

TD1 – Cours de robotique développementale et cognitive

Découverte de la programmation de la plateforme Poppy Humanoid

Pierre-Yves Oudeyer et Pierre Rouanet

Le but de ce TD est de vous familiariser avec l'environnement de programmation de la plateforme Poppy, et en particulier l'utilisation de la librairie de programmation pypot (en python), de l'interface de programmation interactive Jupyter notebooks (<http://jupyter.org/>), du simulateur de robots Poppy VREP (<http://www.coppeliarobotics.com/>), et de la plateforme physique Poppy Humanoid.

Pour réaliser les étapes du TD décrit ci-dessous, vous devrez partir des deux tutoriels suivants, sous la forme de notebooks:

Tutoriel d'installation et de découverte:

<https://github.com/poppy-project/poppy-humanoid/blob/master/software/samples/notebooks/Controlling%20a%20Poppy%20humanoid%20in%20V-REP%20using%20pypot.ipynb>

Tutoriel "Premiers pas":

https://github.com/poppy-project/community-notebooks/blob/master/tutorials-education/poppy-humanoid_poppy-torso_vrep_installation%20et%20prise%20en%20main/poppy%20simulé/premier%20pas%20avec%20poppy%20humanoid%20en%20python%20-%2010%20choses%20à%20savoir.ipynb

Ce TD est à rendre sous la forme d'un notebook Jupyter incluant code, graphiques et explications sous forme de textes (et de pseudo-code ou d'équations si nécessaire).

Etape 1: Installez les outils logiciels python/Anaconda, V-REP, pypot et poppy_humanoid.

Etape 2 : Apprendre à faire bouger le robot Poppy Humanoid et à récupérer la valeur de capteurs (réels ou virtuels) dans V-REP:

- Afficher la liste de tous les moteurs et la valeur de leur position courante
- Faire bouger la tête et les bras avec les commandes goal_position et goto_position. Quelles sont les différences? La position courante (angle) mesurée des moteurs est-elle toujours la même que celle de la commande de position qu'on leur envoie ? Générer des courbes permettant de visualiser les différences s'il y en a, et identifier les situations dans lesquelles il peut y avoir des différences.
- Faire réaliser une trajectoire au bras gauche, et mesurer l'évolution de la position de la main gauche, que vous visualiserez sous forme graphique.
- Avec l'outil "Primitives", réaliser un mouvement des bras dans lequel la trajectoire de chaque moteur est une somme de plusieurs sinus de paramètres différents. Est-il possible de désactiver en temps réel (pendant le mouvement) l'un de ces sinus ?

Etape 3 : Inventer une petite chorégraphie de mouvement pour le robot Poppy Humanoid. Dans un premier temps, testez là sur le simulateur V-REP, en visualisant la trajectoire de position de tous les moteurs. Testez là ensuite sur le vrai robot. Comparez la trajectoire des moteurs (mesurée en temps réel) réalisée avec le vrai robot avec celle réalisée avec le robot simulé. Qu’observez vous ?

Note 1 : Pour lancer votre script sur le robot réel Poppy Humanoid, suivre les instructions du tutoriel suivant : <https://poppy-project.gitbooks.io/poppy-docs/content/en/from-simulation-to-real-robot/using-jupyter.html> . Il faudra penser à installer les bibliothèques 3rd party nécessaires sur le robot, ainsi qu’à uploader votre notebook.

Note 2 : Pour inventer une chorégraphie, vous pourrez vous inspirer des mouvements et chorégraphies présentes sur les vidéos suivantes : <https://www.youtube.com/playlist?list=PLdX8R06QsgB6YCzezJHoYuRToF0hYk3Sf>

En particulier, dans cette liste de mouvements se trouvent ceux créés par Thomas Peyruse (<https://www.poppy-project.org/project/marionnettes-electriques/>) dans le cadre du spectacle de danse School of Moon dans lequel plusieurs robots Poppy Humanoid interviennent : <http://shonen.info/schoolofmoon/>

Etape 4 : A l’aide du tutorial suivant, créer une expérimentation avec le robot Poppy Torso dans V-REP dans laquelle le robot essaie des mouvements paramétrés et observe si (et comment) cela fait bouger un cube placé devant lui. Pour cela réaliser les sous-étapes suivantes:

- Inventez une manière de produire un mouvement avec l’un des bras de telle sorte que ce mouvement soit paramétré par des nombres réels (par exemple, si on lance le mouvement avec les paramètres [1.5; 4.6; 6.5; 1.2], cela produit un trajectoire du bras particulière qui dépend de ces nombres).
- Implémenter une boucle dans laquelle à chaque itération le robot essaie des paramètres aléatoires de ses mouvements et observe le déplacement du cube à partir d’une position initiale du cube qui est toujours la même (donc la position du cube et celle du robot doit être ré-initialisée à chaque itération)
- Implémenter le stockage en mémoire de tous les couples (paramètres moteurs, déplacement de l’objet) observés.

A partir de la base de donnée collectée par le robot, pouvez vous imaginer une méthode simple permettant de réaliser de manière approximative des déplacements du cube choisis arbitrairement par un expérimentateur ? Implémenter cette méthode simple (ou plusieurs variantes si vous avez le temps) et visualiser l’erreur produite (la distance entre la cible et le déplacement effectivement produit) sur un ensemble de déplacements tests choisis au hasard (parmi des petits déplacements du cube).

Tutoriel “Interagir avec des objets dans V-REP”

<https://goo.gl/Nl4Jtz>