Utilisation de la robotique et de l'intelligence artificielle pour améliorer le bien-être des vaches laitières au Canada

Laura Jouvet
Robotique - Promotion 2025

Tuteur ENSTA : Luc Jaulin
Tuteurs UQÀM : Abdoulaye Baniré Diallo et Hayda Almeida

Projet de fin d'études – Mars-Août 2025
Rapport non confidentiel

Remerciements

Je tiens à remercier la chaire Well-E de m'avoir donné l'opportunité d'effectuer mon stage au sein de leur équipe. Tout d'abord, je tiens à remercier le Professeur Abdoulaye Baniré Diallo de m'avoir fait confiance pour travailler sur les bases de ce projet de déploiement d'un robot dans les exploitations laitières qui s'annonce très prometteur. Ensuite, merci à Hayda Almeida d'avoir toujours pris le temps de m'accompagner dans ce projet et de réfléchir avec moi lorsque nous rencontrions des problèmes. Merci également à Voncarlos Marcelo de Araùjo, mon chef de groupe pour mon projet secondaire qui a su bien répartir les tâches et assurer la communication au sein du groupe pour mener le projet à bien. Je tiens également à remercier Marjorie Cellier du laboratoire de sciences animales pour son apport de connaissances sur les vaches qui m'a permis d'adapter au mieux mon projet aux besoins de la chaire. Merci également à Thomas Gisiger et Karen Reynard pour leur disponibilité et leur organisation qui ont favorisé le bon déroulement de mon stage ainsi que l'avancement de mon projet. Enfin, merci à tous les étudiants et membres de la chaire qui ont pris le temps de m'expliquer leurs projets tous très intéressants et de répondre à mes questions. Grâce à eux et aux activités organisées en dehors du travail, je me suis toujours sentie à l'aise au sein du groupe, ce qui m'a permis de bien m'intégrer et d'apprendre beaucoup sur différents sujets.

Résumé

Ce rapport présente mon projet de fin d'études que j'ai réalisé durant 5 mois au sein de la Chaire Well-E à l'Université du Québec à Montréal au Canada. Le but de mon projet était de développer une simulation d'un robot que la chaire utilisera à terme dans les exploitations de vaches laitières afin de surveiller le comportement des vaches pour améliorer leur bien-être physique et mental. Ce stage a impliqué le choix d'un robot, d'un simulateur et d'un environnement de simulation, l'intégration des algorithmes de la chaire à l'architecture ainsi que le développement et l'intégration d'algorithmes de navigation du robot choisi dans le milieu d'application. Ce projet multidisciplinaire a englobé de la simulation, du développement d'une architecture en ROS2, de l'asservissement visuel, du machine learning, de la gestion de projet autonome ainsi que de la collaboration au sein d'un groupe de recherche. Enfin, il a contribué à l'amélioration et à la validation de mes compétences en vue de mon futur professionnel.

Table des matières

1 Introduction

1.1 Présentation de la chaire Well-E

J'ai réalisé mon stage de fin d'études au sein de la Chaire Well-E (WELfare Longevity Electronic), la première chaire de recherche de son genre en bien-être animal et en intelligence artificielle. Son but est d'améliorer le bien-être et de prolonger la longévité des vaches laitières, tout en soutenant les producteurs face aux défis économiques et environnementaux actuels. Elle s'inscrit dans un contexte où les attentes sociétales en matière de production animale responsable sont de plus en plus fortes, tandis que les coûts de production augmentent sans réels gains sur les prix de vente.

FIGURE 1 – Logo de la chaire Well-E

Pour répondre à ces enjeux, le laboratoire CowLife de sciences animales de l'Université McGill et le laboratoire Bioinformatique de l'Université du Québec à Montréal (UQÀM) se sont associés pour mobiliser des technologies innovantes comme l'intelligence artificielle et l'Internet des objets (IoT) au service du bien-être des vaches laitières. L'objectif est de développer des outils capables de détecter précocement des signes de mal-être chez les animaux et d'aider les producteurs à prendre des décisions éclairées pour améliorer les conditions d'élevage et la longévité de leurs vaches. Les expérimentations ont actuellement lieu sur deux fermes de recherche et le but est de les déployer plus tard à plus grande échelle dans des exploitations commerciales à travers le Canada puis dans le monde entier.

L'initiative de cette chaire co-créée par Abdoulaye Baniré Diallo, Professeur en informatique et bio-informatique à l'UQÀM et Elsa Vasseur, Professeure agrégée au Département de sciences animales de l'Université McGill, est financée par le programme Alliance du CRSNG et PROMPT et soutenue par l'Université McGill, l'Université du Québec à Montréal (l'UQÀM) ainsi que par Novalait, Les Producteurs laitiers du Canada (PLC), Dairy Farmers of Ontario (DFO), Les Producteurs de lait du Québec (PLQ) et Lactanet.

FIGURE 2 – Partenaires de la chaire Well-E

1.2 Travaux réalisés par la chaire

La chaire est constituée de deux équipes : des membres du laboratoire CowLife de McGill et des membres du laboratoire Bioinformatique de l'UQÀM où j'ai fait mon stage. Ses travaux de recherche s'organisent autour de quatre grands axes présentés dans la figure ci-dessous :

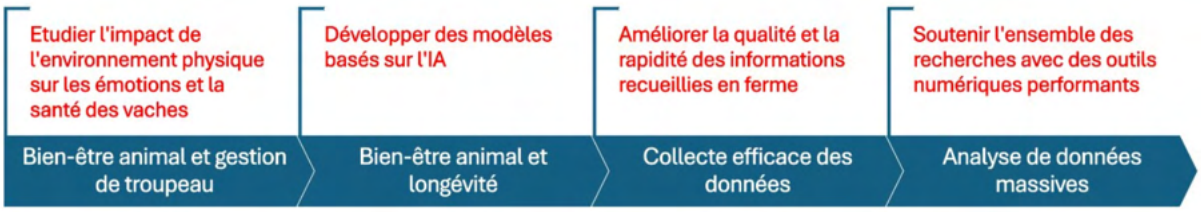


FIGURE 3 – Axes de recherche de la chaire Well-E

Le but du laboratoire CowLife est de réaliser des expériences avec les vaches pour en apprendre plus sur elles, sur leur comportement ainsi que sur leur bien-être car peu d'informations existent à ce sujet. En effet, aujourd'hui on sait par exemple plus de choses sur les rats que sur les vaches. Cependant, la motivation de changement de traitement des animaux ne cesse d'augmenter avec l'opinion publique. Au Canada par exemple, à partir de 2027, les vaches laitières ne pourront plus être entravées durant toute la durée de leur production. Les producteurs laitiers devront donc adapter leur manière de traiter leurs animaux sans subir de perte de productivité. C'est là qu'intervient la recherche de la chaire Well-E, afin d'assister les producteurs laitiers dans leurs décisions tout en aidant les vaches à avoir une vie plus agréable.

Pour atteindre cet objectif, beaucoup d'expériences sont menées sur les vaches afin de mieux comprendre leur condition physique, leur état émotionnel et leur comportement devant des changements dans leur environnement de vie. Cela a lieu sous forme d'enregistrements vidéos sur lesquels chaque changement de comportement est annoté pour en définir l'origine (couchée, debout, en train de manger, de mordre, de jouer, etc.). Or, le processus d'annotation des données est long (environ 1h d'annotation pour 10min de vidéo), laborieux et coûteux car il nécessite une expertise spécifique. L'équipe du laboratoire Bioinformatique de l'UQÀM propose et développe donc des algorithmes basés sur de la vision par ordinateur et de l'intelligence artificielle afin de traiter toutes les données récoltées lors des expériences.

L'un des exemples de projet réalisé dans le cadre de la chaire Well-E est un logiciel qui détecte les mouvements de la vache et qui les indique sous forme de courbe sous la vidéo comme on peut le voir sur la figure 4. Cela permet ainsi d'accélerer les moments où il ne se passe rien dans les vidéos pour se concentrer sur les moments importants et entraîne un gain d'environ 30% de temps d'annotations. Un deuxième projet en cours est une détection automatique en temps réel des comportements pour automatiser complètement l'annotation de ces vidéos. Le principe de cette méthode est de comparer les positions de trois bounding box présentes autour du corps, de la tête et du museau de la vache afin d'en déduire son comportement comme le montre la figure 5.

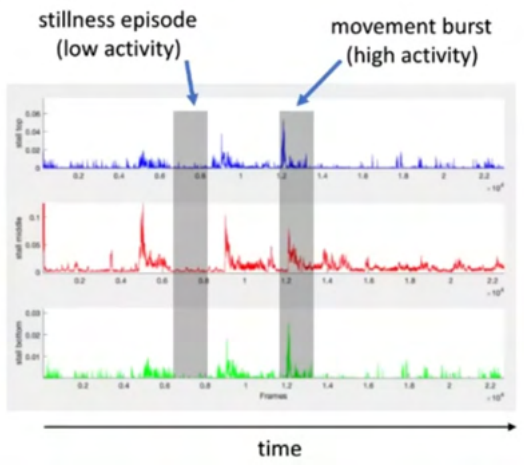


FIGURE 4 – Logiciel de détection de mouvement des vaches

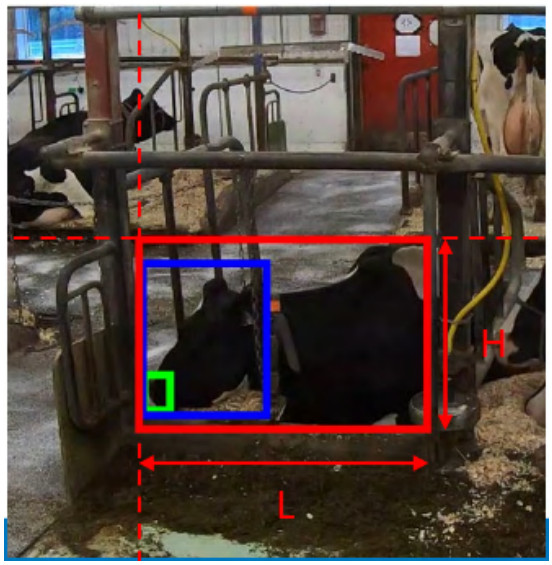


FIGURE 5 – Méthode de détection automatique du comportement des vaches

Enfin, plusieurs méthodes d'intelligence artificielle et de vision par ordinateur sont déployées par la chaire afin de détecter les vaches dans des environnements agricoles complexes et de les identifier afin de pouvoir faire du tracking. La chaire utilise également l'intelligence artificielle pour des méthodes d'analyse des émotions des vaches.

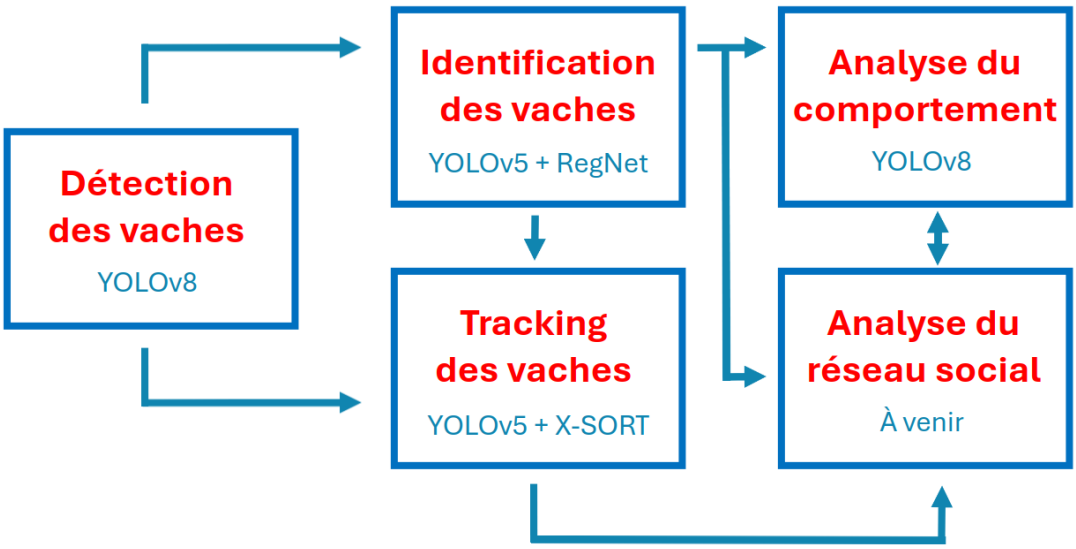


FIGURE 6 – IA dans le cadre de la chaire Well-E

Toutes ces méthodes ont pour but de permettre de suivre automatiquement les vaches et d'observer leurs comportements dans différentes situations. Cela peut par exemple être lors d'expériences telles que des enrichissements cognitifs en ajoutant des jeux dans leur environnement. Etant donné le changement potentiel dans la gestion des vaches dans les fermes et les étables, la chaire étudie également leurs traits de personnalité en vue de regroupements préférentiels pour favoriser leur bien-être et leur longévité.

1.3 Mon projet de fin d'études

Les méthodes développées par Well-E ont pour objectif d'étudier et de comprendre les comportements des vaches laitières afin d'améliorer leur bien-être et leur longévité. À terme, le but est de déployer les approches proposées dans de nombreuses exploitations laitières au Canada et dans le monde entier avec pour promesse faite envers les agriculteurs que ces approches seront faciles à mettre en place et ne nécessiteront pas d'installations complexes qui modifieraient les exploitations laitières ou seraient invasives pour les vaches.

Cependant, actuellement Well-E capture des données d'expériences avec de nombreuses caméras mais des nouveaux moyens de capture des données devront être explorés pour couvrir des environnements ouverts ou avec une demande majeure de mobilité, surtout dans de grandes exploitations. Afin de permettre de mieux suivre les vaches, une solution possible envisagée est donc d'utiliser un robot possédant des caméras et sur lequel on pourrait intégrer tous les algorithmes développés par la chaire. Dans un premier temps, cela permettrait à Well-E de récolter et de traiter des données plus facilement afin de pouvoir concevoir de nouvelles expériences et les conduire d'une façon plus automatisée. Dans un second temps, l'utilisation d'un robot permettrait également de proposer un système unique et autonome aux agriculteurs et qui pourrait suivre les vaches et les étudier en temps réel.

Actuellement, personne ne travaille sur le développement d'un tel robot dans la chaire. Le but de mon stage est donc d'étudier, de comparer et de proposer des options de robots qui seront le plus adaptés à l'utilisation que souhaite en faire Well-E et de créer une simulation de l'environnement computationnel exploité par ce robot en lui intégrant une architecture permettant d'utiliser les algorithmes créés dans le cadre de la chaire. Ce simulateur pourra également me permettre de développer plusieurs algorithmes de déplacement du robot en fonction des tâches qui lui seront assignées (suivi de vache aléatoire, repérage et suivi d'une vache en particulier, etc.). Le planning prévu initialement avec les différentes étapes de mon stage est détaillé en annexe *A | Programme du stage*. Les études pour une recommandation du robot et du simulateur sont détaillées dans les sections 2 et 3 tandis que l'intégration des modèles développés par Well-E et les algorithmes de déplacements sont expliqués en section 4.

En parallèle de ce projet principal, j'ai également travaillé sur un projet d'augmentation de données sur des vidéos de vaches, projet qui a requisitionné tous les membre de la chaire et que je détaillerai en section 5.

2 Etude pour la recommandation d'un robot

Dans le cadre de la chaire de recherche-innovation en bien-être animal et intelligence artificielle Well-E, des robots mobiles capables de se déplacer dans les exploitations de vaches laitières seront utilisés comme outils de support afin d'analyser le bien-être physique et mental des vaches. Cependant, aujourd'hui il n'existe pas de tels robots, si bien que nous ne savons pas comment le robot interagira avec les vaches et quels pourront être les problèmes rencontrés lors de son déploiement en milieu réel. Ainsi, avant de tester le robot directement dans de vrais scénarios d'utilisation comme dans les étables des fermes universitaires et commerciales, il est nécessaire de réaliser des simulations de déploiement afin d'évaluer les besoins en terme de ressources computationnelles et de créer et de tester des algorithmes tels que des algorithmes de navigation autonome ou de détection et d'identification des vaches. Au début de mon stage, j'ai donc dû comparer et évaluer différentes options de robots mobiles en fonction de leur pertinence par rapport au projet de la chaire.

2.1 Choix d'un type de robot

Afin de choisir le robot qui sera le mieux adapté au projet de la chaire, j'ai tout d'abord fait un état de l'art des robots agricoles présenté en annexe C puis j'ai resserré mes recherches en faisant un état de l'art des robots dans les fermes laitières mis en annexe D. L'annexe *C\État de l'art des robots agricoles* montre qu'aujourd'hui l'agriculture est en pleine révolution technologique car l'intégration des robots et des technologies avancées vise à réduire la dépendance à la main d'oeuvre humaine et à apporter des solutions à diverses tâches allant de la gestion des cultures pour optimiser les rendements à l'élevage des animaux. Les principales innovations présentées sont des robots mobiles roulants [51], des tracteurs autonomes [34] ou encore des drones [30][53]. Cependant, très peu de robots sont conçus pour intéragir avec les animaux. Je me suis donc concentrée sur les robots existants dans les exploitations laitières dont l'étude complète est présente dans l'annexe *D\État de l'art des robots dans les fermes laitières*. Dans ce rapport, on remarque que la majeure partie des robots permettent de faciliter le travail humain (robots de traite automatisée [7][33], robots pousseurs [44] [38], robots de nettoyage du fumier [18]). Certaines technologies commencent à s'intéresser à l'amélioration de l'efficacité énergétique ainsi qu'à la diminution des émissions de gaz à effet de serre [35] mais très peu prennent en compte le bien être animal [20]. Quelques systèmes de surveillance des vaches ont été créés grâce au machine learning [6][16] et à l'intelligence artificielle [23][10][37] mais, même si certains d'entre eux sont non invasifs [17], ils nécessitent quand même d'emmener la vache dans un endroit particulier, ce qui peut les déranger, modifier leur comportement et donc biaiser les résultats.

Ainsi, aujourd'hui il n'existe aucun robot dans les exploitations laitières, tel que Well-E le conçoit, c'est-à-dire un robot mobile complètement autonome qui pourrait étudier l'état de santé des vaches en les suivant, en récoltant des données et en les analysant sans modifier leurs habitudes et leur comportement. Or, le fait de déployer un robot mobile dans un milieu agricole au sein même de vaches présente de nombreuses difficultés dues à l'environnement ainsi qu'à l'intéraction avec des animaux. Dans un premier temps, je me suis donc penchée sur le type de robot à choisir car, même si dans les deux états de l'art précédents, la majorité des robots agricoles étaient des robots à roues, ceux-ci ren-

contrent des limites dues à leur configuration cinématique et à la topologie des terrains agricoles [39]. J'ai ainsi décidé de les comparer aux robots quadrupèdes, moins explorés en agriculture mais qui pourraient offrir une alternative intéressante, notamment grâce à leur capacité à s'adapter à des terrains complexes et variés. L'étude complète est présentée dans l'annexe *E\Comparaison de robots mobiles* et montre d'ailleurs que certains chercheurs ont déjà essayé de réaliser des expériences avec des robots quadrupèdes pour des tâches de surveillance et de gestion [52], de gardiennage [29] ou encore des tâches de terrain [19]. J'ai donc rassemblé plusieurs études de comparaison de robots mobiles et quadrupèdes et j'en ai tiré le tableau de comparaison suivant :

Critères	**Robots quadrupèdes**		**Robots roulants**	
	Notation	**Commentaire**	**Notation**	**Commentaire**
Mobilité sur terrain complexe	Excellente	Adaptation naturelle aux obstacles	Faible	Bloqués par pentes, boue
Interaction avec les animaux	Très bonne	Intégration fluide dans les troupeaux	Limitée	Encombrants
Autonomie énergétique	Moyenne	Batterie moyenne durée	Bonne	Sur sol plat en ligne droite
Polyvalence des tâches	Haute	Surveillance, gardiennage, interventions	Moyenne	Transport, pulvérisation
Résilience météo	Bonne	Terrain meuble, humidité	Variable	Risque d'enlisement
Coût	Élevé	Technologie complexe	Moyen à élevé	En fonction du modèle
Acceptabilité sociale	Bonne	Formes biomimétiques	Moyenne	Machines imposantes

TABLE 1 – Comparaison détaillée entre robots quadrupèdes et roulants en agriculture

D'après ce tableau, dans le cadre de notre projet, les robots quadrupèdes semblent être les robots les plus appropriés, et ce pour plusieurs raisons. Tout d'abord, le robot devra avoir une grande mobilité sur des terrains variés. En effet, il pourra être amené à être utilisé dans des étables qui possèdent de nombreux obstacles ainsi qu'un sol assez encombré mais aussi en extérieur, autant sur de la pelouse en été que sur de la neige en hiver ou sur de la boue, des graviers et des flaques par temps de pluie [39]. Il pourra également y avoir des pentes en extérieur et des marches à gravir ou à enjamber en intérieur. Un autre avantage des robots quadrupèdes est leur aspect biomimétique. En effet, leur apparence nous permettra de les faire se déplacer un peu comme des vaches, ce qui favorisera sûrement leur intégration auprès de ces animaux. Il faudrait faire des expériences pour être sûr de cela mais on peut imaginer qu'un tel robot fera moins peur à des vaches s'il leur ressemble. Enfin, les robots quadrupèdes possèdent une importante stabilité dynamique ainsi qu'une grande liberté de mouvements [39]. Ainsi, en cas de déplacement sur un terrain compliqué ou si une vache décide de donner un coup dans le robot, il ne perdra normalement pas l'équilibre. De plus, les robots quadrupèdes possèdent une liberté de mouvement dans toute les directions, ce qui permettra à notre robot d'aller facilement n'importe où et donc de ne pas rentrer dans une vache même s'il se trouve dans

un endroit exigu. Les seuls points faibles de ces robots sont leur prix et leur autonomie [42] mais pour notre projet nous avons besoin de technologies complexes et nous n'aurons peut-être pas besoin du robot Spot en continu. Il pourra donc aller se recharger entre ses différentes phases de navigation. Ainsi, grâce à des capteurs et à la leur grande flexibilité, les robots quadrupèdes associés à des algorithmes avancés de localisation, de cartographie et de planification en trois dimensions sont des robots tout à fait adaptés aux exploitations laitières [15][49][54].

2.2 Sélection d'un robot quadrupède

Comme expliqué dans la partie précédente, le type de robot qui semble être le mieux adapté à notre projet est le robot quadrupède, tant pour sa grande mobilité sur des terrains variés que pour sa stabilité dynamique, sa grande liberté de mouvement et son aspect biomimétique. Cependant, il existe beaucoup de robots quadrupèdes possédant des caractéristiques et des capteurs différents. Cette section présente donc une comparaison entre plusieurs robots quadrupèdes et une analyse de celui qui pourrait être le plus adapté à notre projet.

2.2.1 Etude des robots quadrupèdes

Le premier robot quadrupède à avoir été disponible commercialement est le robot Spot que Boston Dynamics (USA) a lancé publiquement en 2020. Capable de marcher de façon fluide dans le monde réel, Spot a popularisé la robotique dynamique en montrant qu'un robot pouvait intéragir de manière autonome dans un environnement complexe et il a également marqué les débuts de la mobilité bio-inspirée.[5] En 2021, DeepRobotics (Chine) sort son robot Lynx conçu pour les environnements extérieurs complexes (pluie, neige, pente). Le Lynx est un pionnier de la robotique de terrain. Il allie autonomie de mouvement et résistance aux conditions extrêmes, ouvrant la voie à des usages de surveillance, d'inspection et de sauvetage.[11] En 2022, DeepRobotics sort le X20, un robot quadrupède de classe industrielle pensé pour des missions de patrouille autonome, d'inspection intelligente et de téléopération sur de grands sites industriels. Il introduit une intégration poussée de l'IA embarquée, des capteurs multiples (LiDAR, caméras, IMU) et une connectivité 5G. C'est aussi l'un des premiers quadrupèdes à être utilisé en environnements extérieurs réels 24/7, comme dans des centrales électriques, des zones pétrochimiques ou des parcs solaires.[13] Ces trois robots sont illustrés dans la figure 7.

FIGURE 7 – Robots Spot, Lynx et X20

Entre 2021 et 2023, Unitree Robotics (Chine) démocratise complétement les robots quadrupèdes avec ses robots Go1 et Go2. Avec le Go1, pour la première fois, un robot quadrupède agile, ROS-compatible, équipé de capteurs intelligents, était vendu à moins de 4 800 USD$. Le Go2 a ensuite amélioré la puissance, l'IA embarquée, et intégré le LiDAR 4D, rendant ces robots accessibles aux universités et startups.[47] En même temps, en 2023 WLKATA (Chine) sort MarchX, l'un des premiers quadrupèdes éducatifs intégrant LiDAR, caméra de profondeur et SDK AI dans un format abordable. Il vise l'apprentissage de la perception, de la planification et de l'IA embarquée dans des environnements simulés et réels (Sim2Real).[50] En 2023, DeepRobotics poursuit son avancée dans la robotique industrielle avec le lancement du X30, conçu pour des environnements encore plus exigeants, avec une résistance accrue, une plage de température étendue et une meilleure stabilité dynamique. Il intègre des capteurs de couple sur les moteurs, des caméras stéréo, un LiDAR et offre une autonomie optimisée. Le X30 cible des missions d'inspection, de sécurité ou de logistique en terrain difficile, notamment dans des zones industrielles à haut risque. Il renforce l'idée que les robots quadrupèdes peuvent devenir des assistants polyvalents dans l'industrie lourde et les interventions critiques.[12] Enfin, plus récemment, en 2024, Unitree Robotics a proposé B2, l'un des quadrupèdes les plus rapides au monde (6 m/s), capable de porter jusqu'à 40 kg. Il pousse la limite de la puissance, de la vitesse et de la charge utile des robots quadrupèdes. C'est une plateforme de test pour des applications industrielles comme la logistique automatisée.[48] Ces quatre robots sont illustrés sur la figure 8.

FIGURE 8 – Robots Go2, MarchX, X30 et B2

Tous ces robots se challengent et sont très compétitifs sur le marché. Dans la figure 9, un tableau compare leurs principales caractéristiques trouvées sur les sites des fabricants.

Robot	Go2 Air	B2	Spot	Lynx	X20	X30	MarchX
Entreprise	Unitree	Unitree	Boston Dynamics	DeepRobotics	DeepRobotics	DeepRobotics	WLKATA
Année de sortie	2023	2023	2020	2024	2021	2023	-
Poids	15 kg	60 kg	32.7 kg	30 kg	53 kg	56 kg	9.5 kg
Dimensions debout (cm)	70x31x40	109.8x45x64.5	110x50x61	80x50x60	95x47x70	100x58.5x47	60x30x40 cm
Dimensions couché (cm)	76x31x20	88x46x33	110x50x19.1	-	-	-	-
Vitesse max	2.5 m/s	6 m/s	1.6 m/s	5 m/s	4 m/s	4 m/s	1.5 m/s
Autonomie	1-2 h	4-6 h	1h30	3 h	2-4 h	2.5-4 h	2-3 h
Charge utile max	10 kg	120 kg	14 kg	12 kg	20 kg	20 kg	3 kg
Hauteur max obstacle	15 cm	20-25 cm	30 cm	22 cm / 80 cm	20 cm	20 cm	5-10 cm
Inclinaison max	30 °	45 °	30 °	30°	30 °	45°	15-20 °
Type de sol	Tout type	Tout type	Tout type	Tout type (très performant sur neige)	Tout type	Tout type	Surface plane (plutôt intérieur)
Distance de saut fossé	-	0.5-1.2 m	-	-	-	-	Pas de saut
Capteurs	Caméra de profondeur, caméra frontale HD, capteurs IMU	3D LiDAR, 2 caméras de profondeur, 2 caméras optiques	Caméra 360° 4m de profondeur, éviteur de collision, ne nécessite pas de guidage ni de GPS	Caméra grand angle 1080p orientée vers l'avant, connectivités Wi-Fi et GPS intégrées	Caméra de profondeur, LiDAR, algorithmes de navigation autonome, détection d'obstacles et reconnaissance humaine	Caméra stéréo + profondeur, IMU, 4 LiDAR (compatibilité avec capteurs 3D), capteurs de couple sur les moteurs, navigation autonome dans le noir	Caméra de profondeur, LiDAR, IMU, encodeurs de position, capteurs de couple
Indice de protection	IP54	IP67	IP54	IP54	IP66	IP67	-
Etanchéité	Oui, éclaboussures mais pas immersions ni jets puissants	Oui, immersion accidentelle jusqu'à 1m pendant 30min	Oui, éclaboussures mais pas immersions ni jets puissants	Oui, éclaboussures mais pas immersions ni jets puissants	Oui, résiste aux jets d'eau puissants	Oui, immersion accidentelle jusqu'à 1m pendant 30min	-
Résistance à la poussière	Oui mais limitée	Oui, totale	Oui mais limitée	Oui mais limitée	Oui, totale	Oui, totale	-
Températures	-20/55°C	-20/55°C	-20/55°C	0/40°C	-	-20/55°C	-
Niveau sonore	-	Modéré à bruyant	Très silencieux	-	-	-	-
Domaine d'utilisation	Éducation, recherche, navigation autonome, inspection légère, robotique de loisir	Logistique, transport de charge, inspection industrielle, recherche, surveillance de sites, robotique avancée en terrain difficile	Inspection d'infrastructures, construction, sécurité, exploration d'environnements difficiles, collecte de données	Inspection extérieure (incluant neige et terrains difficiles), sécurité, surveillance, missions de recherche et sauvetage, exploration autonome	Inspection industrielle, sécurité, sauvetage, exploration, recherche	Inspection industrielle, sécurité, sauvetage, logistique, exploration	Éducation, démonstration, apprentissage de la robotique, programmation de bras manipulateur
Prix	1850 $	100 000 $	> 75 000 $	17 999 $	15 900 $	65 000 $	18 800 $

FIGURE 9 – Caractéristiques des principaux robots quadrupèdes

D'après ce tableau, nous pouvons voir que les robots varient considérablement en terme de poids, allant de 9,5 kg pour le MarchX à 60 kg pour le B2. Concernant les performances, le B2 se distingue par sa vitesse maximale de 6 m/s et sa capacité à transporter jusqu'à 40 kg, tandis que le Go2 Air et le MarchX se positionnent comme des modèles plus compacts et accessibles, avec une charge utile plus limitée. Concernant l'autonomie, le X20, le X30 et le MarchX offrent les meilleures durées d'utilisation (jusqu'à 4 heures), tandis que le Spot de Boston Dynamics et le Go2 Air sont un peu plus limités dans ce domaine. La capacité à franchir des obstacles ou à sauter des fossés est également variable : seuls le B2 et le Lynx mentionnent une capacité de saut, tandis que d'autres misent sur leur agilité et leur capacité à s'adapter à différents types de terrain, y compris les environnements difficiles ou enneigés.

Tous ces robots sont équipés de capteurs avancés comme des caméras de profondeur, des LiDAR, ou encore des capteurs, ce qui leur permet de naviguer de manière autonome, d'éviter les obstacles, ou encore de reconnaître des éléments dans leur environnement. Le X30 se distingue en intégrant en plus des capteurs de couple sur les moteurs, ce qui améliore la précision dans la navigation autonome, notamment dans le noir. Le niveau de protection contre l'eau et la poussière varie également, avec des modèles comme le B2 (IP67), le X20 (IP66) et le X30 (également IP67) qui sont particulièrement résistants, y

compris en cas d'immersion ou de jets d'eau puissants. Les domaines d'utilisation couvrent un large éventail : éducation, inspection industrielle, robotique de loisir, exploration, sécurité, recherche, ou encore démonstration. Le X30, par exemple, est orienté vers des usages intensifs de sécurité, de logistique ou d'exploration dans des environnements à haut risque. Enfin, le prix de ces robots reflète leurs capacités : de 1 850 $ pour le Go2 Air, accessible et polyvalent, à 100 000 $ pour le B2, un modèle très performant dédié à des usages professionnels avancés.

En résumé, chaque robot a ses spécificités. Certains sont conçus pour la performance et les environnements complexes et d'autres pour l'accessibilité, l'apprentissage ou des missions plus ciblées. Je vais maintenant discuter des options qui pourront permettre de répondre au mieux aux besoins du projet de la chaire.

2.2.2 Choix d'un robot quadrupède pour notre projet

Étant donnés les besoins de couvrir des environnements extérieurs ainsi que d'être potentiellement déployé à des échelles commerciales, nous pouvons remarquer en observant le tableau que certains robots n'atteindront pas ces requis. C'est le cas du MarchX de WLKATA et du Go2Air d'Unitree qui semblent être destinés à un usage plutôt éducatif et en intérieur sur des surfaces planes. Le robot Lynx de DeepRobotics semble être très performant pour se déplacer sur des terrains difficiles mais il ne sera pas assez robuste étant donné qu'il n'opère que sous des températures allant de 0 à 40°C or, en hiver, les températures dans les étables et en extérieur peuvent descendre bien en dessous de 0°C, surtout au Québec. Le X20 semblait également être un bon candidat mais sa plage de températures n'est pas précisée. Il faudrait contacter le constructeur pour obtenir cette information ou préférer prendre sa version améliorée, le X30 pour laquelle les plages de températures sont adaptées à notre projet bien qu'il soit plus cher. Les modèles B2 d'Unitree, Spot de Boston Dynamics et le X30 de DeepRobotics semblent donc être assez équivalents en terme de capteurs et de résistance au milieu agricole. Leurs plages de températures correspondent également à ce dont nous avons besoin pour notre projet. En terme de prix, le B2 et Spot sont équivalents une fois que l'on a ajouté le prix des capteurs, tandis que le X30 est un peu moins cher. Cependant, un autre critère important est le niveau sonore. En effet, si le robot fait trop de bruit, cela risque de déranger les vaches, ce qui va à l'encontre du but d'utilisation du robot qui est d'améliorer leur bien-être. Il n'y a pas beaucoup d'informations disponibles publiquement à propos de cette caractéristique mais d'après les informations présentées par les fabricants, le B2 d'Unitree est modéré à bruyant, le X30 ayant été développé pour des besoins industriels est également bruyant et Spot de Boston Dynamics est connu comme étant le plus silencieux des robots quadrupèdes.

D'après son ensemble de caractéristiques, c'est donc le robot Spot de Boston Dynamics qui semble être le plus approprié pour le projet de la chaire et, vue sa popularité et ses applications dans de nombreux domaines, l'offre d'outils pour son développement et la simulation de son environnement sera possiblement plus riche et facilement disponible. La disponibilité des outils de support ouverts au public est un critère très important car avant de déployer notre robot quadrupède, l'un des buts de ce projet est de construire et de représenter différents cas d'utilisations et de situations en simulation. Dans la partie suivante, nous allons donc étudier les simulateurs disponibles publiquement pour le robot Spot et choisir la configuration d'un environnement qui pourra nous permettre d'obtenir des événements de simulation les plus proches de la réalité.

3 Etude pour la recommandation d'un simulateur et d'un environnement

3.1 Sélection de simulateurs

Dans l'annexe *F\ Choix d'un simulateur robotisé pour soutenir les activités de recherche sur les fermes laitières*, j'ai fait une comparaison détaillée des principaux simulateurs utilisés en robotique. Comme montré dans cette annexe, les critères de comparaison qui reviennent le plus dans les articles de recherche dans le domaine de la simulation sont les suivants [41, 36, 14, 9, 22] :

— la compatibilité avec ROS
— la modélisation de l'environnement
— la précision des moteurs physiques
— le prix
— l'interface utilisateur
— la complexité de modélisation
— la consommation de ressources
— l'utilisation de capteurs

À partir l'étude détaillée dans l'annexe F, j'ai pu construire le tableau comparatif présenté en figure 10.

Simulateur	Compatibilité ROS2	Interface utilisateur	Précision capteurs	Modélisation scène	Consommation CPU
CoppeliaSim	Moyenne	Excellente	Très bonne	Très bonne	Moyenne
GAZEBO	Excellente	Bonne	Très bonne	Moyenne	Elevée
Webots robot simulation	Bonne	Bonne	Moyenne	Bonne	Bonne
morse-simulator/ morse	Moyenne	Faible	Faible	Faible	Moyenne
AIRSIM	Faible à moyenne	Moyenne	Très bonne (drones)	Moyenne	Elevée
unity	Faible	Excellente	Moyenne à bonne	Excellente	Elevée

FIGURE 10 – Tableau comparatif des principaux simulateurs utilisés en robotique

Dans le cadre de la chaire Well-E, le robot déployé dans les étables devra gérer plusieurs tâches en même temps comme par exemple commander ses moteurs, détecter les vaches ou encore naviguer de façon autonome. Pour gérer à la fois les capteurs, les actionneurs et la prise de décision lors de la navigation, ROS sera indispensable afin de permettre à tous ces éléments de communiquer entre eux en temps réel via des topics. Comme montré dans le tableau figure 10, AirSIm et Unity ne semblent pas être des options appropriées étant donné que ce sont des logiciels très spécifiques et difficiles à prendre en

main. L'interface utilisateur et la modélisation de la scène sont également des critères importants pour permettre d'avoir des possibilités de modification et de customisation des paramètres et du contexte de la scène au cours de l'évolution du projet qui ne fait que commencer. De plus, à terme le but est de représenter l'étable et la ferme le plus fidèlement possible d'où l'importance d'utiliser un outil avec lequel cette partie sera facile à mettre en oeuvre. D'après l'analyse de la figure 10, Morse n'est donc pas le bon logiciel à utiliser pour favoriser ce critère. Ensuite, la précision des capteurs doit être correcte afin qu'il n'y ait pas de grande différence dans les algorithmes de navigation lorsque nous passerons de la simulation à la réalité. Enfin, il est préférable que le simulateur soit en open source et avec une faible consommation CPU afin que l'environnement soit le plus reproductible possible.

