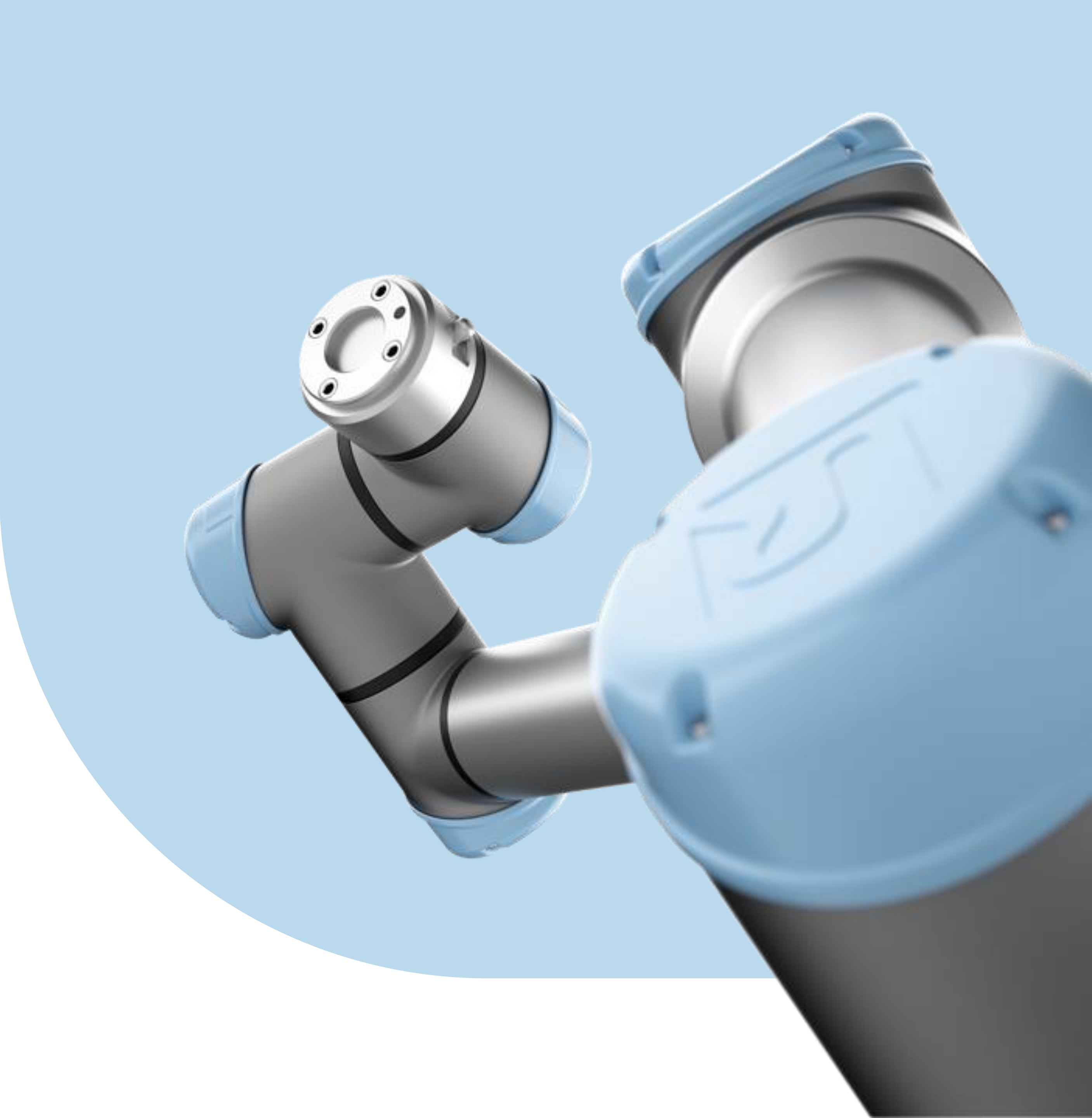
UNIVERSAL ROBOTS

Understanding Protective stops

Applicable to robots running SW 5.6 or newer

Oluf Skov Rosenlund, Tech Lead, Robotics, Product Creation, Universal Robots

Document version 1.0 24/10 2023



UNIVERSAL ROBOTS

Understanding Protective Stops

This presentation will hopefully answer some of these questions:

What exactly is a protective stop?

What are typical application problems that result in protective stops?

When the system fails, what does the errors tell you?

How to analyze and solve protective stop problems?

Agenda

1. Faults, Violations and Protective stops
2. A bit of tech
3. Protective stops
   - Protective stops due to robot (correctly or incorrectly) detects a collision
   - Protective stops that stops the robot before target trajectory exceeds a safety limit
   - Other protective stops
4. Questions?

Faults, Violations and Protective stops

Protective stops

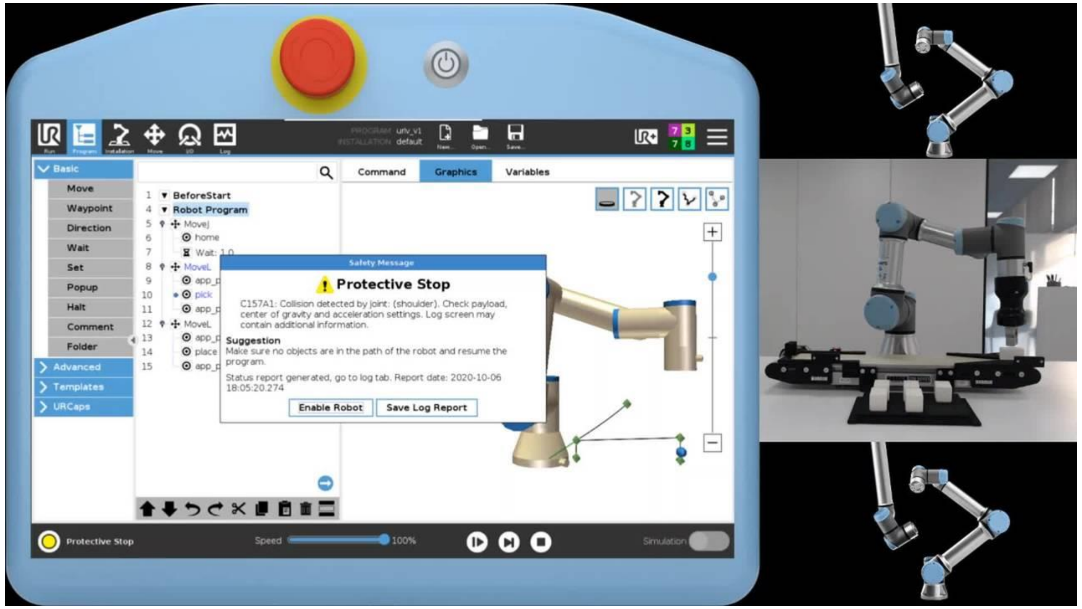
What it is:
A protective stop is issued when the robot can no longer perform the intended motion.

