MANIPAL INSTITUTE OF TECHNOLOGY
MANIPAL
A Constituent Institution of Manipal University

PROJECT MANAS
BRINGING BITS TO LIFE

27th INTELLIGENT GROUND VEHICLE COMPETITION

15th May, 2019

DESIGN REPORT

Team Leader: Siddarth Venkatraman (projectmanas.mit@gmail.com)
Tech Head: Shrijit Singh (tech_head@projectmanas.in)

I hereby certify that the development of vehicle, **Solo**, described in this report has been equivalent to the work involved in a senior design course. This report has been prepared by the students of Project MANAS under my guidance.

Professor & Head
Dept. of Computer Science & Engineering
Manipal Institute of Technology
Manipal - 576 104

Ashalatha Nayak Professor and Head of Department
Department of Computer Science & Engineering
E-mail: asha.nayak@manipal.edu

Contents

1 Introduction

Project MANAS, the AI robotics team from Manipal Institute of Technology, Manipal has designed its newest iteration of its autonomous bot Solo to compete in the 27^{th} Intelligent Ground Vehicle Competition. Solo is the next generation of our autonomous bot and represents the culmination of the hard work of our entire team. It continues to uphold MANAS's vision of pushing the boundaries of Artificial Intelligence and robotics, while ensuring its availability to the general population. Solo includes new cutting edge capabilities which have never been seen in other robots, including but not limited to it robustness, design, swiftness and software architecture.

2 Team Organization

The team is broadly divided into three separate divisions - Artificial Intelligence, Sensing & Automation and Mechanical. All subdivision and division heads are solely responsible for his/her division/subdivision. To assist with the managerial tasks of the team, we have a separate non-technical management division. The primary board comprises of the Team Leader, Tech Head and Team Manager who is responsible for the functioning of the entire team and take all majors decisions pertaining to the team under the guidance of our faculty advisors. The team is comprised exclusively of undergraduate students, numbering 63 in strength consisting of students from all branches of engineering and our interdisciplinary nature is the catalyst for our innovation and creativity.

The list of members who contributed towards Solo are: Avirat Varma, Shrijit Singh, Sahil Swaroop, Shivesh Khaitan, Arya Karani, Rishab Agarwal, Dheeraj Mohan, Sarathkrishnan Ramesh, Rakshit Jain, Raunaq Kalra, Dasarath Selvakumar, Shivanshu Agarwal, Chaitanya, Siddarth Venkatraman, Gaurav Singh Thakur, Manav Sachdeva, Tanaya Mandke, Ansel Dias, Omkar Jadhav, Vibhuti Ravi, Abhineet Choudhary, Adheesh H.M., Aneesh Chawla, Anish Biswas, Anurag Borkar, Apratim Mukherjee, Baidyanath Kundu, Dheeraj Rajaram Reddy, Dhruv Joshi, Garima Singh, Karan Khanna, Leander Melroy Maben, Sahil Khose, Sarthak Mittal, Shivam Agarwal, Aniket Bhawe, Asish Boggavarapu, Debayan Deb, Harsh Barde, H. Sai Manish, Parthesh Savla, Rohit Natu, Achintya Dutta, Akhil Bonagiri, Anirudh Ameya Kashyap, Arpit Chauhan, Kishore K., Gokul P., Nishan D' Almeida, Raj Tulluri, Ritwik Agarwal, Shoumik Dey, Shreesh Tripathi, Yagya Malik, Aditya Veerabahu, Akshat Rawat, Kaashvi Saxena, Manikya Sahai, Nikhil George Savio, Rishi Raj, Rithika Iyer, Ritu Chaturvedi, Siri Rajanahally

Total cost estimate of the entire bot with all components included is **$19,570**. The amount spent by the team this year is: **$3,060**. A huge cut down on the expenditure is the result of team's efforts to reuse the components already used last year.

3 Design Process

The team has used a seven stage design process while designing the vehicle. The first step in our design process is to identify the problem. We analyzed our performance last year and tried to eliminate all the problems we faced. The entire team then has brainstorming sessions where we research and come up with various possible solutions to the problems we have identified. We then evaluate all the viable solutions and choose the best one which fits our criteria. Based on our criteria, prototypes of the various modules are made which then undergoes rigorous physical testing in simulation and real world so as to ensure durability, reliability, accuracy and efficiency of the bot in conditions similar to that of the competition.

The results of our tests are carefully analyzed and points of failures are identified. These failure points undergo redesign and then go through our entire design process again before being integrated into the bot finally.

