The Future of Engineering Education
2024 Annual Conference & Exposition
Oregon Convention Center
Portland, OR . June 23 - 26, 2024
ASEE
Paper ID #43476

Physical Robots for Teaching Mobility & Manipulation using ROS in Remote Learning

Prof. Maria Eugenia Cabrera, University of Massachusetts, Lowell

Maru Cabrera is an Assistant Professor in the Rich Miner School of Computer and Information Sciences at UMass Lowell. Before that, she was a postdoctoral researcher at the University of Washington working with Maya Cakmak in the Human-Centered Robotics Lab. She received her PhD from Purdue University advised by Juan P. Wachs. Her research interests aim to develop robotic systems that work alongside humans, collaborating in tasks performed in home environments.

Prof. John Raiti, University of Washington

Prof. John Raiti is an Associate Teaching Professor in Electrical & Computer Engineering at the University of Washington, and is the Technical Programs Advisor at the Global Innovation Exchange (GIX) where he teaches in the Interdisciplinary UW Master of Science in Technology Innovation degree program. He teaches UW graduate level courses in Sensors & Circuits, IoT and Connected Devices, Capstone-style Launch Projects, and Robotics (Mobility, Navigation, and Manipulation) with a focus on Human Robot Interaction (HRI).

Physical Robots for Teaching Mobility & Manipulation using ROS in Remote Learning

Abstract

Even though remote learning has been present and available in a myriad of topics before pandemic times, robotics remote learning had the limitation of interacting with robotic platforms through simulation alone. With COVID-19, postgraduate education was forced to move to remote learning. Birk et al.[1] conducted a reasonable practice for online teaching of a robotics course at Jacobs University Bremen. Although their lectures covered most of the robotics areas, they used pre-recorded videos to teach and did not hold labs to demonstrate the operations on real robots. The sudden pivot created a paradigm shift for robotics courses traditionally taught in-person where students had the opportunity to experience interaction with robotic platforms in simulation and with the physical platforms. To address the learning challenges and with remote robotics, we present two case studies where physical robots were used to teach concepts in navigation and manipulation. This included instances of long-distance robotic teleoperation as well as autonomous behaviors. The Robot Operating System (ROS) plus Gazebo, and RViz were used for teleoperation and autonomous routines with the Turtlebot3 and Kinova Gen3 lite robot platforms. Our results demonstrate how concepts in robotics such as SLAM can be taught in both simulation and with physical robots, and how students learn how to mitigate the errors and uncertainty produced in the real world with both mobile and manipulator robots. The technical challenges of communications time delay, real world error and uncertainty, and network infrastructure described here exemplify how postgraduate educational goals can be achieved through remote, collaborative-faculty-student project-based learning that can have broader impact for lab and project work.

Introduction

COVID-19 changed abruptly the way in which higher education was delivered by faculty and received by students, moving from in-person to remote learning. In particular, the change has been significant for postgraduate education where more than 1 million students are international[2]. By moving to a distributed classroom with students located around the world, teaching challenges for both faculty and students have been many, such as dealing with synchronous vs. asynchronous activities [3,4], multiple time zones, access to reliable internet [5], and exploring new ways to facilitate practical training and experiences [6]. Additionally, new and existing tools had to be incorporated or adapted to the change in learning environment (e.g. use of teleconferencing technologies, developing pedagogical materials in new formats, adapting interactive pedagogy techniques for online use). This myriad of challenges are particularly true for the field of robotics education. Within the range of robotics education, we specifically highlight teaching the theory and practice understanding how a robot works, with a strong emphasis on the robot's interaction with its environment, and how to develop robot behaviors through programs that make these interactions successful; for these reasons, the embodied aspect of robotics learning, creates a tangible tension with most online learning solutions.

We present two case studies for how the Robot Operating System (ROS) [7] can be used with online conferencing platforms, such as Zoom, to teach postgraduate robotics courses including robotics topics like mobility and manipulation at the Global Innovation Exchange (GIX), a partnership between Microsoft, the University of Washington (UW), and Tsinghua University. GIX offers a new interdisciplinary and multicultural graduate degree program experience focused on project-based learning. Through the two degree options: (1) 18-month Master of Science in Technology Innovation (MSTI) from UW, and (2) the 21-month extended Dual Degree, MSTI (UW) and MSEDSIT (Master of Science in Engineering Data Science and Information Technology, Tsinghua) prepare the next generation of innovators. Students graduate with the entrepreneurial, technical, design thinking, and user experience skills to be successful as a technical program manager or advance their academic goals.

Due to COVID-19 restrictions, the first MSTI Robotics-track cohort pivoted on a very short time frame to a remote learning setup, providing a real-time learning environment with access to a range of state-of-the-art robots via Zoom to enrich the cohort's educational experience. Through the infrastructure at GIX, students and educators from around the world were able log into a robotics server to directly control and program physical robots while viewing live video streams from cameras mounted both onboard each robot as well as an array of overhead cameras. Controlling robots at GIX from remote locations, through ROS, permitted to lower the entry barriers for students and educators to teach and learn robotics using a diverse collection of robots, from the Turtlebot3 the Kinova Gen3 manipulator arms, to sophisticated and expensive ($100,000) mobile manipulators such as the Fetch Mobile Manipulator. Students were also able to collaborate using project-based learning methods while being inspired to solve real-world challenges, enabling students to appreciate the practical value of the concepts being taught in the curriculum through robotic exercises conducted at GIX.

