

Gazebo ထဲမှာ Robot Model ကို control လုပ်ဖို့အတွက် ROS controller တွေ လိုအပ်ပါတယ် ၊
ရှင်းအောင်ပြောရရင် arm မှာပဲဖြစ်ဖြစ် mobile robot မှာပဲဖြစ်ဖြစ် model ရဲ့ joint တွေကို control
လုပ်ဖို့အတွက် controller ရှိဖို့လိုတယ်လို့ ပြောချင်တာပါ ။ joint တွေကို control လုပ်ဖို့ဆိုရင် transmission tag
ထဲမှာ သတ်မှတ်ထားတဲ့ hardware interface နဲ့ အဆင်ပြေမယ့် controller တွေကို လိုအပ်ပါတယ်။

ROS Controller တွေကြောင်း စပြောကြရအောင်

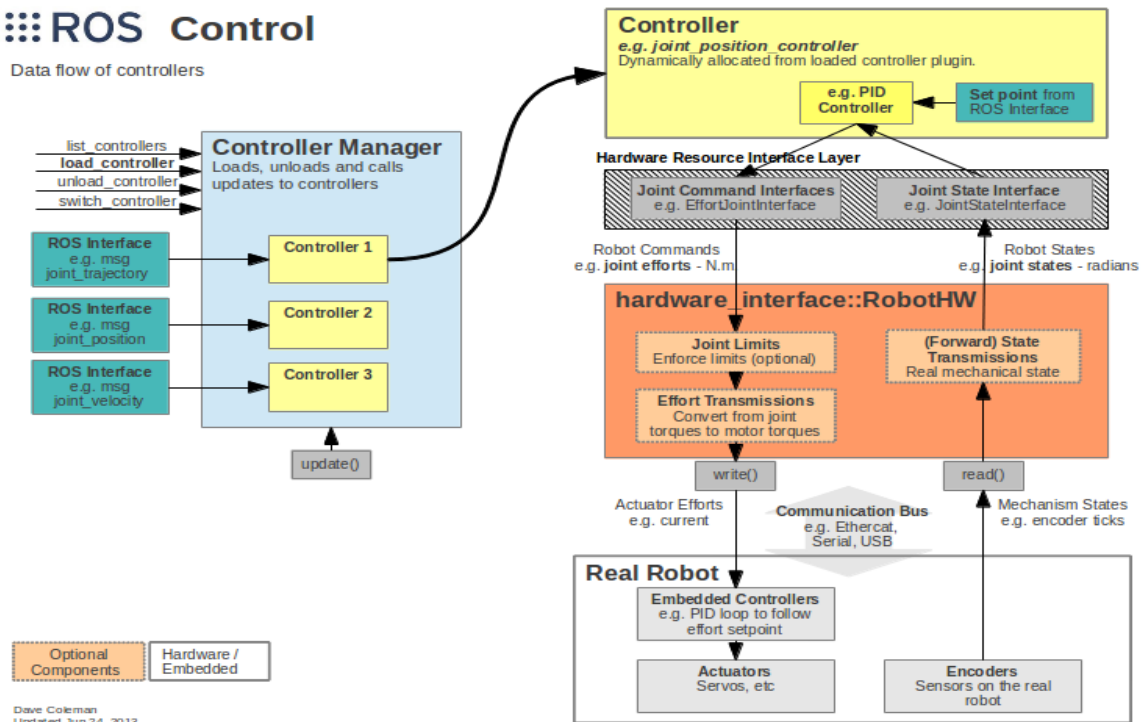
Controller ဆိုတာကတော့ မိမိ robot ရဲ့ joint ကို position ၊ velocity ၊ effort(force or torque) ၊ joint states ၊
joint trajectory စသည်တို့ကို control လုပ်ပေးတဲ့အရာပါ ။ position ကို control လုပ်ဖို့ဆို
joint_position_controller ဆိုတဲ့ package လေးရှိတယ်ပေါ့ ၊ အပေါ်မှာပြောခဲ့တဲ့ အရာတွေလည်း
ထိုနည်းဌာနပါပဲ ။ မြင်အောင် ထပ်ပြီး ဥပမာပေးရရင် PID Controller ပေါ့ ။

*ROS Controller တွေက hardware ကို တိုက်ရိုက် control လုပ်လို့မရပါဖူး ၊ ကြားခံ mediator အနေနဲ့ controller
ကနေ real hardware သို့ fake hardware(in simulation) ကို ဆက်သွယ်ဖို့ interface တစ်ခုလိုအပ်ပါတယ် ၊
controller ကနေ ပေးလာတဲ့ command ကို receive လုပ်ပြီး hardware ဆီကို send လုပ်ပေးနိုင်တဲ့ interface ပေါ့
(Note – Vice Versa) ။ ခုနက အပေါ်မှာပြောခဲ့တဲ့ transmission tag ထဲက hardware interface ဆိုလိုတာပါ။*

