Autonomous & Dynamic Robotics

RS3 Design Report 2008

Presented to the 16th Annual Intelligent Ground Vehicle Competition

Club Capra
École de technologie supérieure
1100 Notre-Dame Ouest
(514) 396-8800 #7779

I certify that the engineering design in the vehicle by the current student team has been significant and equivalent to what might be awarded credit in a senior design course.

Prof. François Coallier, Eng. Ph. D.
Chairman, Department of Software and IT Engineering
Faculty Advisor, Capra
École de technologie supérieure (ETS)

1. Introduction

Following last year's achievements, the Capra team is proud to present an improved RS3. Our vehicle is now more adapted than ever to the various challenges of the Intelligent Ground Vehicle Competition. The team comes forth with lofty goals, with all the effort we invested to bring this improved platform in terms of mechanical, electrical, and software design, meeting these goals is more than likely.

Capra has been evolving in the autonomous vehicle field for many years. It is through innovation and our passion for robotics that we have been able to improve our international standing. What motivates us to build such a vehicle is our thirst for a multidisciplinary knowledge and our drive to apply the most efficient and innovative technologies in our respective fields.

2. Design process

2.1. Methodology

The methodology used this year is based on software iterative methods. This methodology consists of a series of small iterations of 2 to 4 weeks. These iterations start with a meeting to define a set of requirements which are needed to achieve our goal. We then define the most critical or complex needs and start developing and testing the solutions. At the end of the allocated time a new meeting takes place to discuss the success or failure of that solution. This allows us to uncover the areas of risk and to keep the many developing teams aware of our advancements thus minimizing the possibility of incompatible features.

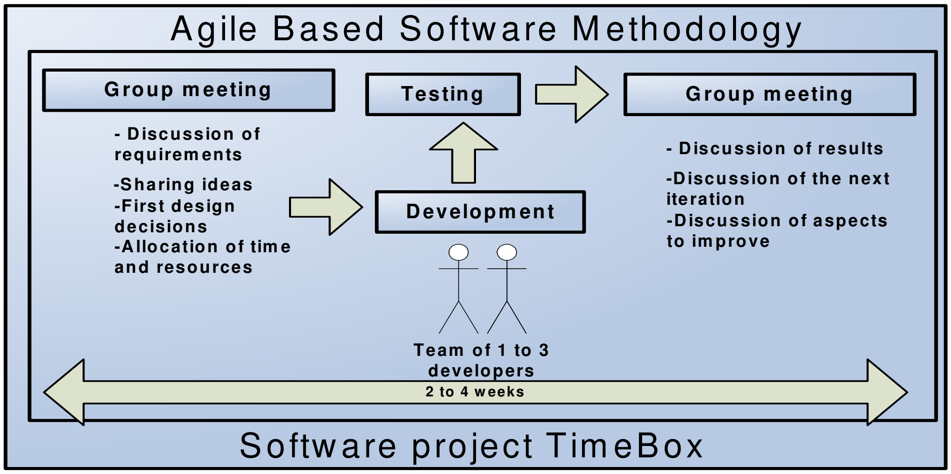

Figure 1: Methodology

Since this technique requires prototyping and proof of concepts we had to use simulators for the mechanical, electrical and software development in order to pinpoint problems and avoid financial losses. The first prototypes were made with the help of Solidworks for the mechanical team, Orcad for the electrical team and a homemade simulator for the software team. Afterwards, we could take the proof of concept one step further either with miniaturized prototypes for the mechanical team or breadboards for the electrical teams. As for the software team the concepts were tested against the simulator.

2.2. Workforce

The team is entirely composed of undergraduate students from ETS in four engineering domains: mechanical, electrical, automated production and software. Here is a chart of their names and domain.

