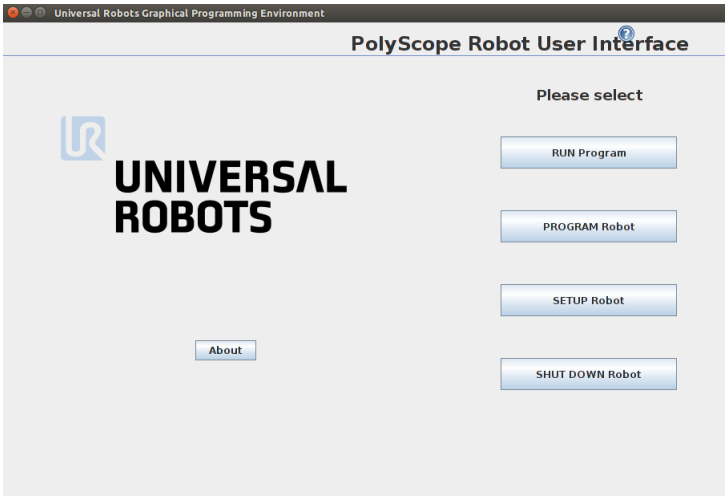

FIGURE 1 – Interface initiale d'URsim

1 Solution de contrôle du bras UR5 développée sous Windows

Le contrôle du bras UR5 sous windows présente quelques défis puisque la majorité des outils de développement, pour cet équipement, sont développés sur la plateforme Linux. Sachant que cette plateforme est souvent méconnue, la solution développée pour interagir avec le bras tentera de minimiser le travail requis avec Linux et fera le lien entre Matlab, URsim et un logiciel de visualisation ce nommant VRep. Ce dernier permet l'insertion et la visualisation de différents objets dans l'environnement du robot puisque URsim ne peut que représenter le robot lui-même. Tout d'abord, cette section expliquera l'installation de URsim et les différentes fonctionnalités utiles à la mise en place de l'infrastructure. Ensuite, le moyen de communication entre Matlab et URsim sera décrit. Finalement, l'interaction avec le logiciel de visualisation VRep sera expliqué.

1.1 URsim et son utilisation

URsim permet l'interface avec un robot virtuel ou un robot réel connecté sur le réseau. Cette caractéristique est importante puisqu'il serait relativement facile de faire un simulateur directement dans matlab, cependant ce dernier ne pourrait pas directement contrôler le bras UR5. En utilisant URsim, le robot simulé agit directement comme un vrai robot, simulant les limitations de forces, de courant et les arrêts d'urgence. Ce logiciel permet la programmation de routines de base ainsi que la définition de fonctions plus complexes permettant, entre autre, la communication UDP avec d'autre logiciels. URsim, en soit, n'est disponible que sur Linux. Cependant, sur le site web de l'entreprise Universal Robots, une machine virtuelle sur laquelle URsim est déjà installé est disponible. Ceci permet de conserver l'environnement de développement Windows en exécutant URsim dans une machine virtuelle Linux. Le logiciel avec lequel cette machine virtuelle a été utilisée pour l'environnement de développement se nomme VMware Workstation. Les instructions quant à la mise en route d'une machine virtuelle sur ce logiciel sont disponible sur le site web de VmWare. Suite au lancement de la machine virtuelle et d'URsim, l'interface initiale d'URsim vous est présentée (Figure 1).

— Le bouton *RUN Program* permet de faire fonctionner des routines contenant différentes fonctions. Ces routines sont communément appelées *URscript*. Ces scripts peuvent varier