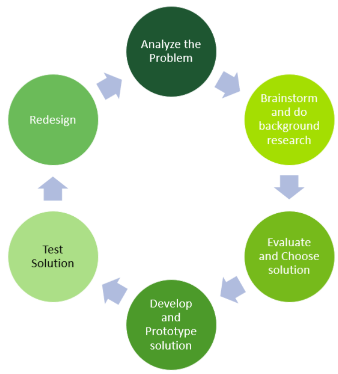

Figure 1: Design Process

4 Innovations

1. **Deep Learning Based Lane Detection:** An encoder-decoder based deep convolutional network is utilized to predict lanes and potholes that are robust to environment changes with minimum heuristics involved.

2. **System Monitor:** A node monitoring system has been built which has been pre-programmed to deal with system failures and provide running status of the system.

3. **Main Housing Unit:** The custom made main housing unit has been developed for weatherproofing and securing all electronic and computation units. The housing unit also accommodates the payload.

4. **Motor Mount:** A single unit to mount the motor has been manufactured to overcome the wheel alignment issue.

5. **Cooling System:** An advanced cooling system specifically designed for the main housing unit to ensure consistent performance of the laptop and battery even under load.

6. **Custom Planner:** The path planner for Solo has been specially designed for a region like IGVC course.

7. **Connection and Wire management:** PCBs have been designed for the connections instead of unreliable breadboard connections.

8. **Accurate modelling of Solo in simulation:** Accurate model of Solo has been designed in Fusion 360 and then imported in Solidworks which converted the model to URDF with accurate dimensions, mass and inertia values to be used in simulation.

5 Mechanical

5.1 Overview

Solo was mechanically designed to be modular, lightweight, weatherproof and accurate. The mechanical design comprises of three main parts: the main housing unit, the chassis and sensor pole structure. Solo is a front wheel differential drive, unlike last year's robot. For superior stability it has a low center of gravity and lightweight structure, which also enables it to to perform accurate point rotations and achieve high velocities very quickly.

Figure 2: Solo Assembly

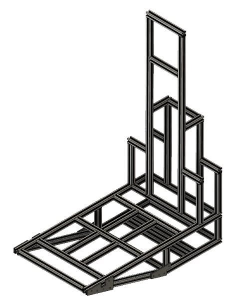

Figure 3: Chassis

5.2 Structure

The structure is designed to be extremely modular, such that all subsystems can work in parallel on different parts of the bot and add new extensions rapidly with ease. This is done by fastening aluminium members together. The structure is also designed keeping in mind the problems faced last year such as wheel alignment, low amplitude sensor pole vibration, weatherproofing and insufficient suspension.

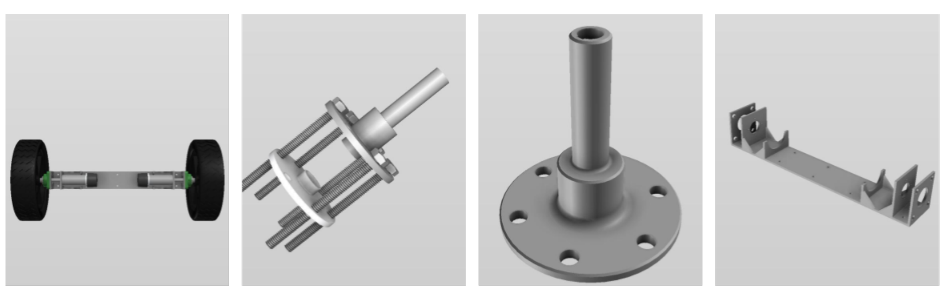

Figure 4: Mechanical Components

5.3 Main Housing Unit

This is a custom manufactured unit made of PVC (Poly Vinyl Chloride) sheets which stores all the electronic components, batteries, laptop, and the payload. It is compartmentalised to keep all the components in separate units to allow an easy work-flow and less clutter. The housing unit gives the bot a very aesthetic and unique look and secures all the components from external factor i.e. inconvenient weather conditions. An emergency stop is mounted on the external surface of this housing unit which is easily accessible even while the bot is moving.

5.4 Chassis

The chassis is the base frame or level zero for the entire robot. Apart from housing the drive train, it also provides support to the main housing unit and the sensor pole with the help of additional aluminium extrusions.

