

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Ecole Nationale Polytechnique



Département d'Automatique
Laboratoire de Commande des Processus



Thèse de Doctorat en Sciences

Thème

Conception et Réalisation d'un Robot Manipulateur à 4 d.d.l dédié à la Télé-Opération en Réalité Augmentée

Fatim Zohra Zendaoui

Sous la direction de Mr Mnd Oulhadj MAHMOUDI, Professeur

Présentée et Soutenue publiquement le : 06/09/2020

Composition du Jury

Président	Mr M ^d Seghir BOUCHERIT	Professeur	ENP
Rapporteur/Promoteur	Mr M nd Oulhadj MAHMOUDI	Professeur	ENP
	Mr FEU Rachid OUIGUINI	MRA	COMENA
Examineurs	Mr Farid BOUCHAFA	Professeur	USTHB
	Mr Mohamed TADJINE	Professeur	ENP
	Mr Said TOUATI	DR	CRNB Birine
	Mr Abdelhakim CHERFIA	MCA	Univ Frères Mentouri Cne1

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Ecole Nationale Polytechnique



Département d'Automatique
Laboratoire de Commande des Processus



Thèse de Doctorat en Sciences

Thème

Conception et Réalisation d'un Robot Manipulateur à 4 d.d.l dédié à la Télé-Opération en Réalité Augmentée

Fatim Zohra Zendaoui

Sous la direction de Mr Mnd Oulhadj MAHMOUDI, Professeur

Présentée et Soutenue publiquement le : 06/09/2020

Composition du Jury

Président	Mr M ^d Seghir BOUCHERIT	Professeur	ENP
Rapporteur/Promoteur	Mr M nd Oulhadj MAHMOUDI	Professeur	ENP
	Mr FEU Rachid OUIGUINI	MRA	COMENA
Examineurs	Mr Farid BOUCHAFA	Professeur	USTHB
	Mr Mohamed TADJINE	Professeur	ENP
	Mr Said TOUATI	DR	CRNB Birine
	Mr Abdelhakim CHERFIA	MCA	Univ Frères Mentouri Cne1

Tell me and I will forget,
Show me and I will remember,
Involve me and I will understand.
Chinese Proverb

Remerciements

Tout d'abord, je remercie "**ALLAH**" le tout puissant, qui m'a gratifié de santé, de moyens et de volonté ardente **dans la quête du savoir**.

Mes remerciements s'adressent aussi, et en particulier, au **Pr Mahmoudi Mnd Oulhadj**, qui, en tant que Directeur de thèse, s'est toujours montré très disponible tout au long de la réalisation de ce travail, ainsi que pour l'aide compétente qu'il m'a apporté, pour sa patience et son encouragement à finir ce travail. Son œil critique m'a été très précieux pour structurer le travail et pour améliorer la qualité des différentes sections.

Au même titre, je tiens également à remercier le Docteur feu **Ouiguini Rachid**, qui, en tant que Co-Directeur de thèse, a su conduire mes recherches et m'accompagner tout au long de ces années en combinant, de façon subtile, les critiques et les encouragements dans les moments difficiles. Merci donc pour votre soutien sans faille, votre confiance et votre disponibilité. Que Dieu vous accueille dans son vaste paradis (الله يرحمك ويجعل مثواك الجنة).

Un grand merci à Monsieur **Zaatri Abdelouahab**, Professeur à l'université Mentouri Constantine, pour sa générosité, l'accueil dans son laboratoire et l'inspiration dont j'ai bénéficié grâce à ses connaissances et son dévouement pour le travail.

Que soit également remercié ici Monsieur **Nezli Lazhari**, Professeur à l'ENP, qui m'a souvent encouragée à continuer à travailler et à qui je dois de précieuses indications sur les démarches à suivre.

Aussi, c'est avec un profond bonheur que je saisis l'occasion pour adresser mes plus sincères remerciements à l'ensemble des membres du jury pour l'intérêt qu'ils ont porté à mon travail en acceptant de l'examiner et de l'enrichir par leurs propositions et critiques:

Je tiens à exprimer ma profonde gratitude à Monsieur **Boucherit Mohamed Seghir**, Professeur à l'ENP, pour m'avoir fait l'honneur de présider le jury d'examen de cette thèse;

Toute ma reconnaissance va à Monsieur **Tadjine Mohamed**, Professeur à l'ENP, pour avoir accepté d'examiner mon travail de thèse, et aussi pour ses précieux conseils;

Je remercie vivement Monsieur **Cherfia Abdelhakim**, Maître de Conférences à l'Université Frères Mentouri Constantine 1, d'avoir accepté de participer dans le jury de ma thèse.

Mes remerciements vont aussi à Monsieur **Bouchafa Farid**, Professeur à l'USTHB, d'avoir accepté de faire partie du jury d'examen de ma thèse;

Je remercie également Monsieur **Touati Said**, Directeur de recherche au CRNB pour sa participation au jury d'examen de ma thèse.

Qu'ils trouvent ici ma gratitude d'avoir accepté la charge de juger ce travail. Leurs compétences, chacun dans son domaine, font que leur présence est pour moi une véritable caution scientifique. Je les remercie chaleureusement pour leur dévouement.

Je tiens à remercier Monsieur **Messai Adnane**, Maître de recherche, de m'avoir encouragé et soutenu dans les moments difficiles. Sans oublier toutes les personnes qui ont contribué de près ou de loin à l'avancement du projet lié à cette thèse.

Enfin, je remercie ceux qui ont veillé sur moi depuis toujours, ceux qui m'ont fait confiance, qui m'ont soutenu sans faille dans tous mes projets et qui ont accepté mes choix sans pour autant toujours forcément les comprendre, merci ma famille !

ملخص

التشغيل عن بعد يسمح للمشغل التحكم في الوقت الفعلي وعن بعد لنظام رباتي أداء مهمته باستخدام واجهة رسومية. يجب أن توفر هذه الواجهة إمكانيات مختلفة لإدارة صعوبات المسافة بين الروبوت وموقع التحكم وهذا يزيل مخاطر الأعمال الخطيرة كالتعامل مع المواد النووية. وبالنسبة للأنظمة الروبوتية عن بعد، غالبا ما تعتبر الحلقة البصرية حلقة ردود الفعل الرئيسية. بشكل عام، يتلقى المشغل الصور التي تحتوي على المعلومات اللازمة لأداء المهمة في الموقع البعيد. لمساعدتها على إنجاز مهمتها، يمكن لأنظمة التشغيل عن بعد دمج أدوات التصور الرسومية وأجهزة تفاعل الأنظمة المختلفة لتخطيط المهام وما إلى ذلك.

هذه الأطروحة تقدم العمل المنجز في مجال التشغيل عن بعد. في هذا المنظور، قمنا بتنفيذ تشغيل عن بعد لذراع قابل للمناورة بثلاثة درجات من الحرية "SEROR"، من خلال شبكة محلية وجهاز التحكم عن بعد على مسافة 430 كم عبر الإنترنت. وعلى ذلك، ظهرت بعض الظواهر فيما يتعلق بجودة الشبكة والتأخير في نقل الصور بين الموقع الرئيسي وموقع التحكم "تأخير"، زيادة على عناصر أخرى. للتغلب على هذه الظواهر، قررنا تحسين التفاعل بين المشغل وبيئة التشغيل، من خلال دمج أدوات مثل الواقع الافتراضي 'RA' والواقع المعزز 'RV' وتقنيات المحاكاة. يسمح الأسلوبان RV و RA بالحصول على واجهة طبيعية وبديهية، ودمج مصادر مختلفة من المعلومات لتحسين إدراك المشغل. المحاكاة من جانبها، تجعل من الممكن تحسين تصميم نموذج الروبوتات قبل تنفيذها "النماذج الأولية". شرعنا في البداية في تصميم نموذج تجريبي للروبوت بأربع درجات من الحرية، ثم تحليل سلوكه باستخدام برامج النمذجة والمحاكاة المستندة إلى منصة SimMechanics و Simulink من Matlab. بعد ذلك قمنا بانجاز الروبوت ولوحة التحكم الخاصة به استنادًا إلى المتحكم Pic18F452. تم العمل السابق من أجل السماح بتنفيذ الواقع المعزز للتشغيل عن بعد. تمت دراسة عدة طرق بما في ذلك التعليق التوضيحي، المستخدم بشكل عام في البرنامج النووي، لأنه يوفر للمشغل أنواعًا مختلفة من المعلومات من نظام رؤية الكاميرا، مما يضمن سلامته وكذلك دقة مهمة المعالجة.

الكلمات الدالة: التشغيل عن بعد، الواقع الافتراضي، الواقع المعزز، التفاعل، رؤية، SimMechanics، Simulink، النماذج الأولية.

ABSTRACT

A tele-operation system allows the operator to remotely control in real time, a robotic system to perform a task using a graphical interface. This interface must offer different possibilities to manage the distance difficulties between the robot and the control site. This eliminates the risks associated with hazardous work such as handling nuclear substances.

For tele-robotic systems, the visual system is often considered as the main feedback loop. Generally, the operator receives images containing the information necessary to perform the tasks in remote site. To help him to accomplish its task, tele-operation systems can integrate graphical visualization tools, different systems interaction devices for task planning, etc.

This thesis presents a work accomplished in the field of tele-operation. In this perspective, we implemented the tele-operation of a manipulator arm with three degrees of freedom "SEROR", through a local network and remotely on 430km via Internet. Nevertheless, certain phenomena appeared; related to the quality of the network, the delay in the transmission of images between the master site and the slave site "Delay", in addition to other parameters. To overcome these phenomena, we decided to improve the interaction between the operator and the operations environment, by integrating tools such as virtual reality 'RV', or augmented 'RA' and simulation. The two techniques RV and RA allow to have a natural and intuitive interface, and to integrate different sources of information to improve the perception of the operator.

Simulation, for its part, makes it possible to better design prototype robots before their implementation "Prototyping". Firstly, we designed a robot prototype with 4 degrees of freedom; analyze its behavior using a modeling and simulation software based on Matlab's SimMechanics and Simulink platform. Then we realized the robot and its control board based on the microcontroller Pic18F452. The previous work was done to allow the implementation of AR for tele-operation. Several methods have been studied including annotation, commonly used in the nuclear program, because it provides the operator different types of information issue from a camera's vision system, thus ensuring his security and accuracy of the handling task.

Key Words: Tele-operation, Virtual reality, Augmented Reality, Interaction, Vision, SimMechanics, Simulink, Prototyping.

Résumé

La télé-opération permet à un opérateur le contrôle en temps réel et à distance, d'un système robotique pour la réalisation d'une tâche au moyen d'une interface graphique. Cette interface doit offrir diverses possibilités pour gérer les difficultés d'éloignement entre le robot et le site de contrôle. Ce qui écarte les risques relevant des travaux dangereux tels que la manipulation de substances nucléaires.

Pour les systèmes télé-robotiques, la boucle visuelle est souvent considérée comme la boucle principale de feedback. Généralement, l'opérateur reçoit des images contenant les informations nécessaires à la réalisation de la tâche du site distant. Pour l'aider à accomplir sa tâche, les systèmes de télé-opération peuvent intégrer des outils de visualisation graphique, différents dispositifs d'interaction des systèmes pour la planification des tâches, etc.

Cette thèse présente un travail accompli dans le domaine de la télé-opération. Dans cette perspective, nous avons mis en œuvre la télé-opération d'un bras manipulateur à trois degrés de liberté "SEROR", à travers un réseau local et à distance sur 430km via Internet. Néanmoins, certains phénomènes ont apparus ; liés à la qualité du réseau, au retard dans la transmission des images entre le site maître et le site esclave "Delay", en plus d'autres paramètres. Pour pallier à ces phénomènes, nous avons décidé d'améliorer l'interaction entre l'opérateur et l'environnement des opérations, en intégrant des outils comme la réalité virtuelle 'RV', ou augmentée 'RA' et la simulation. Les deux techniques RV et RA permettent d'avoir une interface naturelle et intuitive, et d'intégrer différentes sources d'information pour améliorer la perception de l'opérateur. La simulation quant à elle, permet de mieux concevoir des prototypes de robots avant leur mise en œuvre "Prototypage". Dans un premier temps, nous avons conçu un prototype de robot à 4 degrés de liberté, analyser son comportement à l'aide d'un logiciel de modélisation et de simulation basé sur la plateforme SimMechanics et Simulink de Matlab. Ensuite nous avons réalisé le robot et sa carte de commande à base du microcontrôleur Pic18F452. Le précédent travail avait pour but de permettre la mise en œuvre de la RA pour la télé-opération. Plusieurs méthodes ont été étudiées, notamment l'annotation, généralement utilisée dans le programme nucléaire, car elle fournit à l'opérateur différents types d'informations provenant du système de vision de la caméra, garantissant ainsi sa sécurité et la précision de la tâche de manipulation.

Mots clés : Télé-opération, Réalité virtuelle, Réalité augmentée, Interaction, Vision, SimMechanics, Simulink, Prototypage.

TABLE DES MATIERES

1. Introduction générale

1.1 Contexte de la thèse	18
1.2 Motivation.....	18
1.3 Contributions	18
1.4 Structure du manuscrit.....	19

2 Etat de l'art Robotique, Télé-opération, Réalité Virtuelle, Réalité Augmentée, Nucléaire

2.1 Introduction.....	23
2.2 La robotique.....	23
2.2.1 Historique de la robotique télé-opérée ²	24
2.2.2 Programmation des robots	25
2.3 La Télé-opération.....	25
2.3.1 Bref historique	25
2.3.2 Facteurs pouvant influencer la télé-opération.....	26
2.4 Systèmes robotiques basés sur Internet pour la télé-opération	26
2.4.1 Introduction.....	26
2.4.2 Delay sur Internet.	27
2.5 Utilisation de la réalité virtuelle sur l'interface de contrôle	27
2.5.1 Principe de la réalité virtuelle 'RV'	28
2.5.2 L'histoire de la RV ³	28
2.6 Utilisation de la réalité augmentée pour la télé-opération	31
2.6.1 Définition 18.....	31
2.6.2 Définition ² 10.....	31
2.6.3 Travaux réalisés dans le domaine	33
2.7 Milieux hostiles	37
2.7.1 Définition.....	37
2.7.2 Application des systèmes robotiques en milieu hostiles.....	37
2.7.2.1 Introduction.....	37
2.7.2.2 Utilisations en milieu nucléaire	38
2.7.2.3 Entretien et maintenance des sites nucléaires	39
1. Entretien préventif	40
2. Entretien correctif	40

3. Maintenance des sites nucléaires	40
4. Gestion des déchets.....	42

3. Application de la Réalité Augmentée dans la télé-opération du robot

3.1 Introduction.....	44
3.2 Objectif de la RA	44
3.3 Réalité augmentée et vision artificielle.....	45
3.3.1 Utilisation des algorithmes de traitement d'images pour l'extraction des grandeurs d'objets réels dans l'environnement des opérations	46
3.3.2 Essais en laboratoire concernant la mesure des grandeurs et de la profondeur d'objets dans l'image de la scène utilisant la vision	47
3.3.3 Calibration géométrique	48
3.3.3.1 Calibration de la caméra (estimation de ses paramètres intrinsèques et extrinsèques	48
3.3.3.2 Paramètres intrinsèques et extrinsèques de la caméra	49
3.3.3.3 Mesure de distances 2D	49
3.4 Apport de la réalité augmentée	51
3.5 Interaction : Télé-opération et Réalité Augmentée.....	52
3.6 Application de la RA	52
3.7 Domaines d'application de la réalité augmentée	53
3.7.1 La réalité augmentée au service de la sécurité nucléaire	53
3.7.2 Le domaine médical.....	53
3.8 Conclusion	54

4 Mise en œuvre de la télé-opération du robot manipulateur expérimental "SEROR" via le réseau Internet.

4.1 Introduction.....	56
4.2 Objectif du chapitre	56
4.3 Architecture d'un système de contrôle via Internet.....	57
4.4 Télé-opération via Internet du robot SEROR	58
4.4.1 Description du système opérationnel.....	58
4.4.2 Interface graphique "GUI"	59
4.4.3 Intercommunication via les Sockets	60
4.4.4 Commande du bras manipulateur	61
4.4.5 Intégration de l'opérateur humain	62

4.4.6	Modélisation du robot SEROR.....	63
4.4.6.1	Modèle géométrique direct (MGD)	63
4.4.6.2	Modèle géométrique inverse (MGI).....	65
4.4.6.3	Simulation dans l'espace 2D	66
4.4.7	Modélisation cinématique	66
4.4.7.1	Modèle cinématique direct (MCD)	66
4.4.7.2	Modèle cinématique inverse (MCI)	67
4.4.8	Modélisation dynamique.....	67
4.4.8.1	Modèle dynamique direct (MDD).....	67
4.4.8.2	Modèle dynamique inverse (MDI).....	67
4.4.9	Expérimentation du robot	67
4.4.9.1	Introduction	67
4.4.9.2	Essais.....	68
4.4.9.2.1	Contrôle du robot en vision directe (opérateur humain présent sur le site de travail).....	68
4.4.9.2.2	Contrôle du robot sans vision directe (Opérateur distant): Commande par Internet.....	69
	I Contrôle basé sur un clic de souris	69
	II Contrôle basé sur l'image	69
	III Contrôle basé sur le gestuel	70
4.5	Conclusion	70

5 Conception et Simulation d'un robot manipulateur à quatre degrés de liberté utilisant Autodesk/Inventor et la plateforme SimMechanics de Matlab

5.1	Introduction.....	73
5.2	Développement d'un environnement de modélisation et de simulation	74
5.3	Présentation des outils de modélisation et de simulation	75
5.3.1	Le logiciel Autodesk/Inventor	76
5.3.2	Le logiciel Matlab	76
5.3.3	Simulink de Matlab	76
5.3.4	Simscape de Matlab.....	77
5.3.5	Blocs SimMechanics employés.....	77
5.3.5.1	Description du module SimMechanics	77
5.3.5.2	Description des fonctions des blocs utilisés dans le programme SimMechanics.....	78
5.4	Modélisation du robot.....	79

5.4.1	Exportation du robot conçu dans Inventor vers Matlab	80
5.4.2	Etapes suivies dans l'exportation du modèle de robot.....	80
5.4.2.1	Principe de la méthode d'exportation	80
5.4.2.2	Exporter le robot manipulateur dans les formats de 1ère génération	81
5.4.2.3	Exporter dans les formats de seconde génération.....	82
5.4.2.4	Blocs interconnectés constituant le robot manipulateur	84
5.4.2.5	Système de joint 'Revolute'	85
5.4.2.6	Représentation des différents blocs en SimMechanics.....	86
5.4.3	Simulation du robot en environnement SimMechanics.....	86
5.4.4	Modélisation de la pince	87
5.5	Conclusion	88

6 Réalisation du robot manipulateur à 4 ddl Associé à sa carte de commande basée sur le Microcontrôleur PIC18F452

6.1	Introduction et objectif	90
6.2	Architecture matérielle de l'ensemble du système	91
A.	Un ordinateur équipé d'un logiciel de modélisation, de simulation et de pilotage	91
B.	Le robot manipulateur.....	91
6.3	Circuit de commande des moteurs.....	91
6.3.1	Modélisation mathématique d'un moteur électrique.....	92
6.3.2	Asservissement PID	95
6.3.3	Réglage des coefficients du PID	95
6.4	Réalisation et mise en œuvre du robot manipulateur.....	97
6.4.1	Développement de la structure mécanique	97
6.4.1.1	Caractéristiques techniques de la pince.....	99
6.4.1.2	Caractéristiques techniques des servomoteurs de la pince.....	100
1	Servo standard (pour changer l'inclinaison du préhenseur)	100
2	Mini servomoteur (pour ouvrir et fermer le préhenseur)	100
6.4.2	Développement de la carte de commande	100
6.4.2.1	Description de la carte de commande	100
6.4.2.2	Le Microcontrôleur	102
1	Choix du PIC18F452.....	102
2	Programme Assembleur	103
6.4.2.3	Emulateur MPLAB-REAL-ICE.....	103
6.4.2.4	Interface série (MAX232)	104

6.4.3 Développement de la carte de puissance	104
6.4.3.1 Génération des signaux de commande	105
6.4.3.2 Circuit de décodage	106
6.4.3.3 Circuit de protection du courant	107
6.4.4 Principales fonctions gérées par le microcontrôleur	108
6.4.4.1 Initialisation des variables et périphériques	108
6.4.4.2 Réception de la commande et tests d'erreurs	109
1 Frame errors (FERR).....	109
2 Overrun errors (OERR)	109
3 Dépassement de temps (time out)	109
4 Vérification checksum.....	109
6.4.4.3 Décodage de la commande.....	109
6.4.5 Commande des moteurs	109
6.4.5.1 Courbes de position du moteur 29:1 pour une régulation on/off	111
6.4.5.2 Courbes de position du moteur 29:1 pour une régulation proportionnelle	111
6.5 Essais réels de télé-opération du robot	112
6.5.1 Plateforme de la télé-opération	112
6.5.2 Débogage et simulation.....	114
6.5.3 Calcul de la PWM de commande des actionneurs.....	114
6.5.4 Forme du signal PWM appliqué aux actionneurs du robot.....	115
6.5.5 Forme du signal PWM appliqué aux servomoteurs	116
6.6 Application des signaux PWM réels et commande du robot	117
6.6.1 Signaux PWM réels délivré par le microcontrôleur pour la commande des moteurs	118
6.6.2 Signal PWM réel délivré par le microcontrôleur pour la commande du moteur dans les deux sens	118
6.6.3 Signal réel à la sortie de l'encodeur du moteur, sorties A et B	119
6.6.4 Signal PWM réel délivré par le microcontrôleur pour la commande des servomoteurs	120
6.6.5 Signal PWM réel relevé sur la ligne de commande des servomoteurs	120
6.6.6 Evolution en temps réel de la position de l'effecteur du robot	121
6.6.7 Evolution en temps réel de la vitesse des moteurs du robot.....	123
6.7 Comparaison des résultats pratiques avec la simulation.....	123
6.7.1 Forme du signal de courant dans les moteurs	123
6.7.2 Forme du signal de la position de l'effecteur	124

6.8 Séquences vidéo de fonctionnement du robot manipulateur (réalisation d'une tâche de télé-opération type pick-and-place).....	125
6.9 Génération de trajectoire en ligne	126
6.9.1 Effectuer une trajectoire de Robot.....	126
6.9.1.1 Utilisation de la boîte à outils de Matlab	127
6.9.1.2 WaypointTrajectory de Matlab	130
6.9.1.3 Utilisation du logiciel de poursuite "Tracker"	131
1. Traitement des données	132
2. Mise en œuvre du Tracker.....	132
3. Exemples d'exploitation des résultats réels	133
6.10 Conclusion.....	135

7 Conclusion générale & Perspectives

7.1 Conclusion	136
7.2 Perspectives	138
Bibliographie	139

Annexe

Annexe A 1 Cartes de puissance et de commande du robot manipulateur	148
Annexe A 2 Programme en assembleur de toutes les fonctions de commande liées au microcontrôleur.....	149
Annexe B Programme de commande des moteurs.....	161
Annexe C Schémas représentant la conception de quelques pièces constituant le robot Manipulateur	164

LISTE DES TABLEAUX

Tableau 1 Représentation de la vitesse en fonction du rapport cyclique et CCPR1L

LISTE DES FIGURES

2.1 Robot à structure ouverte simple (Robot sériel).....	23
2.2 Robot à structure fermée (Robot parallèle)	23
2.3 Historique de la robotique télé-opérée	25
2.4 Schématisation d'un environnement de télé-opération	26
2.5 La tête du robot, dans laquelle sont situées les caméras vidéo, restituée avec fidélité les six degrés de liberté de la tête de la personne qui le contrôle via le casque de réalité virtuelle.....	30
2.6 Continuum Réalité-Virtualité	33
2.7 Opérateur effectuant une tâche de maintenance en suivant les instructions virtuelles	34
2.8 Utilisation de la prédiction graphique pour la télé-opération spatiale ; à gauche prédiction et confirmation de la tâche, à droite exécution de la tâche	35
2.9 Utilisation de la prédiction graphique pour la télé-opération dans un environnement non structuré; à gauche prédiction et confirmation de la tâche, à droite exécution de la tâche.....	35
2.10 Vue globale du site distant (esclave=Robot), A droite Manipulation du robot ARITI à l'aide de la plateforme Evr@ (site maître), A gauche.....	36
2.11 Robot démineur	38
2.12 Robot MIR 'Machine d'inspection des réacteurs rapides'	39
2.13 Le robot EROS	41
2.14 Le robot LMF	41
2.15 Le bras Maestro au travail dans un des ateliers radioactifs de Marcoule (Gard en France)	41
3.1 Schéma synoptique d'un système RA	46
3.2 Banc de calibration géométrique de la Caméra	47
3.3 Présentation de plusieurs prises de vue de motif géométrique.....	48
3.4 Visualisation des paramètres extrinsèques. Calibration de la caméra.....	49
3.5 Mesure de distances d'une prise de vue	50
3.6 Estimation de l'erreur des paramètres extrinsèques de la caméra.....	51
4.1 Architecture fonctionnelle du système	57
4.2 Illustration d'un système de télé-opération du robot SEROR via Internet.....	58
4.3 Panneaux de contrôle (Robot & ptu). Contrôle du robot à gauche, Netmeeting	

au milieu et commande de la caméra à droite	59
4.4 Algorithme général de l'application	60
4.5 Commande en boucle ouverte du robot SEROR	61
4.6 Asservissement intégrant l'opérateur et une caméra	62
4.7 Coordonnées du robot	63
4.8 Coordonnées de la caméra	63
4.9 Simulation du robot manipulateur par les équations du modèle géométrique	66
4.10 Différentes position du robot manipulateur & réalisation d'une tâche pick-and-place	68
5.1 Structure mécanique du robot manipulateur conçu sur Autodesk/Inventor	74
5.2 Procédure d'exportation du modèle CAD vers SimMechanics	80
5.3 Méthode d'exportation SimMechanics première et seconde génération	81
5.4 Importation du modèle SimMechanics première génération	81
5.5 Représentation du robot manipulateur et des blocs le constituant dans SimMechanics première génération	82
5.6 Exportation de l'assemblage du Robot conçu vers XML fichier pour utilisation dans SimMechanics seconde génération (création du lien "Add-ins") (a).Sauvegarde du fichier XML d'importation SimMechanics d'assemblage (b)	83
5.7 Génération des fichiers STL de chaque élément constituant le robot (1). Exécution du fichier ".xml" et création du fichier ".Slx" utilisant Smimport (2)	83
5.8 Modèle complet du robot dans l'environnement Matlab/SimMechanics	84
5.9 Modèle CAD des assemblages du robot sur la plateforme Matlab/SimMechanics	85
5.10 Compartiment Joint 'Revolute'	85
5.11 Représentation des différentes parties mécaniques du robot Moteur (à gauche), Shape_moteur (à droite)	86
5.12 Modélisation et simulation du robot en environnement SimMechanics de Matlab	86
5.13 Représentation des blocs de l'Assemblage de la pince	87
5.14 Simulation de la pince représentée par le bloc Revolute5	87
5.15 Evolution de la position Revolute5	88
5.16 Evolution de la vitesse	88
6.1 Architecture matérielle de l'ensemble du système	91
6.2 Schéma-bloc d'un moteur asservi en position avec boucle de retour unitaire	92
6.3 Schéma équivalent d'un moteur à courant continu	92
6.4 Schéma-bloc d'un modèle mathématique du moteur à courant continu	93
6.5 Schématisation Moteur à courant continu et son encodeur incrémental avec 64 CPR	94

6.6 Schéma-bloc de l'implantation du PID en régulation de position du moteur	95
6.7 Programmation en Matlab de la fonction PID.....	96
6.8 Utilitaire PID Tuner de Matlab	96
6.9 Contrôleur PID conçu sur Matlab	97
6.10 Schéma général du robot réalisé.....	98
6.11 Base du robot (Socle) avec moteur d'entraînement	98
6.12 Raccordement Epaule et moteur d'entraînement du coude.....	99
6.13 Schématisation de la pince	99
6.14 Carte embarquée associant la carte de puissance à la carte de contrôle-commande....	101
6.15 Schéma synoptique de la carte à microcontrôleur.....	101
6.16 Microcontrôleurs Flash améliorés hautes performances avec A/N converter.....	102
6.17 Carte de puissance : Moteurs & périphériques.....	104
6.18 Schéma bloc du circuit de puissance.....	105
6.19 Architecture interne du circuit de puissance L298.....	105
6.20 Circuit de génération des signaux de commande	106
6.21 Circuit de génération des signaux de commande du L298 tenant compte de la complémentation du signal PWM.....	107
6.22 Circuit de protection de courant	107
6.23 Représentation des principales fonctions gérées par le Microcontrôleur.....	108
6.24 Courbe de vitesse en fonction de l'excitation en tension exprimé en valeur du registre CCPR1L fixant le duty cycle pour le moteur 50:1	110
6.25 Régulation on/off de position du moteur 29 :1 à vide pour différentes valeurs du rapport cyclique	111
6.26 Régulation proportionnelle de position du moteur 29 :1 comparée à une régulation on/off pour un rapport cyclique de 95%	111
6.27 Environnement réel de télé-opération du robot manipulateur réalisé	112
6.28 Carte de commande à base du pic 18F452 et périphériques	113
6.29 MPLAB Real ICE, Câblage du kit ICD	113
6.30 Contrôleur d'interface périphérique.....	114
6.31 Sorties codeur A et B pour motoréducteur avec 64 codeurs CPR (12 V).....	116
6.32 Commande en PWM du servomoteur	117
6.33 Affichage des signaux de commande des actionneurs du robot.....	117
6.34 Signal PWM réel issu du pic18F452 (pin17)	118
6.35 Signal PWM issu du pic18F452, moteur tourne dans un sens(a) et sens opposé(b)....	118
6.36 Forme du signal à la sortie 'output 1' de l'encodeur du moteur de la base.....	119

6.37	Forme du signal à la sortie ‘output 2’ de l’encodeur du moteur de la base.....	119
6.38	Forme du signal PWM issu du Pic18F452 pour commander les servomoteurs.....	120
6.39	Forme du signal à la sortie commande du servomoteur en translation de la pince	121
6.40	Forme du signal à la sortie commande du servomoteur en ouverture/fermeture	121
6.41	Visualisation en temps réel de la position du préhenseur pour un rapport cyclique de 59%	122
6.42	Visualisation en temps réel de la position du préhenseur pour un rapport cyclique de 59%	122
6.43	Visualisation en temps réel de la vitesse du moteur de la base pour un rapport cyclique de 59%	123
6.44	Visualisation du courant au démarrage du moteur. Courbe déterminée par Matlab, à gauche; Courbe affichée par l’oscilloscope, à droite.....	124
6.45	Visualisation du courant au démarrage, décroissance et arrêt du moteur. Courbe déterminée par Matlab, à gauche; Courbe affichée par l’oscilloscope, à droite.....	124
6.46	Evolution de la position du moteur actionnant la base du robot: a) moteur à vide b) moteur en simulation, moteur monté sur le robot.....	124
6.47	Réalisation par le robot d’une tâche pick-and-place	125
6.48	Placement des repères des articulations du Robot.....	128
6.49	Interface de contrôle de la boîte à outils de Matlab	129
6.50	Présentation du mouvement du robot en 2D	129
6.51	Trajectoire effectuée par l’effecteur dans la réalisation par le robot d’une tâche pick-and-place	130
6.52	Génération de mouvement en ligne du robot manipulateur	132
6.53	Représentation suivant X de la position du robot, sa vitesse et son accélération.....	133
6.54	Représentation suivant X de la position, vitesse et accélération angulaire	134
6.55	Représentation suivant Y de la position du robot, sa vitesse et son accélération.....	134

LISTE DES ABREVIATIONS

PWM	Pulse Width Modulation (Modulation en largeur d'impulsions MLI).
CAO	Conception assistée par ordinateur.
CAD	Computer-Assisted Design.
SMA	Structure Mécanique Articulée.
IHM	Interaction Homme-Machine.
DOF	Degree Of Freedom (Degré de liberté DDL)
RA	Réalité Augmentée.
RV	Réalité Virtuelle.
GUI	Graphical User Interface.
PIC	Peripheral Interface Controllers.
CPR	Count Per Revolution.
RM	Réalité Mixte (réalité virtuelle et augmentée)
PTU	Pan-Tilt-Unit.
LSC	Laboratoire des Systèmes Complexes.
ARITI	Augmented Reality Interfaces for Teleoperation via the Internet.
DSC	Digital Signal Controller (Contrôleur de Signal numérique).
PUMA	Programmable Universal Machine for Assembly.
USB	Universal Serial Bus
I2C	Inter Integrated Circuit Bus.
ROM	Read Only Memory.
RAM	Random Acces Memory
CPU	Central Processing Unit.
UART	Universal Asynchronous Receiver Transmitter (équivalent du bus série).
HID	Human Interface Device.
LED	Light Emiting Diode.
IHR	Interaction Homme-Robot.
ISIS	Intelligent Schematic Input System
URDF	Unified Robot Description Format (Format de la Description unifiée des Robots)
STL	STereoLithography (Stéréolithographie)

NOMENCLATURES

θ_1^R : est l'angle de rotation autour de l'axe Y de l'articulation de longueur L1.

θ_2^R : L'angle de rotation autour de l'axe Y, de l'articulation de longueur L2.

θ_3^R : L'angle de rotation autour de l'axe X de l'articulation de longueur L3.

Génération de trajectoires:

X : Composante selon X de la position

Y : Composante selon Y de la position

VX (m/s) : Composante selon X de la vitesse

VY (m/s) : Composante selon Y de la vitesse

aX (m/s²) : Composante selon X de l'accélération

aY (m/s²) : Composante selon Y de l'accélération

θ : Position angulaire

ω : Vitesse angulaire

α : Accélérations angulaire

Chapitre 1 Introduction générale

1.1 Contexte de la Thèse

Un robot est un mécanisme de manipulation, automatique, reprogrammable, polyvalent, capable de positionner et d'orienter des matériaux, pièces, outils ou dispositifs spécialisés au cours de mouvements variables et programmés pour l'exécution d'une variété de tâches [1-4].

L'un de ses premiers objectifs est de fournir une assistance aux humains et parfois de les remplacer pour des activités dangereuses, difficiles ou dans des endroits inaccessibles.

Lorsque la réalisation de la tâche par le robot s'effectue à distance "environnement hostile", on parle de télé-opération [5-8]. La télé-opération peut s'effectuer en réseau local ou via Internet [9-13]. Et dans le cas de figure où l'environnement de travail est simulé par ordinateur, on a ce qu'on appelle : réalité virtuelle (RV) [14-18]. Des objets virtuels peuvent être ajoutés à une image réelle et on parle dans ce cas de la réalité augmentée (RA) [19-29]. Ces deux techniques permettent de réduire un certain nombre de difficultés liées à la télé-opération et pourraient également être utilisées dans des interactions, homme-robot, plus avancées [30-33].

1.2 Motivation

Le travail principal à réaliser lors de cette thèse consiste en le développement d'environnement de programmation de missions offrant à l'opérateur deux fonctions principales : l'entraînement et l'assistance dans l'exécution d'une mission donnée. Ce qui est attendu de cette mission est de pouvoir réaliser des tâches de manipulation dans des milieux hostiles et plus spécialement dans le domaine nucléaire. On mettra l'accent sur le fait que la réalisation de missions, en mode télé-opéré, sur des sites industriels sensibles tel que le nôtre, nécessite des moyens performants pour la préparation, la simulation et l'exécution de ces missions. Vu la sensibilité de ces domaines "risques d'exposition aux rayonnements", plusieurs étapes avaient été nécessaires pour aboutir au résultat final.

1.3 Contributions

Afin d'atteindre nos objectifs à savoir la mise en œuvre de la réalité augmentée dans la télé-opération du robot, des travaux ont été accomplis et se résument par les points suivants :

1. Mise en œuvre la télé-opération du robot manipulateur SEROR¹ à trois degrés de liberté, à travers un réseau local et sur une distance de 430km via Internet;
2. Conception mécanique utilisant le logiciel Autodesk/Inventor, d'un robot manipulateur à 4 d.d.l et utilisant motoréducteurs à encodeurs pour ses articulations;
3. Modélisation du robot manipulateur conçu en utilisant la plateforme SimMechanics de Matlab 'principe du 'prototypage' ;
4. Réalisation mécanique du robot;
5. Conception et réalisation de la carte de commande opérant autour du microcontrôleur PIC18F452;
6. Réalisation de la carte de puissance se basant sur le driver L298 et qui permet le pilotage de deux moteurs à courant continu.

1.4 Structure du manuscrit

Travailler dans un milieu hostile n'est pas une tâche facile, ce qui nécessite de réunir plusieurs conditions pour mettre l'opérateur humain en situation de sécurité par rapport aux risques qu'il peut engendrer, faisant appel à des robots pour effectuer ces tâches. Ceci a nécessité de passer par plusieurs études, en explorant des domaines d'actualité et étudier les nouvelles techniques de télé-opération de robot telle que la réalité augmentée.

Après une introduction générale, détaillée dans le premier chapitre, nous présentons dans le deuxième chapitre un état de l'art couvrant tous les aspects liés à la robotique et à la télé-opération, suivi de quatre chapitres montrant la progression de mon travail.

En chapitre 3, nous cherchons à mettre en œuvre la technique de réalité augmentée pour la télé-opération du robot, pour cela nous passons en revue les avantages de cette technique et son impact dans le cas de notre application. De même, l'interfaçage avec la partie contrôle permet d'obtenir des données réelles et de les comparer avec des données simulées. En superposant l'image réelle du robot sur son modèle virtuelle (simulé par ordinateur), c'est le fondement de la réalité augmentée (RA). Son principe consiste à superposer un modèle informatique, ainsi que d'autres informations appropriées, sur une image réelle correspondante. Par ce cheminement, il est possible de contrôler la réalité de manière plus précise et plus efficace que par des commandes directes. De ce fait, la perception par

¹ SEROR est un robot manipulateur à 3 d.d.l réalisé au Laboratoire des technologies avancées LATA à l'Université Mentouri Constantine dont j'ai eu la chance d'y participer dans la partie programmation en langage JAVA et commande Intranet au laboratoire local, et Internet sur une distance de 430 km à partir de Ain Oussera.

l'opérateur, de l'environnement distant du robot est nettement améliorée et le déroulement de la tâche est effectué en toute sécurité.