Les trois simulateurs adaptés au projet de la chaire et que j'ai donc sélectionnés à la fin de l'annexe F sont ainsi CoppeliaSIM, GAZEBO et Webots. Un autre défi que j'ai ensuite rencontré est que Boston Dynamics ne propose pas de modèle de Spot utilisable sur un simulateur tant que l'on n'a pas acheté le robot. J'ai donc dû trouver un modèle équivalent disponible sur l'un de ces trois simulateurs et étudier les environnements possibles pour le déploiement de ceux-ci.

3.2 CoppeliaSIM et ZeroMQ Remote API

CoppeliaSim est le simulateur qui semble répondre le mieux aux besoins de facilité de modélisation et de modification de la scène. Dans le cadre de ce projet, le déploiement du robot n'en est qu'à ses débuts et nous n'avons encore aucune idée du comportement du robot. Il peut donc être très intéressant d'utiliser CoppeliaSim pour pouvoir personnaliser au mieux l'étable, le déplacement des vaches, le robot ainsi que les interactions entre ces différents éléments. Cependant, un premier défi en utilisant CoppeliaSim était le manque de modèle de Spot disponible. J'ai donc regardé s'il était facile de créer un robot équivalent sur cette plateforme. Pour faire cela, je ne me suis pas tout de suite lancée dans ROS. À la place, j'ai choisi de créer et de contrôler le robot via Python en utilisant l'API remote ZeroMQ car, grâce à cette approche, il est beaucoup plus facile de tester la logique de base comme les capteurs, les déplacements et les algorithmes simples. Tous les scripts créés pour cette partie sont présent sur ce git [31]. Pour modéliser rapidement l'étable, j'ai développé plusieurs scripts Python. Le premier, *create_ barn.py* [31], permet de créer l'étable en réglant de nombreux paramètres tels que la forme de l'étable, ses dimensions ou encore le type de sol. J'ai également ajouté une image au sol grâce au logiciel Inkscape qui permet de personnaliser des fichiers .svg. Ensuite, le script *create_ cows_ barn.py* [31] ajoute le nombre de vaches voulu à la scène en les positionnant de manière aléatoire dans l'étable. J'ai créé les vaches grâce à un modèle .stl trouvé sur le site Internet *free3d.com*[1] puis, dans le script, je leur ai ensuite attribué une couleur aléatoire entre blanc, noir et marron comme montré figure 11.

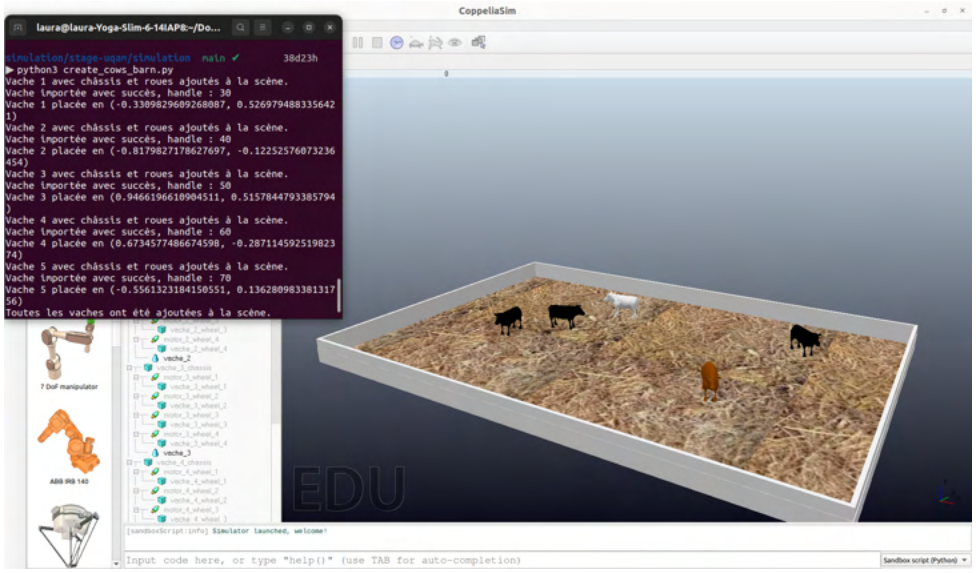


FIGURE 11 – Création de la scène et ajout de vaches

Après la création de la scène, le script *create_robot_barn.py* [31] permet d'ajouter le nombre de robots voulu à la scène. Dans un premier temps, j'ai construit un robot assez simple constitué d'un socle et de quatre roues reliées au socle par des joints. Pour améliorer le rendu visuel, j'ai ajouté des pièces sur le modèle dynamique du robot telles que des roues et des pneus. Enfin, j'ai mis une caméra à l'avant du robot ainsi que quatre LiDAR, un sur chaque face du robot comme nous pouvons le voir dans la figure 12. Ce qui est pratique avec cette méthode de simulation, c'est que l'on peut facilement obtenir les retours des capteurs dans des fenêtre séparées. Par exemple, la figure 13 montre le résultat du code *test_robot_barn_sensors.py* [31]. La fenêtre *front camera image*, montre le rendu de la caméra et à droite dans la fenêtre *figure 1*, nous avons la reconstruction spatiale de la scène réalisée à partir des données des quatre LiDARs.

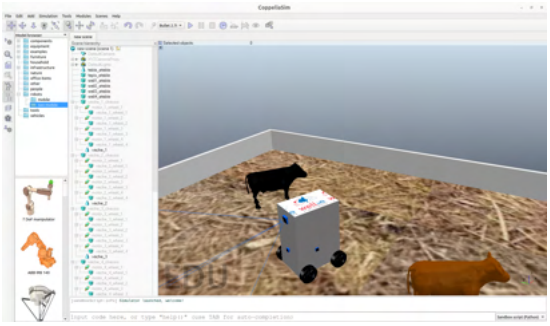


FIGURE 12 – Robot et ses capteurs

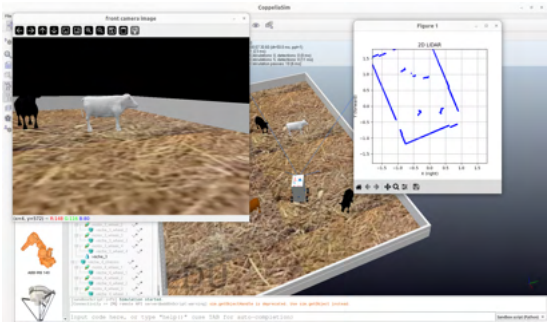


FIGURE 13 – Test des capteurs

Enfin, avec deux scripts Python intitulés *move_cows_barn.py* et *move_robot_barn.py* [31], j'ai fait bouger les vaches et le robot. Pour se rapprocher de la réalité, J'ai programmé la simulation des vaches de manière à ce qu'elles exécutent des mouvements aléatoires indépendants les uns des autres et qu'elles s'arrêtent de temps en temps. Quant au robot, pour l'instant je l'ai fait avancer et tourner sur lui même. Le point fort de CoppeliaSim et de son ZeroMQ remote API est qu'avec des scripts Python nous pouvons rapidement simuler l'étable, le robot, les vaches ainsi que leurs interactions. De plus, si

nous voulons changer d'environnement comme simuler par exemple des vaches à l'extérieur, nous pouvons rapidement changer la scène comme le montre la figure 14 où j'ai rapidement simulé un environnement extérieur grâce aux trois scripts *create_ outside.py*, *create_ cows_ outside.py* et *create_ trees_ outside.py* [31].

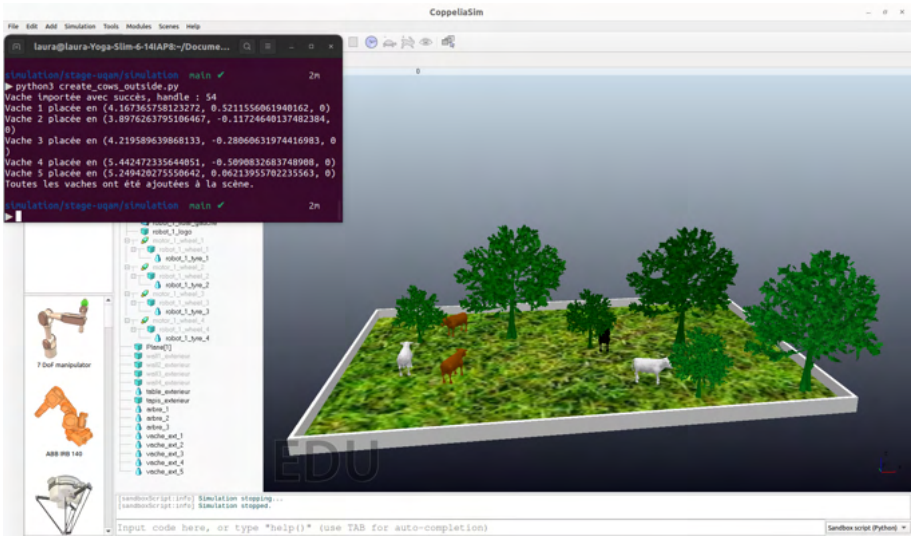


FIGURE 14 – Création d'une scène d'extérieur

La simulation illustrée dans la figure 14 montre qu'avec CoppeliaSim, nous pourrons facilement obtenir des environnements complexes très ressemblants à la réalité, ce qui est un vrai atout pour faciliter l'intégration des algorithmes de détections développés dans le cadre de la chaire Well-E. En effet, ceux-ci ont été entraînés dans des environnements réels donc si la simulation n'est pas assez réalistes, ils pourraient ne pas fonctionner dans le simulateur. Cependant, un gros point bloquant pour ce choix de simulateur est que dans cet exemple il m'a fallu du temps pour créer un robot simple mais il en faudra encore plus pour créer un robot quadrupède représentant Spot avec beaucoup de moteurs et de joints. Il faudrait également définir quels joints utiliser pour faire avancer, reculer ou tourner Spot car les déplacements des robots quadrupèdes sont assez complexes. Ceux-ci s'inspirent en effet des mouvements des animaux et dépendent de la vitesse de déplacement comme présenté en figure 15 [40].

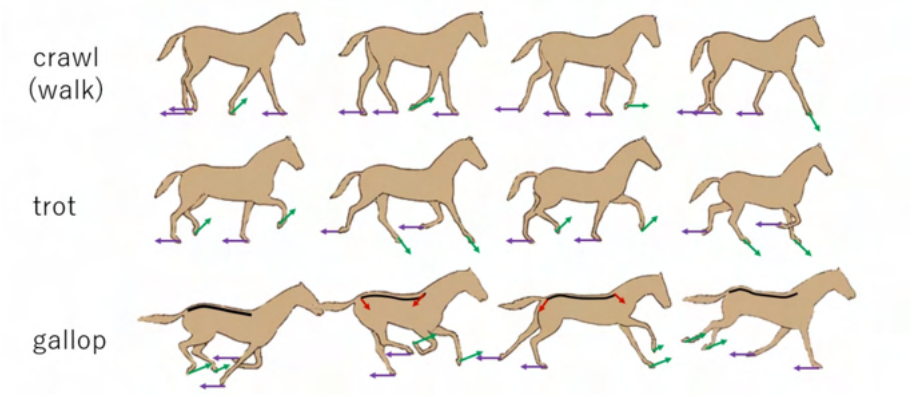


FIGURE 15 – Détails du mouvement des animaux à 4 pattes lors de la marche, du trot et du galop [40]

La figure 15 montre en effet que lorsqu'un animal marche, il a toujours 3 points d'appui au sol et que lorsqu'il trotte il n'en a que deux. Le mouvement est encore plus complexe lors du galop et induit des forces de pression et d'extension dans le dos de l'animal mais, dans le cas d'un robot quadrupède et surtout dans notre application, le galop est beaucoup trop rapide donc nous ne nous y intéresserons pas. Une fois le modèle du robot construit, la marche est le mouvement le plus simple car, avec 3 points d'appuis au sol, le robot est toujours stable. La figure 16 montre la séquence de marche d'un tel robot avec son centre de masse qui passe de droite à gauche en fonction de la jambe qui se trouve en l'air.

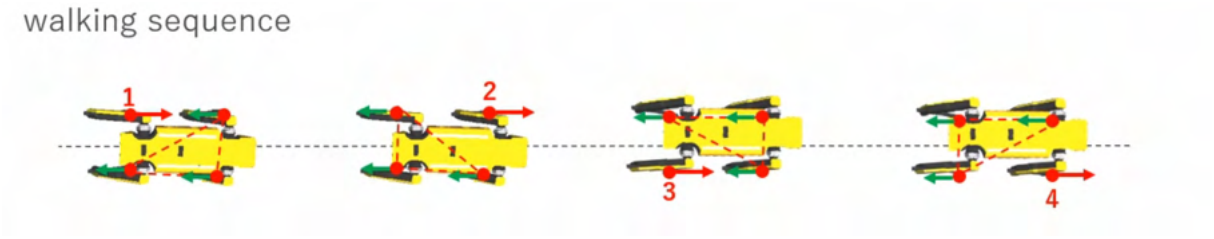


FIGURE 16 – Séquence de marche d'un robot quadrupède [40]

En revanche, dès que l'on veut prendre un peu de vitesse et passer au trot, il faut faire attention à bien placer le centre de masse à la fois de façon verticale et horizontale pour avoir des angles de roulis et de tangage nuls comme le montre la figure 17. Pour stabiliser le robot, il faudrait donc utiliser une IMU (Unité de Mesure Inertielle) et faire un contrôleur PID (Proportionnel Intégral Dérivé) afin de calculer les positions des jambes. De plus, il faut régler la position des 3 joints de chaque patte du robot notés O, A et B sur la figure 18 en fonction des angles que l'on souhaite avoir. Ces angles peuvent se calculer grâce à une méthode de cinématique inverse.

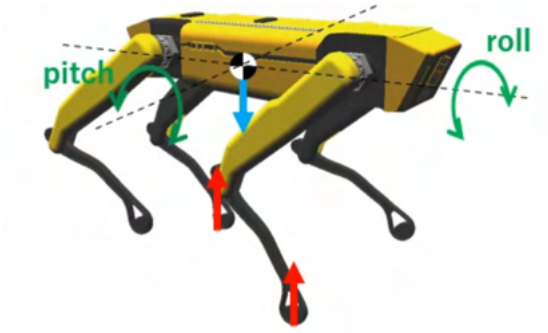


FIGURE 17 – Robot quadrupède au trot [40]

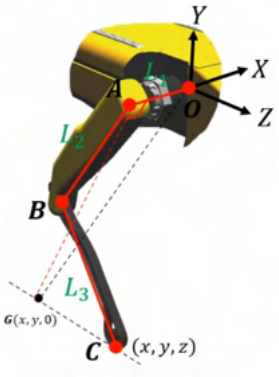


FIGURE 18 – Détails du fonctionnement d'une patte d'un robot quadrupède [40]

Enfin, la figure 19 montre la séquence de trot d'un robot quadrupède pour un déplacement vers l'avant, un déplacement latéral et une rotation sur place.

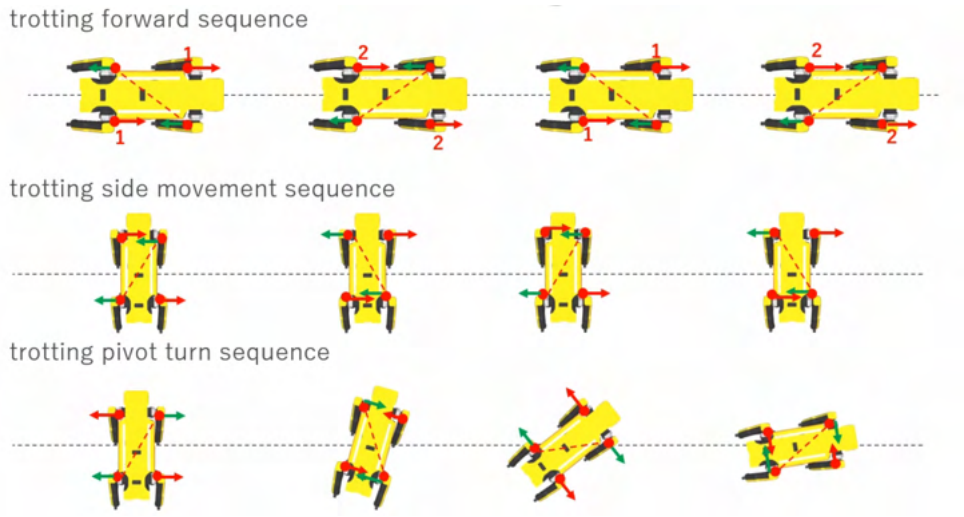


FIGURE 19 – Séquences de trot en avant, latéral et en pivot d'un robot quadrupède [40]

Pour utiliser Spot dans CoppeliaSim, il aurait ainsi fallu construire un robot quadrupède en lui mettant trois joints par patte et en lui ajoutant des capteurs puis développer des contrôleurs en fonction de la vitesse et du type de mouvement. Cependant, étant donné que la chaire avait rapidement besoin de rendu et d'algorithmes de navigation, ce n'était donc pas la bonne solutionc car cela aurait pris trop de temps. J'ai donc décidé de me tourner vers des simulateurs sur lesquels des packages de contrôles de Spot étaient déjà disponibles.

3.3 Gazebo avec ROS1 et Docker

Au cours de mes recherches, j'ai trouvé un modèle de Spot disponible sur CHAMP (Configuration and Hierarchical Algorithm for Motion Planning), un framework open source conçu pour permettre le développement de robots quadrupèdes et l'implémentation de contrôleurs de locomotion dynamiques. Le framework de contrôle est basé sur un article qui s'intéresse au robot Cheetah du MIT [32]. Cette plateforme est compatible avec ROS, Gazebo ou RViz et est très efficace pour mettre en œuvre des stratégies de locomotion dans des environnements simulés complexes, ce qui peut être une excellente chose pour notre projet.

Cependant, CHAMP a été créé pour ROS1, qui nécessite la version d'Ubuntu 16.04 ou 18.04. Or, comme plupart des projets aujourd'hui se font en ROS2, sur mon ordinateur j'avais ROS2 sur Ubuntu 22.04. Pour tester cette méthode de simulation, j'ai donc dû créer un docker. Un docker est un outil qui permet d'exécuter une application dans un conteneur, c'est-à-dire un environnement isolé contenant tout ce qu'il faut pour la faire fonctionner (code, dépendances, configuration). Contrairement à une machine virtuelle, un conteneur est léger et rapide car il partage le noyau du système hôte. Cependant, il fallait faire attention dans la création du docker car celui-ci devait pouvoir avoir une interface graphique dans laquelle visualiser ce qui se passe dans Gazebo ou dans RViz. J'ai donc tout d'abord créé un dockerfile dans lequel j'ai clôné les github nécessaire pour utiliser CHAMP [24]. Ensuite, j'ai créé une image docker et j'ai lancé le conteneur avec une interface graphique.

Dans un premier temps, j'ai tout d'abord lancé RViz pour voir à quoi devait ressembler notre robot Spot. Le résultat attendu et celui obtenu sont montrés respectivement sur les figures 20 et 21. J'ai bien obtenu un robot quadrupède avec des capteurs mais, avec le docker, j'ai perdu certaines composantes comme la couleur de Spot.

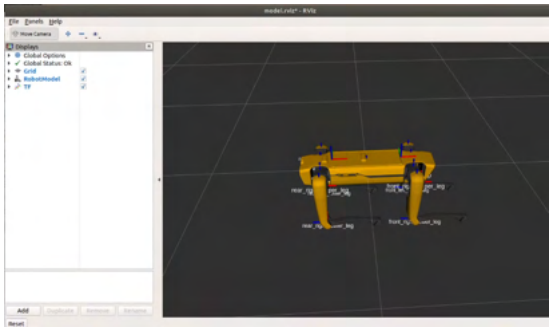


FIGURE 20 – Visualisation de Spot sur RViz attendue avec CHAMP [45]

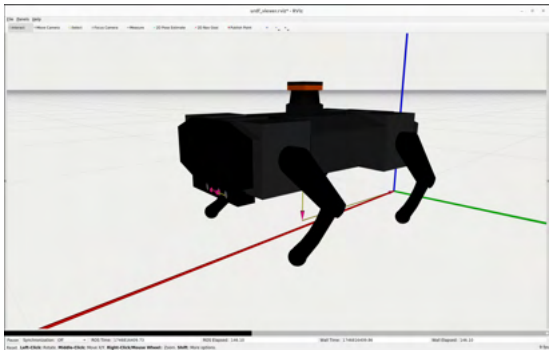


FIGURE 21 – Visualisation de Spot sur RViz obtenue dans le docker

Grâce à un script Python, j'ai pu ensuite contrôler Spot au clavier dans RViz. Malgré l'utilisation d'un docker, le contrôle était plutôt fluide et j'avais bien accès à toutes les directions. Ensuite, j'ai lancé Gazebo toujours dans le docker avec la scène proposée par CHAMP, un chantier de construction montré en figure 22. Dans le docker, j'ai bien obtenu la scène souhaitée et j'ai pu me déplacer dans celle-ci bien qu'il y ait un peu de latence. En revanche, j'ai ensuite commandé Spot au clavier comme montré figure 23 mais les déplacements étaient vraiment très lents.

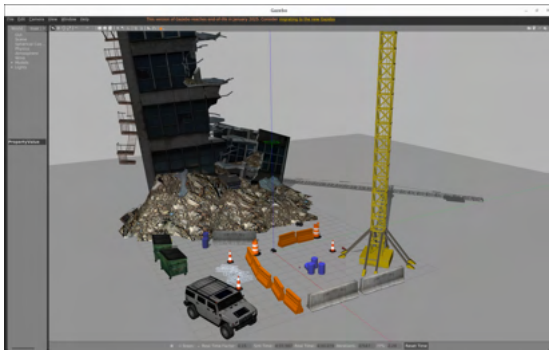


FIGURE 22 – Visualisation de la scène dans Gazebo

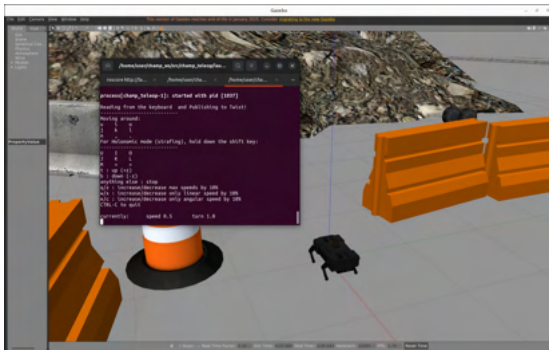


FIGURE 23 – Contrôle de Spot au clavier dans Gazebo

Enfin, j'ai tout de même voulu tester les capteurs de Spot. J'ai tout d'abord affiché dans RViz la carte de la scène qui avait été créée par CHAMP grâce à une méthode de SLAM (Simultaneous Localisation And Mapping) comme présenté dans la figure 24. Sur cette carte, j'ai pu superposer les objets que détecte Spot grâce à son LiDAR. Enfin, sur la figure 25, j'ai pu visualiser l'image obtenue par la caméra située à l'avant de Spot, ce qui sera nécessaire à notre projet pour détecter les vaches et implémenter les algorithmes d'intelligence artificielle.

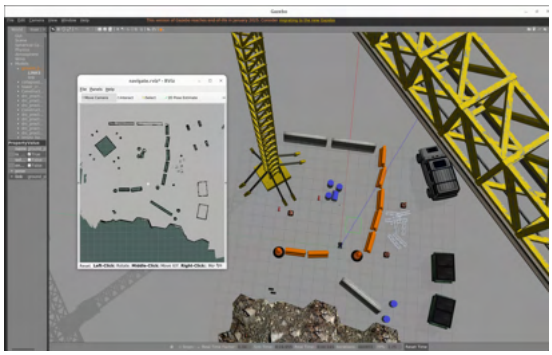


FIGURE 24 – Carte de l'environnement obtenue par SLAM

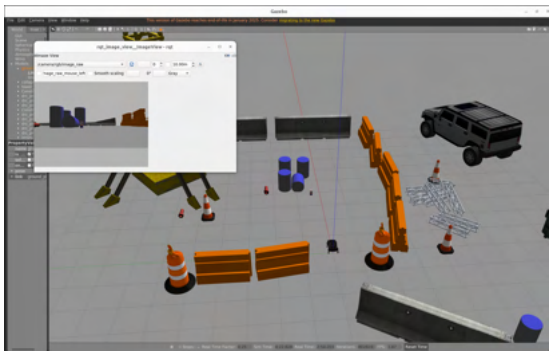


FIGURE 25 – Visualiation de l'image obtenue par la caméra

Pour continuer mon projet, il aurait maintenant fallu que je modélise ma propre scène avec l'étable et les vaches dans Gazebo, chose qui selon la documentation n'est pas facile. Cependant, à ce stade les challenges de ce choix d'environnement s'étaient déjà accumulés entre la difficulté du système docker, la perte d'information sur l'interface graphique ou encore la latence significative. J'ai donc décidé de regarder si d'autres packages n'existaient pas en ROS2, quitte à changer de simulateur même si Gazebo était le plus adapté à ROS2. Une autre possibilité aurait pu être d'essayer de créer un rosbridge pour passer de ROS1 à ROS2 mais cela est assez compliqué. Un git propose par ailleurs une adaptation de CHAMP en ROS2 mais celle-ci est encore en développement et n'est pas du tout documentée [4]. Enfin, un autre git propose depuis quelques mois une adaptation d'un code en ROS2 qui est en fait une extension d'un robot quadrupède open source nommé Dingo. Cependant, ce robot est assez différent de Spot et le git n'est pas encore terminé [2]. Pour poursuivre mon projet, j'ai donc dû continuer à regarder si d'autres packages existaient en ROS2.

3.4 Webots avec ROS2

En poursuivant mes recherches, j'ai trouvé un git créé par le Mobile Autonomous Systems & Cognitive Robotics Institute et qui propose des packages pour simuler Spot sur Webots avec ROS2 humble et Ubuntu 22.04 [21]. Dans un premier temps, j'ai téléchargé Webots et cloné ce git dans un workspace et installé toutes les dépendances nécessaires. Ensuite, après avoir compilé, construit et sourcé les packages, j'ai lancé le package de simulation Webots. En faisant cela, nous voyons que la scène représente Spot dans une pièce comme montré dans la figure 26 et que celui-ci est très bien détaillé et possède même son bras articulé présent figure 27.

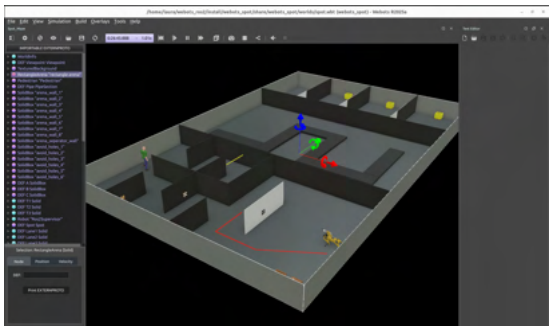


FIGURE 26 – Visualisation de la scène dans Webots

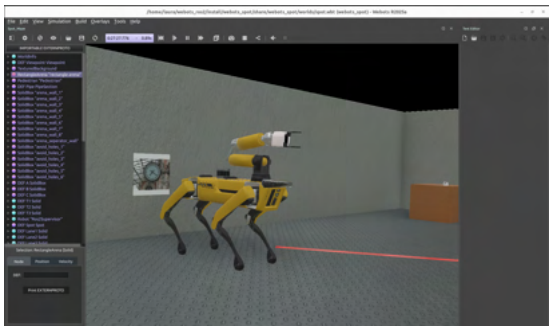


FIGURE 27 – Visualisation de Spot dans Webots

Comme avec Gazebo, dans un second temps j'ai essayé de commander Spot au clavier pour voir si ses déplacements étaient fluides et cela était bien la cas. Grâce à RViz, j'ai également pu visualiser les joints et les transformers qui constituent Spot comme présenté figure 28 et avoir accès aux données relevées par ses capteurs, notamment ses LiDARs dont les données sont visibles figure 29.

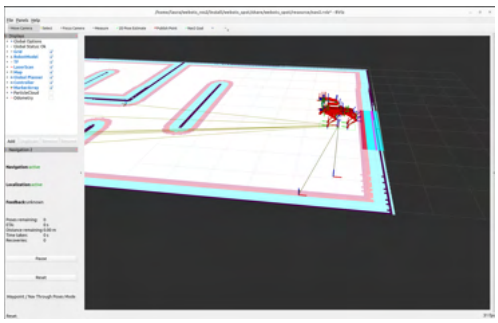


FIGURE 28 – Visualisation de Spot dans RViz

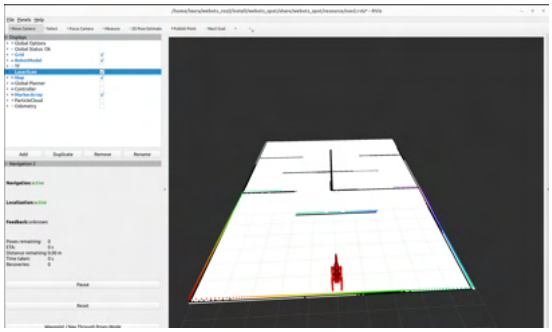


FIGURE 29 – Visualisation des données LiDAR dans RViz

Ensuite, tout comme dans Gazebo avec CHAMP, j'ai pu créer une carte de l'environnement grâce à une méthode de SLAM. La figure 30 montre la carte créée par Spot alors qu'il commence son exploration et la figure 31 présente la mise à jour de celle-ci après que Spot se soit déplacé grâce à un contrôle par clavier.

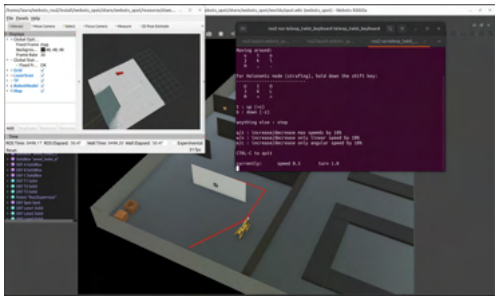


FIGURE 30 – Création de la carte par SLAM

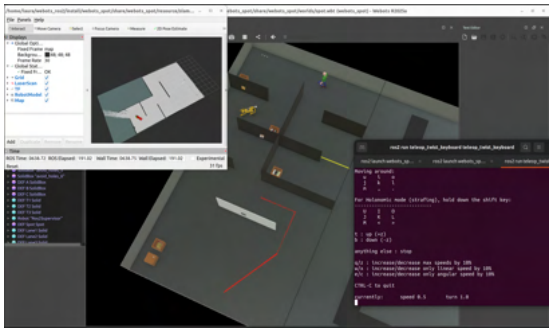


FIGURE 31 – Mise à jour de la carte par SLAM

Enfin, en ouvrant simultanément RViz et Webots, j'ai pu déplacer Spot dans la scène

de Webots grâce au clavier et visualiser les données de son LiDAR grâce à RViz comme montré dans la figure 32.

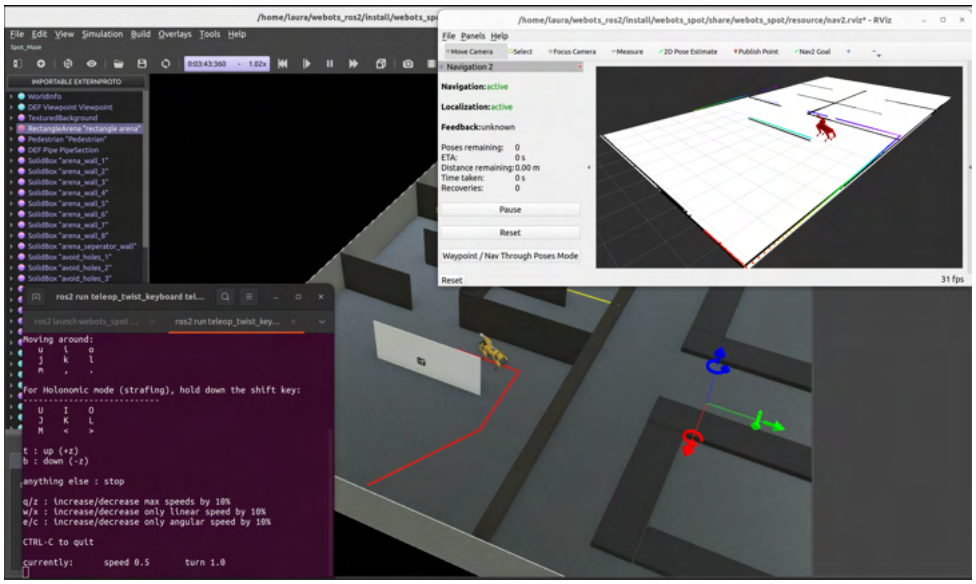


FIGURE 32 – Déplacement de Spot dans Webots et visualisation des données capteurs dans RViz

Tous ces essais d'accès aux capteurs et de fluidité des déplacements de Spot ont été concluants. De plus, ce package est vraiment très complet et permet de simuler un robot quadrupède très proche du vrai robot Spot. Enfin, en me renseignant plus en détail sur Webots, j'ai vu qu'il était assez facile de modéliser ses propres scènes. Pour la suite de mon projet, l'environnement constitué du package de Spot utilisant ROS2 sur Webots et l'intégration de nouveaux codes à cette architecture était donc le choix le plus approprié pour déployer les algorithmes développés par Well-E et simuler la navigation de Spot dans un environnement ressemblant à celui de fermes laitières.

4 Intégration et développement d'algorithmes

Dans cette partie, j'ai ajouté plusieurs codes et modifié l'architecture du répertoire que j'avais clôné [21]. Tous les codes que j'ai créés et ajoutés se trouvent sur ce git du laboratoire de Bioinformatique de l'UQÀM [25]. L'annexe G explique comment éxecuter tous les codes sur son ordinateur.

4.1 Prise en main de Webots et du package de Spot

Avant de me lancer dans l'intégration et le développement d'algorithmes, j'ai dû prendre en main le package de Spot sur Webots afin de découvrir tous les topics et services disponibles et de comprendre comment les utiliser afin de pouvoir y intégrer les algorithmes de la chaire et développer les miens. Pour faire cela, j'ai tout d'abord essayé de modéliser ma propre scène sur Webots. À terme le robot devra être utilisable en intérieur et en extérieur, qui sont deux environnements différents notamment en ce qui concerne le type de sol ou le nombre d'obstacles à éviter mais, dans un premier temps, j'ai décidé de modéliser un environnement simple constitué de 10 vaches situées dans une pièce fermée. Je me suis donc inspirée du fichier `spot.wbt` créant la scène dans le package existant pour créer mon propre fichier `spot2.wbt` dans le dossier world. Après quelques modifications du fichier `spot_launch.py` [25], j'ai réussi à rendre Spot fonctionnel et contrôlable au clavier comme montré figure 33.