1. Drive Train: The drive train consists of a dual motor dual wheel differential drive arrangement, which solves the wheel alignment challenge by mounting both the BLDC motors in a single motor mount to prevent any chance of alignment error. To couple the motors to the wheels a customised flange like coupler was incorporated onto the motor mount using a block bearing.
2. Suspension: Since the bot will not face any extremely rough terrain, Solo is equipped with PU (Polyurethane) Foam wheels to provide vibration damping on the front end and a heavy duty castor with a spring strut to dampen the vibrations on the rear end.

5.5 Sensor Pole

The sensor pole mounts all major sensors like the 3D LiDAR, two ZED cameras and VectorNav INS. The structure is made to be very stable with a wide base frame and L-Structures supporting the sensor pole which provides additional stability. A 3D printed mount is used to hold an LCD screen against the sensor pole.

5.6 Cooling System

Solo is equipped with an ergonomic cooling system, using the laptop's inbuilt fans for maximum efficiency. To keep the inside of the box cool and avoid overheating of the laptop, cooling fans are mounted on the base of the housing unit which allows the cool air from the surrounding directed towards the vents of the laptop and thereby causing the hot air to exit out of the main housing unit. An additional fan is mounted on the front of the box to cool the electronic components and the batteries housed on the inside of the box in case excess heating should occur there.

5.7 Weatherproofing

To overcome the weatherproofing problem the main housing unit is lined with silicone based sealant on the edges leaving no area inside exposed to any kind of external weather threats. Apart from this the Sensor Pole is covered with HDPE (High Density Poly Ethylene) sheets to make sure that the sensors and connections which are exposed are covered.

6 Sensing and Automation

6.1 Overview

This time extensive changes have been done in the electrical system. The power distribution has been completely overhauled. In order to eliminate failure due to loose wiring custom perf boards and PCBs have been designed. For higher precision and control we have switched to brushless direct current motors which is controlled using an ODrive dual channel motor driver which gives us very fine control over the motors. Furthermore, a custom PCB has been manufactured for power distribution, current protection and monitoring.

The power core of solo is 24V lead acid battery pack that feeds into the tailored PCBs which power the motor driver, LED panel, buzzer, magnetic lock, and all other sensors. As the electronic components might be prone to failure, the team laid down the design and documentation of the protection system before producing the hardware. Extra safety measures have been added to ensure failure points are minimized.

To ensure no overheating, the bot is equipped with an advanced active cooling system in the main housing unit. The cooling system has been designed as to take advantage of the aerodynamics of the bot.

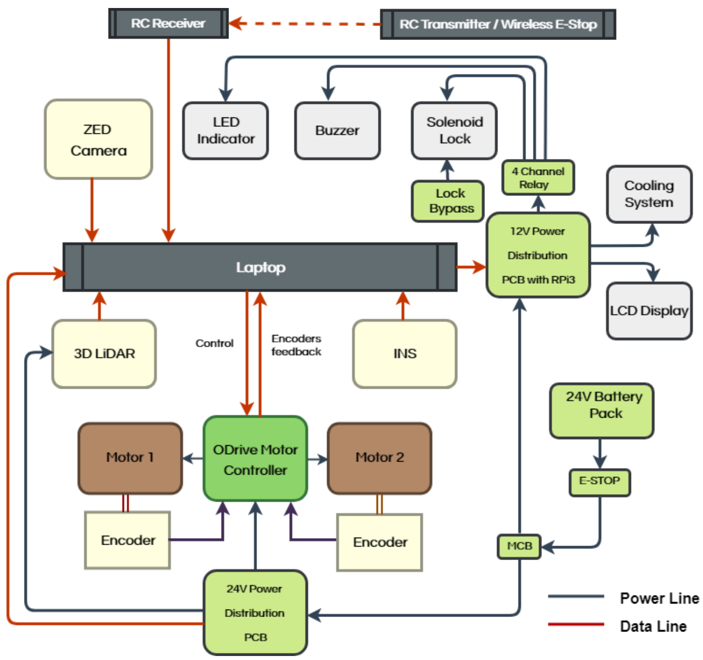

Figure 5: Wire Routing

6.2 Power System

The 24V battery pack is comprised of multiple 12V batteries combined in series, having a total capacity of 7Ah that gives solo a run time of 80-100 minutes. Solo contains 2 PCBs for 12V and 24V supply to isolate the appliances and avoid false wiring. All components powered from the PCB are incorporated with fuses and short circuit protection. Molex XT60 are used as power connectors in order to avoid reverse polarity connection of different electronic modules. The maximum power consumption of the battery is 358 Watts. Maximum continuous power rating is broken down in the table below:-

The power system has also been given additional 24V, 12V and 5V connectors along with dedicated relays for controlling the hardware to append any sensing element in future. Battery charging time is 7 hours while for the laptop it is 3 hours. The mechanical and wireless emergency stop are integrated through solid state relay which controls the power to the motor driver, connected through the power distribution board.

