RS3 Design Report 2007
Presented to the 15th 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

Capra is a scientific club from École de technologie supérieure (ETS) based in Montreal, Canada. In 2006 we presented a brand new robot which gave us a more stable and an easily modifiable platform. With this new robot, our team was able to achieve our best results since our first participation in the competition. With satisfying results in 2006, the team has identified the most urgent improvements and has achieved them over the course of the year. With its improvements in all aspects, RS3 is now fully capable of providing a good competition against the best teams.

This year, we are coming back with an improved version of the robot. We made modifications to the platform to make it easier for the robot to go through obstacles, ease the cooling of the computer, reduce power loss and facilitate the maintenance of the vehicle. Also, we've added a few inches of extra clearance to the robot, so it wouldn't be slowed down as much by high grass. One of the major re-factoring was the software part. The software team developed a brand new architecture that easily adapts to different AIs, and also accelerates the developing time and making it easier to implement new concepts and add sensors.

2. Design process

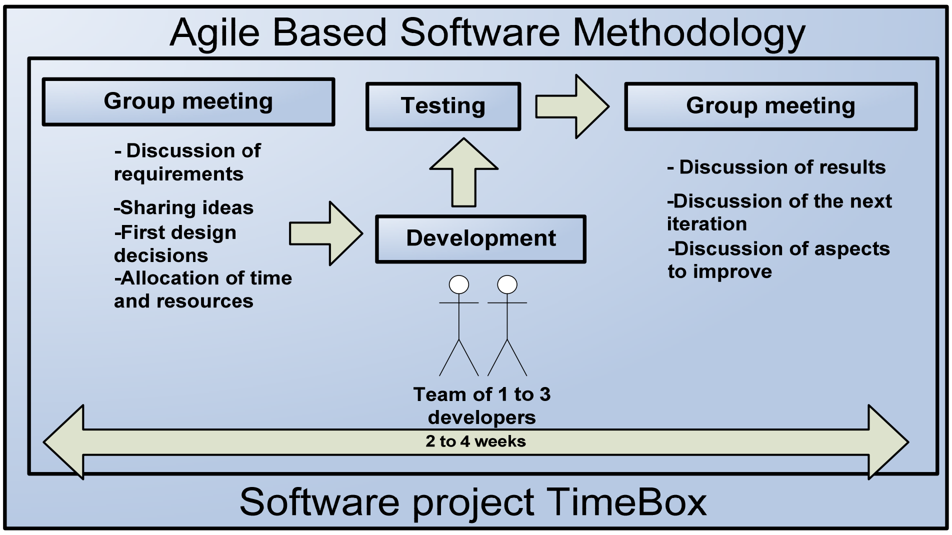
2.1. Design Team Structure

The team is entirely composed of undergraduate students from the ETS in three engineering domains: mechanical, electrical, and software. These three domains are the main areas of expertise of the students of the Capra scientific team.

2.2. Design Methodology

Since the team comes from many engineering backgrounds, they are aware of the importance of a good project design and methodology. The team members are familiar with various design methodologies, and therefore were able to weight the pros and cons of each one before deciding on the one to use.

The methodology decided is based on the agile software development methodology. Agile is a methodology that encourages development in short time periods, and thus is an iterative process. Team members were given small projects, or parts of a larger one, to work on. Having them allocated in short time periods helped the team keep track of progress. In accordance with the agile methodology, these short time spans didn't only include the conception phase, but also other important aspects, like design, and testing.



