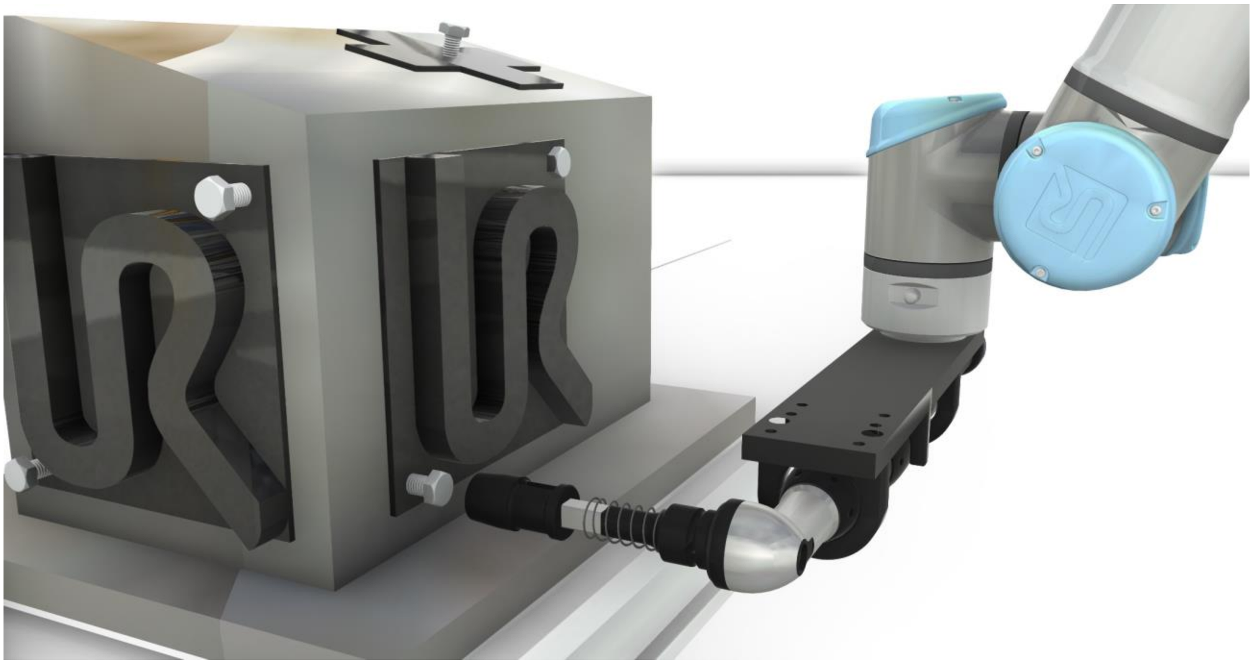


Application manual

High torque screwdriving

Version 1.1

Table of contents

Maximum torque

The maximum tightening torque from an automatic screwdriver that a robot can handle, without triggering a protective stop or a force limitation error, depends on several factors such as:

- Tool mounting
- Robot programming
- Safety settings
- Robot poses
- Payload
- Others not directly related with the robot, e.g., the tightening setup in the controller of the screwdriver that determines the strategy, torque, speed, angle, etc.

The joint positions and payload determine how much torque from the joints is required to maintain the robot position, and how much is left available from its maximum values. The tool positions and force limits can restrict the torque that is allowed from what was left available. The tool mounting changes the force that the robot receives from the process and which requires to oppose to. If the robot is steady (standing still with zero target velocity) the torque restrictions can be less tight and the allowed joint position deviations before triggering a protective stop larger, which would be more favorable to the robot in the process.

Taking all this into account, the screwdriving torque withstanding capabilities for a UR10/UR10e, the most frequent robot in high torque screwdriving applications (> 30 Nm), can be divided into:

- < 30 Nm – basic application, not considered high torque
- 30 – 60 Nm – advanced, requires consideration of some of the above factors
- 60 – 100 Nm – expert, requires consideration of all the above factors
- > 100 Nm – not extensively researched, most likely requires innovative technologies or additional hardware

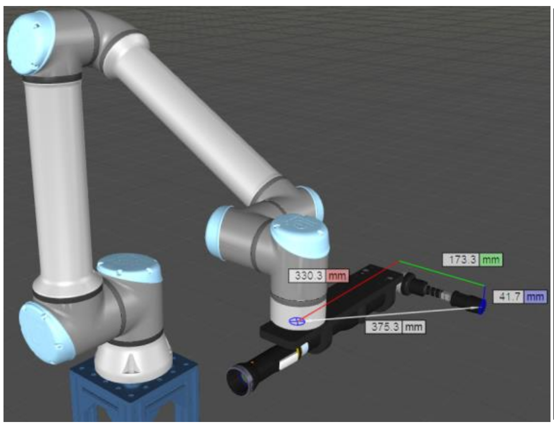
Tool mounting

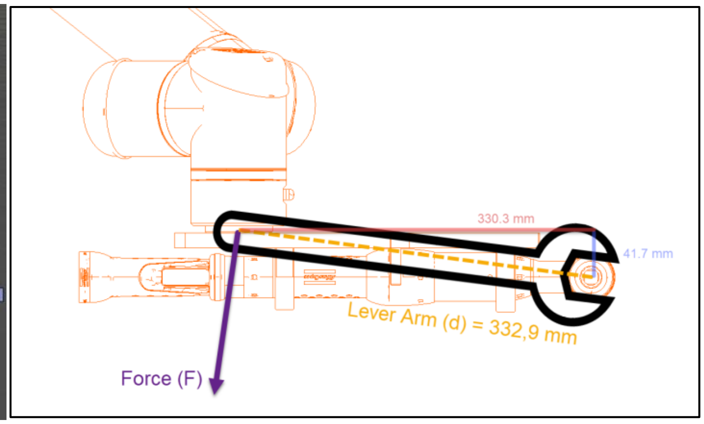
The automatic screwdriver or nut runner should be mounted to the tool output flange of the robot in such way that: the lever arm is maximized, thus the force required from the robot minimized in order to maintain its position during the process; and the possible deviations in a single joint are minimized, avoiding, for example, that the tightening axis is aligned, parallel or close to parallel with the wrist 3 of the robot.

Lever arm

The lever arm is defined as the perpendicular distance from the axis of rotation to the line of action of the force.

