## Requirement Documents

**Construction:**

arm + gripper+integration

**A demo for grasping**

arm：moveit 0.5 (displacement)

* function/node （receive a message, subscribe）
* give a coordinate (x,y,z), go to the certain place.

gripper: allgero hand (grasping excution)

* library
* grasp

integration: robot description arm+gripper(dynamics modelling)

* define a model, assembly, actions and control joints (ur10e),
* URDF -> coordinates transformation (MISSING robot\_state\_descriper node)
* registration between hands and eyes

**Procedure Reference**

<https://blog.csdn.net/frankcreen/article/details/108090729>

<https://www.guyuehome.com/18461>

**Amiga**

(We don't have to finish all the parts of amiga, pick those most important and related)

* gazebo（environment simulation） + rviz （arm simulation）integration
* rviz should be loaded into gazebo
* open rviz，load ur10e，move it （on rviz）control arm
  + by python scripts （as same as ros nodes）
  + by mouse tracking

## Adjustments made to run Allegro\_hand (Melodic) in Noetic:

Clone from:

<https://github.com/yias/allegro-hand-ros/tree/melodic>

With reference to:

<https://github.com/felixduvallet/allegro-hand-ros>

Install dependency Before catkin build:

rosdep install --from-paths src --ignore-src --rosdistro=noetic -y

# left -> right

1. Before run 'roslaunch allegro\_hand allegro\_hand.launch HAND:=right CONTROLLER:=sim',

1) Change ‘allegro\_hand\_description\_left.urdf’ to ‘allegro\_hand\_description\_left\_melodic.urdf’,

2) Add the below lines for robot description import in allegro\_hand.launch:

(Or simply replace the launch file)

<param name="/robot\_description" textfile="$(find allegro\_hand\_description)/allegro\_hand\_description\_$(arg HAND).urdf"/>

<param name="/allegroHand\_$(arg NUM)/robot\_description" textfile="$(find allegro\_hand\_description)/allegro\_hand\_description\_$(arg HAND).urdf"/>

2. 1) Rename 'allegro\_hand\_keyboard.cpp' into 'allegro\_hand\_keyboard\_node',

2) Add '#!/usr/bin/env hbcxx' on the top to run c/cpp file as a script.

(Or add '#!/usr/bin/bash')

3. Rviz visualization:

1) Solution for [Fixed frame: no TF warning]

Open a new terminal, run ‘rosrun tf static\_transform\_publisher 0 0 0 0 0 0 1 map world 5’

Solution for [Error: robot state publisher node missing]

2a)

In allegro\_hand.launch, add below lines:

<node pkg="robot\_state\_publisher" type="robot\_state\_publisher" name="robot\_state\_publisher"></node>

<node name="joint\_state\_publisher" pkg="joint\_state\_publisher" type="joint\_state\_publisher" ></node>

2b) In allegro\_hand.launch:

(around line 133) CHANGE [type="state\_publisher"] > TO

[type="**robot\_**state\_publisher">]

<!-- Joint States (angles) to Joint Transforms -->

<node name="jointState2tf\_$(arg NUM)"

pkg="robot\_state\_publisher"

type="robot\_state\_publisher">

<remap from="tf" to="allegroHand\_$(arg NUM)/tf"/>

<remap from="joint\_states" to="allegroHand\_$(arg NUM)/joint\_states"/>

</node>

3) Solution for Keyboard error:

In allegro\_hand.launch:

a) Rename 'allegro\_hand\_keyboard\_node' into 'allegro\_hand\_keyboard.cpp'

(Will be changed to what have done before Step2. Not sure if there will any difference if just skip step 2)

b) (Around line 144) CHANGE [type="allegro\_hand\_keyboard\_node"] TO [type="allegro\_hand\_keyboard"]

<!-- Keyboard handler (if arg KEYBOARD is true) -->

<node name="keyboard\_$(arg NUM)"

pkg="allegro\_hand\_keyboard"

type="allegro\_hand\_keyboard"

output="screen"

if="$(arg KEYBOARD)">

<remap from="allegroHand/lib\_cmd" to="allegroHand\_$(arg NUM)/lib\_cmd"/>

</node>

NOW the complete model for Allegro hand should be able to be imported, and can be connected to the real gripper.

Install PCAN driver (Follow README instruction: <https://github.com/yias/allegro-hand-ros/tree/melodic>)

RUN ‘roslaunch allegro\_hand.launch HAND:=right’ to test pre-defined grasping using keyboard!

NOTE: For python library,

**ARM CONTROL:**

Reference: <https://github.com/masrim2000/ur10e_moveit_sfm>

## Mark executable scripts (Python etc.) for installation

## in contrast to setup.py, you can choose the destination

catkin\_install\_python(PROGRAMS

scripts/image\_imshow.py scripts/image\_publisher.py

scripts/reciever.py

scripts/edge\_detect.py

DESTINATION ${CATKIN\_PACKAGE\_BIN\_DESTINATION}

)