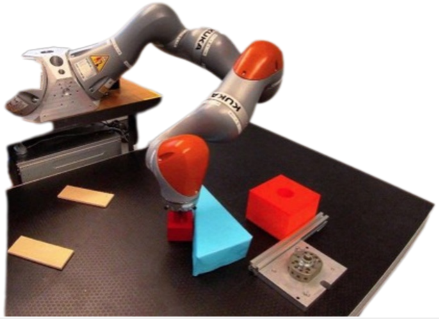
Lernende Roboter mit Skills

Matthias Mayr, Lund University & WASP
matthias.mayr@cs.lth.se



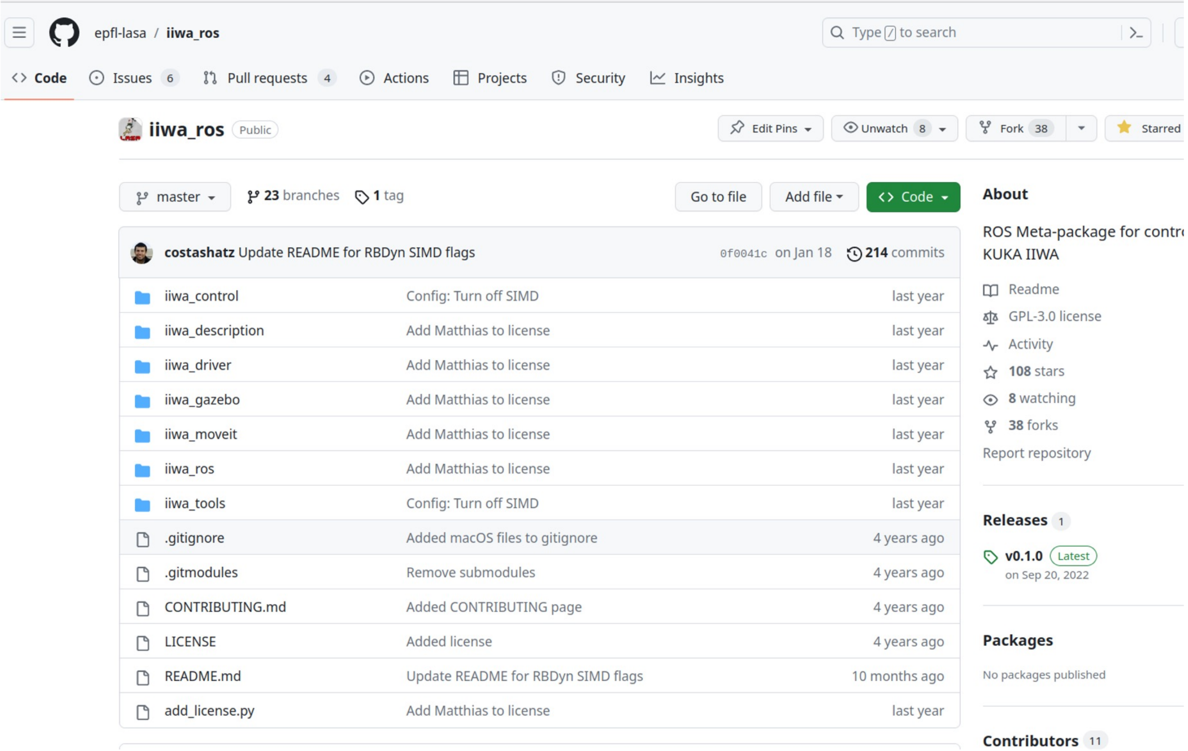
WASP WALLENBERG AI, AUTONOMOUS SYSTEMS AND SOFTWARE PROGRAM

LUNDS UNIVERSITET
Lunds Tekniska Högskola

Was brauchen wir?

- Lernende
- Roboter
- Skills

Robot



epfl-lasa / iiwa_ros

iiwa_ros Public

23 branches 1 tag

costashatz Update README for RBDyn SIMD flags 0f0041c on Jan 18 214 commits

iiwa_control	Config: Turn off SIMD	last year
iiwa_description	Add Matthias to license	last year
iiwa_driver	Add Matthias to license	last year
iiwa_gazebo	Add Matthias to license	last year
iiwa_moveit	Add Matthias to license	last year
iiwa_ros	Add Matthias to license	last year
iiwa_tools	Config: Turn off SIMD	last year
.gitignore	Added macOS files to gitignore	4 years ago
.gitmodules	Remove submodules	4 years ago
CONTRIBUTING.md	Added CONTRIBUTING page	4 years ago
LICENSE	Added license	4 years ago
README.md	Update README for RBDyn SIMD flags	10 months ago
add_license.py	Add Matthias to license	last year

About

ROS Meta-package for contr
KUKA IIWA

- Readme
- GPL-3.0 license
- Activity
- 108 stars
- 8 watching
- 38 forks

Report repository

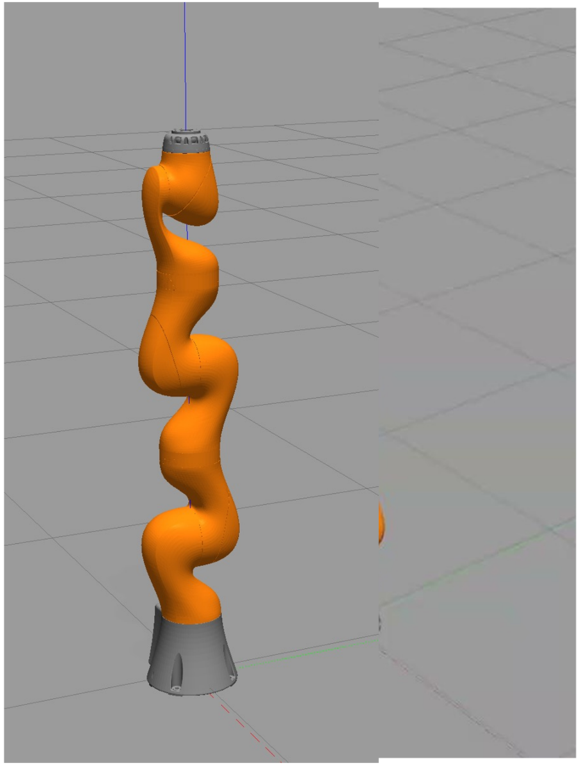
Releases 1

v0.1.0 Latest
on Sep 20, 2022

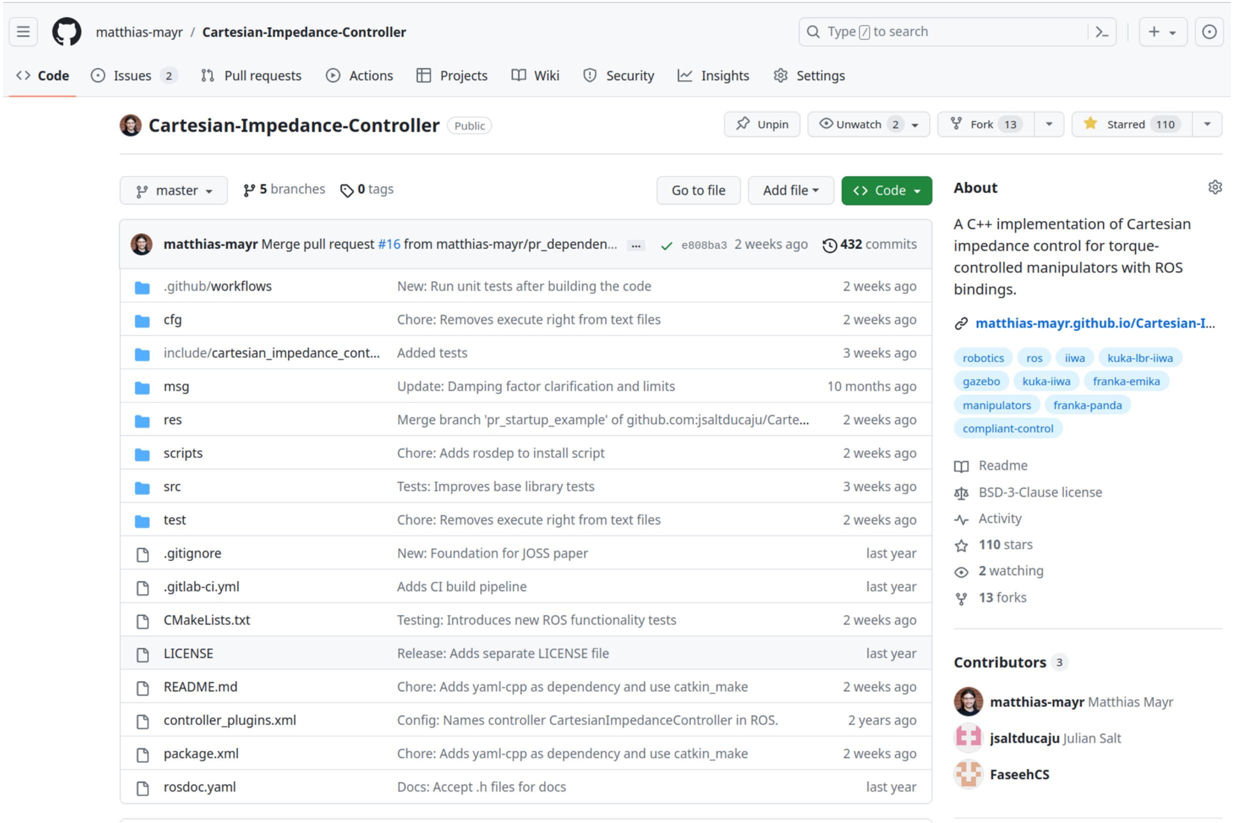
Packages

No packages published

Contributors 11



Control



matthias-mayr / Cartesian-Impedance-Controller

Type / to search

Code · Issues 2 · Pull requests · Actions · Projects · Wiki · Security · Insights · Settings

Cartesian-Impedance-Controller Public

Unpin · Unwatch 2 · Fork 13 · Starred 110

master · 5 branches · 0 tags · Go to file · Add file · Code

matthias-mayr Merge pull request #16 from matthias-mayr/pr_dependen... … ✓ e808ba3 2 weeks ago · 432 commits

.github/workflows	New: Run unit tests after building the code	2 weeks ago
cfg	Chore: Removes execute right from text files	2 weeks ago
include/cartesian_impedance_cont...	Added tests	3 weeks ago
msg	Update: Damping factor clarification and limits	10 months ago
res	Merge branch 'pr_startup_example' of github.com:jsaltducaju/Carte...	2 weeks ago
scripts	Chore: Adds rosdep to install script	2 weeks ago
src	Tests: Improves base library tests	3 weeks ago
test	Chore: Removes execute right from text files	2 weeks ago
.gitignore	New: Foundation for JOSS paper	last year
.gitlab-ci.yml	Adds CI build pipeline	last year
CMakeLists.txt	Testing: Introduces new ROS functionality tests	2 weeks ago
LICENSE	Release: Adds separate LICENSE file	last year
README.md	Chore: Adds yaml-cpp as dependency and use catkin_make	2 weeks ago
controller_plugins.xml	Config: Names controller CartesianImpedanceController in ROS.	2 years ago
package.xml	Chore: Adds yaml-cpp as dependency and use catkin_make	2 weeks ago
rosdoc.yaml	Docs: Accept .h files for docs	last year

About

A C++ implementation of Cartesian impedance control for torque-controlled manipulators with ROS bindings.

matthias-mayr.github.io/Cartesian-I...

robotics · ros · iiwa · kuka-lbr-iiwa · gazebo · kuka-iiwa · franka-emika · manipulators · franka-panda · compliant-control