COVID-19 impacts on higher education COVID-19 started in December 2019 and spread rapidly to all over the world within four months, causing a huge pandemic. On March 11, 2020, The World Health Organization(WHO) declared the COVID-19 outbreak as a global pandemic[8]. Because the COVID-19 virus spreads through respiratory droplets from coughing, sneezing, or talking, mainly from human-to-human transmission, social distancing was enforced in many places around the world, thus the COVID-19 pandemic forced many schools to remain closed. More than 87% (1.5 billion) students around the world have been affected by school closures[9], which accelerated the trend of adopting online learning as an alternative for continuing education. For higher education, online learning represented a big challenge since not only students had to adjust, but many instructors required additional technological and pedagogical support to change the presentation of the materials and assessing students' engagement and performance [10]. As an additional challenge for higher education, there are more than one million international students at US universities, and after the outbreak, some of them were unable to return to the US to continue their studies [11]. This compounds the problem of pivoting to online learning by including aspects associated with latencies in communications and choosing times for synchronous interactions that are appropriate for multiple time zones at once.

Online robotics education In the specific area of robotics education, additional challenges emerge with the need to combine theoretical and practical learning in both hardware and software components of any robotic system. Some online robotics courses available in an online learning platform like Coursera [12], choose to focus on more theoretical and software development aspects of robotics education, whereas some other may include access to simulated robots or pre-recorded videos, limiting the opportunity for online students to gather comprehensive hands-on experience. In recent years, efforts from the Robot Ignite Academy, or the Construct [13], has been including aspects of both practical experience using simulated robots and providing access in their certificate-awarding courses to connect to real robots to test their developed software; this was not available during the pandemic. Bolano et al. [14] proposed a virtual reality framework for programming robotic tasks which can accomplish robotics teaching, but students still cannot gain the experience of operating the real robots. Birk et al. [1] conducted a reasonable practice for online teaching of a robotics course at Jacobs University Bremen. Although their lectures covered most of the robotics areas, they used the pre-recorded videos to teach and did not hold labs to demonstrate the operations on real robots. One of our important contributions is that not only did we provide comprehensive teaching on robotics theory and practice including not only simulated robots, we also developed a novel method to give students hands-on experience on operating physical robots in real time.

Physical robot platforms for robotics education There are several types of robots that can be used as physical robot platforms for robotics education. For mobile robots, Amsters et al. [15] used the Turtlebot3 mobile robot in master courses at KU Leuven. The Turtlebot3 can meet the requirements to teach concepts related to navigation, perception and motion planning as a physical robot platform. For robotic arms, the UR5, and Kinova arms can be used as physical robot platforms to demonstrate robotic manipulation, grasping and perception. White et al.[16] reported on the development of a robotics course intended for students in multiple disciplines like Computer Science, Mechanical and Electrical and Computer Engineering, where they combined hardware design and software development of robotic systems and behaviors through lab assignments and a final project. They used the VEX lego platform as well as Robot kits developed by the KISS Institute for Practical Robotics were purchased (`www.kipr.org`). Radlak et al.[17] integrated a robotic arm with computer vision techniques to create a project-based learning environment. Mobile robotic platforms and robotic arms are the two most common robot platforms for robotics education. In our setting, we used the Turtlebot3 as the mobile robot platform and the Kinova Gen3 lite as the robotic arm platform. They both have comprehensive packages developed on the robot operating system (ROS), which we elaborate on in the following section.

The Robot Operating System (ROS), rosbridge and rosjs, for Remote Labs ROS is an open-source collection of software and a message protocol for robot operation that has come into wide usage in recent years[7]; it is the most used robotic framework available today. And the programming languages most commonly used for robotics programming are mainly Python and C++. The system allows a robot to operate as a collection of independent processes, passing data back and forth among themselves. Research efforts to decrease the barriers of entry into robots have focused on web-based approaches. Over the past years, cloud robotics research by Prof. Chad Jenkins and the Brown Robotics Group have produced tools that allow users to access ROS-controlled robots over the internet: rosbridge, and the companion rosjs JavaScript library. The rosbridge protocol is an applications-layer network protocol and client/server implementation for ROS, that enables any application on the Internet to interface with and control robots[18]. The rosjs library consists of a collection of facilities for controlling a robot through rosbridge[19]. While remote robotics labs have been widely adopted by the research community, remote postgraduate robotics courses with physical robots have not been widely adopted[20,21]. One notable exception is the program at Worcester Polytechnic Institute which offers an online Master's in Robotics, but it is almost exclusively in simulation with part of one course using the PR2 robot.