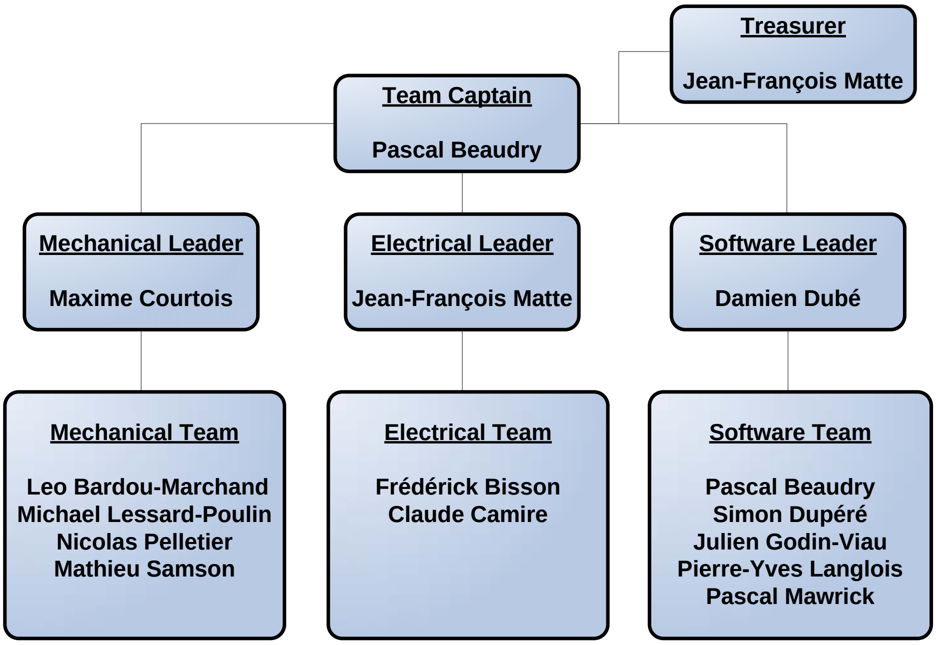
The entire software team also got to participate during the various design phases, allowing proper analysis of all solutions, as well as sharing and discussing ideas. It was also strongly encouraged that group members work in the group's room, as to get easy feedback and help from other persons easily as the projects moved towards completion.

2.3. Production Phase

On the software aspects, the work planned this year was separated in three main components. The first one was the creation of the architecture. The second was the implementation of a system to read and calculate sensor data based on the architectural specifications. And the last one was working on the design and conception of artificial intelligence solutions. The architectural phase was the first priority, as its creation would provide the basic features for the AI developers to get started. After these interfaces were implemented, developing the AI was easier and the interfaces made sure that the code would always work the same way and facilitated the test phase.

The electrical and mechanical teams were also given schedule to follow when they were planning on upgrading some of the robot's features. These time boxes were made in accordance with the software ones, so that a testing period for the artificial intelligence couldn't occur when the robot was undergoing major electrical or mechanical maintenance.

2.4. Team Organization



3. Mechanical Design

Capra has developed a compact vehicle according to the design specifications of the electrical and software teams. The idea is to build a practical and reliable structure. The chassis is made with 6061-T6 aluminum tubes welded and bolted to form two distinctive and very light sections. The upper part is used to contain electronics devices and support the payload. The inferior section holds the drive train and batteries to keep the center of gravity as low as possible. Both sections are covered with aluminum sheet metal and are designed with easy access doors. Design practices of the team are focused to get a better mobility, reliability, durability, and the best robot space management.

3.1. Platform Mobility

The differential wheel configuration of the vehicle allows efficient turning without interference. This design provides zero-turn radius and a good maneuverability for the course in zones where there are obstacles to avoid, making it easy to fit in even the tightest spots. For a better mobility, the drive train section, including batteries and the motors, is positioned to have the center of gravity located in the center of the differential wheel and as low as possible. This gives a smaller inertia and a good stability in rough ground. However, last year's platform had not enough ground clearance in order to face high grass and small obstacles. The mechanical design team came with a solution to add 1 inch of ground clearance without changing the motor assembly. This modification consists in horizontally flipping over the bearing and shaft assembly. To realize it, we built an all new bearing fixation plate and did some small modifications to the main frame.

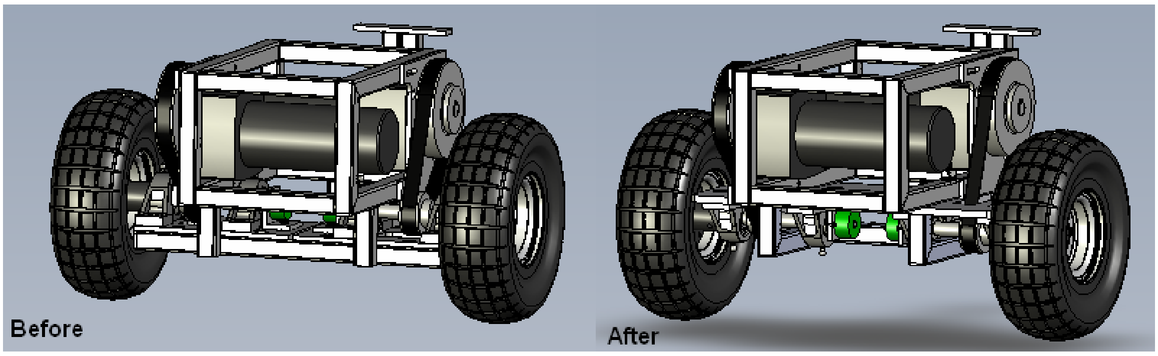
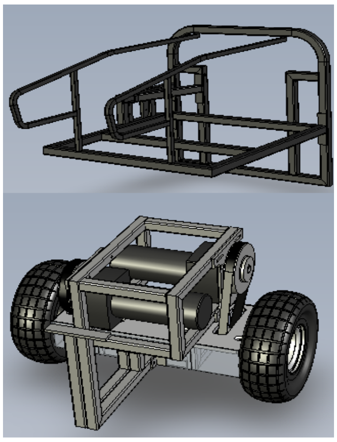


