

IMERIR / NovaLynx

Projet d'intégration 3ème Année

Apprentissage de trajectoires via une souris 3D

02/11/2015 - Document Révision 1.0



Chef de projet : Sébastien Bach / Rémi Parlouar
Contact : 05 82 95 22 95
contact@novalynx.fr

TABLE DES MATIERES

I.	Introduction.....	3
1.	NovaLynx : Intégrateur d’innovations	3
2.	Contexte	3
II.	Le Projet.....	4
1.	Description du projet.....	4
2.	Objectifs.....	4
II.2.1	Phase 1 - Connexion aux différents équipements.....	5
II.2.2	Phase 2 – Asservissement du robot avec la souris 3D.....	6
II.2.3	Phase 3 – Apprentissage et lecture de trajectoires / instructions robot	6
II.2.4	Phase 4 – Apprentissage et gestion d’un stock	6
III.	Le Matériel.....	7
1.	Robot KR 6 R900 sixx (KR AGILUS).....	7
2.	Armoire de commande KRC4 Compact	7
3.	Commande robot SmartPad	7
4.	3D Connexion SpaceMouse Pro	8
IV.	Détails des fonctions de la DLL	8

I. INTRODUCTION

1. NOVALYNX : INTEGRATEUR D'INNOVATIONS

De nos jours, les entreprises sont de plus en plus soumises à la concurrence, et la question de la compétitivité devient un sujet récurrent. De nombreux modèles et méthodes de gestion sont apparus, afin d'apporter des solutions à la recherche de trois objectifs bien connus : réduire les coûts, diminuer les délais et augmenter la qualité. La pensée "Lean", par exemple, amène à se focaliser sur la valeur ajoutée et l'amélioration continue, et de nombreuses entreprises opèrent des changements importants dans leur organisation pour adopter cette pensée.

L'efficacité de ces méthodes n'est plus à démontrer, mais le succès de leur mise en œuvre implique généralement d'avoir :

- ✓ Une vision synthétique, globale et temps réel de son activité,
- ✓ Un contrôle rapide et fiable sur les moyens et les outils de l'entreprise,
- ✓ Une capacité d'évolution et une réactivité permettant de s'adapter efficacement aux changements.

NovaLynx conçoit, développe et commercialise une gamme de produits et de solutions qui répondent à ces besoins. Grâce à une expertise dans les domaines de l'informatique, du réseau et de l'électronique, nous proposons un ensemble de périphériques autonomes, intelligents et connectés. L'expérience utilisateur est au centre de nos préoccupations, nous sommes donc soucieux de concevoir des produits utiles, fiables et simples d'utilisation. Autant d'éléments qui permettent d'optimiser les différentes fonctions d'une entreprise dans un contexte d'agilité, de qualité et d'innovation technologique.

2. CONTEXTE

Dans le cadre d'un projet étudiant de 3ème année, IMERIR a demandé à NovaLynx de créer un projet ayant pour thématique la robotique collaborative. Ce document est un support technique, vous y trouverez les spécifications fonctionnelles de la cellule robot du projet, son architecture et ses spécifications techniques.

Nous vous accompagnerons la première journée pour la mise en service du robot et le lancement du projet. Nous resterons bien entendu disponibles pendant toute la durée du projet directement par téléphone ou par email pour répondre à toutes vos questions.