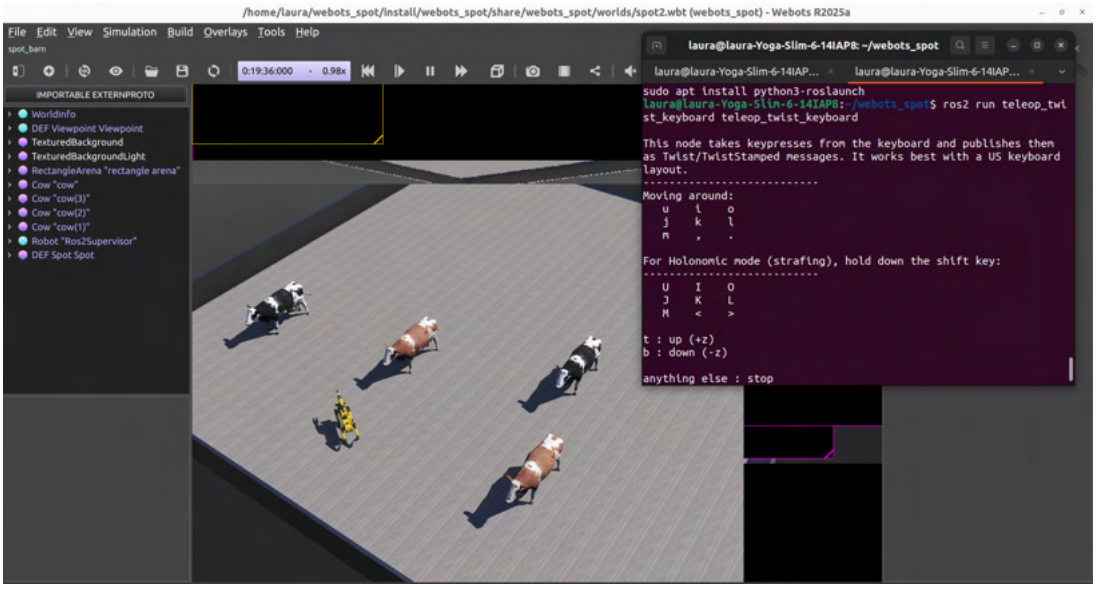


FIGURE 33 – Contrôle de Spot au clavier dans la nouvelle scène avec des vaches

L'accès aux capteurs de Spot était toujours disponible comme le montrent `rqt_image_view` figures 34 et 35 qui nous permettent d'avoir les vues de toutes les caméras de Spot, ce qui nous sera ensuite utile pour se déplacer dans l'étable et suivre des vaches. La figure 34 montre par exemple une image couleur de la caméra présente sur le bras de Spot et la figure 35 une image 3D présentant la profondeur sur le côté droit de Spot. Au final, nous avons accès à 6 images 3D (kinect, côté droit du corps, côté gauche du corps, côté droit de la tête, côté gauche de la tête et arrière de Spot) et à 7 images couleurs (aux mêmes endroits avec le bras en plus).

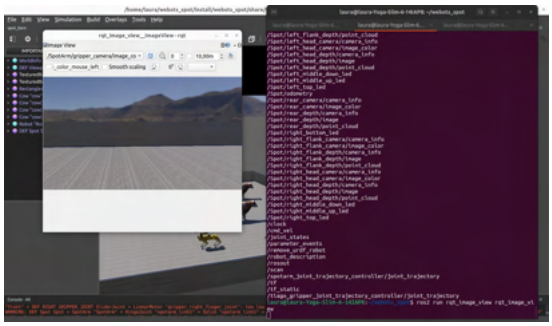


FIGURE 34 – Caméra RGB sur le bras de Spot

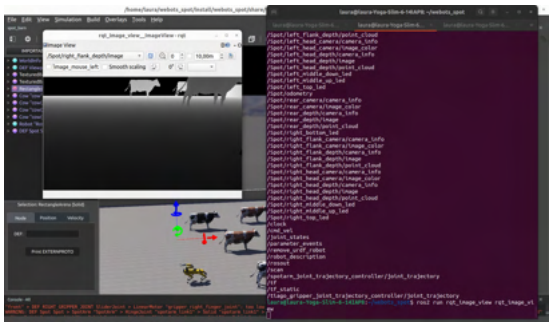


FIGURE 35 – Caméra 3D sur le côté droit de Spot

Je me suis ensuite intéressée aux services et topics disponibles pour faire bouger Spot car, à terme, le but est de contrôler ses déplacements en fonctions des vaches qu'il doit observer. Tout d'abord, les services `Spot/sit_down`, `Spot/lie_down` et `Spot/stand_up` nous permettent de choisir les positions de Spot. Ainsi, avec une ligne de commande nous pouvons faire en sorte que Spot s'asseye, se couche ou se relève comme montré dans la figure 36. J'ai également créé et intégré à l'achitecture ROS les codes `lie_down_service.py` et `stand_up_service.py` afin d'automatiser ces actions [25]. La figure 37 présente par exemple le résultat du code `stand_up_service.py`.

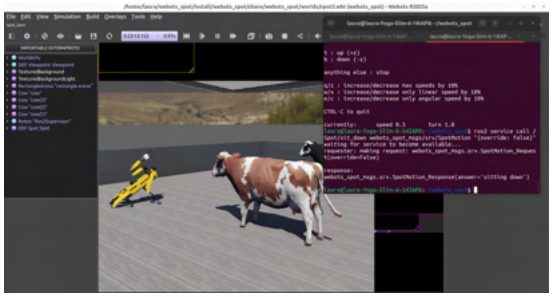


FIGURE 36 – Spot couché grâce à une commande dans le terminal et au service `Spot/sit_down`

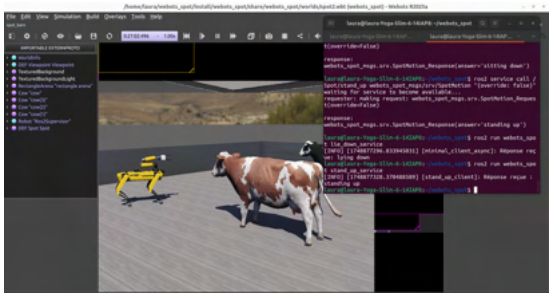


FIGURE 37 – Spot debout grâce au code `stand_up_service.py` et au service `Spot/stand_up`

Ensuite, je me suis penchée sur les topics disponibles et j'ai essayé d'utiliser des publishers pour faire bouger Spot. Dans un premier temps, grâce à une commande dans le terminal j'ai fait avancer et arrêter Spot grâce au topic `cmd_vel` comme on peut le voir sur la figure 38. Puis, j'ai créé deux scripts Python afin d'automatiser les déplacements de Spot en ajoutant ces codes à l'architecture ROS. `send_cmd_vel.py` permet de faire avancer Spot et `move_sequence.py` le fait avancer, reculer et tourner afin de tester plusieurs déplacements comme le montre la figure 39 [25].

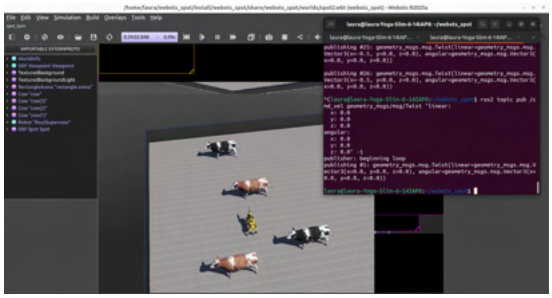


FIGURE 38 – Spot s'arrêtant grâce à une commande dans le terminal et au topic `cmd_vel`

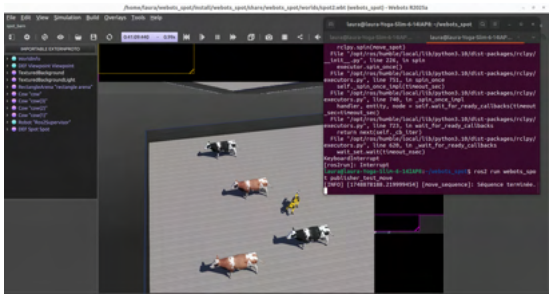


FIGURE 39 – Spot avançant, reculant et tournant grâceau code `publisher_vel` et au topic `cmd_vel`

Enfin, j'ai regardé à quels topics je pouvais m'abonner à l'aide de suscribers et j'en ai trouvé plusieurs assez intéressants pour la suite de mon projet. Le premier est le topic `/joint_states` qui permet d'avoir accès aux positions et à l'état des joints de Spot. Le second est le topic `/scan` qui contient les distances entre Spot et les obstacles situés tout autour de lui et qui me permettra de mesurer les distances entre Spot et les vaches afin qu'il reste assez éloigné d'elles. De plus, ce topic permet notamment d'avoir accès à la distance à l'avant de Spot, ce qui me sera utile quand je voudrai suivre des vaches. Pour faire cela, j'ai donc créé le script `scan_distance_front.py` [25] en calculant quelle ligne de ranges correspond à l'avant de Spot grâce à `angle_min` et `angle_max` donnés par le topic `/scan` comme montré dans les figures 40 et 41.

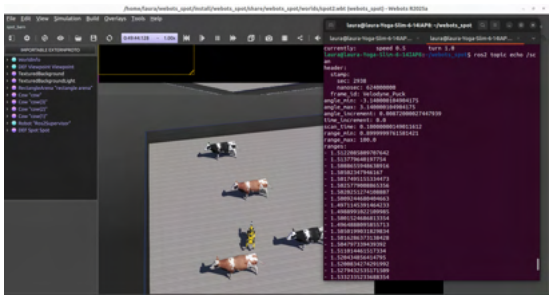


FIGURE 40 – Détails du topic permettant d'avoir accès aux données LiDAR

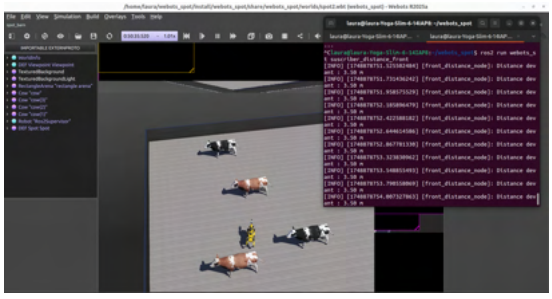


FIGURE 41 – Accès à la distance à l'avant de Spot grâce au code `scan_distance_front.py`

Grâce à tous ces tests d'accès aux différents topics et services, j'ai donc réussi à accéder à des éléments indispensables pour la suite de mon projet tels que faire bouger Spot, avoir accès à ses caméras et connaître les distances autour de lui. La prochaine étape était donc de me servir de ces éléments pour intégrer à cette architecture les algorithmes développés par Well-E.

4.2 Intégration des algorithmes de Well-E à l'architecture

Dans le cadre des activités de la chaire Well-E, l'un des objectifs est d'explorer des algorithmes développés pour la détection et l'identification des vaches dans des cas d'utilisation dynamiques, comme ce le sera avec le robot Spot. Avant de pouvoir développer des algorithmes de navigation de Spot qui utiliseraient la détection et l'identification des vaches, j'ai donc d'abord dû intégrer ces deux algorithmes déployés par Well-E

dans l'architecture. Dans le package open-source que j'ai sélectionné [21], un code Python `yolov8_openvivo.py` était déjà conçu pour utiliser YOLOv8 afin de détecter l'une des 78 classes du dataset Coco dont la classe 19 qui est celle des vaches [25]. En organisant bien les fichiers et en les ajoutant à l'architecture ROS2, j'ai réussi à utiliser ce script afin de détecter les vaches présentes sur la caméra du bras de Spot comme montré figure 42. Même si ce modèle fait quelques erreurs de classification en fonction de l'orientation des vaches, il semble être très performant pour les détecter, et ce même dans des positions plus compliquées avec des jeux de lumière ou encore avec seulement une partie de la vache visible comme présenté figure 43.

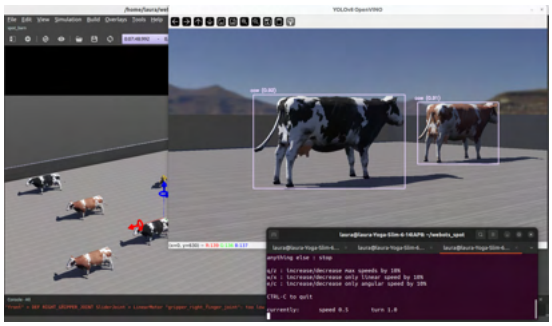


FIGURE 42 – Utilisation de YOLOv8 sur l'image de la caméra du bras de Spot

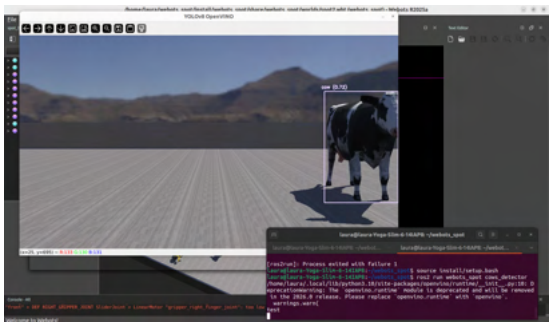


FIGURE 43 – Bonne classification malgré une orientation de la vache complexe

Dans un second temps, j'ai souhaité utiliser notre propre modèle développé dans le cadre des activités de la chaire Well-E pour détecter les vaches dans la simulation car, l'objectif à terme est de déployer ce modèle de détection [3] sur le vrai robot Spot. Après avoir converti le fichier .pt du modèle en fichier .onnx nécessaire pour utiliser la même structure que le script `yolov8_openvivo.py`, j'ai pu utiliser le modèle de détection développé par Araùjo et al. sur Spot grâce à mon script `yolov8_well_e.py` [25] comme le montre la figure 44. Cependant, j'ai constaté que ce modèle était moins performant que le modèle initial pour détecter les vaches, même lorsque celles-ci se présentent avec des orientations simples comme on peut le voit sur la figure 44. Cela peut s'expliquer car ce modèle a été entraîné avec des images de vraies vaches et avec un background vraiment différent de celui de la simulation. À terme, pour obtenir de meilleures performances avec le modèle de Araùjo et al. dans la simulation, nous pourrions essayer de modifier la scène pour la rendre très ressemblante à la réalité mais, cela semble difficile d'obtenir ce résultat. La solution est donc d'utiliser le modèle YOLO initial du package disponible au public dans la simulation car, pour développer les algorithmes de navigation, il faudra que la détection dans la simulation soit aussi performante que dans la réalité. De plus, notre but premier qui est de pouvoir utiliser le modèle de Araùjo et al. sur le robot réel est atteint car celui-ci a été intégré à l'architecture et, nous aurons donc juste à préciser quel algorithme utiliser en fonction d'un déploiement en simulation ou en milieu réel.

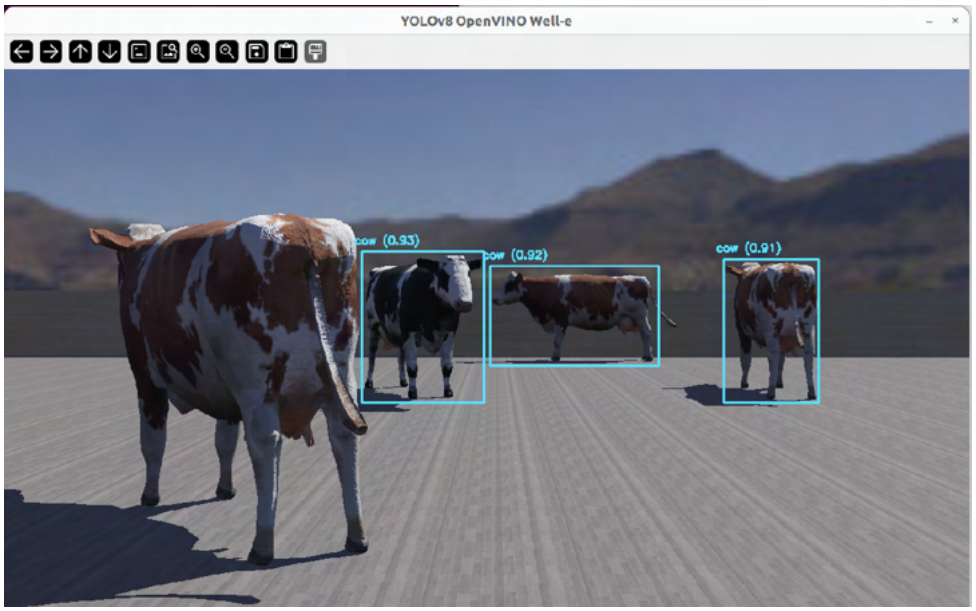


FIGURE 44 – Utilisation de YOLOv8 spécialisé dans la détection des vaches entraîné dans le cadre de la chaire Well-E [3]

Dans un second temps, j'ai voulu intégrer le modèle d'identification de vaches de la chaire à mon architecture. Le principe de mon script `detection_and_tracking.py` est que j'utilise d'abord le modèle de détection qui donne une bounding box en déterminant les coordonnées de l'objet qu'il a identifié puis, sur chaque vache détectée, j'applique le modèle d'identification comme on peut le voir sur la figure 45 [25]. La figure 46 montre un exemple de détection et d'identification réalisé dans la simulation. Au dessus de chaque bounding box, on retrouve l'indice de confiance de la détection ainsi que celui de la classification.

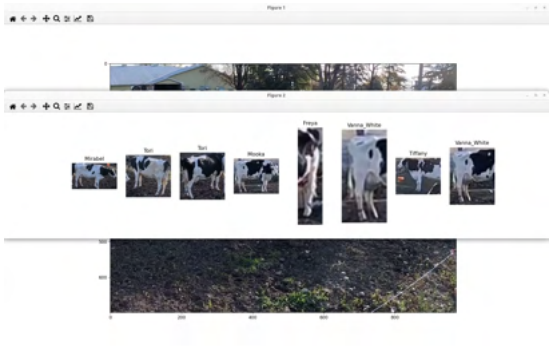


FIGURE 45 – Principe de la détection et de l'identification des vaches

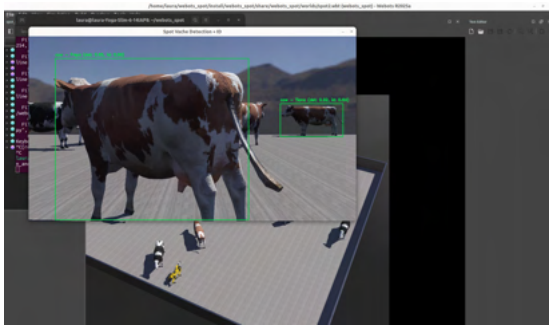


FIGURE 46 – Exemple de détections et d'identifications de vaches

Les modèles de détection et d'identification de vaches ayant été intégrés, j'ai maintenu pu passer à la phase de développement et d'intégration d'algorithmes de navigation pour Spot.

4.3 Développement et intégration d'algorithmes de navigation

Les algorithmes de navigation vont être essentiels pour répondre aux besoins de la chaire et, à terme, déployer des robots autonomes au milieu des vaches. Après avoir discuté

avec des membres du laboratoire CowLife de sciences animales, je me suis rendue compte qu'il allait falloir développer plusieurs algorithmes de navigation différents en fonction de leurs besoins.

4.3.1 Suivi de la vache la plus proche

J'ai commencé par développer un algorithme de suivi de vache la plus proche du robot. D'un point de vue informatique, cela permet d'avoir un critère de sélection initial pour définir la vache cible afin de développer un premier algorithme de suivi de vache. D'un point de vue sciences animales, cela permettra de suivre certaines vaches afin de les surveiller et d'observer leur comportement. Le pipeline de mon algorithme est montré dans la figure 47.

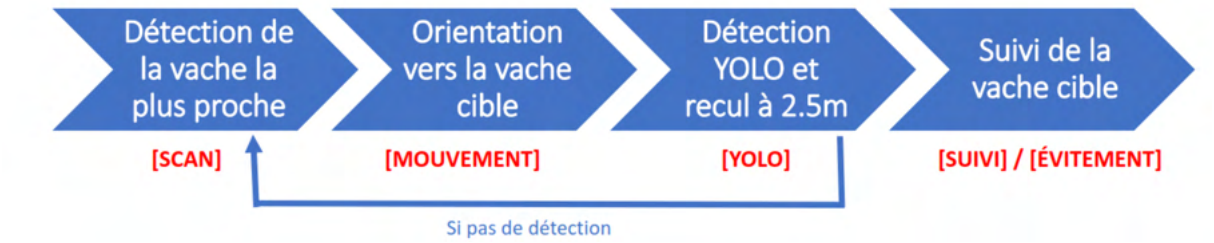


FIGURE 47 – Pipeline de l'algorithme de suivi de la vache la plus proche

Dans le script `suivi_vache_proche.py` Spot commence en mode [SCAN] où il scanne son environnement afin de trouver la vache la plus proche [25]. Spot passe ensuite en mode [MOUVEMENT] et s'oriente face à l'objet détecté comme montré dans la figure 48. Puis, il active la détection de vaches avec YOLOv8 en mode [YOLO] et il recule ou avance de manière à se placer à 2.5m de la vache, distance choisie à la fois pour ne pas s'approcher trop près de la vache et pour avoir la vache en entier dans la caméra. Si une vache est bien détectée, Spot passe alors en mode [SUIVI] comme montré dans la figure 49. Sinon, cela signifie que l'objet scanné n'est pas une vache (cela peut être un mur, un arbre en extérieur, un objet dans l'étable, etc.) et il repasse alors en mode [SCAN] pour choisir une autre cible. Dans le mode [SUIVI], Spot se déplace de manière à toujours garder le centre de la bounding box au centre de sa caméra et à rester à environ 2.5m de la vache cible.

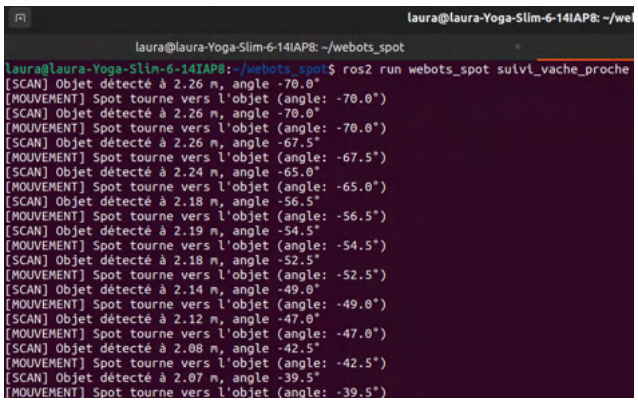


FIGURE 48 – Détection et orientation

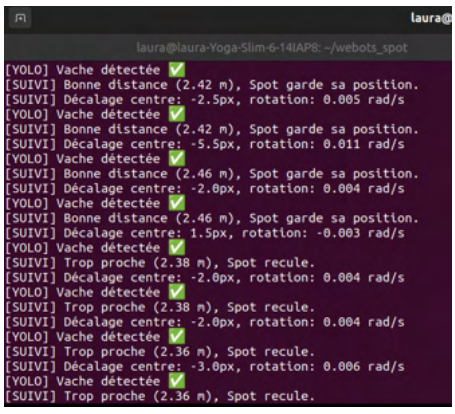


FIGURE 49 – Suivi de la vache cible

Le développement de ce script a amené plusieurs défis pour lesquels j'ai dû trouver des solutions. Le premier était que Spot se mettait à changer de cible lorsque d'autre vaches apparaissaient à la caméra. Une solution possible était de suivre tout le temps la vache avec la plus grande bounding box car j'ai supposé qu'étant la vache la plus proche, sa bounding box serait plus grande que les autres mais ce n'était pas tout le temps le cas. J'ai donc fait en sorte que sa cible reste la bounding box dont le centre était le plus au centre de l'image de la caméra étant donné que les vaches bougent assez lentement et que j'ai donné des consignes de mouvement à Spot également très lentes. Cette seconde solution a présenté les résultats attendus pour Spot..

Le deuxième défi était que, lors de sa phase de suivi, Spot se faisait rentrer dedans par d'autres vaches dont le mouvement est aléatoire et se faisait renverser et, bien que Spot soit capable de se relever, cela pourrait l'abimer. Dans la réalité, on ne sait pas encore quel sera le comportement des vaches vis à vis de Spot mais elles pourront tout à fait se diriger vers lui comme dans la simulation. J'ai donc ajouté un mode de détection des obstacles latéraux (obstacle autour de Spot, sauf dans un cône frontal de plus ou moins 30 degrés) qui utilise les données LiDAR. Si un objet est détecté à moins de 3m de Spot, sauf dans son cône frontal, il se déplace dans la direction opposée à cet objet tout en gardant sa vache cible au centre de sa caméra. La figure 50 montre par exemple Spot qui détecte une vache trop près de lui sur sa droite et qui s'en éloigne donc tout en continuant de suivre la vache cible qui se situe devant lui. Enfin, j'ai ajouté la priorité au suivi de la vache cible car, dans certains cas, lorsque Spot avait une vache à sa droite et une à sa gauche, il n'avançait plus et perdait la vache cible de vue.

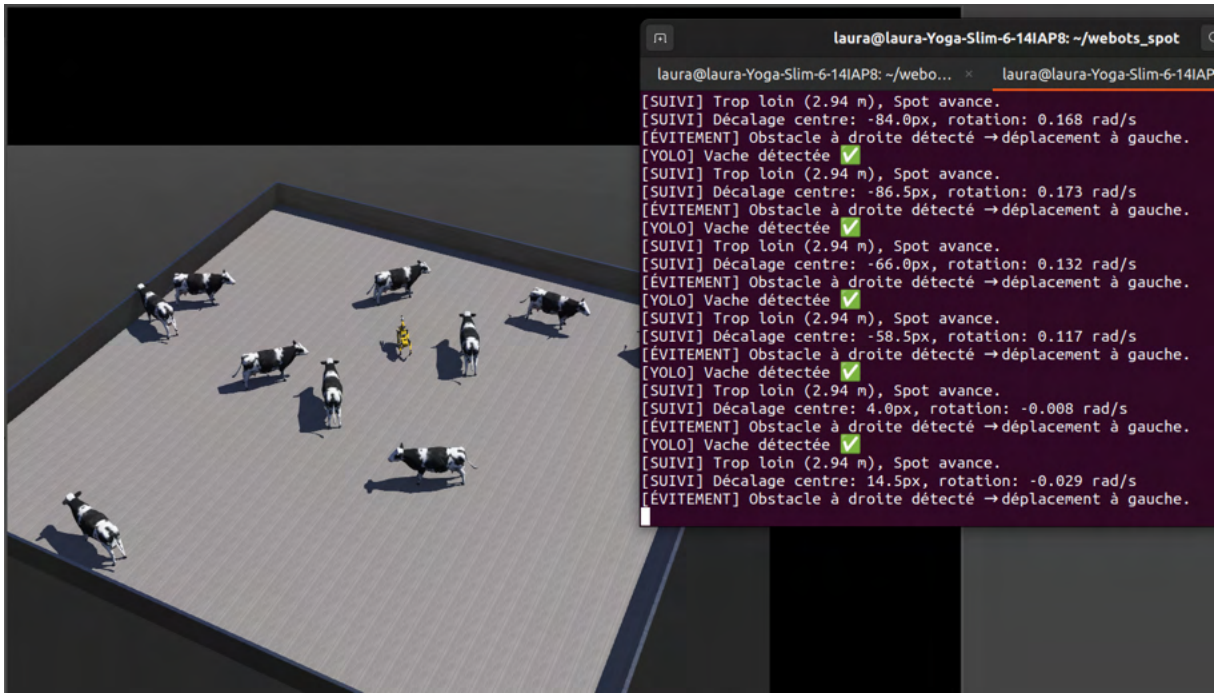


FIGURE 50 – Évitement de l'obstacle détecté à droite

Pour illustrer la performance de l'algorithme de suivi de vache la plus proche, une vidéo de démonstration est fournie [27]. Des futures adaptations du script pourront être réalisées lors de l'avancement du projet et avec la spécification plus détaillée des besoins du laboratoire CowLife. Il sera par exemple possible d'ajouter un paramètre de temps

de suivi de la vache pour changer de vache cible régulièrement afin d'observer toutes les vaches. De plus, des études devront être menées pour déterminer le comportement des vaches vis à vis de Spot. En effet, si celles-ci s'approchent de lui pour le lécher ou le mordre, on pourrait faire en sorte que Spot se couche afin d'éviter qu'il tombe et qu'il se casse.

4.3.2 Suivi d'une vache spécifique

Dans un second temps, j'ai développé un algorithme plus complexe pour faire en sorte que Spot trouve une vache spécifique et la suive. D'un point de vue sciences animales, cela sera très utile, par exemple si le bien-être physique ou mental d'une vache en particulier doit être vérifié. L'objectif est de permettre à Spot de surveiller la vache X toute la journée ou durant un temps donné. Le pipeline pour ce script est montré dans la figure 51.

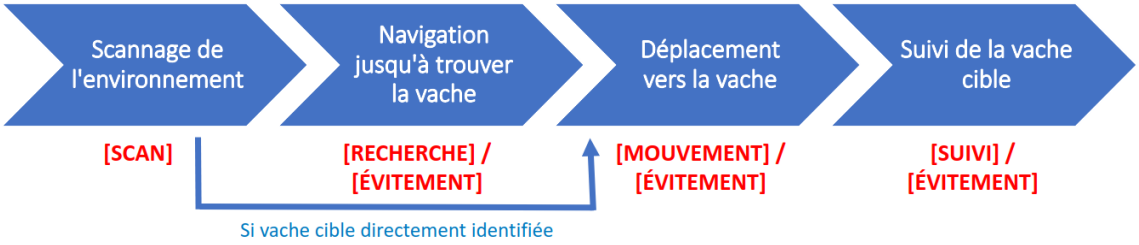


FIGURE 51 – Pipeline de l'algorithme de suivi d'une vache spécifique

Le principe du script `suivi_vache_specifique.py` [25] est que Spot doit tout d'abord activer sa caméra sur son bras et utiliser l'algorithme de détection et d'identification. S'il détecte la vache cible indiquée par l'utilisateur, il se dirige vers elle tout en évitant les autres vaches et entre en mode suivi lorsqu'il est proche d'elle, comme dans le script précédent. Sinon, il se déplace jusqu'à trouver la vache cible puis se rapproche d'elle et la suit.

Le problème avec ce pipeline et la simulation est que j'utilise l'algorithme d'identification des vaches pour trouver celle que je souhaite faire suivre. Or, si cet algorithme fonctionne très bien dans la réalité, il est inutilisable dans la simulation car les vaches sont toutes les mêmes. À terme, dans une simulation très représentative de la réalité, nous pourrons insérer des vaches très ressemblantes aux vraies en leur appliquant une skin grâce à des photos prises de tous les angles des vaches réelles mais, même avec cela, ce n'est pas sûr que l'algorithme d'identification fonctionne dans la simulation. De plus, cela voudrait dire que si l'on change de ferme, il faudra refaire la scène avec les vaches spécifiques de cette ferme. Il vaut mieux donc utiliser un moyen de différencier les vaches dans la simulation de manière indépendante du physique réel des vaches des fermes. Cela nous permettra de ne pas avoir à adapter la simulation en fonction des fermes et de garder le même pipeline que pour le robot réel, qui lui utilisera des modèles d'identification entraînés pour chaque fermes. Ainsi, pour pouvoir utiliser ce même pipeline il fallait trouver une alternative pour identifier les vaches donc j'ai commencé par ajouter des éléments distinctifs aux vaches. Ces éléments devaient être visibles sous tous les angles donc j'ai décidé de mettre deux boules de couleur sur chaque vache en créant une autre scène avec le fichier `spot2_simu.wbt` [25] comme montré dans la figure 52.

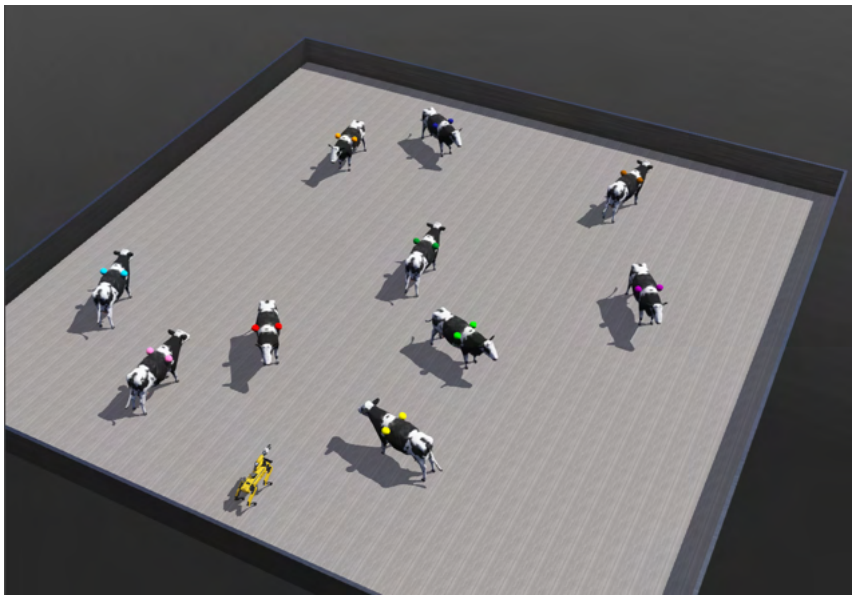


FIGURE 52 – Nouvelle scène avec des vaches différenciées par des boules de couleurs différentes

Ensuite, j'ai créé un dataset pour entraîner YOLOv8 à reconnaître les couleurs des boules pour permettre à Spot de trouver une vache en lui indiquant une couleur de boule de la même manière que, dans la réalité, on lui indiquera un nom de vache. Pour mon dataset, j'ai labelisé 500 images (50 de chaque couleur) avec le logiciel Labelme [43]. Pour obtenir ces images, j'ai fait bouger Spot dans la scène au clavier et j'ai pris 50 captures d'écran par couleur en faisant en sorte d'avoir le plus de points de vue différents des vaches. J'ai ensuite converti les fichiers d'annotation `.json` fournis par Labelme en fichiers .txt demandés par YOLO et j'ai constitué deux jeux de données avec des classes équitablement réparties (un jeu d'entraînement constitué de 350 images et un jeu de validation constitué de 150 images) pour lequels l'entraînement du modèle a donné de bonnes performances de classification. Tout d'abord, en observant des exemples de prédictions d'un batch d'images dans le jeu de validation sur la figure 53, on voit la grande majorité des prédictions sont correctes.

FIGURE 53 – Exemples de prédictions dans le jeu de validation

Les figures 54 et 55 montrent les bonnes performances obtenues par ce modèle d'identification des vaches avec les boules de couleur. Par exemple, la figure 54 présente la matrice de confusion qui permet d'observer les vrais et faux positifs et les vrais et faux négatifs. Sur celle-ci nous pouvons voir que seules deux boules n'ont pas été détectées. La bonne nouvelle est qu'elles n'ont pas pas été classées dans des mauvaises classes, ce qui est très important dans notre cas. En effet, ne pas détecter la vache X que l'on cherche n'est pas très grave car il suffira pour Spot de continuer de se déplacer jusqu'à la trouver. En revanche, Spot ne doit surtout pas se tromper de classe en faisant des faux positifs car, s'il pense surveiller la vache X alors qu'il surveille la vache Y, cela pourra conduire à des erreurs de diagnostic pour les vaches. Le pourcentage de vrais positifs est donc très important dans notre étude et, pour regarder sa valeur, je me suis penchée sur la courbe de précision en fonction de l'indice de confiance présentée sur la figure 55. La précision est le rapport pour chaque classe entre le nombre de vrais positifs sur le nombre total de positifs détectés. La valeur de précision est bornée entre 0 et 1 et une précision est considérée comme bonne à partir de 0.80, très bonne à partir de 0.90 et excellente à partir de 0.95. Pour chaque détection effectuée par le modèle, un indice de confiance est indiqué à côté de la bounding box. La courbe ci-dessous indique la précision en fonction de cet indice de confiance, ce qui nous permettra de choisir le seuil de performance le plus approprié pour la tâche. Dans notre cas, nous voyons qu'un seuil de 0.1 pour l'indice de confiance donne déjà une très bonne précision. La précision moyenne devient excellente vers un indice de confiance de 0.7 et atteint 1 pour 0.983. Ces résultats étant excellents, je n'ai pas fixé de seuil d'indice de confiance dans le script mais, si plus tard on voit que Spot se trompe de cible de temps en temps, je pourrais fixer un tel seuil afin qu'il sélectionne sa cible seulement si l'indice de confiance est supérieur à un certain seuil.

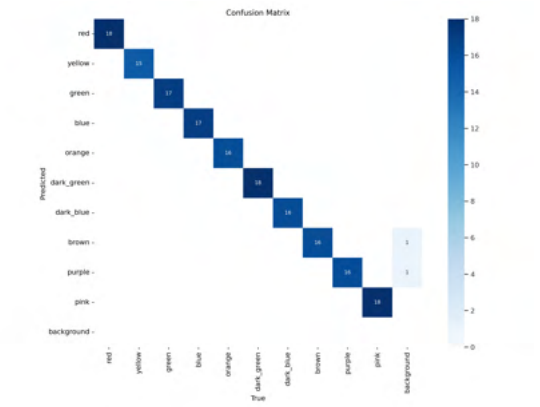


FIGURE 54 – Matrice de confusion

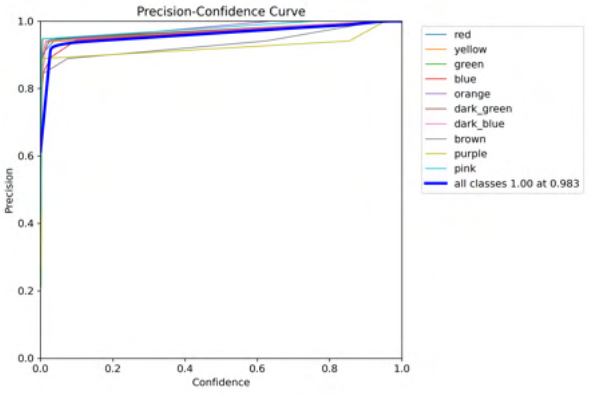


FIGURE 55 – Courbe de précision

Le script `dection_and_tracking_simu_yolo_entraine.py` [25] permet de tester le modèle entraîné dans la simulation. Comme dans le script `detection_and_tracking.py` qui utilise le modèle d'identification de Araùjo et al., ce script utilise le modèle de détection et, pour chaque détection, applique ensuite le modèle d'identification de balles. Ces deux scripts sont identiques sauf que `detection_and_tracking_simu_yolo_entraine.py` est fait pour la simulation et utilise le modèle de détection du package ainsi que le modèle d'identification que j'ai entraîné pour la scène avec les vaches et les boules et le script `detection_and_tracking.py` est fait pour la réalité et utilise les modèles d'identification et de détection de Araùjo et al.. Avec mon sccript, une fenêtre montre bien les détections de vaches et l'identification de la couleur de boule de chaque vache avec les indices de

confiance, comme montré dans la figure 56.

