ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

RSL Robotic Systems Lab
Prof. Dr. Marco Hutter

Bachelor Thesis

Any Robot Controller Interface

Spring Term 2019

Supervised by:
Marco Tranzatto
Russell Buchanan

Author:
Dario Strübin

Declaration of Originality

I hereby declare that the written work I have submitted entitled

Any Robot Controller Interface

is original work which I alone have authored and which is written in my own words.[1]

Author(s)

Dario	Strübin

Student supervisor(s)

Marco	Tranzatto
Russell	Buchanan

Supervising lecturer

Marco	Hutter

With the signature I declare that I have been informed regarding normal academic citation rules and that I have read and understood the information on 'Citation etiquette' (`https://www.ethz.ch/content/dam/ethz/main/education/rechtliches-abschluesse/leistungskontrollen/plagiarism-citationetiquette.pdf`). The citation conventions usual to the discipline in question here have been respected.

The above written work may be tested electronically for plagiarism.

Place and date

Signature

[1] Co-authored work: The signatures of all authors are required. Each signature attests to the originality of the entire piece of written work in its final form.

Intellectual Property Agreement

The student acted under the supervision of Prof. Hutter and contributed to research of his group. Research results of students outside the scope of an employment contract with ETH Zurich belong to the students themselves. The results of the student within the present thesis shall be exploited by ETH Zurich, possibly together with results of other contributors in the same field. To facilitate and to enable a common exploitation of all combined research results, the student hereby assigns his rights to the research results to ETH Zurich. In exchange, the student shall be treated like an employee of ETH Zurich with respect to any income generated due to the research results.
This agreement regulates the rights to the created research results.

1. Intellectual Property Rights

1. The student assigns his/her rights to the research results, including inventions and works protected by copyright, but not including his moral rights ("Urheberpersönlichkeitsrechte"), to ETH Zurich. Herewith, he cedes, in particular, all rights for commercial exploitations of research results to ETH Zurich. He is doing this voluntarily and with full awareness, in order to facilitate the commercial exploitation of the created Research Results. The student's moral rights ("Urheberpersönlichkeitsrechte") shall not be affected by this assignment.
2. In exchange, the student will be compensated by ETH Zurich in the case of income through the commercial exploitation of research results. Compensation will be made as if the student was an employee of ETH Zurich and according to the guidelines "Richtlinien für die wirtschaftliche Verwertung von Forschungsergebnissen der ETH Zürich".
3. The student agrees to keep all research results confidential. This obligation to confidentiality shall persist until he or she is informed by ETH Zurich that the intellectual property rights to the research results have been protected through patent applications or other adequate measures or that no protection is sought, but not longer than 12 months after the collaborator has signed this agreement.
4. If a patent application is filed for an invention based on the research results, the student will duly provide all necessary signatures. He/she also agrees to be available whenever his aid is necessary in the course of the patent application process, e.g. to respond to questions of patent examiners or the like.

2. Settlement of Disagreements

Should disagreements arise out between the parties, the parties will make an effort to settle them between them in good faith. In case of failure of these agreements, Swiss Law shall be applied and the Courts of Zurich shall have exclusive jurisdiction.

Place and date	Signature

Contents

Abstract

This thesis discusses the development of a new controller interface for the quadruped robot ANYmal. Based on literature studies on the topic of "human robot interaction interfaces", design guidelines were elaborated to develop an interface which is intuitive, effective and easy to use. It runs on a touch screen display to ensure mobility and quick deployment.
Central to this task was the development of three widgets that run on the interface. A status bar, a panel to switch between controllers and launch actions and a widget to easily place individual feet of the robot. Together with a RVIZ display, showing the location and surroundings of the robot they form the core of the interface.
The interface was tested on an ANYmal robot at a former mine, to see how it performed under realistic conditions. It was able to successfully control ANYmal and proved to be operational and effective.

Symbols

Acronyms and Abbreviations

ARCI	Any Robot Controller Interface
ETH	Eidgenössische Technische Hochschule
ROS	Robot Operating System
RVIZ	A 3D visualization tool for ROS

Chapter 1

Introduction

1.1 Motivation

This Bachelor's Thesis is based on the ANYmal project. ANYmal is a quadrupedal robot, designed to move around and operate in challenging terrains [1]. To make use of its mobile and agile properties, it would be desirable to have a controller for teleoperation that matches these attributes. Currently ANYmal is operated through an interface that runs on a regular Linux computer and can be extended with a joystick controller to do the steering. The user interface, at its core, is a collection of individual widgets that all work on their own. While this preexisting interface offers all functions and information available to the ANYmal system, it lacks in efficiency and general user experience.