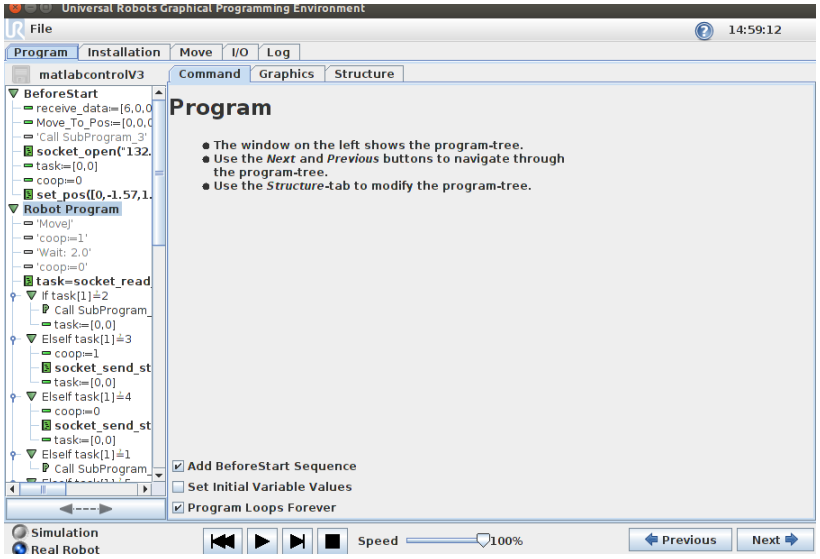

FIGURE 2 – Interface de programmation d'URsim

en complexité allant d'une simple séquence de positions à atteindre jusqu'à un programme complexe gérant de multiples sockets de communications UDP et répondant à des appels de fonctions listées dans la librairie d'URsim.

— Le bouton *PROGRAM Robot* permet la rédaction ou la modification d'URscripts

— Le bouton *SETUP robot* permet la modification de paramètres tels que l'adresse IP d'un robot réel qu'un utilisateur voudrait contrôler via l'instance d'URsim installée sur la machine virtuelle.

Quand un utilisateur charge un URscript, il appui alors sur *PROGRAM ROBOT*, puis sur *Load Program* (ou sur *Empty program* si il n'y a pas de programmes déjà disponibles). Une fois cette opération complétée la fenêtre présente à la Figure 2. Ceci constitue l'interface de programmation d'URscripts. Un manuel de programmation URscript est inclue dans l'appendice B, cependant, pour le fonctionnement de l'environnement de développement tel que présenté, aucune programmation URscript ne sera nécessaire.

1.2 Lien URsim-Matlab

Comme mentionné précédemment, les URscripts peuvent gérer des sockets de communication UDP et utiliser des fonctions tirées de la librairie d'URsim. En premier lieu, les manipulations requises dans le logiciel URsim lancé dans la machine virtuelle seront décrites. Ensuite, les étapes à suivres dans l'environnement de développement Matlab sous windows seront expliquées.

1.2.1 Manipulations dans URsim

Un URscript a été rédigé afin de permettre à URSIM de répondre à certaines commandes envoyées sous format texte via Matlab. Les fichiers URscripts en question sont disponibles dans le dossier *programs* du dépot github suivant. Pour rendre ces URscripts disponibles pour URsim, il faut seulement remplacer le dossier *programs* présent dans le répertoire d'installation d'URsim par celui téléchargé depuis le répertoire github mentionné plus haut. La figure 3 représente grosso modo l'endroit dans lequel il faut remplacer le dossier *programs*. Suivant le dépot du dossier *programs* dans

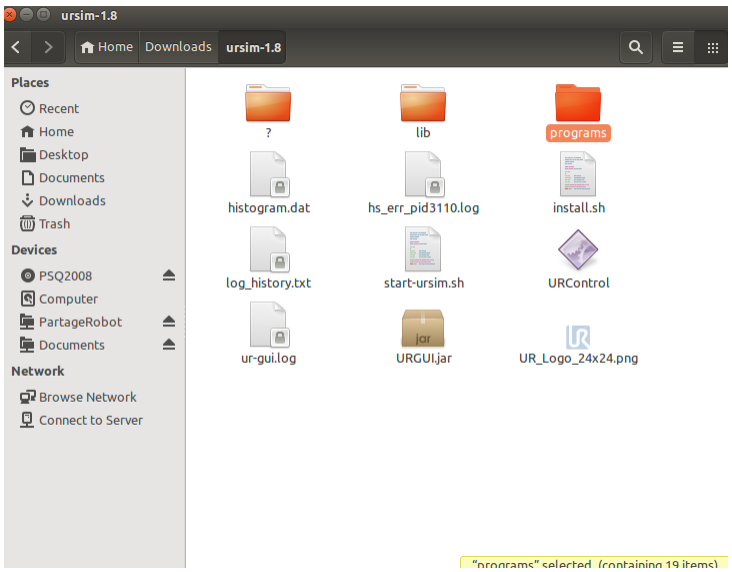

FIGURE 3 – Emplacement du dossier *programs*

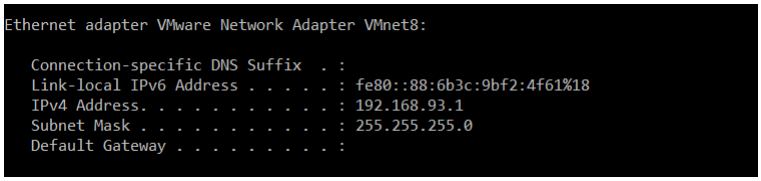

FIGURE 4 – Exemple de l'adresse ip du contrôleur de VMware qu'il faut prendre en note (IPv4 address)

le répertoire d'URsim, il est possible d'ouvrir le programme portant le nom *matlabcontrolV3.urp* après avoir sélectionné *Load Program* tel que vu dans la Figure 1.

