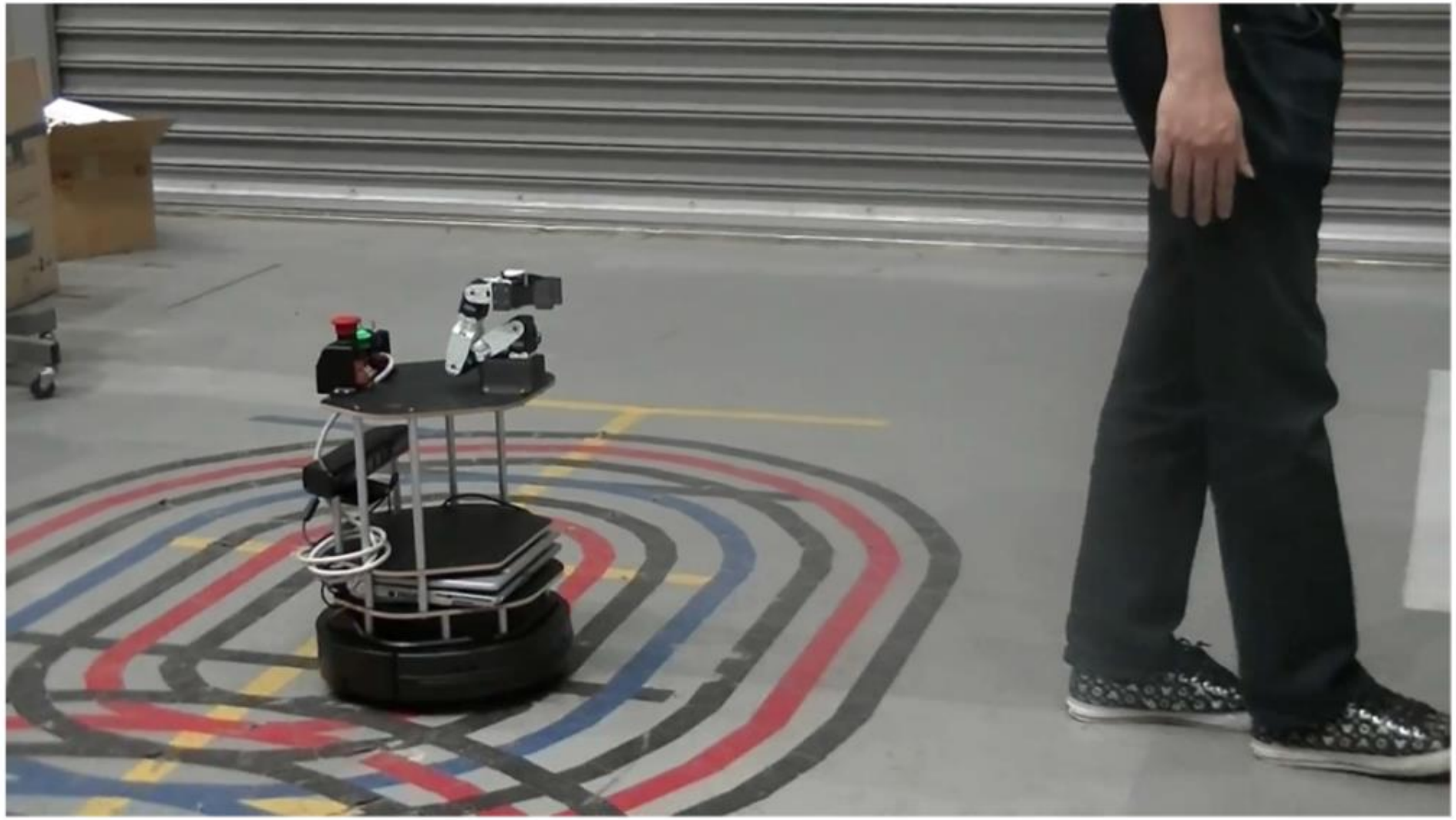
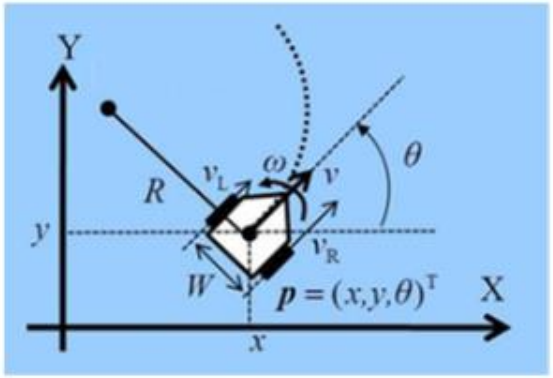
SYSTEM INTEGRATION AND APPLICATIONS