Massive Open Online Course in Robotics (MOOC) Other attempts at remote learning in robotics include Massive Open Online Courses, MOOC. The attraction of this style of course is that it can reach a large number of students without the infrastructure challenges of physical classroom capacity. However, Pozzi et al.[22] describe how preventing students from getting lost in class is a significant challenge, and that the pace of the presentation of concepts should be slower and limited to at most two concepts. Other educators have tried to use flipped-classrooms in MOOC to address some of those limitations. De la Croix and Egerstedt[23] explain how they taught a flipped-class[24] robotics MOOC, Controls of Mobile Robots, offered at Georgia Tech. However, they also experienced significantly negative "But there are virtually no upper-level engineering MOOCs. And there is a reason for this – engineering requires prerequisites, such as calculus, linear algebra, and Laplace transforms. Moreover, engineering courses are made better with hands-on labs,and there is simply no way meaningful, physical labs can be made a part of a free, online course given the current state of the technology." While the work presented in this paper is not at the MOOC scale, the techniques described in our case studies below show the successful use of remote learning and in ways that can be scaled.

Robotics Lab 1: Sensing and Mobility

The graduate-level course is part of the MSTI-Robotics track at GIX. This new interdisciplinary robotics track draws from business, human-centered design, and robotics fundamentals like navigation and manipulation. The program was designed to integrate aspects including design, building, testing, and tuning. It provides the students with the opportunity to learn on real robotic hardware, familiarizing them with the current state-of-the-art robotics middleware ROS, while also understanding the broader context of collaborative robotic systems.

The course introduces students to robotic concepts and provides early exposure to applied work in Human-Robot Interaction (HRI) with teleoperated and autonomous robots. This robotics course enables students to learn how sensors are used in a variety of robotic platforms involved with perceiving the environment and executing actions in this environment, both in simulation and the real world. The course covers three introductory areas of robotics:

- General overview – robotic configurations, applications, sensors, actuators, control and real-world implementations of robotic components.
- Autonomous mobile robots – kinematics of mobile robots, localization and mapping, motion planning and autonomous navigation.
- Robotic arms – forward/inverse kinematics, trajectory generation and manipulation.

While environmental limitations due to COVID-19 limit teaching to remote learning, remote access to real robots is incorporated into the assignments and course structure. Expectations in the course include the deployment of the same robotic platforms both in simulation and physical instances.

This course is targeted towards students who want to gain a basic understanding of technologies used within robotic systems, namely autonomous mobile robots and robotic arms. Upon completion of this course, students should be sufficiently versed in robotics to support career roles in a broad range of industries related to robotics. The course provides an introductory foundation that should support further in-depth study of fundamental robotics technologies.

Assessment methods Given the aforementioned circumstances that combined into the development of this graduate-level course, several assessment methods were considered to promote student engagement and success, while considering diversity in academic backgrounds and previous knowledge on robotics. The class met twice every week. One of the days mostly focused on lectures on different topics, including low-stakes mini quizzes to assess engagement and level of understanding throughout the different lectures; the instructors also included opportunities for discussions, promoting student participation and relating the topics of the lecture with previous experiences that the students might bring from their previous majors. The second day of every week, provided the opportunity for practical work, briefly introduced with mini-lectures on the theme to be covered using simulated or physical robots. Usually, practical work began with simulated robots and following opportunities for practical work were attempted using the physical robotics platforms. During these practical workdays, assignments included deliverables where students needed to convey expectation of the results compared to the actual performance of the robot, discussing potential challenges that could explain the differences. Additionally to technical questions on robot performance on navigation and manipulation topics, the students also needed to submit media deliverables (e.g., images or videos) that displayed the robot behaviors; with these media assessments we also provided the opportunity for students to practice communicating results and using their previous learning experiences producing those deliverables.

Methods

In this section, we describe the preparation process around the robotics course, outside of the syllabus content. We provide details regarding the robotic platforms selected for the course, what the necessary infrastructure and logistics were, and how they were implemented to guarantee that remotely located students, some internationally, successfully connected with the robot and completed assignments pertaining to different objectives of the robotics course.

Robot Platforms Two robotic platforms were selected to address the different components of the robotics course syllabus: the Turtlebot3 for mobility and navigation component, and the Kinova Gen3 lite for manipulation. Among the common characteristics are: compactness, cost-efficiency, available code-base with examples ready to use, and compatibility with ROS.

1. Turtlebot3: as the name suggests, Turtlebot3 is the third instance of Turtlebot robotic platforms. This platform was released in 2017 with Dynamixel motors by Robotis, taking advantage of 3D printing technology to create modular plates where customizable sensors can be attached to. Additionally, using Arduino-based computing as its main onboard PC, they were able to reduce both the size and the price of the platform without sacrificing functionality and offering expandability.

   The model used during this course rendition was the smaller Burger model. Turtlebot3 Burger consists of a base, two Dynamixel motors, a 1,800mAh battery pack, a 360-degree LIDAR, an SBC (single board computer: Raspberry PI 3 and Intel Joule 570x) and a hardware mounting kit attaching everything together; its dimensions are L: 13.8, W: 17.8, H: 19.2 cm. The main difference in platform architecture between the Burger and the Waffle models, aside from size, is the lack of a camera sensor. Nevertheless, the Burger is a suitable robot to teach core robotics concepts associated with mapping, localization and navigation. The Turtlebot3 ROS packages (available as repositories in GitHub) provide the starting point for using the platform both in a physical application and in simulation; some of the most useful components are the teleoperation, SLAM and Navigation packages. They additionally provide example nodes with python scripts that showcase the use of sensor readings like odometry or laser data.

