RoboCup Rescue 2020 Team Description Paper Club Capra

Alexandre Francoeur, Ludovic Vanasse, and Edouard Belval

Info

Team Name:	Capra
Team Institution:	École de technologie supérieure
Team Country:	Canada
Team Leader:	Alexandre Francoeur
Team URL:	http://capra.etsmtl.ca/

RoboCup Rescue TDP collection: 2019+: https://tdp.robocup.org/ Pre 2019: https://robocup-rescue.github.io/team_description_papers/

***Abstract*—RoboCup Rescue is a division of RoboCup competition that focuses on the use of robots in search and rescue applications. This robot is the second prototype of our club built for this competition. One of the main features in our design is the four flippers, no central track geometry, we inspired our selves from robots currently on the market like the Remotec Andros and the Telerob telemax Pro. On the software side, we're working on a Q-Learning algorithm to control the flippers semi-autonomously and on a control algorithm for our manipulator using a SpaceMouse as an input.**

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

I. Introduction

THE RoboCup Rescue competition allows our team to participate in an international challenge with different universities and organizations. The competition aims to encourage the teams to build the best robot for search and rescue operations and this is the main motive behind our team's effort. We were lucky enough to participate at the RoboCup 2019 in Australia and we are aiming to compete in 2020 at Bordeaux. We learned a lot from our experience and we made great improvements to our prototype especially in the mechanical and software department of the project.

A. Improvements over Previous Contributions

To design this year's system, we've started by analyzing the flaws of our previous robot. Then, we aimed to fix most of them. Our robot now has a longer footprint, a lower center of gravity and better mobility caused by the use of flippers. We've also decided to improve the usability by having a fully symmetric camera setup and to change our control algorithm to control the robot as easily going backward as when moving forward. We've also improved our UI based on the suggestions made by last year's operator.

A. Francoeur, L. Vanasse and E. Belval are with the scientific student club Capra, École de technologie supérieure, Montréal, e-mail: capra@clubcapra.com.

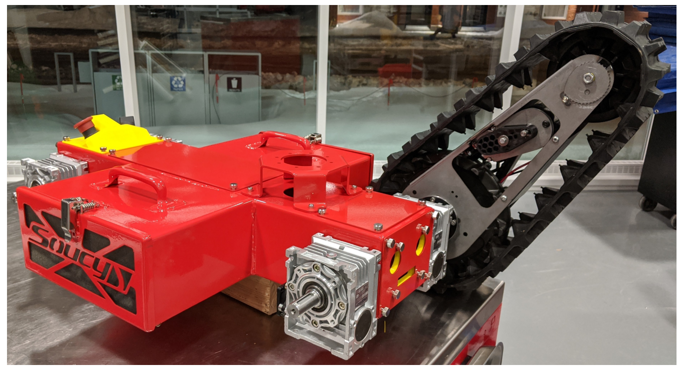

Fig. 1. The Markhor Robotic Platform currently on the assembly bench

II. System Description

A. Hardware

The robot's hardware is designed for long and complex tasks and will be described in this section

1) Locomotion: The robot has 4 motors for locomotion and 4 others to control the flipper angles. Each flipper angle can be controlled independently. All drives are connected to the on-board computer by the CAN Bus network and powered through our power distribution system. The electrical power is provided by 6 power tool batteries all connected in parallel to provide a maximum of current to the motors.

2) Manipulation: The robotic arm is the same as last year, it's mounted on top of the robotic platform. It has 6 degrees of freedom with a gripper, it can lift, poke and move objects.

Fig. 2. Ovis the Robotic Arm

3) Vision: The robot is equipped with a front camera. This camera is an RGB-D, it can record a visual image along with the depth field of the environment. There is another 3D camera in the rear which helps navigate backward and see a potential obstacle. On the robotic arm, we added a thermal camera to record the heat sources inside the lanes. There's also an RGB camera and a microphone mounted on the robotic arm to detect victims in hard to reach locations.