1.2 Goal

The aim for this thesis is the development of a new remote controller for ANYmal. For this controller a new custom graphical user interface needs to be designed to access ANYmal's main functionalities and enable teleoperation. The user interface should be effective and intuitive to use, thus making it easy for new users to pick it up. Furthermore the controller needs to be quickly deployable and mobile. To achieve this, the interface will be designed to run on a touchscreen display, connected to a joystick controller.

Figure 1.1: Example controller setup

Chapter 2

Methods

2.1 Definition of the User

In a first step it had to be defined who the target user for the Any Robot User Interface would be. On one hand it had to open up the audience compared to the old interface, which was too specifically targeted at people, who were themselves involved in the development of ANYmal and thus very familiar with the system. On the other hand it wouldn't make sense to define a too general audience for a product which was still almost exclusively used by robotics experts and developers. So it was decided to define the user as a person with some prior knowledge about the ANYmal robot or robotics in general. However the new interface still needed to improve on the shortcomings of the old one. It should feel intuitive and uncomplicated, enabling new operators to use it with little to no training or explanation.

2.2 Design Guidelines

In parallel to the development of this thesis, literature studies on the topic of "human-robot interaction interfaces" were conducted [2]. From the result of those studies, lessons and design guidelines to be used on the development of the new ANYmal interface were selected.

- **Natural and Intuitive Design**

  The use of the interface should be self-explanatory and only take minimal effort and attention of the user [3]. To achieve that, the design should be simple, minimalist and consistent [4]. A good approach is to use widely familiar design elements, e.g. from popular phone apps [5].

- **High Interaction Effectiveness**

  Interaction with the robot should take as little time as possible and lead to the result that the user intended [4].

- **Establishing Situation Awareness**

  The user should have a clear understanding of the robot's location, environment, status and activities [6]. Central role for teleoperation interfaces play the map-display and the video-feed, which should therefore receive the majority of screen real estate [6].

- **Attention Management**

  The interface should direct the users attention to relevant information and to the active task [3]. In order to do so, only provide relevant information in the given context [4] and use highlighting techniques such as color labeling [7].

- **Error Prevention**

  The interface should prevent errors and make sure the robot always falls back into safe states [7].

2.3 Framework

All the communication with the robot happens through ROS, which was predetermined by the ANYmal system. ROS offers a network in which individual processes, called nodes, can communicate by exchanging messages. The graphical user interface was written in C++ with the RQT framework, which combines ROS with the graphical library Qt. It enables you to write individual standalone widgets, which can then be arranged together in a window. Each widget is a node in the ROS network and can exchange messages and commands with other nodes and processes running on ANYmal. The advantages of this setup are that widgets from the old interface could be reintegrated if needed and it makes it easy for future development to add widgets for new functionality or specific tasks. The interface runs on a regular Linux Ubuntu 18.04 operating system, which was also held uniform to the rest of ANYmal development.

Chapter 3

User Interface

3.1 ARCI Overview

Figure 3.1 shows a screenshot of the new Any Robot Controller Interface. It consists of four main elements. In the center is the RVIZ display, showing the environment map and a third person view of ANYmal. The environment map is constantly updated using sensor input and can be used to determine the location of ANYmal as well as to detect moving objects and obstacles. The RVIZ display is the main tool at establishing situation awareness and following the design guidelines, was therefore chosen to be the most dominant element of the interface.
For the rest of the interface three new widgets were developed. Stretching across the top of the screen is the Status Bar widget, informing the operator about some critical status values. To run controllers and actions, the Fast Control widget on the bottom right was developed. And lastly placed above the Fast Control widget is the Foot Placement widget, a tool to quickly and easily place the feet of ANYmal on a selected location in the RVIZ display. This last widget is placed in a tab-window, where other widgets can be placed. Therefore allowing the user to switch between different tools, such as a startup utility or a live camera view.
The next sections are going to describe these three new widgets in more detail.

