

PSC MEC-11
Coupe de France de Robotique 2019

Rapport Intermédiaire



Agier Julien
Colson Raphaël
Chauiq Elbadre Hamza
Escande Maxime
Iben Brahim Yahya

Introduction

La coupe de France de robotique est une compétition qui a lieu chaque année et dont la participation fait l'objet d'un PSC depuis la X2014. Pour l'anecdote, une équipe de l'Ecole Polytechnique a même gagné en 1998. L'un des principes de la coupe de robotique est que les règles, au sens des actions que doit accomplir le robot changent chaque année de telle sorte que les participants doivent réinventer leur technologies d'une année à l'autre. Cependant, les dimensions et les formes autorisées pour le robot évoluent peu lors des changements de règles.

L'expérience montre que les équipes les plus performantes ne créent en général pas un robot ex-nihilo et repartent plutôt d'une base fiable capable de se déplacer et de s'orienter efficacement. C'est là la stratégie que nous souhaitons adopter dans ce projet en construisant un châssis motorisé adaptable selon les besoins que rencontreront les équipes ultérieures pour qu'elles puisse petit à petit en accumulant de l'expérience monter dans les classements. C'est autour de cet intérêt que s'est formé notre équipe, deux groupes de PSC distincts travaillant sur un même projet. Notre groupe est en charge de la partie mécanique du robot, le groupe INF09 de la partie Software du robot.

Ne sachant pas trop quelles difficultés nous allions rencontrer, nous avons établi un partenariat avec Exotec Solutions, une entreprise de robotique française, qui a accepté de nous sponsoriser et nous introduire à la démarche industrielle qu'ils utilisent pour fabriquer leur propres robots.

Nous présenterons à travers ce rapport les résultats obtenus jusqu'à maintenant ainsi que nos objectifs pour la deuxième partie de l'année.

I. Où en sommes-nous ?

a. Les objectifs

Nos objectifs initiaux dans ce PSC étaient de réaliser un robot particulièrement précis en suivant une démarche proche d'une démarche industrielle. Il était particulièrement important que le robot réalisé soit réutilisable de la manière la plus efficace possible dans les prochaines années, le but étant que l'Ecole polytechnique puisse présenter une équipe de plus en plus compétitive qui se concentrerait sur l'amélioration et l'adaptation du robot plutôt que devant repartir de rien.

La démarche industrielle que nous avons adoptée nous a été proposée dans le cadre d'un partenariat avec une entreprise spécialisée dans la fabrication de robots, cependant la démarche nous a semblée d'autant plus utile qu'elle semble très flexible et pouvoir s'adapter à la majorité des projets. Nous aurons l'occasion de décrire plus précisément cette manière de fonctionner dans la suite, mais en quelques mots, nous avons essayé d'apporter une grande réflexion aux capacités que devait avoir le robot d'un point de vue quantitatif (sa vitesse, ses distances de freinages...), réfléchi aux composants nécessaires pour avoir ces capacités et établi une timeline précise.

Un intérêt à avoir des échéances précises s'est vite fait sentir, puisque cela nous a permis de déceler que nous prenions du retard. Voilà pourquoi nous avons décidé de revoir nos objectifs à la baisse en ne cherchant plus à participer à tout prix à la coupe : il nous semble en effet plus important de bien faire les choses et de réaliser une base solide, bien conçue et testée précisément plutôt que de bâcler les étapes moins cruciales comme l'agencement des composants ou l'établissement de test précis, qui à terme nuiraient forcément aux performances.

b. Le cadre

Notre PSC est un projet qui comporte de nombreuses dimensions qui ne sont pas strictement scientifiques, mais qui sont des composantes inévitables de tout projet industriel ou de recherche : interagir avec des contacts extérieurs et coordonner des équipes travaillant sur différents aspects du projet.

Pour ce qui est des relations extérieures, nous avons démarché plusieurs sponsors avant d'obtenir une réponse positive de la part d'*Exotec Solutions*. Il nous a ensuite fallu communiquer et négocier avec eux un contrat acceptable par les deux parties. Pour l'instant, 5.000€ ont été versés sur les 10.000 au total prévus à terme du contrat, et les représentants d'*Exotec* sont venus le 23 janvier à Polytechnique présenter leur entreprise et rencontrer les élèves, ce qui est ce que nous avons convenu de leur offrir en échange de leur aide technique et financière.

Par ailleurs, notre projet a la particularité d'être rattaché à deux groupes distincts, ce qui n'empêche pas que le travail réalisé par chaque partie ait régulièrement besoin d'être mis en commun. Pour assurer la coordination des travaux, une réunion est assurée chaque semaine pour mettre en commun le travail des deux groupes. Parallèlement à notre avancement, le groupe INF 09 travaille sur trois points essentiels :

- L'asservissement des moteurs : le but de cette partie est d'asservir les moteurs en position. C'est à dire que le robot doit atteindre le plus rapidement possible la valeur de la consigne de position et de la maintenir, quelles que soient les perturbations externes. Le groupe INF 09 s'occupe donc, avec nous d'écrire l'algorithme d'asservissement qui consiste à comparer continuellement la consigne et l'état actuel du système afin de le corriger efficacement.

