Robotic Operation System (ROS) report

# Introduction

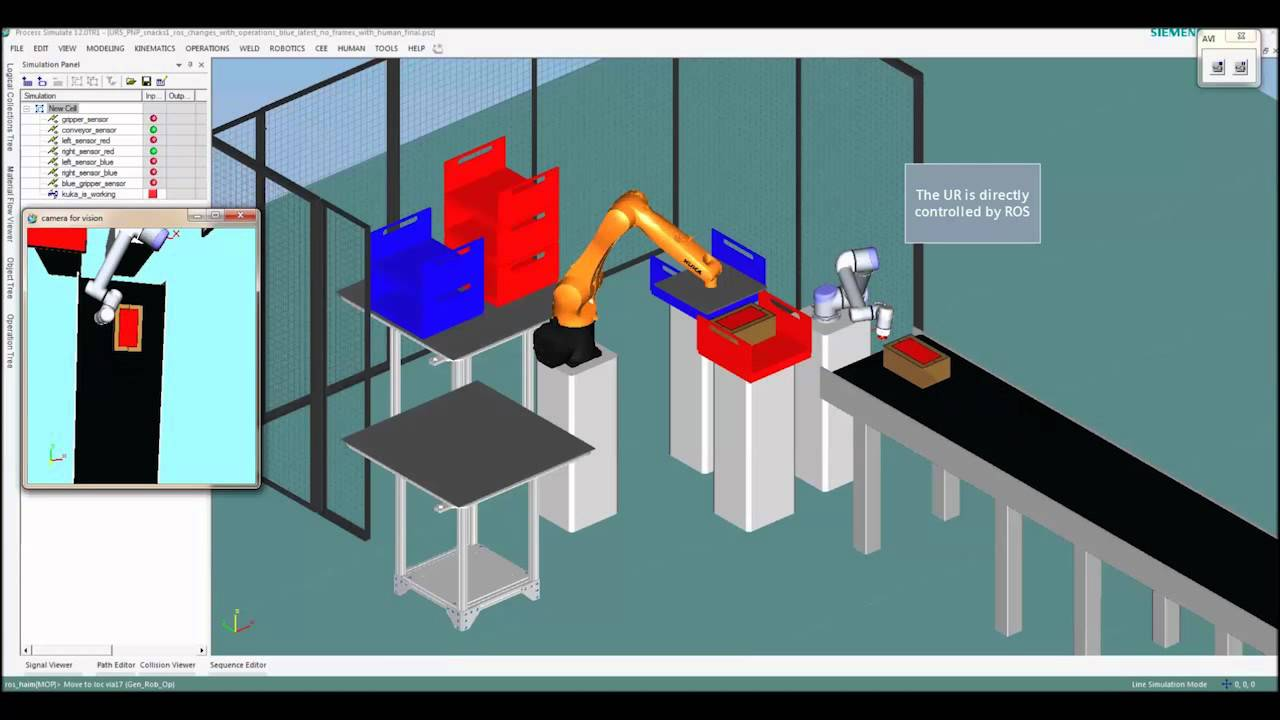
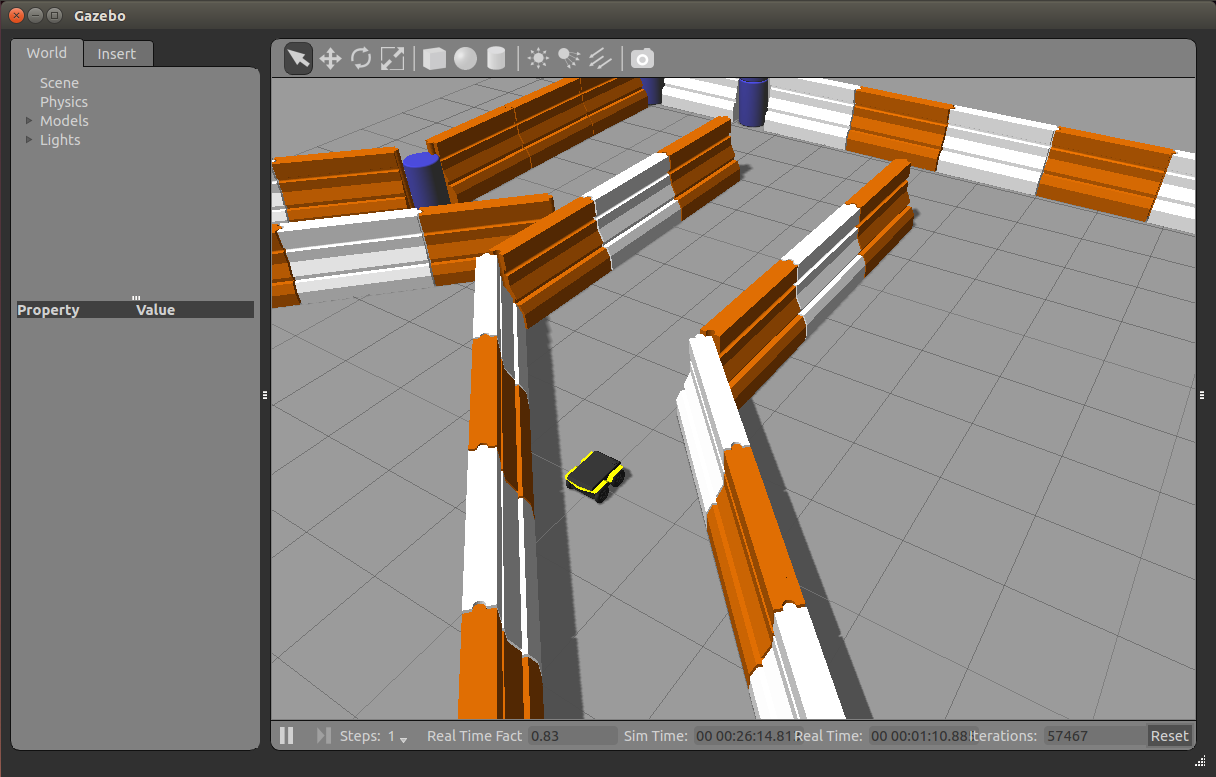
ROS (Robot Operating System) is kind of open source, providing tons of libraries and tools to help software developers create robot applications. hardware abstraction, device drivers, libraries, visualizers, message-passing, package management are also provided in ROS. A robot is made up of many hardware components such as sensors, camera, Servo motors, microcontrollers and LEDs. Each of them requires its software. Even a small robot arm with a handful of servo motors uses a servo motor library. If attach some ultrasonic sensors for collision avoidance, a camera for vision and wheel to this robotic. Probably you want to split this complex task into small processes. Therefore, ROS, providing organization of package and node communication, is a good choice.

