RS3 Design Report 2006

Presented to the 14th 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.

François Coallier Eng., PhD.
Chair, Department of Software and IT Engineering
Faculty Advisor, Capra
École de technologie supérieure

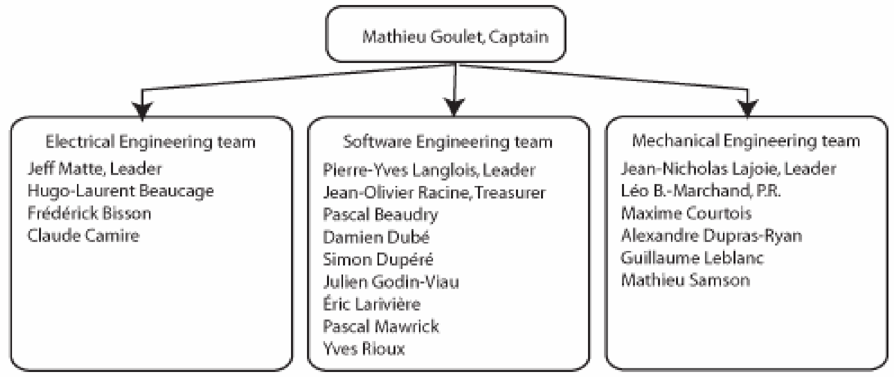
Table of content

1. Overview

Formerly composed of master students working on a walking robot, Capra was founded with the ambitious goal to develop unmanned vehicles. Few years later, the team is now composed of undergraduate students from the *École de technologie supérieure* (ÉTS), an engineering university that strongly supports student participation to many competitions.

1.1 Team Organization

The scientific club is composed of 20 students supervised by their faculty advisor, François Coallier, and the scientific clubs manager, Réjean Tétreault. The members are divided into 3 development teams. Their participation into the project is only in a voluntary basis, since it is not credited by the university. However, the ÉTS supports the projects financially and logistically as its main public relation resources. See Appendix A for the complete list of members. 4000 hours are estimated for this year on RS3.



2. Innovations

Since the end of 2005, Capra developed a brand new vehicle from scratch using its experience and lessons learned from Mentis 2.

2.1 Mechanically Improved platform

The aluminum based frame lowers the weight while it improves the durability of the vehicle. RS3's size is also greatly reduced from its predecessor to enhance the navigation and autonomy. Laser cutting and CNC milling were used for machining several of RS3's components, which allow better precision and increase the interchangeability of different parts. More details are found in the mechanical design part of this report.

2.2 Modularity and Standardization

To improve its usability, the modular design allows RS3 to carry extra components, adjusting its goals to the required task or presentation. The vehicle can carry extra modules such as a lawnmower or extra battery packs on its back. Masts, sensors, computers, other electronic components and batteries can easily be changed or removed. The team used standard brackets to fix those components and emphases onto accessibility of those components. Specific details of standardization are explained in the design sections of this report.

2.3 Improved Processing Power and Software Design

The embedded system of RS3 is by far superior to the one used previously in Mentis 2 with a Pentium M processing, 128 bit controllers, and a strong commercial off-the-shelf drive. The navigation software is enhanced with behavior-based algorithms, grid-based maps and cost maps to improve the decision process for path planning.

3. Design Process

The design process used for RS3 is based on the Attribute Driven Design developed by the Carnegie-Mellon Software Engineering Institute.

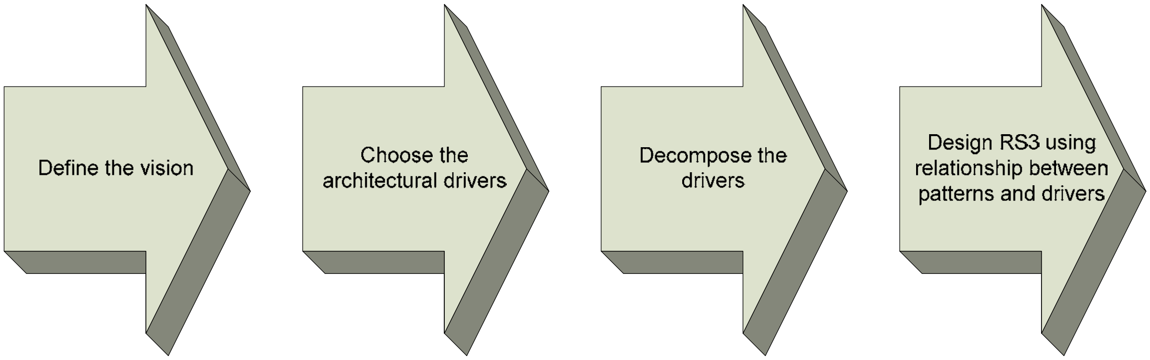


Figure 1 - RS3 Design Process

The vision defined for RS3 covers the different contexts in which the vehicle is used. First of all, Capra participates to the IGVC, and competing at the various challenges is the first priority. Capra also represents the ÉTS at various educational presentations. This year only, Mentis 2 went to secondary schools, scientific events, military and business presentations. Finally, the main goal of this club is research in autonomous robotic.

