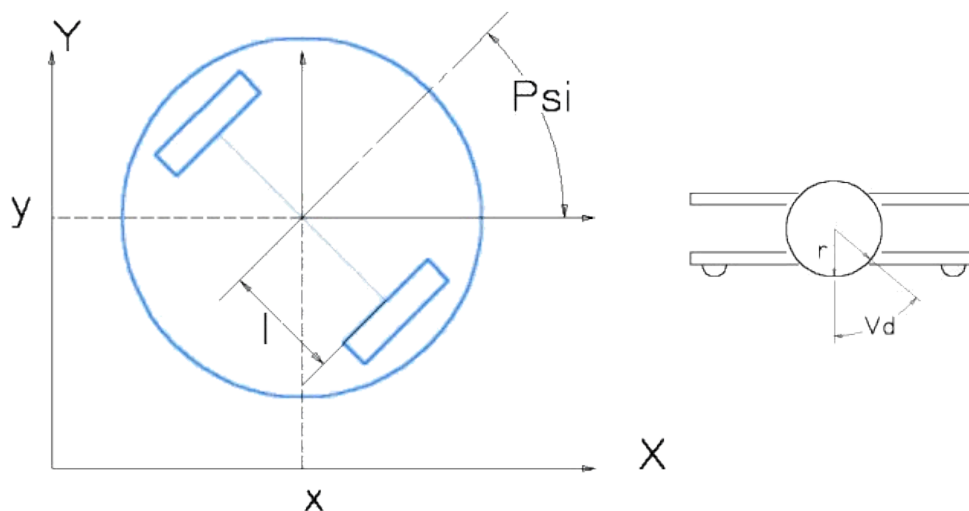


# MODELE GEOMETRIQUE

Modèle géométrique, derrière ce nom vulgaire, ce cache la méthode de localisation la plus courante. Généralement pour connaître sa position, un robot compte ses tours de roues (odométrie) et estime sa position à partir de cette information. Le modèle géométrique donne la transformation qui permet de passer des vitesses angulaires des roues à la vitesses instantanée du robot. Cette fiche technique présente le modèle géométrique d'un robot différentiel (type char ou pelleuse). Cette transformation se passe en 3 étapes. Nous allons calculer la vitesse du milieu des deux roues. Ensuite, nous calculerons la vitesse de n'importe quel point du robot, et enfin, à partir des vitesses nous calculerons la position.

## Modèle géométrique du milieu des roues

Voici deux schémas du robot, respectivement vue de dessus et vue de côté :



Avec :

- $2.l$  : distance entre les roues
- $r$  : rayon de la roue
- $V_d, V_g$  : vitesse des roues droite et gauche
- $x, y$  : position du robot
- $\Psi$  : orientation du robot

Pour commencer, nous allons calculer la vitesse linéaire des roues ( $V_{rd}$  et  $V_{rg}$ ) :

$$V_{rd} = r.V_d \quad V_{rg} = r.V_g$$

Nous pouvons en déduire la vitesse moyenne du centre des roues :

$$V_{moy} = \frac{V_{rg} + V_{rd}}{2}$$

Ensuite, pour ramener cette vitesse dans le repère (O,X,Y), nous allons la décomposer en X et en Y :

$$V_x = V_{moy} \cdot \cos(\Psi) = \frac{r.V_d}{2} \cdot \cos(\Psi) + \frac{r.V_g}{2} \cdot \cos(\Psi)$$

$$V_y = V_{moy} \cdot \sin(\Psi) = \frac{r.V_d}{2} \cdot \sin(\Psi) + \frac{r.V_g}{2} \cdot \sin(\Psi)$$

Il ne nous reste plus qu'à calculer la vitesse angulaire du robot. Elle nous est donnée par la relation suivante :

$$2.l.a_{psi} = r.V_d - r.V_g$$

Avec Psi pris dans le sens trigonométrique et  $a_{psi}$  la vitesse angulaire du robot. En isolant  $a_{psi}$  on obtient la relation suivante :