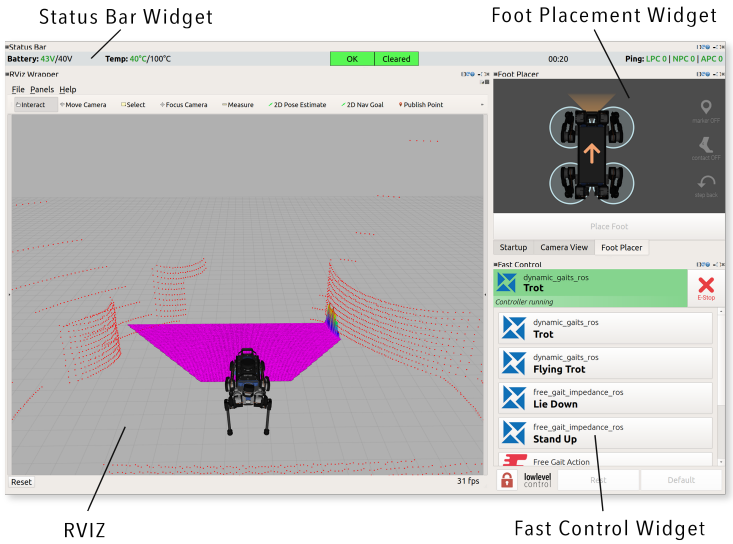


Figure 3.1: Screenshot of the Any Robot Controller Interface

3.2 Status Bar Widget

The status bar is a thin widget at the very top of the interface. As the name suggests, its task is to inform the user of the robots status and to give constant updates on some of the most critical information about the system. It is a very common interface element, known for instance from smartphones or tablets. This use of familiar design offers a comfortable user experience [5] and is in compliance with the design guidelines.

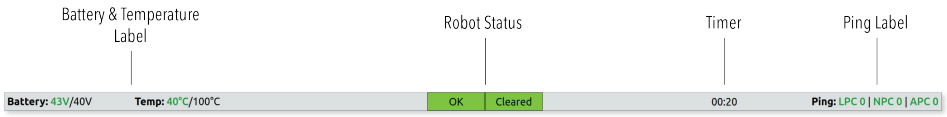


Figure 3.2: Status Bar Widget

On the very left it displays the battery voltage and the temperature. Since the battery of ANYmal doesn't directly output its voltage, the widget looks at the data from all of the twelve drives installed on ANYmal and displays the lowest value. Similarly the temperature label shows the highest temperature across all the drives.

Thresholds to change the color of the two labels can be set in the Status Bar's parameter file. Once the voltage goes below, or in the case of the temperature above the critical threshold, the whole status bar changes its color to red to ensure the operator becomes aware of the emergency and stops the operation. The critical threshold can always be seen next to the measured value to help the user interpret the data and put it into perspective.

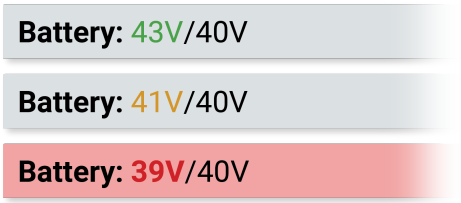


Figure 3.3: Battery thresholds

In the middle of the Status Bar is information on the robot state and the clearing status. The robot state can switch between 'Failproof', 'Emergency' and 'OK'. The clearing status is either true or false, signalized by the label with a green/red background and gives information on whether ANYmal is cleared to engage in any action or not.

To give information about the time passed since the start of operation, a small timer was added to the Status Bar. This can be a helpful tool for any tasks with time constraints.

And lastly on the very right is a label that displays the ping to the three computers running on ANYmal. This is crucial to inform the user on whether the interface is connected to the robot and on the delay of that connection. The three ping values are also color coded to describe the quality of the connection.

3.3 Fast Control Widget

The aim for the Fast Control widget was to develop a compact and easy to use tool to switch between controllers on ANYmal. Previously on the old interface this task was split up between different widgets, one for low-level controllers, one for high-level controllers and then two more responsible for Free Gait actions, located in a different tab.

High-level controllers for ANYmal have a certain structure hierarchy. At the base there are the 'controllers' (such as Dynamic Gaits or Free Gait), which in itself offer just the framework for individual motions or gaits. Those are then specified as 'modes' within a controller and describe an actual control behaviour. In some cases there are even 'sub-modes', for example 'trot' is a sub-mode of 'walk', which is a mode for the Dynamic Gaits controller. In the old interface changing the gait of ANYmal required multiple actions since the user had to first run the necessary controller, then select and run a mode and in some cases even a sub-mode. Making switches between gaits and motions more efficient was an additional goal for the Fast Control widget.

