NAELIC - Team Description Paper Small Size League RoboCup 2021 Application of Qualification in Division B

E. Schmitz[2], P. Felix[1], G. Passault[1], L.Verdier[3], J-B.Bonnemaison[7], M.Caillet[8], P-G.Praderas[7], M.Billaud[1], A.Audebert[1], T.Berland[1], B.Bordenave[1], C.Brocaire[1], Y.Chancrin[1], M.Ducamin[1], G.Froidcourt[1], M.Lapeyre[1], G.Le Direach[6], A.Mahieu[1], J.Pestel[1], L.Polla[1], C.Riboulet[1], N.Saifoudine[1], C.Chukry[4] A.Couthures[4], L.Darnaud[4], M.Decultot[4], T.De Faria[4], L.Gazui[4], E.Lascourréges-Berdeü[5], V.Luc[4], H.Mocaër[4], L.Monclus[4], A.Sureau-Samit[4]

[1] IUT - Université de Bordeaux, Bordeaux, France
[2] ENSEIRB-MATMECA, Bordeaux-INP, France
[3] Lycée Saint Cricq, Pau, France
[4] ESME Sudria, Bordeaux, France
[5] ENIB, Bretagne, France
[6] Licence Informatique - Université de Bordeaux, Bordeaux, France
[7] Coh@bit - Université de Bordeaux, Bordeaux, France
[8] TechnoShop - Université de Bordeaux, Bordeaux, France
contact@etienne-schmitz.com (corresponding author)

Abstract. This paper presents a short overview of the design of the robots of the SSL RoboCup NAELIC team.
The team is new, even though there are some experienced people from the NAMeC[2] team, that participated twice in the RoboCup. This is the first attempt of qualification for this new team.
We present the main hardware and software components in their current state and some improvements that we want to make for June.

1 INTRODUCTION

We are a new French team named NAELIC (Nouvelle-Aquitaine Elektrons Libres Innovation Club). It is a partnership between two associations called “Elektrons libres” and “Cohabit”.

Elektrons Libres is an association that brings together young people, parents and trainers. Created in September 2019 in Pau, it aims to facilitate access to science for all young people and to promote their international mobility, while allowing them to participate in competitions and supporting them in their career orientation. The association has been participating for 3 years in the Junior Robocup and each time participates in the international finals.

Cohabit is an association bringing together the fablab and the technoshop of the University of Bordeaux. Thus, Cohabit combines a collaborative space for digital manufacturing through the fablab and a technoshop for the development of innovative technological projects. The association also offers workshops for youth, but also for professionals or individuals on technological concepts.

Furthermore, we are attached to the University of Bordeaux, and it is thanks to a project in our region (Nouvelle Aquitaine), who wants to promote robotics in France, that we are financed.

Our team was born from the meeting between 2 entities: a student belonging to the NAMeC (Nouvelle Aquitaine Mechatronic Club) team - having already participated twice in the SSL Robocup - and a group of motivated students, some of whom have already participated in the Junior Football Robocup in Sydney.
This NAMeC student wanted to pass on his knowledge to other enthusiasts and is now Team Leader of our team.

Our team is not limited to one school or university, all our members come from all over France (although most are studying in the Nouvelle Aquitaine region). Only one restriction: to be passionate, have the will to learn and work hard, in particular in this Covid19 period, which caused us some problems of delay.

Fortunately, since the beginning of the year, we can rely on the NAMeC team, always ready to help us, to give us guidance, advice and who even lent us one of their robots to do some tests.

In addition, a member of the Rhoban team (quadruple world champion in the Humanoid size soccer category), Gregoire Passault, helps us by giving us lessons on the Firmware part.

Our work this year has been to improve the NAMeC robot used at the 2019 RoboCup, in Sydney.

2 Mechanics

2.1 General Presentation

This year, we worked with the French NAMeC team who allowed us to retrieve the mechanical drawings of their robots. They informed us of several important problems detected during their participation in the 2019 Sydney Robocup competition, which allowed us to work on some technical improvements in addition to our appropriation of the robot and its programming within the Robocup competition.

We will see the improvements made by our team divided into modules : Dribbler, Kicker and Wheels, then we will see the fairing, the mainboard, the realization of mathematical models and to conclude, experimental tests.

2.2 Dribbler : Resin Molding

Concerning the dribbler module which allows the robot to hold the ball in place while moving. In order to catch the ball without causing too many vibrations we need a relatively soft part that can withstand the impact as well as cushion it. The technical solution that has been selected for this job is the PU(Polyurethane) resin injection. It is the most suited solution due to the fact that it is easy to change the molds which are 3d printed and allow us to make several tests while being economical at the same time.

In fact with the CAO, we can easily change some options such as the thread, the pitch, stop height,... We have tested several moldings and we chose the best one out of them (see Fig1). In order to choose the best molding, we have established a benchmark with a dremel that can rotate up to 20000 rpm to test with a dribbler speed variation too.

