MEC1315 TI en ingénierie

Travail Synthèse 1 – Automne 2025

Cahier des charges

Date de remise Moodle le 24 octobre 2025 avant 17h00

Équipe de 4 étudiants

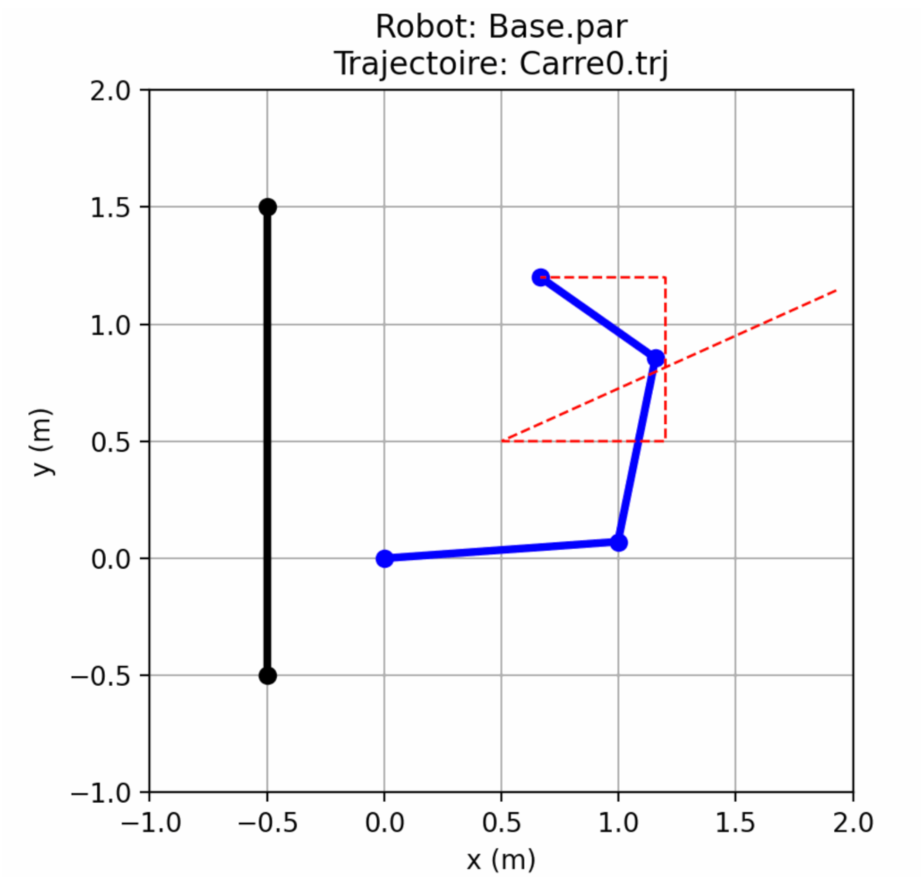


Figure 1 – Animation d'un robot 3R planaire

Polytechnique Montréal

Objectif

Le travail synthèse no. 1 (TS1) porte sur la résolution numérique de l'équation de vitesse d'un robot 3R planaire pour le déplacement en ligne droite de son bout tout en évitant d'outrepasser les limites articulaires. Un simulateur graphique et un générateur de trajectoire doivent être conçus en Python.

1. Simulateur graphique du robot 3R (30%)

En utilisant le module **Animation** de la bibliothèque **matplotlib**, vous devez préparer un simulateur graphique du robot planaire 3R, tel que défini par les 23 paramètres du fichier de configuration (robot.par), alors que les trajectoires sont définies dans l'espace articulaire par un fichier (carre.trj). Des exemples sont présentés en annexe. Votre simulateur doit utiliser seulement les bibliothèques standard disponibles dans les salles informatiques de Polytechnique sans aucune autre dépendance externe.

Requis :
1- Lecture d'un fichier de configuration, tel que robot.par (format obligatoire);
2- Lecture d'un fichier de trajectoire, tel que carre.trj (format obligatoire);
3- Afficher la figure avec le nom des fichiers .par et .trj, ainsi que les étiquettes aux axes x et y;
2- Afficher la figure avec une échelle égale en x et y, jusqu'aux limites demandées;
3- Afficher le robot en bleu avec points, si ses angles sont à l'intérieur des limites, sinon en rouge;
4- Afficher le mur fixe en noir avec points;
5- Tracer le trajet parcouru par le bout du robot en tireté rouge.