Moment = Force x Distance or M = (F)(d)





The force has maximum effectiveness in producing (or opposing to) torque if it is exerted perpendicular to the wrench. This drawing is for illustrative purposes only, since the robot doesn't need to apply force to produce torque with a wrench; the torque is produced by the automatic screwdriver or nut runner, and the robot needs to oppose to the force that is transferred, called torque reaction.

It is important to understand that the lever arm shall be measured in the plane that is perpendicular to the tightening axis. In the example above, the distance of 173.3 mm in the direction of the tightening axis is not considered for the lever arm calculation.

How much force is required at the tool output flange of the robot, can be calculated with the target torque (moment) and lever arm, using the equation: Moment = Force x Distance. For example:

Target Torque: 30 Nm

Lever Arm: 0.3329 m

Force** = Moment / Distance = 30 Nm / 0.3329 m = **90 N

Universal Robots safety system by default limits the force at the tool output flange of the robot at 150 N (125 N considering the tolerance of 25 N), which can be increased up to 250 N. Although these values are not representative of forces that could actually be achieved with the robot, neither necessarily limiting (but part of what needs to be considered for) possible higher torque screwdriving values, they can give an idea of the effort and the relationship with the target torque and the lever arm.

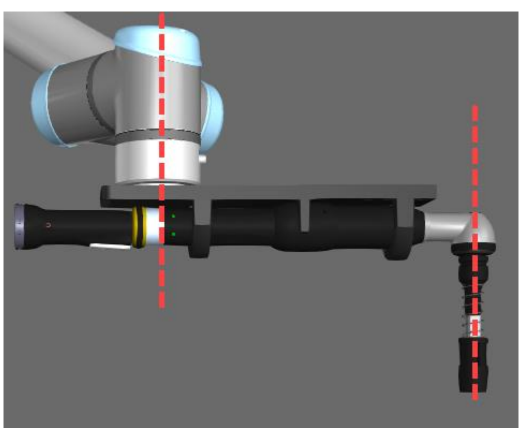
Torque [Nm]	Lever Arm [m]	Force [N]
7.5	0.1	75
15	0.2	75
22.5	0.3	75
30	0.4	75
12.5	0.1	125
25	0.2	125
37.5	0.3	125
50	0.4	125
22.5	0.1	225
45	0.2	225
67.5	0.3	225
90	0.4	225

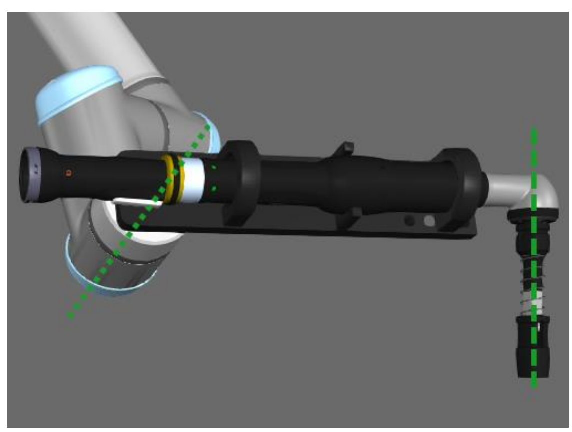
While increasing the lever arm reduces the torque reaction (transferred force to the robot), it should be considered that it may reduce too the capacity of the robot to apply force against the screw.

Joint deviations

Position deviations should be avoided during the tightening process, which may occur when some of the joints reach their allowed torques. Joint deviations can trigger protective stops, and having a single joint aligned, parallel or close to parallel with the tightening axis, can favor deviations in that joint.

While the angles between most of the joint axes and the tightening axis change with the robot motion, that can be reoriented too at a screw position to find a suitable position, the angle between Wrist 3 and the tightening axis remains fixed by the way the tool is mounted to the output flange. To minimize possible deviations at this joint, avoid the alignment or parallelism between the directions of the Wrist 3 and screwdriving - at least an angle of 30 degrees is recommended.





Robot programming

The robot controller allows the joints to use more current/torque if needed to maintain the arm in position, and accepts larger joint deviations before triggering a protective stop, when the robot has been standing still with zero target velocity for 500 ms (aka "steady mode") or when using the Screwdriving functionality.