Figure 1: Before and after ground clearance modification.

3.2. Practical chassis

The chassis of RS3 is designed to be a strong and light modular frame. This geometric design makes the maintenance easier and provides a sufficient space to install all required components inside the vehicle. The upper frame section is easily removable and covered with aluminum sheet metal. It is optimized with lighter tube and can support the payload.



The lower assembly contains everything related to the drive train. The upper one contains the electrical and computer systems. The chassis is made of welded 1 inch aluminum square tube. We estimate the weight of the robot is 170 lbs. The overall dimensions are 36in long, 24in width and 62in high.

Figure 2: Chassis

3.3. Major upgrades

The mechanical team, in collaboration with the electrical team, modified the original casing of the computer and electronics devices. The goal is to optimize air flow and to have a better space management in the robot for future upgrades. Instead of mounting all the electronic boards into the original computer case, we design two brand new cases. They are shown in the figure 3.

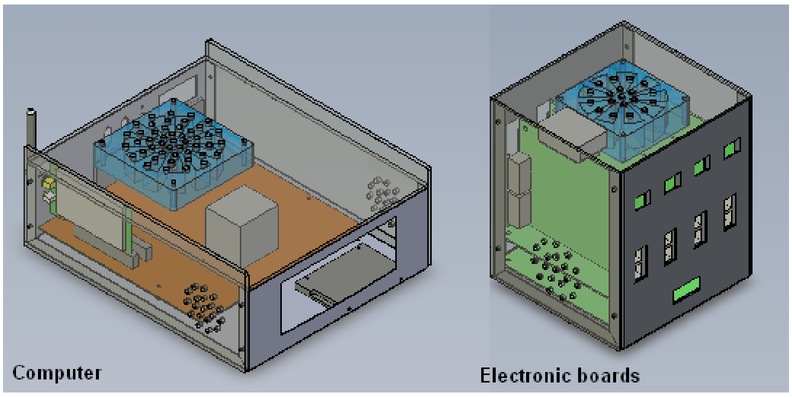


Figure 3: Computer and electronic board cases.

3.4. Physical Parameters

The drive system is made of two 1/3Hp electrical motors with a winding of 24 volts. A gear ratio of 1:8 at the end of the motors offers a maximum torques of 176 lb-in at the wheels. With these specifications, the robot can climb an incline plan of 12°, manage an acceleration of 3.28 Ft/s^2 and reach the maximum speed of 5 miles per hour.

4. Electrical System

4.1. Safety

For safety measures, two different systems can be used to stop the robot in case something goes wrong. The first one is a red e-stop button, located at the back end of the robot. Whenever the button is pressed, the power relay will be deactivated and thus the motors will stop.

The second safety system is a remote control, which activates the same line than the e-stop. To activate the robot, we need to release the push button and activate the relay by remote control to let the current go through the motors.

4.2. Electrical System Layout

This year, RS3 has four different electrical systems. A new power supply is placed in the top front of the robot. With this new configuration, the efficiency is now 15 to 20 percent better than last year. The gain will be noticed as an increase in autonomy and a reduction of heat dissipation inside the robot.

The next group is the computer. Just like last year's competition, we will use a mini ATX mother board with Intel Pentium M 2.2 GHz and 1GB of RAM. The new part for the computer is its case that is completely modified. The new one is smaller and will control the temperature more efficiently than the old one. Also, the connectors are better located on the box to facilitate wiring.

