

Chemins de Dubins et Reeds-Shepp

Benjamin Loison

Janvier 2022

Dans cette présentation je vais vous présenter les chemins de Dubins et Reeds-Shepp.

SLIDE

Dans un premier tant nous verrons les chemins de Dubins, puis certains détails d'implémentation de ce type d'algorithme, puis nous étudierons les chemins de Reeds-Shepp et les différentes simplifications possibles. Je vais maintenant vous introduire les chemins de Dubins et Reeds-Shepp avec des exemples pour commencer :

SLIDE

Par exemple, si je vous demande, imaginons qu'au point de départ vert ce soit un être humain qui souhaite se rendre au point d'arrivée rouge, quel est un plus court chemin, vous allez me répondre qu'il suffit d'aller en ligne de droite du point vert au point rouge.

SLIDE

D'accord, mais en fait, en réalité l'être humain est orienté vers le nord au point de départ vert et on souhaite qu'il arrive orienté vers l'est au point d'arrivée rouge. Si je vous pose la même question : quel est un chemin le plus court entre ces deux positions orientées, en supposant que tourner sur soi-même est instantané, vous allez toujours me répondre que le problème est trivial et qu'il suffit de tourner au point de départ vert en direction du point d'arrivée rouge puis de marcher tout droit jusqu'au point d'arrivée rouge et de s'orienter vers l'est.

D'accord, mais en fait, on va considérer un robot qui ne peut avancer que vers l'avant au lieu de l'être humain, on reste sur le même problème mais cette fois-ci physiquement on se rend compte que le robot ne peut pas tourner sur lui-même en restant sur place.

SLIDE

C'est-à-dire lorsqu'on dit au robot de braquer au maximum à gauche, on observe un rayon de braquage, de quelques mètres ici. Avec cette contrainte de ne pouvoir soit qu'avancer, soit que tourner à gauche au maximum, soit que tourner à droite au maximum, je vous repose la même question : quel est un plus court chemin pour ce robot entre ces deux positions orientées ?

SLIDE

Alors là le problème n'est plus trivial dans le cas général, bien que sur cet exemple, on pressent bien le fait que ceci est un plus court chemin. Ce plus court chemin traité dans le cas du robot est appelé chemin de Dubins car ce problème, de trouver un plus court chemin entre deux positions orientées pour un robot

qui ne peut qu'aller en marche avant ou tourner à gauche au maximum ou tourner à droite au maximum avec un rayon de braquage non nul, a été résolu par Lester Dubins en 1957.

SLIDE

Effectivement dans le cas général où les deux positions orientées sont quelconques, il a démontré qu'un plus court chemin était forcément de la forme CSC ou CCC. Avec C signifiant "curve" (c'est-à-dire tourner à gauche ou à droite) et S signifiant "straight" (c'est-à-dire aller tout droit). De même R signifie "right" (c'est-à-dire tourner à droite) et L signifie "left" (c'est-à-dire tourner à gauche).

SLIDE

Pour revenir à l'exemple précédent, le chemin de Dubins trouvé était un LSR c'est-à-dire qu'au début le robot tourne à gauche, puis va tout droit puis tourne à droite.

SLIDE

Quant à l'implémentation, j'ai essayé d'implémenter les chemins de Dubins en Python avec la librairie graphique Qt cependant prendre en main cette librairie sous Python a été assez chronophage car j'ai cherché, en vain, pendant plusieurs heures comment proprement faire tourner une image de voiture bien que du point de vue théorique ce problème est trivial. Donc je tiens à préciser que les illustrations vues dans cette première partie ne sont pas originales de moi bien que j'ai effectué quelques photo montages pour faciliter la compréhension.

SLIDE

Il est intéressant de remarquer que l'algorithme consistant à trouver un plus court chemin, en utilisant l'approche de Dubins, a une complexité constante car cet algorithme consiste à calculer les 6 trajectoires possibles et à déterminer la plus courte.

