

Modélisation et commande d'un robot RPR planaire

Luc BARON, ing., Ph.D.
Polytechnique Montréal

25 décembre 2025

1 Modélisation du robot

Le robot RPR planaire a 2 axes rotatifs et un axe prismatique. La position du bout est donnée par

$$\mathbf{p} = \begin{bmatrix} (L_1 + t_2) \cos(t_1) + L_3 \cos(t_1 + t_3) \\ (L_1 + t_2) \sin(t_1) + L_3 \sin(t_1 + t_3) \end{bmatrix}, \quad \mathbf{p} \equiv \begin{bmatrix} x \\ y \end{bmatrix}, \quad \boldsymbol{\theta} \equiv \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix}, \quad (1)$$

où L_1 , L_2 et L_3 sont des longueurs constantes. La vitesse du bout est obtenue par dérivé de (1) :

$$\dot{\mathbf{p}} = \mathbf{J}\dot{\boldsymbol{\theta}}, \quad \dot{\mathbf{p}} \equiv \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix}, \quad \dot{\boldsymbol{\theta}} \equiv \begin{bmatrix} \dot{t}_1 \\ \dot{t}_2 \\ \dot{t}_3 \end{bmatrix}, \quad \mathbf{J} \equiv \begin{bmatrix} \frac{\partial x}{\partial t_1} & \frac{\partial x}{\partial t_2} & \frac{\partial x}{\partial t_3} \\ \frac{\partial y}{\partial t_1} & \frac{\partial y}{\partial t_2} & \frac{\partial y}{\partial t_3} \end{bmatrix}, \quad (2)$$

où \mathbf{J} est la matrice *Jacobienne* (2×3) en position du robot. C'est un ensemble de 6 coefficients d'influences de $\dot{\boldsymbol{\theta}}$ sur $\dot{\mathbf{p}}$ qui varient avec la position $\boldsymbol{\theta}$. Ces équations forment le modèle cinématique position-vitesse du robot (forward kinematic). Les coefficients de \mathbf{J} sont présentés en annexe.

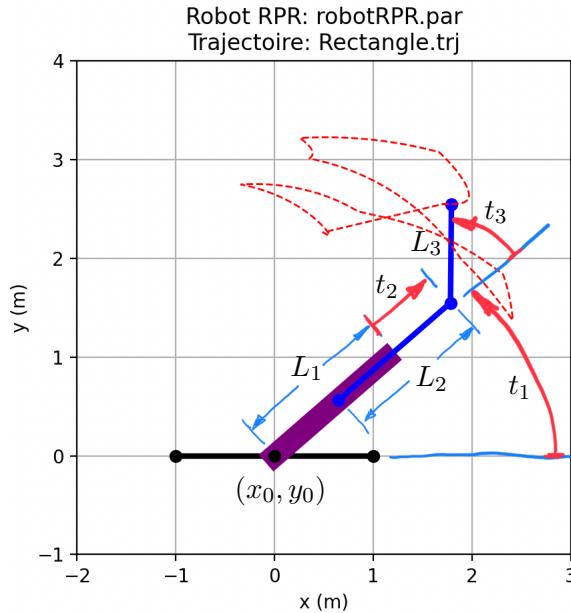


FIGURE 1 – Géométrie du robot RPR planaire : $L_1 = 1.5$, $L_2 = 1.5$ et $L_3 = 1.0$.

2 Commande du robot

La commande du robot se fait dans l'espace articulaire $\boldsymbol{\theta}$, alors que la tâche est définie dans l'espace cartésien \mathbf{p} . Un générateur de trajectoire doit calculer une suite de position articulaire $\boldsymbol{\theta}_k$ permettant au bout du robot de parcourir une suite de position cartésienne \mathbf{p}_k . Sous l'hypothèse de petit Δt , nous avons $\dot{\mathbf{p}} \rightarrow \Delta\mathbf{p}$ et $\dot{\boldsymbol{\theta}} \rightarrow \Delta\boldsymbol{\theta}$. Il est alors possible d'intégrer numériquement l'équation de vitesse (2) et résoudre pour $\boldsymbol{\theta}_k$. Ainsi, la cinématique inverse se calcule par