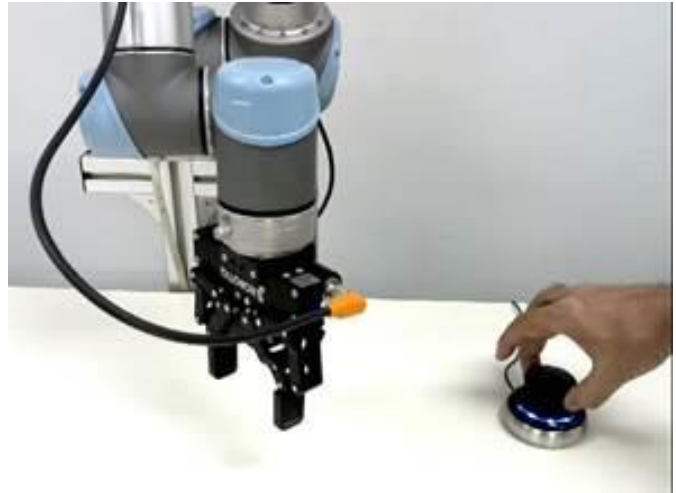
II. LE PROJET

1. DESCRIPTION DU PROJET

Vous devrez créer au moyen des éléments qui vous seront fournis, une solution d'apprentissage et de gestion de stock robotisé. C'est un projet très innovant, KUKA travaille en partenariat avec NovaLynx pour effectuer de la recherche sur ce sujet.

Dans un premier temps, vous devrez créer un programme qui permet de manipuler le robot grâce à une souris 3D.

Ensuite, en manipulant le robot au moyen de la souris, le même programme devra permettre l'apprentissage et la lecture de points/trajectoires dans le but de gérer un stock de pièces.



Les **avantages** de ce projet sont les suivants :

- Vous permettre de **manipuler un robot simplement** via une DLL bien documentée. Il n'y a aucune intégration à faire, notre librairie exposera via des classes très claires les différents éléments de la cellule (robot, pince, souris 3D).
- Vous sensibiliser à la **cobotique** (coopération d'un robot et d'un humain).
- Créer un asservissement afin de synchroniser les mouvements du robot et de la pince pour créer des actions.
- **Manipuler les différents concepts 3D** (ex : récupérer le vecteur de force de la souris 3D et appliquer un déplacement sur le robot).
- Utiliser le filtrage et le **traitement du signal** (il faut simplement normaliser et borner les valeurs que renvoie la souris 3D).
- **Créer un processus industriel** en créant une machine à état et gérant une librairie de mouvement.

2. OBJECTIFS

Vous devez développer (en C# ou autre, langage au choix...) une application qui permettra l'apprentissage et la relecture de trajectoires. Notre API vous permettra d'être immédiatement connectés au robot, à la pince et à la souris 3D. Votre objectif est d'écrire le code métier et de superviser les différents éléments pour répondre au besoin d'apprentissage et de gestion d'un stock.

Afin de procéder pas à pas, nous avons décomposé ce projet en plusieurs phases :

Document Confidentiel

Tous les textes, titres et images de ce document sont la propriété de NovaLynx SAS.

II.2.1 Phase 1 - Connexion aux différents équipements

La première étape de ce projet est de se connecter aux différents équipements et de les piloter de manière indépendante.

Objectif de la phase :

- Se connecter au robot et le piloter
- Se connecter à la souris et récupérer les données
- Gérer les différents actionneurs / capteurs de la cellule

II.2.2 Phase 2 – Asservissement du robot avec la souris 3D

Après avoir pris le contrôle de tous les éléments de la cellule, vous devrez créer un programme qui transforme les forces appliquées sur la souris 3D en déplacement pour le robot (le tout en temps réel bien sur...).

Objectif de la phase :

- Manipuler le robot en temps réel grâce à la souris 3D

II.2.3 Phase 3 – Apprentissage et lecture de trajectoires / instructions robot

Bien joué ! Vous pouvez maintenant faire bouger le robot au moyen de la souris. Dans cette phase, votre objectif est d'alterner le déplacement du robot au moyen de la souris et la capture de sa position.

Après avoir capturé plusieurs points, vous devrez pouvoir rejouer les points appris à la suite afin de former une trajectoire.

Une fois les trajectoires maîtrisées, séquencer la lecture de trajectoires et d'instructions pour créer un programme complet de prise et de dépose d'une pièce, intégrant l'utilisation de la pince.

Objectif de la phase :

- Enregistrer des trajectoires (point clé) en capturant la position du robot
- Rejouer les trajectoires apprises
- Créer un programme qui déplace une pièce en mélangeant des trajectoires et des instructions.