- Calcul et optimisation de trajectoires : Cette deuxième partie consiste à générer la consigne de position en fonction des tâches que le robot doit effectuer, et d'optimiser ces consignes pour avoir une trajectoire minimale quels que soient les obstacles (typiquement le robot adverse) sur le chemin du robot.

- Repérage vidéo : Notre robot aura deux système de repérage dans l'espace. Le groupe MEC 11 s'occupe du repérage avec les encodeurs et les capteurs de distance tandis que le groupe INF 09 s'occupera du repérage avec caméra. Ce type de repérage permettra un positionnement absolu et ainsi de redémarrer l'erreur de l'odométrie à zéro. Il permettra aussi de voir où se trouvent les palets et de repérer leurs couleurs afin de les classer au bon endroit.

c. La démarche : le Design Control

Dans le cadre de notre partenariat avec Exotec Solutions nous avons été initiés à la méthode du Design Control. Le Design Control est l'application d'une méthodologie formelle pour le développement d'un produit. Le but étant de planifier toutes les étapes de la conception du robot et de les vérifier grâce à des tests précis et non ambigus définis à l'avance, selon le schéma ci-dessous :

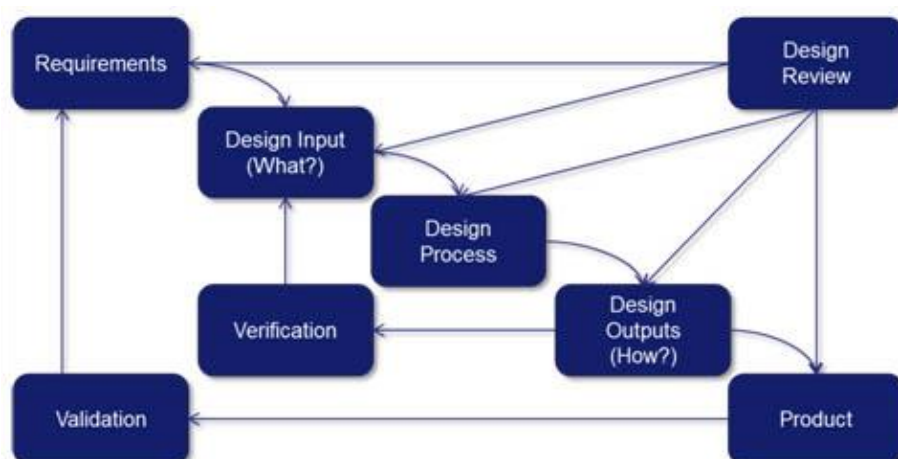


Figure 1 : Schéma d'action de la démarche industrielle adoptée

Après un après-midi de formation avec Exotec Solutions, nous avons donc commencé la planification, qui consistait en la rédaction de plusieurs documents :

« SRS (system requirement specification) »

Un tableau qui décrit exactement ce que le système doit être capable de faire selon les demandes de l'utilisateur et des autres acteurs. Il faut prendre en compte les contraintes environnementales et réglementaires.

Figure 2 : Exemple de l'un de nos SRS

6 - Robot abilities	
REQU 24	The robot shall be able to play 3 matches (3 x 100s) in a row when its batteries are fully charged
REQU 25	The robot shall be able to move forward on a straight line with a precision of 5cm after 1m
REQU 26	The robot shall be able to spin around a vertical axis with a precision of 1° (rotation <= 90°) or 2° (rotation > 90°)
REQU 27	The top speed of the robot shall be >= 0,5m/s
REQU 28	The robot shall reach the speed of 0,5m/s in less than 1s
REQU 29	When rolling at full speed (0,5 m/s), the robot shall be able to brake and stop in less than 200mm

« TDR (technical detail review) »

Une présentation qui résume le brainstorming et les différentes réflexions qui ont conduites au choix d'un élément du système. Elle doit résoudre un problème et justifier les choix du matériel du robot. Le document commence d'abord par l'exposition du problème à traiter : "input", ensuite on passe à la phase "Analysis" où on expose les solutions existantes, leurs avantages et inconvénients. Enfin, la présentation se clôt par une partie "Choice" où on explique la solution retenue pour notre problème initial.

2 - Analysis : What capacity ?

Energy (Wh) = tension (V) * capacity (Ah).

We need around 20Wh in order to stay on for at least 30min without recharging the battery.

Capacity = 20/12 = 1.7 Ah = **1700 mAh at least**





Figure 3 : Extrait de la TDR Batteries

« TP (test plan) »

Un tableau qui décrit la façon dont on va tester le système pour vérifier qu'il remplit les exigences des SRS. Il faut prendre des marges sur les besoins du SRS pour s'assurer de respecter le cahier des charges. Le document TP doit absolument être non ambiguë : le robot passe ou échoue le test. Il faut aussi penser à tout tester, même les points les plus évidents. Les SRS ont normalement étaient établis de manière propres et complètes, en les respectant scrupuleusement, on peut-être sur que le système remplira au moins la fonction voulue.

Un dernier point qui nous a été conseillé est que le seul moyen de s'assurer qu'un robot est fiable n'est pas de lui faire passer une fois un test mais de s'assurer que le robot est capable de réussir dix fois d'affilée le test.

