

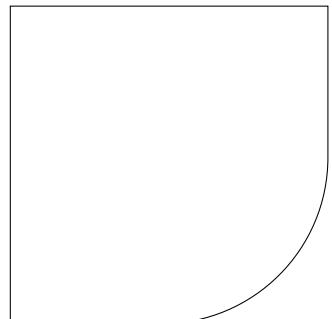
ENSTA Bretagne
2, rue François Verny
29806 BREST cedex
FRANCE
Tel +33 (0)2 98 34 88
00
[www.ensta-bretagne.
fr](http://www.ensta-bretagne.fr)



Simulation des
robots mobiles

UE 4.2

6 mai 2020



Simulation d'un robot –Warthog–



Colin Baumgard, Ludovic Diguet, Hamid Hacene,
Corentin Lemoine, Antonin Lizé

Table des matières

Introduction	1
1 Modélisation du robot	1
1.1 Choix techniques	1
1.2 Répartition du poids	2
1.3 Articulations des deux "bascules" portes roues	3
1.4 Interface avec ROS	4
1.5 Validation du modèle	4
2 Simulation du robot dans l'eau	5
2.1 Simulation d'une simple sphère	5
2.2 Application au robot Warthog	6
Conclusion	6

Introduction

Ce projet s'inscrit dans le cadre des enseignements de l'ENSTA BRETAGNE dispensés aux élèves de deuxième année de la filière *Robotique Autonome* au sein de l'U.E 4.2 : *Simulation des Robots Mobiles*.

Ce rapport présente le travail de notre équipe sur la simulation sur *V-REP* d'un robot : le robot *Warthog* fraîchement acheté par l'ENSTA BRETAGNE. Dans la suite du projet nous allons aborder différents aspects de la simulation : le choix de la modélisation des liaisons du robot, le choix de la répartition des masses, le franchissement d'obstacles qui est l'une des caractéristiques principales de ce robot et enfin son comportement et sa simulation dans l'eau.

1 Modélisation du robot

1.1 Choix techniques

Après avoir étudier la documentation présente sur internet à propos du robot *Warthog* et après visualisation de quelques vidéos, nous avons opté pour la modélisation

dynamique suivante :

- Les 4 roues seront modélisées avec 4 cylindres ;
- Les deux trains ainsi que le châssis principal seront modélisés par des parallélépipèdes ;
- Pour modéliser les actionneurs des roues, nous utiliserons 4 liaisons pivots simples ;
- La modélisation des deux articulation "bascules" est détaillée plus bas. [1.3](#)

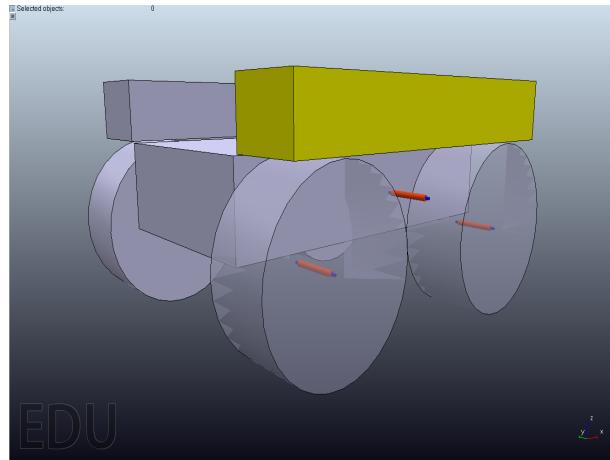


FIGURE 1 – Modèle dynamique

1.2 Répartition du poids

Dans cette partie nous allons aborder notre choix de répartition de la masse. En effet dans la documentation disponible sur le site du fournisseur, le robot a une masse de 250kg. Nous avons décidé la répartition suivante :



FIGURE 2 – Répartition des masses

1.3 Articulations des deux "bascules" portes roues

Dans un premier temps, nous avons modélisé cette articulation avec une liaison pivot en mode *Spring-damper mode*. Après plusieurs tests pour régler les deux coefficients k et C , le comportement de notre robot ne se rapprochait pas de ce que nous avons pu voir sur les vidéos de présentation du *Warthog*.