Detection:
Controller detects that the target trajectory cannot be followed.

Effect:
Robot decelerates, program can be resumed.

Note:
Different types pf protective stop have diffferent root causes and mitigations.

Robot can be freedrived or just pushed if it has collided with something.



Safety Message

Protective Stop

C157A1: Collision detected by joint: (shoulder). Check payload, center of gravity and acceleration settings. Log screen may contain additional information.

Suggestion
Make sure no objects are in the path of the robot and resume the program.

Status report generated, go to log tab. Report date: 2020-10-06 18:05:20.274

Safety system Violations

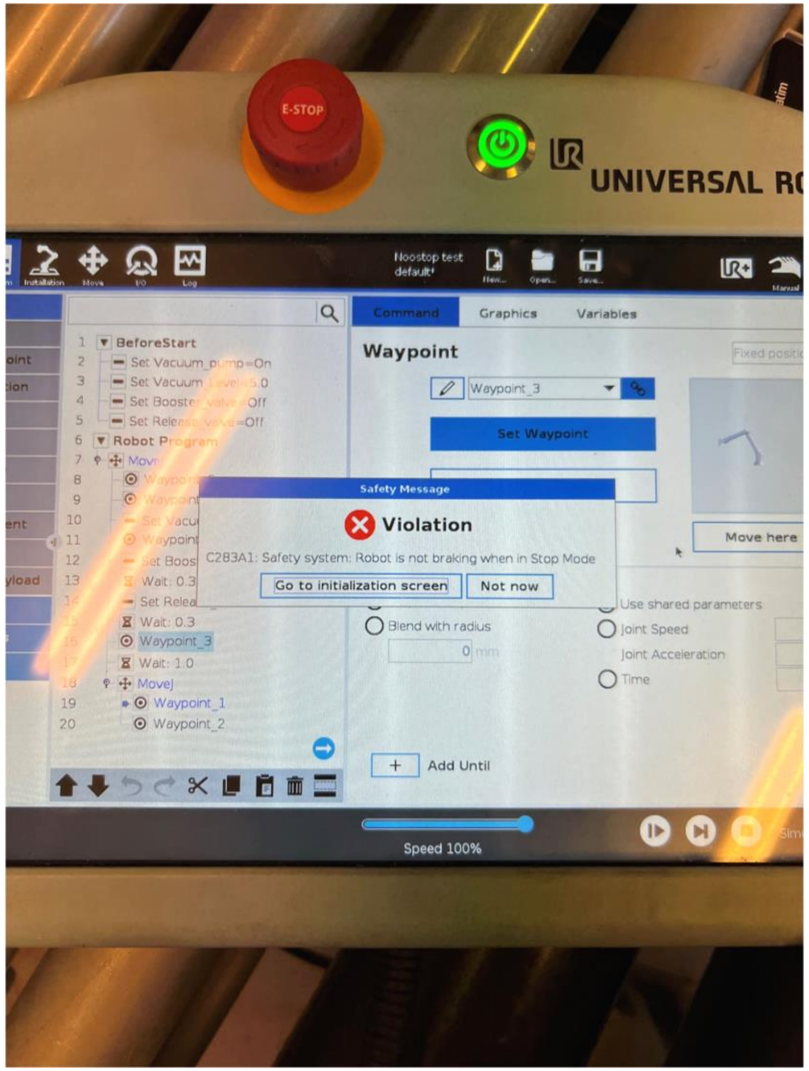
Detection:
Safety system observes a safety limt is exceeded.

Effect:
Robot decelerates and powers off motors, program can be resumed.

Note:
This should in theory never happen, Controller ensures to always react before safety system activates.

(the error on the picture usually happens if you pull the robot hard while it is e.g. Safeguard stopped).

The robot should just be restarted and the application can resume.



Safety Message

Violation

C283A1: Safety system: Robot is not braking when in Stop Mode

Go to initialization screen | Not now

Safety system Faults

Detection:
Safety system detects a fault in hardware or software.