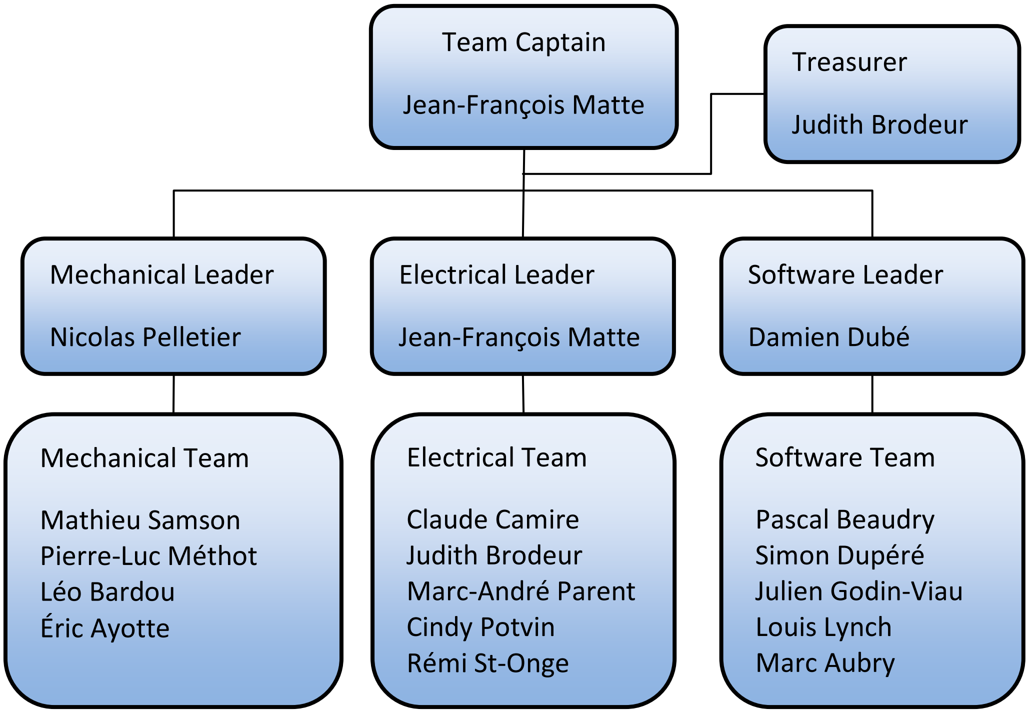

Figure 2: Workforce

3. Mechanical Subsystem

3.1. Structure of the vehicle

With its good design and maneuverability, RS3 is suited for the kind of challenges in the IGVC competition. Furthermore, the good mechanical design doesn't come at the expense of complexity, since we have built RS3 to be easy to work in.

The structure of the vehicle is divided into three parts shown in Figure 4. The upper structure contains electrical equipment and electronics. The bottom structure part contains the drive train equipment, as well as the batteries. Finally, the front structure contains swivel wheel and the bumper.

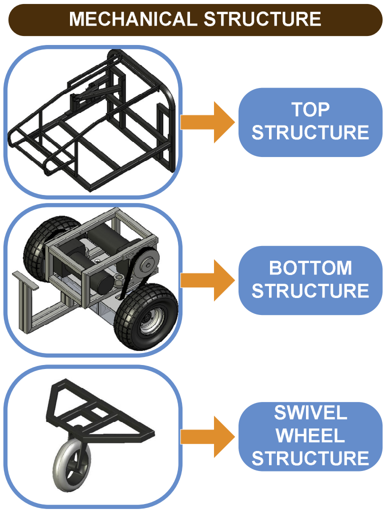

Figure 3

The vehicle is equipped with a propulsion system for a differential zero turn radius. RS3 can easily move even in narrow passages. The chassis is made of aluminum tubes 6061-T6 lightweight and rigid designed to ensure ease of manufacture and reliability. The mechanical design also considered getting the best center of gravity possible. To accomplish this goal we have placed the batteries and motors in order to make the center of gravity as low as possible.

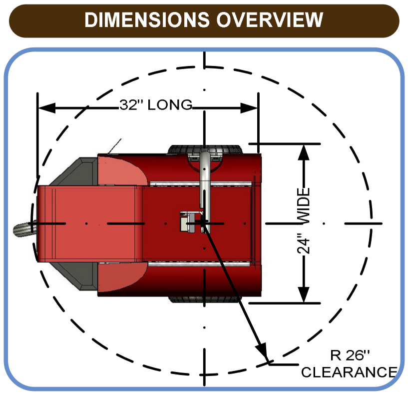

Figure 4

The vehicle's chassis, or top structure, was built with easy accessibility in mind. All the electrical components stored inside can be easily reached. The top structure's dimensions also provide enough space to easily work on components inside the robot, as well as providing room for further upgrades. The size of the top structures also reduces the risk of computer overheating. Lighter tubes of aluminum have been used for the upper frame, making it resistant to the payload.

3.2. Aluminum sheets improvement

For this edition of IGVC, improvements have been made on the outside appearance of the vehicle. At last year's competition, we encountered issues with aluminum panels. Two major phenomenon posed problems. First, the reflection of light on the polished aluminum distorted the camera's captured image. Secondly, panels were electrically conductive. Several types of solutions were available, like paint coating or masking reflective area, but a more sophisticated way was proposed following the Creative Problem Solving Approach. The team decided to anodize exterior aluminum panels. This solution added a layer of colored oxide, reducing the coefficient of reflectivity and adds resistance to wear. Most of all, panels don't conduct electricity anymore. This modification solves problems and also gives a new look.