$$\Delta\boldsymbol{\theta}_M = \mathbf{J}^p \Delta\mathbf{p}, \quad \mathbf{J}^p \equiv \mathbf{J}^T (\mathbf{J}\mathbf{J}^T)^{-1} \quad (3)$$

où \mathbf{J}^p est le pseudoinverse gauche de \mathbf{J} (matrice 2×3) permettant de calculer la solution de norme-minimale $\Delta\boldsymbol{\theta}_M$, c'est-à-dire le $\Delta\boldsymbol{\theta}$ qui minimise $\|\Delta\boldsymbol{\theta}\|$. Avec un choix judicieux de pas de déplacement, le générateur de trajectoire doit découper chaque segment en une suite de position \mathbf{p}_k . La fonction Reach (voir l'algorithme 1) permet de calculer $\boldsymbol{\theta}_{k+1}$ à partir de $\boldsymbol{\theta}_k$, afin d'atteindre \mathbf{p}_{k+1} avec l'éq.(3). Les points \mathbf{p}_k doivent être très près les uns des autres.

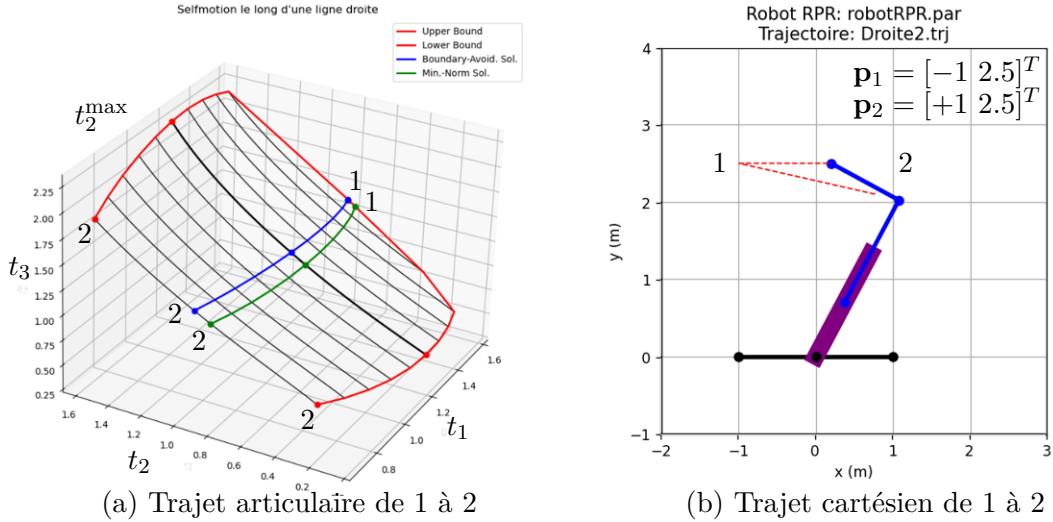


FIGURE 2 – Trajet en ligne droite de 1 à 2 avec les limites $\boldsymbol{\theta}^{\min} = [0 \ 0 \ 0]^T$ et $\boldsymbol{\theta}^{\max} = [\pi/2 \ 1.50 \ \pi]^T$: a) surface du noyau dans la boîte des limites articulaires, la courbe en vert est la solution de norme minimale, soit le chemin le plus court de 1 à 2; b) mouvement du robot de 1 à 2, alors que 1 est à la limite de l'enveloppe de travail du robot ($t_1 = \pi/2$).

