

Projet ROBOCUP

Parcours Innovation - S2

GAZUI Laetitia - DECULTOT Martin - MOCAER Hugo - LUC Valentin - MONCLUS Louis

ESME Sudria Bordeaux

2020-2021

TABLE DES MATIERES :

I/- PARTENARIAT ET ORGANISATION	2
A. Partenariat	2
1. Elektrons Libres	2
2. Cohabit	2
B. Qui sommes-nous et quel est notre projet ?.....	3
1. Membres	3
2. Projet RoboCup SSL	3
→ Essentiel des règles et fonctionnement :	4
3. Organisation Interne.....	5
→ Répartition des tâches :	5
→ Notre planning :	5
→ Outils utilisés :	6
C. Nos Inspirations	7
1. NAMEC (Nouvelle-Aquitaine Mécatronique Club).....	7
2. Tigers Mannheim	7
D. L'Equipe NAELIC	8
E. Développement de l'association Bureau des Robots	8
II/- LE PROJET.....	9
A. Récapitulatif du 1er semestre.....	9
1. Mécanique	9
2. Informatique Embarquée	9
3. Software	10
B. X-Proto 1 : une version Low-cost d'un robot SSL.....	10
1. Mécanique	10
→ Problème sur les composants visserie à acheter :	11
→ Découpe Laser :	11
→ Conception 3D :	11
→ Montage du robot :	12
2. Electronique	12
3. Software	13
C. Robot final NAELIC	14
1. Mécanique	14
→ Système de réarmement du kicker :	14
→ Isolation Electromagnétique "Cage de Faraday" :	15
→ Accroche du carénage :	15
→ Pièces en impression sur lit de poudre :	15
2. Electronique	16
→ Travail théorique :	17
3. Software	18
Présentation de ROS 2	20
Navigation.....	23
D. Tests expérimentaux et améliorations.....	24
1. Banc de test du dribbleur	24
2. Le Kicker rotatif.....	25
III/- CONCLUSION	27
IV/- ANNEXES.....	28

I/- Partenariat et Organisation

A. Partenariat

1. Elektrons Libres



Elektrons Libres est une association, loi 1901, qui réunit des jeunes, parents et formateurs. Créée en septembre 2019 à Pau, elle a pour vocation de faciliter l'accès aux sciences à tous les jeunes de l'agglomération paloise, de favoriser leur mobilité à l'international, de renforcer leur identité européenne, tout en leur permettant de participer à des concours et en les soutenant dans leur orientation professionnelle.

Les jeunes à l'initiative de la création d'Elektrons Libres se sont rencontrés dans le cadre de projets auxquels ils ont participé au sein du lycée Saint-Cricq : concours de robotique, projet d'astronomie, séminaires scientifiques en Allemagne et en Écosse, etc.

En créant cette association, ces lycéens souhaitent partager leurs expériences et intégrer des jeunes extérieurs à leur établissement afin que ces opportunités soient offertes au plus grand nombre.

C'est Elektrons Libres qui nous a proposé de participer à la RoboCup au sein de l'association. Ainsi nous avons accès à leur réseau d'information et leur partenaire, Cohabit.

2. Cohabit



Cohabit est une association regroupant le fablab et le technoshop de l'université de Bordeaux. Ainsi Cohabit regroupe un espace collaboratif de fabrication numérique au travers du fablab et un technoshop pour le développement de projets technologiques innovants.

L'association propose des ateliers à destination des jeunes, mais également des professionnels ou particuliers sur des concepts technologiques.

Au travers de cette association, nous avons accès au **Club Innovation** de l'université de Bordeaux. Cette entité est à l'origine de rassemblement Elektrons libres/Cohabit et de la formation de l'équipe. C'est également le club innovation qui nous permet d'assister à des cours d'informatique embarquée et de software en rapport avec la RoboCup SSL ainsi qu'à des conférences en rapport avec les nouvelles technologies.

B. Qui sommes-nous et quel est notre projet ?

1. Membres

Nous sommes 10 élèves de l'école d'ingénieur ESME Sudria Bordeaux, dont 5 en parcours innovation :

- En parcours innovation :



GAZUI Laëticia
Leader



DECULTOT Martin
Mécanique



MOCAER Hugo
Mécanique



LUC Valentin
Software



MONCLUS Louis
Software

- Hors parcours innovation :



DARNAUD Louis
Mécanique



DE-FARIA Tenessy
Info Embarquée



CHUKRY Clément
Info Embarquée



SUREAU-SAMIT
Arthur
Info Embarquée



COUTHURES
Aurélien
Software

2. Projet RoboCup SSL

La **RoboCup** est la plus grande compétition de Robotique et d'intelligence artificielle au monde. Elle rassemble chaque année plus de 3500 participants et autant de robots venants de 45 pays différents. De même, des milliers de chercheurs venus du monde entier viennent assister à des conférences dans le but de partager connaissances, savoir-faire, idées et technologies.

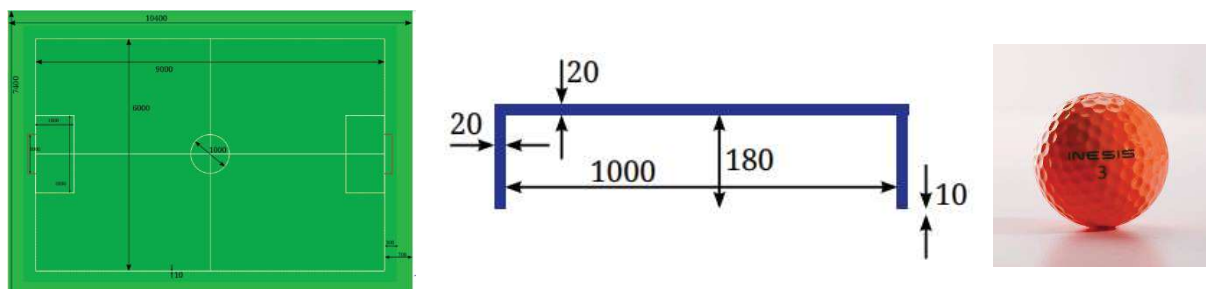
C'est une initiative scientifique internationale dont le but est de faire progresser l'état de l'art des robots intelligents. Lors de sa création en 1997, la mission initiale était d'aligner une équipe de robots capable de gagner contre les champions de la Coupe du monde de football humain d'ici 2050.

La RoboCup contient de nombreuses compétitions différentes réparties par ligues : Soccer, Rescue, @Home, Junior. Et dans chaque ligue, il existe plusieurs compétitions.

Nous avons choisi de participer à la compétition SSL (Small Size League) de la ligue Soccer, compétition se concentrant sur le problème de coopération et de contrôle intelligent de plusieurs robots dans un environnement dynamique avec un système de contrôle centralisé : l'ordinateur contrôle et envoie continuellement des ordres aux robots.

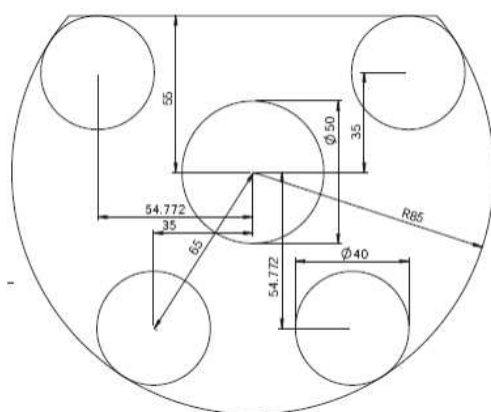
→ Essentiel des règles et fonctionnement :

La SSL est un **match de football** se déroulant entre deux équipes ayant chacune six robots. Chaque robot doit être conforme aux dimensions spécifiées dans les règles F180 : le robot doit pouvoir entrer dans un cercle de 18cm de diamètre et ne doit pas dépasser 15cm de hauteur. Les robots jouent au football avec une balle de golf orange sur un terrain recouvert de tapis vert de 9 m de long sur 6 m de large (voir image 1 ci-dessous).

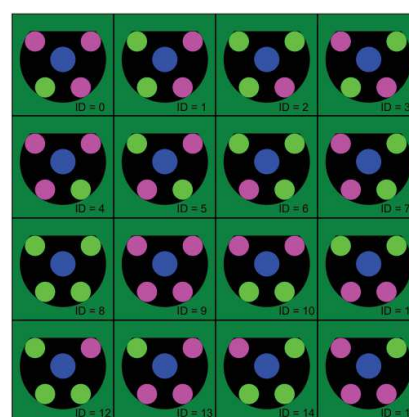


Tous les objets sur le terrain sont suivis par un système de vision standardisé appelé SSL Vision, un projet open source maintenu par la communauté de la ligue. Ce dernier traite les données fournies par quatre caméras attachées à une barre de caméra située à 4m au-dessus de la surface de jeu. Ces caméras envoient les images à un ordinateur coordinateur central qui traite les images et renvoie aux deux équipes la position, vitesse et orientation de chaque robot sur le terrain à chaque instant t (environ toutes les quelques millisecondes en simulation).

Pour avoir l'orientation du robot, l'ordinateur analyse les images de chaque robot grâce à un identifiant couleur composé de 5 cercles : un central représentant la couleur de l'équipe (bleue ou jaune), les quatre autres utilisés pour reconnaître chaque robot dans l'équipe (voir ci-dessous).



Placement conforme des identifiants couleur sur un robot



Signification des identifiants couleur pour l'ordinateur central

Les ordinateurs de chaque équipe sont utilisés pour le traitement nécessaire à la coordination et au contrôle des robots. Les communications sont sans fil entre l'ordinateur de chaque équipe et les robots et utilisent des émetteurs / récepteurs radio commerciaux dédiés.