	B	C	D	E
41				
42	REQU 24	The robot shall be able to play 3 matches (3 x 100s) in a row when its batteries are fully charged	See TP/R1 validation	See TP/R1 validation
43	REQU 25	The robot shall be able to move forward on a straight line with a precision of 5mm after 1m	Place the robot at rest in front of a virtual square (side 5cm) 1m away from the robot. Program the robot to move forward for 1m when the string is pulled. We consider that the position of the robot on the table is given by the vertical projection of the center of the axis of the motor wheels.	The robot starts, rolls to the square and stops within it.
44	REQU 26	The robot shall be able to spin around a vertical axis with a precision of 1° (rotation <= 90°) or 2° (rotation > 90°)	Place the robot at rest. Program the robot to spin around a vertical axis when the string is pulled. The rotation angle shall vary from 20° to 180° with intervals of 20°. Measure the actual rotation angle with a protractor	The correct angles are reached with a tolerance of +/- 1° if the rotation angle is <= 90°, and with a tolerance of +/- 2° if the rotation angle is > 90°
45	REQU 27	The top speed of the robot shall be >= 0.5m/s	Set a trapezoidal speed. With a chronometer (smartphone), measure the time taken by the robot to cover a distance of 1m after a 500mm run-up.	Duration <= 2s
46	REQU 28	The robot shall reach the speed of 0.5m/s in less than 1s	Set a trapezoidal speed. Let the robot accelerate along a straight line, parallel to a tape measure. Film the robot. Measure the speed of the robot between 1s and 1.2s after the start with videotracking : - If the recording has a temporal resolution of 25 fps, measure the distance covered during 5 frames - If it has a resolution of 50 fps, measure the distance during 10 frames	At speed >= 0.5 m/s i.e. The distance covered in 0.2s is >= 100mm

Figure 4 : Extrait d'un TP

« TR (test result) »

C'est une mise à jour du document *Test Plan* qui fait figurer les résultats des tests et les valident ou non. Ce document permet de garder une trace des résultats obtenus à chaque test, en effet, on ne peut pas parier sur le fait que le système passe tous les tests du premier coup. La possibilité de pouvoir comparer les résultats des différentes versions permet de revenir en arrière sur des modifications entraînant des pertes de performances.

« Budget »

Dans tout projet, on a en général des fonds en quantité limité, qui peut être grande mais reste limitée. Il est donc nécessaire dans le but de mener l'ensemble du projet à bien de prévoir et de garder à jour un budget précis. Le budget consiste donc en la liste de tout le matériel nécessaire à la construction du robot, avec les prix et le nombre de pièces tout en prévoyant des pièces de remplacements ainsi que des dépenses secondaires comme le déplacement à la coupe de France.

« Timeline »

L'un des points cruciaux de tout projet, un planning précis des échéances. La Timeline permet de prendre conscience d'éventuels retards accumulés et de se réorganiser pour être en mesure de tenir les échéances intermédiaires et mener le projet à bien en temps et en heure.

d. Dimensionnement

Pour que notre robot soit capable de remplir les objectifs que nous lui avons fixés, une des étapes clef consistait à choisir des composants bien dimensionnés.

« Motorisation »

Après avoir un peu réfléchi aux opérations à réaliser pendant la coupe et aux dimensions du plateau, nous avons estimé que notre robot devait avoir une vitesse de pointe de 1m/s (2m/s avec marge), être capable de freiner en moins de 0,1m lorsque lancé à 1m/s et avoir un couple le plus élevé possible au niveau des roues motrices puisque c'est cela qui permettra d'être précis sur des déplacements à faible vitesse. Pour des raisons pratiques, nous souhaitons que le moteur soit alimenté en 12V. Les calculs ont été réalisés en prenant des roues de 6 cm de rayon et une masse de 8kg.

On a $C = a_{\max} m \frac{R_{\text{roue}}}{2} = 0.35 \text{ Nm}$, en négligeant le moment d'inertie de la roue autour de son axe.

$$\omega_{\text{arbre moteur}} = 136 \text{ rpm}$$

Par ailleurs, plusieurs technologies de moteurs existent, on peut en particulier citer les moteurs *brushless* possédant des aimants permanents et les moteurs *brushed* avec des électro-aimants. Nous avons choisi d'avoir recours à des *brushed* moteurs car ceux-ci sont beaucoup plus classiques et facile d'emploi (au sens où on peut les piloter avec des branchements plus simples que les moteurs à aimants permanents). Notre choix s'est vite porté sur les moteurs de la marque suisse Maxon réputée pour leur robustesse et leur fiabilité. Le moteur est généralement associé à un réducteur et, un avantage de la marque Maxon est qu'elle permet de composer son assemblage à la carte pour répondre à ses besoins. Nous avons donc porté notre dévolu sur un moteur de puissance 14W et de voltage nominal de 12V qui associé à un réducteur de rapport 44 :1 et d'efficacité 0.71 remplit efficacement les prérequis comme le montre le tableau suivant. Les valeurs requises calculées sont suivies des valeurs nominales, soit les valeurs auxquelles le moteur est conçu pour fonctionner.