Il est permis d'utiliser un moteur conversationnel d'intelligence artificielle (IA), tel que ChatGPT, Copilot ou autre, pour vous assister dans la conception de votre simulateur. Il est alors requis de fournir en annexe, une **transcription de votre conversation** (max 4 pages) initiale avec l'IA jusqu'à votre premier script élémentaire fonctionnel. Ce premier script poursuivra son évolution afin de satisfaire à tous les requis. Finalement, vous devez préparer, avec l'aide de l'IA, une **analyse critique** (~4 pages) de votre script final.

2. Générateur de trajectoires (30%)

Vous devez préparer un **générateur** de trajectoires articulaires, tel que défini par les fichiers (.trj). La configuration du robot est définie par les mêmes 23 paramètres du fichier de configuration (robot.par), alors que les trajectoires Cartésiennes sont définies par un fichier (carre.xy). Des exemples sont présentés en annexe. Votre générateur doit utiliser seulement les bibliothèques standard disponibles dans les salles informatiques de Polytechnique sans aucune autre dépendance externe.

Requis :
1- Lecture d'un fichier de configuration, tel que robot.par (format obligatoire);
2- Lecture d'un fichier de trajectoire Cartésienne, tel que carre.xy (format obligatoire);
3- Écriture d'un fichier de trajectoire articulaire, tel que carre.trj (format obligatoire);
4- Discrétiser les droites avec un pas suffisamment petit (theta ~ sin(theta));
5- Résoudre l'équation de vitesse du robot pour des petits déplacements Cartésien du bout;
6- Appliquer une stratégie d'évitement des limites articulaires.

Il est permis d'utiliser un moteur conversationnel d'intelligence artificielle (IA), tel que ChatGPT, Copilot ou autre, pour vous assister dans la conception de votre générateur. Il n'est pas requis de fournir une transcription de votre conversation avec l'IA. Aucune analyse critique de votre script n'est requise. Cependant, votre **méthode d'évitement des limites articulaires** doit être présenté dans votre rapport.

3. Livrables

Vos 8 fichiers doivent être remis en un seul fichier ZIP nommé **TS1_Equipe_XX.zip** (max. 80 Mo):

1- Le script Python **Generateur_##.py**;
2- Le script Python **Simulateur_##.py**;
3- <u>Votre exemple</u> de fichiers **Robot_##.par** , **Trajet_##.xy** et **Trajet_##.trj**;
4- Le fichier **Rapport_##.pdf** contenant :
   a) l'analyse critique du script du simulateur (~4 pages);
   b) la méthode d'évitement des limites articulaires avec vos résultats (~4 pages);
   c) la contribution des membres de l'équipe
   d) en annexe : le début de la conversation de co-conception avec l'IA d'un script élémentaire fonctionnel du simulateur, même si incomplet (max 4 pages en annexe);
5- <u>Vos fichiers</u> GIF animés : **Trajet0_##.gif** (sans évitement) et **Trajet1_##.gif** (avec évitement).

Vos scripts doivent pouvoir s'exécuter avec l'exemple de robot/trajectoire présentés en annexe, ainsi qu'avec votre exemple original de robot/trajectoire présentés dans votre rapport. Pour réduire la taille des fichiers GIF, il est possible de réduire la variable **fps** (frame-par-seconde), ainsi que le nombre total de frame.

4. Critères d'évaluation

Ce TS1 compte pour 20% de la note globale du cours MEC1315. Il doit être fait en équipe de 4 étudiants.

1- Qualité du script générateur vérifié sur les fichiers exemples en annexe (30%);
2- Qualité du script simulateur vérifié sur les fichiers exemples en annexe (30%);
3- Rapport : analyse critique du script final du simulateur et verbatim de co-conception en IA (10%);
4- Rapport : stratégie et originalité de <u>votre exemple</u> de robot/trajectoire (20%);
5- Rapport : pertinence des GIF animés (10%);
6- Qualité d'un script : encapsulation de fonctions, robustesse, ergonomie et lisibilité;
7- Pénalités pour le non-respect des requis et directives;