En ce qui concerne les fonctionnements de certaines parties précises du robot (comment tirer dans la balle ? Comment le robot se déplace-t-il ? Comment attrape-t-il la balle pour dribbler ?) sont expliqués dans le paragraphe II/A. "Mécanique" en page 10.

3. Organisation Interne

→ Répartition des tâches :

Au vu de la densité de travail et de la taille du projet, nous sommes répartis sur 4 pôles : Mécanique, Electronique, Informatique embarquée et Software.

- Au pôle **Mécanique**, les étudiants étudient la modélisation 3D puis réalisent la fabrication du robot. Cela passe par toutes les étapes : utilisation du logiciel de modélisation (Inventor et Solidworks), élaboration de tableurs de budget répertoriant tous les composants à fabriquer, acheter ou à faire usiner, et l'impression des pièces 3D et de découpe laser au fablab.
- Au pôle **Electronique**, les étudiants apprennent à comprendre et modifier des schémas électroniques.
- Au pôle **Informatique embarquée**, les étudiants programment la partie embarquée du robot, comme par exemple les moteurs sur la carte électronique Maple Mini.
- Au pôle **Software**, les étudiants programment "l'intelligence stratégique" des robots. Sans faire de l'IA, ils travaillent sur des parcours d'arbres pour la prise de décision d'action de chaque robot dans le but d'une stratégie collective.

→ Notre planning :

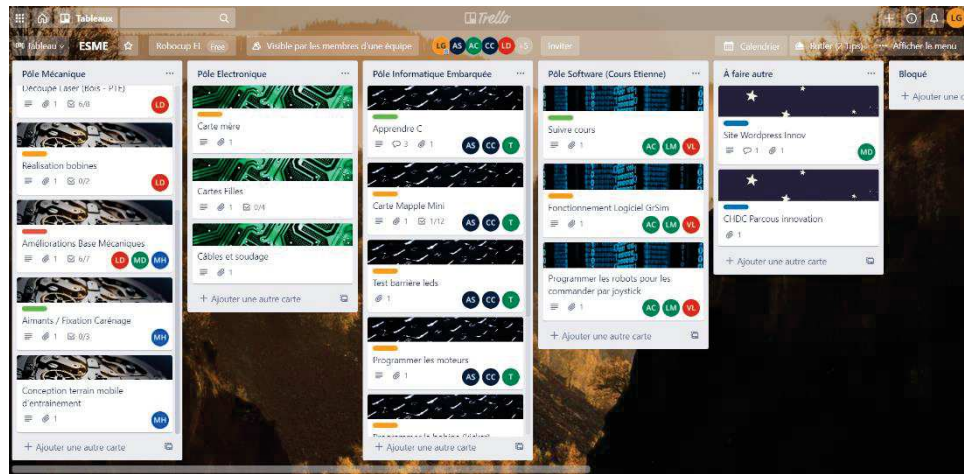
- Les **mardis soir** de 18h à 20h, les élèves s'occupant de la partie **Informatique embarquée** suivent les cours de Grégoire Passsault, professeur à l'Enseirb Matméca, à l'IUT et à l'université de Bordeaux et également membre de l'équipe Rhoban (quadruple championne du monde en ligue Humanoid Soccer de la RoboCup).
- Les **mardis après-midi** de 14h à 18h nous avançons le projet en parcours innovation.
- Les **jeudis après-midi**, lorsque la situation le permet, nous venons au fablab de l'ESME continuer le projet.
- Les **jeudis soir**, nous avons des réunions hebdomadaires de l'équipe qui intègre aussi des étudiants Cohabit ou Elektrons libres, totalisant une trentaine de participants.
Le même soir après la réunion, le pôle **Mécanique** tient une réunion pour échanger sur l'avancée du travail et se répartir les tâches de la semaine.
De même, de son côté le pôle **Software** suit les cours d'Etienne Schmitz, étudiant en dernière année à l'Enseirb (filiale robotique), membre de l'équipe NAMEC et qui enseigne la SSL au sein du club innovation de l'IUT de Bordeaux .
- Les **samedis matin**, 2 étudiants suivent des cours d'automatisme à l'Enseirb Matméca.

En plus, nous travaillons chacun de notre côté pour effectuer diverses tâches à réaliser dans la semaine, comme par exemple le moulage en résine des dribbleurs que l'on ne peut faire à l'école.

→ Outils utilisés :

- **Trello** : un outil de gestion de projets en ligne partagé entre les collaborateurs.

Cela permet de nous organiser entre les différents pôles et de suivre notre avancée générale et respectant des dates clés.



- **Teams** : Plateforme Collaborative de télétravail utilisée par l'école.

• **Discord** : Discord est un logiciel d'échanges très pratique par ses sous-groupes de discussions textuelles ou vocales permettant de parler d'un seul sujet par conversation sans créer un groupe pour chaque sujet. Initialement créé pour les gamers et pouvant accueillir de très nombreux membres dans un même groupe, il est utilisé de façon plus universelle dorénavant.



C'est la plateforme d'échanges de toute notre l'équipe, utilisée également pour suivre les cours proposés par le club innovation.

- **Google Drive** : Stockage et partage de fichiers.

On l'utilise pour partager tous nos fichiers administratifs, nos tableurs de commandes et tout fichier relevant de la mécanique.



- **GitHub** : un service web d'hébergement et de gestion de développement de logiciels, utilisé couramment pour l'hébergement de codes informatiques. Nous partageons nos codes informatiques en *Software* par ce service web.



- **GitLab** : GitLab est un logiciel libre basé sur git proposant les fonctionnalités d'un wiki, un système de suivi des bugs, l'intégration continue et la livraison continue. C'est le moyen de partage utilisé par l'équipe NAMEC.



- **OnShape** : Logiciel de conception assistée par ordinateur en ligne partagé entre collaborateurs.



- **Inventor** : Logiciel de modélisation 3D distribution Autodesk disponible librement pour les étudiants.



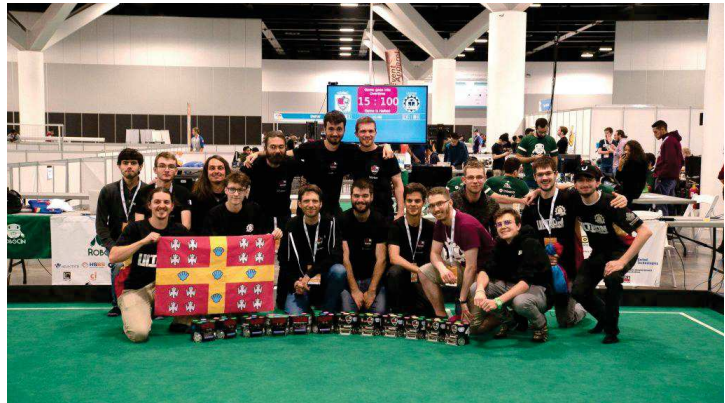
- **GrSim** : Logiciel fourni par la RoboCup pour simuler informatiquement le mouvement des robots lors d'un match de foot.

C. Nos Inspirations

La RoboCup repose avant tout sur le partage et la collaboration dans le but de faire avancer la technologie. Alors, participer à la compétition impose à tous les participants de publier leurs recherches en open source une fois le championnat du monde passé.

Ainsi, cette année nous nous inspirons principalement de 2 équipes : les *NAMEC* (Nouvelle Aquitaine Mécatronique Club) (Français) et les *Tigers Mannheim* (Allemands)

1. NAMEC (Nouvelle-Aquitaine Mécatronique Club)



L'équipe *NAMEC* a été lancée en 2017 avec le soutien de l'équipe *Rhoban* (quadruple championne du monde en Humanoïd size de la RoboCup) et d'enseignants chercheurs du LaBRI (Laboratoire Bordelais de Recherche en Informatique). L'objectif de ce projet régional est d'allier pédagogie innovante et collaboration entre enseignants chercheurs autour de la robotique.

L'équipe a participé à la RoboCup au cours des deux dernières années, atteignant les quarts de finale l'année passée.

Nous sommes en étroite relation avec leur équipe, notamment avec Etienne Schmitz, leader de l'équipe entière, Grégoire Passault nous donnant des cours d'informatique embarquée et Patrick Félix, à l'origine de la création de notre nouvelle équipe. Les *NAMEC* étant la seule équipe française en SSL, il leur tenait à cœur de soutenir une nouvelle équipe.

Ainsi, l'équipe *NAMEC* a accepté de nous prêter un de leurs robots utilisé lors de la RoboCup 2019 et nous a permis d'accéder à leurs fichiers mécaniques, électroniques et informatiques afin de nous lancer dans la compétition.

2. Tigers Mannheim



Les *Tigers Mannheim* est une équipe de RoboCup SSL allemande fondée en 2011 qui s'est classée en haut du classement lors des dernières RoboCup.



Après avoir étudié le robot *NAMEC*, nous nous inspirons également des *Tigers* pour ses composants placés par "modules" plus pratiques en cas de casse et ses recherches théoriques poussées, écrites et transmises aux autres équipes.

Ainsi, les *Tigers* nous ont beaucoup inspiré lors de la conception mécanique de notre propre robot.

D. L'Equipe NAELIC

L'équipe NAELIC (Nouvelle-Aquitaine Elektrons Libres Innovation Club) est le nom donné à l'équipe formée par le partenariat Elektrons Libres-Cohabit, dont nous faisons partie.

L'équipe est formée d'étudiants venant de différents horizons : Fac, Université (Informatique, mécanique, technique,...) et écoles d'ingénieurs. Cette diversité permet d'avoir accès à des compétences très variées au sein de l'équipe et pourvoir travailler avec différents états d'esprits, différentes compétences... ce qui sera également le cas lors de notre travail futur !

E. Développement de l'association **Bureau des Robots**

Durant ce second semestre, nous avons également créé notre propre association, nommée Bureau des Robots. L'objectif étant de recruter de nouveaux étudiants qui voudraient s'investir dans la robotique et participer à la Robocup.