Vitesse de rotation du moteur	6002,415	rpm
Nominale	10700	rpm
Couple en amont du réducteur	1,12E-02	N.m
Nominal	1,46E-02	N.m
Puissance	6,72E+01	W

Figure 5 : Confrontation des valeurs nécessaires estimées aux valeurs nominales

« Alimentation »

L'objectif du dimensionnement de trouver le bon compromis énergie / poids, sans négliger la résistance à l'usure et le prix de la batterie. Il faut aussi tenir compte des questions de sécurité relatives aux risques d'explosion et de surchauffe, par exemple les règles de la coupe de France spécifient que toutes les batteries au lithiums doivent être dans des sacs ignifugés ce qui augmente beaucoup leur encombrement.

Une fois que nous avons lu les règles, choisi les moteurs et une vague idée des composants électroniques qui seront dans le robot, nous avons pu établir un cahier des charges pour la batterie :

- Le robot doit être capable de fonctionner au moins 30 min en continu sans recharger la batterie.
- La tension de sortie de la batterie ne doit pas être inférieure à 12V afin de pouvoir alimenter les moteurs
- La batterie doit avoir un courant de décharge suffisant pour fournir la puissance nécessaire aux moteurs dans les phases de démarrage et de freinage sans qu'elle ne soit endommagée.
- La batterie ne doit pas surchauffer à cause des courants de fonctionnement des moteurs.

Le robot a trois sources principales de consommation : ce sont les deux moteurs (14 W chacun), la carte Arduino (5 W) et la carte Raspberry (5W). Les autres actionneurs et capteurs sont directement alimentés par la carte Arduino, ou ne fonctionnent pas en même temps que les moteurs (nous pensons ici aux futures actions que le robot pourrait être conduit à faire dans les prochaines années). On a donc besoin d'une puissance totale de 38W ce qui correspond à une énergie de 20Wh pour un fonctionnement d'au moins 30 min sans recharge.

En choisissant une batterie de 14V (pour pouvoir alimenter sans problèmes des composants en 12V), on aura besoin d'une capacité d'au moins $20\text{Wh} / 14\text{V} = 1.5\text{Ah} = 1500 \text{mAh}$.

Quatre types de batteries sont disponibles sur le marché :

- Li-Ion : le meilleur ratio capacité/poids, mais une mauvaise résistance et un courant de décharge limité.
- Li-Po : bon ratio capacité/poids et un excellent courant de décharge.
- NiMh : un ratio raisonnable, une bonne résistance et un prix raisonnable.
- Pb : un mauvais ratio, mais très grande résistance à l'usure, facile à utiliser.

Nous avons d'abord opté pour une batterie Li-Po. Mais les contraintes logistiques (risque d'explosion et utilisation obligatoire d'un sac de sécurité pour la coupe) qui les accompagne nous a conduit à privilégier une batterie NiMh.

Une autre question s'est posée : veut-on avoir deux batteries distinctes, une pour la motorisation et une pour l'électronique ? Cela nous a paru une bonne idée pour deux raisons.

D'abord, les appareils électroniques (comme la carte Arduino) ont très peu de variations dans leur demande de courant contrairement aux moteurs : une batterie Li-ion serait alors

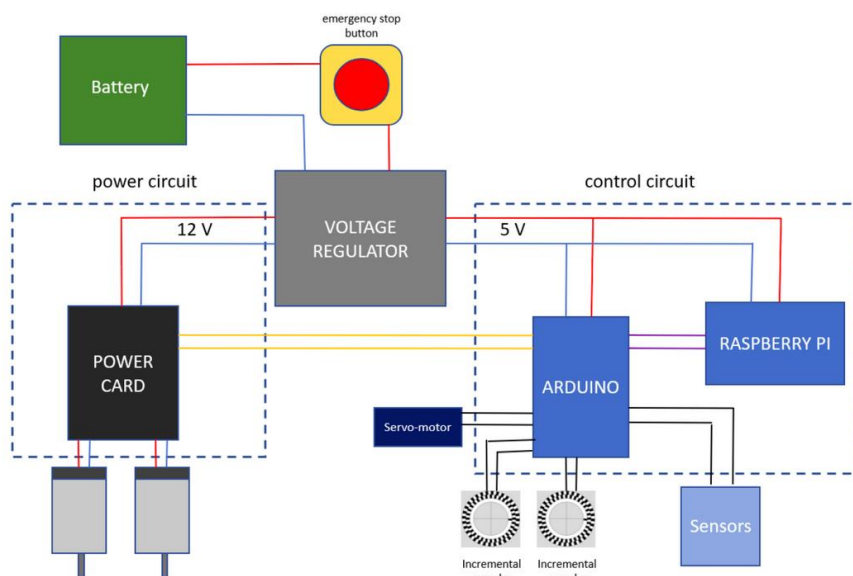


Figure 6 : Schéma électrique du robot

adaptée car elle délivre un courant stable. De plus, les courants de retour des moteurs peuvent causer le redémarrage du circuit de commande en plein milieu du fonctionnement du robot en cas de demande excessive, les données des cartes électroniques sont alors perdues.