Readme

BSD-3-Clause license

Activity

110 stars

2 watching

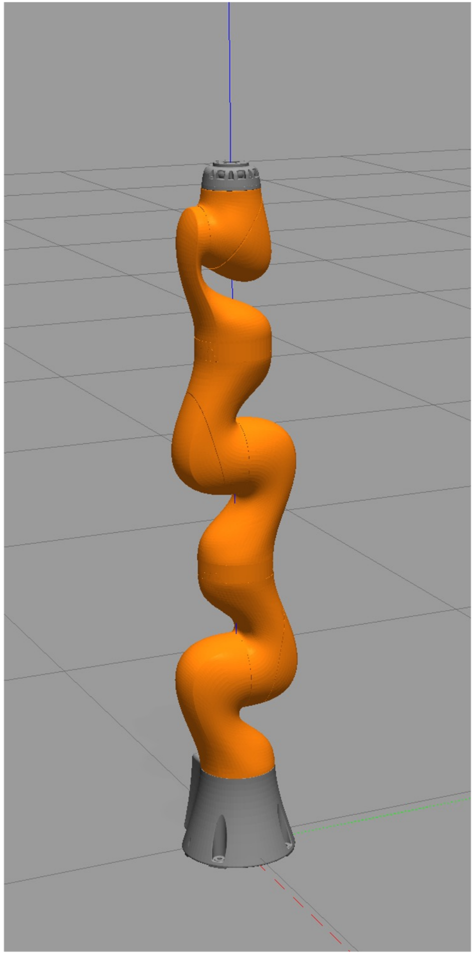
13 forks

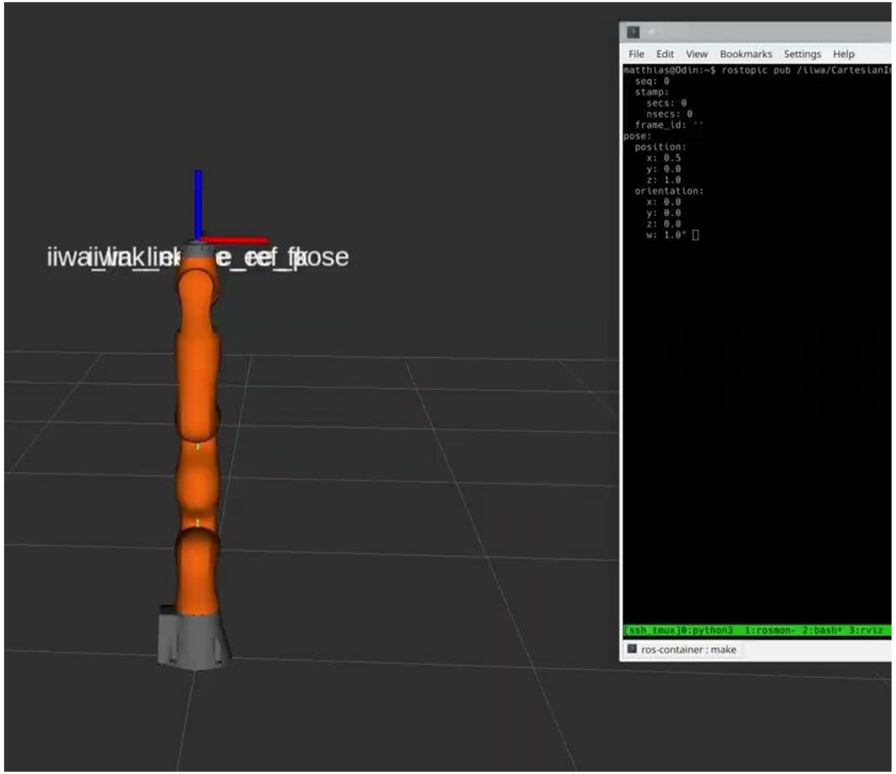
Contributors 3

matthias-mayr Matthias Mayr

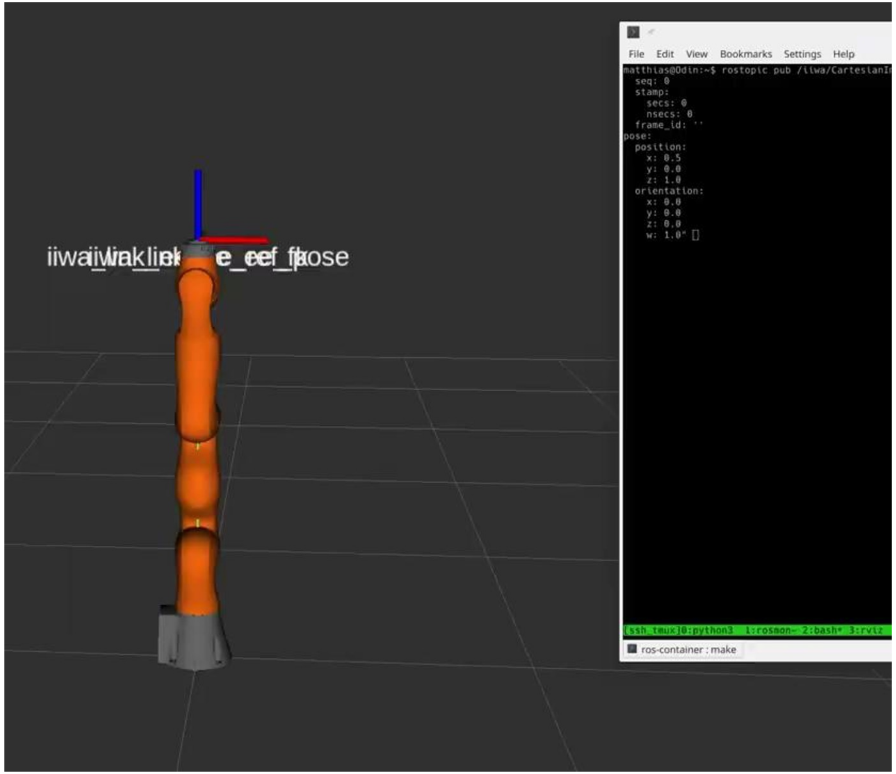
jsaltducaju Julian Salt

FaseehCS



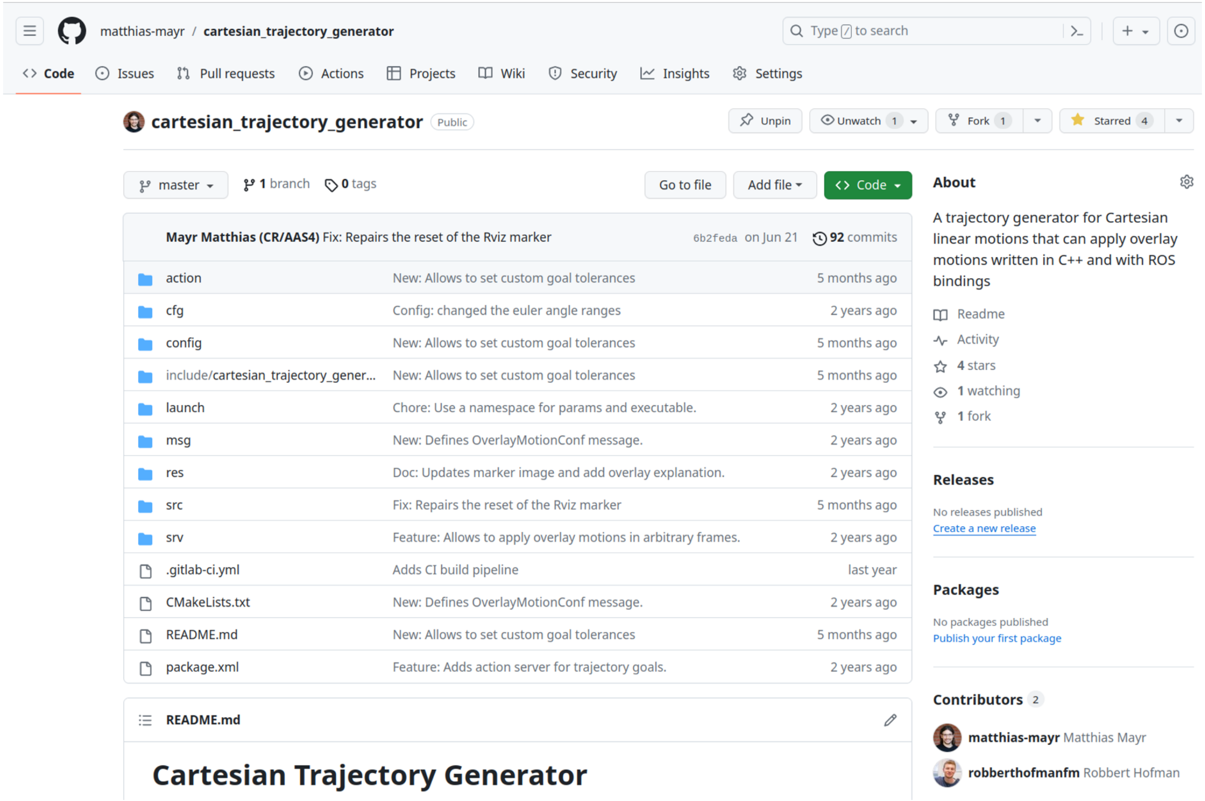

iiwaiiwaklink e_eef_fpose
File Edit View Bookmarks Settings Help
ros-container : make

```
matthias@Odin:~$ rostopic pub /iiwa/CartesianI
  seq: 0
  stamp:
    secs: 0
    nsecs: 0
  frame_id: ''
pose:
  position:
    x: 0.5
    y: 0.0
    z: 1.0
  orientation:
    x: 0.0
    y: 0.0
    z: 0.0
    w: 1.0" []
[ssh_tmux]0:python3  1:rosmon- 2:bash* 3:rviz
```


iiwaiiwaklink e_eef_fpose
File Edit View Bookmarks Settings Help
ros-container : make

```
matthias@Odin:~$ rostopic pub /iwa/CartesianI
  seq: 0
  stamp:
    secs: 0
    nsecs: 0
  frame_id: ''
pose:
  position:
    x: 0.5
    y: 0.0
    z: 1.0
  orientation:
    x: 0.0
    y: 0.0
    z: 0.0
    w: 1.0" []
[ssh_tmux]0:python3  1:rosmon- 2:bash* 3:rviz
```

Cartesian Trajectories



cartesian_trajectory_generator Public

1 branch 0 tags

Mayr Matthias (CR/AAS4) Fix: Repairs the reset of the Rviz marker 6b2feda on Jun 21 92 commits

action	New: Allows to set custom goal tolerances	5 months ago
cfg	Config: changed the euler angle ranges	2 years ago
config	New: Allows to set custom goal tolerances	5 months ago
include/cartesian_trajectory_gener...	New: Allows to set custom goal tolerances	5 months ago
launch	Chore: Use a namespace for params and executable.	2 years ago
msg	New: Defines OverlayMotionConf message.	2 years ago
res	Doc: Updates marker image and add overlay explanation.	2 years ago
src	Fix: Repairs the reset of the Rviz marker	5 months ago
srv	Feature: Allows to apply overlay motions in arbitrary frames.	2 years ago
.gitlab-ci.yml	Adds CI build pipeline	last year
CMakeLists.txt	New: Defines OverlayMotionConf message.	2 years ago
README.md	New: Allows to set custom goal tolerances	5 months ago
package.xml	Feature: Adds action server for trajectory goals.	2 years ago

README.md

Cartesian Trajectory Generator

About

A trajectory generator for Cartesian linear motions that can apply overlay motions written in C++ and with ROS bindings

- Readme
- Activity
- 4 stars
- 1 watching
- 1 fork

Releases

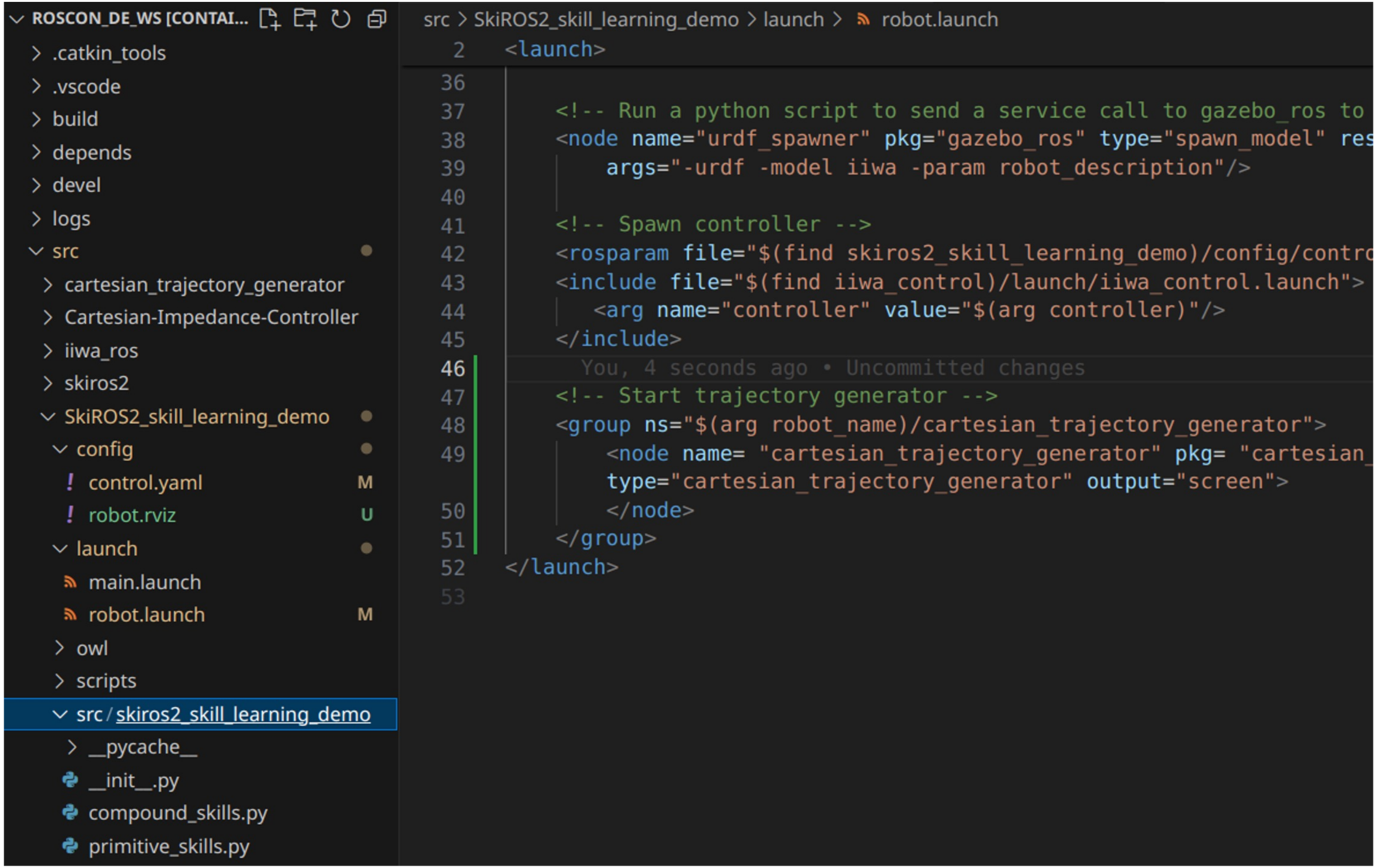
No releases published
Create a new release