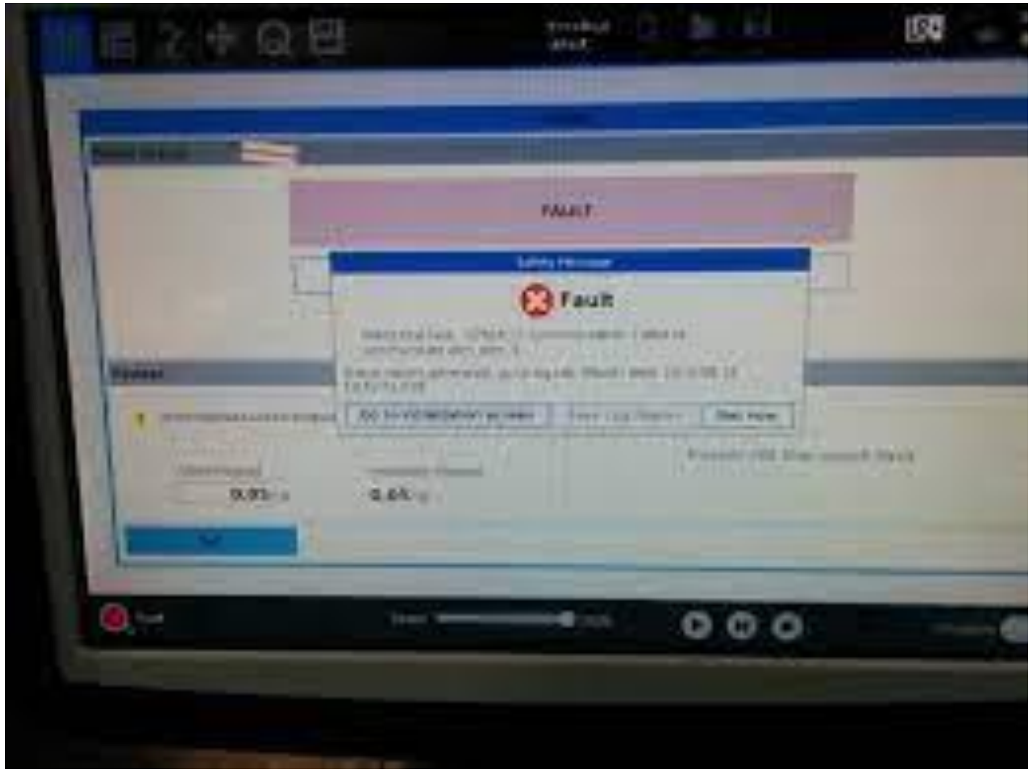
Effect:
Robot decelerates and powers off motors.

Note:
Faults can be caused by incorrectly wired safety IO (causes safety system disagreemets)

Try installing latest software and see if problem is resolved (in case of a software issue)

If problem occurs multiple times, component may need repair

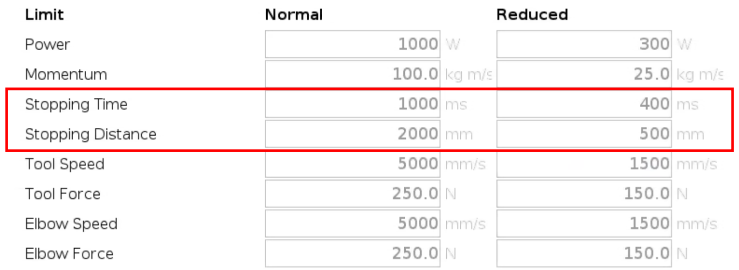
See log in Log Viewer for details



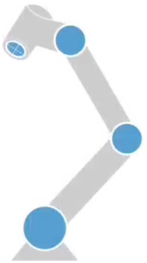
Safety Stop approach

Protective stops, Safeguard stops, E-stops and violations all pause the program so it can be resumed from where it was paused.

Robot always stops within the *stopping time* and *stopping distance* safety limits



Limit	Normal		Reduced	
Power	1000	W	300	W
Momentum	100.0	kg m/s	25.0	kg m/s
Stopping Time	1000	ms	400	ms
Stopping Distance	2000	mm	500	mm
Tool Speed	5000	mm/s	1500	mm/s
Tool Force	250.0	N	150.0	N
Elbow Speed	5000	mm/s	1500	mm/s
Elbow Force	250.0	N	150.0	N



Flow of E-stops, Violations and Faults (Cat. 0 and 1 stops)*

1. Decellerate movement on trajectory down to zero velocity
2. Engage brakes when standing still
3. Remove power from motors so robot ”parks” on the brake (robot drops a few mm during that process)

Flow of Protective stops, Safeguard stops and 3 position enabling device (Cat. 2 stops)*

1. Decellerate movement on trajectory down to zero velocity

2. If a collision is detected (C153, C157, C158, C159), robots performs a small reverse movement

*except very rare fault scenarios where encoders or motor is brokan and a Cat.0 stop

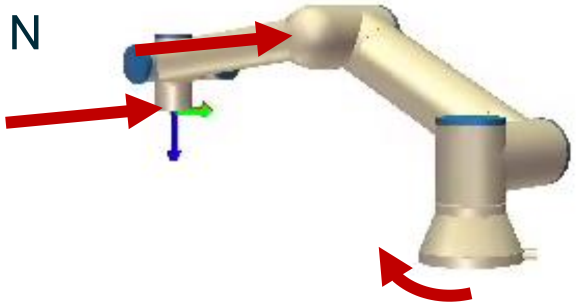
A bit of tech

Tech: Torque windows

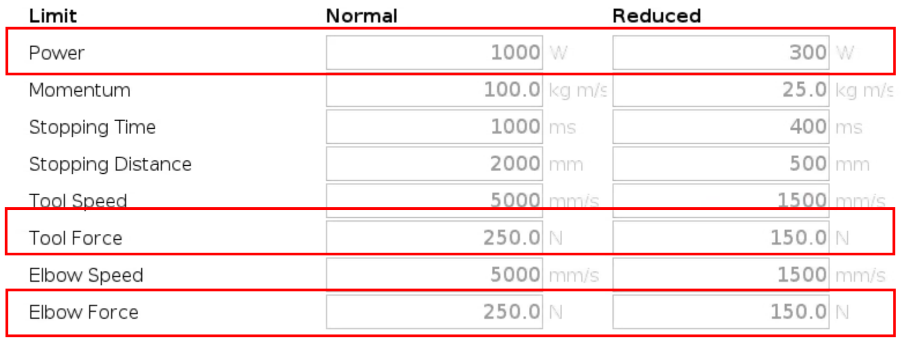
- Force and power limits affect allowed torque/current in motors
- Window is affected by robot pose
- Torque window is centered around target current