3 Stratégie d'évitement des limites articulaires

Ce robot 3R en position est redondant. Il possède 3 moteurs pour seulement 2 variables de position (x, y). Il est donc possible de maintenir fixe la position du bout et déplacer le manipulateur. Cette redondance permet de choisir une position articulaire parmi toutes celles permettant de placer le bout au même point \mathbf{p}_k . D'un point de vue algébrique, cela signifie qu'il existe un $\Delta\boldsymbol{\theta}_H$ qui appartient au noyau de \mathbf{J} , c'est-à-dire que son déplacement cartésien $\Delta\mathbf{p}$ sera nulle :

$$\Delta\mathbf{p} = \mathbf{J}\Delta\boldsymbol{\theta}_H = \mathbf{0} \quad (4)$$

Ainsi, il est possible d'ajouter $\Delta\boldsymbol{\theta}_H$ à $\Delta\boldsymbol{\theta}_M$ sans compromettre l'atteinte de la position cible \mathbf{p}_k . Une façon simple de calculer un vecteur appartenant au noyau de \mathbf{J} est de projeter sur ce dernier un vecteur

arbitraire \mathbf{h} (3×1) en un vecteur $\Delta\boldsymbol{\theta}_H$ (aussi 3×1) de la façon suivante :

$$\Delta\boldsymbol{\theta}_H = (\mathbf{1} - \mathbf{J}^p \mathbf{J}) \mathbf{h}, \quad \mathbf{J}^p \equiv \mathbf{J}^T (\mathbf{J} \mathbf{J}^T)^{-1}. \quad (5)$$

Rappelons que $\Delta\boldsymbol{\theta}_H$ produit aucun déplacement $\Delta\mathbf{p}$, il permet plutôt de satisfaire une tâche secondaire en indiquant une direction préférée dans l'espace articulaire. En général, seul $\Delta\boldsymbol{\theta}_M$ est nécessaire pour générer des trajectoires. Mais lorsque le robot est près de ses limites articulaires, il est alors préférable d'utiliser les deux termes, $\Delta\boldsymbol{\theta}_M$ et $\Delta\boldsymbol{\theta}_H$. L'algorithme 1 doit alors être légèrement modifié.

Pour l'évitement des limites articulaires, la fonction objectif z demande de maintenir le robot le plus près possible de sa position articulaire milieu $\bar{\boldsymbol{\theta}}$, c'est-à-dire

$$z = \frac{1}{2}(\bar{\boldsymbol{\theta}} - \boldsymbol{\theta})^T \mathbf{W}^T \mathbf{W}(\bar{\boldsymbol{\theta}} - \boldsymbol{\theta}) \rightarrow \min \quad (6)$$

avec $\bar{\boldsymbol{\theta}}$ et la matrice diagonale de pondération \mathbf{W} définie par

$$\bar{\boldsymbol{\theta}} \equiv \frac{1}{2}(\boldsymbol{\theta}_{max} + \boldsymbol{\theta}_{min}), \quad \underline{\boldsymbol{\theta}} \equiv \frac{1}{(\boldsymbol{\theta}_{max} - \boldsymbol{\theta}_{min})}, \quad \mathbf{W} \equiv \text{diag}(\underline{\boldsymbol{\theta}}) \quad (7)$$

Clairement, $\bar{\boldsymbol{\theta}}$ et \mathbf{W} sont calculés à partir des limites du robot. Le vecteur gradient de z , appelé ∇z , est un vecteur de l'espace articulaire pointant vers la maximisation de z . Puisque nous désirons minimiser z , nous choisissons $\mathbf{h} = -\nabla z = \mathbf{W}(\bar{\boldsymbol{\theta}} - \boldsymbol{\theta})$. La matrice de pondération \mathbf{W} peut être choisi autrement.

4 Algorithme de résolution

L'algorithme 1 présente une procédure permettant d'effectuer un petit déplacement du bout du robot à partir de $\mathbf{p}_k = \text{forward_kinematic}(\boldsymbol{\theta}_k)$ vers \mathbf{p}_{k+1} afin de calculer $\boldsymbol{\theta}_{k+1}$. Cette fonction doit donc être appelée pour chacun des points discrétisés \mathbf{p}_k de la trajectoire cartésienne. La ligne 7 calcule la solution de norme minimale $\Delta\boldsymbol{\theta}_M$. Le calcul de la solution homogène $\Delta\boldsymbol{\theta}_H$ avec l'éq.(5) peut être ajouté afin de s'éloigner des limites articulaires. Le calcul du vecteur \mathbf{h} doit alors être ajouté. La saturation à la ligne 8 permet de bloquer les dépassements articulaires, si désirée.