Cependant, cette solution n'est pas parfaite non plus, notre tuteur nous a par exemple expliqué

avoir connu bon nombre d'équipes ayant perdu beaucoup de temps à réparer un robot dont le seul problème était qu'une des deux batteries était déchargée. Ainsi, un compromis qui a en plus l'avantage d'être moins gourmand en volume de robot est d'utiliser des régulateurs de tension qui stabilise les courants du circuit de commande.

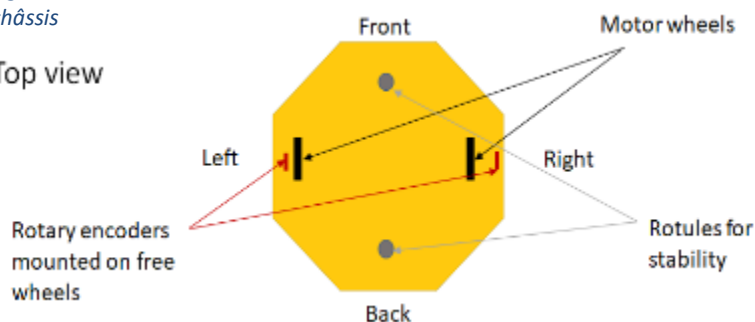
« Architecture »

Comme expliqué lors de la réunion de lancement, notre châssis aura une forme octogonale les roues motrices étant positionnées sur l'axe médian. Cette forme est un compromis entre le carré qui permet de positionner facilement les cartes et composants et le cercle qui permet de tourner sur place sans que le robot ne soit amené à heurter des obstacles. L'un des points clefs de notre stratégie consiste à s'appuyer sur une odométrie précise et fiable, à laquelle vient s'ajouter un système d'analyse d'images réalisées par une caméra fixe 1m au-dessus du plateau développé par le groupe informatique. Pour optimiser l'odométrie, nous avons décidé rapidement de séparer la motricité de l'encodage du mouvement pour s'affranchir du compromis adhérence-précision (la roue motrice veut avoir une grande surface de contact tandis qu'une roue codeuse optimale souhaite un contact ponctuel). Ce choix est en même temps un pari car si le principe est simple, la mise en œuvre se complique pour plusieurs raisons :

-Tout d'abord, avoir des encodeurs et des moteurs séparés signifie un espace occupé dans le robot plus important, il faut parvenir à

Figure 7 : Schéma du châssis

Top view



agencer les composants pour que l'ensemble de la motricité et de l'encodage tienne sur un axe ce qui contraint fortement la longueur maximale de chaque composant,

- Ensuite pour que l'odométrie soit fonctionnelle, il est primordial que les roues encodeuses soient en permanence en contact avec le sol mais aussi que l'axe des roues codeuses appartienne à tout instant au plan orthogonal au sol passant par l'axe des roues motrices, ce sans quoi les valeurs mesurées ne permettraient pas de connaître la position et l'orientation du robot à cause en particulier d'un patinage latéral des roues codeuses,

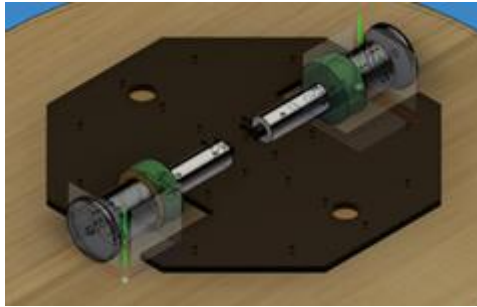


Figure 8 : design 3D de la base roulante

-Dernièrement, il faut parvenir à avoir un contact de frottement suffisant pour que la roue codeuse roule sans glisser.

Les calculs de dimensionnement de la précision des roues codeuses nous ont amenés à constater que l'erreur principal d'une odométrie était la quantification des rotations. Pour tenir nos engagements définis dans les SRS, nous avons observé que (pour une rotation de θ du robot, α l'angle de rotation de la roue codeuse) :

$$\theta = \frac{R}{l} \alpha \quad \text{donc} \quad \frac{u(\theta)}{\theta} = \sqrt{\left(\frac{\alpha}{R}\right)^2 u(R)^2 + \left(\frac{R\alpha}{l^2}\right)^2 u(l)^2 + \left(\frac{R}{l}\right)^2 u(\alpha)^2}$$

Nous avons originellement fait une application numérique sur un déplacement surévalué de coupe de robotique, en comptant que le robot devait parvenir à revenir connaître sa position finale avec une erreur de moins de 5cm. Pour une roue codeuse de 3 cm de diamètre, on trouve une précision d'au moins 1/1000^{ème} de tour pour l'encodeur. Seulement, il est nécessaire que ce soit la roue qui soit en contact avec le sol et non pas le corps de l'encodeur, la gamme de modèles que nous avons choisie (du fait de disponibilités d'approvisionnement) ayant un diamètre de 38mm, nous avons choisi de redimensionner la roue à 50mm de diamètre ce qui nous demande alors une précision d'au moins 1/1700^{ème} de tour pour garder la précision précédente. Les différentes problématiques qui ont conduit à ces choix seront détaillées plus précisément dans la suite de ce document.