En chapitre 4, nous réalisons la télé-opération du robot manipulateur expérimental SEROR à travers les réseaux Intranet et Internet [34,35]. Ce travail nous permet la maîtrise des techniques de télé-opération, la familiarisation avec les langages orientés objet et de ce fait la programmation des robots. La commande du robot est basée sur les modèles géométriques directs et inverse du robot ainsi que ceux de la caméra "webcam" qui sert à la prise des images du robot et de son environnement. Néanmoins, certains phénomènes apparaissent lors de la télé-opération du robot, comme le décalage dans le temps entre l'envoi et la réception de l'ordre au robot "Time-delay". Ce qui provoque une instabilité du système et un manque de précision dans la réalisation de la tâche.

Ces phénomènes sont principalement dus à la distance séparant l'opérateur humain du site des opérations, une perturbation du réseau utilisé [36-39], un manque d'éclairage de la scène, une ombre [40,41], ou tout simplement l'inexpérience de l'opérateur.

On a essayé d'aborder ce problème sous différents angles et différents schémas de contrôle avaient été proposés et testés dans la littérature spécialisée [42-44]. Dans certains travaux, on a effectué une comparaison de quelques systèmes de contrôle en robotique, comme la passivité, la prédiction, les systèmes de contrôle en mode glissant et autres [45-48]. La comparaison est basée sur le niveau de transparence, l'exigence de stabilité, le type de délai et la plage de retard.

Avec l'évolution de la technologie du numérique et de la vision artificielle, la recherche s'est tournée vers trois domaines étroitement liés qui sont : La réalité augmentée, la réalité virtuelle et la vision artificielle. L'association Réel et Virtuel consiste en l'ajout, d'objets virtuels calculés par un système informatique en temps réel, à la scène réelle. Le but étant d'intensifier la compréhension et l'interaction de l'utilisateur avec son environnement [49] ; elle s'appuie généralement sur des informations issues de capteurs ou de caméras. La vision consiste en l'élaboration de méthodes de modélisation et calibration des capteurs utilisés, de modélisation 2D-3D d'environnement [50].

Dans le but de permettre l'intégration de la réalité augmentée à notre système de télé-opération, il était primordial de procéder à la mise en œuvre d'un système robotique plus adéquat en matière de robustesse et de précision. Ce travail a été étalé sur deux chapitres.

En premier lieu en chapitre 5, nous faisons l'étude de conception du robot manipulateur à quatre degrés de liberté en utilisant le logiciel Autodesk/Inventor, qui est un logiciel CAD. Le modèle complet du système conçu est transféré vers l'environnement Matlab via le link

ADD-IN, qui permet le transfert de tous les éléments du robot vers la plateforme SimMechanics, pour effectuer sa modélisation puis sa simulation. La modélisation de chaque partie de sa structure est nécessaire pour pouvoir effectuer des simulations sur ce modèle et extraire des algorithmes de contrôle (contrôle de vitesse, positions, etc.).

L'étape précédente a débouché sur la réalisation du robot manipulateur; trois motoréducteurs sont utilisés pour la rotation de ses articulations, et deux servomoteurs pour la manipulation de la pince. Le système de pilotage du robot se fait à l'aide d'une carte embarquée à base du microcontrôleur PIC18F452, ce choix a été motivé par sa richesse relative en ressources (instructions, adressages, mémoires...etc.). Les étapes de réalisation du robot ainsi que les détails techniques suivies pour mettre sur pied cette réalisation font sujet du chapitre 6.

Le système de commande du robot prend en charge la gestion de toutes les tâches liées à la commande et le fonctionnement du bras manipulateur. Ce système est constitué de deux parties, une carte de commande embarquée et le PC. Ce dernier assure la communication avec la carte de commande pour le contrôle du robot par l'utilisation d'un langage évolué (Matlab), ce qui facilite la tâche pour l'utilisateur tout en permettant aussi le traitement en parallèle d'images émanant des caméras supervisant la scène accomplissant ainsi la mise en place d'une plateforme pour la réalisation de la réalité virtuelle et/ou augmentée.

Les commandes sont écrites en langages évolués dans un fichier script de Matlab et envoyés vers la carte embarquée à base de microcontrôleur PIC18F452 et sont interprétés à ce niveau en une suite de tâches exécutées en langage assembleur du microcontrôleur.

La carte de contrôle et de commande embarquée est conçue et réalisée autour du microcontrôleur PIC18F452. Cette carte remplit les principales fonctions de :

- Commande et régulation (de position) des moteurs
- Lecture des positions des moteurs
- Communication vers le PC
- Envoi des états des moteurs (positions) vers le PC

Une conclusion générale ainsi que des perspectives sont données en chapitre 7.

Chapitre 2 Etat de l'art

Robotique, Télé-opération, Réalité Virtuelle,

Réalité Augmentée et nucléaire

2.1 Introduction

L'utilisation de manipulateurs dans l'industrie n'a cessé d'augmenter ces dernières années. Ils ont permis d'augmenter la production et la précision des mouvements effectués. La plus grande partie de ces manipulateurs est constituée des manipulateurs sériels dont la morphologie peut facilement être comparée à celle de l'être humain, tels les robots SCARA (figure 2.1). Contrairement aux manipulateurs parallèles, considérés comme des mécanismes en chaînes cinématique fermées dont l'organe terminal est relié à la base par deux ou plusieurs chaînes cinématiques indépendantes (figure 2.2).

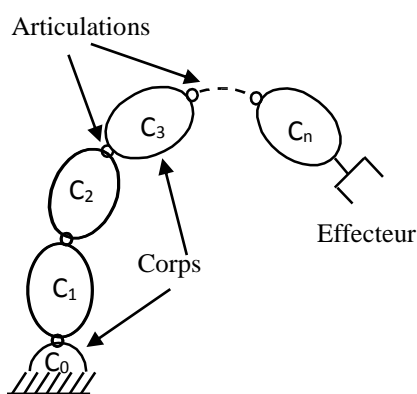


Figure 2.1. Robot à structure ouverte simple (Robot sériel)

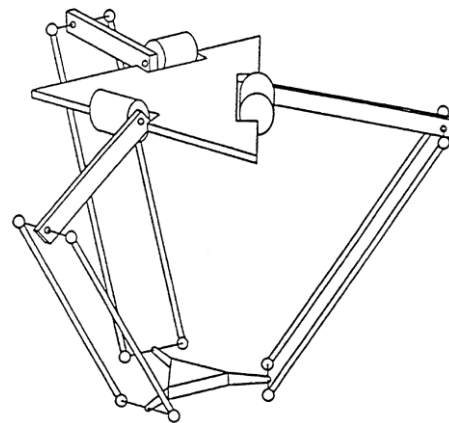


Figure 2.2. Robot à structure fermée (Robot parallèle)

Ce travail vise à présenter l'état de l'art dans les techniques d'interaction de télé-opération entre les robots à caractère sériels et leurs utilisateurs en environnement distant. Nous commençons par effectuer un bref historique sur la robotique en général, puis des systèmes de télé-opération et les facteurs pouvant influencer ces systèmes. Nous poursuivons avec un état de l'art sur les systèmes de réalité virtuelle et réalité augmentée pour la télé-opération de robots. Ensuite nous présentons un aperçu des milieux hostiles en commençant par une définition de ces milieux.

2.2 La robotique

Les premiers robots ont été conçus à la fin de la seconde guerre mondiale aux Etats Unis afin de manipuler les substances radioactives à la place des chercheurs. En 1947 l'Argonne National Laboratory démarra un projet destiné à développer des manipulateurs à usage général qui puissent copier les mouvements et les fonctionnalités de la main humaine.

Le développement des systèmes humains-robots ont évolué chronologiquement et couvrent actuellement une grande variété d'applications et sont généralement très différents les uns des autres. Les champs scientifiques couverts par la robotique s'insèrent dans plusieurs disciplines parmi lesquelles la mécanique (SMA), l'informatique (programmation), l'automatique, l'électronique (Contrôle commande de processus) et la physique. Ce caractère pluridisciplinaire de la robotique engendre ainsi un nouveau domaine scientifique et technique qui étudie les différents composants d'un robot (mécanisme, actionneurs, capteurs, loi de commande,...). Les techniques d'interaction conçues pour chaque système sont également très différentes.

Un système robotique est constitué d'un robot manipulateur, d'une interface et d'un moyen de contrôle nécessaire à la réalisation d'une tâche.

La réalisation de missions en mode télé-opéré sur des sites industriels sensibles nécessite des moyens performants pour la préparation, la simulation et l'exécution de missions.

Les robots sont utilisés aujourd'hui pour accomplir de nombreuses tâches dans la société, que ce soit dans l'industrie, à la maison, ou comme outils d'aide sur des incidents tragiques.

2.2.1 Historique de la robotique télé-opérée²

En 1898 Nicolas TESLA réalise les premiers démonstrateurs de navires téléguidés. "Method of an Apparatus for Controlling Mechanism of Moving Vehicle or Vehicles". En 1979, la NASA commence les premières opérations de pilotage au travers d'un écran du drone aérien HiMAT.

La robotique militaire a débuté durant la période 1939-1945 avec le robot "Goliath", robot chenillé filoguidé utilisé par la Wehrmacht durant la seconde Guerre mondiale (figure 2.3a). L'année 1970 a été le début de la robotique spatiale avec le robot "LUNOKHOD", robot huit roues motrices téléguidées depuis la Terre envoyé sur la Lune par l'Union soviétique (figure 2.3b). En 1974, réalisation du robot français radiocommandé "CENTAURE", destiné à intervenir en milieu hostile et irradié en cas d'accident nucléaire (figure 2.3c).

²https://www.wisg.fr/Mise_en_Ligne_PPT/11_fevrier_2016/Session1_Les_outils_technologiques_lutte_contre_menaces_multiformes/Robot_Demineur_Jean-Marc_Denis.pdf

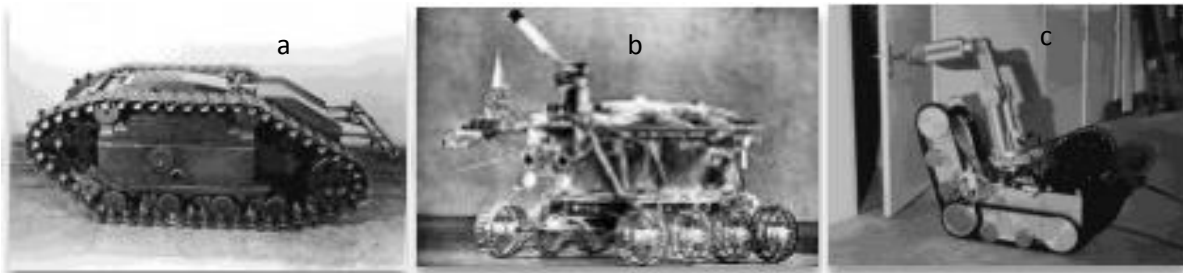


Figure 2.3. Historique de la robotique télé-opérée

2.2.2 Programmation des robots

Pour effectuer une définition des tâches que devra effectuer un robot grâce à un langage de programmation, l'opérateur écrit un programme informatique sur une machine dédiée ou un ordinateur classique. Les langages de programmation diffèrent selon le procédé de commande utilisé et la nature de la tâche à réaliser.

2.3 La Télé-opération

La notion de télé-opération désigne une interaction entre l'opérateur et le robot. L'opérateur envoie une consigne et le robot l'exécute.

Le développement des systèmes de télé-opération a été stimulé par la nécessité d'intervention humaine dans des environnements hostiles pour réaliser des tâches dangereuses, très pénibles ou tout simplement impossibles à effectuer par l'homme. Ce besoin est apparu dans différents domaines à savoir: entretien, maintenance, déminage, militaire,...etc. [51-55]. L'objectif du robot est donc de fournir une assistance aux humains et parfois de les remplacer pour des activités dangereuses et difficiles; C'est le fondement de la télé-opération.

2.3.1 Bref historique

Le nom télé-opération n'est apparu qu'autour de 1975 bien que ses spécialistes l'utilisent depuis avant la seconde guerre mondiale. Pourtant, c'est un mot qui tient son origine du mot grecque télé qui signifie loin et du mot latin 'operatio', qui signifie travailler, œuvrer ou accomplir. Télé-opération désigne donc un mode d'exécution d'une tâche, d'une action, d'un mouvement ou d'un travail à distance.

Le moyen de la télé-opération sera ainsi le téléopérateur, qui s'applique à la désignation de l'homme manipulant un système de télé-opération. Cette action se base sur les principes et les techniques qui permettent à l'opérateur humain d'accomplir une tâche à distance, à l'aide

d'un système robotique d'intervention (dispositif esclave), commandé à partir d'une station de contrôle (dispositif maître), par l'intermédiaire d'un canal de télécommunication (figure 2.4). L'organe terminal porte l'outil de travail, qui dépend des tâches que l'on désire réaliser. Il s'agit souvent de pinces à deux ou trois doigts que l'on nomme préhenseurs.

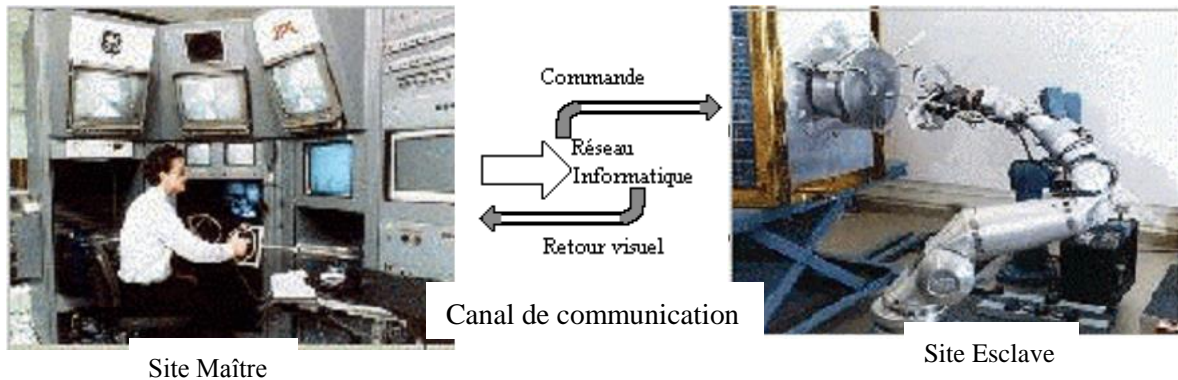


Figure 2.4. Schématisation d'un environnement de télé-opération

2.3.2 Facteurs pouvant influencer la télé-opération

Le principal problème rencontré lors d'une télé-opération est le délai. Ce délai est principalement dû à deux causes : la distance (physique) entre les deux parties et la saturation du réseau.

D'autres problèmes peuvent avoir des effets sur le système de télé-opération, comme la visibilité du site de travail, l'occlusion des objets manipulés causée parfois par des obstacles ou une simple ombre [56].

En outre, un facteur non négligeable est l'opérateur humain lui-même et ses capacités. S'il a une expérience suffisante du système et de son environnement, il est capable de se former un modèle mental et de l'utiliser pour anticiper l'état futur du robot, obtenant ainsi une marge plus large pour le délai.

De nos jours, de nombreux systèmes de télé-robotique sont des applications orientées Internet [57]. L'avènement d'Internet a permis d'introduire des systèmes télé-robotiques dans plusieurs domaines comme la télé-chirurgie ou la télémaintenance.

2.4 Systèmes robotiques basés sur Internet pour la télé-opération

2.4.1 Introduction

La télé-opération basée sur Internet permet aux utilisateurs de travailler et de collaborer ensemble pour contrôler des robots de n'importe quel endroit dans le monde. Elle ne dépend non plus de l'environnement où se trouve le matériel qui permet de contrôler le robot qui, lui se trouve sur le site esclave généralement distant.

2.4.2 Delay sur Internet

Internet représente un moyen de communication entre les opérateurs et les machines lors de la transmission des informations. Néanmoins, lors de cette transmission, un problème de caractère aléatoire survient et cause la perturbation du fonctionnement du système. Ce phénomène est le délai de communication entre l'opérateur humain et le robot distant.

Ce délai peut être provoqué par la distance physique potentiellement importante entre les sites maîtres (de l'opérateur) et esclaves (du robot), le temps de traitement des contrôleurs, la qualité du réseau et la bande passante de communication.

Il faut mentionner que le retard n'est pas la seule cause d'instabilité d'un système de télé-opération. En fait, d'autres sources comprennent le bruit, les perturbations externes, l'environnement non passif, etc. Un autre facteur influençant, est le matériel utilisé pour le poste de commande et le robot.

Des études faites ont démontré que le délai, sur des distances importantes (intercontinentales), a une composante continue d'environ 400 ms et, de temps en temps, des sauts importants [58].

Pour des tâches qui demandent des réponses très rapides, des études ont trouvé que les performances de l'opérateur humain commencent à diminuer si le délai dépasse 180 ms, dans ce cas, l'opérateur pourra accomplir ses tâches, mais uniquement que ses performances seront dégradées. Le délai maximal qui n'empêchera pas le fonctionnement normal du système téléopéré n'est pas fixe. Il dépend d'une multitude de facteurs. La tâche à accomplir est un élément important. Avec le même robot, les contraintes seront différentes s'il faut naviguer dans un environnement ou chercher un certain objet.

Il a été démontré que même un retard inférieur à 50 ms peut déstabiliser un système de télé-opération bilatéral [59].

Afin de palier à ce phénomène perturbant, des études récentes en conception d'interface opérateur ont démontré sommairement que l'intégration de techniques de réalité virtuelle à des images vidéo obtenues d'une caméra montée sur un robot permet d'améliorer les performances de télé-opération.

2.5 Utilisation de la réalité virtuelle sur l'interface de contrôle

L'exploitation des nouvelles technologies comme la réalité virtuelle/augmentée est un atout favorable pour la poursuite des recherches.

2.5.1 Principe de la réalité virtuelle 'RV'

Depuis son introduction, la RV a été définie de plusieurs manières. Elle peut être définie par sa finalité, ses applications, ses fonctions ou encore les techniques sur lesquelles elle repose. Elle peut être désignée par plusieurs termes que nous trouvons dans la littérature scientifique telle que "Environnement Virtuel", "Espace Virtuel", "Monde Artificiel", "Réalité Synthétique" et bien d'autres.

Quelle que soit la définition de cette réalité dite virtuelle, sa finalité reste la même. Elle permet d'avoir une interface naturelle et intuitive, et d'intégrer différentes sources d'information pour améliorer la perception de l'opérateur. Elle nécessite généralement l'utilisation d'un casque HMD "Head Mounted Display" dans lequel des objets (images, vidéos) virtuels sont projetés et avec lesquels l'utilisateur peut interagir en temps réel. Ce casque peut aussi s'appeler casque de visualisation, casque immersif, casque de réalité virtuelle ou casque-écran.

La RV a donc offert à l'homme un nouveau mode de vie qui lui permet de réaliser certaines tâches difficiles ou même impossibles dans la vie réelle. De même, avec l'arrivée de la réalité virtuelle, de nombreuses applications ont vu le jour. Ses applications sont fondées sur une représentation 3D de l'environnement distant que l'opérateur humain peut observer avec le point de vue le plus approprié.

2.5.2 L'histoire de la RV³

La réalité virtuelle ne peut être envisagée que depuis peu, grâce à l'augmentation importante de la puissance intrinsèque des ordinateurs, en particulier la possibilité de créer en temps réel des images de synthèse et de permettre une interactivité, toujours en temps réel, entre l'utilisateur et le monde virtuel. Elle est devenue accessible au grand public, depuis 2016, grâce à des casques comme l'Oculus Rift⁴, le HTC⁵ Vive ou le PlayStation VR. Tout un chacun peut désormais se plonger dans une réalité alternative sans bouger de l'endroit où il se trouve. Mais pour arriver à cela, il aura fallu beaucoup de temps et de tentatives pour mettre au point des systèmes de réalité virtuelle avec une simplicité d'usage :

- ❖ Le cinéaste Morton Heilig, réalisa En 1957-1962, le premier casque VR "Sensorama", une machine conçue pour *immerger* les utilisateurs dans un monde imaginaire. Son but est de permettre aux spectateurs de plonger dans les films en trompant leur sens, afin de donner

³ <https://www.realite-virtuelle.com/histoire-vr-7-etapes-1511>.

<https://fr.linkedin.com/pulse/la-r%C3%A9alit%C3%A9-augment%C3%A9e-histoire-opportunit%C3%A9s-et-limites-laurent-legrain>

⁴ https://fr.wikipedia.org/wiki/Oculus_Rift

⁵ https://fr.wikipedia.org/wiki/HTC_Vive

une nouvelle envergure à l'expérience cinématographique. Il imagine le *Sensorama* comme une machine capable de simuler des odeurs, de produire du son en stéréo, des vibrations, et même des effets atmosphériques comme le vent dans les cheveux⁶.

- ❖ Le premier casque VR fonctionnel a été créé en 1968 par Ivan Sutherland "l'épée de Damoclès", en référence au célèbre mythe romain; ce casque embarque un écran stéréoscopique permettant d'afficher des formes simples, changeant la perspective en fonction des mouvements de tête de l'utilisateur. L'écran n'étant pas occultant, ces formes sont superposées aux images du monde réel et cet appareil peut donc aussi être considéré comme le premier casque de réalité augmentée.
- ❖ En 1970, Thomas Furness, ingénieur militaire a travaillé sur un *ambitieux projet de simulateur de vol immersif*, le *Super Cockpit*, jusque dans les années 80. Il en résulta un cockpit d'entraînement capable de projet des cartes 3D générées par ordinateur, des images infrarouges et radar, et des données d'aviations en temps réel au sein d'un espace 3D. Ce projet permettait déjà aux pilotes de s'entraîner en utilisant des gestes, des paroles et même des mouvements oculaires.
- ❖ A partir de l'an 2010, a eu la vulgarisation de la RV avec l'arrivée d'un premier prototype de casque VR "*Oculus Rift*" réalisé par l'américain *Palmer Luckey*. La version commercialisée de ce casque a eu lieu en 2016.

En télé-opération, la RV est considérée comme une technologie qui immerge un ou plusieurs utilisateurs dans un monde synthétique représentant un environnement réel ou imaginaire et qui permet à ces utilisateurs d'être des acteurs capables de changer les propriétés de l'environnement et d'interagir avec les différentes entités composant l'univers simulé.

Quel que soit le moyen d'agir en télé-opération, avec un système maître-esclave, assistée par ordinateur ou par la réalité virtuelle, l'opérateur doit percevoir efficacement le monde téléopéré.

Dans un avenir proche, les avancées combinées de la robotique et de la réalité virtuelle vont donner lieu à de multiples applications et usages inédits. On pense notamment à des expériences extracorporelles comme celle du rover lunaire développé par des étudiants et chercheurs de l'université Carnegie Mellon, aux États-Unis, et qui ambitionne de rendre accessible au grand public l'exploration de la Lune en vue subjective.

⁶ En 2017, Tecmo a réalisé cette idée avec le VR Sense, une cabine destinée aux salles d'arcade VR japonaises, proposant une immersion sensorielle complète avec des effets atmosphériques comme ceux imaginés par Heilig pour le Sensorama.

D'un point de vue plus réaliste, l'association d'un casque de réalité virtuelle et d'un robot téléopéré pourrait aider des équipes de secours à évaluer avec une meilleure précision les conditions dans des zones difficiles d'accès ou dangereuses, sans parler des nombreuses applications militaires.

C'est précisément ce genre d'immersion qu'a cherché à reproduire une équipe de l'université de Pennsylvanie (États-Unis) avec son projet Dora (*Dexterous Observational Roving Automaton*) qui se présente comme une "plateforme robotique téléopérée immersive". Il s'agit d'associer un casque de réalité virtuelle "Oculus Rift" à un robot télécommandé dont la tête est munie de deux caméras vidéo (figure 2.5)⁷.

La personne qui porte le casque voit en quelque sorte à travers les yeux du robot, comme si elle était sur place. Si ce genre de technique n'est pas une nouveauté en soi, l'innovation tient à la liberté de mouvement qu'offre Dora. Le système suit avec précision les mouvements de la tête sur six degrés de liberté. L'objectif est d'atteindre un niveau d'immersion tel que la personne ait l'impression d'être réellement sur place. La vidéo de démonstration, publiée sur Vimeo, permet de constater que l'Homme et la machine font quasiment corps.



Figure 2.5.

La tête du robot, dans laquelle sont situées les caméras vidéo, restitue avec fidélité les six degrés de liberté de la tête de la personne qui le contrôle via le casque de réalité virtuelle.

© Dora Platform, University of

Le casque Oculus détecte à la fois l'orientation de la tête grâce à sa centrale inertielle ainsi que la position au moyen de balises infrarouge. Les informations sont transmises à des microcontrôleurs Arduino et Edison (fabriqués par Intel) grâce auxquels le robot reproduit fidèlement les mouvements de la tête. Les caméras de l'engin filment avec une définition de 976 x 582 pixels à 30 images par seconde. Le casque Oculus pourrait toutefois supporter une qualité supérieure.

Plusieurs étapes doivent se dérouler entre le moment où la personne bouge la tête et le moment où l'affichage vidéo du casque restitue cette action : réception par l'ordinateur des informations relatives aux mouvements, traitement, capture de l'image vidéo correspondante

⁷ <https://www.futura-sciences.com/tech/actualites/robotique-piloter-robot-grace-casque-realite-virtuelle-58139/>

puis restitution au casque Oculus. Cette procédure nécessite un temps de latence qui, Selon Oculus, le minimum acceptable pour garantir l'immersion et le réalisme de la réalité virtuelle est de 60 millisecondes.

Actuellement, le système Dora effectue un temps de latence d'environ 70 millisecondes. La différence n'est donc pas énorme, sachant que, dans le cas du projet Dora, les concepteurs doivent, en plus, composer avec la vitesse de la connexion sans fil entre le casque et le robot ainsi que le frottement des pièces mécaniques en mouvement. Cependant, ils estiment pouvoir optimiser le système pour réduire l'écart.

2.6 Utilisation de la réalité augmentée pour la télé-opération.

2.6.1 Définition 1⁸.

Réalité: n.f (latin médiéval *realitas*, du latin classique *realis*, de *res*, chose)

1. Caractère de ce qui est réel, de ce qui existe effectivement.
2. Ce qui est réel, ce qui existe en fait, par opposition à ce qui est imaginé, rêvé, fictif.

Augmenter: v.tr (bas latin *augmentare*, du latin classique *augere*, accroître)

1. Rendre quelque chose plus grand, plus considérable.
2. Rendre quelque chose plus important, plus intensif.

L'association de ces deux mots en une seule expression "réalité augmentée" a été introduite au début des années 90 par Caudell⁹.

2.6.2 Définition 2¹⁰

Le terme *réalité augmentée* a été utilisé pour la première fois en 1992 par Tom Caudell et David Mizell pour nommer la superposition de matériel informatisé sur le monde réel. Par la suite, l'expression a été précisée par Paul Milgram et Fumio Kishino (1994) dans leur article fondateur "Taxonomy of Mixed Reality Visual Displays".

La réalité augmentée est une image réelle à laquelle sont ajoutés un modèle informatique et des informations pertinentes.

⁸ Définition étymologique du dictionnaire Larousse

⁹ T.P.Caudell and D.W.Mizell. Augmented reality: an application of heads-up display technology to manual manufacturing processes. Proceedings of the Twenty-Fifth Hawaii International Conference on System Sciences, 2:659-669, 1992.

¹⁰ Jean-Marc Cieutat, "Quelques applications de la réalité augmentée : Nouveaux modes de traitement de l'information et de la communication. Effets sur la perception, la cognition et l'action", Mémoire pour Habilitation à Diriger des Recherches-2013.

Prenons l'exemple d'un environnement hostile dans lequel un homme ne peut pas se déplacer. L'opérateur humain télécommandera un engin via une caméra placée sur celui-ci. La réalité augmentée permettra alors d'assister l'opérateur qui ne peut pas voir un certain nombre d'informations comme le relief ou un objet caché par exemple. Par ordinateur, il est fait une modélisation de cet environnement sur lequel on superpose le monde virtuel au monde réel. L'opérateur peut ainsi avoir à sa disposition des points de vue différents que la caméra ne lui montre pas. "On parle donc de réalité augmentée dans la mesure où au monde réel sont ajoutées des fonctionnalités pour assister l'opérateur et permettre ainsi un meilleur contrôle".

La réalité augmentée désigne selon Ronald T Azuma, chercheur à l'Université de Caroline du Nord et auteur d'une des premières études sur la réalité augmentée intitulée "A survey of Augmented reality", publié en 1997, la réalité augmentée peut se définir comme une interface entre des données "virtuelles" et le monde réel.

Concrètement, la réalité augmentée combine le monde réel et les éléments numériques en temps réel, offre à l'utilisateur des possibilités d'interaction en temps réel, et repose généralement sur un environnement 3D.

Techniquement, ce n'est pas la réalité qui est augmentée, mais bien la perception de l'utilisateur. La plupart du temps, la réalité augmentée permet d'altérer la vision de l'usager. Toutefois, les cinq sens peuvent être affectés par cette technologie. Il est par exemple possible d'ajouter des sons artificiels à un environnement sonore.

Selon Azuma, un système de réalité augmentée est un système qui complète le monde réel avec des objets virtuels (générés par ordinateur) de telle sorte qu'ils semblent coexister dans le même espace que le monde réel [60]. Ce qui dans les deux cas l'amènera à définir les caractéristiques d'un système de réalité augmentée par les trois propriétés suivantes:

1. "Combiner le réel et le virtuel". Au monde réel en trois dimensions doivent être intégrées des entités également en trois dimensions.
2. "Interactivité en temps réel" ; Cela exclut notamment les films bien que la condition précédente soit respectée.
3. "Recalage en trois dimensions" ; Cela permet de faire coïncider visuellement les entités virtuelles avec la réalité.

Pour définir la réalité augmentée, on évoque bien souvent les travaux de Milgram et Kishino sur le continuum entre réel et virtuel publiés en 1994. Selon ces deux experts, la réalité

mixte va de l'environnement réel à l'environnement virtuel en passant par la réalité augmentée et la virtualité augmentée (figure 2.6).

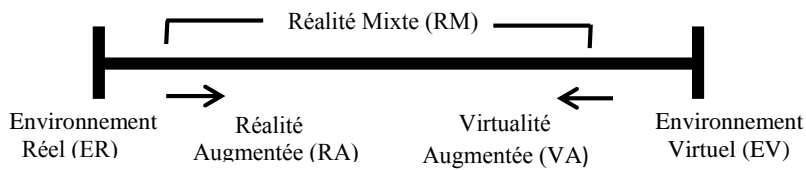


Figure 2.6. *Continuum Réalité-Virtualité*

2.6.3 Travaux réalisés dans le domaine

La réalité augmentée pour la télé-opération, consiste généralement à superposer un modèle virtuel sur une image réelle grâce à un mécanisme appelé la calibration¹¹ Cette superposition du virtuel sur le réel est utilisée soit, pour faire de la visualisation prédictive, soit pour superviser des tâches en cours d'une mission.

La visualisation prédictive est utilisée généralement en télé-opération pour pallier le problème de délai, ou bien pour rehausser l'image vidéo afin de compenser ses défauts.

Ce principe de superposition a été utilisé, et adopté généralement par la plus part des laboratoires qui s'intéressent à la télé-opération et à la supervision de missions de télérobotique.

En bref, quelques travaux de certains laboratoires:

- ❖ KARMA "Knowledge-based Augmented Reality for Maintenance Assistance", à l'origine du concept *Steven Feiner* en 1993 (figure 2.7).

¹¹ Permet l'identification des paramètres (internes et externes) de la caméra réelle qui seront appliqués par la suite sur la caméra virtuelle afin de superposer les deux mondes virtuel et réel.

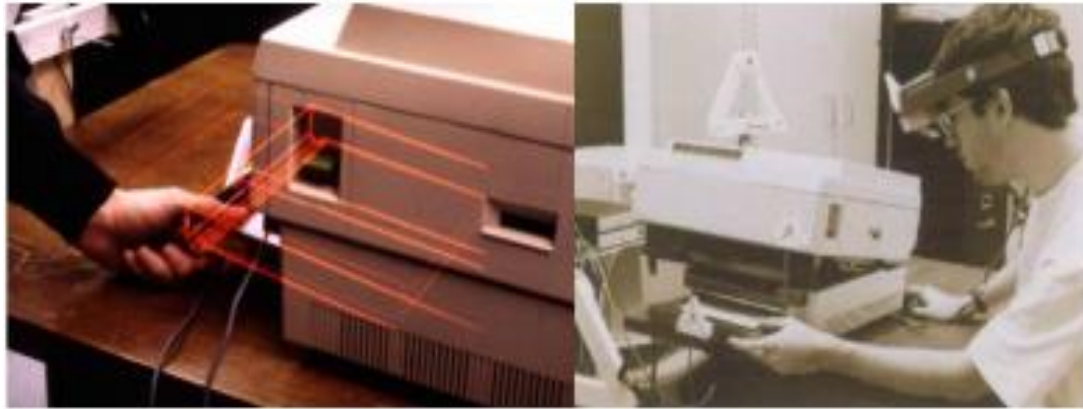


Figure 2.7. Opérateur effectuant une tâche de maintenance en suivant les instructions virtuelles

- ❖ Jet Propulsion Laboratory (JPL-NASA, USA)¹²: Cette équipe travaille sur un système permettant à l'opérateur d'être indépendant du temps de propagation entre le site esclave et le site maître. L'opérateur dispose dans le site maître d'une interface lui permettant de configurer le système, de gérer l'acquisition des données pertinentes et de visualiser en temps réel les forces et couples s'exerçant sur le bras.

Une interface vidéo permet de gérer le retour d'information des caméras ainsi que la superposition à ces dernières des images synthétiques (prédicteurs graphiques, figure), dont certaines issues d'une base de données géométriques mise à jour en temps réel. Pour s'affranchir des temps de transfert des informations, l'image prédite du déroulement de la tâche est présentée à l'opérateur et recalée sur l'image réelle (figure 2.8, envoyée depuis JPL). L'aide principale apportée à l'opérateur est visuelle et lui permet d'avoir une estimation de la distance et de l'orientation de l'effecteur par rapport à la cible.

¹² Télé-opération, télérobotique et Internet: Techniques & applications, Samir Otmae, Université d'Evry Val d'Essonne / UFR S&T Laboratoire IBISC (Informatique, Biologie Intégrative et Systèmes Complexes). CNRS-FR 2873

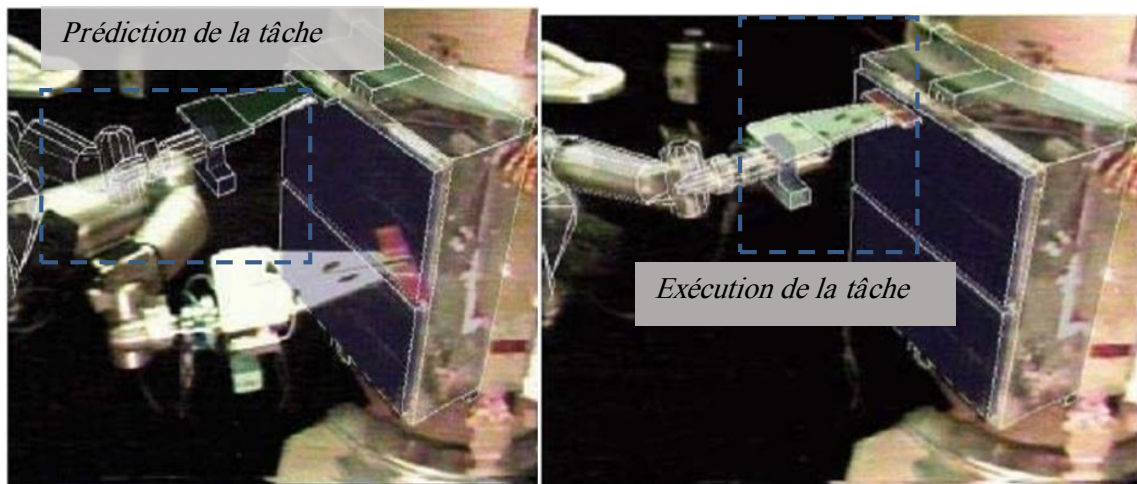


Figure 2.8. Utilisation de la prédiction graphique pour la télé-opération spatiale; à gauche prédiction et confirmation de la tâche, à droite exécution de la tâche

- ❖ Ergonomics in Teleoperation and Control Laboratory (ETCL Université de Toronto, Canada): Dans ce laboratoire, a été réalisé le système ARTEMIS (Augmented Reality TELeManipulation Interface System). Ce système permet de générer les modèles de l'environnement distant (non structuré) en utilisant la stéréovision. Il permet de superposer le modèle stéréographique du robot sur les images stéréo-vidéo du site distant, le modèle stéréographique du robot est utilisé pour simuler les tâches de manipulation et des instructions sont ensuite envoyées au robot réel pour l'exécution (figure 2.9, envoyée depuis ETCL).