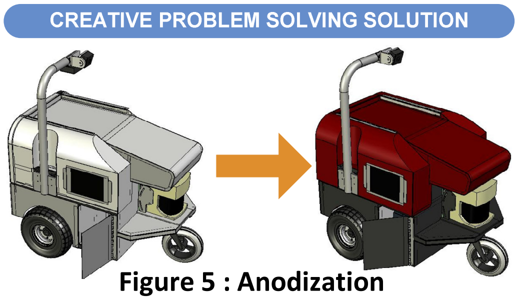

Figure 5 : Anodization

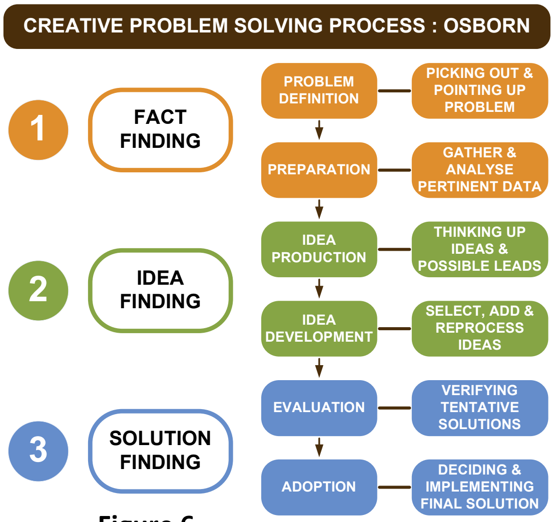

Figure 6

3.3. Vibration isolation

The second problem encountered in the mechanical aspect was vibrations on electronic components. This problem was observed during last year's edition. The electronic components were exposed to vibration and shock when the vehicle moved and created premature wear. As a result, loose nuts and occasionally computer shutdowns could happen. Solutions for

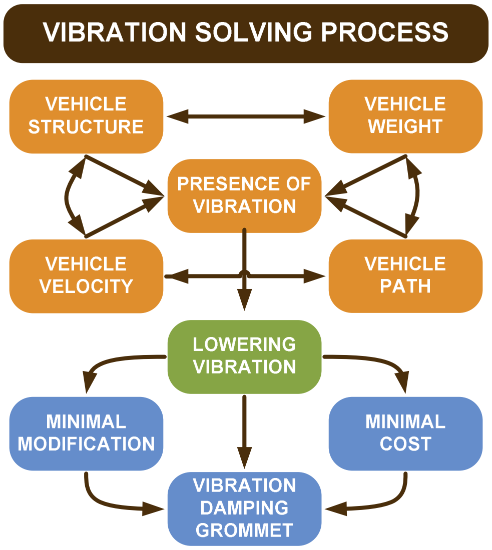

Figure 7

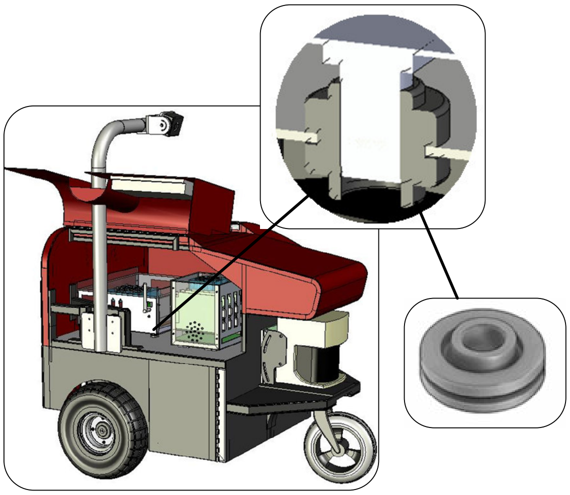

Figure 8 : Damping system

adding suspension mechanisms have been studied with criteria of economy, simplicity and of course efficiency. The team did not want to make major changes to the chassis. The chosen solution was to add vibration damping grommet. This solution is very cheap, assembles easily and reduces vibration in the electronic components. This time also Creative Problem Solving was integrated in design process.

4. Electrical Subsystem

The electrical group is the department that merges software and mechanical systems. The reliability and efficiency are of primary importance to get a good robot behavior. The main objective of the electrical group is to improve the power and precision of our components without sacrificing reliability.

4.1. Computers

We wanted to replace the mini ATX motherboard used in the past because it was starting to show some wear and tear due to the vibrations produced by uneven terrain. Also, we needed a computer with better heat dissipation without using a sophisticated cooling system. So we choose to explore two options: a laptop or an embedded computer.

We choose to use the embedded computer which was more flexible and easier to cool than the laptop. The embedded computer used is the EXT-CD Computer-On-Module by Kontron powered by an Intel Core 2 Duo @ 1.66 GHz. We added a Asus EN8600GT graphic card (NVIDIA GeForce 8600GT chipset), which is required by the software team for their improved image processing. The last innovation regarding the computer is a solid state hard drive. Unlike the old hard drive, these memory devices aren’t affected by vibrations. These type of hard drives are also a lot more flexible, since the card can be removed and plugged into any computer equipped with a Compact Flash reader for quick troubleshooting or reformatting. Overall, it is an optimized solution for the unmanned vehicle world.