FIGURE 56 – Détection et identification avec le modèle entraîné

Ce problème d'identification étant réglé, j'ai pu ensuite me pencher sur le code `suivi_vache_specifique.py`. Comme montré dans le pipeline de la figure 51, Spot commence en mode [SCAN] en activant d'abord la caméra de son bras et il utilise les modèles de détection et d'identification afin de trouver la vache que l'utilisateur lui indique en ligne de commande en lançant le code de la manière suivante :

```
$ ros2 run webots\_spot suivi\_vache\_specifique --ros-args -p target\_color
    ↪ :=red
```

Si Spot ne voit pas la vache cible, pour l'instant il tourne sur lui-même en mode [RECHERCHE] et on voit la mention "SEARCHING_TARGET" dans la fenêtre de visualisation de la caméra et des détections comme montré dans la figure 57. Lorsque Spot repère sa cible, il passe dans la phase [MOUVEMENT] et se dirige vers la vache cible tout en évitant les autres vaches. Dans la fenêtre de visualisation, on voit la mention "TRACKING_TARGET" et un cadre "TARGET" permet de visualiser la cible de Spot comme montré dans la figure 58. Une fois la cible atteinte, le but est de passer en mode [SUIVI] comme dans le code précédent. Enfin, dès qu'il se déplace, Spot doit être en mode [ÉVITEMENT] pour ne pas rentrer dans les autres vaches.

FIGURE 57 – Mode `SEARCHING_TARGET`

FIGURE 58 – Mode `TRACKING_TARGET`

Un autre défi qui est alors apparu avec ce code est que j'utilise deux modèles (détection et identification) et que cela est trop lourd pour mon ordinateur. La vidéo suivante

[28] démontre ce défi lorsque Spot finit par trouver la vache rouge que nous lui avons demandé de suivre mais qu'il finit par la perdre car, étant données les ressources computationnelles, les images de la caméra sont saccadées et diffèrent trop de l'une à l'autre. Je n'ai d'ailleurs pas pu faire un enregistrement vidéo de mon écran car mon ordinateur ne le supportait pas en même temps que le lancement du script. Spot retrouve ensuite la vache cible perdue en effectuant un autre tour sur lui-même mais cela prend beaucoup de temps. Une solution possible est d'utiliser le modèle d'identification seulement dans les phases [SCAN] et [RECHERCHE] puis de ne garder que le modèle de détection dans les phases [MOUVEMENT] et [SUIVI] mais, lors de certains tests, cela entraînait des changements de cible lorsque la vache cible était trop loin. L'exploration de ce défi n'était plus possible étant données les contraintes de temps de mon stage mais, pour la suite, il faudrait trouver un moyen de ne pas perdre la vache cible malgré la latence du retour caméra due à l'utilisation coûteuse des deux modèles en même temps. Ensuite, il faudrait également développer un algorithme de recherche de la vache cible plus optimal car, pour l'instant, Spot se contente de tourner sur lui même. On pourrait même développer une méthode de SLAM afin que Spot puisse cartographier son environnement et se déplacer plus efficacement en recherchant les vaches.

5 Augmentation de données vidéos

En parallèle de mon projet principal de développement d'un simulateur robotique, j'ai travaillé sur un projet d'augmentation de données qui a réquisitionné le travail de tout le département de bio-informatique de l'UQÀM.

5.1 Explication du projet

Ce projet d'augmentation de données vidéos de vaches est innovant car, dans le domaine des sciences animales, rien de tel n'a jamais été fait. Pourtant, dans les 3/4 des projets de recherches réalisés dans ce domaine, la conclusion est que les résultats ne sont pas bons ou pas valides car les jeux de données étaient trop petits.

Les départements de sciences animales de McGill et de bio-informatique de l'UQÀM ont décidé de changer les choses et de se lancer dans un projet d'augmentation de données avec des vaches afin d'obtenir de meilleurs résultats pour une expérience qui n'avait pas donné de résultats satisfaisants. Cette expériences a été réalisée récemment par le département de sciences animales auprès de 10 vaches durant deux semaines et son but était de proposer deux jouets kongs dont l'un est présent sur la figure 59 aux vaches et de déterminer si elles avaient une préférence pour l'un ou l'autre. Cependant, les résultats ont été mitigés et le département de sciences animales a donc mobilisé le département de bio-informatique pour augmenter les données car il leur était impossible d'aller reprendre des vidéos de vaches, cela nécessitant trop de temps (les vaches passent en moyenne seulement 7% de leur temps à intéragir avec les kongs) et de moyens humains et matériels. De plus, étant donné qu'il existe déjà beaucoup de vidéo à annoter, le but est de générer directement des données annotées afin de ne pas rajouter du travail.

FIGURE 59 – Vache et kong

Le département de bio-informatique de l'UQÀM a été partagé en trois groupes :

- Augmentation of Video Data : augmentation des données vidéos
- Augmentation of Bounding Box Data : annotation automatique en identifiant le comportement et la posture à partir de 3 bounding boxes (corps, tête et museau)

— Behaviour Annotation Data Augmentation and Generation : génération d'enchaînements de séquences d'actions et de comportements réalistes dans des tableurs (génération d'éthogramme)

Durant mon stage, j'étais dans le groupe numéro un dont le but était de faire de l'augmentation de données vidéos. En effet, afin d'améliorer la robustesse statistique et l'interprétabilité de la recherche comportementale sur les animaux, le département des sciences animales a besoin d'un plus grand nombre de vidéos du comportement des vaches car, comme mentionné précédemment, seules 10 vaches ont été incluses dans leur étude, ce qui limite la puissance statistique et rend plus difficile la détection de différences comportementales significatives. Dans un premier temps, nous avons donc fait de l'augmentation de données classique puis nous avons utilisé des méthodes basées sur le machine learning. Tous les codes cités sont présents sur le git au lien suivant [26].

5.2 Augmentation de données classique

Dans un premier temps, mon groupe s'est concentré sur de l'augmentation de données vidéos classique à partir d'une vidéo d'une vache de 45min. Une personne était chargée de découper cette vidéo en 9 vidéos de 5min afin d'avoir plus de vidéos. Une autre devait modifier la luminosité et le contraste de la vidéo tandis que quelqu'un d'autre devait s'occuper de modifier la netteté et le bruit. Mon rôle était de remplacer ou de modifier l'arrière plan de la vidéo en utilisant des modèles de segmentation. Cela est très intéressant car, avec de nouveaux arrière-plans, il est possible de simuler divers environnements, ce qui améliore la robustesse de l'ensemble des données sans nécessiter davantage d'enregistrements qui prennent du temps, nécessitent de l'équipement et peuvent perturber les vaches. Tout d'abord, je n'ai segmenté que la vache et le kong puis, dans un second temps, le département de sciences animales nous a dit qu'il fallait aussi segmenter la corde et la chaîne. Je présente donc ces deux méthodes dans les parties suivantes.

5.2.1 Segmentation de la vache et du kong

Tout d'abord, dans le code principal `change_background.py` [26], je me suis concentrée sur la segmentation de la vache et du kong. J'ai commencé par extraire les frames d'un extrait de vidéo grâce à OpenCV puis j'ai segmenté les vaches avec DeepLabV3 qui contient une classe vache. Ensuite, j'ai gardé la vache principale en supprimant les vaches les moins grandes.

Dans un second temps, je me suis intéressée à la segmentation du kong. Pour faire ceci, j'ai utilisé une méthode de traitement d'image en créant un masque en utilisant les valeurs HSV du kong. Tout d'abord, grâce au code `color_kong.py` [26] j'ai pu tester les masques que j'obtenais en faisant varier les valeurs HSV avec un curseur afin de sélectionner celles pour lesquelles il ne restait que le kong. Ensuite, j'ai reporté ces valeurs dans mon code principal pour créer un masque pour le kong. Enfin, j'ai superposé les masques du kong et de la vache et j'ai ajouté un fond noir, blanc ou composé d'une image d'une autre étable. J'ai appliqué cette méthode à chaque frame et à la fin j'ai reconstitué la vidéo avec les frames modifiées. Ce pipeline est résumé sur la figure 60.

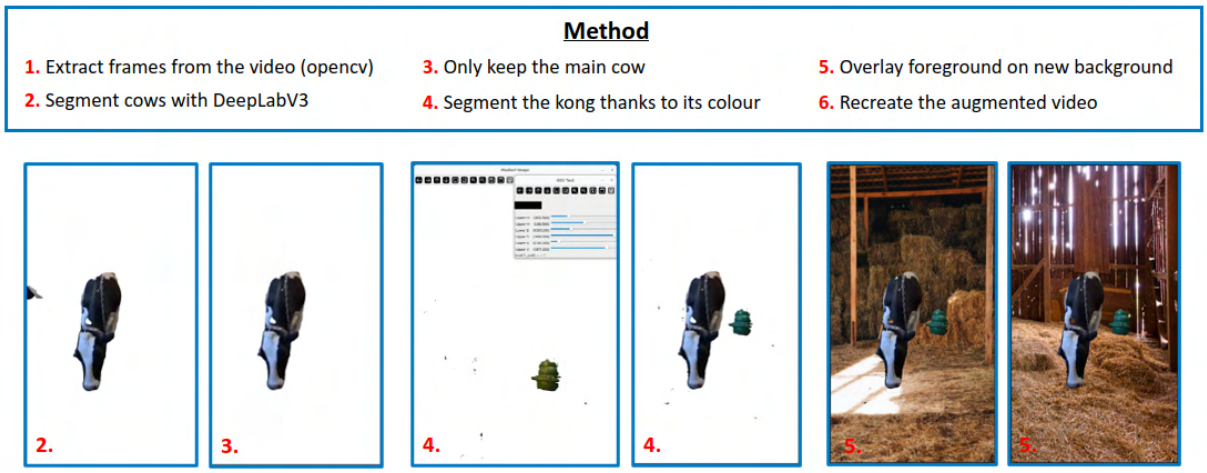


FIGURE 60 – Pipeline de segmentation de la vache et du kong

Le code était beaucoup trop lourd pour tourner sur mon ordinateur sans GPU. À titre d'exemple, pour seulement 1 min de vidéo, j'avais 1750 frames qui occupaient 17,4GB et le code mettaient environ 1h à s'exécuter. Pour pallier ce problème, j'ai donc choisi de travailler sur de courtes vidéos et de créer un fichier Jupyter Notebook que j'ai fait tourner sur Compute Canada. Compute Canada est une organisation à but non lucratif financée par la Fondation canadienne pour l'innovation et soutenue par des partenariats régionaux afin de fournir l'infrastructure numérique essentielle à l'industrie et aux chercheurs au Canada. C'est maintenant l'Alliance de recherche numérique au Canada qui a repris ces responsabilités et qui est désormais l'organisme national de coordination de l'informatique de recherche avancée, de la gestion des données de recherche et des logiciels de recherche. Ils fournissent plusieurs systèmes sur lesquel nous pouvons utiliser des CPU, des GPU ou encore des systèmes de stockage comme présenté sur la figure 61.

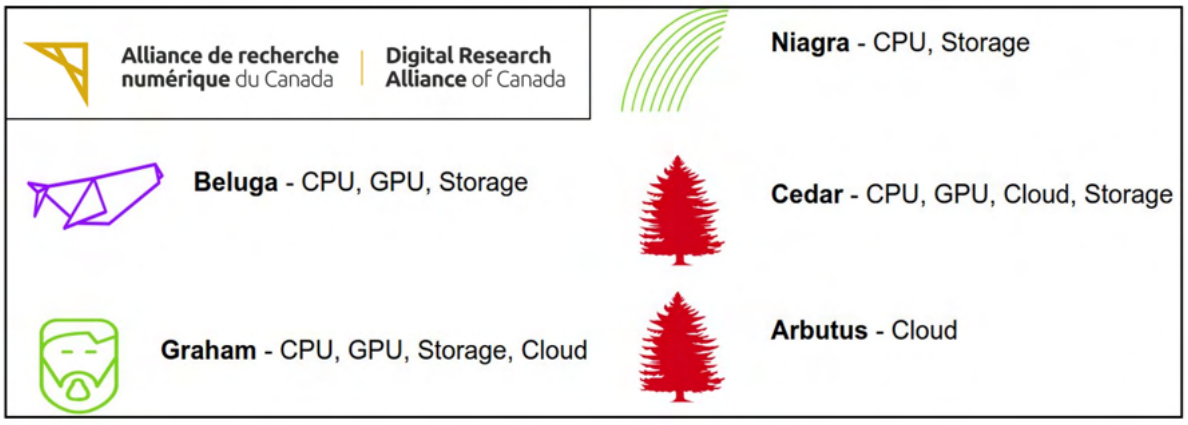


FIGURE 61 – Systèmes mis à disposition par l'Alliance de recherche numérique au Canada

Pour exécuter mon Jupyter Notebook sur Compute Canada, j'ai dû me créer un compte sur la plateforme et créer et activer un environnement virtuel dans lequel j'ai chargé tous les modules dont j'avais besoin puis installer les packages Python dont j'avais besoin. Ensuite, j'ai dû faire une demande d'allocation pour obtenir les ressources nécessaires comme présenté sur la capture d'écran de mon terminal sur la figure 62 et, une fois cette demande acceptée, j'ai créé un tunnel SSH entre mon ordinateur local et le Jupyter

Notebook. Enfin, j'ai pu faire tourner mon code et obtenir les vidéos grâce à ma méthode de segmentation.

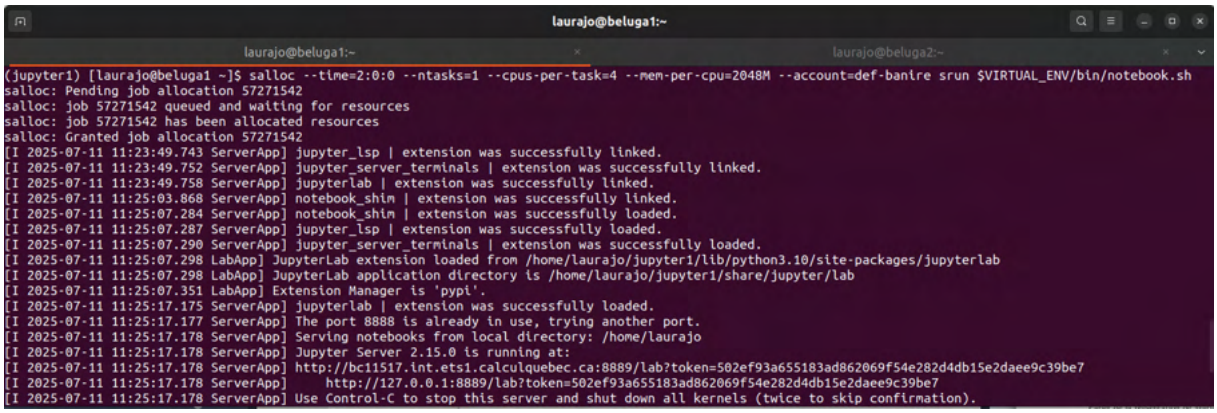


FIGURE 62 – Création d'un tunnel SSH

Les limites de cette méthode sont tout d'abord que le masque du kong ne fonctionne plus si on utilise un kong d'une autre couleur ou que la luminosité change trop au cours de la vidéo. De plus, le département de sciences animales nous a ensuite dit qu'il fallait également segmenter la chaîne et la corde mais DeepLabV3 ne possédait pas de telles classes donc j'ai dû réfléchir à une autre méthode.

5.2.2 Segmentation de la vache, du kong, de la corde et de la chaîne

Pour segmenter la vache, le kong, la corde et la chaîne, j'ai décidé d'utiliser le Segment Anything Model (SAM) développé par META. SAM propose trois sortes de segmentations : une segmentation directe et totale d'une image, une segmentation d'une zone délimitée par une bounding box et une segmentation à partir d'un point.

Tout d'abord, comme dans la méthode précédente, j'ai extrait les frames de ma vidéo grâce à OpenCV puis, dans un premier temps, j'ai donné une image de vache à SAM pour voir si la segmentation directe fonctionnait mais ce n'était pas le cas car les objets étaient segmentés en différentes parties comme on peut le voir sur la figure 63 avec le point 2. La chaîne et la corde se confondant avec l'arrière-plan, la méthode de bounding box n'était pas performante non plus. J'ai donc décidé de sélectionner moi-même des points pour segmenter les parties que je souhaitais obtenir afin de créer un masque dans le code `sam_points.py` [26]. Enfin, j'ai changé l'arrière-plan et recréé la vidéo à partir des frames modifiées. Je me suis une nouvelle fois servie de Compute Canada pour faire tourner mon code. Le pipeline de cette méthode est présenté figure 63.

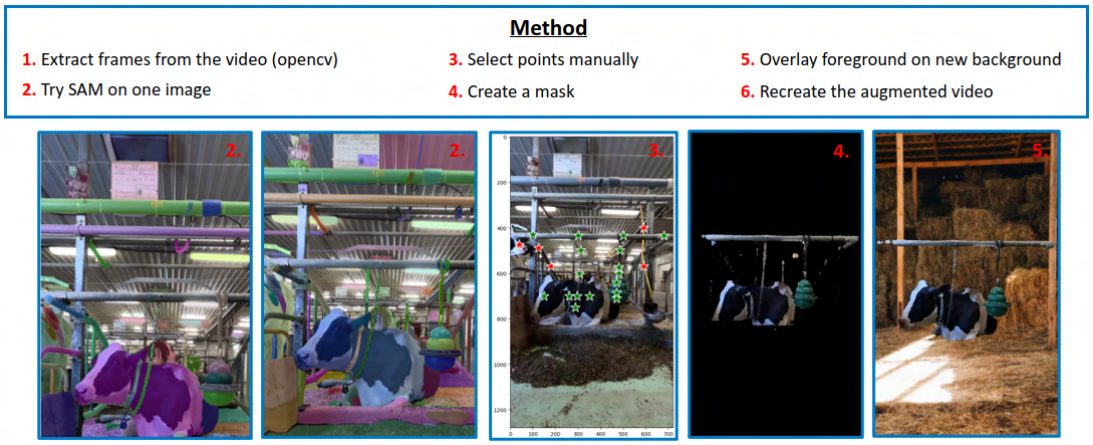


FIGURE 63 – Pipeline de segmentation de la vache, du kong, de la corde et de la chaîne

La principale limite de cette méthode est que dans notre vidéo cela fonctionne car la vache ne bouge pas beaucoup mais si elle se déplace, les points sélectionnés manuellement ne seront plus situés sur les objets choisis et on segmentera autre chose. De plus, elle n'est pas du tout appliquable à d'autres vidéos.

Cependant, le but de ce projet très court était simplement de fournir rapidement des données au département de sciences animales pour leur présentation au sujet de leur projet d'augmentation de données donc cette méthode était satisfaisante pour l'instant. D'ailleurs, chaque personne du groupe n'a fourni que de courts extraits vidéos car traiter la vidéo de 45min aurait pris trop de temps. En revanche, pour le projet à long terme, il serait préférable d'entraîner un modèle de segmentation pour qu'il segmente automatiquement le kong, la chaîne, la corde et la barre. Un tel projet nécessiterait beaucoup de monde et de temps pour annoter de nombreuses images. Pour la segmentation de la vache, DeepLabV3 serait suffisant car dans la sous-partie précédente nous avons vu que ce modèle était très efficace pour réaliser cette tâche.

5.3 Augmentation de données basée sur le machine learning

Dans un second temps, mon groupe a utilisé des méthodes basées sur le machine learning pour réaliser l'augmentation de données. L'un des membres avait pour but d'extraire les morceaux de vidéos correspondants aux intéractions entre les vaches et les kongs. Deux autres personnes devaient générer des images entre les frames pour augmenter la fréquence de frame afin d'avoir plus de détails dans les mouvements de la vache en simulant un slow motion grâces à deux modèles pré-entraînés. Enfin, mon rôle était d'utiliser un GAN (Generative Adversarial Network) pour générer de nouvelles séquences vidéos afin de simuler des comportements de vaches réalistes. L'objectif avec ce GAN est de créer des comportements rares et de simuler la variation naturelle de la posture, de la vitesse et de l'intensité des actions des vaches en augmentant la diversité comportementale sans ajouter de vraies vaches. En effet, comme expliqué précédemment, la tâche aurait été très longue si nous avions fait cela dans l'environnement réel car, tout au long de la journée, les vaches ne passent qu'un faible pourcentage de temps à interagir avec les dispositifs d'enrichissement. Pour obtenir suffisamment de données, il aurait donc encore une fois fallu beaucoup de temps et d'équipement, ce qui aurait pu perturber les vaches.

À terme, le but de cette tâche est de créer un GAN conditionnel, c'est-à-dire un GAN avec une classe par comportement et qui peut générer des vidéos d'un comportement en particulier. La méthode envisagée était tout d'abord de segmenter la vidéo selon les différents types de comportements (mordre le kong, lécher le kong, mordre la corde, lécher la corde, dormir, se mettre debout, etc.) puis de ranger le dataset par comportement. Ensuite il aurait fallu entraîner un GAN avec ce dataset et à la fin générer des vidéos pour chaque type de comportement. Ce pipeline est décrit figure 64.

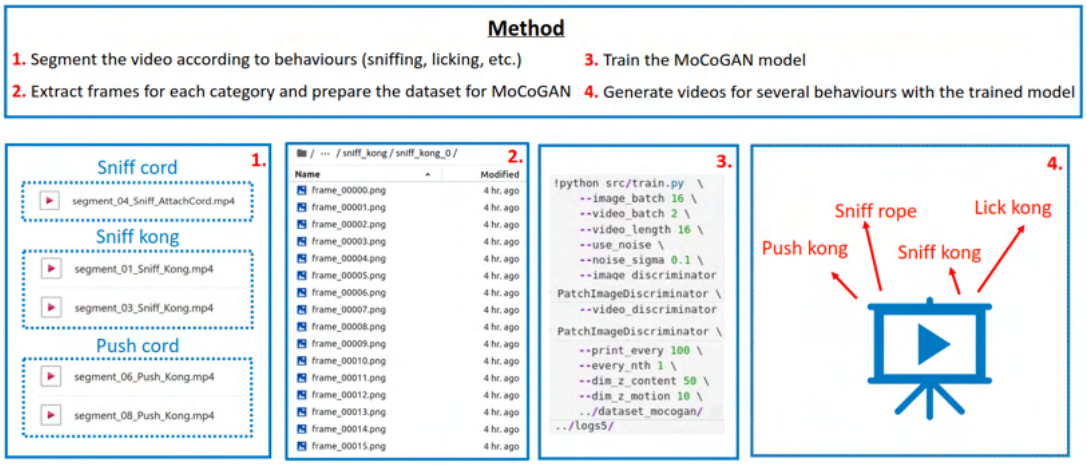


FIGURE 64 – Méthode envisagée pour l'utilisation du GAN conditionnel

Afin de réaliser cette méthode, j'ai d'abord créé un code `times.py` [26] pour trouver les séquences de temps correspondant à chaque action de la vache à partir de l'éthogramme que j'avais à disposition, dans le but ensuite de segmenter la vidéo en fonction de ces différents comportements. Un éthogramme est un tableau de données résultant de l'annotation des vidéos et qui comporte tous les comportements ou changements de comportements des vaches. Un exemple d'éthogramme est présent figure 65.

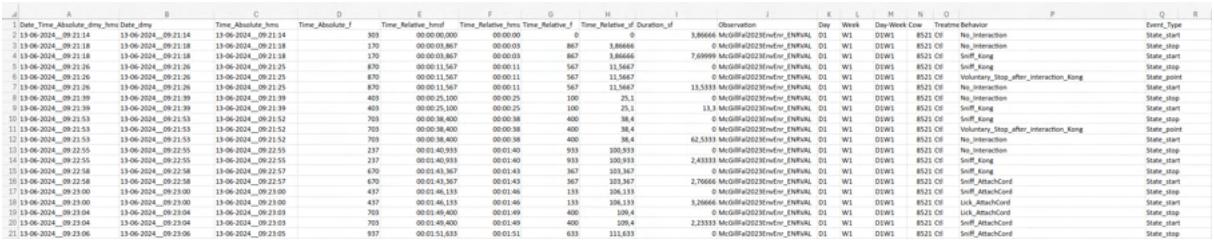


FIGURE 65 – Exemple d'éthogramme

D'après les résultats du code, je me suis rendue compte que je ne disposais que de très peu de données (deux vidéos ou moins de quelques secondes pour 3 actions et pas de vidéo pour les autres). J'ai donc abandonné l'idée d'un GAN conditionnel dans un premier temps car il m'aurait fallu beaucoup plus de vidéos pour chaque comportement. Ainsi, je n'ai pas fait de distinctions dans les actions de la vache et j'ai d'essayé de produire simplement une vidéo de vache quelconque. J'ai également pu récupérer une autre vidéo de 45min dans laquelle la vache bougeait plus que dans la vidéo précédente, ce qui était indispensable pour entraîner le GAN.
Ensuite, j'ai préparé mon dataset pour utiliser MoCoGAN (Motion and Content decomposed GAN)[46]. Ce modèle de GAN utilise des collages horizontaux composés d'images

se suivant dans la vidéo. J'ai donc créé un code `GAN.ipynb` [26] qui extrait des frames et les redimensionne pour MoCoGAN. Il crée ensuite un datataset constitué de 4 dossiers composés chacun de 50 collages de 50 images. Encore une fois, pour une question de stockage j'ai utilisé Compute Canada.

J'ai ensuite créé un dossier pour les logs et j'ai lancé l'entraînement avec la commande suivante dans le Jupyter Notebook :

```
!python src/train.py \
--image_batch 8 \
--video_batch 1 \
--video_length 16 \
--use_noise \
--noise_sigma 0.1 \
--image_discriminator PatchImageDiscriminator \
--video_discriminator PatchVideoDiscriminator \
--print_every 100 \
--every_nth 1 \
--dim_z_content 50 \
--dim_z_motion 10 \
../final_output_collages/ ../logs5/
```

Après plusieurs modifications de paramètres comme la taille des batchs et la longueur de vidéo, l'entraînement s'est lancé sans erreur mais, au bout de 2h, alors que j'étais à un peu plus de 10% de l'entraînement, mon noeud de calcul sur Compute Canada s'est rompu. J'ai donc dû favoriser la demande d'allocation et le lancement du script par sbatch. Mon dataset étant déjà constitué, j'ai écrit le fichier .sh en annexe B\ que j'ai exécuté dans Compute Canada pour lancer mon entraînement qui, cette fois-ci, s'est bien terminé. Après cela, je me suis placée dans le dossier qui contenait le modèle entraîné et j'ai importé la classe VideoGenerator depuis ce modèle. Ensuite, j'ai chargé le modèle correspondant à la dernière itération effectuée (99900) et j'ai configuré le générateur en mode évaluation pour ensuite générer une vidéo. Enfin, j'ai généré une vidéo grâce à PyTorch, en convertissant d'abord le tenseur PyTorch au format NumPy pour la compatibilité vidéo puis en mettant les frames au bon format. Malheureusement, celle-ci n'était pas satisfaisante. J'ai donc essayé de refaire le processus avec des images provenant d'autres itérations au cas où le modèle aurait trop appris mais, même si autour de 80000 nous voyons des formes apparaître sur la figure 66, dans chaque cas je n'ai obtenu que des pixels de couleurs différentes.

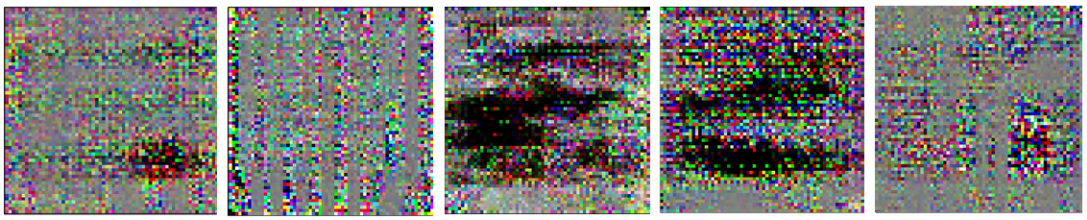


FIGURE 66 – Exemples de frames obtenues pour les itérations 30000, 50000, 70000, 85000 et 99900

Afin d'obtenir une vidéo exploitable, il faudrait jouer avec les paramètres de l'entraînement et surtout utiliser plus de vidéos de vaches actives ou faire de l'augmentation de données. Malheureusement, je ne disposais pas de plus de vidéos et avec les contraintes de temps imposées par mon stage, je n'ai pas eu le temps de terminer cette tâche.

5.4 Conclusion et perspectives

Comme je l'ai expliqué, ce projet d'augmentation de données est tout récent au sein de la chaire car il a débuté pendant mon stage. J'ai participé à la première étape du projet qui était simplement de fournir des éléments de vidéo pour montrer ce qu'on était capable d'obtenir, sans pour autant être capable de traiter des vidéos de 45min ou d'avoir des modèles robustes utilisables pour n'importe quelle vidéo. Ces objectifs interviendront dans un second temps si le projet se poursuit. Il faudra alors entraîner de vrais modèles pour qu'ils soient performants dans tous les cas de figure et remodeler nos codes pour que les codes de chacuns puissent être appelés au sein d'une même fonction (par exemple, appliquer d'abord du contraste puis améliorer la netteté et enfin changer l'arrière-plan).

Marjorie Cellier, membre du département de sciences animales de McGill a présenté ce que notre département avait fait au cours de cette première étape lors de la conférence 2025 North American Regional Meeting of the International Society of Applied Ethology (ISAE) (Page de présentation de la conférence [8]). L'ISAE comprend des chercheurs, des étudiants et des universités partenaires qui étudient le comportement animal et qui contribuent ainsi à une meilleure compréhension des interactions entre les humains et les autres animaux et aident à créer un meilleur équilibre entre le bien-être des animaux et les exigences des humains envers eux. Les membres de l'ISAE travaillent sur une grande variété d'espèces animales et de sujets comme la gestion et le bien-être du bétail, les interactions entre les humains et les animaux de compagnie, l'impact du logement sur le comportement et le bien-être des animaux de zoo et de laboratoire. L'ISAE constitue une réserve d'experts indépendants pour les gouvernements, les organismes internationaux, l'industrie et les ONG.

La présentation de Marjorie Cellier intitulée *"Enhancing Statistical Power in Animal Behavior Research : A Case Study on Enrichment in Dairy Cows Using Data Augmentation Methods"* a été plutôt bien accueillie au sein des chercheurs en sciences animales. Si quelques personnes étaient inquiètes à propos de l'utilisation de l'augmentation de données en sciences animales car elles avaient peur que cela nous pousse à nous débarasser des animaux dans des recherches les concernant, la majorité des chercheurs a trouvé que cela était courageux d'essayer de changer les choses dans un domaine ou le peu de données mène souvent à des résultats inexploitables. Si l'étude se poursuit et est concluante, ce projet d'augmentation de données pourrait permettre d'avoir enfin assez de données dans les projets de recherche en sciences animales et de valider ou d'infirmer des résultats qui n'avaient pas pu l'être jusqu'à aujourd'hui.

6 Conclusion et perspectives

6.1 Conclusion du projet

Durant mes 5 mois de stage, j'ai pu réussir à poser le bases du projet de déploiement d'un robot dans les exploitations laitières. J'ai choisi le robot qui serait le plus adapté aux besoins de la chaire et à l'environnement des exploitations laitières et j'ai réussi à proposer un simulateur et un environnement open source pratiques à utiliser. J'ai également intégré les algorithmes de détection et d'identification de vaches à mon architecture et proposé des solutions utilisables dans la simulation tout en faisant en sorte qu'il soit facile de passer de l'un à l'autre en vue de tester ces codes sur le robot dans la réalité. Enfin, j'ai pu développer deux algorithmes de navigations pour le robot choisi. Le premier permettant de suivre la vache la plus proche du robot est opérationnel et le second est déjà bien avancé mais nécessite quelques ajustements à cause des lourdes ressources qu'il utilise.

A présent, pour continuer le projet il faudrait tester ces algorithmes sur un vrai robot Spot pour pouvoir adapter les commandes si les réactions du robot ne sont pas les mêmes que dans la simulation, notamment en ce qui concerne les vitesses de déplacement ou de rotation. De plus, avant de continuer le développement des algorithmes de navigation, il faudrait travailler avec le laboratoire de sciences animales CowLife à propos de plusieurs sujets. Tout d'abord, il faudrait connaître leurs besoins exacts pour développer de nouveaux algorithmes. Par exemple on pourrait adapter le suivi de vache proche en changeant de cible toutes les X minutes. De plus, d'après le laboratoire CowLife, il pourrait être intéressant de développer un algorithme qui permet de toujours orienter Spot face à la vache car ce sont les informations de position de la tête et du museau qui leur permet de déduire certaines informations concernant le comportement des vaches. Enfin, avant de continuer il serait intéressant de mettre un vrai robot Spot au milieu des vaches pour voir comment elles se comportent avec lui car nous ne pouvons pas le deviner. Les experts en science animale de CowLife ont aussi expliqué que certaines vaches viendront forcément renifler, mordre ou lécher Spot, voire même qu'elle pourraient lui rentrer dedans. Il faudra alors adapter l'algorithme en faisant par exemple en sorte que Spot se couche si une vache se rapproche trop rapidement de lui afin d'éviter qu'il se fasse renverser et qu'il s'abîme. Il faudrait également étudier l'impact de Spot sur le bien-être mental des vaches car celui-ci ne doit pas les effrayer ou influencer leur comportement. On pourrait alors faire des expériences en camouflant Spot ou en lui donnant une odeur semblable à celle des vaches.

Pour finir, tout comme mon projet secondaire dont j'ai évoqué les perspectives dans la partie précédente, ce projet principal est très prometteur. D'après les chercheurs des laboratoires de bio-informatique à l'UQÀM et de sciences animales à McGill, un tel robot pourrait grandement faciliter leur travail de recherche pour développer des algorithmes permettant d'étuder le bien-être physique et mental des vaches ainsi que leur comportement. À terme, Spot pourrait également être déployé dans plusieurs exploitations laitières afin de permettre aux agriculteurs de mieux prendre soin de leurs vaches. La chaire prévoit d'ailleurs bientôt d'entraîner leur modèle d'identification avec des vaches provenant d'autres fermes afin que ceux-ci soit utilisables dans ces fermes avec toujours la promesse de fournir un moyen facile à mettre en place aux agriculteurs, sans qu'ils n'aient besoin de faire des adaptations dans leur ferme ou d'installer des moyens invasifs pour leurs vaches.

6.2 Conclusion du stage

Ce stage était une excellente expérience durant laquelle j'ai appris beaucoup de choses autant professionnelles que personnelles. Au début ce n'était pas évident de travailler seule sur le début d'un tel projet d'intégration d'un robot dans les exploitations laitières car le sujet était très vaste mais, au fur et à mesure des choix que j'ai été amenée à faire, j'ai pu resserrer le sujet. C'était également la première fois que j'ai dû travailler de manière aussi autonome car personne n'avait les réponses à mes questions lorsque j'étais bloquée étant donné qu'un tel projet n'avait jamais été fait. Après 5 mois, c'est donc très satisfaisant de voir les bases que j'ai construites pour ce projet qui sera continué et apportera beaucoup aux agriculteurs et au bien-être des vaches laitières.

Un autre point que j'ai beaucoup apprécié est le fait d'échanger avec le laboratoire de sciences animales. Cela m'a montré que mes compétences informatiques pouvaient être associées à d'autres connaissances comme ici les connaissances des comportements des vaches afin de créer des technologies performantes. Je trouve cela très intéressant car grâce au laboratoire de bio-informatique, le laboratoire de sciences animales peut obtenir des résultats plus performants et plus rapides et, grâce au laboratoire de sciences animales, nous pouvons exploiter les résultats que nous obtenons grâce à ce que nous développons. De plus, j'ai également beaucoup aimé travailler en groupe lors de mon projet secondaire d'augmentation de données. C'était très instructif de voir qu'à partir d'une demande du laboratoire de sciences animales, le laboratoire de bio-informatique a su se séparer en trois groupes pour répartir les tâches tout en gardant un système de communication efficace. J'ai également pu participer à des activités de cohésion avec le laboratoire ainsi qu'à des lectures organisées par les étudiants autour des Large Language Models, ce qui m'a permis à la fois d'appprendre de nouvelles choses et d'être parfaitement intégrée à la chaire.