With RS3's scope defined, the architectural drivers were also based on previous experience with Mentis 2. The complexity and design flaws made this vehicle very hard to control with a high maintenance time. Finally and nonetheless, the students are working on a voluntary basis; they are often inexperienced and need to improve the vehicle attractiveness to help recruitment. Based on those facts, the architectural drivers became security, reliability and durability.

3.1 Security

Security is a major architectural driver as it is required for our competition eligibility and because we are doing many public shows. For our robot, security implies problem detection and reaction systems. Our safety system is composed of one accessible E-Stop button on top and a remote E-Stop which has 250 feet range. The robot also detects electrical breakdowns to prevent unwanted behaviors. For instance, the Roboteq's drive shutdowns as soon as it detects a problem. Activation of the E-Stop or breakdown detections cuts the engine's power and stops the robot within 6 feet on a grass land.

3.2 Reliability

Reliability of a system improves its predictability and lowers its maintenance. The overall design is based on simplicity and modularity. The simplicity reduces risk of unwanted behaviors or problems as we knew with Mentis 2. The modularity improves cohesion by keeping single goals to components. Unnecessary components are simply removed.

3.3 Durability

Durability lowers the maintenance due to stronger components. The hard drive is more shock resistant than the previous version on Mentis 2. The transmission shaft has been meticulously calculated using the Von Mises-Hencky technique. Supporting parts are 5/8in to support a total weight of 480 lbs, 240 more than the RS3 estimated weight.

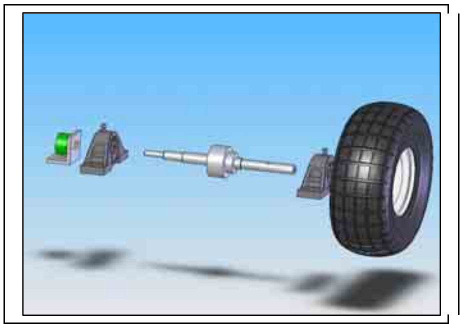


Figure 2 – Driving shaft assembly

4. Mechanical Design

Capra developed a completely new vehicle for 2006. To reach the goals selected with the Attribute Driven Design, the Mechanical Engineering Team spent over 6 months to design the frame with the contribution of the electrical and the Software Engineering Team to document specifications and to review the design at each milestone. With a strong design, the team was able to start the construction of RS3 in mid February.

Figure 3 – Robot design using Dassault System's Solidworks

Figure 4 – CNC machining

Figure 5 – TIG Welding on aluminum

The main structure of the robot witch is built in aluminum holds the engines and the driving components. Other structural parts are bolted on the main frame to rapidly put together the robot and simplify its maintenance.

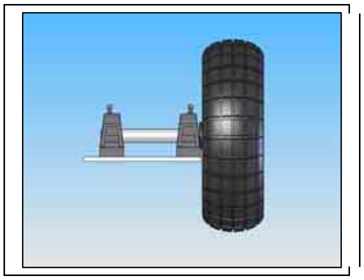


Figure 6 – Drive Shaft and Wheel

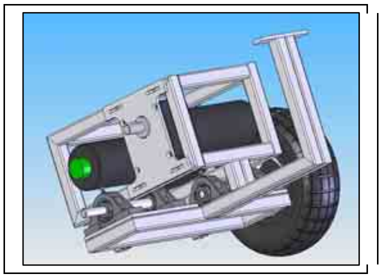


Figure 7 – Motors and Wheel Assembly

Figure 8 – RS3 aluminum structure

We estimate the weight of the robot at 170lbs. The dimensions are 36in in length, 24in of width and 61in of height.

During the design process, Capra used the product line engineering technique to maximize the modularity of the robot. This allows major configuration changes on RS3 depending of the required task. A lawnmower has been designed to be attached at the back of the robot to apply the unmanned vehicle mentality in a day to day context and to participate in the ION competition.

