École Publique d'Ingénieurs en 3 ans

Report

Second Year Internship

August 22, 2023,

Eliott ECCIDIO,
2A ISIA
eliott.eccidio@ecole.ensicaen.fr

School Tutor:
Sebastien FOUREY
Laboratory Supervisor:
Ian MANCHESTER
Nicholas BARBARA

ÉCOLE PUBLIQUE D'INGÉNIEURS
CENTRE DE RECHERCHE

www.ensicaen.fr

ACKNOWLEDGEMENT

First, I would like to sincerely thank Ian Manchester, my supervisor, for allowing me to work within his team and projects. Working with this team allowed me to discover the usage of artificial intelligence with robots and also to have a first glance at the work done by PhD students.

I would also like to thank Nicholas Barbara for giving me guidance and advice throughout this internship. He gave me a lot of his time to help me with my contribution to his project.

I’d want to express my gratitude to Mrs. Annette Karydis for her constant support and help with logistics during the entire internship.

I cannot conclude this acknowledgment without expressing my sincere thanks to all the members of the lab who helped me get introduced to the team. Especially Jack Naylor and Jesse Morris, with whom I shared unforgettable moments and made these 17 weeks pleasant.

Thus, I am justly thanking the entire ENSICAEN teaching team for the quality and rigor of their academic teaching.

TABLE OF CONTENTS

INTRODUCTION

The starting point for this internship was the research period, and so the research for internships in domains that interested me. As this is my first internship, I needed to gain as much knowledge as possible, so it had to meet certain criteria. In fact, this internship had to be in an English-speaking country and related to artificial intelligence. The reasons behind these criteria are that fluency in English is mandatory for an engineer, and artificial intelligence is the field in which I want to work.

Given these criteria, I contacted many companies and laboratories, especially in Australia and Canada. I was not particularly picky because I was looking for the usage of artificial intelligence by these entities to understand the stakes of its use for them.

Furthermore, I focused on laboratories because they allow me to study some of the latest scientific contributions that use artificial intelligence. In addition, I may be doing a PhD after my degree, so it would be the best opportunity to discover how it is organised and to know the keys to success in the pursuit of a PhD.

In this respect, I contacted many researchers, and my supervisor gave me the opportunity I was looking for. Not only did it meet my criteria, but it would give me a deeper knowledge of reinforcement learning, which I was not very educated in during my studies.

So you can discover the world in which I evolved, I will present to you a detailed description of the laboratory, my contribution to one of their projects, and an assessment of my acquired skills.

At the end of this development and in conclusion, I would like to share with you my feedback and my outlook for next year and beyond.

DEVELOPMENT

B.1/ ACFR: Laboratory presentation & organisation

In just a few decades, the University of Sydney has become one of the most prestigious universities in the world. In particular, it is part of the famous "Group of Eight", the Australian equivalent of the Ivy League in the United States.

I had the opportunity to carry out my internship at the **Australian Centre for Field Robotics (ACFR)**. The laboratory is dedicated to the research and teaching of concepts relating to intelligent autonomous systems at *the University of Sydney.* Originally established as an ARC Key Centre of Teaching and Research in 1999, it now forms part of the ARC Centre of Excellence for Autonomous Systems along with groups at the University of Technology, Sydney, and the University of New South Wales.

The Australian Centre for Field Robotics is one of the largest robotics research institutes in the world and is well-known in the robotic field. They are often represented at the biggest scientific conferences, such as ICRA, ICML, IROS, ect.

Here is a detailed description of the ACFR:

1.1/ Main mission & focus

The ACFR is dedicated to developing innovative robotic theories and real-life applications of them. Some of these applications are developed in collaboration with companies such as Thales. These partnerships are a considerable help in funding the different teams in the laboratory. The laboratory is divided into teams to be able to cover a wide range of environments (marine agriculture, mining, ect.).

1.2/ Research Areas

The ACFR aims to develop technologies in four core areas:

- *sensors, fusion, and perception*
- *movement, control, and decisions*
- *modeling, learning, and adapting*
- *architectures, systems, and cooperation of robotics and intelligent systems.*

Let me elaborate a bit more on those areas:

1.2.1/ Perception and Sensing

In this category, we will focus mainly on *the agriculture team* by giving a quick presentation of their work and the purpose of their research. Professor *Salah Sukkarieh* is the one in charge of this subteam. The agriculture team works with different robotic platforms in collaboration with local farmers. These robotic platforms have different purposes:

- Taking pictures of the crops & classifying the pictures taken
- Spraying some chemicals on some unhealthy crops
- Navigating in farm environments

I'll detail, more specifically, the research of one of my good coworkers, *Nicholas Harrison*. His research consists of making maps and inferences about farms. He uses some samples of soil and spectral images of crops to be able to map the crops health within the farm. In this regard, he shares the data analysis with biologists to help him reach the highest accuracy on the crop's health conditions. *The first figure* is an image of the robot used for such tasks:

Figure 1: ACFR's SwagBot [1]

Besides, the robotic platforms used by the agriculture team are also used for auxiliary purposes, such as evaluating the density of a fire over time to help the firefighters.