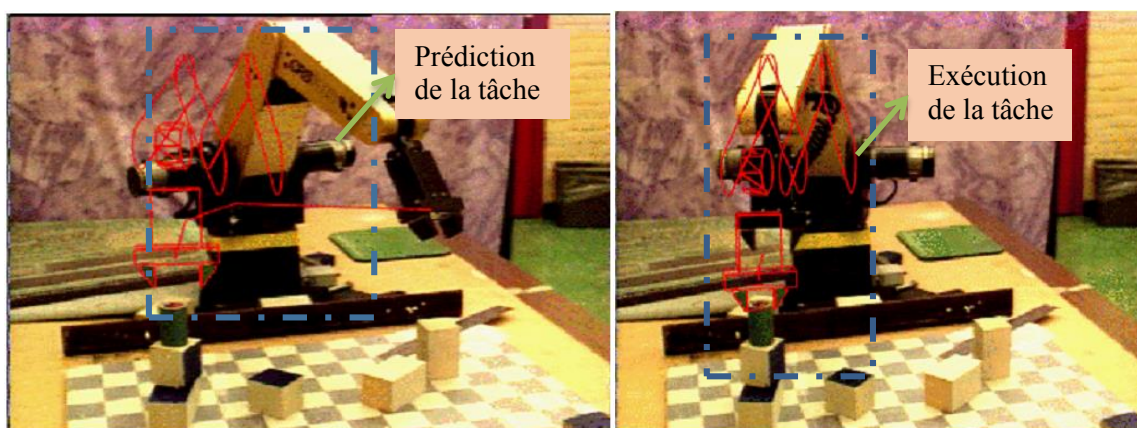


Figure 2.9. Utilisation de la prédiction graphique pour la télé-opération dans un environnement non structuré ; à gauche prédiction et confirmation de la tâche, à droite exécution de la tâche

❖ Le laboratoire IBISC de l'Université Evry Val d'Essonne travaille, depuis plus d'une décennie, sur les systèmes de réalité virtuelle et/ou de réalité augmentée.

Parmi les projets réalisés, le projet AMRA¹³ (Assistance à la Maintenance en Réalité Augmentée). Ce projet regroupe plusieurs partenaires entre autres : le CEA-SRSI, CEA-LCEI, ALSTOM,...etc. Son objectif est d'augmenter la disponibilité de l'information sur le lieu de maintenance en utilisant les techniques de réalité augmentée, et d'apporter aux agents de maintenance une assistance permettant d'accéder sur leurs postes de travail à des informations pertinentes telles que la documentation de maintenance, les modes opératoires et films de montage.

Le premier système réalisé en laboratoire est un système d'assistance à la télé-opération de robots en réalité augmentée via le réseau Internet (le système ARITI¹⁴, 1998-2003).

Le système ARITI est réalisé dans le but de permettre à un Opérateur Humain (OH) de contrôler et de commander un robot à distance.

Développé au LSC (Laboratoire des Systèmes Complexes), ce système est accessible sur le site Web du laboratoire en 1998¹⁵ et il est référencé sur le site WEB de la NASA depuis janvier 2000¹⁶. Il permet d'assister un utilisateur dans la réalisation des tâches de télémanipulation d'objets réels par un robot distant via un navigateur WEB (figure 2.10). Les techniques utilisées ont permis d'apporter à l'opérateur en situation de télétravail une assistance à la perception de l'environnement distant (par la réalité augmentée) et à la commande d'un robot (par des guides virtuels actifs). Ces assistances ont pour objectif l'amélioration des performances (temps, précision et sécurité) des tâches de télémanipulation.

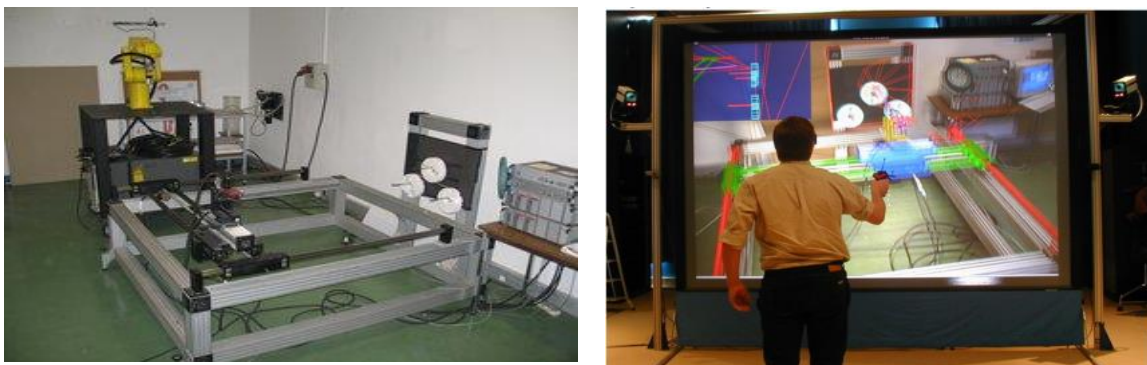


Figure 2.10. *Vue globale du site distant (esclave=Robot), A droite, Manipulation du robot ARITI à l'aide de la plateforme Evr@ (site maître), A gauche,*

¹³ <http://evra.ibisc.univ-evry.fr/index.php/AMRA>

¹⁴ Augmented Reality Interface for Teleoperation via the Internet, <http://ariti.ibisc.univ-evry.fr>

¹⁵ <http://lsc.univ-evry.fr/Projets/ARITI/>

¹⁶ http://ranier.oact.hq.nasa.gov/telerobotics_page/realrobots.html

Le contrôle du robot, se trouvant au laboratoire IBISC, peut s'effectuer depuis n'importe quel emplacement en utilisant un navigateur Web via une interface Homme-Machine (IHM). Cette interface est basée sur le concept de la Réalité Mixée, regroupant ainsi la Réalité Virtuelle (RV) et la Réalité Augmentée (RA).

Le serveur est exécuté donc sur la machine locale. Le client est exécuté sur la machine distante et permet d'une part une supervision du robot et d'autre part une commande de niveau objet de chacun des composants du robot.

La télécommande du robot est bien entendu réservée à un seul client à la fois, il s'agit du premier connecté.

2.7 Milieux hostiles

2.7.1 Définition

Dans le domaine de la robotique, un milieu hostile peut être défini comme un environnement où l'opérateur humain ne peut pas y accéder : température extrême, terrains parsemé de mines, sous-marins, sites nucléaires...etc. Nocifs pour les exploitants, ces milieux sont inaccessibles et présentent un réel danger pour l'être humain. Ce qui a nécessité un recours à la robotique et télé-opération.

2.7.2 Application des systèmes robotiques en milieu hostiles

2.7.2.1 Introduction

La télé-opération est nécessaire lors de l'intervention dans certains lieux inaccessibles ou hostiles, directement par l'homme. C'est le cas de la manipulation d'objets à forte irradiation dans les sites nucléaires, où il faut éviter les risques de contaminations et offrir à l'opérateur de meilleures conditions de travail. En effet, les robots utilisés dans ce domaine résistent bien aux radiations destructrices, et doivent s'équiper d'une électronique durcie qui résiste à des taux nettement supérieurs à l'électronique classique.

Les systèmes humains-robots actuellement développés couvrent une grande variété d'applications et sont généralement très différents les uns des autres. Les techniques d'interaction conçues pour chaque système sont également très différentes.

L'utilisation de robots en milieu hostile se développe et peut permettre de sauver des vies. Les robots démineurs sont utilisés depuis longtemps par l'armée américaine. Le robot présenté ci-dessous est capable de se mouvoir facilement en milieu hostile ou dangereux pour l'homme (figure 2.11).

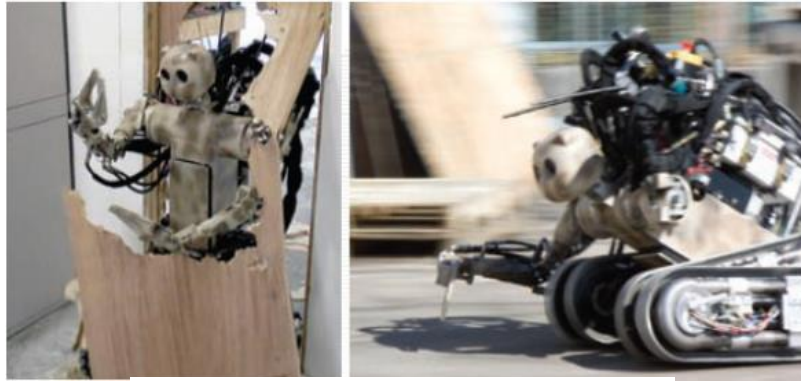


Figure 2.11. Robot démineur

2.7.2.2 Utilisations en milieu nucléaire

Face aux nombreuses missions du domaine nucléaire sur le plan industrielle, médicale...etc. il est toutefois connu pour ses risques d'exposition et de contamination par la radioactivité lors des opérations de maintenance, de la manipulation des sources ou des déchets radioactifs...etc. ceci a entraîné un croissant développement de la sureté nucléaire.

De ce fait, a eu le début du domaine de la robotique pour la prise en charge des opérations d'inspection et de maintenance des matériaux radioactifs.

En matière nucléaire, la robotique connaît plusieurs domaines d'application. Les premiers robots développés aux Etats-Unis dès les années 1950, se résument à des bras de manipulation, dotés de caméras et de projecteurs. Après l'accident de Three Mile Island en 1979, les tâches de surveillance, d'inspection et de décontamination étaient effectuées par des manipulateurs télécommandés, dont les célèbres Fred (pour la surveillance à distance), SISI, ROSA (pour enlever le combustible du cœur) et d'autres.

En France, un des premiers prototypes de robot télé-opéré construit pour intervenir en terrain très radioactif a été imaginé dès 1975 par le CEA. MERIT (Module d'Entrée et de Réparation Interne de Tuyauteries d'eaux) était capable de monter et descendre les escaliers, franchir des obstacles et se saisir d'objets, tout en résistant aux radiations, pour effectuer des observations vidéo, contrôles de sécurité et des opérations de maintenance dans les circuits hydrauliques des centrales nucléaires.

Depuis cette date a eu la réalisation de plusieurs types de robots : En 1983, le robot MIR¹⁷ (Machine d'inspection des réacteurs rapides) pour réaliser des contrôles de soudures par ultrasons entre la cuve principale et la cuve de sécurité de Superphénix (figure 2.12). Ensuite

¹⁷ http://neutronique.free.fr/Module_EIC/EIC_MachineInspectionCuve.pdf

Centaure, commandé à 500 mètres, pour intervenir en milieu hostile après un accident nucléaire et Sherpa, doté de six jambes télescopiques et d'un plateau pour porter des charges lourdes.



Figure 2.12. Robot MIR 'Machine d'inspection des réacteurs rapides'

2.7.2.3 Entretien et maintenance des sites nucléaires

Face aux nombreuses missions du domaine nucléaire sur le plan industrielle, médicale...etc. il est toutefois connu pour ses risques d'exposition et de contamination par la radioactivité lors des opérations de maintenance, de la manipulation des sources ou des déchets radioactifs...etc. Ceci a entraîné un croissant développement de la sureté nucléaire.

De ce fait, a eu le début du domaine de la robotique pour la prise en charge des opérations d'inspection et de maintenance des matériaux radioactifs, la télé-opération est principalement utilisée ainsi que les robots mobiles [61].

Pour bon nombre des tâches à accomplir dans un site nucléaire, les robots viendraient utilement compléter les effectifs; ils libéreraient l'homme de certains des travaux les plus pénibles et, éventuellement permettraient d'accomplir certaines tâches pendant le fonctionnement d'un réacteur ou même d'une centrale nucléaires, réduisant ainsi les temps d'immobilisation nécessaires à l'inspection et à la maintenance.

Le recours à des robots pour l'inspection et la maintenance. Les centrales nucléaires sont motivées, sur le plan économique par les possibilités qu'ils offrent d'augmenter la disponibilité des centrales; accessoirement, ils permettent de réduire l'exposition des personnes. De nombreuses tâches d'inspection et de maintenance ne peuvent être assurées que si le réacteur est à l'arrêt, les niveaux de rayonnement étant trop élevés en conditions normales d'exploitation, même pour des personnes revêtues de vêtements protecteurs.

Un entretien efficace est essentiel à l'exploitation sûre d'une installation nucléaire. Les activités d'entretien comprennent la surveillance, l'inspection, les essais, l'évaluation, l'étalonnage, l'entretien ordinaire, la remise en état, la réparation et le remplacement des pièces.

Le type d'activité d'entretien et sa fréquence doivent tenir compte de l'importance de chaque structure, systèmes et composants du point de vue de la sûreté, des fonctions pour lesquelles il est conçu et de la performance exigée. On peut utiliser divers concepts d'entretien afin d'élaborer une bonne stratégie. La majeure partie des activités d'entretien sont habituellement associées au concept de l'entretien préventif.

1. Entretien préventif

- ❖ Périodique : basé sur la durée, remplacement de pièces ou remise en état. comprend l'inspection périodique et le remplacement pendant la durée de vie.
- ❖ Prédicatif : surveillance, contrôle, test, inspection en service.
- ❖ Anticipé : entretien basé sur l'état, remplacement de pièces, révision, remise à neuf ou modification.

2. Entretien correctif

- ❖ Réparation après défaillance : Réparation, remise à neuf, remplacement ou modification.
- ❖ Défaillance imprévue : inspection et diagnostic des défaillances.

3. Maintenance des sites nucléaires

La maintenance des sites opérationnels ou des futures installations fait souvent appel à des équipements télé-opérés. Chaque problématique étant unique, sa solution doit être adaptée à la géométrie des lieux, à la nature des opérations à effectuer et aux contraintes environnementales.

Cybernetix¹⁸ commercialise une gamme de robots mobiles d'intervention sur colis suspects et engins explosifs pour les artificiers de la Sécurité Civile ainsi que des robots équipés d'une électronique durcie pour l'industrie nucléaire :

- ❖ Le robot EROS, Robot mobile à 6 d.d.l, dédié aux opérations post-accidentelles d'inspection et de maintenance en milieu nucléaire. Ce robot de 320 kg son poids, résiste à des radiations pouvant atteindre 10^6 Rad et possède une d'autonomie de 3 h (figure 2.13).
- ❖ Le robot LMF/MENHIR (figure 2.14), robot modulaire utilisé pour diverses missions dans l'industrie nucléaire telles que surveillance, inspection, maintenance et démantèlement. Ce robot peut être équipé d'un ou deux bras manipulateur (hydraulique, électrique, maître/esclave, retour de force) comme MAESTRO ou MA23. Il est opéré à distance par câble ou par radio utilisant la technologie d'étalement de spectre¹⁹.

¹⁸ Cybernetix est l'un des leaders européens en robotique mobile télé-opérée adaptée aux milieux hostiles et dangereux pour l'homme

¹⁹ Les techniques d'étalement de spectre sont des méthodes de transmission de signal (sur ondes hertziennes) dans lesquelles l'énergie émise avec une ou plusieurs harmoniques est délibérément étalée ou distribuée dans le domaine fréquentiel.



Figure 2.13. Le robot EROS



Figure 2.14. Le robot LMF

Mais le plus célèbre des robots réalisés reste le robot *Maestro* (figure 2.15).



Figure 2.15. Le bras Maestro au travail dans un des ateliers radioactifs de Marcoule (Gard en France)

Issu de 20 ans de R&D entre le CEA, *Cybernetix* et l'*IFREMER*, Maestro a été conçu pour intervenir en milieu hostile, il est capable d'opérer dans des installations hautement radioactives et inaccessibles pour l'homme.

Maestro est un bras articulé tenant une main multifonction à son extrémité. Toutes les articulations utilisent des moteurs hydrauliques asservis à des commandes informatiques encapsulées dans des enveloppes de titane durci²⁰, c'est-à-dire résistant à une radioactivité de 10 000 grays (unité de dose radioactive absorbée, une exposition d'une demi-heure à moins de 1 gray provoque la mort d'un homme)²¹. Sa dextérité, sa précision et sa force sont ses principales qualités. Il peut être utilisé en mode robot (séquence automatique) ou en mode télé opéré avec ou sans contrôle du retour d'effort. A cause de sa résistance aux rayonnements nucléaires, Maestro réalise des opérations de maintenance, d'inspection et de

²⁰ Le titane a l'avantage de pouvoir être lavé de sa radioactivité absorbée une fois sa tâche accomplie.

²¹ Rapport détaillé par Didier Leterq, chef du département de technologie au CEA

nettoyage dans ces milieux qu'ils soient en surface ou sous l'eau avec une décontamination facilitée. Sa capacité de charge peut atteindre 100 kg avec une portée de 2,4 mètres maximale.

4. Gestion des déchets

La *gestion des déchets* est considérée comme le problème majeur du nucléaire pour l'opinion publique.

Les objectifs fondamentaux de la sûreté nucléaire sont de limiter les rejets d'effluents radioactifs dans l'environnement des installations en fonctionnement normal, de prévenir les incidents ou accidents d'exploitation et de minimiser les conséquences de ces événements lorsqu'ils surviennent.

Techniquement, les dispositifs permanents d'une installation nucléaires ont pour fonction de maîtriser la réactivité, de refroidir le combustible en évacuant l'énergie produite, et d'isoler les substances radioactives.

Chapitre 3

Application de la Réalité Augmentée dans la télé-opération du robot

3.1 Introduction

Aujourd'hui, la tendance est de ne pas réaliser des applications purement virtuelles, mais, avec les méthodes de ces dernières, d'augmenter la perception des utilisateurs.

Ce nouveau concept se nomme la réalité augmentée. Dans ce cas, l'utilisateur n'est plus complètement immergé dans un monde non réel, mais des informations virtuelles, viennent renforcer sa perception de la réalité.

Nous cherchons à améliorer la perception de l'opérateur vis à vis de l'environnement réel, généralement par superposition d'images de synthèse sur des images réelles.

Le concept de guides virtuels permet d'assister l'opérateur dans la perception et la commande à distance du robot. Donc l'objet virtuel consiste en l'information à ajouter (augmenté), cette information doit être interactive c'est-à-dire s'il y a une évolution dans la scène l'affichage de cette information, ou objet virtuelle, doit s'adapter automatiquement en tenant compte des phénomènes entrant en scène tels qu'une l'ombre, l'occlusion, ... etc.

3.2 Objectif de la RA

Le principal objectif de la RA est donc de procurer des assistances sensorielles à l'utilisateur dans son monde réel. Ces assistances à l'opérateur ont pour but d'améliorer la précision et assurer la sécurité du déroulement de la tâche de télé-opération. Son principe repose sur le concept de réalité augmentée dans lequel la scène modélisée (monde virtuel) est superposée au monde réel vu par une caméra.

Au sens large du terme, la réalité augmentée est rendue possible par un système capable de faire coexister spatialement et temporellement, un environnement virtuel avec le monde réel. Cette coexistence a pour objectif l'enrichissement de la perception de l'utilisateur de son environnement réel par des augmentations visuelles (texte, vidéo, photo, image animées 2D ou 3D), sonores ou haptiques. Pour cela, elle définit de nouvelles perspectives d'interaction qui permettent d'exploiter les différents contextes d'utilisation pour une meilleure intégration des mondes réels et informatiques.

L'utilisation du modèle 3D permet d'anticiper une situation réelle ou prévoir les cas dangereux, et de les simuler complètement avant de les mettre en pratique. Ses applications sont multiples et touchent de plus en plus de domaines, tels que les jeux vidéo, l'éducation par le jeu, le cinéma et la télévision, mais aussi l'industrie : conception, design, maintenance, assemblage, pilotage, robotique et télérobotique, implantation, étude d'impact... Ses

domaines favoris sont la télé-opération, la télé-robotique ou la télé-présence, qui permet l'intervention et les manipulations dans des milieux lointains et/ou hostiles.

D'une façon générale, un système de RA doit répondre aux principales caractéristiques qui définissent ce système [62] à savoir :

- ❖ Combiner réel et virtuel dans un environnement réel ;
- ❖ Interactive en temps réel ;
- ❖ Alignement²² virtuel et réel en3D (Registration).

La mise en œuvre d'un système de Réalité Augmentée (RA) nécessite la modélisation de l'environnement du robot. Plusieurs types de capteurs sont utilisés pour véhiculer l'information à l'opérateur pour pouvoir effectuer sa tâche.

Le moyen le plus utilisé étant la vision au moyen d'une caméra, celle-ci permet non seulement de transmettre l'image de la scène mais aussi, moyennant des techniques, à localiser les objets, estimer les dimensions 3D, ...etc.

Pour que la superposition d'images puisse être effectuée correctement, il faut connaître la transformation mathématique qui fait passer des coordonnées tridimensionnelles de l'objet aux coordonnées bidimensionnelles de son image et vice-versa. Cette transformation est obtenue par la calibration de la caméra [63].

3.3 Réalité augmentée et vision artificielle

La commande référencée vision 'Asservissement visuel' consiste à contrôler les mouvements d'un système robotique en intégrant des informations fournies par des capteurs ou par des caméras. L'emploi des caméras est préférable car elles couvrent un champ de vision important et tous les détails dans ce champ de vision constituent une source d'information, les capteurs quant à eux couvrent un champ de détection limité.

Le problème du contrôle robotique par la vision artificielle réside dans l'association intelligente d'un système de perception et d'un système d'action en vue de l'exécution de tâches. Cette association est univoque. Le système sensoriel de vision artificielle fournit au système moteur robotique les informations utiles à la réalisation des tâches, mettant en évidence une articulation intelligente de la perception sur l'action. De nombreux travaux en vision pour la robotique ont porté sur l'interaction entre l'homme et la machine durant ces dernières années.

²² La registration ou l'alignement précis avec l'objet réel dans la scène est le paramètre le plus important dans la réalité augmentée et surtout dans des applications spécifiques comme dans le cas des opérations chirurgicales. L'amélioration de la registration continue de faire l'objet de plusieurs travaux actuellement [64, 65].

La vision implique la mise en œuvre d'au moins 3 sous-systèmes : un capteur visuel, un dispositif d'acquisition/traitement et un système mécanique articulé SMA. Le capteur visuel utilisé dans notre travail est une caméra (figure 3.1).

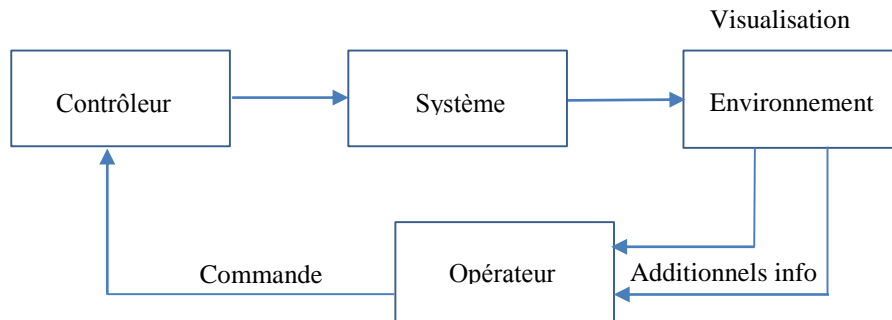


Figure 3.1. Schéma synoptique d'un système RA

Le dispositif d'acquisition et de traitement "ordinateur" permet de traiter l'information visuelle numérisée issue de la caméra. Une caméra vidéo comprend un système optique composé de lentilles qui forme une projection bidimensionnelle d'une scène de l'environnement observée sur un plan image photosensible (ou rétine). Chaque caméra délivre, à cadence vidéo, une copie de l'image optique reçue par sa rétine électronique. La rétine d'une caméra est divisée, en abscisse et ordonnée, en un certain nombre de points appelés pixels. Au cours du traitement de l'image, l'usage de l'informatique implique cette division en un nombre défini mais programmable de pixels. Souvent, c'est le même système qui est chargé du traitement de l'image et de la commande.

Cette commande qui résulte du retour visuel est envoyée au système mécanique qui est le robot manipulateur.

3.3.1 Utilisation des algorithmes de traitement d'images pour l'extraction des grandeurs d'objets réels dans l'environnement des opérations.

Une multitude de procédés de traitement d'images sont représentés dans la littérature entre autres la méthode de Tsai, Heikkila et Zhang [66,67]. Ces trois méthodes ont été analysées et comparées par Wei Sun et al [68]. A l'exception dans le choix du paramètre de la distorsion, ces méthodes utilisent le même modèle projective de la caméra appelé 'pinhole'. Les résultats de cette comparaison ont montré que la méthode de Zhang est mieux adaptée en termes de simplicité et précision des résultats, elle utilise l'approche non coplanaire ou son principe consiste à utiliser un échiquier à coller sur un plan plat et rigide afin d'éviter les déformations, puis prendre plusieurs vues de l'échiquier sous différents angles et orientations.

A partir de ces images, le logiciel calcule les paramètres intrinsèques et extrinsèques de la caméra en utilisant quelques fonctions de Matlab orientées à ceci. En général, l'extraction des paramètres intrinsèques se fait une seule fois car ces paramètres sont constants, alors que les paramètres extrinsèques sont mis à jour au fur et à mesure du mouvement du système, car ces paramètres sont des paramètres géométriques ou la position du système de coordonnées de la scène.

3.3.2 Essais en laboratoire concernant la mesure des grandeurs et de la profondeur d'objets dans l'image de la scène utilisant la vision.

Dans le but de mettre au point un système de télé-opération en réalité augmentée, nous avons conçu en laboratoire, après une étude approfondie des méthodes utilisées dans la littérature, un banc d'essais de calibration de la caméra nécessaire pour la mesure 2D et 3D de la caméra montée sur le robot. La figure 3.2 montre le procédé de calibration utilisé. La méthode de calibration que nous avons utilisée est la méthode de Zhang offrant des caractéristiques fiables et précises. La caméra utilisée dans notre cas est une Webcam3100²³, elle constitue un atout, vu le volume de son champ de mesure 3D, sa résolution (dixièmes de millimètres), et son faible coût.



Figure 3.2. Banc de calibration géométrique de la Caméra

Se basant sur les fonctions de Matlab en ce qui concerne la vision et le traitement d'image, Plusieurs essais ont été effectués en laboratoire pour procéder à la calibration de notre caméra.

²³ Les paramètres de la caméra obtenue après calibration : Camera Intrinsics, Lens Distortion, Camera Extrinsics, Accuracy of Estimation et Calibration Settings.

3.3.3 Calibration géométrique

Pour la mesure des distances 2D (Plan) il faut dans une première étape, estimer les paramètres extrinsèques (géométriques) de ce plan (rotation et translation par rapport au centre optique de la caméra). Ce plan doit être maintenu fixe pour toutes les mesures à effectuer dans ce plan.

3.3.3.1 Calibration de la caméra (estimation de ses paramètres intrinsèques et extrinsèques)

La calibration de la caméra est une procédure nécessaire pour extraire ses paramètres intrinsèques et extrinsèques. Cette étape consiste, selon la méthode, à prendre des images d'un motif géométrique (figure 3.3) avec différentes positions et angles. Les points constituant ces images seront utilisés pour résoudre un système d'équations selon un algorithme pour extraire les paramètres intrinsèques et extrinsèques de la caméra.

Dans le but de réaliser de meilleurs résultats lors des prises de vues des, certains éléments sont à prendre en compte tels que :

- ✚ la taille de l'échiquier ne doit pas être inférieure à 20% de la taille de l'image ;
- ✚ le nombre de carrés dans l'échiquier doit être différent selon x ou y (carré blanc et carré noir) ;
- ✚ les prises de vues doivent être à peu près à la même distance où les mesures vont être prises par la suite.

Une fois fixés, les paramètres intrinsèques ne changent plus sauf dans le cas où on change le plan de mesure. Ils sont donc sauvegardés pour une éventuelle utilisation. Les paramètres extrinsèques ou de position sont mis à jour au fur et à mesure lors du mouvement comme dans le cas du tracking.

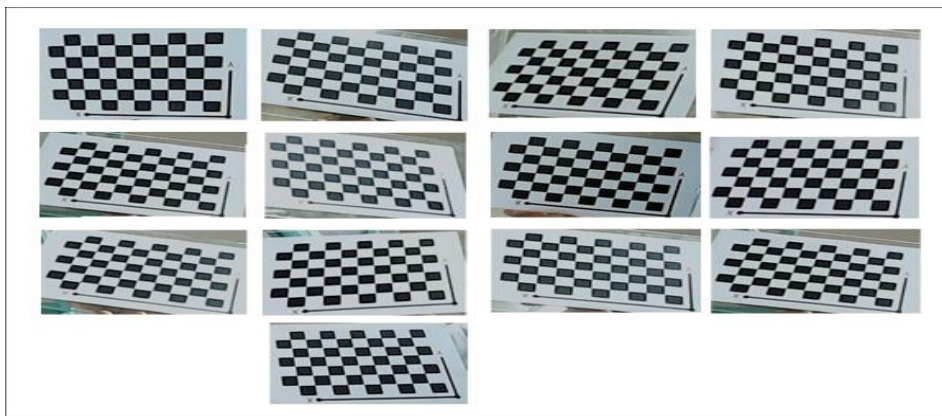
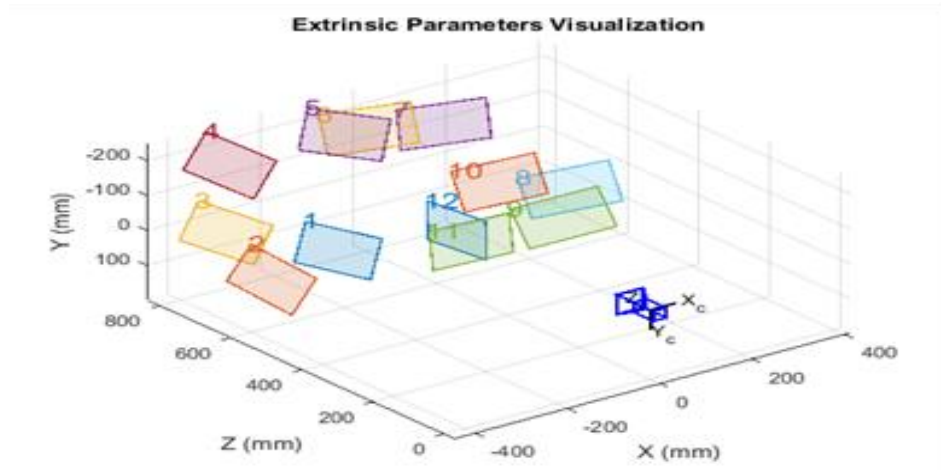


Figure 3.3. Présentation de plusieurs prises de vue de motif géométrique

3.3.3.2 Paramètres intrinsèques et extrinsèques de la caméra

Après traitement par le logiciel de toutes ces images, la figure 3.3 affiche toutes les images dans leur position de prise, ce qui permet le calcul des paramètres extrinsèques et intrinsèques de la caméra utilisée. Le logiciel permet également de calculer l'erreur liée à chaque image vue par le logiciel utilisé (Matlab) et montrée par le diagramme de la figure 3.4.



*Figure 3.4. Visualisation des paramètres extrinsèques.
Calibration de la caméra*

3.3.3.3 Mesure de distances 2D

Des méthodes de calibration de la caméra ont été utilisées pour l'acquisition de l'information des dimensions d'objets dans une scène en vue de son exploitation. Ces méthodes se basent généralement sur deux approches:

- Coplanaire (un seul plan 2D) ;
- Non coplanaire (3D). cette approche utilise plusieurs procédés de calibration, entre autres :
 - ✚ Calibration basé sur une référence externe (World-reference based) ;
 - ✚ Méthode basé sur l'invariance géométrique (Geometrie-invariant-based methods) ;
 - ✚ L'approche autocalibration (self-calibration) ;
 - ✚ Méthode de la calibration implicite.

Le logiciel Matlab propose quelques fonctions pour préparer et calculer les paramètres intrinsèques et extrinsèques de la caméra exécutés dans les séquences suivantes :

- ✚ `[imagePoints, boardSize] = detectCheckerboardPoints(files);`
- ✚ `worldPoints = generateCheckerboardPoints(boardSize,squareSize);`
- ✚ `cameraParams=estimateCameraParameters(imagePoints,worldPoints);`
- ✚ `showReprojectionErrors(cameraParams.`

Dans cette étape, les paramètres extrinsèques et intrinsèques sont disponibles, la mesure des distances sont possibles moyennant la détection des motifs dans l'image (détection des coins par exemple). Dans cet exemple (figure 3.5), nous avons pu mesurer dans notre laboratoire la largeur et la longueur d'un rectangle avec une *précision de 1mm*. Cette précision sera améliorée davantage en améliorant les prises de vue lors de la calibration.



Figure 3.5. *Mesure de distances pour une prise de vue*

Toutes ces fonctions sont réalisées en utilisant le logiciel Matlab, qui emploie des boites à outils "Toolbox" possédant des fonctions spécifiques de traitement d'images "toolbox Image-Acquisition et toolbox Image-Processing". Ceci permet de réaliser du calcul numérique et de tracer des graphiques pour visualiser et analyser les données.

Exécuter dans l'éditeur de commande de Matlab, certaines fonctions sont nécessaires et servent comme premières étapes au déroulement du programme, parmi lesquelles :

- a) `imaqhwinfo` ('winvideo'): Cette fonction permet l'obtention d'informations sur le matériel et les pilotes installés ;
- b) Initialisation matérielle de la capture utilisant la subroutine `init.m` ;
- c) Acquisition d'images (`acq.m`) : Le matériel d'acquisition est une WebCam reliée au PC via le port USB, un pilote Windows spécifique permet de communiquer entre le PC et la webcam. L'acquisition d'une image sous Matlab passe par la création d'un objet d'entrée vidéo en utilisant la fonction `videoinput`. Cette fonction affiche également les principales propriétés de l'objet d'entrée vidéo créé, qui est de type structure sur lequel il est possible de régler des paramètres liés au périphérique telles que format de l'image, l'espace de codage de l'image (ex : RGB et niveaux de gris) et des paramètres liés à l'acquisition comme la luminosité (Brightness), le contraste (Contrast), le temps

d'exposition (Exposure), etc. L'acquisition d'une seule image se fait avec la fonction `getsnapshot (I= getsnapshot(obj))`.

Et, en plus des fonctions dédiées au calcul des paramètres intrinsèques et extrinsèques de la camera, d'autres fonctions sont aussi nécessaires comme "estimationErrors" qui permet d'estimer l'erreur du modèle obtenu (figure 3.6).

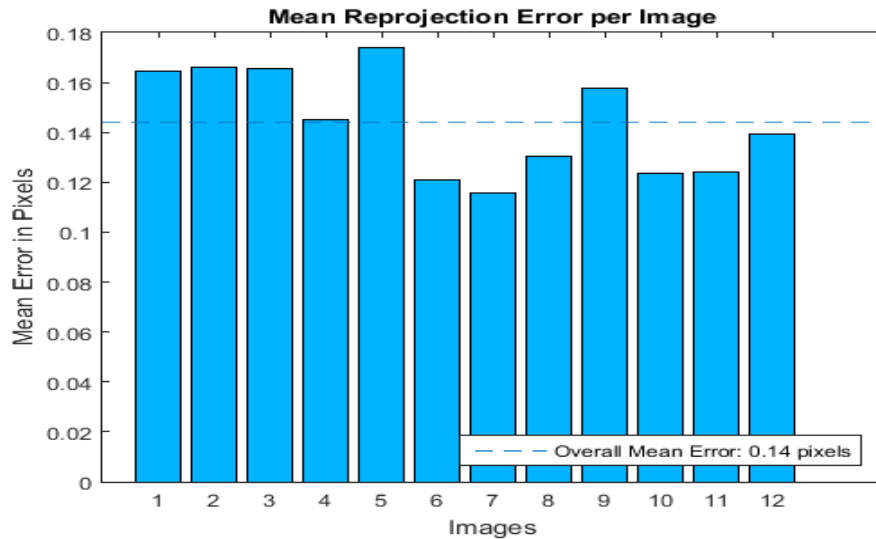


Figure 3.6. Estimation de l'erreur des paramètres extrinsèques de la caméra

3.4 Apport de la réalité augmentée

Le principal problème lié à la commande des systèmes télé-opérés est l'instabilité due entre autres au retard de communication entre l'opérateur (le maître) et le système robotique (l'esclave), qui induit à une information incomplète des deux côtés. Dans ce contexte, des recherches ont été menées afin de pouvoir palier à ce phénomène contraignant et plusieurs analyses et approches en télé-opération ont été proposées, cherchant chacune à fournir un ensemble de techniques pour le réglage automatique des contrôleurs en temps réel afin d'atteindre ou de maintenir un niveau souhaité de performance du système de contrôle. Parmi ces approches :

- ❖ Le contrôle par la méthode "Move-and-wait" [69,70] ;
- ❖ Heuristique approche ;
- ❖ Logiciel basée sur la télé-opération (Software-based teleoperation) [71] ;
- ❖ Le contrôle prédictif MPC (Model Predictive Control) [72-75]
- ❖ Le contrôle par mode de glissement (Sliding-mode) [76-79] ;
- ❖ etc.

Toutes ces méthodes ont contribuées à minimiser le retard de transmission ainsi que d'autres problèmes liés à la télé-opération tels qu'une occlusion ou un manque d'éclairages.

Toutefois, si l'objectif final est d'améliorer la performance d'une tâche, la réalité augmentée s'avère la méthode la plus adaptée et a prouvé son efficacité car elle offre de meilleures performances en télé-opération et le time-delay est nettement amélioré. La facilité de l'utilisateur à la perception de l'environnement et la réalisation de la tâche est nettement améliorée [80,81].

Les apports de l'approche RA sont multiples :

- i. La possibilité de représenter l'environnement de travail uniquement par un aspect fonctionnel. Le but recherché est alors une transformation du site de travail en un environnement virtuel où les tâches à effectuer sont simples à réaliser pour un opérateur non-spécialiste de la télé-opération.
- ii. La possibilité de préparer et de planifier des missions de télétravail hors ligne afin d'acquérir une expertise (entraînement) ou de parer à d'éventuels accidents (prédiction).
- iii. La possibilité de concevoir des interfaces de commande ergonomiques permettant une interaction multimodale avec l'environnement virtuel.

