

Interaction Robot-sol : état de l'art et étude

Rédigé par AMBLARD Lucas, BERNAT Loan, GAUDILLAT Eliott, MISSIER Samuel, PHAN Quentin et SOUBIRANT Damien, étudiants en M1 à l'UPSSITECH, Systèmes Robotiques et Interactifs, dans le cadre du projet TER avec la société Agreenculture.

Résumé

Simuler en 3D un robot à chenilles est une tâche ardue à laquelle plusieurs chercheurs ont essayé d'apporter une solution. Nous tâcherons à travers ces pages de présenter un modèle mathématique et physique permettant de modéliser l'interaction chenille-sol. Nous utiliserons les différentes équations proposées par Becker, McKyes et Janosi. De nombreux outils permettent la mise en œuvre de notre modèle, et suite aux comparaisons de ces derniers et des travaux réalisés dessus, nous choisirons ceux qui nous paraissent les plus intéressants pour la suite du projet.

Mots-clés Robotique, Chenille, Modélisation, Interaction, Simulation

Introduction

La simulation 3D de robot permet de concevoir, simuler et tester des applications robotiques dans un environnement 3D indépendamment de la disponibilité et l'avancement réel de l'hardware du robot. Cela représente un gain de temps pour le développement et une économie financière non négligeable tout en évitant de possibles dégradations du matériel durant les phases de test. Il existe de nombreuses solutions logicielles et techniques pour modéliser et simuler un robot (plus ou moins complexes) sur un ordinateur. Dans le cadre qui nous intéresse, un des principaux enjeux de ces modélisations est de représenter l'interaction "Robot-sol" le plus fidèlement possible. Tout au long de cet ouvrage, nous nous concentrerons sur l'étude d'une possible modélisation d'un robot à chenilles. En effet, cette modélisation ne présente pas encore de solution

parfaitement aboutie. Nous verrons donc **comment est-il possible de modéliser un robot à chenilles en 3D afin de simuler son comportement de manière exhaustive ?**

Il semble important de définir, dans un premier temps, l'ensemble des relations mathématiques et physiques qui régissent une telle interaction. Nous présenterons ensuite des méthodes de modélisation réalisées par différents chercheurs. Pour chacune d'entre elles, nous présenterons les hypothèses, les méthodes de réalisation, les outils logiciels et les résultats obtenus. Nous terminerons par une comparaison afin de conclure sur quelle méthode et quel logiciel semblent les plus performants afin de l'appliquer au robot CEOL de la société Agreenculture avec laquelle nous travaillons.

I - Contexte mathématique

1 - Modèle général

L'interaction chenille-sol est soumise à différentes contraintes : cisaillement du sol lors du déplacement, l'effet plastique du sol, l'effet des dents de la chenille sur le sol ou encore les forces normales [1]. Avant de détailler ces équations, il est nécessaire de présenter la forme que prend une chenille et les propriétés qui en découlent. La Fig 1 représente la coupe verticale d'un maillon de chenille. D'après cette figure, on observe qu'une chenille possède 3 zones de contact avec le sol : la zone plane, la zone Grouser 1, et la zone Grouser 2.

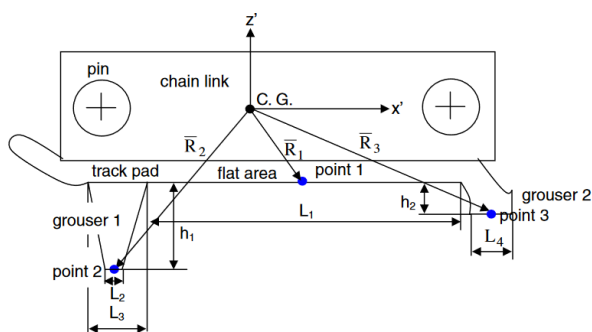


Fig 1 : Description schématique d'un maillon de chenille [1]

$x' y' z'$: Coordonnée du centre de masse du système.

$\vec{R}_1 \vec{R}_2 \vec{R}_3$: Vecteurs de localisation vers les points 1, 2 et 3, respectivement.

L_1 : Longueur de la surface plane.

L_2 : Longueur inférieure du cran 1.

L_3 : Longueur supérieure du cran 1.

L_4 : Longueur du cran 2.

$h_1 h_2$: Hauteur des crans 1 et 2, respectivement.

De manière plus générale, un chenillard est constitué de différents éléments : une roue motrice à l'avant, les galets porteurs permettant de transmettre le poids de la machine au sol, et

un galet de soutien pour soutenir le poids. Le chenillard est également soumis à plusieurs forces. La représentation de la Fig 2 correspond à un environnement particulier (évolution du chenillard sur un sol meuble), mais nous permet d'avoir un premier aperçu des forces globales agissant sur ce dernier.

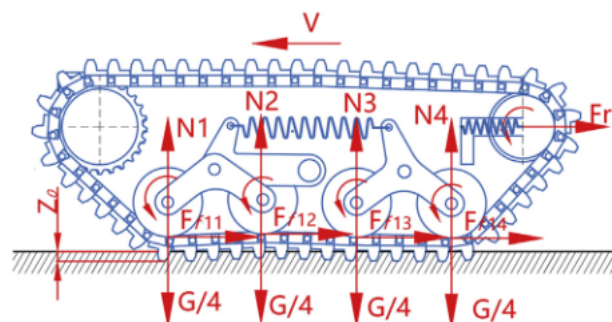


Fig 2 : Représentation globale des forces présentes [2]

G : Masse du robot

$F_{11}, F_{12}, F_{13}, F_{14}$: Friction entre la roue et la chenille

$N1, N2, N3, N4$: Réaction du sol sur les roues

Fr : Précharge du chenillard

Un modèle souvent évoqué est le modèle de Becker. Ce dernier prend en compte plusieurs conditions de l'environnement (sa porosité, sa composition...) et les différentes forces, afin de pouvoir étudier et prévoir le comportement du véhicule.