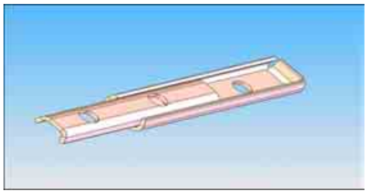


Figure 9 – Taper sliding bracket

Power calculations were made to establish the power required to propels the robot at a convenient speed and acceleration with its maximum load capacity. Some other calculations were also made to optimize the size of the power transmission shafts. According to those results, RS3 has two 1/3HP (250W) electric motors with a winding of 24V. A gear ratio of 8 at the end of the motors offers a maximum torque of 176lbgpo (19.9Nm) at the wheels.

5. Electronic Design

Since Capra focuses on reliability and durability as its architectural drivers, the Electrical Engineering Team designed a simple, but effective system. The team had learned their lesson from Mentis 2, which was a complex spaghetti-like system which made it very hard to understand and maintain. However, they kept the idea of having 2 independent electrical systems to isolate the noisy engines from the fragile electronic components.

5.1 Power Management

Both electrical systems are powered with 12V batteries at 20A per hour, sealed lead-acid serially connected to provide 24V. Using the same set of batteries simplifies their management, as they are balanced between efficiency, durability, weight and price. In addition, a switch disconnects the system when the battery compartment door is opened to prevent sparks while changing batteries.

The computers and sensors system can be powered by the conventional electricity network for development and testing purpose. The team added an inverted to convert DC 110V AC electricity to provide power to the electronic parts. A coil is also added for better quality of the AC output, which had important noise caused by harmonics. The AC current goes to the PC power supply (95% efficiency) for an output of +3.3V, +5V, -5V, +12V and -12V. A PCB when distributes the power trough a 48 ports terminal block through vibration proof clamps instead of screws used with Mentis 2.

Table 1 - Power Consumption of RS3

Component	Voltage (V)	Current (A)	Power (W)
Computer	ATX	3.5 (on 24V battery)	84
Controller cards	5	.3 each	1.5 each
Range Finder	24	.9	21.6
GPS	5	.065	.325
Camera	5	.25	1.25
inverter	24	8	192

5.2 Wiring

Wiring has been carefully designed to reduce its amount and to use standardized coloring to improve readability of the architecture. AWG 10 gage is used for the main power line (max 25A), AWG 14 gage is used between the drive and e-stop relays (max 14A), and AWG 12 gage is used to connect the drives to the motors (20A max). This design improves efficiency and durability of the electronic components.

5.3 Engines Electrical System

RS3 is propelled by 2 motors selected by both electrical and mechanical engineering teams, since the autonomy of the vehicle depends on its power consumption. Safety considerations were also considered with the use of a relay for each E-Stop to cut the power source from the engines. The model #6884 12V motors bought from Bodine Electric are protected by 24V fuses. The selected model had a flaw despite the documentation: they are not designed with sockets for encoders. The team decided to create a socket on the shaft itself for the encoders.

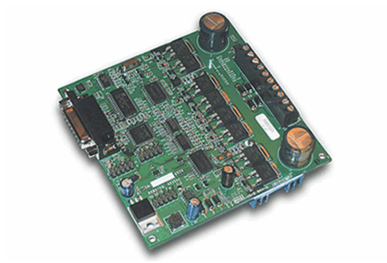


Figure 10 - AX1500 Drive used for RS3. (Source RoboteQ)

Our previous robot also had custom drives to power the engines. Since Capra does not have enough knowledge to maintain this component and since the club had many, many problems with it, the electrical engineering team chose the AX1500 model from RoboteQ for its extensive documentation and good reliability. The command interface is a RS-232, using a proprietary but simple communication protocol. It can handle an input voltage between 12 and 40V and can deliver 30A peak or a continuous 20A per channel.

5.4 Computers and Sensors Electrical System

The second electrical system in RS3 feeds the electronic components required for the unmanned navigation system. This system includes sensors, signal treatment components, and the embedded computer.

RS3 uses a FireFly 2 camera from Point Grey Research for its vision. This model is powered through firewire and is directly connected to the embedded computer. It offers auto-brightness and auto-exposure to enhance vision with minimum calibration.

The unmanned vehicle also uses a Sick LMS 291 range finder. This sensor is connected to the AC circuit through a dedicated converter to avoid power source dependency. It provides a good precision of half a degree on a 180 degrees range for a 30m distance. It also provides fog correction.

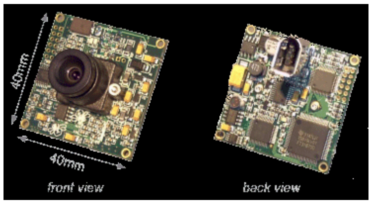


Figure 11 - Firefly 2 Camera (source: Point Grey Research)

Figure 12 - Range finder (Source: Sick USA)

Figure 13 - GPS (Source Garmin)

The GPS is made by Garmin. The reason why we chose this one is because it has accuracy less than 3m and because it gives 5 positions per second. The compass is a simple 2 coils module that we installed on an Atmel controller card.