3.5 Interaction : Télé-opération et Réalité Augmentée

Plusieurs types de capteurs sont utilisés pour véhiculer l'information à l'opérateur pour pouvoir effectuer sa manipulation, mais la vision au moyen d'une caméra est généralement le moyen utilisé car elle permet non seulement de transmettre les images de la scène mais aussi de constituer un moyen d'interaction pour l'utilisateur.

L'information ajoutée doit s'aligner 'régistration' avec l'objet réel dans la scène du système de réalité augmenté surtout dans des applications spécifiques comme dans le cas des opérations chirurgicales. Et pour une meilleure précision concernant l'information affichée, la solution est le suivi par vision à l'aide de repères visuels. L'environnement est modifié par la pose d'un certain nombre de marqueurs de tailles variables disposés à des endroits connus, qui vont permettre de se localiser dans l'environnement. Un algorithme de traitement d'image suit ces marqueurs et calcule la pose de la caméra en temps réel et permet de reconnaître la forme d'objet et le localiser dans l'espace.

3.6 Application de la RA

L'objectif principal étant de permettre au robot de manipuler des objets à distance, et d'une façon interactive en augmentant sa perception avec des objets virtuels. Cette augmentation de la réalité provient de plusieurs issues : les capteurs, la modélisation 3D de l'environnement ou

par la vision, qui est la méthode choisie pour notre application. Son principe consiste à extraire la localisation des objets en 3D en utilisant les caméras pour la vision de l'environnement. Dans cette optique, on utilise *l'annotation*²⁴, qui consiste en la superposition de l'image réel du robot manipulateur avec l'information de la localisation précise du préhenseur. Cette information de la localisation est transmise par le capteur au PC via la carte embarquée.

De ce fait, l'installation de ce système pour la réalité augmentée doit satisfaire certaines conditions comme les mesures des distances et des profondeurs, le tracking (Poursuite). Pour cela, le modèle projective des caméras (paramètres intrinsèques et extrinsèques) et l'algorithme de tracking sont à développer. Néanmoins, lors d'une registration, des erreurs apparaissent qui sont soit des erreurs dynamiques ou bien des erreurs liées au tracking.

3.7 Domaines d'application de la réalité augmentée

En robotique, l'intérêt de la RA dépend du domaine d'application. Cette application peut concerner plusieurs environnements tels que :

3.7.1 La réalité augmentée au service de la sécurité nucléaire

Le domaine où la sécurité est une priorité, c'est bien l'industrie nucléaire. Des opérations de contrôle de conformité sont effectuées en permanence pour prévenir tout incident ou l'utilisation de la réalité augmentée facilite les opérations de contrôle.

Généralement dans ce domaine la réalité augmentée utilise l'annotation pour mettre à la disposition de l'opérateur différents types d'informations pour assurer la sécurité et la précision de sa tâche de manipulation [82-84].

3.7.2 Le domaine médical

Dans le domaine médical, de nombreux systèmes fonctionnent déjà sur le principe de la réalité augmentée pour simuler informatiquement un environnement réel qui permet à l'utilisateur de s'y immerger et d'y interagir. On cherche à obtenir des simulations de plus en plus réalistes des tissus d'un patient et de leurs composants anatomiques (par exemple nerfs, veines, artères), afin de guider au plus près les gestes chirurgicaux. Le domaine médical concerne beaucoup plus la chirurgie orthopédique, l'idée pourrait être de s'entraîner à manipuler un robot chirurgical grâce à la réalité augmentée, avant de passer à une opération réelle.

²⁴ Un opérateur télérobotique utilise une image visuelle de l'espace de travail à distance pour guider le robot. Une annotation de la vue apparaîtrait comme si la scène était devant l'opérateur.

3.8 Conclusion

La réalité augmentée a trouvé sa place et a apporté des solutions à des problèmes rencontrés dans de nombreux domaines d'application. Plus précisément dans le domaine de la télé-opération où la technique de RA est adoptée afin de palier certains problèmes liés généralement à la distance qui sépare le site maître (Opérateur) du site esclave (Robot), et aussi au délai de transmission qui s'avère parfois énorme.

En effet, la télé-opération assistée par la réalité augmentée offre plusieurs avantages²⁵ :

1. Il ne nécessite pas de construire à l'avance la modélisation de l'environnement d'exploitation distant. Il convient à un environnement structuré et non structuré, même s'il est complètement inconnu de l'environnement d'exploitation distant.
2. Ce qui améliore l'efficacité et la précision de la télé-opération.
3. Affichage rehaussé : L'image vidéo est augmentée par la présence d'informations abstraites comme les distances entre objets du site distant, l'environnement du robot et l'affichage des trajectoires générées.

²⁵ Zhang a développé une plate-forme de système de télé-opération basée sur la réalité augmentée pour améliorer la stabilité du système, facilitant ainsi la tâche de l'opérateur, observation et fonctionnement.

Chapitre 4

Mise en œuvre de la télé-opération du robot manipulateur expérimental "SEROR" via le Réseau Internet

4.1 Introduction

Un robot manipulateur est constitué de deux sous-ensembles distincts ; une structure mécanique articulée (SMA) et de un ou de plusieurs organe terminal. L'organe terminal est le dispositif d'interaction, fixé à l'extrémité mobile de la structure mécanique et destiné à manipuler des objets.

Le SMA est une séquence de corps rigides (appelées liens) qui sont reliés entre eux par des articulations. La principale distinction entre les différents robots concerne leur structure mécanique, ou de la façon dont les liaisons sont reliées. Ces liaisons sont en général en série pour les robots sériels, où chaque corps est mobile par rapport au précédent. Cette mobilité s'exprime en termes de degrés de liberté (d.d.l), qui est par définition le nombre de mouvements indépendants possibles d'un corps par rapport à un autre. Quant aux robots parallèles, ceux-ci consistent en une plateforme fixe (base) reliée à une plateforme mobile (nacelle) par l'intermédiaire de segments.

Le rôle de la structure mécanique articulée est d'amener l'organe terminal dans une situation (position et orientation) donnée, selon des caractéristiques de vitesse et d'accélération données. Il est désigné par différentes appellations : préhenseur, outil, effecteur, ou pince.

4.2 Objectif du chapitre

Les recherches réalisées concernent principalement le développement de méthodes et d'outils pour l'interfaçage efficace d'un opérateur dans le cadre de la télé-opération d'un robot via Internet.

L'objectif étant de permettre à l'opérateur humain d'effectuer diverses opérations de télémanipulation et de télémaintenance dans les sites distants, inaccessibles ou dangereux pour les humains. Ceci est réalisable à l'aide d'un robot manipulateur, en plus d'une interface graphique permettant la supervision par l'opérateur du système télé-opéré en se basant sur une ou plusieurs caméras situées dans l'environnement de travail.

Dans ce chapitre, nous faisons l'étude du robot SEROR, un robot expérimental à trois degrés de liberté avec une pince à son extrémité pour la préhension des objets. Pour le contrôle du robot, nous avons développé une interface graphique en langage Java qui permet la communication entre l'opérateur humain et le robot distant. Quelques modes de contrôle réalisés par ce robot sont décrites dans ce rapport.

4.3 Architecture d'un système de contrôle via Internet

La structure d'un système robotique contrôlé par Internet, est constituée de deux unités de communication entre le site maître et le site esclave (Client/serveur) ainsi que des systèmes de supervision et de télé-opération. La communication entre les deux sites se fait via le réseau Internet ou Intranet (figure 4.1).

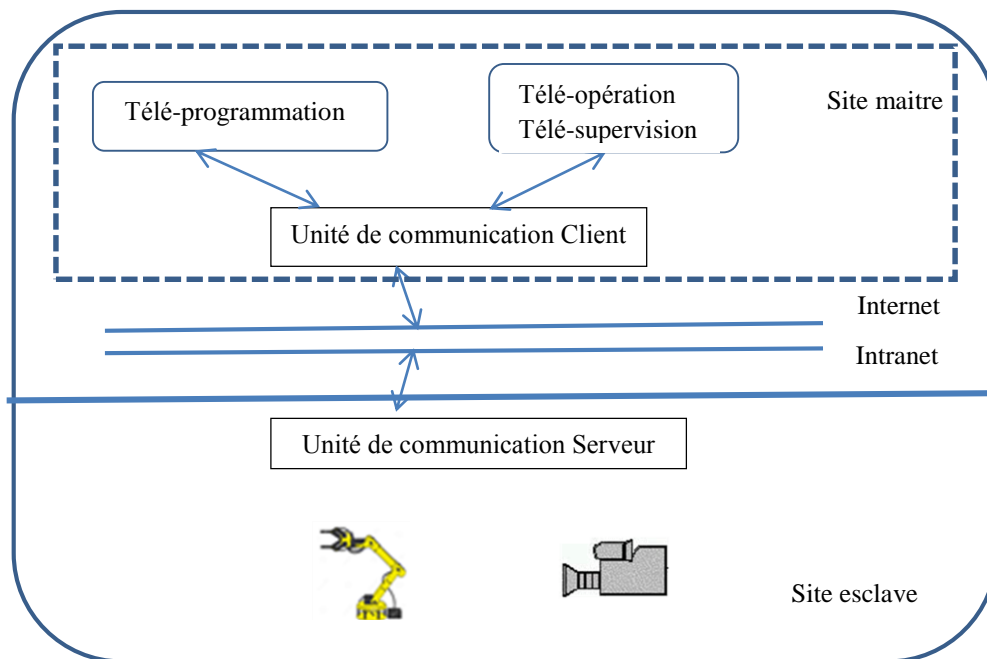


Figure 4.1. Architecture fonctionnelle du système

Dans le site esclave se trouve un robot et une caméra vidéo ainsi que l'unité de communication serveur. Ce dernier est constitué de deux serveurs, un pour les images vidéo et un autre pour les consignes robot. Le premier serveur se charge de capturer des images via la carte d'acquisition vidéo, de les compresser et ensuite de les envoyer directement au client connecté. Le deuxième serveur quant à lui est chargé de récupérer les ordres envoyés par le client et les transmet directement au robot pour exécution. Il peut se charger éventuellement de transférer des informations depuis le robot vers le client.

Dans le site maître se trouve un opérateur humain, un PC connectée à Internet, une interface Homme Machine (IHM) et une unité de communication client. L'unité permet l'échange d'informations avec le site esclave et l'IHM gère les interactions entre l'opérateur humain et le module de communication client. Aussi, l'IHM permet à l'opérateur de spécifier le mode de travail désiré: Télé-opération, Télé-programmation et Télé-supervision & Télé-coopération,...etc. La station de contrôle commande est constituée de deux calculateurs reliés

en réseau Ethernet. Les différents logiciels organisés en client/serveur communiquent entre eux via Internet, en utilisant les sockets selon les protocoles TCP/IP.

4.4 Télé-opération via Internet du robot SEROR

4.4.1 Description du système opérationnel

Le système de télé-opération comprend un robot manipulateur "SEROR", un système de caméra de surveillance auto contrôlable "PTU", et une interface de coordination entre l'opérateur et les deux systèmes (figure 4.2). Le robot est à trois degrés de liberté, il permet donc une visualisation de l'environnement de travail en 3D. Muni à son extrémité d'une pince, lui permet l'exécution de différentes tâches comme la prise et le déplacement des objets, opération pick-and-place.

La caméra constitue le système d'observation pour la transmission des commandes vers le robot et la réception des images du lieu où s'effectuent les opérations. Nous utilisons une WebCam montée sur un système motorisé lui conférant une rotation autour d'un axe horizontal et une deuxième rotation autour d'un axe vertical. Ce qui permet à l'opérateur la perception des déplacements des éléments constituant le robot, à savoir le bras, l'épaule et la pince.

Le système offre des facilités à l'opérateur pour la supervision et le contrôle des tâches à réaliser. Un programme écrit en java représente l'interface entre l'opérateur et le système à télé-manipuler. L'ensemble est conçu conformément à la technique *Human-centred*.

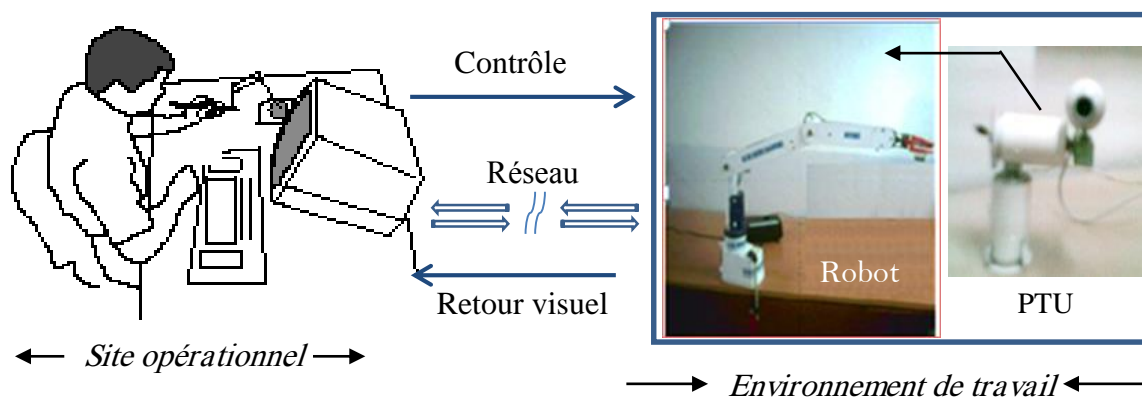


Figure 4.2. Illustration du système de télé-opération du robot SEROR via Internet

Dans le site opérationnel, on retrouve le robot SEROR, la caméra vidéo et le module de communication avec le serveur ; ce dernier contient les programmes de contrôle du robot et les routines de capture et de traitement d'image.

L'ensemble du système est commandé à partir du PC à l'aide d'un programme écrit en java. Une interface graphique permet la transmission des commandes et la réception des images ainsi que d'autres informations liées à l'environnement des opérations (figure 4.3).

4.4.2 Interface graphique "GUI"

Le contrôle et la supervision du robot se font via l'interface du panneau de commande (figure 4.3). Une interface simple se construit pour fournir autant d'informations que possible pour la télé-opération. Cette interface utilisateur est composée de plusieurs Java Applets, elle réalise plusieurs fonctionnalités. En plus d'assurer la connexion et la déconnexion du client au serveur, il permet séparément l'ordre des différents éléments du robot manipulateur : le corps, l'épaule, le bras du robot manipulateur et l'ouverture ou la fermeture du préhenseur "effecteur". Il permet également la commande et l'orientation de la caméra dans les quatre directions (Est, Ouest, Nord et Sud).

La connexion au système robotique peut s'effectuer de deux manières, indirecte ou directe. Dans le premier cas, l'opérateur distant accède au système via le serveur Web à l'aide des sockets TCP/IP, et dans le second cas (connexion directe), l'opérateur utilise directement le système pour télé-opérer le robot et réaliser des tâches de manipulation ou tout simplement faire de la simulation.



Figure 4.3. *Panneaux de contrôle (Robot & PTU). Contrôle du robot à gauche, Netmeeting au milieu, commande de la caméra à droite*

Le retour d'information visuel est effectué en utilisant les images transmises par la caméra au moyen de NetMeeting. Ceci permet à l'opérateur de localiser la prise par rapport aux objets à manipuler.

Au moyen de cette interface utilisateur graphique, un programme de simulation a été mené pour tester la mise en œuvre de la commande du robot manipulateur en utilisant les modèles géométriques inverse et direct, ce qui s'avère tout à fait suffisant pour les applications de type pick-and-place.

4.4.3 Intercommunication via les Sockets

Notre application est de type Client-Serveur. Elle peut être subdivisée en deux parties qui sont le Serveur du côté robot et le Client du côté opérateur. La communication entre ces deux parties se fait au moyen de sockets via un réseau informatique type LAN en utilisant le protocole de communication TCP/IP.

Le serveur constitue un moyen de communication entre l'opérateur et le site distant contenant, en plus du robot et de la caméra, les programmes d'actionnement du robot ainsi que les routines de capture des images.

L'exécution d'une tâche en robotique nécessite de positionner et de diriger l'effecteur, selon les instructions de l'opérateur. Celui-ci doit spécifier les coordonnées opérationnelles ou articulaires de la pince (figure 4.4).

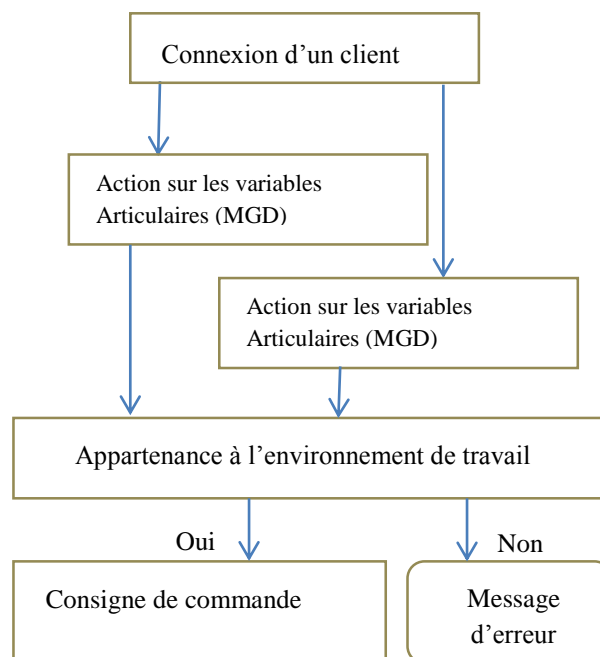


Figure 4.4. Algorithme général de l'application

L'opérateur, à partir du site local, gère les commandes nécessaires au déroulement de la tâche et se prépare aux éventuelles modifications en cas de nécessité. Le programme informatique, réalisé en langage java, rassemble une suite d'instructions interprétables par le contrôleur du robot (circuit électrique) qui générera les signaux à envoyer aux moteurs pour

effectuer les mouvements envisagés. Et une fois toutes les informations recueillies, le robot met en exécution la tâche.

4.4.4 Commande du bras manipulateur

La commande représente l'outil de pilotage de la structure mécanique articulée, c'est une partie très importante dans la robotique. En effet, l'exécution d'une tâche en robotique nécessite au préalable de positionner et d'orienter l'organe terminal "l'effecteur" en fonction des consignes de l'opérateur. Celui-ci doit préciser les coordonnées opérationnelles de la pince dans le but de positionner le bras manipulateur dans une configuration désirée. Ce travail nécessite d'une part de déterminer les modèles géométriques direct et inverse du bras manipulateur, et d'autre part d'être capable de commander ce bras à l'aide d'une interface informatique.

L'asservissement que nous appliquons pour ce robot est un retour d'information au niveau du PC, à partir d'un programme de commande réalisé (figure 4.5). Les trois moteurs du robot ainsi que celui de la pince sont entièrement commandés par un programme informatique écrit en langage JAVA. Ce programme permet d'avoir les solutions pour les modèles géométriques, directs et inverses.

En introduisant les coordonnées cartésiennes ou articulaires par l'intermédiaire du clavier sur l'interface graphique et après les phases calcul et exécution, un signal électrique est transmis à la plaque électronique pour commander l'alimentation des actionneurs afin d'avoir la position désirée.

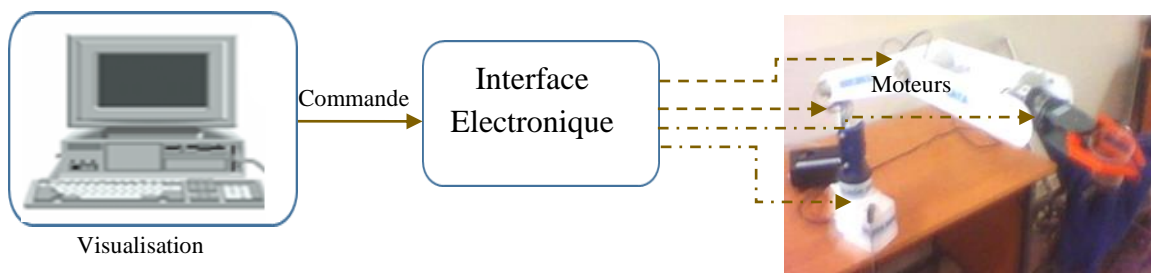


Figure 4.5. Commande en boucle ouverte du robot SEROR

Le mode de contrôle utilisée pour notre robot correspond à un mode de commande en boucle ouverte. Pour effectuer des tâches de type "pick-and-place", ceci se traduit par des commandes en position. L'expérimentation de ce mode de commande a montré une grande sensibilité aux perturbations internes et externes. Il en résulte un manque de précision en commande en position.

En vue d'améliorer la précision de la commande en position du robot, il est nécessaire de réaliser un asservissement en position pour toutes les articulations. Une solution qui a été adoptée au laboratoire LATA est la réalisation d'un asservissement par intégration d'un opérateur humain. Ceci a permis de mener de nombreuses expériences en robotiques ainsi qu'en télé robotiques.

4.4.5 Intégration de l'opérateur humain

L'intégration de l'opérateur humain dans la chaîne de mesure et de commande permet de travailler en boucle fermée améliorant et compensant les écarts de position. Deux solutions testées au laboratoire, une première solution intègre l'opérateur humain en utilisant sa vision directe du robot et de l'environnement de travail. La deuxième solution est surtout utilisée en télé-robotique lorsque l'opérateur est éloigné du site de travail et reçoit les informations à partir d'une séquence vidéo transmise par une caméra liée au site de travail du robot (figure 4.6).

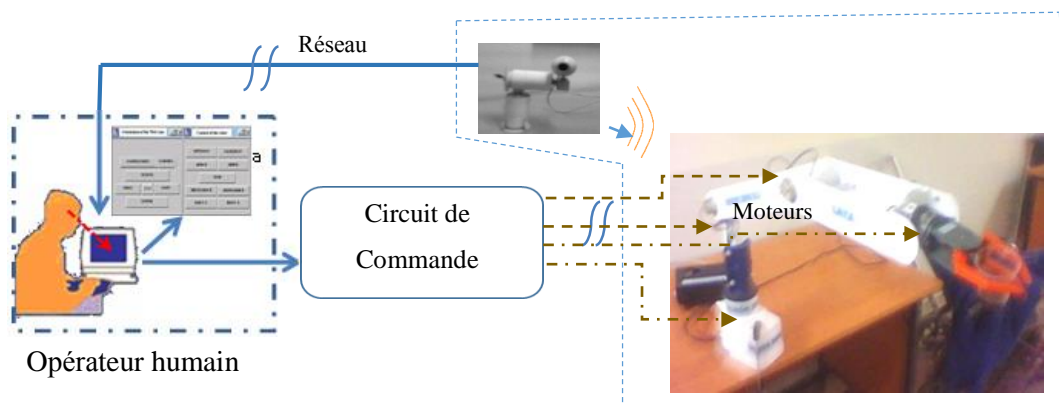


Figure 4.6. Asservissement intégrant l'opérateur et une caméra

Ce mode de commande, intégrant l'opérateur dans le système d'asservissement et une caméra, est utilisé lorsqu'il y a distance entre l'opérateur et le robot ou bien lorsque la vision directe n'est pas facile ou possible même en site local. L'opérateur humain reçoit de l'information grâce à une séquence vidéo transmise par une caméra située dans le site de travail. La séquence vidéo est visualisée en temps réel sur l'écran du PC. Son observation permet à l'opérateur d'estimer le positionnement du robot et d'effectuer les corrections nécessaires.

Les essais effectués en site local ou à distance ont concernés des opérations du type 'Pick-and-place'. Dans ce cas, nous avons simplement implémenté le modèle géométrique direct et inverse, avérés suffisants pour les opérations de manipulation à distance de nos applications.

La caméra, qui sert uniquement à la visualisation du site de travail, utilise aussi les deux modèles géométriques "direct et inverse".

4.4.6 Modélisation du robot SEROR

La modélisation des robots nécessite une méthode adéquate pour la description de leur morphologie. Plusieurs méthodes et notations ont été proposées [85]. Le plus commun est celui de Denavit-Hartenberg [86], développé pour de simples structures ouvertes.

A cette fin, les modèles géométriques direct et inverse sur lesquels la commande du robot sera basée sont présentés schématiquement en figure 4.7. Celles de la caméra sont schématisées en figure 4.8.

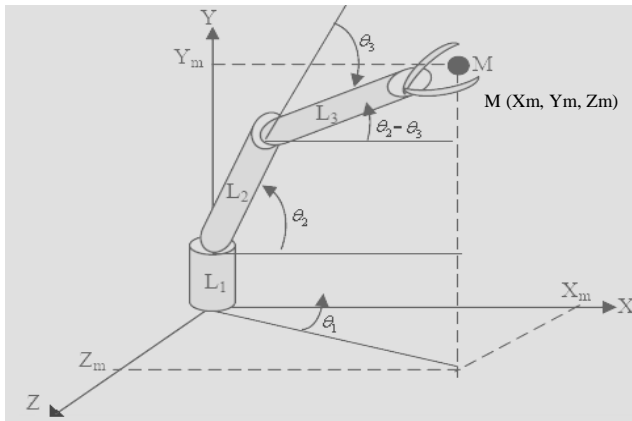


Figure 4.7. Coordonnées du robot

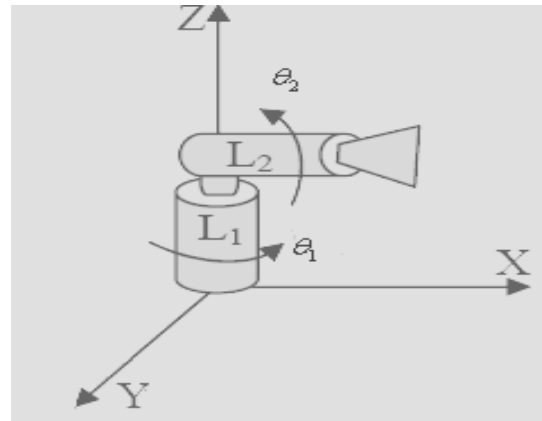


Figure 4.8. Coordonnées de la caméra

Les programmes de contrôle et de la simulation du robot sont écrits en langage java. L'opérateur exécute la tâche au moyen d'un panneau graphique en cliquant sur le bouton correspondant du panneau de contrôle présenté en figure 4.3.

4.4.6.1 Modèle géométrique direct (MGD)

La conception et la commande des robots nécessitent le calcul de certains modèles mathématiques, tel que: les modèles de transformation entre l'espace opérationnel (dans lequel est définie la configuration de l'organe terminal) et l'espace articulaire (dans lequel est définie la configuration du robot). Parmi ces modèles:

1. Les modèles géométriques directs et inverses qui expriment la position de l'effecteur terminal en tenant compte des variables communes du mécanisme ou des variables cartésiennes ;

2. Les modèles cinématiques directs et inverses qui expriment la vitesse de l'effecteur terminal en fonction des vitesses d'articulation et vice versa ;
3. Modèles dynamiques définissant les équations de mouvement du robot, qui établissent la relation entre les couples, les forces exercées par les actionneurs et les positions, vitesses et accélérations des articulations.

Les modèles géométriques sont les relations mathématiques qui expriment la situation de l'organe terminal en fonction des coordonnées articulaires du bras manipulateur et inversement. L'opérateur, à partir du site local, gère les commandes nécessaires au déroulement de la tâche et se prépare aux éventuelles modifications en cas de nécessité.

Le modèle géométrique direct permet d'exprimer la position de la pince du robot en fonction de ses coordonnées articulaires sous la forme vectorielle :

$$X = f(\theta) \quad (4.1)$$

Où $X = (X_m, Y_m, Z_m)^t$ sont les coordonnées opérationnelles du vecteur de l'effecteur terminal.

Et $\theta = (\theta_1^R, \theta_2^R, \theta_3^R)^t$ sont le vecteur représentant les coordonnées articulaires.

θ_1^R : L'angle de rotation ou de déplacement dans le plan XZ.

θ_2^R : L'angle de rotation ou de déplacement dans le plan XY.

θ_3^R : L'angle de rotation ou de déplacement dans le plan XY.

Comme le robot est de type RRR (trois rotations), on considère un référentiel R (X, Y, Z) situé à la base du robot. Nous associons Ri à chaque segment. Ainsi, les coordonnées opérationnelles des effecteurs terminaux sont données par ces équations :

$$\begin{bmatrix} X_m = (L_2^R \cos \theta_2^R + L_3^R \cos(\theta_2^R + \theta_3^R)) \cos \theta_1^R \\ Y_m = L_2^R \sin \theta_2^R + L_3^R \sin(\theta_2^R + \theta_3^R) + L_1^R \\ Z_m = (L_2^R \cos \theta_2^R + L_3^R \cos(\theta_2^R + \theta_3^R)) \sin \theta_1^R \end{bmatrix} \quad (4.2)$$

Où, L_1^R , L_2^R et L_3^R , sont les longueurs des segments du robot.

De même, puisque la caméra a deux degrés de liberté de type RR (figure 4.8), nous obtenons les relations entre les coordonnées opérationnelles et les coordonnées articulaires comme indiqué ci-dessous. L'indice ^R désigne le robot et l'indice ^C de la caméra.

$$\begin{bmatrix} X_c = (L_2^c \cos \theta_1^c) \cos \theta_2^c \\ Y_c = (L_2^c \sin \theta_1^c) \cos \theta_2^c \\ Z_c = L_1^c + L_2^c \sin \theta_2^c \end{bmatrix} \quad (4.3)$$

4.4.6.2 Modèle géométrique inverse (MGI)

Par le modèle géométrique inverse, nous calculons les coordonnées articulaires qui amènent la pince dans une situation désirée, spécifiée par ses coordonnées opérationnelles "xm, ym, zm" (expression 4.4).

$$\begin{bmatrix} \theta_1^R = \operatorname{artg} \frac{Z_m}{X_m} \\ \theta_2^R = \arccos \frac{\left[\frac{X_m}{\cos \theta_1^R} (L_2 + L_3 \cos \theta_3^R) \right] - [(Y_m - L_1)(L_3 \sin \theta_3^R)]}{L_2^2 + L_3^2 + 2L_2L_3 \cos \theta_3^R} \\ \theta_3^R = \arccos \frac{\left[\frac{X_m}{\cos \theta_1^R} \right]^2 + (Y_m - L_1)^2 - L_2^2 - L_3^2}{2L_2L_3} \end{bmatrix} \quad (4.4)$$

De même, les coordonnées géométriques inverses pour la caméra sont données par l'expression 4.5.

$$\begin{bmatrix} \theta_1^c = \operatorname{artg} \frac{Y_c}{X_c} \\ \theta_2^c = \arcsin \left(\frac{Z_c - L_1^c}{L_2^c} \right) \end{bmatrix} \quad (4.5)$$

Nous avons implémenté les algorithmes de calcul des coordonnées opérationnelles et articulaires du bras manipulateur conformément aux expressions présentées ci-dessus. Pour la technique MGD, nous utilisons un mode interactif dans lequel l'opérateur guide le robot au moyen d'une interface graphique vers l'endroit et les objets à manipuler. Pour la technique MGI, nous désignons les coordonnées cartésiennes du point à atteindre et le robot se dirige automatiquement vers ce point.

4.4.6.3 Simulation dans l'espace 2D

Le programme de simulation, basé sur les modèles géométriques direct et inverse, permet de pré- vérifier l'exécution des commandes (figure 4.9).

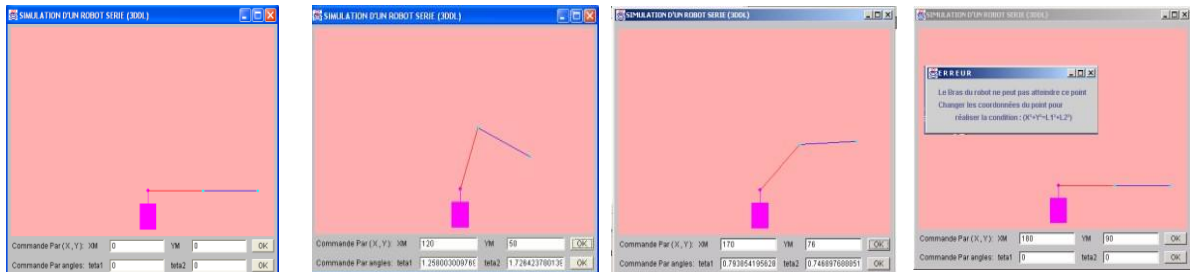


Figure 4.9. Simulation du robot manipulateur par les équations du modèle géométrique

Les commandes peuvent être données sous la forme de coordonnées opérationnelles (modèles géométriques inverse et direct). Cette simulation a été effectuée pour tester la validité de nos équations afin de les appliquer sur le modèle réel.

4.4.7 Modélisation cinématique

Les modèles cinématiques direct et inverse qui expriment la vitesse de l'organe terminal en fonction des vitesses articulaires et inversement. La cinématique complète la modélisation géométrique en établissant les relations entre les vitesses des paramètres articulation (q) et les vitesses des variables articulation. Ces relations s'expriment par :

$$\dot{X} = J \cdot \dot{q} \quad \text{Où } j : \text{ est une matrice suivant la configuration du SMA.}$$

La propriété évidente du modèle cinématique est sa linéarité par rapport aux vitesses, il est donc a priori plus aisée à manipuler que le modèle géométrique.

4.4.7.1 Modèle cinématique direct (MCD)

Le modèle cinématique directe d'un robot manipulateur décrit les vitesses des coordonnées opérationnelles (\dot{X}) en fonction des vitesses articulaires (\dot{q}).

$$\dot{X} = J \cdot \dot{q} \quad \text{Où } j \text{ désigne la matrice jacobéenne de dimension } (m \times n)$$

4.4.7.2 Modèle cinématique inverse (MCI)

Le modèle cinématique inverse d'un robot manipulateur exprime les vitesses articulaires (\dot{q}) en fonction des vitesses opérationnelles (\dot{X}).

$$\dot{q} = J^{-1} \cdot \dot{X} \quad (4.6)$$

4.4.8 Modélisation dynamique

Les modèles dynamiques définissant les équations du mouvement du robot, qui permettent d'établir les relations entre les couples ou forces exercés par les actionneurs et les positions, vitesses et accélérations des articulations.

4.4.8.1 Modèle dynamique direct (MDD)

Le modèle dynamique direct est représenté par la forme :

$$\Gamma = f(q, \dot{q}, \ddot{q}, fe) \quad (4.7)$$

Γ Vecteur des couples/forces des actionneurs, selon que l'articulation soit rotoïde ou prismatique.

q vecteur des positions articulaires

\dot{q} Vecteur des vitesses articulaires

\ddot{q} Vecteur des accélérations articulaires

fe : vecteur représentant l'effort extérieur (forces et moments) qu'exerce le robot sur l'environnement.

4.4.8.2 Modèle dynamique inverse (MDI)

Le modèle dynamique inverse exprime les accélérations en fonction des positions, vitesses et couples des actionneurs.

$$\ddot{q} = f(q, \dot{q}, \Gamma, fe) \quad (4.8)$$

Toutes ces fonctions sont basées sur la détermination des paramètres de Denavit-Hartenberg.

4.4.9 Expérimentation du robot

4.4.9.1 Introduction

Dans le domaine de la télé-opération, la grande majorité des tâches sont du type pick-and-place. L'opérateur, se basant sur des informations visuelles des images transmises par la caméra, puisse situer la pince par rapport aux objets à manipuler et de ce fait il positionne le

robot pour accomplir sa tâche. Plusieurs chemins se présentent à lui pour accomplir cette tâche ; dans ce cas il doit choisir soigneusement la configuration adéquate pour éviter tout obstacle.

4.4.9.2 Essais

Nous avons implémenté le modèle d'équations, précédemment défini, pour contrôler le robot. Les valeurs des segments du robot sont: $L_1 = 360$ mm, $L_2 = 340$ mm, $L_3 = 260$ mm.

A l'état initial du robot, les valeurs des θ sont mises à : $\theta_1 = 0^\circ$, $\theta_2 = 90^\circ$ et $\theta_3 = 270^\circ$ et par conséquent : $X_0 = 0$ mm, $Y_0 = 700$ mm et $Z_0 = 0$ mm.

Pour différentes valeurs de X_m , Y_m et Z_m , nous obtenons les différentes positions du robot (figure 4.10).

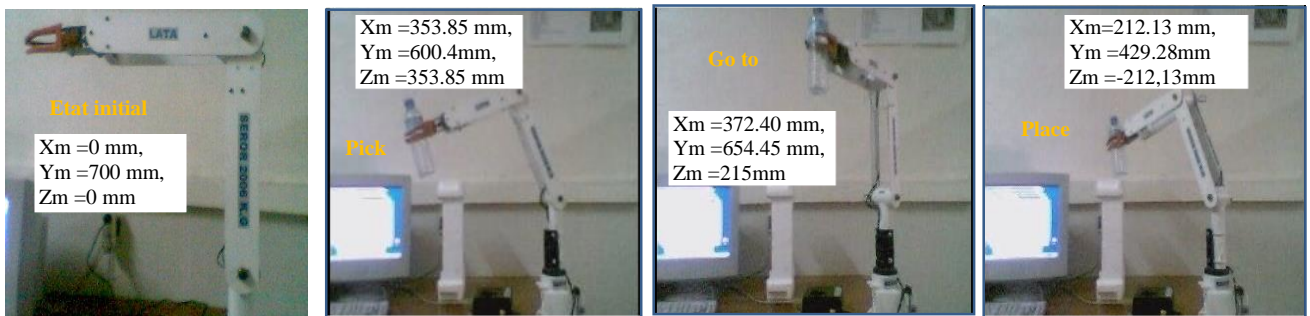


Figure 4.10. Différentes position du robot manipulateur & réalisation d'une tâche *pick-and-place*

Les expériences réalisées concernent deux situations différentes : avec une vision directe du site de travail par l'opérateur, ou sans vision directe en se basant sur l'interconnexion Internet.