The Fast Control widget is divided in three parts: A top section giving information on the currently running controller, a middle section where the user can select high-level controllers and modes, and a bottom section with low-level controllers.

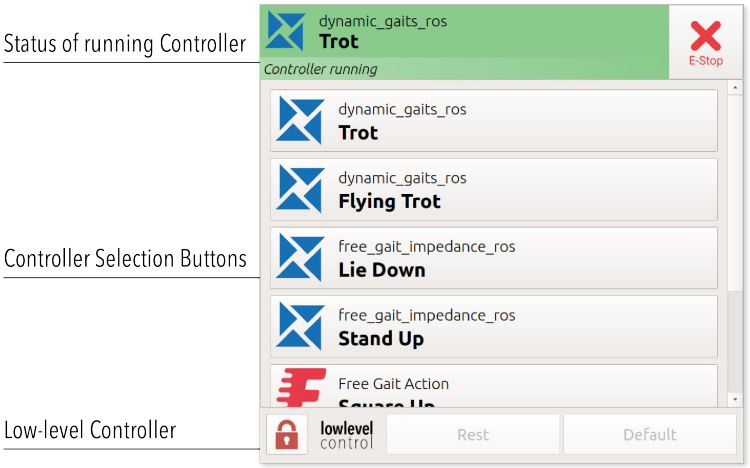


Figure 3.4: Fast Control Widget

Top - Status of running Controller
the top section of the widget displays the currently running controller and mode with information on it's status. A green background symbolizes that the controller is running as expected. In case of failure the background color turns red to bring the error to the attention of the operator. If a Free Gait action is running, the background transforms into a progress bar, updating on the current stand of the execution. Also for Free Gait action buttons a semi-transparent label displaying 'action' overlays the icon to distinguish it from regular modes. A small bar at the bottom of the label informs the user of the widgets activity with a short description. the E-stop button allows the user to stop the controller at any moment and put ANYmal into emergency mode.

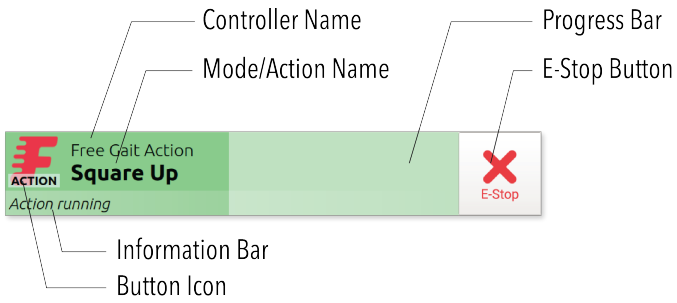


Figure 3.5: Top section of the Fast Control widget during execution of a Free Gait action

Middle - Controller Selection Buttons
In the middle section of the widget are the buttons that can launch the high-level controllers and modes. Each button contains a desired mode, its corresponding controller and where necessary a sub-mode. So to change the gait of ANYmal, the user only has to press one button and the widget will automatically do all intermediate steps to end up in the desired mode. Internally this is achieved by creating a dedicated thread to make the controller and mode switches. This use of multitasking ensures that the user interface doesn't freeze during the switching process. The thread switches the low-level controller to 'operational', then launches the controller, afterwards the mode and if necessary the sub-mode. Reducing this process to a single tap of the operator improves the interaction effectiveness and makes this somewhat confusing process more simple and intuitive.

The button height was set to a comfortable 1.5cm to make the interaction by touch easy and reduce the possibility of selecting any wrong command by mistake. The buttons were arranged in a scrollable panel, which was chosen as design element because of its familiarity from smartphones and because it allows to easily expand the list with new buttons.

To ensure that the user only gets the controllers that he wants, the widget builds its buttons off of a parameter file. In that file he can add either a controller/mode combination, a Free Gait action or an entire Free Gait collection. Further it is possible to add optional attributes such as an icon or a different display name. See section A in the appendix for a description of the parameter file.

Bottom - low-level Controller
On the bottom of the widget are the buttons to put ANYmal into the low-level states 'Rest' or 'Default'. Since selecting any of these buttons by accident may lead to catastrophic consequences, they are protected with a lock mechanism. The buttons are disabled and greyed-out by default and only become clickable by pressing the lock button first.