II.2.4 Phase 4 – Apprentissage et gestion d'un stock

Si vous avez atteint les trois premiers objectifs, vous devrez mettre en place un programme d'apprentissage et de gestion de stock de pièces. Pour cela il faut :

- Apprendre des points clés pour déterminer la dimension du stock
- Appliquer des transformations sur les points enregistrés pour déduire les positions des autres pièces

Attention ! Afin d'éviter les collisions, vous devrez impérativement mettre en place une stratégie de mémorisation du nombre de pièces placées, de la pièce courante, et du nombre de pièces restantes.

Objectif de la phase :

- Apprentissage et définition d'un stock de pièces dans l'espace (plan dans l'espace, nombre de colonnes et nombre de lignes).
- Attraper et déplacer les pièces grâce à la pince dans le stock appris.

III. LE MATERIEL

Pour mener à bien ce projet, vous aurez à votre disposition le matériel suivant :

- un robot KUKA KR6 Agilus et sa cellule
- une souris 3D de la marque 3D Connexion
- une pince pneumatique
- un capteur inductif pour la détection des pièces

1. ROBOT KR 6 R900 SIXX (KR AGILUS)



C'est un robot manipulateur 6 axes ayant un rayon d'action de 901 millimètres avec une charge utile maximale de 6 kg en bout de bras et une répétabilité de l'ordre de $\pm 0,03\text{mm}$.

Il s'agit d'un robot industriel de haute précision particulièrement adapté aux tâches de manipulation de petites pièces. Plus de détails concernant ce robot peuvent être trouvés dans la documentation du constructeur ci-jointe.

2. ARMOIRE DE COMMANDE KRC4 COMPACT

Le robot est piloté grâce à l'armoire de commande standard fournie par KUKA pour ce type de robot, une armoire de commande KRC4 compact. Elle comporte tous les éléments permettant le contrôle du robot, ainsi qu'un ensemble d'entrées/sorties qui permettront de faire communiquer le robot avec les autres éléments du système (pince, boucle de sécurité, capteur inductif...). Plus de détails concernant cette armoire peuvent être trouvés dans la documentation constructeur jointe.



3. COMMANDE ROBOT SMARTPAD



L'armoire de commande KRC4 compact est équipée de série avec un contrôleur tactile très ergonomique, le Kuka SmartPad. Cette interface permet le contrôle total de l'armoire de commande, en offrant la possibilité de démarrer des programmes, de déplacer manuellement le robot, de piloter manuellement des périphériques (ex : pince), ou encore de visualiser la position courante du robot.

4. 3D CONNEXION SPACEMOUSE PRO

Cette souris 3D de 3Dconnexion, intègre le capteur 3Dconnexion breveté offrant 6 degrés de liberté, permettant une navigation facilitée au sein de modèles numériques, positions de caméra ou pilotage de robot dans les espaces 3D.

Il suffit de pousser, tirer, incliner ou faire tourner la molette du contrôleur 3Dconnexion pour réaliser simultanément des panoramiques, zooms et rotations de votre modèle ou de votre caméra, avec précision et de manière intuitive.



IV. DETAILS DES FONCTIONS DE LA DLL

Descriptions des fonctions disponibles dans la librairie NLX.Robot.Kuka.Controller.dll:

```
public RobotController()  
Default constructor.
```

```
public void Connect(string robotIpAddress)  
Connects the controller to the robot. The robot program must be running.
```

```
public void StartRelativeMovement()  
Starts relative movement mode.
```

```
public void SetRelativeMovement(CartesianPosition relativePosition)  
Sets the relative transformation to apply.
```

```
public void StopRelativeMovement()  
Stops relative movement mode.
```

```
public void PlayTrajectory(List<CartesianPosition> trajectory)  
Plays a trajectory defined by a list of cartesian positions.
```

```
public void OpenGripper()  
Opens the robot gripper.
```

```
public void CloseGripper()  
Closes the robot gripper.
```

```
public void IsGripperOpen()  
Reads the gripper state.
```

```
public bool ReadSensor()  
Reads the inductive sensor.
```



```
public CartesianPosition GetCurrentPosition()  
Returns the current position of the robot.
```