De même, on souhaite proposer la participation d'une équipe ESME en catégorie RoboCup Junior. Cette équipe serait notamment ouverte aux 1ères années, ayant l'âge de concourir et pouvant ainsi s'initier à la robotique de manière ludique.

Enfin, la création de cette association nous permet de disposer d'un compte bancaire. Cela est très utile pour faciliter nos achats de pièces et matériels (et ainsi pouvoir faire nos propres tests). De plus, il nous faudra rechercher du financement pour pouvoir participer l'année prochaine à la RoboCup à Bangkok (frais de voyage, logement,...).

Ainsi nos objectifs de l'année prochaine sont :

- La création d'une équipe RoboCup Junior en partenariat avec l'association Elektrons Libres (projet réservé principalement aux Sup, donc sous réserve de membres motivés)
- La création d'un partenariat NAELIC - Bureau des Robots pour pouvoir continuer dans la compétition SSL dans l'équipe NAELIC tout en pouvant communiquer au sein de l'école pour promouvoir encore plus la robotique en France.

II/- Le Projet

Le projet relaté ci-dessous contient le travail réalisé au second semestre, ainsi c'est la suite du dossier de présentation n°1.

A. Récapitulatif du 1er semestre

Lors du premier semestre, nous avons réalisé un travail essentiellement basé sur la découverte et l'amélioration du robot SSL des NAMEC en un robot NAELIC. L'objectif du second semestre était de tester nos améliorations et maîtriser notre robot NAELIC.

Néanmoins nous avons été confronté à de nombreux retards (livraisons, sous-traitants,...) et nous avons dû nous réorienter vers une autre partie du projet.

Tout le travail effectué au 1^{er} semestre est loin d'avoir été inutile, bien au contraire, nous avons découvert le robot, avons appris énormément que ce soit au niveau du fonctionnement mécanique des pièces qu'en programmation ou encore sur les logiciels (modélisation 3D, découpe laser,...).

Nous allons voir un petit récapitulatif de ce qui a été réalisé dans les différents pôles.

1. Mécanique

Durant la première partie du 1^{er} semestre, nous avons découvert le robot NAMEC grâce à un prêt de quelques semaines de cette équipe. Nous avons référencé toutes les pièces se trouvant sur le robot (des composants électroniques à la moindre vis présente). Ainsi, nous avons acquis des compétences de gestion, référencement et recherche de pièces sur des sites professionnels. Nous avons également appris l'importance de tenir des documents à jour, complets, propres, sans quoi un projet ne peut avancer.

Une fois ce travail effectué (début novembre), nous avons étudié les robots des différentes équipes participant à la RoboCup SSL, et une équipe en particulier est sortie du lot : les Tigers Mannheim. En plus d'être 2nd au classement mondial, ils tiennent à jour leurs documents de recherche et les laissent open source (ce qui est normalement exigé par le comité d'organisation de la Robocup). Ainsi nous nous sommes inspirés de leur robot pour améliorer le robot NAMEC. Lors de cette étape, nous avons fait beaucoup de modélisation 3D, d'étude de solidité des différents matériaux, des méthodes d'usinage.... Cela fut très enrichissant.

A la fin de ce premier semestre, nous avons un robot amélioré mêlant les procédés techniques des NAMEC et des Tigers Mannheim et attendions qu'une seule chose : pouvoir tester ces améliorations.

2. Informatique Embarquée

Au 1^{er} semestre, la partie Informatique embarquée venait de commencer (seulement 3 cours avec un professeur connaissant le robot avaient eu lieu). Ainsi, tout était à développer.

Nous codions sur un microcontrôleur Maple Mini en langage C. Nous utilisons l'IDE Platformio de VisualStudioCode.

3. Software

Durant le premier semestre, la partie Software a consisté en la prise en main du logiciel de simulation grSim, permettant de simuler une partie de football SSL. Nous avons utilisé un Framework nommé Moleculer (s'appuyant sur Node.js) dans lequel nous codions en Typescript. A cette époque, nous étions déjà capables d'effectuer des mouvements avec les robots et souhaitions, entre autres, implémenter un algorithme d'évitement d'obstacle.

B. X-Proto 1 : une version Low-cost d'un robot SSL

Au début du second semestre, nous avons été informés d'un retard de commandes de plusieurs mois. Nous ne pouvions donc pas continuer le projet et réaliser les différents tests comme prévu.

Heureusement, nous avons rapidement eu l'idée de concevoir un robot SSL dit "Low-cost" en bois pour commencer les tests. Il servira également à terme de robot d'essai et pourra être répliqué pour que plus d'étudiants puissent travailler depuis chez eux directement sur un robot.

L'avantage du prototype est qu'il est plus facile à réaliser : il est en bois et en impression 3D. Ainsi, nous pouvons nous-même le créer entièrement (en commandant néanmoins quelques pièces de visserie). Il est également beaucoup moins cher (400-500€ au lieu de 4000€)

Toute la structure du robot est la même que sur le robot original. Les composants électroniques sont également les mêmes pour permettre la création de programmes rigoureux pouvant être appliqués directement sur les robots originaux.

1. Mécanique

Une fois ce nouvel objectif de conception d'un robot Low-Cost décidé, nous avons fait un référencement des pièces nécessaires et décidant dans quel matériau les usiner. Notre but était d'avoir affaire au moins de sous-traitants possible ; concevoir un robot rapidement, à moindre coût, et duplicable facilement pour les personnes voulant tester le robot depuis chez eux et faire avancer le projet.

Ainsi, toutes les pièces allaient être usinées sur place à l'ESME (découpe laser et impression 3D) et les quelques composants de visserie indispensables et non réalisables devaient être achetés sur le moins de sites possibles.

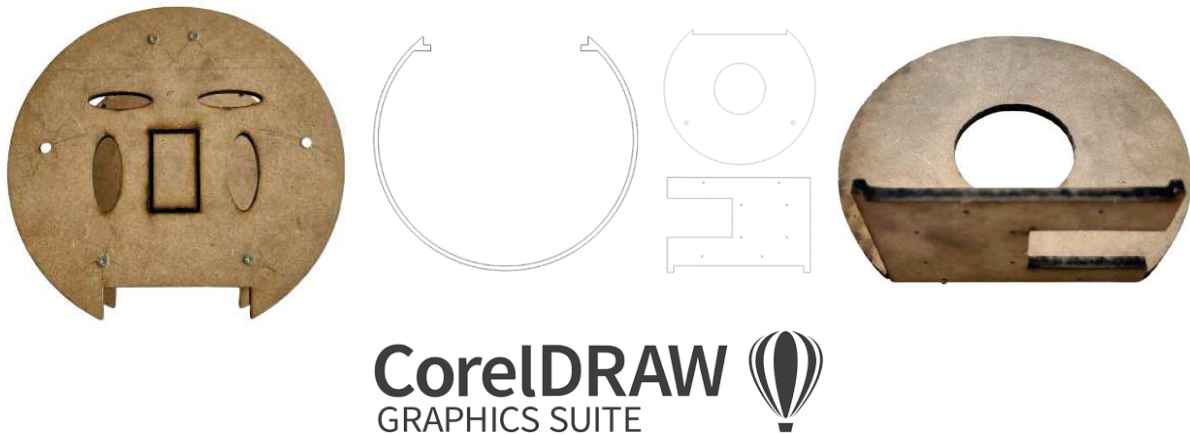
→ Problème sur les composants visserie à acheter :

Bien qu'étant peu nombreux, les composants visserie à commander nous ont donné du fil à retordre. En effet, ils nécessitent beaucoup plus de précision qu'une imprimante 3D ne peut réaliser, nous obligeant à les acheter. Ces composants sont assez difficiles à trouver dans le commerce. Nous avons réussi à les trouver au semestre dernier sur des sites professionnels de visserie à l'étranger.

Mais voilà qu'est arrivée la crise économique due au Covid19 : le coût des livraisons à l'étranger a grimpé (30€ minimum pour des délais assez longs). Nous avons alors été obligés de reprendre nos recherches et les réorienter sur des sites français professionnels de composants.

→ Découpe Laser :

Après avoir réalisé un tableur référençant les méthodes d'usinage pour chaque pièce, la première étape fut de dessiner les pièces nécessaires sur Corel Draw, un logiciel de modélisation pour découpe laser de bois. Ensuite, nous avons usiné ces pièces.



→ Conception 3D :

En parallèle, nous avons conçu les pièces à imprimer en 3D. Ce travail fut plus facile en récupérant les pièces dans la modélisation entière du robot, exceptées certaines qu'il a fallu créer ou réadapter pour les rendre plus solides. Ensuite, nous les avons imprimées.



→ Montage du robot :

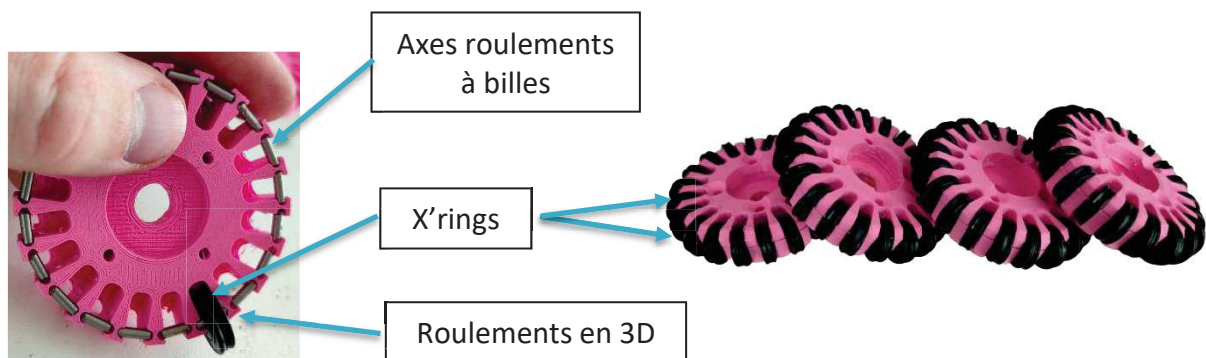
Une fois toutes les pièces mécaniques usinées, nous avons monté le robot :



Au cours de ce montage, nous avons rencontré plusieurs problèmes.

