# RoboCup Rescue 2022 Team Description Paper Capra

David Caron, Marc-Olivier Champagne, Asma Djoual, Léo-Daniel Gosselin, Benoit Malenfant, Gabriel Rondeau-Bouvrette, Ludovic Vanasse

**Info**

| | |
|---|---|
| Team Name: | Capra |
| Team Institution: | École de Technologie Supérieure |
| Team Country: | Canada |
| Team Leaders: | Asma Djoual & Léo-Daniel Gosselin |
| Team URL: | https://clubcapra.com |
| Team Qualification Video: | Youtube link |

RoboCup Rescue TDP collection:

2020
2019

***Abstract*—Markhor is Capra's second prototype. Its main new feature is the 4 flippers it has for vertical movement and all-terrain adaptability. Markhor's design is inspired from robots currently available on the market such as the Telerob telemax Pro and the Remotec Andros. This robot's main purpose is to go into Hazardous fields that might be too dangerous for emergency responders to go in, such as buildings that are filled with gases or are at risk of collapsing. RoboCup Rescue is a division of RoboCup competition that focuses on the use of robots in search and rescue applications, which is perfect to put Markhor's abilities to the test.**

***Index Terms*—RoboCup Rescue, Team Description Paper, Search and Rescue Robots, Robotics in Hazardous Fields.**

## I. INTRODUCTION

ROBOCUP RESCUE allows our team to compete against the best. It allows everyone to see each other's strengths and weaknesses in order to improve on our designs to create better prototypes. Last year, our team participated in the RoboCup Rescue German Open 2021 - DRZ Edition where we noticed that our robot's performances were underwhelming. The performance issues on the control portion of the robot are now fixed, along with the robotic arm being most likely ready for the competition making Markhor a fierce competitor.

### *A. Improvements over Previous Contributions*

The last showing that Capra had at the Robocup was at the 2019 edition in Australia. That year, the prototype that participated was our first ever robot built for the Robocup Rescue League. This year, we are bringing a whole new prototype. This new robot has a completely different locomotion system. With this configuration, every track and flipper is controlled with a closed loop mechanism to allow for finer movement capacities.

Robo Hero is with the Department of Robotics Gurus, Moon University, Moon, e-mail: .

J. Doe and J. Doe are with Anonymous University.


Fig. 1. Markhor at RoboCup Rescue German Open 2021 - DRZ

Also in this new prototype, we have implemented several new software features, such as a reverse control scheme for increased speed and maneuverability when it's required to go backwards. Furthermore, we are working on a basic operation system of the robotic arm mounted on the robot in time for the competition.

### *B. Scientific Publications*

At ÉTS (École de Technologie Supérieure), we have the opportunity of replacing an elective course with a project-course related to a student club participating in an engineering competition, such as the Robocup Rescue. Furthermore, this concept also applies to the final engineering project each student must do before completing their degree. This results in publishing independently a fair number of research reports. Although they are not equal in every aspect to a typical research paper, they represent work and research that went into improving the robot for the Robocup Rescue competition.

Since the 2020 competition was canceled, and we didn't participate in the 2021 competition, we decided to include all the papers we published between 2020 and 2021. Also, it is worth mentioning that all the papers are published in French since it is required by our university.

To facilitate the software development of our prototype, we explore the possibility of using Unity as our simulation engine for our robot[1].

Another paper describes the implementation of using a custom inverse kinematic controller for our robot arm[2]. It is based on the jog-control package.

Again, for the robotic arm, we wanted to capitalize as much as possible with the current hardware that we have. Right now, the 6 actuators for the arm's joints are linked together by a 24 pin flat cable in a daisy-chain configuration. He is routed through the arms limbs. Position commands and power are sent through that cable. We try to use two free pins of this flat cable to also read data from sensors placed in the wrist of the arm with the onboard computer[3].

And, in another paper, we tried to implement an augmented control system to reduce the cognitive load of the operator by controlling the front flippers based on perceived obstacles[4].