2 - Forces Normales

Suivant le modèle proposé par Becker [3] et en y ajoutant un élément de frottement visqueux, on peut décrire la relation entre la direction normale de la pression d'enfoncement et la vitesse d'enfoncement via l'équation (1.1.1) :

$$p_i = \begin{cases} ((\frac{k_c}{s_i} + k_\phi) D_i^n + C D_i) & \text{si } D_i > 0 \text{ et } p_i > 0 \\ 0 & \text{sinon.} \end{cases}$$

(1.1.1)

p_i : Pression au point i
 b : Largeur de la chenille,
 s_i : La plus petite taille de la chenille au point i
 C : Coefficient d'amortissement par unité de surface
 D_i (respectivement \bar{D}_i) : Distance (vitesse) d'enfoncement selon l'axe z'
 k_c : Module de cohésion de la déformation du sol
 k_ϕ : Module de frottement interne de la déformation du sol
 n : Indice de déformation du sol
 $i \in \{1, 2, 3\}$: Indice représentant l'un des 3 points de contact définie sur la Fig 1

On distingue donc 3 valeurs de s_i selon i :

$$s_i = \begin{cases} \min(L_1, b) & \text{si } i = 1 \\ \min(\frac{L_2+L_3}{2}, b) & \text{si } i = 2 \\ \min(L_4, b) & \text{si } i = 3 \end{cases} \quad (1.1.2)$$

Dans notre cas, l'enfoncement sera perpendiculaire à la surface du plan des maillons de la chenille. Lorsque le maillon pénètre dans le sol (**appelé phase de chargement**), les deux composantes de (1.1.1) sont dans la même direction. Opposément à cela, lorsque le maillon n'est plus en contact (**phase de déchargement**), les forces sont opposées. En conséquence, il y a un point où la pression sera nulle. Ce point, c'est justement le moment exact où le maillon et le sol ne sont plus en contact [1]. On représentera la force appliquée au point i par :

$$F_i = p_i A_i, \quad (1.1.3)$$

A_i étant la surface de contact au point i

3 - Forces de cisaillement

Maintenant que nous avons une meilleure idée de l'expression de la pression, il est nécessaire de prendre en compte les forces

de cisaillement. Elles se développent entre le sol et les maillons de notre chenille dans la direction longitudinale de ces derniers [1]. Lorsqu'un couple est appliqué au pneu ou au pignon d'une chenille, une action de cisaillement est initiée, comme le montre la Fig 3.

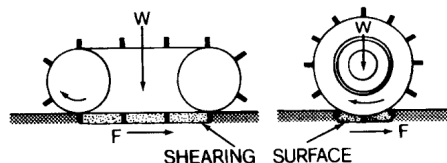


Fig 3 : Action de cisaillement d'une chenille et d'une roue.

Pour le sable meuble, l'argile saturée, la neige fraîche sèche et la plupart des sols répartis, la relation entre la contrainte de cisaillement et le déplacement en cisaillement présente les caractéristiques illustrées à la Fig 4.

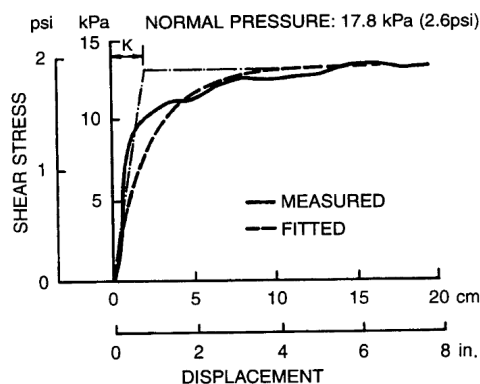


Fig 4 : Courbe de cisaillement de forme exponentielle

La relation entre la tension de cisaillement et son déplacement peut être décrite par une fonction exponentielle (1.2.1) selon Janosi.

$$\tau_i = (c + p_i \tan(\phi)) (1 - e^{-\frac{j_i}{k}}) \quad (1.2.1)$$

τ_i : Tension de cisaillement au point i .
 p_i : Pression au point i .
 j_i : Déplacement du cisaillement au point i .

c : La cohésion du sol.
 ϕ : L'angle de friction interne.
 k : Constante empirique.

On pourrait déduire de (1.2.1) que la tension de cisaillement augmente si le déplacement du cisaillement augmente aussi. Mais en réalité, à partir d'une certaine valeur de déplacement, la tension de cisaillement diminue [1], comme le montre la Fig 5.

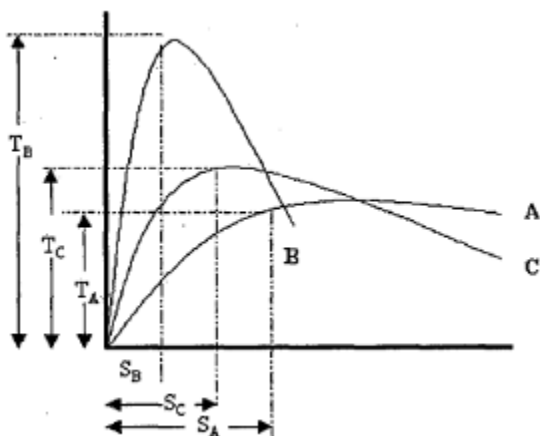


Fig 5 : Déplacement du sol sous cisaillement pour 3 types de sols [4]

A : Sol sans cohésion (Ex : argile humide)
B : Sol cohésif (Ex : argile cohésif)
C : Sol avec des propriétés intermédiaires
 T_A : Déplacement maximal du sol A.
 T_B : Déplacement maximal du sol B.
 T_C : Déplacement maximal du sol C.
 S_A : Moment où la tension de cisaillement du sol A n'augmente plus
 S_B : Moment où la tension de cisaillement du sol B n'augmente plus
 S_C : Moment où la tension de cisaillement du sol C n'augmente plus

Ce phénomène est dû à l'**effondrement du sol**. La tension maximale de cisaillement au point i peut donc s'exprimer :

$$\tau_{max_i} = \begin{cases} c + p_i \tan \phi & j_i < j_{max}, \\ (c + p_i \tan \phi) e^{\frac{j_{max} - j_i}{k_1}} & j_{max} \leq j_i \leq j_u, \\ (c + p_i \tan \phi) r & j_i > j_u, \end{cases} \quad (1.2.2)$$

τ_{max_i} : Tension maximale de cisaillement au point i .
 p_i : Pression au point i .
 j_i : Déplacement du cisaillement au point i .
 j_{max} : Déplacement du cisaillement à partir duquel la tension n'augmente plus.
 j_u : Déplacement du cisaillement à partir duquel la tension n'est plus affectée.
 c : La cohésion du sol.
 ϕ : L'angle de friction interne.
 r : Ratio de cisaillement maximal. $1 > r \geq 0$
 k_1 : Constante

