ADVANTAGES OF USING FLEXIBLE ELECTRIC GRIPPERS FOR INDUSTRIAL ROBOTIC APPLICATIONS

Robotiq 2-Finger Adaptive Gripper

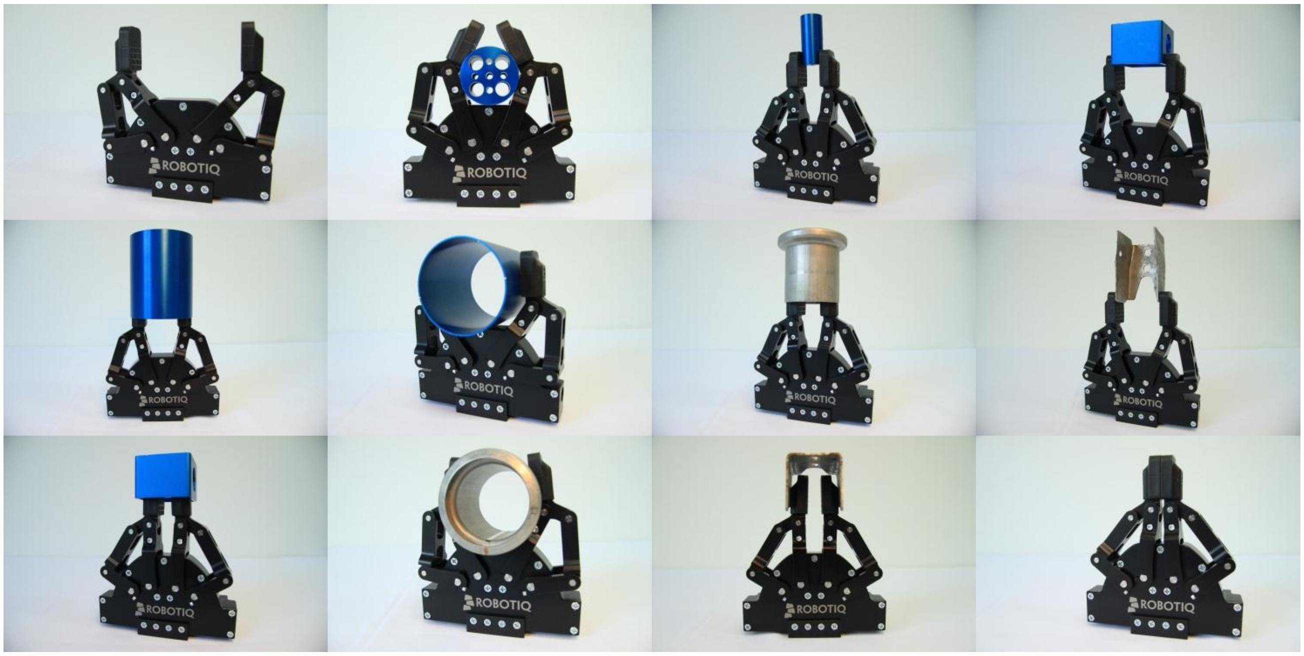


TABLE OF CONTENT

INTRODUCTION

Today's manufacturers are under relentless pressure to reduce costs, increase agility and automate more processes, including the high-mix, labor intensive ones. However, existing technologies such as tool changers provide only limited and expensive solutions when it comes to handling different parts. Changeover costs, custom gripper designs and tooling costs often represent issues which can stop investment in the automation of a process. The return on investment is crucial when automating high-mix part types of applications and it is often hard to achieve. That is why 90% of US manufacturers still don't use robots in their production processes.

This is why Robotiq has designed the Adaptive Robot Gripper to provide manufacturers with maximum flexibility in robotic parts handling.

2-Finger Gripper Main Features

In keeping with the idea of giving manufacturers the flexibility needed to automate high-mix processes, Robotiq designed the 2-Finger Adaptive Gripper to provide unmatched versatility in handling a broad range of parts.

Fully programmable, the robotic Gripper provides:

- Three distinct gripping modes – parallel, encompassing and inside – enable handling of different part geometries including flat, square, cylindrical and irregular.
- High payload to weight ratio coupled with long stroke enable handling of a variety of sizes in a compact form factor.
- Precise speed and force controls enable handling of parts of different rigidities, ranging from brittle, to deformable, to stiff.
- Accurate finger control enables fast cycle times through partial open/closing.

Taken together, these capabilities make the 2-Finger Adaptive Gripper a perfect fit for high-mix processes such as light assembly, sorting and kitting, packaging, machine tending, bin picking and parts transfer

Visit the product page of our website to get more technical information and see videos on our product's main features.