## II. System Description

### A. Hardware

*1) General:* Markhor's chassis is custom made from bent sheet metal by one of our sponsors. Many of the other mechanical parts have been purchased off-the-shelf and adapted. The platform's wheelbase is 62cm and its axle track is 44.5cm. The overall outside dimensions of the platform, excluding the arm, are 145cm long by 60cm wide by 25cm high. The dimensions of the arm, folded at rest, are: 74cm long by 8cm wide by 30cm high. The outstretched arm measures 145cm. The total weight of the robot, including the batteries and the arm, is 77.5kg.

*2) Locomotion:* Markhor's locomotion system consists of four independent flippers, with fairly aggressive rubber tracks to provide traction on uneven terrain. They can be raised or lowered to modify the robot's footprint and navigate the obstacles. Each flipper is powered by two DC motors: one for the propulsion and one for the up and down movement.

*3) Power:* One of our requirement for the power source was that we should be able to transport the robot by plane. With that in mind, we opted to use Milwaukee M18 power tools battery packs. While this choice limits us in terms of power efficiency, the packs are known to be safe and reliable. As we are also using them in our tools, it is an extremely convenient solution when deploying Markhor in the field.

*4) Electronics:* The first electronic system is the emergency stop system which provides a safe way to cut power to the motor drivers. All power electronics can be disabled by either the mushroom type E-stop button on top of the robot or remotely via the user interface. For motor control, we opted for the Talon SRX drives. They can be easily interfaced with in C++ with their library, are ultra robust and handle closed loop control on the drive itself, providing a higher refresh rate and lower latency for the control system.

There is two voltage rails in the robot which are 12 and 24 volts DC provided by three DC to DC converters which adapt the voltage level of the batteries.

The main computer for Markhor is the Jetson Xavier AGX which will give us more than enough computational power for our visual processing and deep learning needs in the future. To interface various hardware, we decided to create a new board we call the GPIO board, leveraging the atmega328p microcontroler and the Arduino software. This board will allow us to efficiently integrate new sensors and or actuators in markhor.

*5) Manipulation:* For the manipulation, we are using a custom robotic arm with Kinova's actuators. Those actuators are powered via a 24V DC-DC Boost converter in the robot. Their communication line is daisy chained inside the robotic arm itself from actuator to actuator making it easy to move every actuator individually without having to manage the wires. In order to grab onto objects, we are using a Robotiq gripper as an end effector.

*6) Sensors:* Here is a list of the sensors that are used in the robot:

- **Thermal camera:** used in order to find the heatmap of the victim board. Powered through a USB-hub plugged into a Jetson.
- **ambient $CO_2$ sensor:** used in order to find how much CO2 particles per million are around Markhor. Powered through a 5 volt DC-DC regulator plugged into the Jetson's I2C communication BUS.
- **3D Cameras:** Used in order to detect QR codes, detect motion and in order for the user to see what's around the robot. The cameras and powered and communicate via the same USB hub
- **Temperature sensors:** Used in order to detect the motor's temperature. Those sensors are eight thermistors plugged into an analog pin.

*7) Communication:* For communication between the robot and the operator station, we are using a small form factor Wi-Fi router embedded in Markhor as an access point. For communication inside the robot, we opted for a combination of Ethernet, USB, CAN, serial and I2C.

### B. Software

*1) Tracks control:* Our new prototype, Markhor, uses four tracks instead of the two that we used in our previous prototype, Takin. Also, one of the issue was the computation of the velocity command sent to the tracks. When sending certain types of inputs with a video game controller, the robot would have improper behaviors in its movement.

With Markhor, we decided to use a differential drive control mechanism to drive the tracks. This ROS controller allows for correct movement of the robot with a simple velocity command. This greatly simplifies the control of the robot.

Conjointly, in our previous prototype, there was no PID controller to correct the velocity of the tracks. In Markhor, each track is configured in a closed-loop system to guarantee an efficient velocity control.

*2) Flipper control:* In Takin, there wasn't any locomotion mechanism to climb stairs or obstacles. In Markhor, we tried our first attempt at a flipper system. The software aspect of the flippers is similar to the tracks control system. For each flipper, we have a closed loop PID controller to correct the position of the flipper.