Dans le but d'obtenir des roues codeuses remplissant au mieux nos besoins, nous avons opté pour les réaliser nous-mêmes à l'aide d'une impression 3d pour le corp et d'un joint torique pour le roulement. Le corps plastique sera peut-être à terme remplacé par des pièces en métal usinées pour avoir une longévité de plusieurs années si le système s'avère fiable. Pour parer aux deux derniers problèmes évoqués ci-dessus, nous nous sommes aussi lancés dans le design d'un amortisseur chargé de mettre en liaison glissière l'ensemble roue codeuse-encodeur avec le châssis. Cet amortisseur, à l'aide d'un jeu d'élastiques se chargera de maintenir en contact avec une intensité quasiment constante les roues codeuses sur le sol. Cette pièce étant l'élément crucial de notre pari de séparation encodage/motricité, nous avons décidés de ne pas la bâcler, c'est pourquoi dans un premier temps et pour ne pas retarder le reste du projet, nous avons réalisé en impression 3d une bride chargée de la même fonctionnalité mais mécaniquement plus simple puisque plutôt que de réellement amortir la roue, elle se contente par déformation mécanique de la maintenir en contact avec le sol.

« Capteurs »

L'utilisation de capteurs est cruciale pour tout robot devant détecter des obstacles, comme le robot adverse ou encore les bords de la table de jeu. Ainsi, il nous fallait un système de détection capable de mesurer assez précisément la distance entre notre robot et l'obstacle, surtout quand celui-ci est assez proche pour éviter les collisions. Rappelons que si les obstacles peuvent être détectés par la caméra fournissant en plus un positionnement absolu, certaines décisions comme éviter l'adversaire demande un temps de réponse très rapide qui peut ne pas forcément être fourni par la caméra (nous ne sommes en effet pas encore en mesure de nous prononcer sur les délais de communications entre le robot et le dispositif de caméra), d'où la nécessité de capteurs intégrés dans le robot.

Des capteurs simples fournissant la distance dans une direction nous ont cependant paru suffisant. En effet, avoir recours à un système type lidar permettant de cartographier en deux dimensions nous est apparu comme difficile à mettre en œuvre pour un gain d'une simple redondance d'information avec le système de caméra.

Il pourrait également être intéressant de mettre un capteur à l'arrière du robot pour mesurer la distance au mur, nous aviserons de cela une fois les premiers tests effectués.

Deux grandes familles de capteurs de distances existent : ultra-son et l'infrarouge. Leurs avantages et inconvénients peuvent être résumés dans le tableau ci-dessous :

	Infrarouge	Ultra-son
Précision	Précision meilleure	Précision plus faible
Directivité du signal émis	Signal plus ciblé : moins de perturbations à cause du sol ou d'autres obstacles	Signal moins directif : peut détecter des obstacles pas nécessairement devant le robot
Interférences	Peut être perturbé par un signal infrarouge de caméra ou du robot adverse (peu probable:signaux sont de direction précise).	Facilement perturbable par un autre émetteur ultra-son si les fréquences coïncident.

L'équipe de la X2016 avait utilisé des capteurs infrarouges de STMicroelectronics, mais nous adopterons une autre technologie pour les raisons suivantes :

- La communication entre ST Microelectronics et l'électronique Arduino est compliquée.
- Une carte ST ne peut contrôler que 3 capteurs à la fois (système peu flexible).

- Les protocoles de communication à mettre en oeuvre fond que les données des capteurs ne peuvent être écrites que sur 4 bits (16 valeurs) ce qui est une perte de précision non négligeable.

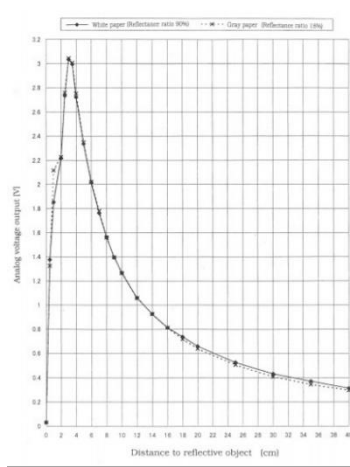


Figure 9 : tension délivrée par le capteur selon la distance à l'obstacle

En conséquence, nous avons choisi d'utiliser le capteur infrarouge SHARP GP2Y0A41SK0F ci-dessous, qui est capable de mesurer un obstacle de 4 à 30 cm. Ce capteur délivre une tension qui dépend de la distance mesurée selon le graphe caractéristique suivant :

Toutefois, nous remarquons que cette courbe n'est pas bijective : une tension de 2,2V peut correspondre à deux distances différentes, ce qui peut induire des erreurs de déplacement. Or cette courbe devient bijective à partir d'une distance de 3 cm à peu près. Ainsi pour passer outre ce problème, nous avons décidé de placer le capteur 3cm à l'intérieur du robot, ce qui nous permet de retrouver une bijection entre tension délivrée et distance lue.