To optimize the computer's case, we had to remove the controller and I/O boards. So we built an entire new box for our controller and I/O cards. Because all of them use Atmel AVR micro-controller, the amount of electrical current consumed by the cards is quite small. So the total heat produced by the boards is not a problem. In this way, we installed them in a smaller and more compact box. The boards have been redesigned to correct the last year bugs and add some new features.

Finally, the last group is the one for powering the motors. There's no big change for this one. We still use a RoboteQ 1500 drive that is controlled by the controller's board and there is a relay that activates the motors by the safety system, since it has been reliable to use.

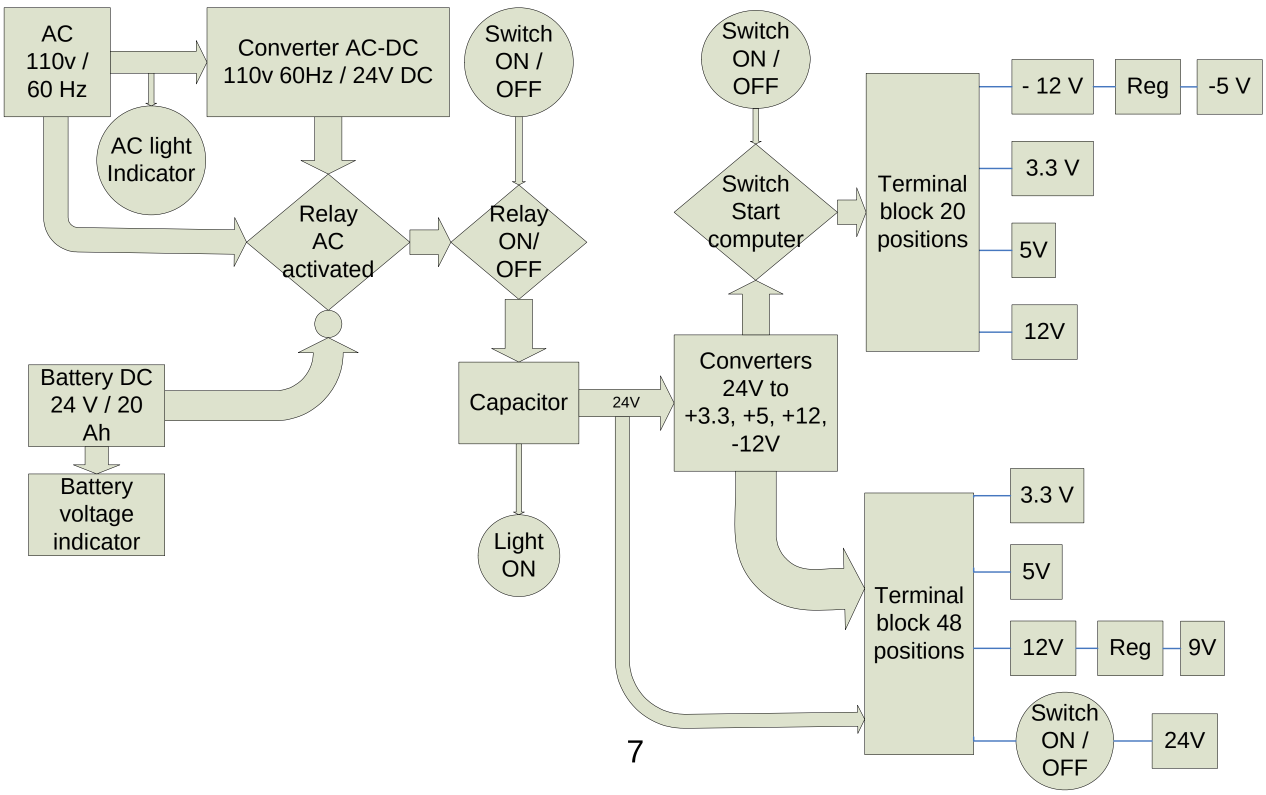
4.3. Motors

RS3 is propelled by 2 DC motors selected by both electrical and mechanical engineering teams, since the autonomy of the vehicle depends on its power consumption. Safety components were also considered with the use of a relay for each E-Stop system to cut the power source from the engines. The model #6884 24V motors bought from Bodine Electric are protected by fuses. The team created a specific socket on the motors' shafts itself for the encoders.