6.3 Battery Management System

The power distribution board is integrated with an INA226 current monitoring and protection IC that monitors the power consumption of the motor driver and indicates the battery voltage in which is mapped to a battery percentage that is displayed on the LCD screen.

6.4 Safety Measures

The Power distribution and protection PCBs are incorporated with fuses with an over current limit and a fly-back diode for reverse polarity protection from motors due to misconnection. Dual level protection is added to ensure uninterrupted function of the bot. INA226 on the circuit board encompasses the alert pin which has been set to an over current alert signal for protection of the motor driver and battery pack in case of stalling of motors or high current surges.

There are two separate emergency stops implemented. The mechanical e-stop is on the hardware level, when pressed disconnects the connection to the PCB from the power supply. The wireless e-stop is controlled via 2.4 GHz RC transmitter joystick that sends a signal to PCB micro controller unit which in turn deactivate the solid state relay hence disconnecting only the motor driver from the power supply.

The main housing unit contains all the electronics along with the laptop and batteries. Thus it is equipped with various safety features such as a magnetic lock that prevents unauthorized access, buzzer for an audio signal that gets triggered when someone tries to move the bot outside a specific region. All these features are controlled and powered via a dedicated PCB that works on the 12V battery through a Raspberry Pi to fetch the signal from the main computation unit.

6.5 Electronic Suite Description

6.5.1 Computation

The onboard laptop is used to process and compute all the sensor data and camera feeds, giving an input to the motor controllers with encoders to obtain a closed loop operation. Solo carries an Acer Predator G9 793 with 7th generation Intel core i7 clocked at 2.8GHz, Nvidia GeForce GTX 1070 6GB graphics card and 16 GB RAM. This computer is responsible for obtaining sensor data and running the path planning and control algorithms.

6.5.2 Microcontroller Unit

Apart from of the onboard laptop there also is a Raspberry Pi 3B+, an Arduino Nano and a ODrive motor controller. The Raspberry Pi is to control the peripheral devices like the lock, light, and buzzer. It has its own PCB to reduce wiring clutter and for better power management.

The Arduino Nano handles the battery monitoring and e-stop functionality. The ODrive motor driver is used to control the motor which receive commands directly from the laptop through USB communication.

6.5.3 Battery Monitoring System

The new power distribution board this year is integrated with a INA226 current monitoring and protection IC which monitors the power consumption of the motor driver in real time . In addition, it gives the battery voltage which is also is mapped as a battery percentage and displayed on the LCD screen.

6.5.4 Encoders

Each motor shaft is connected to a quadrature optical rotary encoder with a resolution of 2048 CPR. The motor shaft is further geared down by 43 times to give very precise angular position of the wheel. The feedback is used for improved closed loop BLDC motor velocity control. The data is further used for localization by calculating absolute position.

6.5.5 LiDAR

Solo is equipped with the Quanergy M8 LiDAR sensor which has 8 laser detection layers employing Time-of-Flight (TOF) depth perception resulting in 3D point clouds for spatial sensing. With a 360°horizontal and 20°vertical FOV and each beam generating close to 150,000 points per second, at full FOV the sensor generates over 1.2 million data points per second. Operating at 20 Hz, it gives 0.2°angular resolution and the data is transmitted to the laptop via Ethernet using TCP/IP which is used to identify the ramp and avoid obstacles.

6.5.6 INS

The VN-300 is a miniature, high-performance Dual Antenna GNSS-Aided Inertial Navigation System that combines MEMS inertial sensors, two high-sensitivity GNSS receivers, and advanced Kalman filtering algorithms to provide optimal estimates of position, velocity, and orientation. By utilizing two separate GNSS receivers and antennas, the VN-300 enables accurate heading measurements without reliance on vehicle dynamics or magnetic sensors, providing unmatched performance under both static and dynamic conditions.

6.5.7 Stereo Camera

The stereo camera utilized is a ZED from stereolabs with 2K resolution, dept range of 15m, 6 axis positional tracking, depth perception at large distance and a USB interface with laptop.