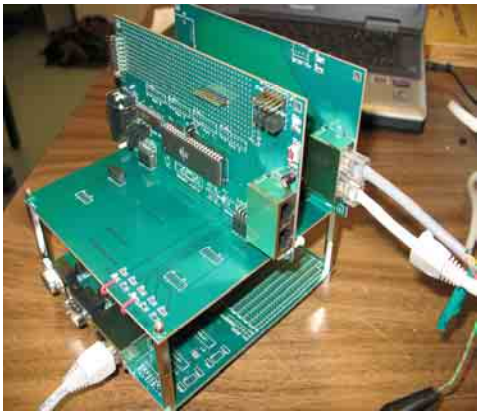


Figure 14 - The backplanes with cards.

The Electric Engineering Team repositioned the embedded computer's power supply outside of its case to install microcontroller cards connected on a backplane for the case protection. This backplane provides power and CANBus connectivity to reduce the wiring. Most cards are

Atmel Atmega32 microcontrollers on standard PCBs. Those components link peripherals such as the GPS, the encoders, the compass, and the range finder to the embedded computer. In addition, a low-level controller links the computer and the remote control to the drive to switch between manual and autonomous mode. This card uses an Atmega128 microcontroller with an additional 32KB RAM memory for better expandability. Every card has expansion holes and a serial port for development and debugging purpose. Their features are activated using jumpers.

The embedded computer used for the autonomous navigation has a Centrino 2.2Ghz processor, a Seagate 2.2” 7200RPM laptop hard drive in a Aspire X-QPack case with a 450W power supply. This model was selected for its efficiency ratio and shock resistance to improve durability. This system is much more powerful than the previous system. A D-Link wireless PCI card antenna has been installed to allow a direct communication with the computer and for JAUS.

The remote used is from Futaba. It requires 8 AA NiMH rechargeable batteries.