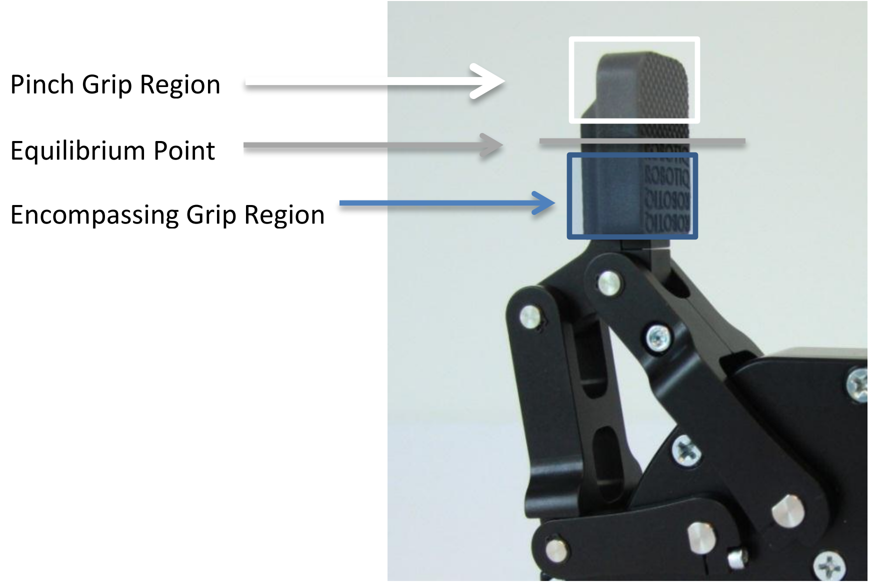
Section 1

THE EASY GRIPPING INTELLIGENCE

When watching how the 2-Finger Adaptive Robot Gripper works you might ask yourself; how this Gripper can be so versatile while using only one actuator? The secret lies in its unique mechanical architecture.

Encompassing Grip vs Pinch Grip

The mechanism driving the fingers of this Gripper is optimized to obtain two distinct contact regions. The first one, called the “encompassing grip region”, is located at the base of the fingers, while the second one, called the “pinch grip region”, is located at the end/tip of the finger. The boundary between these two adjacent regions is called the “equilibrium point”.



When the contact of the finger with the object to be grasped occurs in the encompassing grip region, the finger automatically adapts to the shape of the object and curls around it. On the other hand, when the contact is made in the pinch grip region, the finger maintains its parallel motion and the object is pinched. This comportment may be observed in the following video, where the equilibrium point is indicated by an arrow sticker.

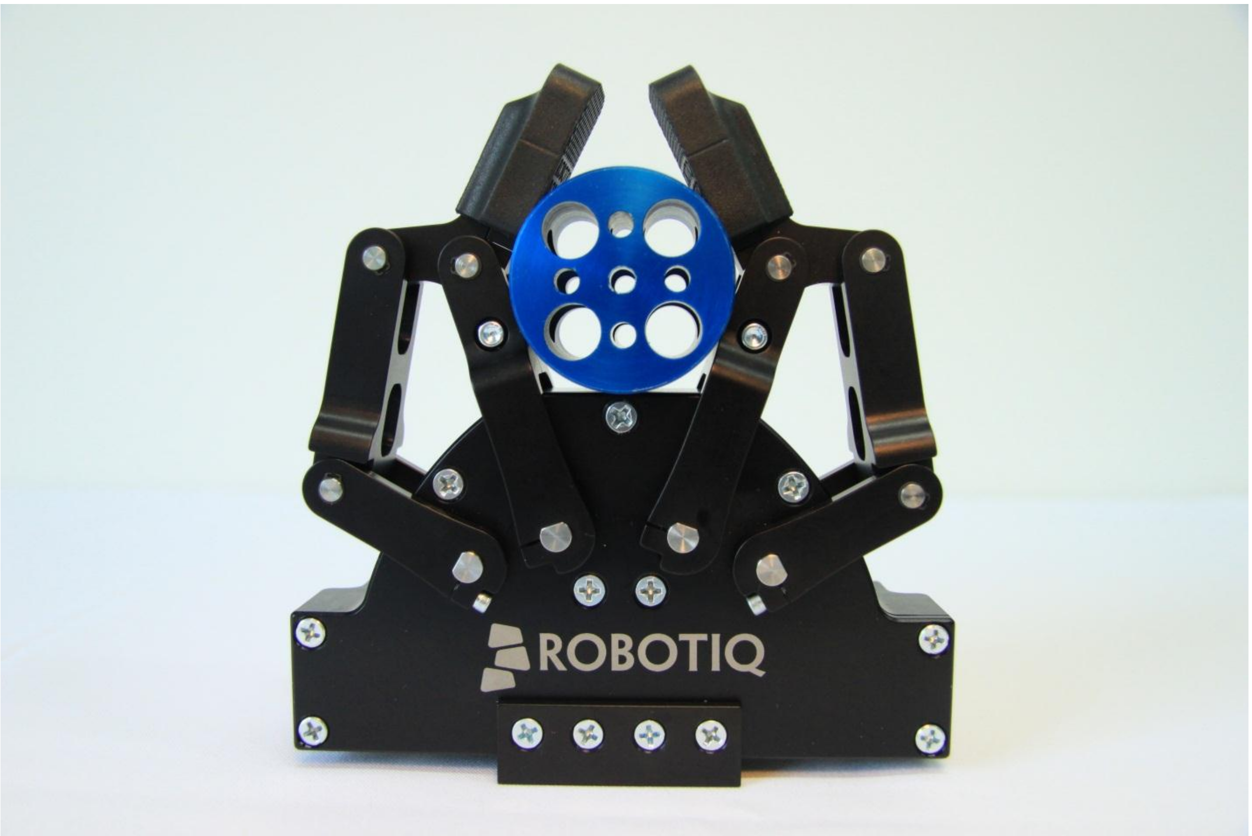
The finger keeps its parallel motion when a contact is made above the equilibrium point during a pinch grip, the same is true for a contact made below the equilibrium point during an inside grip, i.e. for a force applied at the back of the finger. This unique feature allows the Gripper to pick up objects from the inside, which proves to be very useful in many situations.

