

ROS Repository & Filesystem

Vivek Rijal Upadhyaya

Sprint 3 Documentation of Work

Objective of Research

The objective of this research was to learn and understand how ROS (Robot Operating System) can be used in the project. Having no prior knowledge in the stated subject and it being the critical to the project integration and development, It is essential to have a deep understanding of the framework.

Findings of Research

- ROS is meta-operating system for autonomous devices (not exactly a full operating system but a source framework that provides all the fundamental functionalities that can be applied to robotic autonomous machines)
- Supports reuse of codes from robotics research and development with a lot of open-source materials to choose from.
- Provides nodes that enable executables to be designed individually and can be put together at run time to perform difficult tasks
- Easy to implement with languages such as Python and C++
- Scalable i.e. can be used for large runtime systems and development processes

Some Risk associated with ROS are:

- ROS involves a good understanding in how the ROS repository can be setup. It basically has three main parts that it requires to run any simulation that includes the simulation environment package, Control package and a interface package that necessary to provide the services.
- As of right now ROS supports Python and C++ programming which I have very few familiarity with.
- Although ROS has its own file-system management shell commands it is going to need some dedication getting used-to and being familiar with the scripting which is essential to launch services.

Conclusion

In conclusion to this research document I am now somewhat familiar with ROS and do realize that ROS is not like any other frameworks that you can learn on the fly and it involves a lot of understanding of its repository scheme. In all fairness, it provides deep knowledge in how autonomous systems can be created which is an interesting learning experience with this project.