II. Des problèmes inattendus

a. Des encodeurs trop précis

Comme expliqué plus haut, les calculs effectués lors de la conception montraient que nous avons besoin de roues encodeuses précises au moins au 1/1700^{ème} de tour. Notre budget le permettant, nous avons commencé par acheter des encodeurs précis au 1/5000^{ème} car leur encombrement est exactement le même, et nous pensions ainsi améliorer encore la précision du placement du robot. Toutefois, un problème auquel nous n'avions pas pensé s'est alors posé : les roues du robot tournant à environ 10 tours/seconde, les deux encodeurs produisaient, à eux deux, 100.000 « clicks » par seconde, et cette fréquence est trop élevée pour la carte Arduino Mega, qui ratait des clicks et n'arrivait pas à suivre la cadence.

Nous avons d'abord essayé de faire malgré tout fonctionner ces encodeurs en remplaçant la carte Arduino par une carte Raspberry Pi, mais les résultats n'ont pas été concluants (voir section iv. ci-dessous).

Nous avons donc revu à la baisse notre envie d'une précision trop poussée, en optant pour des encodeurs à deux canaux en quadrature 1/600. Ces deux canaux peuvent prendre la valeur 0 ou 1 (0 ou 5V), et le changement s'opère 600 fois par tour pour chaque canal. Grâce au fait que les deux canaux sont décalés de $\pi/2$, cela permet de connaître le déplacement algébrique (valeur absolue + sens de rotation), et en fait cela permet d'atteindre une précision allant jusqu'à 1/2400^{ème} de tour, en détectant la montée ET la descente des deux canaux simultanément, retrouvant ainsi une précision suffisante vis-à-vis des calculs de dimensionnement.

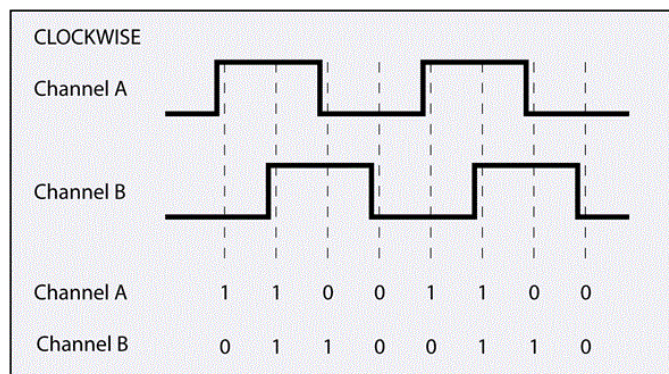


Figure 10 : Schéma du fonctionnement d'un encodeur rotatif à deux canaux en quadrature

Ce problème, que nous avons en définitive surmonté (les encodeurs fonctionnent maintenant très bien), nous a appris les dangers du surdimensionnement, et l'importance de rester « humble » en quelque sorte : même si le budget nous permet d'acquérir un composant très performant, c'est en définitive la pièce la plus limitante qui impose sa performance au système.

b. Résistance pull-up

Un encodeur rotatif possède deux canaux de sortie, qui oscillent entre la valeur 0 et la valeur 5V (interprétables comme des 0 ou 1 numériques par la carte Arduino). Toutefois, quand nous avons testé notre premier encodeur, nous n'avions pas du tout la sortie espérée : la tension en sortie était quasi constante et autour de 2V. Après de nombreuses tentatives infructueuses, c'est finalement notre sponsor et notre tuteur qui nous ont mis sur la piste : en

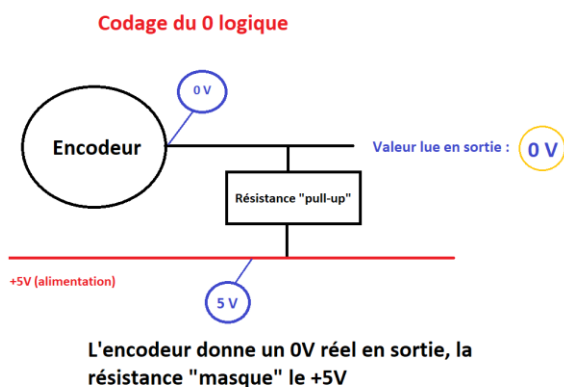
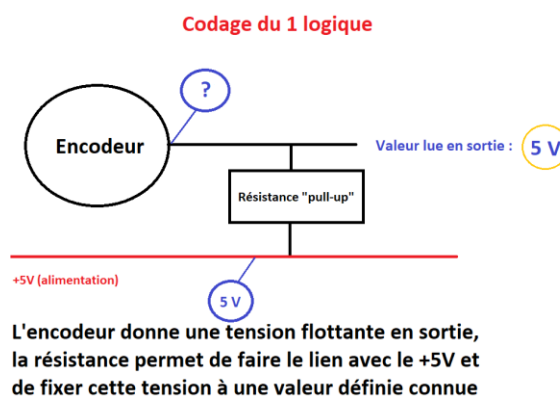


Figure 11 : Schéma logique d'un branchement en pull-up



réalité, l'encodeur rotatif donne en sortie une tension de 0V pour coder 0, et une tension non définie (un fil « flottant » relié à rien) pour coder 1. Il faut donc rajouter entre la sortie et le +5V de l'alimentation une résistance dite « pull-up » (voir figure 11).

Après quelques tâtonnements, nous avons trouvé la valeur idéale de cette résistance (environ 150 Ω), et l'encodeur nous donne maintenant des 0 et des 1 logiques parfaitement reconnaissables.