Tout d'abord, les "trous" nécessaires à la fixation des pièces entre elles étant réalisés à la perceuse de même qu'un manque de précision sur l'usinage, donne certains décalages des pièces entre elles (voir robot).

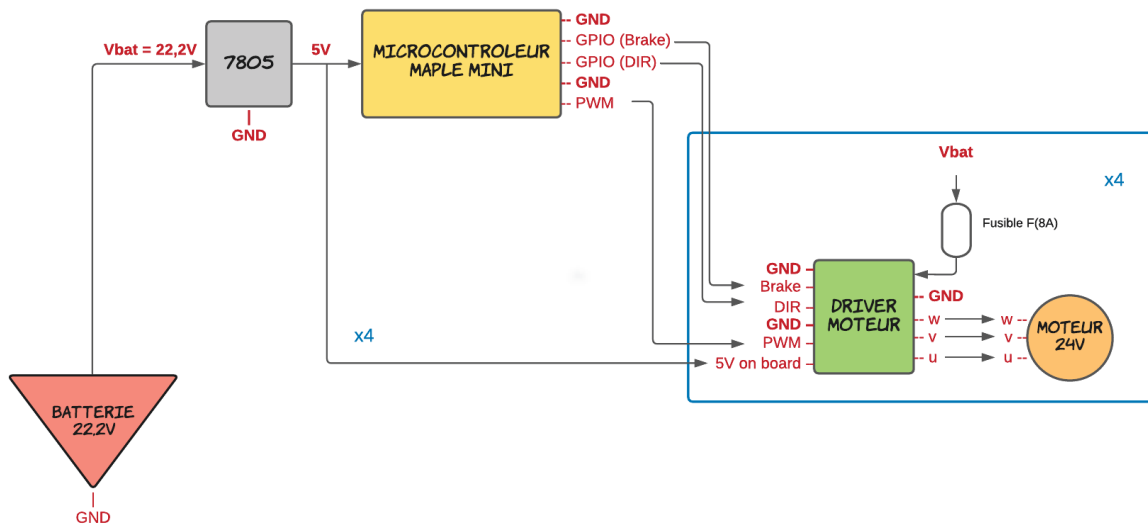
De plus, on a rencontré un problème au niveau du montage des roues. En raison de la Covid19 et de l'augmentation des prix, nous n'arrivions pas à trouver des axes de 3mm de diamètre pour 6mm de longueur en France (à l'étranger les coûts de livraison étant exorbitants). Ainsi nous avons choisi de prendre de grands axes de 12cm et de les redécouper nous-même. Ce fut un travail long et fastidieux, pour lequel nous avons dû redoubler d'imagination pour couper ces axes de la taille adéquate. Voici le résultat :



2. Electronique

Pour faire fonctionner ce robot low-cost, nous avons réalisé un circuit électronique basée sur la carte Maple Mini car cela permettra aux élèves en informatique embarquée de tester directement leurs programmes sur le robot low-cost.

Voici le schéma électronique :



Le circuit est composé d'un microcontrôleur Maple Mini qui contrôle 4 drivers moteurs, ces 4 drivers moteurs alimentant chacun 1 moteur du robot. Le tout est alimenté par une batterie 22,2V qui fournit de l'énergie directement au driver moteur. De même, on note la présence d'un composant 7805 qui permet d'abaisser la tension de la batterie pour alimenter convenablement le microcontrôleur (en 5V).

De plus, en raison du porte-conteneurs bloqué dans le canal de Suez qui a retardé la livraison de millions de pièces à travers le monde, il nous manque toujours à ce jour deux composants absolument nécessaires pour faire fonctionner le robot : les moteurs et les drivers moteur !

Ainsi, au cours de la semaine dernière nous avons décidé, dans l'espoir de voir le robot bouger au moins une fois, de remplacer les moteurs Brushless par des moteurs pas à pas présents dans le stock du fablab.

Ce n'est pas du tout la même technologie (brushless - pas à pas) et ce n'est pas contrôlé de la même manière : Maple Mini en C à Arduino. Néanmoins, il tenait à cœur des membres du projet de voir enfin le robot se déplacer.

3. Software

La partie Software étant la même du prototype à la version final du robot NAELIC, elle sera développée dans la **partie B** du cahier des charges.

C. Robot final NAELIC

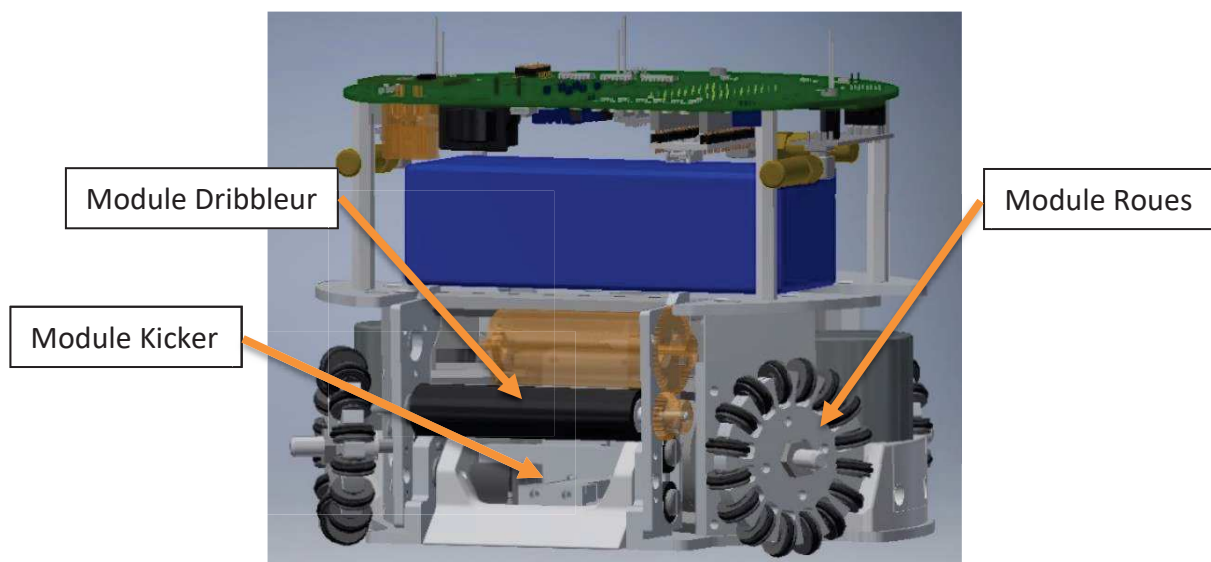
Parallèlement à la conception du robot Low-Cost, nous avons évidemment continué le travail sur le robot utilisé lors de la compétition RoboCup SSL.

Suite au retard de commandes/productions, nous avons décidé (l'équipe NAELIC) de rester cette année sur le robot NAMEC en améliorant quelques points mécaniques, changeant le microcontrôleur et le software.

Ce fut une décision qui a remis en question une bonne partie de notre travail effectué au premier semestre mais néanmoins nécessaire en vue du timing et des capacités en informatique embarquée de l'équipe. En effet, il est préférable de savoir contrôler un robot avant de commencer à l'améliorer, et pour l'instant, nous n'avons jamais essayé de contrôler un robot (pas de robot encore construit en raison d'un retard de commandes des cartes électroniques).

1. Mécanique

Voici le robot final NAELIC : (cf dossier n°1 pour l'explication des différents modules)



→ Système de réarmement du kicker :

La modification la plus importante fut le système de réarmement du kicker. L'équipe NAMEC utilisait deux élastiques maintenus par une plaque imprimée en 3D. Dès le début de l'année, lors du prêt de NAMEC de leur robot, nous avons remarqué que cette pièce de maintien était cassée et les élastiques détendus. Ce système n'était ni optimal, ni précis.

Ainsi plusieurs membres de l'équipe ont travaillé sur la modélisation d'une nouvelle pièce de maintien et sur la recherche d'un ressort adéquat en fonction de la raideur et longueur nécessaire.

Sur ce dernier point, nous avons découvert un détail à prendre en compte : les ressorts peuvent être de traction ou de compression. Si un ressort de compression est choisi pour de la traction, n'étant pas prévu pour cela, il risque de casser très facilement.

→ Isolation Electromagnétique “Cage de Faraday” :

Lors de la compétition RoboCup SSL 2019 à Sydney, l'équipe NAMEC avait remarqué la présence de problèmes d'interférences électromagnétiques dans leurs robots. Ils avaient réglé ce problème informatiquement sur le moment, mais il fallait trouver une solution pour pallier de manière plus simple cela : mécaniquement.

Ainsi, cette année ils ont tenté de créer une “cage de faraday” mécanique isolante.

La cage de faraday permet de faire une isolation électromagnétique



Dans cette solution, il est également nécessaire de renverser les condensateurs pour que les fils les alimentant n'interfèrent plus non plus.

→ Accroche du carénage :

Ce point peut paraître superflu mais est en fait assez important. Lors de la compétition RoboCup SSL, les robots se déplacent très rapidement. Il peut y avoir des chocs qui entraînent un mouvement du carénage et qui peuvent ensuite altérer le fonctionnement du robot (blocage des roues, accroche d'un autre robot,...). Un robot peut même avoir un carton jaune s'il accroche un robot adverse dans son carénage !