Nous avons donc décidé de changer d'approche. Le comportement de cette liaison pivot dans les vidéos montre qu'elle agit comme un vérin : elle essaie de maintenir le plus possible le châssis principal à l'horizontale. Quand l'environnement ne le permet pas (passage sur une roche par exemple), la liaison ramène progressivement le châssis à sa position d'équilibre.

Nous avons donc décidé de la modéliser par une liaison pivot simple avec les propriétés de la figure 3. En effet, un contrôleur *PD* se comporte comme un système ressort-amortisseur : il ramène notre système à la position d'équilibre.

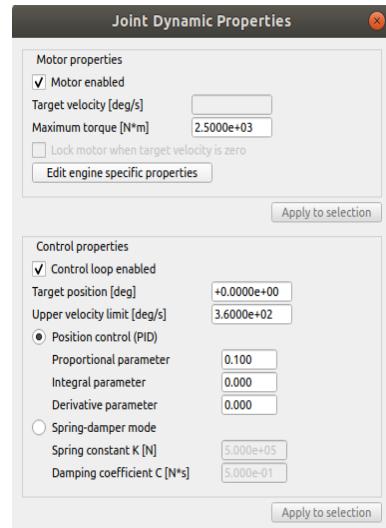


FIGURE 3 – Propriétés de l'articulation

1.4 Interface avec ROS

Pour communiquer et pouvoir tester le robot et sa modélisation, nous avons établi une interface avec ROS en *lua*. Cette interface permet de contrôler les quatre moteurs de notre robot.

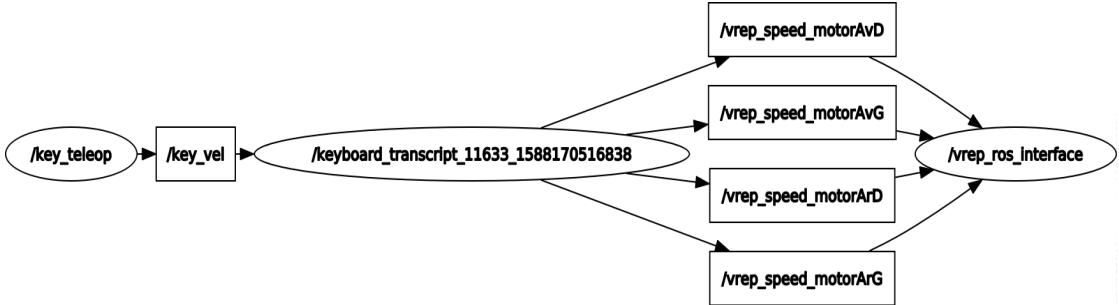


FIGURE 4 – Interface ROS

1.5 Validation du modèle

Afin de valider le modèle dynamique, nous avons construit une scène avec différents obstacles. Un script *python* est utilisé pour contrôler le robot avec les touches du clavier. Cette utilisation nous a permis de valider les différentes valeurs des coefficients du *PID*.

Une vidéo de la simulation en fonctionnement est disponible à cette [adresse](#)¹. Les nodes nécessaires pour contrôler le robot avec le clavier sont `kb_t` du package `keyboard_transcript` ainsi que `key_teleop` du package éponyme.



FIGURE 5 – Simulation et validation

1. <https://youtu.be/z00VfYflSLs>

2 Simulation du robot dans l'eau

2.1 Simulation d'une simple sphère

Afin de mettre en place une simulation simplifiée comme preuve de faisabilité, nous avons cherché à simuler une sphère dans l'eau.