Coupling between the fingers

In addition to the mechanism used inside each of its fingers, the Gripper also relies on a special coupling architecture between the fingers. In fact, it is mechanically designed to ensure that the two fingers move in conjunction with each other in order to center the object grasped in the middle of the Gripper. This self-centering avoids the need to use expensive sensors and is above all safer.

In the same vein to make this Robot Gripper as reliable as possible, a self-locking feature has been incorporated into it between the actuator and the fingers. By doing so, we are sure that the Gripper will never release the object and let it fall if the power is shut down. It is also economically interesting, as the actuator doesn't need to apply torque continually when an object is grasped, thus in addition to the power saved, the lifespan of the Gripper is thereby maximized.

Finally, it is important to note that the Gripper may be manually opened when the power is shut down, even if the Gripper is rigidly grasping an object.



Section 2

GRIPPER COMMUNICATION BASICS

As explained in the previous section, the 2-Finger Adaptive Robot Gripper is able to achieve both pinching and encompassing grips by automatically conforming to the shape of the object. The Gripper is therefore simply programmed using straightforward open/close commands (everything else is taken care of by the Gripper controller and the finger mechanism). We will now take a look at how the Gripper can be controlled: how to communicate with it, how to perform simple actions, how to adapt its behavior and how to test different commands.

How to Connect to the Robot Gripper?

The 2-Finger Adaptive Robot Gripper has a shared memory which can be accessed by its master (the robot). Some registers can only be read, for example to obtain the status of the Gripper, whereas other registers will also accept written commands. The read/write operations depend on the communication protocol of the Gripper which is chosen to be compatible with the robot. At the moment, the 2-Finger Adaptive Robot Gripper is compatible with six communication protocols, which makes it one of the most versatile products on the market. The current protocols are: EtherNet/IP, TCP/IP, DeviceNet, CANopen, EtherCAT and Modbus RTU.

How to Send Simple Commands?

The first thing to do when the Gripper is powered up is to initialize it. The initialization procedure consists of fully opening the Gripper and is required to set the "zero" position of the Gripper (or its "home" position, if you prefer). When the Gripper is initialized, it will wait for its "Go to Position Request" bit to be set before moving towards its requested position. The position request is an integer in the range: 0 to 255 and is controlled by writing the appropriate number in one of the Gripper's registers (the third one, actually). The position 0 corresponds to the fully opened position and 255 to the fully closed position (there are software limits which prevent the Gripper from reaching these theoretical limits). Once the "go to" flag is set, the Gripper can be fully closed by writing 255 in the position request register (and fully opened by writing 0).