4.2. Actuators

We needed motors that could pull and provide the needed speed in rugged and grassy terrain and with low energy consumption. We opted for 2 Bodine brushed DC motors that can produce 1/3 Hp on 24V. They are powered by two 12V batteries at 20A per hour, serially connected to provide 24V. In between, there is a

RoboteQ AX1500 drive. This rugged component can handle 30 A per motor for 30 seconds. With all of these elements, RS3 can pull heavy loads with ease.

4.3. Electrical System

Last year we tried to replace our power supply system with a more efficient one, but it didn't work as intended. The main problem with the power system was the energy loss, which complicated lengthy test sessions and created excess heat. This year we designed a brand new homemade power supply system based on last year teachings. The first step was to find out the energy needs of every components (see table 1) to have an efficient system. As a result the power supply became sturdier and more compact. It event allowed us to start the robot without powering on the computer. This is useful when we need to test something that doesn't require the computer or when we want to move the robot with the remote control.

Table 1: Power consumption

Components	Nominal Voltage (V)	Max Power (W)
PC (CPU =100%)	-12, -5, 3.3, 5, 12	59
Graphic card	3.3, 12	43
Range Finder	24	29
Camera	12	3.6
GPS receiver	12	2.64
GPS antenna	5	.5
3 Controller boards	5	3x1.25
IMU	9	2.7
Others	5, 12	5
Total	-	149

Another benefit of this revised power supply is that we are able to switch between AC power and battery power without the need to turn off the computer. This is a really useful feature when we need to change empty batteries.

Secondly, last year we used a low quality inverter, which we replaced with a pure sine one. The pure sine improved the reliability of the power supply because there are fewer harmonics affecting the electrical parts.

Just like the motor section and the electronic are each powered by two 12V batteries connected in series to get 24V.

4.4. Electronics

Many peripheral are controlled by a microcontroller system. There is a main card that controls the drive and acquires the information from the encoders. Another card is used to control the robot’s fan, and a small LCD screen that show batteries voltage. This technology simplifies the process of checking the batteries voltage and reduces the need for a voltmeter. The last card is used to activate accessories like the horn and the light.

They are all powered by Atmel AVR microcontroller, and the communication between the cards is handled by CAN bus and RS-232. We also use SPI to communicate with some chips like the quadrature counter for the encoders.

As requested, we installed a simple but efficient safety system. There a mushroom like button in series with a relay activated by remote control. We need to activate both of them to give the power to the drive and allow the motors to start. The button is also large and easy to press, making it easily accessible in case of an emergency shutdown.

4.5. Sensors

To acquire the speed of the wheels, we use Grayhill optical encoders with 128 counts per turn. A Sick LMS 291 laser scanner is used to detect obstacles in front of the robot. It gives information on 180 degrees, an ample range for proper obstacle detection. In addition, we replaced our old Point Gray camera for a new ISG LightWise LW-1-1.3 with a Fujinon DF6HA-1B fixed focal (1:1.2/6mm). We can now get better information to detect lines and even obstacles, since the amount of noise has been reduced compared to the previously used camera. Another acquisition is the Novatel OEMV-3 GPS with the GPS-532-C L1/L2 aircraft antenna that replaced our old Garmin GPS-18. The given precision has greatly increased, thus making the robot more suited for the navigation challenge. Last but not least is the Microstrain 3DM-GX1 inertial measurement unit. We use it as a three axis compass, which provides us precise robot orientation, as well as pitch data to detect ramps.

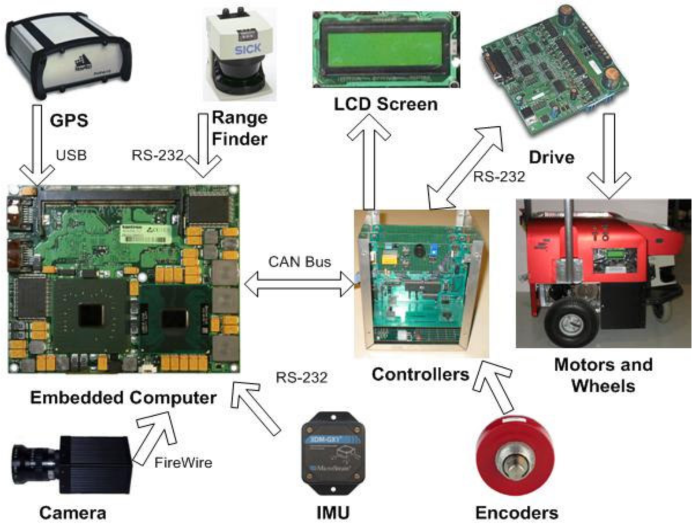

Figure 9 : Intercommunication

4.6. Battery Charging System

Because we use lead-acid rechargeable batteries, we need to manage their recharging process. In the past we always trouble managing the charging cycle and they take at least 4 times longer to charge than the autonomy time of the robot. Also they are heavy to handle and hard to plug in the charger correctly. So we decided to make our own intelligent charging system.