4) Sensors: To further help the robot be aware of its environment, there is a LiDAR on top of the frame to help localize and map the surroundings. The robot has an IMU which provides gravitational force and acceleration data. A speaker is installed to communicate instructions or to speak with a victim. There is also a CO2 sensor that can detect human presence in the surroundings.

5) Operator Station: The operator station consists of a computer that runs a web-based interface. The station is connected to the robot using Wi-Fi or a wired connection. An Xbox One controller is connected to the computer using USB to control the robot platform and a SpaceMouse is used to control the robotic arm. Finally, the audio is handled with a gaming headset.

B. Software

The software architecture is based on ROS from the UI to the motor control. We implemented many nodes in the system to help distribute the information across the architecture. We used the APIs and SDKs provided by the manufacturers to interact with the different sensors and actuators.

1) User Interface: For the UI, we started by porting our web application from VueJS to ReactJS. Over time, we realized that the application was better suited as a desktop application so we decided to use Electron. This allowed us to access a few more native APIs. For example, we now have an embedded terminal that automatically connects over SSH into the robot when we start the application. The audio is also handled by the web interface. The UI lets you see the camera stream in real-time, but it can also show you the robot URDF model in real-time. We made sure to remain as platform-agnostic as possible so that it could work with any robot, not just robots made by Capra. [1]

2) Arm Control: Our robotic arm is made of actuators from Kinova. They provide an SDK along with ROS nodes. This year, we're trying a new control algorithm based on MoveIt [1] and the package jog_control [2]. We're currently programming an algorithm to use a SpaceMouse as an input to control the arm in cartesian coordinates.

3) End effector: On the end effector, there's a multitude of electronic components. The gripper will be used to manipulate the objects of the dexterity tasks. The sensors will allow detection of thermal activity, CO2 levels along with a camera. All this data will be relayed to the embedded computer inside the robotic platform.

4) Mapping: Currently, the mapping relies on the navigation stack from ROS. We use Cartographer for the map generation but we might switch to RTAB-MAP [3] to improve the operator awareness of the robot environment. Additionally, if time allows, we also plan on making a mission controller to facilitate the generation and archiving of the different maps throughout the competition. The mapping system will also integrate a POI manager, that will listen to the detection nodes. The POI manager will record and archive the position on the map of detected elements such as hazmat labels, motion, and others.

[1] Capra's UI is available at github.com/clubcapra/capra_web_ui

5) Hazmat label detection: Hazmat detection is done by using the MobileNetV3 architecture with an R-ASPP segmentation head as its final layer [4]. This approach allows for a way lighter networks which uses about a tenth of the FLOPS used by our previous solution based on YOLOv3 [5]. In practice, the model should be small enough to run on a Pixel smartphone at 10 FPS. We trained the new model by using a previously existing dataset [6] to which we added our own samples taken from a replica of the victim board. By doing so, we are successfully able to identify 10 to 11 labels from the 13 possible ones, with a lot of headroom for further improvement by enriching the dataset with more images.

6) Hardware abstraction layer: Since we drastically changed the design of the robot, we had to review our approach towards the control of the motors. Thus to increase the flexibility of our setup for the future, we decided to adopt the approach of a hardware abstraction layer. To do so, we created a hardware interface for our motors. This configuration allows us to use a standard ROS controller to control the hardware interfaces. This is a significant improvement since we don't need to recreate already existing algorithms to control our hardware. As an example, we'll be using the differential drive controller to control the robot locomotion.

7) Two-way audio: Last time we participated at the RoboCup Rescue, we used a very unstable solution for the two-way audio with the robot. Today, we are using a modified version of the ROS audio common nodes for the two way audio. This is an improvement because it's integrated with the rest of the software stack and increases the reliability of our setup.