Algorithm 1 : Atteindre \mathbf{p}_{cible} à partir de $\boldsymbol{\theta}$ avec $\Delta\boldsymbol{\theta}_M$ et limites articulaires bloquées par saturation

```

1: function  $\boldsymbol{\theta} = \text{REACH}(\mathbf{p}_{cible}, \boldsymbol{\theta})$  ▷ Hypothèse :  $\boldsymbol{\theta}$  est très près de  $\mathbf{p}_{cible}$ 
2:    $n, \delta \leftarrow 25, 10^{-5}$  (nombre maximum d'itérations, précision de position en mètre)
3:    $d, dt \leftarrow 0.75, 0.025$  (facteur d'amortissement, pas de temps en seconde)
4:    $\mathbf{p} \leftarrow \text{forward\_kinematic}(\boldsymbol{\theta})$  ▷ Position initiale par cinématique directe
5:   repeat
6:      $\Delta\mathbf{p} = d(\mathbf{p}_{cible} - \mathbf{p})/dt$  ▷ Déplacement cartésien amorti
7:      $\Delta\boldsymbol{\theta}_M = \mathbf{J}^p \Delta\mathbf{p}$  ▷ Solution de norme minimale
8:      $\boldsymbol{\theta} = \text{Saturation}(\boldsymbol{\theta} + dt\Delta\boldsymbol{\theta}_M)$  ▷ Avancement articulaire sans dépassements
9:      $\mathbf{p} \leftarrow \text{forward\_kinematic}(\boldsymbol{\theta})$  ▷ Position actuelle par cinématique directe
10:     $n = n - 1$ 
11:   until  $n = 0$  or  $\|(\mathbf{p}_{cible} - \mathbf{p})\| < \delta$  ▷ Nombre maximum d'itération ou précision atteinte
12:   if  $n = 0$  then
13:     display(Précision non atteinte)

```

Annexe - Matrice Jacobienne

La matrice *Jacobienne* en position \mathbf{J} du robot RPR est de dimension 2×3 . Elle est définie par les dérivées partielles de x et y par rapport à t_1 , t_2 et t_3 , telle que :

$$\dot{\mathbf{p}} = \mathbf{J}\dot{\boldsymbol{\theta}}, \quad \dot{\mathbf{p}} \equiv \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix}, \quad \dot{\boldsymbol{\theta}} \equiv \begin{bmatrix} \dot{t}_1 \\ \dot{t}_2 \\ \dot{t}_3 \end{bmatrix}, \quad \mathbf{J} \equiv \begin{bmatrix} \frac{\partial x}{\partial t_1} & \frac{\partial x}{\partial t_2} & \frac{\partial x}{\partial t_3} \\ \frac{\partial y}{\partial t_1} & \frac{\partial y}{\partial t_2} & \frac{\partial y}{\partial t_3} \end{bmatrix}, \quad (8)$$

avec

$$\begin{aligned} \frac{\partial x}{\partial t_1} &= -(L_1 + t_2) \sin(t_1) - L_3 \sin(t_1 + t_3) \\ \frac{\partial y}{\partial t_1} &= (L_1 + t_2) \cos(t_1) + L_3 \cos(t_1 + t_3) \\ \frac{\partial x}{\partial t_2} &= \cos(t_1) \\ \frac{\partial y}{\partial t_2} &= \sin(t_1) \\ \frac{\partial x}{\partial t_3} &= -L_3 \sin(t_1 + t_3) \\ \frac{\partial y}{\partial t_3} &= L_3 \cos(t_1 + t_3) \end{aligned} \quad (9)$$