Clear Button
The clearing state is a safety mechanism of ANYmal that ensures the robot does not engage in any activity before the operator tells it explicitly that it is safe. Whenever ANYmal has not been cleared yet, the top section of the Fast Control widget will inform the operator of it and offer a large 'Clear' button. At the same time all the other buttons on the widget will be disabled, to make sure the user doesn't try to select a button in vain and becomes aware of the need to clear ANYmal first. Once the Clear button has been pressed, it will disappear. As soon as ANYmal needs to be cleared again, e.g. after an e-stop, the Clear button will reappear. Making the Clear button only show up when needed was chosen over having it fixed because it

wouldn't serve any purpose anymore once ANYmal was cleared. This ensures that the operator knows intuitively when to use the clearing function, avoids confusion about what it does and saves space in the interface.

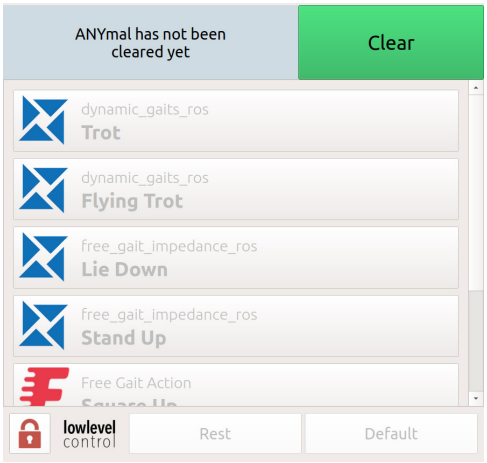


Figure 3.6: Clear Button

3.4 Foot Placement Widget

The Foot Placement widget is a tool that makes it easy to move the feet of ANYmal to a selected target on the RVIZ display. Previously the fastest way of doing this was to manually write a Yaml-file with the destination coordinates and the instructions for a Free Gait action. This was a very slow process, required the operator to know the Free Gait commands and didn't give any visual feedback on where the foot would be placed. The Foot Placement widget offers an easy graphical interface to facilitate this process. Placing a foot happens in three steps:

1. The user places an interactive marker in the RVIZ display at the location where he wants to place a foot.
2. In the Foot Placement widget the user selects the foot that needs to be moved. They are represented as four circles around the legs of ANYmal.
3. By pressing 'Place Foot' the action is generated and the robot will execute the command.

To execute the task the Foot Placement widget will switch to the Free Gait controller and generate an action. The target will be reached using a 'Footstep' action with a triangular trajectory to avoid collisions with the ground or other objects. Additionally an automatic base motion will be generated to balance the robot.

Interactive Marker
An interactive marker symbolizes a 3D-point in the RVIZ display that can be moved around by the user. It represents the target position for the step action.

The interactive marker can be activated in two ways:

- By toggling on the marker option in the foot placement widget. This will create an interactive marker at the base of ANYmal.
- By publishing a point in RVIZ. This works by pressing the 'Publish Point' button and tapping on a surface in the RVIZ map. The interactive marker will be created on this selected point.

Once the marker was created the user can change its position in the RVIZ display. Dragging the marker will move it in the xy-plane without changing the height. To change the z-coordinate the two arrows can be pressed and dragged. This enables the user to reposition the marker with intuitive touch-interaction.

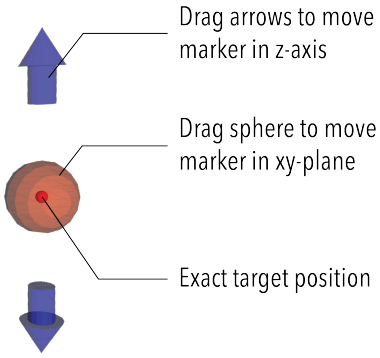
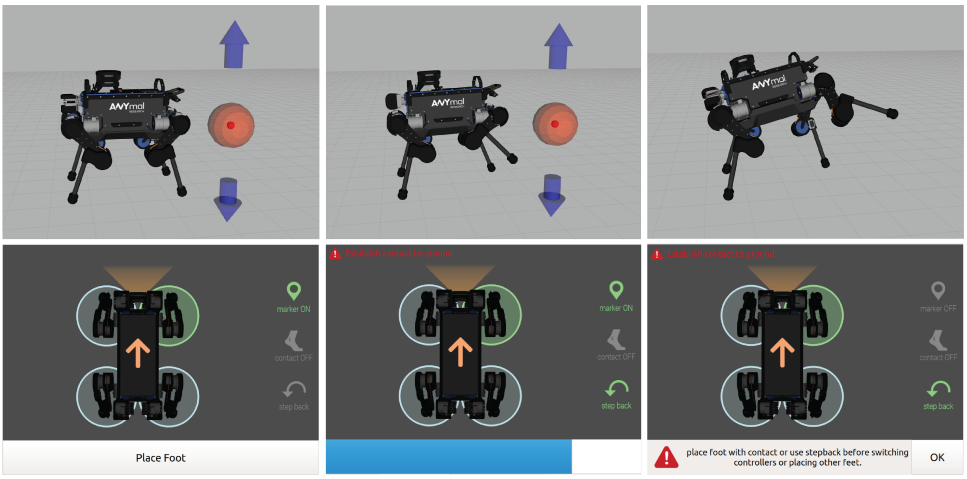


