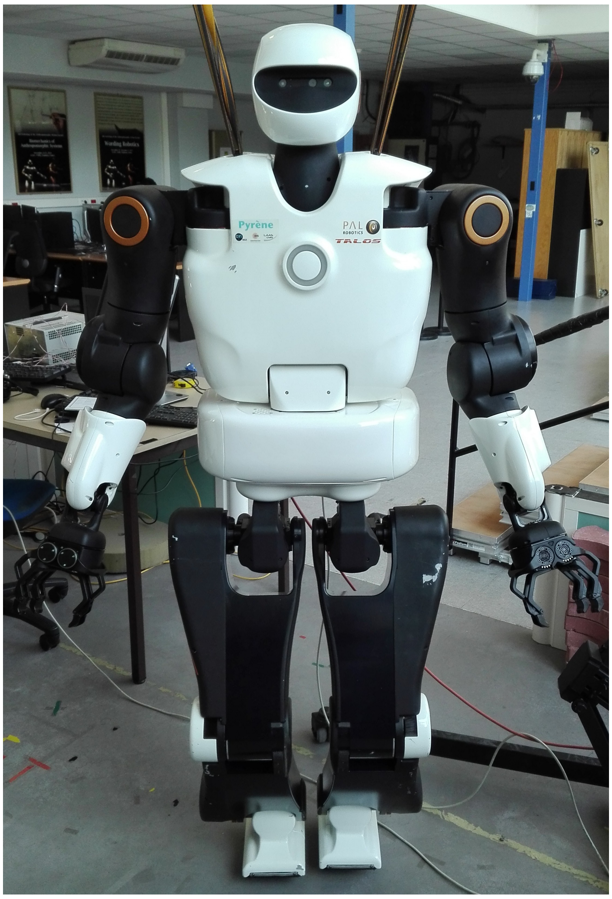
Internship report: Study of the temperature evolution in the humanoid robot Pyrène's motors

Florian VALETTE, Tutor: Olivier STASSE

August 28, 2018



1 Introduction

This internship comes as part of my second year in engineering school. The purpose is to discover what is waiting for us after the diploma, and gain as much experience as possible. I decided to make it in a lab, because I wasn't sure if I wanted to make a thesis: I wanted to discover the research environment, to get an idea if this would meet my expectations or not. It is supposed to be at least six weeks, but I chose to make it three months as I thought this would allow me to learn more things. In my opinion, six weeks were too short to get familiar with the environment and be able to do some actual work.

As I was interested in robotics and wanted to know more about it, I started my internship in the Gepetto team, at LAAS, with Olivier STASSE. With their expertise in humanoid robotics, this certainly was one of the best places to learn more about it. This came as a great opportunity for me to talk with experts in this domain and work with advanced technology.

During this internship, my main work was to study the temperature evolution in the humanoid robot Pyrène's motors. But the biggest part of it was to apprehend what humanoid robotic really is, and discover the tools that would allow me to do my job.

In this report, I first describe the lab, then talk about the working environment and finally detail my work and its purpose. The motors' data are confidential and hence won't be disclosed in this report.

2 The Lab

The LAAS (Laboratoire d'Analyse et d'Architecture des Systèmes) is a part of the CNRS (Centre National de la Recherche Scientifique), one of the world leading scientific research organisms. With its 700 employees and its 26 research teams, the LAAS is a lab at the cutting edge of technology. Gepetto [8], the team in which I have made my internship and led by Philippe SOUÈRES, focuses its work on motion generation for anthropomorphic systems. They develop their own software for optimisation or motion planning and control, among other things. There are two different types of movements generation. The first one, playback motion generation, is a generation in advance, then simulated and finally, if it seems acceptable, played on the robot. Most of the movements are generated this way. The second one, online motion generation, is a generation directly on the robot while moving and immediately played, allowing it to make real time decisions. It is the one we eventually want to master.

The group is in possession of different platforms for experimentation. One of them is HRP-2, a 15 year old Japanese humanoid robot, and the only one of its series that has left Japan. This is the one that has mostly been used for experiments until now. More recently, a collaboration with the Spanish company PAL-ROBOTICS [1] has led to the creation of Pyrène, first robot of the Talos [10] series. This robot, which has benefited from scientific advancements, is equipped with numerous sensors that weren't on HRP-2, like torque and temperature sensors. It also possesses much more efficient motors which allow it to have a wider range of movements.

As one of the world leaders in this domain, Gepetto is a team that attracts industrials, like Airbus with whom a partnership has been established. One of the ongoing work for a practical application is the control in torque of Pyrène, as we want it to be able to drill holes in the air-plane's hull. This places the group at the front-line of technological advancement, developing what may very well be the future of our world.