Packages

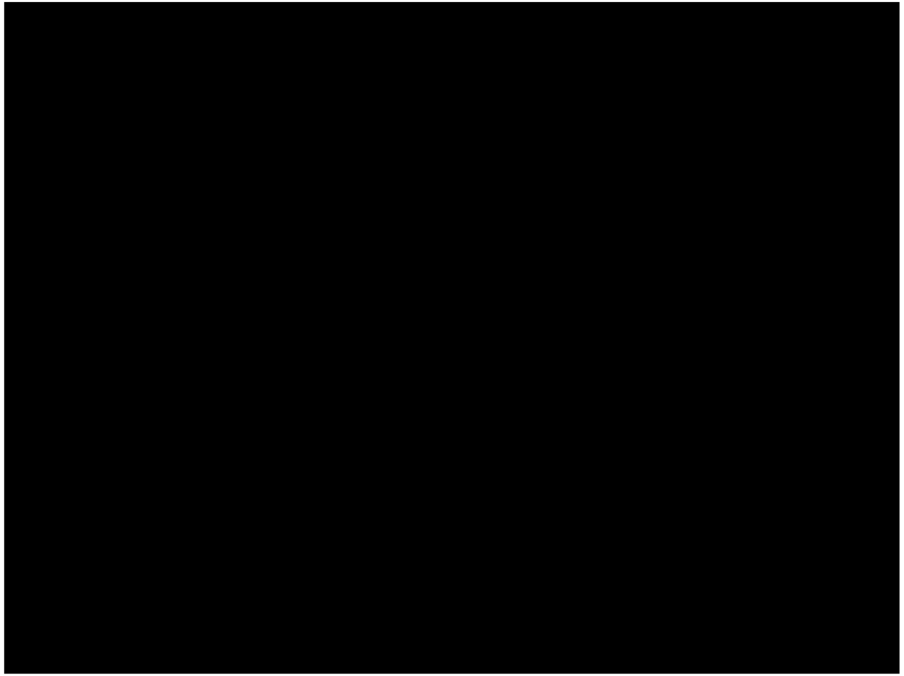
No packages published
Publish your first package

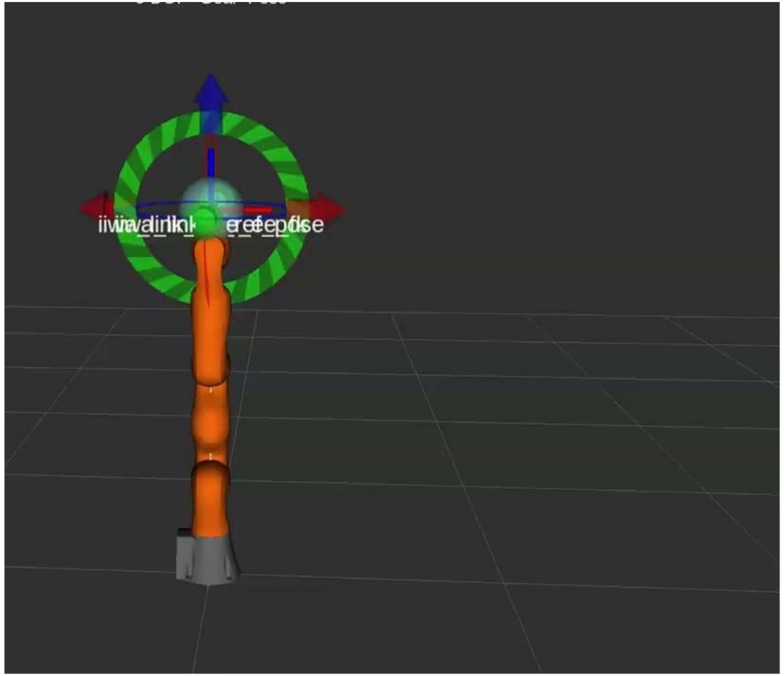
Contributors 2

- matthias-mayr Matthias Mayr
- robberthofmanfm Robbert Hofman

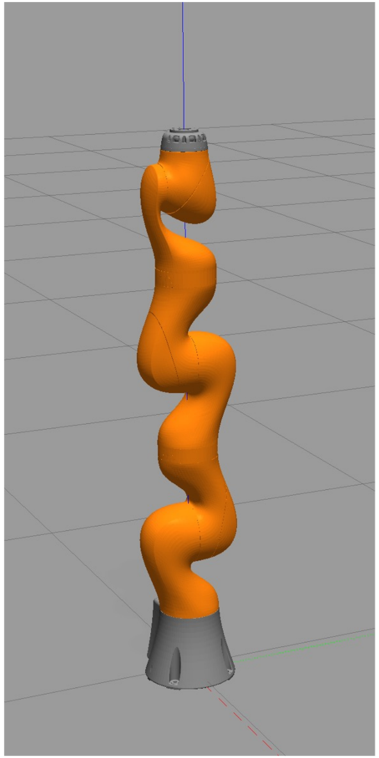

ROSCON_DE_WS [CONTAI...
.catkin_tools
.vscode
build
depends
devel
logs
src
cartesian_trajectory_generator
Cartesian-Impedance-Controller
iiwa_ros
skiros2
SkiROS2_skill_learning_demo
config
control.yaml M
robot.rviz U
launch
main.launch
robot.launch M
owl
scripts
src / skiros2_skill_learning_demo
__pycache__
__init__.py
compound_skills.py
primitive_skills.py
src > SkiROS2_skill_learning_demo > launch > robot.launch
You, 4 seconds ago • Uncommitted changes

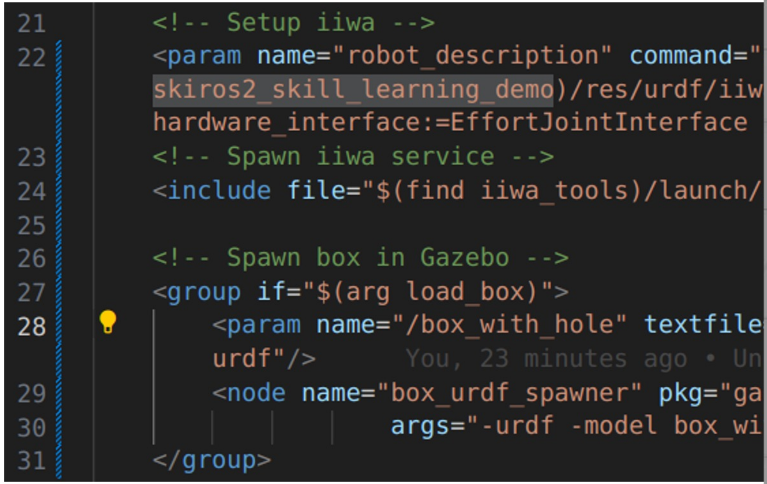
```
<launch>

    <!-- Run a python script to send a service call to gazebo_ros to
    <node name="urdf_spawner" pkg="gazebo_ros" type="spawn_model" res
        args="-urdf -model iiwa -param robot_description"/>

    <!-- Spawn controller -->
    <rosparam file="$(find skiros2_skill_learning_demo)/config/contro
    <include file="$(find iiwa_control)/launch/iiwa_control.launch">
      <arg name="controller" value="$(arg controller)"/>
    </include>

    <!-- Start trajectory generator -->
    <group ns="$(arg robot_name)/cartesian_trajectory_generator">
        <node name= "cartesian_trajectory_generator" pkg= "cartesian_
        type="cartesian_trajectory_generator" output="screen">
        </node>
    </group>
</launch>
```



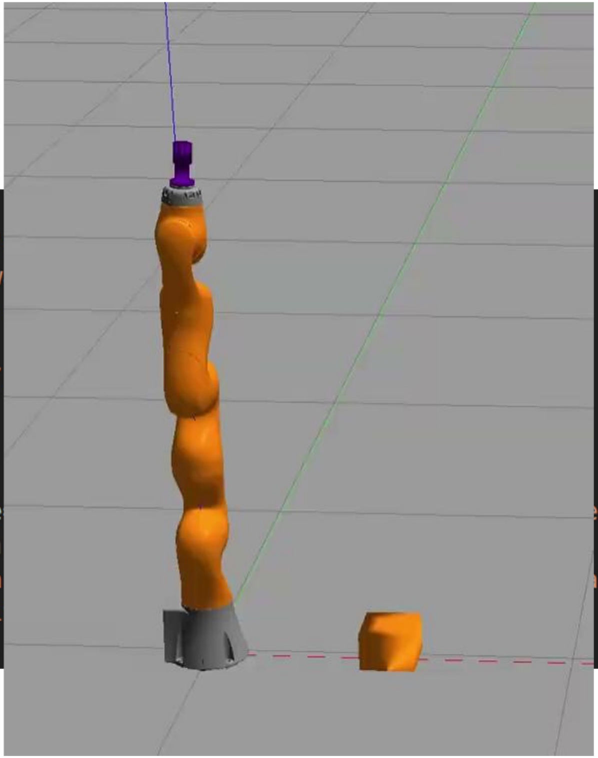

iiwa_link
ee_pose

The Peg and the Hole

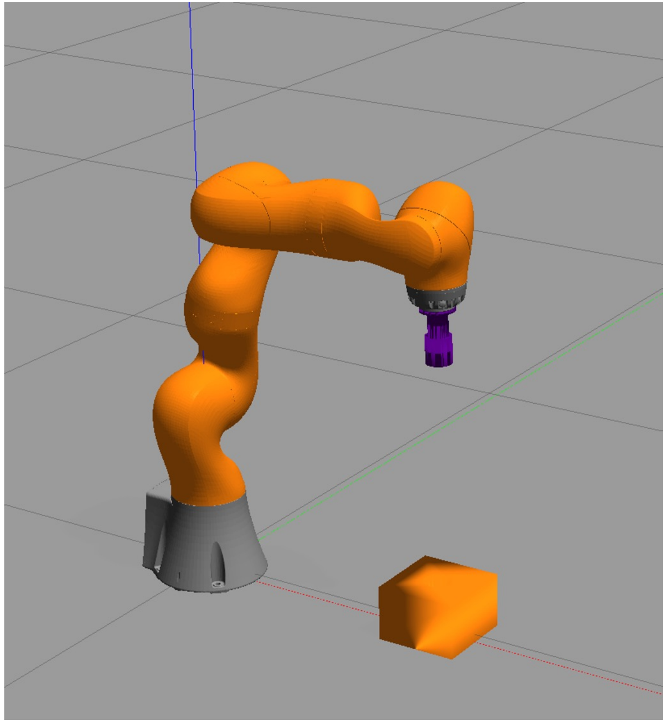


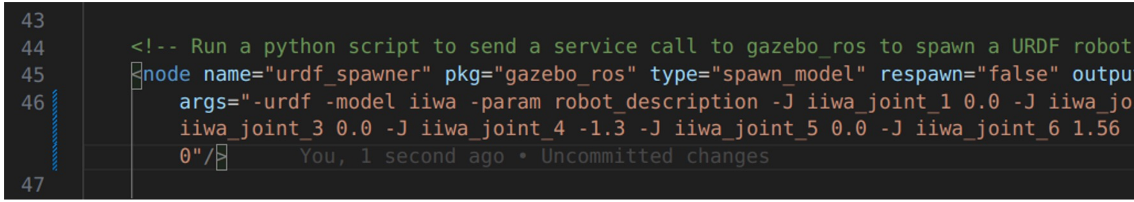


```
<!-- Setup iiwa -->
<param name="robot_description" command="
skiros2_skill_learning_demo)/res/urdf/iiw
hardware_interface:=EffortJointInterface
<!-- Spawn iiwa service -->
<include file="$(find iiwa_tools)/launch/

<!-- Spawn box in Gazebo -->
<group if="$(arg load_box)">
    <param name="/box_with_hole" textfile                    es/ur
    urdf"/>       You, 23 minutes ago • Un
    <node name="box_urdf_spawner" pkg="ga                    lse"
                args="-urdf -model box_wi
</group>
```