Ensuite, il faut executer les étapes suivante :

— Modifier l'adresse IP dans la fonction *socket_open* du URscript pour qu'elle concorde avec l'adresse IP du controleur réseau de VMware. Cette infromation est disponible en ouvrant un invité de commande dans la session Windows et de tapper ipconfig. La Figure 4 représente la sortie de l'invité de commande qu'il faut remarquer lors de l'envoie de la commande ipconfig.

— Sélectionner l'option *Simulation* dans le bas de la fenêtre d'URsim.

— Appuyer sur le bouton *Play* présent de le bas, au centre, de la fenêtre d'URsim.

Ces étapes lancent l'URscript sur un robot virtuel et permet l'écoute et la réponse à des commandes envoyées par Matlab.

1.2.2 Interface Matlab-URsim

Une fois l'environnement virtuel d'URsim mis en place, l'environement matlab requis pour communiquer avec URsim doit être implémenté. Un environnement de travail matlab, situé dans le répertoire github mentionné plus haut, contient plusieurs fichiers nécessaires à la communiquation avec URsim.

Un survol rapide du URscript *matlabcontrolV3.urp* permet d'observer qu'URsim attend l'arrivée

Tableau 1 – Liste de fonctions Matlab communiquant avec URsim.

Fonction	Description
readrobotpose_j(Socket_conn)	Lecture des positions articulaires du robot.
readrobotpose(Socket_conn)	Lecture de la position à l'effecteur du robot.
moverobot(Socket_conn,Goal_Pose,Orientation)	Envoie de commande en positions articulaires.
readrobotspeed(Socket_conn)	Lecture des vitesses cartésiennes du robot.
readrobotspeed_j(Socket_conn)	Lecture des vitesses articulaires du robot.
speedrobot(Socket_conn,theta_dot)	Envoie de commande en vitesses articulaires.
getinversekin(Socket_conn,next_pose)	Calcul de la cinématique inverse du UR5.

de différents types de packets et détermine la commande à accomplir avec la valeur de la première donnée comprise dans une liste de longueur prédéterminée. Les fichiers Matlab permettant de faire le lien avec URsim font seulement traduire les arguments dans la forme qui rend possible la lecture par VRep.

Voici la liste des fonctions disponibles dans le répertoire github fourni :

Ces fonctions permettent l'appel des fonctions d'URsim listées dans l'annexe B. Une fois les packets lus par URsim, le URscript détermine quelle fonction appeler. Suivant la méthodologie utilisée, il est facile d'ajouter des fonctions au URscript et d'adapter des fonctions Matlab à la nouvelle commande à envoyer.

1.3 Logiciel de visualistion VRep

Bien que le logiciel URsim permet de représenter le robot dans un environnement 3D de base, il ne permet pas la représentation d'objets composant l'environnement de travail du robot. Pour se faire, le logiciel VRep est utilisé. Ce dernier permet d'afficher une multitude d'objets localisés dans une librairie déjà prédéfinie. Cette section couvre brièvement les étapes requises pour l'installation et l'utilisatiuon de VRep avec matlab. Il est à noter que VRep peut également servir de simulateur pour le UR5, cependant, il n'est pas aussi fidèle qu'URsim pour représenter les limitations en couple, courrant ou les arrêts d'urgence.

1.3.1 Installation de VRep

L'installation de VRep peut se faire rapidement en téléchargeant la version d'éducation sur le site web officiel suivant.

1.3.2 Interface Matlab-VRep

VRep est doté d'une API très versatile pouvant être utilisée dans plusieurs langages de programation tels que C/C++, Python, Matlab, Java, Octave et Lua. Pour avoir accès aux fonction de l'API dans un projet Matlab, il faut aller chercher les fichiers suivants :

— remApi.m
— remoteApiProto.m
— remoteApi.dll

Ces fichiers sont situés dans les dossiers d'installation de VRep. Sur le système d'exploitation Windows, cet endroit est normalement :

C :\Program Files (x86)\V-REP3\V-REP_PRO_EDU\programming\remoteApiBindings\matlab\matlab pour le fichier ".m" et :
C :\Program Files (x86)\V-REP3\V-REP_PRO_EDU\programming\remoteApiBindings\lib\lib\64Bit pour le fichier ".dll".