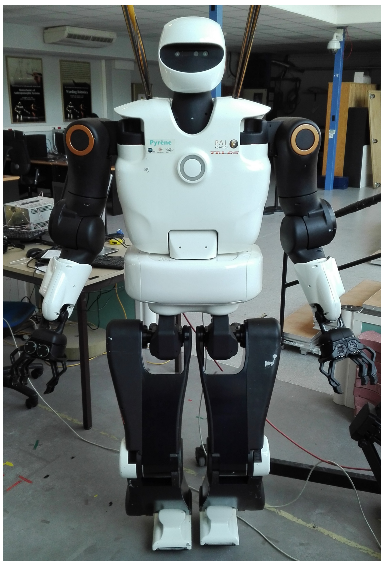


Figure 1: Pyrène

Figure 2: HRP-2

3 Discovering the working environment

One of the most important parts of this internship has been to get to know the different software that were created and are used by the team. Indeed, those result from a work of a decade or more for some of them, and are in constant evolution, also meaning that they are really complex. The first challenge to overcome was to install everything I would need for my work. Being relatively new to the Linux operating system, I faced some issues I did not know how to solve, and it took me quite a while to get everything working. Some of the software I have used the most during this internship are the following:

- pinocchio [6]: One of the biggest works of the team, pinocchio is a library that allows the simulation of poly-articulated systems using Rigid Body Algorithms. It is particularly made for legged robots but its field of application is not limited only to this. Its main purpose is to determine the efforts on the different joints in a certain position using inverse dynamics. It can also calculate the movements created by the different joint torques using direct dynamics, and has some other functionalities that are less used, like collision detection. It is one of the tools used to generate movements.

- Stack of Tasks (SoT) [9]: Another software developed by the Gepetto team, it offers the possibility to prioritize one task over an other for the robot. For example, the most important task will be the balance task (which means keeping its center of mass in a position where the robot is stable), and only then will come the position control. This implies that the robot won't do a movement if it will make it fall.

- Dynamic-graph : This software allows the other ones (pinocchio, SoT, Talos-torque-control etc.) to interact with each-other. It allows the creation of different blocs, like controllers or estimators, and makes them communicate. It is possible de connect or disconnect the different blocs "online", without stopping the robot.

- sot-torque-control : It is a software using both SoT and Dynamic-graph in order to control the real humanoid robot using joint torques.

- robotpkg [3]: It is a packaging system that offers the possibility to install the different software used in robotics, and makes sure every dependencies required for a package are available. This is what is used to compile all the software (in development or not) on which we work at the lab.

Most of the software that are used are still in development. In order for everyone to always have up-to-date versions we use Git, a version control tool. Even though this has a lot of advantages, it also has the major inconvenient that updates are made a few times a day, and it sometimes creates some issues when someone carelessly changes the name or the content of a function which is used somewhere else. This can sometimes lead to long sessions of debugging (all the longer that I'm not experienced). All the software are available in open-source on Github and Gitlab.

In order for me to get familiar with those software, I spent the first weeks doing tutorials. They are well documented, but as Pyrène is a new platform at the lab, I have had to adapt them. This has been beneficial for me as I was forced to browse through the code and it made me understand different aspects of its architecture, clarifying things.

To understand the complexity of the architecture of the system, here is a simplified graph of sot-sorque-control, which is only a small part of the SoT:

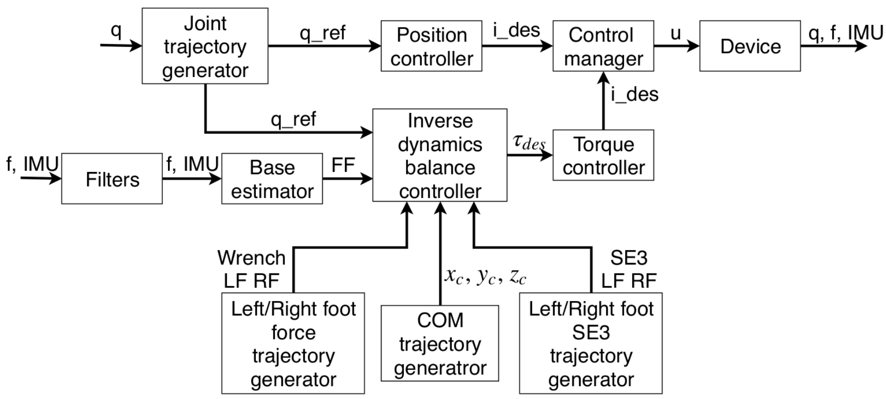


Figure 3: Simplified graph of the sot-torque-control architecture

4 My work in the lab

4.1 The goals

The purpose of my internship at LAAS was to identify the temperature evolution in the motors of Pyrène. This was not the original subject but a few days before my arrival, the robot has suffered a failure due to high temperatures. Olivier STASSE, my tutor, has then decided to have me working on what could be an immediate problem. Indeed, in the lab, things move at an unpredictable pace, so my subject was adapted to the needs of the moment.