Force Limits Example

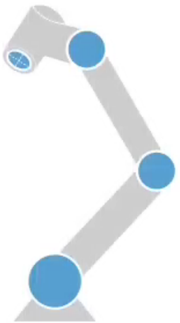
Example base torque limits:
Tool and elbow force limit = 150 N

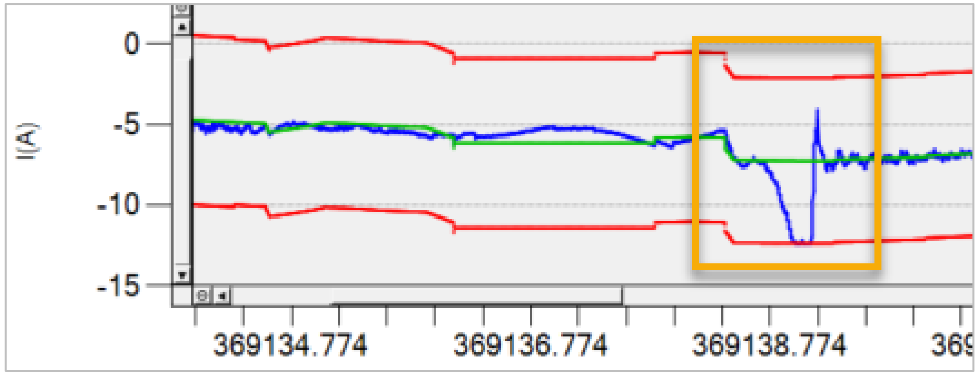


Base-tool dist: 0.9 m
0.9m x 150N = 135 Nm
Base-elbow dist: 0.5 m
0.5m x 150N = **75 Nm**
Base torque window: 75Nm



Limit	Normal	Reduced
Power	1000 W	300 W
Momentum	100.0 kg m/s	25.0 kg m/s
Stopping Time	1000 ms	400 ms
Stopping Distance	2000 mm	500 mm
Tool Speed	5000 mm/s	1500 mm/s
Tool Force	250.0 N	150.0 N
Elbow Speed	5000 mm/s	1500 mm/s
Elbow Force	250.0 N	150.0 N



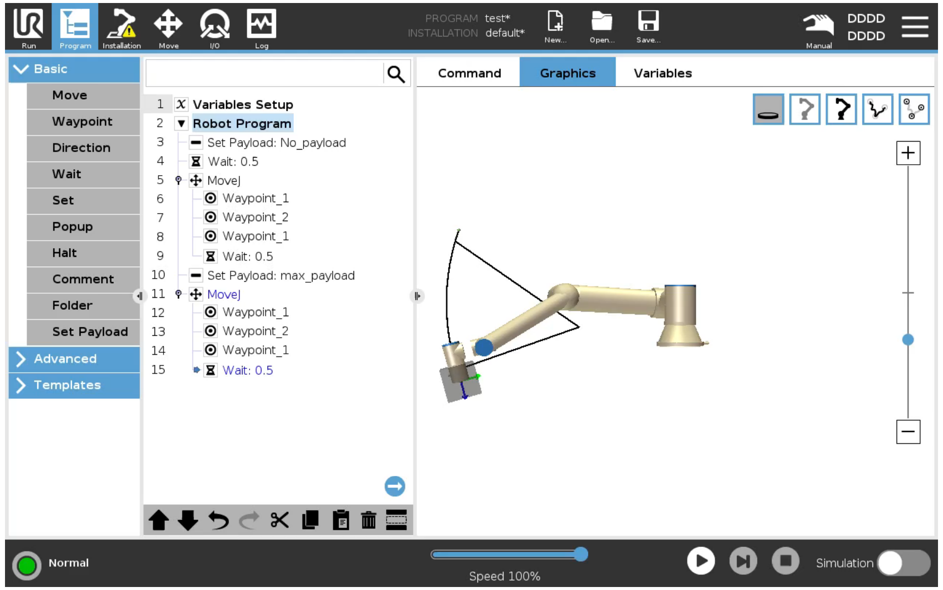


Tech: Time scaling

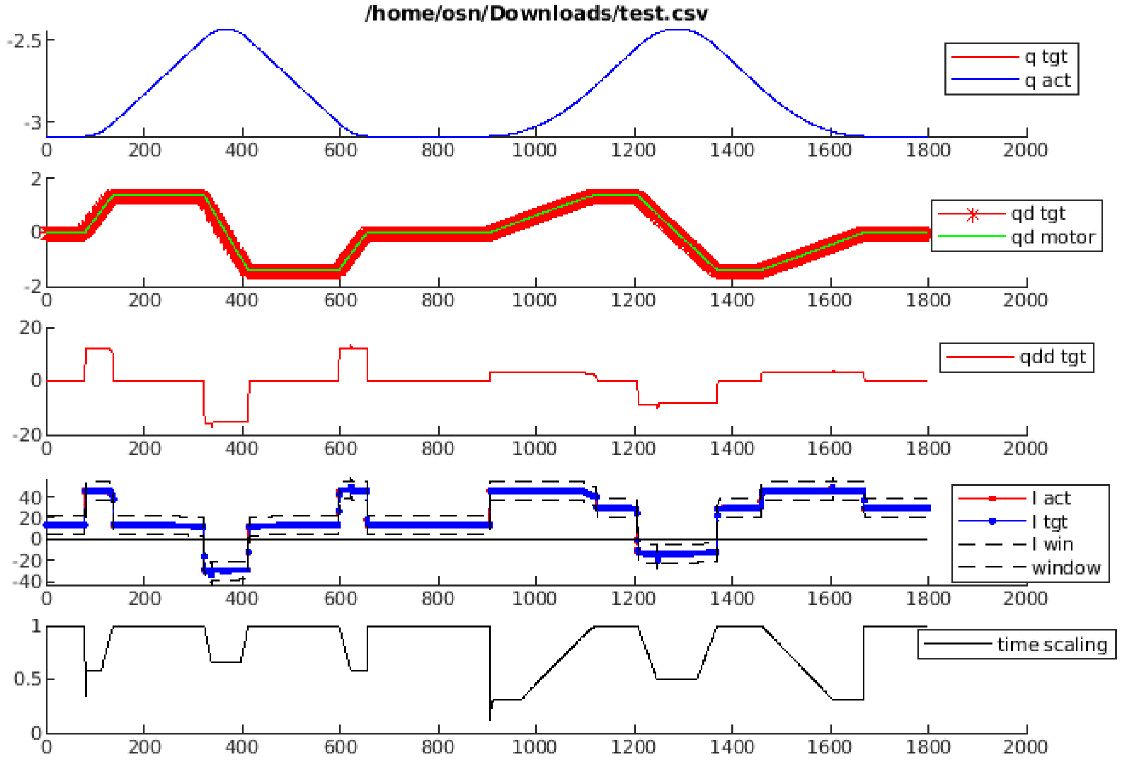
MoveJ and Movel inSW5.0 and newer:

Same velocity and acceleration for two moves.

Movement acceleration is automatically scaled due to payload change



```
Variables Setup
▼ Robot Program
    Set Payload: No_payload
    Wait: 0.5
    MoveJ
        Waypoint_1
        Waypoint_2
        Waypoint_1
    Wait: 0.5
    Set Payload: max_payload
    MoveJ
        Waypoint_1
        Waypoint_2
        Waypoint_1
    Wait: 0.5
```



Which ways do we have to know if safety system is taking effect on any moment of the cycle? Is there a way to identify which of the safety parameters is actually affecting?

It is quite hard to see what limit is scaling the robot.

- During accelerations/decelerations it is fixed limits the prevent exceeding mechanics and gear limits, or PSU limits (e.g. for the OEM CB)

- During cruise phases, it can be joint speed, TCP speed, elbow speed, momentum.

- During vertical movements typically, it is stopping time or stopping distance (a stretched robot with full payload will move slowly downwards as it almost does not have enough torque to brake if needed)

Protective stops

Types of protective stops

A protective stop is issued when the robot can no longer perform the intended motion.

Different types of protective stop have diffferent root causes and mitigations.

Protective stops due to robot (correctly or incorrectly) detects a collision:

C153: Position deviates from path
C159: Position deviates from path (zero payload)
C157: Collision detected by joint
C158: Collision detected by joint (zero payload)

Protective stops that stops the robot before target trajectory exceeds a safety limit:

C150: Position close to joint limits
C151: Tool orientation close to limits
C152: Position close to safety plane limits

Other protective stops:

C154: Position in singularity
C155: Robot cannot maintain its position, check if payload is correct
C156: Wrong payload or mounting detected, or something is pushing the robot when entering Freedrive mode

Protective stops due to robot (correctly or incorrectly) detects a collision

Error codes:

C153: Position deviates from path
C159: Position deviates from path (zero payload)
C157: Collision detected by joint
C158: Collision detected by joint (zero payload)

Check:
Deviation between target and actual position and velocity

Note:

Root causes:

- Wrong payload, wrong mounting
- External cabling dragging the robot, or collisions
- Flexible mounting or vibrating tools
- Robot on external axis
- Very high accelerations combined with blends
- External control with `servoj()` movements with very high accelerations and decelerations, or jitter
- Startup in cold environments

Let's dig into what you can do about those root causes

Protective stops due to wrong payload or mounting

Detection:
Test if the payload is correct by setting the robot in freedrive. If the robot feels gravity compensated in multiple positions and orientation, these settings are good.

Note:
TCP offset does **not** affect the dynamic model of the robot, so it does not influence protective stops.

Guidelines:

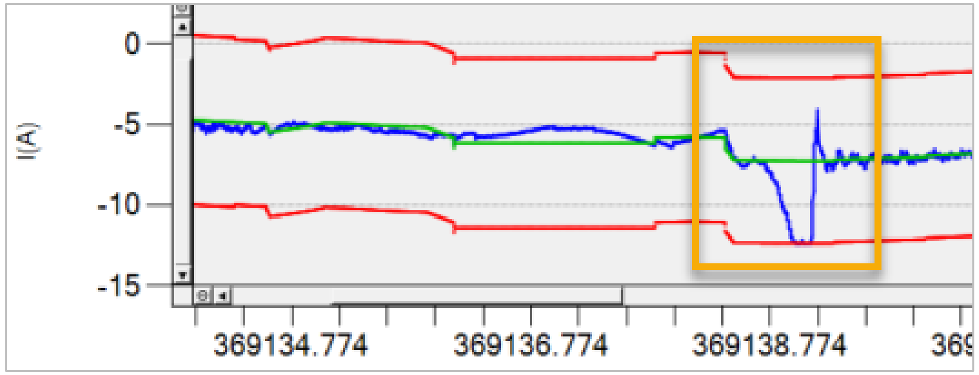
- Set payload, Center of Gravity in installation settings
- If picking and placing objects heavier than 1kg, payload should be adjusted during program using Payload program node in PolyScope or set_target_payload(m, CoG) script command
- For picking heavy objects with suction cup grippers you may use **payload transition time** to smoothly adjust the payload set_target_payload(m, cog, inertia=[0, 0, 0, 0, 0, 0], transition_time=0)

See also https://www.universal-robots.com/articles/ur/application-installation/how-to-design-for-higher-payloads/