4.4. Power System and System Integration

The electrical and motor systems are powered by the same 12V sealed lead-acid batteries at 20A per hour, serially connected to provide 24V. Using the same set of batteries simplifies their management. We picked this type of batteries for their efficiency, durability, weight and price ratio, since we can go up to having an hour and a half of autonomy with the same set of batteries. In addition, a switch disconnects the system from the batteries when the batteries' compartment's door is opened to prevent sparks while changing batteries.

Two means of powering the computer and sensors system can be used: the AC input for development and testing purpose, or the 24v batteries when the vehicle needs to go autonomous. This makes sure we don't waste battery power when we're running tests. As mentioned, the power supply works differently than last year. We added an AC-DC converter



that produces 24V with a current up to 8.5 A. From this, we pass through a relay that is activated if AC is present and give the energy to a series of DC-DC converters that produce all needed voltages, as shown in the upper figure that describes its conception. Those components have an efficiency of 80 to 95%. Also, we don’t need the 12V-24V step up converter anymore. If there’s no AC, the relay is set by default on 24V batteries. The power is accessible through a 48 ports terminal block for the main electronics and another set of terminal block is dedicated to the computer. The last set of terminal blocks are powered only if we boot the computer. With this system we don’t need to boot the computer to use the remote control when we need to manually move the robot. We added a voltage indicator for the batteries which is able to show the voltage on a LCD for both batteries and sends this information to the computer. This way, the robot is able to warn us when batteries are low.

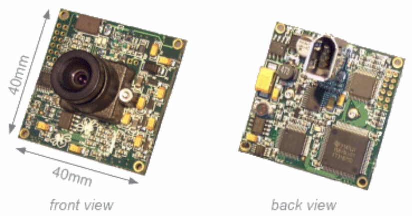
4.5. System User Interface

For the sake of simplicity, there’s a switch to power the robot without giving power to the computer. This way, we can control it remotely. If we want the robot to be autonomous, we push a second button which starts the computer. In both ways, we need to deactivate the e-stop button and activate the remote e-stop so the robot is ready to move. Finally, we can see the batteries voltage on the LCD screen simply by pushing a button.

5. Sensors

The team’s autonomous vehicle relies on a variety of sensors for optimal detection of the obstacles to avoid and precise positioning. The sensors used in the previous years’ competitions have proved to be effective. As an improvement for this year’s competition we have purchased an Inertia Measurement Unit. The table below explains the broad details of sensors used by RS3.

Device	**Characteristic**
3DM-GX1 Inertia Measurement Unit 	3 orthogonal DC accelerometers. 3 angular rate gyros. 3 magnetometers. 360 degrees of angular motion on all three axes. 76Hz update rate.

Device	**Characteristic**
Sick LMS 291 Range Finder  	Up to 360 distance samples on a 180 degrees angle. Maximum range of 80 meters. Precision of 10mm.
PointGrey FireFly 2 Camera  	Resolution of 640X480. Frame rate up to 15 FPS. FireWire (IEEE 1394).
Garmin GPS 18 5Hz  	12-parallel-channel. WAAS-enabled. Update rate of 5 Hz. Accuracy: Position: < 3 meters (WAAS). Velocity : 0.1 knots RMS.

A combination of all these sensors makes the vehicle suited to handle all the possible obstacles at the IGVC competition.

5.1. Digital Camera

Purchased in 2006 and having shown its effectiveness during last year's competitions, the FireFly 2 camera will once again be the hardware of choice for all vision related things. The camera is used for quality image display, and is used with collaboration of the Matrox Imaging Library to detect lines.

5.2. Laser Range Finder

A commonly found sensor in plenty of autonomous vehicles, the Sick brand laser range finder provides precise and accurate readings of nearby obstacles. For quite a few years now, it has been the sensor used for detection of barrels and other solid objects to avoid.

5.3. Global Positioning System (GPS)

A vital element for the navigation competition, the Garmin GPS 18 provides the autonomous vehicle a trusted and reliable input of positioning data. It can also serve as a tool to adjust the positioning data during the robot's run.

5.4. Inertia Measurement Unit (IMU)

The Inertia Measurement Unit is the most recent sensor purchased for the vehicle. Not only does it serve as a replacement for the compass sensor, but if offers new and useful features. First, the IMU includes a gyroscope and an accelerometer, which gives the robot information about its angular and linear speeds. The information gained from these sensors can be used to detect if the robot is drifting off from its planned destination. Secondly, a magnetometer that feeds the machine attitude data is integrated within the IMU, thus making it convenient to detect whether or not the robot is going up a ramp by inspecting the pitch data.