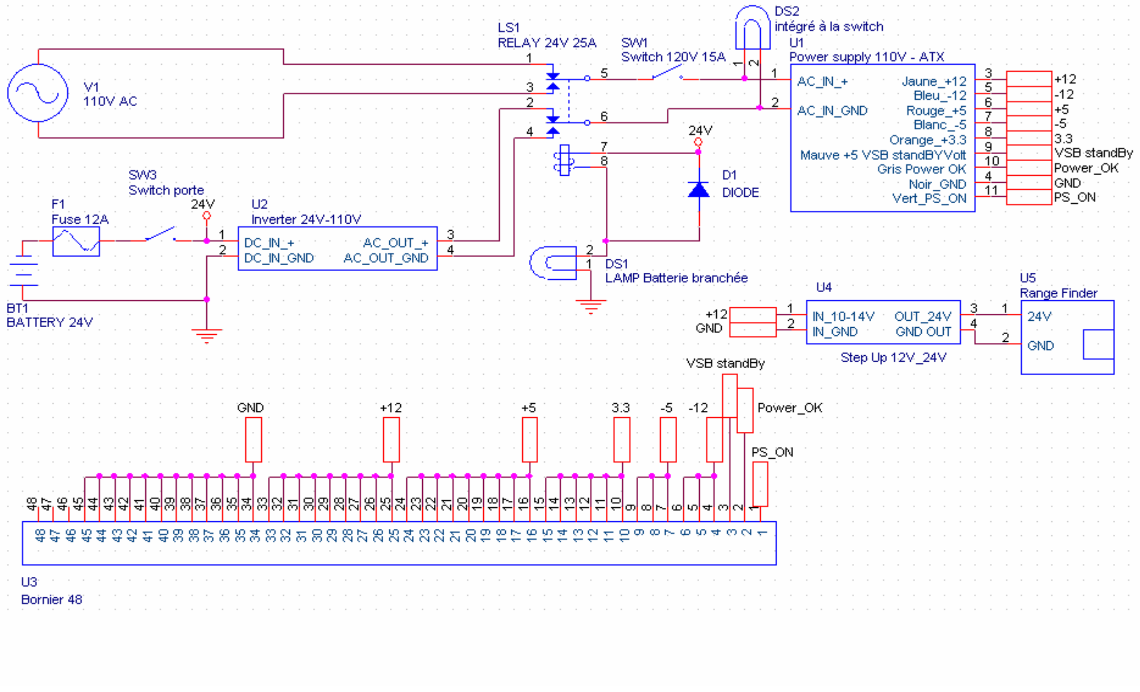


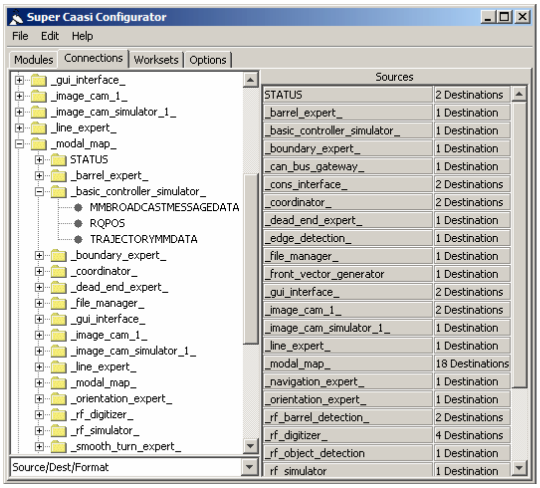
Figure 15 - Computer and sensors electrical scheme

6. Software Strategy

Capra is a software-oriented scientific club, aiming for scientific research on unmanned navigation algorithms and artificial intelligence. A major goal for the Software Engineering Team is reusable long term development. The students also chose C++ over Java for its performance and for the integration of commercial off the shelf tools.

6.1 Middleware

To help reusability through modularity and configurability, the Software Engineering Team developed a middleware, ARCHIE, to support continuous improvement of features. Based on a mixture of a subscription and a pipe and filter pattern, it allows encapsulation of each component into a separate module that can be enabled or disabled. Communication between the different modules is achieved through logical connections that regulate messages types and routes. Instantiation, configuration and connections of the different modules are managed through an XML configuration file. An interface was developed to manage this configuration (see figure). This architecture allows transparent replacement of modules: enabling a camera simulation module instead of the camera module will not affect the system in any way. This decoupling of functionality allows long term improvability and maintenance.



- _gui_interface_
- _image_cam_1_
- _image_cam_simulator_1_
- _line_expert_
- _modal_map_
  - STATUS
  - _barrel_expert_
  - _basic_controller_simulator_
    - MMBROADCASTMESSAGEDATA
    - RQPOS
    - TRAJECTORYMMDATA
  - _boundary_expert_
  - _coordinator_
  - _dead_end_expert_
  - _file_manager_
  - _gui_interface_
  - _image_cam_1_
  - _image_cam_simulator_1_
  - _line_expert_
  - _modal_map_
  - _orientation_expert_
  - _rf_digitizer_
  - _rf_simulator_
  - _smooth_turn_expert_

Source/Dest/Format

Sources	
STATUS	2 Destinations
_barrel_expert_	1 Destination
_basic_controller_simulator_	1 Destination
_boundary_expert_	1 Destination
_can_bus_gateway_	1 Destination
_cons_interface_	2 Destinations
coordinator	2 Destinations
_dead_end_expert_	1 Destination
_edge_detection_	1 Destination
_file_manager_	1 Destination
_front_vector_generator	1 Destination
_gui_interface_	2 Destinations
_image_cam_1_	2 Destinations
_image_cam_simulator_1_	1 Destination
_line_expert_	1 Destination
_modal_map_	18 Destinations
_navigation_expert_	1 Destination
_orientation_expert_	1 Destination
_rf_barrel_detection_	2 Destinations
_rf_digitizer_	4 Destinations
_rf_object_detection	1 Destination
rf simulator	1 Destination

Figure 16 – XML configuration utility

6.2 Navigation Software

The navigation software controlling RS3 developed by Capra is based on the Atlantis architecture from Erann Gat. This hybrid solution incorporates both fast reactive and slower deliberative modules within three layers. The following sections describe the layers roles.