L'équipe d'enseignement

Automne 2025

Annexe

Exemple de fichier (format obligatoire) : Robot.par

```
# Fichier Classique.par: Paramètres du robot 3R Planaire
# MEC1315 Équipe no: XX (Automne 2025)
 1.0      # L_1     : longueur du segment 1
 0.8      # L_2     : longueur du segment 2
 0.6      # L_3     : longueur du segment 3
-1.0      # xmin    : borne de gauche du graphique
 2.0      # xmax    : borne de droite du graphique
-1.0      # ymin    : borne du bas du graphique
 2.0      # ymax    : borne du haut du graphique
 0.0      # x0      : Position x de l'axe theta1
 0.0      # y0      : Position y de l'axe theta1
 0.157079 # t1_dep  : Position départ de theta1
 0.457079 # t2_dep  : Position départ de theta2
 0.457079 # t3_dep  : Position départ de theta3
-0.35     # t1_min  : Limite minimale de theta1
 2.0      # t1_max  : Limite maximale de theta1
-2.5      # t2_min  : Limite minimale de theta2
 2.5      # t2_max  : Limite maximale de theta2
-2.0      # t3_min  : Limite minimale de theta3
 2.0      # t3_max  : Limite maximale de theta3
-0.5      # xmur[0] : Position x0 du mur
-0.5      # xmur[1] : Position x1 du mur
-0.5      # ymur[0] : Position y0 du mur
 1.5      # ymur[1] : Position y1 du mur
 25       # dt      : Pas de temps (msec.)
```

Exemple de fichier (format obligatoire) : Trajet.xy (coordonnées Cartésiennes x y)

```
0.5  0.5
1.2  0.5
1.2  1.2
0.5  1.2
0.5  0.5
```

Exemple de fichier (format obligatoire): Trajet.trj (coordonnées theta_1 theta_2 theta_3)

```
1.570789999999999964e-01 4.570790000000000131e-01 4.570790000000000131e-01
1.508438904978509354e-01 4.649936731818883895e-01 4.652399057200604648e-01
1.447183251527879855e-01 4.727837796223884892e-01 4.732716704308652100e-01

...                      ...                      ...

2.358300270500452156e-01 2.113728334922261864e-01 7.469520981815528371e-01
2.411688297036222450e-01 2.053389924210800022e-01 7.368047866269827884e-01
2.466011817027726483e-01 1.992361984060130875e-01 7.264773760591812790e-01
```

Modélisation et commande d'un robot 3R planaire

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

29 septembre 2025

1 Modélisation du robot

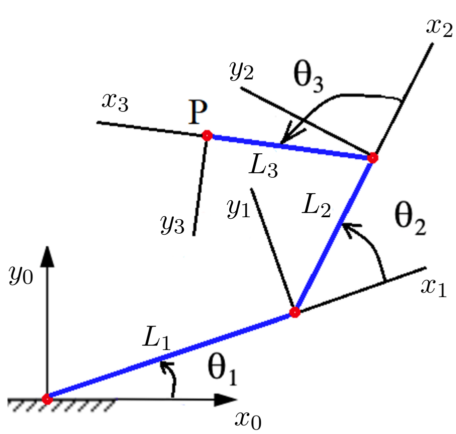
Le robot a 3 axes de rotation perpendiculaires au plan XY. La position du bout est donnée par

$$\mathbf{p} = \left[\begin{array}{c} L_1\cos(\theta_1) + L_2\cos(\theta_1+\theta_2) + L_3\cos(\theta_1+\theta_2+\theta_3) \\ L_1\sin(\theta_1) + L_2\sin(\theta_1+\theta_2) + L_3\sin(\theta_1+\theta_2+\theta_3) \end{array}\right], \quad \mathbf{p} \equiv \left[\begin{array}{c} x \\ y \end{array}\right], \quad \boldsymbol{\theta} \equiv \left[\begin{array}{c} \theta_1 \\ \theta_2 \\ \theta_3 \end{array}\right], \tag{1}$$