Enfin, j'ai trouvé que les tâches effectuées durant mon stage étaient vraiment nombreuses et variées, si bien que j'ai l'impression d'avoir pu réutiliser presque tous les cours que j'ai eu durant mes deux années de robotique à l'ENSTA comme par exemple ROS, simulation, machine learning, linux embarqué ou encore asservissement visuel. Ce stage était donc comme une synthèse de ces deux dernières années et m'a vraiment permis de me rendre compte de toutes les compétences que j'avais acquises à l'ENSTA et que je pourrais réutiliser dans mon futur professionnel.

Références

[1] Modèle de vache. https://free3d.com/fr/3d-models/vache.

[2] Vijay Anand. magbot. https://github.com/VijayAnand2k20/magbot/tree/vj, 2025.

[3] Voncarlos M. Araújo, Ines Rili, Thomas Gisiger, Sébastien Gambs, Elsa Vasseur, Marjorie Cellier, and Abdoulaye Baniré Diallo. Ai-powered cow detection in complex farm environments. *Smart Agricultural Technology*, 10 :100770, 2025.

[4] Fetullah Atas. quadruped_ros2 : Adaption of champ quadruped robot to ros2. https://github.com/jediofgever/quadruped_ros2, 2022. Accessed : 2025-05-21.

[5] Boston Dynamics. Spot - the agile mobile robot that climbs stairs and traverses rough terrain, 2024.

[6] Marco Bovo, Miki Agrusti, Stefano Benni, Daniele Torreggiani, and Patrizia Tassinari. Random forest modelling of milk yield of dairy cows under heat stress conditions. *Animals*, 11(5), 2021.

[7] Le Bulletin. Les robots de traite ont doublé en cinq ans, 2025. Accessed : 2025-03-27.

[8] Campbell Centre for the Study of Animal Welfare. 2025 isae north american regional meeting. Réunion régionale de l'International Society of Applied Ethology, May 2025. Arboretum Centre, University of Guelph ; dates : 14–15 mai 2025.

[9] Jack Collins, Shelvin Chand, Anthony Vanderkop, and David Howard. A review of physics simulators for robotic applications. *IEEE Access*, 9 :51416–51431, 2021.

[10] Albert De Vries, Nikolay Bliznyuk, and Pablo Pinedo. Invited review : Examples and opportunities for artificial intelligence (ai) in dairy farms*. *Applied Animal Science*, 39(1) :14–22, 2023.

[11] DEEPRobotics. Deeproboticslynx– all-terrain wheeled-legged robot, 2025.

[12] DEEPRobotics. Product series 3 – deeprobotics (x20, x30, lite3, lynx, j-series, dr01), 2025.

[13] DEEPRobotics. Products – deep robotics (x20, x30, lite3, lynx, j-series, dr01), 2025.

[14] Andrew Farley, Jie Wang, and Joshua A. Marshall. How to pick a mobile robot simulator : A quantitative comparison of coppeliasim, gazebo, morse and webots with a focus on accuracy of motion. *Simulation Modelling Practice and Theory*, 120 :102629, 2022.

[15] Jorge Ferreira, A. Paulo Moreira, Manuel Silva, and Filipe Santos. A survey on localization, mapping, and trajectory planning for quadruped robots in vineyards. In *2022 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, pages 237–242, 2022.

[16] Sigfredo Fuentes, Claudia Gonzalez Viejo, Brendan Cullen, Eden Tongson, Surinder S. Chauhan, and Frank R. Dunshea. Artificial intelligence applied to a robotic dairy farm to model milk productivity and quality based on cow data and daily environmental parameters. *Sensors*, 20(10), 2020.

[17] Sigfredo Fuentes, Claudia Gonzalez Viejo, Eden Tongson, Nir Lipovetzky, and Frank R. Dunshea. Biometric physiological responses from dairy cows measured by visible remote sensing are good predictors of milk productivity and quality through artificial intelligence. *Sensors*, 21(20), 2021.

[18] Berry Gerrits, Martijn Mes, Peter Schuur, and Robert Andringa. A simulation model for cooperative robotics in dairy farms. In *2022 Winter Simulation Conference (WSC)*, pages 831–842, 2022.

[19] Hansen Hendra, Yubin Liu, Ryoichi Ishikawa, Takeshi Oishi, and Yoshihiro Sato. Quadruped robot platform for selective pesticide spraying. In *2023 18th International Conference on Machine Vision and Applications (MVA)*, pages 1–6, 2023.

[20] Lewis Holloway, Christopher Bear, and Katy Wilkinson. Re-capturing bovine life : Robot–cow relationships, freedom and control in dairy farming. *Journal of Rural Studies*, 33 :131–140, 2014.

[21] Mobile Autonomous Systems Cognitive Robotics Institute. Webots ros2 spot. https://github.com/MASKOR/webots_ros2_spot, 2023.

[22] Serena Ivaldi, Vincent Padois, and Francesco Nori. Tools for dynamics simulation of robots : a survey based on user feedback. 02 2014.

[23] Boyu Ji, Thomas Banhazi, Clive J.C. Phillips, Chaoyuan Wang, and Baoming Li. A machine learning framework to predict the next month's daily milk yield, milk composition and milking frequency for cows in a robotic dairy farm. *Biosystems Engineering*, 216 :186–197, 2022.

[24] Juan Miguel Jimeno. Champ : An open-source development framework for quadrupedal robots. https://github.com/chvmp/champ, 2019.

[25] Laura Jouvet. Quad robot simulator. https://github.com/bioinfoUQAM/quad-robot-simulator, 2025.

[26] Laura Jouvet. UQAM Data Augmentation Project. Dépôt GitLab, ENSTA Bretagne https://gitlab.ensta-bretagne.fr/jouvetla/uqam_data_augmentation_project.git, 2025.

[27] Laura Jouvet. Vache proche - démonstration finale. https://youtu.be/ynQjb4JUOTw, 2025.

[28] Laura Jouvet. Vache specifique - perte de la cible. https://www.youtube.com/watch?v=qafUxz77IdM, 2025.

[29] Beatriz Jové, Alexis Gutiérrez, Camino Fernández, Lidia Sánchez, Francisco J. Rodríguez-Lera, and Vicente Matellán. Quadrupeds robots in herding : Metrics for experimental validation of animal-robot interactions. Grupo de Robótica, Universidad de León, 2023.

[30] L'Actualité. Des robots en agriculture, 2025. Accessed : 2025-03-27.

[31] Laura Jouvet. Stage-UQAM — dépôt GitLab CoppeliaSim. GitLab repository, ENSTA Bretagne (private).

[32] Jongwoo Lee. *Hierarchical Controller for Highly Dynamic Locomotion Utilizing Pattern Modulation and Impedance Control : Implementation on the MIT Cheetah Robot.* Ph.d. thesis, Massachusetts Institute of Technology, Cambridge, MA, 2013.

[33] T. Martin, P. Gasselin, N. Hostiou, and et al. Robots and transformations of work in farm : a systematic review of the literature and a research agenda. *Agronomy for Sustainable Development*, 42(66), 2022.

[34] Nur Supadmana Muda. Design and develop autonomous 3 in 1 agricultural robots for farming, 2024. Thesis, October 2024.

[35] Suresh Neethirajan. Net zero dairy farming—advancing climate goals with big data and artificial intelligence. *Climate*, 12(2), 2024.

[36] Lucas Nogueira. Comparative analysis between gazebo and v-rep robotic simulators. *School of Electrical and Computer Engineering, Universidade de Campinas*, 2023. Email : lucas.afonso@hotmail.com.

[37] M. Pastell, H. Takko, H. Gröhn, M. Hautala, V. Poikalainen, J. Praks, I. Veermäe, M. Kujala, and J. Ahokas. Assessing cows' welfare : weighing the cow in a milking robot. *Biosystems Engineering*, 93(1) :81–87, 2006.

[38] Dmitriy Yu. Pavkin, Denis V. Shilin, Evgeniy A. Nikitin, and Ivan A. Kiryushin. Designing and simulating the control process of a feed pusher robot used on a dairy farm. *Applied Sciences*, 11(22), 2021.

[39] Christopher Quail, Evrard Emonot–de Carolis, and Fernando Auat Cheein. Legged robots in the agricultural context : Analysing their traverse capabilities and performance. In *IECON 2023- 49th Annual Conference of the IEEE Industrial Electronics Society*, pages 01–07, 2023.

[40] robot mania. Quadruped robot basics, 2021.

[41] Robotics Knowledgebase. Choose a simulator, 2025. Consulté le 7 avril 2025.

[42] Francisco J. Rodríguez-Lera, Miguel A. González-Santamarta, Jose Manuel Gonzalo Orden, Camino Fernández-Llamas, Vicente Matellán-Olivera, and Lidia Sánchez-González. Lessons learned in quadruped deployment in livestock farming. 2023.

[43] Bryan C. Russell, Antonio Torralba, Kevin P. Murphy, and William T. Freeman. Labelme : A database and web-based tool for image annotation. *International Journal of Computer Vision*, 77(1–3) :157–173, 2008.

[44] Fuyang Tian, Xinwei Wang, Sufang Yu, Ruixue Wang, Zhanhua Song, Yinfa Yan, Fade Li, Zhonghua Wang, and Zhenwei Yu. Research on navigation path extraction and obstacle avoidance strategy for pusher robot in dairy farm. *Agriculture*, 12(7), 2022.

[45] Pranshu Tople. Ca how to try $75000 robot dog without actually buying | @bostondynamics spot mini simulation package. https://www.youtube.com/watch?v=C9quuhNuWIM, 2021.

[46] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. MoCoGAN : Decomposing motion and content for video generation. https://github.com/sergeytulyakov/mocogan, 2018.

[47] Unitree Robotics. Go2 – intelligent quadruped robot with 4d lidar and ai modes, 2025.

[48] Unitree Robotics. Unitreeb2 – all-terrain industrial-grade quadruped robot with wheeled-leg option, 2025.

[49] Xinyue Wang, Jinkun Liu, Ting Zhang, and Jie Zhang. Applied research of agricultural quadruped robots. In *2024 IEEE 13th International Conference on Communication Systems and Network Technologies (CSNT)*, pages 954–957, 2024.

[50] WLKATA Robotics. Marchxpro lidar kit – quadruped robotic dog with lidar, depth camera, sdk and ai functions, 2025.

[51] Donger Yang, Di Cui, and Yibin Ying. Development and trends of chicken farming robots in chicken farming tasks : A review. *Computers and Electronics in Agriculture*, 221 :108916, 2024.

[52] Xiao Yang, Jinchang Zhang, Bidur Paneru, Jiakai Lin, Ramesh Bahadur Bist, Guoyu Lu, and Lilong Chai. Precision monitoring of dead chickens and floor eggs with a robotic machine vision method. *AgriEngineering*, 7(2), 2025.

[53] Darío Fernando Yépez-Ponce, José Vicente Salcedo, Paúl D. Rosero-Montalvo, and Javier Sanchis. Mobile robotics in smart farming : current trends and applications. *Frontiers in Artificial Intelligence*, 6, 2023.

[54] Jie Zhang, Xinyue Wang, and Liang Zheng. Research on autonomous navigation system of agricultural quadruped robot. In *2024 IEEE International Conference on Mechatronics and Automation (ICMA)*, pages 1286–1290, 2024.

Annexe A/ Programme du stage

Mois	Tâche	Description
Avril	Tâche 1-A	Revue de littérature
Avril	Tâche 1-B	Recherche et documentation d'un outil utilisable pour la simulation de l'environnement
Avril	Jalon 1	Résumé et raisonnement pour le choix de l'outil et le déploiement du simulateur d'environnement
Mai	Tâche 2-A	Évaluer et identifier l'infrastructure disponible la plus faisable pour développer le simulateur
Mai	Tâche 2-B	Déployer l'environnement et choisir le simulateur
Mai	Tâche 2-C	Créer le premier modèle de grange (peuplé) dans l'environnement
Mai	Jalon 2	Déploiement du premier modèle de grange avec des acteurs et des objets (vaches, robot)
Juin	Tâche 3-A	TBD
Juin	Jalon 3	Déploiement d'opérations simples (arrêter, continuer, sortir, etc)
Juillet	Tâche 4-A	TBD
Juillet	Jalon 4	Peupler le modèle avec de nouvelles propriétés (TBD)
Juillet	Rapport	Rédaction du rapport de stage
Août	Tâche 5-A	Comprendre les applications des outils développés en interne
Août	Tâche 5-B	Étudier les moyens potentiels d'intégration de la boîte à outils dans le simulateur
Août	Jalon 5	Déploiement et test des outils développés en interne dans l'environnement virtuel robotique
Août	Rapport	Rédaction du rapport de stage

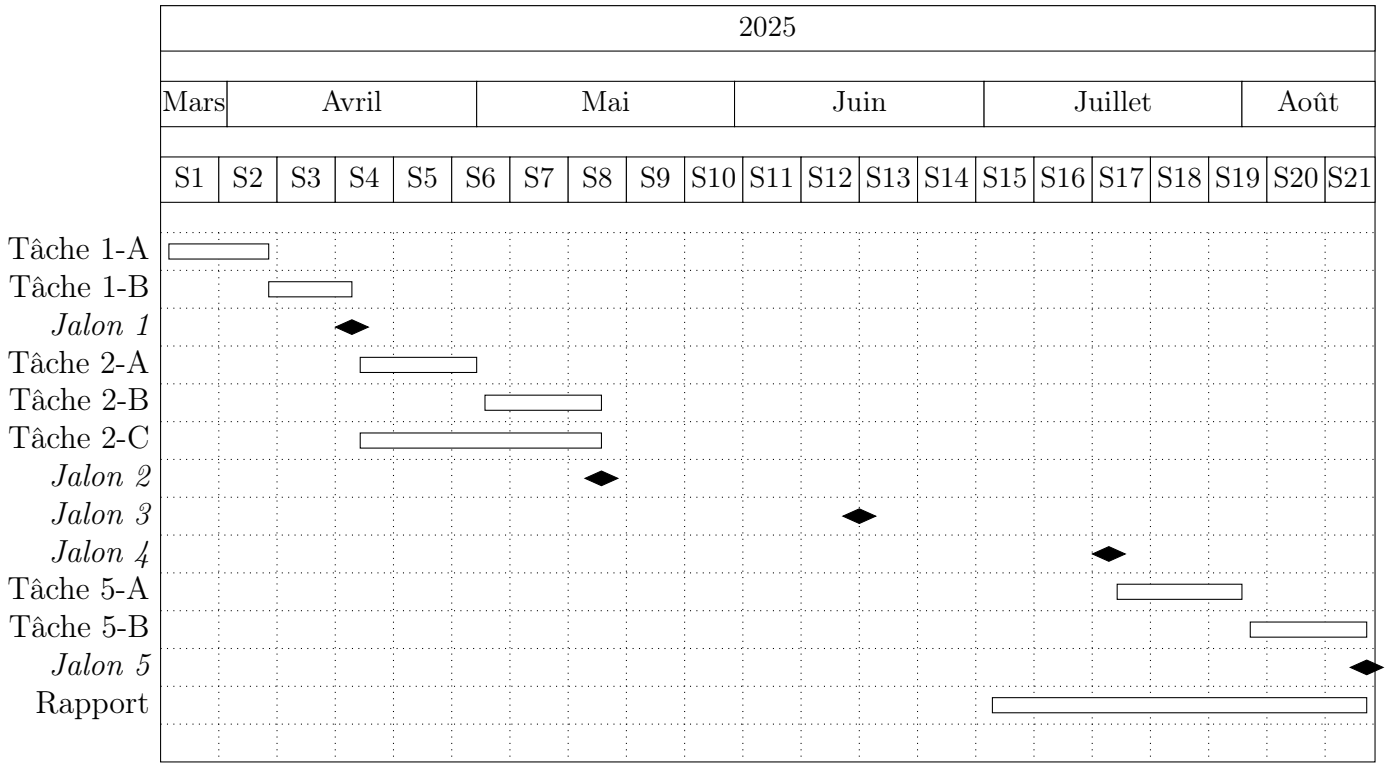


FIGURE 67 – Diagramme de Gantt du projet principal

Annexe B/ Fichier `run_training.sh` pour lancer l'entraînement de MoCoGAN

```
#!/bin/bash

#SBATCH --account=def-banire # Nom du groupe sur Compute Canada
#SBATCH --gpus-per-node=1 # Nombre de GPU
#SBATCH --cpus-per-task=4 # Nombre de CPU
#SBATCH --mem=16G # RAM necessaire
#SBATCH --time=22:00:00 # Duree (max 24h)
#SBATCH --mail-user=ton.email@UQM.ca # Mail a indiquer si besoin
#SBATCH --mail-type=ALL
#SBATCH --output=train-%j.out # Log de sortie

# === ENVIRONNEMENT ===

module load python
module load gcc
module load opencv
module load StdEnv/2020
module load ffmpeg
module load gcc opencv

source ~/jupyter1/bin/activate # Activation de l'environnement virtuel

# === LANCEMENT DU SCRIPT PYTHON ===

python src/train.py \
    --image_batch 8 \
    --video_batch 1 \
    --video_length 16 \
    --use_noise \
    --noise_sigma 0.1 \
    --image_discriminator PatchImageDiscriminator \
    --video_discriminator PatchVideoDiscriminator \
    --print_every 100 \
    --every_nth 1 \
    --dim_z_content 50 \
    --dim_z_motion 10 \
    ../dataset_mocogan/ ../logs5/
```

Annexe C
Etat de l'art des robots agricoles

Laura Jouvet

McGill
UQÀM
Université du Québec à Montréal
well e
AI Serving Welfare
ENSTA
POLYTECHNIQUE INSTITUT DE PARIS
IP PARIS

Table des matières

1 Introduction

L'agriculture est un secteur fondamental depuis les débuts de la civilisation humaine et, aujourd'hui, elle est en pleine révolution technologique. L'intégration des robots et des technologies avancées dans l'agriculture vise à augmenter l'efficacité, la productivité et à réduire la dépendance à la main-d'œuvre humaine. Les robots agricoles se diversifient et apportent des solutions à diverses tâches, allant de la gestion des cultures à l'élevage des animaux.
Cet état de l'art présente les principales innovations récentes dans le domaine des robots agricoles, notamment les robots de traite, les tracteurs autonomes, les drones et l'usage de l'intelligence artificielle (IA).

2 Robots de traite automatisée

Les systèmes de traite robotisée représentent l'une des applications les plus anciennes et les plus étudiées dans le domaine de l'automatisation agricole. Martin et al. ont en effet étudié 90 publications parlant de robotique agricole et, entre 1992 et 2022, 60 articles mentionnent des systèmes de traite automatisée.[4]

En Europe, l'adoption de ces robots a débuté il y a 30 ans et leur déploiement s'est accéléré dans les années 2010.[4]. Au Québec aussi l'utilisation de ces robots explose. En 2021, 19% des fermes laitières déclaraient posséder une ou des trayeuses entièrement automatisées, ce qui concernait 840 fermes. La vitesse d'adoption de ces système est le même dans le reste du Canada pour une moyenne de 20%. Il est le plus populaire dans l'Ouest où il est présent dans 27% des fermes. Le taux varie selon la taille des fermes et les revenus : plus les fermes sont grosses, plus elles ont tendance à être dotées d'un robot de traite.[1]

Le coût élevé de ces robots et leur rentabilité dépendent de facteurs comme la gestion du troupeau, les conditions climatiques et les paramètres biologiques des animaux. Cependant, ils possèdent de nombreux avantages. Ils permettent par exemple aux agriculteurs de réduire le temps passé à traire leurs vaches, de réduire leurs coûts de main-d'œuvre et d'avoir une meilleure qualité de lait. Certains robot de traite automatisée permettent même de s'intéresser au bien-être animal. C'est ce qu'ont fait Ji et al. en utilisant le machine learning pour améliorer la précision des prédictions par rapport aux méthodes statistiques traditionnelles. Ils ont utilisé des systèmes de traite robotisée pour collecter une importante quantité de données sur la production laitière et la santé des animaux. Grâce à celles-ci, ils ont ensuite développé des modèles prédictifs afin de détecter des problèmes comme le stress thermique, les maladies ou les variations de la production de lait. Par exemple, grâce à un cadre basé sur l'apprentissage automatique utilisant des modèles comme XGBOOST, ils ont pu prévoir la production laitière, la composition du lait et la fréquence de traite de chaque vache sur une période de 28 jours, offrant ainsi une gestion plus individualisée des animaux. Un écart entre les prédictions et la réalité peut signifier qu'il y a un problème de santé mental ou psychologique de la vache.[2]

Dans l'études de Martin et al. dont les résultats sont présentés sur la figure ci-dessous, nous pouvons voir qu'à partir des années 2010, d'autres types de robots sont apparus dans le milieu agricole tels que des robots pour nourrir les animaux ou bien des

robots pour s'occuper des cultures. Nous allons donc à présent évoquer certains de ces robots existants.

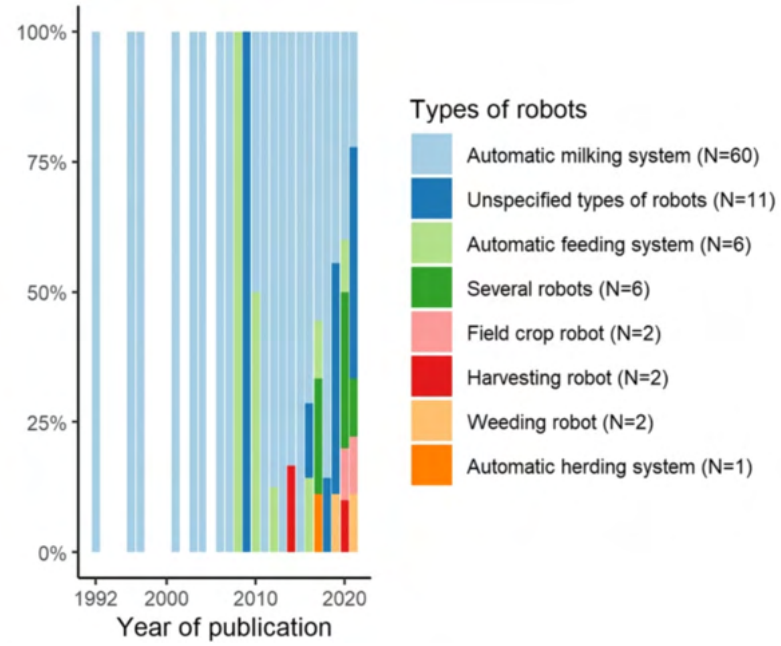


FIGURE 1 – Nombre de publications par sujets entre 1992 et 2022 sur 90 publications [4]

3 Tracteurs autonomes et véhicules à guidage autonome

Les tracteurs autonomes sont l'une des innovations importantes dans l'agriculture. Ces véhicules à guidage autonome par GPS, équipés de capteurs et d'actionneurs, sont utilisés pour la surveillance des cultures, l'irrigation, la récolte et la gestion des maladies. Des prototypes de tracteurs autonomes de labourage de rizières équipés d'intelligence artificielle ont montré qu'ils pouvaient réduire la dépendance à la main-d'œuvre et accélérer le processus de travail augmentant ainsi l'efficacité de manière significative (jusqu'à 40% d'accélération du processus de labourage).[5]

Muda et al. ont également conçu le prototype d'un robot agricole 3 en 1 visant à combiner 3 fonctions : arroser les sols, appliquer des pesticides et décompacter le sol. Pour faire cela, ils ont utilisé plusieurs composants tels qu'un microcontrôleur Arduino, un GPS, des capteurs ainsi qu'un système d'arrosage et une fourche pour le labourage. La hauteur du robot peut être ajustée en fonction de la taille des plantes pour permettre son passage. Ils ont tout d'abord réussi à contrôler manuellement le robot en transmettant des commandes de mouvement via une télécommande. Le microcontrôleur Arduino permettait de transmettre les commandes aux moteurs. La portée de communication était de 500 mètres, ce qui montre que le robot est capable de couvrir de vastes zones agricoles d'environ 25 hectares et qui le rend particulièrement adapté à l'utilisation dans des champs étendus. Enfin, Muda et al. ont intégré une navigation autonome basée sur un suivi de waypoints GPS. Les tests ont montré que le robot pouvait naviguer avec succès dans un champ de 1 hectare avec un taux de réussite de 80% lors de 10 expériences. Cependant, certaines erreurs ont été identifiées. Celles-ci étaient principalement liées à la

précision du GPS et à la stabilité du logiciel, ce qui a provoqué des dysfonctionnements occasionnels. Ces résultats suggèrent que bien que le robot démontre un fort potentiel pour les opérations agricoles autonomes, des améliorations sont nécessaires au niveau du logiciel et de la calibration du GPS pour optimiser ses performances.[5]

Néanmoins, certains véhicules autonomes restent très performants. C'est le cas de La Chèvre de Nexus Robotics. Ce robot possède trois bras qui arrachent les mauvaises herbes qu'ils différencient des pousses d'oignons encore jeunes grâce à des caméras et à leur intelligence artificielle qui a été entraînée à les reconnaître. La Chèvre est complètement autonome et peut brouter jour et nuit. De plus, d'autres véhicules autonomes possèdent même de meilleures capacités que les véhicules plus traditionnels. Par exemple, le Laser-Weeder de l'entreprise américaine Carbon Robotics n'arrache pas les mauvaises herbes. Au lieu de faire cela il les pulvérise au laser, ce qui permet d'avoir une plus grande précision. Cependant, ce robot reste encore trop cher pour pouvoir être acheté, même par les gros producteurs.[3]

FIGURE 2 – Robot LaserWeeder

4 Robots agricoles multifonctions

Les robots multifonctions représentent un axe d'innovation intéressant dans l'automatisation agricole. Comme nous l'avons vu dans la partie précédente, le prototype de robot agricole 3-en-1 de Muda et al. conçu pour l'arrosage, l'application de pesticides et l'aération du sol, montre comment la robotique peut combiner plusieurs tâches en un seul appareil. Bien que des améliorations soient encore nécessaires pour affiner la précision de la navigation et la stabilité du logiciel, le fait de combiner trois robots en un seul est très avantageux en terme de rendement temporel et de stockage matériel. En effet, bien que ce robot soit assez cher, le fait de stocker un seul robot au lieu de trois permet de réduire les coûts de stockage.[5]

D'autre part, Elmec, un fabricant de bornes de recharge résidentielles pour véhicules électriques, travaille actuellement sur une nouvelle génération de robots très performants. A terme, le but est de déployer sur de vastes superficies un ensemble de robots, tous inter-

connectés avec leurs outils pour sarcler, arroser, récolter, etc. D'après l'ingénieur Samuel Pittet qui a créé le robot Erion donc nous parlons dans la partie suivante, ces robots pourront se recharger sur une remorque et travailler 24 heures sur 24. Contrairement au cas précédent du robot 3 en 1, cela nécessiterait d'avoir plusieurs robots mais le fait qu'ils soient tous interconnectés réduirait de la même façon le temps d'utilisation ainsi que la main d'oeuvre.

5 Drones agricoles

Les drones représentent également une technologie de plus en plus utilisée pour la surveillance des cultures, l'inspection des infrastructures agricoles et la gestion de la chaîne d'approvisionnement alimentaire. Ils permettent de collecter des informations en temps réel sur la santé des cultures, la qualité du sol et d'autres paramètres cruciaux pour la gestion agricole. Ils sont notamment utilisés pour la détection précoce de maladies et la cartographie des champs, permettant aux agriculteurs de prendre des mesures rapides pour éviter la propagation des infections. Il existe même des logiciels pour la prise de décision qui analysent les données acquises par les drones et/ou certains capteurs. Ces logiciels fournissent généralement des informations sur la météo, le sol, les rendements des cultures et d'autres facteurs pertinents pour la production agricole afin d'améliorer la prise de décision.[7]

Les drones permettent également à certains robot de s'orienter. C'est le cas du robot Erion, un robot-tracteur québécois alimenté par une batterie récupérée d'une Tesla accidentée et qui permet de sarcler le sol. Ce robot s'oriente à l'aide de la caméra installée sur son capot et grâce à des images prises par un drone et qui sont décodées par son intelligence artificielle.[3]

FIGURE 3 – Robot québécois Erion

6 Intelligence artificielle et agriculture de précision

Nous avons vu précédemment que l'intelligence artificielle (IA) pouvait avoir des effets positifs dans de nombreuses applications agricoles telles que le bien-être animal avec les prédictions laitières, l'augmentation des performances de certains véhicules autonomes ou encore l'analyse de certaines données collectées par des drônes. L'IA joue donc un rôle croissant dans l'agriculture de précision.

Suharyanto et al. ont par exemple développé une application utilisant le deep learning pour détecter des maladies des plantes, comme celles du piment, en utilisant des images analysées par des réseaux neuronaux convolutifs (CNN). Ce système permet une détection rapide et précise des maladies, aidant les agriculteurs à intervenir avant que les infections ne se propagent. De même, l'optimisation de l'utilisation des engrais grâce à l'IA a été explorée par Wahyuni et al. qui ont démontré que l'utilisation de capteurs de sol et d'algorithmes d'apprentissage automatique permet de réduire l'utilisation d'engrais de 25 % tout en augmentant les rendements des cultures de riz de 15 %. Cette technologie permet une gestion plus durable des ressources et une amélioration de la productivité agricole.[5]

Enfin, en Montérégie, un producteur de maïs et de soja exploite des données depuis 2012 pour savoir quels produits utiliser et en quelle quantité en fonction des rendements. Les données sont de plus en plus pointues et il en collecte autant que possible dans l'espoir qu'un jour, des algorithmes lui indiqueront quoi faire au champ ou quelle variété planter. Sur une tablette électronique, il possède déjà des cartes de rendements obtenues grâce à sa moissoneurse-batteuse. Sur ces cartes, on peut voir la quantité de grains récoltée par seconde à chaque endroit de sa parcelle. Les données sont très précises car un algorithme corrige même le poids du grain en fonction du taux d'humidité pour calculer le nombre de kilos récoltés par hectare. Un autre algorithme compare également les résultats des récoltes avec les conditions météo, des images satellites et les registres d'épandage et d'arrosage.[3]

FIGURE 4 – Carte de rendement de récoltes de grains

En plus de réduire les temps des travaux agricoles ainsi que la main d'oeuvre, l'IA ainsi que l'automatisation et la robotisation des engins agricoles permet donc aux agriculteurs de prendre des décisions et d'améliorer leurs rendements.

7 Robotique dans l'élevage avicole

Ces dernières années, de nombreux progrès ont été faits en terme de robotisation dans l'élevage avicole et plus précisément dans l'élevage du poulet. En effet, l'élevage de poulets connaît une expansion pour répondre à la demande croissante du marché, mais fait face à des défis tels que l'augmentation de la charge de travail et la pénurie de main-d'œuvre. C'est donc pour remédier à cela que l'industrie évolue vers l'automatisation, soutenue par le développement des robots mobiles. Ces technologies permettent de standardiser la production, d'améliorer le bien-être animal et d'optimiser l'efficacité.

Les robots d'élevage de poulets sont aujourd'hui capables d'accomplir des tâches agricoles spécifiques. Ils peuvent par exemple effectuer des missions telles que l'inspection, la collecte des poulets morts, le nettoyage des poulaillers et la collecte des œufs au sol. Bien que de tels robots spécialisés, comme "ChickenBoy" pour l'inspection des bâtiments ou "XO" pour la désinfection, aient été développés pour répondre aux besoins spécifiques de l'élevage avicole, les défis techniques restent nombreux, notamment l'adaptation des robots aux environnements complexes des fermes avicoles, la gestion des interactions avec les animaux et la navigation autonome.

FIGURE 5 – Robot de désinfection de poulailler "XO"

Des recherches sont en cours pour améliorer l'autonomie des robots et leur capacité à travailler dans des environnements en constante évolution, tels que les systèmes d'élevage en volière sans cage. Cependant, pour atteindre un élevage entièrement automatisé et intelligent, des recherches et des développements supplémentaires sont nécessaires.[6]

8 Perspectives d'avenir et défis

L'intégration des robots dans les fermes transforme profondément le secteur agricole en offrant des solutions efficaces pour de nombreux défis, tels que la réduction des coûts de main-d'œuvre, l'optimisation des rendements et la gestion plus précise des ressources. De plus, la transition vers des systèmes multi-robots collaboratifs, où plusieurs robots peuvent travailler ensemble pour optimiser les tâches agricoles, représente une avancée majeure.

Cependant, plusieurs défis subsistent, notamment la compatibilité des technologies robotiques avec des environnements agricoles spécifiques, comme les fermes avicoles, et l'amélioration des interactions entre robots et animaux pour minimiser le stress animal. En effet, lors du développement de robots agricoles, l'accent est souvent mis sur les performances dans un but de gagner du temps et de réduire la main d'oeuvre mais le bien-être des animaux est très peu pris en compte. Par exemple, dans le cas des robots dans les poulaillers ou des robots de traite, la question du stress des poulets et des vaches n'a pas été étudiée. Les recherches sur les interactions entre les robots et les animaux restent superficielles, et peu de robots sont conçus pour minimiser le stress des animaux. De plus, avec l'évolution des pratiques avicoles vers des systèmes comme l'élevage en volière sans cage, de nouveaux défis apparaissent. Or, les robots actuels ne sont pas encore adaptés à ces nouveaux modes d'élevage et aux tâches spécifiques qu'ils impliquent, comme la gestion des poulets malades ou la récolte des volailles. De même, au Canada en 2027, il ne sera plus permis d'enfermer les vaches laitières durant tout leur cycle de production. Dans les fermes, les animaux seront donc de moins en moins enfermés car la question de leur bien-être devient de plus en plus importante. Que ce soit pour les poulets, pour les vaches ou pour d'autre animaux, il faudra donc développer des algorithmes optimisés pour la localisation et la planification des trajectoires dans les bâtiments et à l'extérieur de ceux-ci.

Des études devront également être faites pour réduire le stress animal qui pourrait être généré à cause des robots. Il faudrait par exemple étudier l'impact du design des robots en modifiant leur taille, leur vitesse ou encore leur couleur. De plus, si d'un côté l'utilisation de robot permet par exemple aux vaches d'avoir plus de liberté en allant à la traite quand elles le veulent et d'être moins stressées par les humains, de l'autre, cette automatisation change également la perception des éleveurs envers leurs animaux. Ceux-ci viennent alors à les percevoir comme un produit ou une combinaison de paramètres à surveiller, ce qui a des répercussions sur leur bien-être. Enfin, il serait intéressant d'intégrer de nouvelles fonctions à ces robots comme la capacité de gérer les soins des animaux, de détecter des maladies ou d'évaluer leur état de santé physique et mentale.

Les innovations actuelles, combinées à l'IA, à la robotique mobile et à l'automatisation des tâches, ouvrent donc la voie à une agriculture plus intelligente et durable mais il existe encore des progrès à faire en terme de bien-être animal afin de favoriser la meilleure des cohabitations possibles.

Références

[1] Le Bulletin. Les robots de traite ont doublé en cinq ans, 2025. Accessed : 2025-03-27.

[2] Boyu Ji, Thomas Banhazi, Clive J.C. Phillips, Chaoyuan Wang, and Baoming Li. A machine learning framework to predict the next month's daily milk yield, milk composition and milking frequency for cows in a robotic dairy farm. *Biosystems Engineering*, 216 :186–197, 2022.

[3] L'Actualité. Des robots en agriculture, 2025. Accessed : 2025-03-27.

[4] T. Martin, P. Gasselin, N. Hostiou, and et al. Robots and transformations of work in farm : a systematic review of the literature and a research agenda. *Agronomy for Sustainable Development*, 42(66), 2022.