1.2.2/ Control and Planning

This is the team I was working for. The team is directed by my supervisor, Professor *Ian Manchester*. Within the team, we have various subteams dealing with various robotic platforms.

The subteam for whom I worked will be explained further in this report. Nonetheless, Tara Bartlett, who works with us, had the opportunity to do an internship at JPL (Jet Propulsion Laboratory) to enrich her knowledge and skills. Her goal was to build a sample recovery helicopter. ***The Sample Recovery Helicopter*** is a backup sample tube collection method that may be part of ***the Mars Sample Return Campaign***. The design is based on ***the Ingenuity Helicopter***, but with the added ability to drive and pick up sample tubes with a robotic arm. She was working on the software development, integration, and testing of the ground mobility subsystem. This involved working with the prototype testbed pictured here in a live simulation environment.

Figure 2: Sample Recovery Helicopter

Apart from the subteam in which I was, there is the subteam run by Johnny Cheng. His team works on surgical robots to improve the current knowledge of treatment planning for critical diseases. To do so, he works on ***The KUKA*** (highlighted in the 3rd figure), which is part of a unique place called ***The Hybrid Theatre***.

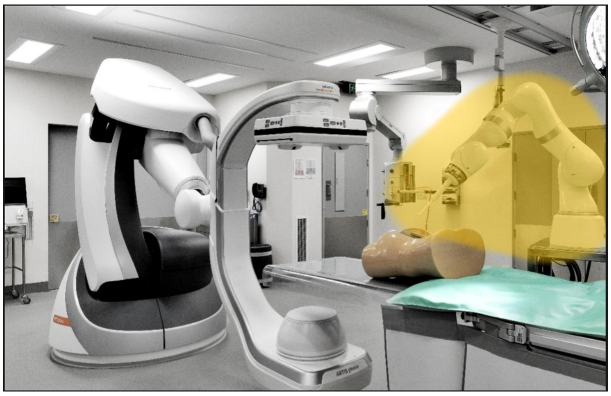


Figure 3: KUKA Robot

1.2.3/ Field & Marine Robotics

The ACFR Marine Robotics Program is one of the biggest and most influential groups in the ACFR. This group is mainly dedicated to the analysis of the seafloor. They have developed applications such as «high-fidelity, three-dimensional models of the seafloor precisely matching survey locations across years to allow scientists to understand variability in these environments» [2]. To obtain the data, the ACFR possesses a major national **AUV** (Autonomous Underwater Vehicle). The “**SIRIUS**” *(represented in the fourth figure)* robot has been developed from that perspective by Professor Stefan Williams and his team. It is used not only in Australia but also Greece.

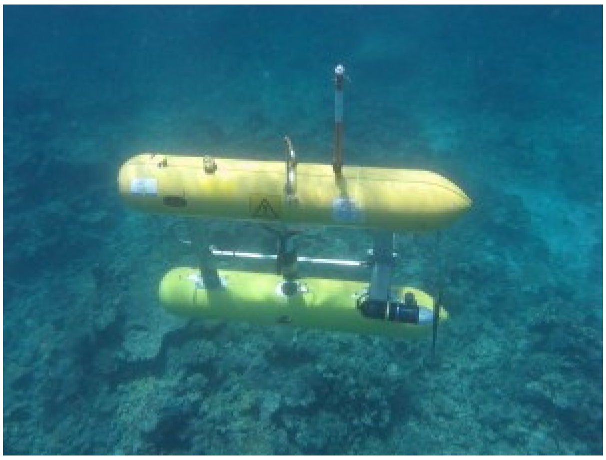


Figure 4: AUV Sirius [3]

1.3/ Inter-Group Communication

The laboratory allocated a weekly seminar entitled The ACFR's Seminar. This seminar is a weekly presentation of a project done by an ACFR researcher or a private company. These seminars are an opportunity for researchers to gain knowledge about other groups' research and to practice presenting their own. Moreover, it allowed us to discover some outstanding and inspiring projects. I was fully inspired by Dr. Oluwarotimi Williams Samuel's presentation entitled «*Towards Intuitively Robust Control Schemes for Assistive and Rehabilitation Devices*.» [9]. It inspired me because it showcased how artificial intelligence can be used to provide new solutions for disabled people and showed how one's work can positively affect the world.

1.4/ Organisation and People

- ***Academic Staff***
  - Professor *Ian Manchester*
  - Professor *Salah Sukkarieh*
  - Dr. *Donald Dansereau*
  - Dr. *Graham Brooker*
  - Dr. *Stewart Worall*
  - Professor *Stefan Williams*
  - Associate Professor *Guodong Shi*
  - Dr. *Viorela Ila*
  - Dr. *Mitch Bryson*
  - Dr. *Andrew Hill*
- ***Administration***
  - Mrs. *Annette Karydis*, Administration Manager
  - Mrs. *Liz Dimond*, Administration Officer
  - Mr. *Kieran Parker*, Business Development Manager

B.2/ Thesis/Subject Presentation

Firstly, this internship was carried out as a contribution to Nicholas Barbara's thesis. Given that, a detailed presentation of the thesis is needed to understand my contribution. His research provides an answer to the question,

"Can we learn feedback controllers with robust stability guarantees for complex nonlinear systems?"