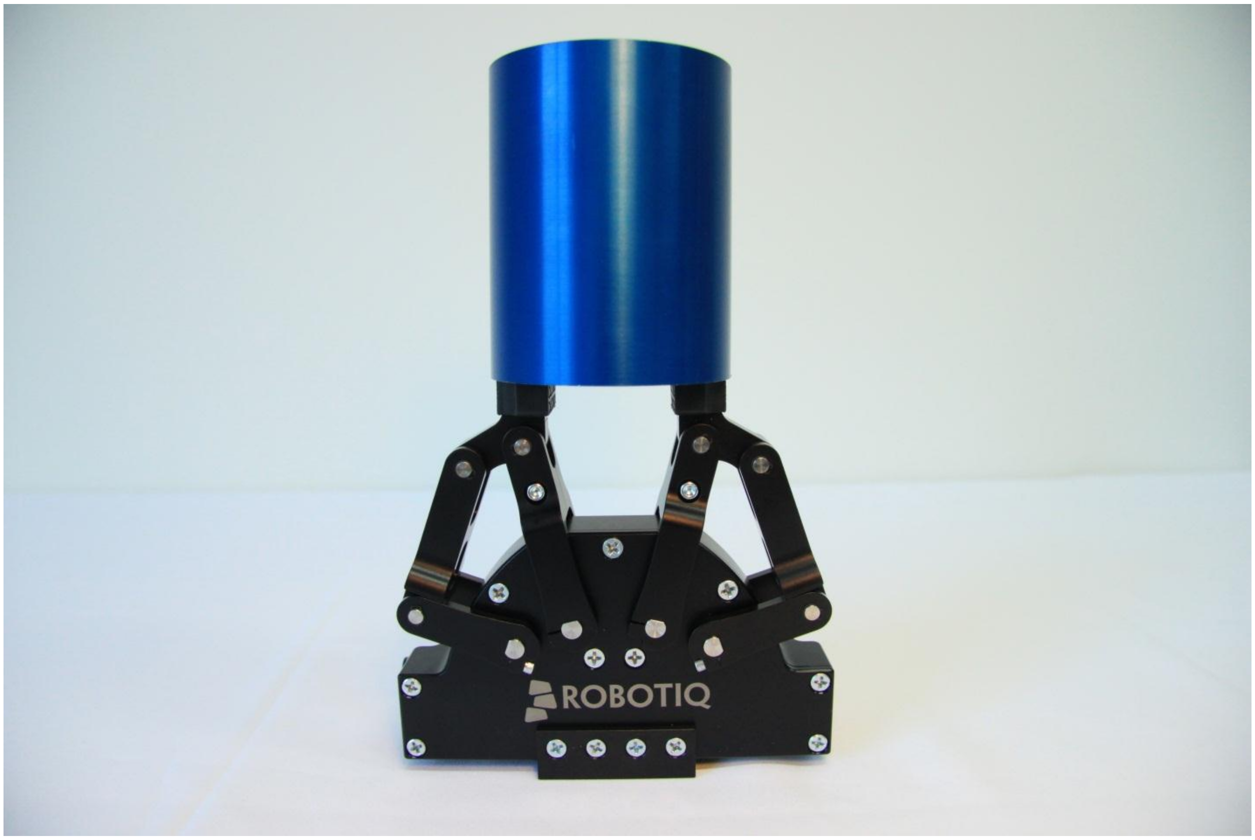
How to Change the Gripper Control Parameters?

As noted above, we can establish a position request from 0 to 255 to make the Gripper fingers move towards this reference. There are also two parameters that can be adjusted similarly (with values from 0 to 255): speed and force. The Gripper motor is primarily controlled using a speed reference. Therefore, adjusting the speed parameter has a lot of influence on the Gripper's behavior. The value 0 corresponds to the minimum speed (not zero speed!) whereas the value 255 corresponds to the maximum speed of the fingers. The force parameter modifies the maximum current that is sent to the motor prior to reaching the stop position. It is worth noting that the Gripper has an auto-locking mechanism which prevents the fingers from moving when external forces are applied on them. Therefore, the maximum current sent to the motor during the grip determines the gripping force. The value 255 sets the motor current limit to its maximum value whereas the value 0 sets the Gripper to the lowest value to which the Gripper can properly move.

How to Test the Commands?

When the Gripper is received by the end-user, it is often useful to test its movement potential before placing it on the robot. This allows the programmer to experiment with the effect of the control parameters, to establish the best way to grip the part(s) for the targeted application or just to become comfortable with the Gripper before programming it for real. This is possible with our user interface which is distributed with the Gripper. The user interface is compatible with Windows XP and Windows 7. The user interface allows you to modify all the control parameters and to read all the status registers.

Well, this explains in a nutshell how to control the 2-Finger Adaptive Robot Gripper. If you need more information, you can take a look at the control section of the user manual or, of course, you can contact us at info@robotiq.com.