Figure 3.7: Interaction Marker

Contact Flag and Step Back Button
The second button in the Foot Placement widget is called 'Contact' and can be toggled on or off. Its function is to tell Free Gait whether it should look for contact to the ground or not. Toggling it off will allow to place a foot up in the air or on a vertical wall. However it's important that a foot always reestablishes contact to the ground before a different leg is moved or another controller is activated. Otherwise a crash of the robot is possible, since Free Gait will not use that leg as a support-leg. To facilitate this process the user can make use of the 'step back' button. Pressing it will always return the foot to the last position where it had contact to the ground. Since it could result in catastrophic failure if the user was not aware of the need to reestablish contact to the ground, warnings were put in place. In the top left corner a warning sign will always appear whenever a foot has no contact to the ground and is not used as support-leg. Additionally a prompt will appear and force the operator to acknowledge the need to use the step back button.



Place interactive marker and choose foot | Step action in progress | Step action completed, warning prompt

Figure 3.8: Process of placing a foot with the Foot Placement widget

Chapter 4

Results

During development of the Any Robot Controller Interface, it was frequently tested on a simulation of ANYmal running on Gazebo, to save time and effort. At the end of the development process, the interface was then tested twice on a real ANYmal robot. Once in a lab environment and once at the "Bergwerk Gonzen", a former mine in Sargans Switzerland.

Tests at the Lab
The first tests mainly served the purpose of checking whether the interface was operational and could be used for tests in less controlled environments, such as the Gonzen mine.

The connection to the ANYmal robot was established successfully and information and commands could be sent back and forth. To test the Foot Placement widget, a plastic bottle was placed on a box to knock over. Using an intermediate step, placing the foot in the air in front of the bottle first, the experiment could be completed successfully.

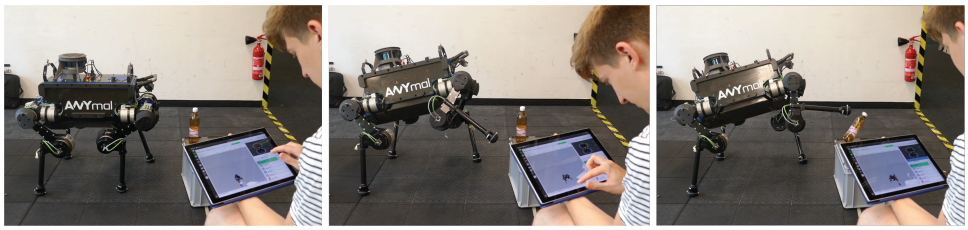


Figure 4.1: Pushing over a bottle with the Foot Placement widget

Tests at the Bergwerk Gonzen
At the mine in Gonzen more tests were done in an environment that could match a real use scenario. As an example ANYmal could be used to determine if the machinery down in the mine is still running by measuring vibrations with haptic sensors on its feet. Therefore it was set as a goal to walk up to the machine and use the Foot Placement tool to touch it.
The experiment could be completed successfully, with ANYmal touching the machine from different positions. However in a similar experiment where the goal was to touch a wall of the mine, it was observed that sometimes ANYmal would not completely touch the wall but stop a few centimeters in front of it. The reason for this being that the widget doesn't get any feedback from its environment but only

tries to touch a point in its virtual map. If the point cloud from the sensors doesn't precisely match the real environment then the targeted point will not match with the desired destination.