2. Kinova Gen3 lite: the newest and most compact member of the Kinova ultra-lightweight robot series. The Gen3 lite is a 6 degree-of-freedom robotic arm, with an integrated 2-finger gripper, ideal for light manipulation and mobile applications; it comes with a quick-connect base that easily attaches the robot's base to a surface.

   It is a more affordable option compared to the Gen3 version which includes a carbon fiber exterior, integrated torque sensors in each joint, and an integrated vision module. However, its cost-effective and ultra-lightweight presentation, provides the necessary tools to address educational materials comprising manipulation. The robot's base includes a controller with a Linux web server that manages connectivity between the controller and the arm devices, and between the controller and an external computer. This allows the robot to exploit the Kortex API functionality, both Gen3 and Gen3 lite use the same code-base, which provides services that act as translators between ROS nodes and the Kortex API, as well as the tools to simulate the robot using MoveIt! and Gazebo.

Infrastructure and Logistics Operating robots in the physical world requires infrastructure and logistics. In the context of a remote robotics course with assignments including physical robot operation, there were components of the course associated with: creating the conditions for the robot to interact in a constraint and safe environment, successfully connecting to the robots, as well as visualizing the robot performing actions in the environment.

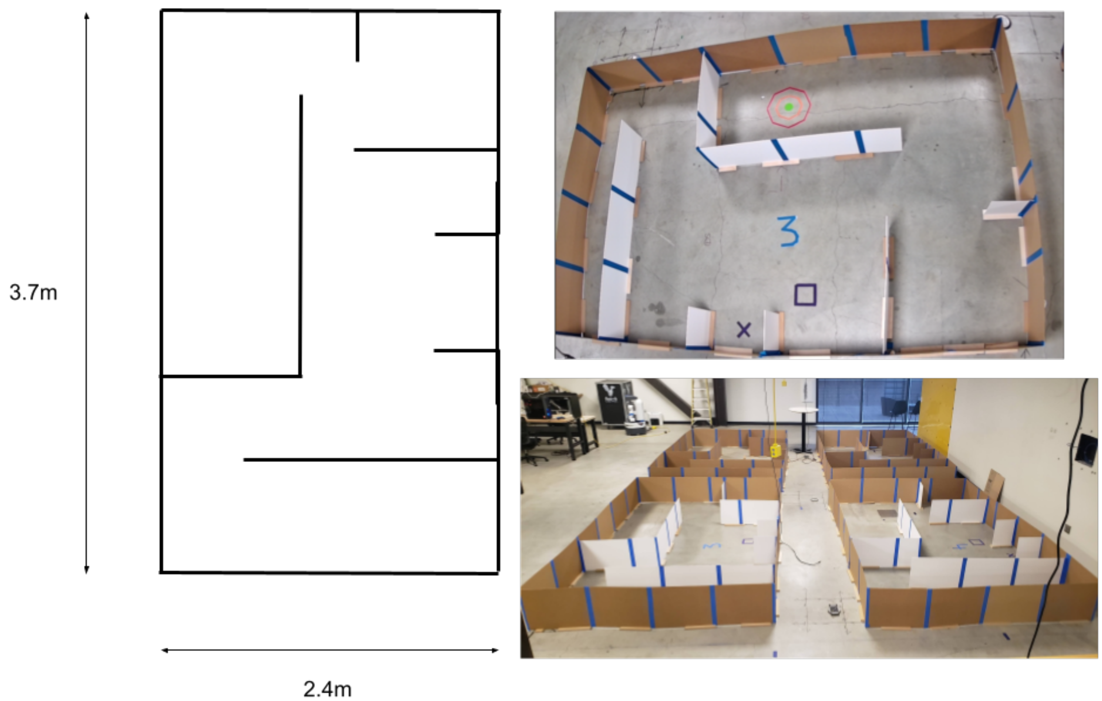

Figure 1: Navigation environment for the Turtlebot3. Left: diagram with measurements 3.7m long by 2.4m wide. Right Top: top-view from cameras. Right Down: all four built environments

1. Physical environment. Depending on the course topic and the robot used, the physical environment needs to provide context about the task to be completed and, of particular importance in remote conditions, the ability to visualize the environment and the results of said task.

   In the case of the Turtlebot3, the course topics were related to navigation, mapping and localization; hence, the workspace needed to guarantee the conditions to complete the task associated with those topics successfully. We designed a physical environment enclosed with walls about 40cm tall. To allow students the ability to work in parallel, four environments were built (shown in Figure 1, Right bottom). Each enclosed environment is 3.7 by 2.4 m, and can be navigated by the Turtlebot3 Burger, which has the LIDAR sensor at a height of 19cm, to effectively create a map and test navigation and path planning algorithms. In terms of providing a third person view of the environments, four cameras were respectively attached to the ceiling with a slight angle to avoid large fish-eye distortions when the camera was placed parallel to the ground; this view also avoided occlusions caused by the walls hiding the robot from view. Figure 1, Right top shows an example top view of one of the enclosed maps.