The non-linear systems that we are focusing on in this study are bipedal robots. Before detailing the robot we work on, we will enumerate aspects that make such robots difficult to control. First and foremost, bipedal robots are unstable and very hard to model. The difficulty of modeling such systems will be one of the key challenges that I faced in my contribution. Secondly, for behaviors such as walking, these robots are harder to control due to the environment and its irregularities.

The bipedal robot we work on is the **Cassie robot**. It was introduced by *Oregon State University* in 2017. Moreover, it was developed by the Agility Robotics company under the direction of *Professor Jonathan Hurst*. The Cassie robot is the Guinness World Record holder for the fastest bipedal robot in 100 meters. This record was done on a running track, so there were no instability factors brought on by the environment.

Figure 5: ACFR's Cassie robot

That is where this thesis can change things. We will detail **The Youla-Ren Learning**, which is the model that will guarantee the *stability* and *robustness* we are looking for. Having robustness will grant the model better resistance against adversarial attacks and, therefore, improve its performance.

There are several ways to enforce stability. They are based on controllers' theory, and they are effective for robustness and stability purposes, but all the methods present some tradeoffs. We can illustrate this by citing *the projection method [4]*:

We have a defined set of stabilising controllers. Defining such a set is very difficult and problem-dependent. During each iteration, we realise the projection of the control policy to enforce, and so guarantee, stability. Even though this method functions, it has a learning speed tradeoff that can't be scaled for complex systems like ours.

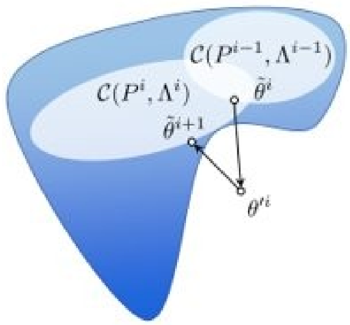
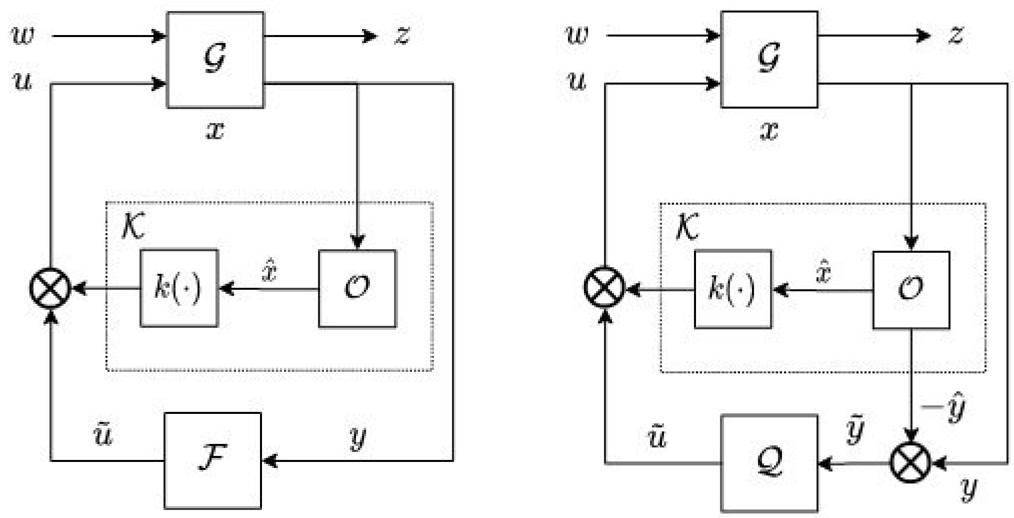


Figure 6: Projection Method

Unlike the projection method, **the Youla-Ren learning method [5]** will be scalable for complex non-linear systems. This method is based on the **Youla-Kucera parameterisation**. The seventh figure details the difference between a system with regular control feedback and a system using the Youla Parameterisation. Before going deeper into the non-linear case, we will present the linear one to understand the system's operation.



Controller Augmentation — The Youla Parametrisation

Figure 7: The Youla-Kucera Parametrisation

These two **linear** systems are defined with: u the input of the environment, y the output of the environment, *G* the environment we are interacting with, x the current state, K the stabilising base controller, ũ the prediction of the next input, and O an observer.

The key difference between these two is the usage of **ỹ** instead of y. Therefore, **ỹ** defines the Prediction Error / Innovation of the system, and it is what is provided to the learnable/optimisable model Q. With this implementation, the system is in a closed loop with Q to optimise it. In fact, we learn over the space of all linear stabilising controllers for a given linear system.