En 2019, l'équipe NAMEC utilisait des aimants pour fixer le carénage. Mais comme dans le cas du système de réarmement du kicker, nous avons remarqué dès le début de l'année que cette solution n'était pas optimale : les aimants n'accrochaient pas.

Ainsi, nous avons pensé à une solution mécanique simple mais efficace : l'utilisation de “boutons” d'accrochage de couturier.



→ Pièces en impression sur lit de poudre :

L'impression sur lit de poudre consiste à projeter une poudre d'alliage (métal ou céramique) et de le faire entrer en fusion de manière à assembler une pièce modélisée plus tôt en 3D. Cela permet de faire des pièces complexes et solides.



Grâce à ce procédé, nous avons pu obtenir nos rampes qui résisteront à la fois à de puissants chocs tout en ayant une géométrie complexe.

→ Travail théorique :

Ce deuxième semestre, 2 élèves de l'ESME ont suivi des cours d'automatisme le samedi matin à l'ENSEIRB Matméca. De plus, des calculs théoriques ont commencé à être réalisés avec le pôle système embarqué (détaillé ci-dessous).

2. Electronique

La première partie du second semestre fut la prise en main de l'environnement PlatformIO, mbed et la carte Maple Mini grâce aux cours suivis tous les mardis soir par les élèves en Informatique Embarquée.

Maple Mini



Platformio

Lors de ces premiers cours, a été étudié :

- L'Os : DigitalIn, DigitalOut, PwmOut, AnalogIn
- Communication : SPI, I2C
- Interruption

Ensuite, il fallait commencer à passer du code vers mbed. En effet, les NAMEC utilisaient une bibliothèque Maple Mini qu'ils devaient eux-mêmes tenir à jour. Pour faciliter ce point pour l'équipe NAELIC, le professeur Gregoire Passault a créé une passerelle entre mbed (codé sur PlatformIO) et le code Maple Mini.

Ainsi un cahier des charges a été rédigé, contenant tous les codes à réécrire sur mbed :

- Communication sans fil
- Communication USB/PC pour le master
- Communication moteurs, cinématique et odométrie
- Contrôle du dribbleur
- Carte kicker
- Barrière IR
- Buzzer
- Mémoire flash du robot

L'avantage de tout réécrire sur mbed est qu'il rend possible de changer de microcontrôleur facilement. Il n'est plus nécessaire de retranscrire tout le code à chaque fois puisque le mbed est utilisable partout.

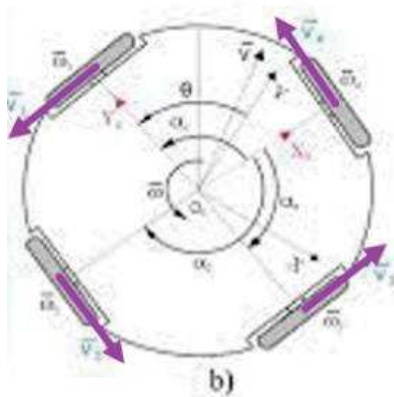
Dernièrement, l'équipe NAMEC a décidé de changer de microcontrôleur : passer de Maple Mini à une carte appelés Citron. Elle serait plus puissante et adaptée aux besoins de la RoboCup SSL. L'équipe NAELIC a décidé de suivre NAMEC en prenant également cette carte. Ces dernières semaines, quelques personnes de l'équipe ont commencé à tester cette carte.

→ Travail théorique :

Un travail théorique a été commencé sur les 2 pôles mécaniques et informatique embarquée.

Ci-après, nous allons voir le fonctionnement d'un module : **le contrôle des moteurs.**

Tout d'abord, voici la représentation vectorielle des vitesses de chaque roues : v_0, v_1, v_2, v_3



Voici la relation matricielle entre la vitesse des roues : v_1, v_2, v_3, v_4 et la vitesse du robot : v, v_n, ω pour une disposition des roues à 90° .

$$\begin{array}{c} \text{Vitesses en} \\ \text{translation} \\ \text{des roues} \end{array} \rightarrow \begin{bmatrix} v_0(t) \\ v_1(t) \\ v_2(t) \\ v_3(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & d \\ -1 & 0 & d \\ 0 & -1 & d \\ 1 & 0 & d \end{bmatrix} \begin{bmatrix} v(t) \\ v_n(t) \\ \omega(t) \end{bmatrix} \leftarrow \begin{array}{c} \text{"Ordre" du} \\ \text{Software} \end{array}$$

On obtient les relations suivantes :

- $v(t) = \frac{1}{2} (v_3(t) - v_1(t))$
- $v_n(t) = \frac{1}{2} (v_0(t) - v_2(t))$
- $\omega(t) = \frac{1}{4d} (v_0(t) + v_1(t) + v_2(t) + v_3(t))$

De même, les contrôles moteurs sont :

- $v_0(t) = v_n(t) + d\omega(t)$
- $v_1(t) = -v(t) + d\omega(t)$
- $v_2(t) = -v_n(t) + d\omega(t)$
- $v_3(t) = v(t) + d\omega(t)$

Le principe est que le software envoie un vecteur "ordre" contenant la vitesse de translation du robot (selon x), la vitesse normale (selon y) et la vitesse de déplacement totale voulue. Ces 3 valeurs passent par la matrice de passage, ce qui permet d'obtenir les vitesses en translation des 4 moteurs nécessaires pour avancer à la vitesse voulue par le software jusqu'à l'endroit désiré.

3. Software

Après la 1^{ère} soutenance, au mois de Janvier 2021, notre objectif a été de continuer les améliorations et développements prévus durant ce 1^{er} semestre, à savoir, en particulier la mise en place d'un robot « gardien » et le développement de l'algorithme d'évitement d'obstacle A*(pathfinding). La création du robot « gardien » fut assez aisé et rapide. Toutefois, le pathfinding nous a, quant à lui, posé beaucoup de problèmes et après plusieurs jours voire semaines, notre algorithme n'était toujours pas fonctionnel et possédait un bug dans sa conception. Afin de remédier à ceci, nous avons prévu de créer un outil de débogage dans le but de faire, entre-autre, un affichage de la grille et montrer le chemin parcouru par le robot.

En parallèle, nous avons également appris à utiliser GitHub et travaillé à la participation pour la RoboCup 2021 qui devait se dérouler uniquement en simulation. Pour ce faire, il a fallu, avec toute l'équipe NAELIC, écrire le TDP (Team Description Paper), présentant un aperçu de la conception de nos robots, avec les principaux composants matériels et logiciels dans leur état actuel et les améliorations que nous souhaitions apporter pour le mois de juin, lors de la compétition. Par ailleurs, nous avons également dû créer la vidéo de qualification : <https://www.youtube.com/watch?v=zFDJqvT1Y-8> dans laquelle nous présentons, cette fois-ci en vidéo, ce que nous savions faire avec les robots.

A partir de la fin du mois de Février/début Mars, nous avons reçu la réponse de la part du comité de la RoboCup. Dans leur réponse, il souhaitait que nous améliorions notre TDP et refassions une vidéo pour la fin du mois de Mars ; dans lequel nous pouvions effectuer un match dans sa globalité sur le logiciel de simulation GrSim (et non seulement certains mouvements comme nous savions faire).

Le challenge était très ambitieux, peut-être même trop, selon le leader de NAELIC, Etienne. Après plusieurs semaines à travailler ardemment sur le développement de stratégie afin d'effectuer un match, Etienne a proposé à l'équipe de temporiser, qu'il n'était pas judicieux de participer à cette édition de la RoboCup. Cela entrainerait notamment trop de stress pour être dans les temps et dans la précipitation nous ferions mal les choses. Il a paru à l'équipe plus opportun de se focaliser sur la prochaine compétition de 2022, à Bangkok.

A la suite, Etienne, actuellement en stage au CATIE (Centre Aquitain des Technologies de l'Information et Electroniques), décida de changer complètement de Framework (auparavant Molecular, présenté au 1^{er} Semestre), et d'utiliser ROS2, qu'il utilise depuis quelques temps dans son stage. ROS2 et plus particulièrement ROS (Robot Operating System) est un ensemble d'outils informatiques sous formes de logiciels libres open source, permettant de développer des logiciels pour la robotique en général. Chaque personne souhaitant faire de la robotique se doit d'apprendre ROS car dans quelques années il sera universel. On estime aujourd'hui que 50% des robots utilisent déjà ROS.

Depuis notre transition, très récente, vers ROS2 (qui a duré quelques temps dû à quelques problèmes d'installation notamment), notre objectif a été, dans un premier temps, de se familiariser avec ce nouvel environnement avec les différents concepts qu'offre ce

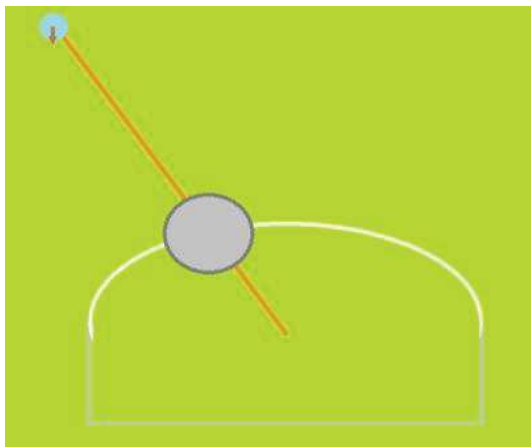
Framework. Puis, depuis peu, nous avons essayé de contrôler un robot avec une manette de type Playstation 4, en simulation d'abord, puis à terme sur un vrai robot.

En parallèle de cela, avant le changement sous ROS2, nous avons développés quelques nouveaux services sous Molecular et notamment des comportements pour les robots. Nous avons par exemple développé un gardien, un algorithme de positionnement circulaire ainsi qu'un algorithme de pathfinding.

