



Cours: LOG791, session Hiver 2020

**CONTRÔLE AUGMENTÉ POUR LA MOBILITÉ DU ROBOT
MARKHOR
CONÇU PAR LE CLUB CAPRA**

Travail remis à:

Alain April

Remis par:

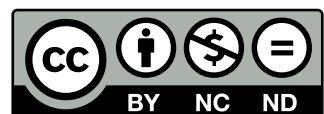
Charles Giguère - GIGC22109306,
Marc-Olivier Bélisle - BELM05039606,
Ludovic Vanasse - VANL21109404

Date de remise: 14 avril 2020



C.Giguère, M-O Bélisle, L. Vanasse, 2020

Ce document est mis à disposition selon les termes de la licence Creative Commons “Attribution - Pas d’utilisation commerciale - Pas de modification 3.0 non transposé”.



Résumé

Ce projet a comme objectif d'améliorer la mobilité du robot Markhor, fait par le club Capra, en utilisant de l'apprentissage machine pour faciliter l'opération des quatre chenilles motrices pouvant être inclinées en fonction des besoins. Plusieurs solutions ont été analysées pour arriver à la conclusion d'utiliser du Q-learning combiné avec un CNN pour filtrer les données d'entrées. De plus, les modules de contrôle du robot ainsi qu'une interface utilisateur ont été implémentés. Ce robot est fait avec le Robot Operating System (ROS).

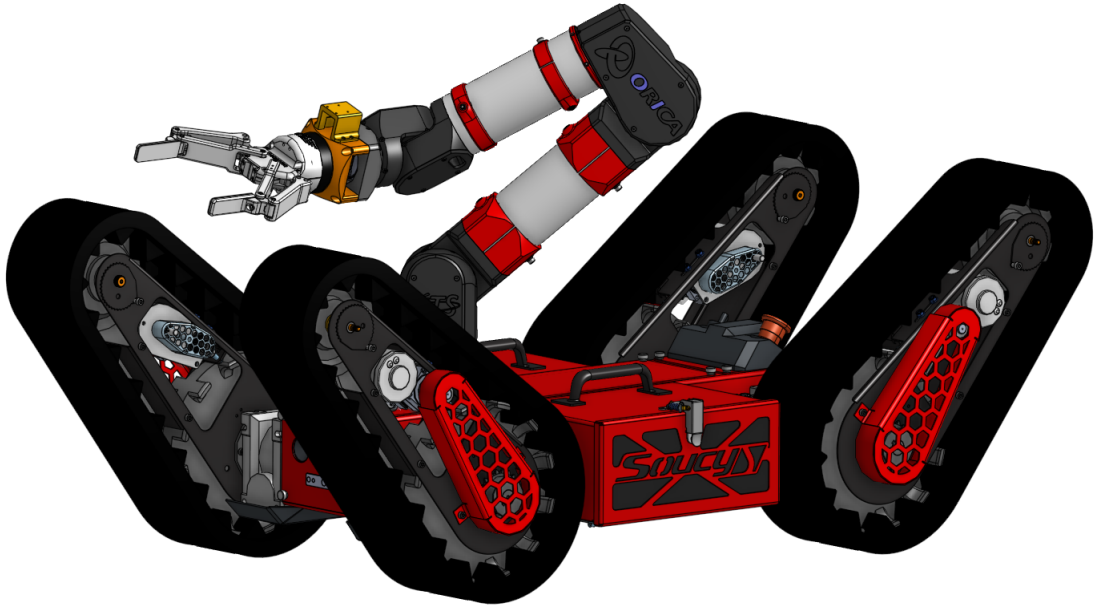


Table des matières

1	Glossaire	4
2	Présentation du Projet	5
2.1	Contexte et problématique	5
2.2	Objectifs du projet	5
2.2.1	Phase d'exploration de solution	5
3	Méthodologie et organisation du travail	6
3.1	Composition de l'équipe et rôles	6
3.2	Livrables	6
4	Présentation de la solution	7
4.1	Fonctionnalités à réaliser (théorie)	7
4.1.1	Collecte des données	7
4.1.2	Transformation des données	7
4.1.3	Traitement algorithmique	7
4.2	Description de la solution choisie (pratique)	7
4.2.1	Données utilisées	7
4.2.2	Vecteur d'apprentissage	8
4.2.3	Préparation de l'expérimentation	8
4.3	Expérimentation	8
4.3.1	Exécution et description des essais	8
4.3.2	Résultats obtenus	8
5	Améliorations et suite du projet	9
5.1	Limites du système et solution	9
5.2	Améliorations suggérées	9
6	Bibliographie	10
7	Annexe 1- Questions concernant les qualités de l'ingénieur	11
7.1	Évaluez l'étape de Formulation du problème	11
7.2	Évaluez l'étape de Formulation de solutions	11
7.3	Évaluez les Communications	11
7.4	Évaluez votre approche de gestion des Enjeux Environnementaux	12
7.5	Évaluer votre capacité d'effectuer une Recherche judicieuse de solutions	12

