The development and implementation of robots is definitely an iterative task that requires the right tools for development. The right tool in this context is the **Robotic Operating System**. Commonly referred to as **ROS**, it is an open-source framework used by researchers and developers to build and reuse code between robotics applications. This offers the best platform to prototype a robot while learning and also doing research.

Leveraging its a rich platform, as listed below, led to the following **Achievements**:

* Its open source therefore, its rich libraries and functionalities were used in the enitre design of the robot function. Data, such as Odometry data using Transform Function to assess motion of each wheel with respect to the user input data – **[Demostrated in the Control Videos RVIZ and Transform\_Publishing\_tree]**.
* Has numerous libraries, allows code re-usability, has visualization and simulation tools. Using **RVIZ** (A visualization tool) to monitor the change in transforms that inturn dictate the motion of the robot and **Gazebo** to monitor its environmental behaviour.
* Communication System is based on Publish, Subscribe and Remote Method Invocation), it is also cross-platfrom, therefore allows the use of various low-level, high-level, scripting and markup languages to be used in simulation. In the project, XML, YAML, URDF- (Unified Robot Description Format): All these, being Markup languages were used in robot description for simulation. This made it easy to transfer a SolidWorks file into ROS environment for simulation and control.
* Framework & Tools that were utilized to build the system and dependency management, visualization, record and replay. This made correctional-iteration in development and design phases easier.
* The ROS ecosystem is event-driven, therefore, robot motion control (driving) was achieved by triggering communication between various services such as keyboard control, a designed Graphical User Interface and command-line control.
* ROS is dynamic and active, therefore the frequency of information, data and instruction transaction is characterized by null latency. The core technology of ROS lies in its distributed communication mechanism—the ROS Master helps different Nodes to find each other and establish connections

**Challenge(s)**

During simulation, one of the key challenges is designing an interactive environment. This is a Graphical Intensive and hungry task. Therefore, the design of an interactive environment to simulate the behaviour of the robot in case of object collision and lane detection was not within our graphical reach. However, with the help of scripting, it was possible to simulate the behaviour and interaction in the case of object detection and lane-keeping.