မိတ်ဆက်အနေနဲ့ မြင်အောင်ပြောပြတာပါ ။ အောက်ကပုံကို ကြည့်လိုက်ရင် နည်းနည်းမြင်သာသွားပါလိမ့်မယ် ။

ROS Control

Data flow of controllers



Controller အမျိုးအစားတွေနဲ့ Hardware Interface အမျိုးအစားတွေအကြောင်း ဆက်ရှင်းပါမယ် ။

Types of ROS Controllers

- effort controllers** - မိမိ robot ရဲ့ joints တွေဆီ desired force or torque ကို command ပေးနိုင်တဲ့ controllers တွေပါ ။ command ပေးတဲ့ပုံစံအပေါ်မူတည်ပြီးတော့လည်း သုံးရတဲ့ package တွေက ကွဲပြားပါတယ် ။ အောက်မှာဆက်ပြောမယ့် controller package တွေနဲ့ မရောသွားစေဖို့ မှတ်ထားရမှာက ဒီ controller အမျိုးအစားမှာပါတဲ့ ဘယ်လို controller ပဲ ဖြစ်ဖြစ် သူ့ရဲ့ နောက်ဆုံး desired output က force or torque ဖြစ်တယ်ဆိုတာပါ ။ ငါးခုရှိပါတယ် ။ အောက်မှာကြည့်ပါ ။

- `joint_effort_controller`
- `joint_group_effort_controller`
- `joint_group_position_controller`
- `joint_position_controller`
- `joint_velocity_controller`

2. **position_controllers** - မိမိ robot ရဲ့တစ်ခု သို့ တစ်ခုထက်ပိုတဲ့ joint တွေရဲ့ position တွေကို သတ်မှတ်ပေးလို့ရတဲ့ controller တွေပါ။ နှစ်ခုရှိပါတယ် ။ အောက်မှာကြည့်ပါ ။

- joint_position_controller
- joint_group_position_controller

3. **velocity_controllers** - မိမိ robot ရဲ့တစ်ခု သို့ တစ်ခုထက်ပိုတဲ့ joint တွေရဲ့ velocity တွေကို သတ်မှတ်ပေးလို့ရတဲ့ controller တွေပါ ။ နှစ်ခုရှိပါတယ် ။ အောက်မှာကြည့်ပါ ။

- joint_velocity_controller
- joint_group_velocity_controller

4. **joint_state_controller** - မိမိ robot ရဲ့ states ကို publisher လုပ်ပေးတဲ့ controller ပါ။ တစ်ခုပဲရှိပါတယ် ။ သူကိုယ်တိုင်ပါပဲ ။

- joint_state_controller

5. **joint_trajectory_controller** - မိမိ robot ရဲ့ joint သွားမယ့် trajectory (path+time) ကို သတ်မှတ်ပေးတဲ့ controller ပါ ။ ငါးခုရှိပါတယ် ။ အောက်မှာကြည့်ပါ ။

- position_controller
- velocity_controller
- effort_controller
- position_velocity_controller
- position_velocity_acceleration_controller
-

အထက်ပါ Controller တွေကို အသုံးချဖို့အတွက် Plugins တွေကို ဒီ Link မှာ (https://github.com/ros-controls/ros_controllers) ဒေါင်းလိုလည်းရသလို terminal ကနေလည်း install လို့ရကြောင်းပါ ။

Enter following line in terminal to install ,

```
sudo apt-get install ros-noetic-ros-control ros-noetic-ros-controllers
```

Types of Hardware Interface

1. **Joint Command Interface** - ဒီ Interface ကတော့ controller ကနေ command ပေးလိုက်တဲ့ command အမျိုးအစားပေါ်မူတည်ပြီး hardware joint တွေဆီကို send လုပ်ပေးပါတယ်။ command အမျိုးအစားဆိုတာကတော့ အောက်မှာပါတဲ့ သုံးခုပေါ့။
 - **Effort Joint Interface** - for commanding effort-based joints.
 - **Velocity Joint Interface** - for commanding velocity-based joints.
 - **Position Joint Interface** - for commanding position-based joints.
2. **Joint State Interfaces** - ဒီ Interface ကတော့ hardware ဆီကနေ feedback ပြန်ရမယ့် Joint States တွေကို controller ဆီ ပြန်ပို့ပေးတာပါ။ ဥပမာနဲ့ မြင်အောင်ပြောရရင် PID Controller ပါ။ controller နဲ့ သုံးမယ့် real or fake hardware ကြား loop ဖြစ်နေတာပါ။