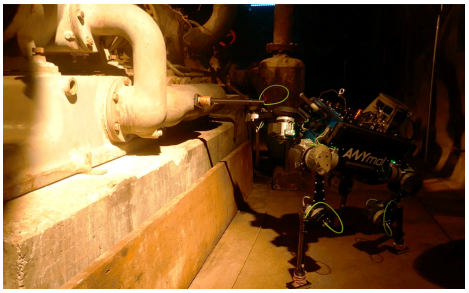


Figure 4.2: ANYmal making contact to mine machinery

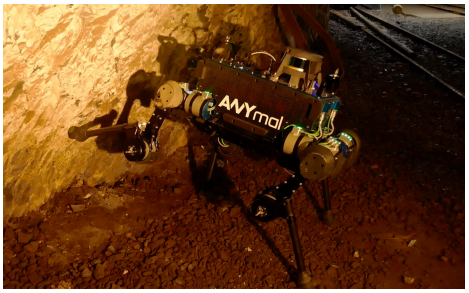


Figure 4.3: ANYmal missing the wall by a few centimeters

Chapter 5

Conclusion and Outlook

The new Any Robot Controller Interface was successful at offering the core functionality of the old ANYmal interface in a new and improved design, following the guidelines extracted from the literature:

- The interface was designed to be as intuitive as possible, making it easy to switch between controllers or to create Free Gait actions to place the feet of ANYmal. Inspiration for the design of the widgets was always taken from already familiar elements in smartphones or tablets, e.g. the Status Bar or the scrollable button list in the Fast Control widget.
- High interaction effectiveness was achieved by simplifying the interface where possible and reducing all the necessary interaction to a minimum, e.g. by making the switch of a controller and mode only need one tap.
- Situation awareness is established with the large RVIZ display, informing about location, surroundings and activity, as well as the Status Bar and Fast Control widget reporting on the status and activity as well.
- The interface guides the attention of the user to important sections. Color coding was used in the Status Bar and the Fast Control widget to indicate when something is failing (switch from green to red). The RVIZ display was made comfortably big to be the natural center of attention.
- To prevent user errors, warnings were placed where they could avoid catastrophic failures. E.g. the Foot Placement widget forces the user to acknowledge the need to make contact to the ground or the Status Bar informs of critical voltage and temperature levels, such that the operator can stop the robot in time.

In the end during the two tests on real ANYmal robots the interface proved to be functioning and operating as intended.

While the interface has come to the point in development where the core functionality of ANYmal is covered and functioning, it also has to be noted that there's still work needed to complete the project.

A still missing part is the physical joystick to be connected to the interface. This is a crucial part to complete the teleoperation capabilities and making the Any Robot Controller Interface fully operational. Since the joystick hardware was not acquired at the time of the ARCI development, its implementation could not make it into this thesis.

Other work to be done is to further improve the user error prevention of the interface. The Foot Placement widget so far lacks the ability to verify whether a requested footstep is safe to perform. Possibilities for the implementation could be checks for self-collision or range of the step. Further the Fast Control widget should ensure that the operator cannot switch to controllers, which are not safe to launch from ANYmal's current pose and activity.

Lastly it would be beneficial to implement a controller into the Foot Placement widget that incorporates haptic feedback, to improve its capabilities of making contact with objects.

Bibliography

[1] L. A. G. F. B. C. D. T. V. F. P. D. R. B. S. B. M. K. H. B. M. I. L. Hutter M., Gehring C. and M. K., "ANYmal - toward legged robots for harsh environments," *Advanced Robotics*, vol. 31, no. 17, pp. 918–931, 2017.

[2] S. D., "Human robot interaction interfaces," 2019.

[3] M. A. Goodrich and D. R. Olsen, "Seven principles of efficient human robot interaction," *IEEE International Conference on Systems, Man and Cybernetics*, 2003.

[4] K. C. X. M. Adamides George, Christou Georgious and H. Thanasis, "Usability guidelines for the design of robot teleoperation: A taxonomy," *IEEE Transactions on Human-Machine Systems*, vol. 45, no. 2, 2015.

[5] M. M. Frank Jared A. and K. Vikram, "Realizing Mixed-Reality Environments with Tablets for Intuitive Human-Robot Collaboration for Object Manipulation Tasks," *IEEE*, 2016.

[6] K. B. Drury Jill L. and Y. H. A., "Lassoing hri: Analyzing situation awareness in map-centric and video-centric interfaces," *HRI*, 2007.

[7] S. Aaron, "Interface lessons for fully and semi-autonomous mobile robots," *IEEE*, 2004.

Appendix A

Fast Control Parameters

The buttons in the Fast Control widget can be created and edited in the 'params.yaml' file, located in the config folder of the Fast Control repository.