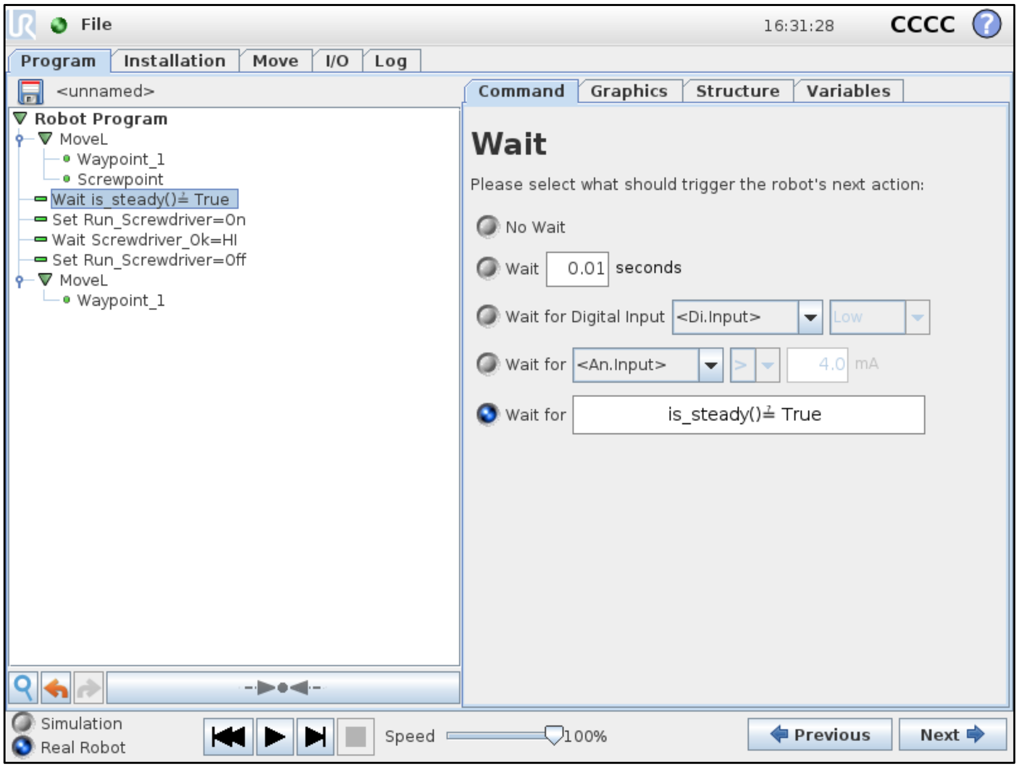
Steady mode

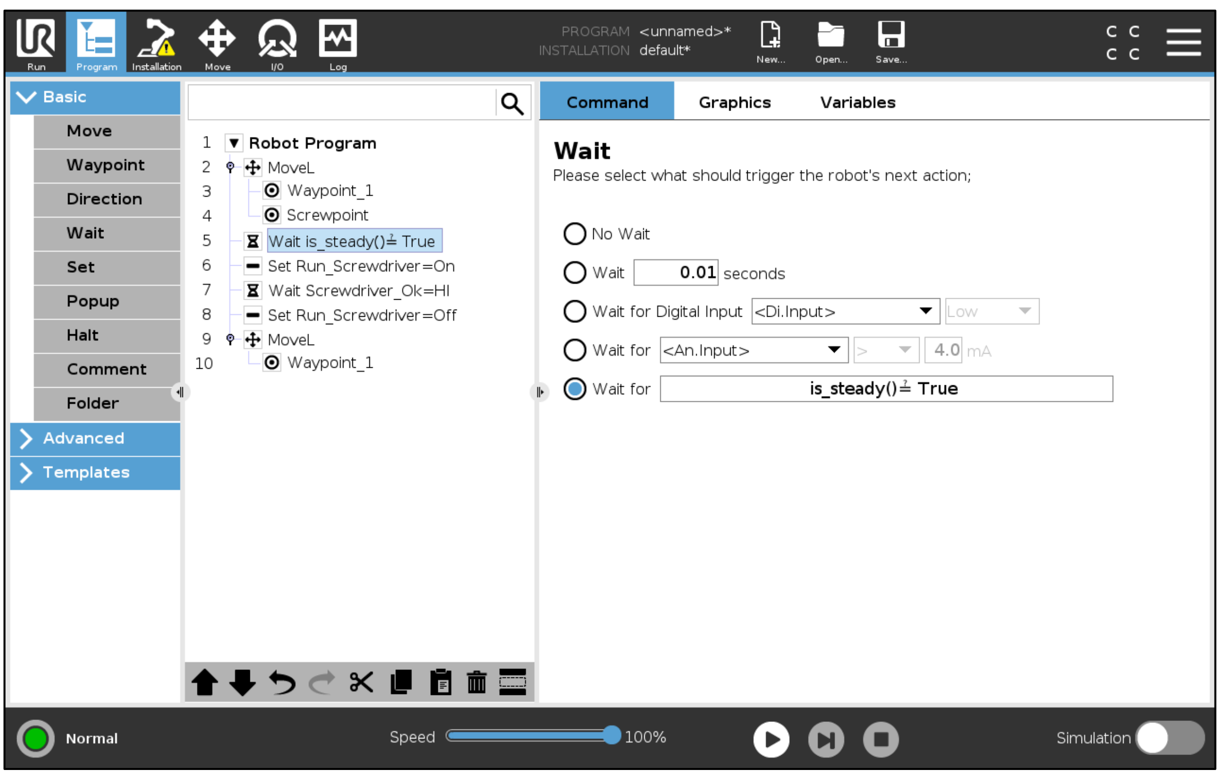
This behavior was introduced from SW version 3.2.20175 in CB3 and has been available since then, including all SW versions for e-Series.

When the robot has been standing still with zero target velocity from 500 ms it automatically goes into steady mode, a state where more torque is allowed and larger deviations are accepted, as described before. The URScript function *is_steady()* returns *True* when the robot is in steady mode, *False* otherwise. There is no need to call *is_steady()* to transition into steady mode, the robot just always goes into this state after 0.5 s standing still with zero target velocity, and is_steady() can simply be used to look into this internal state without adding any delay.

It is recommended to use the script *is_steady()* in a **Wait** command before running the screwdriver, to confirm that the robot is in steady mode before starting the tightening process. To robot should remain in steady mode until the screwdriver ends the tightening of the screw, which means that no motion command should be executed meanwhile – should this happen, the robot would simply stop being in steady mode and execute that command, but it would not be able to benefit from the extra torque and larger deviation, if needed at the end of the tightening process when the torque reaction happens.

Since the robot is meant to be standing still in steady mode, it can't be used to push or follow the screw going into the thread. If steady mode is required, use a spring or other mechanical compensation for the length of the screw, so the robot can be fully at rest during the process.


File
16:31:28
CCCC
Program
Installation
Move
I/O
Log
<unnamed>
Command
Graphics
Structure
Variables
Robot Program
MoveL
Waypoint_1
Screwpoint
Wait is_steady()≟ True
Set Run_Screwdriver=On
Wait Screwdriver_Ok=HI
Set Run_Screwdriver=Off
MoveL
Waypoint_1
Wait
Please select what should trigger the robot's next action:
No Wait
Wait 0.01 seconds
Wait for Digital Input <Di.Input> Low
Wait for <An.Input> > 4.0 mA
Wait for is_steady()≟ True
Simulation
Real Robot
Speed
100%
Previous
Next


Run
Program
Installation
Move
I/O
Log
PROGRAM <unnamed>*
INSTALLATION default*
New...
Open...
Save...
C C
C C
Basic
Move
Waypoint
Direction
Wait
Set
Popup
Halt
Comment
Folder
Advanced
Templates
1 Robot Program
2 MoveL
3 Waypoint_1
4 Screwpoint
5 Wait is_steady()≟ True
6 Set Run_Screwdriver=On
7 Wait Screwdriver_Ok=HI
8 Set Run_Screwdriver=Off
9 MoveL
10 Waypoint_1
Command
Graphics
Variables
Wait
Please select what should trigger the robot's next action;
No Wait
Wait 0.01 seconds
Wait for Digital Input <Di.Input> Low
Wait for <An.Input> > 4.0 mA
Wait for is_steady()≟ True
Normal
Speed
100%
Simulation