1 Glossaire

Terme	Définition
ROS	Robot Operating System. Librairie de robotique.
Q-Learning	Type d'apprentissage machine
Robocup	Compétition internationale de robotique
URDF	Unified Robot Description Format
Gazebo	Logiciel de simulation de robot ROS
IMU	Inertial Measurement Unit, Unité qui permet de connaître la vitesse et l'orientation dans l'espace
CNN	Convolutional Neural Network, Réseau Neuronal Convolutionnel
LSTM	Long Short-Term Memory, architecture de réseau neuronal
IA	Intelligence artificielle

Table 1: Glossaire

2 Présentation du Projet

2.1 Contexte et problématique

Capra participe à la Robocup Rescue, une compétition internationale de robotique pour des missions de recherche et de sauvetage dans des terrains sinistrés. Dans cette compétition, le robot doit naviguer dans un terrain accidenté pour trouver une cible et faire le plus d’aller retour possible. Notre performance est calculée à partir de l’efficacité et la vitesse du robot. Donc, il faut maximiser notre pointage en minimisant le travail du pilote.

2.2 Objectifs du projet

Après une analyse du robot, il a été observé que la majorité du temps perdu était dans le contrôle de l’inclinaison des chenilles. Donc, notre objectif est de trouver une solution afin améliorer l’efficacité et rapidité du contrôle pour l’inclinaison des chenilles.

2.2.1 Phase d’exploration de solution

D’abord, l’inclinaison des chenilles étaient contrôlées à partir d’une manette de Xbox. Notre première idée était de connecter des chenilles ensemble pour contrôler seulement deux chenilles plutôt que quatre chenilles. Cette idée ne fonctionnerait pas puisque les terrains sont de telle sorte que chaque chenille doit s’incliner, selon des angles différents indépendamment l’une de l’autre. Une autre solution était d’acheter ou construire une autre manette qui permettrait un meilleur contrôle. L’idée a été refusée puisque la manette de Xbox a été choisi d’abord et avant tout pour sa simplicité. Une de nos grandes forces étaient la simplicité de notre interface utilisateur et notre but est de garder cela. Il semblerait qu’il n’était pas possible de rendre le contrôle de l’inclinaison plus simple sans rendre la manette plus complexe. Nos recherches se sont portés alors sur l’apprentissage machine. Notre but était d’avoir une IA qui puisse elle-même contrôler l’inclinaison des chenilles sans l’aide du pilote. En plus de grandement aider notre pilote, cela permet au robot d’être semi-autonome, ce qui augmente notre pointage dans la compétition. Après avoir cherché une solution préexistante, nos recherches ont abouti sur des documents de recherche et des thèses à ce sujet.

3 Méthodologie et organisation du travail

3.1 Composition de l'équipe et rôles

Nom	Rôle	Responsabilité
Marc-Olivier Bélisle	Chef logiciel	Implémentation de l'apprentissage machine
Ludovic Vanasse	Programmeur	Intégration avec le robot
Charles Giguère	Programmeur	Développement de l'interface utilisateur

3.2 Livrables

- Pour simplifier le contrôle du robot, l'interface utilisateur devait être capable d'afficher le modèle 3D du robot pour pouvoir facilement visualiser les positions des quatre chenilles. De plus, c'est l'interface utilisateur qui est responsable de lire la manette de Xbox et de transformer les contrôles en données faciles à comprendre pour le robot. Plusieurs fonctionnalités ont été ajoutées qui facilitent le développement tel qu'un terminal intégré. Interface Utilisateur
- Un des livrables du projet était d'améliorer le contrôle du robot à partir de l'infrastructure logicielle ROS. Une des fonctionnalités de *framework* ROS est d'utilisation des interfaces physiques virtualisé pour être contrôlés avec différents contrôleurs. Ces contrôleurs implémentent des règles de contrôle pour manier le robot correctement. Dans notre cas, il était question d'utiliser les chenilles du robot avec un contrôleur pour moteurs différentiel. Cela permet de prendre en ingérant deux vecteurs de direction et de vitesse et les convertis en valeur décimale pour les deux paires de chenilles.
Markhor Control
Talon SRX Hardware Interface
- Un des livrables du projet était de faire le URDF du robot pour la simulation. Le URDF permet de décrire à Gazebo comment le robot est supposé être affiché et interagir avec le monde.
Markhor Description

4 Présentation de la solution

Pour permettre un contrôle intelligent de l'inclinaison des chenilles, il faut entraîner une IA pour s'occuper de déterminer l'inclinaison optimale sans l'aide du pilote.

