



# ROS Advanced

Anas Abou Allaban

([abouallaban.a@husky.neu.edu](mailto:abouallaban.a@husky.neu.edu))



Northeastern University

*Student Branch Chapter*

# Agenda

- Launch files
- Configuration files
- Arm Control

# Refresher

- Publisher/Subscriber
- Services
- Actions
- Messages

# Refresher

- Execute a file: `roslaunch <package_name> <file_name>`
- Mark file as executable: `chmod +x <file_name>`
- Refresh terminal: `source ~/.bashrc`
- Build packages: `catkin build`
- View topics: `rostopic list`
- Show topic data: `rostopic echo <topic_name>`

# Launch files

- Launch multiple nodes and other launch files at the same time.
- Uses XML (eXtensive Markup Language) ([Documentation](#))
- Set global parameters available to whole network
- Syntax: `roslaunch <package_name> <file_name.launch>`

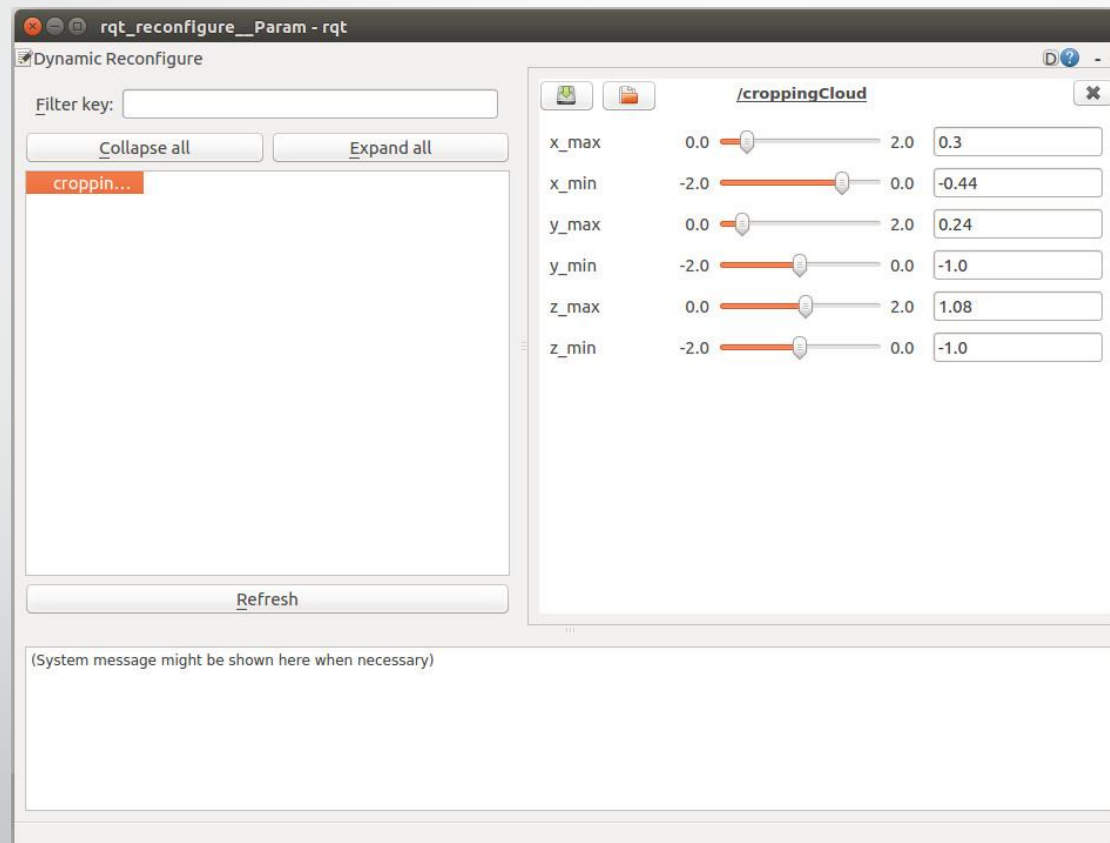
Create a directory called `launch` in your package.

Create a file called `ik.launch`.

# Launch files

- `include`: Import another launch file
  - `<include file="$(find pkg_name)/launch/file_name.launch/>`
- `node`: Run an executable file/script
  - `<node name="my_node" pkg="pkg_name" type="file_name.py"/>`
- `param`: Global parameters available to all ROS nodes
  - `<param name="/namespace/param_a" type="str" value="hello world"/>`

# Configuration Files



# Configuration Files

- Similar to messages (.msg) we need dynamic configurations (.cfg).
- Allow us to change variables instantaneously.

Create a `cfg` directory in your package.

Create a file called `WorkshopParams.cfg`



# Configuration Files

- Parameter generator has the following options:
  - Name: String which specifies the name of the parameter.
  - Type: Type of value stored, and can be any of: `int_t`, `double_t`, `str_t`, or `bool_t`
  - Level - When the callback is called all of the level values for parameters that have been changed are ORed together and the resulting value is passed to the callback.
  - Description - String which describes the parameter
  - Default - Default value
  - Min - Min value (optional and does not apply to strings and bools)
  - Max - Max value (optional and does not apply to strings and bools)

# Robot control

- Clone the github repo:
  - `cd ~/catkin_ws/src`
  - `git clone https://github.com/pirakago11/ros_workshop`
  - `catkin build`
- Might need `sudo apt install git`

- 2-D planar manipulator (again)
- Solve for  $\theta_2$

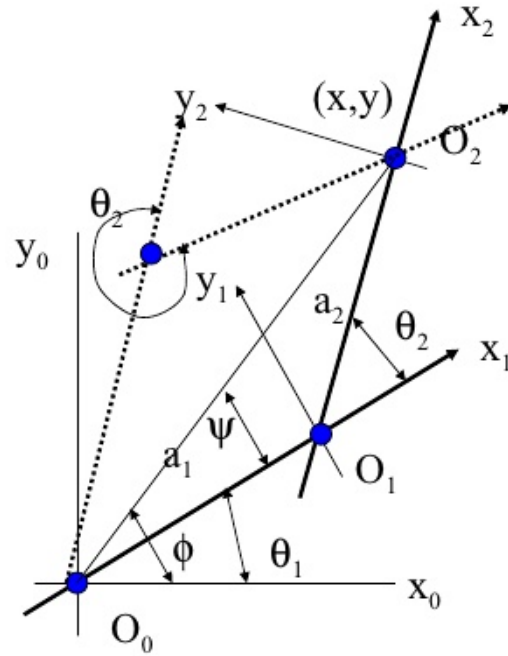
$$x^2 + y^2 = a_1^2 + a_2^2 - 2a_1a_2 \cos(\pi - \theta_2)$$

$$\cos \theta_2 = \frac{x^2 + y^2 - a_1^2 - a_2^2}{2a_1a_2}$$

for greater accuracy

$$\begin{aligned} \tan^2 \frac{\theta_2}{2} &= \frac{1 - \cos \theta}{1 + \cos \theta} = \frac{2a_1a_2 - x^2 - y^2 + a_1^2 + a_2^2}{2a_1a_2 + x^2 + y^2 - a_1^2 - a_2^2} \\ &= \frac{(a_1^2 + a_2^2)^2 - (x^2 + y^2)}{(x^2 + y^2) - (a_1^2 - a_2^2)^2} \end{aligned}$$

$$\theta_2 = \pm 2 \tan^{-1} \sqrt{\frac{(a_1^2 + a_2^2)^2 - (x^2 + y^2)}{(x^2 + y^2) - (a_1^2 - a_2^2)^2}}$$



# Inverse Kinematics

We'll just skip the math...