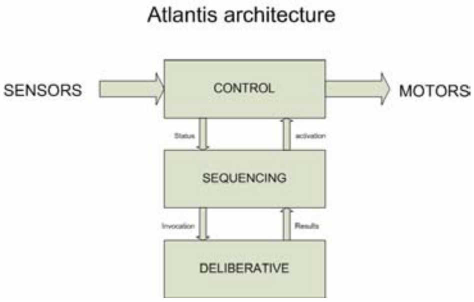


Figure 17 – Atlantis architecture overview

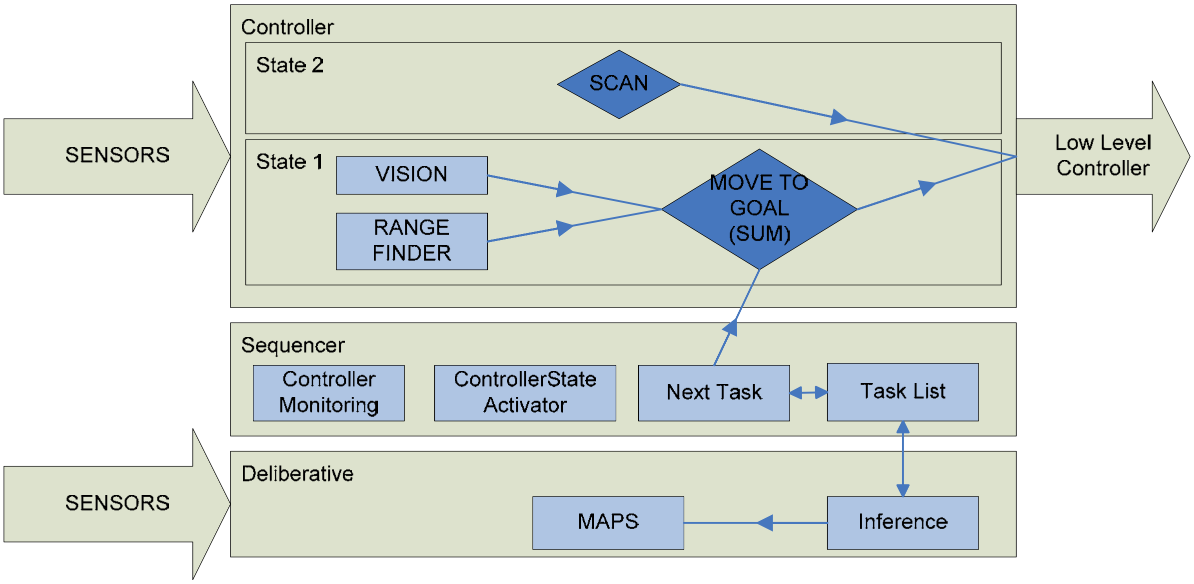


Figure 19 – Atlantis architecture implementation

6.3 Control Layer

The Control Layer's role is to receive sensors information and to generate navigation command. It follows vector field reactive control theory principles: the camera and the range finder data are processed into a repulsion vector field (see figure) for obstacle avoidance. Added to this field, a complementary vector map is added by the deliberative layer as a goal to reach by RS3.

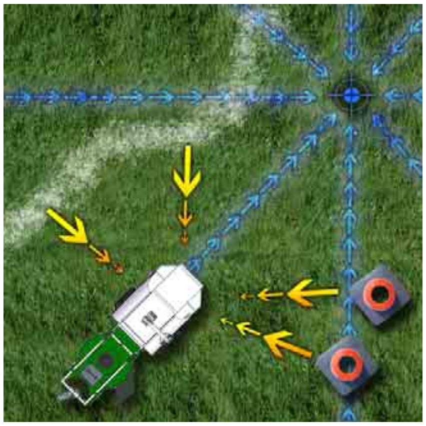


Figure 18 – Vector generation and summation

In order to generate the vector field to avoid obstacles and follow paths, the data collected from sensors must be processed first to extract relevant information to be then converted into valid data types.

For instance, images recorded by the camera are applied using a threshold to isolate whitish pixels associated with lines. Lines are subsequently extracted using Matrox Imaging Library (see figure below) in order to generate a vector field. In addition, the range finder reports physical obstacles in front of RS3, which also generates a vector field. Both fields are then summed to an objective vector map, which generates a velocity and orientation vector at RS3's position to be forwarded to the low-level controller.

Figure 20 – Image processing steps involved in line detection

Because image processing requires a lot of calculations, the response time of the control layer is 160 ms.

6.4 Sequencing Layer

The sequencing layer is a state machine interfacing the control and deliberative layers. It manages objectives needed to be accomplished, monitors control components and requests new objectives from the deliberative layer. The sequencer is also responsible for monitoring sensor availability and generating alternate course of action upon failures.

6.5 Deliberative Layer

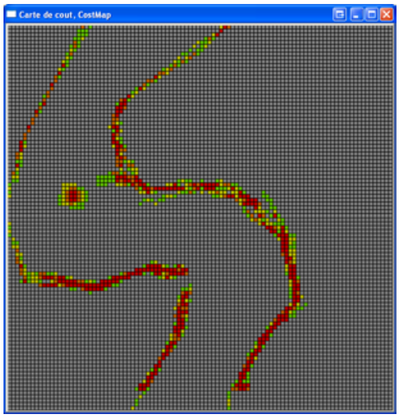


Figure 21 – Local grid based map

The deliberative layer uses real time data in conjunction with historical data to generate high-level objectives. It uses three data stores to do so: a local grid based map, a global grid based map and an object oriented map. Grid based maps are sets of counters incremented every time an obstacle is detected on a square (see figure to the right). Data from the sensors is directly forwarded to produce these maps. From the local map, which has a resolution of 20cm x 20cm, heuristics are used to generate the global map, which has a resolution of 1m x 1m. From this global map, strategic information is extracted to populate the object oriented map. Includes observed/not observed and mowed/not mowed regions are identified, and mission and intermediate objective points are managed. It is through cost analysis (safe zones identification) that objective prioritization is achieved. Once an objective is added and prioritized, it is available for selection and execution by the sequencing layer.