Sachant que τ_{max_i} doit être une fonction continue, r doit vérifier :

$$r = e^{-\frac{j_{max} - j_u}{k_1}} \quad \text{Alors } k_1 = \frac{j_{max} - j_u}{\ln r}$$

La cohésion du matériau est le lien qui cimente les particules entre elles, indépendamment de la pression normale exercée par une particule sur l'autre. Ainsi, en théorie, la résistance au cisaillement de l'argile saturée et des matériaux similaires ne dépend pas de la charge normale, alors que la résistance au cisaillement du sable sec augmente avec l'accroissement de la charge normale. [5]

Pour le sable sec, la résistance au cisaillement peut donc être exprimée par :

$$\tau = \sigma \tan \phi \quad [5]$$

avec σ , la contrainte normale sur la surface cisailée. Dans notre cas, σ se ramène à p .

Pour l'argile saturée et les matériaux similaires, elle peut prendre la forme suivante :

$$\tau = c \quad [5]$$

La Fig 6 représente l'évolution de la tension de cisaillement τ^* (non-dimensionnelle)

en fonction du déplacement du cisaillement j^* (non-dimensionnel)

Avec:

$$\tau^* = \frac{\tau}{c + p \tan \phi}, \quad j^* = \frac{j}{k}, \quad j_{\max}^* = \frac{j_{\max}}{k} = 10,$$

$$j_u^* = \frac{j_u}{k} = 17, \quad r = 0.5.$$

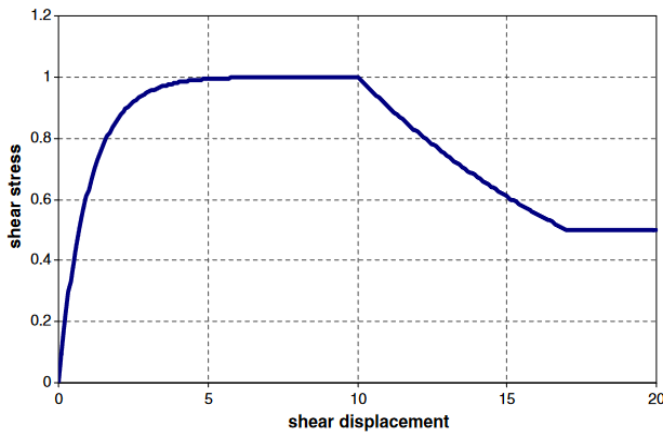


Fig 6 : Contrainte de cisaillement non dimensionnelle par rapport au déplacement de cisaillement non dimensionnel [1]

4 - Impact des maillons

Un point important que nous n'avons pas encore pris en compte est l'impact des maillons sur la force de traction. En réalité, cet impact est très similaire à celui d'une lame. On peut donc utiliser les modèles définis par McKyes . Cependant, le modèle bi-dimensionnel ne suffit pas à exprimer l'interaction de notre robot dans la direction latérale. On utilisera alors un modèle à trois dimensions pour exprimer l'interaction dans la direction latérale et le bi-dimensionnel pour exprimer en longitudinal :

$$P_i = (\gamma h^2 k_p + c h k_c + c_a h k_{ca} + q h k_q + \chi \rho h k_a v^2) b, \quad i = 2, 3 \quad (1.3.1)$$

avec $h = \min\{D_i, h_{i-1}\}$ l'enfoncement.

Nous ne considérons que les points 2 et 3 de la Fig 1, car le point 1 n'est pas situé sur un maillon. Pour définir l'équation (1.3.1) nous nous appuyons sur la Fig 1

P_i : Force du maillon au point i
 h_1, h_2 : Hauteur des crans 1 et 2, respectivement.
 b : largeur de la chenille (direction longitudinale) / $b_2 = (L_2 + L_3)/2$ et $b_3 = L_4$ (direction latérale)
 q : Force uniformément distribué $q = p_1 b$ (longitudinale) avec p_1 définie en (1.1.1), $q = 0$ (latérale)
 v : Vitesse
 k_p, k_c, k_{ca}, k_q et k_a : Constantes

Ici, il serait intéressant de décomposer (1.3.1) dans une direction verticale et dans une direction horizontale [1]. On obtient alors :

$$P_{Ni} = [P_i \sin(\alpha + \delta) + c_a h b \cot \alpha] \text{sign}(v) \left(1 - e^{-\frac{j}{k}}\right) \Psi_i,$$

$$P_{Vi} = [-P_i \cos(\alpha + \delta) + c_a h b] \left(1 - e^{-\frac{j}{k}}\right) \Psi_i, \quad (1.3.2)$$

avec :

$$\Psi_i = \begin{cases} 1 & j_i < j_{\max}, \\ e^{-\frac{j_{\max} - j_i}{k_i}} & j_{\max} \leq j_i \leq j_u, \\ r & j_i > j_u. \end{cases}$$

$j_i, j_{\max}, j_u, k, k_i$ et r sont expliqués en (1.2.2)

5 - L'effet plastique du sol

Un point important que nous n'avons pas encore pris en compte dans notre modèle est la déformation du sol. En fait, cette déformation peut être caractérisée par de petites déformations élastiques, de grandes déformations plastiques et des zones de rupture dans lesquelles les particules du sol se séparent [1].

Certains types de terrains, tels que l'argile saturée et le sable compact, qui couvrent une partie de la surface terrestre praticable, peuvent être comparés à un matériau élastoplastique idéal, présentant la relation contrainte-déformation illustrée à la Fig 7. Lorsque le niveau de contrainte dans le terrain ne dépasse pas une certaine limite, telle que celle indiquée par "a" sur la Fig 7, le terrain peut présenter un comportement élastique. [5]

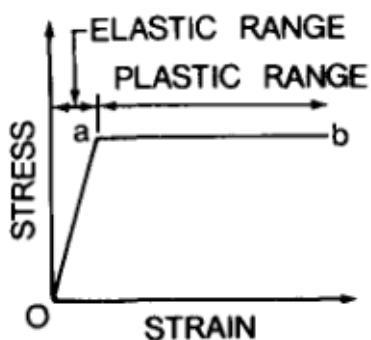


Fig 7 : Comportement d'un matériau élastoplastique idéalisé [5].

