

# Summary

Jingyi Li

Junwei Li:

Main part: Introduction, Summary of reference, Function of ROS (SIMULATION YOUR ROBOT, PROCESSES ARE NODES, COMMUNICATING BETWEEN NODES, PROS VS CONS ), Recommendation, Conclusion

First, in introduction part, he describes what is ROS system and analyze the advantage of ROS system and why a robot need this system. In function of ROS part, he introduces how to make a Unified Robot Description Format (URDF) document to simulate robot and the ROS system how to communicate between nodes. Then, he summarizes the pos and cons of this system. Pos: 1. Multi-lingual 2. Free and open source. 3. Distributed. 4. Start easily. Cons: 1. Not reliable if do precise task. 2. Security and scalability are not first-class concerns. 3. Operation system other than Ubuntu and Linux are not well supported. Finally, he thinks this system cannot be used to do out-of-the-box self-driving wheelchair, because ROS cannot simulate so complex and unpredictable outdoor environment. We may need to train wheelchair with tones of picture from API or train it in real environment.

Michael Brunsman:

Main part: Introduction, ROS System, Analysis, Recommendations, Conclusions, References

In his report, he introduces the main function of ROS System and the reasons why people like use this system. In ROS System part, he introduces the ROS system can be broken down into three major components: Communications Infrastructure, Robot-Specific Features, and Tools and describes this system how to work. (These major groupings are the basis of all ROS code. The communications infrastructure works to effectively transfer data between various parts of a robot.) In Analysis part, he thinks if we had all hardware, the ROS system has enough capable to create a self-driving wheelchair, because the *rviz* library would be instrumental in this application. *Rviz* is easy to use and can be run using both a GUI and a command line. These are the two basic libraries and functions that would be needed to create a self-driving wheelchair.

Merlin Hoffman:

Main part: What is ROS? Sources, Pros & Cons, Recommendations, Conclusion

In his report, first he told us ROS is a framework used for writing software for robots. In Sources part, he summaries the content of three essays about ROS System. Then, he gives us several advantages and disadvantages about this system. Pros – 1. It is a modular system: It can be summarized that it is much easier to update and try different code in a modular system. 2. Large collaborative community: There are over 3000 packages that be used within the OS. 3. Open source: Anyone can use ROS and its multitude of packages as long as they adhere to the licensing agreements. 4. ROS uses modern programming languages. ROS is compatible with Python, C++, LISP, and there are experimental libraries for Java and Lua. Cons – 1. Higher complexity with large learning curve. 2. Currently only runs on Unix-based platforms. In recommendations part, ROS is a very powerful and flexible framework. It is state of the art in giving you the most control over the framework. This is ideal for cutting edge research and complex systems, but it is not a great framework for everyone though. When working with simple systems or highly focused and specialized systems other OS systems may be better.

Elise DeCarli:

In this report, he introduces the development of robotics' history. Early on, robotics just does some simple tasks that needed to be repeated multiple times with extreme precision, but now people can build robotic elements with dexterity that rivals a human's. Then, he introduces the OpenAI's robotic hand. This hand used artificial intelligence and machine learning to determine how it works. OpenAI was able to broaden the ability of their hand by training it in a multitude of randomized simulations.