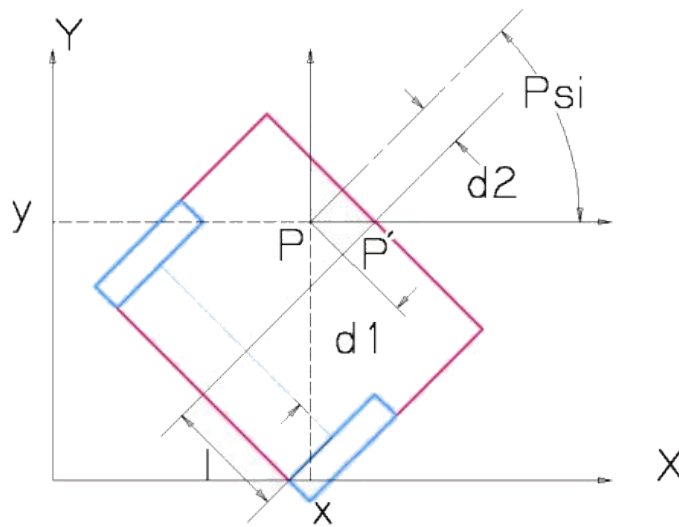
$$a_{psi} = \frac{r.V_d - r.V_g}{2.l}$$

Nous avons donc notre modèle géométrique, il ne nous reste plus qu'à le mettre sous forme matricielle :

$$\begin{bmatrix} V_x \\ V_y \\ a_{psi} \end{bmatrix} = \frac{r}{2} \cdot \begin{bmatrix} \cos(Psi) & \cos(Psi) \\ \sin(Psi) & \sin(Psi) \\ \frac{1}{l} & \frac{-1}{l} \end{bmatrix} \cdot \begin{bmatrix} V_d \\ V_g \end{bmatrix}$$

### Modèle géométrique du robot

Parfois, le centre du robot n'est pas le milieu des deux roues. Ou alors, le robot est équipé d'un outil, et c'est la position de cet outil qui nous intéresse. Il est alors possible de calculer le modèle géométrique en prenant en compte cette nouvelle contrainte. Supposons que le point du robot qui nous intéresse est décalé de  $d1$  et  $d2$  :



Voici la transformation (en position) qui permet de passer du centre des roues au point P'.

$$\begin{aligned} x^{P'} &= x + d1 \cdot \cos(Psi) \\ y^{P'} &= y + d1 \cdot \sin(Psi) \\ Psi^{P'} &= Psi \end{aligned}$$

En dérivant ces relations, on obtient les vitesses au point P'

$$V_x^{P'} = V_x - d1 \cdot \sin(Psi) \cdot a_{psi} = \frac{r}{2} \cdot \left( \left( \cos(Psi) - \frac{d1}{l} \cdot \sin(Psi) \right) V_d + \left( \cos(Psi) + \frac{d1}{l} \cdot \sin(Psi) \right) V_g \right)$$

$$V_y^{P'} = V_y + d1 \cdot \cos(Psi) \cdot a_{psi} = \frac{r}{2} \cdot \left( \left( \sin(Psi) + \frac{d1}{l} \cdot \cos(Psi) \right) V_d + \left( \sin(Psi) - \frac{d1}{l} \cdot \cos(Psi) \right) V_g \right)$$

$$a_{psi}^{P'} = a_{psi}$$

Sous forme matricielle, nous obtenons :

$$\begin{bmatrix} V_x^{P'} \\ V_y^{P'} \\ a_{psi}^{P'} \end{bmatrix} = \frac{r}{2} \begin{bmatrix} \cos(Psi) - \frac{d1}{l} \cdot \sin(Psi) & \cos(Psi) + \frac{d1}{l} \cdot \sin(Psi) \\ \sin(Psi) + \frac{d1}{l} \cdot \cos(Psi) & \sin(Psi) - \frac{d1}{l} \cdot \cos(Psi) \\ \frac{1}{l} & \frac{-1}{l} \end{bmatrix} \begin{bmatrix} V_d \\ V_g \end{bmatrix}$$