7 Software System

7.1 Overview

The software system is built on Robot Operating System (ROS). It is divided into two modules namely perception and planning. The perception module interprets the incoming data from various sensors. It is responsible for obstacles and ramp detection using a 3D LiDAR. Identification of lanes and potholes is done on images from a stereo camera system. Lane following is achieved from local way-points generated using the detected lanes and potholes. Subsequently, the planning module accepts the interpreted data for creating a map of the environment which distinguishes between feasible and vulnerable regions for the robot to travel. Path and trajectory planning algorithms then generate the route and command velocity for the movement of the robot. The performance of the entire system is optimized by reducing the amount of data transfer between modules. ROS nodelets feature is exploited for passing data using pointers instead of normal messages over network connection.

7.2 Perception

Our shortcomings of last year‘s module helped us identify extreme sensitivity of conventional algorithms towards environmental factors. Hence this year we designed a deep learning solution which is known for its high capacity to generalize. We employ two ZED cameras to improve the range of lane detection and eliminate blind spots.

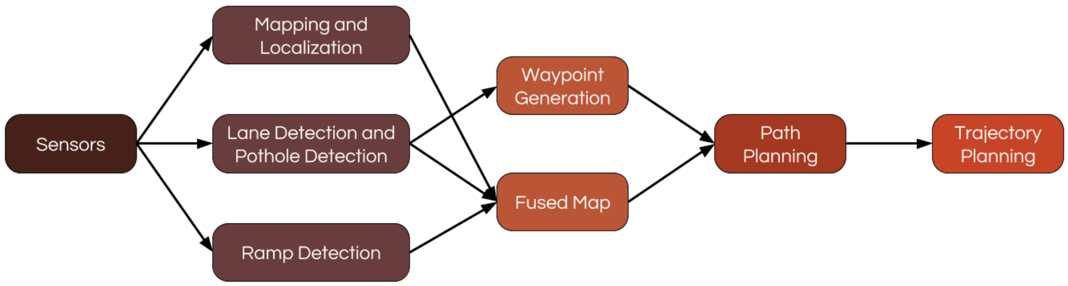

Figure 6: Software System

Figure 7: Perception Pipeline

7.2.1 Lane and pothole detection

Conventional algorithms for lane detection are highly dependent on parameter tuning for specific environments, and hence have low generalizability. They also are are not robust enough to capture geometric properties due to which they are unable to differentiate between lanes and obstacles. We thus improved upon our perception from last year by using a Deep Learning solution this time. The lane and pothole marking is posed as a binary segmentation problem which classifies each pixel of input image as point of interest or background. A modified version of the state of the art DeeplabV3+[1] model is employed to train on our own internally generated lane/potholes IGVC dataset. DeeplabV3+ is a segmentation model that specializes in detection of object boundaries without attachment of CRF units. It employs Atrous Convolutions which capture global context in the image or feature map.

Deeplab uses an encoder-decoder pattern in which the encoder is unaware of the decoder and vice versa. The encoder through learning gains the ability to map information in higher dimension into information in lower dimension which still can be decoded into the higher dimension without loss of much information. This feature is vital in our case because the bottle-neck in the model forces it to eliminate redundant information and preserve only information that is absolutely necessary.

The binary mask generated consisted of a high number of misclassified pixels due to large imbalance in the foreground and background classes. This problem is solved by using a unique loss function called 'Focal Loss' proposed in the RetinaNet[2] paper which pushed our model to classify less conservatively. The Focal Loss adds a modulating factor to the standard cross entropy loss that is defined by a focusing parameter which controls the amount of focus required.