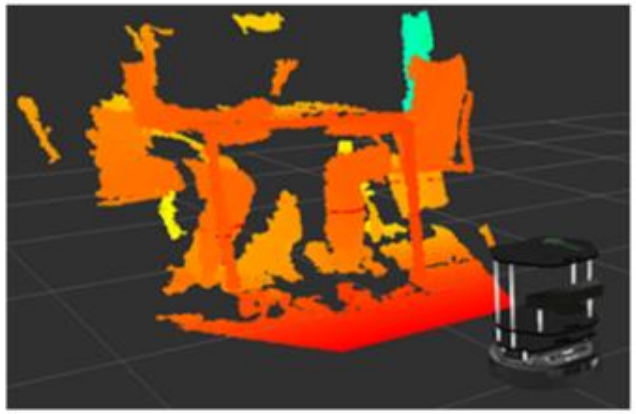
People Tracking (Follow Me)



People Tracking (Follow Me)

- Parallel Two-Wheel Vehicle (Mobile Robot)



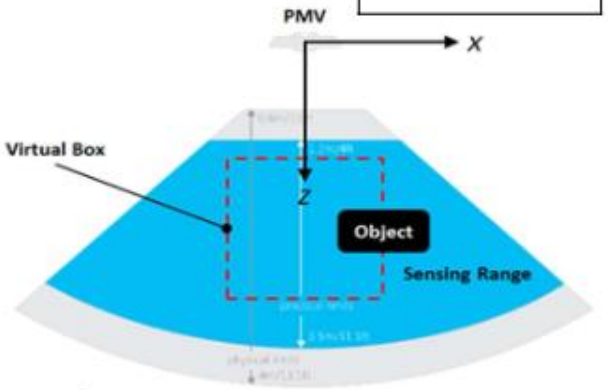


Data Point Cloud

Control Parameters

Parameter
min_y
max_y
min_x
max_x
max_z
goal_z
x_scale
z_scale

- Vehicle Following Control
  - Lateral, $X_{centroid} = \frac{\sum_{i=1}^{n} x_i}{n}$
  - Longitudinal, $Z_{centroid} = \frac{\sum_{i=1}^{n} z_i}{n}$
  - Rotation, $\omega_{PMV} = -X_{centroid} \times x\_scale$
  - Speed, $v_{PMV} = (Z_{centroid} - goal\_z) \times z\_scale$



System Integration: People Tracking (Follow Me)

- TurtleBot Follower Demo
  - http://wiki.ros.org/turtlebot_follower/Tutorials/Demo
- Bring up
  - $ roslaunch jupiterobot_bringup **jupiterobot_bringup.launch**
- Launch node
  - $ roslaunch rchomeedu_follower **follower2.launch**
- Changing Follower Parameters
  - $ rosrun rqt_reconfigure rqt_reconfigure
- Follower start/stop service control
  - $ rosrun rchomeedu_follower **follower_control.py**

System Integration: PartyBot

PartyBot

- Speech recognition with lm mode to recognize questions from guest.
- Speech synthesis to speak the answers and play audio files.
- Take photo by speech command.
- Integrated with Follower to start/stop by speech commands.
- Move mobile base and arm for dance sequence.

Robot Applications (rchomeedu_apps)

PartyBot

Launch

- $ roslaunch jupiterobot_bringup **jupiterobot_bringup.launch**
- $ roslaunch rchomeedu_arm **arm.launch**
- $ roslaunch rchomeedu_follower **follower2.launch**
- $ roslaunch rchomeedu_partybot **partybot.launch** dict:=/home/mustar/catkin_ws/src/rc-home-edu-learn-ros/rchomeedu_apps/rchomeedu_partybot/config/partybot.dic lm:=/home/mustar/catkin_ws/src/rc-home-edu-learn-ros/rchomeedu_apps/rchomeedu_partybot/config/partybot.lm

Exercise: Mini Challenge

1. Integrate all the previous exercises to develop the robot application for Mini Challenge
   - Refer to the PartyBot sample code