There are two types of buttons that can be added:

- In the section fast_control_modes buttons can be created that launch a controller first, followed by a desired mode and optionally a sub-mode.
- In the section fast_control_actions buttons can be created that launch a Free Gait action. A single action can be added with its action id. Alternatively by adding a collection id, a button for each action in that collection will be created.

To add a new button append the name of the mode, the action id or the collection id to the corresponding list. Then add the parameters of the button, see figure A.1 for a description of the parameters and the general structure of the document. The parameters marked as 'optional' can be left empty by typing two single apostrophes.

config/params.yaml

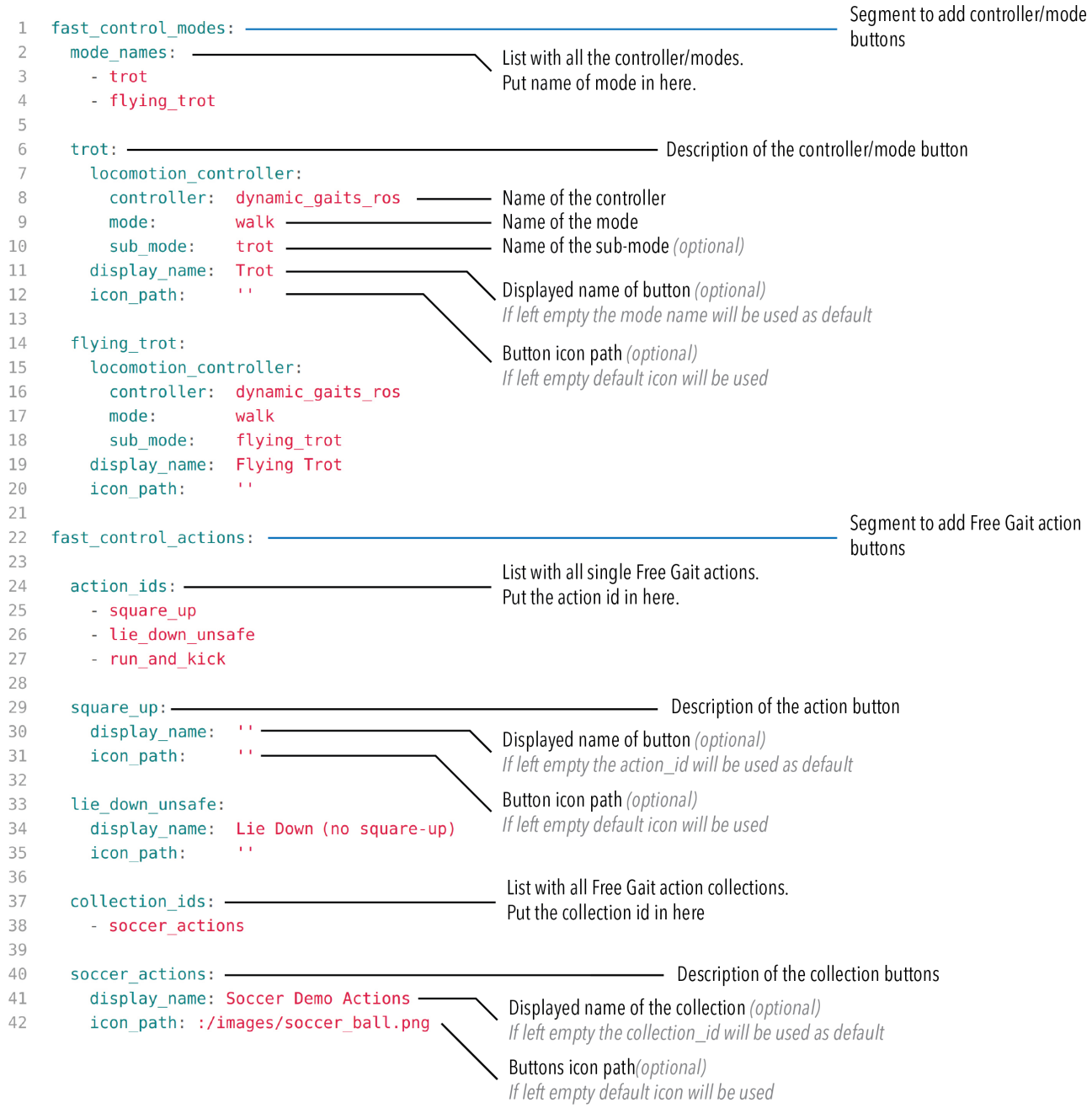


Figure A.1: Parameters