C. Communication

The robot will be emitting Wi-Fi. An external Ethernet port will be available should the wireless connection be too weak. The operating station consists of a single laptop computer with an integrated wireless antenna that will connect to the robot using the Wi-Fi or the wired network. The robot wireless antenna will be mounted on top of the frame. The model of your wireless router is a Microtik RouterBOARD RB911G-5HPacD. The router will use conventional 802.11ac Wi-Fi and we estimate that the range of the router will be around 15 meters before the connection start to deteriorate. The wireless network is set to only use 5GHz band like the rule book specifies and the SSID of our network will be RRL_Capra.

D. Human-Robot Interface

We reviewed our approach to the Human-Robot interface (HRI) compared to last year. We set our self's a simple goal to aim towards.

1) The HRI should be as user-friendly as possible, to measure our achievement, we settled that after 60 minutes, any adult should be able to operate approximately 80% of the robot features with ease. This metric is qualitative and will vary greatly from a subject to another, but based on our research even NASA-TLX is prone to human bias.

1) Description of the UI: The UI has tabs for different usage. The main one is for the teleoperation of the robot, there's one to visualize the data provided by the arm-mounted sensors and finally, there's a tab to set up the user preferences. The teleoperation tab has a camera view and a 3D representation of the robot in space. The arm tab has a camera view and a graph of the CO2 readings.

III. Application

A. Set-up and Break-Down

Robot deployment is as easy as dropping it on the ground, checking battery charge levels, booting the computer, releasing the E-Stop and connecting the user interface to the robot using our UI.

B. Mission Strategy

Since this is year will be our second participation at RoboCup Rescue. We'd like to perform better overall than last year. First, we'd like to test thoroughly our robot on the mobility lanes as this robot will allow it. Secondly, if all goes as planned, we will be able to control the robot in a semi-autonomous fashion, the flippers' angle of contact will be controlled without any user input. Finally, we'd like to score all the points related to the victim board and get some for dexterity. This is a new design and we're confident that we've addressed the biggest issue our previous robot had which was the lack of mobility. The only lane we're thinking we might avoid is the sand lane because we'd like to protect our hardware.

C. Experiments

To test the robot, we built a small scale test method. We'll be able to test the robot before getting to the competition. This will help us address issues with the control algorithm before the competition. Figure 4 of the appendix shows the test method.

D. Application in the Field

We believe that this prototype will perform better in a real-world scenario. The new design allows the robot to navigate heavily accidented terrains. The only issue that we'd need to address would be the ingress protection of the robot. At the moment, we estimate our robot to have an IP40B, which means that the internals of the robot are not quite protected against dust. Also, the software is coming along quite nicely, we are almost certain of being able to clear all the victim board tests. We are still working on the dexterity test, but we are building the software so it will be very easy for the operator to control the robotic arm. Additionally, since we are using a web-based interface for the control of the robot, it is possible to allow multiple people to visualize the robot data in real-time. This could multiply the number of information that the first responders could observe and yield greater operational success.

IV. Conclusion

The new design of the robot is very promising and we are eager to test it at RoboCup 2020. Also, we believe that the improvement made in the software will greatly help to operate the robot and hopefully will have an impact on within the RoboCup Rescue community. We believe that we will outperform the results we got last year. We are excited to test the robot and find out how it performs and what will be the next improvement or goals we set for ourselves.

Appendix A
Team Members and Their Contributions

A. Team members

Our team is composed of undergraduate students in multiple fields of engineering.

- Alexandre Francoeur — Systems Engineering
- Marc-Olivier Belisle — Software System
- Ludovic Vanasse — Software System
- Alexandre Mongrain — Mechanical Design
- James Mackay — Mechanical Design
- Frédérick Giard — Mechanical Design
- Edouard Belval — ML Trainer
- Charles Giguère — User Interface
- Maxime Vigneault — Electrical Design

B. Sponsors

- Soucy International inc.
- Robotiq
- Kinova Robotics
- Creaform Engineering
- Mekanisk 3D Printing Services