4.4.9.2.1 Contrôle du robot en vision directe (opérateur humain présent sur le site de travail)

Dans le cas de la télé-opération à vision directe, l'environnement de l'opérateur et l'environnement opérationnel sont dans le même endroit. L'opérateur peut guider le robot et superviser directement l'exécution de la tâche demandée. Cette technique avec vision directe peut être avantageusement exploitée pour exécuter des tâches pick-and-place. Cependant, l'opérateur doit garder le contrôle sur le système pour superviser la tâche à l'aide d'une interface graphique et d'effectuer les corrections nécessaires afin de compenser les imperfections du modèle. Néanmoins, ce mode de contrôle peut devenir difficile pour conduire la tâche car il dépend de la position de la pince dans le champ de vision de l'opérateur ; A titre d'exemple, la non coïncidence entre l'axe du regard et la direction du mouvement. Pour cela l'opérateur

doit choisir un emplacement lui permettant d'atteindre les positions voulues avec une certaine précision; et cela n'est pas toujours facile à réaliser [87].

4.4.9.2.2 Contrôle du robot sans vision directe (Opérateur distant) : Commande par Internet

Dans ce mode de contrôle, l'opérateur est loin de l'environnement du robot. Il guide le robot distant à l'aide d'images captées par une caméra. Avec une seule caméra, cette technique génère des difficultés qui limitent les performances du système ; à savoir la qualité de l'éclairage de la caméra, la limitation du champ de vision, et en particulier, on note le temps de retard aléatoire qui lui est propre à Internet et la variabilité du transfert de l'information. Le retard dans la transmission d'images entre le robot et le site de contrôle.

Cette commande nécessite la présence d'une caméra sur l'environnement de travail pour permettre une vision panoramique de cet environnement, en plus de celle fixée sur le robot pour la visualisation de l'organe de préhension. Plusieurs modes de contrôle ont été testés avec le robot manipulateur, nous citons entre autres :

I. Contrôle basé sur un clic de souris

Avec ce mode, nous pouvons contrôler le robot et la caméra par de simples clics de souris sur les boutons appropriés d'un panneau. Chaque bouton représente une fonction spécifique ou une direction de mouvement spécifique (figure 4.3).

Dans ce mode de contrôle, l'opérateur peut effectuer des tâches de prélèvement, comme prendre une bouteille au-dessus d'une table et la placer à un autre endroit (figure 4.9). Pour réaliser des tâches avec ce mode, l'opérateur dirige le robot par une série de clics sur les boutons appropriés.

II. Contrôle basé sur l'image

Ce mode de contrôle permet un contrôle de haut niveau. Dans ce mode, l'opérateur dirige le robot vers des emplacements dans un espace 2D ou 3D en déplaçant uniquement ses images en utilisant des clics de souris.

A partir des images reçues, l'opérateur spécifie l'objet cible d'un simple clic, le système de vision considère ces coordonnées et le robot peut se déplacer vers l'objet désigné dans l'image, un second clic sur la webcam du panneau de contrôle le dirige là où on veut libérer l'objet, l'image reçue par la caméra, nous indiquons l'endroit montrant le positionnement de

l'objet. Cette technique est à la fois puissante et nécessaire pour lever de nombreuses difficultés et inconvénients associés au téléopérateur.

Certaines expériences ont été réalisées en utilisant le contrôle basé sur l'image. Pratiquement, ce mode de contrôle est utilisé pour envoyer le robot à un endroit quelconque. D'abord, une image du site distant est saisie. Ensuite, l'opérateur sélectionne un objet d'intérêt. Le logiciel de stéréovision extrait les coordonnées de cet objet qui sont utilisées pour déplacer le robot vers l'objet dans le monde réel.

III. Contrôle basé sur le gestuel

Dans ce mode de commande, le mouvement gestuel est présenté comme un moyen de contrôler un robot à partir d'une séquence d'images transmises par une caméra. Le système expérimental consiste en un ordinateur, une caméra webcam et un robot à trois degrés de liberté. L'algorithme développé pour la poursuite d'objets en mouvement utilise le traqueur KLT pour la détermination des points correspondants dans les séquences d'images, pour orienter le robot dans la direction correspondante.

4.5 Conclusion

Ce chapitre nous a permis de nous familiariser avec les systèmes robotiques en utilisant un bras manipulateur expérimental à 3 d.d.l (SEROR) pour effectuer certaines tâches préliminaires de saisie d'objet à distance. Les expériences réalisées avaient pour but d'évaluer l'interface utilisateur d'une part et d'autre part, l'architecture réseau du système en prenant en compte la distance séparant les deux sites maître et esclave.

Une valeur ajoutée importante est l'utilisation des modes de contrôle interactifs modernes (basé sur le clic de la souris, sur l'image et sur le geste). Les différents modes peuvent être intégrés dans une interface multimodale construite en fonction de la conception centrée sur l'utilisateur, ce qui permet de les utiliser individuellement ou dans une combinaison séquentielle. Cette capacité améliore la flexibilité du système par rapport à l'opérateur et permet d'effectuer diverses tâches complexes dans différents environnements.

Testés sur des sites locaux et dans des sites distants sur des centaines de kilomètres via Internet. Les résultats obtenus sont encourageants et confirment la possibilité d'utiliser ce système télérobotique avec une amélioration des moteurs et de la méthode de contrôle. Ce robot pourra alors effectuer des tâches industrielles, nucléaires, de soudage, d'intervention et d'exploration en environnement hostiles.

D'autre part, certains problèmes rencontrés au cours de nos expériences et pouvant affecter les performances du système, comme le retard de transmission, ont besoin d'être améliorés.

Pour pallier à ces problèmes et après une étude approfondie des méthodes utilisées, la plus appropriée dans la télé-opération des robots est l'utilisation de la réalité augmentée. Pour sa mise en œuvre, nous avons besoin d'un système robotique d'une certaine performance, d'un système de vision, d'outils informatiques, etc.

Nous présentons dans ce qui suit, la conception d'un robot manipulateur à 4 degrés de liberté, sa conception, sa modélisation et simulation puis sa réalisation. La réalisation a concerné la mécanique du robot et sa commande à base du microcontrôleur PIC18F452.

Chapitre 5

**Conception et Simulation d'un robot
manipulateur à quatre degrés de liberté
utilisant Autodesk/Inventor et la plateforme
SimMechanics de Matlab**

5.1 Introduction

Dans le processus de conception des systèmes mécaniques articulés, différentes étapes doivent être effectuées. Tout d'abord, le système doit être modélisé puis simulé et si les résultats de la phase finale de simulation sont satisfaisants, les algorithmes de contrôle développés sont testés sur un système réel.

En fait, les systèmes robotiques sont des systèmes à structure mécanique complexe, composés de plusieurs corps rigides reliés entre eux par des liaisons appelées articulations. Leur simulation nécessite des outils développés qui peuvent modéliser avec suffisamment de précision le monde physique permettant l'interaction de l'utilisateur.

Le principal objectif de cette étude ou recherche répond à deux besoins recherchés qui sont :

- ❖ Le prototypage rapide de robots manipulateurs à structure quelconque en vue de leur analyse (cinématique, comportement, etc.),
- ❖ Une interface intuitive pour la télé-opération (définition de tâche) de nouveaux robots (environnement de télé-opération, etc.).

Ces principes sont atteints par la création d'une interface de communication homme-robot interactive réalisée grâce à un nouveau fichier de description des structures de robots, à un moteur de résolution de contraintes fonctionnant en temps réel ainsi qu'à des outils d'entrée et sortie flexible pour l'utilisateur.

Lors de la conception de la commande d'un robot, nous devons connaître le couple et l'angle de rotation nécessaires de chaque moteur, visualiser le comportement du robot et obtenir un modèle mathématique de chaque pièce. Généralement, cette tâche cinématique inverse ne peut pas être résolue analytiquement et le calcul numérique comporte souvent des difficultés. La modélisation physique dans l'environnement SimMechanics facilite considérablement les efforts de simulation de systèmes mécaniques complexes, quelle que soit leur complication par des éléments élastiques et amortisseurs et par le nombre de degrés de libertés [88]. Le schéma du programme SimMechanics se présentant sous la forme de blocs interconnectés montre comment les composants physiques avec des relations géométriques et cinématiques du robot sont interconnectés [89,90].

Plusieurs approches sont à prendre en considération dans le développement d'un environnement de modélisation et de simulation : la première concerne la conception du robot, plus précisément, comment créer des pièces individuelles et les assembler ensemble. Ensuite

nous décrivons comment traduire le modèle CAO en SimMechanics et solder par l'édition du modèle et la connexion des blocs Simulink aux blocs SimMechanics.

5.2 Développement d'un environnement de modélisation et de simulation

Un robot manipulateur est une séquence de corps rigides (appelées liens) qui sont reliés entre eux par des articulations, la principale distinction entre les différents robots concerne leur structure mécanique, ou de la façon dont les liaisons sont reliées et la façon dont ils se déplacent par rapport à l'autre. Ils sont constitués de sous-ensembles de parties mécaniques et moteurs qui interagissent entre eux. Par conséquent, leur simulation avec les méthodes classiques telles que les méthodes géométriques, algébriques, itératives, réseau de neurones...etc, devient difficile ou parfois impossible, avec l'augmentation des degrés de liberté. Il existe des logiciels qui offrent ce service comme SimMechanics de Matlab qui est un environnement de modélisation physique dédiés entre autres à la robotique [91-93]. L'utilisation de ces logiciels facilitent la tâche de simulation et permettent de pallier aux difficultés rencontrées avec les systèmes de modélisation cinématiques courantes [94-99].

Nous avons conçu un robot comprenant quatre degrés de liberté, piloté par trois moteurs à courants continu au niveau de ses trois articulations et deux servomoteurs pour actionner la pince. Le logiciel Autodesk/Inventor, a été utilisé pour la conception de la partie mécanique du robot car il contient des outils adaptés à la conception mécanique 3D, à la documentation et à la simulation de produit.

La figure ci-dessous illustre le schéma général du robot, où sont montrés les différents segments constituant le robot (figure 5.1).

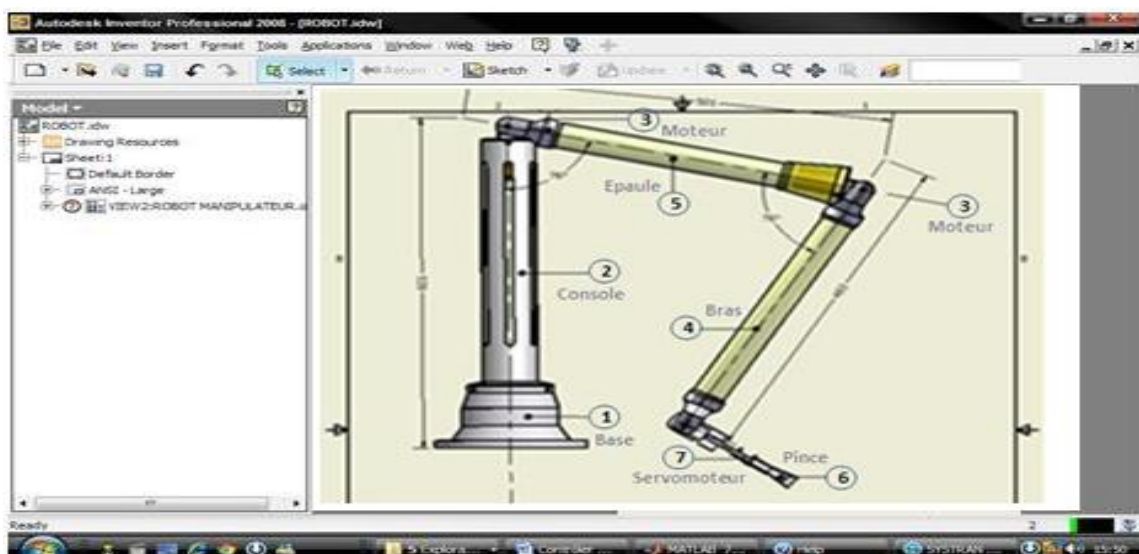


Figure 5.1. Structure mécanique du robot manipulateur conçu sur Autodesk/Inventor

5.3 Présentation des outils de modélisation et de simulation

Parallèlement au développement des robots, de nombreux outils pour les simuler en vue de leur étude ont vu le jour. Certains logiciels relèvent de la recherche, mais de nombreux outils dans ce domaine sont proposés par le marché. Les logiciels de conception mécanique tels que 'Autodesk/Inventor' possèdent des outils très puissants et performants.

Ces logiciels comportent tous les mêmes fonctionnalités de base : principalement un moteur de résolution qui formule les équations automatiquement et fournit les solutions pour la statique et pour la modélisation du comportement dynamique des systèmes à corps rigides multiples, formant plusieurs boucles cinématiques. En effet, l'analyse dynamique prend une place croissante dans la conception des systèmes de plus en plus complexes afin de trouver les comportements optimums et une fiabilité accrue. Les moteurs de résolution de ces outils sont principalement basés sur des méthodes numériques. Des modules de visualisation contenus dans ces outils permettent d'effectuer une simulation en trois dimensions du comportement calculé. Ils contiennent des programmes de dessin tridimensionnel assisté par ordinateur, ou alors sont interfacés avec de tels programmes.

Les logiciels dédiés à la simulation des robots permettent aux concepteurs et utilisateurs de robots de mettre en œuvre virtuellement des robots dans leur environnement de travail en vue de la validation d'un nouveau système et/ou de l'optimisation d'un système existant. Ces programmes sont essentiellement utilisés pour la programmation hors-ligne des robots. Cette fonctionnalité permet la programmation complète d'une installation robotisée sans immobiliser les ressources. La plupart de ces programmes sont fournis avec une bibliothèque de robots existants permettant la résolution de leur cinématique (voire dynamique). Ceci permet une simulation performante et exacte des mouvements des robots [101-104].

L'intégration de ces outils de simulation avec les technologies de conception assistée par ordinateur amène au concept de "Virtual Prototyping" : la conception de prototypage virtuel qui nous permet :

- ❖ D'éviter de construire des prototypes dans la première phase de développement ;
- ❖ De réduire le temps de conception d'un produit et d'améliorer sa qualité ;
- ❖ Optimiser et mieux comprendre le fonctionnement de systèmes complexes tels que les systèmes robotiques.

5.3.1 Le logiciel Autodesk/Inventor

Autodesk Inventor est le logiciel fondateur de la solution Autodesk pour le prototypage numérique "Digital Prototyping". C'est un logiciel CAO comprenant des programmes de dessin tridimensionnel qui permet la conception et la modélisation des systèmes mécaniques (figure 5.1).

Il fournit des outils de conception mécanique 3D de qualité professionnelle, de documentation et de simulation de produits.

Le modèle de logiciel Inventor est un modèle numérique 3D précis qui permet de valider la forme, l'ajustement et la fonction d'une conception, en minimisant la nécessité de tester via la conception de prototypes physiques.

Les produits Autodesk Inventor incluent un environnement de conception paramétrique intuitif pour développer des schémas de conception et des modèles cinématiques de pièces et assemblages.

5.3.2 Le logiciel Matlab

Le logiciel *Matlab* possède de nombreux outils permettant de mettre en œuvre une démarche de modélisation globale multi-physique. Il est un puissant outil de calcul, il permet de saisir des instructions sous la forme de lignes de commandes. MATLAB possède des bibliothèques de fonctions regroupées dans des '*Toolbox*' relatives à un domaine de connaissance, '*Control System Toolbox*' pour le contrôle commande des systèmes, '*Signal Process Toolbox*' pour le traitement du signal,...etc.

5.3.3 Simulink de Matlab

Simulink représente une interface graphique de Matlab, qui permet de s'affranchir du code et de la syntaxe indispensable à la saisie des lignes de commandes. Il possède des bibliothèques de blocs, regroupés dans des Blocksets (*Simulink Control Design* pour le contrôle commande des systèmes par exemple). Le comportement dynamique d'un système est caractérisé par un bloc contenant, par exemple, la fonction de transfert du système avec une entrée et une sortie. L'information circulant dans les connexions entre deux blocs est un signal numérique orienté. La sortie du bloc est calculée numériquement en déterminant pour chaque pas de calcul la transformation du signal d'entrée imposé par le contenu du bloc.

Simulink propose également toute une palette d'outils très évolués pour le contrôle commande des systèmes asservis. Le comportement physique des composants est pris en

compte directement par le logiciel, il est donc possible de modéliser un système sans avoir à écrire l'équation différentielle qui caractérise son comportement.

5.3.4 Simscape de Matlab

Simscape propose une approche qui permet de faire de la modélisation par assemblage de composants. Le comportement physique des composants est pris en compte directement par le logiciel, il est donc possible de modéliser un système sans avoir à écrire l'équation différentielle qui caractérise son comportement.

Simscape inclue une bibliothèque de composants élémentaires dans toutes les technologies avec leurs différents modules : mécanique (*SimMechanics*), électrique (*SimPowerSystems*), hydraulique (*SimHydraulics*), pneumatique (*SimDrivelines*), thermique, magnétique...etc.

Chacun de ces modules explore un domaine technique spécifique et propose une modélisation de haut niveau pour les bibliothèques de base de *Simscape*.

Donc, les principaux supports de modélisation et de la simulation des systèmes robotiques sont, *Simulink*, et *SimMechanics* de *Simscape*.

5.3.5 Blocs SimMechanics employés

5.3.5.1 Description du module SimMechanics

Le module *SimMechanics* est un outil interactif de modélisation en trois dimensions, des systèmes mécaniques dans l'environnement *Simulink*.

SimMechanics contient un ensemble de bibliothèques de blocs et d'interfaces de simulation spéciales (blocs de capteurs et d'actionneurs) permettant l'interconnexion du schéma *SimMechanics* avec l'environnement *Simulink*. Les blocs *SimMechanics* présentent des éléments permettant de modéliser des systèmes mécaniques constitués de corps rigides reliés par des articulations représentant des degrés de liberté en translation et en rotation. *SimMechanics* configure automatiquement un seul référentiel inertiel absolu et un système de coordonnées (CS) appelé *World*.

Pour une interprétation plus facile des schémas fonctionnels, plusieurs blocs *SimMechanics* sont utilisés présentés par leurs fonctions.

5.3.5.2 Description des fonctions des blocs utilisés dans le programme SimMechanics

1. **Env** : Le bloc *environnement machine* définit l'environnement pour le calcul du schéma. Chaque modèle SimMechanics contient un bloc de ce type connecté au bloc *Ground*. Hormis la saisie de la précision du calcul et des paramètres de l'environnement, le type d'analyse requis peut être défini :
 - ❖ **Dynamique avant** : sur la base des valeurs et des forces initiales dans le système, le programme calcule les valeurs des positions et des vitesses.
 - ❖ **Linéarisation** : ce mode calcule le modèle linéaire du système.
 - ❖ **Découpage** : trouve l'état stable de la machine.
 - ❖ **Inverse Dynamics pour boucle ouverte** : Dans ce mode, SimMechanics calcule les forces nécessaires à l'exécution du mouvement forcé par excitation cinématique.
 - ❖ **Kinematics** : fait de même pour les systèmes en boucle fermée en incluant des contraintes invisibles internes supplémentaires découlant de ces structures.
2. **Ground** : Le bloc Ground représente un point fixe ayant une masse infinie. Au moins un bloc de terre connecté à l'environnement machine doit être impliqué.
3. **Body (Corps)** : Le bloc Body de SimMechanics remplace tous les corps rigides fixes auxquels sont ajoutés les degrés de liberté. Les corps sont définis par leurs masses finales et non nulles, leur inertie, leurs positions, leurs directions et par des systèmes de coordonnées qui leur sont connectés.
4. **Revolute** : Le bloc *Revolute* du groupe *Articulations* représente un degré de liberté (rotation).
5. **Actionneur** : Les capteurs et les actionneurs sont les blocs utilisés comme interfaces entre les blocs Simulink non SimMechanics et les blocs SimMechanics. Grâce aux *actionneurs*, il est possible de transformer un signal Simulink en signal physique activant les corps dans le diagramme SimMechanics. Les *capteurs* intègrent des fonctions inverses en transformant le signal de SimMechanics en environnement Simulink. Les sorties des capteurs sont : positions, vitesses, accélérations, forces de réaction, etc.

5.4 Modélisation du robot

Les modèles de robot ont été développés sur la base de la configuration du robot. Pour simplifier le développement, nous avons utilisé une fonctionnalité spéciale offerte par MathWorks et qui intègre SimMechanics à certains logiciels de conception mécanique, tel qu'Autodesk/Inventor (utilisé dans la conception de notre système robotique). La collaboration de Mathworks avec Inventor a étendu les capacités d'analyse technique de SimMechanics en permettant une intégration transparente des assemblages CAD dans l'environnement de conception et de simulation SimMechanics.

Considéré comme un assemblage de pièces mécaniques, le robot est constitué de sous-ensembles de parties mécaniques et moteurs qui interagissent entre eux. Pour le contrôler, le programme SimMechanics permet de modéliser sa structure par corps et par articulation, de simuler leur mouvement, de modifier facilement l'assemblage, d'optimiser les paramètres du système et d'analyser les résultats, le tout dans l'environnement Simulink. Cette approche ne nécessite pas de lourdes dérivations des équations différentielles du système et constitue un moyen simple et rapide pour l'analyse des forces et des couples dans les joints mécaniques, le tracé des accélérations et les déplacements de chaque partie du système, pour visualiser le mouvement de l'assemblage CAO. Cette commodité est activée par l'installation d'un plug-in adapté dans SimMechanics qui importe le modèle CAO 3D du système complet avec les corps, les joints, les couplages et les masses du programme Inventor dans SimMechanics pour une utilisation ultérieure du modèle.

L'objectif du chapitre proposé est de présenter une application de manière efficace de la modélisation de la mécanique de robot et de sa simulation dynamique à l'aide de modules additionnels add-on de l'environnement avancé de MATLAB: SimMechanics et Interface graphique utilisateur (GUI) MATLAB.

Matlab est utilisé comme plateforme pour effectuer la simulation du système robotique réalisé, parties mécaniques et commande. Le modèle complet du robot conçu sur Autodesk/Inventor peut être exporté vers Matlab en exploitant les fonctionnalités qui lui sont nécessaires. SimMechanics de Matlab permet une modélisation multi-physique des modèles CAO 3D importés à partir du logiciel Inventor.

Les propriétés cinétiques des pièces sont prises en compte et le comportement dynamique est automatiquement intégré. Il est donc nécessaire de concevoir un environnement de simulation physique pour notre système robotique.

5.4.1 Exportation du modèle robotique de Inventor vers Matlab

SimMechanics est un système de modélisation mécanique 3-D avec possibilité d'importer des modèles conçus sur des logiciels mécaniques entre autres Autodesk/Inventor. Associé à Simulink, il fournit un environnement de simulation multi-corps pour les systèmes mécaniques 3D, permet de modéliser le système multi-corps à l'aide de blocs représentant des corps, des articulations, des contraintes et des éléments de force, puis SimMechanics formule et résout les équations de mouvement pour le système mécanique complet.

Les modèles des systèmes de CAO, y compris la masse, l'inertie, l'articulation, la contrainte et la géométrie 3D, peuvent être importés dans SimMechanics. Une animation 3D générée automatiquement permet à l'utilisateur de visualiser la dynamique du système.

5.4.2 Etapes suivies dans l'exportation du modèle de robot

La première étape contient des instructions sur la façon de concevoir un bras robotique. Plus précisément, comment créer des pièces individuelles et les assembler ensemble. Ce qui inclut des informations sur le modèle de conception CAD 3D dans Autodesk Inventor du robot. L'étape suivante concerne l'exportation du modèle CAO en SimMechanics de Matlab (génération du .xml file). Cette procédure a nécessité l'installation d'un plug-in (Add-Ins) approprié dans SimMechanics (figure 5.2), qui permet d'exporter l'assemblage du robot conçu sur Inventor vers Matlab.

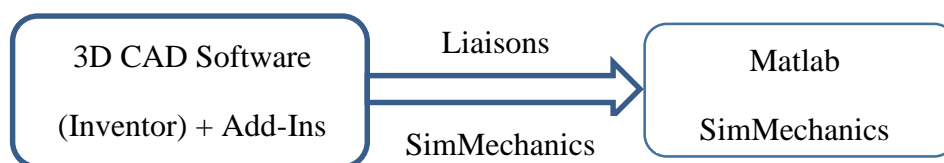


Figure 5.2. Procédure d'exportation du modèle CAD vers SimMechanics

La dernière étape concerne l'installation du système robotique sur la plateforme Matlab/SimMechanics, pour pouvoir être modélisé puis simulé.

5.4.2.1 Principe de la méthode d'exportation

Add-Ins/Export SimMechanics étant la première et la principale étape de ce transfert (figure 5.3). C'est à partir de cet utilitaire que l'on peut procéder à l'exportation vers SimMechanics dans les formats première ou seconde génération. Ce qui conduit à la création d'une liste de fichiers *STL* pour chaque bloc constituant le robot et les fichiers de structure .xml qui définissent les contraintes entre ces blocs.

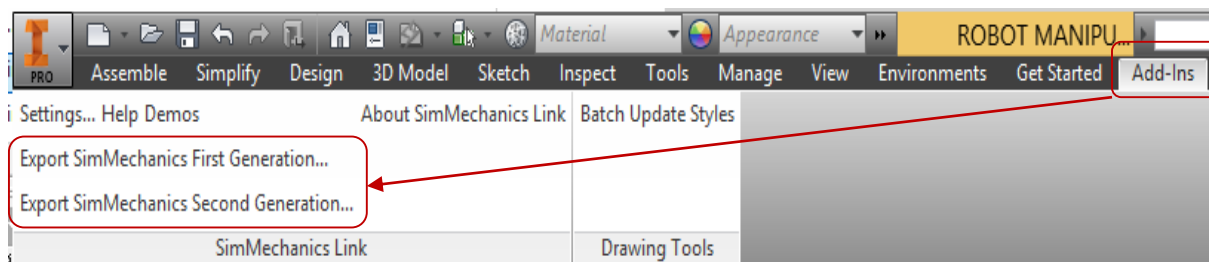


Figure 5.3. Méthode d'exportation SimMechanics première et seconde génération

La procédure d'exportation CAO génère un fichier XML d'importation SimMechanics et un ensemble de fichiers STL. Le fichier XML contient la structure de l'assemblage et les paramètres qui définissent chaque pièce. Les fichiers STL définissent la géométrie 3D de chaque pièce. Une fois la procédure d'exportation terminée, le fichier XML est importé dans SimMechanics. SimMechanics utilise le fichier pour générer automatiquement un nouveau modèle SimMechanics.

5.4.2.2 Exporter le robot manipulateur dans les formats de 1^{ère} génération

L'exportation au format de première génération génère un fichier XML de modélisation physique. La commande appropriée pour ce format utilise l'instruction *mech_import* (figure 5.4).

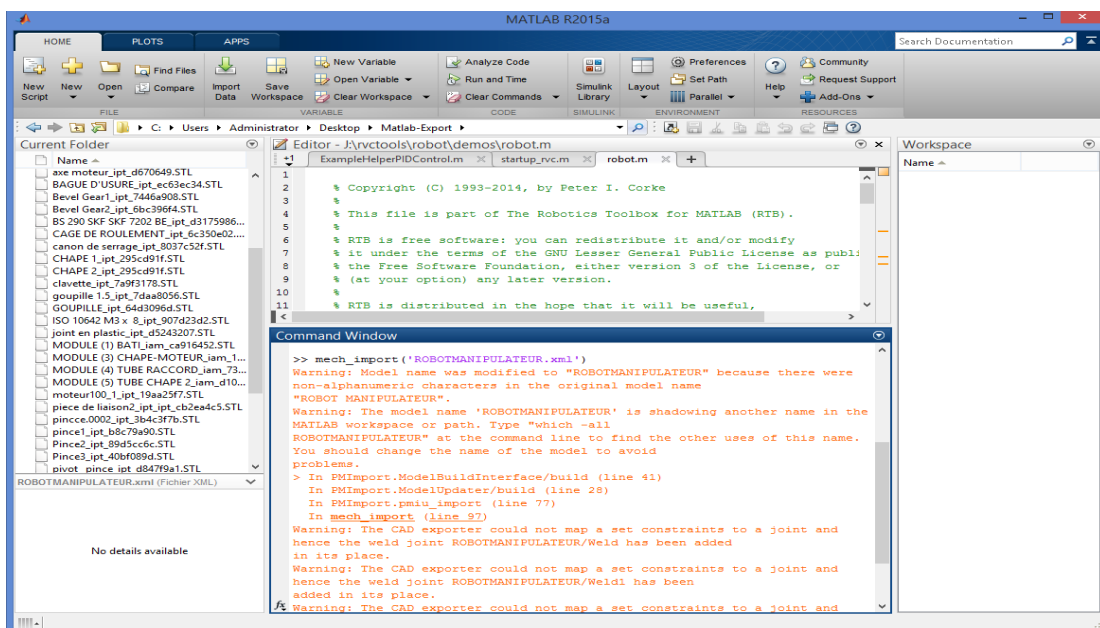


Figure 5.4. Importation du modèle SimMechanics première génération

Le résultat donne naissance à un fichier *.slx* permettant de présenter le modèle complet du robot manipulateur sous forme de blocs constituant chacun de ses éléments (figure 5.5).

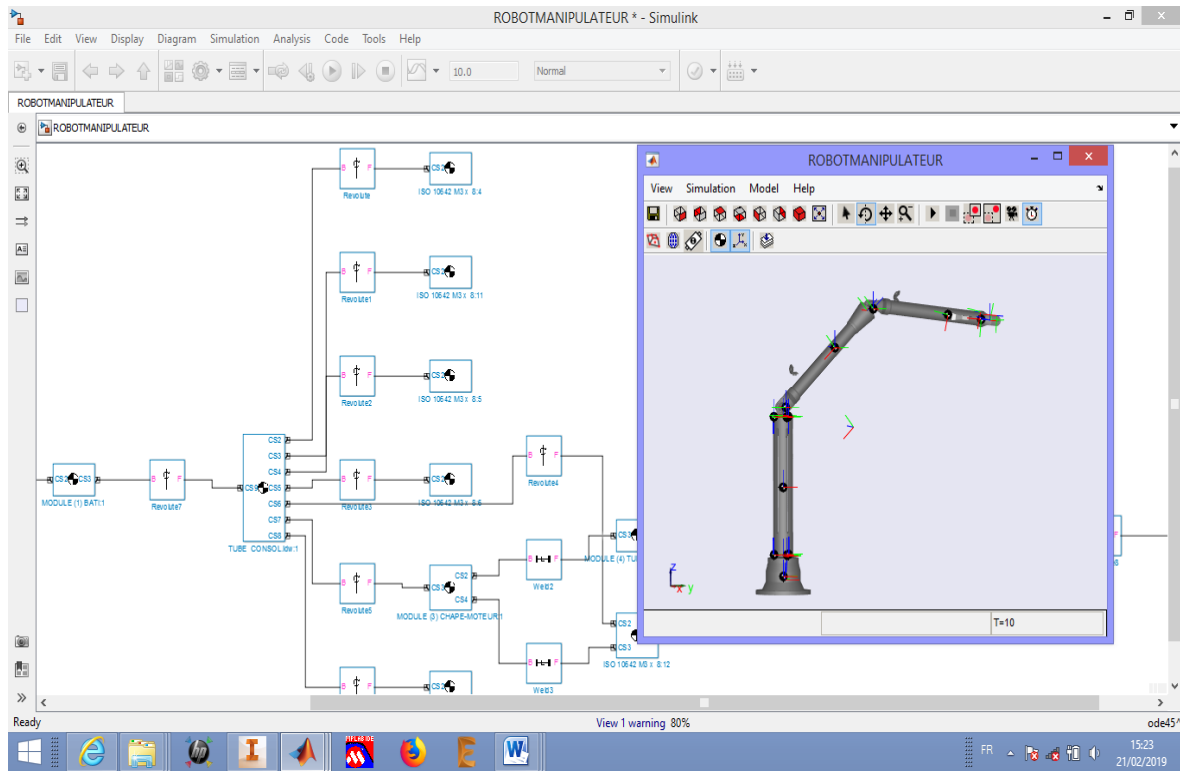


Figure 5.5. Représentation du robot manipulateur et des blocs le constituant dans SimMechanics première génération

5.4.2.3 Exporter dans les formats de seconde génération

L'exportation au format de deuxième génération engendre un fichier XML d'importation SimMechanics.

En utilisant les liaisons SimMechanics, on peut automatiquement créer l'assemblage en 3D du robot dans SimMechanics à partir de l'assemblage CAD conçu sur Inventor. Le modèle du robot importé de CAD (Inventor) est sauvegardé dans un fichier *.xml* et exporté vers SimMechanics. La visualisation est générée à partir des fichiers STL exportés automatiquement du modèle CAD. Pour ce faire, nous avons procédé par étape en suivant les schémas des figures 5.6 et 5.7.

Chapitre 5 Conception et Simulation d'un robot manipulateur à quatre degrés de liberté (4 d.d.l) utilisant Autodesk/Inventor et la plateforme SimMechanics de Matlab

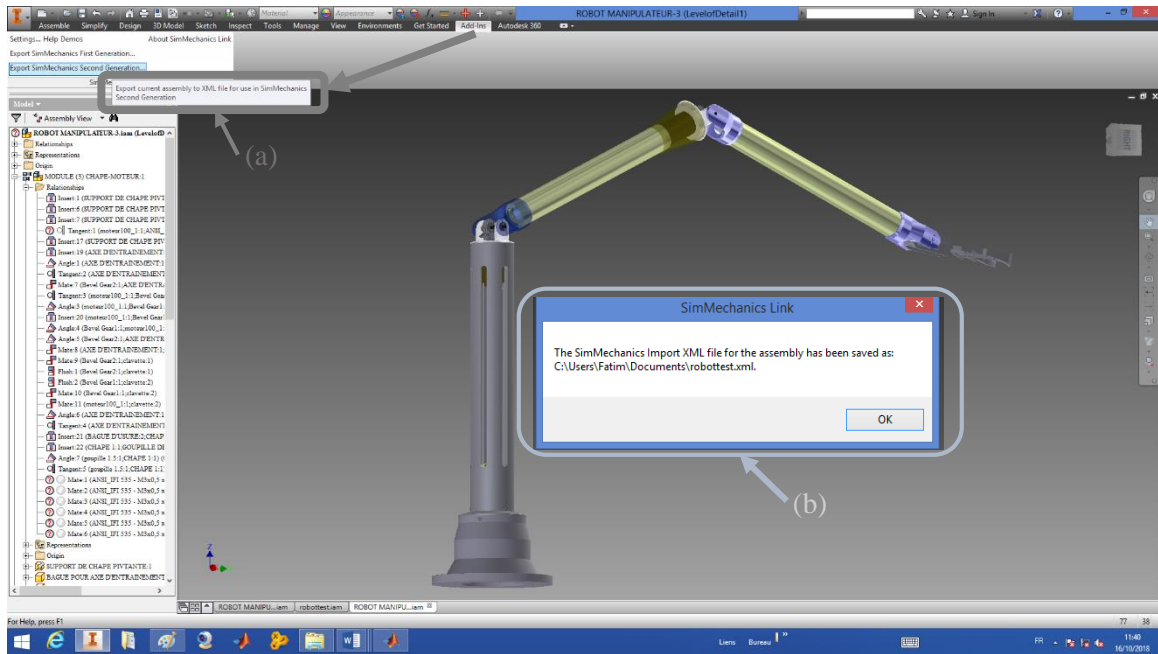


Figure 5.6. Exportation de l'assemblage du Robot conçu vers XML fichier pour utilisation dans SimMechanics seconde génération (création du lien "Add-ins") (a). Sauvegarde du fichier XML d'importation SimMechanics d'assemblage (b).

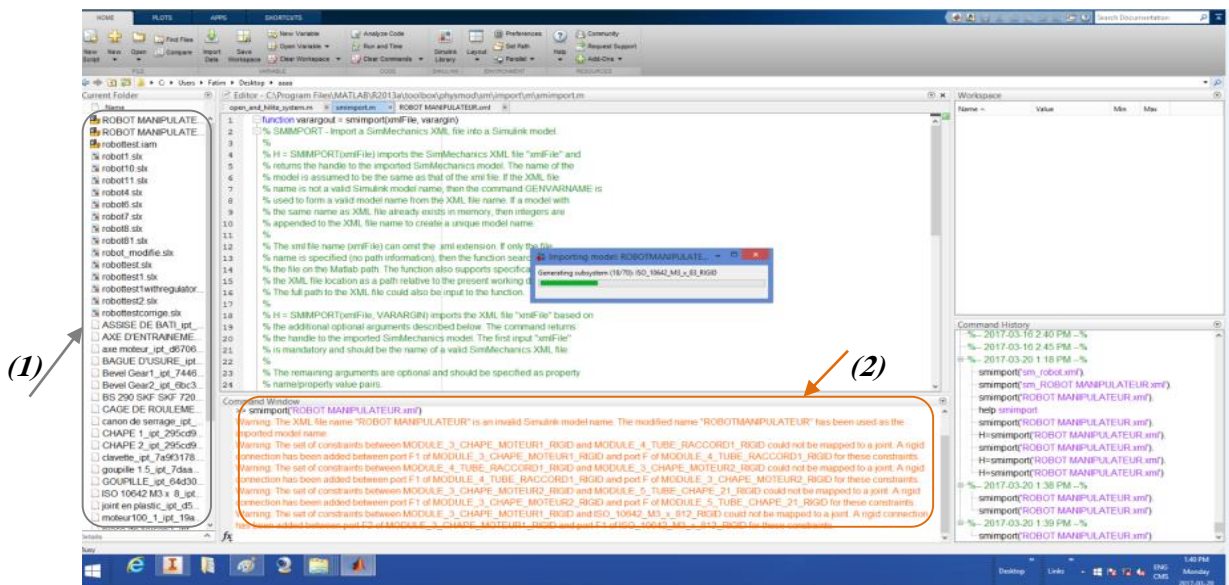


Figure 5.7. Génération des fichiers STL de chaque élément constituant le robot (1). Exécution du fichier ".xml" et création du fichier ".Slx" utilisant Smimport (2).