6 - Impact du sol sur les forces normales

Dans un premier temps, nous allons modifier l'expression de notre force normale obtenue en (1.1.1). La Fig 8 représente les courbes liées aux phases de chargement et aux phases de déchargement en un point i.

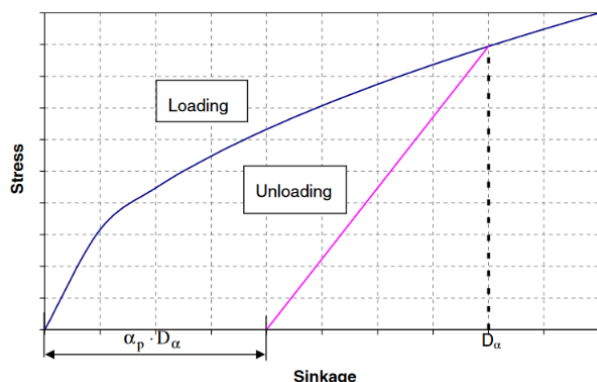


Fig 8 : Courbes de chargement et de déchargement pour notre modèle : Tension en fonction de l'enfoncement [1]

La tension du sol sous le chenillard augmente durant la phase de chargement jusqu'à un certain point maximal noté (D_α, p_α) et diminue lors de la phase de déchargement. Pour déterminer la plasticité du sol, on repère l'endroit où la courbe de déchargement et l'axe 0 de la tension se croisent. La valeur d'enfoncement vaut alors $\alpha_p \cdot D_\alpha$ et va nous permettre de caractériser la plasticité du sol ($0 \leq \alpha_p \leq 1$). Si $\alpha_p = 0$, alors la déformation n'est pas plastique (le sol revient à son enfoncement d'origine), si $\alpha_p = 1$ la déformation est purement plastique (le sol reste à l'enfoncement maximal subi). Enfin, en cas de nouveau passage sur le sol, la courbe de déchargement correspond à la courbe de recharge jusqu'à D_α . Après ce point, on réutilise la courbe de chargement pour définir la tension et un nouveau D_α . [1]

Basé sur les travaux de Dror Rubinstein et James L. Coppock, nous suivons l'hypothèse que la courbe de déchargement est linéaire et s'exprime donc de cette façon :

$$p = K(D - \alpha_p \cdot D_\alpha) \quad (1.4.1)$$

avec K dépendant des propriétés du sol :

$$K = \frac{\left(\frac{k_c}{b} + k_\phi\right) D_\alpha^{n-1}}{1 - \alpha_p}$$

On obtient alors l'expression finale de la pression normale basé sur (1.1.1) et (1.4.1) :

$$p = \begin{cases} \left(\frac{k_c}{b} + k_\phi\right) D^n + C\dot{D} & \text{si } D > D_\alpha \text{ et } p > 0 \\ K(D - \alpha_p \cdot D_\alpha) + C\dot{D} & \text{si } D_\alpha > D > 0 \text{ et } p > 0 \\ 0 & \text{sinon.} \end{cases}$$

7 - Impact du sol sur les forces de cisaillements

En seconde partie, il faut modifier les forces de cisaillement. Un infime dépassement du point "a" de la Fig 7 va produire une

augmentation rapide de la déformation, ce qui constitue un écoulement plastique. L'état suivant de l'écoulement plastique est généralement appelé équilibre plastique. Le passage de l'état d'équilibre à celui d'écoulement représente la rupture de la masse.

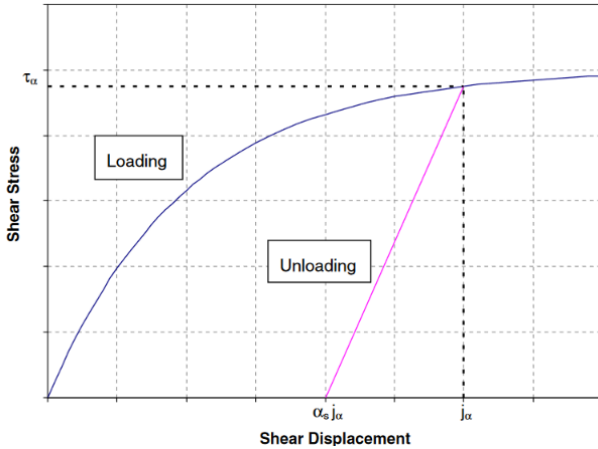


Fig 9 : Courbes de chargement et de déchargement pour notre modèle : Tension du cisaillement en fonction du déplacement du cisaillement [1]

Nous appuyant sur nos équations (1.2.1) et (1.2.2) ainsi que la Fig 9. Dror Rubinstein et James L. Coppock définissent le point maximal du déplacement du cisaillement en (τ_α, j_α) avec :

$$\tau_\alpha = \tau_{\max} \left(1 - e^{-\frac{j}{k}} \right) \quad (1.4.2)$$

Similairement à l'étude réalisée sur les forces normales via la Fig 8. La plasticité du sol se définit via l'intersection de la courbe de déchargement et l'axe 0 de la tension du cisaillement, où la valeur du déplacement est de $\alpha_s \cdot j_\alpha$ avec $0 \leq \alpha_s \leq 1$. Si $\alpha_s = 0$, alors la déformation n'est pas plastique, si $\alpha_s = 1$ La déformation est purement plastique.

$$\tau = \tau_\alpha \frac{j - \alpha_s j_\alpha}{j_\alpha (1 - \alpha_s)} \quad (1.4.3)$$

Combinant l'expression de la tension le long de la courbe de déchargement (1.4.3) et

celle le long de la courbe de chargement (1.4.2), on obtient alors (1.4.4) :

$$\tau = \begin{cases} \tau_{\max} (1 - e^{-\frac{j}{k}}) & \text{si } j \geq j_\alpha \\ \tau_\alpha \frac{j - \alpha_s j_\alpha}{j_\alpha (1 - \alpha_s)} & \text{si } j_\alpha > j > \alpha_s j_\alpha \end{cases} \quad (1.4.4)$$

Les deux hypothèses importantes de ce modèle sont que :

- lorsque $j = \alpha_s \cdot j_\alpha$, les valeurs de j et de j_α sont mises à 0.
- Les valeurs de j et de j_α sont positives. (Valeur absolue)

La direction de la force de cisaillement dépend donc de la direction de j tel que :

$$F_s = \begin{cases} -\tau A & \text{si } j > 0 \\ \tau A & \text{si } j < 0 \\ 0 & \text{si } j = 0 \end{cases} \quad (1.4.5)$$

Avec A la surface de contact.