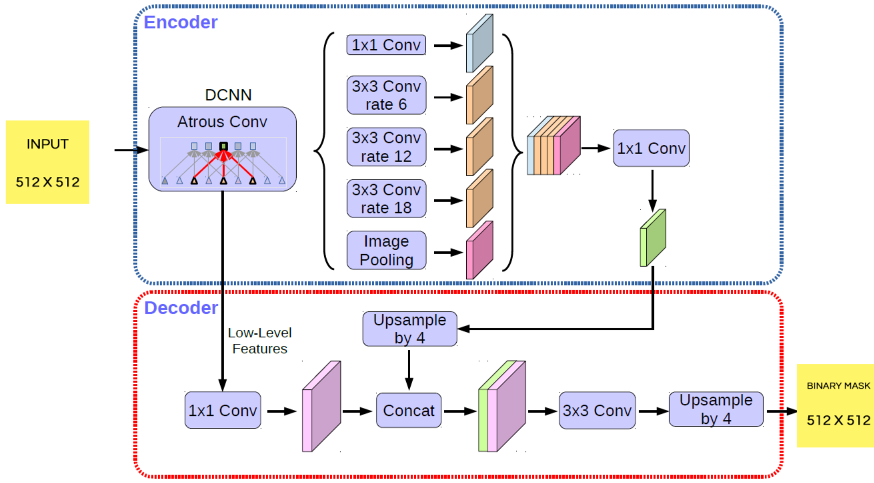

Figure 8: DeepLabv3+ Architecture

Figure 9: Lane Detection

7.2.2 Post Prediction

On the predicted binary mask a series of post processing techniques is applied, such as skeletonizing and median blurring which reduce random errors. The output of the model is a probability map of foreground and background which is to be mapped to real world coordinates. For this problem we employed Inverse Perspective Mapping (IPM). A shortcoming of IPM is that we lose all information of height, hence a mislabelled region at a height would be mapped to a long stretch of wrongly predicted lanes. To overcome this problem we utilized ground plane segmentation enabled by the point clouds of ZED cameras to consider only lanes predicted on the lowest plane.

7.2.3 Efficiency Considerations

- **Model Compression:**
  The Deeplab model is compressed by almost 50% using a simple trick of merging batch normalization and convolution layers. This boosts network inference speed by a significant amount.
- **In-graph postprocessing and batching:**

The post processing operations are made to be a part of the tensorflow graph so that the post processing and model prediction cane be done in one shot all in GPU. Also the two sources of images are batched together to predict in a single shot.

- **MobileNet Backbone:**
  The DEEPLABv3+ model runs on a MobileNet backbone, thus allowing real time prediction and light GPU usage.
- **Reduced Latency:**
  The ZED-ROS system for messages is completely replaced with native function calls to ZED API to improve computation usage and reduce latency in predicting the final binary mask.

7.2.4 Lane Following and Waypoint Generation

To follow the lanes appropriately and be within it throughout the course, the system is commanded to follow intermediate points through which the global GPS goal is to be achieved. The position of these goals in the map are generated using the detected lanes over which a density-based clustering algorithm (DBSCAN) is applied to cluster the lanes and potholes separately. The distance of each cluster from the system's position is calculated. A sampling of the cluster points after forcing a beta distribution provides us points that are farther away from the current position. The midpoint of the line through this sample is chosen as a potential local goal. In case only one or no lanes are detected, a temporal dependency on the previous goal is used to find the appropriate goal. If this continues for long, the goal is switched to the GPS goal to be executed next. Additionally, the robot's current position and orientation are tracked to prevent execution of plans resulting from noisy perception information.

7.3 Planning

7.3.1 Mapping

For autonomous motion, the system requires a good understanding of the surrounding. But since the IGVC course is an unknown environment, this is achieved by a SLAM (Simultaneous Localization and Mapping) technique called openslam-gmapping [3]. Owing to the sparsely filled environment of IGVC course, gmapping in our system is supported with a 3D LiDAR. This expands the bounds of the region that can be mapped and the increases the number of particles from which to sample from for a particle filter based filtering. The 3D LiDAR has an 360°field of view. This helps in keeping track of bot's back as well.

Following the region static map, two maps comprising of the obstacles, ramp, lanes and potholes is composed using data from perception. The maps are analogous to a single channel top view image of the course where each cell is a probability of an obstacle being present in the cell. The first map namely the global map is a large low resolution map covering almost the whole IGVC course. Another map namely the local map is a small, fast updating, high resolution map.

For efficiency, the previous code base has been updated to rebuild the map for only that region which is in current view of the bot. Additionally, a time based decay of cost is introduced to get rid of random noise from perception output data. Classical image processing algorithms have been applied on the maps for noise removal. The maps are used for path planning and trajectory generation respectively.

7.3.2 Localization

Though lately, but we realised that localization has great significance when it comes to autonomous navigation. It becomes more fundamental in a region like IGVC course where SLAM algorithms are unable to perform well. Keeping this in mind, an entirely new system is developed for localization. We have used an Inertial Navigation System (INS) and its associated GPS alongwith wheel encoders for localization. The sensor data is fused using Unscented Kalman filters (UKF nodes) for localization. Only GPS data is made available to the first filter. Its output is supplied to the subsequent filter along with wheel encoders and INS data. The final output on an average has error of 1m in translation and 0.03 degrees in rotation after completing a IGVC sized course. Visual odometry has not been used this year because of its excessive dependence on sunlight and weather conditions.

7.3.3 Path Planner

A modified version of A* algorithm is used as the base of Solo's path planner. A new approach has been used in developing it. Unlike traditional path planners which generate the shortest distance path, we have added special heuristics and checks which penalize the path planner if it tries to go out of the lanes or very close to obstacles.