With a microcontroller, we can charge two 24V batteries pack at the same time and the system automatically switches to another battery when the one charging is full. There is a LED system that indicates if the battery is charging, empty or full. With this system, we greatly simplify the process of charging batteries and identifying the status of the batteries. This way, we can leave 8 batteries to charge for a full night and the microcontroller will manage the rotation.

4.7. Electrical Innovations

As we wrote in previous sections, many improvements have been made in electronic department to get a better robot. The new computer is more reliable and powerful. It is faster with its 2 cores, even if each of them is slower compare to the Pentium M 2 GHz which it replaced. It is easier to get a good solid heat dissipater and the acquisition of a Compact Flash card diminishes the vibrating and impact problems that happened with a standard hard drive.

Another major improvement is the GPS system. The Novatel technology gives 4 times more information per second and we can get a precision of 0.7 meters instead of the 3 meters the Garmin gave us by using L1/L2 technologies. We also use the USB port instead of the RS-232 to interface it with our computer.

The new camera, a ISG LightWise LW-1-1.3, produces a better image, with a resolution of 1280x1024 instead of the 640x480 from our old Dragon Fly. In addition, the frame rate has improved from 15 to 27.5 FPS. We still use an IEEE-1394 interface to plug it to our computer since it worked pretty well.

Last one but not the least, is our new power supply. We improved the electric current capability and the system is now able switch between an AC source and the batteries live. It gives more autonomy to the robot, because there is less heat loss compared to the inverter system that we previously used. The space gained in the robot also improved the maintenance and air flow in the robot.

5. Software Subsystem

5.1. Robotic Vision

This year we purchased a new camera, the ISG LightWise LW-1-1.3, which provides us with a cleaner image than with the previous model. The objective is to find the possible lines in an image. Once the picture is taken, a Gaussian filter is used to remove most of the remaining noise. Then we use a blue threshold to get regions corresponding to areas of the image that might be lines.

With these images, we try to discern lines and remove the barrels, fences, sand-trap and left-over noise. In order to do so we use a geometric based approach backed with a statistical classifier to detect which connected pixels are lines. We use the nearest neighbor classifier (KNN) to classify the remaining connected dots. The KNN algorithm was previously trained with a large amount of classified data.

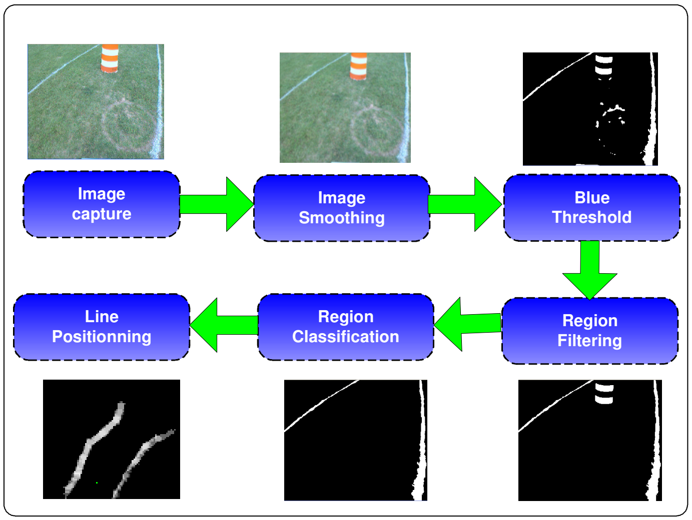

Figure 10 : Vision analysing

The major improvements in the basic pattern recognition are the use of the CUDA library, combined with the Graphical Processor Unit (GPU) of the NVIDIA graphics card. Not only these GPU are well suited to process images but they relieve the CPU from lengthy image processing calculations. The more complex geometric recognition is made using the Matrox Imaging Library 9.0 Alpha, which uses the brand new GPU based algorithms.

Finally, our camera has been calibrated in order to map the detected lines into our obstacle map. With calibration it is easy to map image pixels to real world coordinates. Furthermore we can also use geometric features analysis in order to get the general orientation of a line, providing us with further information for the path finding process.

5.2. Obstacle detection

After lines have been properly detected, the next challenge is detection of solid obstacles like barrels, barriers, or fences. Thankfully, the range finder sensor provides us with precise positioning of any solid obstacles. The difficulty here is not so much precise positioning relative to the robot's position, but filtering out noise and keeping obstacle position even when the robot is in movement.

The solution we propose is keeping the detected obstacle points in a movable grid, which is stored in an image. Using an image provides us with an easy way to map obstacles around the robot, with low computational cost whenever the grid needs to be moved. Each pixel represents a 5 centimeters area. With a 320x320 pixels image, we can represent obstacles as far as 8 meters from the robot.