6.6 Positioning

Positioning of the robot is achieved through sensor fusion. Information from four sensors is combined to evaluate the position of the robot. The main positioning component is the dead reckoning which uses optical encoders attached to the propulsion wheels to calculate the orientation and position of the robot. Orientation (in radians) is calculated using equation 1 (with positive orientation represented as clockwise). Position is calculated using a temporal summation (numerical estimation) of linear displacement, based on the orientation of the robot at the time of the summation (equation 2). Both parameters are calculated at every sampling cycle of the low level controller, which is 40ms.

Equation 1

$$\theta = \frac{(\alpha_L - \alpha_R) \times R}{L}$$

Equation 2

$$\Delta x = \sin(\theta) \times R \times \frac{(\Delta\alpha_L + \Delta\alpha_R)}{2}$$

$$\Delta y = \cos(\theta) \times R \times \frac{(\Delta\alpha_L + \Delta\alpha_R)}{2}$$

Dead reckoning tends to accumulate an error over time due to wheel sliding. Long term corrections must be applied to improve precision. Orientation and position errors are calculated and threshold are used (empirical measurements) to insure values are within acceptable ranges. Correction factors are estimated and applied to the dead reckoning values to improve precision. In addition to these methods, landmark based positioning is also used. Once a known obstacle is identified using the range finder (in this case, the known barrel), its relative position is calculated and used to adjust the correction factors.

6.7 Simulator

To help the navigation software development, a simulator has been developed to reproduce sensors and motors behaviors. Both the camera and the range finder have specialized imitations to reproduce the range, imprecision, frequent errors and noise generally associated with these sensors. Simulation data is generated from a second instance of the object oriented map that stores fake obstacles, from which lines and barrels can be detected. The low level controller and motor response are also simulated using a dedicated module to represent RS3's realistic progression in the map.

7. Systems Integration

Systems integration is a key design element in Capra. All systems are designed by a multidisciplinary team of undergraduates in order to select an approach that fits in the overall design of RS3.

7.1 Integrated Development

Since an unmanned vehicle requires work from various engineering disciplines, Capra holds integration meetings to help developers consider requirements from each relevant fields and contexts. This heightens the coordination process in combining the different point of views of each development team. For example, the Mechanical Engineering Team first designed a swivel wheel with a good suspension for the stability of the vehicle. However, the Software Engineering Team objected to this idea because it would have forced the range finder to be higher, which would have diminished the detection of barrels. Another example is when the Software Engineering Team wanted to add a second computer dedicated to the robotic vision. The Electrical Engineering had to object because it would have compromised the autonomy of the vehicle. In both examples, integrated team meetings helped Capra to consider various constraints at design time, preventing expensive changes later during development.

7.2 Systems Communication

Capra decided to choose a single communication protocol for sensor communications. Both Electric and Software Engineering Teams have chosen the CAN communication protocol for its robustness and reliability. This noise resistant protocol is conceived for and by the automotive industry. The embedded computer communicates through an IXXAT USB-to-CAN interface while other components use CAN MCP2515 microchips through their SPI port. The backplane uses a RJ-45 connection to communicate with other devices, also using CAN protocol.

The only other communication protocol used for RS3 is to connect the embedded computer to the network. The IEEE 802.11g protocol is required for JAUS integration to RS3, as well as to deploy new software versions developed by the Software Engineering Team during development time.

7.3 Components Integration

The Mechanical Engineering Team had the challenge to optimize the design of RS3 between room for new components integration and miniaturization. A problem Capra had with Mentis 2 was the decentralization of electronic parts within the robot, which required a lot of wires to link and powers everything. With RS3, those parts are more centralized; they are directly connected on the backplane. This centralization greatly improves and speeds up integration of every component, which also improves reliability and durability. In addition, this backplane also contains extra slots for further development.

8. JAUS Integration

The JAUS section of the IGVC competition consists of controlling an unmanned vehicle from an external source. Although it has been added this year, the Software Engineering Team designed an architecture that facilitates the addition of new modules.