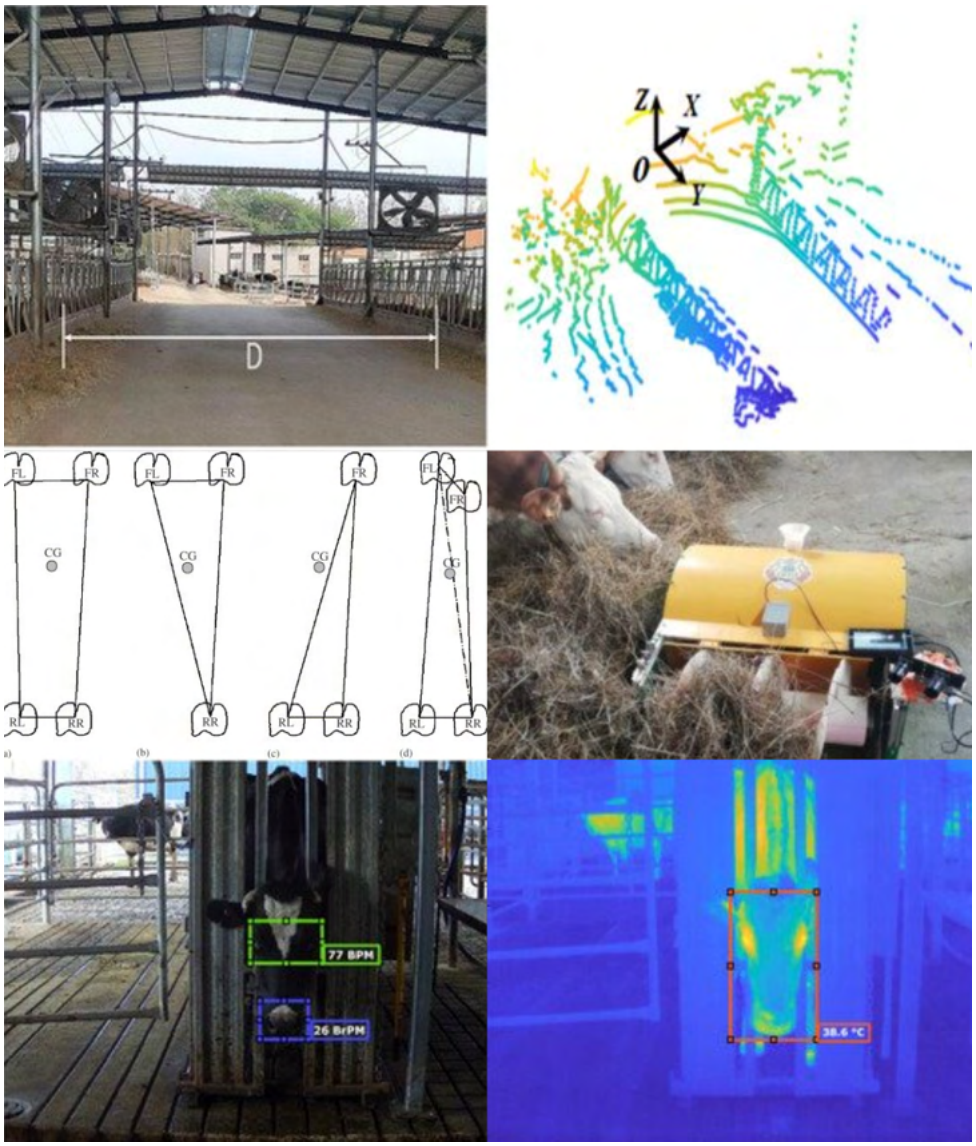
[5] Nur Supadmana Muda. Design and develop autonomous 3 in 1 agricultural robots for farming, 2024. Thesis, October 2024.

[6] Donger Yang, Di Cui, and Yibin Ying. Development and trends of chicken farming robots in chicken farming tasks : A review. *Computers and Electronics in Agriculture*, 221 :108916, 2024.

[7] Darío Fernando Yépez-Ponce, José Vicente Salcedo, Paúl D. Rosero-Montalvo, and Javier Sanchis. Mobile robotics in smart farming : current trends and applications. *Frontiers in Artificial Intelligence*, 6, 2023.

Annexe D
Etat de l'art des robots dans les fermes laitières

Laura Jouvet



McGill
UQÀM
Université du Québec à Montréal
well-e
AI Serving Welfare
ENSTA
IP PARIS

Table des matières

1 Introduction

Les robots dans les fermes laitières représentent une évolution technologique majeure dans l'industrie agricole permettant d'améliorer la gestion de la production laitière, la rentabilité, le bien-être des animaux et la durabilité environnementale. Ce développement est renforcé par l'introduction de l'intelligence artificielle (IA) et des mégadonnées, qui permettent aux systèmes de traiter des volumes massifs de données pour prendre des décisions plus précises et efficaces. Cet état de l'art explore les applications actuelles des robots dans les fermes laitières, en s'appuyant sur les travaux de plusieurs chercheurs dans ce domaine.

2 Robots de traite automatisée

Les systèmes de traite robotisée représentent l'une des applications les plus anciennes et les plus étudiées dans le domaine de l'automatisation agricole. Martin et al. ont en effet étudié 90 publications parlant de robotique agricole et, entre 1992 et 2022, 60 articles mentionnent des systèmes de traite automatisée comme on peut le voir dans l'image ci-dessous.[9]

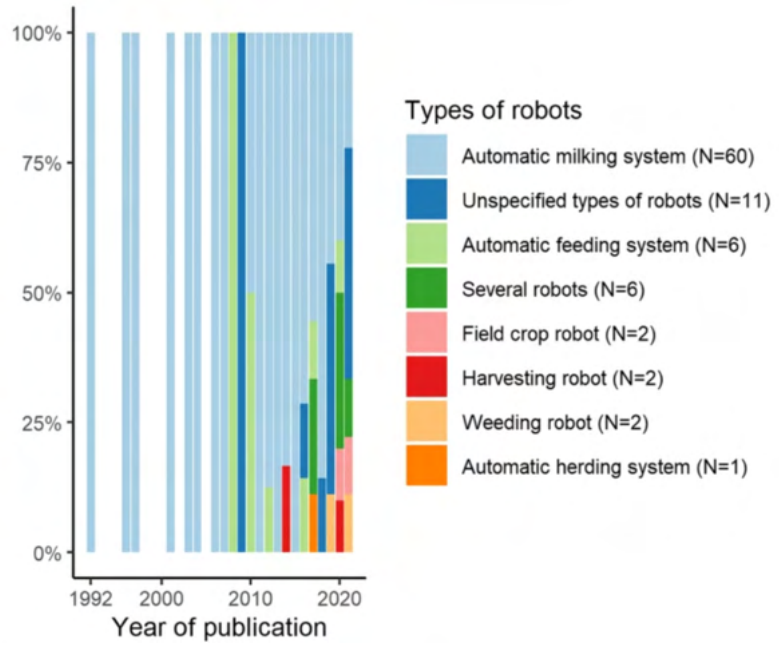


FIGURE 1 – Nombre de publications par sujets entre 1992 et 2022 sur 90 publications [9]

En Europe, l'adoption de ces robots a débuté il y a 30 ans et leur déploiement s'est accéléré dans les années 2010.[9]. Au Québec aussi l'utilisation de ces robots explose. En 2021, 19% des fermes laitières déclaraient posséder une ou des trayeuses entièrement automatisées, ce qui concernait 840 fermes. La vitesse d'adoption de ces système est le même dans le reste du Canada pour une moyenne de 20%. Il est le plus populaire dans l'Ouest où il est présent dans 27% des fermes. Le taux varie selon la taille des fermes et les revenus : plus les fermes sont grosses, plus elles ont tendance à être dotées d'un robot de traite.[2]

Le coût élevé de ces robots et leur rentabilité dépendent de facteurs comme la gestion du troupeau, les conditions climatiques et les paramètres biologiques des animaux. Cependant, ils possèdent de nombreux avantages et permettent de répondre à la demande croissante du marché mondial, avec une prévision de croissance de 35 % d'ici 2030. Ils permettent par exemple aux agriculteurs de réduire le temps passé à traire leurs vaches, de réduire leurs coûts de main-d'œuvre et d'avoir une meilleure qualité de lait. Certains robot de traite automatisée permettent même de s'intéresser au bien-être animal. C'est ce qu'ont fait Ji et al. en utilisant le machine learning pour améliorer la précision des prédictions par rapport aux méthodes statistiques traditionnelles. Ils ont utilisé des systèmes de traite robotisée pour collecter une importante quantité de données sur la production laitière et la santé des animaux. Grâce à celles-ci, ils ont ensuite développé des modèles prédictifs afin de détecter des problèmes comme le stress thermique, les maladies ou les variations de la production de lait. Par exemple, grâce à un cadre basé sur l'apprentissage automatique utilisant des modèles comme XGBOOST, ils ont pu prévoir la production laitière, la composition du lait et la fréquence de traite de chaque vache sur une période de 28 jours, offrant ainsi une gestion plus individualisée des animaux. Un écart entre les prédictions et la réalité peut signifier qu'il y a un problème de santé mental ou psychologique de la vache.[8]

3 Robots pousseurs dans les étables

La demande croissante de produits laitiers de qualité supérieure stimule le développement de l'industrie laitière, mais ce progrès expose aussi de nouveaux défis. Par exemple, dans le processus d'alimentation des vaches, une partie de la nourriture est déplacée, ce qui crée une accumulation qui peut détériorer les aliments non consommés. Actuellement, cette accumulation est gérée manuellement, ce qui nécessite plus de main-d'œuvre et peut réduire l'efficacité de l'alimentation et la productivité du lait. Les robots d'alimentation automatiques sont une solution prometteuse pour résoudre ce problème.

Avec les avancées du deep learning (DL) et de l'apprentissage automatique, des technologies de reconnaissance multimédia ont été appliquées dans divers domaines comme la vidéosurveillance et la navigation robotisée. Ces techniques sont désormais utilisées dans des robots agricoles pour des tâches comme l'alimentation, le transport et la cueillette. Parmi ces robots, les robots automoteurs, qui utilisent des technologies de navigation basées sur des capteurs comme le LiDAR (télédétection), sont particulièrement recherchés. Le LiDAR, avec sa haute précision, est largement utilisé pour l'extraction d'informations dans les environnements agricoles. Cependant, bien que des progrès aient été réalisés pour résoudre les problèmes de navigation des robots dans des scénarios agricoles, les spécificités des fermes laitières n'ont pas été suffisamment étudiées.

Tian et al. proposent alors un nouveau système de vision industrielle pour améliorer la navigation autonome des robots pousseurs dans les étables. Le système utilise des robots auto-conçus et des capteurs LiDAR pour collecter des données puis applique des algorithmes pour extraire des caractéristiques et optimiser les trajectoires de navigation. Plus précisément, le robot pousseur auto-conçu et le lidar 3D ont été utilisés pour recueillir les données du nuage de points de l'étable. Le nuage de points au sol est ensuite retiré par prétraitement et l'algorithme de filtrage extrait les données du nuage de points de la région d'intérêt. Puis, la méthode des moindres carrés (LSM) et le consensus d'échantillonnage

aléatoire (RANSAC) sont utilisés pour extraire les lignes de clôture, les projeter et obtenir des caractéristiques du contour de la frontière, et extraire les lignes de clôture et les chemins initiaux.[13]

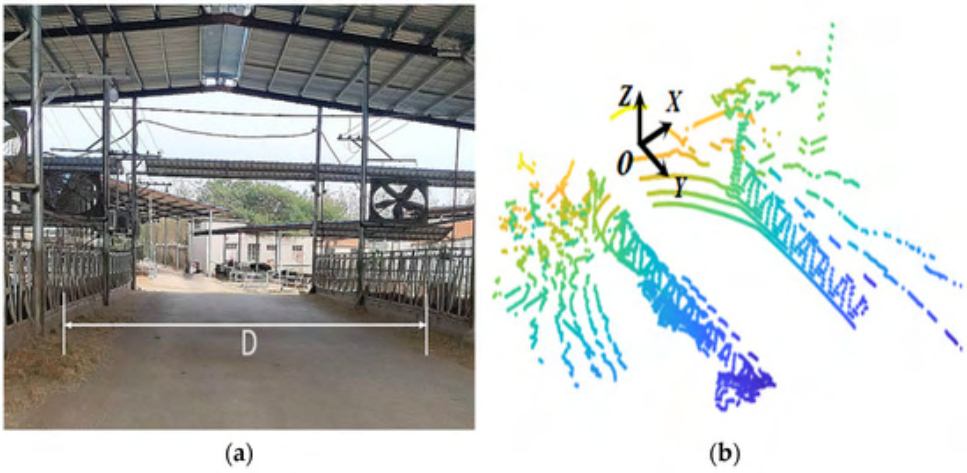


FIGURE 2 – Acquisition de nuages de points en 3D dans une ferme bovine : (a) élevage bovin expérimental, (b) nuage de points en 3D dans la ferme bovin d'origine. [13]

Ce système permettrait aux robots agricoles de naviguer de manière autonome et d'effectuer des tâches de poussée dans des environnements agricoles complexes, offrant ainsi des solutions pour l'élevage de précision et la robotisation des fermes laitières.

Une autre étude propose d'utiliser un distributeur contrôlé d'additifs alimentaires, visant à optimiser le dosage des additifs et faciliter le processus d'alimentation des vaches. Les chercheurs ont en effet observé qu'une distribution alimentaire fréquente (trois fois ou plus par jour) améliore l'intérêt des animaux et augmente leur consommation alimentaire, bien que cela soit coûteux. Par ailleurs, il a été noté que les bovins trient les aliments en faveur de ceux à haute valeur énergétique, négligeant souvent des aliments qui sont essentiels pour leur apport en fibres. L'étude propose donc un robot pousseur abordable, équipé d'un corps de travail à vis et d'un distributeur pour l'alimentation fréquente des animaux avec des aliments énergétiques ou des additifs minéraux. Ce robot améliore ainsi l'efficacité de l'alimentation en augmentant l'intérêt des animaux et en assurant un dosage précis des suppléments. Il peut également effectuer des cycles d'alimentation à intervalles réguliers.[12]

FIGURE 3 – Analogue physique d'un robot pousseur ; 1, sortie du distributeur d'additifs alimentaires ; 2, vis de poussée ; 3, remplisseur avec une fermeture [12]

Le robot se déplace sur une table d'alimentation guidée par des capteurs inductifs, avec un système de télémétrie pour corriger la trajectoire en cas d'écart. En cas d'obstacles ou de contamination sur la trajectoire, un algorithme intelligent, combiné à une caméra stéréoscopique, ajuste la position du robot pour maintenir l'orientation correcte. Ce modèle mathématique du robot pousseur offre une base solide pour la conception et l'amélioration du robot, en analysant les actionneurs et en optimisant son efficacité. L'utilisation de ce robot pourrait grandement réduire l'intensité du travail manuel pour les agriculteurs, facilitant les tâches liées à l'alimentation du bétail, même dans des conditions complexes sur la table d'alimentation.[12]

4 Robots de nettoyage du fumier

Des planchers propres dans les fermes laitières sont d'une importance vitale pour atténuer les émissions d'ammoniac et limiter les risques pour la santé des vaches. Or, le nettoyage des planchers est un acte qui demande une main d'oeuvre importante et régulière. C'est pourquoi dans les exploitations laitières modernes, des robots de nettoyage du fumier se développent pour automatiser la tâche de nettoyage.

Aujourd'hui, de tels robots opèrent dans une relative solitude sans coordination ni coopération au sein du parc. Cependant, Gerrits et al. ont utilisé la simulation d'événements discrets pour tester l'efficacité de différentes stratégies de nettoyage coopératif en équipe dans les fermes laitières. Une attention particulière est accordée à l'impact des différentes compositions d'équipe et des caractéristiques du robot sur le routage de l'équipe. Leurs résultats semblent prometteurs car ils ont montré que la propreté minimale peut être augmentée en déployant des équipes tout en réduisant le nombre d'interactions vache-robot.[6]

5 Impact de ces nouveaux robots sur le bien-être animal

Nous venons de voir que les nouveaux robots présents dans les fermes laitières avaient de nombreux avantages. Ils permettent de réduire la main d'oeuvre et d'accélérer les processus mais aussi de favoriser le bien-être animal en contrôlant par exemple la qualité du lait des vaches dans les robots de traite automatisée, en améliorant leur alimentation grâce à des robots pousseurs ou en nettoyant leurs planchers grâce aux robots de nettoyage du fumier. Cependant, peu d'études sont encore réalisées sur le stress que ces robots peuvent apporter aux animaux alors que leur apparition n'est pas sans conséquence sur la vie des vaches. Certains chercheur commencent donc à s'intéresser à l'impact de ces robots sur le bien-être physique et mental des vaches.

Prenons l'exemple des robots de traite qui sont à première vue bénéfiques pour le bien-être des vaches. Ce sont en effet des technologies novatrices qui prennent en charge le travail de l'élevage laitier et réduisent la nécessité des interactions entre les humains et les animaux. En remplaçant la traite « conventionnelle » effectuée deux fois par jour par des humains par un système censé permettre aux vaches d'être traites automatiquement quand elles le souhaitent, on affirme que la traite robotisée présente des avantages en matière de santé et de bien-être pour les vaches, accroît la productivité et offre des

bénéfices en termes de mode de vie pour les éleveurs laitiers. [7] Cependant, si ces robots permettent aux vaches d'avoir plus de liberté en allant à la traite quand elles le veulent et d'être moins stressées par les humains, peu d'études sont menées sur l'impact des interactions entre ces robots et les vaches. De plus, cette automatisation change également la perception des éleveurs envers leurs animaux, ce qui a un impact sur leur bien-être.

Une machine à été créée par Pastell et al. pour mesurer le stress des vaches au moment de cette traite automatisée. Quatre balances à jauge de contrainte ont été installées dans un robot de traite après une inspection minutieuse des positions des jambes des 40 vaches d'un troupeau.

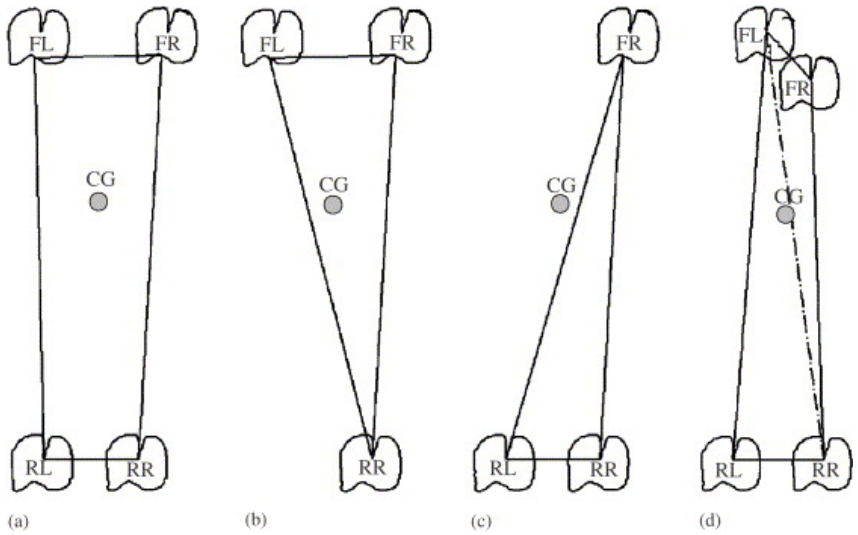


FIGURE 4 – Stabilité d'une vache. CG, centre de gravité ; RL, sabot arrière gauche ; RR, sabot arrière droit ; FL, sabot avant gauche ; FR, sabot avant droit. (a) Debout sur quatre jambes ; (b) jambe gauche arrière vers le haut ; (c) jambe gauche avant vers le haut ; (d) jambes avant proches l'une de l'autre [11]

Pastell et al. ont ensuite relié les balances à un ordinateur pour récolter des données permettant de mesurer la charque de chaque jambe ainsi que le poids moyen, la variation du poids de chaque jambe, le poids total, le nombre de coups de pied, la fréquence des coups de pieds ou encore le temps total passé dans le robot de traite. Grâce à cette machine, il est également possible d'analyser le comportement de pas et de coups de pied de la vache pendant les différentes étapes de la traite, du lavage, et de la déconnexion. De cette manière, il est aussi possible de surveiller le niveau d'activité de la vache et la façon dont celui-ci change au moment du passage dans le robot. Pastell et al. ont réussi à montrer que leur machine fonctionne. Elle pourra donc être utilisée à l'avenir pour déterminer si les robots de traite automatisée entraînent du stress chez les vaches ou non.[11]

6 Utilisation de la robotique et du machine learning au service du bien-être animal

Etant donné que le bien-être physique et mental des vaches devient de plus en plus important, des technologies ont été développées spécialement pour se consacrer à cela. En effet, nous avons déjà vu dans cet article des robots permettant de vérifier le bon état de

santé des vaches mais jusqu'à présent la fonction principale du robot était avant tout de réduire la main d'oeuvre et d'améliorer les rendements. A présent nous allons donc voir comment la robotique et le machine learning peuvent être utilisés directement au service du bien-être animal.

6.1 Intégration de la vision par ordinateur et des techniques non-invasives

La surveillance du stress et du bien-être des vaches est essentielle, mais les méthodes actuelles (stéthoscopes, thermomètres, etc.) sont souvent invasives et biaisées. De nouvelles approches, comme la vision par ordinateur et la télédétection, permettent une évaluation non invasive des paramètres physiologiques et comportementaux.

Dans leur étude, Fuentes et al. explorent l'utilisation de caméras vidéos RVB et infrarouges thermiques (IRTV) couplées à des modèles d'apprentissage automatique pour surveiller le bien-être et la productivité des vaches laitières. Grâce à des installations de traite robotisées, ils ont testé des méthodes non invasives permettant d'estimer la fréquence cardiaque, la respiration et les mouvements des vaches, tout en prédisant la température oculaire, la production et la qualité du lait. Le modèle développé, basé uniquement sur des caméras RVB, a montré une grande précision (R = 0,96) dans la prédiction de ces paramètres. De plus, Fuentes et al. ont démontré que la température oculaire, extraite des images thermiques, peut être un indicateur fiable du stress des vaches. Cette étude présente donc une nouvelle approche pour estimer les informations physiologiques et de productivité importantes basées sur des caméras vidéos visibles qui peuvent être abordables, précises et permettent l'automatisation de la surveillance du stress des vaches sans les déranger.[5]

6.2 Utilisation du machine learning pour réduire le stress thermique

Le stress thermique est l'un des principaux facteurs affectant le bien-être des vaches. Il peut entraîner une baisse de l'appétit, réduisant la production laitière, une augmentation du rythme respiratoire, signalant une détresse physiologique ou encore une réduction de l'activité et une modification des comportements naturels. Avec le réchauffement climatique, les vaches sont de plus en plus touchées par le stress thermique.

Dans une étude, Bovo et al. essayent d'améliorer le bien-être des vaches laitières de plusieurs manières grâce à l'intégration des technologies et de l'analyse prédictive. Ils utilisent un modèle basé sur l'algorithme Random Forest pour prédire le rendement laitier quotidien des vaches en fonction de sept caractéristiques, dont l'indice température-humidité (THI) sur plusieurs jours. Ce modèle a été testé sur des données collectées en Italie entre 2016 et 2017 et a montré une capacité fiable à détecter l'impact du stress thermique sur la production laitière, avec une erreur de prédiction relative de 18 % sur un jour et seulement 2 % sur la production totale.

Cet algorithme permet donc de prévenir les baisses de rendement en détectant l'impact du climat sur la production laitière et permet aux éleveurs d'anticiper les vagues de chaleur et de mettre en place des mesures adaptées (ventilation, brumisation, ajustement de l'alimentation) pour leurs vaches. Une gestion proactive du climat de l'étable améliore

ainsi le confort thermique des animaux.[1]

Dans leur étude, Fuentes et al. utilisent également des capteurs environnementaux pour collecter des données provenant de stations météorologiques automatiques. Ces données sont intégrées aux informations sur les animaux pour mieux comprendre l'impact de l'environnement sur la santé des vaches et la production laitière. Ils utilisent également un suivi biométrique avec des images thermiques et des vidéos RGB comme nous l'avons vu dans la partie précédente. Grâce à cela, ils peuvent surveiller la fréquence cardiaque, le rythme respiratoire et la température corporelle des vaches. Ces technologies sont cruciales pour détecter les signes de stress thermique et d'autres problèmes physiologiques.[4]

Grâce à des modèles de machine learning, ils ont ensuite analysé les données. Ces modèles ont permis de prédire avec une grande précision certains paramètres clés, comme le rendement laitier, la composition du lait, et la fréquence de traite des vaches sur une période donnée. Les chercheurs ont utilisé des techniques de validation croisée pour s'assurer de la robustesse et de la fiabilité des modèles. Comme expliqué précédemment, l'étude de ces facteurs permet de remonter au niveau de stress thermique des vaches.

En cas de stress thermique, Fuentes et al. proposent de faire passer les vaches dans une station de refroidissement avant de les traire.

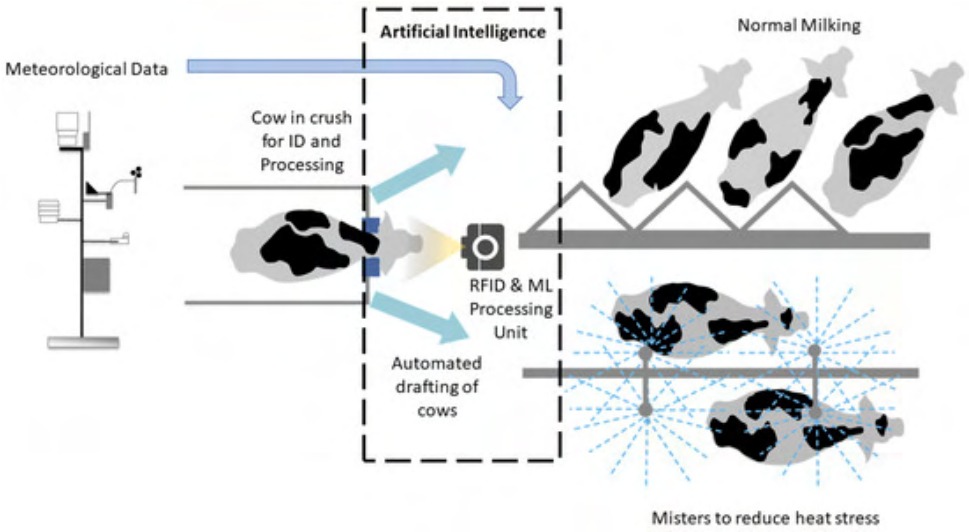


FIGURE 5 – Application proposée d'intelligence artificielle (IA) basée sur le traitement automatisé des stations météorologiques et du système d'identification par radiofréquence (RFID) pour la saisie de données spécifiques sur les vaches et le traitement par apprentissage automatique (ML). Ce système active le système de portillon pour amener les vaches à un système de refroidissement ou à une traite normale.[4]

L'un des principaux avantages des technologies utilisées dans cette étude est leur capacité à surveiller le bien-être des vaches en temps réel. En intégrant des systèmes d'IA, il est possible de détecter les signes de stress thermique chez les animaux, ce qui permet d'intervenir rapidement pour améliorer leur confort et réduire les risques de baisse de productivité. Fuentes et al. ont également proposé l'implémentation de ces modèles de machine learning dans les fermes laitières pour évaluer et améliorer automatiquement le stress thermique des vaches. Ces modèles, une fois intégrés aux systèmes de refroidissement et aux grilles automatisées de tri des animaux, pourraient permettre de maintenir les animaux dans des conditions optimales. De plus, bien que les technologies utilisées dans

cette étude proviennent d'une ferme laitière robotisée, les chercheurs soulignent que les modèles d'IA et de machine learning développés peuvent être appliqués à des fermes laitières traditionnelles. En effet, l'application de ces technologies ne nécessite que des ajouts technologiques minimes, comme des systèmes automatisés de refroidissement et de tri. Cela permettrait donc aux petites et moyennes exploitations laitières de rester compétitives sur un marché mondial de plus en plus exigeant.[4]

6.3 La robotique et l'intelligence artificielle au service du bien-être physique des vaches

La robotique et l'intelligence artificielle permettent également de s'intéresser au bien-être physique des vaches laitières. En effet, dans leur étude, De Vries et al. mettent en évidence le rôle croissant de l'intelligence artificielle et de l'apprentissage automatique dans l'industrie laitière. Ces technologies permettent d'analyser en temps réel des images vidéo pour identifier le bétail, évaluer son état corporel et sa température, et surveiller la consommation d'aliments. Elles détectent également des changements de comportement pouvant signaler des maladies comme la boiterie ou l'oestrus. En intégrant divers ensembles de données (santé, génétique, comportement, environnement), l'IA améliore les prévisions de performances, notamment la fertilité des vaches.[3]

La machine de Pastell et al. dont nous avons parlé dans la partie 5 permet également de s'intéresser au bien-être physique des vaches en plus de leur stress. En effet, lors de cette expérience, les variations des valeurs de chaque vache ont été suivies et la santé des jambes des vaches a été observée. Une analyse préliminaire des données a également fourni des éléments de preuve suggérant que les troubles des membres et des sabots pouvaient être détectés grâce à ce système.[11]

7 Amélioration de l'efficacité énergétique et réduction des émissions de gaz à effet de serre

La robotique et l'intelligence artificielle permettent également de réduire l'impact environnemental des fermes laitières. Cela est indispensable car l'industrie laitière est confrontée à des défis majeurs en matière de réduction des émissions de gaz à effet de serre, notamment le méthane (CH_4) produit par la fermentation entérique des vaches et l'oxyde nitreux (N_2O) issu de la gestion du fumier. Face à l'accélération du changement climatique, il devient essentiel d'adopter des solutions innovantes pour limiter ces impacts.

Par exemple, dans leur publication, Neethirajan et al. explorent comment les méga-données et l'intelligence artificielle peuvent transformer l'industrie laitière pour atteindre la carboneutralité, un objectif crucial face aux changements climatiques. En prenant le secteur laitier canadien comme étude de cas, ils montrent que ces avancées technologiques peuvent être appliquées à l'échelle mondiale pour améliorer la durabilité environnementale dans l'agriculture.[10]

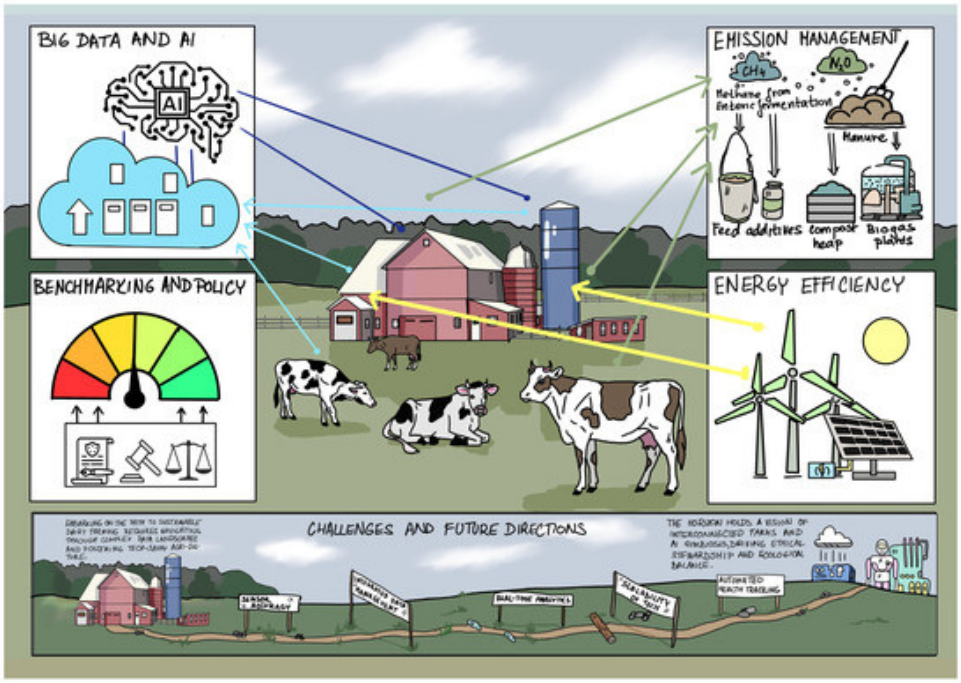


FIGURE 6 – Exploiter les mégadonnées et l'IA pour une production laitière durable – un guide visuel de réduction des émissions et d'étalonnage.[10]

Neethirajan et al. commencent par proposer une optimisation de l'alimentation des vaches afin de réduire les émissions de méthane et d'améliorer l'efficacité alimentaire. Pour faire cela ils ont créé des algorithmes d'IA analysant les données pour ajuster les rations selon la santé et la productivité de chaque vache, les conditions environnementales ainsi que l'impact des aliments sur le rendement laitier et la santé du troupeau. Grâce à cela, ils ont réussi à émettre moins de méthane, à obtenir moins de gaspillage alimentaire et à engendrer une meilleure productivité ainsi qu'une réduction des coûts pour les éleveurs.[10]

Ensuite, Neethirajan et al. ont proposé une gestion intelligente du fumier pour minimiser les émissions de gaz à effet de serre issues du stockage et de l'épandage du fumier. Pour faire cela, ils ont essayé de prédire les meilleurs moments et méthodes d'épandage en fonction des conditions météorologiques et du sol et d'optimiser le processus de digestion anaérobie pour convertir le fumier en biogaz. Ils ont ainsi réussi à obtenir moins d'émissions de méthane et d'oxyde nitreux et à valoriser le fumier en source d'énergie renouvelable.

Enfin, Neethirajan et al. ont optimisé la consommation d'énergie afin d'améliorer l'efficacité énergétique des fermes laitières en utilisant des réseaux de capteurs et d'algorithmes. Le but était de prédire et d'ajuster la consommation d'énergie en optimisant l'utilisation des sources d'énergie renouvelable (solaire, biogaz) et en effectuant une maintenance prédictive des équipements pour éviter les pertes d'énergie. Ainsi, ils ont réussi à réduire la consommation d'énergie fossile, à diminuer les coûts énergétiques et à obtenir une exploitation plus autonome et écologique.[10]

Ainsi, cette étude montre que l'IA et le Big Data offrent des solutions puissantes pour rendre l'industrie laitière plus durable et carboneutre. En optimisant l'alimentation, la gestion du fumier et la consommation d'énergie, ces technologies permettent à la fois de réduire l'impact environnemental et d'améliorer la rentabilité des fermes. Cette transformation s'inscrit dans une vision plus large d'une agriculture durable et résiliente,

contribuant aux efforts mondiaux de lutte contre le changement climatique.

Cependant, certaines limites persistent. L'implémentation de ces technologies requiert des infrastructures coûteuses et une connectivité stable, ce qui peut limiter leur adoption, notamment dans les petites exploitations ou les zones rurales. De plus, l'absence de normes universelles pour mesurer l'efficacité des solutions basées sur l'IA pose un défi en matière de régulation et d'évaluation des performances environnementales. L'interprétation des données reste également un enjeu, car les modèles d'IA dépendent de données agricoles souvent incomplètes ou biaisées, ce qui peut fausser les résultats. Par ailleurs, l'utilisation de ces technologies par les agriculteurs nécessite une formation et un accompagnement adaptés, tandis que la consommation énergétique liée au traitement des données pourrait contrebalancer les bénéfices écologiques attendus. Enfin, l'étude repose sur une vision optimiste des impacts à long terme, sans prise en compte des délais d'adoption ni des contraintes économiques qui pourraient freiner l'intégration de ces technologies.[10]

8 Perspectives d'avenir et défis

L'avenir de l'industrie laitière semble prometteur avec l'intégration croissante de la robotique et de l'intelligence artificielle. Les progrès technologiques permettront une automatisation accrue, augmentant ainsi l'efficacité des fermes laitières tout en améliorant le bien-être animal. L'intégration de technologies avancées, telles que la vision par ordinateur, les capteurs biométriques et les algorithmes de machine learning, ouvrira la voie à des systèmes de production laitière plus intelligents, capables de répondre aux besoins spécifiques de chaque animal et d'optimiser la gestion des ressources.

Les robots de traite et les autres systèmes automatisés devraient se perfectionner pour effectuer des tâches avec une plus grande précision, réduisant ainsi les besoins en main-d'œuvre et permettant aux agriculteurs de mieux gérer la santé et la productivité des vaches. Parallèlement, ces technologies offriront des solutions de plus en plus personnalisées pour chaque vache, permettant un suivi individualisé de leur état de santé, de leur alimentation et de leur confort.

L'un des domaines où les technologies émergentes peuvent avoir un impact significatif est la gestion du stress thermique. L'introduction de systèmes automatisés permettant de réguler l'environnement des vaches en fonction de leur température corporelle en temps réel aidera à réduire le stress thermique, contribuant ainsi à maintenir leur productivité tout en favorisant leur bien-être. Ces technologies pourront également s'étendre à d'autres aspects de la gestion de la ferme, comme la régulation des conditions de vie et le suivi des comportements.

Enfin, l'amélioration de l'efficacité énergétique et la réduction des émissions de gaz à effet de serre sont également des objectifs importants pour l'industrie laitière de demain. Les robots et les systèmes automatisés, en optimisant l'utilisation des ressources, pourraient jouer un rôle clé dans la réduction de l'empreinte écologique des exploitations. Cette évolution pourrait se traduire par une gestion plus efficace des intrants, un meilleur contrôle des émissions et une exploitation plus durable des terres agricoles.