Screwdriving functionality with fast-steady mode

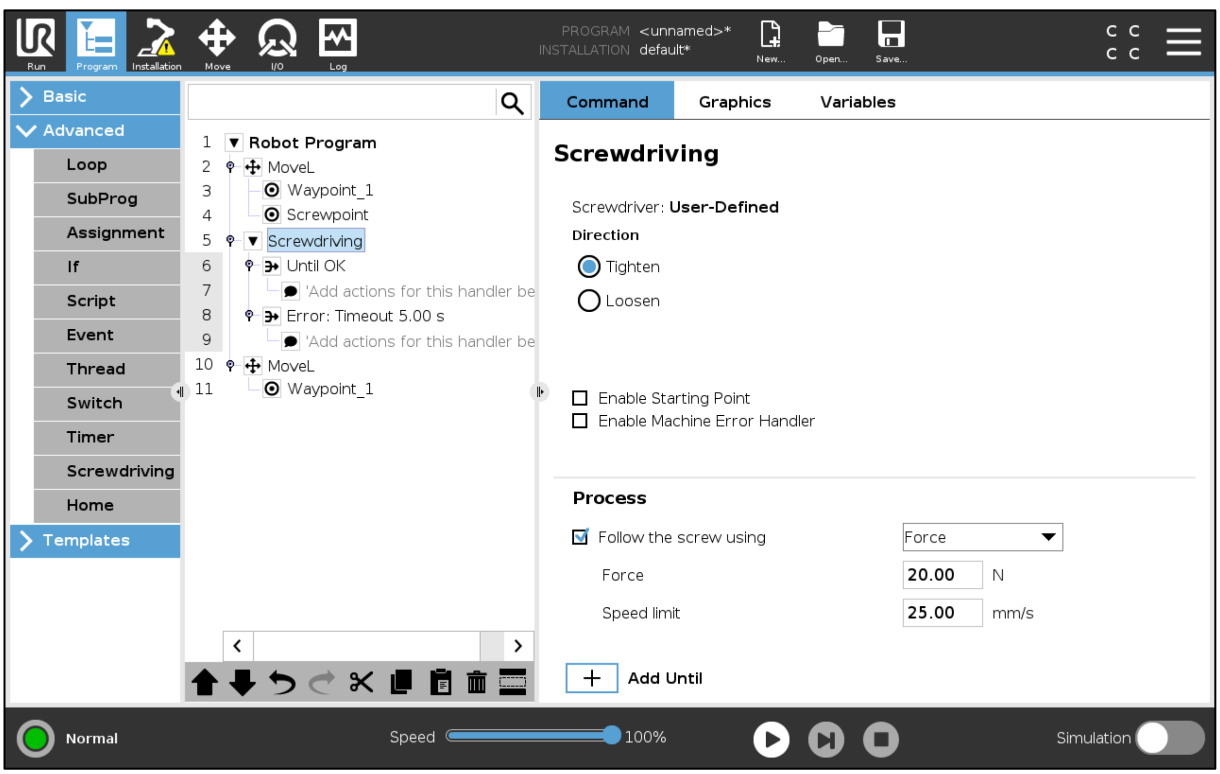
The Screwdriving functionality was introduced from SW version 5.4.0 in e-Series and has been available since then. This functionality is not available in CB3.

The Screwdriving functionality consists of an advanced command and an installation setup in PolyScope, and a URScript function *screw_driving(f, v_limit)*.

While the Screwdriving functionality makes it easier to setup a screwdriving application in general by enabling TCP selection, I/O exchange, direction selection, starting point, several strategies to follow the screw, and multiple success and errors conditions to end the process; one of its main features is the ability to instantaneously change into steady mode when the torque reaction is detected (aka "fast-steady mode").

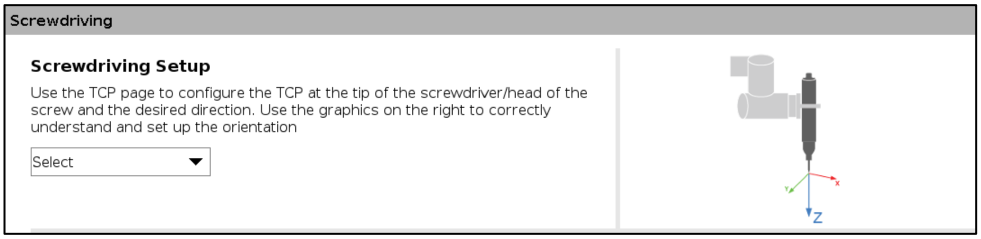
When executing a Screwdriving command that uses controlled force and speed to push and follow the screw while it goes in the thread, and the torque reaction from the automatic screwdriver is detected, the robot automatically goes into fast-steady mode, a state where more torque is allowed and larger deviations are accepted, as described before.

The Screwdriving command with the fast-steady mode behavior enables that the robot can still move during the tightening process, pushing and following the screw going into the thread, while it also ensures that when the torque reaction is detected, this motion will be immediately stopped and the robot will be standing still with the maximum allowed current to maintain its position, and ready to accept larger joint deviations, should they occur, before triggering a protective stop.



The fast-steady mode behavior is only enabled when the checkbox “Follow the screw using” is selected with the “Force” option from its dropdown, in the process section of the Screwdriving command; or by using directly the URScript function *screw_driving(f, v_limit)*.

The Tool Center Position should be defined as instructed in the Screwdriving installation setup in order to successfully detect, through the Force and Torque Sensor readings and internal calculations, the torque reaction that triggers the fast-steady mode. It should be observed that the positive Z direction of the TCP Orientation must be aligned to the length of the screws to be tightened.



Since the torque reaction coming from the screwdriver is detected using the Force and Torque Sensor from the robot, its zeroing should be carefully considered. The Force and Torque Sensor zero must be done without the screwdriver pushing against the screw, neither under the influence of any external forces.

When using the Screwdriving command from PolyScope, a zero of the F/T Sensor is automatically done with the start of the execution of each of these. Just before running a Screwdriving command, the screwdriver must be in the right position to start the tightening process, but without any contact with the screw, workpiece or any other thing that could bias the F/T Sensor readings.

On the other hand, if using directly the URScript function *screw_driving(f, v_limit)*, the *zero_ftsensor()* function should be called before it, under the same conditions described above.

screw_driving(*f, v_limit*)

Enter screw driving mode. The robot will exert a force in the TCP Z-axis direction at limited speed. This allows the robot to follow the screw during tightening/loosening operations.

Parameters

`f`: The amount of force the robot will exert along the TCP Z-axis (Newtons).