Furthermore, we can define a fading rate for the obstacle points. This gives us the option to use anything from keeping obstacles in mind for a long period of time, to quickly fading obstacles to reduce the memory cost. This approach makes our vehicle easily configurable for different kinds of challenges. Also, since obstacle points are refreshed at each reading of range finder data, points generated by noise will eventually fade out, while real obstacles will be refreshed.

5.3. Robot positioning

The position used by the system is averaged based on 3 independent sensors. The GPS returns information in the form of the longitude and latitude. The IMU returns the attitude of the robot in the form a yaw, pitch and roll. The wheel encoders return the relative displacement of each wheel.

An estimated position is then obtained using an Unscented Kalman filter (UKF). This estimated position is used throughout the system to navigate, reach waypoints and identify obstacles and lines. We have found this to be not time consuming and a effective method to combine all of our error-prone positioning data.

5.4. Autonomous Challenge Strategy

Since the autonomous challenge is an exploration challenge in which the environment is unknown, our system has no long term goal. The robot will try to create the best path based on sensor information and several heuristics.

- The information used by the pathfinder is:
- An obstacle grid represented by a grayscale image

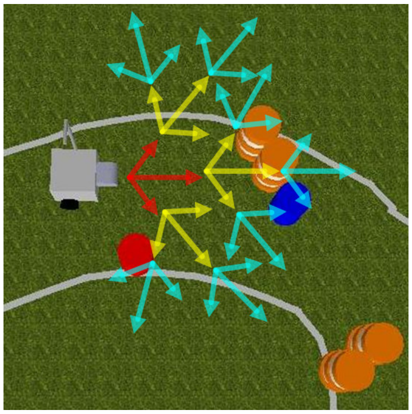

Figure 11 : path finding

- A line grid represented by a grayscale image
- A "been there" grid represented by a grayscale image
- The estimated robot position
- The average orientation of the lines surrounding the vehicle
- The average orientation of the robot during the last 2 meters

The heuristics are:

- Minimizing the position that has a high value in the obstacles, lines and "been there" grids.
- Maintaining the line orientation or at least the average orientation of the robot.
- Minimizing the turn angle between each waypoint.

The path finder algorithm is a basic tree base algorithm in which the cost of every path is calculated with the information and the heuristics described above. A cost is associated with each type of information to give a priority to certain type of information over some other less important ones. For example, the obstacles may have a cost of 1000 and the angular increment a 1. The final cost of the best path will be used for heading and to evaluate the speed of the robot. This method will give us a complete path at each run cycle.

5.5. Navigation Challenge Strategy

Our navigation strategy is pretty similar to the autonomous one. The major difference between the two is that we have a long term goal, reaching every waypoint.

The type of algorithm used is a mix between the autonomous algorithm and a D*.

- The information used by the pathfinder is:
- An obstacles represented by a grayscale image.
- The estimated position.
- The average orientation of the robot during the last 2 meters.

The heuristics are:

- Minimizing the position that has a high value in the obstacles grids.
- Maintaining the average orientation of the robot.
- Minimizing the turn angle between each waypoint.

- Minimizing the distance from the waypoint.

Using this algorithm, the system will try to reach every waypoint successively. The order of the waypoints is given manually. We have chosen the manual approach in order to properly optimize the order of the waypoints and thus optimize the time taken to reach them all. In addition, we allow the system to receive GPS coordinates to reach using Jaus commands. Further details of this feature will be given in the “Jaus Challenge” section.

5.6. Software Architecture

Architecture the backbone of the software controlling the robot, thus it needs to be well taught up. We decided to make an architecture which could stand the test of time by being scalable and robust. In order to achieve this goal we opted for a layers based system. We have the IO/sensor layer, the knowledge base layer and the AI layer. Each layer performs a specific task.

The IO/sensor layer feeds the information to the knowledge base layer and finally the AI layer processes this information to create an optimal path. These 3 layers minimize coupling, so a modification on the implementation of one of them has minor or no impact on the other layers. This makes the system more cohesive and defines different level of abstraction. Furthermore, we wanted the system to be modular. Each module is a plug-in for a specific layer, allowing us to easily change the behavior of the system at load time. This way, it is simple to emulate any sensor and integrate new ones.