Cependant, l'adoption généralisée de ces technologies soulève plusieurs défis. D'une

part, des questions éthiques liées à l'automatisation et à la collecte de données sensibles devront être abordées : les régulations concernant la protection des données et la sécurité des informations des exploitants seront primordiales. D'autre part, la robotisation croissante pourrait avoir un impact sur l'emploi dans les zones rurales, en particulier dans les secteurs moins qualifiés de l'agriculture. Il sera alors essentiel de trouver un équilibre entre la modernisation des fermes et les impacts sociaux et économiques liés à l'automatisation. De plus, le coût d'investissement initial pour l'achat de ces systèmes et leur entretien peuvent constituer un obstacle pour certaines exploitations. Par ailleurs, l'intégration des données collectées par ces robots dans des systèmes de gestion efficaces reste un défi technologique et logistique. Il est également important de mentionner que de nombreuses bases de données liées aux systèmes robotisés existent souvent en silos non connectés, ce qui peut limiter le potentiel des technologies d'IA à exploiter pleinement ces informations pour la prise de décision.[10]

L'application de la robotique et de l'intelligence artificielle dans les fermes laitières ouvre donc la voie à une gestion plus précise et plus durable de la production laitière. Grâce aux robots et aux algorithmes de machine learning, les fermes peuvent optimiser l'alimentation des vaches, réduire les émissions de gaz à effet de serre, améliorer l'efficacité énergétique et surveiller la santé des animaux de manière plus proactive. Cependant, pour que ces technologies aient un impact significatif, il est crucial de résoudre les défis liés à l'intégration des données et aux coûts d'investissement. L'avenir de la production laitière sera probablement marqué par une plus grande adoption de ces technologies, avec une attention particulière portée sur la durabilité et le bien-être animal. Les perspectives d'avenir dans l'agriculture laitière robotisée sont donc riches, mais elles nécessitent une prise en compte des défis technologiques, éthiques et sociaux pour garantir que ces avancées profitent à l'ensemble de la société et des animaux.

Références

[1] Marco Bovo, Miki Agrusti, Stefano Benni, Daniele Torreggiani, and Patrizia Tassinari. Random forest modelling of milk yield of dairy cows under heat stress conditions. *Animals*, 11(5), 2021.

[2] Le Bulletin. Les robots de traite ont doublé en cinq ans, 2025. Accessed : 2025-03-27.

[3] Albert De Vries, Nikolay Bliznyuk, and Pablo Pinedo. Invited review : Examples and opportunities for artificial intelligence (ai) in dairy farms*. *Applied Animal Science*, 39(1) :14–22, 2023.

[4] Sigfredo Fuentes, Claudia Gonzalez Viejo, Brendan Cullen, Eden Tongson, Surinder S. Chauhan, and Frank R. Dunshea. Artificial intelligence applied to a robotic dairy farm to model milk productivity and quality based on cow data and daily environmental parameters. *Sensors*, 20(10), 2020.

[5] Sigfredo Fuentes, Claudia Gonzalez Viejo, Eden Tongson, Nir Lipovetzky, and Frank R. Dunshea. Biometric physiological responses from dairy cows measured by visible remote sensing are good predictors of milk productivity and quality through artificial intelligence. *Sensors*, 21(20), 2021.

[6] Berry Gerrits, Martijn Mes, Peter Schuur, and Robert Andringa. A simulation model for cooperative robotics in dairy farms. In *2022 Winter Simulation Conference (WSC)*, pages 831–842, 2022.

[7] Lewis Holloway, Christopher Bear, and Katy Wilkinson. Re-capturing bovine life : Robot–cow relationships, freedom and control in dairy farming. *Journal of Rural Studies*, 33 :131–140, 2014.

[8] Boyu Ji, Thomas Banhazi, Clive J.C. Phillips, Chaoyuan Wang, and Baoming Li. A machine learning framework to predict the next month's daily milk yield, milk composition and milking frequency for cows in a robotic dairy farm. *Biosystems Engineering*, 216 :186–197, 2022.

[9] T. Martin, P. Gasselin, N. Hostiou, and et al. Robots and transformations of work in farm : a systematic review of the literature and a research agenda. *Agronomy for Sustainable Development*, 42(66), 2022.

[10] Suresh Neethirajan. Net zero dairy farming—advancing climate goals with big data and artificial intelligence. *Climate*, 12(2), 2024.

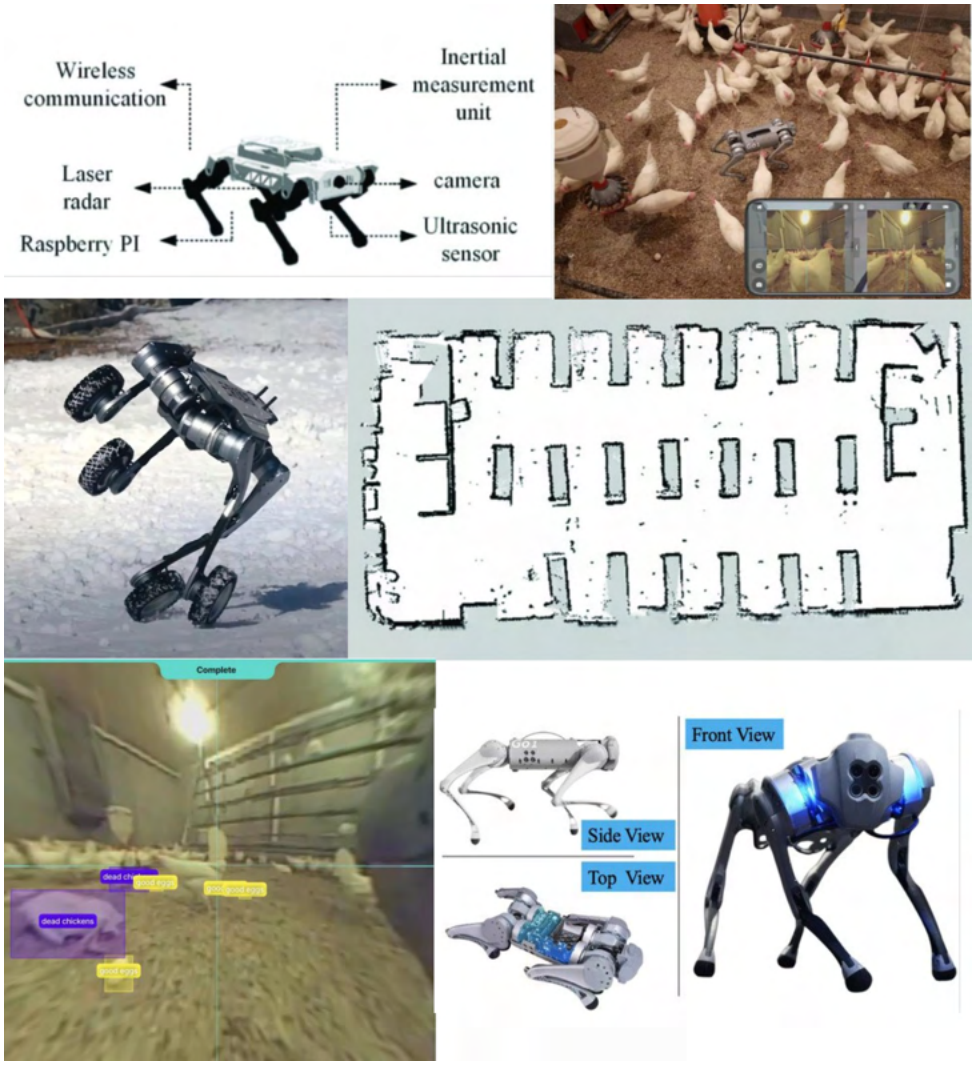
[11] M. Pastell, H. Takko, H. Gröhn, M. Hautala, V. Poikalainen, J. Praks, I. Veermäe, M. Kujala, and J. Ahokas. Assessing cows' welfare : weighing the cow in a milking robot. *Biosystems Engineering*, 93(1) :81–87, 2006.

[12] Dmitriy Yu. Pavkin, Denis V. Shilin, Evgeniy A. Nikitin, and Ivan A. Kiryushin. Designing and simulating the control process of a feed pusher robot used on a dairy farm. *Applied Sciences*, 11(22), 2021.

[13] Fuyang Tian, Xinwei Wang, Sufang Yu, Ruixue Wang, Zhanhua Song, Yinfa Yan, Fade Li, Zhonghua Wang, and Zhenwei Yu. Research on navigation path extraction and obstacle avoidance strategy for pusher robot in dairy farm. *Agriculture*, 12(7), 2022.

Annexe E
Comparaison de robots mobiles agricoles

Laura Jouvet



McGill
UQÀM
Université du Québec à Montréal
well-e
AI Serving Welfare
ENSTA
IP PARIS

Table des matières

1 Introduction

Face aux enjeux croissants du secteur agricole, tels que la pénurie de main-d'œuvre, la nécessité de pratiques durables et la gestion de vastes territoires, l'automatisation devient une réponse incontournable. L'utilisation des robots mobiles dans l'agriculture moderne est en forte croissance et principalement l'utilisation de robots à roues pour des tâches comme la récolte et la surveillance comme nous pouvons en voir un exemple sur l'image suivante avec un robot arracheur de mauvaises herbes. Cependant, les robots à roues rencontrent des limites dues à leur configuration cinématique et à la topologie du terrain et bien que certains robots à roues améliorés offrent une meilleure manœuvrabilité, leur utilisation reste limitée à la surveillance.[4]

FIGURE 1 – Robot La Chèvre de Nexus Robotics arracheur de mauvaises herbes

Dans ce contexte, les robots à pattes, moins explorés en agriculture, pourraient offrir une alternative intéressante, notamment grâce à leur capacité à s'adapter à des terrains variés comme nous pouvons le voir avec le Lynx de DeepRobotics roulant sur de la neige sur la photo suivante, et à imiter le comportement des animaux. En effet, ces dernières années, les robots à pattes et en particulier les robots quadrupèdes, ont suscité un fort intérêt en raison de leur capacité unique à évoluer dans des environnements complexes, accidentés ou irréguliers. Contrairement aux robots à roues ou à chenilles, les robots à pattes peuvent surmonter efficacement des obstacles tels que des escaliers, se déplacer sur presque tous les types de terrains et exercer une pression moindre sur le sol. Ces qualités en font des candidats idéaux pour des secteurs comme l'agriculture et la foresterie, où les terrains sont souvent difficiles d'accès et peu structurés. L'utilisation croissante de ces robots s'accompagne toutefois de défis techniques majeurs : il est crucial de pouvoir localiser précisément le robot, cartographier l'environnement en 3D et planifier des trajectoires sûres, indépendamment des conditions météorologiques ou du terrain. Ces fonctions exigent des algorithmes avancés de localisation, de cartographie et de planification en trois dimensions. [1]

FIGURE 2 – Robot quadrupède Lynx de DeepRobotics

Les robots quadrupèdes se distinguent ainsi des autres types de robots agricoles (roulants, volants, ou fixes) par leur capacité à se déplacer efficacement sur des terrains accidentés et variés, imitant la locomotion animale. Leur morphologie inspirée des quadrupèdes naturels leur confère plus d'agilité et d'adaptabilité par rapport à d'autres types de robots.[4] Ce document vise à explorer les fondements technologiques, les applications pratiques et les enjeux liés à l'utilisation de ces robots dans l'agriculture, en mettant en avant leur valeur ajoutée face aux autres formes de robotique.

2 Robots quadrupèdes

2.1 Fonctionnement

Les robots quadrupèdes sont des robots marcheurs équipés de quatre jambes leur permettant de se déplacer de façon autonome dans des environnements complexes. Leur intérêt principal par rapport aux autres types de robots réside dans leur capacité à s'adapter aux irrégularités du terrain. Ils sont choisis pour leur compromis entre mobilité, stabilité et facilité de contrôle, en comparaison avec les robots à six pattes (plus stables mais plus complexes à contrôler) et les robots bipèdes (moins stables). Inspirés des mouvements d'animaux comme les chiens, ces robots possèdent une redondance cinématique élevée, ce qui permet une grande variété de mouvements.[4] En équipant ces robots avec des capteurs et des systèmes embarqués, ils sont donc capables de se déplacer de manière autonome dans des environnements présentant de nombreux obstacles, ce qui leur permet de devenir des ressources précieuses pour de nombreuses activités comme par exemple les activités agricoles.

Wang et al. détaillent la conception et la mise en œuvre des aspects mécaniques, électroniques et logiciels de robots quadrupèdes afin de pouvoir les utiliser dans des environnements complexes. Le système de contrôle du robot inclut la mécanique (structure et articulation), les capteurs (pour la détection du terrain et des obstacles) et des algorithmes pour la planification des trajets et l'évitement d'obstacles. En termes de programmation, le robot utilise Python et C++, et les capteurs recueillent des informations environnementales (température, humidité, etc.) qui sont ensuite envoyées au contrôleur. Le contrôle de la direction du robot offre une flexibilité et une précision accrues grâce à des capteurs tels qu'une caméra et des capteurs de pression et d'accélération. Le robot émet également

des alertes en cas de variations anormales de température ou d'humidité.[6]

Zhang et al. proposent également d'utiliser de la technologie SLAM (Simultaneous Localization and Mapping) dans le cadre d'un robot quadrupède agricole autonome, technologie qui permet au robot, équipé de capteurs LiDAR, de se localiser avec précision tout en construisant une carte de son environnement en temps réel. Cette capacité est essentielle pour assurer une navigation autonome efficace, notamment dans des environnements inconnus ou dynamiques. La figure suivante montre des exemples de composants pour un système de navigation autonome d'un robot quadrupède.[8]

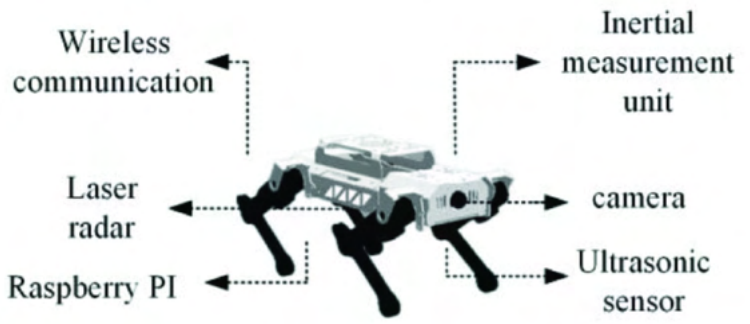


FIGURE 3 – Structure du système de navigation autonome

L'étude se concentre sur les principaux défis rencontrés dans le domaine agricole, notamment la planification de trajectoire, la perception de l'environnement, la localisation et l'évitement d'obstacles. Le système proposé par Zhang et al. repose sur une double cartographie : une carte globale construite par le SLAM, utilisée pour la planification générale, et une carte locale générée en temps réel pour détecter les obstacles et ajuster le parcours. La figure suivante montre un exemple d'une carte globale créée par SLAM. Une fois la carte construite, le robot peut naviguer de manière autonome en suivant un itinéraire prédéfini et en s'adaptant aux obstacles rencontrés en chemin. Le système a été implémenté, testé et validé expérimentalement, prouvant que le robot est capable de réaliser efficacement des tâches agricoles tout en optimisant le temps d'opération et la consommation d'énergie.[8]

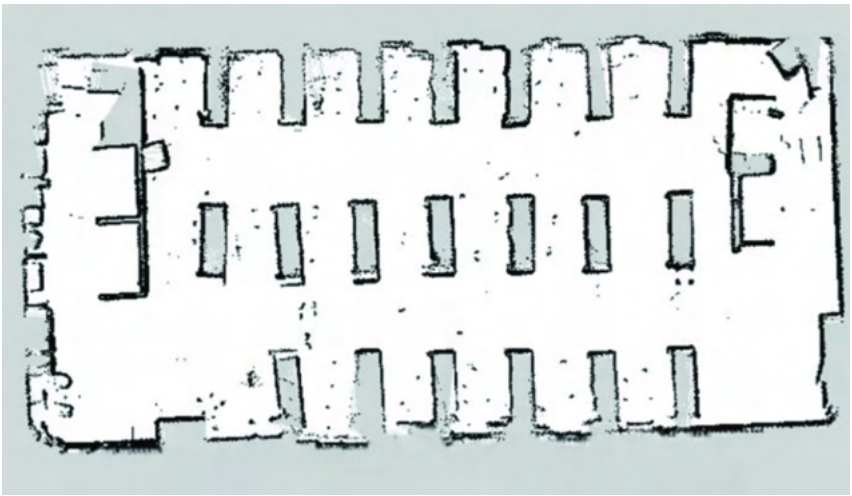


FIGURE 4 – Carte globale créée par SLAM [8]

En conclusion, ces articles démontrent que le SLAM, associé à une combinaison de capteurs intelligents et d'algorithmes de navigation avancés, permet de doter les robots quadrupèdes agricoles d'une véritable autonomie sur le terrain.

2.2 Avantages par rapport aux autres types de robots

Comme nous l'avons vu dans l'état de l'art des robots agricoles en annexe, de nombreux robots à roues sont utilisés pour diverses tâches agricoles. Cependant, ceux-ci sont limités dans les environnements complexes et dès qu'il y a des obstacles à franchir ou des marches. Quail et al. étudient la différence de performances entre un robot à roues (Husky) et un robot à pattes (Spot) sur divers terrains (gravier, herbe, sable, argile, ou même chaussée) et avec différents parcours (circulaires, carrés et oblongs). [4] Les robots ont suivi des trajectoires programmées en Python via ROS à une vitesse de 0,5 m/s. Trois essais ont été réalisés pour analyser l'odométrie ainsi que la consommation d'énergie totale et instantanée pour chaque trajectoire et manœuvre. Les résultats montrent des performances odométriques similaires comme nous pouvons le voir sur la figure suivante, mais avec des problèmes de glissement pour le Husky sur le gravier. Le robot à pattes, lui, a présenté un comportement odométrique stable quel que soit le terrain.

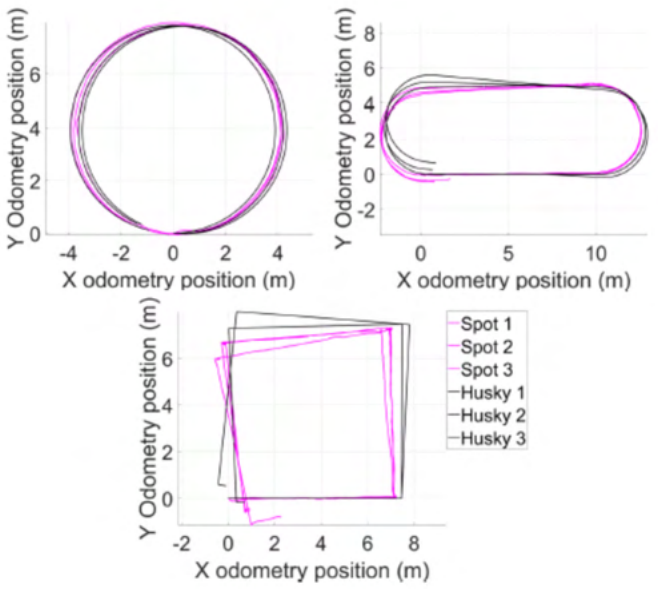


FIGURE 5 – Reconstruction odométrique du parcours suivi sur argile. Des résultats similaires ont été obtenus pour les autres terrains à l'exception du gravier pour le robot à roues. [4]

Comme montré dans la figure 6, Quail et al. ont trouvé que le Spot consomme généralement plus d'énergie que le Husky, mais de manière constante, quel que soit le terrain. Le Husky consomme en effet globalement moins d'énergie sauf sur la trajectoire carrée, où sa vitesse angulaire plus faible a prolongé la durée du test. Cela témoigne de la grande adaptabilité du robot à pattes, le rendant particulièrement adapté aux environnements complexes et aux tâches variées. En revanche, le Husky est plus efficace en ligne droite, mais devient moins performant lors des ajustements fréquents d'orientation. Cela le rend mieux adapté aux environnements agricoles réguliers, mais moins au suivi de

cultures spéciales ou à l'inspection de précision. Grâce à ses multiples degrés de liberté, le Spot peut effectuer des inspections sous différents angles sans dispositifs mécaniques supplémentaires, ce qui le rend utile pour des applications comme l'application ciblée de pesticides ou la détection de maladies. En conclusion, le robot à pattes (Spot) est plus constant et adaptable, ce qui pourrait être un atout pour l'agriculture de précision.[4]

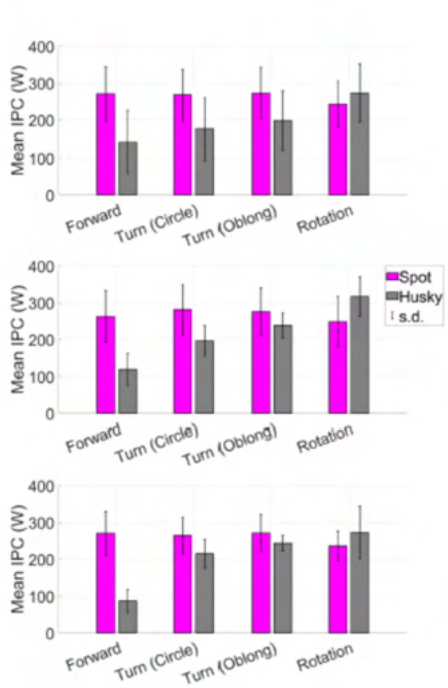


FIGURE 6 – Consommation moyenne de puissance pendant les manœuvres : pour l'argile (figure du haut), le gazon plat (milieu) et la chaussée (bas) [4]

3 Utilisation en agriculture

Comme nous l'avons vu dans la partie précédente, l'aspect mécanique des robots quadrupèdes couplé à certains capteurs et systèmes embarqués permettent à ces robots de se déplacer dans des environnemnt très complexes, ce qui est un atout majeur pour de nombreuses applications en agriculture.

3.1 Surveillance et gestion

Aujourd'hui, l'agriculture évolue en faveur du bien-être animal si bien que les pratiques changent. Les animaux qui étaient autrefois enfermés dans ces petits enclos commencent à devenir plus libres de se déplacer où ils veulent. C'est la cas par exemple des vaches laitières ou des poules qui ne vivent plus dans des cages. Cependant, ces changements impliquent des nouvelles méthodes de surveillance et de gestion des animaux. Yang et al. expliquent par exemple que dans les poulaillers sans cage il y a des problèmes de ramassage des oeufs et des poulets morts. Pour faire face à ce problème, ils ont décidé d'utiliser le "Unitree Go 1", un robot quadrupède équipé de 34 articulations et d'une perception panoramique à 360 degrés présenté sous différents angles sur la figure 7. Il intègre un système d'IA avec un processeur puissant pour analyser en temps réel les flux

vidéo captés par ses caméras RVB et de profondeur, permettant de détecter les œufs et les poulets morts.[7]

FIGURE 7 – Vue tridimensionnelle du chien robot «Unitree Go 1» [7]

Ce robot, déployé deux fois par jour, effectue une inspection complète de la ferme en suivant un parcours défini par un opérateur humain. Un modèle de réseau neuronal convolutif (CNN) a été utilisé pour détecter les œufs au sol et les poulets morts comme présenté dans la figure 8. En utilisant un ensemble de 1200 images, il a montré une précision moyenne de 85,39 % à 90,59 % avec le modèle YOLOv8m.

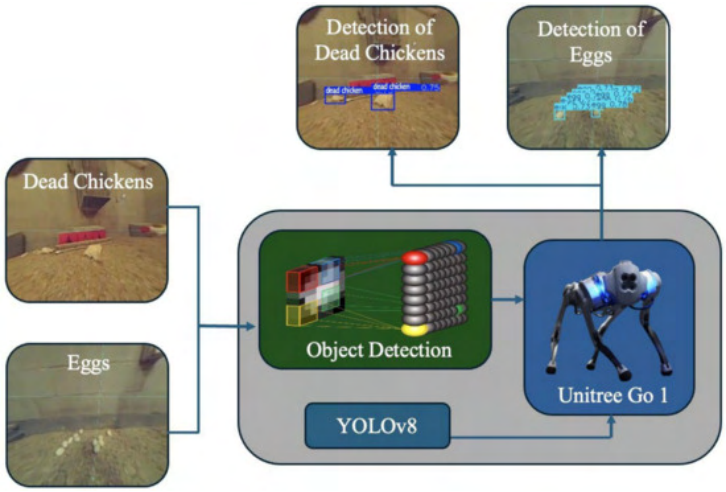


FIGURE 8 – Aperçu du système de détection automatisé pour la gestion des volailles à l'aide de YOLOv8 et Unitree Go 1 Robotic [7]

Ce système permet de repérer efficacement les œufs au sol, même dans des conditions de faible luminosité, et d'identifier les poulets morts au sein du troupeau. Ce robot permet donc de ramasser les oeufs et les poulets morts de manière efficace, sans introduire d'humain au contact des poules, ce qui pourrait par ailleurs favoriser leur bien être. L'étude s'intéresse d'ailleurs également à l'interaction des poules avec le robot. Initialement, les poules ont manifesté une réaction de peur, suivie d'une phase de curiosité et d'interaction, avant de les accepter comme un élément normal de leur environnement. Cela souligne l'importance de la gestion de l'acclimatation des animaux à la présence de la robotique.

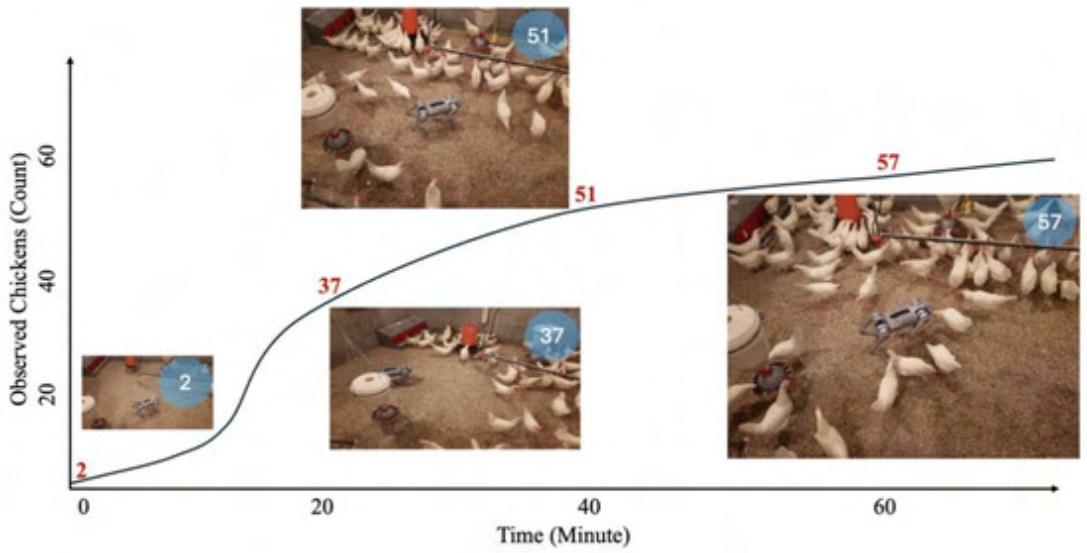


FIGURE 9 – Série chronologique des interactions entre les poulets et le robot. L'axe des x représente le temps (en minutes) et l'axe des y représente le nombre de poulets observés. [7]

En comparaison avec des études antérieures, où d'autres robots comme le PoultryBot ont utilisé des scanners laser et des bras robotisés pour détecter les œufs, les résultats obtenus par Yang et al. ont montré une efficacité supérieure dans un environnement sans cage, avec un seul robot capable de détecter à la fois les œufs et les poulets morts. Toutefois, la collecte des œufs et l'enlèvement des poulets morts restent des défis. Les solutions actuelles incluent l'utilisation de bras mécaniques et de systèmes d'aspiration souple pour les œufs, et des bras plus puissants ou des robots supplémentaires pour les poulets morts. Cette recherche démontre néanmoins le potentiel des robots bioniques quadrupèdes pour automatiser efficacement la gestion des œufs au sol et des poulets morts dans des installations avicoles sans cage, tout en ouvrant la voie à des améliorations futures pour la manipulation d'objets plus lourds et la navigation dans des espaces plus restreints.[7]

3.2 Gardiennage

Si les robot quadrupèdes permettent de remplacer certaines tâches humaines, ils pourraient également remplacer les chiens de troupeau dans les tâches de gardiennage. C'est ce qu'ont essayé de montrer Jové et al. en présentant une méthode innovante d'évaluation des interactions entre des troupeaux et des chiens de berger, qu'ils soient biologiques ou robotiques. [3] L'objectif est de proposer une approche rigoureuse, automatisée et objective pour comparer leurs performances tout en prenant en compte le bien-être animal. Un premier test a été effectué avec un chien de berger qui était évalué selon deux critères : un critère d'obéissance quant au parcours à suivre et un critère d'efficacité pour ramener les moutons dans un enclos que l'on peut voir sur la figure 10. Un second test a été effectué avec un robot dans un enclos fermé cette fois comme montré figure 11. Le test consistait à déplacer calmement le troupeau d'un bout à l'autre de l'enclos à plusieurs reprises, sans test d'obéissance. Le troupeau, peu habitué à tout type de chien, avait été exposé au robot 4 fois en plusieurs mois.

FIGURE 10 – Échantillon de la fin d'un essai réussi où le troupeau est à l'intérieur de l'enclos [3]

FIGURE 11 – Échantillon des enregistrements de l'interaction robot-troupeau à l'intérieur de l'enclos [3]

En terme de rapidité, le chien courait plus vite que le robot (qui est limité à 1,6 m/s) mais, en contrepartie, les moutons paniquaient moins avec les mouvements lents du robot. Finalement, l'étude n'a pas réussi à montrer si un tel robot pourrait être aussi efficace qu'un chien de berger car les expériences étaient différentes et trop d'éléments perturbateurs étaient présents. Cependant, Jové et al. ont quand même réussi à mettre en place la première méthode d'évaluation automatisée utilisant YOLOv8 pour comparer chiens de berger réels et robotiques. Elle représente un pas important vers l'intégration responsable de la robotique dans l'élevage, en conciliant efficacité et respect du bien-être animal.[3]

3.3 Tâches de terrain

En plus de leur capacité à intéragir avec les animaux, les robots quadrupèdes peuvent être très utiles pour les tâches agricoles de terrain demandant une certaine précision dans des environnements complexes.

Dans leur étude, Hendra et al. se concentrent sur l'automatisation de la pulvérisation de pesticides, un processus souvent inefficace et coûteux lorsqu'il est réalisé de manière systématique. En effet, actuellement, ce sont des drones équipés de réservoirs de pesticides et de pulvérisateurs, qui sont utilisés pour la pulvérisation de pesticides, offrant une approche récente avec des véhicules aériens sans pilote (UAV). Cependant, cette méthode a des limites, notamment la pulvérisation uniforme depuis des hauteurs, ce qui peut entraîner une pulvérisation excessive, surtout en présence de vents imprévisibles. Une méthode plus avancée implique des véhicules terrestres sans pilote (UGV), qui utilisent principalement des roues et opèrent dans des zones limitées comme des serres. Un exemple est le robot Bonirob, présenté en figure 12, initialement conçu pour le phénotypage fongique du blé, et désormais utilisé pour des tâches agricoles variées, telles que la lutte contre les mauvaises herbes et l'épandage sélectif d'herbicides.

FIGURE 12 – Robot Bonirob

Cependant, les robots à roues sont très limités par le type de terrain. C'est pourquoi Hendra et al. présentent une plateforme robotique à quatre pattes conçue pour pulvériser des pesticides de manière sélective, en détectant en temps réel les parasites, comme les vers, grâce à la vision par ordinateur. Le robot quadrupède Unitree A1 permet de manœuvrer sur des sols inégaux et de travailler en proximité des cultures. Le module de pulvérisation inclut un bras robotisé, une pompe à eau et une caméra RGB-D pour détecter les ravageurs. Des batteries externes alimentent ces composants. Tous les modules sont connectés via ROS et Wi-Fi et, lorsqu'un ravageur est détecté, le programme d'opération contrôle le bras robotisé pour pulvériser les pesticides avec précision. La distance de pulvérisation est déterminée par la caméra RGB-D, et un contrôleur de bras permet de diriger l'effecteur final vers la cible. Le système présenté figure 13 offre également un positionnement à basse perspective pour identifier les vers cachés sous les feuilles. Il été testé dans des champs de brocolis pour détecter et pulvériser des vers. Des tests préliminaires ont été effectués avec des vers d'imitation dans différentes conditions météorologiques avant d'appliquer le système aux vrais vers. La méthode de suivi des vers a réduit les faux suivis de 89,13 % en temps ensoleillé et de 83,09 % en temps nuageux et le robot a pulvérisé efficacement 13 des 14 zones cibles en un temps de transition rapide. Bien que le système ait bien filtré les tiges et nervures, il a parfois échoué à filtrer des contours ressemblant à des vers, en raison de flous de mouvement et de couleurs similaires.

En conclusion, bien que des améliorations soient nécessaires dans la détection des vers pour gérer les variations d'éclairage, le robot a lui montré une grande précision dans la pulvérisation de pesticides et une grande adaptabilité de terrain.[2]

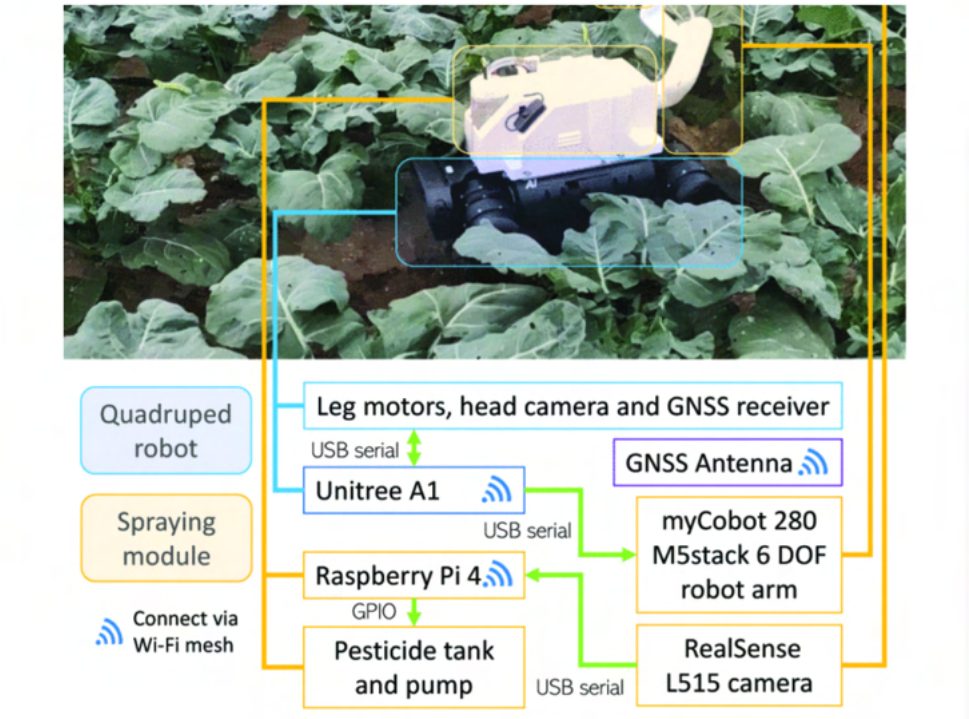


FIGURE 13 – Le robot quadrupède recherche les vers dans un champ de brocoli cru. Les flèches vertes et les symboles de maillage Wi-Fi indiquent les méthodes de communication entre les différents modules.

4 Tableau récapitulatif

Critères	Robots quadrupèdes		Robots roulants	
	Notation	Commentaire	Notation	Commentaire
Mobilité sur terrain complexe	Excellente	Adaptation naturelle aux obstacles	Faible	Bloqués par pentes, boue
Interaction avec les animaux	Très bonne	Intégration fluide dans les troupeaux	Limitée	Encombrants
Autonomie énergétique	Moyenne	Batterie moyenne durée	Bonne	Sur sol plat en ligne droite
Polyvalence des tâches	Haute	Surveillance, gardiennage, interventions	Moyenne	Transport, pulvérisation
Résilience météo	Bonne	Terrain meuble, humidité	Variable	Risque d'enlisement
Coût	Élevé	Technologie complexe	Moyen à élevé	En fonction du modèle
Acceptabilité sociale	Bonne	Formes biomimétiques	Moyenne	Machines imposantes

TABLE 1 – Comparaison détaillée entre robots quadrupèdes et roulants en agriculture

5 Limites actuelles et défis à relever

Comme nous l'avons vu, les robots quadrupèdes possèdent de nombreux avantages pour le milieux agricoles. Néanmoins, leur application implique des défis comme les intéractions homme/robot/animal car l'introduction des robots peut changer l'environnement et la dynamique dans les espaces d'élevage et dans les fermes, que ce soit pour les humains ou les animaux. Par exemple, comme le notent Rodríguez-Lera et al., les moutons peuvent être stressés par les robots, et les bergers doivent être formés pour travailler avec ces nouvelles technologies.[5]

Deux autres obstacles sont les limitations techniques et énergétiques. En effet, le manque d'autonomie (1 à 2,5h pour les robots les plus légers, jusqu'à 10h pour les plus lourds), les erreurs de détection ou encore les pertes de signal de contrôle restent des obstacles majeurs pour l'intégration de ces robots dans des milieux agricoles.[5][3]

Enfin, le coût d'adoption de ces robots peut être un frein pour les agriculteurs. Comme le soulignent Wang et al, des études coût-bénéfice sont nécessaires pour convaincre les exploitants agricoles traditionnels.[6]

6 Conclusion

Les robots quadrupèdes agricoles représentent une avancée technologique majeure pour relever les défis de l'agriculture contemporaine tels que l'efficacité, la productivité et le bien-être des animaux dans l'élevage intensif. Bien que leur adoption soit encore limitée par des contraintes techniques, économiques et sociales, les recherches récentes montrent qu'ils sont capables de surveiller des troupeaux, de faciliter le travail des éleveurs et d'assurer une agriculture plus durable. L'avenir de ces robots passera par une meilleure intégration dans les systèmes de production, une collaboration avec les éleveurs, et un engagement fort pour le bien-être animal.