`v_limit`: Maximum TCP velocity along the Z axis (m/s).

Notes:

- Zero the F/T sensor without the screw driver pushing against the screw.
- Call `end_screw_driving` when the screw driving operation has completed.

```
>>> def testScrewDriver():
>>>     # Zero F/T sensor
>>>     sleep(0.02)
>>>     zero_ftsensor()
>>>
>>>     # Move the robot to the tightening position
>>>     # (i.e. just before contact with the screw)
>>>     ...
>>>
>>>     # Start following the screw while tightening
>>>     screw_driving(5.0, 0.1)
>>>
>>>     # Wait until screw driver reports OK or NOK
>>>     ...
>>>
>>>     # Exit screw driving mode
>>>     end_screw_driving()
>>> end
```

Safety settings

The safety system limits the torque/current of each joint according to the force limit and tool positions settings, among others. The tool position should be defined at tip of the screwdriver or nut runner, where the tool force limit acts. Having the TCP/tool positions defined properly and the tool force limit value as least restricted as possible according to the risk assessment, will enable the robot arm to handle higher tightening torques from an automatic screwdriver or nut runner.

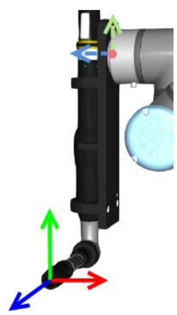
Force limits

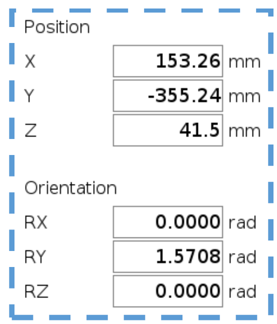
The force limits are the forces exerted by the robot at the TCP/tool positions (Tool Force) and elbow joint (Elbow Force, e-Series only). The safety function continuously calculates the torque allowed for each joint, so that the TCP/tool positions and elbow joint (e-Series only) stay within the force limits.

The force limits should be set as least restricted as possible according to the risk assessment, to allow the joints to use more current/torque if needed.

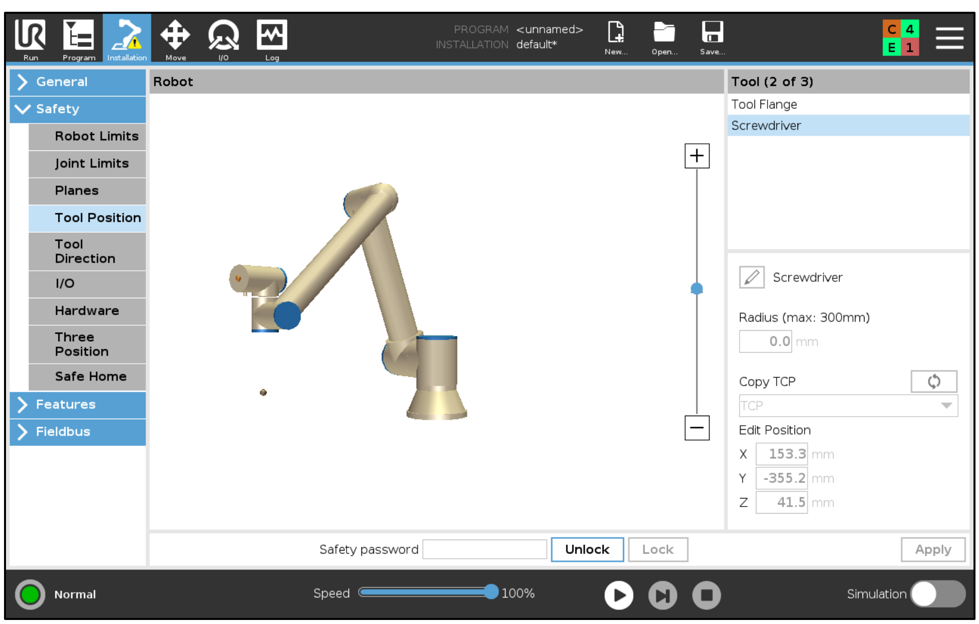
Tool positions

Define the Tool Center Point Position at the tip of the screwdriver or nut runner, and the Orientation to obtain the positive Z direction aligned to the length of the screws to be tightened.





In e-Series the TCP should be copied as a Tool Position in the Safety settings.

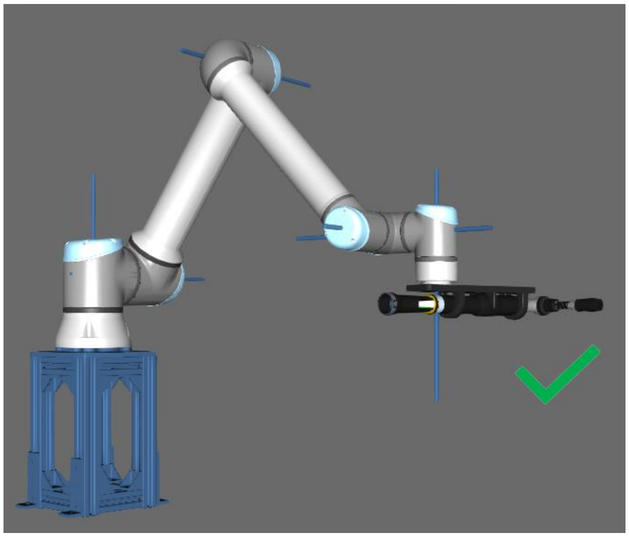


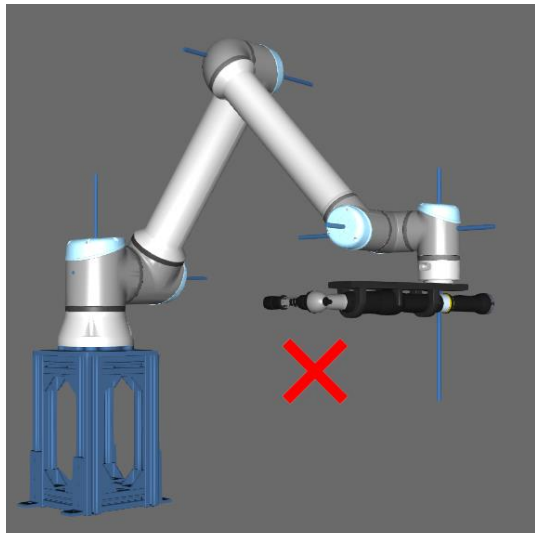
Robot poses

The robot poses for automatic screwdriving should be such that the screwdriving points are as far as possible to the robot joint axes, and none of the joints is loaded close to its maximum torque.