8.1 Learning Process

To understand the basics of the JAUS architecture, the team used the official documentation for the understanding and reviews for a good overview of different opinions. The JAUS "Strategic Plan" gives a basic history and long term goals of this project, the "Domain Model" gives a good idea of where the vehicle is located in the overall architecture, and nonetheless the "Reference Architecture Specification" gives the information to implement the specific messages to achieve the goals for the IGVC JAUS competition.

8.2 Message Integration

The implementation of the JAUS protocol in the software architecture requires several components:

- Socket based UDP-IP client-server component for 802.11g support
- JAUS message parsing component
- JAUS command execution component.
- An external light

The socket client-server is built using the "winsock" library from Microsoft Windows. There is no specific implementation that had to be done to support wireless of DSSS protocol, since they are all supported by the IP protocol.

The JAUS message parsing component has the responsibility to validate the UDP messages and to dispatch them to command execution components. The execution component module has the responsibility to interact with the internal architecture to alert each impacted internal modules. The impacted internal modules are the start/stop command of the AI and the light operation module.

8.3 Challenges to implementation

The major challenges encountered are the changes to the decision process so that the robot would be controlled by external signal rather that by the internal AI. However, with the modular architecture, this can be done using the module activation interface described in the middleware section. Despite this fact, the implementation is a pretty forward task. The modularity of the architecture gives the team the required liberty to add such functionalities as those asked by this competition.

9. Vehicle Performances and Cost

With the limited human and monetary resources, the need of well-scooped requirements and prioritization is important for Capra.

9.1 Performance Analysis

Table 2 - Predicted versus Measured Performance

Performance	Predicted
Speed	8 km/h
Ramp Climbing	12°
Battery Life	1.5 h

The following table sums the performance analysis for RS3. Using the mechanical calculations to choose power train components, we predict that the robot can climb an incline plan of 12° managing an acceleration of 1m/s^2 and can reach the maximum speed of 8km/h. With a three inches ground clearance, the robot could probably climb an incline plan up to 15° without any problems. Tests still have to be made in order to determine the official capacities of the robot.

The electronic component required 7 A of continuous current on the 24V battery to work properly. At this rate, the battery can give enough energy for 1.5 hour.

9.2 Vehicle Cost

The following table represents the overall cost for RS3. The students are trying hard to conclude sponsorship deals with industrial partners. In another way, those deals represent a good return on investment with the ability to help students into studying unmanned robotics. For example, Matrox graciously gave development license to their MIL library to help the detection of lines and fences. As an indirect return, many software engineering students from Capra were hired for an internship.

Table 3 - Vehicle cost

Item	Price
Lawnmower	300$
Wheels	290$
Structure	890$
Engines	900$
Encoders	340$
Camera	320$
GPS	200$
Compass	50$
Range Finder	8000$
Electronic Components	500$
Batteries	660$
Timing pulley and bearings	400$
Embedded Computer	1450$
Total	14300$

10. Conclusion

Capra's new robot, RS3, is the result of a multidisciplinary effort. Mechanical, electrical and software design tools and processes have been used to insure high quality design and good system integration. Although this team is a club of undergraduate, volunteer students, passion for robotics and engineering allowed delivering a robot that can reach goals, bounce on lines, and avoid obstacles. Technologies used in space (Atlantis architecture), automotive (CAN Bus) and military (JAUS) have been combined to produced a state-of-the-art autonomous vehicle. RS3 will safely, precisely and rapidly reach goals and follow the path to glory.

Appendix A – Team members

Name	Team	Course	Years
Léo Bardou-Marchand	Sponsors, Mechanical	MEC	2
Pascal Beaudry	Software	LOG	2
Alexandre Beaulieu	Software	LOG	3
Frédérick Bisson	Electric	ELE	2
Claude Camire	Electric	ELE	2
Maxime Courtois	Mechanical	MEC	3
Damien Dubé	Software	LOG	1
Simon Dupéré	Software	LOG	1
Alexandre Dupras-Ryan	Mechanical	MEC	3
Julien Godin-Viau	Software	LOG	2
Mathieu Goulet	Captain, Software	LOG	4
Mathieu Labrecque-Samson	Mechanical	MEC	2
Jean-Nicholas Lajoie	Chief Mechanical	GPA	4
Pierre-Yves Langlois	Software	LOG	4
Éric Larivière	Software	LOG	1
Guillaume Leblanc	Mechanical	GPA	4
Jean-François Matte	Chief Electric	ELE	2
Pascal Mawrick	Software	LOG	3
Jean-Olivier Racine	Treasurer, Software	GPA	4
Yves Rioux	Software	LOG	4

LOG: Software Engineering Degree

ELE: Electrical Engineering Degree

MEC: Mechanical Engineering Degree

GPA: Automate Production Engineering Degree

Note: Team members are undergraduates only.