SLIDE

Bien que la complexité soit constante, de même pour l'algorithme qu'on va voir d'ici quelques instants, il peut être intéressant d'éviter de faire certains calculs lourds impliquant notamment des fonctions trigonométriques en se basant sur un tableau d'angles qui indiquent dans quels bornes et seulement dans celles-ci les différents chemins sont optimisés.

SLIDE

Les différents points d'intérêt où le comportement du robot change, c'est-à-dire passe de tourner à droite à aller tout droit par exemple, se déterminent avec les tangentes intérieures et extérieures aux deux cercles définis par les positions orientées de départ et d'arrivée. Effectivement dans le cas où les deux cercles sont extérieurs l'un de l'autre, il existe en général quatre droites distinctes qui sont tangentes aux deux cercles. Pour deux d'entre elles, les tangentes extérieures, ici en jaune, les deux cercles sont du même côté de la tangente, pour les deux autres, ici en bleu clair, les tangentes intérieures, les cercles sont de chaque côté. Une fois ces tangentes tracés on redevine vite le chemin le plus court obtenu. Je ne vais pas détailler comment on obtient ces tangentes mais cela se fait par de simples calculs.

SLIDE

De plus une fois la séquence de mouvements que le robot doit effectuer est calculée, on peut se servir des

équations différentielles suivantes pour obtenir les coordonnées du robot à un moment donné. On peut soit intégrer avec la méthode d'Euler ou on peut faire un calcul plus précis en considérant le point débutant l'arc de cercle ou le segment et calculer où en est le robot sur cette portion du chemin. De plus si l'on utilise la première méthode on peut faire dépendre la taille du pas δ en fonction de la distance du chemin à parcourir.

SLIDE

Bon alors si on revient au problème initial, d'un robot qui cherche un plus court chemin d'une position orientée de départ à une position orientée d'arrivée, en restreignant les mouvements de ce robot à aller tout droit, tourner à gauche et tourner à droite avec un rayon de braquage non nul. Si cette fois-ci on ajoute la possibilité au robot de pouvoir reculer tout, reculer à gauche et reculer à droite et que je vous pose la même question : quel est un chemin le plus court, là il y a de quoi se donner des mots de tête.

SLIDE

Bon la réponse la voici. Le robot va dans un premier tant à reculons tourner à gauche, puis aller en ligne droite en arrière puis tourner légèrement à droite en arrière. Sachant que bien évidemment qu'un chemin le plus court de Reeds-Shepp est au moins aussi court qu'un chemin le plus court de Dubins.

SLIDE

Dans l'article de J. A. Reeds et L. A. Shepp publié en 1990, ils résolvent ce problème du robot aussi capable d'aller en marche arrière dans le cas général. Contrairement aux chemins de Dubins où il n'y avait que 6 possibilités, les chemins de Reeds et Shepp comportent 48 possibilités différentes. De plus ici les indices indiquent la longueur de l'arc de cercle ou la longueur du segment impliqué et l'indice plus ou moins à la puissance signifie lorsque c'est plus, un mouvement vers l'avant et moins lorsque c'est un mouvement vers l'arrière. Et les barres verticales signifient un changement de sens. Reeds et Shepp ont résolu ce problème à l'aide d'un ordinateur, en réalité l'ordinateur les a aidé à trouver cet ensemble de possibilités et les a aidé à rédiger une preuve que cet ensemble est suffisant et Reeds et Shepp ont facilité la preuve afin qu'elle soit lisible par un être humain. De plus ils font remarquer que l'on peut se rapporter des 48 possibilités à 9 possibilités en remarquant des similarités dans les différents chemins. Nous allons dès maintenant étudier ces simplifications.

SLIDE

Effectivement en remarquant le fait qu'un chemin $l^-r^+s^+l^+$ ici en bleu foncé allant de $(0, 0, 0)$ à (x, y, ϕ) et que ce même chemin où l'on change la direction de chaque sous-chemin donc $l^+r^-s^-l^-$ ici en bleu clair mène de $(0, 0, 0)$ à $(-x, y, -\phi)$. Sur le dessin que j'ai fait, on pressent bien cette propriété.