The idea was to set up a theoretical model of the temperature that would be used in an actuator's control architecture exploiting a DDP (Differential Dynamic Programming) solver that Florent FORGET, a doctoral student, is working on. The DDP is an algorithm that uses the current state of a motor in order to determine the optimal command, on a limited horizon of time, that will satisfy previously set constraints, as explained in [11]. It uses motor data measured in real time in order to adjust the command at any time. In our case, we use a DDP because the studied models are not too complicated. The DDP, which is rather simple to set up, is then a good compromise. The use of other tools, like neural networks, does not make sense because the time saved does not justify the complexity of the implementation. In order for the DDP to work correctly, it requires precise models of the actual system, which have to be at least once differentiable. A good example of movement generation using a DDP algorithm can be found in [5].

Given the correct models, it would be possible to anticipate the heat evolution and then generate movements that we are sure are not going to overheat the actuators. In the current state of things, there is a safety on the robot that shuts down the motors when the temperature goes over 84°C. This means we lose a lot of the actuators' capabilities, as they are supposed to work correctly up to approximately 140°C. Using the DDP algorithm, it would be possible to use the motors much more efficiently by going to the short-term range of operation (see Figure 4) without risks, and hence we could have the robot make movements that require a lot more power. The goal is to see if the robot is able to lift 40kg using only the elbow motor. If things prove to be working correctly, we will try and adapt the models for all of the robot's joints so we can monitor and anticipate the temperature in the whole robot.

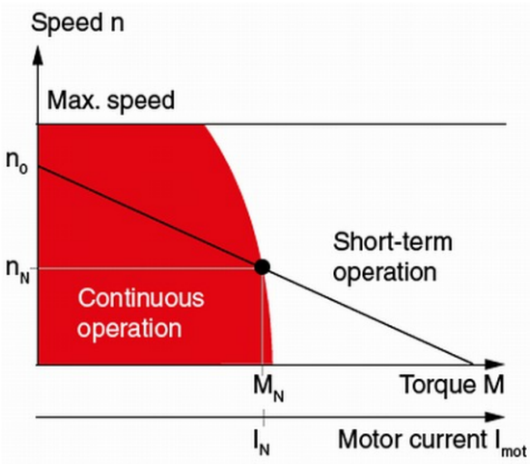


Figure 4: Motor's range of operation (Maxon datasheet [4])

The interest in humanoid robot's joints temperature is fairly new, as before there were no temperature sensors. There has been temperature studies on smaller robots like the climbing robot [12] or the tendon-driven robot Kojiro [13], but the systems are not as complex as Pyrène, and they did not use it to generate movements like we want to. More recently, there was a study on HRP-4 [7], a full size humanoid robot, in which they limit the joint torques in order not to overheat the motors. But in our case, the goal is to directly take into account the temperature while generating motions.

4.2 The theoretical models

When I started my internship, a model had already been quickly done, so I had to make sure it was a correct one. But it was not. Indeed, they had based their work on the Maxon datasheet [4], but this one was a bit ambiguous. This led to an equation that was not homogeneous, so I decided to start from scratch.

We want a model of temperature that is quite simple as we want to have as few calculations as possible when using the DDP. Indeed, when working with robots and real time programming, there is no time to spare.

We have really few informations about the motors from the constructor, and none of them are of any use when working on the temperature. We will have to make a general model and then identify the informations we need thanks to experiments on the robot. But first, in order to describe correctly the motor, we agreed on a dynamic model:

$$B_m\ddot{\theta}_m + D_m\dot{\theta}_m = \tau_m - \tau_l \tag{1}$$

with θ_m the robot position, B_m the rotor inertia, D_m the motor friction which includes friction in the gears, τ_m the motor torque and τ_l the motor load.

Let V_b the back electromotive force (E.M.F.) given by:

$$V_b = K_b\dot{\theta}_m \tag{2}$$

where K_b is the back E.M.F. constant, and R the winding resistor which increases linearly with the temperature:

$$R = R_{TA}(1 + \alpha_{Cu}(T - T_A)) \tag{3}$$

with R_{TA} the armature resistance at ambient temperature and α_{Cu} the temperature coefficient of resistance of copper. We can now write τ_m as:

$$\tau_m = K_m i_a = K_m\frac{V - V_b}{R} \tag{4}$$

with i_a the armature current, V the armature voltage and K_m the motor torque constant.

