Coordinated Motion Programmer URCap Example

Description

The Coordinated Motion Programmer URCap is an example implementation of a Polyscope program node for teaching and playing a coordinated trajectory. The example includes programming nodes for teaching movep and movec waypoints that coordinate robot motion with axis motion. The programming nodes implement a GUI that helps programmers capture the current pose of the robot and external axes for coordinated motion using the teach pendant.

The example provides a Polyscope program tree node called "Coordinated Move" to which MovePWithAxisGroup and MoveCWithAxisGroup children nodes can be added.

The example demonstrates:

1. Generating URScript code by traversing the Polyscope program tree
2. Saving waypoint information in the Polyscope node DataModel
3. Sending URScript programs for the robot to execute over the primary interface
4. Performing socket communication between a Polyscope program node and the robot
5. Retrieving TCP poses in the external axis frame from the robot
6. Retrieving external axis positions from the robot

Build

- The URCap SDK must be installed and available in the build environment

1. `cd coordinated-motion-programing`
2. `mvn clean package -f ./pom.xml`

   The built URCap is placed in coordinated-motion-programming/target

Usage

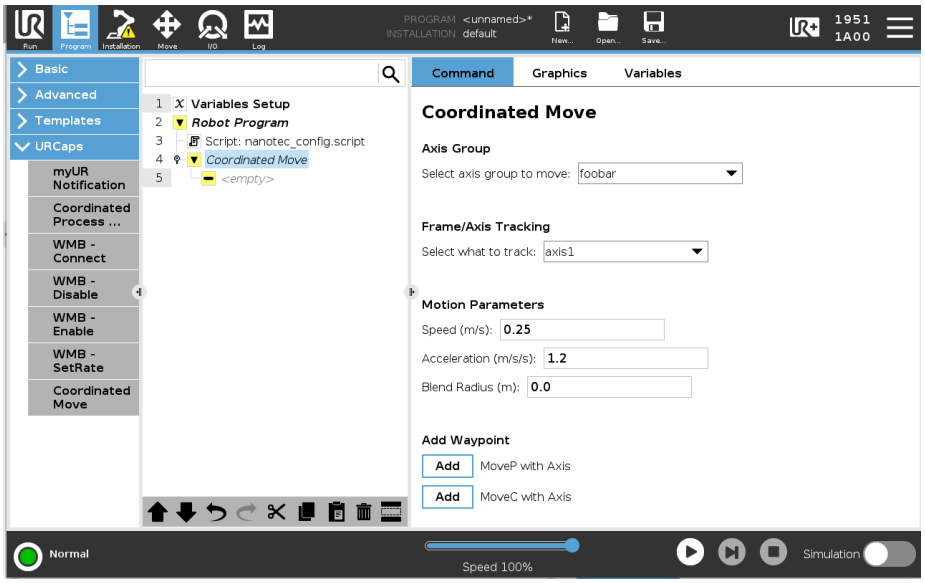
- Note: The Coordinated Motion Programmer URCap assumes that the external axis has already been set up. To successfully use the "teach" functionality where the waypoints are set from current axis and robot positions, the robot must be powered on and the EtherCAT master must be started.

1. On the robot, execute a program or script that creates an axis group, an axis, and starts the EtherCAT master.
   - Example script:

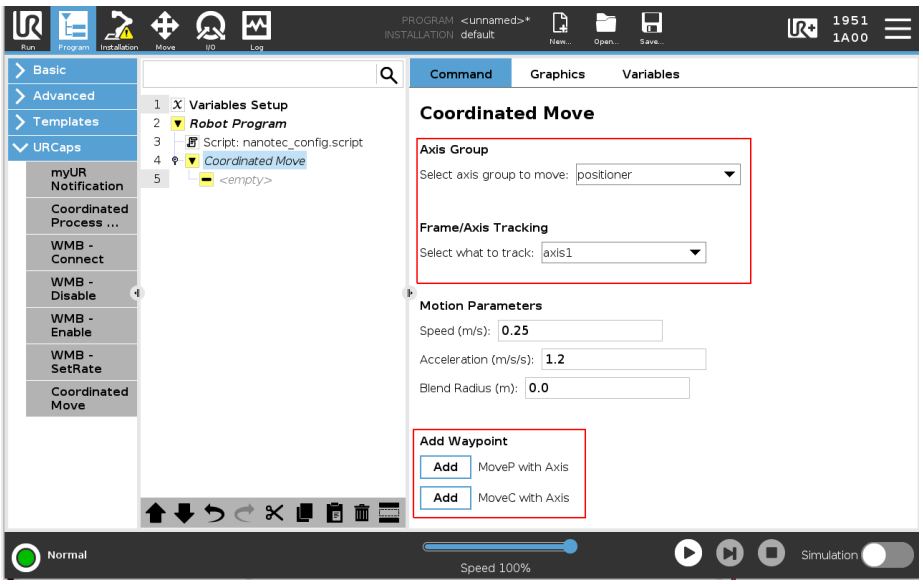
```
global PI = acos(-1)
POS_CALIBRATION=p[-0.21193751615757192,-0.41208313289342907,0.1804325232993879,
-1.2343073254390127,-1.1788473562139317,1.1924086318005989]
POSITIONER_AXIS_TYPE = 0            # 0:revolute, 1:prismatic
POSITIONER_VELOCITY_LIMIT = 0.7*2     # rad/s
POSITIONER_ACCELERATION_LIMIT = 0.1*100 # rad/s^2
POSITIONER_ENCODER_RESOLUTION = 3600
POSITIONER_FEED_CONSTANT = 2 * PI
POSITIONER_GEAR_RATIO = 10
POSITIONER_ZERO_OFFSET = 480

# stop EtherCAT master first, otherwise, reset_world_model will throw an error
ethercat_clear_error()
ethercat_stop(True)

reset_world_model()
axis_group_add("positioner", POS_CALIBRATION, "base")
axis_group_add_axis("positioner", "axis1", "", p[0,0,0,0,0,0],
POSITIONER_AXIS_TYPE, POSITIONER_VELOCITY_LIMIT, POSITIONER_ACCELERATION_LIMIT)

ethercat_config_axis("axis1", 1, POSITIONER_ENCODER_RESOLUTION,
POSITIONER_GEAR_RATIO, POSITIONER_FEED_CONSTANT, POSITIONER_ZERO_OFFSET)
ethercat_start(10)
```

- You can also use the ethercat-axis-config reference design URCap to configure the axis group and axis.

2. After the program has finished, go to the program tree and add a new "Coordinated Process Move" node to the tree:

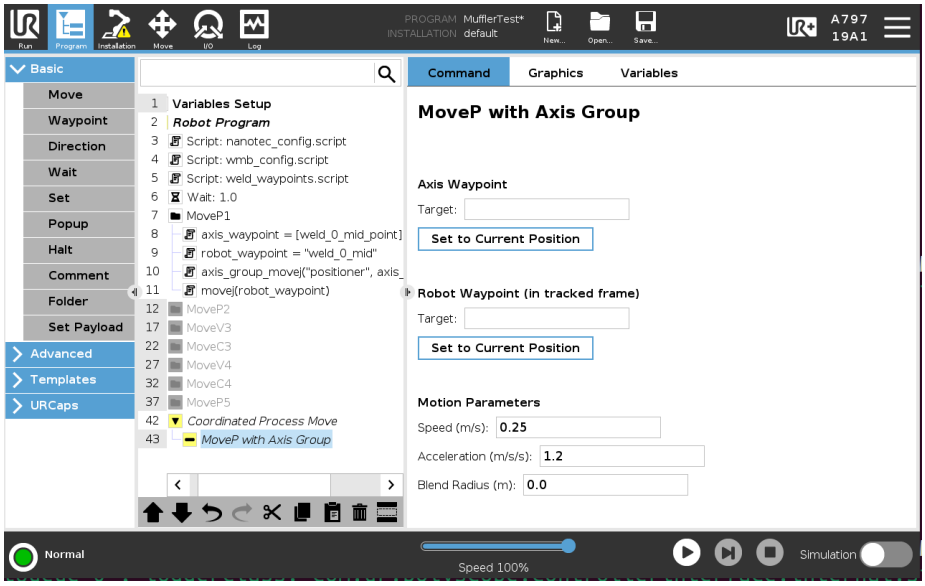


1. Select the parameters for the axis group and frame/axis to track for coordinated motion and then add children move nodes by pressing the "Add" button next to the desired type.