Moreover, we chose to inject a resin called Polyurethane with a hardness of A60 in shore scale which is perfect for our purpose and our use.

Fig. 1. Molding used during Injection

2.3 Kicker : Modifications and Experiments on the shape

The kicker is the robot's module that propels the ball in two different ways: by making a "straight shot" or by "lobbing" thanks to a ramp that lifts the ball. The module is composed of two coils propelling two ferromagnetic parts called plungers thanks to electromagnetic force.

Once the system is triggered, the plungers are brought back to their initial position thanks to a reloading system.

First, we tested different shapes of plungers (bending) to improve the accuracy of the direction of fire, highlighting other possible improvements.

We chose to modify the reloading system. Indeed, this system was previously realized by simple elastic bands that could relax after a few shots. In the same way, the part holding these elastics was printed in 3D and badly fixed on the robot, which led to many breakages of the part, and thus a robot unable to shoot during a match. Our solution is to use springs instead of elastics. (see below)

2.4 Kicker : Reloading of the Kicker module with springs

A spring is an elastic member, usually made of hardened steel, which has the property of returning to its original shape and position after undergoing a deformation, and entering a mechanism to absorb work or to produce a movement.

Here is a diagram showing the behaviour of the spring during a shot: it will relax, then bring the mass representing the plunger back to its equilibrium position in order to start again :

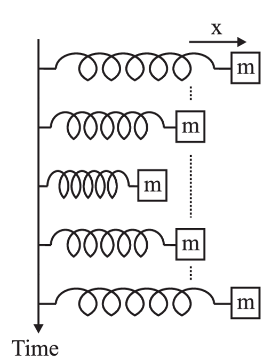

Fig. 2. Diagram of spring

We have chosen experimentally to use a 15mm long spring (initial position), and we are making an approximation on the necessary stiffness of the spring.

In addition, in order to be able to fix the springs, we made modifications on the plungers and created a new CNC-cut workpiece to fix the springs. This last part is now much more solid and efficiently maintains the reset system. We also added a part used as a damper to reduce the vibrations and damage that some parts can suffer from the shock created by a shot. With these improvements, we have a safer and more stable rearming system that should hold over time.

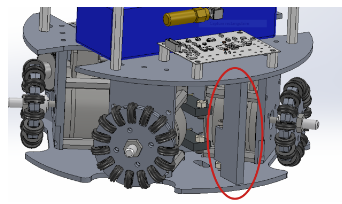

Fig. 3. ISO view of the new piece

2.5 The Wheels

Inspired by the studies of the Manheim Tigers team, we decided to equip the robots with 20 X'rings wheels, instead of 16 O'rings ones. The difference is that the O'rings are round-ended circles, while the X'rings are cross-ended circles.

This increases the number of possible contacts in order to reduce the vibrations that the robot undergoes due to its movement and to improve its adhesion to the ground. These modifications are again to be tested. (see Fig 4)

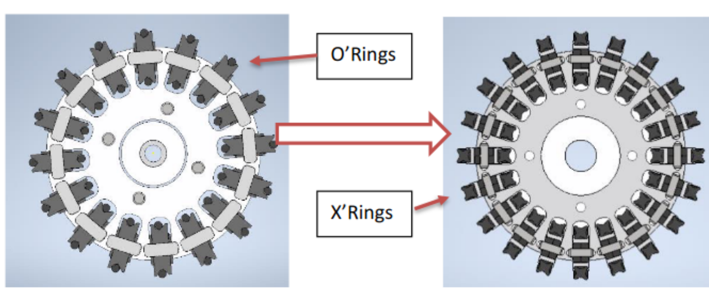

Fig. 4. Change from O'rings to X'rings

2.6 Mainboard modifications

In order to solve the electromagnetic problems present on the NAMeC robot, we grouped the electronic board controlling the dribbler motor and the mainboard in a single mainboard located on the top of the robot.

2.7 Fairing Holding

The previous technical solution selected by the NAMeC team (magnets) was not very efficient. Therefore, we decided to choose a new way to maintain the fairing.

The chosen solution is to use pressure buttons which offers a good mechanical holding as well as a reasonable price. We attached the female parts of the buttons to the maintaining parts of the Capacitor, and the male parts to the fairing.(see Fig 5)

Fig. 5. Pressure Buttons

2.8 Realization of mathematical models and Automatism

Since the beginning of the year, we have been working essentially by experimental approaches. Thus, we are just starting the mathematical and theoretical models of our robot's systems.

We have also started to model the robot's automatism to make it easier for the computer scientists in the software part. As a result, their codes will be more rigorous and they will be able to afford more things.

2.9 Experimental Tests

We are a new team entering the competition and we were delayed by the Covid-19 situation. So, we haven't had time yet to make all the ideas we have been thinking about for the RoboCup competition.