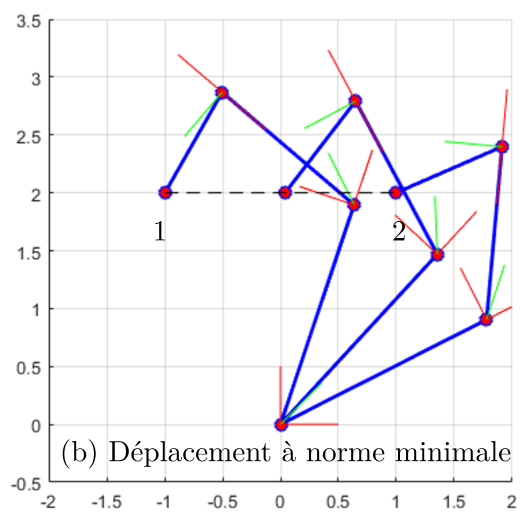
où L_1, L_2, L_3 sont les longueurs des membrures. La vitesse du bout est obtenue par dérivé de (1) :

$$\dot{\mathbf{p}} = \mathbf{J}\dot{\boldsymbol{\theta}}, \quad \dot{\mathbf{p}} \equiv \left[\begin{array}{c} \dot{x} \\ \dot{y} \end{array}\right], \quad \dot{\boldsymbol{\theta}} \equiv \left[\begin{array}{c} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{array}\right], \quad \mathbf{J} \equiv \left[\begin{array}{ccc} \frac{\partial x}{\partial \theta_1} & \frac{\partial x}{\partial \theta_2} & \frac{\partial x}{\partial \theta_3} \\ \frac{\partial y}{\partial \theta_1} & \frac{\partial y}{\partial \theta_2} & \frac{\partial y}{\partial \theta_3} \end{array}\right], \tag{2}$$

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.



(a) Géométrie du robot 3R



(b) Déplacement à norme minimale de 1 à 2

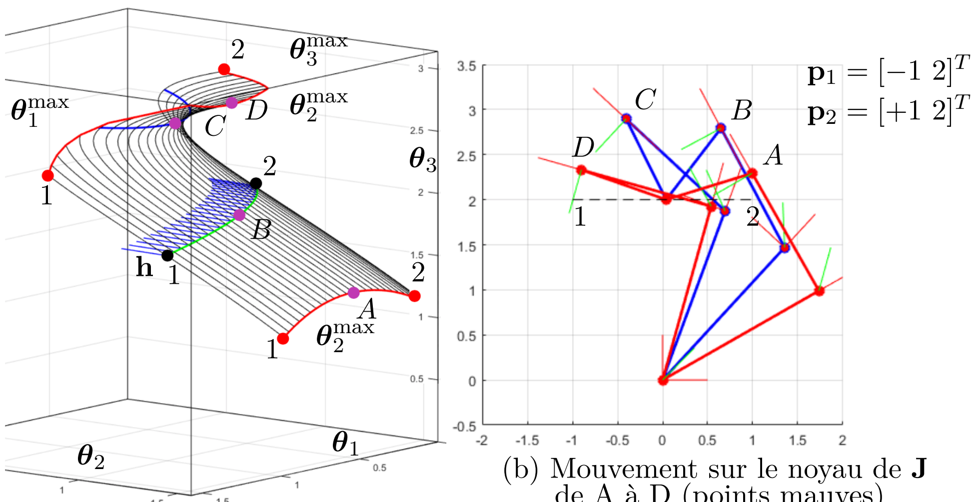
FIGURE 1 – Robot 3R planaire : Géométrie et déplacement à norme minimale avec $L_1 = 2$, $L_2 = 1.5$ et $L_3 = 1$. Les limites articulaires sont illustrées par des lignes rouges et la position milieu en vert.

2 Commande du robot

La commande d'un 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 $\dot{\mathbf{p}} \to \Delta\mathbf{p}$ et $\dot{\boldsymbol{\theta}} \to \Delta\boldsymbol{\theta}$. 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

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

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.



(a) Surface du noyau du segment 1-2

(b) Mouvement sur le noyau de $\mathbf{J}$ de A à D (points mauves).

FIGURE 2 – Ligne droite de 1 à 2 avec les limites $\boldsymbol{\theta}^{\min} = [0\ 0\ 0]^T$ et $\boldsymbol{\theta}^{\max} = [\pi/2\ \ \pi/2\ \ \pi]^T$: a) surface du noyau dans la boîte des limites articulaires, la courbe en vert est la trajectoire de norme minimale entre 1 à 2, soit le chemin le plus court ; b) mouvement sur le noyau à 50% du segment, où la configuration B est la solution de norme minimale, alors que A et D sont aux limites articulaires. Les vecteurs $\mathbf{h}$, illustrés en bleu, sont projetés sur la surface du noyau.