Section 3

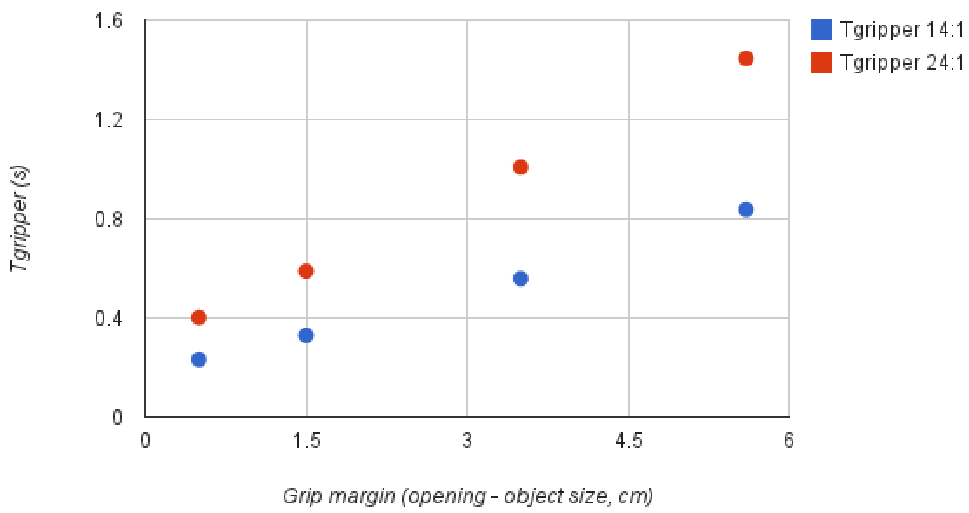
CREATING FASTER TIME CYCLE WITH ELECTRIC GRIPPER

One of the advantages of a servo-electric robot gripper is that it allows you to control the position of the fingers. If the size of the object you want to pick up is known, the fingers can be partially closed before the robot has reached the pick-up location. This, in turn, will reduce the time required to pick up the object and therefore the cycle time will be reduced. As we all know, reducing the cycle time is highly desirable as it allows us to increase the production rate of a robot. In this section, we will present how the cycle time can be reduced by programming the partial opening of the 2-Finger Robot Gripper.

How to Calculate the Robot Gripper Cycle Time?

If we consider a pick and place application, the cycle time can be split into two components: the time required by the robot to move during the whole cycle (Trobot) and the time required by the Gripper to pick up the object and release it (the Gripper cycle time, Tgripper). Let's focus on the Gripper cycle time as the robot component will vary too much depending on the application.

The Gripper cycle time can in turn be split into three components: the time required to grip the object (Tgrip), the time required to release it (Trelease) and the idle time, which comprises all the time required for the Gripper to update its status to the master and for the master to send new commands to the Gripper, plus the time lost during the acceleration and deceleration of the fingers (Tidle). The idle time is tricky as it depends on the communication protocol and on the master and will vary slightly from one cycle to the next. Therefore, the Gripper cycle time can be split into a fixed delay (Tidle) and two delays which vary according to the grip margin (Tgrip, Trelease). The grip margin is the opening of the fingers minus the object size and is required to avoid interference between the Gripper and the object during the approach trajectory. To measure the Gripper cycle time, we ran some tests using the TCP/IP communication protocol on a standard PC. We ran the tests for both gearbox options (24:1 and 14:1) of the Gripper, using the full speed setting. We found that the cycle time varies proportionally to the grip margin, as shown on the graph below:



On the graph, it can clearly be seen that minimizing the Grip margin will have an enormous effect on the robot gripper cycle time, and as a consequence on the total cycle time. The cycle time can more than double if no attention is paid to the partial opening position!