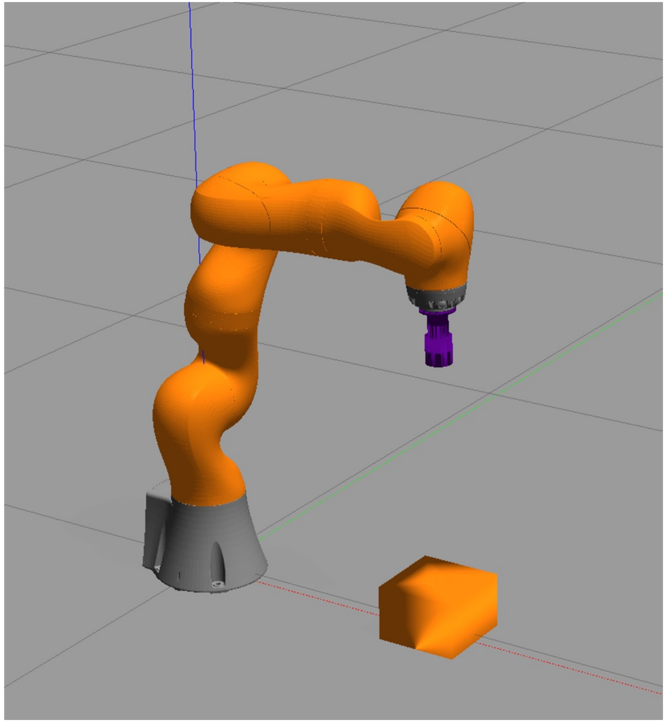
Start Configuration

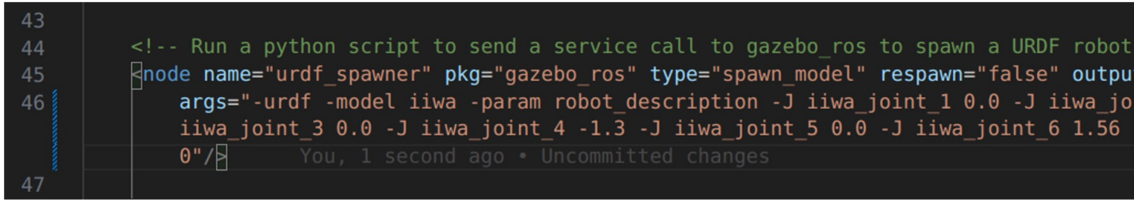




```
<!-- Run a python script to send a service call to gazebo_ros to spawn a URDF robot
<node name="urdf_spawner" pkg="gazebo_ros" type="spawn_model" respawn="false" outpu
    args="-urdf -model iiwa -param robot_description -J iiwa_joint_1 0.0 -J iiwa_jo
    iiwa_joint_3 0.0 -J iiwa_joint_4 -1.3 -J iiwa_joint_5 0.0 -J iiwa_joint_6 1.56
    0"/>      You, 1 second ago • Uncommitted changes
```

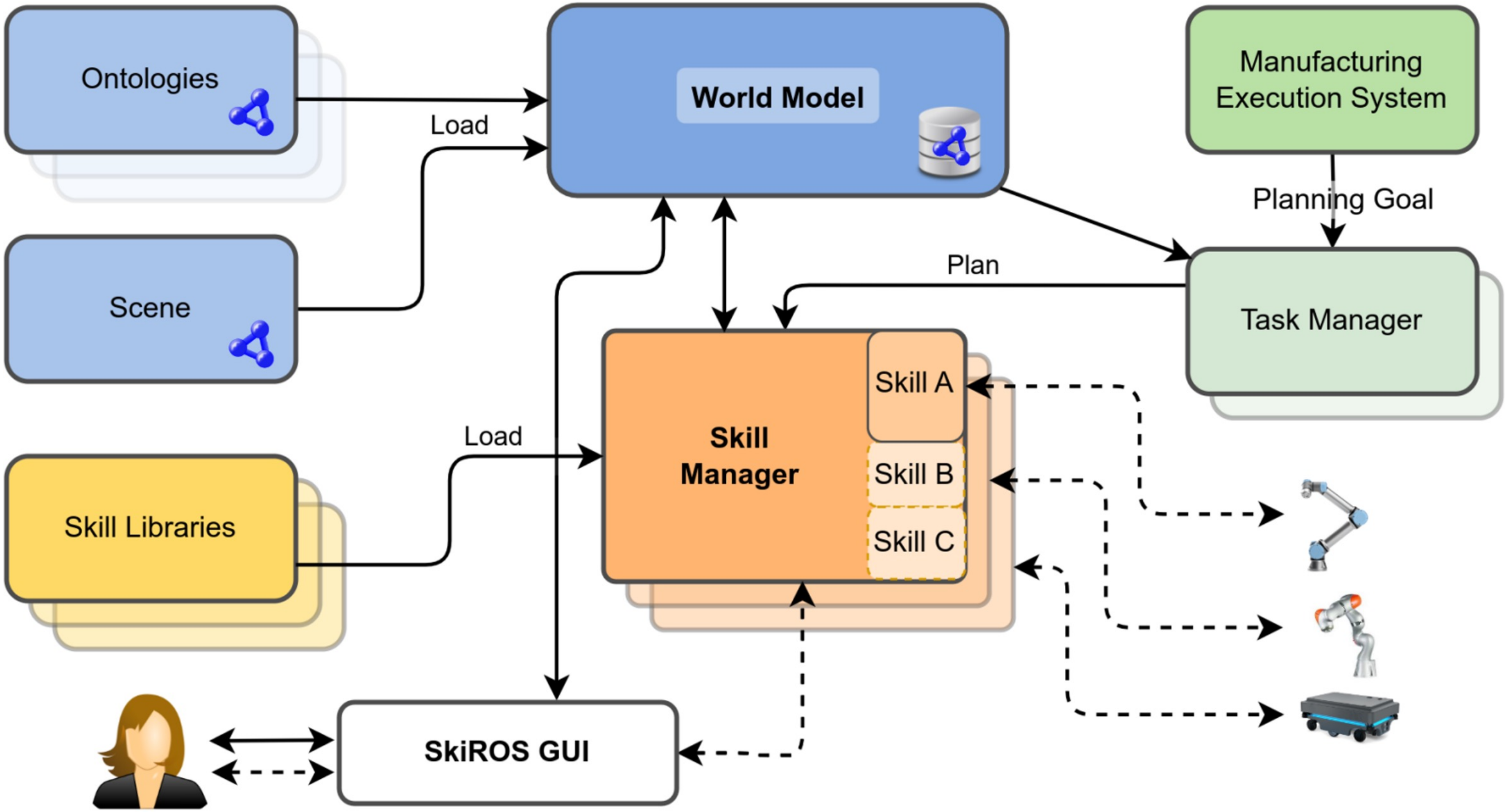
Start Configuration



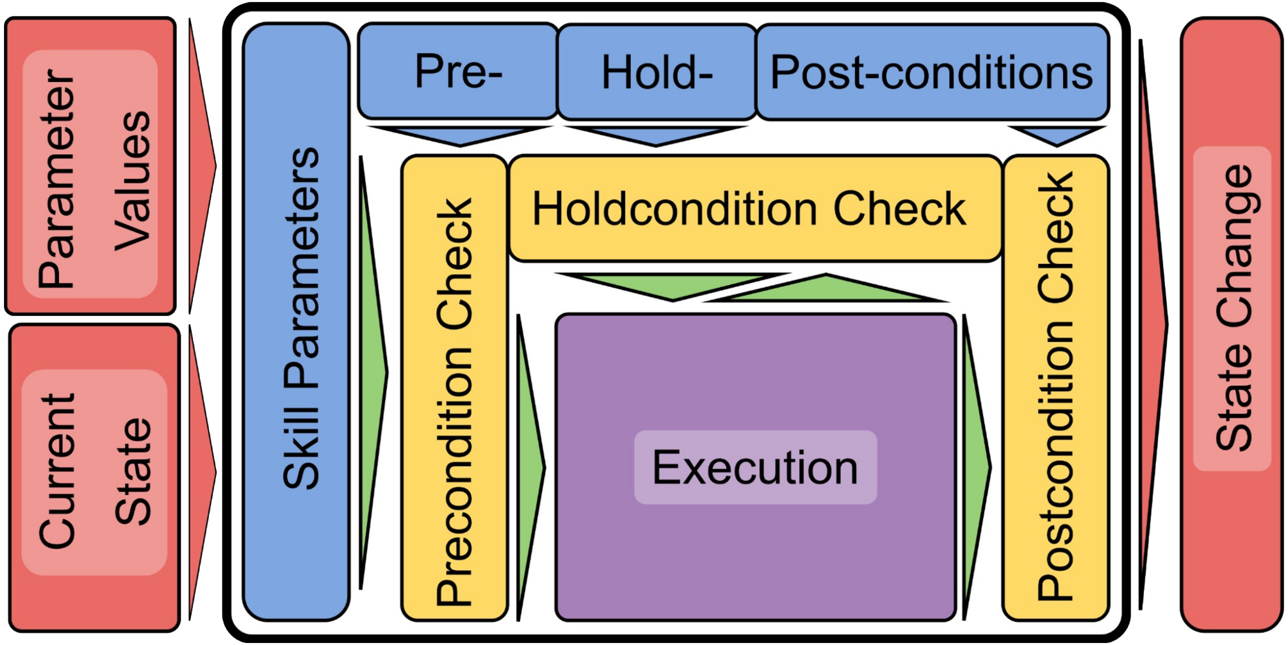


```
<!-- Run a python script to send a service call to gazebo_ros to spawn a URDF robot
<node name="urdf_spawner" pkg="gazebo_ros" type="spawn_model" respawn="false" outpu
   args="-urdf -model iiwa -param robot_description -J iiwa_joint_1 0.0 -J iiwa_jo
   iiwa_joint_3 0.0 -J iiwa_joint_4 -1.3 -J iiwa_joint_5 0.0 -J iiwa_joint_6 1.56
   0"/>      You, 1 second ago • Uncommitted changes
```

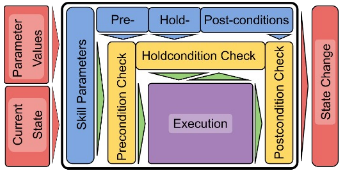
A platform for Intelligent and Autonomous Robots



Skill Model

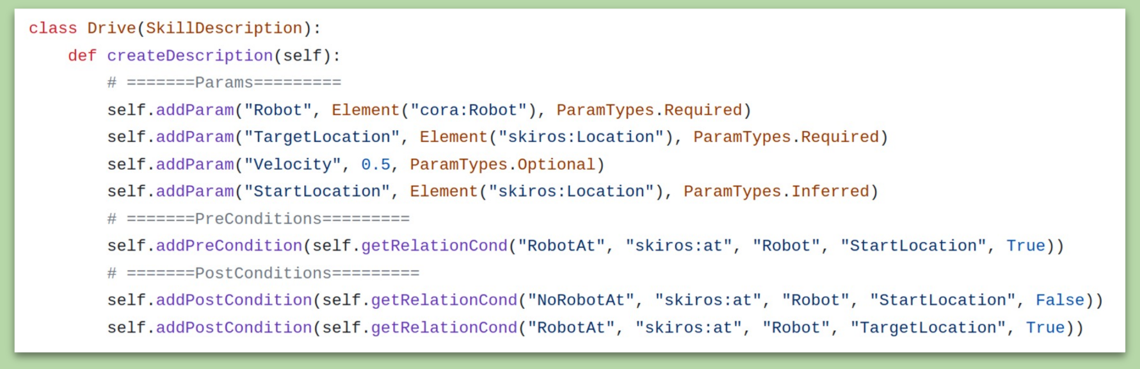

Parameter Values
Current State
Skill Parameters
Pre-
Hold-
Post-conditions
Precondition Check
Holdcondition Check
Execution
Postcondition Check
State Change

Skill Model



Skill Description

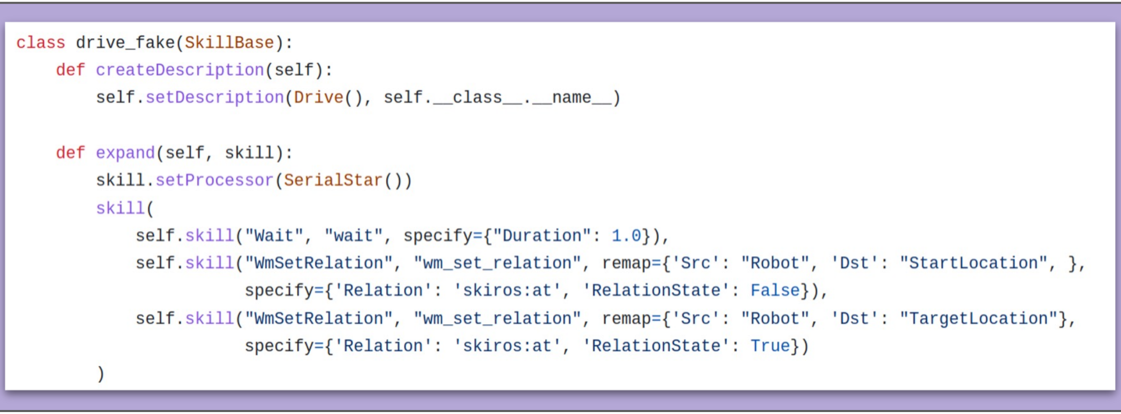
- Semantic level
- Parameters
- Pre-, hold- and post-conditions



```
class Drive(SkillDescription):
    def createDescription(self):
        # =======Params=========
        self.addParam("Robot", Element("cora:Robot"), ParamTypes.Required)
        self.addParam("TargetLocation", Element("skiros:Location"), ParamTypes.Required)
        self.addParam("Velocity", 0.5, ParamTypes.Optional)
        self.addParam("StartLocation", Element("skiros:Location"), ParamTypes.Inferred)
        # =======PreConditions=========
        self.addPreCondition(self.getRelationCond("RobotAt", "skiros:at", "Robot", "StartLocation", True))
        # =======PostConditions=========
        self.addPostCondition(self.getRelationCond("NoRobotAt", "skiros:at", "Robot", "StartLocation", False))
        self.addPostCondition(self.getRelationCond("RobotAt", "skiros:at", "Robot", "TargetLocation", True))
```

