Tools for creating new ignition projects

Organization - Open Source Robotics Foundation

Contact Information

Name: Harsh Mahesheka
Phone Number: +91 9470292105
Email Address: harsh.mahesheka.eee20@iitbhu.ac.in
LinkedIn: https://in.linkedin.com/in/harsh-mahesheka-ab0725212
Website: https://harshmahesheka.github.io/

Course Work

I am a second-year undergraduate student pursuing an Integrated Bachelor of Technology (B.Tech)+ Masters of Technology (M.Tech) in Electrical Engineering at the Indian Institute of Technology (IIT), Varanasi.

- **Robotics**

  I am a core member of my college's Robotics Club and Robotics Research Wing (Roboreg). I have been working in them for over one and half years now and have done multiple projects and won numerous competitions through it. I have also played an instrumental role in ROS integration of our robots.

  Relevant courses: ROS for Beginners 1, ROS for Beginners 2, ROS for Beginners 3 by Anis Koubaa on Udemy, Learn ROS2 as ROS1 Developer by Eduoard Renard on Udemy, Robotics: basics and advanced concepts by IISc, Linear control systems.

- **Simulation**

  Through projects and competitions, I have worked with multiple simulators like Gazebo, Ignition Gazebo, Pybullet, Webots, Carla, etc. I have also organized various workshops on pybullet, gazebo, etc., in multiple colleges.

- **Software Engineering**

  I have been involved in programming since the start of my college. I have experience in Python, C, C++, CMake, Matlab, etc.

  Relevant courses: Computer Programming (CSO101), Complete Python BootCamp by Jose Portilla on Udemy.

Experience

Multi-Agent Pickup And Delivery System(MAPD):

I led a team of 10 in the Flipkart Grid Robotics Challenge Competition, in which we designed a multi-agent pickup and delivery system for warehouses using omnidirectional mobile bots.

As a team leader, I contributed in almost all respects, especially in bot control, ROS integration, navigation, path planning, and bot communication.

Simulation

Created a realistic simulation environment and robot description for our mobile bots. URDF and xacro were used to create the robot description. We implemented our controllers and planning algorithms in simulation, and observations from it helped us a lot in real-life implementation.

Planning and Control

Implemented CBS-based Multi-Agent Path Planning and Scheduling as Global Planner for the swarm of bots. We used tracking_pid in both simulation and hardware for local planning and control.

Localization and Communication

Designed a centralized vision-based localization system based on Cameras and April tags which were fused to high-speed odometry data. We implemented multi ROS master communication between our bots, Jetson Nanos, and laptops using fkie_multimaster package.

Hardware

Designed and fabricated our own Omni-directional ROS-integrated mobile robot with a delivery mechanism and implemented our controls and planning algorithms on it. Robots were designed in a modular manner to allow usage in other applications.

For visuals and detailed descriptions, you can visit Dot-Bot.

All the required code is uploaded on- https://github.com/Robotics-Club-IIT-BHU/Dot-Bot

The code used in hardware is uploaded on -

https://github.com/Robotics-Club-IIT-BHU/Swarm-Bot-Hardware

Multi-Purpose Household Bot:

Led a research project in my institute under the guidance of Prof. Shyam Kamal with the objective to design an affordable multi-purpose household bot that could help in day-to-day activities.

As a team leader, I contributed in almost all respects, especially in simulation, ROS integration, bot control, and navigation.

Simulation
The CAD model of the bot was created using Solidworks. Further, a URDF file was created using the model, and all the links were controlled and simulated using gazebo in a household environment.

Navigation
We used Differential Drive along with ROS Navigation Stack on our Bot to autonomously navigate it through the house. We developed four moods for its navigation-Teleop Controlled, Autonomous Mapping, Autonomous Navigation, and Autonomous Complete Coverage.

Baby Care
Designed and integrated face and object recognition algorithm to detect threats like knives or unknown people around babies. Also designed and tested baby following algorithms in the gazebo.

We are currently working on the hardware implementation of the project. You can find further details and code at https://github.com/harshmahesheka/Multi-Purpose-HouseHold-Bot.

UAV Guided UGV on Mountainous Terrain:

The project was done under Inter IIT Tech Meet 2022, in which we had to guide a car through a drone on a pre-mapped snow-covered hilly area in Gazebo. The project included Road Segmentation from Visual Data for Mapping, UAV Localization Controls and Planning, UGV Localization Controls, and Planning.

UGV Localization, Controls, and Path Planning
My major contribution was to UGV Localization, Controls, and Path Planning. UGV was localized with respect to UAV using YoloV5, and UAV was localized with respect to the map using GPS and odometry data fusion. Now for Car Control and Low-level planning, we used the Se_2 Navigation package and Teb_Global_Planner for High-level planning.

You can find further details and code at https://github.com/InterIITRepo/MP_DR_T14.

Cutlery Sorting Arm:

Designed a 4 DOF arm with an on-arm camera that could sort cutlery; it used Deep learning models for object identification and the move_it framework for control. This project was done in collaboration with NTU Singapore, and my major contribution was to integrate move_it, gazebo, and ros with their arm URDF.