   The course topics associated with the Kinova Gen3 lite robot involved arm kinematics, manipulation and trajectory generation. All available arms (three Gen3 lite robots and one Gen3 research) were attached to an assembled workbench, all directed towards the outside. This setup guaranteed the robots' workspaces would not overlap and students could have multiple arm settings to share and work in parallel. In front of each robot, an additional table was placed with a printed $5x5$cm grid that aligned the origin of the Y-axis with respect to the base links; this also provided the students with a fixed reference to estimate size and distance in the real world. Given the 3D nature of robotic arm's operation, two camera views were offered per robot. A tripod was placed in the middle of the island table, with over-the-shoulder oblique views; two other cameras offered top view visualization for two robots at the time. The resulting views are shown in Figure 2.

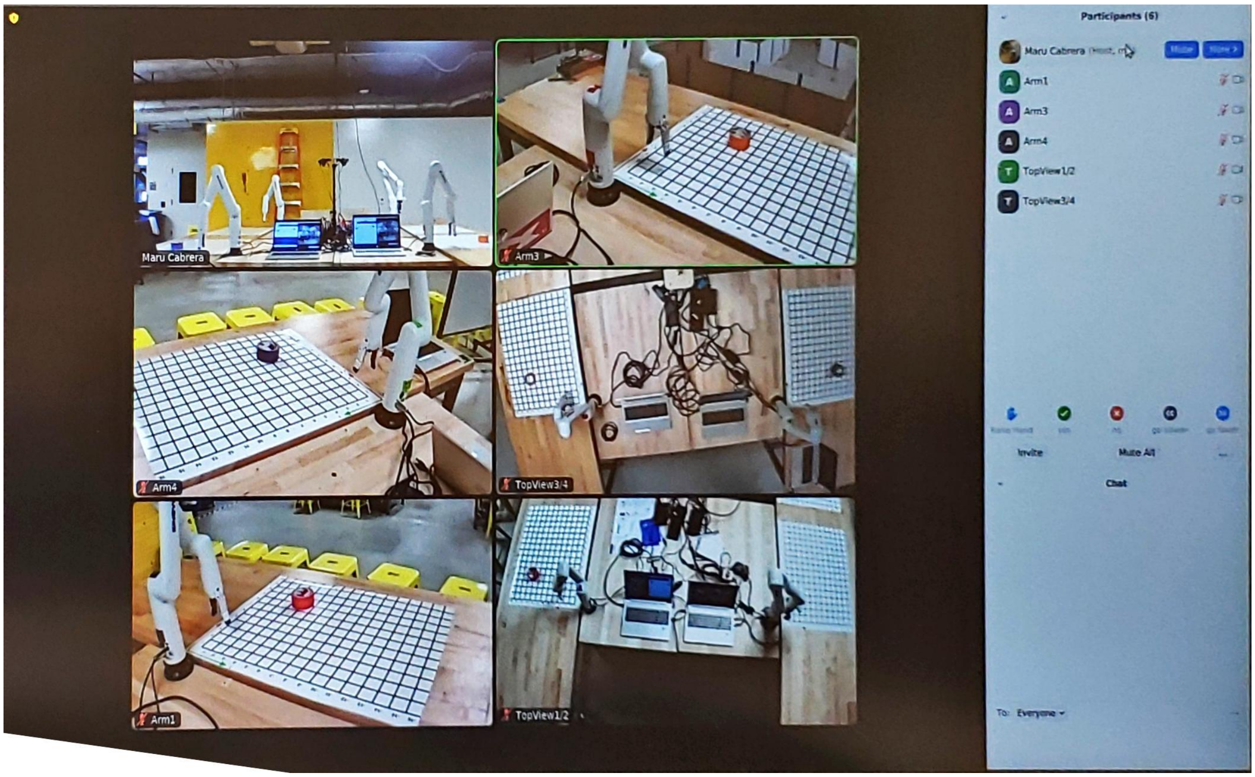

Figure 2: Multiple camera views available to operate the Kinova Gen3 lite remotely. Top left camera view is from the instructor's laptop showing the "Kinova island" setup.

2. Network Connectivity. Once the physical environment was set up, the following hurdle related to network connectivity to allow students to successfully connect and operate the robots.

   The students gained access to the university network using HuskyOnNet, an IT service which provides individual users (current UW students, faculty and staff) with a secure connection to the UW network from remote locations. The service needed to give access and route all their internet traffic to the correct sub-network corresponding to the department and building. Inside the building, a router with Wi-Fi capabilities was used as an access point to provide static DHCP IP addresses to all robots, filtered by their corresponding MAC addresses.

   In the case of the Turtlebot3 Burgers, a computer was used as a robotic server (also connected to the access point with another DHCP static IP). This server was used as the master for all ROS connections between robots and students computers. The connection architecture is shown in Figure 3. Both robots and remote computers, would assign their environment variables to point to a ROS master served by the external server. This distributed network also allowed better use of resources at each component, not overloading the robot's Arduino onboard computer with unnecessary processes, and keeping anything related to heavy visualization processes at the students' end.