8 - Impact du sol sur les maillons

Enfin, il ne reste plus qu'à modifier les équations (1.3.2) afin de prendre en compte la déformation du sol. Pour ce faire, on va multiplier nos équations par une fonction Ω représentant l'effet réaction de la force, la direction de la force et la plasticité du sol ($-1 \leq \Omega \leq 1$). Une description de Ω est donné en Fig 10. Lors de la courbe de chargement, on note que $|\Omega|$ augmente si $|j|$ augmente. Si $j > 0$, on représente le minimum du déplacement du cisaillement par j_β et si $j < 0$, j_γ . On désigne Ω_β (respectivement Ω_γ) la valeur de Ω à j_β (j_γ).

$$\Omega_\beta = e^{\frac{j_\beta}{k}} - 1,$$

$$\Omega_\gamma = 1 - e^{\frac{j_\gamma}{k}}.$$

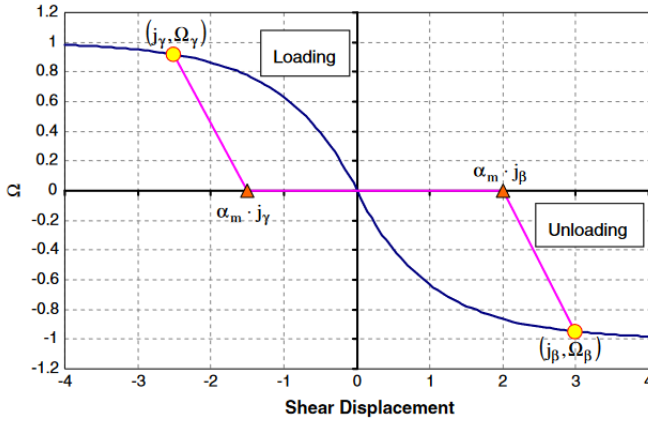


Fig 10 : Ω en fonction du déplacement du cisaillement [1]

Lors du premier contact entre le chenillard et le sol, les valeurs j_β et j_γ sont mises à 0. A partir de ce point, la valeur absolue de j augmente jusqu'à j_β ou j_γ et la valeur de Ω se détermine le long de la courbe de chargement. Lorsque $j_\beta > j > j_\gamma$, Ω se détermine le long de la courbe de déchargement.

Le paramètre α_m décrit la plasticité du sol ($0 \leq \alpha_m \leq 1$).

$$\Omega = \begin{cases} e^{-\frac{j}{k}} - 1 & \text{si } j \geq j_\beta \\ 1 - e^{\frac{j}{k}} & \text{si } j \leq j_\gamma \\ \frac{\Omega_\beta}{1-\alpha_m} \left(\frac{j}{j_\beta} - \alpha_m \right) & \text{si } \alpha_m \cdot j_\beta < j < j_\beta \\ \frac{\Omega_\gamma}{1-\alpha_m} \left(\frac{j}{j_\gamma} - \alpha_m \right) & \text{si } \alpha_m \cdot j_\gamma > j > j_\gamma \\ 0 & \text{si } \alpha_m \cdot j_\gamma \leq j \leq \alpha_m \cdot j_\beta \end{cases}$$

(1.4.6)

9 - Les paramètres du sol

Pour déterminer les paramètres du sol, on se base sur les travaux de Dror Rubinstein et James L. Coppock [1].

$$\alpha_s = 1 - \frac{k}{j_\alpha} \left(1 - e^{-\frac{j_\alpha}{k}} \right). \quad (1.4.7)$$

$$\alpha_m = \begin{cases} 1 - \frac{k}{j_\beta} - \left(1 - e^{\frac{j_\beta}{k}} \right) & \text{si } j > 0 \\ 1 + \frac{k}{j_\gamma} - \left(1 - e^{\frac{j_\gamma}{k}} \right) & \text{si } j < 0 \end{cases}$$

(1.4.8)

II - Modélisation et Simulation

La simulation physique est un domaine complexe mais également un outil de test remarquable. Cependant, la simulation d'un chenillard n'a pas encore de solution générale que l'on peut importer pour tout type de robot à chenilles. Nous avons présenté en (1) un modèle mathématique pour définir l'interaction entre le chenillard et le sol, mais la modélisation d'un robot à chenilles demande également d'autres études liées au robot directement, telles que la flexibilité et la déformabilité de la chenille. Une pratique répandue à l'heure actuelle et de remplacer les chenilles du robot par des roues lors de la simulation. Mais cette méthode peut présenter des soucis de fiabilité : les résultats expérimentaux peuvent alors différer. Dans cette partie nous allons présenter les différents travaux de chercheurs qui ont tenté de modéliser le plus fidèlement possible un robot à chenilles.

1 - Dror Rubinstein et James L. Coppock

Suivant le modèle mathématique présenté en (1), Rubinstein et L. Coppock, utilisent un **programme multi-corps** : Ils ajoutent alors, à la modélisation de ce modèle, le programme LMS-DADS. Ce programme leur permet alors de créer un modèle détaillé d'une chenille en modifiant uniquement 3 sous-routines [1] :

- **frcdf** - Interface pour les forces définies par l'utilisateur.
- **inudf** - Lire les entrées définies par l'utilisateur.
- **urstop** - Utilisé pour stopper le programme en cas d'erreur.

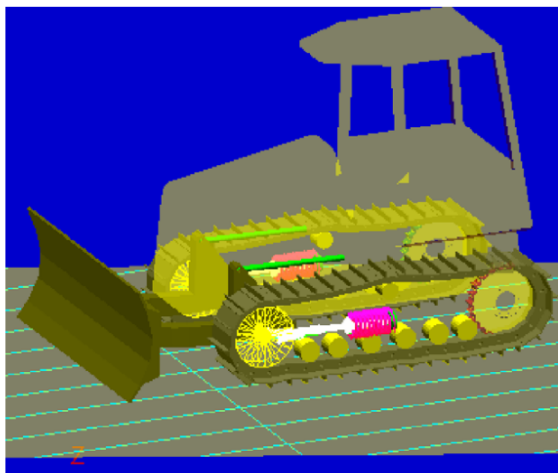


Fig 11 : Schéma du programme multi-corps utilisé par Rubinstein et L. Coppock [1]

Ils ont alors comparé les résultats obtenus via la simulation avec une expérimentation sur 2 types de sol différents à 3 vitesses différentes et estiment que les résultats corrèlent raisonnablement bien avec les données expérimentales. [1] Malheureusement, il nous est impossible d'accéder à cet outil.