Now that this is done, we can focus on the temperature. First, we have to determine the causes of the heating, the biggest being the joule power losses:

$$P_J = RI^2 = R_{TA}(1 + \alpha_{Cu}(T - T_A))I^2 \tag{5}$$

We could have considered the friction losses, but as our motor is a DC brushless, we decided that those could be neglected. We also decided to neglect the remagnetization losses and the Eddy current losses which do not really account for much in the heat increase, and make our model a lot more complicated, which we do not want.

We now have to determine the thermal model of the motor. As we do not know in details what kind of DC brushless motor we have, we came up with two different models.

In the first model, we consider that the motor casing has its own thermal resistance and capacity. This leads us to the following equivalent model:

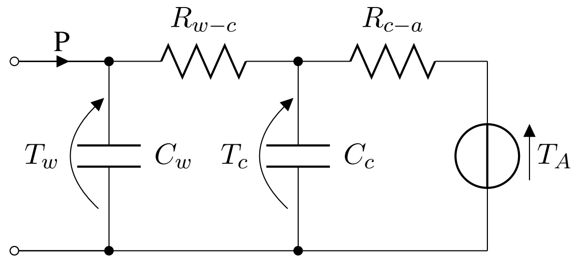


Figure 5: Motor's thermal equivalent circuit considering the casing

This model is quite interesting in our case as we measure the temperature on the outer side of the casing. The equations that come out of this are the following:

$$\begin{cases} C_w \frac{dT_w}{dt} = P - \frac{T_w - T_c}{R_{w-c}} \\ C_c \frac{dT_c}{dt} = \frac{T_w - T_c}{R_{w-c}} + \frac{T_A - T_c}{R_{c-a}} \end{cases} \tag{6}$$

where C_w and C_c are the winding and casing thermal capacities, R_{w-c} and R_{c-a} are the winding to casing and casing to air thermal resistances, T_A the ambient temperature, T_w the winding temperature, T_c the measured temperature (casing) and P the motor's power losses.

This model is close to the one used in [13], except that we do not neglect any term while they consider that C_c and R_{c-a} are considerably bigger than C_w and R_{w-c}, leading to solme simplifications. With this model, we have four constants to identify, which could prove to be a bit complicated and maybe useless if we can neglect the casing part on the actual motor. Which is why we came up with a second model.

In this second model, we consider that the casing conducts the temperature well enough so that the outer side of the casing (where we measure the temperature) is approximately the same temperature as the windings. This allows us to have a very simple model of the motor:

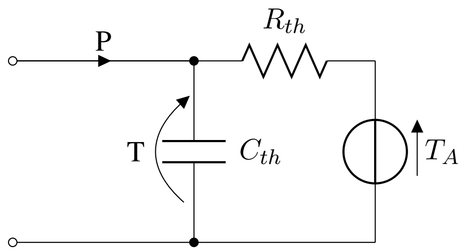


Figure 6: Motor's thermal equivalent circuit 2

We have the corresponding equation:

$$C_{th}\dot{T} = P_J - \frac{1}{R_{th}}(T - T_A) \tag{7}$$

where C_{th} is the thermal capacity, R_{th} is the thermal resistance, T_A the ambient temperature, T the measured temperature and P_J the Joule power losses.

This model is actually the same as in [7] as they also considered the measured temperature to be the same as the winding temperature. We will have to verify if this works correctly to represent the temperature evolution in our motors.

Considering the ambient temperature constant, and using a Laplace transform, we can write it as following:

$$\frac{\Delta T(p)}{I^2(p)} = \frac{\frac{R_{TA}R_{th}}{1-R_{TA}R_{th}\alpha_{Cu}I^2}}{1 + \frac{C_{th}R_{th}}{1-R_{TA}R_{th}\alpha_{Cu}I^2}p} \tag{8}$$

This is quite interesting as when we input a constant current, the system reacts like a first order system which should be quite easy to identify.

It is important to raise that in both cases, the equations are non linear. They also are differentiable, which is a condition that has to be met in order for the DDP algorithm to work correctly.

Having looked at different motors characteristics on the Maxon website [2], we can hope to get a time constant of a few minutes with the nominal input, but this is something we will be able to observe through the experiments. The experiments will also allow us to chose the model that fits the best.