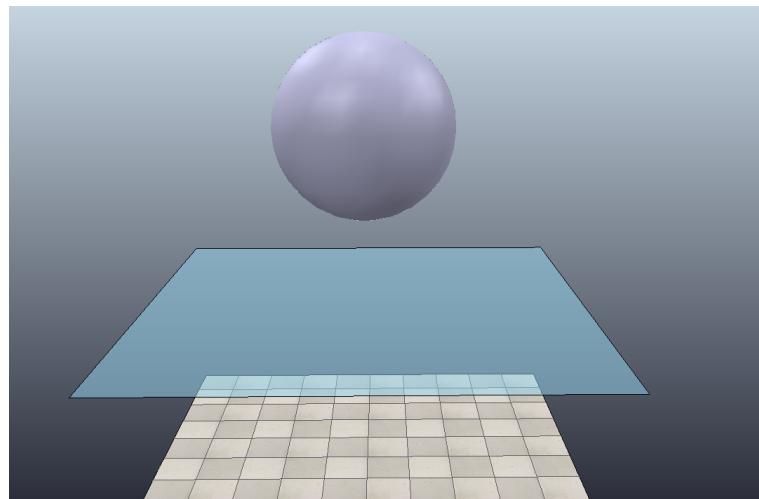


FIGURE 6 – Scène V-rep de la sphère au dessus d'un volume d'eau

Dans la figure ci-dessus, on voit la scène V-rep. Un script en lua fait ensuite le lien avec un node ROS codé en C++. Pour simuler l'action de l'eau sur la sphère, le script récupère la position de celle-ci. Un calcul de la poussée d'Archimède est ensuite effectué. On a besoin pour cela de la formule de la calotte sphérique, énoncé ci-dessous :

$$V = \frac{\pi h^2}{3} * 3(R - h)$$

Avec h la hauteur immergée et R le rayon de la sphère. On en définit ensuite facilement la poussée d'Archimède, que l'on applique au centre de la sphère (la force s'applique au fait au barycentre du volume immergé, mais comme celle-ci est co-axiale et passant par le centre de gravité, cette hypothèse est vérifiée puisqu'aucun moment n'est créé). Il ne reste alors qu'à renvoyer la valeur de cette force au script lua qui l'applique dans V-rep.

Nous obtenons alors une balle rebondissante. Il faut, pour une simulation plus réaliste, intégrer un coefficient de frottement. Pour cela, le code lua transmettra en plus de la position, la vitesse. Une première approche a été de calculer la vitesse en dérivant la position, mais le résultat est moins propre qu'avec la vitesse extraite

de V-rep, qui est donc ajoutée aux topics. Pour le coefficient de frottement, nous avons choisi un frottement linéaire et quadratique.

Nous obtenons un résultat tout à fait conforme à nos attentes. Une petite tricherie est toutefois nécessaire pour éviter la divergence de la simulation. On place une liaison glissière entre le sol et la sphère, en ne lui autorisant que la translation selon z, hypothèse vérifiée par la symétrie selon cet axe.

2.2 Application au robot Warthog

Pour la simulation du robot complet dans l'eau, nous avons implémenté la méthode qui consiste à appliquer une force sur chaque roue égale à la force d'Archimète résultante de leur partie immergée. Cette méthode fonctionne mais le résultat obtenu n'est pas le bon puisque le robot se retourne. Ce comportement est la conséquence du placement des points d'application des forces vis-à-vis du centre de gravité du robot. L'équilibre instable initial est perturbé par des erreurs de calculs liées à la simulation et le robot atteint un équilibre stable sur le dos.

Le meilleur moyen de corriger ces erreurs serait de faire des calculs de stabilité pour trouver le moment du couple de redressement ainsi que la poussée d'Archimète pour une grande plage d'attitudes du robot, et d'utiliser ces calculs dans des *Look Up Tables* pour notre simulation.

Conclusion

Pour conclure, ce projet nous a permis de mettre en pratique les premiers cours d'utilisation de *V-REP*. En associant ces connaissances ainsi que l'utilisation de ROS pour contrôler le robot pendant la simulation, nous avons réussi à modéliser simplement et rapidement un robot dans une situation donnée (*évitement d'obstacles, navigation dans l'eau*).

Table des figures

1	Modèle dynamique	2
2	Répartition des masses	3
3	Propriétés de l'articulation	3
4	Interface ROS	4
5	Simulation et validation	4
6	Scène V-rep de la sphère au dessus d'un volume d'eau	5