To boot, since there are 4 flippers, we set up the option on the game controller to manage one pair of flippers at the time. The idea was not to overwhelm the pilot with too many controls at the same time on a simple gamepad.

*3) Video System:* For our video system, our robot is equipped with four 3D cameras. Each camera is a nodelet of the astra_camera package. This package gives us finer control over the cameras and allows for specific processing on the image.

The third-person camera will be free to move on 2 axes to allow for almost 360°vision without depending on the camera on the robot's arm.

The view of our cameras can be rearranged by the operator and is reorganized automatically when going backwards by the simple press of a button.

*4) Audio System:* The robot is equipped with a conference speakerphone to allow for bidirectional audio communication with the operator and potential victim. This communication will soon be on a push-to-talk activation scheme, which is more practical than a constant one distracting the operator. This speaker is connected through USB to the onboard computer. We decided to use a customized version of the ros_audio_common package. There's two ROS nodes running on the robot, one node for capturing and sending the audio data over the operator station and one for playing the audio data received from the operator station.

*5) Sensors:* Grouped under the "victim board" tab on the operation station, we have a view of all our sensors. Each one of these has its own ROS node that reads the associated hardware and publishes its readings on its own topic for our user interface to display. These include:

- A thermal camera communicating through the UART and SPI protocols
- An ambient $CO_2$ particles per million graph with which we communicate through I2C
- The arm camera with a motion detection OpenCV-based program activated over its feed
- The same arm camera but with a QR code detection and reading visp_auto_tracker based program screening it

*6) Arm control:* Our robotic arm system is the same as for our last prototype. But we didn't have a lot of control implemented in the last edition of the Robocup Rescue League. This year, we are working on a basic control system of the arm and hand.

The arm has Kinova actuators, which use a proprietary SDK to allow for the control of each actuator. We are working on a joint control scheme and afterward we will add a Cartesian control. The Kinova actuator uses a controller, which implements an inverse kinematic for any type of arm built with their actuator and controller.

### C. Communication

Right now, the communication setup between the operator station and the robot is done with the GL-AR750S-Ext router from GL-iNet. This is a small and powerful travelling router. It allows for controlling the robot over Wi-Fi and with a tether connection. We are also in the process of testing the networking setup with a more powerful and reliable router, the ESPRESSObin Ultra.

For the GL-AR750S-Ext router, the antenna for the wireless communication are set up on top of the robot. This allows for connecting to it if needed. The router supports the Wi-Fi protocols IEEE 802.11a/b/g/n/ac. It's maximum transmission power is 20dBm for the 5GHz spectrum. The router consumes less than 6W and its power setting will be the default one on the router. And its SSID will be RRL_CAPRA.

### D. Human-Robot Interface

Our robot is controlled using an Xbox Controller to control our tracks, flippers and camera movements and a 3D Space-Mouse to control the robot's arm. The inputs are read and sent to our robot by our user interface, which is built as an Electron app using TypeScript and React. Our UI allows the operator to see every camera feed. A small, but useful, addition that we are making currently is to be able to the position of every flipper, which can sometimes be hard to see through our camera feeds. The user interface is simple to use for any user, and it includes a controller scheme that shows what each input is for. It is composed of multiple tabs, permitting to change easily from the camera views to the victim board reading options. Additionally, the "config" tab allows the operator to reconfigure completely any view on the fly. This final tab also allows launching the entirety of our robot's functionalities or any combination of these without the need of a single command line.

## III. Application

### A. Set-up and Break-Down

Markhor's deployment is as easy as dropping it on the ground, plugging in fully charged batteries, booting operation station, releasing the E-Stop and connecting the user interface to its UI.