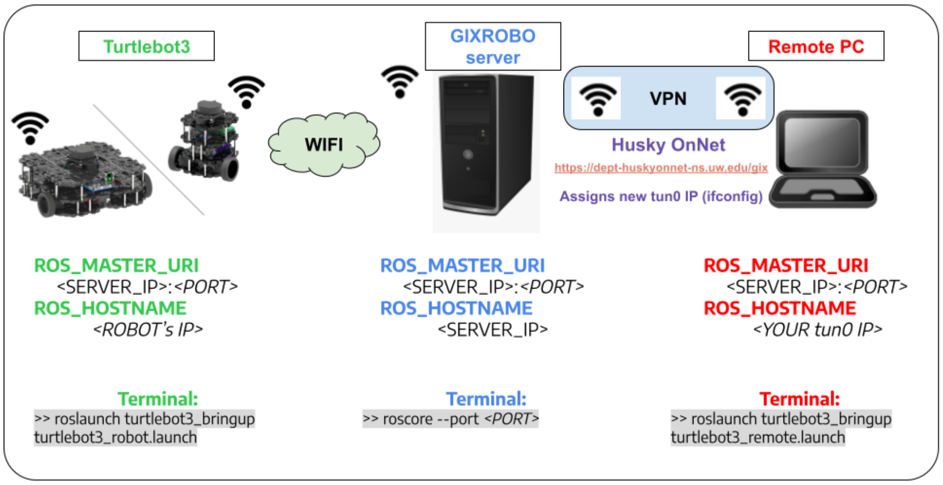

Figure 3: Network architecture for Turtlebot3 Remote Connection. The robot connects through Wi-Fi to a local server acting as the master, both robot and students' computers assign ROS environment variables to register nodes under the same ROS master with different hostnames.

The connectivity to the Kinova Gen3 lite started with the same remote access through the HuskyOnNet VPN service. However, the Kinova Gen3 lite offers a DHCP server on the robot's base, and receives commands from ROS through a node that connects to the Kortex API. By having the robot connect to the Wi-Fi network through our router access point, the students' remote IPs were able to successfully communicate with the robot on the same network. No additional configuration to the ROS environment variables were needed.

Case Study 1: Mobility using the Turtlebot3

We prepared two practical sessions with graded assignments that used the physical Turtlebot3 Burger to address mobility concepts. In previous practical sessions, the course had taken the students through topics like mapping and localization using the simulated robot in Gazebo. The practical sessions initially covered testing the remote connection and checking that all ROS nodes and topics were brought up correctly. The instructors were physically present in the building, and were responsible for launching the code to bring up the robots and physically place the robots in the enclosed environments. The learning objectives were:

- Address sensor inaccuracies and mitigate their impact in robot performance from simulation to the real world. They specifically had to implement code, observe the robot's behavior and compare and contrast odometry performance between simulation and real world implementation.

- Compare and contrast LIDAR sensor performance by executing the SLAM gmapping process. This requires teleoperation through the remote connection to explore the physical map.
- Explore challenges and limitations of different planning algorithms and their performance in real world operation (see Figure 4). This portion of the lab included autonomous navigation components.

The students were required to familiarize themselves with the remote connection and access to the physical environment and robot by using some of the code-base provided by Robotis github repository. They started by teleoperating the robot, estimating lag and adjusting their operation accordingly. There were instances were the camera views would glitch and freeze and recover after one second or so. Students reported using the visualization of the robot's status through RViZ as a secondary method of assessment. Issues with lag, both for sending robot commands and observing changes through the camera views, were normally below 0.5 seconds.

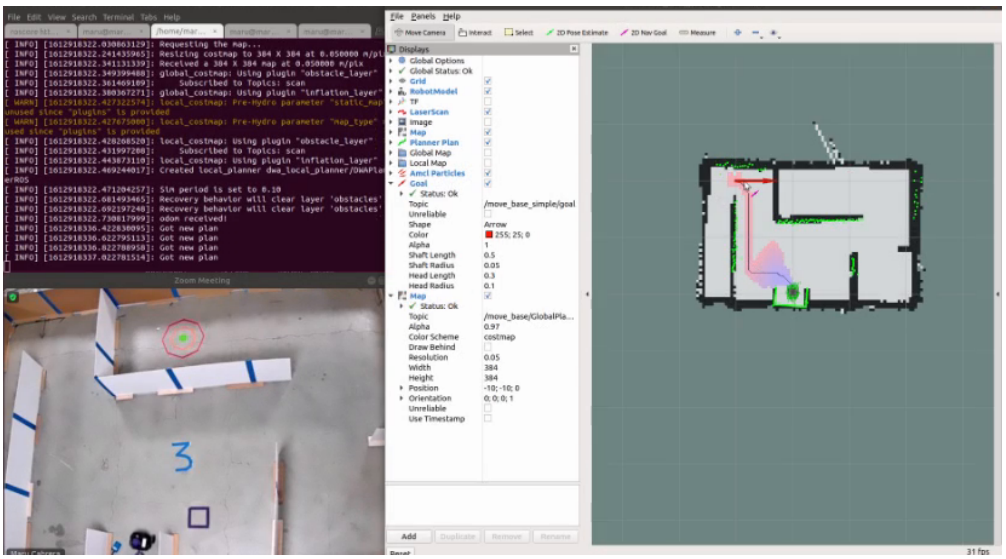

Figure 4: Screenshot including terminal output, camera top-view, and RViZ window with path planned.