Nonetheless, we stated before that this could be scaled to a more complex non-linear system. Consequently, here are the assumptions taken to scale it:

- Contracting and Lipschitz closed-loop system under K
- Contracting Observer O
- Lipschitz Observer System
- Observer correctness
- Contracting and Lipschitz Q

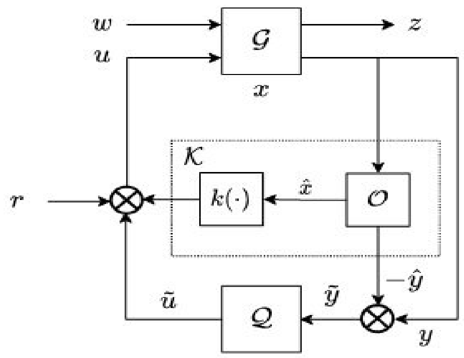


Figure 8: Non-Linear Youla

The contraction and Lipschitz bounding are mandatory to enforce stability during the learning process. Working with contracting systems means that the system will forget the initial conditions over time. Given two different initial conditions, the contraction can be represented by the following figure:

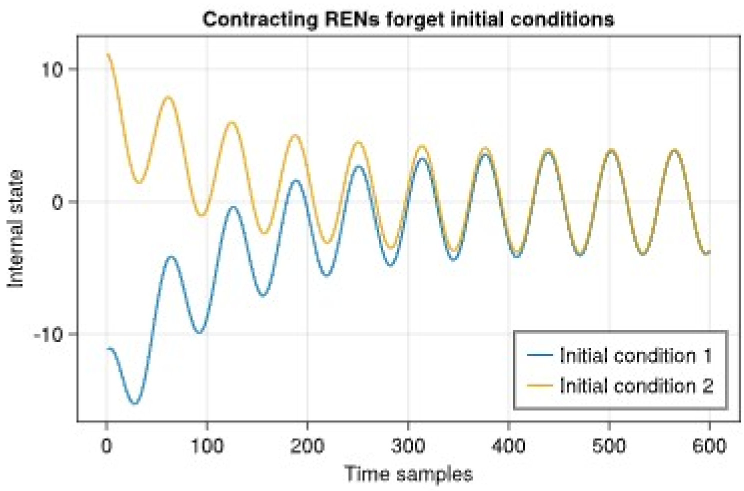


Figure 9: The contracting system forgets initial conditions

There are a couple of additional robustness criteria that are taken into account during the learning process. They are the **Integral Quadratic Constraints (IQC) [6]** and **the Lipschitz Bounds Smoothness [7]**. For additional information on these theories, refer to the associated resources provided.

B.3/ Contribution

3.1/ Introduction & State of the Art

The Youla-Ren method doesn't have a lot of resources since it is fairly new. To carry out his thesis, Nicholas Barbara has developed a library in the Julia language entitled **RobustNeuralNetworks.jl [8]**. This package is the application of the Youla-Ren Learning. Beforehand, he collected some "Preliminary Numerical results" to confirm the effectiveness of the method. In fact, he measured the robustness of the method for a pole stabiliser. The training process was done through Augmented Random Search (ARS). The performance was on the same order as existing methods such as Feedback-LSTM. Nonetheless, the tenth figure illustrates the leap in effectiveness for built-in robustness compared with former methods:

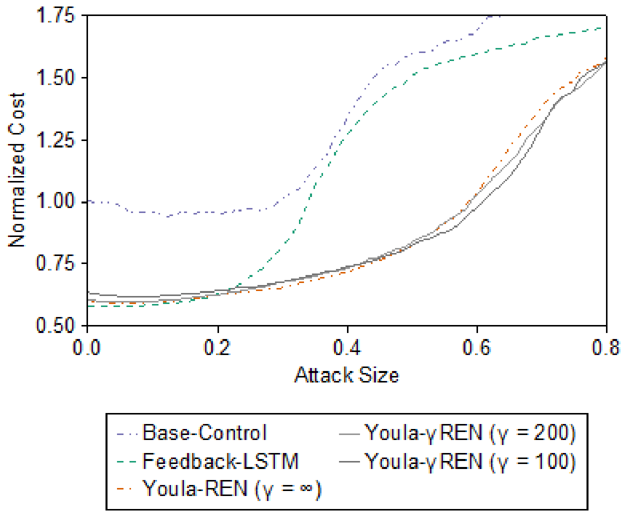


Figure 10: Built-in Robustness

The work he did for his library was so well executed that it earned him an invitation to the annual Julia language conference (JuliaCon) at MIT. Now, his future work is going to be on the application of the Youla-Ren method to more complex non-linear systems, such as **Cassie**. Consequently, being a much more complex and expensive device, a proper simulation is needed to conduct experiments. **The purpose of this internship was to help him build an effective and precise simulation of Cassie to realise robust reinforcement learning.** Precision is required for such systems in order to accurately emulate their physical aspects. Furthermore, for the simulation, we were using **Mujoco**. Mujoco allows «advanced physics simulations». This physics engine allows us to simulate complex controllers while having an accurate physical representation of our system/environment.