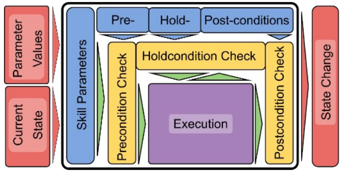
Skill Implementation

- Implements one description
- Different implementations of one description
- Can modify the description

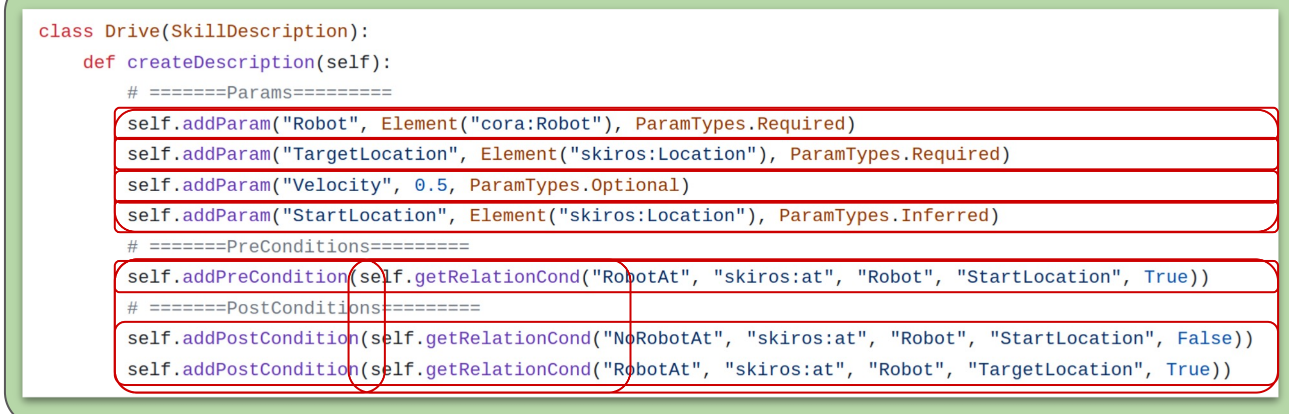


```
class drive_fake(SkillBase):
    def createDescription(self):
        self.setDescription(Drive(), self.__class__.__name__)

    def expand(self, skill):
        skill.setProcessor(SerialStar())
        skill(
            self.skill("Wait", "wait", specify={"Duration": 1.0}),
            self.skill("WmSetRelation", "wm_set_relation", remap={'Src': "Robot", 'Dst': "StartLocation", },
                       specify={'Relation': 'skiros:at', 'RelationState': False}),
            self.skill("WmSetRelation", "wm_set_relation", remap={'Src': "Robot", 'Dst': "TargetLocation"},
                       specify={'Relation': 'skiros:at', 'RelationState': True})
        )
```

Skill Description

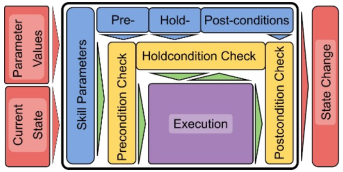


- Parameters
  1. Required
  2. Optional
  3. Inferred
- Conditions
  1. Preconditions
  2. Holdconditions
  3. Postconditions
- Condition Types
  1. Relation Condition
  2. Property Existence
  3. Property Value



```
class Drive(SkillDescription):
    def createDescription(self):
        # =======Params=========
        self.addParam("Robot", Element("cora:Robot"), ParamTypes.Required)
        self.addParam("TargetLocation", Element("skiros:Location"), ParamTypes.Required)
        self.addParam("Velocity", 0.5, ParamTypes.Optional)
        self.addParam("StartLocation", Element("skiros:Location"), ParamTypes.Inferred)
        # =======PreConditions=========
        self.addPreCondition(self.getRelationCond("RobotAt", "skiros:at", "Robot", "StartLocation", True))
        # =======PostConditions=========
        self.addPostCondition(self.getRelationCond("NoRobotAt", "skiros:at", "Robot", "StartLocation", False))
        self.addPostCondition(self.getRelationCond("RobotAt", "skiros:at", "Robot", "TargetLocation", True))
```

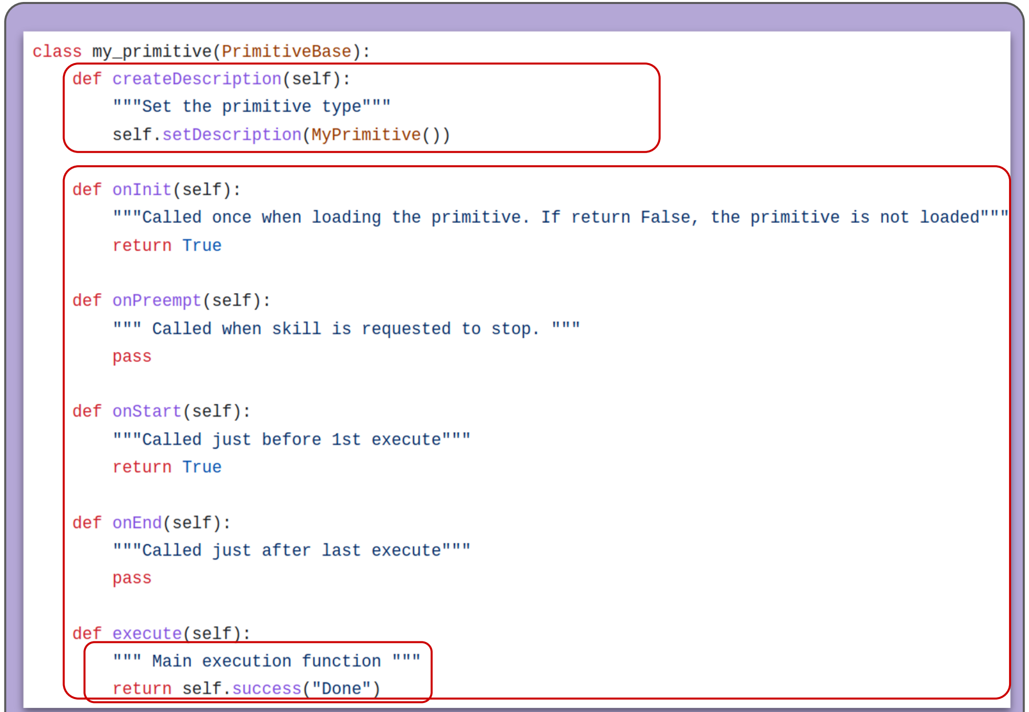
Skill Implementations: **Primitive Skills**



- Semantically atomic actions
- Typically directly interact with an API
- Examples:
  - Gripper actuation
  - Arm manipulation
  - Sensor input

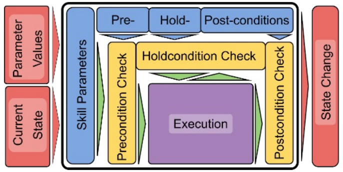
Code Skeleton:

- Implement one skill description
- Python functions for start, execution, …
- Return “running”, “success” and “failure”



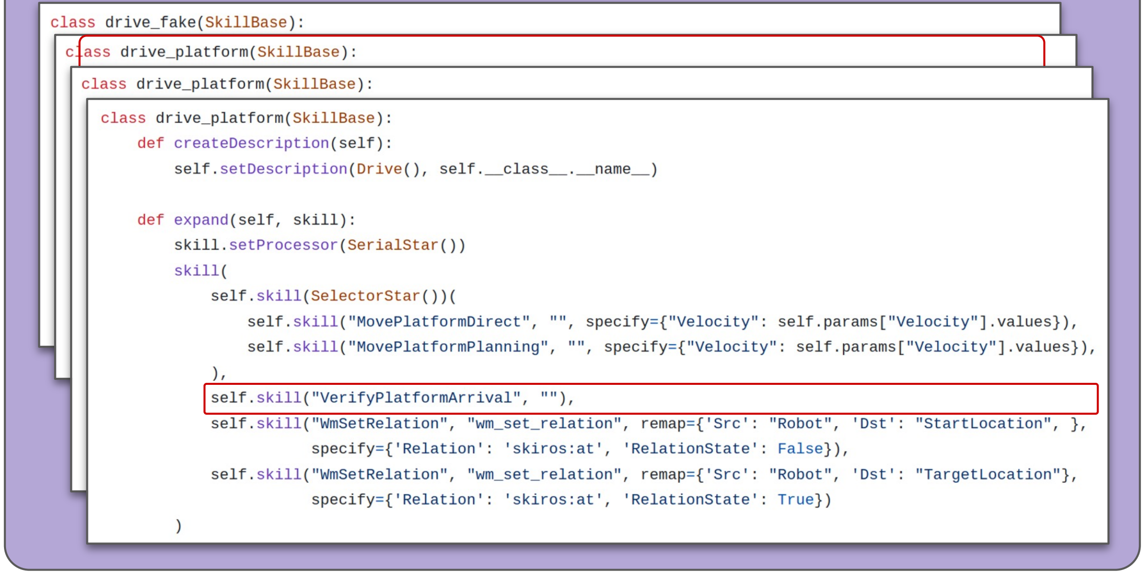
```
class my_primitive(PrimitiveBase):
    def createDescription(self):
        """Set the primitive type"""
        self.setDescription(MyPrimitive())

    def onInit(self):
        """Called once when loading the primitive. If return False, the primitive is not loaded"""
        return True

    def onPreempt(self):
        """ Called when skill is requested to stop. """
        pass

    def onStart(self):
        """Called just before 1st execute"""
        return True

    def onEnd(self):
        """Called just after last execute"""
        pass

    def execute(self):
        """ Main execution function """
        return self.success("Done")
```

Skill Implementations: **Compound Skills**



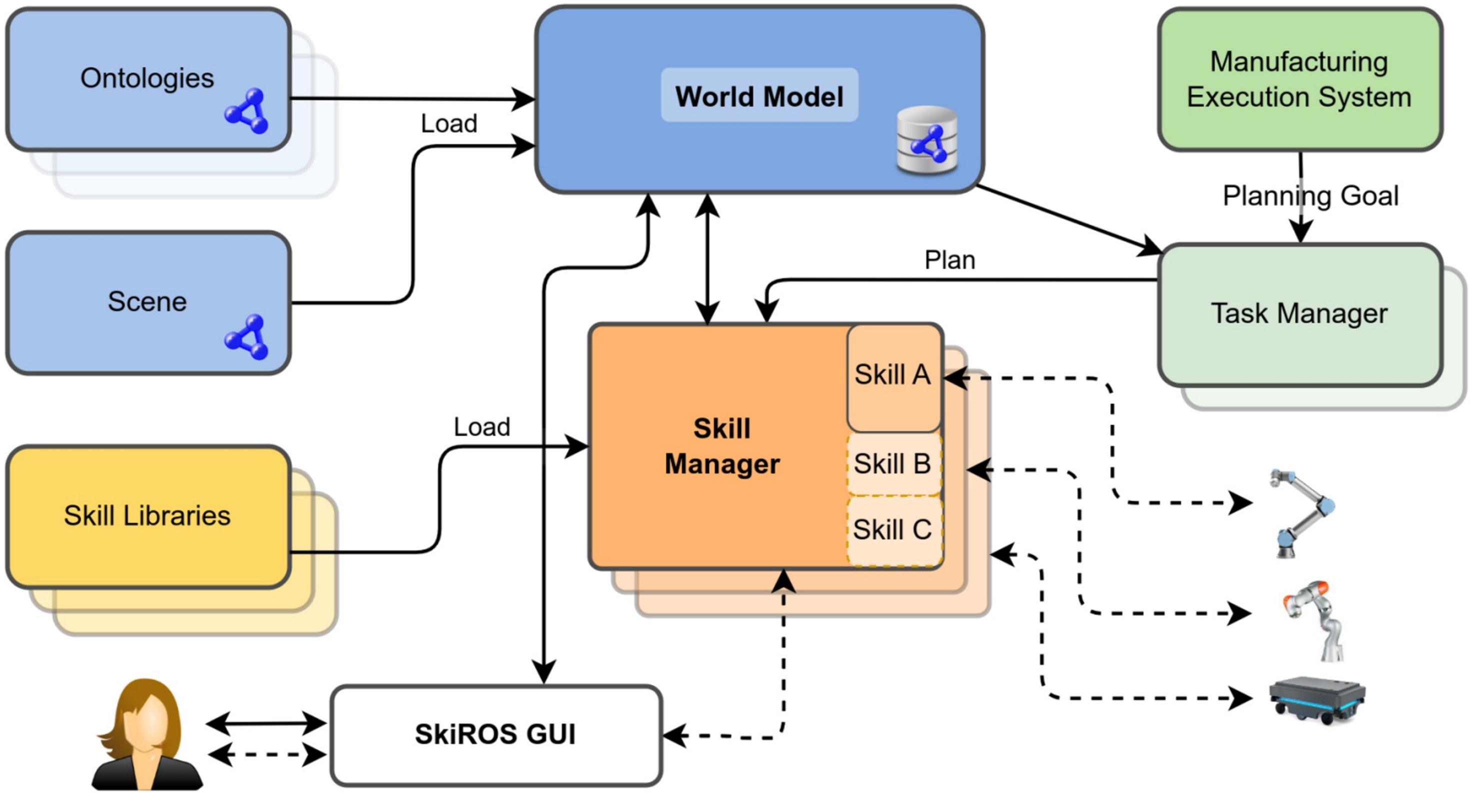
- Combine several compound skills and primitives
- *Extended Behavior trees*
- Processors
  - Serial (AND)
  - Selector (OR)
  - Parallel
  - …
- Automatic selection of implementations