Screwdriving points

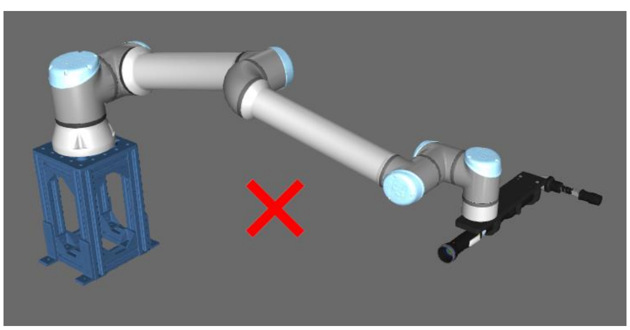
With the TCP/tool position properly defined, the tool force limit restricts at the screwdriving points the allowed torque from each of the joints: the farther the screwdriving points are to the robot joint axes, the bigger the torque allowed – the same concept of the lever arm, but applied to the torques of the joints and the force they may produce at the tool position.

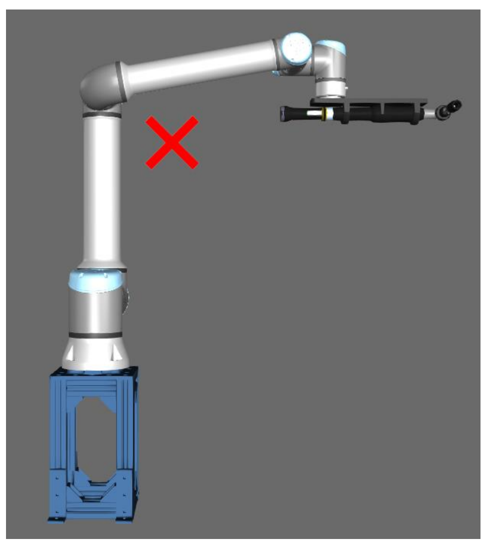




Maximum joint torque values

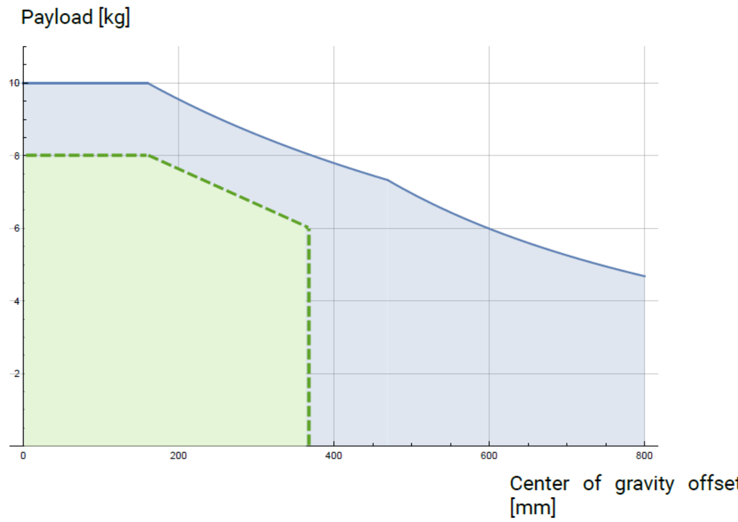
Robot poses with the arm fully extended (tool flange close to the outer workspace limit), or in such way that one or more joints are close to their maximum torque values should be avoided. The idea is to use as much as possible of the available torque in every joint in holding the screwdriving torque, rather than the robot position.





Payload

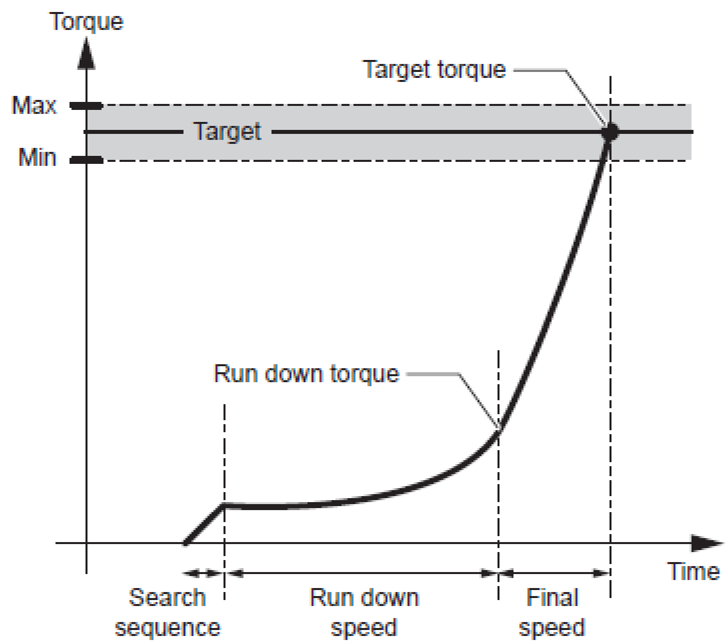
The weight and center of gravity of the payload attached to the tool output flange of the robot should be defined accurately. In high torque screwdriving applications, the weight of the tool mounting should be kept as low as possible and center of gravity should not be too close to the limits of the maximum payload diagram in the user manual, like the example below for UR10.