Protective stops due to collisions

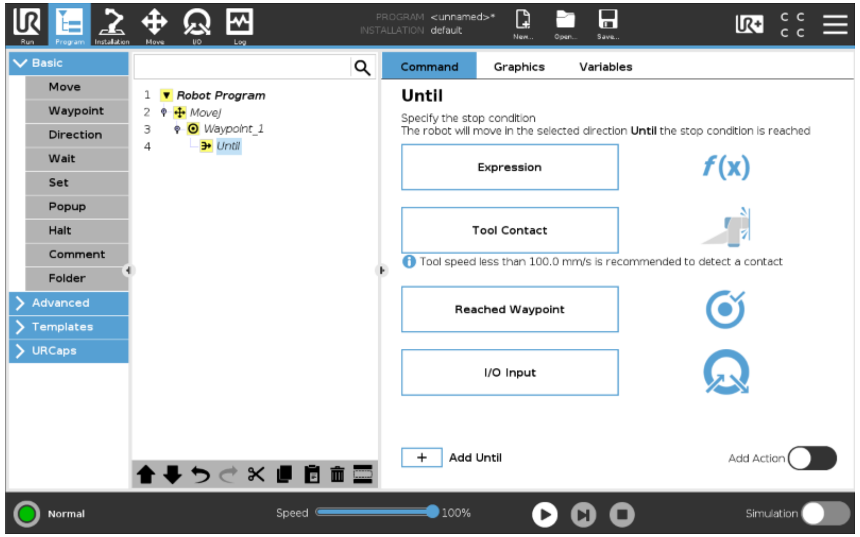
Detection:
Actual current suddently deviates from target.

Position error builds up fast



Guidelines:

- Visually inspect what caused the collision
- Use ”move until contact” PolyScope node or Force control when e.g. stacking.



1 Robot Program
2 MoveJ
3 Waypoint_1
4 Until

Until

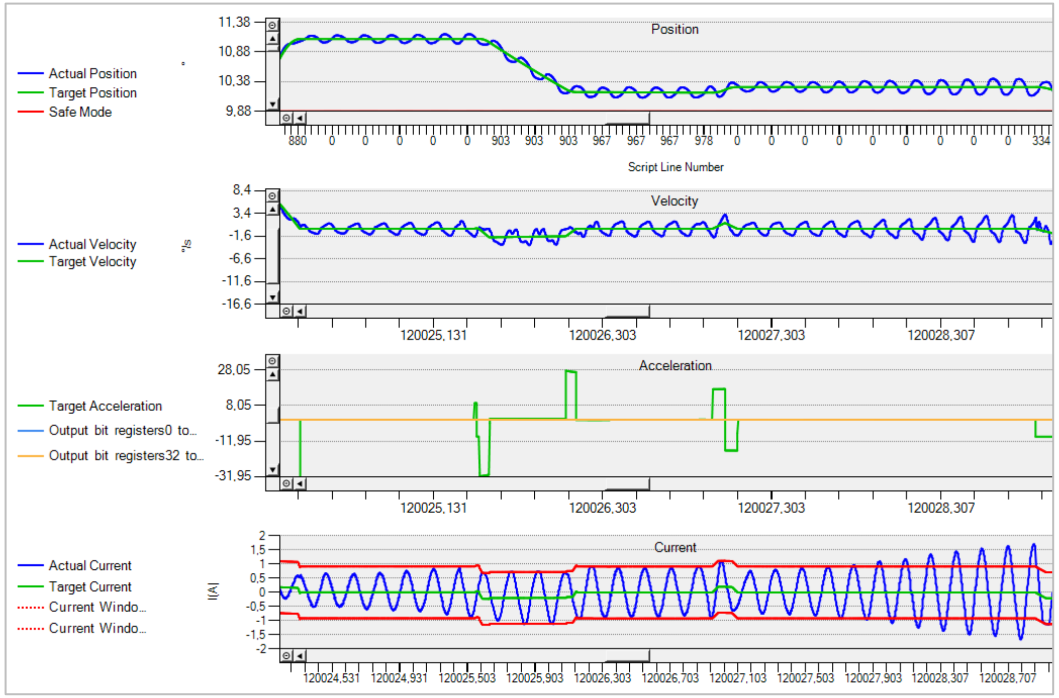
Specify the stop condition
The robot will move in the selected direction **Until** the stop condition is reached

Tool speed less than 100.0 mm/s is recommended to detect a contact

- Use urscript command position_deviation_warning() to check if parts of a program is close to causing a protective stop
- Higher force limits for tool and elbow increases allowed motor torque to move on the trajectory

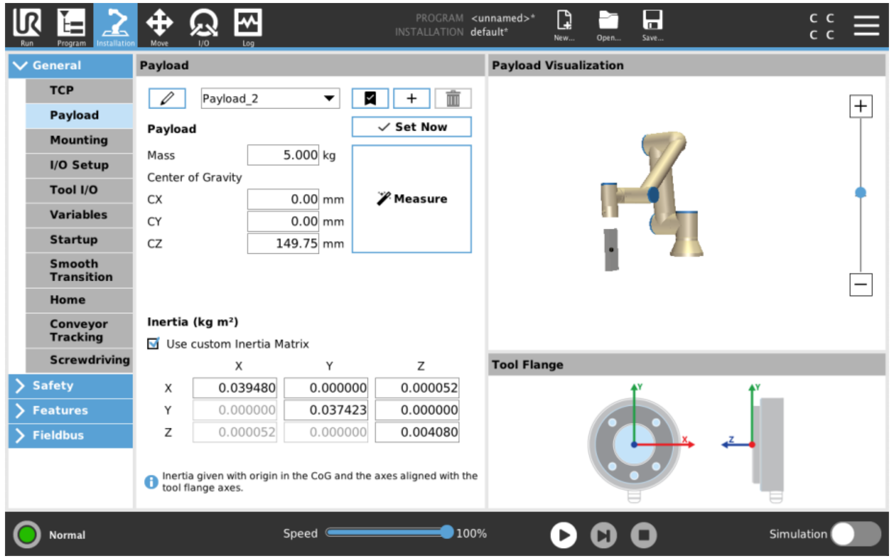
Protective stops due to vibrations

Detection:
Robot vibrates, oscillations in realtimedata



Guidelines:

- Ensure stiff mounting
- Set payload inertia in PolyScope



Payload

Payload_2

✓ Set Now

Payload

Mass 5.000 kg

Center of Gravity

CX 0.00 mm

CY 0.00 mm

CZ 149.75 mm

Measure

Inertia (kg m²)

☑ Use custom Inertia Matrix

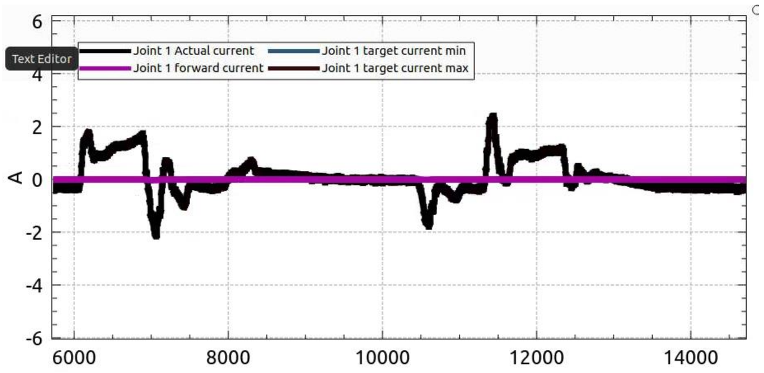
	X	Y	Z
X	0.039480	0.000000	0.000052
Y	0.000000	0.037423	0.000000
Z	0.000052	0.000000	0.004080

Inertia given with origin in the CoG and the axes aligned with the tool flange axes.

See also https://www.universal-robots.com/articles/ur/application-installation/how-to-design-for-higher-payloads/

Protective stops due to accelerations of external axes

Detection:
Varying current consumption when external axis accelerates



Guidelines:

New in SW 5.14:

Use `set_base_acceleration(a)` script command to compensate for acceleration

Use `high_holding_torque_disable()` to make the robot detect collisions when the axis is moving and the robot is standing still

In older software, you can use `set_gravity()` script command

See also new commands in https://www.universal-robots.com/articles/ur/programming/set-base-acceleration-and-high-holding-torque-disableenable/

Protective stops due to very high accelerations combined with blends

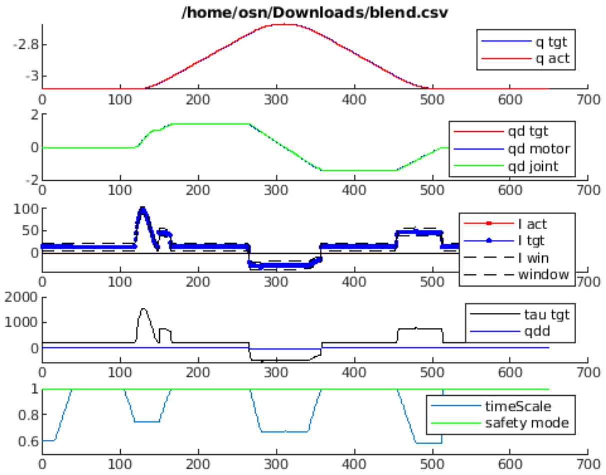
Notes:
MoveJ and MoveL without blends scales torques to be within hardware limits

Detection:
Exceeding max torque during blends
Look for warning C173 joint torque overload

Guidelines:

- Adjust accelerations to lower required torque during blends
- Scaled MoveJ and MoveL gives **optimal** cycle times
- In SW 5.6 and newer, robot will do a controlled deceleration within torque limits if leaving a “continuous if” statement or having a blend in the last waypoint of a move command. It is still a good idea to add a `stopj()` or a `stopl()` if you want a lower acceleration.

Look out for future software releases with improved handling of blends



External control with `servoj()` movements with very high accelerations and decelerations, or jitter

Notes:
* Accelerations are scaled to be within torque limts, and stay on trajectory
* Decelerations are the challenge

Detection:
Robot overshooting waypoints, or too high torque spikes during decelerations, or protective stops during decelerations

Guidelines:

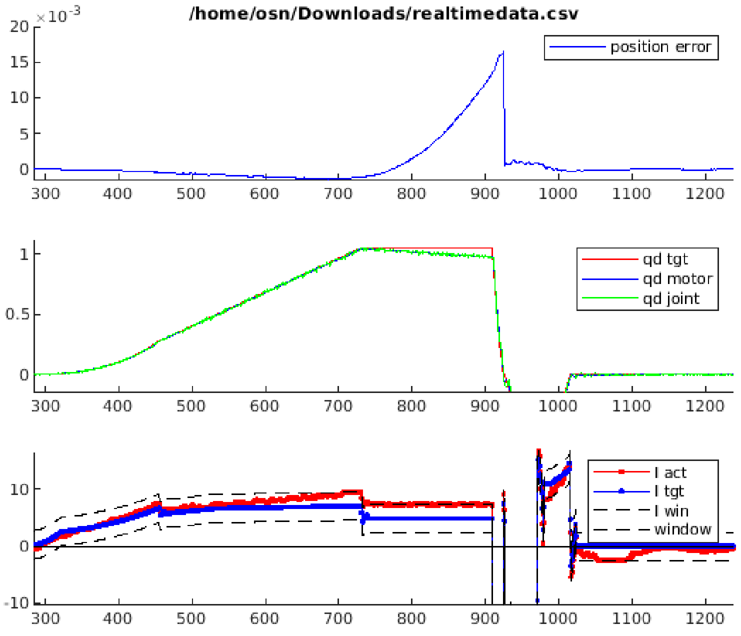
- Consider if you can switch to MoveJ or MoveL, then robot controller scales accelerations optimally for you.
- If possible, run real-time part of trajectory generation on robot controller, either as a daemon or in urscript to avoid jitter. Then you can feed positions with 500Hz
- Adjust decelerations in your trajectory generator software
- Drivers exist here:
  - https://github.com/UniversalRobots/Universal_Robots_ROS_Driver
  - Or https://gitlab.com/sdurobotics/ur_rtde

These drivers do not resolve the challenge of choosing optimal accelerations yet.
Drivers can be used as inspiration; extend with the functions you need!