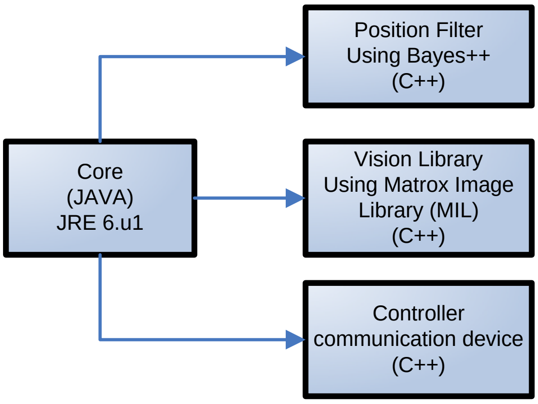
6. Software

6.1. Software operation

The software is composed of a core Java6 component and several C++ components wrapped using the Java Native Interface (JNI).

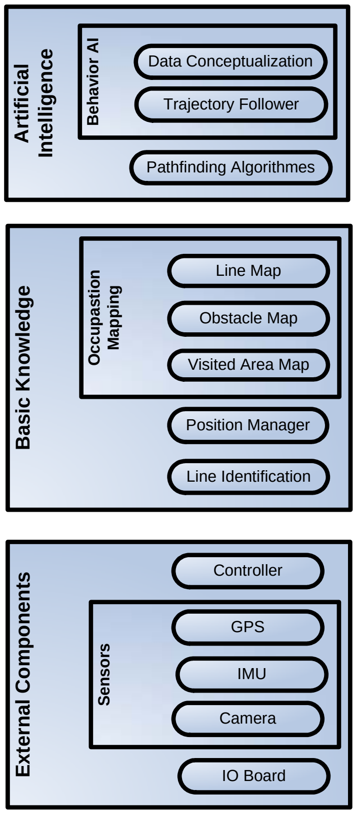
The Matrox Image Library, a specialized library for real-time image processing, was used for all vision related computation.

For the positioning filtering, Bayes++, a specialized library for unscented filtering, was used.



6.2. Software Architecture

This layer based architecture is composed of three layers.



Artificial Intelligence:

The top layer is responsible for the decision taking. It is responsible for every action taken by the robot.

Basic Knowledge:

The middle layer is responsible for the data mapping and is used as a memory for the top layer. The position filtering and the line identification are also done at this level.

External Components:

The bottom layer manages the connections to the different sensors, I/O board and controller. Every component in this layer can be simulated for development purpose.

6.3. Positioning

The positioning and attitude of the robot is based on three (3) independent sensors.

The current attitude returned by the IMU, in the form of the yaw, is used without any filtering because it’s already stabilized using the gyroscopes. The pitch and roll is also used to know if the robot is in a slope.

The position estimation is an issue because every sensor has a different type of error. The GPS has a drift error that is less than 3 meters in as WAAS enabled mode. The position returned by the encoders can accumulate errors over time due to the wheel skidding. The IMU linear acceleration and angular rates are noisy because of the poor quality of the MEM based accelerometers and gyroscopes.

To integrate data from every sensor, it's important to convert them all in the same coordinate system. The data from the wheel encoders, the linear acceleration and angular rates are rotated using the attitude to be in the same orientation as the GPS.

The resulting data can then be passed in an Unscented Kalman Filter (UKF) to obtain a position estimation that minimizes the noise on every sensor. This makes the robot's positioning more accurate, despite the precision issues of individual sensors.

6.4. Mapping

Another important improvement is the extensive use of mapping. All information that comes in from the camera (lines) and the range finder (obstacle) is stored in maps. Every pixel of these maps corresponds to the probability of occupation by a line or obstacle at this specific position. The map covers 15 meters around the robot. To reduce the required computing, the only information that is updated is the region which is currently observed by the sensors. These maps help to memorize important information over time and standardize the way information is passed to the AI.

6.5. Line Detection

The line detection is done in 4 simple steps.