4.1 Fonctionnalités à réaliser (théorie)

Pour réaliser cela, il faut entraîner deux IA. D'abord, il faut entraîner un CNN qui va permettre de transformer l'image de profondeur de la caméra du robot en un tableau de valeurs. Ensuite, ce tableau sera concaténer avec les valeurs du IMU et l'inclinaison courante des chenilles. Finalement, le robot sera entraîné, dans notre simulation, en utilisant du Q-Learning.

4.1.1 Collecte des données

Pour la collecte de données, le robot sera simulé dans Gazebo, un simulateur pour des robots utilisant ROS. Durant la simulation, les données seront collectées en utilisant un ROSBag, un outil de ROS qui permet de collecter les valeurs du robot et de les rejouer dans l'ordre.

4.1.2 Transformation des données

Pour augmenter la rapidité de l'entraînement, un script sera utilisé pour prendre seulement les segments les plus importants de la simulation.

4.1.3 Traitement algorithmique

Les images prises dans la simulation sont utilisées pour entraîner le CNN. Quand le CNN est prêt, on peut commencer à entraîner le Q-Learning. Le Q-Learning prend en entrée la sortie du CNN, les valeurs du IMU et l'inclinaison courante des chenilles et crée un tableau de coefficient. Ce tableau permet de calculer la prochaine action que le robot doit prendre à partir des entrées du Q-Learning.

4.2 Description de la solution choisie (pratique)

4.2.1 Données utilisées

Les données utilisées sont l'image de la caméra de profondeur, les valeurs du IMU, l'inclinaison des chenilles et pour le CNN les commandes de l'opérateur pendant qu'il contrôle le robot.

4.2.2 Vecteur d'apprentissage

D'abord, le CNN transforme l'image de profondeur et les commandes du pilote en un tableau de 60 éléments. Sur ce tableau, on ajoute les 6 valeurs du IMU et plus les quatre valeurs d'inclinaisons des chenilles.

4.2.3 Préparation de l'expérimentation

Pour faire l'expérience, il faut préparer le simulateur. Celui-ci a besoin d'un fichier SDF qui décrit le terrain et les paramètres du monde simuler. De plus, le simulateur a besoin d'un fichier URDF. Celui-ci décrit les composantes du robot et comment il interagit avec ses composantes. Il faut aussi faire l'interface de nos moteurs pour permettre de standardiser les commandes du contrôleur. Aussi pour faciliter l'entraînement du CNN, il faut faire un conteneur pour permettre de facilement déployer la simulation sur plusieurs machines et permettre le UI de se connecter à ceux-ci. En dernier lieu, il faut configurer les tâches et l'environnement de la librairie "OpenAI Ros" pour expérimenter notre Q-Learning.

4.3 Expérimentation

Notre travail a permis de simuler le robot dans Gazebo avec le contrôle des chenilles et de l'inclinaison de ceux-ci sur un ordinateur local.

4.3.1 Exécution et description des essais

La plus récente version de Gazebo n'avait pas le support pour simuler la force sur une surface de contact. Il a fallu installer une version expérimentale de Gazebo pour permettre une simulation réaliste des chenilles. Dans nos premières tentatives, les chenilles semblaient bien se comporter. Par contre, les chenilles du robot n'était pas fixe et il bougeait avec l'inertie du robot, ce qui n'était pas vrai avec le robot réel. Donc, on a ajouté de l'amortissement sur les joints pour régler le problème.

4.3.2 Résultats obtenus

En contrôlant le robot manuellement avec un clavier et en envoyant une variétés commandes différentes dans plusieurs terminaux, il est capable de monter les escaliers et surmonter plusieurs obstacles. Il peut bouger lorsque les chenilles sont partiellement en contact avec le sol et le robot peut entrer en collision avec le terrain.

5 Améliorations et suite du projet

La prochaine étape était de faire un conteneur pour la simulation qui va permettre de collecté des données pour le CNN. Ensuite, configurer "OpenAI ROS" pour entraîner le robot. La dernière étape serait d'expérimenter plusieurs stratégie de Q-Learning et de pointage pour trouver la solution optimale.

5.1 Limites du système et solution

L'entraînement et la simulation sont deux applications très demandantes pour un ordinateur. Il est donc préférable de l'entraîner sur des ordinateurs très puissants ou avec des ordinateurs dans les nuages.