SLIDE

De cette manière avec cette simplification dite "timeflip" on retire la moitié des cas puisque l'on ne peut se concentrer que sur les trajectoires commençant par un mouvement par l'avant indiqué par un exposant plus. J'ai donc barré en rouge ici toutes les trajectoires qui commençait par un mouvement vers l'arrière indiqué par un exposant moins.

SLIDE

De manière similaire on peut remarquer la propriété suivante : un chemin $r^+l^-s^-r^-$ allant de $(0, 0, 0)$ à (x, y, ϕ) et que ce même chemin on l'on inverse les directions, c'est-à-dire que les tourner à gauche

deviennent des tourner à droite et inversement, ce chemin est donc $l^+r^-s^-l^-$ allant de $(0, 0, 0)$ à $(x, -y, -\phi)$. A nouveau, on pressent bien cette propriété depuis mon exemple.

SLIDE

Avec cette simplification dite "reflect" on retire une trajectoire possible dans chaque groupe de 4. Effectivement on peut se restreindre par la simplification précédente à ne conserver que les trajectoires commençant par un mouvement à gauche en avant, c'est-à-dire un l^+ . J'ai donc barré en vert toutes les trajectoires plus nécessaires grâce à la simplification "reflect".

SLIDE

De manière similaire on peut remarquer qu'en inversant chaque sous-chemin d'un chemin, c'est-à-dire en changeant les exposants moins par des plus et inversement, $l^-s^-r^-l^+$, obtenant donc le chemin $l^+s^+r^+l^-$ le point final n'est plus (x, y, ϕ) mais $(x\cos\phi + y\sin\phi, x\sin\phi - y\cos\phi, \phi)$. Un peu plus dur de se convaincre de la véracité de cette propriété mais en tout cas sur l'exemple, cette propriété est vérifiée.

SLIDE

De cette manière on retire à nouveau 3 cas pour ceux qui en comportait 8 dans leur section. On choisit de se concentrer que sur les premières trajectoires de chaque section. J'ai donc barré en bleu ici toutes les trajectoires plus nécessaires par la simplification dite "reverse".

De plus Reeds et Shepp affirment que toutes les formules ayant deux sous-possibilités comportent une sous-possibilité de trop car jamais observée au sein d'un trajet optimal et que une des deux sous-possibilité peut donc à chaque fois ne pas être considéré.

SLIDE

Pour traiter le cas le plus simple de $l_t^+s_u^+l_v^+$ afin de vous donner une idée de la trigonométrie qui est appliquée dans chaque cas. Ici les indices t, u et v permettent d'indiquer la longueur de l'arc de cercle ou du segment considéré. Du coup on a u et t obtenu grâce à l'inverse d'une transformation polaire de $x - \sin\phi$ et $y - 1 + \cos\phi$, on obtient v en reportant $\theta - t$ entre 0 et 2π . De plus comme le font remarquer Reeds et Shepp, le chemin ne peut pas être optimal si t ou v est en dehors de $[0, \pi]$

SLIDE

En pratique les chemins de Dubins et Reeds-Shepp ne peuvent pas être utilisés tels quels dans la réalité car il est difficile de faire parfaitement suivre un chemin calculé à un robot. En plus du fait que les chemins de Dubins et Reeds-Shepp sont peu réalistes vus qu'ils ne traitent ni d'une variation de vitesse, ni de la possibilité de ne pas braquer au maximum, ni de la non parfaite adhérence des véhicules dans les virages.

SLIDE

Un autre problème est le fait que ces algorithmes ne traitent pas de la présence d'obstacles, ce qui est peu réaliste. Toutefois on peut utiliser un rapidly-exploring random tree, qui consiste à chercher efficacement dans un espace en construisant de manière aléatoire un arbre qui remplit l'espace. Effectivement au lieu de considérer la distance usuelle dans \mathbb{R}^2 on peut utiliser la pseudo distance induite par les chemins de Dubins voire celle de Reeds-Shepp qui est vraisemblablement une distance car symétrique puisque l'on autorise les mouvements en arrière.

SLIDE

Si certains désirent connaître les références que j'ai utilisé, les voici.