Le gardien a été développé en plusieurs itérations, la première consiste simplement à se mettre entre la balle et le centre des buts, à une distance définie. Cette première itération était efficace pour des balles arrivant à faible vitesse mais en augmentant la vitesse de la balle, le gardien n'arrive plus à suivre la balle et n'arrive pas à l'empêcher de rentrer.

Ce souci est récurrent dans les implémentations de ce type car le robot n'anticipe pas la position de la balle puisqu'il se met juste entre le centre des buts et la balle. Toutes les itérations testées fonctionnent sur ce principe mais il existe des ensembles de positions possibles pour le robot permettant d'augmenter l'efficacité du gardien.

Pour faire face à ce souci, nous avons commencé à travailler sur un système prenant en compte la vitesse de la balle pour anticiper sa position d'intersection avec l'ensemble de positions du robot.



Sur le schéma ci-contre, l'ensemble de positions du robot est en blanc tandis que la ligne balle-centre des buts est en orange. Le robot se situe à l'intersection des deux domaines. L'idée d'amélioration consisterai à mettre la ligne balle-centre suivant le vecteur vitesse de la balle, le robot pourrait alors anticiper dès le début la trajectoire de la balle.

L'algorithme de positionnement circulaire consiste à placer les robots de façon circulaire ou semi-circulaire autour d'un point donné en entrée du service, l'intérêt de ce service est avant tout éducatif car son développement nous a permis de bien prendre en main le Type script et le développement de service sous Molecular.

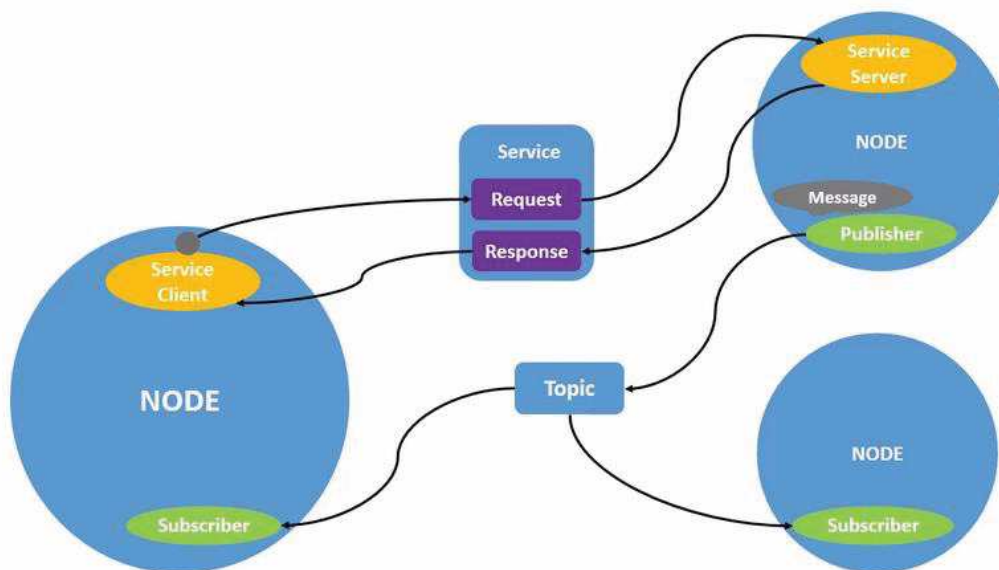
L'algorithme de pathfinding consiste à chercher la trajectoire la plus courte entre deux points tout en évitant les obstacles sur cette trajectoire. Nous avons implémenté un pathfinding basé sur A*, le principe est de diviser le terrain en cases et de vérifier ces cases dans un ordre dicté selon des critères donnés, ces critères sont principalement la distance et la distance de Manhattan (addition des coordonnées x et y). L'ordre dépend de ces distances, et plus une case est proche de l'arrivée et plus elle a de poids dans l'ordre et est prioritaire. Malheureusement, nous n'avons pu finir l'algorithme à cause du passage sous ROS2 dans lequel il existe une librairie le proposant.

Présentation de ROS 2

Depuis le lancement de ROS en 2007, beaucoup de choses ont changé dans la communauté robotique et ROS. L'objectif du projet ROS 2 est de s'adapter à ces changements, en tirant partie de ce qui est intéressant dans ROS 1 et en améliorant ce qui ne l'est pas.

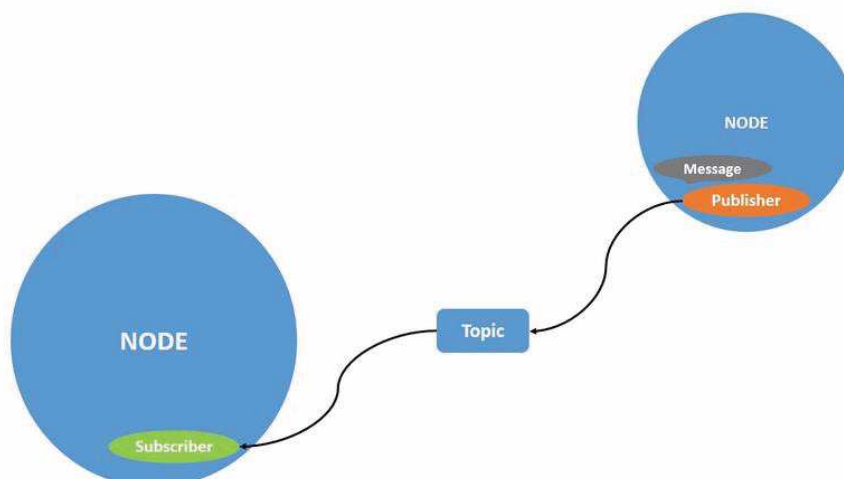
→ Nœud

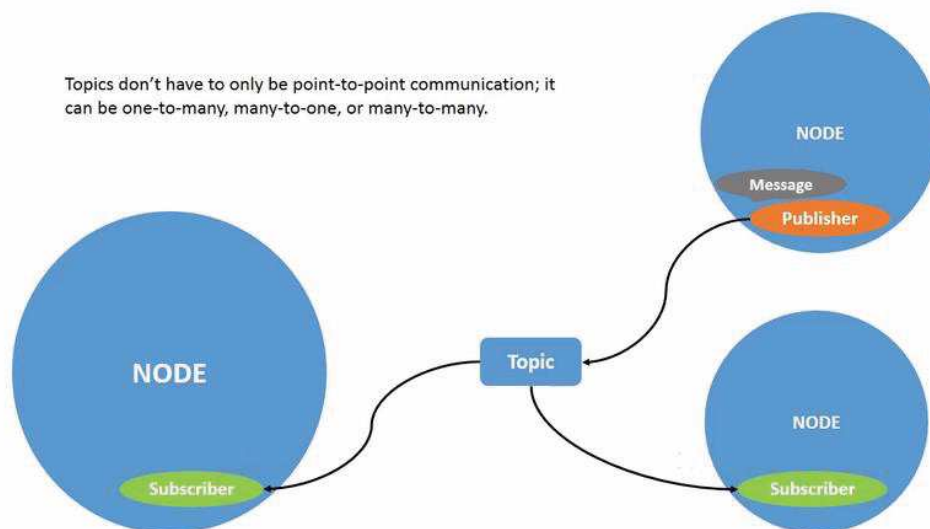
Chaque nœud de ROS est responsable d'un objectif unique (par exemple, un nœud pour contrôler les moteurs des roues, etc...) Chaque nœud peut envoyer et recevoir des données à d'autres nœuds via des topics, des services, des actions ou des paramètres. Un système robotique complet est composé de nombreux nœuds travaillant ensemble. Dans ROS 2, un seul exécutable (programme C++, programme Python, etc.) peut contenir un ou plusieurs nœuds.



→ Topic

Les topics sont un élément essentiel de ROS qui sert de « bus » aux nœuds pour échanger des messages.