5.2 Améliorations suggérées

Dans la thèse, l'auteur étudie l'idée d'une solution avec une architecture LSTM. Cette solution semble donner de bons résultats et cela permettrait d'entraîner seule une IA.

6 Bibliographie

- Azayev, T. (Juin, 2017), Czech Technical University in Prague, Prague, Deep Learning for Autonomous Control of Robot's Flippers in Simulation <https://core.ac.uk/download/pdf/151072651.pdf> Consulté pendant l'hiver 2020

7 Annexe 1- Questions concernant les qualités de l'ingénieur

7.1 Évaluez l'étape de Formulation du problème

Comment avez-vous réussi à bien décrire le problème, à le résoudre en tenant compte des besoins et contraintes identifiées dans votre projet.

On a d'abord pris les contraintes de notre compétition. Le terrain est dépendant du type d'épreuve. Il y a quatre type d'épreuves, "Maneuvering", "Mobility", "Dexterity" et "Exploration". Les quatre type d'épreuves ont leurs propres enjeux et leurs difficultés, mais il y a généralement toujours une chose en commun, le déplacement. Généralement, chaque épreuve a comme étape de trouver la victime, collecté de l'information et de faire le plus d'aller retour. Donc la majorité de notre compétition demande de se déplacer rapidement et efficacement dans des terrains accidentés. De plus, le robot ne doit pas toucher les murs et doit rester stables pour éviter de se briser. Donc, le pilote doit être rapide, mais veiller à ne pas faire d'erreurs. Pour ces raisons, le pilote est forcé de ralentir et prendre son temps. Ce qui a un impact sur notre pointage.

7.2 Évaluez l'étape de Formulation de solutions

Comment avez-vous réussi à décrire des solutions possibles et choisir celle qui a été préférée pour votre projet.

Une grande partie des commandes envoyés sont en rapport à l'inclinaison des chenilles. D'abord, on a cherché une formule mathématique simple pour répondre à nos besoins. On a vite réalisé qu'il serait difficile de simplement utiliser une formule. On s'est alors tourné vers l'apprentissage machine. Notre équipe a fait des recherches pour trouver des solutions déjà existante ou des recherches dans le sujet. Notre équipe n'a pas trouvé une solution déjà existante, mais on a trouvé plusieurs papiers de recherches et thèses dans le contrôle intelligent des chenilles.

7.3 Évaluez les Communications

Comment avez-vous communiqué avec le client/utilisateurs et entre vous durant ce projet.

Le client de notre projet est le club Capra et la majorité des membres de l'équipe sont aussi des membres exécutifs du club, donc la communication était simple. Pour communiquer, notre équipe a principalement utilisé la plateforme Discord pour écrire et parler à distance. On a aussi fait plusieurs rencontres en face à face dans les locaux du club. De plus, notre équipe avait régulièrement des rencontres dans un local du département de génie logiciel avec notre superviseur.

7.4 Évaluez votre approche de gestion des Enjeux Environnementaux

Expliquez comment vous avez tenu compte de la réduction de votre empreinte écologique durant le projet.

Ce qui impact le plus notre empreinte écologique, c'est la consommation électrique du robot. Plus le robot fait de calcul, plus le robot consomme de l'électricité. Donc, il faut optimiser notre solution pour notre empreinte écologique, mais aussi pour notre limitation en temps de CPU. Étant donné que la performance de la solution est un des points les plus importants dans notre recherche, on a indirectement travaillé sur notre empreinte écologique.

7.5 Évaluer votre capacité d'effectuer une Recherche judicieuse de solutions

Évaluer votre capacité d'effectuer une Recherche judicieuse de solutions et de la nécessité de compréhension de nouveaux concepts pour proposer des solutions pour ce projet.

À partir de nos recherches, notre équipe a regroupé une dizaine de solutions en apprentissage machine. On a rejeté ceux qui avaient des résultats non-concluants pour se concentrer sur les autres. Une thèse en particulier a été choisie. Cette thèse est arrivée à des résultats beaucoup plus intéressants et elle utilise les mêmes technologies que notre robot. On a validé le tout avec notre superviseur pour valider notre solution et commencer à implémenter notre solution. La thèse utilisait des nouveaux concepts qu'on n'avait jamais utilisés comme un CNN et le Q-Learning. Pour connaître la base, notre superviseur a proposé à notre équipe un exercice du MIT sur le Q-Learning. De plus, notre équipe a travaillé avec une personne externe et notre superviseur pour comprendre les bases du CNN.