2 - Mise en place sur Ros/Gazebo

Pour essayer de pallier ce problème, des modèles simplifiés utilisant des roues sont utilisés pour essayer de modéliser la physique du chenillard [6], comme le montre la Fig 12

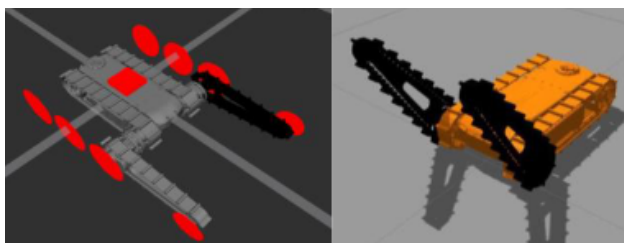


Fig 12 : Utilisation de roues imaginaire pour modéliser le chenillard [6]

Cette implémentation permet d'utiliser des modèles déjà existants pouvant être appliqués sur des roues (contrôle de rotation angulaire et de vitesse par exemple). Cela dit, le problème est qu'il y a quand même une différence entre les interactions et les

réactions régies par des roues ou bien par des mécanismes à chenilles.

Un autre exemple de mise en place sous ROS et Gazebo est celui de Sokolov et al. [7], dans le cadre de la modélisation d'un robot à chenilles, présenté ci-dessous en Fig 13.



Fig 13 : Chenillard 'Servolisa' type UGV [7]

Le simulateur Gazebo, intégré à ROS leur a permis de modéliser le robot en 7 parties distinctes. Les 7 modèles, créés en CAO, ont permis d'implémenter les caractéristiques physiques du robot. L'assemblage de ces différents modèles a permis la modélisation du robot. Les différentes parties sont liées par des joints ROS prenant en compte les limites physiques du robot réel.

C'est dans la simulation des chenilles que réside la principale difficulté. Il n'existe pas réellement de modèle robuste dans ROS pour simuler des chenilles et leurs particularités physiques particulières.

La première idée de Sokolov et al. a été de contourner le problème en approximant les chenilles à de nombreuses roues de part et d'autre du robot. Augmenter leur nombre et réduire leur taille permet de s'approcher des propriétés physiques d'une chenille. Toutefois cette modélisation est simplifiée, donc moins réaliste qu'une véritable modélisation de chenilles, qui prendrait en compte les phénomènes physiques de contact chenille-sol.

Pour aller plus loin, les chercheurs ont utilisé le seul package ROS disponible à l'époque pour modéliser des chenilles : *Gazebo-tracks*. Ce package permet de définir des robots, des objets statiques, le terrain et la physique liée à chacun de ces éléments. Toutefois, le principal défaut de ce package est qu'il ne permet que de modéliser des systèmes à chenilles de deux galets, de même rayon (voir Fig 14). Ce package est donc inapplicable pour des chenillards plus complexes.

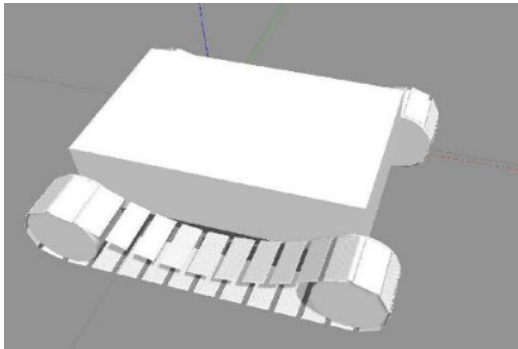


Fig 14 : Type de chenillard modélisable grâce au package *Gazebo-tracks* [7]

Malgré des paramètres assez complets : prise en compte de la distance entre les galets, entre les chenilles, leur largeur et épaisseur et le nombre de mailles, le package souffre d'autres défauts.

L'utilisation du format SDF permet de relier et de connecter les chenilles modélisées au corps du robot. Cette solution (car le format initial URDF ne permettait pas cette connexion des chenilles) présente toutefois certains inconvénients. En effet, d'après les études menées par Sokolov et al. sur leur modèle, les chenilles ne sont pas stables et finissent par se désolidariser du robot à cause des forces d'inertie et du décalage chenille-galets engendré. De plus comme évoqué précédemment, le package ne permet que de modéliser 2 galets, et ceux-ci ne peuvent être synchronisés avec l'échange de messages ROS (problème d'API).

La modélisation sur ROS de chenillards semble donc être délicate, certaines solutions approximatives existent, comme remplacer les chenilles par un certain nombre de roues, pour approcher leurs propriétés physiques. Les package existant souffrent de lacunes et d'inconvénient ce qui réduit drastiquement leur champ d'application.

3 - Mise en place sur Ros/Webots

Basé sur le même robot que la partie précédente, c'est-à-dire le chenillard Servolisa, une étude dans un environnement de simulation différent à été réalisée. Ce dernier est Webots, un simulateur Open-Source de robotique utilisé dans l'industrie ou bien la recherche, qui supporte ROS et permet donc d'utiliser ses services classiques.

Webots présente aussi un avantage face à Gazebo. La construction d'un modèle de robot dans Gazebo passe par la création et la complétion d'un fichier de configuration au format URDF. Dans Webots, cette construction se fait directement via l'interface graphique, où il est possible de déplacer manuellement des parties du modèle et de les connecter. Cette fonctionnalité est bien plus intuitive et simple d'utilisation que le fichier URDF.

De plus, le plus gros problème de l'utilisation du modèle Gazebo proposé était causé par un facteur temps réel (RTF) de la simulation. Le RTF représente une corrélation du temps en simulation au temps réel. Par exemple, si dans une simulation, il faut dix secondes pour calculer une action d'une seconde (monde réel), le RTF est de 0,1. Il est bon de savoir que plus la valeur du RTF est faible, moins efficace et moins rapide est la simulation.