Suite à l'incorporation de ces fichiers dans l'environnement de travail, il est possible d'initialiser l'objet interagissant avec l'API dans matlab avec cette commande :

```
vrep=remApi('remoteApi');
```

Ensuite, il est possible d'envoyer des commandes à une instance de VRep. Dans l'environnement du simulateur, l'interaction du script matlab avec VRep se résume à l'envoie des positions articulaires lues dans URsim.

1.4 Exemple d'utilisation

Les étapes suivantes réfèrent aux scripts matlab situés dans le dossier *URsim-vrep-matlab* du dépot github mentionné plus haut. L'utilisation de la solution présentée dans cette section peut être résumée par les étapes suivantes :

- Connection d'un joystick dans l'ordinateur (requis pour l'exemple).
- Ouverture de la machine virtuelle et démarage d'URsim
- Chargement du URscript dans URsim.
- Ouverture de VRep sur la session windows.
- Chargement de la scène *scene_ur5_1.ttt* dans VRep.
- Démarage du script matlab nommé *Exemple_controle_ur5_vrep.m*.
- Appuyer sur le bouton "play" de URsim

Suite à l'exécution de ces étapes, le robot est contrôlable avec le joystick et devrait être visualisé dans VRep. La Figure 5 représente l'interface VRep lorsque la scène est chargée. Pour plus de détails quant à l'utilisation des fonctions dans matlab, il est possible de lire le code exemple qui a été commenté pour faciliter la compréhension. De plus, l'annexe C est constituée d'un document listant la magoritée des commandes disponibles.

1.5 Conclusion

La solution de contrôle du bras UR5 avec matlab à été présentée. L'installation et l'utilisation de URsim dans une machine virtuelle à été abordée. Cependant la nécessité d'utiliser une machine virtuelle ainsi qu'un logiciel de visualisation tel que VRep est quelque peu compliquée.

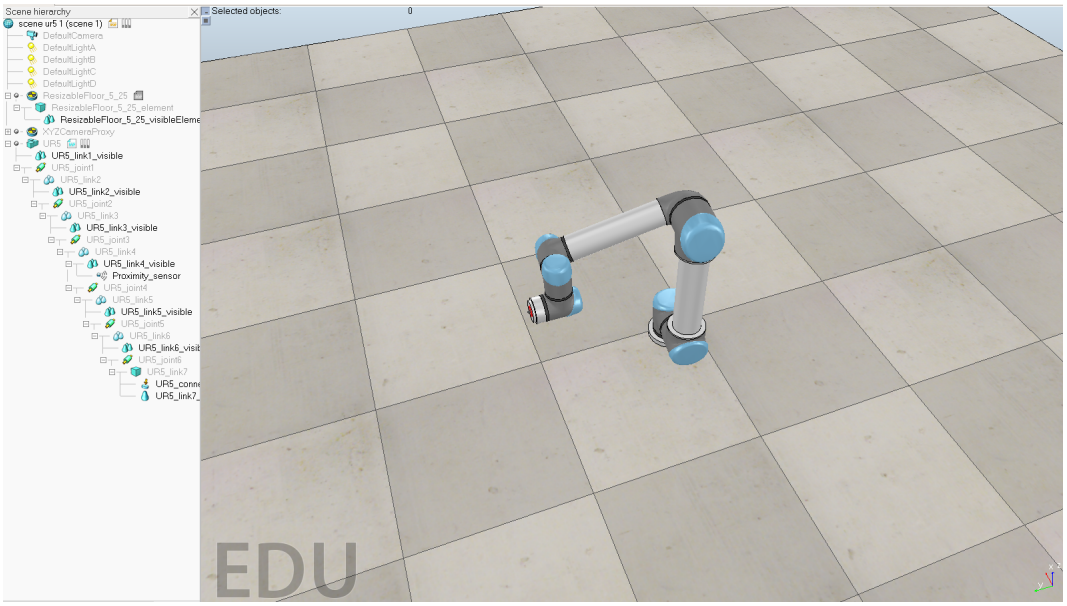

FIGURE 5 – Interface du simulateur VRep avec la scène chargée.