Appendix B
CAD Drawings

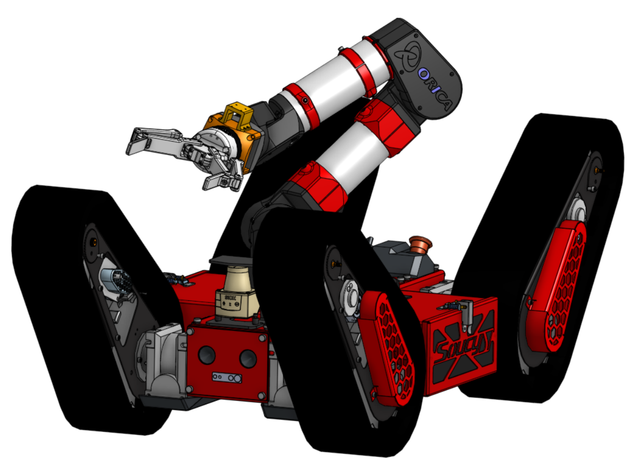

Fig. 3. Markhor Robot CAD Drawing

Fig. 4. Picture of our small scale test method – This test method is 8’x12’ and is used to test the following cases: stairs, parallel rails, step-fields, and inclined plane

TABLE I
HARDWARE COMPONENTS LIST

Part	Brand & Model	Unit Price	Num.
Motors	BAG Motor	38.99$	4
Motors	Mini-CIM Motor	38.99$	4
Drives	Talon SRX	90$	8
Gearbox	VersaPlanetary Gearbox	170$	8
Batteries	Milwaukee M18 Battery	300$	12
Embedded computer	Nvidia Jetson Xavier AGX	600$	1
Wi-Fi Router	Mikrotik RB911G-5HPacD	99$	1
IMU	Vector VN-300	300$	1
3D Camera	Astra Embedded S	170$	2
Infrared Camera	Flir Lepton 3.5	300$	1
LiDAR	SICK TiM5xx	5000$	1
CO_2 Sensor	Telaire T6713	170$	1
6-axis Robot Arm actuators	Kinova Actuator	12000$	6

APPENDIX C
TEST METHOD

APPENDIX D
LISTS

A. Hardware Components List

The table I list the big pieces of hardware that are used on this robotic system.

B. Software List

The table II lists most software in use in our robot.

ACKNOWLEDGMENT

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

TABLE II
SOFTWARE LIST

Name	Version	License	Usage
Ubuntu	18.04	Various (mostly GPL)	Nvidia Jetson Xavier
ROS	Melodic	BSD	Robotic framework
OpenCV	3.4.0	BSD	QR code
OpenCV	3.4.0	BSD	Landolt C
OpenCV	3.4.0	BSD	Motion detection
Octomap	1.7.1	BSD	2D SLAM
VectorNav	-	MIT	IMU
CTRE Phoenix API	5.6	Closed source	Motors control
Sick Tim	0.0.14	BSD	LiDAR
MobileNet	3	MIT	Hazmat Detection
TypeScript	3.6.5	Apache License 2.0	Web UI
ReactJS	16.12.0	MIT	Web UI
ElectronJS	8.0.1	MIT	Web UI
RobotWebTools	3.4.0	Proprietary	Web UI
MoveIt	1.0.0	BSD	Arm control

REFERENCES

[1] Ioan A. Sucan and Sachin Chitta, “MoveIt.” [Online]. Available: http://moveit.ros.org/

[2] Ryosuke Tajima, “jog_control.” [Online]. Available: https://github.com/tork-a/jog_control

[3] M. Labbé and F. Michaud, “Rtab-map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation,” *Journal of Field Robotics*, vol. 36, no. 2, pp. 416–446, 2019. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21831

[4] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, Q. V. Le, and H. Adam, “Searching for mobilenetv3,” 2019. [Online]. Available: https://arxiv.org/abs/1905.02244

[5] J. Redmon and A. Farhadi, “Yolov3: An incremental improvement,” *arXiv*, 2018. [Online]. Available: https://arxiv.org/abs/1804.02767

[6] J. T. Mohamed, Mahmoud and B. Mertsching, “Hazmat label dataset - seeing signs of danger,” 2018. [Online]. Available: https://osf.io/b5dap/