4.3 The experiments

We want to identify the different thermal parameters of the actuators in order to confirm (or not) one of our models. We will work on the elbow actuators, in which high torques appear when lifting heavy objects. The simplest method for this would be to input a constant current and let the motors heat. Two different set-ups were discussed:

- The first idea was to have the robot standing up, and control the forearms in position so they are parallel to the ground. Then, we would block them with a fixed bar, and have the elbow actuators controlled in torque with the desired value. The arms being blocked, we would have a constant current going through the motor and no movement, which is good to identify the parameters, but not a really practical thing to do, especially since the torque control is not totally implemented yet.

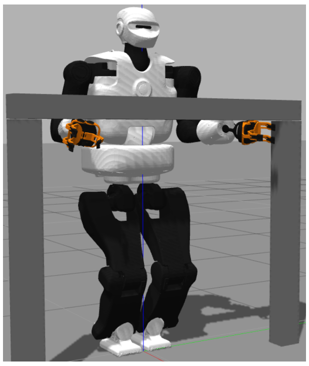


Figure 7: Simulation: experiment with torque control

- The second idea was to control the robot in position like before, but instead of using the torque control, we would just hang a weight to its forearm. The robot being controlled in position, it would try to compensate and hence increase the torque. This would also lead to a constant current input with no movement, and would be much easier to set up. The current depends on the weight we hang and the distance to the joint. It is not easy to have precise values with this, but as we can access current data in real time, we can adjust it by displacing the application point of the weight along the forearm.

Figure 8: Actual setup of the experiment

It is the second setup that we will use. At this point, there is a major problem: we don't know the limits in torque of the harmonic drive (the reducer), so we don't know how much weight we can put without it being too risky. We will therefore be careful not to put weights that could lead to a current higher to the nominal value for the first experiments, as this is the only limit we know.

For the first experiment, we try to be really cautious not to break the robot: we do not go over 3A, that is almost 2 amps under the nominal current, and we put the weight not far from the elbow joint because this is a solid part which does not risk to break. As we are close to the joint, to get to the 3A desired, we add bricks in a bag up to a total of 25kg. The robot does not seem to have any problem, so we wait for the motor to heat. After a certain time, we take the weight off the robot arm and totally deactivate the motors to let the motor cool. We then get a big .bag file which contains all the data we collected. We have to use a python script to put all this data in different text files in order to have only what's interesting for us, and in a format we can use. After that, using Matlab, we can observe the following curve (Figure 9):

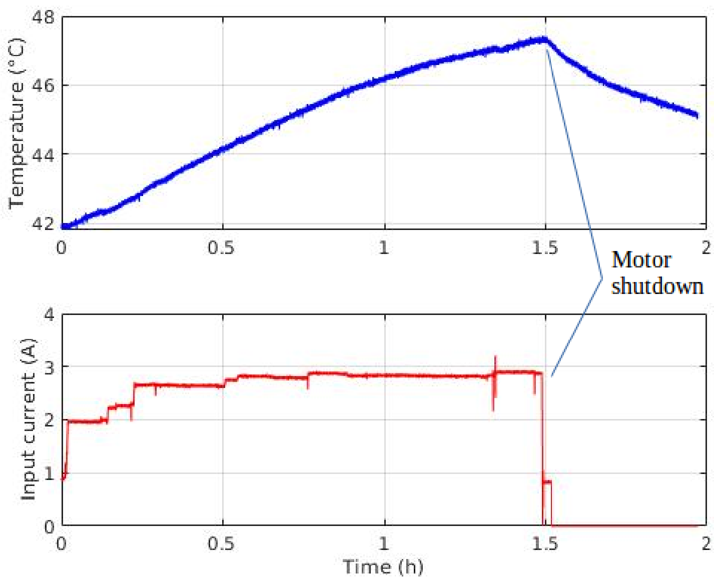


Figure 9: Temperature evolution, 1st experiment

The experiment was not ideal as we slowly added weight, which is why there is not a real step in. It was a first so we kept touching the robot, reason why we can observe spikes on the command. But this was enough for us to realise that the dynamic was way slower than expected. We then decided to make the same experiment once again but with a higher current. For this, we hung 20kg at the robot's wrist, leading to a continuous 4A in the motor. We can observe the curve on Figure 10.

The input is much cleaner on this one, even though there is a secondary step caused by someone bumping into the robot (it caused the bag to slip a bit along the arm, increasing the torque). But we made a mistake: we made the experiment just after the robot was started, but we can clearly see that the room temperature ($\simeq$26°C) is not the same as the ambient temperature for the motor. Indeed, this robot has a slight problem which is that the drives release a lot of heat, heating the motor to approximately 43°C even when not in use. Apart from this, we still have a really slow dynamic which has no interest for us as it can not be taken into account in the DDP algorithm, and we never have movements lasting 3 hours.