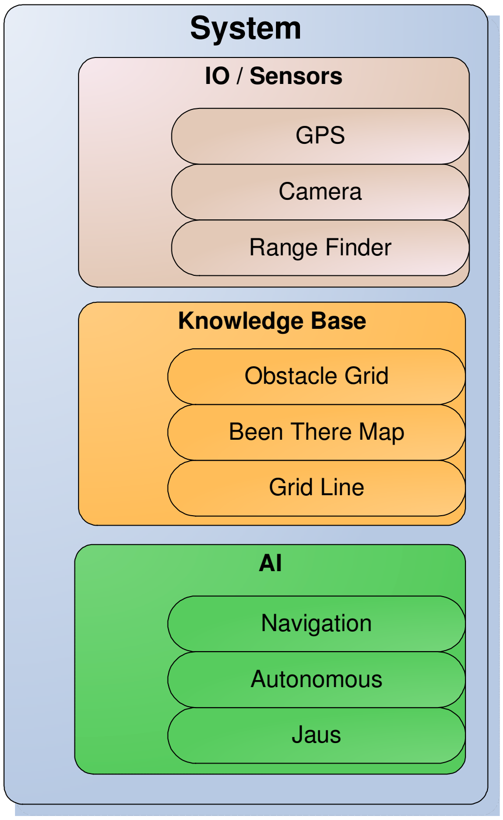

Figure 12 : Software

This architecture has the advantage of allowing the system to parameterize any module for any type of requirement very easily. A configuration file provides the modules with the needed configuration, so we don’t have to manually modify the settings every time we change from navigation, autonomy or JAUSS settings.

5.7. 3D Simulator

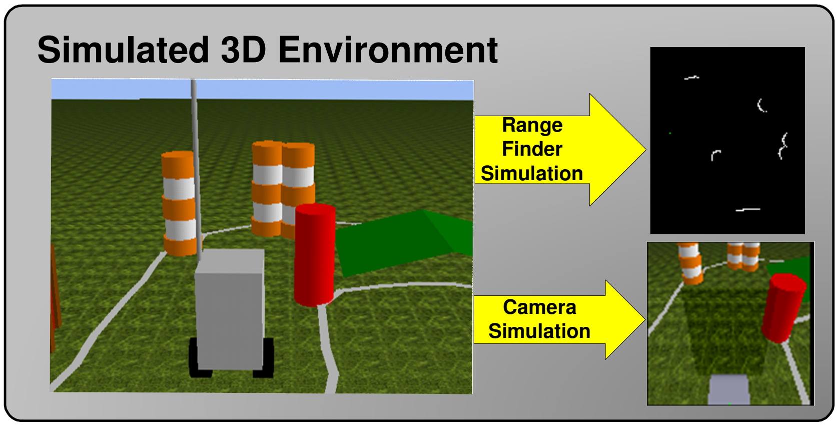

Figure 13 : Simulator environnements

One of the major improvements this year is the development of a 3D environment simulator. With this simulator, we can design and test new behaviors for the robot without the need of any real robot environment. The simulator allows us to reproduce many different terrains and make our robot evolve in competitions like environments.

Our modular architecture allows us to easily define a virtual sensor based on our simulated environment. These simulated sensors are far more accurate than their 2D counterparts. They create a very accurate virtual environment close to reality, greatly increasing our productivity. The robot's sensors, like the camera or range finder, are also simulated and follow closely the real behavior of these sensors. This tool allows us to focus on debug and improving certain parts of the AI by placing it in a specific situation. It helps debugging because we can easily monitor the state of the environment and the knowledge base. With this tool multiple developers can be testing the robot at the same time in different contexts.

5.8. JAUS Challenge

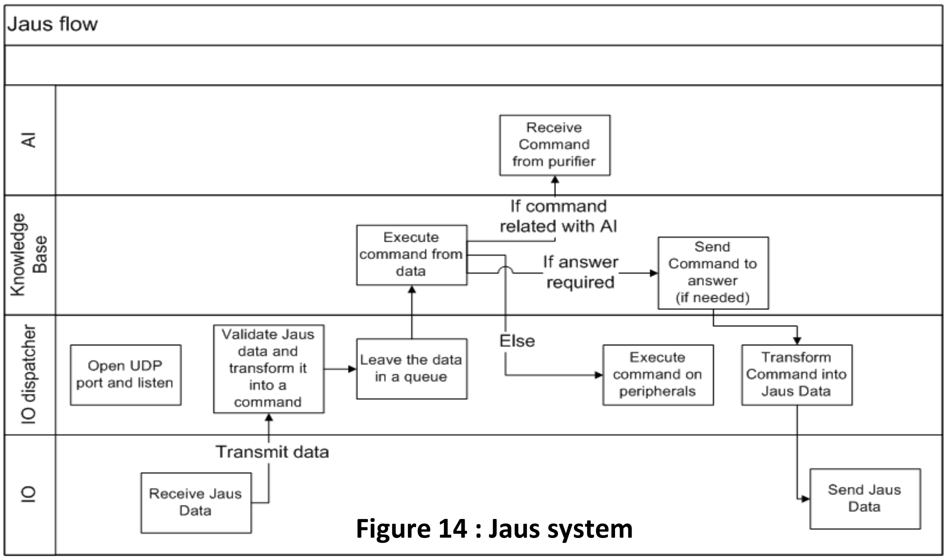

Figure 14 : Jaus system

In previous years, we always participated in the JAUS challenge. So this year we are taking the new challenge with enthusiasm. Our current layer-based software architecture made the addition of a JAUS module simple since it was designed for easy extendibility. Previously we had a module that interpreted JAUS commands and was independent of the currently running AI. This year things changed, we had to give waypoints to out robot for him to reach. Thus we needed to use the AI developed for the navigation challenge. Thanks to our well constructed architecture doing so wasn't difficult.