Références

[1] Jorge Ferreira, A. Paulo Moreira, Manuel Silva, and Filipe Santos. A survey on localization, mapping, and trajectory planning for quadruped robots in vineyards. In *2022 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, pages 237–242, 2022.

[2] Hansen Hendra, Yubin Liu, Ryoichi Ishikawa, Takeshi Oishi, and Yoshihiro Sato. Quadruped robot platform for selective pesticide spraying. In *2023 18th International Conference on Machine Vision and Applications (MVA)*, pages 1–6, 2023.

[3] Beatriz Jové, Alexis Gutiérrez, Camino Fernández, Lidia Sánchez, Francisco J. Rodríguez-Lera, and Vicente Matellán. Quadrupeds robots in herding : Metrics for experimental validation of animal-robot interactions. Grupo de Robótica, Universidad de León, 2023.

[4] Christopher Quail, Evrard Emonot–de Carolis, and Fernando Auat Cheein. Legged robots in the agricultural context : Analysing their traverse capabilities and performance. In *IECON 2023- 49th Annual Conference of the IEEE Industrial Electronics Society*, pages 01–07, 2023.

[5] Francisco J. Rodríguez-Lera, Miguel A. González-Santamarta, Jose Manuel Gonzalo Orden, Camino Fernández-Llamas, Vicente Matellán-Olivera, and Lidia Sánchez-González. Lessons learned in quadruped deployment in livestock farming. 2023.

[6] Xinyue Wang, Jinkun Liu, Ting Zhang, and Jie Zhang. Applied research of agricultural quadruped robots. In *2024 IEEE 13th International Conference on Communication Systems and Network Technologies (CSNT)*, pages 954–957, 2024.

[7] Xiao Yang, Jinchang Zhang, Bidur Paneru, Jiakai Lin, Ramesh Bahadur Bist, Guoyu Lu, and Lilong Chai. Precision monitoring of dead chickens and floor eggs with a robotic machine vision method. *AgriEngineering*, 7(2), 2025.

[8] Jie Zhang, Xinyue Wang, and Liang Zheng. Research on autonomous navigation system of agricultural quadruped robot. In *2024 IEEE International Conference on Mechatronics and Automation (ICMA)*, pages 1286–1290, 2024.

Annexe F
Choix d'un simulateur robotisé pour soutenir les activités de recherche sur les fermes laitières

Laura Jouvet

McGill
UQÀM
Université du Québec à Montréal
wellie
AI Serving Welfare
ENSTA
POLYTECHNIQUE INSTITUT DE PARIS
IP PARIS

Table des matières

1 Introduction

Dans le cadre de la chaire de recherche-innovation en bien-être animal et intelligence artificielle Well-E, des robots mobiles capables de se déplacer dans les exploitations de vaches laitières seront utilisés comme outils de support afin d'analyser le bien-être physique et mental des vaches. Cependant, aujourd'hui il n'existe pas de tels robots, si bien que nous ne savons pas comment le robot interagira avec les vaches et quels pourront être les problèmes rencontrés lors de son déploiement en milieu réel. Ainsi, avant de tester le robot directement dans de vrais scénarios d'utilisation comme dans les étables des fermes universitaires et commerciales, il est nécessaire de réaliser des simulations de déploiement afin d'évaluer les besoins en terme de ressources computationnelles et de créer et de tester des algorithmes tels que des algorithmes de navigation autonome ou de détection et d'identification des vaches.

Le recours aux simulateurs robotiques s'est en effet imposé comme une étape cruciale dans le développement, la validation et le test d'algorithmes pour la robotique mobile. En offrant un environnement sûr, reproductible et économique, la simulation permet d'anticiper des problématiques complexes sans exposer de matériel coûteux ou fragile à des risques inutiles.[1] Elle est d'autant plus importante lorsque le robot interagit avec des êtres vivants comme avec des vaches dans notre cas. Toutefois, le choix du simulateur le plus adapté reste une tâche complexe, tant l'offre est vaste et hétérogène. À travers l'analyse de cinq références récentes, cette synthèse vise donc à dresser un panorama argumenté des principales plateformes de simulation, en mettant en lumière leurs atouts, limites et spécificités, afin d'orienter un choix raisonné pour notre projet de simulation de déploiement d'un robot mobile dans les exploitations laitières.

2 Méthodologie de comparaison

Dans les références étudiées, les simulateurs sont évalués sur la base de critères tirés à la fois d'études expérimentales et de retours d'utilisateurs. Les critères d'évaluation dépendent bien évidemment du type de robot que l'on souhaite simuler mais les critères qui reviennent le plus sont [5, 4, 2] :

— la compatibilité avec ROS
— la modélisation de l'environnement
— la précision des moteurs physiques
— le prix
— l'interface utilisateur
— la complexité de modélisation
— la consommation de ressources
— l'utilisation de capteurs

Tous ces éléments ont été comparés tant de manière qualitative que quantitative. Par exemple, Farley et al. ont mis en place une évaluation quantitative pondérée de quatre simulateurs — CoppeliaSim (anciennement V-Rep), Gazebo, MORSE et Webots — en les confrontant à des données réelles issues du robot Husky A200. [2] Leurs résultats sont présentés dans le tableau suivant et seront décrits dans la partie 3 :

Metric name	Weight	CoppeliaSim	Gazebo	MORSE	Webots
Free to use	4	1.000	1.000	1.000	1.000
Open source	2	0	1.000	1.000	1.000
ROS integration	6	0.800	1.000	1.000	0.800
Programming languages	3	1.000	0.667	0.333	1.000
UI functionality	6	1.000	1.000	0.333	1.000
Model format support	4	1.000	1.000	0	0.250
Physics engine support	3	1.000	1.000	0	0
Real time factor	4	0.914	1.000	0.789	0.849
Average load CPU efficiency	2	0.364	0.174	0.333	1.000
Intense load CPU efficiency	2	0.583	0.304	0.583	1.000
IMU angular velocity accuracy	10	0.875	1.000	0.792	0.867
IMU linear acceleration accuracy	10	1.000	0.713	0.424	0.736
Total	**56**	**49.100**	**49.087**	**32.148**	**44.226**

FIGURE 1 – Résultats d'évaluation de la valeur des métriques pondérées [2]

Ivaldi et al. quant à eux ont effectué une analyse qualitative en effectuant un sondage en ligne sur l'utilisation des outils de simulation dynamique en robotique. L'objectif est ainsi d'aider les chercheurs à choisir le meilleur outil pour leurs projets en robotique, en se basant sur des retours d'expérience des utilisateurs. Le sondage a été rempli par 119 participants, principalement des chercheurs ayant un doctorat (62%) et travaillant principalement dans des universités ou des instituts de recherche. L'article analyse les réponses et fournit des fiches descriptives des outils de simulation les plus pertinents, en se concentrant sur des outils populaires comme Gazebo et CoppeliaSim. Le document met en lumière l'importance de combiner des comparaisons quantitatives avec des retours qualitatifs des utilisateurs pour aider les roboticiens à faire des choix éclairés concernant les outils de simulation adaptés à leurs besoins de recherche. [3]

3 Description de différents simulateurs

Dans cette partie, nous allons décrire les principaux simulateurs cités dans les 5 références en mettant en avant leurs atouts, leurs spécificités et leurs limites afin de pouvoir ensuite choisir celui qui sera le plus adapté à notre projet.

3.1 Gazebo

Gazebo apparaît comme l'un des simulateurs les plus complets pour la robotique mobile, notamment en raison de sa profonde intégration avec ROS. Dans le sondage d'Ivaldi et al., il est le simulateur le plus utilisé avec CoppeliaSim.[3] Il permet de simuler une grande variété de capteurs, d'environnements et de comportements robotiques, ce qui en fait une solution privilégiée pour les recherches en navigation, localisation ou contrôle. Farley et al. lui attribuent un score très élevé de 49.087/56 1, saluant notamment la compatibilité ROS (note parfaite de 1.000), la précision de ses mesures IMU ainsi que son support opensource complet.[2]

FIGURE 2 – Logiciel de simulation Gazebo

Toutefois, Gazebo est également pointé du doigt pour sa gourmandise en ressources CPU et sa complexité de prise en main, [5] comme le rappellent Nogueira et al. qui notent que la personnalisation de robots passe par des fichiers XML (SDF) qui sont peu intuitifs. Gazebo ne permet donc pas de modéliser facilement des scènes, ce qui rend la personnalisation et la modification des robots complexe. [4]

3.2 CoppeliaSim

CoppeliaSim rivalise avec Gazebo en termes de précision et de polyvalence. Dans le sondage d'Ivaldi et al., il est également le simulateur le plus utilisé avec Gazebo.[3] Il arrive également en première position dans l'étude quantitative de Farley et al. avec un score total de 49.100 sur 56 1, grâce à sa grande polyvalence. Il obtient notamment un score parfait (1.000) pour des critères tels que l'interface utilisateur, la prise en charge de plusieurs langages de programmation, ou encore la précision de l'accélération linéaire IMU. [2] CoppeliaSim permet égalelent d'ajouter facilement des capteurs et actionneurs à des robots via une interface graphique, ce qui simplifie la création de robots personnalisés. Les plugins ROS pour CoppeliaSim permettent également une intégration facile des capteurs avec ROS.

FIGURE 3 – Logiciel de simulation CoppeliaSim

CoppeliaSim se distingue donc par une interface graphique intuitive et une grande facilité de modélisation d'environnements complexes, grâce à des fonctionnalités permettant de glisser-déposer des objets et d'inspecter et modifier facilement leurs propriétés. Il prend en charge plusieurs moteurs physiques et propose une personnalisation fine des robots. Nogueira et al. soulignent néanmoins que son intégration avec ROS, bien que possible via des plugins, reste moins fluide que celle de Gazebo.[4]

3.3 Webots

Webots constitue une solution robuste, particulièrement bien adaptée à la simulation de robots mobiles au sol. Il offre un bon compromis entre simplicité d'usage et richesse fonctionnelle. Farley et al. lui attribuent un score de 44.226/56 1, avec de bonnes performances en termes d'efficacité CPU et d'interface utilisateur. [2] Il prend en charge une large gamme de capteurs, comme le LiDAR, les caméras et le GPS. L'étude de Collins et al. confirme son intérêt pour la recherche académique, notamment grâce à une base de données riche en modèles robotiques. [1]

FIGURE 4 – Logiciel de simulation Webots

Cependant, Webots est limité par son support restreint des moteurs physiques et des formats de modèles. [2]

3.4 MORSE

Moins utilisé que ses concurrents, MORSE reste un simulateur open-source pertinent pour des projets spécifiques. Toutefois, son interface réduite et la faible précision de ses capteurs simulés limitent ses applications. Il obtient un score de 32.148 1 chez Farley et al., avec une note très basse pour la précision de l'accélération linéaire IMU (0.424), ce qui en fait une option secondaire pour des projets exigeants. [2]

FIGURE 5 – Logiciel de simulation MORSE

3.5 Unity et AirSim

D'autres outils comme Unity et AirSim sont cités dans l'étude de Robotics Knowledgebase pour leur capacité à offrir une visualisation photoréaliste, notamment dans des domaines comme les véhicules autonomes. AirSim, développé par Microsoft, offre une modélisation physique fine des drones, tandis que Unity, grâce à son moteur de jeu, permet la création d'environnements riches et immersifs. Toutefois, leur usage en robotique nécessite souvent des ajustements importants, notamment pour l'intégration avec ROS, et leur prise en main est très compliquée. [5]

FIGURE 6 – Logiciel de simulation Unity

FIGURE 7 – Logiciel de simulation AirSim

3.6 Tableau récapitulatif

Voici un tableau récapitulatif qui rassemble les principaux critères des simulateurs décrits ci-dessus :

Simulateur	Compatibilité ROS2	Interface utilisateur	Précision capteurs	Modélisation scène	Consommation CPU	Score global [4]
CoppeliaSim	Moyenne	Excellente	Très bonne	Très bonne	Moyenne	49.100
GAZEBO	Excellente	Bonne	Très bonne	Moyenne	Elevée	49.087
Webots	Bonne	Bonne	Moyenne	Bonne	Bonne	44.226
morse-simulator/ morse	Moyenne	Faible	Faible	Faible	Moyenne	32.148
AIRSIM	Faible à moyenne	Moyenne	Très bonne (drones)	Moyenne	Elevée	-
unity	Faible	Excellente	Moyenne à bonne	Excellente	Elevée	-

FIGURE 8 – Tableau comparatif des principaux simulateurs utilisés en robotique

4 Comparaison selon les critères de notre projet

4.1 Compatibilité avec ROS

Le robot que nous allons déployer dans les étables devra gérer plusieurs tâches en même temps : commander ses moteurs, détecter les vaches, naviguer de façon autonome etc. Pour gérer à la fois les capteurs, les actionneurs et la prise de décision lors de la navigation, ROS sera indispensable afin de permettre à tous ces éléments de communiquer entre eux en temps réel via des topics. Nous pouvons donc déjà éliminer AirSim et Unity qui étaient de toute façon des logiciels très spécifiques et difficiles à prendre en main. Nous allons donc devoir choisir entre CoppeliaSim, Gazebo, MORSE et Webots.

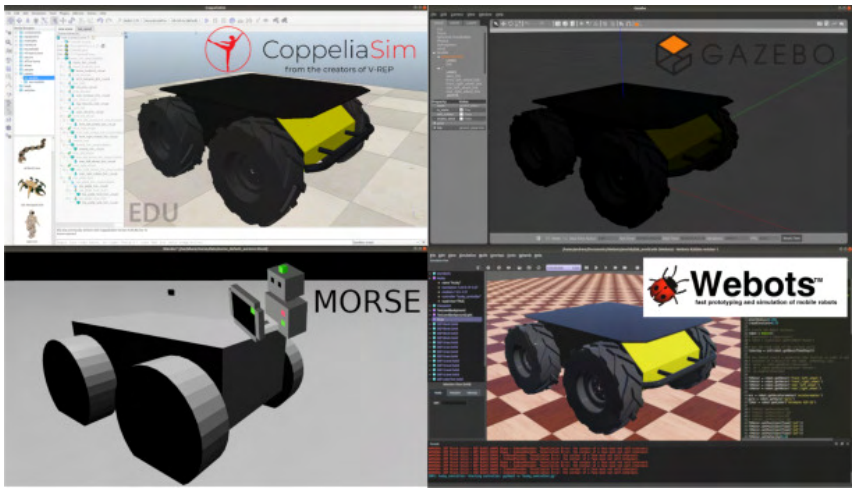


FIGURE 9 – Principaux simulateurs utilisés en robotique

Gazebo est l'intégration par défaut de ROS, avec un ensemble de plugins permettant une communication facile avec ROS via des sujets et services. Il dispose d'une forte communauté et de nombreux plugins, rendant l'intégration simple, notamment avec des robots comme le Pioneer 2DX. CoppeliaSim, n'a pas de nœud ROS natif mais propose des plugins permettant l'intégration à ROS via des scripts Lua.[4] Webots permet aussi de contrôler des robots via ROS assez facilement tandis qu'avec MORSE cela est beaucoup plus compliqué.

4.2 Précision des capteurs

L'un des points les plus importants de cette simulation sera le déplacement du robot de manière autonome. Pour cela, le robot sera équipé de différents capteurs tels que des LiDAR et possiblement d'une IMU.

Dans Gazebo, la modification des modèles de robot nécessite de travailler avec des fichiers SDF, ce qui peut être difficile pour les débutants. Bien que Gazebo propose des modèles préconçus, leur personnalisation peut donc être complexe. CoppeliaSim, en revanche, permet d'ajouter facilement des capteurs et actionneurs à des robots via une interface graphique, simplifiant la création de robots personnalisés. Les plugins ROS pour CoppeliaSim permettent également une intégration facile des capteurs avec ROS.[4] Webots quant à lui prend en charge une large gamme de capteurs et une base de donnée riche en modèles robotiques, ce qui nous permettra de trouver les capteurs que possède Spot et de les utiliser sur un modèle de robot quadrupède. MORSE en revanche n'a pas une bonne précision sur ses capteurs ce qui est un point négatif pour notre projet.[2]

4.3 Modélisation de la scène

CoppeliaSim excelle dans la modélisation d'environnements grâce à son interface graphique simple permettant de glisser et déposer des objets, et d'inspecter et modifier facilement leurs propriétés. Il dispose de nombreux modèles préconçus pour des objets d'infrastructure, facilitant la création rapide de scènes complexes. Gazebo est plus limité à cet égard et nécessite l'utilisation de fichiers XML (SDF) pour personnaliser les modèles, ce qui rend la modélisation plus complexe et moins intuitive.[4] Webots quant à lui se situe entre les deux. La modélisation de scène n'est pas aussi intuitive qu'avec Coppelia mais plus facile qu'avec Gazebo. Pour MORSE, encore une fois, ce critère n'est pas facile à mettre en place.

4.4 Modification du modèle

Dans Gazebo, la modification des modèles de robot nécessite de travailler avec des fichiers SDF, ce qui peut être difficile pour les débutants. Bien que Gazebo propose des modèles préconçus, leur personnalisation peut donc être complexe. CoppeliaSim, en revanche, permet d'ajouter facilement des capteurs et actionneurs à des robots via une interface graphique, simplifiant la création de robots personnalisés[4], tout comme c'est aussi le cas pour Webots.

4.5 Contrôle de la simulation

Gazebo, CoppeliaSim et Webots permettent un contrôle programmé via ROS, avec des services permettant de démarrer/arrêter la simulation et de récupérer ou définir la position des robots. Cependant, Gazebo, bien qu'offrant plus de contrôle sur la simulation à travers ses services ROS, est plus exigeant en termes de ressources matérielles par rapport à CoppeliaSim et Webots.[4]

4.6 Utilisation des ressources CPU

Lors d'une simulation avec plusieurs robots, Gazebo est plus exigeant en termes de CPU, consommant jusqu'à 311 % de la puissance processeur contre 267 % pour CoppeliaSim. Cela est dû à la façon dont chaque simulateur gère la simulation, Gazebo étant constamment actif alors que CoppeliaSim alterne entre des phases actives et inactives.[4] Webots quant à lui engendre moins de consommation GPU que Gazebo et CoppeliaSim.

5 Conclusion

5.1 Synthèse des analyses et des comparaisons

Le choix du simulateur dépend étroitement des objectifs du projet. Pour des développements intégrés à ROS et des expérimentations réalistes sur robots mobiles, Gazebo, Webots et CoppeliaSim restent les références principales, chacune ayant ses forces que ce soit pour l'intégration ROS ou pour la flexibilité et la précision physique. MORSE reste une option plus limitée notamment en terme de précision des capteurs. AirSim et Unity, quant à eux, conviennent mieux à des projets où la visualisation est centrale, comme la simulation de drones ou de véhicules autonomes.

D'après le sondage d'Ivaldi et al., Gazebo et CoppeliaSim sont les simulateurs les plus utilisés par les étudiants et chercheurs en robotique, notamment car ces outils se distinguent par leur support communautaire et leur compatibilité avec différents moteurs physiques [3]. Farley et al. placent également ces deux simulateurs aux deux premières places de leur étude comparative. Ils concluent que CoppeliaSim est aujourd'hui la solution la plus aboutie, tandis que Gazebo constitue une excellente alternative, en particulier pour les projets fortement intégrés à ROS. [2] Webots n'arrive pas loin derrière et peut être une bonne alternative en cas de faible disponibilité de CPU et est même mieux compatible avec ROS que CoppeliaSim.

Gazebo est donc plus intégré avec ROS et mieux adapté pour des projets nécessitant une personnalisation poussée avec ROS. Cependant, Nogueira et al. rappellent qu'il demande plus de ressources et est beaucoup plus difficile à prendre en main et à personnaliser. CoppeliaSim, de son côté, est plus intuitif et offre une interface plus conviviale pour la modélisation et l'ajout de capteurs/actionneurs. Il est donc mieux adapté pour les utilisateurs en robotique pour qui la modélisation de la scène est un critère important ou pour des projets pour lesquels les robots seraient au premier stade de conception et pour lesquels on n'aurait pas encore beaucoup d'information quant à leur comportement.[4] Cependant, l'intégration avec ROS est un peu plus compliqué. Le site internet Robotics Knowledgebase propose donc une bonne conclusion en disant que pour les projets utilisant ROS, Gazebo et CoppeliaSim sont des choix solides grâce à leur compatibilité et leur

capacité de personnalisation mais que Gazebo est plus difficile à prendre en main et plus limité dans la personnalisation des scènes et des robots. [5]

Enfin, Webots propose une bonne compatibilité avec ROS et une consommation de CPU correcte et ce logiciel est un entre deux entre Gazebo et CoppeliaSim en terme de facilité de modélisation des scènes.

5.2 Choix pour notre projet

D'après les critères fixés par notre projet et décrits en partie 4, le meilleur simulateur pour notre robot serait CoppeliaSim, Gazebo ou Webots. Comme nous l'avons vu, Gazebo est le mieux adapté à ROS mais ce simulateur est plus difficile à prendre en main et plus lourd, ce qui posera problème si l'on utilise nos propres ordinateurs. De plus, sur Gazebo il est difficile de personnaliser son robot, d'ajouter des capteurs et de créer et de modifier une scène, ce qui sont des parties imoportantes de notre projet étant donné que nous n'en sommes qu'au début et que nous n'avons aucune idée de comment sera notre robot et des problèmes que nous allons rencontrer dans la simulation. Nous allons donc sûrement devoir réaliser de nombreuses modifications, ce qui n'est pas évident avec Gazebo. CoppeliaSim, bien que moins adapté à ROS permet néanmoins de l'utiliser et est beaucoup plus facile à prendre en main. Il permet également de mieux modéliser des scènes, de contrôler la simulation et de personnaliser des robots, notamment en terme d'ajout de capteurs. Enfin, Webots pourrait être une autre alternative qui possède une bonne compatibilité avec ROS et qui permet de modèliser les scènes de manière assez facile sans demander trop de ressources.

Références

[1] Jack Collins, Shelvin Chand, Anthony Vanderkop, and David Howard. A review of physics simulators for robotic applications. *IEEE Access*, 9 :51416–51431, 2021.

[2] Andrew Farley, Jie Wang, and Joshua A. Marshall. How to pick a mobile robot simulator : A quantitative comparison of coppeliasim, gazebo, morse and webots with a focus on accuracy of motion. *Simulation Modelling Practice and Theory*, 120 :102629, 2022.

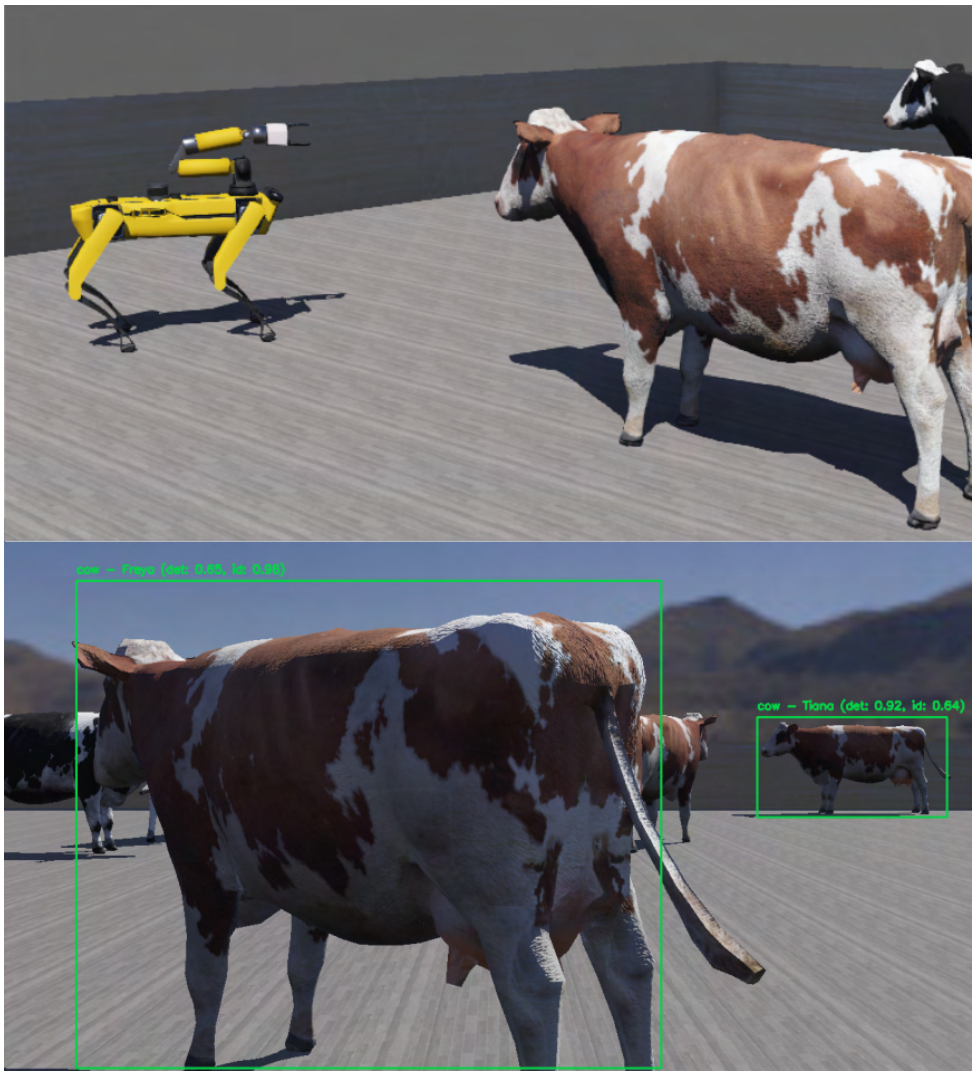
[3] Serena Ivaldi, Vincent Padois, and Francesco Nori. Tools for dynamics simulation of robots : a survey based on user feedback. 02 2014.

[4] Lucas Nogueira. Comparative analysis between gazebo and v-rep robotic simulators. *School of Electrical and Computer Engineering, Universidade de Campinas*, 2023. Email : lucas.afonso@hotmail.com.

[5] Robotics Knowledgebase. Choose a simulator, 2025. Consulté le 7 avril 2025.

Annexe G
Setup Spot WEBOTS

Laura Jouvet



1 Prérequis

L'utilisation de Spot sur Webots nécessite d'utiliser Ubuntu 22.04 et ROS2 Humble.

Installation d'Ubuntu 22.04 et de ROS2 Humble

- Ubuntu 22.04
- ROS2 Humble : Instructions d'installation

Installez ensuite les outils de développement ROS2 et initialisez et mettez à jour rosdep.

```
$ sudo apt install -y ros-dev-tools
$ source /opt/ros/humble/setup.bash
$ sudo rosdep init
$ rosdep update
```

2 Installer WEBOTS

Pour installer webots, tout est indiqué au lien suivant : Installation Webots.

3 Clôner le dossier GitHub

Créez un dossier pour le projet, par exemple *webots_spot*.
Dans ce dossier, créez un dossier *src*.
Clônez le git `https://github.com/MASKOR/webots_ros2_spot.git` à la racine du projet.

4 Installer les dépendances et construire le package

Toujours à la racine du dossier, extraire les packages pertinents, installer les dépendances, compiler et sourcer l'espace de travail.

```
$ mkdir -p webots_spot/src
$ cd webots_spot
$ git clone https://github.com/MASKOR/webots_ros2_spot.git
$ rosdep install --ignore-src --from-paths src -y -r
$ vcs import --recursive src --skip-existing --input src/webots_ros2_spot/
    ↪ webots_ros2_spot.repos
$ chmod +x src/webots_ros2/webots_ros2_driver/webots_ros2_driver/
    ↪ ros2_supervisor.py
```

Enfin, construisez le package et sourcez l'environnement.

```
$ colcon build --symlink-install
$ source install/setup.bash
```

5 Utiliser les commandes de bases

Pour lancer la simulation, aller à la racine du projet et taper la commande suivante :

```
$ ros2 launch webots_spot spot_launch.py
```

Pour ouvrir Rviz2 et visualiser les capteurs de Spot, faire :

```
$ ros2 launch webots_spot nav_launch.py set_initial_pose:=true
```

Pour faire bouger Spot, un script Python existe dans le package :

```
$ ros2 launch webots_spot moveit_launch.py
```

Enfin, pour télécommander Spot au clavier, tapez dans un autre terminal après avoir sourcé l'environnement :

```
$ ros2 run teleop_twist_keyboard teleop_twist_keyboard
ou
$ ros2 run spot_teleop spot_teleop_keyboard for body_pose control
```

6 Apprendre à contrôler Spot

Pour pouvoir contrôler Spot, il faut savoir à quelles commandes nous donne accès le package. Pour cela, nous pouvons observer la liste des topics et des services.

```
$ ros2 service list # Pour obtenir la liste des services
$ ros2 topic list # Pour obtenir la liste des topics
```

De plus, pour visualiser les caméras disponibles sur Spot et les images qu'elles renvoient, nous pouvons utiliser **rqt image view** :

```
$ ros2 run rqt_image_view rqt_image_view
```

6.1 Lignes de commande

Afin de tester rapidement ces services et ces topics pour les prendre en main, nous pouvons tout d'abord taper de simples lignes de commandes.

Par exemple, voici les commandes pour faire s'assoir Spot puis pour le relever grâce aux services `Spot/sit_down` et `Spot/stand_up` :

```
$ ros2 service call /Spot/sit_down webots_spot_msgs/srv/SpotMotion "{
    ↪ override: true}" # Commande pour que Spot s'asseye
$ ros2 service call /Spot/sit_down webots_spot_msgs/srv/SpotMotion "{
    ↪ override: true}" # Commande pour que Spot se leve
```

Ensuite, à partir des topics disponibles, nous pouvons également utiliser des publishers pour faire bouger Spot toujours grâce à de simples lignes de commande. Par exemple, grâce au topic `cmd_vel`, nous pouvons faire avancer et s'arrêter Spot. Les topics `/joint_states` et `/scan` sont également intéressants pour avoir accès à l'état des joints de Spot ou à la distance entre Spot et les objets qui se situent tout autour de lui. Voici les lignes de commande pour obtenir toutes ces informations :

```
$ ros2 topic pub /cmd_vel geometry_msgs/msg/Twist "linear:
     x: 0.5
```

```
      y: 0.0
      z: 0.0
    angular:
      x: 0.0
      y: 0.0
      z: 0.0" -r 5 # Commande pour faire avancer Spot
$ ros2 topic pub /cmd_vel geometry_msgs/msg/Twist "linear:
      x: 0.0
      y: 0.0
      z: 0.0
    angular:
      x: 0.0
      y: 0.0
      z: 0.0" -1 # Commande pour faire s'arreter Spot
$ ros2 topic echo /joint_states # Commande pour voir ce que publie /
    ↪ joint_states
$ ros2 topic echo /scan # Commande pour voir ce que publie /scan
```

6.2 Scripts Python

Pour mieux comprendre comment utiliser les services et les topics disponibles pour programmer les déplacements de Spot, j'ai ensuite écrit plusieurs scripts Python que j'ai intégrés à l'architecture ROS2 déjà présente grâce au fichier setup.py. Ces scripts sont disponibles sur mon git au lien suivant.

Les scripts `lie_down_service.py` et `stand_up_service.py` permettent par exemple de faire assoir Spot et de le relever.

```
$ ros2 run webots_spot lie_down_service # Commande pour lancer
    ↪ lie_down_service.py
$ ros2 run webots_spot stand_up_service # Commande pour lancer
    ↪ stand_up_service.py
```

Le script `send_cmd_vel.py` permet de faire avancer Spot et le script `move_sequence.py` permet de le faire avancer, reculer et tourner pendant quelques secondes.

```
$ ros2 run webots_spot publisher_vel # Commande pour lancer send_cmd_vel.py
$ ros2 run webots_spot publisher_test_move # Commande pour lancer
    ↪ move_sequence.py
```

Enfin, dans le script `scan_distance_front.py` nous calculons la distance à l'avant de Spot et la renvoyons en direct.

```
$ ros2 run webots_spot suscriber_distance_front # Commande pour lancer
    ↪ scan_distance_front.py
```

7 Ajout de scripts à l'architecture pour Well-E

Dans le cadre de la chaire Well-E, la scène proposée dans le package ne nous intéresse pas. A la place, j'ai créé une scène constituée de 10 vaches avec le fichier `spot2.wbt` dans

le dossier **worlds**. Pour charger cette scène en lançant Webots, il suffit de remplacer `spot.wbt` par `spot2.wbt` dans le fichier `spot_launch.py` situé dans le dossier **launch**.

7.1 Détection des vaches

Dans le dossier clôné, il existe un script `yolov8_openvivo.py` permettant de détecter les vaches en utilisant la caméra du bras de Spot.

```
$ ros2 run webots_spot cows_detector_webots # Commande pour lancer
    ↪ yolov8_openvivo.py
```

Or, nous voulions utiliser les modèles développés par Well-E. J'ai donc créé le script `yolov8_well_e.py` et ajouté un fichier `vache_classes.yaml` ne contenant que la classe vache à l'architecture ROS2 ainsi qu'un fichier `best_yolov8_detection_10.onnx` représentant le modèle YOLO entraîné par Well-E. Ce script est également présent dans mon git.

```
$ ros2 run webots_spot cows_detector_well_e # Commande pour lancer
    ↪ yolov8_well_e.py
```

Le modèle développé dans le cadre de la chaire est moins performant que celui initialement présent dans le package. Pour la simulation, il est donc plus intéressant de garder celui du package initial mais, pour être utilisable dans la réalité, le modèle de la chaire a été intégré à l'architecture.

7.2 Identification des vaches

Nous voulions ensuite intégrer l'algorithme d'identification des vaches à l'architecture ROS2 donc j'ai fait cela avec le script `detection_and_tracking.py` en ajoutant également le fichier `cows_names.yaml` contenant les noms des vaches et le modèle entraîné par Well-E : `efficientnet_b0_best_model_local_SNA_29_classes.pth`. Ce script permet de réaliser la détection des vaches grâce au modèle .onnx puis à identifier les vaches présentes sur chaque détection en extrayant les frames grâce au modèle d'identification .pth.

```
$ ros2 run webots_spot cows_detection_and_tracking # Commande pour lancer
    ↪ detection_and_tracking.py
```

7.3 Algorithmes de navigation

Nous avons ensuite développé des scripts de navigation pour Spot. Le premier s'appelle `suivi_vache_proche.py`. Il permet de faire en sorte que Spot suive la vache la plus proche de lui au début de la simulation, tout en évitant les autres vaches qui ont un mouvement aléatoire.

```
$ ros2 run webots_spot suivi_vache_proche # Commande pour lancer
    ↪ suivi_vache_proche.py
```

Puis, nous avons travaillé sur un autre code pour faire en sorte que Spot suive une vache spécifiée par l'utilisateur. Le problème est que ce code nécessitait une phase d'identification et que dans la simulation les vaches étaient toutes pareil. Nous avons donc créé une autre scène `spot2_simu.wbt` située dans le dossier **worlds**. Pour l'utiliser, il faut encore une fois indiquer le nom de cette scène dans `spot_launch.py` dans le dossier **launch**. Cette

scène ajoute deux boules de couleur sur chaque vache afin de les différencier. Nous avons alors développé un script de détection et d'identification de vaches pour la simulation dans cette scène nommé `detection_and_tracking_simu_yolo_entraine.py`.

```
$ ros2 run webots_spot cows_detection_and_tracking_simu_yolo # Commande
    ↪ pour lancer detection_and_tracking_simu_yolo_entraine.py
```

Enfin, le code `suivi_vache_specifique.py` permet de suivre la vache indiquée avec la ligne de commande suivante :

```
$ ros2 run webots_spot suivi_vache_specifique --ros-args -p target_color:=
    ↪ red # Commande pour lancer suivi_vache_specifique.py
```

Ce code est encore en développement car nous faisons face à des défis dus aux ressources computationnelles.