Pour obtenir le modèle au point P, il suffit de recommencer le changement de repère de P' à P :

$$\begin{bmatrix} V_x^{P'} \\ V_y^{P'} \\ a_{psi}^{P'} \end{bmatrix} = \frac{r}{2} \begin{bmatrix} \cos(Psi) - \frac{d1}{l} \cdot \sin(Psi) - \frac{d2}{l} \cdot \cos(Psi) & \cos(Psi) + \frac{d1}{l} \cdot \sin(Psi) + \frac{d2}{l} \cdot \cos(Psi) \\ \sin(Psi) + \frac{d1}{l} \cdot \cos(Psi) - \frac{d2}{l} \cdot \sin(Psi) & \sin(Psi) - \frac{d1}{l} \cdot \cos(Psi) + \frac{d2}{l} \cdot \sin(Psi) \\ \frac{1}{l} & \frac{-1}{l} \end{bmatrix} \begin{bmatrix} V_d \\ V_g \end{bmatrix}$$

### Calcul de la position

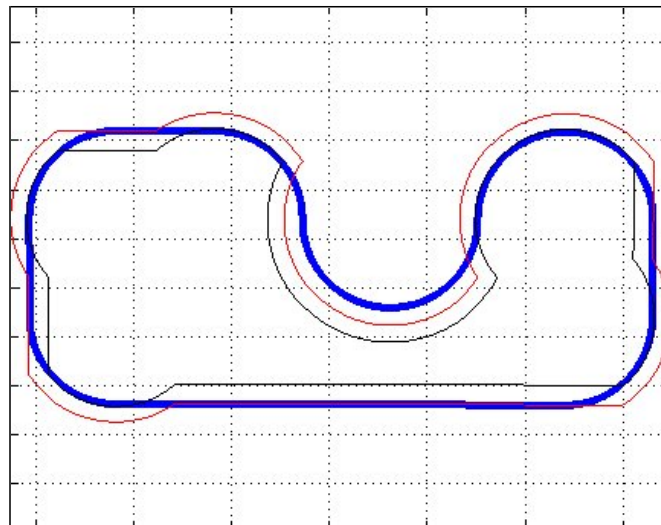
Une fois que l'on connaît les vitesses instantanées, on peut facilement calculer la position du robot, en intégrant  $V_x$ ,  $V_y$  et  $a_{psi}$ . Pour intégrer ces valeurs, il suffit de les sommer depuis le début :

$$x = \sum_0^n V_x$$

$$y = \sum_0^n V_y$$

$$Psi = \sum_0^n a_{psi}$$

Plutôt que de calculer la somme à chaque cycle, on ajoute le nouvelle incrément à la dernière position connue. Pour vérifier toutes ces équations, nous avons réalisé [ce petit programme](#) sous Matlab. Il calcule la trajectoires des trois point en fonction de la commande qui est appliquée sur les roues. La courbe ci-dessous représente cette trajectoire. Le milieu des roues est en bleu, le point intermédiaire P' est en rouge, et enfin la trajectoire du point P est en noir.



Ici, en simulation la trajectoire est parfaite. Mais la réalité est tout autre : les roues pattines, on ne connaît pas les distances exactes etc ... Et comme on intègre une vitesse pour en déduire une position, on intègre aussi l'erreur, ce qui veut dire que l'erreur s'accumule en fonction du temps. Pour pallier à ce problème voici quelques solutions :

- Faire un petit trajet
- Se recaller
- Avoir un système de localisation absolue

Ce modèle est lourd à implémenter dans un robot, mais il peut aussi servir de base à un simulateur : sur l'un des

robot de l'ancr, ce modèle sert à générer les trajectoires du robot en simulation. Ensuite, nous n'avons plus qu'à transférer les commandes des roues dans un microcontrôleur et le tour est joué !

## Liens

[Une page sur les codeurs et l'odométrie](#) (Merci Frédéric)

## Contact

Pour toutes questions envoyer moi un mail: [Sinclair](#)

[ICQ# : 144345434](#)