Besides, we had several possibilities to realise the reinforcement learning on Cassie. First and foremost, we wanted to build the Cassie environment with **GYM**. Gym is an open-source library created by OpenAI that allows us to create custom environments and apply reinforcement learning algorithms. Gym being one of the standards for designing custom environments for reinforcement learning, there are already groups that have developed several resources on Cassie using Gym. Most of these groups used the Cassie model provided on Oregon State University's Github to use a reliable model. Although most of these resources were developed in Python, we wanted to use Julia as much as possible. The main reason is to reduce embedding between Python and Julia. However, since Julia is a fairly new language, it doesn't benefit from all the well-developed libraries existing in Python.

3.2/ Project management methodology

3.2.1/ Control Group's Management

Firstly, we will be detailing the project management methodology of my group, the control group. Each week, we had a meeting with the whole group. These meetings were at 3 p.m. every Thursday. In most cases, these meetings were held to hear from two members of the group about their research. They might discuss a recent discovery and how it would affect their research or the study of a new paper that appears to be useful to them. Therefore, it was a great opportunity to understand the direction taken by members of the group or to discover methods that we may use in the future. Another way to discover new methods was when we received foreign researchers who came to present their research. Furthermore, the meetings were used for preparing presentations. For instance, we held a meeting to help my tutor, Nicholas Barbara, prepare for JuliaCon. Also, it helps the group's members prepare for their thesis defence and the *ACFR Seminars*.

Secondly, within the control group, we have meetings for each team. These gatherings serve as an evolution tracker for each project. For instance, Johnny Cheng's team is meeting with the medical staff of the hospital where the *KUKA robot* is. Our group is also organizing such gatherings, one every Monday to track the evolution of Nicholas Barbara's Julia package, **RobustNeuralNetworks.jl [8]**. We either discuss the Github issues that we need to take care of or the various new usages we have done with the package. Another one is held to track the evolution of the work with the Cassie robot. We are not only tracking how Cassie's software is being maintained and improved but also the inquiries from the group that wants to use it in the future.

3.2.2/ Personal Management Methods

Firstly, the methodology that I used was different for the different tasks I was given. Nonetheless, all the tasks required some learning to master each area. For this process, I dedicated sessions to learning what was needed to provide for every domain. For instance, we had to restructure the software to manipulate the Cassie Robot. Thus, I needed to learn **ROS** (Robot Operating System), which is the distribution used to interact with robotic systems. To gain efficacy for this process, I asked at the very beginning what needed to be learned to understand our platform. This prior work defined my objectives and how I should segment the learning process. Then, I used a scheduling application online to order my schedule during the week to be as efficient as possible and to focus on what was mandatory for me. The scheduling application used is called **Clockify**. The eleventh figure is an example of two days for me.

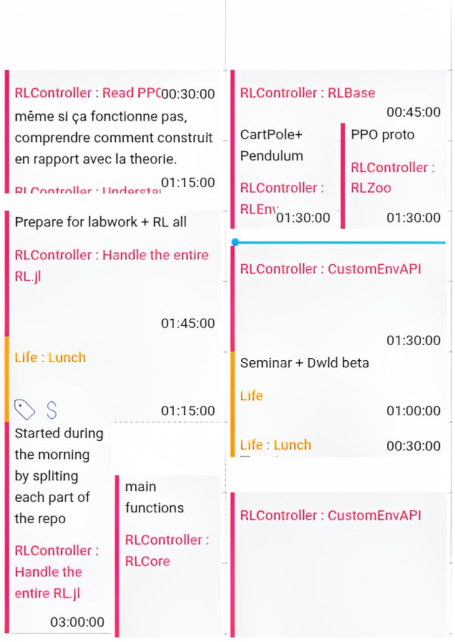


Figure 11: Clockify Usage Example

While learning or carrying out a request, I could always ask questions to my tutor, Nicholas Barbara, or Jesse Morris, who are the collegues with whom I worked the most. Moreover, we would meet quite often to either test some of the new code or measure how things were going. It would help me not to lose myself in a single task that would have been secondary compared with others.

Thus, when I was researching frameworks and methods that we could use, I was working in *sprints*. In fact, I would allocate a few days to each method to determine how useful it was and if it was worth continuing to investigate. This return on investment was one of the topics discussed during our meetings.

3.3/ Work carried out

3.3.2/ Rebuilding Cassie's software structure

The control team that I was part of is using the Cassie robot as a platform to perform tests on the different controllers the team is developing. My tutor, Nicholas Barbara, is in charge of the robot. His research will be applied to the robot. Consequently, the software used to manipulate Cassie needs to be user-friendly. We wanted it to be user-friendly because then the researchers could focus only on their work with their controllers and not on the robot's behavior. The first version of this code had other issues. The coupling between the submodules of the system and between *ROS* and *non-ROS* files was too high. Moreover, the former software was difficult to expand and test. That is why Jesse Morris decided to write the entire structure again. We discussed our different views regarding the new structure, but my knowledge of *ROS* was insufficient to grasp the way to go.

To make up for this insufficiency, I dedicated time to clearly understand *ROS* and the plugin we were using to design the new structure. I'll briefly detail the key *ROS* notions that we used in our code. *ROS* is a middleware that can be used in either Python or C++. Also, ROS communicates via *nodes (files using ROS)* in a distributed system. In fact, *ROS* is a standard in robotic software systems. There are several ways to communicate data between *nodes* in *ROS.* The most frequently used are *ROS publishers and subscribers* *(described in the following figure)*. We create a publisher node that will publish on a topic. Every time new data is provided on this topic, all of the subscribing nodes to that topic will receive the data and process it using their callback function.