Our implementation of the JAUS module is simple. We first receive the packets on the computer's Wi-Fi card. We open a listener on the computer that receives all of the data transmitted to an aforementioned port. A module is then used to filter the incoming data, it will check for the correctness of the message, and whether or not our vehicle is the intended recipient. We will keep the valid messages in a queue, where another module will be responsible of executing the command. If the command requires AI, like reaching a waypoint, then the request is transmitted to upper layers, where the AI layer will execute the action. If the command requires a response from our vehicle, we use another module to generate JAUS messages, which will then be sent to the JAUS client using our network card.

5.9. Software innovations

Last year we got our best results yet with a 5th place in the autonomous challenge. Now closer than even before to a first place, the software team was able to design from a good starting point this year. Our artificial intelligence for the autonomous challenge performed nicely, but was unfortunately costly on computing power. It was our main goal this year to reduce the processing time of certain components.

The most significant performance issue with previous years was the image analysis. Heavy on computational cost and ineffective in certain situations, our previous solution wasn't adapted for the autonomous challenge. That's why we chose to design our own basic, speed-optimized image analysis library which uses the GPU. While we had plenty of image treatment libraries to choose from, we ultimately decided to design our own using the CUDA library from NVIDIA. Therefore we now have a GPU accelerated library, which contains the functions we need in our image processing operations. This new system greatly reduces the computing power needed for image processing, and takes advantage of our GPU's computing power, thus leaving more time for the AI to do its job.

Another aspect we added to the image processing is machine learning with a nearest-neighbor classifier (KNN). The line classification training can be done on the competition field, making it suited for the challenges. With a good collection of line examples, we will have a shape classification technique that can effectively identify the lines by comparing them with previously acquired line shape data. Given sufficient training time, the use of a statistical classifier makes our vehicle more apt to recognizing many obstacles in different landscapes and lighting. Also with the use of geometric features to classify possible lines we are robust to changes in line orientation.

Also the use of a 3D simulator is a significant innovation, providing the AI developers with a testing environment that is close to reality. So before we even get to the competition fields, we are confident that the AI behavior we observed in simulation is close to reality. The use of such an effective testing environment makes the AI developers far more productive.

6. Costs & Technical chart

The following table represents the overall cost for RS3.

Table 2: RS3 main components cost

Component	Detail	Cost paid
IMU	1500$	1275$
Wheels	290$	0$
Structure	890$	500$
Engines	900$	600$
Encoders	340$	120$
Camera	1000$	500$
GPS	9600$	1800$
Range Finder	6500$	2550$
Electronics	800$	600$
Batteries	660$	250$
Mechanical	400$	400$
PC	1450$	450$
Total	24330$	9045$

All members worked hard to gain sponsorship deals with industrial partners. Those deals offer a good return on investment. For example, Matrox graciously gave us development licenses of their MIL library to aid in the lines and fences recognition. As a result Matrox has hired many members for internships, thus having a smaller learning curve of their system than regular interns.

Table 3: RS3 specifications

Item	Description
Speed	5 m/h
Torque	80 lb/in at the wheels
Acceleration	1 m/s^2
Ramp climbing ability	Incline of 12 degrees
Reaction time	1/10 of a second
Battery life	2h for electronic and 3h
Obstacle Detection Distance	10 m with the laser
Line Detection Distance	4 m
Accuracy of waypoints arrival	0.5m to 1m

7. Safety, reliability, durability

7.1. Safety

As required by the competition’s rules, RS3 comes equipped with the requested safety features: an E-Stop and remote control activation. The e-stop is within easy reach at the top of the robot, and can be activated by pressing on it even slightly. The remote control has a range of 18 meters, allowing for a quick response should the robot be stopped remotely in case of an emergency.

We have also implemented a safety measure if the robot’s communication fails. A software watchdog has been added, and the robot will stop if no new direction and speed commands have been sent for a period of time.

We changed the configuration of our batteries and their connectors to simplify their use. Now, people can drop a metallic component over the batteries without the risk to short them. Our new connectors simplify the way to charge them and to plug them in the robot. That security issue was a big problem in the past.

7.2. Reliability

The electrical team this year has added a new pure sine inverter. The benefits of such a device are that the electrical components will be less damaged over time. Also the electrical wiring uses an easy to understand color code. This code prevents misunderstood and anyone can work with the electrical components without the risk to break the parts. Also, the wheels are propelled by the motors with straps and not chains, which require less maintenance.

7.3. Durability

The main issue we had to deal with is reducing the vibrations, which had the unfortunate consequence of damaging certain components. Also, we have made some changes to the camera’s support pole to solidified it

As we said above, the anodization process of the sheet metal gives more roughness to them against scratch and marks.