It is the case of the experimental tests. Our work was essentially remote, and the robot tests were postponed. However, the first tests are encouraging and we are more than happy to have our robot finally built in its entirety.

3 Firmware

3.1 Mbed

The electronic board of the robots is based on the architecture of the NAMeC team. The architecture is based on a 32-bit STM32 micro-controller. Originally NAMeC did not use a development library which is quite restrictive. This is why, we want to make a transition to a system using the mbed library which allows us to have an additional layer of abstraction and also to be able to use tools such as platformIO which allows us to manage Threads more finely.

3.2 Trinamic

We also want to use drivers from Trinamic for the motor of each wheel, they will allow us to modulate the speed of the motors more easily, the advantage is that the robot can modulate its speed according to the distance of the objective in a simpler way.

3.3 Dribbler

We have integrated the dribbler driver on the mainboard, which avoids the need for a specific card.

3.4 SPI fix

The NAMeC team, during its participation to the Sydney RoboCup, had communication problems in SPI, so we fixed this bug. The fault was due to electromagnetic disturbances during the loading of the capacitors.

3.5 Micro-controller

We plan to move on the next STM32 micro-controller, next year, named Black-Pill. This move permits to use a FPU to use float in our computation.

4 Software (Nodetron-core)

4.1 Introduction

Being the child of NAMeC, we were inspired by an educational work started by NAMeC to create our software. This work was to transform their SSL sofware (initially in C++) in the programming language TypeScript. So we have continued this work and create our software entirely in TypeScript - named Nodetron.

4.2 Micro-services

We use the architecture called SOA (Service-Oriented Architecture) instead of the monolithic architecture like NAMeC uses in their C++ project.
We have chosen this architecture for serveral reasons. First, it allows a better organisation of the code, with independent service, well-coupled and easy-change services. We can even switch to a different language if Typescript doesn't have sufficient performance.
This organization in services allows :

- Reusability of services : These components can be reused in multiple applications without influencing other services
- Better maintainability : It's easy to update and maintain it without hurting other services
- Higher reliability : Services are easier to debug and test rather than huge chunks of code like in the monolithic approach.
- Parallel development : Independent services can be developed in parallel and completed at the same time.

4.3 Services

Nodetron-core is therefore composed of several services, it contains multiple services to move robots https://github.com/NAELIC/nodetron-core, MSB is outside this core, because we can easily change our decisions systems by Artificial Intelligence for example.

- Network : Receives input data, including the position of the different robots and the ball on the field but also the moment in the match.
- World : Filter the input data of Network to only keep the real information.
- API : Service to connect with other tools such as a viewer, log tools, etc...
- MSB (Manager, Services, Behavior) : Robot decision making service where we develop strategies and manage robot control It's based on the architecture of NAMeC.
- Control : Transforms orders of MSB in understandable instruction for robots (angular speed, normal and tangent, ...).
  Planning, obstacle avoidance and control of the robot is situated here.
- Bots : Send orders of simulation or real robots, read for status packet on robot.

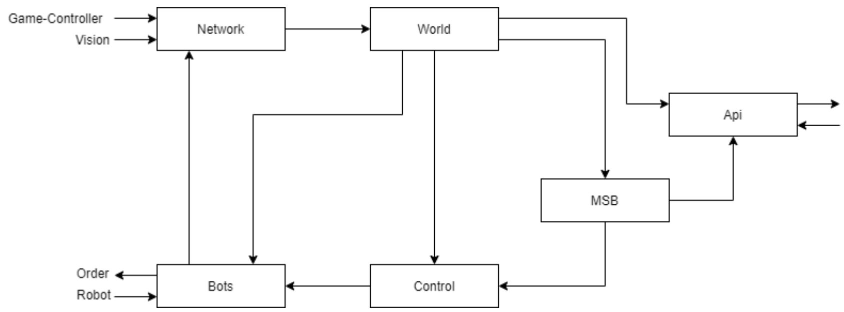

Fig. 6. Architectural diagram

4.4 Improvement, future research

With this architecture, we can search for intelligent multi-agent cooperation, implementing one service that handles one robot. In next years, we plan to move on this field search.

5 Conclusion

Regarding the mechanical part of the robot, we have improved several systems according to the conclusions made by the NAMeC team and we have started to develop the mathematical and kinetic model of the robot for the competition in June.
For this year's Robocup competition, we are only improving some points of the NAMeC 2019 robot in Sydney, but we are already working on a new version that we would like to use in future competitions.

For the software part, we continue a working progress made by some NAMeC's members.
For this next month, we plan to improve the motion control system thanks to a real study of our robot's physics capability. On top of the motion control system, we will introduce global navigation algorithms. Currently, we don't have a good navigation algorithm.
Another plan is to improve our current tools such as our visualization software and log tools.

References

1. Github's team - https://github.com/NAELIC
2. NAMeC Team - https://namec.fr/