

Travail Synthèse 1 – Hiver 2026

Cahier des charges

Date de remise Moodle le 27 février 2026 avant 23h59

Équipe de 4 étudiants

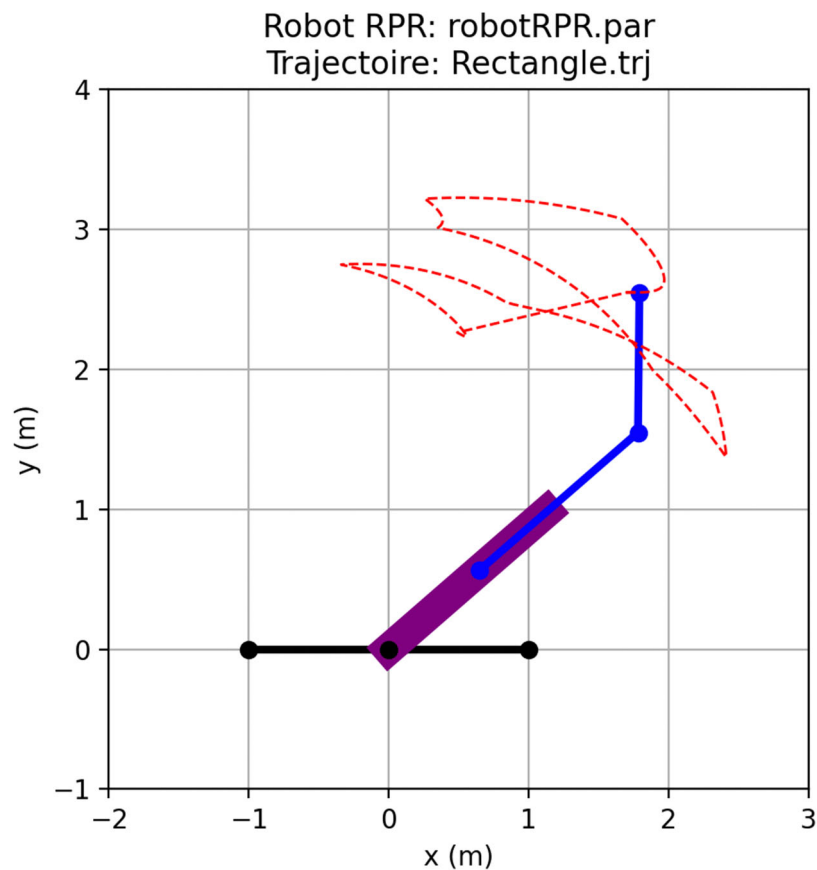


Figure 1 – Animation d'un robot RPR planaire

Objectif

Le travail synthèse no. 1 (TS1) porte sur la résolution numérique de l'équation de vitesse d'un robot RPR 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 (30%)

En utilisant le module **Animation** de la bibliothèque **matplotlib**, vous devez préparer un simulateur graphique du robot RPR planaire, tel que défini par les 24 paramètres du fichier de configuration ([robotRPR.par](#)), alors que les trajectoires sont définies dans l'espace articulaire par un fichier ([exemple.trj](#)). Votre simulateur doit utiliser seulement les bibliothèques standard disponibles dans les salles informatiques de Polytechnique sans aucune autre dépendance externe.

Requis du script :

- 1- Lecture du fichier de configuration [robotRPR.par](#) (fournit);
- 2- Lecture du fichier de trajectoire, tel que [exemple.trj](#) (fournit);
- 3- Afficher la figure avec le nom du 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 du fichier [.par](#);
- 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 selon le fichier [.par](#);
- 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) un **verbatim** du début de votre conversation (max 4 pages) avec l'IA. Vous devez préparer (avec l'aide de l'IA) une **analyse critique commentée** de votre script final (max 2 pages).

2. Générateur de trajectoires (30%)

Vous devez préparer un **générateur** de trajectoires, afin produire un fichiers (.trj) à partir d'un fichier de segment de droite ([rectangle.xy](#)). La configuration du robot est définie par les mêmes 24 paramètres du fichier de configuration ([robotRPR.par](#)). Le paramètre limites du fichier [.par](#) est 0 si aucun évitement des limites articulaires et 1 avec une stratégie d'évitement active. Vous devez choisir un trajet [.xy](#), puis produire un fichier [.trj](#) sans et avec évitement des limites articulaires. Vous devrez produire des GIF animés de vos deux résultats. Votre générateur doit utiliser seulement les bibliothèques standard disponibles dans les salles informatiques de Polytechnique sans aucune autre dépendance externe.