Comme évoqué précédemment, une des difficultés pour la simulation de robot à chenilles est la représentation de la chenille. Un nombre élevé de roues pour se rapprocher de son fonctionnement, est la cause d'une mauvaise valeur du RTF. Webots permet

d'apporter une solution sur le nombre de roues nécessaires. La Fig 15 permet de comparer les modèles de Webots et celui de Gazebo.

Parameter	Gazebo	Webots
Number of elements	221	11
Real time factor (RTF)	0.1 ~ 0.2	0.95~1

Fig 15 : Comparaison entre le modèle Gazebo et le modèle Webots

Pour le même robot et la même mise en situation, la représentation sous Webots permet de diminuer le nombre de roues et donc d'avoir une meilleure valeur du RTF. Cela dit, malgré le fait que pour des conditions similaires Webots ait de meilleurs résultats sur cette partie, ce n'est pas pour autant que Webots permet de résoudre le problème de la représentation d'un chenillard en simulation.

4 - Mise en place sur Project Chrono

Project Chrono est un moteur graphique open-source, initialement en C++ mais qui existe en python (PyChrono), qui permet la simulation 3D de modèles très variés comme par exemple la simulation de véhicules à roues et à chenilles évoluant sur des terrains déformables.

Ce moteur de simulation multi-physique est utilisé dans de nombreux domaines comme la robotique, la mécatronique, la réalité virtuelle et augmentée, la terra-mécanique, la détection de collisions, l'étude de la sismologie et des flux granulaires mais aussi et surtout : l'étude des comportements de véhicules variés.

Grâce à la dynamique multicorps proposée par Project Chrono, on peut :

- Exécuter des simulations de mécanismes constitués de corps rigides.
- Appliquez des contraintes aux pièces en utilisant un large éventail d'articulations.

- Ajoutez des moteurs, des actionneurs linéaires, des ressorts et des amortisseurs.
- Appliquez des forces et des couples.
- Calculez les forces de contact par friction à l'aide d'algorithmes de détection de collision de pointe.
- Utilisez différents modèles de sols

Côté modélisation, comme le véhicule à roues, un véhicule à chenilles est défini comme une hiérarchie de sous-systèmes.

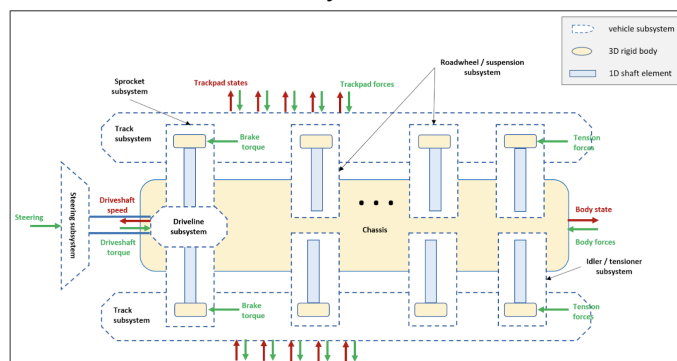


Fig 16 : Hiérarchie des systèmes du véhicule à chenilles

Malheureusement, une seule topologie de véhicules à chenilles est prise en charge. Elle comprend, au niveau supérieur, un sous-système de châssis, la chaîne cinématique du véhicule, un mécanisme de direction et deux sous-systèmes de chenilles. Ces derniers sont des conteneurs d'autres sous-systèmes et chacun comprend un mécanisme de pignon, un sous-système de tendeur de roue folle, un nombre arbitraire de composants de suspension et un nombre arbitraire d'ensembles de patins de chenille.

Chaque sous-système du véhicule à chenilles doit alors être précisément défini pour pouvoir simuler un comportement identique au système réel. Voici ci-dessous le diagramme de collaboration représentant les sous-systèmes pour les véhicules à chenilles.

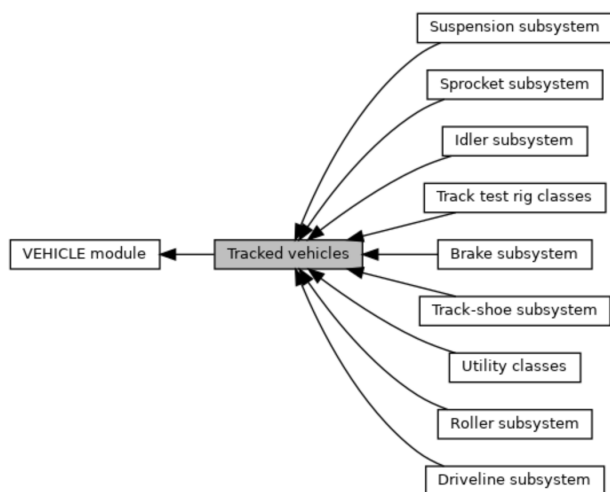


Fig 17 : Diagramme de collaboration pour les véhicules à chenilles

Project Chrono est donc un moteur parfaitement capable de simuler un véhicule à chenilles et son interaction avec différents types de sol dans la mesure où notre système correspond à la topologie prise en charge par ce dernier.

5 - Mise en place sur CoppeliaSim

Le 4ème logiciel de simulation sur lequel nous nous sommes penché est CoppeliaSim (connu comme V-Rep par le passé) de Coppelia Robotics.

Tout d'abord ce logiciel propose certains avantages:

- cross-platform et portabilité entre Windows/ Mac/ Linux
- prise en charge de différents langages de programmation (C/C++/Python)
- 5 moteurs physiques différents
- Enregistrement et visualisation des données

CoppeliaSim offre un large éventail de fonctionnalités qui le distingue des autres environnements de simulation. Une table de comparaison avec gazebo est disponible dans le document [8] aux pages 358 à 361.

Cependant l'un des inconvénients de CoppeliaSim est que le logiciel est très demandant en ressources. CoppeliaSim

nécessite de plus une modélisation du robot plus complexe et complète pour fonctionner correctement.