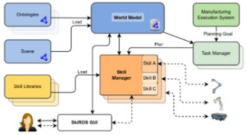
Compound Skill Implementation:



```
class drive_fake(SkillBase):
class drive_platform(SkillBase):
class drive_platform(SkillBase):
class drive_platform(SkillBase):
    def createDescription(self):
        self.setDescription(Drive(), self.__class__.__name__)

    def expand(self, skill):
        skill.setProcessor(SerialStar())
        skill(
            self.skill(SelectorStar())(
                self.skill("MovePlatformDirect", "", specify={"Velocity": self.params["Velocity"].values}),
                self.skill("MovePlatformPlanning", "", specify={"Velocity": self.params["Velocity"].values}),
            ),
            self.skill("VerifyPlatformArrival", ""),
            self.skill("WmSetRelation", "wm_set_relation", remap={'Src': "Robot", 'Dst': "StartLocation", },
                       specify={'Relation': 'skiros:at', 'RelationState': False}),
            self.skill("WmSetRelation", "wm_set_relation", remap={'Src': "Robot", 'Dst': "TargetLocation"},
                       specify={'Relation': 'skiros:at', 'RelationState': True})
        )
```

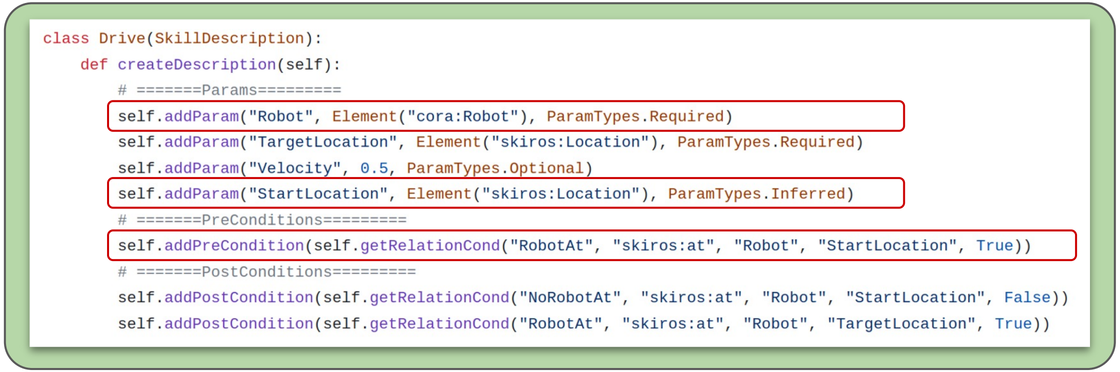
SkiROS2 Architecture





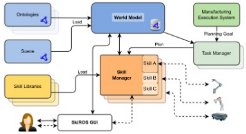
World Model

- Stores knowledge in an RDF graph
- Ontologies
  - Concepts
  - Properties
  - Relations
- Scene has concrete instances
- Enables reasoning and planning

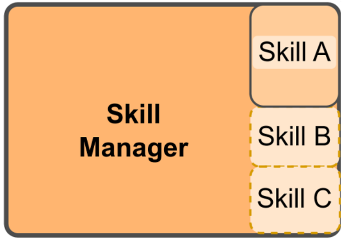


```
class Drive(SkillDescription):
    def createDescription(self):
        # =======Params=========
        self.addParam("Robot", Element("cora:Robot"), ParamTypes.Required)
        self.addParam("TargetLocation", Element("skiros:Location"), ParamTypes.Required)
        self.addParam("Velocity", 0.5, ParamTypes.Optional)
        self.addParam("StartLocation", Element("skiros:Location"), ParamTypes.Inferred)
        # =======PreConditions=========
        self.addPreCondition(self.getRelationCond("RobotAt", "skiros:at", "Robot", "StartLocation", True))
        # =======PostConditions=========
        self.addPostCondition(self.getRelationCond("NoRobotAt", "skiros:at", "Robot", "StartLocation", False))
        self.addPostCondition(self.getRelationCond("RobotAt", "skiros:at", "Robot", "TargetLocation", True))
```

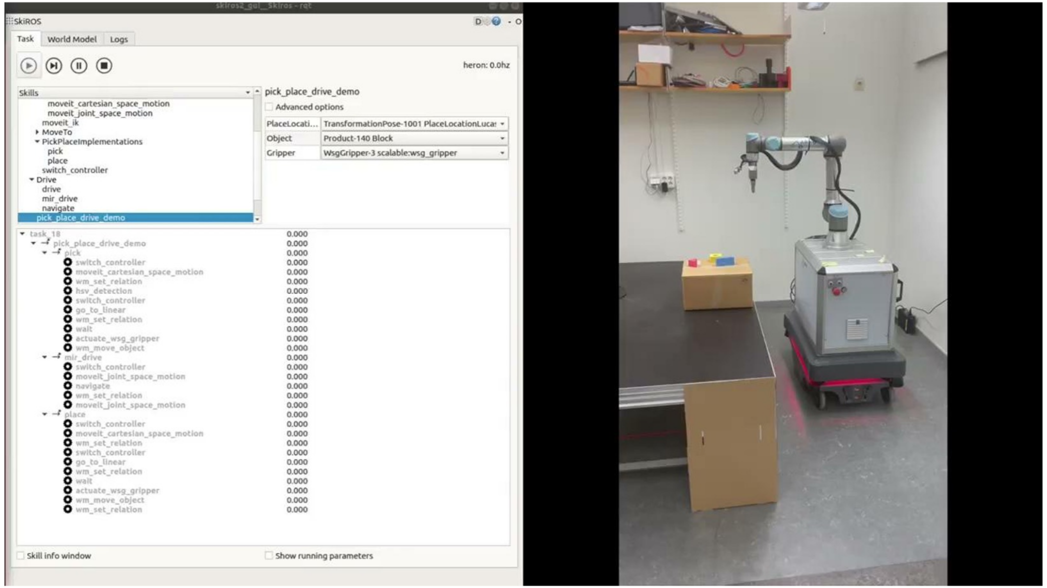
Subject	**Predicate**	**Object**
skiros:Container	rdfs:subclassOf	skiros:Location
skiros:DriverAddress	rdfs:subPropertyOf	skiros:DeviceProperty
skiros:Scene-0	skiros:contains	skiros:Location-1
skiros:Robot-2	skiros:at	skiros:Location-1



Skill Manager



- Loads skills from skill libraries
- Populates the world model with skill information
- Executes skills
  - Creates a task
  - Skills share a blackboard
  - Grounds skills
  - Automatically selects skills



```
Skills
      moveit_cartesian_space_motion
      moveit_joint_space_motion
    moveit_ik
  ▸ MoveTo
  ▾ PickPlaceImplementations
      pick
      place
    switch_controller
▾ Drive
    drive
    mir_drive
    navigate
  pick_place_drive_demo
```

pick_place_drive_demo

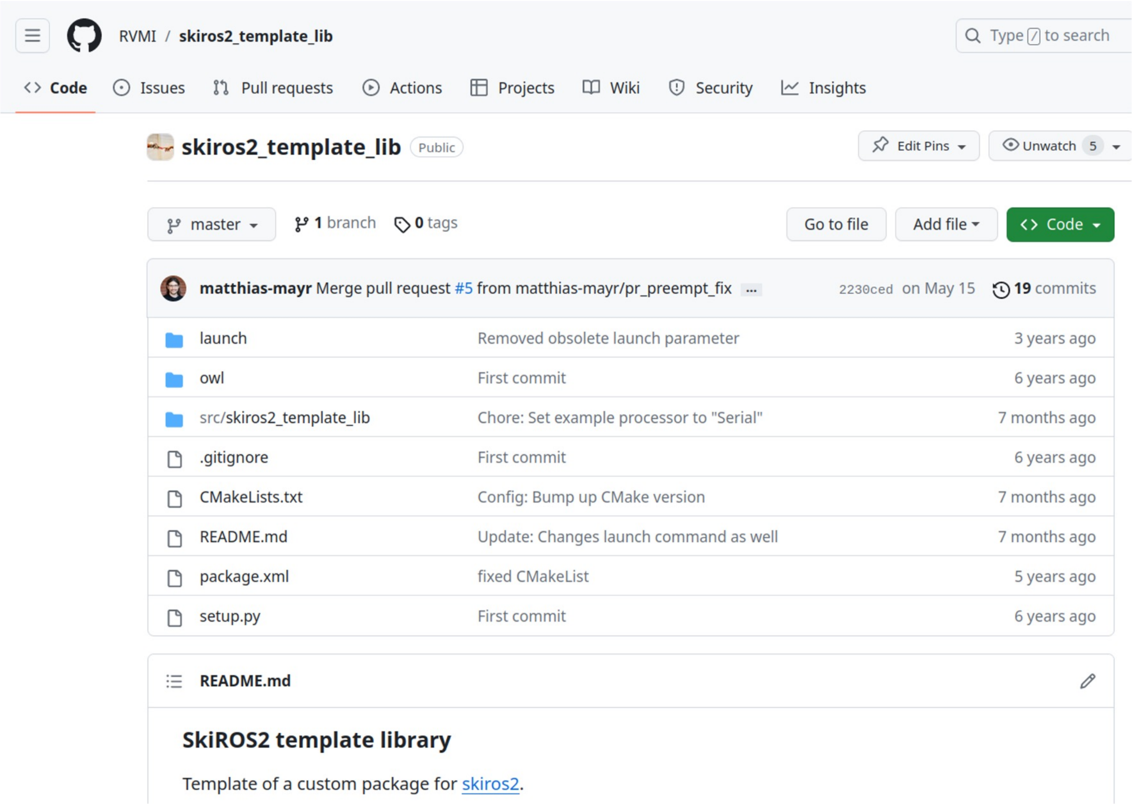
□ Advanced options

PlaceLocati...	TransformationPose-1001 PlaceLocationLuca:
Object	Product-140 Block
Gripper	WsgGripper-3 scalable:wsg_gripper

```
▾ task_18                                         0.000
  ▾ pick_place_drive_demo                         0.000
    ▾ pick                                        0.000
        switch_controller                         0.000
        moveit_cartesian_space_motion             0.000
        wm_set_relation                           0.000
        hsv_detection                             0.000
        switch_controller                         0.000
        go_to_linear                              0.000
        wm_set_relation                           0.000
        wait                                      0.000
        actuate_wsg_gripper                       0.000
        wm_move_object                            0.000
    ▾ mir_drive                                   0.000
        switch_controller                         0.000
        moveit_joint_space_motion                 0.000
        navigate                                  0.000
        wm_set_relation                           0.000
        moveit_joint_space_motion                 0.000
    ▾ place                                       0.000
        switch_controller                         0.000
        moveit_cartesian_space_motion             0.000
        wm_set_relation                           0.000
        switch_controller                         0.000
        go_to_linear                              0.000
        wm_set_relation                           0.000
        wait                                      0.000
        actuate_wsg_gripper                       0.000
        wm_move_object                            0.000
        wm_set_relation                           0.000
```

□ Skill info window □ Show running parameters

SkiROS2 Skills



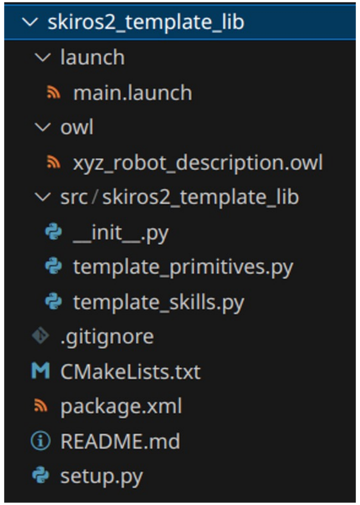
matthias-mayr Merge pull request #5 from matthias-mayr/pr_preempt_fix ... 2230ced on May 15 19 commits

launch	Removed obsolete launch parameter	3 years ago
owl	First commit	6 years ago
src/skiros2_template_lib	Chore: Set example processor to "Serial"	7 months ago
.gitignore	First commit	6 years ago
CMakeLists.txt	Config: Bump up CMake version	7 months ago
README.md	Update: Changes launch command as well	7 months ago
package.xml	fixed CMakeList	5 years ago
setup.py	First commit	6 years ago

README.md

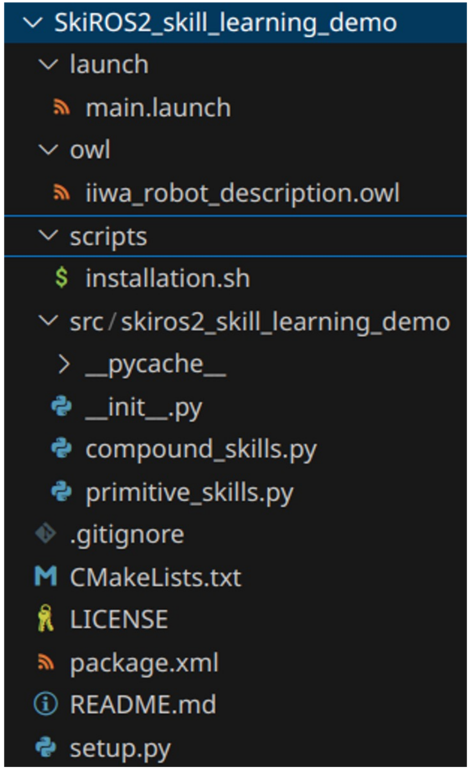
SkiROS2 template library

Template of a custom package for skiros2.