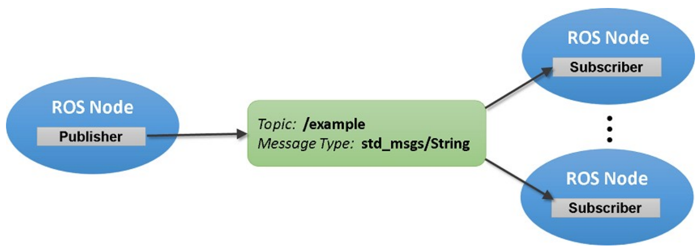


Figure 12: Exchange Data with ROS Publishers and Subscribers [10]

The plugin we are using is for registering the controllers used by Cassie. This plugin is especially useful when we want to use controllers with non-torque outputs. However, Cassie can only interpret torque, thus, the plugin will register another controller to make the torque conversion. Considering all these things including, the testability and safety of the structure, we finalized the design and code. The following figure displays a simplified version of this new structure.

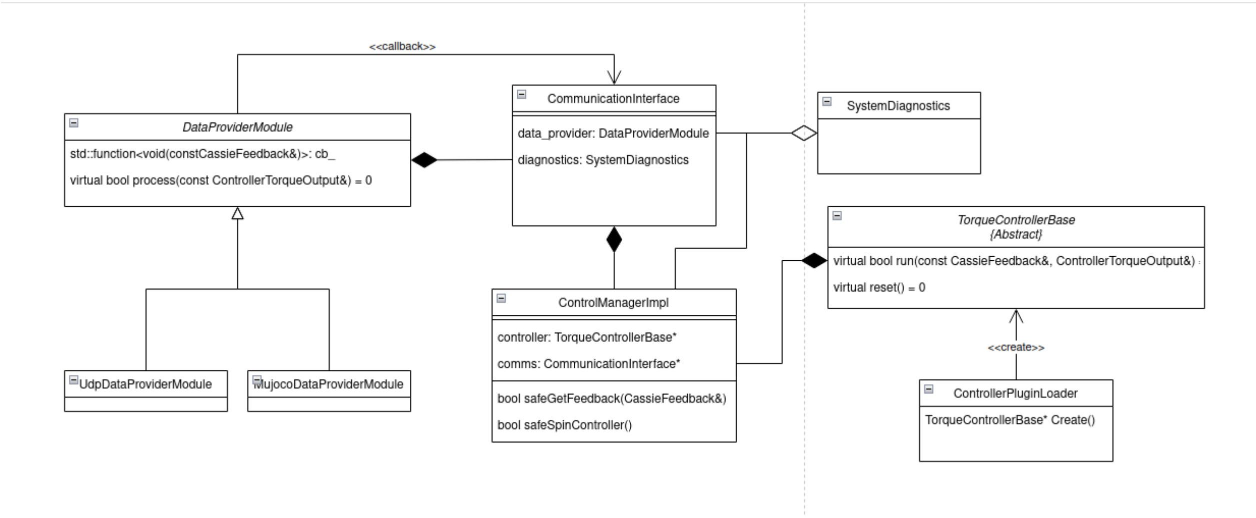


Figure 13: Simplified New Structure

This new structure is now acting as a sort of black box for other users, which was the challenge we wanted to take up. Most of the *ROS* nodes are now separated from the regular files. We used a lot of callback functions for this design to imitate the behavior of an observer. The other design pattern we used is factories. We wanted to implement more design patterns, but we focused on a rather simple version and maintained the regular way to implement ROS communication.

After being designed, we added some tests using Google's frameworks, ***GMOCKS*** and ***GTESTS*.** Mocking the newly implemented classes helped us test the newly implemented safety features, such as *ESTOP*. ESTOP prevents the controllers from reaching the robot while still publishing data. Nonetheless, we can still analyse the data provided. When the testing phase was done, we used it on the robot with a simple controller, and here are the results:

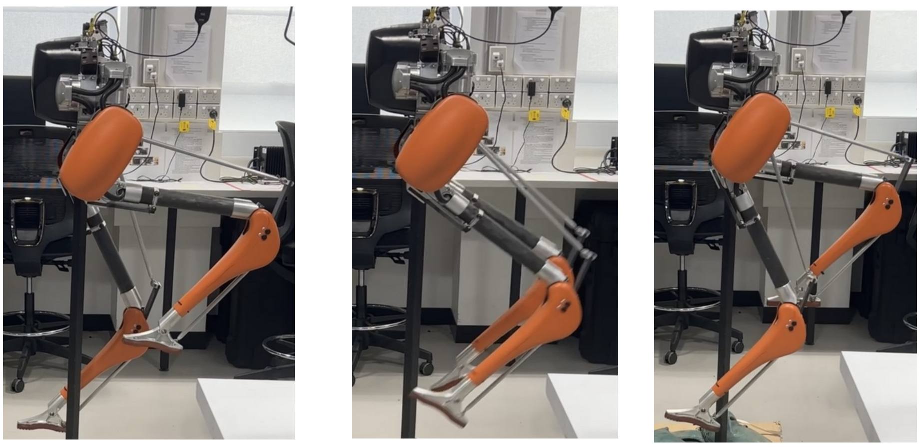


Figure 14: Pd controller test on Cassie

3.3.2/ Reinforcement Learning on a Cassie Simulator

Not only did I work on the real robot's software but also on designing an accurate Cassie simulator compatible with robust reinforcement learning to help my tutor, Nicholas Barbara, with his thesis. First of all, we were focusing on the resources offered by Julia since my tutor's library is in Julia. The reinforcement learning library in Julia [11] underwent changes during my internship. So, for the time being, I dedicated time to understanding how to customise an environment in Gym. We used Gym because they already have built-in wrappers for environments that use various simulators, such as *Mujoco*. The steps to create one are pretty straightforward. First, we create a class that inherits from GymEnv; then we define the observation space and the initial state in an init function, a step function, and a render function to show the simulation. Obviously, the learning algorithm chosen for the training may vary depending on our environment's distinctiveness. The custom environment for Cassie has a continuous action space and is complex in order to be physically correct.

The algorithm we use to suit this kind of environment is the **PPO** (Proximal Policy Optimisation) for continuous action spaces. We'll briefly explain its behavior and why it is the current standard for such training. This method was introduced by OpenAI in 2017 and was inspired by a former method that had the same purpose, *TRPO* [12]. For our case, the PPO would be used, unlike DQN, as an online method that would optimise the policy of our system to execute a controller's behavior. The PPO method uses an Actor-Critic Style process for its loss function. This loss function is defined by the equation below:

$$L^{PPO}(\theta) = \hat{E}_t[L^{CLIP}(\theta) - c_1 L_t^{VF}(\theta) + c_2 S[\pi_\theta](s_t)]$$

Figure 15: PPO's Loss Function

The legacy of the TRPO lies in the L_{CLIP}. This part is defined as taking actions that favor positive rewards using the baseline of the former timesteps. The clipped aspect is to maintain stability during the training process and not make too many drastic changes to the policy. This method is effective and robust.

We applied this method to the most common Gym environment, which is *CartPole*. Everything went well, but we faced some issues with scaling it up to more complex systems. My tutor kept looking for solutions in Julia while I started to use Python. Using Python was not an issue for us since it would be easy to embed it using the Julia library PyCall.jl. We started by using the most commonly used libraries such as *TensorFlow* and *Pytorch* to implement the PPO. We faced issues with TensorFlow because the reinforcement learning is not being maintained anymore. We tried to resolve the source code by ourselves, but it was not worth it. That is when we found **CleanRL**, which is what we are going to use. This library was developed by Costa Huang. The library is presents all the aspects we were looking for: Mujoco compatibility, PPO with continuous action space, and being maintained. Moreover, the code was trackable using ***Weight and Biases*** which is not only useful but does not exist in Julia. At that point, I trained a humanoid system with a continuous action space using Mujoco as a simulator. We train the model for one day (15 million iterations) to try to make it stand.

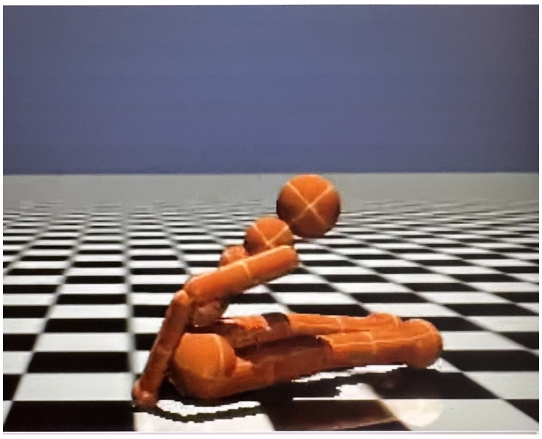


Figure 16: Training on the Humanoid Model

Afterward, we looked for the Cassie models that we were going to use to apply. My tutor and I found two Github repositories that we could use. Both of them were compatible with Mujoco-py, which is the legacy version before it was open-sourced and the version we use with **ClearnRL**. [13] [14]

B.4/ Conclusion & Future Work

Throughout this internship, I have dealt with many aspects of engineering. Starting with adapting to the environment and learning what I was lacking to adapt, we designed an object-oriented structure using our knowledge of design patterns and our experience. Also, we pursue the research currently being done by helping with certain aspects that require additional work. In fact, we dedicated time to finding the appropriate resources to obtain the most accurate simulator for Cassie. This work will save time for my tutor because it will make the transition between the simulation and the real robot easier.

The next step for this work is to design it as an API. To be more specific, we want it to be user-friendly and act as a black box, as is the case with the new structure. To do so, we would need to create a GitHub repository that uses CleanRL and the simulator we chose. Moreover, using the package ArgParse, we'll make the controller we're using tunable. Thus, the final step would be to use this API with the new software structure and to be able to select beforehand if we want it to be simulated or applied to the robot.