### B. Mission Strategy

This is Markhor's first time in the official Robocup Rescue league, but it previously participated in the RoboCup Rescue German Open 2021 - DRZ Edition. At the DRZ it was put to the test, it had multiple issues regarding how we managed the power put into the motors and how we managed the commands sent to them. This year, we plan on improving everything that is related to mobility challenges such as the traverse, maneuvering and mobility. On top of that, we have a multitude of sensors such as the $CO_2$ sensor that is now included in the UI capable of getting us all the points from the victim boards. However, Markhor's Ingress Protection rating isn't high enough to protect himself against sand. Therefore, we'd like to avoid anything that requires anything equal or above to the IP50 rating.

### C. Experiments

In order to test Markhor's ability to perform in this competition, we built a small-scaled test zone where the robot needs to get through typical RRL challenges such as debris, stairs, inclined floors, etc. This puts the robot's maneuvering to the test and lets our pilot practice for the real challenges coming ahead.

On top of that, we also have a victim board made in order to test Markhor's sensors that allows us to evaluate how well he can acquire multipliers in the challenges.

Fig. 2. Field used for testing Markhor's abilities

### *D. Application in the Field*

Since Markhor's design is made to be particularly rugged, and its body is made of thick steel, able to resist collisions, we believe he would shine in real world scenarios. It's four snowblower tracks allow him to drive easily over rough terrains while being able to climb substantial debris with its flippers. Markhor's user is also able to easily see anything that is moving around it with the help of all its cameras. On top of that, he is capable of coming up to anyone stuck in said terrain. However, he still has a couple flaws. Originally, there was the idea of using fans to keep its interiors component cooled and create a positive pressure chamber to keep dust and sand outside the case of Markhor. Since he has so many holes, the pressure is far from high enough to keep sand out. Therefore, that makes his electronic components vulnerable to sand.

## IV. Conclusion

Markhor's new design now allows us to do everything that is related to mobility. We are impatient to try it out against other teams in order to compare it against its competition at RoboCup 2022. Not only are the software improvements going to be a huge boost to help us get as many points as possible, but we're also eager to push the build of the robot to its limit in order to find out what we need to improve upon next.

TABLE I
Operator Station

| Attribute | Value |
|---|---|
| Name | Dell Latitude 5424 Rugged |
| System Weight | 2.5 kg |
| Dimensions | 347.0 x 244.5 x 44.4 mm |
| Unpack and assembly time | 1 min |
| Startup time (off to full operation) | 1 min |
| Power consumption | 90 W |
| Battery endurance | 8 h |
| Any other interesting attribute | Rugged & Outdoor-Readable Screen |
| Cost | 3059$ |

## Appendix A
## Team members and Their Contributions

Capra's team is mostly made of undergraduate students in the fields of engineering

- Léo-Daniel Gosselin — Team co-captain
- Asma Djoual — Team co-captain
- David Caron — Electrical team co-captain
- Benoit Malenfant — Electrical team co-captain
- Rondeau-Bouvrette, Gabriel — Mechanical team captain
- Marc-Olivier Champagne — Software team captain
- Ludovic Vanasse — Software architect
- Christian Belleau — EE & SW engineering advisors
- Simon Joncas — Mechanical engineering advisors

## Appendix B
## CAD Drawings

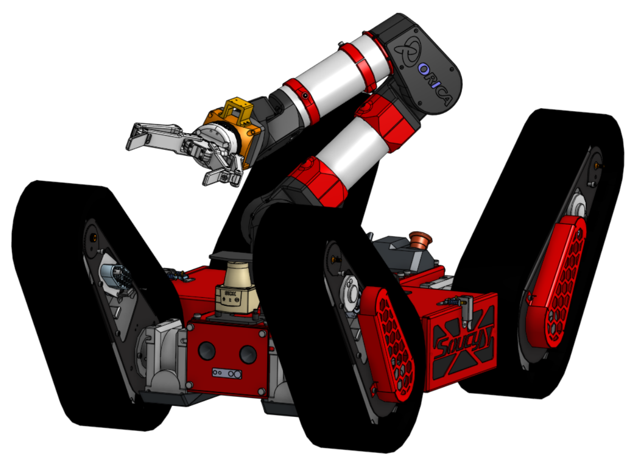

Fig. 3. Markhor Robot CAD Drawing

## Appendix C
## Lists

### *A. Hardware Components List*

The table II lists most of the hardware used in our robot.