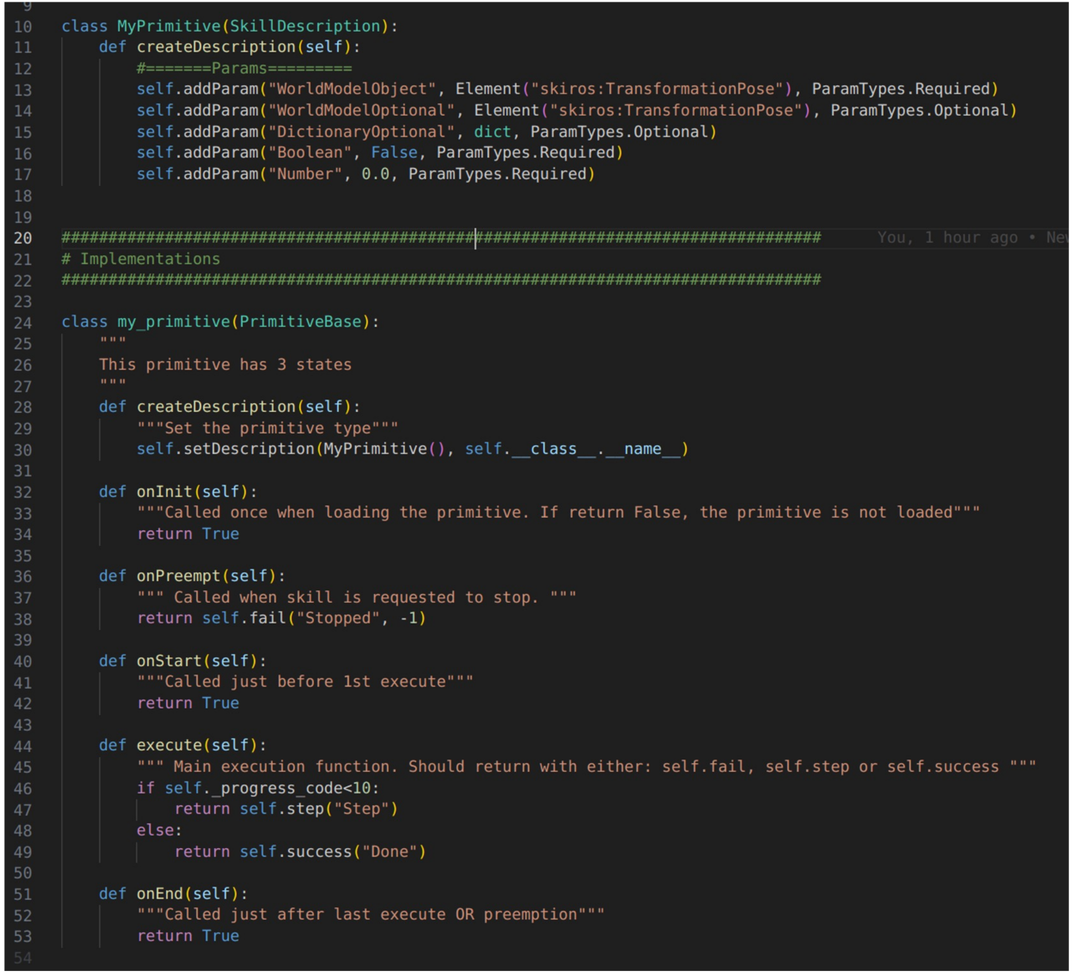


```
skiros2_template_lib
  launch
    main.launch
  owl
    xyz_robot_description.owl
  src / skiros2_template_lib
    __init__.py
    template_primitives.py
    template_skills.py
  .gitignore
  CMakeLists.txt
  package.xml
  README.md
  setup.py
```

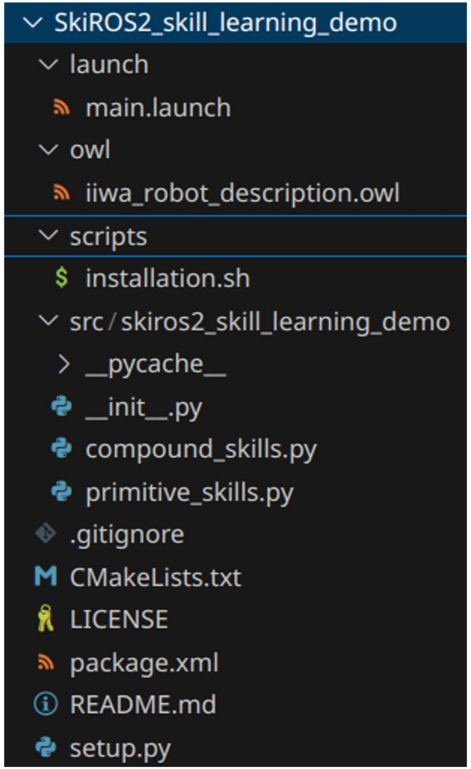
Skills



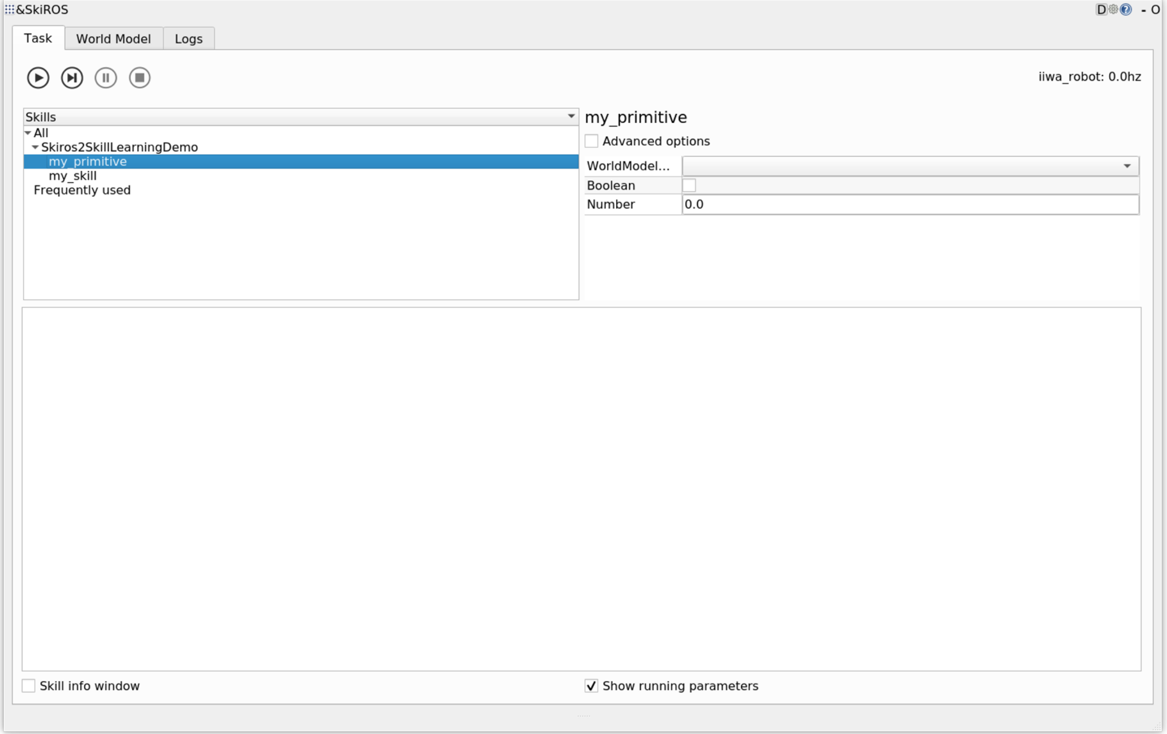
```
⌄ SkiROS2_skill_learning_demo
  ⌄ launch
      main.launch
  ⌄ owl
      iiwa_robot_description.owl
  ⌄ scripts
    $ installation.sh
  ⌄ src / skiros2_skill_learning_demo
    > __pycache__
      __init__.py
      compound_skills.py
      primitive_skills.py
  .gitignore
  CMakeLists.txt
  LICENSE
  package.xml
  README.md
  setup.py
```



```python
class MyPrimitive(SkillDescription):
    def createDescription(self):
        #=======Params=========
        self.addParam("WorldModelObject", Element("skiros:TransformationPose"), ParamTypes.Required)
        self.addParam("WorldModelOptional", Element("skiros:TransformationPose"), ParamTypes.Optional)
        self.addParam("DictionaryOptional", dict, ParamTypes.Optional)
        self.addParam("Boolean", False, ParamTypes.Required)
        self.addParam("Number", 0.0, ParamTypes.Required)


##################################################################################    You, 1 hour ago • Ne
# Implementations
##################################################################################

class my_primitive(PrimitiveBase):
    """
    This primitive has 3 states
    """
    def createDescription(self):
        """Set the primitive type"""
        self.setDescription(MyPrimitive(), self.__class__.__name__)

    def onInit(self):
        """Called once when loading the primitive. If return False, the primitive is not loaded"""
        return True

    def onPreempt(self):
        """ Called when skill is requested to stop. """
        return self.fail("Stopped", -1)

    def onStart(self):
        """Called just before 1st execute"""
        return True

    def execute(self):
        """ Main execution function. Should return with either: self.fail, self.step or self.success """
        if self._progress_code<10:
            return self.step("Step")
        else:
            return self.success("Done")

    def onEnd(self):
        """Called just after last execute OR preemption"""
        return True
```



```
SkiROS2_skill_learning_demo
  launch
    main.launch
  owl
    iiwa_robot_description.owl
  scripts
    installation.sh
  src / skiros2_skill_learning_demo
    > __pycache__
    __init__.py
    compound_skills.py
    primitive_skills.py
  .gitignore
  CMakeLists.txt
  LICENSE
  package.xml
  README.md
  setup.py
```



iiwa_robot: 0.0hz

Skills
- All
  - Skiros2SkillLearningDemo
    - my_primitive
    - my_skill
  - Frequently used