Aside of connectivity issues like lag among the challenges present during the practical sessions with the physical robot, the students also experimented with discrepancies in performance of the same example code between simulation and real-world operation. One of the Turtlebot3 examples includes a patrol server/client. The students successfully used the pair of nodes to create patrol iterations in different shapes (circle, square) in the simulated world in Gazebo. However, during real-world implementation the 90^o turns taken during the square patrol example, resulted in a hit or miss where the robot would just pause and continue going straight.

Another one of the students, who was connecting remotely from China, experienced intermittent issues with the connection in terms of synchronicity and exchange of information between nodes across the different components of the system. At times, these issues made it impossible for the student to carry out a navigation goal for the robot to move autonomously.

Among the outcomes of these practical sessions, all students successfully teleoperated the Turtlebot3 around the enclosed environment, created the map, and sent navigation values through RViZ for the robot to autonomously go from an initial point to a selected position in the map.

Case Study 2: Manipulation using the Kinova Gen3 lite

We prepared one practical session with graded assignments that used the physical Kinova Gen3 lite to address manipulation concepts. In previous practical sessions, the course had taken the students through topics like forward kinematics/inverse kinematics, motion planning using MoveIt! and Gazebo. The learning objectives were:

- Develop the ability to control the robot either using the Kinova web app or using the Kortex Driver in Kinova arm ROS packages.
- Control the robotic arm in both configuration space and joint space to reach the same pose and make a comparison.
- Get familiar with MoveIt! as a ROS component with a physical Kinova Gen3 lite.
- Control the robotic arm to finish a task in real world such as ”pick and place”.

Most of the challenges that arose during the practical experience with the physical robots were similar to the case study 1, but for the robotic arms, there are some additional challenges. One challenge was that because of the latency, sometimes when students aimed to move a small value for one joint using the Kinova web app, the visual feedback would delay and the real robot would move by a greater value, which increased the probabilities of collisions with the desk, thus the instructor needed to pay a close attention to the arms' poses to mitigate those potentially disastrous situations. In the end, all of the students successfully controlled the robotic arm to complete the pick and place task using both RViz on ROS and Kinova web app.

Results

From both experiences using physical robots in remote robotics learning, the learning objectives were met by obtaining above passing grades in all proposed assignments. We believe there is still room from improvement in terms of infrastructure, systems architecture and flow of information.

There were issues that still need to be improved upon. For example, synchronicity issues when adding remote connections, and passing through VPNs caused additional strain on the distributed framework ROS is based on.

Some of the challenges also provided the opportunity for additional learning experiences. Even when the patrol server/client example failed during the real-world operation of the Turtlebot3, the process of evaluating the performance of the robot and comparing it with simulation had added value evaluating system's performances and trade-offs. The students were able to get first hand knowledge on the gap between operating simulated robots and physical robotic systems.

On a more subjective level, adding the physical component of the robotic platforms to the remote lessons gave the students an additional sense of accomplishment. Having the experience of watching an embodied platform perform a task is one of the most rewarding moments of working with robots. These remote connections and setups allowed the students to maintain those learning aspects, and also experience the challenges associated with sensor uncertainty (i.e. wheels slipping result in accumulation of errors in odometry).

Discussion

The two case studies above demonstrate the use of physical robots for navigation and manipulation tasks with communications time delays in order to show the feasibility of complex tasks for both teleoperation and autonomous robot actions. The technical challenges of communications time delay, real world error and uncertainty, as well as network infrastructure described here exemplify how postgraduate educational goals can be achieved through remote, collaborative-faculty-student, project-based learning.

Controlling robots in the lab from remote locations both domestically and internationally increases the accessibility of robotics education and decreases some of the fiscal and logistical barriers of entry. Students and instructors can program and run experiments with a diverse collection of robots, from Turtlebot3 mobile platform to Kinova Manipulators. Students and instructors can engage in the learning experience of robotics concepts while being inspired to solve real-world challenges, enabling students to appreciate the practical value of robotics concepts being taught in the curriculum through robotic exercises conducted in the lab with physical robots, including learning the nuances that arise while using the physical robot versus simulation. Since schools would not need to purchase the robots, nor pay to maintain or upgrade the robots, there exists the potential for rapid adoption. This has the potential to accelerate research and increase the opportunities for multi-institutional grant proposals.

There also exists the possibility to extend learning through web-based robotics for collaborative courses at multiple institutions both domestically and internationally. For example, our future work consists of piloting an international course offering this Summer in Tokyo between Shibaura Institute of Technology (SIT) and GIX on Human Robot Interaction to explore the business opportunities, understand the cultural differences for robot integration, while using international teleoperation plus autonomous robots remotely from Tokyo with the robots at GIX in Seattle. This experience will be both multicultural plus interdisciplinary as the course will be open to students from all majors and students will work in teams to learn how the next generation of service robotics is being developed in Japan – robots that are no longer limited to industrial and pre-programmed applications, but applications interacting with human users in everyday settings. Plus, learn the challenges in developing robots that operate in dynamic spaces with human users, and compare the trends in development between Japan and the United States.