Step 1: Smoothing + Saturation

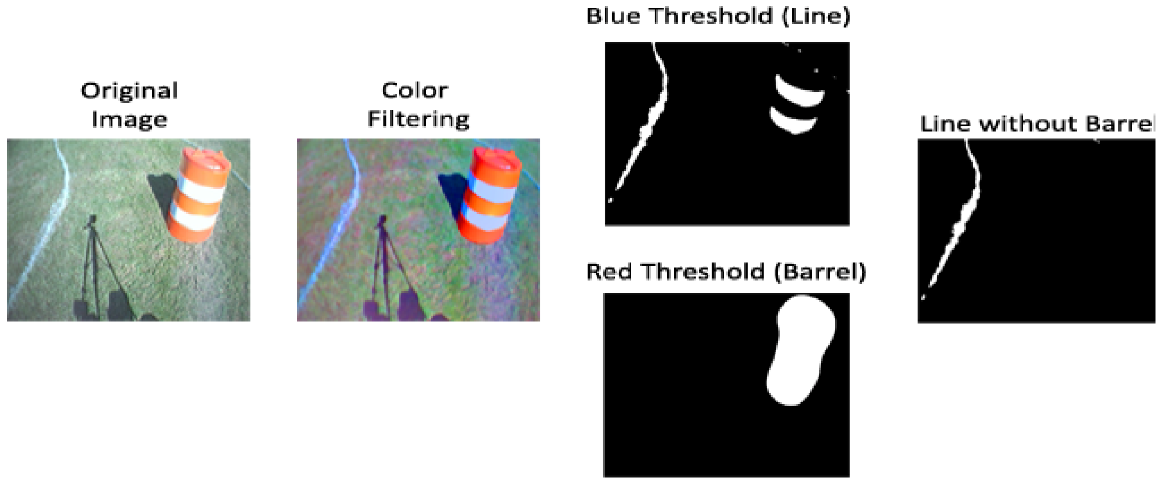
This step is important to prepare the image for the lines and the barrel identification. The first step is to smooth the image with a median filter. The second step is to increase the saturation of the image to intensify the colors.

Step 2: Blue threshold

A blue threshold will isolate the white elements of the image (Lines). The resulting is a binary image. An erode operation is done after the threshold to eliminate small and unattached white elements in the image.

Step 3: Red Threshold

A red threshold will isolate the red elements of the image (Barrels). The resulting is a binary image. An erode operation followed by a binary close operation is done on the image to attach the unattached parts of a barrel.

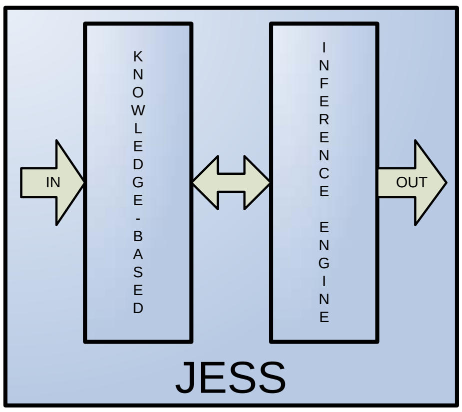


Step 4: Line Isolation

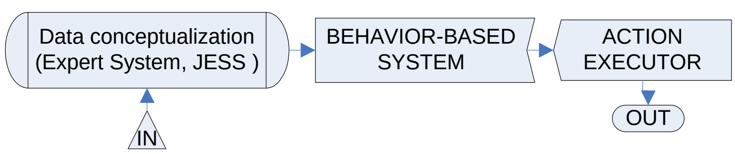
A binary subtract is done on the line image with the barrel image so there is no resulting elements from the barrel in the final line image.

6.6. Artificial Intelligence

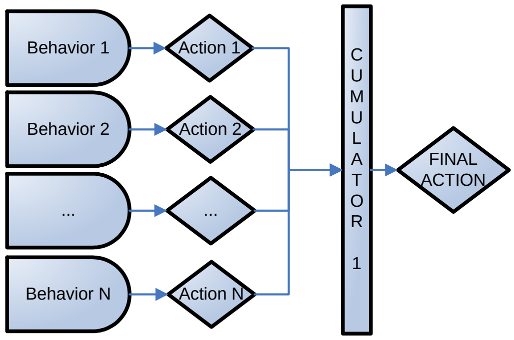
To make the development of artificial intelligence easier, the software team has integrated an intelligence system as a system linked to the most recent software architecture. The artificial intelligence is split in three parts: inspection of nearby areas, creation of an action, and execution of the said action. A behavior based structure is used to generate actions. We use the symbolic approach to analyze what's going on, and then change the acquired data to concepts.



