1. Theoretical background

Oficial documentation

- Definition:
  - A parameter is a configuration value or setting of a node.
  - They can be integers, floats, booleans, strings and lists.
  - If the node stops then the parameters values are lost.
- They are used to modify internal values of a node while executing to change its behaviour.
  - For instance, the subscription to an specific node, the name of a device to establish a connection, the speed of a communication, etc.
- Parameters can be set:
  - Through the CLI when the node is running
  - At the beginning through some extra commands or a launch file

2. Python parameters

2.1 Structure

1. Import the related **modules and classes**
2. Declare the parameter in the **constructor**
3. Create a **callback** method

2.2 Methods

- Modules and classes to import

```
from rclpy.parameter import Parameter
from rcl_interfaces.msg import ParameterType,
ParameterDescriptor, SetParametersResult
```

- Declaration: Parameters **must always be declarated** to be binded to rclpy

```
Node.declare_parameter(<'parameter_name'>,
                                            <default value>,

ParameterDescriptor(description=<'message'>)
)
```

- Get a parameter value

```
value = Node.get_parameter(<'parameter_name'>).value
```

- Override a parameter
    - It is necesary to Instanciate a non-declared parameter
    - It is used to **set the value** of a aparameter

```
# Non-declared parameters
prm1 = Parameter(<'name'>, Parameter.Type.<TYPE>,
<new_value>)
prm2 = Parameter(<'name'>, Parameter.Type.<TYPE>,
<new_value>)
# TYPE: STRING, INTEGER, DOUBLE, BOOL, STRING_ARRAY,
BYTE_ARRAY, ...

# Override all the previously declared parameters
Node.set_parameters([prm1, prm2, ...])
```

- Bind a parameter callback

```
Node.add_on_set_parameters_callback(<callback>)
```

2.3 Callbacks

- Arguments
    - parameters: A list of all the declared parameters
- Returns:
    - SetParameterResult: Indicates that the callback was executed successfully

```
def callback(self, parameters):
        ...
        return SetParameterResult(successful=True)
```

3. C++ parameters

- To be written...

4. CLI tools

1. Get the current value

```
ros2 param get <node_name> <parameter_name>
```

2. Modify at the begining the execution

```
ros2 run <package_name> <node_name> --ros-args -p
<parameter_name>:=<value>
```

3. Modify during execution

```
ros2 param set <node_name> <parameter_name> <value>
```

Alternative sources

Robotics backend