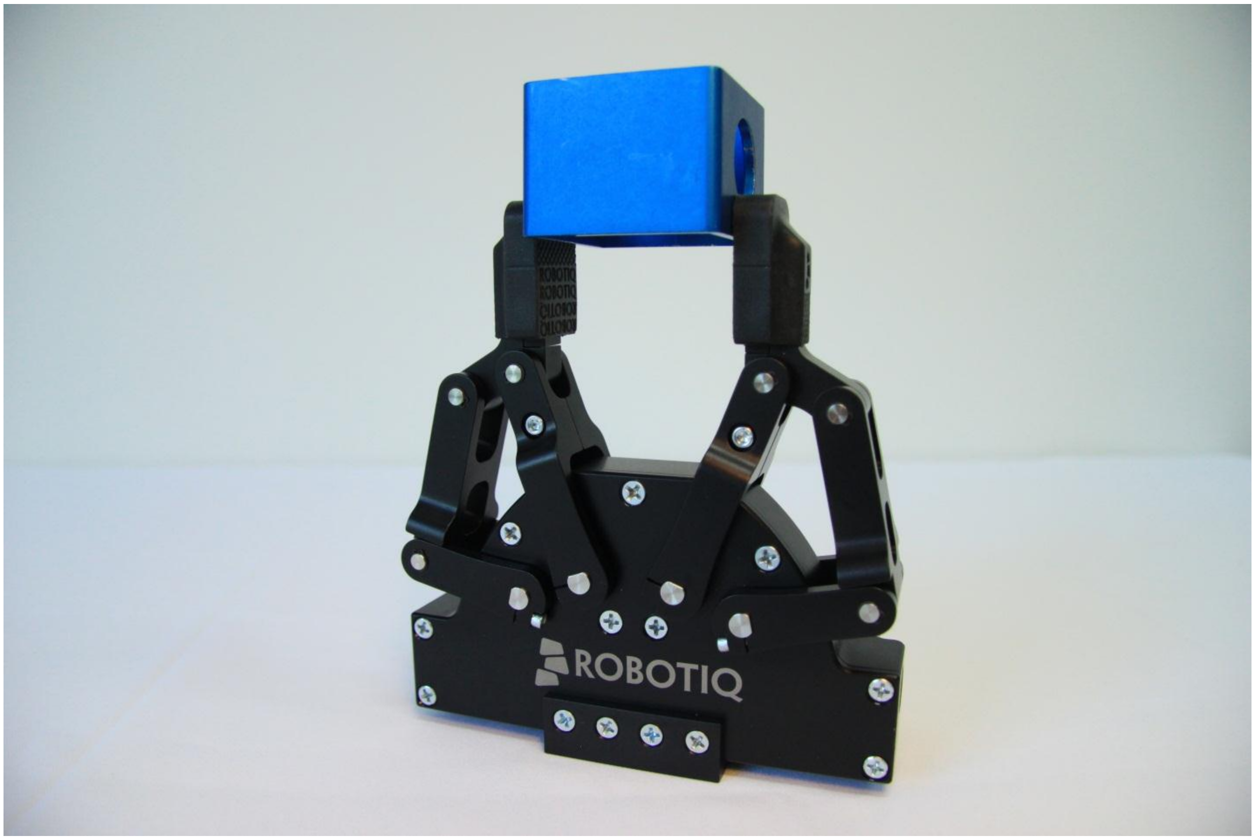
For these tests, Tidle was found to be 0.16 seconds for the 14:1 gearbox option and 0.31 seconds for the 24:1 gearbox option. Again, these values will vary depending on the master and the communication protocol of the Gripper. Tgrip and Trelease can be approximated by dividing the grip margin by the finger speed. The Gripper cycle time is therefore computed as:

Tgripper = Tidle + 2 x grip margin / finger speed.

How to Choose the Grip Margin?

In the last section, I showed the effect of the grip margin (finger opening before the grip minus the object size) on the Gripper cycle time. Clearly, this distance has to be minimized in order to get the best production speed out of your robot. How small of a margin can we use? Well, it depends. On one hand, the Gripper repeatability is good enough to lower the margin to a very small value (under 1 mm). However, a small grip margin will require the robot to be very precise in its approach trajectory to avoid hitting the object. This may require the reduction of the robot speed, which will in turn increase the cycle time. Therefore, a trade-off has to be achieved which will depend on the application (robot, object shape variability, object positioning, etc.). Generally, it is better to use a slightly larger grip margin to avoid hitting the object in all possible situations (reliability is often more important than cycle time!). As a result, the choice of the grip margin is not always easy to optimize.

To conclude, let's just say that being able to set the partial opening position is a key feature in the optimization of the cycle time. This is a clear advantage of servo-electric grippers such as the 2-finger Adaptive Robot Gripper!



Section 4

USING FEEDBACK OF 2-FINGER ADAPTIVE GRIPPER

Feedback might be very useful in industrial applications. Here are few types of feedback given by our robot Gripper and how they could be used.

Encoder Value

The 2-Finger Adaptive Gripper has a feature which is called partial opening. This feature has two input parameters: opening limit and closing limit. In order to optimize cycle time, this feature can be set to the object's dimensions through the encoder value, which is ranged from 0 to 255. By doing so, the Gripper will respect these opening and closing limits, thus reducing cycle time by not having to fully open or close. Speed and force must still be set accordingly to your application.