You can find further details and code at https://harshmahesheka.github.io/csa.html or https://github.com/ntu-maerc/a1-AY2122Sem1

Open Source Contributions

Pull Requests:

Serial Number	Title	Status
1	Adding tests for CONFIG option	Merged
2	Adding CONFIG option	Merged
3	Macros for generating cmd*.rb and conf/*.yaml files	Open
4	Improving CONFIG test	Merged
5	Adding ign_gazebo_dev package	Open
6	Error in ign_gazebo5 installation documentation	Merged
7	Removing Drag and Drop	Open

Issues:

Serial Number	Title	Status
1	Gazebo crashes in specific worlds	Closed
2	Error in ign_gazebo5 installation documentation	Closed
3	No drag me option	Open

Specialized Skills

Languages: Python, MATLAB, C, C++, HTML

Development Tools: Git, CMake

Technologies: ROS, ROS2, Gazebo, RViz, OpenCV, Move_it, PyBullet, Tensorflow, OpenAI gym

Statement of Intent

I have always been interested in Robotics and Software development and was introduced to ROS and Gazebo a year ago. Since then, I have spent a major portion of my time working with them. I integrated ROS and Gazebo into all my robotics projects and felt a huge difference. I was amazed at how simply one can apply such complicated algorithms to real bots. This along with reading about the organization and how it is democratizing robotics, pushed me to work with ROS and Gazebo.

Also, I recently started working with ROS2 and Ignition Gazebo and loved the range of improvements. I wanted to give back to the community, so I organized multiple ROS workshops in various colleges across India. I also wanted to contribute to the development and found GSOC to be an ideal way to do so.

Project Implementation

Now, on the part of project implementation. There are three significant segments to the project-

1. <u>Macros for project creation</u>:
I plan to create five macros initially to help users add and source all of the common elements in an ignition project.

I. **ign_add_resources** (`/path/to/models_or_world`)-This will install all the models inside the mentioned folder and add required paths in hooks.
II. **ign_add_plugins**(`/path/to/plugin`)-This will add all the plugins inside the mentioned folder as libraries, install them and add required paths in hooks. We can have an option for adding common target_link_libraries, dependencies, etc., and an option if you want to add something to some specific plugin from the folder.
III. **ign_add_examples**(`/path/to/example`)-This will be similar to the last one, just instead of libraries; this will add files as executables.
IV. **ign_add_tests(**`/path/to/tests`)-First, we can add macros for gtest for the non-amnet package(basically a counterpart to ament_cmake_gtest), and then similar to the last 2, this will build all the tests inside the directory with options to add common and specific dependencies.
V. **ign_environment_hook**(): This will generate and install hooks based on the macros mentioned above.

The macros will work for all package types (including plain cmake dependent). Also, these are just some of the macros I felt most relevant. I plan to **work further on other macros** that can help users create an ignition package like adding custom messages, etc., based on suggestions from mentors and the community.

I have created a prototype in which I have created ***ign_add_resources, ign_add_plugins,ign_add_executables,*** **and** ***ign_environemnt_hook*** for the colcon compile type.

You can check the code at Prototype_Macros
Video demonstration can be found at Macros_Demonstration

2. <u>Base Package Generation Tool</u>-
This tool will generate a base package for users just like the command ***ros2 pkg create*** does based on inputs from the user. I was planning to create an **ign subcommand** for the tool like ***ign create pkg***. For implementation, I will try directly plugging in my python prototype code or refactor it in ruby. The tool will offer users a range of options like

- **built_type**: Whether the package will depend on amnet or not
- **package_type**: basic/advanced(advanced pkg will offer folders for worlds, plugins, etc. along with Hooks preinstalled). It can also have more specific package types based on common ignition packages used.
- **standard_dependencies**: Whether to include standard dependencies (like ignition-gazebo) or not
- **ignition_version**: version of ignition on which package will depend

I have created a prototype of the tool using python script. Currently, it is in the development phase, and many more features like adding placeholders for C++/Python code, the option to add standard boilerplate codes, etc., along with macros created above, will be added later.

You can check the code out at Ign_pkg_create_prototype
Video demonstration can be found at Ign_pkg_create_demo

3. Tutorials and Tests-
I will create **detailed tutorials on creating various ignition packages** using the macros and tools mentioned above and also add detailed comments around all the code and generated files. The idea being to help a completely new user to get started with ignition. Also, I will **create tests for various features and add support for unit tests**.

Above mentioned ideas will be the base of the project but I am open and would love to **increase this project's scope based on the guidance from mentors and the community** with the idea to create a complete tool helping all levels of users create ignition packages.

Timeline

- **Pre GSoC Period |** Before 20th May
    - Dig deeper into the ignition-cmake
    - Get further familiarized with ignition packages
    - Read the Ignition Contributor Guide to get acquainted with the practices and standards followed by the organization.
    - Further, develop simple macros under ign-cmake
    - Construct a suitable software architecture prototype for the package create tool compatible with ignition packages
- **Community Bonding Period |** 20th May to 12th June
    - Extensive discussion with the mentor regarding the final architecture of the project. (Iteratively updating the architecture during the development cycle)
    - Setup the project repository on GitHub. Create a GitHub project board to keep track of ongoing, to be done, and completed tasks (issues).
- **Phase I |** 12th June to 25th July
    - Create all the discussed macros under ign_cmake for all package types
    - Write test scripts for developed macros
    - Update template repositories with new macros
    - Add documentation for the new macros
    - Read and research about ign subcommands
- **Phase II |** 25th July to 20th August
    - Finalize contents of template repositories
    - Develop the ign package create tool for all package types
    - Write test scripts for the tool.
    - Create documentation for the tool
- **Phase III |** 20th August to 12th September
    - Create detailed tutorials on creating all types of ignition projects.
    - Address any open issues
    - Extend features with suggestions from the community.

This is a rough estimate of the work and the timeline for the GSoC program. I would like to take any input on the same from the mentors and work accordingly.

Post GSoC

I would like to maintain the project and address issues even after the conclusion of the coding period. I would also love to work on other issues people face in getting started with ignition and **help them by making/improving tutorials, macros, tools for getting started, etc.** As I earlier mentioned, I am deeply influenced by the work you do at Open Robotics and would love to maintain a **long-term relationship with the organization and give back to the community.**