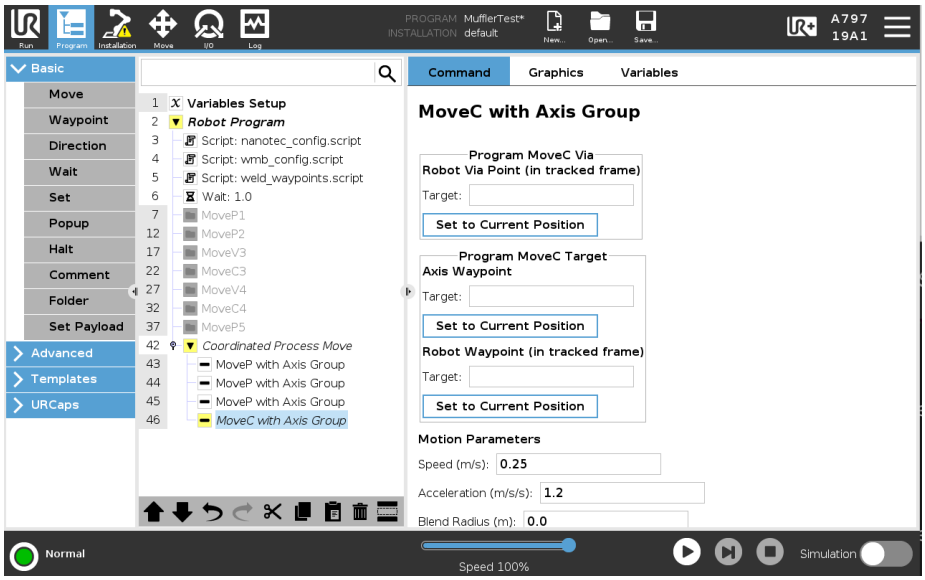


1. When adding a MoveP child node, set the axis and robot waypoints. This can be done by typing in the text boxes or by pressing the "Set to Current Position" which stores the current position of the axis or robot for the waypoint. If you manually type the axis waypoint in the text box, the input must be a comma separated list of numbers inside square braces

where the number of elements corresponds to the number of axes in the group (specified in the parent node). Here is an example of the waypoint for a two-axis system: "[0.1234, 1.2345]". If you manually type the robot waypoint in the text box, it must be the string representation of a URScript pose, e.g., "p[1,2,3,4,5,6]". The pose is represented in the frame or axis being tracked, which is defined in the parent node.



1. When adding a MoveC child node, first set the via robot waypoint for the movec motion, then set the axis and robot targets for the endpoint of the movec motion.


Run
Program
Installation
Move
I/O
Log
PROGRAM MufflerTest*
INSTALLATION default
New...
Open...
Save...
A797
19A1
Basic
Move
Waypoint
Direction
Wait
Set
Popup
Halt
Comment
Folder
Set Payload
Advanced
Templates
URCaps
1 Variables Setup
2 Robot Program
3 Script: nanotec_config.script
4 Script: wmb_config.script
5 Script: weld_waypoints.script
6 Wait: 1.0
7 MoveP1
12 MoveP2
17 MoveV3
22 MoveC3
27 MoveV4
32 MoveC4
37 MoveP5
42 Coordinated Process Move
43 MoveP with Axis Group
44 MoveP with Axis Group
45 MoveP with Axis Group
46 MoveC with Axis Group
Command
Graphics
Variables
MoveC with Axis Group
Program MoveC Via
Robot Via Point (in tracked frame)
Target:
Set to Current Position
Program MoveC Target
Axis Waypoint
Target:
Set to Current Position
Robot Waypoint (in tracked frame)
Target:
Set to Current Position
Motion Parameters
Speed (m/s): 0.25
Acceleration (m/s/s): 1.2
Blend Radius (m): 0.0
Normal
Speed 100%
Simulation