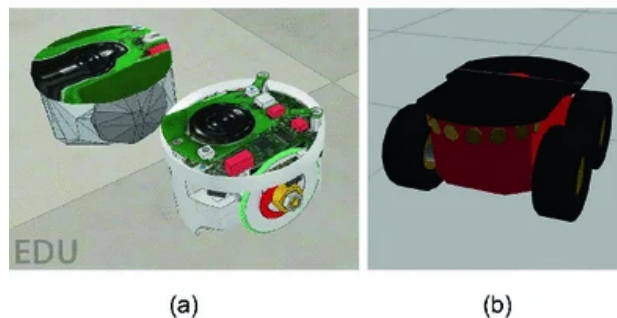


Fig 18 : (a) e-puck modèle sur CoppeliaSim (b) The Pioneer 3AT modèle de Gazebo

Enfin, la modélisation du terrain est très avancée sur ce logiciel. En effet, des terrains complexes peuvent être modélisés : neige, sable, boue, pentus... C'est pourquoi CoppeliaSim a été préféré pour simuler de la planification de robot à chenilles, lorsque le terrain possède des propriétés physiques particulières [9].

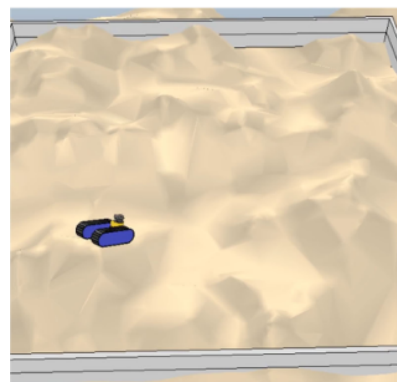


Fig 19 : Robot à chenilles sur un terrain extrême sous CoppeliaSim

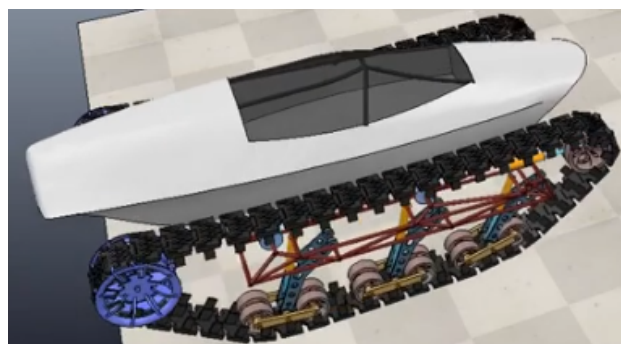


Fig 20 : Robot à chenilles sous CoppeliaSim

Il est possible de simuler des chenillards assez élaborés, comme en Fig 11, CoppeliaSim est donc un outil performant et complet, disposant de nombreuses fonctionnalités utiles à notre modèle. C'est une piste non écartable, nonobstant des demandes sérieuses en ressources et en complexité de modélisation.

6 - Mise en place sur Unreal Engine/Unity

La mise en place sur Unreal Engine ou Unity est une solution relativement "simple" et esthétique. En effet, ces moteurs initialement utilisés pour les jeux vidéos, disposent déjà de leur propre gestion de la physique (gravité, friction). Cependant, il n'y est pas possible de définir des paramètres de sols extrêmement précis comme nous le souhaiterions pour notre modèle.

III - Conclusion

Dans le but de pouvoir simuler un robot à chenilles, le travail se divise en deux parties : Dans un premier temps il faut être conscient et comprendre les différentes interactions entre ce type de véhicule et le sol. Par la suite, il faut être capable de pouvoir implémenter notre modèle dans un environnement de simulation adapté.

Comme évoqué dans notre étude, ce dernier point reste encore un des problèmes majeurs pour réaliser une simulation la plus proche possible de la réalité et de nos attentes. Au vu des différents travaux et de nos observations, nous allons nous orienter vers les outils suivants pour réaliser nos tests : Project Chrono, CoppeliaSim (V-rep) mais aussi Unreal Engine.

IV - Références

- [1] D. Rubinstein et J. L. Coppock, « A detailed single-link track model for multi-body dynamic simulation of crawlers », *J. Terramechanics*, vol. 44, n° 5, p. 355-364, nov. 2007, doi: 10.1016/j.jterra.2007.10.004.
- [2] Z. Li, L. Chen, Q. Zheng, X. Dou, et L. Yang, « Control of a path following caterpillar robot based on a sliding mode variable structure algorithm », *Biosyst. Eng.*, vol. 186, p. 293-306, oct. 2019, doi: 10.1016/j.biosystemseng.2019.07.004.
- [3] M. G. Bekker et M. G. Bekker, « Mechanics of Off-the-Road Locomotion », *Proc. Inst. Mech. Eng. Automob. Div.*, vol. 16, n° 1, p. 25-44, janv. 1962, doi: 10.1243/PIME_AUTO_1962_000_009_02.
- [4] S. Laughery, G. Gerhart, et R. Goetz, « Bekker's Terramechanics Model for Off-Road Vehicle Research: », Defense Technical Information Center, Fort Belvoir, VA, janv. 1990. doi: 10.21236/ADA457955.
- [5] J. Y. Wong, *Theory of ground vehicles*, 3rd ed. New York: John Wiley, 2001.
- [6] M. Sokolov, R. Lavrenov, A. Gabdullin, I. Afanasyev, et E. Magid, « 3D modelling and simulation of a crawler robot in ROS/Gazebo », in *Proceedings of the 4th International Conference on Control, Mechatronics and Automation - ICCMA '16*, Barcelona, Spain, 2016, p. 61-65. doi: 10.1145/3029610.3029641.
- [7] M. Sokolov, I. Afanasyev, R. Lavrenov, A. Sagitov, L. Sabirova, et E. Magid, « Modelling a crawler-type UGV for urban search and rescue in Gazebo environment », *Proc. Int. Conf. Artif. Life Robot.*, vol. 22, p. 360-363, janv. 2017, doi: 10.5954/ICAROB.2017.GS2-4.
- [8] M. Giuliani, T. Assaf, et M. E. Giannaccini, Éd., *Towards Autonomous Robotic Systems: 19th Annual Conference, TAROS 2018, Bristol, UK July 25-27, 2018, Proceedings*, vol. 10965. Cham: Springer International Publishing, 2018. doi: 10.1007/978-3-319-96728-8.
- [9] B. Sebastian et P. Ben-Tzvi, « Physics Based Path Planning for Autonomous Tracked Vehicle in Challenging Terrain », *J. Intell. Robot. Syst.*, vol. 95, n° 2, p. 511-526, août 2019, doi: 10.1007/s10846-018-0851-3