7.3.4 Trajectory Planner

After the generation of a feasible path, velocity commands have to be calculated to achieve the goal. This is done with the help of Time Elastic Bands(TEB) planner[4]. It optimizes the robot's trajectory with respect to trajectory execution time, separation from obstacles and compliance with kinodynamic constraints at runtime.

The shift to TEB from last year's Dynamic Window Approach(DWA) is because of better performance and ability to execute complex maneuvers.

7.4 Interoperability Profiles

7.4.1 Overview

Interoperability Profile was originally an initiative started by the United States Department of Defense (DoD) to organize and maintain open architecture interoperability standards for Unmanned Ground Vehicles.

With regards to the IOP challenge, we successfully implemented a number of IOP attributes from the JAUS Profiling Rules document in order to make Solo's system JAUS compliant. Since, the software system is implemented and integrated using ROS, we have implemented a JAUS layer to achieve JAUS compliance using the OpenJAUS Development Kit.

OpenJaus is a C++ based middleware toolkit based on the latest JAUS standards. Unlike last year's JausToolSet(JTS), we used OpenJaus this year because of its simpler and cleaner C++ API, a completely open codebase, compliance with the latest JAUS standards, and implementation of several service and message sets.

7.4.2 System Integration

The Jaus layer essentially acts as a communication interface between the Conformance Verification Tool (CVT) running on the Judges Test Client (JTC) and Solo's ROS system.

The Navigation and Reporting Component and the Platform Management Component were combined into a single component which runs on Solo as its subsystem and provides all of the services that were individually offered by these components. To the CVT, this appears as a JAUS component when in fact this component runs as a ROS node and does the interfacing between ROS and JAUS.

The Performance tasks, as well as, the Navigation and Reporting subtask pose as the main problems which require interfacing between ROS and JAUS. For the Performance tasks, which require the physical execution of certain commands, instructions were sent to Solo's system using ROS services. Different tasks, such as setting waypoints, changing maximum travel speed, setting local pose, executing a list of waypoints, etc. were accomplished by calling appropriate services advertised in Solo's ROS system.

7.4.3 Conformance Verification Tool

This year we have also built a system similar to Conformance Verification Tool(CVT). This helped us to achieve extensive testing on Solo before it can truly be called Jaus compliant. Our CVT uses a text-based interface that can be used to find subsystems available on the network and send specific JAUS messages to the subsystem. The development of a makeshift CVT allowed us to carry out tests as mentioned in the IGVC rules to see if Solo behaves the way we want it to. The system also allowed us to perform some extensive tests to figure out problems in the implementation of certain services, along with other network problems we could face, such as dropping of packets due to poor wireless network connection. Every single test, as mentioned in the rule book, was carried out to make sure Solo is Jaus compliant in every sense of the word.

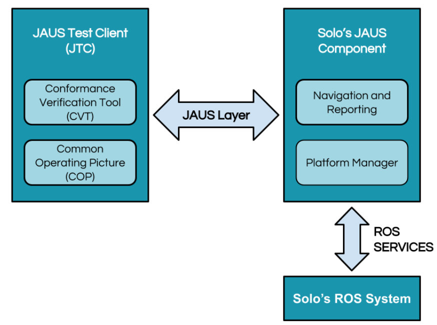

Figure 10: Jaus System

7.5 Additional Creative Concepts

7.5.1 Overwatch

Overwatch is Solo's custom made node monitoring and controlling system. It checks for the status of all running modules of the software system of the bot along with all the sensor connections. It has been designed for easy debugging of errors during testing and identification of points of failures. This also implements a login system for getting access to the main code base of Solo. This ensures the safety of the bot and its code base from potential threat.

7.5.2 Person Tracking

Solo has another mode apart from manual and autonomous mode. This is called the person tracking mode. In this mode, Solo can be made to follow a person keeping a minimum safe distance from him/her. This is done by clustering on 3D LiDAR point cloud and tracking it as it moves. This mode has been added for easy movement of the bot without the overhead of controlling it using a joystick.

8 Simulation

One of the most essential part during the fabrication of Solo was creating a simulation for the robot. A well-designed simulation helps in testing and development of different algorithms extensively before transferring it to the real bot and running any real world tests. The simulator of our choice was Gazebo because of its accurate physics engine and readily available ROS support.

We developed an accurate model for Solo using the differential drive mechanics from the ANSYS model. Since the model planner uses forward simulation to find the optimal path, the model thus created had accurate dimensions, mass and inertial values. The simulation model also helped in deciding the optimal sensor position for every sensor on board. Gaussian noise was added to the simulation sensors to mimic the real world readings as much as possible.