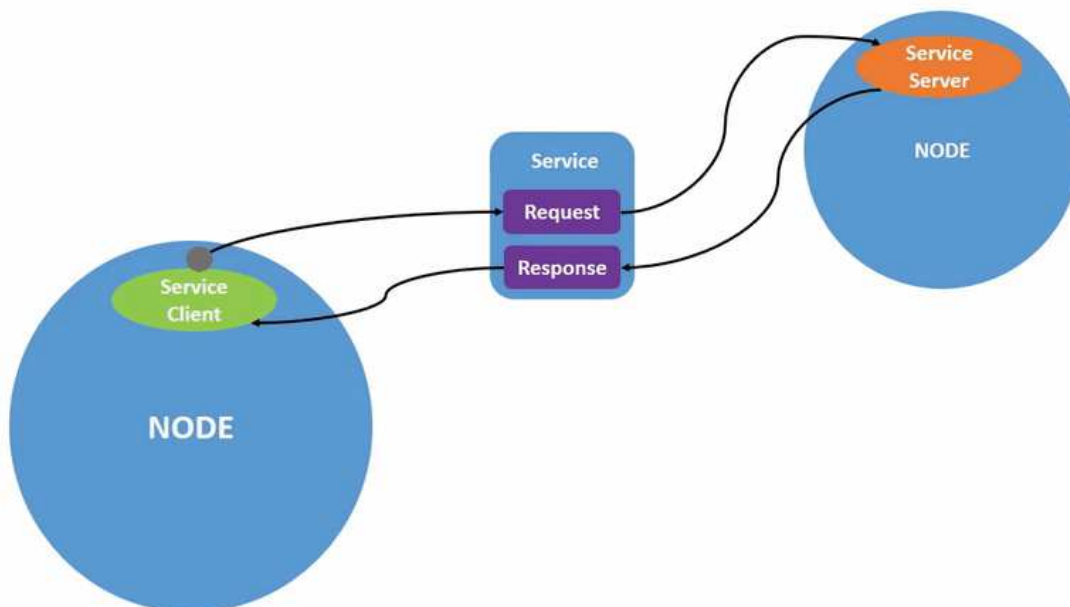


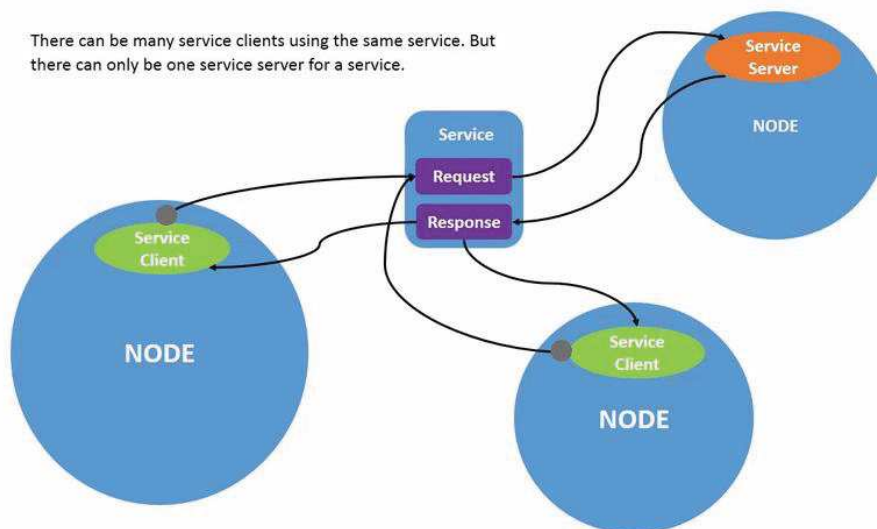


Un nœud peut publier des données dans un nombre quelconque de topics et avoir simultanément des abonnements à un nombre quelconque de topics.

→ Services

Les services constituent une autre méthode de communication pour les nœuds. Les services sont basés sur un modèle de requête et de réponse, contrairement au modèle Publisher (éditeur) et Subscriber (abonné) des topics. Alors que les topics permettent aux nœuds de s'abonner à des flux de données et d'obtenir des mises à jour continues, les services ne fournissent des données que lorsqu'ils sont spécifiquement appelés par un client.





→ Paramètres

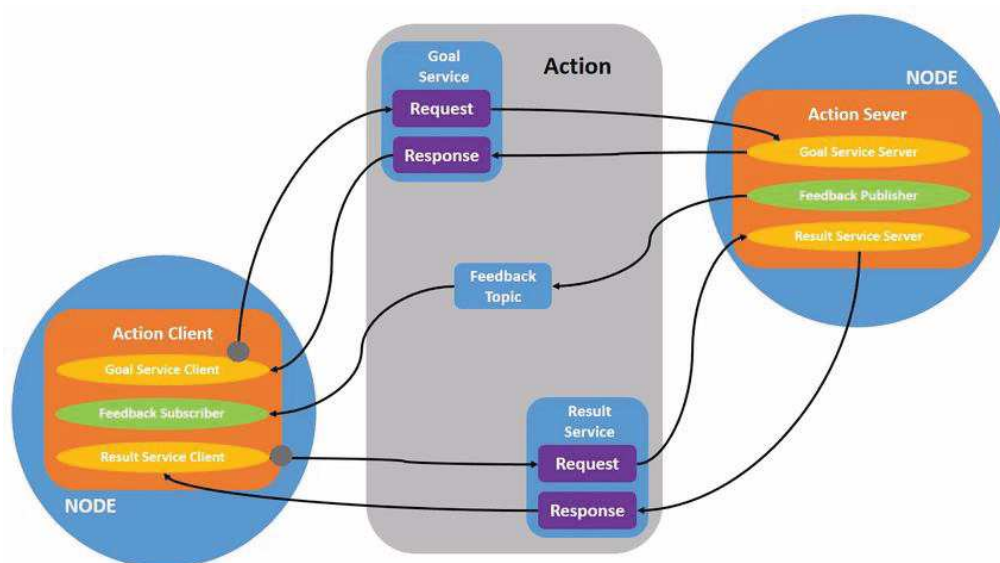
Un paramètre est une valeur de configuration d'un nœud ou un réglage d'un nœud. Un nœud peut stocker des paramètres sous forme d'entiers, de flottants, de booléens, de chaînes de caractères et de listes.

→ Actions

Les actions sont l'un des types de communication de ROS 2 destinés aux tâches de longue durée. Elles se composent de trois parties : un objectif, un résultat et un retour d'information ou feedback.

Les actions sont construites sur des topics et des services. Leur fonctionnalité est similaire à celle des services, sauf que les actions peuvent être annulés en cours d'exécution. Elles fournissent également un retour d'information constant, contrairement aux services qui renvoient une seule réponse.

Les actions utilisent un modèle client-serveur, similaire au modèle Publisher-Subscriber. Le nœud "client d'action" envoie un objectif à un nœud "serveur d'action" qui accuse réception de l'objectif et renvoie un flux de retour et un résultat.



Navigation

Le but de la navigation pour un robot est de parcourir une distance, d'un point A à un point B sans rien toucher sur son chemin, c'est à dire en évitant les obstacles, ce qui fera intervenir le pathfinding déjà abordé au premier semestre. Le planner, controller et recoveries que nous allons aborder maintenant, nous permettront de faire ceci.

→ Planner

Le planner crée une trajectoire en évitant des obstacles déjà connus. Ce sont des obstacles à un instant T qui sont donc fixes. Il y a donc dans notre cas la délimitation du terrain. Grâce au planner, le robot restera à l'intérieur du terrain lors d'un déplacement. Le planner utilise une costmap qui est une carte en 2D sur laquelle la navigation va s'effectuer. Cette costmap est dite globale car elle recouvre tout l'environnement (tout le terrain).

→ Controller

Le controller crée une trajectoire qui évite des obstacles dynamiques (en mouvement). A la différence du planner, le controller utilise une costmap locale qui est caractérisé par un rayon autour du robot, que l'on a défini à l'avance. Il permet donc d'éviter les autres robots qui sont en déplacement lors d'un trajet.

→ Recoveries

Les recoveries interviennent lorsque la position que veut atteindre le robot n'est pas atteignable. Cela peut arriver lorsqu'il y a un énorme obstacle devant qui empêche le robot d'avancer plus. Lorsque cela arrive, il y a plusieurs solutions :

- Spin : fait effectuer une rotation au robot
- Backup : fait retourner le robot vers la position initiale
- Wait : laisse le robot immobile

La Navigation (Nav2) est une pile logiciel complémentaire à ROS2, permettant donc à un robot d'éviter des obstacles. Cela est très pratique car l'algorithme d'évitement d'obstacle (pathfinding) est donc déjà codé et intégré à ROS ce qui facilite grandement le développement. Toutefois, cet algorithme déjà intégré n'est pas complètement optimal et il se pourrait que nous en développons un autre, plus optimisé, à l'avenir, si cet algorithme se révèle insuffisant durant un match.

D. Tests expérimentaux et améliorations

1. Banc de test du dribbleur

Pendant ce second semestre, un élève en particulier a été dédié à la réalisation d'un banc d'essai dribbleur et des tests d'efficacité. Nous avons rédigé un compte rendu expérimental à remplir au fur et à mesure des tests.

De même, deux autres élèves ont été chargés de maîtriser le procédé de moulage résine des moules, dont vous pouvez voir les résultats ci-dessous. Un procédé expérimental a également été rédigé (cf dossier de présentation n°1).

Tous les moules réalisés et déclarés "bons" sur le banc d'essai sont testés sur le robot réel pour voir son efficacité lorsque le robot est en mouvement.

Sur le banc de test, nous utilisons une Dremel pour simuler la rotation du moteur ainsi qu'un coupleur reliant les barres des dribbleurs à l'arbre de la Dremel de manière à ne pas faire subir de sur-effort à nos composants et de ne pas avoir des résultats biaisés. Enfin nous avons utilisé des roulements à billes pour permettre une meilleure rotation.

Banc d'essai seul



Banc d'essai prêt à l'emploi



Pour les tests nous avons opté pour la vitesse 5 de la Dremel ce qui correspond à 5000 tr/min car nous avons remarqué qu'au-delà, le dribbleur avait du mal à récupérer la balle et avait tendance à la repousser ce qui est contre-productif.

Pour être précis dans nos études nous avons filmé toutes les vidéos sans changer la position de la caméra. De plus, nous avons utilisé une rampe de lancement de manière à faire tomber la balle à la même vitesse et du même endroit pour tous nos tests.

Pour le traitement des vidéos, nous utilisons le logiciel *Tracker* qui est une alternative gratuite et libre à *Latis Pro*.



Le logiciel permet de faire la même chose que le célèbre LatisPro des lycées : poser des points pour chaque image, puis réaliser des calculs mathématiques. Les résultats obtenus sont précis et faciles à analyser dans des tableurs excels.

Cela peut être notamment pratique pour calculer qualitativement le temps que prend la balle à arriver pile au milieu du dribbleur ou d'autres critères que nous avons étudié.