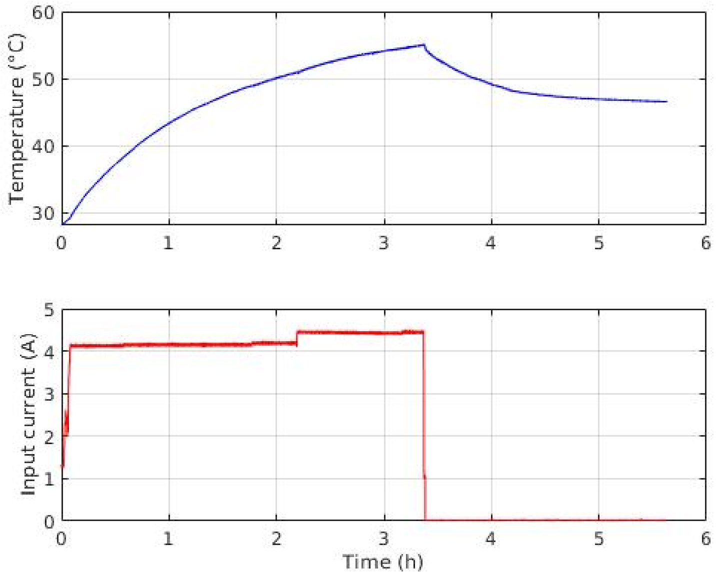


Figure 10: Temperature evolution, 2nd experiment

We will then try a new experiment in order to dissipate more power. This time, we will be careful to let the robot heat before we start. The idea here is to input a sinusoidal command in the elbow actuators. The robot will then make movements like if it was doing biceps body-building. First, we have to create a function to create the corresponding command. We then simulate it using Gazebo to make sure that the robot acts the way we want him to. When this works correctly, we can try on the real robot. But for this experiment, we have to implement the Stack of Task on Pyrène which has not been done before. It did not work at first but after different adjustments made by Olivier STASSE, we are able to proceed with the experiment. We get the following curves:

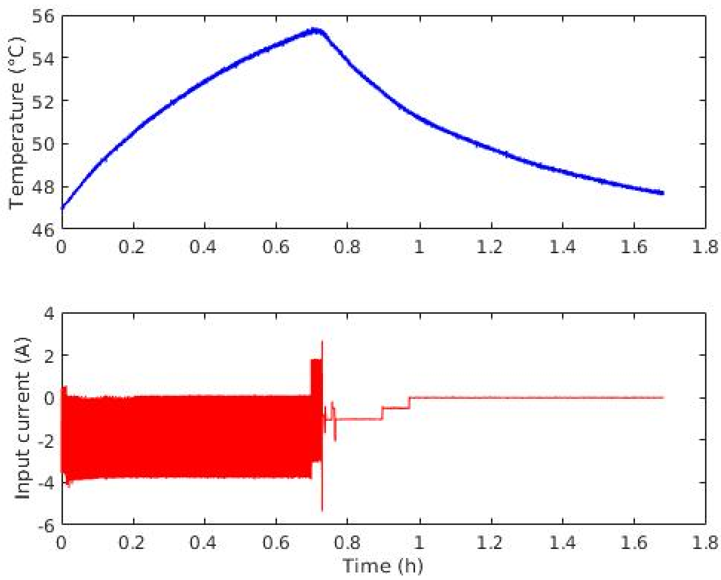


Figure 11: Temperature evolution, 3rd experiment

On the heating part, we have the current oscillating due to the sinusoidal position command. Its RMS value is 2.57A, which is less than the first experiment, but we can observe that it is heating faster. This means that while moving, there is an other heating factor which can not be neglected. It is most likely the frictions that we will probably have to take into account in our model.

As these experiments gave us a lot of data, we take this opportunity to make sure the value given by the constructor for the torque constant is correct. It is important to say that the torque is measured after the harmonic drive. For this we plot the torque according to the current and we identify it to a linear function using a least squares algorithm:

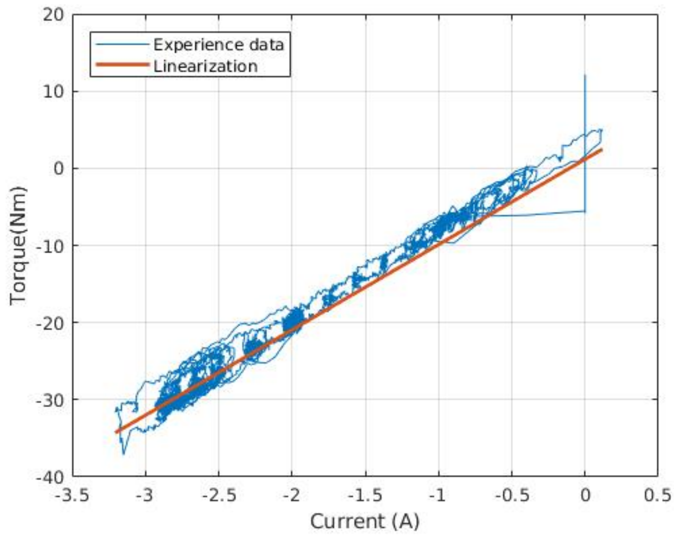


Figure 12: Torque depending on the current

We get the following function:

$$torque = 11.06 * current + 1.17 \tag{9}$$

The offset is due to the different frictions in the arm, but we get a slope very close to the torque constant given by the constructor, which is a very good news. We now proceed with the thermal parameters identification.

4.4 Identification

Now that we have different sets of data, we can try to identify the different parameters of the motor. For this, we use a least square identification algorithm. Given the shape of the curves, we decided to use the simplest model. We have to determine the heat resistor and capacity, and then make sure that with those values, we can have simulations that fits different measured data.

We try the identification algorithm on the first part of the first experiment. We have the following result:

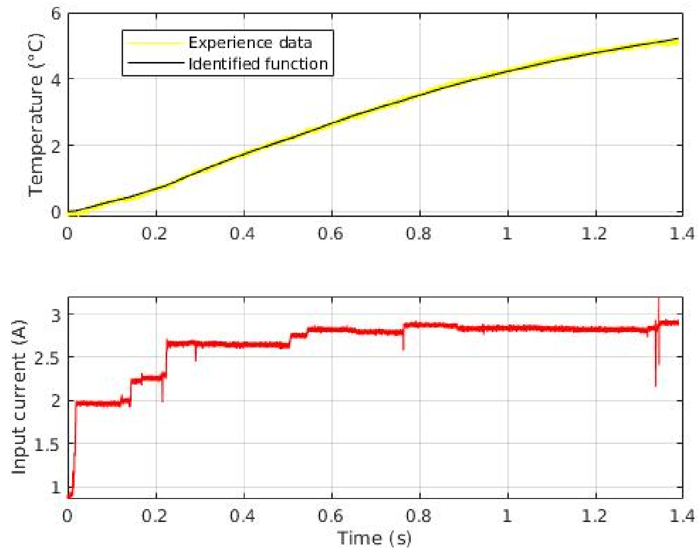


Figure 13: Identification result, 1st experiment

The identification seems to work quite well as the function with identified parameters remains in the range of oscillation due to the sensors of the measurements. The choice of the simplest model seems coherent. We now make the identification on the full set of data. We get the following curves for the second experiment:

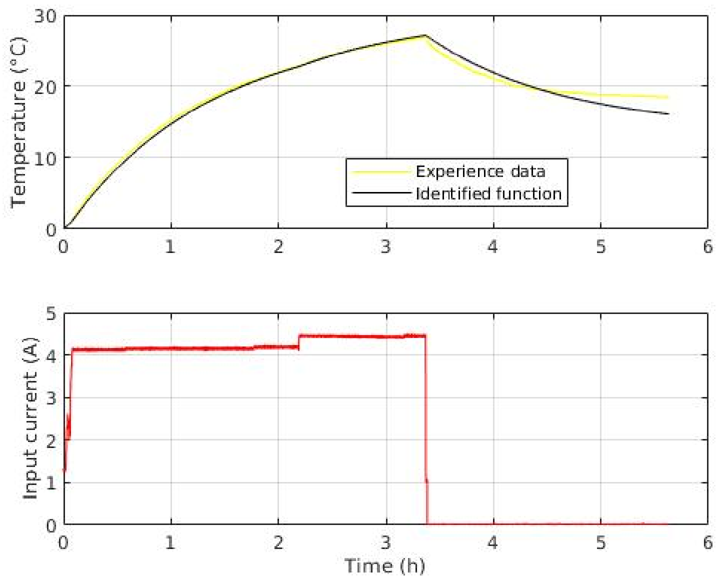


Figure 14: Identification result, 2nd experiment

We can observe that in the heating part, the identified function fits the data relatively well. On the other hand, we can see that in the cooling part, the curves tend to diverge. We think the heat of the drives causes the problem. Indeed, the drives' temperature also increases during the experiment. But the drives and the motor are really close to one another, meaning that the drives' temperature can not be set aside while studying the motors' temperature: it makes the motor cool much slower. This also implies that our hypothesis about a constant ambient temperature is not verified. In order to get a better model, we will have to take the drives' heat into account, which will be a difficult task as we do not know the exact influence it has on the motor. It will also make make the model more complicated.