APPENDICES & BIBLIOGRAPHY

- **[1] Figure 1:** Blain, L. (2019) «SwagBot autonomous agricultural robot to hit the farmers market in 2020», https://modernfarmer.com/2016/07/swagbot/, July 2023.

- **[2] :** ACFR Marine, (2014) «ACFR Marine Robotics Program», https://marine.acfr.usyd.edu.au/ ,July 2023.

- **[3] Figure 4:** ACFR Marine, (2014) «ACFR Marine Systems», https://marine.acfr.usyd.edu.au/systems/, July 2023.

- **[4] :** Neelay Junnarkar, (2008) «Synthesis of Stabilising Recurrent Equilibrium Network Controllers»,https://ieeexplore.ieee.org/abstract/document/9992684/authors , June 2023.

- **[5] :** Ruigang Wang & Ian Manchester, (2021) «Youla-REN: Learning Nonlinear Feedback Policies with Robust Stability Guarantees», https://arxiv.org/abs/2112.01253, May 2023.

- **[6] :** Peter Seiler, (2020) «Stability Analysis With Dissipation Inequalities and Integral Quadratic Constraints», https://ieeexplore.ieee.org/abstract/document/6915700 , June 2023.

- **[7] :** Wikipedia, « Lipschitz Continuity»,

https://en.wikipedia.org/wiki/Lipschitz_continuity, June 2023.

- **[8] :** Nicholas Barbara, (2022) «RobustNeuralNetwork.jl», https://acfr.github.io/RobustNeuralNetworks.jl/dev/introduction/package_overview/, May 2023.

- **[9] :** Oluwarotimi Williams Samuel, (2016) «Intellingent EMG Pattern Recognition Control Method for Upper-Limb Multifunctional Prostheses : Advances, Current Challenges, and Future Prospects», https://ieeexplore.ieee.org/abstract/document/8612917 , July 2023.

- **[10] :** MathWorks, «Exchange Data with ROS Publishers and Subscribers», https://www.mathworks.com/help/ros/ug/exchange-data-with-ros-publishers-and-subscribers.html , May 2023.

- **[11] :** Johanni Brea Jun Tian, «ReinforcementLearning.jl», https://juliareinforcementlearning.org/, June 2023.

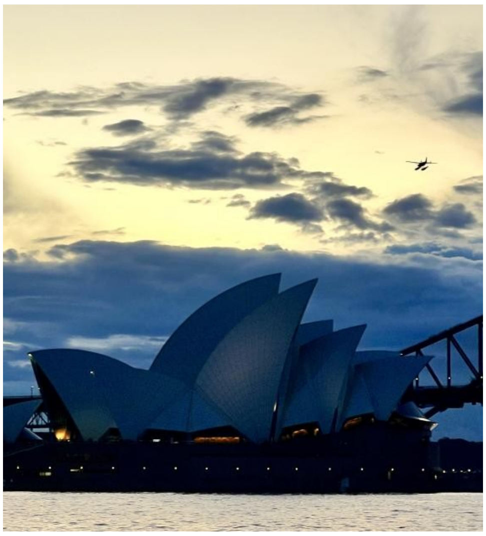
- **[12] :** OpenAI, (2016) «Trust Region Policy Optimisation», https://spinningup.openai.com/en/latest/algorithms/trpo.html#background , June 2023.

- **[13] :** Jonah Siekman, (2020), «An OpenAI Gym-like environment for simulating Agility Robotics' Bipedal Cassie Robot.»,https://github.com/siekmanj/cassie , July 2023.

- **[14] :** Feidedao, (2023), «A cassie walking simulation test example based on cassie-mujoco-sim with reference to gym-cassie.»,https://github.com/feidedao/cassie-sim-RL, July 2023.

SUMMARY

This first internship was conducted at the Australian Field for Robotics (ACFR). It was a good opportunity for me to discover reinforcement learning in robotics and also to witness the work realised by PhD students. Throughout this internship, I worked for the control group led by my supervisor, Ian Manchester. To be more specific, within the group, I worked for the team led by my tutor, Nicholas Barbara. The two main focuses of this internship were designing a new structure for our robot's software and providing new tools to improve the simulation of our robot: **the Cassie robot**. Cassie is a platform used to do research on various controllers. Being a platform that will be used by others, we changed the structure to make sure that it would act as a black box. In addition, worked with methods that use deep reinforcement learning to obtain a stable and robust controller. This stability is brought by a theoretical concept called the **Youla-Kucera parameterisation** which is used for my tutor's thesis. Thus, the simulator tools needed to be compatible with reinforcement learning. Apart from the great opportunity this internship was to reinforce my skills (Julia, Python, reinforcement Learning, C++), it was also a great social opportunity. I have met wonderful people with whom I have shared wonderful moments. Moreover, living in Sydney was amazing, and it would be a pleasure to do my next internship in this marvelous city.



Ecole Publique d’Ingénieurs en 3 ans

6 boulevard Maréchal Juin, CS 45053
14050 CAEN cedex 04