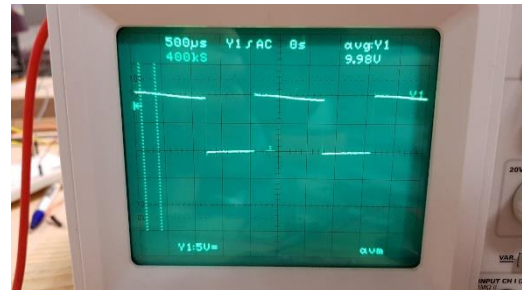


Figure 12 : Tension de sortie de l'encodeur (le 1 logique est à 10V car l'encodeur est alimenté en +10V)

c. Arduino ou Raspberry Pi ?

Comme expliqué plus haut, nous avons un temps envisagé de remplacer la carte Arduino Mega par une Raspberry Pi pour la lecture des encodeurs.

La difficulté liée à la lecture de l'encodeur est que le programme doit en permanence être « attentif » aux encodeurs : on ne sait jamais quand les roues vont tourner. Le programme doit donc guetter le changement de valeur d'un des deux canaux même lorsqu'il est en train d'effectuer un autre calcul. Heureusement, les cartes Arduino et Raspberry Pi implémentent une solution à ce problème : elles possèdent des pins capables de générer des « interruptions » : en permanence, le programme va guetter un changement de valeur de la tension mesurée par ces pins, et lorsqu'un changement se produit, le programme s'interrompt pour traiter une petite procédure (ici, incrémenter le compteur de tours de la roue droite ou gauche de +1 ou -1).

En revanche, la fréquence maximale de ces interruptions est limitée, et comme nous l'avons vu, la carte Arduino n'arrivait pas à suivre la fréquence de 100kHz que nous lui imposions. Nous avons donc essayé de la remplacer par une Raspberry Pi (la fréquence du processeur d'une Arduino Mega est de 16MHz, alors que celle d'une Raspberry dépasse le GHz, donc nous pouvions raisonnablement espérer que la Raspberry arrive à suivre la cadence).

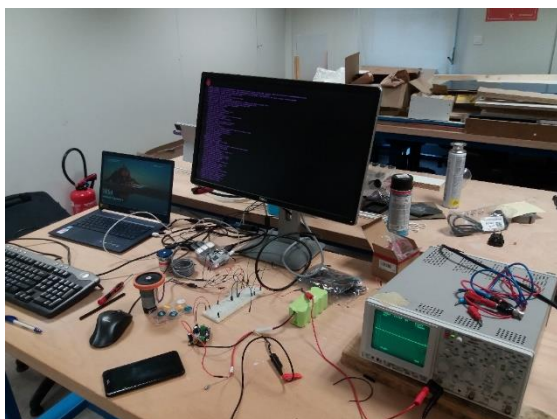


Figure 13 : Montage testant la capacité de la Raspberry Pi (données visualisées sur le grand écran)

Malheureusement, la puissance supérieure de la Raspberry Pi sert plutôt à faire tourner un système d'exploitation. Ses pins d'interruption ne sont en fait que peu optimisées et, contre toute

attente, encore moins performants que ceux de la carte Arduino. Nous avons donc dû renoncer à l'utiliser, et à redimensionner nos encodeurs en les exploitant mieux pour atteindre la précision nécessaire sans submerger notre carte Arduino d'interruptions.

d. Des délais d'approvisionnement sous-estimés

Si une fois l'ensemble du design contrôle réalisé nous nous sentions prêts à attaquer le côté pratique du sujet, il nous est rapidement apparu que nous avions gravement sous-estimé une étape pouvant être très limitante, à savoir les approvisionnements en matériels. En effet, les dates de la coupe et les délais du psc nous ont parfois contraints à changer nos choix de composants non pas pour des problèmes de prix mais de disponibilités. Nous pouvons par exemple ici penser au choix des encodeurs où la même précision de 600 points par tours était possible sur un dispositif d'encombrement deux fois moins important mais indisponible avant fin février. D'autres mauvaises surprises nous sont aussi arrivées sur des produits normalement disponibles rapidement qui ne sont pas encore arrivés au moment de l'écriture de ce rapport.

Avec un peu de recul, nous aurions dû tenir à jour un document de suivi des pièces pour avoir une trace de ce que nous avons déjà reçu et en attente, ainsi nous aurions été plus à l'abri d'éventuels oublis de composants qui auraient pour effet de retarder encore plus l'avancement du robot. Par ailleurs, ces délais d'achats nous ont fait réaliser l'importance de savoir s'avancer et prototyper ses pièces en amont pour pouvoir intégrer les composants reçus au prototype dès leur réception. Pour cela, un design 3 dimensions très propres et prenant en compte les caractéristiques techniques des composants sélectionnés est nécessaire. Nous avons cependant réalisé qu'il était assez difficile de faire mouche du premier coup lorsque l'on est néophyte. Notre premier design de châssis en impression laser a par exemple dû subir un certain nombre de corrections à la scie et à la lime avant de pouvoir accueillir la première base roulante que nous ne manquerons pas d'intégrer dès l'impression laser la prochaine fois.