After studying the results of the third experience, which were not concluding as the model did not fit, we can say different things. The model we came up with is a valid model for a motor with no frictions. But in our case, the frictions can not be neglected. We will have to add them to our model. The other parameter we have to take into account is the drive's temperature. As it is close to the motor in a closed space, it interacts with the motor so it is not in an environment with a constant temperature anymore, the most noticeable effect being the slower cooling of the motor.

Knowing this, we have different options available to us. The first one is saying that we don't need a precise model as the dynamic is so slow. Indeed, with the DDP, we recalculate the evolution every minute so the error would be small. The second option is to try and create a model that is more precise. For this, we will have to do more experiments. An idea to determine the losses due to the frictions would be to make an experiment in movement, determine the RMS current, then make a second experiment with no movement with the current equal to the RMS current of the previous experiment, and finally compare the results. This leaves the problem of the drives heating. For this we don't know the current flowing through, so we could try to guess a model that would be correct enough so we can have a complete modelization of the elbow of our robot.

5 Conclusion

Through the different experiments we made, we realised that the model we came up with was not correct for our robot, even though it works for other teams [7]. It seems the reason lies in the differences in the way the robots were designed. Due to a lack of time, we did not manage to come up with a correct model for Pyrène, but we have ideas about how to improve it.

In the end, this internship went very well. There was a great atmosphere, people were always happy to help or answer questions, and this has made my stay both pleasant and informative. I got to handle cutting edge technology and learn a lot about humanoid robotics, but also, on a larger scale, about how things work in a lab. Even though I'm not sure about making a thesis yet, it has confirmed that this is an option I have to carefully look into in the futur.

Acknowledgments

I would like to thank Olivier STASSE for taking me as an intern during these three months, as well as the rest of the Gepetto team who made my internship a great experience.

References

[1] Pal-robotics, 2018. https://www.pal-robotics.com/en/home/.

[2] maxon motor, 2018. https://www.maxonmotor.com/maxon/view/content/index.

[3] robotpkg, mar 2018. http://robotpkg.openrobots.org/.

[4] Jan Braun. *Formulae Handbook*. maxon motor, 2016. http://maxon.blaetterkatalog.ch/b9991/catalog/index.html?data=b9991/b999145.

[5] Florent Forget, Kevin Giraud-Esclasse, Rodolphe Gelin, Nicolas Mansard, and Olivier Stasse. Implementation, identification and control of an efficient electric actuator for humanoid robots. 2017.

[6] Gepetto. Pinocchio, 2018. https://stack-of-tasks.github.io/pinocchio/.

[7] Iori Kumagai, Shintaro Noda, Shunichi Nozawa, Yohei Kakiuchi, Kei Okada, and Masayuki Inaba. Whole body joint load reduction control for high-load tasks of humanoid robot through adapting joint torque limitation based on onlinejoint temperature estimation. *IEEE-RAS*, page 463, nov 2014.

[8] LAAS-CNRS. Mouvement des systèmes anthropomorphes - gepetto, 2018. https://www.laas.fr/public/fr/gepetto.

[9] N. Mansard, O. Stasse, P. Evrard, and A. Kheddar. A versatile generalized inverted kinematics implementation for collaborative working humanoid robots: The stack of tasks. In *International Conference on Advanced Robotics (ICAR)*, page 119, June 2009. URL `http://hal-lirmm.ccsd.cnrs.fr/file/index/docid/796736/filename/2009_icar_mansard-Stack_of_Tasks.pdf`.

[10] Olivier Stasse, Thomas Flayols, Rohan Budhiraja, Kevin Giraud-Esclasse, Justin Carpentier, Andrea Del Prete, Philippe Souères, Nicolas Mansard, Florent Lamiraux, Jean-Paul Laumond, Luca Marchionni, Hilario Tome, and Francesco Ferro. Talos: A new humanoid research platform targeted for industrial applications. In *IEEE/RAS Int. Conf. on Humanoid Robotics (ICHR)*, 2017.

[11] Yuval Tassa. Theory and implementation of biomimetic motor controllers. *Senate of the Hebrew University of Jerusalem*, feb 2011.

[12] Salomon Trujillo and Mark Cutkosky. Thermally constrained motor operation for a climbing robot. *IEEE*, page 2362, may 2009.

[13] Junichi Urata, Toshinori Hirose, Yuta Namiki, Yuto Nakanishi, Ikuo Mizuuchi, and Masayuki Inaba. Thermal control of electrical motors for high-power humanoid robots. *IEEE/RSJ*, page 2047, sep 2008.