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 :

$$\Delta\mathbf{p} = \mathbf{J}\Delta\boldsymbol{\theta}_H = \mathbf{0} \tag{4}$$

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 :

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

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 articulaire, 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

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

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

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

Clairement, $\bar{\boldsymbol{\theta}}$ et $\mathbf{W}$ sont calculés à partir des limites du robot. Le vecteur gradient de z, appelé $\boldsymbol{\nabla} z$, est un vecteur de l'espace articulaire pointant vers la maximisation de z. Puisque nous désirons minimiser z, nous choisissons $\mathbf{h} = -\boldsymbol{\nabla} z = \mathbf{W}(\bar{\boldsymbol{\theta}} - \boldsymbol{\theta})$. 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 $\mathbf{p}_k = \textbf{forward\_kinematic}(\boldsymbol{\theta}_k)$ 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 8 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 9 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: $n \leftarrow 25$ (nombre maximum d'itérations)
3: $d \leftarrow 0.75$ (facteur d'amortissement)
4: $\delta \leftarrow 10^{-5}$ (précision en position)
5: $\mathbf{p} \leftarrow \textbf{forward\_kinematic}(\boldsymbol{\theta})$ ▷ Position initiale par cinématique directe
6: **repeat**
7: $\Delta\mathbf{p} = d(\mathbf{p}_{cible} - \mathbf{p})$ ▷ Déplacement cartésien amorti
8: $\boldsymbol{\Delta\theta}_M = \mathbf{J}^p\boldsymbol{\Delta}\mathbf{p}$ ▷ Solution de norme minimale
9: $\boldsymbol{\theta} = \text{Saturation}(\boldsymbol{\theta} + d\boldsymbol{\Delta\theta}_M)$ ▷ Avancement amorti sans dépasser les limites
10: $\mathbf{p} \leftarrow \textbf{forward\_kinematic}(\boldsymbol{\theta})$ ▷ Position actuelle par cinématique directe
11: $n = n - 1$
12: **until** $n = 0$ or $||(\mathbf{p}_{cible} - \mathbf{p})|| < \delta$ ▷ Nombre maximum d'itération ou précision atteinte
13: **if** $n = 0$ **then**
14: display(Précision non atteinte)

Annexe - Matrice Jacobienne

La matrice *Jacobienne* en position $\mathbf{J}$ du robot 3R est de dimension 2×3. Elle est définie par les dérivées partielles de x et y par rapport à θ_1, θ_2 et θ_3, telle que :

$$\dot{\mathbf{p}} = \mathbf{J}\dot{\boldsymbol{\theta}}, \quad \dot{\mathbf{p}} \equiv \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix}, \quad \dot{\boldsymbol{\theta}} \equiv \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix}, \quad \mathbf{J} \equiv \begin{bmatrix} \frac{\partial x}{\partial \theta_1} & \frac{\partial x}{\partial \theta_2} & \frac{\partial x}{\partial \theta_3} \\ \frac{\partial y}{\partial \theta_1} & \frac{\partial y}{\partial \theta_2} & \frac{\partial y}{\partial \theta_3} \end{bmatrix}, \tag{8}$$

avec

$$\begin{aligned}
\frac{\partial x}{\partial \theta_1} &= -L_1 \sin(\theta_1) - L_2 \sin(\theta_1 + \theta_2) - L_3 \sin(\theta_1 + \theta_2 + \theta_3) \\
\frac{\partial y}{\partial \theta_1} &= L_1 \cos(\theta_1) + L_2 \cos(\theta_1 + \theta_2) + L_3 \cos(\theta_1 + \theta_2 + \theta_3) \\
\frac{\partial x}{\partial \theta_2} &= -L_2 \sin(\theta_1 + \theta_2) - L_3 \sin(\theta_1 + \theta_2 + \theta_3) \\
\frac{\partial y}{\partial \theta_2} &= L_2 \cos(\theta_1 + \theta_2) + L_3 \cos(\theta_1 + \theta_2 + \theta_3) \\
\frac{\partial x}{\partial \theta_3} &= -L_3 \sin(\theta_1 + \theta_2 + \theta_3) \\
\frac{\partial y}{\partial \theta_3} &= L_3 \cos(\theta_1 + \theta_2 + \theta_3)
\end{aligned} \tag{9}$$