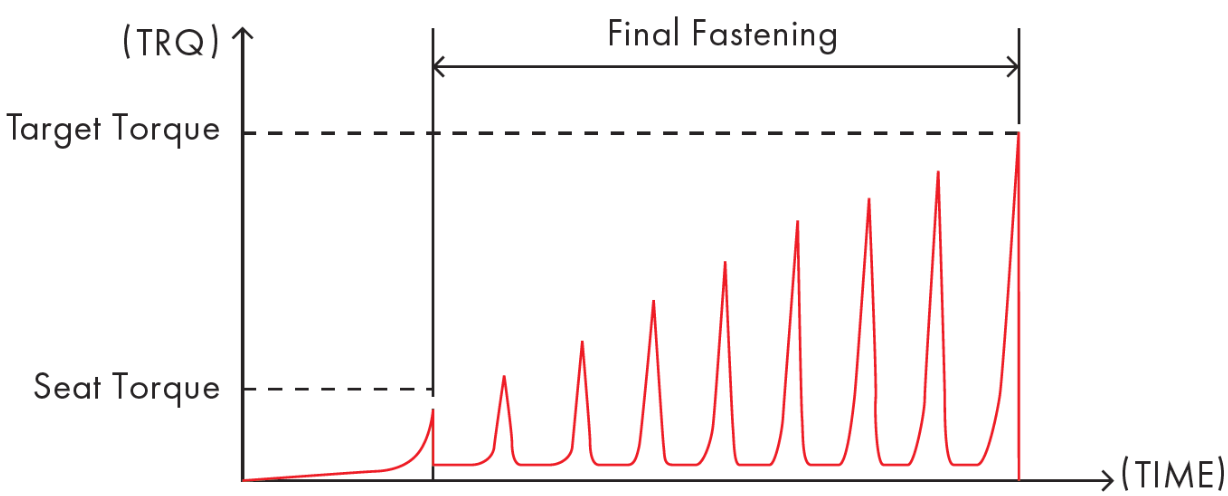
Alternative solutions

Pulse-driven tools

Most of the automatic screwdrivers and nut runners use continuous fastening techniques, in which the torque is increased continuously until reaching the target:



When the process allows it, there are some tools that use pulse-driven fastening techniques, in which the torque reaction is reduced compared to the continuous ones (some manufacturers claims reductions of the torque reaction up to 70-80 % with pulse-driven tools) by applying the torque in steps or sequence of increasing torque pulses:



Pulse-driven tools can reach tightening torques up to 150 Nm – 180 Nm depending on the manufacturer, but accuracy compared to a continuous tool could be worse, so this is one of the factors, among others, why they might not be suitable in some processes.

While reducing the torque reaction with a pulse-driven tool could be better for the robot, the vibrations associated with this technique should be carefully observed. Nevertheless, there has been successful trials using pulse-driven tools mounted to UR10/UR10e robots tightening up to 140 Nm.

Some references from manufacturers that have pulse-driven tools are:

- ESTIC
  - Handy 2000 Lite plus (some models, e.g., EH2-R2120-A)
  - Handy 2000 Touch (some models)
  - Handy 2000 Cordless (some models)
- ATLAS COPCO
  - Pistol Cordless Pulse Tool TBP

Advanced tightening strategies

Some manufacturers have functionalities to reduce the torque reaction of their tools by using advanced tightening strategies. The tightening strategy is one of the factors that the maximum torque withstanding capabilities of the robot depends on; the speed, acceleration and torque progression (all of them of the screwdriver), among others, are significant to the maximum tightening torque that the robot can handle.

One example of this is Atlas Copco and their TurboTight strategy, in which the screwdriver controller reads from the tool information about joint stiffness, and based on this it regulates the speed of the tool in such way that it slows down at the exact time when the target torque is reached, almost eliminating the reaction force in the tool.

Torque arms

The torque arms can be used to absorb and transfer the reaction force generated at the screwdriver during fastening from the robot to an external frame or structure, so the robots just needs to move to the screwdriving points and then the torque arm will be the one receiving the torque reaction. Some of them they can balance the weight of the tool. However, they may also have a significant effect on freedom of movement, reducing the flexibility and reach of the robot.

One example is the SMC Carbon Arm from Atlas Copco, with a telescopic design to maximize flexibility, a light weight and smooth movements.

While there are several options available with different mechanical configurations and payloads, the make the installation and programming of the robot more complex.

Reaction devices

This method consists of mounting a reaction bar or plate onto the front part of the tool. Then the reaction device must have something solid to react against, it could be the same workpiece if this is allowed. The torque reaction is then absorbed within the reaction device against the workpiece or fixture.

Real-time data analysis

Data collection

When doing a recording of data from the robot, the preferred interface would be the Real-Time Data Exchange (port number 30004), and the most valuable fields would be the ones that give information about:

- Joint angles/deviations
    - Target joint positions (*target_q*)
    - Actual joint positions (*actual_q*)
- Joint currents/torques
    - Target joint currents (*target_current*)
    - Actual joint currents (*actual_current*)
    - Joint current windows (*current_window*)
    - Target joint moments (*target_moment*)
- TCP pose
    - Target Cartesian coordinates of the tool (*target_TCP_pose*)
    - Actual Cartesian coordinates of the tool (*actual_TCP_pose*)
- TCP force
    - Generalized forces in the TCP (*actual_TCP_force*)
    - Raw force and torque measurement (*ft_raw_wrench*) – e-Series only, SW > 5.9.0
- Robot status
    - Safety mode (*safety_mode*)
    - Program state (*runtime_state*)