References

[1] Andreas Birk, Evelina Dineva, Francesco Maurelli, and Andreas Nabor. A robotics course during covid-19: Lessons learned and best practices for online teaching beyond the pandemic. *Robotics*, 10(1):5, 2021.

[2] Amanda B Click. International graduate students in the united states: Research processes and challenges. *Library & Information Science Research*, 40(2):153–162, 2018.

[3] Elena V Frolova, Olga V Rogach, Alexander G Tyurikov, and Pavel V Razov. Online student education in a pandemic: New challenges and risks. *European Journal of Contemporary Education*, 10(1):43–52, 2021.

[4] Curtis J Bonk. Pandemic ponderings, 30 years to today: Synchronous signals, saviors, or survivors? *Distance Education*, 41(4):589–599, 2020.

[5] Tamer Sari and Funda Nayır. Challenges in distance education during the (covid-19) pandemic period. *Qualitative Research in Education*, 9(3):328–360, 2020.

[6] Tea Kasradze and Ekaterine Gulua. Challenges and opportunities for teaching practical skills at higher education institutions under the conditions of covid-19 pandemic. *European Journal of Social Science Education and Research*, 8(1s):63–78, 2021.

[7] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng. Ros: an open-source robot operating system. In *ICRA Workshop on Open Source Software*, 2009.

[8] Domenico Cucinotta and Maurizio Vanelli. Who declares covid-19 a pandemic. *Acta Bio Medica: Atenei Parmensis*, 91(1):157, 2020.

[9] Francisco Jonathan de Oliveira Araújo, Ligia Samara Abrantes de Lima, Pedro Ivo Martins Cidade, Camila Bezerra Nobre, and Modesto Leite Rolim Neto. Impact of sars-cov-2 and its reverberation in global higher education and mental health. *Psychiatry Research*, 288:112977, 2020.

[10] Ismail Celik, Egle Gedrimiene, Anni Silvola, and Hanni Muukkonen. Response of learning analytics to the online education challenges during pandemic: Opportunities and key examples in higher education. *Policy Futures in Education*, 21(4):387–404, 2023.

[11] Catherine E. Shoichet. Why the impact of coronavirus could be particularly bad on college campuses, 2020. URL `https://www.cnn.com/2020/03/07/health/universities-coronavirus-impact`.

[12] Coursera | Degrees, Certificates, & Free Online Courses, . URL `https://www.coursera.org/`.

[13] ROS Courses Library | Robot Ignite Academy, . URL `https://www.theconstructsim.com/`.

[14] Gabriele Bolano, Arne Roennau, Ruediger Dillmann, and Albert Groz. Virtual reality for offline programming of robotic applications with online teaching methods. In *2020 17th International Conference on Ubiquitous Robots (UR)*, pages 625–630. IEEE, 2020.

[15] Robin Amsters and Peter Slaets. Turtlebot 3 as a robotics education platform. In *International Conference on Robotics in Education (RiE)*, pages 170–181. Springer, 2019.

[16] William White, Jerry Weinberg, George Engel, Cen Karacal, and Ai-Ping Hu. Assessing an interdisciplinary robotics course. In *2005 Annual Conference*, pages 10–225, 2005.

[17] Krystian Radlak and Marcin Fojcik. Integration of robotic arm manipulator with computer vision in a project-based learning environment. In *2015 IEEE Frontiers in Education Conference (FIE)*, pages 1–4. IEEE, 2015.

[18] Christopher Crick, Graylin Trevor Jay, Sarah Osentoski, Benjamin Pitzer, and Odest Chadwicke Jenkins. rosbridge: ROS for non-ROS users. In *International Symposium on Robotics Research (ISRR 2011)*, Flagstaff, AZ, August 2011.

[19] S. Osentoski, G. Jay, C. Crick, B. Pitzer, C. DuHadway, and O. C. Jenkins. Robots as web services: Reproducible experimentation and application development using rosjs. In *Proceedings of the 2011 IEEE International Conference on Robotics & Automation*, 2011.

[20] Baichang Zhong, Jijun Zheng, and Zehui Zhan. An exploration of combining virtual and physical robots in robotics education. *Interactive Learning Environments*, pages 1–13, 2020.

[21] Saira Anwar, Nicholas Alexander Bascou, Muhsin Menekse, and Asefeh Kardgar. A systematic review of studies on educational robotics. *Journal of Pre-College Engineering Education Research (J-PEER)*, 9(2):2, 2019.

[22] Maria Pozzi, Monica Malvezzi, and Domenico Prattichizzo. Mooc on the art of grasping and manipulation in robotics: Design choices and lessons learned. In *International Conference on Robotics and Education RiE 2017*, pages 71–78. Springer, 2018.

[23] Jean-Pierre De La Croix and Magnus Egerstedt. Flipping the controls classroom around a mooc. In *2014 American Control Conference*, pages 2557–2562. IEEE, 2014.

[24] B Tucker. The flipped classroom-education next: Education next. *Education Next*, 12(1), 2012.