Ces deux étapes concernent les deux modes d'exportation en SimMechanics première et seconde génération.

5.4.2.4 Blocs interconnectés constituant le robot manipulateur

Lors de l'exécution du fichier .slx, nous créons le modèle complet du robot dans l'environnement Matlab/SimMechanics (figure 5.8). Ce modèle est prêt à l'exécution et permet de visualiser le comportement du robot manipulateur en 3D.

D'autre part, le modèle généré peut être modélisé selon l'application à envisager en jouant sur certains paramètres comme le couple, qui permet d'afficher en ligne, l'évolution de la position et/ou la vitesse correspondant.

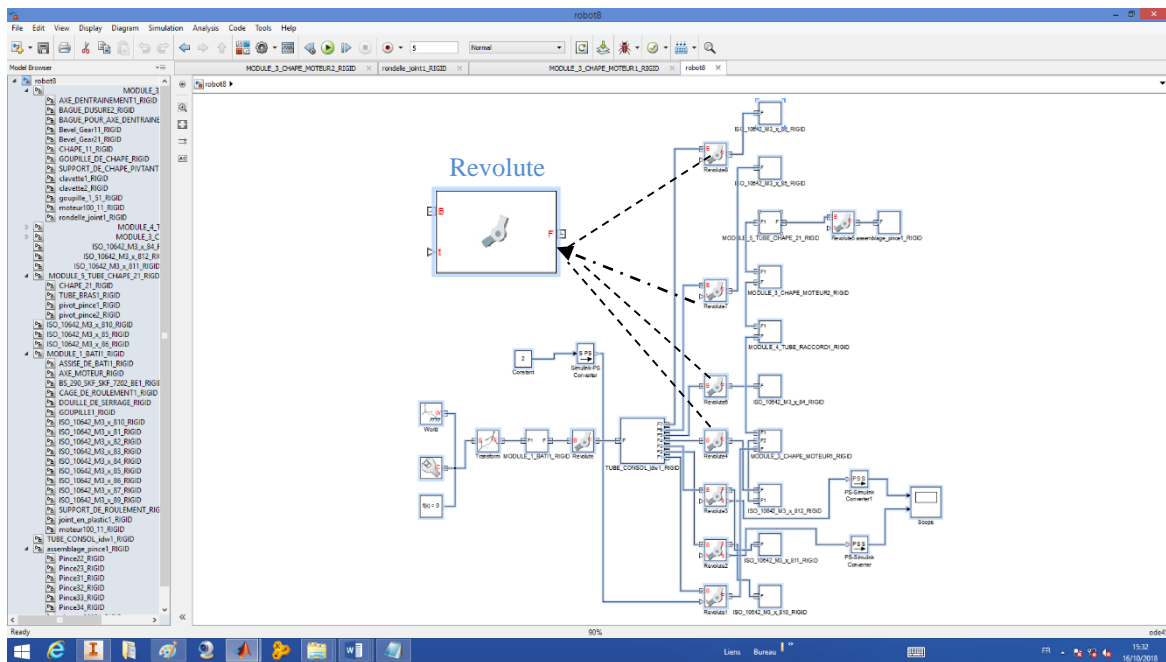


Figure 5.8. Modèle complet du robot dans l'environnement Matlab/SimMechanics

Le modèle du robot présenté par le schéma de la figure ci-dessus, peut être simulé en agissant sur les paramètres des joints "Revolute" (un bloc Revolute représente un degré de liberté de rotation unique autour d'un axe spécifié entre deux corps).

Chaque bloc est identifié séparément avec le même nom du fichier source et chaque pièce peut être affichée et configurée séparément (masse, inertie, dimension, coordonnées, ...etc.) afin de procéder à une éventuelle modification ou un ajustement (figure 5.9).

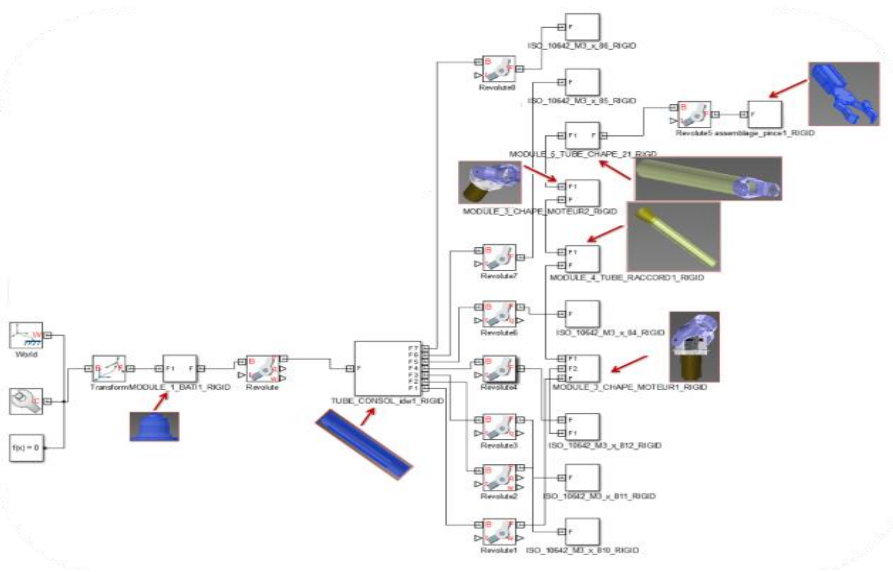


Figure 5.9. Modèle CAD des assemblages du robot dans SimMechanics

5.4.2.5 Système de joint 'Revolute'

Le système de joint présenté sur la figure 5.10 est un élément essentiel de modélisation. C'est à partir de ce système que l'on puisse agir sur le couple moteur pour voir l'évolution de la position et de la vitesse.

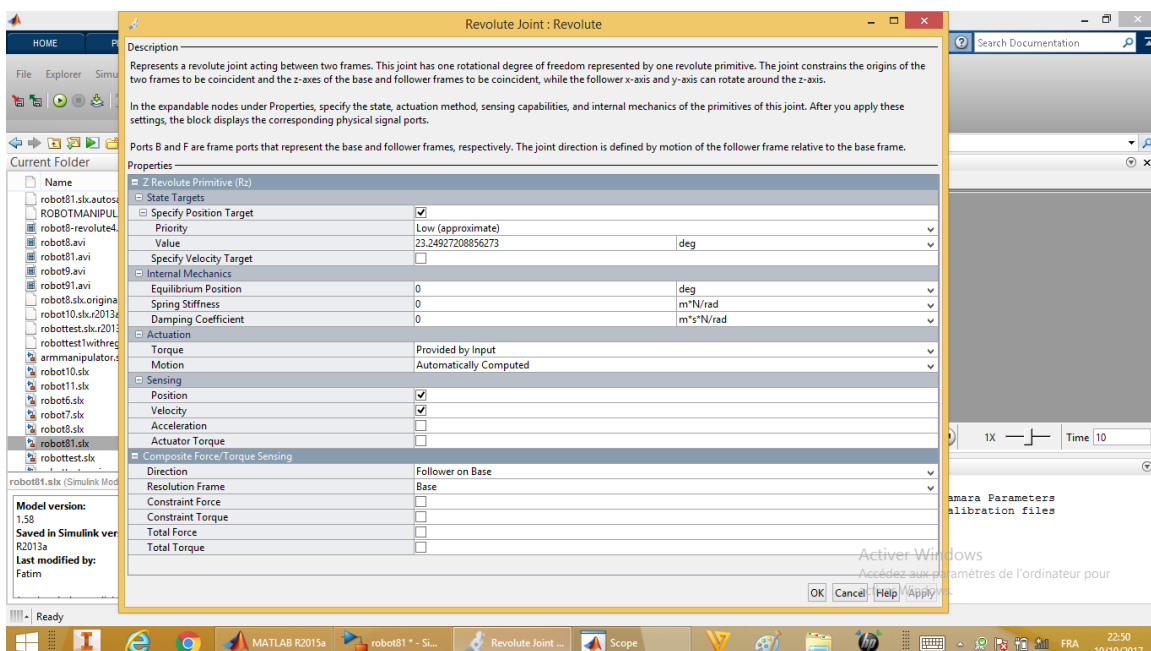


Figure 5.10. Compartiment 'Joint Revolute'

Ce système représente une articulation rotative agissant entre deux frames. Cette articulation a un degré de liberté rotationnel.

5.4.2.6 Représentation des différents blocs en SimMechanics

Chaque bloc du robot manipulateur a été étudié séparément et peut être affiché avec ses caractéristique, en exemple le schéma de représentation du moteur et du Shape moteur (figure 5.11).

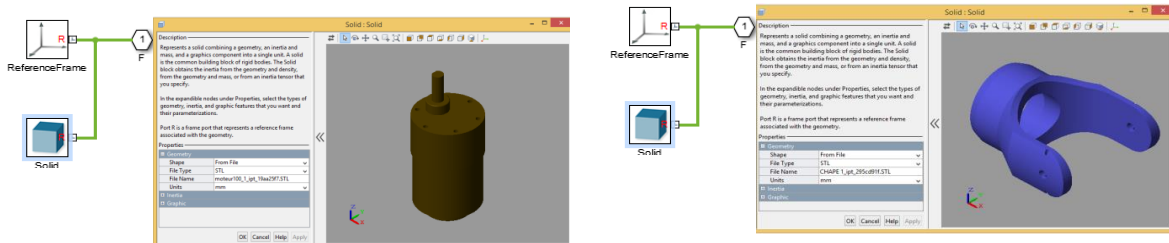


Figure 5.11. Représentation des différentes parties mécaniques du robot Moteur (à gauche), Shape moteur (à droite)

5.4.3 Simulation du robot en environnement SimMechanics

Afin de pouvoir simuler le robot manipulateur, nous avons besoin d'effectuer une translation du modèle CAD réalisé sur Inventor vers Matlab/SimMechanics. Ceci concerne tous les blocs constituant le robot manipulateur : moteurs, joints, engrenages, coudes, épaules, base,...etc. D'autres éléments peuvent être ajoutés au modèle comme les blocs de régulation, en se servant des modèles existant dans Simulink. Des moyens d'affichage sont aussi disponibles sur Simulink "Scopes". Ainsi, les résultats de simulation sont affichés en ligne. Les étapes de modélisation sont présentées en figure 5.12.

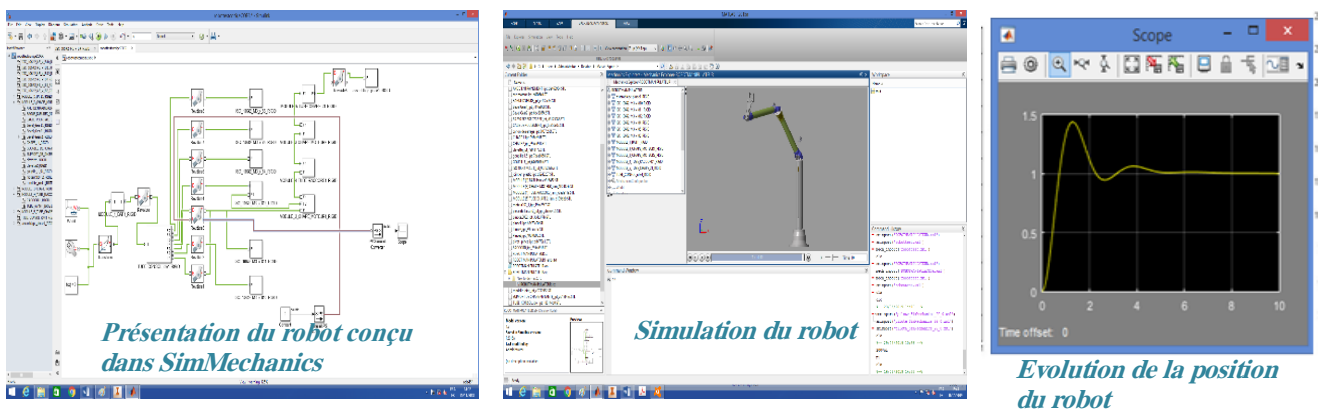


Figure 5.12. Modélisation et simulation du robot en environnement SimMechanics de Matlab

La simulation peut être opérée sur toutes les parties du robot, la figure 5.12 schématise le mouvement du robot et trace en ligne l'évolution de la position du robot lors du mouvement du moteur de la base.

5.4.4 Modélisation de la pince

Le bloc 'Revolute 5' montre l'assemblage de la pince désigné par l'ensemble de blocs de la figure 5.13. Le tube_bras, Shape_moteur et les pivots représentent les éléments du montage de la pince.

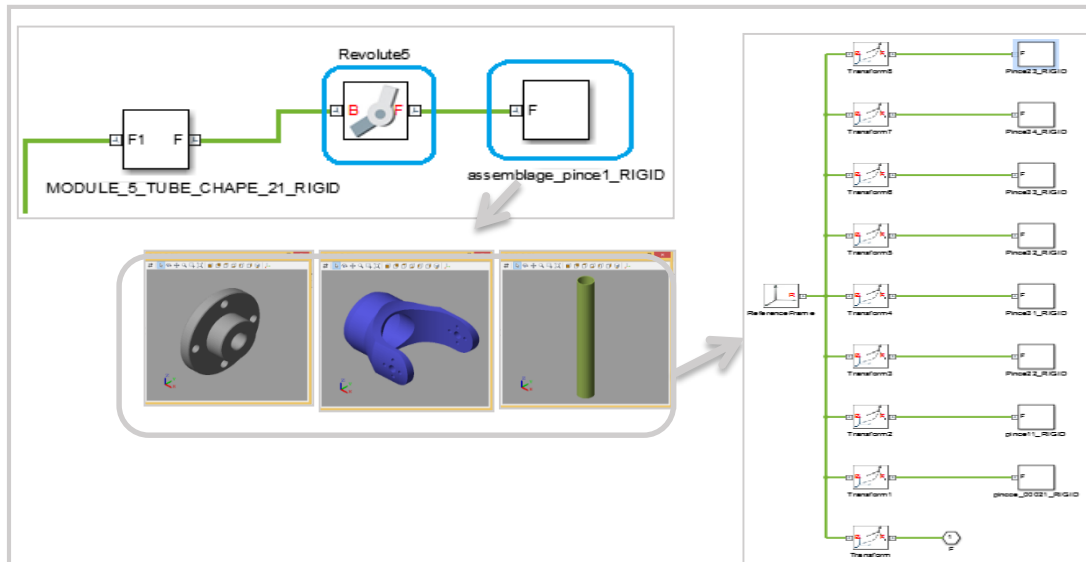


Figure 5.13. Représentation des blocs de l'Assemblage de la pince

En exerçant un couple sur le bloc revolute 5 comme le montre la figure 5.14, on observe sur oscilloscope 2, l'évolution de la position (figure 5.15) et en oscilloscope 3, celle de la vitesse (figure 5.16).

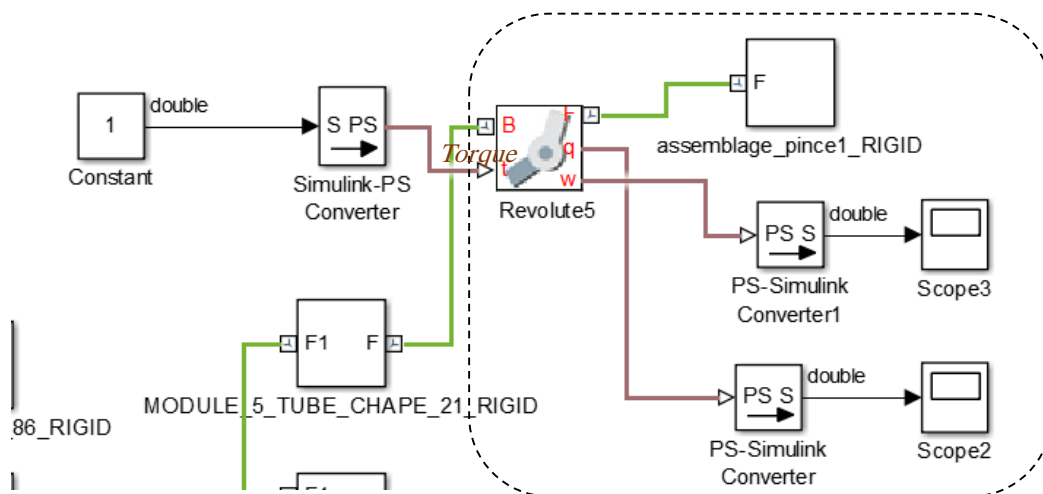


Figure 5.14. Simulation de la pince représentée par le bloc Revolute5

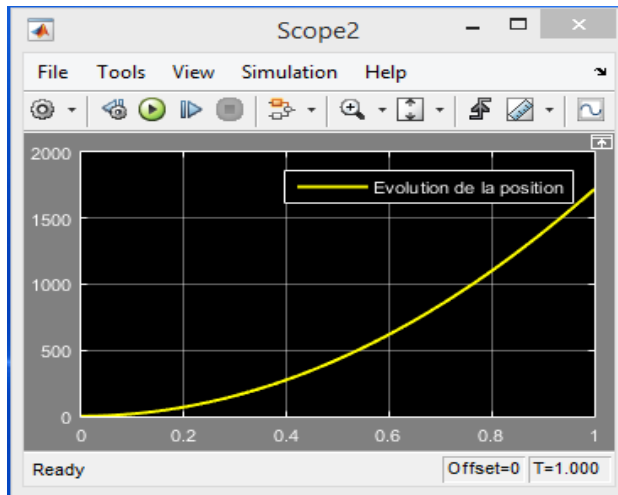


Figure 5.15. Evolution de la position Revolute5

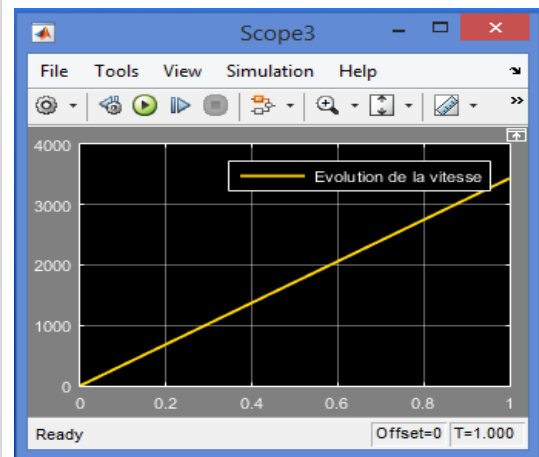


Figure 5.16. Evolution de la vitesse

On constate que la vitesse est constante, ce qui est tout à fait cohérent vue que la fonction de préhension réalisée par la pince fait un mouvement d'ouverture ayant une vitesse constante.

5.5 Conclusion

La simulation de mécanismes, les méthodes de télé-opération de robots et les interfaces homme-machine sont devenus des domaines clef dans la recherche et l'industrie. Ils permettent la maîtrise de la conception et du contrôle de dispositifs complexes.

La modélisation du robot réalisé permet de détecter et de corriger les défauts enregistrés dans sa conception (s'il y a lieu). Ce qui facilite la conception d'un prototype conférant à l'ensemble des performances des systèmes mécaniques.

La simulation étant un outil pour l'analyse d'un système de robot et la planification des tâches, puis une plateforme ouverte pour développer de nouveaux systèmes de robots. De ce fait, la modification du design du robot manipulateur nécessite l'introduction des nouveaux éléments tels que les moteurs, les joints des articulations adaptées à la mobilité du robot, les contraintes, les forces, etc.

A l'issue de cette étude, la conception de l'environnement de simulation physique, a nécessité l'utilisation du logiciel Autodesk/Inventor (CAD mécanique) et SimMechanics, qui est un outil interactif de modélisation en trois dimensions, des systèmes mécaniques dans l'environnement Simulink. Les essais pratiques nous ont permis d'observer le comportement du robot réalisé et constater les différents défauts liés à sa conception.

Chapitre 6

Réalisation du Robot manipulateur à quatre degrés de liberté associé à sa carte de commande basée sur le Microcontrôleur

PIC18F452

6.1 Introduction et objectif

L'Organisation Internationale de Normalisation (ISO 8373:2012) définit le robot comme étant un manipulateur à plusieurs degrés de liberté, à commande automatique, reprogrammable, multi-applications, mobile ou non, destiné à être utilisé dans une application d'automatisation industrielle donnée.

Selon le type de tâche et le secteur d'application, le robot aura une architecture mécanique et un système de commande-contrôle différent. Dans le domaine du nucléaire par exemple les robots sont utilisés pour répondre à un besoin de sécurité et de protection des personnels. En effet, certaines zones des installations nucléaires présentent des risques (radiologiques, chimiques, environnementaux) limitant l'accès au travail humain. D'où l'idée d'utiliser des robots spécialement adaptés à ces conditions extrêmes pour réaliser des opérations de maintenance, de manipulation et d'inspection.

Dans le présent chapitre, nous décrivons la conception, la mise au point et le contrôle d'un bras manipulateur à quatre degrés de liberté (dof) utilisant des motoréducteurs et des servomoteurs. Le contrôle du bras robotique est réalisé par un microcontrôleur PIC 18F452. Le principal devoir du microcontrôleur est de générer des signaux de modulation de largeur d'impulsion (PWM) qui sont appliqués aux différents moteurs et servomoteurs pour obtenir la rotation désirée est d'amener l'organe terminal dans une situation (position et orientation) donnée, selon des caractéristiques de vitesse et d'accélération données. Son architecture est une chaîne cinématique de corps rigides, assemblés par des liaisons appelées articulations. Sa motorisation est réalisée par des actionneurs électriques, i. e. des moteurs DC à caractéristiques différents, qui transmettent leurs mouvements aux articulations par des systèmes appropriés dont les détails seront fournis dans ce qui suit.

Nous nous intéresserons aussi au circuit de commande dédié au contrôle de ce robot. Nous décrivons en particulier l'organe de préhension adaptés à la tâche à effectuer qui sera une "*pince*" à 2 d.d.l.

6.2 Architecture matérielle de l'ensemble du système

Afin de mettre en œuvre la démarche, la figure 6.1 présente l'environnement matériel du projet.

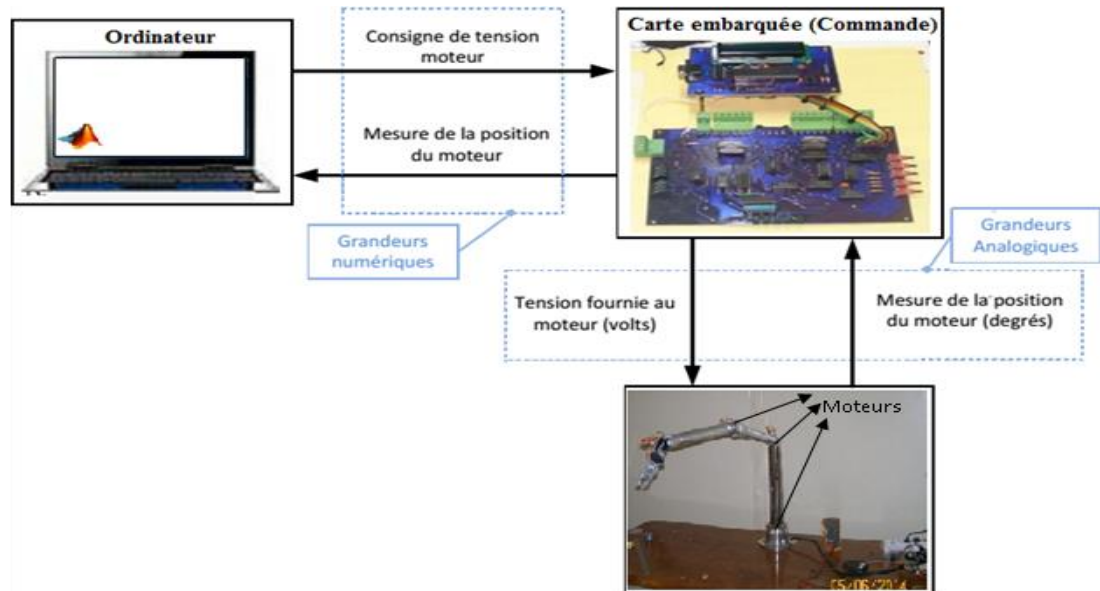


Figure 6.1. Architecture matérielle de l'ensemble du système

Cet environnement contient :

A. Un ordinateur équipé d'un logiciel de modélisation, de simulation et de pilotage

Le rôle de l'ordinateur est de servir de support à la conception des modèles numériques, à leur simulation et à la compilation des résultats. L'ordinateur va également servir à créer des interfaces de mesures et de pilotage afin de commander le robot manipulateur via la carte de contrôle-commande.

B. Le robot manipulateur

Le robot utilise des moteurs à courant continu. Ces moteurs sont commandés par une carte embarquée reliant le circuit de contrôle et le circuit de puissance. La tension fournie est modulable en fonction de la logique de commande programmée.

6.3 Circuit de commande des moteurs

Généralement dans le domaine de la robotique, lorsque l'on souhaite qu'un moteur ou un vérin atteigne une certaine vitesse ou position avec une précision donnée, il est souvent nécessaire d'avoir recours à un asservissement, c'est à dire un système capable d'atteindre et de maintenir une consigne en utilisant une mesure. Il s'agit donc d'un système bouclé, dont il

reste à déterminer la fonction permettant de corriger la commande en fonction de la consigne initiale et de l'erreur mesurée.

Le PID est une technique qui a fait ses preuves dans ce contexte et qui donne de bons résultats. Dans le cas du présent travail, nous avons à faire à un asservissement en position des moteurs à courant continu MCC, munis d'un capteurs de position/vitesse, et par conséquent le schéma bloc du système à implémenter peut être représenté par la forme suivante :

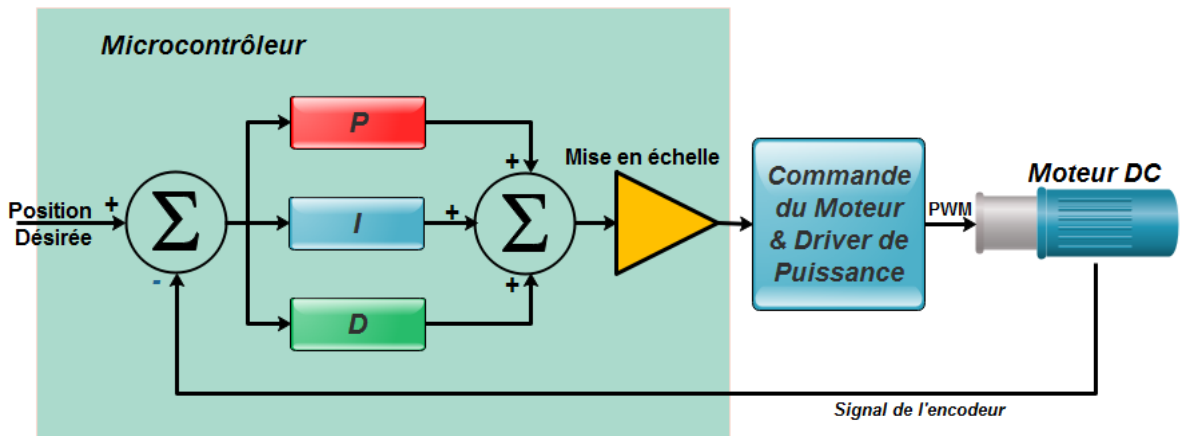


Figure 6.2. Schéma-bloc d'un moteur asservi en position avec boucle de retour unitaire

La simulation du système tant qu'il est nécessite une modélisation mathématique des principaux constituants.

6.3.1 Modélisation mathématique d'un moteur électrique

Avant de procéder à la commande du moteur, une phase essentielle étant sa modélisation. La modélisation du moteur à courant continu consiste à établir la relation entre la tension aux bornes de l'induit du moteur et sa vitesse de rotation (figure 6.3).

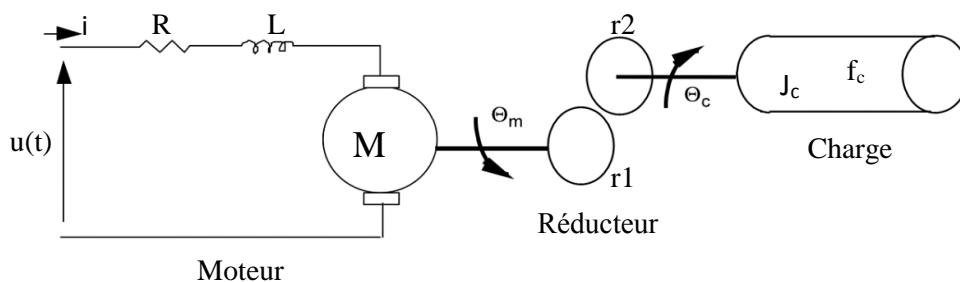


Figure 6.3. Schéma équivalent d'un moteur à courant continu

En utilisant le Théorème du Moment Cinétique et des équations d'électromagnétique, et en négligeant les frottements visqueux, on obtient un système d'équations différentielles linéaires :

$$u(t) = e(t) + Ri(t) + L \frac{di(t)}{dt}$$

$$e(t) = K_e \omega_m(t)$$

$$C_m = K_c i(t)$$

$$J \frac{d\omega_m(t)}{dt} = C_m(t) - C_r(t) - f \omega_m(t)$$

$$C_m(t) - C_r(t) = J_T \frac{d\omega(t)}{dt}$$

- $u(t)$: tension de commande du moteur (V)
- $e(t)$: force contre électromotrice du moteur (V)
- $i(t)$: intensité dans le moteur (A)
- $C_m(t)$: couple exercé par le moteur (N.m)
- $C_r(t)$: couple résistant ramené sur l'arbre moteur (N.m)
- $\omega_m(t)$: vitesse de rotation du moteur (rad/s)
- R : résistance de l'induit (Ω)
- L : inductance de l'induit (H)
- J : inertie équivalente ramenée sur l'arbre moteur (kg.m^2)
- f : paramètre de frottement visqueux (N.m.s)
- K_m : constante de couple (N.m/A)
- K_c : coefficient de force contre électromotrice (V.s/rad)

Parmi les paramètres qui caractérisent le comportement du moteur, certains sont parfaitement connus et d'autres restent indéterminés. L'inductance L et la résistance R de l'induit sont données dans la documentation technique du moteur, il en va de même pour les coefficients K_e et K_m ²⁶. Par contre le paramètre de frottement visqueux 'f' n'est pas forcément connu, en particulier si le moteur est ensuite couplé à un mécanisme. Il faudra soit négliger ce paramètre f ou alors en faire une estimation pour espérer obtenir des résultats valides avec.

La force électromotrice est une tension créée par le bobinage du moteur pour s'opposer à la variation du flux du champ magnétique le traversant.

On en déduit donc le modèle suivant dans le domaine de Laplace (figure 6.4):

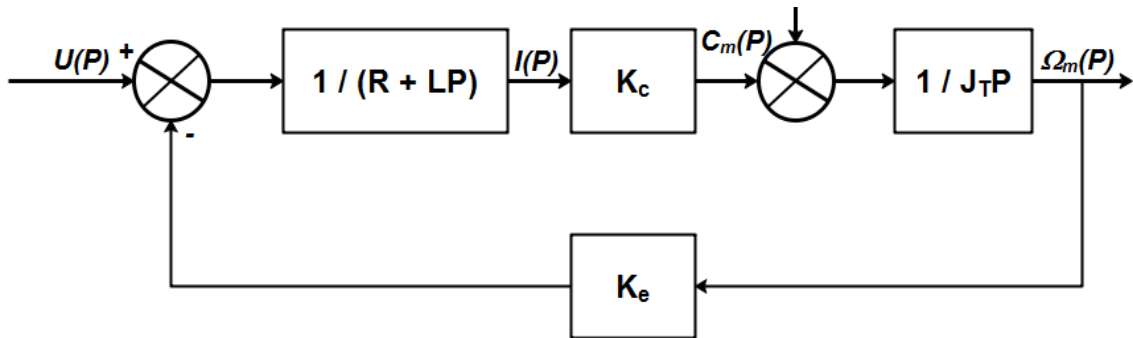


Figure 6.4. Schéma-bloc d'un modèle mathématique du moteur à courant continu

La fonction de transfert associée à U possède les caractéristiques suivantes :

$$H(P) = \frac{\Omega(P)}{U(P)} = \frac{A}{1 + \frac{2\zeta}{\omega_0} P + \frac{1}{\omega_0^2} P^2}$$

²⁶ En règle générale les coefficients K_e et K_m sont si proches qu'il est raisonnable de les considérer égaux, négligeant alors les pertes durant la conversion électromécanique de puissance.

Avec :

$$A = \frac{1}{K_e}$$
$$\zeta = \frac{R}{2} \sqrt{\frac{J_T}{K_e K_c L}}$$
$$\omega_0 = \sqrt{\frac{K_e K_c}{L J_T}}$$

La simulation et la résolution des modèles par le logiciel de simulation imposent de connaître les valeurs de tous les paramètres numériques. La modélisation adoptée est basée sur les données du constructeur du moteur utilisé (figure 6.5) dont les caractéristiques sont :



Figure 6.5. Schématisation Moteur à courant continu et son encodeur incrémental avec 64 CPR

Le moteur utilisé est à Engrenage en métal 50:1 Avec Encodeur (64 CPR)

- Tension nominale : $U_{nom} = 12 \text{ V}$;
- Courant à vide : $I = 300 \text{ mA}$;
- Courant en charge $I = 5 \text{ A}$;
- Résistance aux bornes : $R = 1.91 \Omega$,
- Couple moteur: $C_m = 12 \text{ kg-cm}$ (170 oz-in)
- Inductance : $L = 0.63 \text{ mH}$;

Boucle de retour

Étant donné que l'asservissement du moteur utilise une boucle fermée, il est nécessaire de disposer d'un capteur de position angulaire. L'information retournée est retraitée par calculs pour en déduire une information plus exploitable, dans notre cas c'est l'angle en sortie d'un réducteur. Pour avoir cette information, nous avons utilisé un encodeur rotatif incrémentaux à effet hall: capteurs équipés d'un dispositif magnétique émettant N impulsions par tour.

6.3.2 Asservissement PID

Les termes proportionnel et intégral du contrôleur PID peuvent amener un dépassement de la consigne et des oscillations. Cela implique pour le moteur des inversions de polarité, ce qui est loin d'être idéal. Pour limiter ce phénomène indésirable, on introduit un troisième élément : le terme dérivé. Son action va dépendre du signe et de la vitesse de variation de l'erreur, et sera opposée à l'action proportionnelle. Elle devient prépondérante aux abords de la valeur demandée lorsque l'erreur devient faible, que l'action du terme proportionnel faiblit et que l'intégrale varie peu : elle freine alors le système, limitant le dépassement et diminuant le temps de stabilisation.

$$\text{Consigne}(P) = K_p \varepsilon(P) + K_i \frac{\varepsilon(P)}{P} + K_d P \varepsilon(P) = \varepsilon(P) \left[K_p + K_i \frac{1}{P} + K_d P \right]$$

On obtient alors un asservissement composée d'un terme proportionnel, un terme intégral et un terme dérivé : on parle d'asservissement PID (figure 6.6).

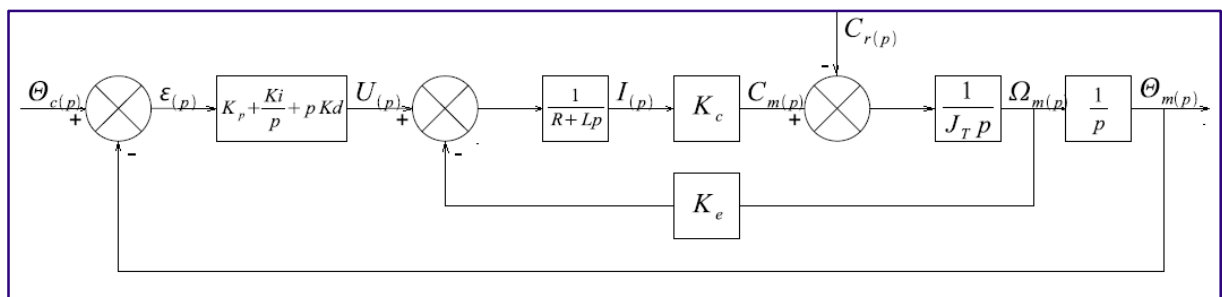


Figure 6.6. Schéma-bloc de l'implantation du PID en régulation de position du moteur

6.3.3 Réglage des coefficients du PID

Pour le choix des coefficients du PID ainsi conçu, nous avons utilisé l'utilitaire Matlab PID Tuner (figure 6.7).

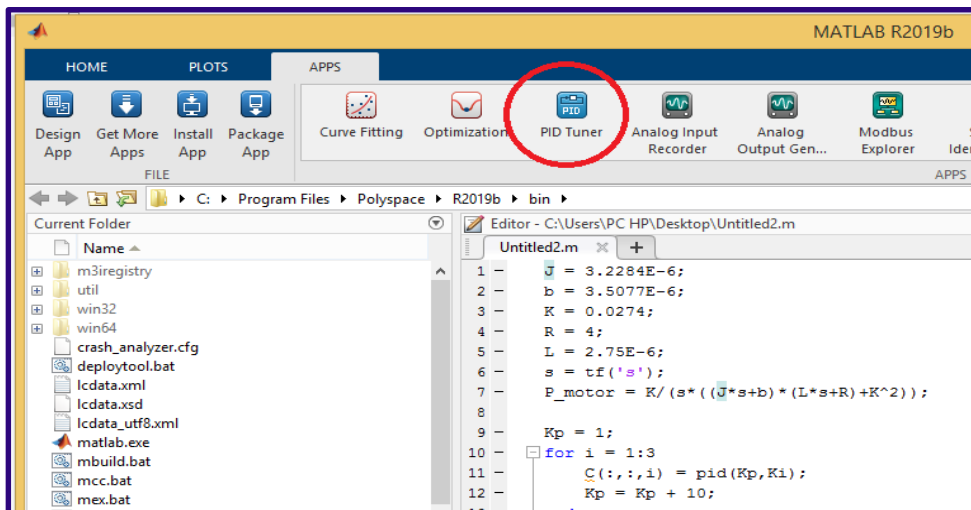


Figure 6.7. Programmation en Matlab de la fonction PID

Après l'accès à cet utilitaire, on importe la fonction du transfert du moteur déjà modélisé en haut. A noter que la fonction de transfert en question, et qui reflète le moteur utilisé, a été introduite dans Matlab sous forme de système linéaire et mise dans l'environnement de travail actuel sous le nom de P_motor (figure 6.8).