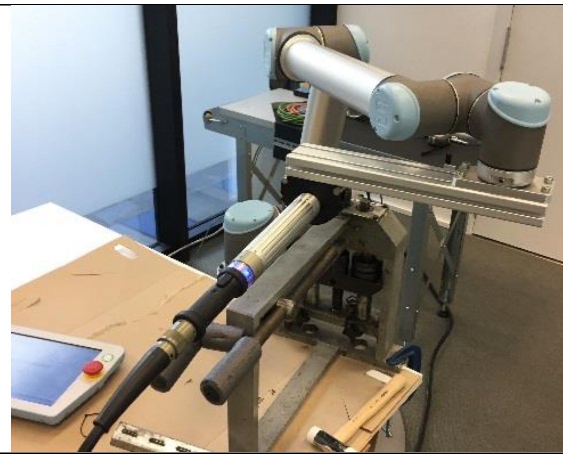
Data interpretation

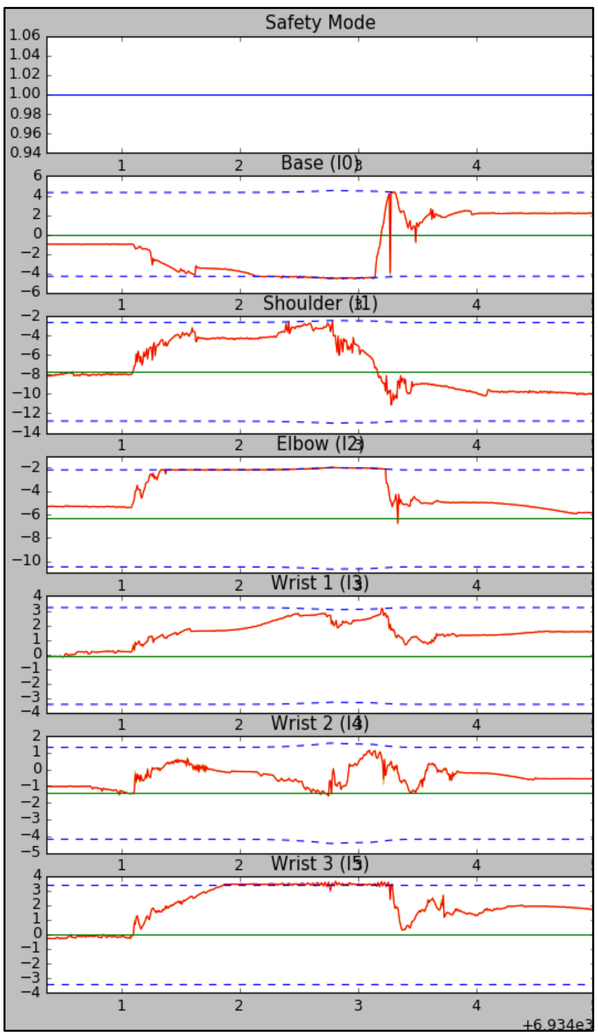
Looking at this data in a high torque screwdriving application with the payload properly defined, it should be observed in the main joints withstanding the tightening torque, that the actual currents break away from the target ones during run down of the screw, but specially at the end, when the torque reaction happens.

If one or more of the actual currents moving away from their targets, flatten at any point like they hit a limit, then it could be due to their joint torque maximum values or safety system limitations (acting on the allowed joint torques) - the limit values can change significantly depending on whether the robot is steady or not. In any case, then the joints hitting their limits would be expected to start deviating, i.e., actual joint positions break away from the target ones, since they don't have any more torque available to hold their positions. Once the joint angles start to deviate, a protective stop could be triggered if the accumulated deviation in any of the joints exceeds what is allowed for that joint in terms of position deviation, which can be larger too when the robot is in steady mode.

While having a gap between the actual and the target currents during the screw or nut tightening process would be acceptable, hitting one or more joint torque limits should be avoided; should that happen, deviations must be minimum (not evident to the naked eye), but protective stops (and any other of errors or warnings), even if isolated and random, can never be accepted as part of a normal application.

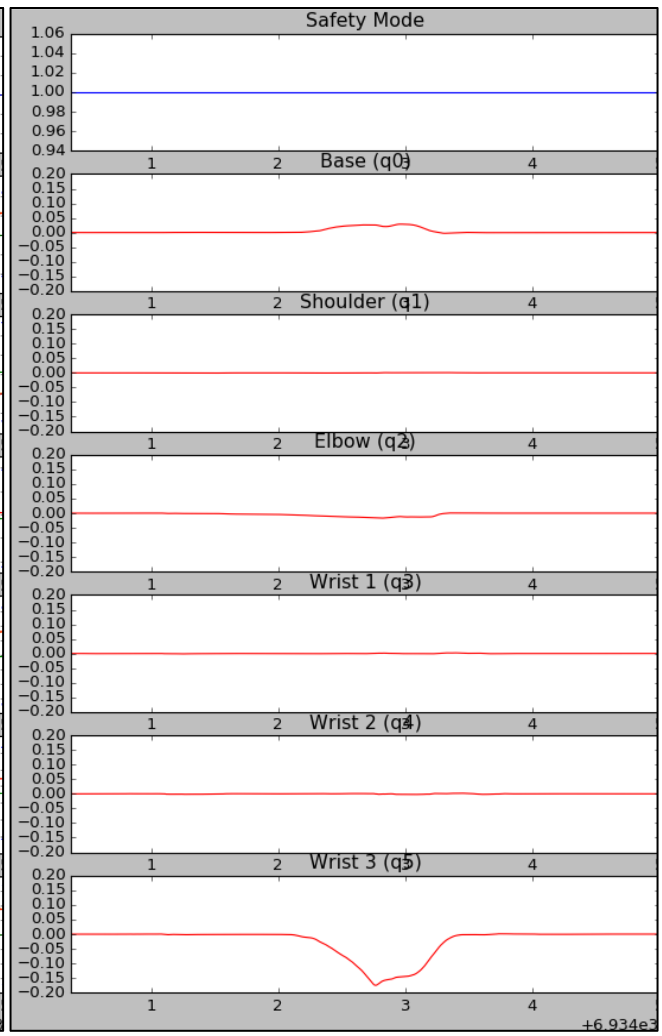
Example

Robot	UR10 CB3.0	
Software	3.4.5-100	
Safety settings	Default (CCCC)	
Steady mode	Yes	
Fastening technique	Continuous	
Target torque	120 Nm	
Tool type	Right-angle nut runner	



Joint currents [A]

— target — actual



Joint deviations [rad]

Although the robot is in steady mode and there is no protective stop according to the safety mode, this is not an acceptable application.

The Base, Elbow and Wrist 3 currents are not only breaking away from their targets as it could have been expected, but hitting a limit that can be seen when they flatten. This means that these joints can't apply any more torque to oppose to the force that is transferred from the screwdriver, and they are likely to start to deviate.

While angle deviations can be observed in the three joints that hit the limit, the biggest one is in Wrist 3. Being a right-angle nut runner mounted in such way that the Wrist 3 is aligned with the tightening axis, is likely to have a large deviation in that joint when torque limits are reached. The deviation for Wrist 3 goes up to ~10 degrees, then it should be visible to the naked eye too.

The robot pose has the TCP, located at the tip of the screwdriver, very close to the elbow joint axis of the robot, beneath the lower arm. This condition together with the force limit of the default safety settings, can restrict the torque allowed to this joint that is the first one in reaching its limit.

While some of the points presented in this document to increase the screwdriving torque withstanding capabilities for the robot are applied, due to the target of 120 Nm that lays beyond the expert range, it would require a careful consideration of all of them and an exhaustive analysis of data. Even then, alternative solutions such as pulse-driven tools or torque arms could end up being necessary to successfully deploy this application.

Changelog

Date	Version	Description	Author
31. December 2020	1.0	- document creation in final version	Germán Baños
4. January 2021	1.1	- added real-time data analysis information - added more information to Screwdriving functionality with fast-steady mode	Germán Baños