Startup in cold environments

Detection:

Slow buildup of position and velocity error



Guidelines:

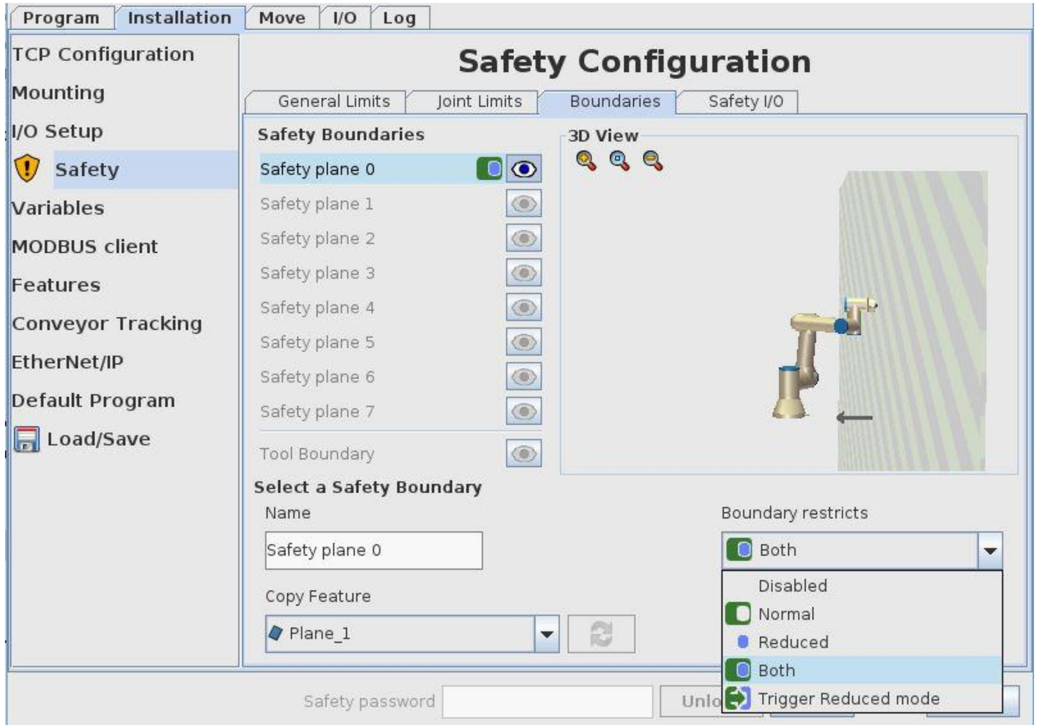
- With warmup program, robot can work down to minus 15°C ambient
- Make program with slow warmup movement
- Use SW 5.13.1 or newer
- Tested in climate chamber at UR

Protective stops that stops the robot before target trajectory exceeds a safety limit

Error codes:
C150: Position close to joint limits
C151: Tool orientation close to limits
C152: Position close to safety plane limits

Root cause:
Target trajectory exceeds a safety limit.
Robot software prevents exceeding the limit by issuing a protective stop.
Can happen if safety planes are enabled/disabled in reduced mode.

Guidelines:
Check that waypoints are placed on the right side of safety planes



Safety Configuration

Safety Boundaries

- Safety plane 0
- Safety plane 1
- Safety plane 2
- Safety plane 3
- Safety plane 4
- Safety plane 5
- Safety plane 6
- Safety plane 7
- Tool Boundary

Select a Safety Boundary

Name: Safety plane 0

Boundary restricts: Both

Copy Feature: Plane_1

Other protective stops: C155 and C156: Robot cannot maintain its position

C155: Robot cannot maintain its position, check if payload is correct

Detection:
Happens if a very wrong payload or mounting is entered. Robot starts moving upwards or downwards due to wrong dynamics. Movement is detected and stop is issued

Guideline:
Make sure payload and mounting is correct

Is something external pushing the robot?

C156: Wrong payload or mounting detected, or something is pushing the robot when entering Freedrive mode

Detection:
The robot software checks if a sudden acceleration happens right after entering freedrive. If that happens, it protective stops as this could be caused by wrong payload setting

Guideline
The check is a bit sensitive…

Try to avoid pulling the robot before the freedrive button is pressed

Don’t use freedrive when mounting or unmounting tools, unless you know what you do…

C154: Position in singularity

C154: Position in singularity

Detection:
Happens if the target trajectory + offsets from conveyor tracking, force mode, or path offset

Guidelines:
Consider if base should be tilted to avoid base singularity

Consider mounting tool at an angle to avoid wrist alignment singularity

See also https://www.universal-robots.com/articles/ur/application-installation/what-is-a-singularity/

Resources

UNIVERSAL ROBOTS

Find the latest UR software at https://www.universal-robots.com/download/?filters[]=98763&query= If experiencing problems with a robot, always try installing the latest software.

Manuals online: https://myur.universal-robots.com/manuals/content/SW_5_14/Documentation%20Menu

Release notes: https://www.universal-robots.com/articles/ur/release-notes/release-note-software-version-514xx/

Support articles: https://www.universal-robots.com/articles/?

Forum: https://forum.universal-robots.com/

Discord: https://discord.com/invite/sEjRgEf6fp

Inspect logs with Log Viewer: https://www.universal-robots.com/download/software-e-series/support/ur-log-viewer/log-viewer-v1210/

Collect realtimedata with https://github.com/UniversalRobots/RTDE_Python_Client_Library

GitHub https://github.com/orgs/UniversalRobots/repositories

Questions?

Thank you

Let’s change the world!