Voici une partie du compte rendu expérimental :

Nom du moule	Moule_f1_p5	Moule_f2_p	Moule_f3_p	Moule_f4
Nombre de filet	1	2	3	4
Pas	5mm	10mm	15mm	20mm
Conclusion orthogonale		Accroche sur différentes vitesses malgré un rebond sur la grande vitesse	Accroche sur les vitesses moyennes et faibles malgré un rebond sur la vitesse moyenne	Accroche sur les vitesses moyennes et faibles malgré un rebond sur la vitesse moyenne
Conclusion angulaire		Retiens la balle que pour une vitesse de la balle très faible	Retiens la balle que pour une vitesse de la balle très faible	Retiens la balle que pour une vitesse de la balle très faible
Conclusion de dessus		Tendance a aller a droite (peut-être du au système et non au dribbler	Tendance a aller a droite (peut-être du au système et non au dribbler	Tendance a aller a droite (peut-être du au système et non au dribbler
Volume	8700 mm ³ 8,7 mL	8700 mm ³ 8,7 mL	8700 mm ³ 8,7 mL	8700 mm ³ 8,7 mL

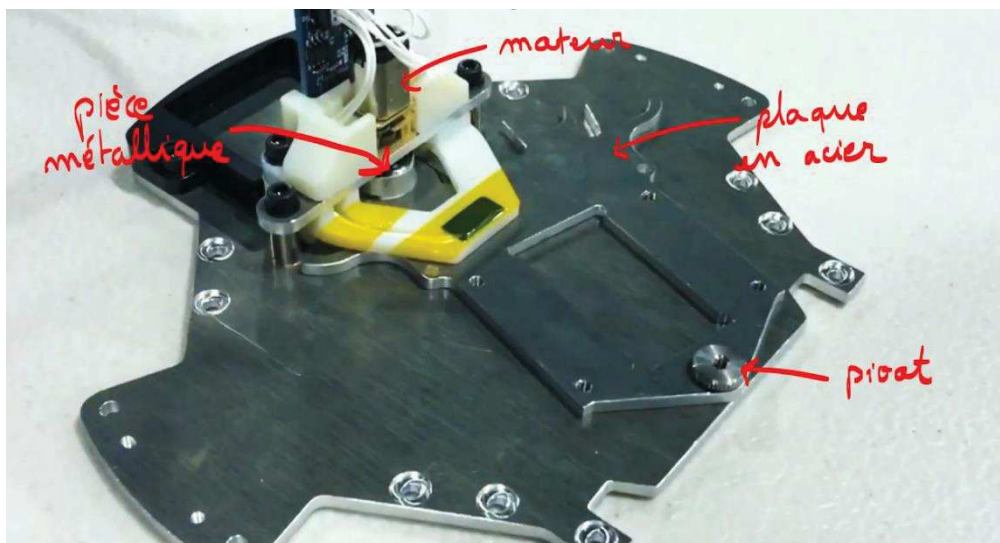
2. Le Kicker rotatif

En nous renseignant sur les autres équipes, le fonctionnement et la composition de leurs robots, nous avons découvert beaucoup de systèmes intéressants dont un tout particulièrement : le kicker rotatif.

Le kicker rotatif est un kicker qui a la particularité de tourner autour d'un axe de rotation. Cette amélioration du kicker est très intéressante car elle permet de tirer dans la balle avec un angle.

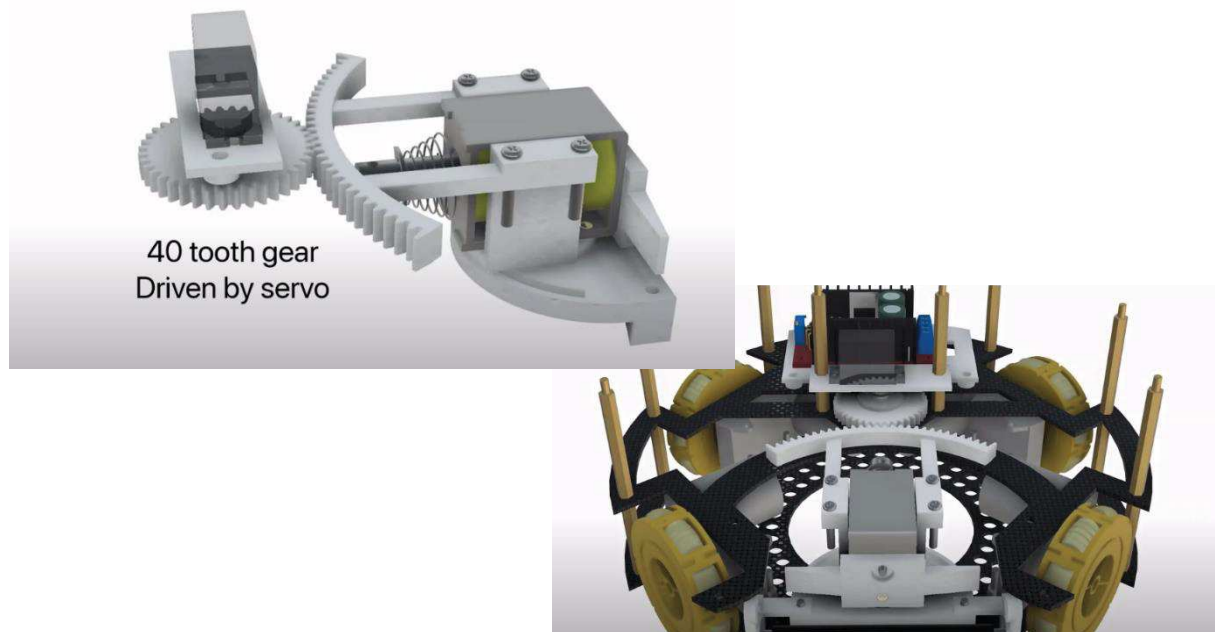
Les robots sont en général codés pour se placer devant le tireur pour stopper son tir et c'est là que ce kicker rotatif nous donnerait un gros avantage pour passer les contres adverses et simplifier l'avancée de l'équipe vers les buts. De plus, combinée avec le backspin du dribbler, la balle peut prendre une trajectoire courbée et par conséquent rendre son arrêt beaucoup plus difficile.

Différents systèmes ont été inventés afin de permettre au kicker de prendre un angle. Voici un exemple :



Ici une pièce métallique est reliée à l'arbre du moteur et va tirer ou pousser la plaque en acier reliée au pivot.

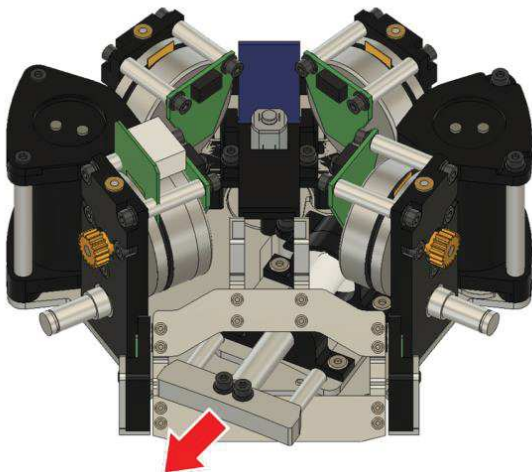
Ce que nous avons prévu de faire ressemblerait plus à ce modèle :



Cette fois-ci un servo moteur dirige un engrenage qui est relié à un fragment d'engrenage. Ce dernier permet d'orienter le kicker qui peut prendre un angle de 25° à gauche et 25° à droite.

Pour mettre en application ce kicker, il faut prendre en compte de nombreux paramètres. Dans un premier temps, il est nécessaire de miniaturiser le kicker pour que sa rotation soit possible. Il faut donc faire un compromis au niveau des performances. Les moteurs devront être surélevés et la transmission jusqu'aux roues sera assurée par des engrenages.

La structure entière du robot devra par ailleurs être revue car, en remontant les moteurs, cela engendrera un décalage des composants qui se trouvent au-dessus. Cependant les réglementations de la taille du robot doivent être respectées (20*15 cm)



Le travail sur le kicker rotatif reste très dense et est donc un projet qui sera réalisé à plus long terme. Il faudra réaliser de nombreux tests et trouver le système le plus approprié pour angler le kicker.

III/- Conclusion

Ce projet nous a apporté beaucoup de compétences, à la fois techniques et humaines.

Techniquement, les élèves du pôle mécanique ont appris à modéliser en 3D sur Inventor, à dessiner des pièces sur Corel Draw, à utiliser les machines présentes au fablab (imprimantes 3D et découpeuse laser). Ils ont également appris à s'organiser et chercher des composants de manière efficace sur des sites professionnels (Maxon motor, Misumi,...).

De même, les élèves travaillant sur l'informatique commencent à apprendre le fonctionnement de ROS, le framework de l'avenir pour la robotique.

Humainement, cette expérience nous apporte beaucoup. Que ce soit en travail d'équipe, organisation, respect de délais, adaptation et recherche de solutions alternatives, mais également en menant à bien ce projet concret en ayant pour but final la participation à la RoboCup à Bangkok en juin 2022 !

→ Remerciements

Nous souhaitons remercier l'association Elektrons Libres, qui nous a permis de participer à ce projet.

Nous remercierons de même son partenaire, Cohabit ainsi que le Club Innovation nous permettant chaque semaine d'apprendre et innover autour de la RoboCup.

Nos remerciements vont également à notre école, l'ESME Sudria Bordeaux, son directeur M.Crouzil ainsi que le responsable du FabLab M.Guillemard qui nous soutiennent dans ce projet complexe depuis le début.

Un grand merci à Patrick Félix et surtout à Etienne Schmitz, leader de l'équipe entière qui se démène chaque jour pour coordonner et satisfaire les besoins de chaque pôle, que ce soit en termes de commandes, dossiers administratifs ou problèmes informatiques et qui trouve également le temps de donner des cours de Software et de mener des conférences autour de la robotique !



IV/- Annexes

- Un lien Google Drive donnant accès aux fichiers suivant :
 - Résumé en français des règles de la SSL
 - Notice d'installation de grSim et d'un dual boot Ubuntu

https://drive.google.com/drive/folders/1sA1x16VK1wc-dBBk1QlpG-cocq1k_cEH?usp=sharing

- Sources :

<https://elektronlibres.fr/>

<https://www.iut.u-bordeaux.fr/cohabit/>

<https://ssl.RoboCup.org/>

<https://ssl.RoboCup.org/rules/>

<https://namec.fr/>

<https://www.tigers-mannheim.de/index.php?id=14>

<https://docs.ros.org/en/foxy/index.html>

<http://217.182.168.167:6875/>

- Quelques Images :