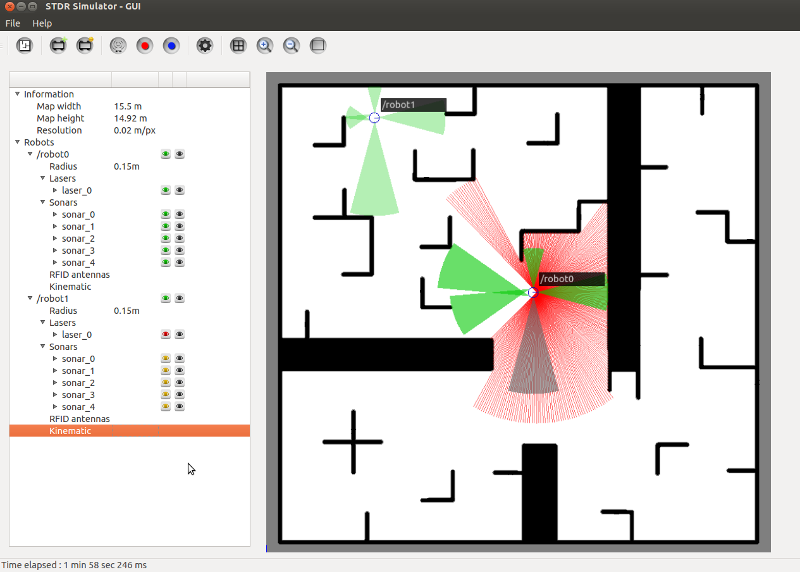
# Summary of reference

1. Dufresne, S., Dufresne, S., Adam, Darin, Saabman, Steven, … Mre. (2018, June 1). Modular Robotics Made Easier With ROS. Retrieved from <https://hackaday.com/2018/05/31/modular-robotics-made-easier-with-ros/>.
2. S. Murata, E. Yoshida, A. Kamimura, H. Kurokawa, K. Tomita and S. Kokaji, "M-TRAN: self-reconfigurable modular robotic system," in IEEE/ASME Transactions on Mechatronics, vol. 7, no. 4, pp. 431-441, Dec. 2002. doi: 10.1109/TMECH.2002.806220 <http://ieeexplore.ieee.org.ezproxy.bu.edu/stamp/stamp.jsp?tp=&arnumber=1159221&isnumber=25977>
3. Wiki.ros.org. (2019). Documentation - ROS Wiki. [online] Available at: http://wiki.ros.org/ [Accessed 16 Oct. 2019]. <http://wiki.ros.org/>

# Function of ROS

## SIMULATION YOUR ROBOT

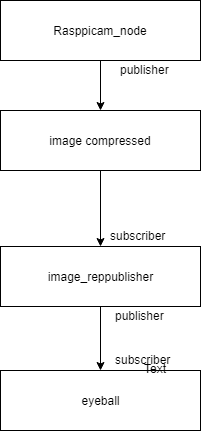
Before your robot is formed, you may want to simulate your software. For that, you can make a Unified Robot Description Format (URDF) document, which is an XML file that depicts your robot in detail. There’s even a URDF exporter add-in available for SolidWorks. You then use it with ROS and the Gazebo robot simulator to see and manipulate your robot on the screen before building anything.

## PROCESSES ARE NODES

A sufficiently complex robot will require multiple processes, possibly even spread out over different computers. In ROS, each process is a node. A package contains one or more nodes. For my example, I use raspicam\_node which reads from the Pi Camera and produces a compressed stream of video. A node called image\_republisher decompresses that video. My eyeball node then processes images from the video. Of course, to do that they need to communicate somehow.

## COMMUNICATING BETWEEN NODES

ROS provides two mechanisms for nodes to communicate with each other.



ROS is not a real operation system. It is tool for software developers. It contains lots of library for many components of robotics.

# PROS VS CONS

## PROS:

1. Multi-lingual: ROS modules can be written in any language for which a client library exists (C++, python, MATLAB, Java, etc.)
2. Free and open source: Most ROS software is open source and free to use
3. Distributed: Programs can be run on multiple computers and communication over network.
4. Start easily: other people’s work is provided, and you can try and share your own project.

## 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

# Recommendation

This is tool and library for robotic design. Based on knowing C++ or python or other language, you can use different language in ROS and combine them from different computer and different language files.

Also, ROS contains lots of hardware drivers you can use directly. It will reduce lots of software work in your robotic project.

# Conclusion

ROS is open source and free. I recommend robotic beginners try ROS to build their robotic project. Because there are many project codes developed by others. You can download and try other people’s work. In addition, lots of tutorial can be found on online.

**Can we do out-of-the-box self-driving wheelchair using what is in ROS in open source?**

My answer is no. although designer can simulation the software in ROS, it is not reliable. After all, the first concerned is safety for designing wheelchair. Probably, 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.