

UNIVERSITÉ DE SHERBROOKE  
Faculté de génie  
Département de génie électrique et génie informatique

## **BILAN PERSONNEL**

Conception agile et ouverte en robotique  
GRO400

Présenté à  
L'équipe professorale de la session 4

Présenté par  
Loïc Boileau-boil3205

Sherbrooke – 20 avril 20

## Contribution personnelle technique

Tout au long du projet de session, j'ai pu être en charge du développement de l'interface graphique. De plus, une fois la mécanique installée, il y a eu aussi l'aspect de programmation des moteurs sur la carte OpenCR. À cause du virus, j'ai dû prendre en charge le robot vu que j'étais celui qui développait la structure de programmation. Cela m'a permis d'avoir du temps pour développer le contrôle complet du robot.

Pour commencer, mon mandat était d'établir un environnement de programmation pour que l'on puisse commencer à programmer. Vu les contraintes du cahier de charge, il était clair que nous devions programmer sur un PI. C'est pourquoi j'ai décidé que le programme principal de contrôle serait intégré à l'interface graphique et le tout serait en python. Il a donc fallu que je trouve une librairie pour coder l'interface graphique. J'ai pu choisir la librairie PySide (de QT en python) vu qu'elle venait avec une interface graphique pour designer l'interface graphique du projet.

J'ai pu donc commencer à développer l'interface jusqu'à ce que j'obtienne quelque chose d'utilisable avec plusieurs fonctionnalités (envoyer des commandes avec des arguments). Cela mène au premier « release ».

Ensuite, mes partenaires de travail ont pu installer les moteurs et commencer leur code de contrôle. J'ai, par la suite, établi la communication entre le PI et l'OpenCR en développant un protocole de communication entre les deux modules. Celui-ci consiste à une communication en port sériel avec 4 octets. Pour ce faire, j'ai dû programmer en C++ pour établir la boucle d'attente de commande du OpenCR et en Python pour que le Pi puisse envoyer le bon message à partir des commandes de l'interface graphique.

C'est en décidant de l'architecture des modules et en terminant l'interface graphique (ajouter de nouvelles commandes est très simple) que j'ai pu finaliser l'environnement de programmation. J'ai aussi conçu une méthode pour envoyer un message au OpenCR et faire bouger les moteurs à leur guise.

Pour finir, j'ai tenté, avec l'aide de Katryne, d'implémenté le contrôle cartésien du robot, mais ce ne fut pas un succès; il aurait fallu plus de temps.

## Nouvelles connaissances et connaissances futures

Dans ce projet, j'ai pu acquérir plusieurs nouvelles connaissances. De même qu'il y en a que j'aurais aimées acquérir.

Pour commencer, je n'avais jamais programmé une interface graphique au complet à partir de 0. Il a fallu que je cherche des librairies et que je choisisse une d'elle. Ensuite, que je compose les méthodes pour

faire interagir les composantes de l'interface. Et le tout en ajoutant un module de communication en série dont je n'y connaissais rien. Cela m'a permis de comprendre la représentation d'un message en binaire (octet) et les fonctionnalités du port en série. Aussi tout ce qui correspond aux fonctionnalités de QT (les fonctions « slot » et les signaux).

La programmation des moteurs a été mystérieuse vu que la documentation sur les dynamixels n'était pas claire. J'ai pu donc apprendre à configurer un moteur. Cependant, j'aurais aimé ça joué un peu plus avec les moteurs, leurs vitesses et leurs PID. Ça n'a pas été le cas vu que modifier les paramètres semblait complexe et aurait apporté plus de problèmes que de solutions. J'ai pu, aussi, apprendre à programmer en temps réel sur l'OpenCR.

En ce qui concerne les outils de travail, j'ai pu apprendre à bien utiliser GitHub, ce qui a permis de toujours avoir une version du code à jour.

J'aurais aimé en apprendre plus sur les approximations quadratiques pour être capable de rendre lisse le déplacement des moteurs vu qu'ils se déplacent en mouvements saccadés.

## Autoévaluation

Par rapport à mes partenaires, je pense que j'ai effectué très grande partie du travail technique du projet. Cependant, j'aurais pu m'investir un peu plus en ce qui concerne l'utilisation des outils mise à disposition pour coordonner le projet. Je n'ai pas tout le temps maintenu à jour mes tâches sur ZenHub ce qui pouvait mener à de la confusion dans l'équipe. Il est vrai que j'étais seul responsable de mes tâches, mais il est important pour un ingénieur de tout le temps laissé des traces du travail effectués que ce soit pour le client ou pour le reste de l'équipe.

À recommencer ce projet, je m'investirais autant dans le développement technique (ce que j'ai beaucoup apprécié) mais je prendrais quelqu'un d'autre pour m'aider ce qui me permettrait d'alléger ma charge de travail et de maintenir plus à jour le tableau des tâches.

Je pense que pour les réunions d'équipe je participais assez bien.