The encoder value can also be used as a qualitative way to measure the same object or to verify whether the Robot Gripper has picked a part in the desired way.

Motor Current

The motor current can be used to monitor the energy consumed in a complete cycle. This might be useful for integrators in the design of a robot cell. By knowing the motor current, power supplies and their related protection devices can be appropriately dimensioned.

The motor current value, range is from 0 to 255, can be used in a qualitative way to know what final force will be applied by the fingers to the object. The 2-Finger Robot Gripper also has an auto-locking mechanism. This means that the motor doesn't need to apply force continuously on the object to maintain the designated final force. Care must be taken when using the motor current in this way. Tests must be performed with different force settings to be sure the Gripper doesn't damage any fragile parts.

Object Detection and Error Messages

The 2-Finger Gripper is able to detect whether an object has been picked up or not. This feedback can be very helpful when used in conjunction with a teach pendant program. In fact, recognition of object detection can be used as a conditional statement in a routine.

The 2-Finger Gripper gives error messages. Based on the seriousness of these messages, the robot may be programmed in such a way that it behaves safely when Gripper errors occur. See the list of error messages in the user manual to integrate them properly into your robot program.

Now that you have some ideas about how the Gripper feedback can be used with your robot, go ahead an experiment till you find the settings that work best for your application!

Section 5

HOW TO VERIFY GRIPPER CONTACT

Both of our Adaptive Grippers are able to grip objects of various shapes using their innovative finger mechanism. This allows for firm grips on various objects without the need to build custom grippers for each application. In addition to this adaptability, the Gripper is able to determine when it has gripped an object. This is useful to determine if the pick-up procedure was performed correctly and if the robot can move to the next step of its program.

In some cases, however, it is also useful to verify if the Gripper is still holding the object after a motion is executed by the robot. For example, if the object position was not initially determined accurately (for instance, if the object has moved or if the vision system had a problem), it is possible that the object could be picked up in an awkward position. In this situation, the object might slip out of the Gripper if the robot moves very rapidly to its next position. Knowing that the object was dropped is crucial for many applications. So below, we will explain the appropriate procedure to verify if the Adaptive Gripper still has the object after a robot motion.

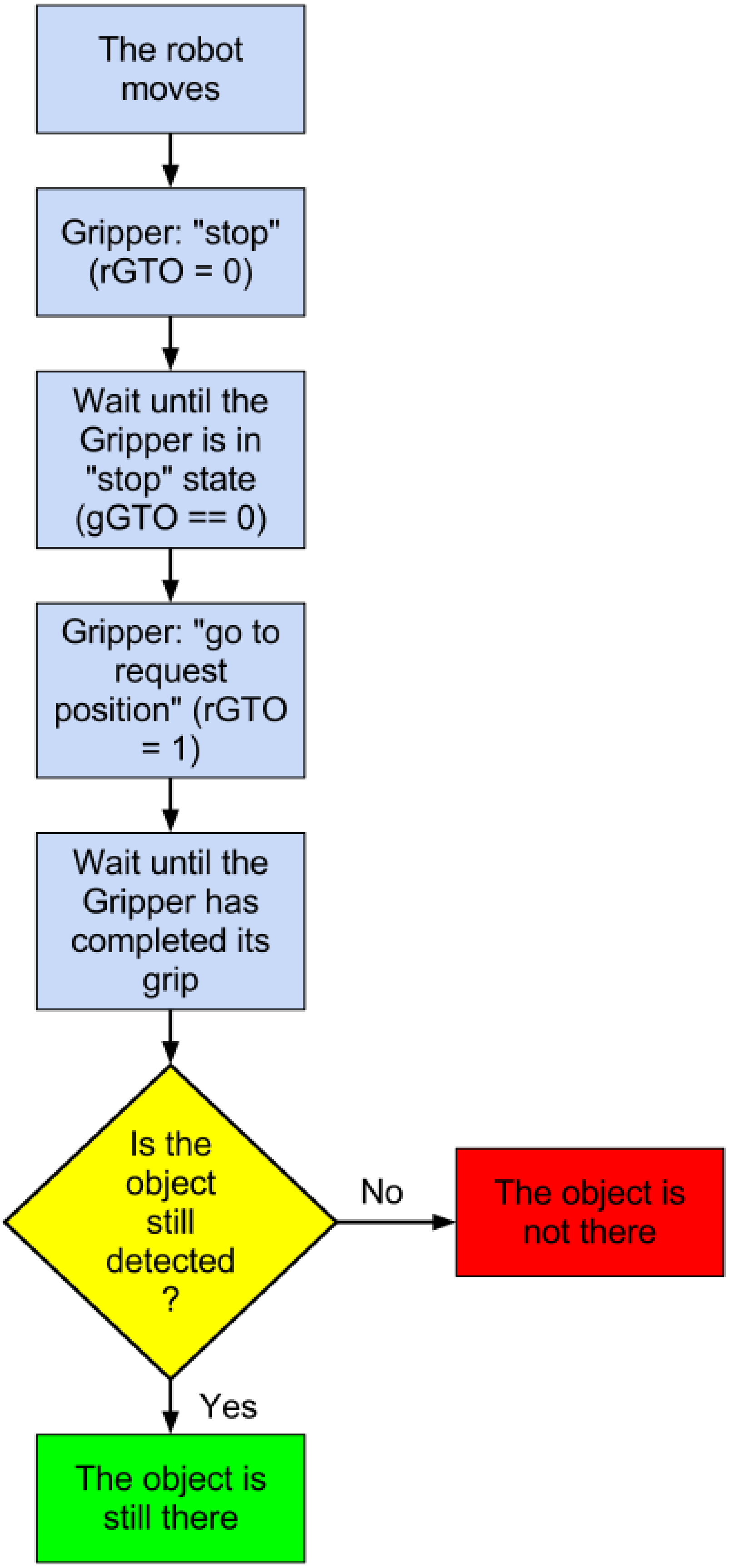
How the object is detected during the grip