my_primitive

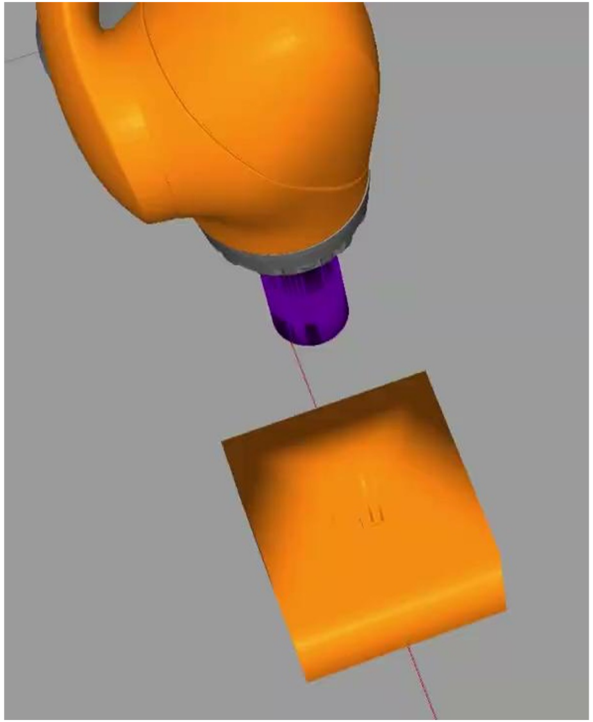
□ Advanced options

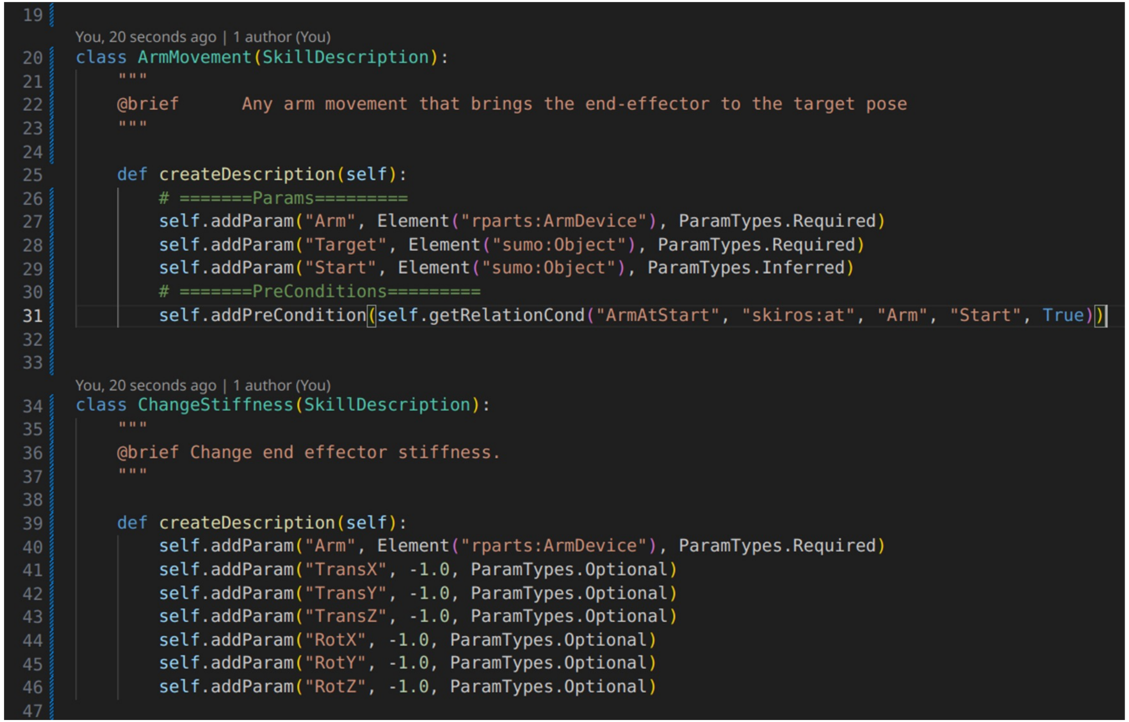
WorldModel...	
Boolean	□
Number	0.0

□ Skill info window

☑ Show running parameters

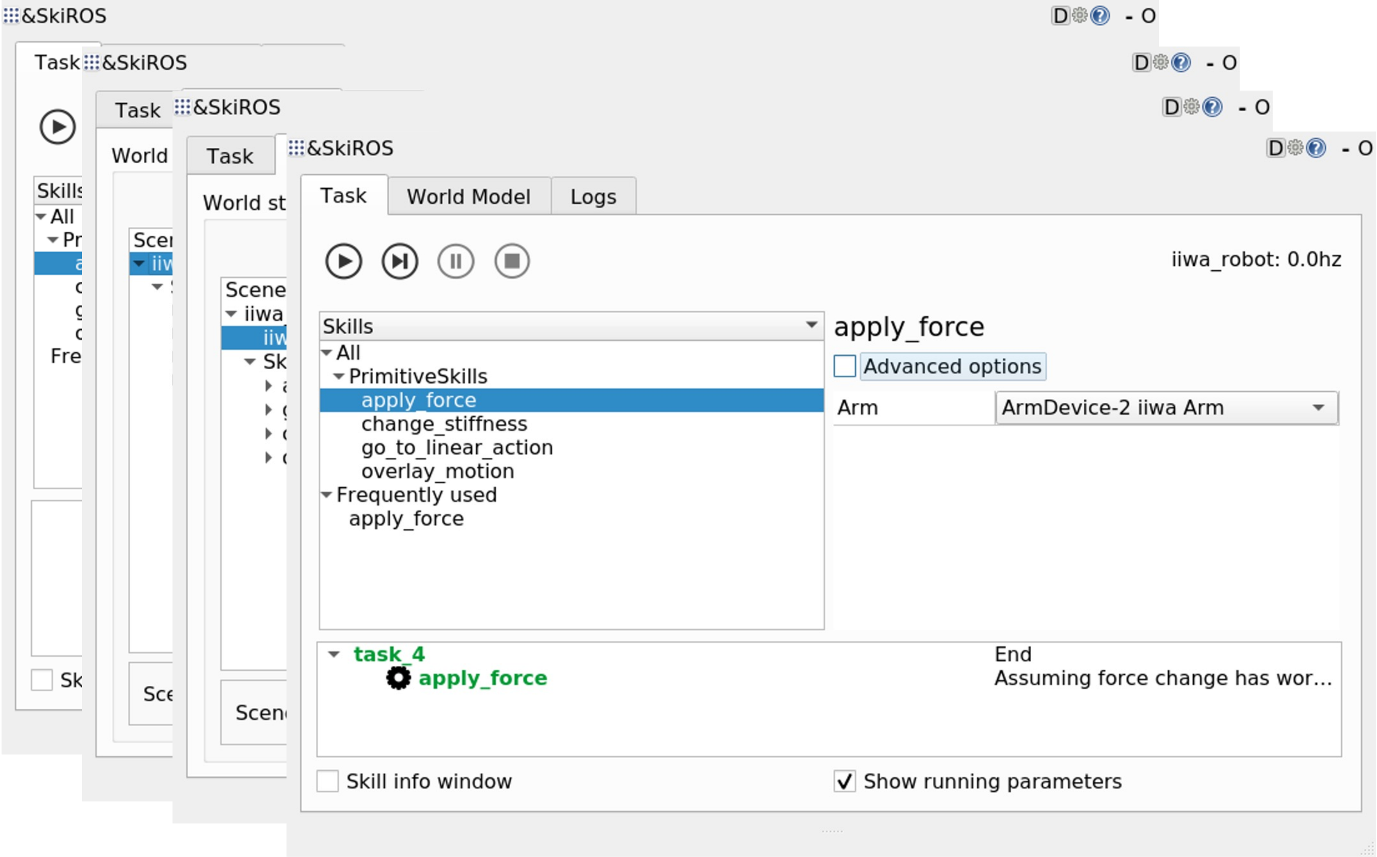
Skills





```
You, 20 seconds ago | 1 author (You)
class ArmMovement(SkillDescription):
    """
    @brief      Any arm movement that brings the end-effector to the target pose
    """

    def createDescription(self):
        # =======Params=========
        self.addParam("Arm", Element("rparts:ArmDevice"), ParamTypes.Required)
        self.addParam("Target", Element("sumo:Object"), ParamTypes.Required)
        self.addParam("Start", Element("sumo:Object"), ParamTypes.Inferred)
        # =======PreConditions=========
        self.addPreCondition(self.getRelationCond("ArmAtStart", "skiros:at", "Arm", "Start", True))


You, 20 seconds ago | 1 author (You)
class ChangeStiffness(SkillDescription):
    """
    @brief Change end effector stiffness.
    """

    def createDescription(self):
        self.addParam("Arm", Element("rparts:ArmDevice"), ParamTypes.Required)
        self.addParam("TransX", -1.0, ParamTypes.Optional)
        self.addParam("TransY", -1.0, ParamTypes.Optional)
        self.addParam("TransZ", -1.0, ParamTypes.Optional)
        self.addParam("RotX", -1.0, ParamTypes.Optional)
        self.addParam("RotY", -1.0, ParamTypes.Optional)
        self.addParam("RotZ", -1.0, ParamTypes.Optional)
```

Skills



iiwa_robot: 0.0hz

- All
  - PrimitiveSkills
    - apply_force
    - change_stiffness
    - go_to_linear_action
    - overlay_motion
  - Frequently used
    - apply_force

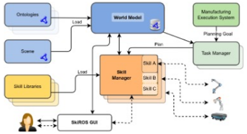
apply_force

□ Advanced options

Arm	ArmDevice-2 iiwa Arm

task_4	End
apply_force	Assuming force change has wor…

□ Skill info window ☑ Show running parameters



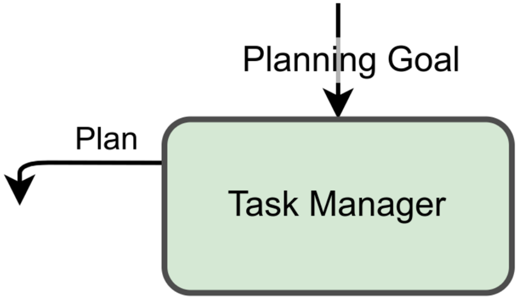
Task Manager for Task-Level Plans

- Receives planning goal such as

  `(skiros:at skiros:Robot-2 skiros:Location-3)`

- Automatically creates a PDDL planning domain
  - Based on the knowledge in the world model

- Uses a PDDL planner (tfd)
- Execution in the skill manager

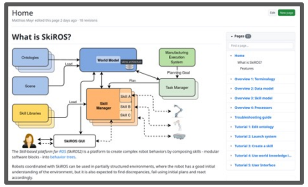


SkiROS2 - Summary

- Flexible robot control platform
- Targeted for semi-structured environments
- Knowledge integration and reasoning
- Automatic task-level planning
- Behavior trees
- Reinforcement learning
- ROS 2 support

What will you do with it?

Documentation:



Home

Matthias Mayr edited this page 2 days ago · 18 revisions

What is SkiROS?

The *Skill-based platform for ROS (SkiROS2)* is a platform to create complex robot behaviors by composing *skills* - modular software blocks - into behavior trees.

Robots coordinated with SkiROS can be used in partially structured environments, where the robot has a good initial understanding of the environment, but it is also expected to find discrepancies, fail using initial plans and react accordingly.

https://github.com/RVMI/SkiROS2/wiki

{Code}:

https://github.com/RVMI/SkiROS2

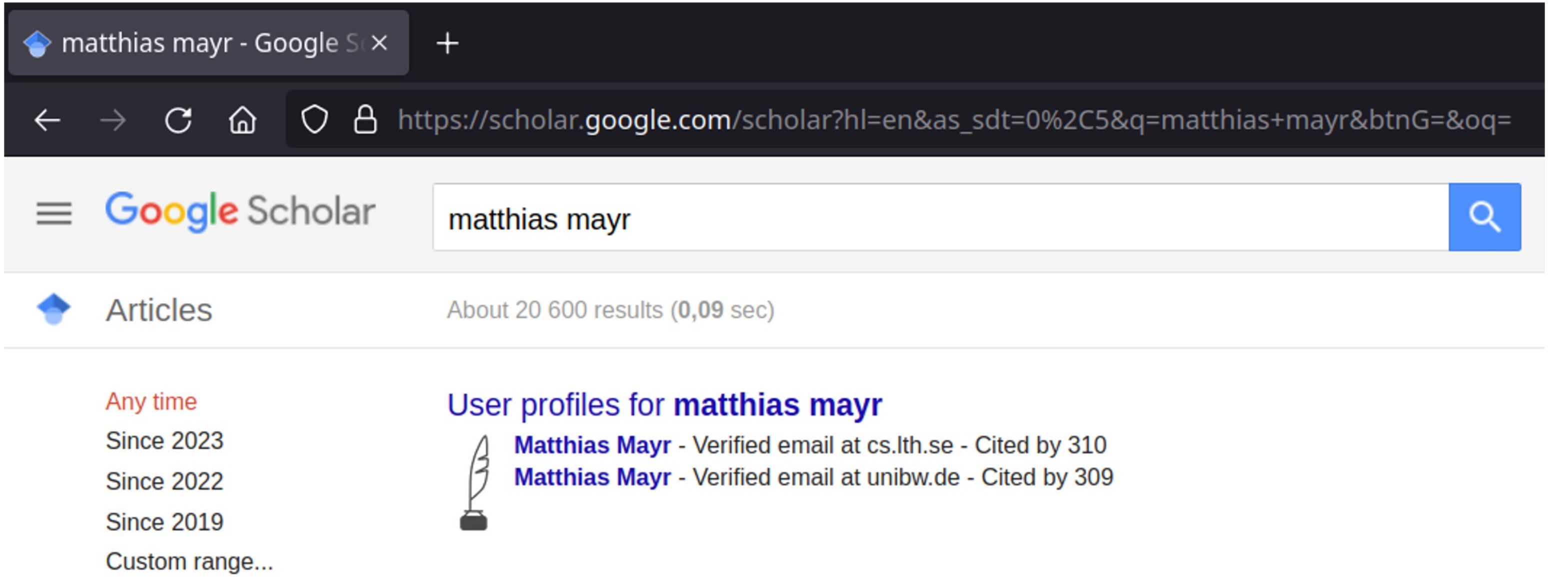
Paper:

SkiROS2: A skill-based Robot Control Platform for ROS

Matthias Mayr[1], Francesco Rovida[2] and Volker Krueger[1]

arXiv:2306.17030v1 [cs.RO] 29 Jun 2023

SkiROS2: A skill-based robot control platform for ROS
https://arxiv.org/abs/2306.17030



matthias mayr - Google Sc
https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=matthias+mayr&btnG=&oq=
Google Scholar

matthias mayr

Articles

About 20 600 results (**0,09** sec)

Any time

Since 2023

Since 2022

Since 2019

Custom range...

User profiles for **matthias mayr**

Matthias Mayr - Verified email at cs.lth.se - Cited by 310

Matthias Mayr - Verified email at unibw.de - Cited by 309