Requis du script :

- 1- Lecture du fichier de configuration [robotRPR.par](#) (fournit);
- 2- Lecture du fichier de trajectoire Cartésienne [rectangle.xy](#) (fournit);
- 3- Écriture du fichier de trajectoire articulaire [Trajet.trj](#) (format idem à [exemple.trj](#));
- 4- Discrétiser les segments avec un pas suffisamment petit ($\theta \sim \sin(\theta)$);
- 5- Résoudre l'équation de vitesse du robot pour des petits déplacements du bout;
- 6- Implanter votre 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_XX.py**;
- 2- Le script Python **Simulateur_XX.py**;
- 3- Vos fichiers **Trajet_XX.xy**, **Trajet_libre_XX.trj** et **Trajet_limite_XX.trj**;
- 4- Le fichier **Rapport_##.pdf** contenant :
 - a) l'analyse critique du script du simulateur (max 2 pages);
 - b) la méthode d'évitement des limites articulaires avec vos résultats (max 2 pages);
 - c) la contribution des membres de l'équipe
 - d) en annexe : le verbatim du début de la conversation avec l'IA de la co-conception du simulateur (max 4 pages);
- 5- Vos GIF animés : **Trajet_libre_XX.gif** (sans évitement) et **Trajet_limite_XX.gif** (avec évitement).

Vos scripts doivent pouvoir s'exécuter avec les fichiers fournis, 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 simulateur vérifié sur nos fichiers fournis et vos fichiers (30%);
- 2- Qualité du script générateur vérifié sur nos fichiers fournis et vos fichiers (30%);
- 3- Rapport : analyse critique du script du simulateur avec vos commentaires (10%);
- 4- Rapport : présentation et résultats de votre stratégie d'évitement (10%);
- 5- Rapport : pertinence des GIF animés de votre stratégie (10%);
- 6- Qualité d'un script : fonctions, robustesse, ergonomie et lisibilité (10%);
- 7- Pénalités pour le non-respect des requis et directives;

Note : Vos scripts doivent utiliser des fonctions, tels que `reach()`, `forward()`, `saturation()`, `jacobienne()` et autres, afin d'encapsuler les opérations mathématiques dans des outils de plus haut niveau. Cela permet également de mieux répartir le travail entre les membres de l'équipe, ainsi que, réutiliser les fonctions dans les scripts du simulateur et du générateur.

L'équipe d'enseignement

Hiver 2026

Annexe

Exemple de fichier (format obligatoire) : robotRPR.par

```
# Fichier robotRPR.par Paramètres du robot RPR Planaire
# MEC1315 Équipe no: XX (Hiver 2026)
1.5      # L_1      : longueur du segment 1
1.5      # L_2      : longueur du segment 2
1.0      # L_3      : longueur du segment 3
-2.0     # xmin     : borne de gauche du graphique
3.0      # xmax     : borne de droite du graphique
-1.0     # ymin     : borne du bas du graphique
4.0      # ymax     : borne du haut du graphique
0.0      # x0       : Position x de l'axe t1
0.0      # y0       : Position y de l'axe t1
0.757079 # t1_dep   : Position départ de t1
0.50     # t2_dep   : Position départ de t2
1.570796 # t3_dep   : Position départ de t3
0.0      # t1_min   : Limite minimale de t1
1.570796 # t1_max   : Limite maximale de t1
0.0      # t2_min   : Limite minimale de t2
1.500000 # t2_max   : Limite maximale de t2
0.0      # t3_min   : Limite minimale de t3
3.141593 # t3_max   : Limite maximale de t3
-1.0     # xmur[0]  : Position x0 du mur
1.0      # xmur[1]  : Position x1 du mur
0.0      # ymur[0]  : Position y0 du mur
0.0      # ymur[1]  : Position y1 du mur
25       # dt       : Pas de temps (msec.)
1        # limites  : 0 pas d'évitement ou 1 avec évitement
```

Fichier XY : rectangle.xy (fournit) => (coordonnées X et Y en m)

```
0.0  1.0
0.0  3.0
1.0  3.0
1.0  1.0
0.0  1.0
```

Fichier trajectoire : exemple.trj (fournit) => (theta_1 theta_2 theta_3 en radian)

```
1.570789999999999964e-01 4.57079000000000000131e-01 4.57079000000000000131e-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
```