A life-sized IGVC environment was created using the proposed map given in the 2019 rules. Grass plane with powdered texture along with traffic barrels and ramp were introduced. Modelling the world's lighting conditions and friction values accurately helped making the underlying algorithms robust to any environment.

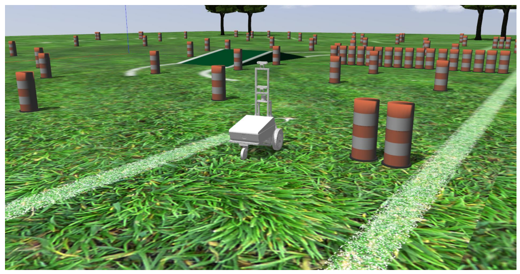

Figure 11: Simulation

9 Failure Modes, Failure Points and Resolutions

Failure Points	Resolution
Mechanical Failure	
The vibration in rough terrain could not be fixed.	Finally to make it further stable, the base frame is made wide with supporting L-structures.
Wheel was tilted at small angles	3D wheel alignment using precision placement of Square pillow block bearings
Electrical Failure	
Sensor are not working as expected	We have replacements for wheel encoders. The remaining sensor have software to perfom re-calibration
Batteries getting drained out during testing	Battery charge indicators have been put which can be constantly monitored without the need of a DMM to make sure that the battery is sufficiently charged
Software Failure	
One of the ROS nodes crashes in middle of the run	Overwatch is a custom made tool which constantly monitors all the nodes and restores them to previous state in case of failure
Lanes are detected at incorrect places	To deal with such noise from perception, the marking of the lanes on costmap is done with a special moving average technique in which a part of the map is predicted as lanes continuously to be marked as one.

10 Performance Testing

Solo was tested in an open grass field with a scaled down replica of the track, with similar lanes, obstacles and complexity as well as in other open areas around our university.

- **Mechanical:** Solo was designed from the ground up on Autodesk Fusion 360. This was followed by load bearing capacity studies and other analysis using the ANSYS workbench. Multiple iterations were made and after discussion with other divisions and testing, the design was finalized and the bot was assembled using lightweight and durable material to ensure structural integrity. The bot was tested on uneven terrain at speed of up to 1.8 m/s and inclined ramps of upto 30°to test its reliability.
- **Electrical and Electronics:** All the circuits and sensors were first calibrated in the workshop to ensure their proper and accurate working before being tested. Regular checks were conducted at the end of each day to ensure the data collected was error-free, and damaged components and wires were continually replaced.
- **Software:** All the algorithms were first implemented and tested on the simulator. For tuning the various parameters, we used ROS bag files which were recorded while it was being manually driven. Each algorithm was then tested on the track individually, followed by integration tests. The parameters for the Kalman Filter and PID were manually tuned and the parameters which gave the most accurate results and worked at high speeds were chosen.

11 Initial Performance Assessments

Solo was completely redesigned for IGVC 2019 and meets all the requirements. From our preliminary testing the bot checks of all the major marks listed below:

- The motors were able maintain the minimum speed and operational speed of 1.2m/s has been verified during testing.
- The bot was able to ascend and descend ramps of more than 15°without any issues.
- Our tests constituted of 1hr of continuous testing and the battery lasted till the end of testing.
- With our current localization setup, Solo arrives near a waypoint with an accuracy between 0.1m to 0.8m.
- Solo is able to handle sharp turns, complex obstacles and follow lanes with ease.
- Recovery behaviours have been tested in various situations including when stuck close to lanes, obstacles and in dead ends.
- All visible obstacles within 1m to 50m at steps of 0.2°detected without any issue.
- On the software front Solo was able to perform lane detection, lane driving, mapping, localization, path planning, motion planning and waypoint handling as well as IOP challenges in accordance to our expectations.
- The various mounts of the bot were able to withstand rocky terrain even at high speeds.
- All sensor readings were within error bounds during testing.

The next few days will be spent in fine tuning our localization, waypoint generation and motion planning algorithms.

References

[1] L. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," *CoRR*, vol. abs/1802.02611, 2018.

[2] T. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *CoRR*, vol. abs/1708.02002, 2017.

[3] G. Grisetti, C. Stachniss, and W. Burgard;, "Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling,"

[4] C. RÃűsmann, F. Hoffmann, and T. Bertram, "Kinodynamic trajectory optimization and control for car-like robots," 2017.