### *B. Software List*

The table III lists most software used in our robot.

TABLE II
HARDWARE COMPONENTS LIST

| Part | Brand & Model | Unit Price | Num. |
|---|---|---|---|
| Tracks motors | VEX Robotics BAG Motor | 38,99$ | 4 |
| Flipper motors | VEX Robotics Mini-Cim Motor | 38,99$ | 4 |
| Encoder | VEX Robotics TalonSRX | 116.99$ | 8 |
| Gearbox | VersaPlanetary Gearbox | 170$ | 8 |
| Embedded computer | Nvidia Jetson Xavier AGX | 600$ | 1 |
| Wi-Fi Router | GL-AR750S-Ext | 129.99$ | 1 |
| Batteries | Milwaukee M18 Battery | 300$ | 12 |
| IMU | Vector VN-300 | 300$ | 1 |
| 3D Camera | Astra Embedded S | 170$ | 3 |
| Thermal Camera | Flir Lepton 3.5 | 300$ | 1 |
| LiDAR | SICK TiM5xx | 5000$ | 1 |
| $CO_2$ Sensor | KS0457 CS811 CO2 Sensor | 16$ | 1 |
| Bidirectional Audio system | Jabra Speak 410 | 180$ | 1 |
| 6-axis Robot Arm actuators | Kinova Actuator | 12000$ | 6 |
| Robotic hand | Robotiq 2F-140 | 5000$ | 1 |
| Rugged Operator Laptop | Latitude 5424 Rugged | 3059$ | 1 |
| 3D controller | SpaceMouse® Compact | 200$ | 1 |
| Telecommunication headphone | Hyper X Headphone | 100$ | 1 |

TABLE III
SOFTWARE LIST

| Name | Version | License | Usage |
|---|---|---|---|
| Ubuntu | 18.04 | Various | Operating system |
| ROS | Melodic | BSD | Comm framework |
| Phoenix Framework | 5.18 | Closed source | Motor control |
| Sick Tim | 0.0.17 | BSD | LiDAR |
| Kinova SDK | 1.5.1 | Closed source | Arm control |
| ros_control | Melodic | BSD | Tracks & Flipper control |
| OpenCV | 3.2.0 | Apache | Motion detection |
| ros_audio_common | Melodic | BSD | Audio communication |
| ros_astra_camera | Melodic | BSD | Camera ROS wrapper |
| React JS | 17.0.2 | MIT | Operator Station |
| Electron | 16.0.1 | MIT | Operator Station |

## ACKNOWLEDGMENT

The authors would like to thank all team members for their efforts toward this project. A special thanks goes to all our sponsors and École de Technologie Supérieure student club support staff.

## REFERENCES

[1] N. Beaudet, “Simulation de contrôle dans un environnement virtuelle du robot markhor pour le club Capra.” [Online]. Available: https://raw.githubusercontent.com/clubcapra/papers/master/simulation_de_contr%C3%B4le_dans_un_environnement_virtuelle_du_robot_markhor_pour_le_club_capra.pdf

[2] A. Francoeur, “Conception et implémentation du contrôle pour le manipulateur d’une plateforme robotique mobile.” [Online]. Available: https://raw.githubusercontent.com/clubcapra/papers/master/GPA791_Controle_Cartesien_Ovis.pdf

[3] Arnold Chan Kao Chung, Charles Masson, and Maxime Vigneault, “Conception d’un système d’acquisition de données de capteurs pour un bras robotique.” [Online]. Available: https://raw.githubusercontent.com/clubcapra/papers/master/ELE795_PFE_Rapport_Technique_CHAA_MASC_VIGM.pdf

[4] Charles Giguère, Marc-Olivier Bélisle, and Ludovic Vanasse, “Contrôle augmenté pour la mobilité du robot Markhor conçu par le club Capra.” [Online]. Available: https://raw.githubusercontent.com/clubcapra/papers/master/Rapport_LOG791_controle_flipper_UI.pdf