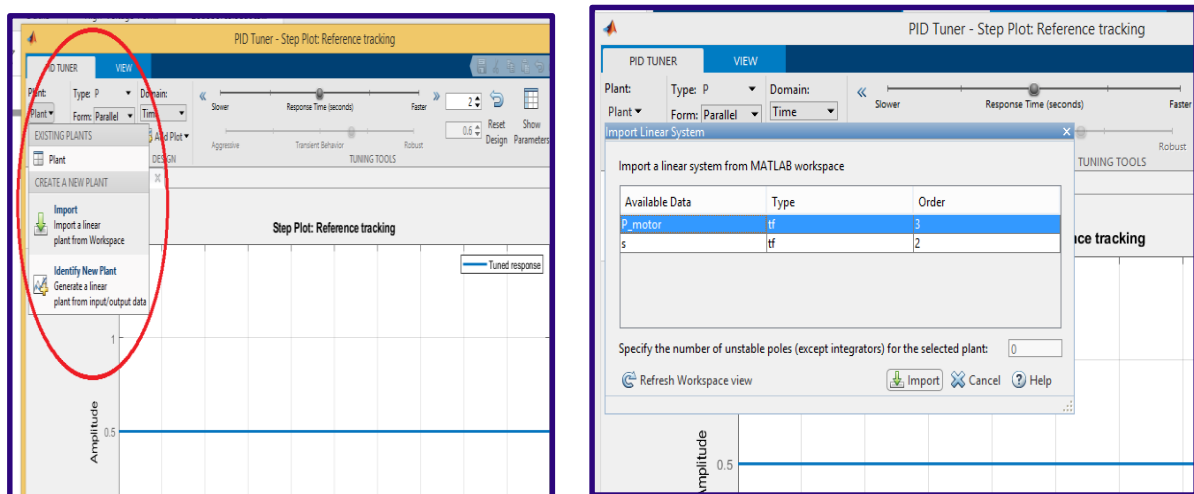


Figure 6.8. Utilitaire PID Tuner de Matlab

Sur ces fenêtres, on choisira l'algorithme du calcul des coefficients et les performances temporels voulu (temps de réponse) et on obtiendra automatiquement les coefficients K_i , K_p et K_d .

Les coefficients ainsi obtenus, seront par la suite inséré dans le programme en assembleur (au niveau du microcontrôleur) pour implémenter le contrôleur PID conçu (figure 6.9).

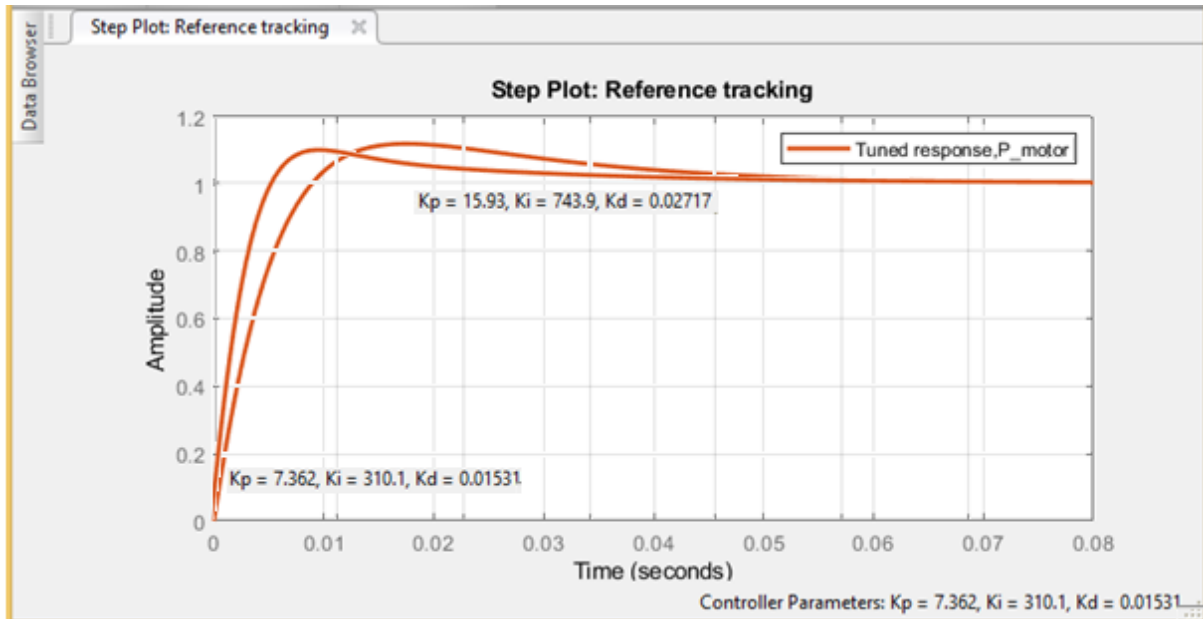


Figure 6.9. Contrôleur PID conçu sur Matlab

6.4 Réalisation et mise en œuvre du robot manipulateur

La réalisation et la mise en œuvre du système nécessite de définir un cahier des charges afin d'identifier les moyens nécessaires, de programmer la réalisation des tâches, et de déterminer les rôles dévolus. Une plateforme de télé-opération est donc nécessaire.

6.4.1 Développement de la structure mécanique

Le robot réalisé est une structure mécanique articulée à quatre degrés de liberté. Sa réalisation vient par la nécessité d'intervenir pour des opérations de contrôle ou de maintenance dans des zones présentant un risque pour les opérateurs humains (contamination par rayonnement dans les installations du réacteur).

Dans sa *première version* (figure 6.10), ce robot utilise des moteurs à courants continu avec réduction de la vitesse et permet ainsi l'augmentation du couple. La valeur du couple dépend du moteur utilisé⁵. Chaque moteur est muni d'un encodeur intégré pour la mesure en ligne de sa position et de sa vitesse. L'encodeur fournit une résolution de 64 par tour de l'arbre moteur.

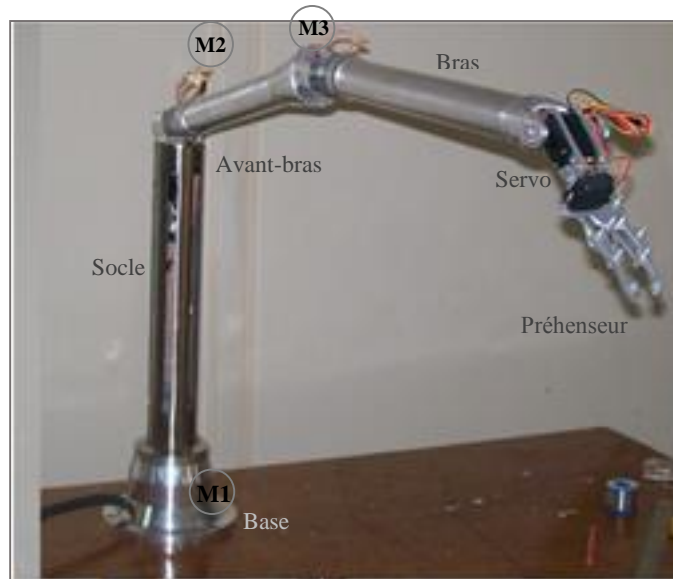


Figure 6.10. Schéma général du robot réalisé

La base du robot constitue l'élément support de l'ensemble ; elle est faite pour supporter le poids de la structure et assurer la stabilité de l'ensemble.

Les segments du robot sont en aluminium pour réduire le couple appliqué sur les moteurs. L'axe du moteur de la base est couplé, via un roulement, à la tige de rotation avec une bague à travers des vis facilitant ainsi son montage-démontage (figure 6.11). Ce qui permet la rotation de l'ensemble d'un angle de 180°.

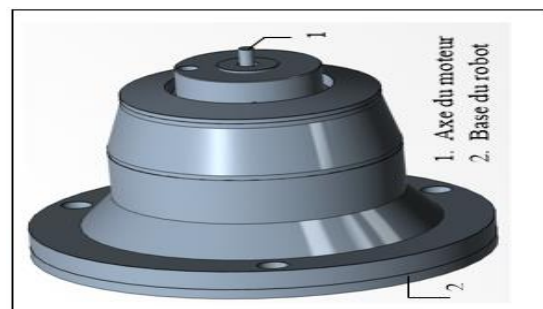


Figure 6.11. Base du robot (Socle) avec moteur d'entraînement

Le couplage du bras et de l'avant-bras avec les moteurs se fait grâce à des engrenages coniques en aluminium (figure 6.12), et couvre un angle de mouvement de 180°.

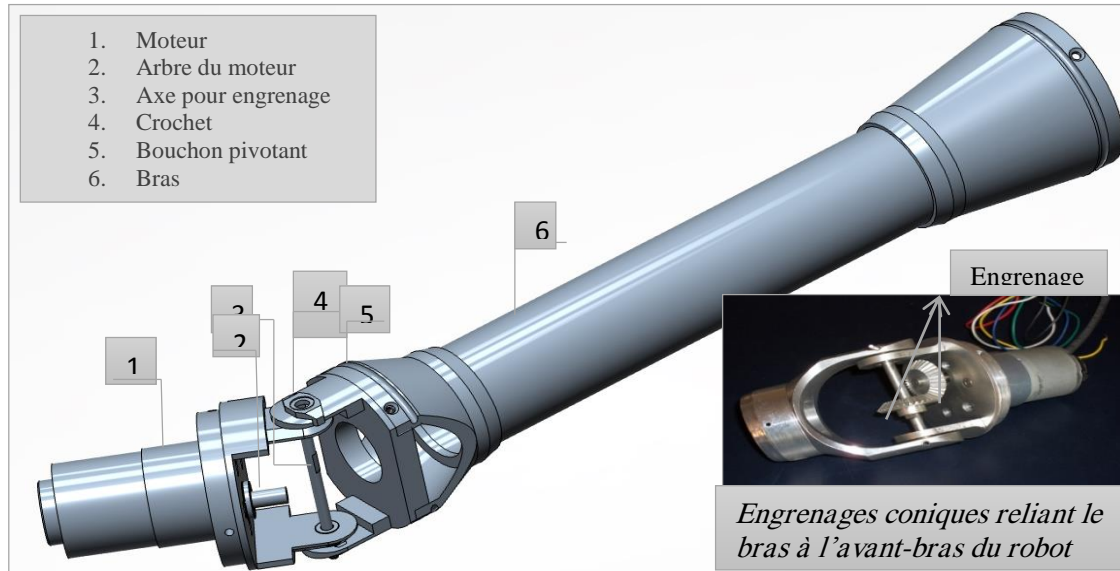


Figure 6.12. Raccordement Epaule et moteur d'entraînement du coude

L'effecteur ou la pince du robot utilise deux servomoteurs pour ses deux degrés de liberté. Un pour effectuer le mouvement de la pince "S06NFSTD", et l'autre pour la préhension de l'objet "HS485HB" et donc l'ouverture et la fermeture de la pince (figure 6.13).

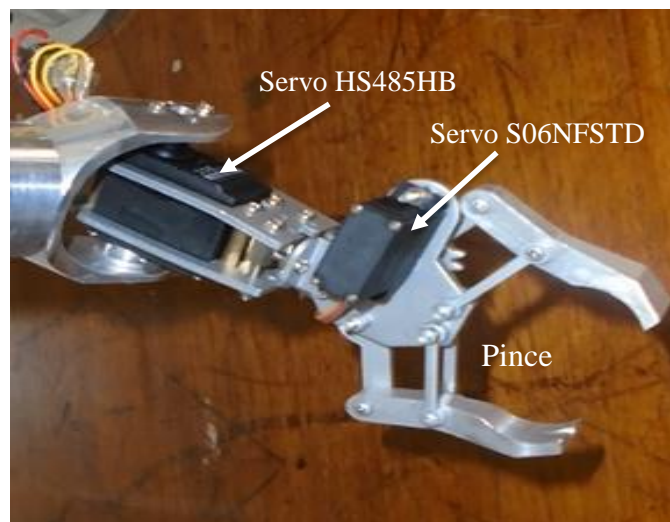


Figure 6.13. Schématisation de la pince

6.4.1.1 Caractéristiques techniques de la pince

La Pince en Aluminium de Dagu est faite en aluminium de 3 mm d'épaisseur et pèse 0.22 kg. Elle est livrée avec 2 servomoteurs et s'ouvre sur plus de 50 mm (5,3 cm). La longueur du préhenseur est de 21 cm.

6.4.1.2 Caractéristiques techniques des servomoteurs de la pince

1. Servo standard (pour changer l'inclinaison du préhenseur)

Le servomoteur S06NFSTD à pignons métalliques, possède un couple de 180,5 oz-in (13 kg-cm) sous 6V, et de 187,4 oz-in (13.5 kg-cm) sous 7.2V ; et une vitesse de 0.18 sec/60° à 6.0V, et de 0.16 sec/60° sous 7.2V.

2. Mini servomoteur (pour ouvrir et fermer le préhenseur)

Le servomoteur HS485HB à pignons karbonite Hitec, possède un couple de 67.00 oz-in (4.82 kg-cm) sous 4.8V, et de 83.00 oz-in (5.98 kg-cm) à 6V ; et une vitesse de 0.22 sec/60° sous 4.8V, et de 0.18 sec/60° à 6V.

6.4.2 Développement de la carte de commande

Le contrôle du robot s'opère au niveau d'une carte embarquée réalisée à base de microcontrôleur PIC18F452. Ce choix a été motivé par sa richesse relative en ressources (instructions, adressages, mémoires...etc.). Toute les fonctions nécessaires pour la commande du robot y sont implantées, à savoir l'alimentation variable des moteurs, lecture des encodeurs, régulation de la position des moteurs, communication avec le PC pour recevoir les commandes et envoyer les informations, etc... Quant au PC il se limite à envoyer les commandes et recevoir les informations de la carte simplifiant ainsi la programmation. Cet échange de tâches entre la carte embarquée et le PC permet la réduction de la complexité de la programmation et donc le risque d'erreur.

6.4.2.1 Description de la carte de commande

Le circuit de commande en puissance des moteurs a été séparé de celui de basse puissance du microcontrôleur (figure 6.14).

Ce circuit remplit les principales fonctions de :

- ✚ Commande et régulation (de position) des moteurs ;
- ✚ Lecture des positions des moteurs ;
- ✚ Communication vers le PC ;
- ✚ Envoie des états des moteurs (positions) vers le PC ;
- ✚ Réception, interprétations et exécution des commandes émanant du PC.

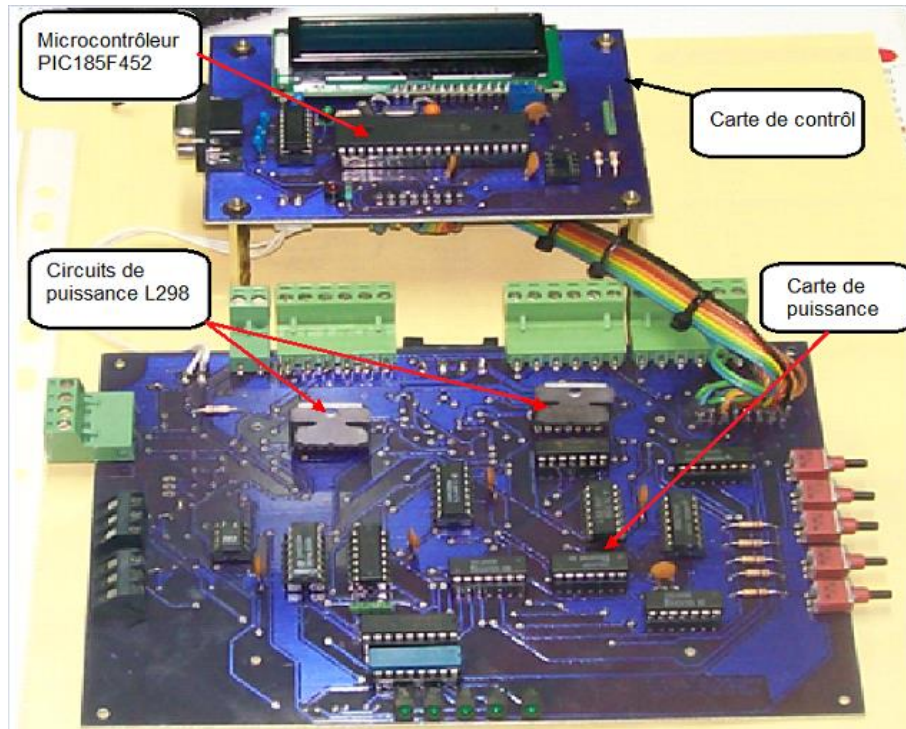


Figure 6.14. Carte embarquée associant la carte de puissance à la carte de contrôle-commande

Un autre aspect pris en compte dans la conception est la flexibilité pour faire face à une éventuelle modification. De ce fait la partie hardware a été scindé en deux parties ; la carte de puissance et la carte de contrôle. La carte de puissance s'articule autour d'un circuit Driver double pont le L298, tandis que la carte de contrôle s'articule autour du circuit microcontrôleur PIC18F452. Le schéma synoptique en figure 6.15, montre les différents blocs de la carte de commande, parmi ces blocs : les encodeurs à quadrature de phases, le port série RS232, l'afficheur LCD, les connecteurs et les mémoires.

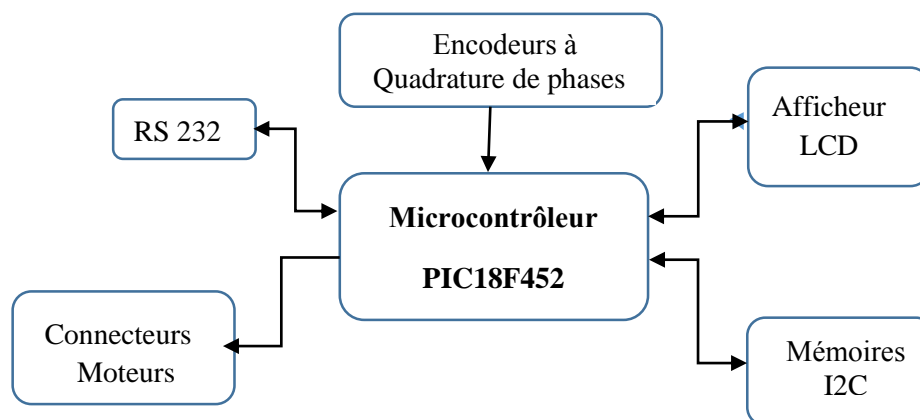


Figure 6.15. Schéma synoptique de la carte à microcontrôleur

6.4.2.2 Le Microcontrôleur

Le microcontrôleur gère plusieurs périphériques dont principalement l'encodeur, qui se trouve monté sur l'arbre du moteur délivrant un nombre de 16 impulsions par tour. Le microcontrôleur compte les impulsions en quadrature de phase afin de déterminer le sens de rotation et la position de la rotation. Les trois lignes d'interruptions du microcontrôleur de plus hautes priorités sont utilisées pour les encodeurs vu l'importance de la position. Il gère également la communication série avec le PC via la liaison RS232. L'afficheur LCD est destiné à afficher la position du moteur en temps réel, et la mémoire I2C permet de mémoriser certaines données comme les données de calibrations ou certaines positions ou parcours, elle sera utilisée en cas où la mémoire on-chip du microcontrôleur est insuffisante.

1. Choix du PIC18F452

Les microcontrôleurs 8 bits de Microchip sont un excellent choix pour les concepteurs d'une grande variété de systèmes embarqués. Ces appareils offrent une combinaison unique de performance, d'efficacité énergétique et de flexibilité de conception. Optimisés pour réduire le temps de développement, ils sont basés sur l'architecture la plus efficace du code pour la programmation en C ou en assembleur.

Faisant partie des systèmes embarqués, le Pic 18F452-I/P (figure 6.16), est un microcontrôleur 8 bits de la famille Mid-Range amélioré (Enhanced). Il a été choisi pour sa puissance (richesse en ses ressources), sa fréquence d'horloge de 40 MHz ainsi que son utilisation relativement simple et présentant les avantages suivants :

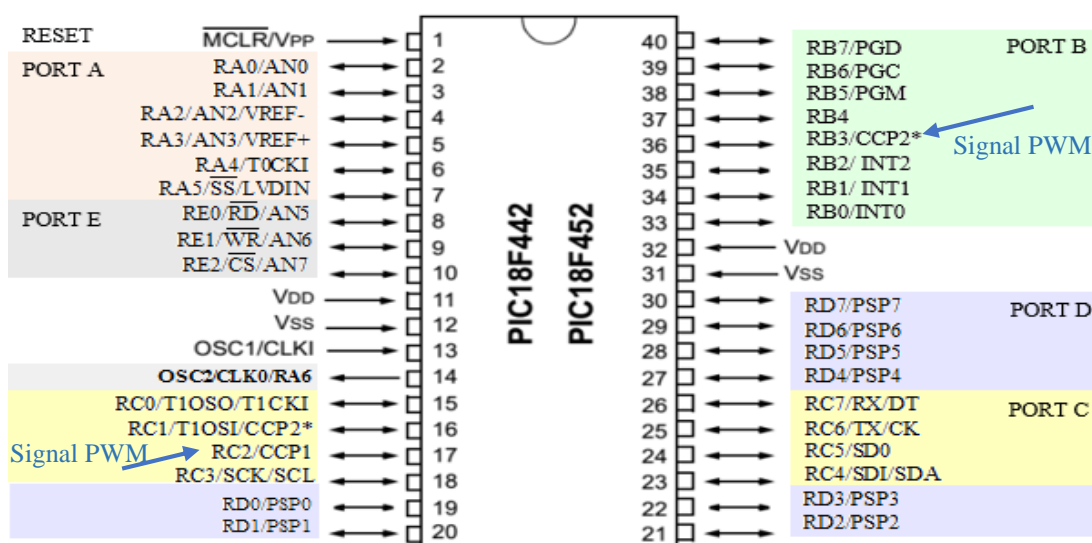


Figure 6.16. Microcontrôleurs Flash améliorés hautes performances avec convertisseur A/N

- ❖ Mémoire et adressage mémoire riche (Mémoire programme et données);
- ❖ Jeux d'instructions variés, largeur d'instruction 16 bit, niveau de priorité pour les interruptions;
- ❖ Grande fréquence d'horloge (40 MHz);
- ❖ Tous les périphériques nécessaires à savoir le PWM, le capture/compare, les Timers (en plus des interruptions de l'unité);
- ❖ Un multiplicateur hardware 8 x 8 bit.

Le PIC18F452 fait partie des microcontrôleurs qui sont optimisés pour les applications nécessitant des performances et une facilité d'utilisation.

L'ADC intégré, le convertisseur numérique-analogique, les comparateurs, la mémoire EEPROM et le détecteur de baisse de tension permettent de créer des applications sans ajouter de composants externes.

2. Programme Assembleur

Le programme du microcontrôleur a été réalisé en assembleur du PIC pour une plus grande optimisation sur le plan de vitesse d'exécution et synchronisation des tâches. La communication avec le PC est faite en utilisant le langage de programmation Matlab pour ces capacités de traitements.

6.4.2.3 Emulateur MPLAB-REAL-ICE

Les produits d'émulation et de débogage de Microchip Technology (MPLAB IDE, MPLAB-ICE) sont très utiles pour le débogage, ils codent et émulent des programmes écrits par l'utilisateur.

Le système de débogage MPLAB® ICD 3, est le débogueur/programmeur matériel haut débit le plus rentable de Microchip pour les dispositifs de contrôleur de signal numérique (DSC) et de microcontrôleur (MCU) Microchip Flash. Ce sont des Microcontrôleurs-Flash facile à utiliser et possèdent une puissante interface utilisateur graphique (de l'environnement de développement intégré IDE MPLAB), inclus avec chaque kit. Toutefois, ces outils ne fournissent pas de capacités de co-traitement et de visualisation avancée des données.

La sonde MPLAB ICD 3 In-Circuit Debugger est connectée au PC du concepteur à l'aide d'une interface USB 2.0 haut débit et est connectée à la cible avec un connecteur compatible avec les systèmes MPLAB ICD 2 ou MPLAB REAL ICE (RJ-11) (figure 6.23). MPLAB ICD 3 prend en charge tous les en-têtes d'émulation.

6.4.2.4 Interface série (MAX232)

Le Max 232 est un pilote et un récepteur RS232 à 2 canaux fabriqués par Maxim Integrated Products, Inc. Il nécessite une alimentation de 5VDC et convertit les niveaux de tension entre logique PC et logique microcontrôleur.

6.4.3 Développement de la carte de puissance

La carte de puissance utilise, en plus du circuit L298 (double pont en H), des circuits logiques qui permettent en association avec le microcontrôleur, d'envoyer la commande vers les moteurs et les servomoteurs (figure 6.17).

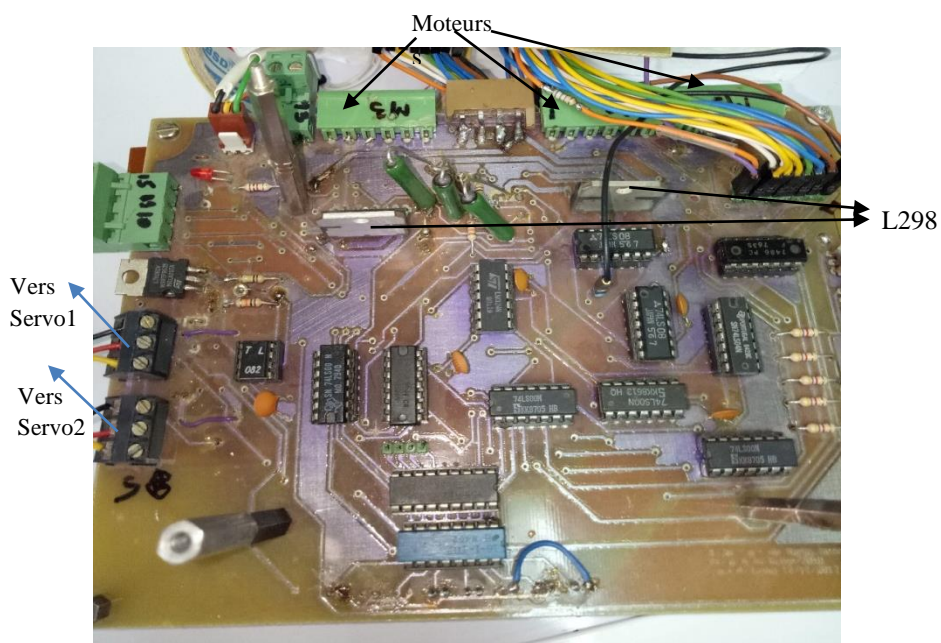


Figure 6.17. Carte de puissance : Moteurs & périphériques

Elle permet de piloter les trois moteurs des articulations du robot et les deux servomoteurs de la pince. Elle est constituée de blocs distincts contenant un circuit de décodage, un circuit de puissance, un circuit de protection de courant et de la signalisation, comme le montre la figure 6.18.

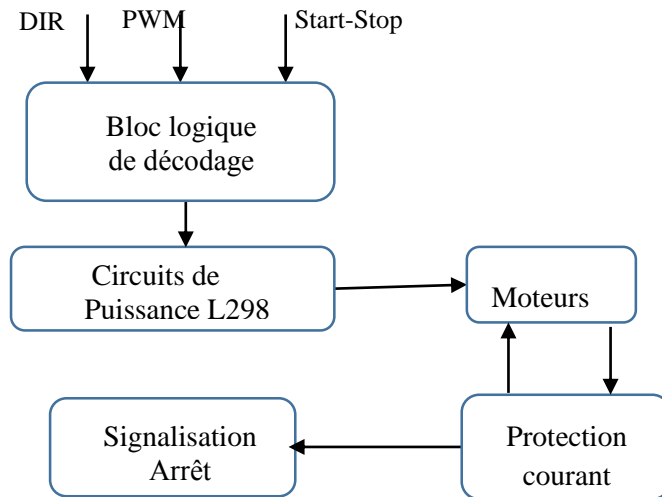


Figure 6.18. Schéma bloc du circuit de puissance

L'alimentation continue des moteurs (12V) ainsi que celle des autres circuits de la carte de puissance est fournie par une alimentation de PC.

6.4.3.1 Génération des signaux de commande

Le circuit de puissance utilisé est un double pont ayant une sortie 2A chacun. Le circuit logique de contrôle intégré permet de faciliter sa commande, à savoir, les entrées logiques pour marche/arrêt, marche directe/inverse et sortie pour la détection du courant (figure 6.19).

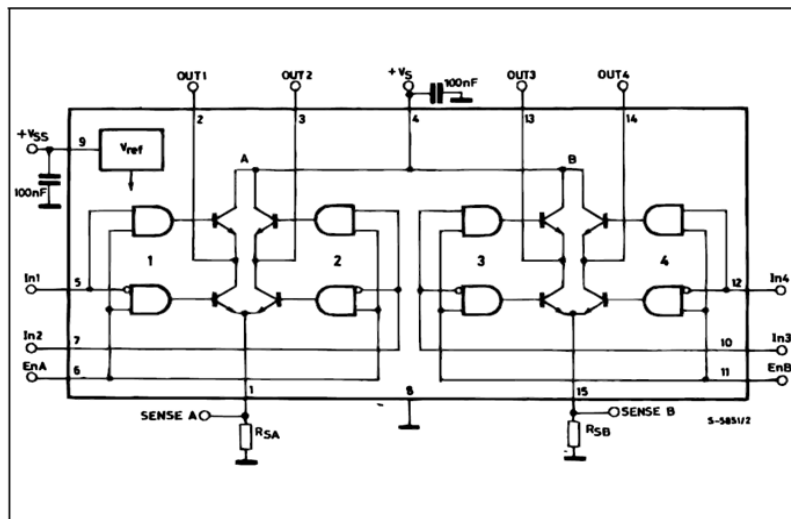


Figure 6.19. Architecture interne du circuit de puissance L298

Le microcontrôleur commande le circuit de puissance à travers trois lignes, une ligne pour le signal PWM pour varier la tension moyenne appliqué au moteur, une ligne "DIR" pour

commander le sens du courant dans le moteur donc sa rotation, et une ligne pour la commande marche/arrêt (Start/Stop).

Ces lignes sont combinées dans un circuit combinatoire logique pour générer des signaux exploitables par le L298 (figure 6.20). Le problème qui n'a pas été tenu en compte dans ce circuit est que lors de l'inversion du sens, le rapport cyclique devient le complément, ceci a été corrigé au niveau software du microcontrôleur en faisant le complément du registre PWM par l'instruction `comf` lors de l'inversion du sens.

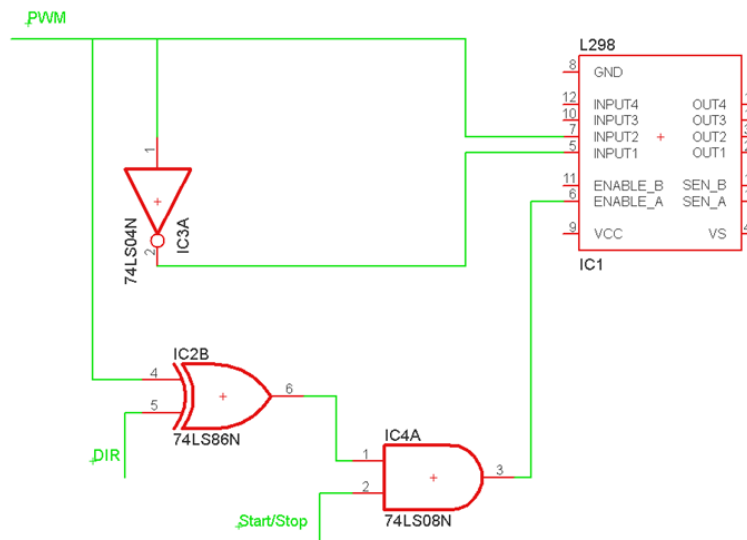


Figure 6.20. Circuit de génération des signaux de commande

6.4.3.2 Circuit de décodage

Ce circuit logique de décodage permet de combiner les signaux PWM, DIR et StartStop afin de commander le circuit de puissance du moteur. Néanmoins le problème dans ce circuit de décodage se pose lors de l'inversion de la rotation, le signal PWM se trouve complémenté, ce qui risque de causer l'arrêt du moteur avec la baisse de tension moyenne. Un circuit de remplacement proposé en figure 6.21, permet de contourner ce problème.

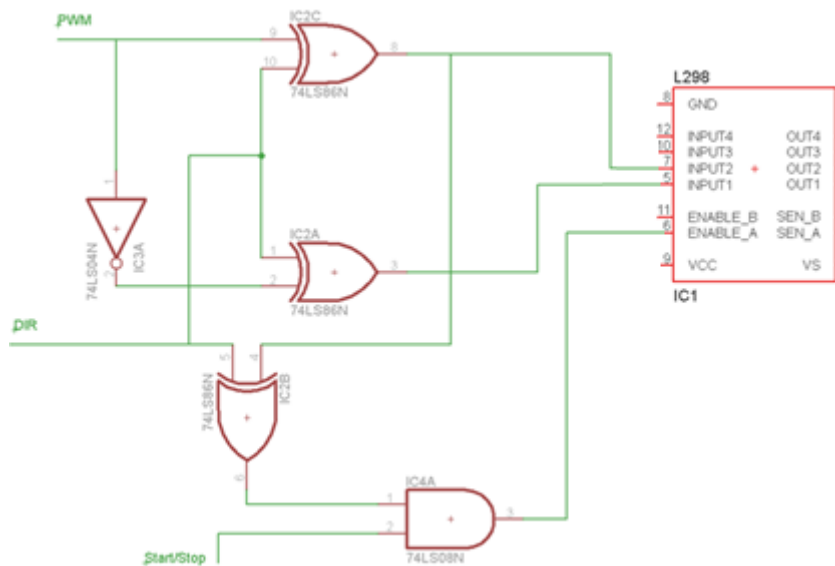


Figure 6.21. Circuit de génération des signaux de commande du L298 tenant compte de la complémentarité du signal PWM

6.4.3.3 Circuit de protection du courant

Afin de protéger le circuit driver L298 d'une surintensité on a ajouté un circuit de protection de courant comme le montre le schéma de la figure 6.22. Le courant maximum de fonctionnement du driver ne dépasse pas 2A.

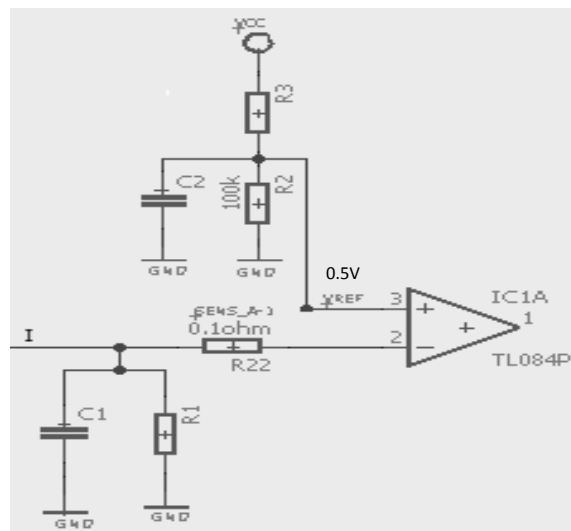


Figure 6.22. Circuit de protection du courant

Le circuit de protection a un temps de réponse fixé par le constante de temps RC^* , répondant aux limites de protection du driver et en même temps pour réduire les effets des transitoires du courant moteur.

6.4.4 Principales fonctions gérées par le microcontrôleur

La figure 6.23 montre les principales fonctions gérées par le microcontrôleur partant de l'initiation des variables et des périphériques jusqu'à la commande des moteurs.

6.4.4.1 Initialisation des variables et périphériques

Le programme commence par initialiser les différents périphériques utilisés dans l'accomplissement des différentes tâches effectuées par le microcontrôleur, à savoir :

- ✚ les trois timers TMR0, TMR1 et TMR2 ;
- ✚ les deux comparateurs COMPAR1, COMPAR2 ;
- ✚ le module PWM ;
- ✚ le périphérique RS232 ;
- ✚ les ports, PORTA, PORTB, PORTC et PORTD ;
- ✚ les différentes interruptions ;
- ✚ initialisation des différentes variables utilisées (espaces mémoires).