The conceptualization module is divided in two sections: knowledge-based and a rule-based engine. The behavior based structure receives all the concepts generated by the



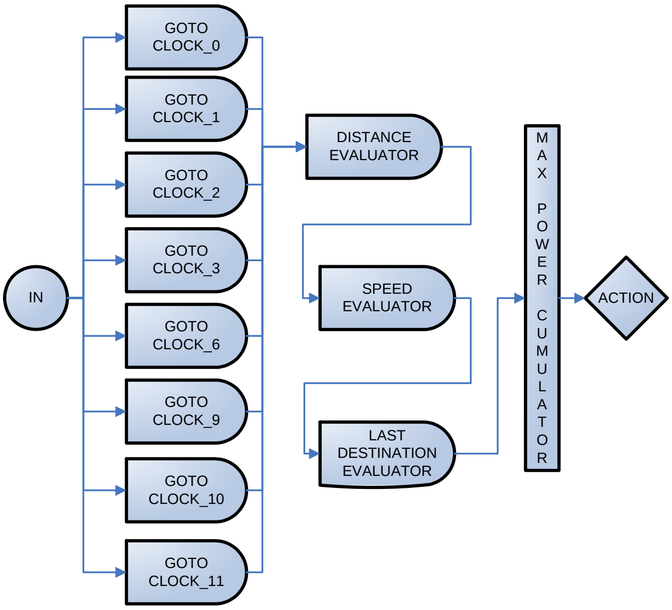
conceptualisation module and generates a final action for the vehicle to do. For the creation of actions, each behavior generates one action depending on their specialized tasks and the concepts contained in the knowledge-based section. The actions coming from the behaviors are then sent to a cumulator, who has the responsibility to generate the final: either by choosing one of the previously generated actions, or by creating a new one with the sum of all previous actions.



6.7. Autonomous Challenge Algorithm

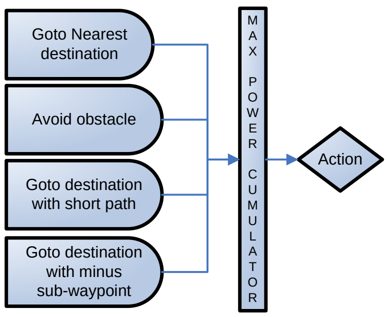
When the vehicle is in autonomous mode, a behavior is run for each direction the robot wants to go.

The behaviors thus create an action if it can be executed. Then, the previous actions are then sent towards another behavior that calculates the maximum distance each action can go up to. Following that, a behavior will check the complexity level associated to execution said action, and thus picks appropriate speed for the action. Finally, a last behavior is used to add a weight that can alter the strength of the chosen operation. The last cumulator picks the final action to be run in an exclusive way, thus picks only the action with the highest weight to send to the action executor.



6.7.1.Navigation

For the navigation competition, the robot has a sequencer used to manage the different steps of the mission. As long as the expected destination is not reaches, the robot will keep moving towards it. The choice of the best position is still done using a cumulator that picks the action with the highest calculated weight.



If the robot heads towards an obstacle, it will avoid it instead of continuing its path towards the waypoint. Waypoints are defined using a converter that will change GPS coordinates to distances relative to the robot's starting position. The vehicle uses a module to calculate the best path, based on what has been previously detected.

6.7.2.Jaus

Another advantage of our current architecture is the ease with which we can integrate new modules, like Jaus. Our current system has a module whose responsibility is listening and reading commands received on the UDP port. The module then verifies that it is a Jaus message, and that it is intended for one of the robot's components. The message is then read by the addressed component, and turned into an operation that will be treated by the system.

7. Cost

The following table represents the overall cost for RS3. All members worked hard to gain sponsorship deals with industrial partners. Those deals offer a good return on investment. For example, Matrox graciously gave 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.

Item	Price
IMU	1500$
Wheels	290$
Structure	890$
Engines	900$
Encoders	340$
Camera	320$
GPS	200$
Compass	50$
Range Finder	8000$
Electronic Components	800$
Batteries	660$
Timing pulley and bearings	400$
Embedded Computer	1450$
Total	14300$

8. Conclusion

This year, our robot is smarter, has heightened awareness of its location, has improved its power consumption and is more agile thus giving us the capacity to exceed ourselves like never before. We respected our deadlines by using techniques to accelerate the development and new tools. Our robot is now ready to face the challenges of the IGVC competition with the best mechanical, electrical, and software components since the beginning of Capra. Not like earlier competitions, we are now ready for the top five.