ဒီ အောက်က interface လေးခုကတော့ မိမိသုံးတဲ့အပေါ်မူတည်ပြီး ကွဲပြားမှာဖြစ်တဲ့အတွက် အသေးစိတ်မပြောတော့ပါဘူး (ပျင်းလို့) ၊ ကျနော် အနေနဲ့ကလည်း အရမ်းကြီးလည်း သုံးတာ သိပ်မတွေ့ဖူးလို့ပါ။

3. **Actuator State Interfaces**
4. **Actuator Command Interfaces**
 - **Effort Actuator Interface**
 - **Velocity Actuator Interface**
 - **Position Actuator Interface**
5. **Force-torque sensor Interface**
6. **IMU sensor Interface**

အမှန်တော့ ကျနော် ခုပြောခဲ့တဲ့ Types of Controllers နဲ့ Types of Hardware Interface တွေက အပေါ်ကပြခဲ့တဲ့ ပုံကို နားလည်တာနဲ့ အကုန်လုံး ရှင်းရှင်းလင်းလင်းနဲ့ကို သိသွားမှာပါ။

အပေါ်မှာ သေချာပြောပြခဲ့တဲ့ ROS controller နဲ့ Hardware Interface တွေ ကို Gazebo မှာ အသုံးပြုနိုင်ဖို့ဆို ROS မှာရှိတဲ့ **ros_control package** ကို နားလည်ဖို့လိုပါတယ်။

ros_control Package

ros_control ဆိုတာကတော့ metapackage ကြီးပါ။ သူ့ထဲမှာ တခြားသော packages တွေ ပါဝင်ပါတယ်။

1. **control_toolbox** - ဒီ Package ကတော့ controller တွေရေးတဲ့အခါ သုံးတဲ့ tools တွေ ပါဝင်ပါတယ်။
2. **controller_interface** - ဒီ Package ကတော့ controller တွေအတွက် interface base class ပါဝင်ပါတယ်။
3. **controller_manager** - ဒီ Package ကတော့ controller တွေကို load,unload,start,stop လုပ်တာပါ။
။ရှင်းအောင်ပြောရရင် controller တွေကို ခိုင်းတဲ့ package ပါ။
4. **controller_manager_msgs** - ဒီ Package ကတော့ controller_manager အတွက် လိုအပ်တဲ့ message တွေ service တွေကို provide လုပ်ပေးပါတယ်။
5. **hardware_interface** - ဒီ package ကတော့ အပေါ်မှာ ရှင်းပြခဲ့ပြီးလို့ သိပြီထင်ပါတယ် ၊ controller တွေက command ပေးတာကို real hardware သို့ fake hardware(in simulation) ကိုလုပ်ဆောင်နိုင်ဖို့ ကြားခံအဖြစ် ဆောင်ရွက်ပေးတဲ့ Package ပါ။ (**Note – Vice Versa**)
6. **transmission_interface** - ဒီ Package ကတော့ hardware interface အတွက်လိုအပ်တဲ့ interface classes တွေပါဝင်ပါတယ်။

Package တွေများလို့ စိတ်ညစ်မသွားပါနဲ့ ၊ ကျနော်က ပြည့်ပြည့်စုံစုံဖြစ်အောင်လို့ ပြောပြတာပါ။

Coding မှာအဓိကသုံးမယ့် controllers တွေနဲ့ဆိုင်တဲ့ အရေးကြီးတဲ့ Package တွေကို ပြောပါမယ် ။
အောက်မှာကြည့်ပါ ။

ROS Controller Packages

1. **joint_position_controller** - ဒီ Package ကိုရှင်းမယ်ထင်ပါတယ် ၊ အပေါ်မှာပြောခဲ့တဲ့ Types of Controllers မှာပါတဲ့ **position_controllers** အမျိုးအစားပါ ။
2. **joint_state_controller** - ဒီ Package ကိုရှင်းမယ်ထင်ပါတယ် ၊ အပေါ်မှာပြောခဲ့တဲ့ Types of Controllers မှာပါတဲ့ **joint_state_controller** အမျိုးအစားပါ ။
3. **joint_effort_controller** - ဒီ Package ကိုရှင်းမယ်ထင်ပါတယ် ၊ အပေါ်မှာပြောခဲ့တဲ့ Types of Controllers မှာပါတဲ့ **efforts_controllers** အမျိုးအစားပါ ။

Program ပိုင်းကို သေသေချာချာ နားလည်နိုင်ဖို့အတွက် လိုအပ်တဲ့ Information မှန်သမျှကို
ရှင်းပြပြီးပြီဖြစ်တဲ့အတွက် Gazebo မှာ မိမိတို့ Robot Model ကို ဘယ်လို Control လုပ်မလဲဆိုတာကို
ဆက်ရှင်းပါမယ်