When a command is sent to close the fingers of the Gripper, the motor moves towards a target position. If the motion is stopped because the Gripper has found an object, the force applied by the fingers will increase until the current sent to the motor exceeds its limit (which is fixed using the force parameter). At that moment, the motor will stop moving and the grip force will be maintained by the Gripper's auto-locking mechanism. By reading the motor position, the Gripper is able to determine if an object was gripped or if the motion was stopped due to the fingers touching themselves. However, from that moment, the Gripper will consider that the object is gripped and will not detect an object loss unless the procedure explained in the next section is executed.

How to verify if object contact is maintained

Once the object is gripped, no more current is sent to the motor as the grip force is maintained by the Gripper's auto-locking mechanism. In order to verify if the Gripper still has the object, motor motion has to be re-initiated. This is simply done by first sending a "stop" command and then by sending another "go to" command with the same parameters as the initial grip. If the Gripper still has the object, no motion will be recorded (or a small one if the object is compliant) and the object will remain detected. However, if the object is no longer being grasped by the Gripper, the fingers will move and no object will be detected.

So, from a technical standpoint, how is this programmed in the robot? The first thing to know is that the Gripper motion is initiated using a "go to" variable which is named **rGTO**. When **rGTO = 0**, the robot tells the Gripper to stop and when **rGTO = 1** the robot tells the Gripper to move. Also, the Gripper has an output variable **gGTO** which is simply an echo of **rGTO**. This variable is useful to verify if the new command was received and executed. In a more visual manner, the following diagram shows how the robot can verify if the Gripper still has the object after it has moved to the position where the Gripper will release it:


The robot moves
Gripper: "stop" (rGTO = 0)
Wait until the Gripper is in "stop" state (gGTO == 0)
Gripper: "go to request position" (rGTO = 1)
Wait until the Gripper has completed its grip
Is the object still detected ?
No
The object is not there
Yes
The object is still there

This procedure is useful in handling problems that may arise in the robot cell if, for example, the position of the object can change between the time the robot localizes the object and the time the Gripper initiates its grip.

However, in many applications, the parameters are defined so precisely that this procedure might not be necessary. Anyways, as engineers we generally aspire to find the best scenario while planning for the worst. In this case, double checking everything may be a good idea!

ABOUT ROBOTIQ

Robotiq designs and manufactures flexible robot grippers. We aim to give to industrial manufacturing – from large businesses to SMEs – flexible peripheral solutions to optimize automated processes by providing universal, agile and rugged robotic tools.

We work with robot manufacturers, system integrators and end-users to enable new applications and improve productivity.

We are Robotiq. We make tools for agile automation.

TO LEARN MORE

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.

Phone (USA and Canada): 1 888 762-6847 extension 122
Phone (Anywhere in the world): 1 418 380-2788 extension 122

Email: info@robotiq.com

Website: www.robotiq.com

Let's keep in touch via social media

ROBOTIQ'S BLOG

LEARN ABOUT ROBOTICS INDUSTRY NEWS, APPLICATIONS AND TRENDS

GET THE LATEST NEWS ABOUT ROBOTIQ

WATCH VIDEOS OF OUR PRODUCTS

GET THE LATEST NEWS ABOUT ROBOTIQ