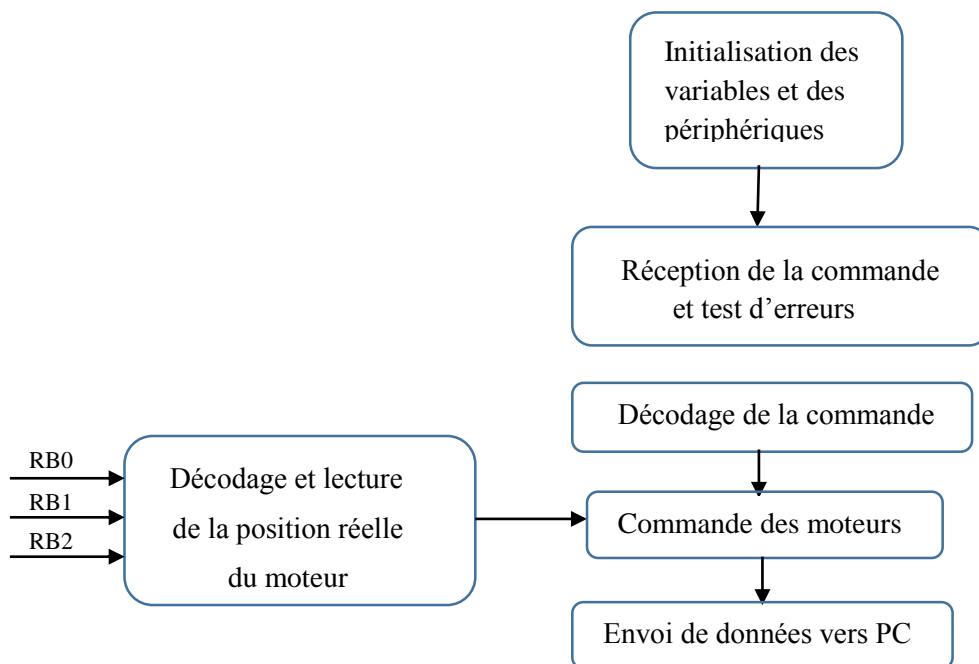


Figure 6.23. Représentation des principales fonctions gérées par le Microcontrôleur

* Le temps de réponse est lié à la constante de temps par la relation : $t_r = 2.2 \tau$

6.4.4.2 Réception de la commande et tests d'erreurs

Cette routine consiste à recevoir et vérifier l'intégrité des données (commandes) provenant du PC. Cette étape est importante et évite la corruption des données et le plantage du système. Ces tests d'erreurs comportent les routines suivantes :

1. Frame errors (FERR)

C'est l'éventuelle erreur détectée au niveau du périphérique RS232 (Erreur de bits) sur l'octet reçu. Cette erreur est lue dans l'état du bit FERR dans le registre d'état RCSTA (Receive Status Register) du périphérique du port série.

2. Overrun errors (OERR)

C'est l'erreur détectée au niveau du port série, due à l'écrasement de l'octet au niveau du receive register dans le cas du non lecture de ce bit au temps opportun. Cette erreur est localisée par la lecture du bit OERR dans le registre d'état RCSTA.

3. Dépassement de temps (time out)

C'est une routine écrite qui permet de vérifier si le nombre total d'octets du message est présent dans un laps de temps déterminé, faute de quoi il faut réinitialiser le buffer de réception. Ceci permet d'éviter le blocage du programme.

4. Vérification checksum

Une vérification de checksum est opérée sur les octets du message après leur réception.

6.4.4.3 Décodage de la commande

L'instruction de commande de la position du moteur est établie par la syntaxe suivante : Type de moteur, position, checksum. En exemple "A, 200, checksum".

En réception au microcontrôleur, le programme décode cette séquence en commençant par l'entête du message (type de moteurs), et la position (à atteindre) en vérifiant aussi le checksum puis ces informations de commande sont affectées à la routine de la commande de position du moteur.

6.4.5 Commande des moteurs

La routine de commande du moteur consiste à appliquer la tension PWM en comparant la position réelle (lue sur l'encodeur) avec la position désirée (consigne).

Dans une première étape, la commande du moteur a été effectuée sur un moteur à vide en boucle ouverte (tout ou rien). Suite au dépassement inhérent dans ces cas de commande, une

régulation proportionnelle a été ajoutée dans un premier temps. Dans ce but, une caractérisation a été réalisée pour s’assurer de la proportionnalité entre le signal d’excitation PWM et la vitesse du moteur. La figure 6.24 montre une nette proportionnalité à partir d’un rapport cyclique de 47%.

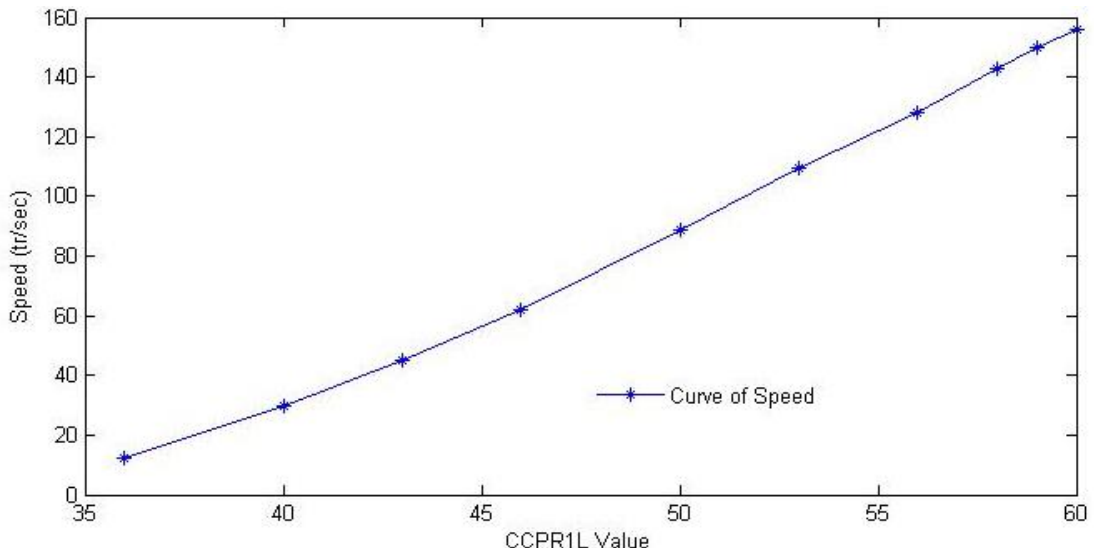


Figure 6.24. Courbe de vitesse en fonction de l’excitation en tension exprimé en valeur du registre CCPR1L fixant le duty cycle pour le moteur 50 : 1

Cette courbe présente la vitesse en fonction du DutyCycle ou CCPR1L déduites du tableau ci-dessous.

Rapport cyclique (%)	Tension moyenne (V)	Vitesse (tr/sec)	CCPR1L
59.68	7.1616	15.63	25/37
64.5	7.74	28.13	28/40
69.35	8.32	43.75	2B /43
74.20	8.90	57.81	2E/46
80.64	9.67	81.25	32/50
85.48	10.25	98.44	35/53
90.32	10.83	115.6	38/56
93.55	11.22	129.7	3A/58
95.16	11.42	134.4	3B/59
96.78	11.61	137.5	3C/60

Tableau 1. Représentation de la vitesse en fonction du rapport cyclique et CCPR1L

CCPR1L=3D n’est pas réalisable car la valeur du PR2 ne doit pas dépasser celle du CCPR1L. PR2 détermine la période du timer ‘TMR0’, la plage permise au fonctionnement du moteur varie entre 58% et 98% (tableau 1).

6.4.5.1 Courbes de position du moteur 29 :1 pour une régulation on/off

La base du robot est faite pour assurer la stabilité de la structure. L'axe du moteur de la base (rapport de réduction 29 :1) est couplé à la tige de rotation via un roulement, ce qui permet la rotation de l'ensemble de la structure sur une plage allant jusqu'à 180°. Le nombre de révolution par tour d'arbre du moteur est estimé à 16.6. La position consigne représentant un demi cycle est égale à 241 (qui est $29 \times 16.6 / 2$). La figure 6.25 montre l'évolution de la position du moteur pour différentes valeurs du rapport cyclique.

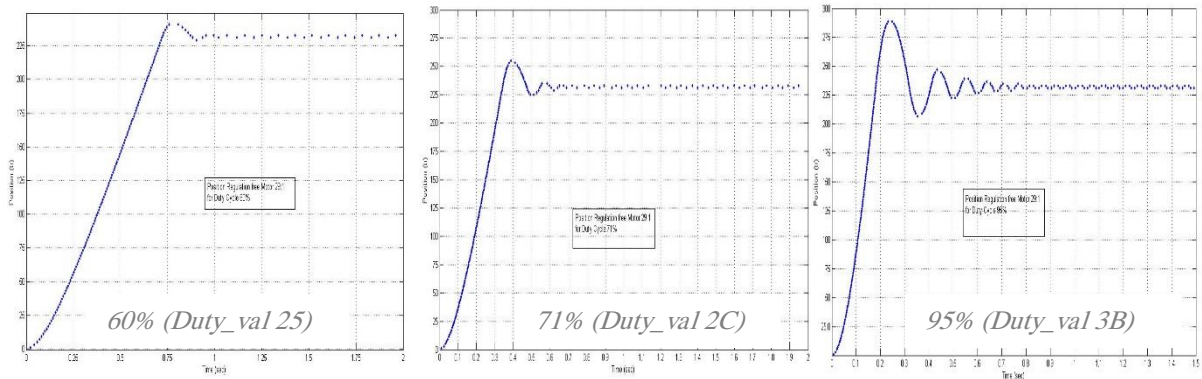


Figure 6.25. Régulation on/off de position du moteur 29 :1 à vide pour différentes valeurs du rapport cyclique

6.4.5.2 Courbes de position du moteur 29 :1 pour une régulation proportionnelle

En appliquant une régulation proportionnelle sur ce moteur, la courbe de position est comparée à celle en boucle ouverte pour un rapport cyclique de 95% (CCPR1L=3B) (figure 6.26).

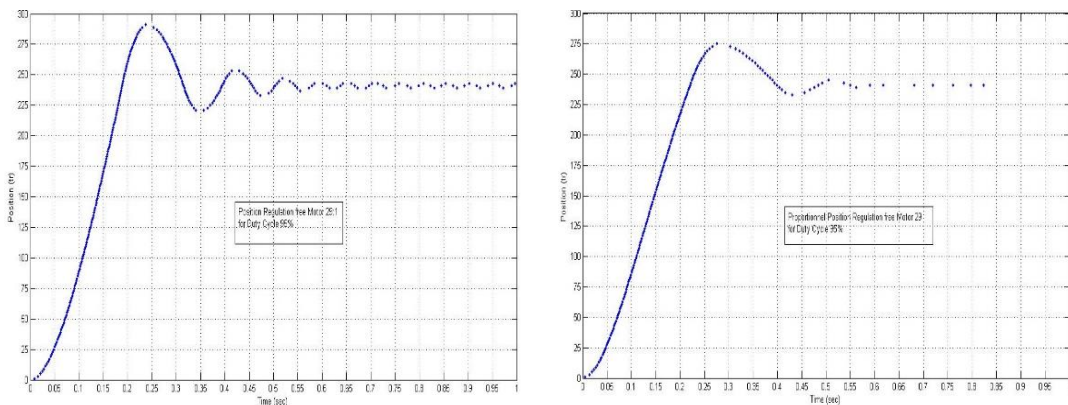


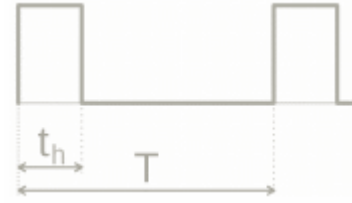
Figure 6.26. Régulation proportionnelle de position du moteur 29 :1 comparée à une régulation on/off pour un rapport cyclique de 95%

Le rapport cyclique (Duty cycle) qu'on note par α et est représenté par l'expression: $\alpha = 100 \frac{t_h}{T}$ Exprimé en pourcent

Si $t_h=0$, $\alpha=0\%$ et la tension moyenne de sortie est nulle.

Si $t_h=T$, $\alpha=100\%$ et la tension moyenne de sortie est égale à

$$V_{cc}. \quad V_m = \frac{t_h}{T} V_{cc}$$



6.5 Essais réels de télé-opération du robot

Le principal objectif de ce travail consiste à concevoir, développer et mettre en œuvre un bras de robot²⁷ compétitif avec un système de contrôle-commande amélioré

La réalisation du robot manipulateur a été accomplie dans le but de pouvoir répondre au minimum requis de mouvement pour le besoin de notre utilisation. La principale exigence considérée dans notre conception est la précision afin de minimiser l'erreur, garantir la répétabilité,...etc. [105,106].

6.5.1 Plateforme de la télé-opération

La réalisation par le robot d'une tâche de télé-opération se base sur la plateforme de télé-opération (figure 6.27).



Figure 6.27. Environnement réel de télé-opération du robot réalisé

²⁷Les critères principaux pour choisir une structure en vue d'une application donnée sont: ses performances géométriques (forme et dimensions de son volume de travail) et les caractéristiques de la tâche à réaliser.

La tâche à effectuer peut être du type pick-and-place, perçage, découpage, sciage, démantèlement (dans le cas des sites nucléaires),...etc.

La pièce mère dans cette plateforme est le robot manipulateur. Pour son pilotage, nous avons utilisé une carte de commande à base du microcontrôleur, le PIC18F452, l'interfaçage avec le PC via le port série utilisant ainsi le circuit Max 232 (figure 6.22), MPLAB Real ICE (figure 6.28) et une carte de puissance basée sur le driver L298.



Figure 6.28. Carte de commande à base du pic 18F452 et périphériques

Le PIC18F452 utilise différentes ressources à savoir les périphériques PWM, Capture/Compare, Timer, ...etc. Ces ressources ont permis l'exécution adéquate des différentes tâches. MPLAB, un environnement informatique de développement pour PIC et installé sur PC (figure 6.29), permet d'éditer le code assembleur (reconnaissance des instructions et des variables internes), le compiler et simuler le comportement du PIC (ceci permet de tester le bon fonctionnement d'un programme "débogage" par simulation).



Figure 6.29. MPLAB Real ICE, Câblage du kit ICD

6.5.2 Débogage et simulation

L'environnement de développement MPLAB avec le kit ICD 2 permet en premier lieu, la simulation du PIC sur PC avec débogage, ensuite l'exécution avec débogage sur PIC grâce au débogueur du kit ICD. Tout ceci s'effectue sur un contrôleur d'interface périphérique.

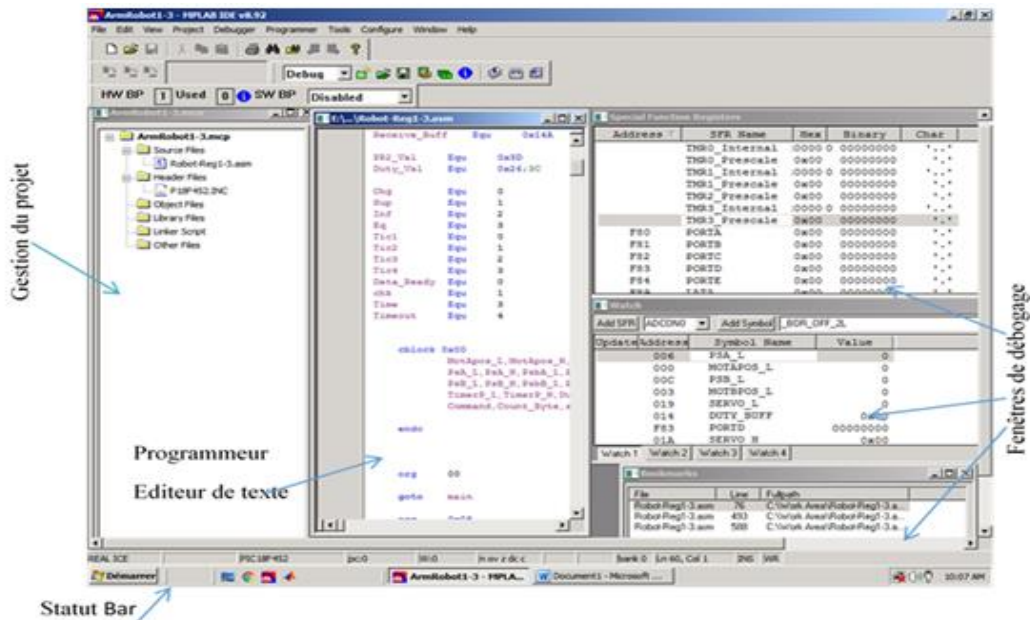


Figure 6.30. Contrôleur d'interface périphérique

6.5.3 Calcul de la PWM de commande des actionneurs

Afin de permettre le calcul de la position et de la vitesse des actionneurs du robot, nous devons d'abord configurer les ports du microcontrôleur pour déterminer les entrées/sorties, la sortie PWM nécessaire à la commande des moteurs.

- ✚ Configurer les ports en entrée '1' ou en sortie '0'
- ✚ Configurer le périphérique en PWM, Mettre RC2/CCPR1L en sortie
- ✚ L'oscillateur est configuré dans le timer 1 T1CON
- ✚ Créer 1s utilisant TMR0 pour compter 0.1ms et incrémenter jusqu'à 1s. Sec_L et Sec_H.
- ✚ Configurer Timer 3 pour l'oscillateur 32768 kHz
- ✚ Configurer DIR3 en sortie
- ✚ Le TRISC commande le sens
- ✚ PR2 détermine la période de TMR2

Le microcontrôleur permet, à l'aide de toutes ses fonctions, d'actionner les moteurs et déterminer leurs positions par la lecture des encodeurs. Les trois principales lignes du microcontrôleur sont:

- ✚ Le signal PWM pour faire varier la tension appliquée au moteur (expression 6.3) ;
- ✚ Une ligne "DIR" pour commander le sens du courant dans le moteur et donc sa rotation et une ligne pour démarrage ou arrêt des moteurs (Start/Stop).

$$PWM\ Period = 10^{-7} * [(PR2) + 1] \quad (6.3)$$

$$Duty\ \% = \frac{[CCPR1L]+4}{4 \times [PR2+1]} \frac{CCPR1L}{[PR2+1]} \quad (6.4)$$

1. La tension moyenne du moteur

$$V = Duty\ (\%)*12V \quad (6.5)$$

2. Calcul de la vitesse en fonction du rapport cyclique

$$\frac{PWM_{DutyCycle}}{PWM_{Period}} = \frac{CCPR1L}{3E} \rightarrow \gg CCPR1L = 3E * \frac{PWM_{DutyCycle}}{PWM_{Period}} \quad (6.6)$$

6.5.4 Forme du signal PWM appliqué aux actionneurs du robot

Notre robot utilise trois moteurs pour chaque articulation, Ces moteurs sont prévus pour une utilisation à 12 V, ils sont dotés d'une boîte de vitesse en métal (50:1 et 100:1 pour l'entraînement du bras et de l'épaule du robot, 29:1 pour la rotation de la base)²⁸ en plus d'un encodeur en quadrature intégré, qui fournit une résolution de 64 compte par tour de l'arbre du moteur. Ceci est équivalent à 3200 coups par tour de l'arbre de sortie, pour le rapport de réduction 50:1, et 6400 coups par tour de l'arbre de sortie, pour le rapport de réduction 100:1.

Le codeur à effet Hall à deux canaux est utilisé pour détecter la rotation d'un disque magnétique sur saillie arrière de l'arbre moteur. Le codeur en quadrature offre une résolution de 64 comptes par tour de l'arbre du moteur. Pour calculer le nombre de tours par tour de la sortie de la boîte de vitesses, multipliez le rapport de transmission par 64. Le moteur/codeur dispose de six conducteurs codés par couleur de 28 cm (11").

²⁸Les principales caractéristiques des trois moteurs à 12 :

- 100 tr / min et 300 mA à vide, 220 oz-in (16 kg-cm) et à 5 A de décrochage (rapport de vitesse 100:1).
- 200 tr / min et 300 mA à vide, 170 oz-in (12 kg-cm) et 5 A de décrochage (rapport de vitesse 50:1).
- 350 tr / min et 300 mA à vide, 110 oz-in (8 kg-cm) et 5 A de décrochage (rapport de vitesse 29:1).

L'alimentation du moteur s'effectue à travers les deux fils : noir (GND) et rouge (12V). Le capteur à effet Hall nécessite une tension d'entrée Vcc (fil bleu), comprise entre 3,5 et 20 V et consomme un maximum de 10 mA.

Les sorties A (fil jaune) et B (fil blanc), sont des ondes carrées de 0 V à Vcc déphasées d'environ 90° (figure II.2).

La capture d'oscilloscope suivante (figure 6.31) montre les sorties de codeur A et B (jaune et bleu) en utilisant une tension de moteur de 12 V et celle du capteur Hall Vcc de 5 V.

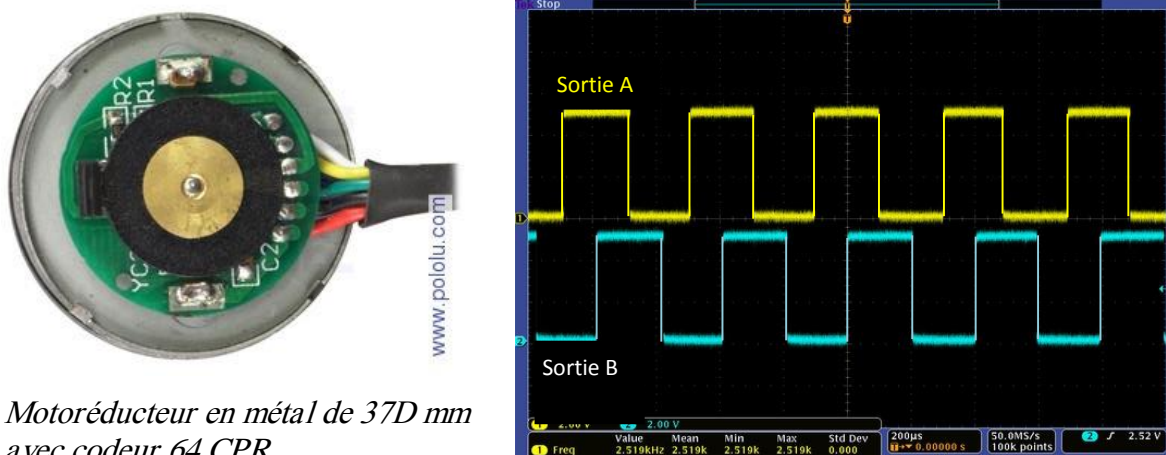


Figure 6.31. Sorties codeur A et B pour motoréducteur avec 64 codeurs CPR (12 V).

En comptant à la fois les fronts montants et descendants des sorties A et B, il est possible d'obtenir 64 coups par tour de l'arbre moteur.

L'utilisation d'un seul bord d'un canal entraîne 16 coups par tour de l'arbre du moteur, de sorte que la fréquence de la sortie A dans la capture de l'oscilloscope est égale à 16 fois la fréquence de rotation du moteur.

6.5.5 Forme du signal PWM appliqué aux servomoteurs

La préhension des objets s'effectue par une pince à deux degrés de liberté utilisant deux servomoteurs un pour la translation et l'autre pour la saisie des objets, et sont asservis en position angulaire sur une plage allant de 0° à 180°. Trois fils de connexion du servomoteur, deux pour l'alimentation (Vcc et GND) et l'autre fil représente l'entrée de commande, un signal numérique modulé en impulsions.

Le servomoteur fonctionne sur une période allant de 0 à 2ms et le rapport cyclique définit l'angle du servomoteur. Une valeur de 5% donne un état haut de 1ms, ce qui correspond à un

angle de 0° ; 10% donne un état haut de 2ms ce qui convient à un angle de 180° et par linéarité, la valeur 7.5 % donne un état de 1.5 ms, qui correspond à 90° (figure 6.32).

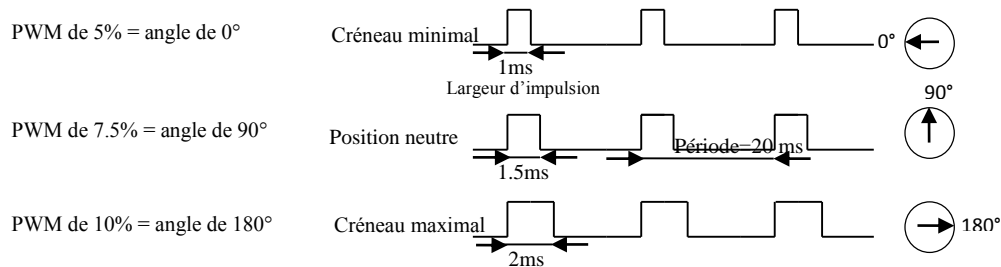


Figure 6.32. Commande en PWM du servomoteur

6.6 Application des signaux PWM réels et commande du robot

Lors des essais de télé-opération sur le robot, les signaux de commande en PWM envoyés vers les moteurs/servomoteurs, sont affichés sur l'oscilloscope (figure 6.33).



Figure 6.33. Affichage des signaux de commande des actionneurs du robot

Le circuit de commande à base de microcontrôleur permet d'interfacer des périphériques tels que des capteurs de distance, ou une carte de contrôle de moteur. Le circuit permet également la communication avec un ordinateur via une liaison série RS232. Cependant, et avant toute action, il est nécessaire de configurer certaines fonctions du microcontrôleur comme les ports, les registres, les périphériques, les timers,...etc.

La fonction PWM destinée à la commande des moteurs a été générée à partir du pic via une ligne de commande utilisant le port C "RC2/CCPR1L" cette commande est envoyée aux moteurs à travers les circuits de puissance L298 pour actionner les articulations du robot.

Des essais ont été réalisés sur les moteurs de l'articulation de la base et celle du coude.

Nous représentons la forme du signal PWM issu du microcontrôleur et les signaux à la sortie des encodeurs des moteurs dans les deux canaux A et B.

6.6.1 Signaux PWM réels délivré par le microcontrôleur pour la commande des moteurs

Le signal de commande des différents moteurs destinés au pilotage du robot à travers ses articulations est généré par le microcontrôleur en pin 17. Sa forme est schématisée par la figure 6.34.

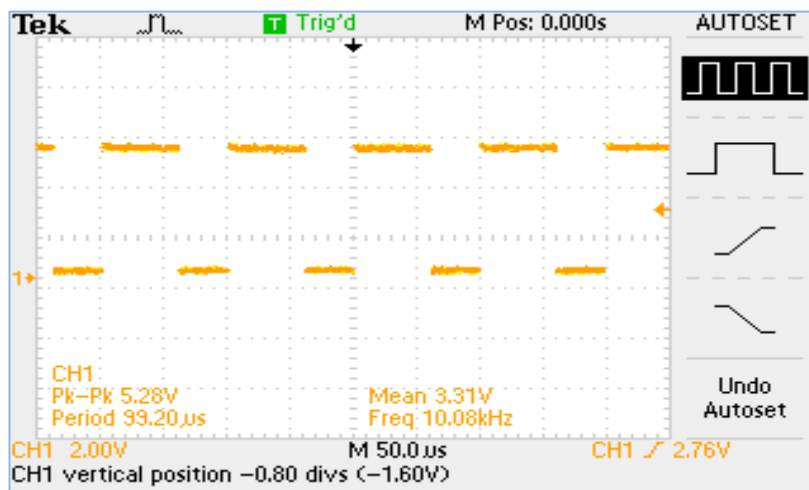


Figure 6.34. Signal PWM réel issu du pic18F452 (pin17)

6.6.2 Signal PWM réel délivré par le microcontrôleur pour la commande du moteur dans les deux sens

Le signal de commande PWM issu du microcontrôleur prend la forme de la figure 6.35 lorsque l'un des moteurs est en fonctionnement ((a) Lorsque le moteur tourne dans un sens et (b) lorsque le moteur tourne en sens inverse).

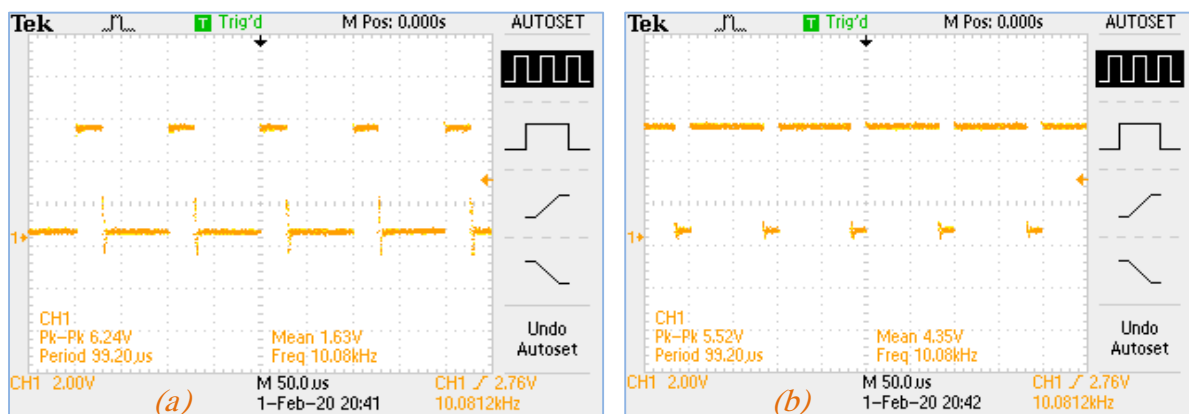


Figure 6.35. Signal PWM issu du pic18F452, moteur tourne dans un sens(a) et sens opposé(b)

6.6.3 Signal réel à la sortie de l'encodeur du moteur, sorties A et B

L'encodeur du moteur possède six connexions, deux des connexions (rouge et noir) sont destinés à l'alimentation du moteur, deux autres connexions (bleu et noir) pour l'alimentation 5V de l'encodeur et les deux derniers (jaune et blanc) sont les sortie A et B.

Le signal en sortie de l'encodeur sur le canal A est représenté par la figure 6.36 pour différentes valeurs de la fréquence.

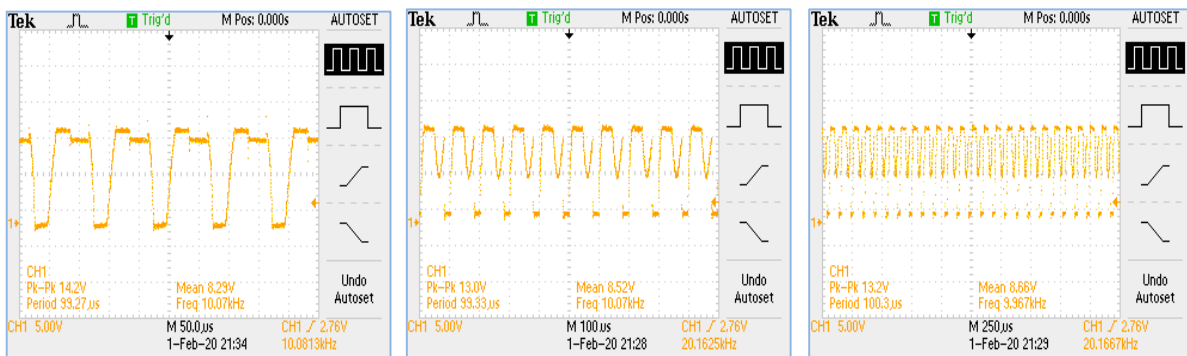


Figure 6.36. Forme du signal à la sortie 'output 1' de l'encodeur du moteur de la base

Le signal en sortie de l'encodeur sur le canal B est représenté par la figure 6.37 pour deux valeurs de la fréquence.

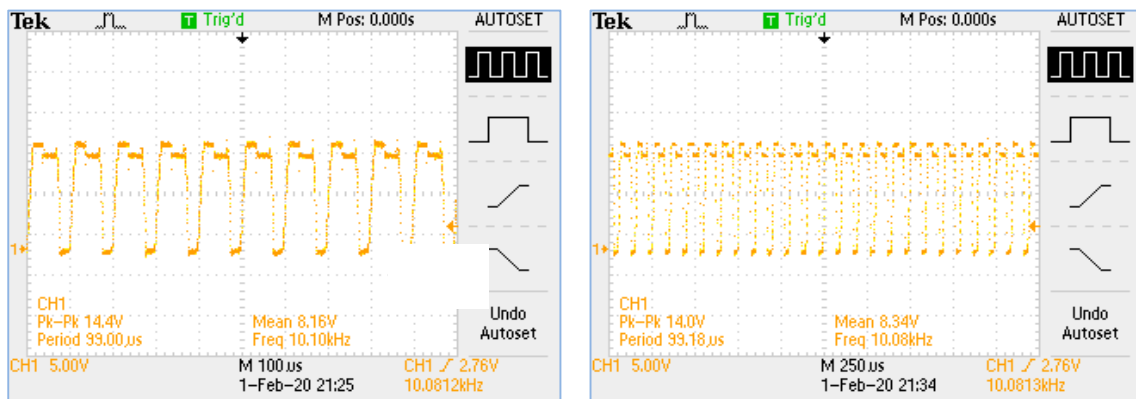


Figure 6.37. Forme du signal à la sortie 'output 2' de l'encodeur du moteur de la base

A partir des figures précédentes, on constate un décalage entre le signal des deux canaux de sortie de l'encodeur du moteur comme en théorie (figure 6.31).

6.6.4 Signal PWM réel délivré par le microcontrôleur pour la commande des servomoteurs

Le servomoteur utilisé pour la translation est du type 'S06NFSTD' ayant pour caractéristique, une alimentation de 6 à 7.2 V, un couple de 13 Kg.cm sous 6V et une vitesse de 0.18s/60°.

Le servomoteur utilisé pour la saisie d'objets du type 'HS485HB', est un servo au format standard de haute performance. Il intègre un roulement à billes assurant précision et des pignons karbonite pour la solidité. Il possède un couple de 4,8 kg.cm à 4,8V et 6,0 kg.cm à 6.0V, et une vitesse de 0,22sec/60° à 4,8V et 0,18sec/60° à 6.0V.

La commande des deux servomoteurs se fait en PWM, générée sur le port B pin 36, 'CCP2CON' module compare en toggle mode (mode basculement) et programmé en mode Compare avec la configuration du timer 3 en 'T3CON' avec un rapport cyclique de 95% (CCPR1L=3B). La forme du signal PWM est donnée en figure 6.38.

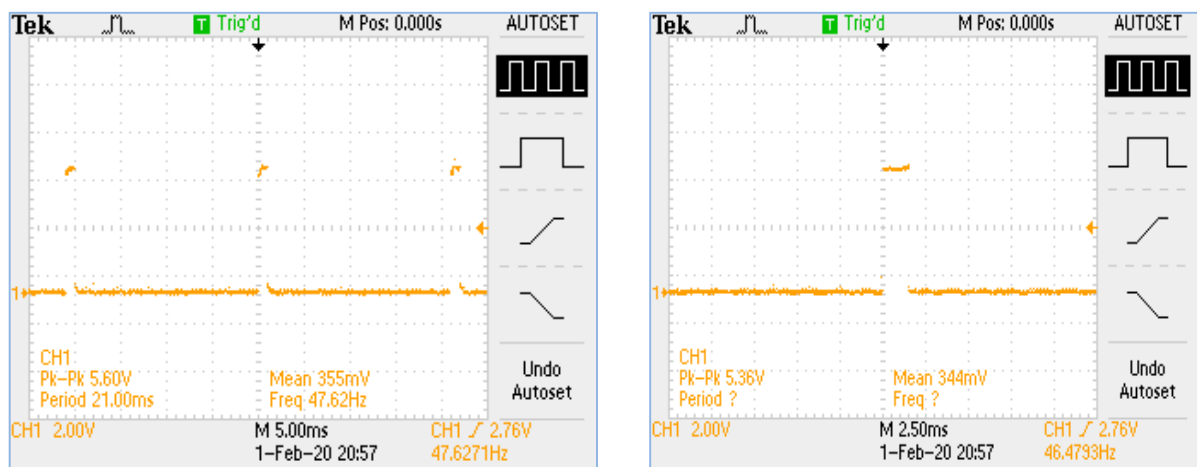


Figure 6.38. Forme du signal PWM issu du Pic18F452 pour commander les servomoteurs

La valeur **0x21** → **1ms** Correspond à la position minimale permise (dans le programme).

La valeur **0x41** → **1ms** Correspond à la position maximale permise (dans le programme).

6.6.5 Signal PWM réel relevé sur la ligne de commande des servomoteurs

Le servomoteur possède 3 fils de connexion, deux pour l'alimentation Vcc et GND et l'autre fil (de couleur jaune en général) est destiné à la commande du servomoteur. La forme du signal de commande PWM relevé sur oscilloscope pour le servomoteur destiné à la translation de la pince, est représentée par la figure 6.39 pour différentes valeurs de fréquence.

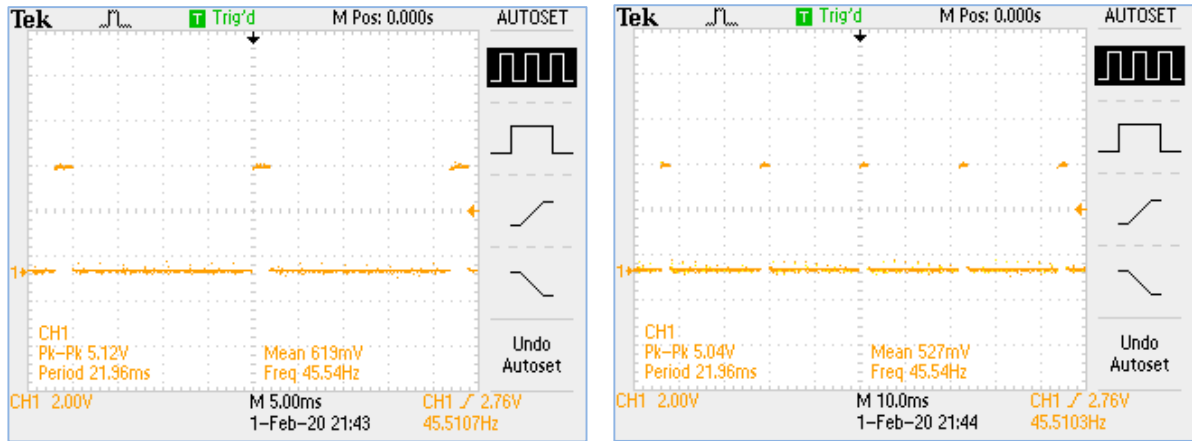


Figure 6.39. Forme du signal à la sortie commande du servomoteur en translation de la pince

La forme du signal de commande PWM relevé sur oscilloscope pour le servomoteur destiné à l'ouverture/fermeture de la pince, est représentée par la figure 6.40 pour différentes valeurs de la fréquence.

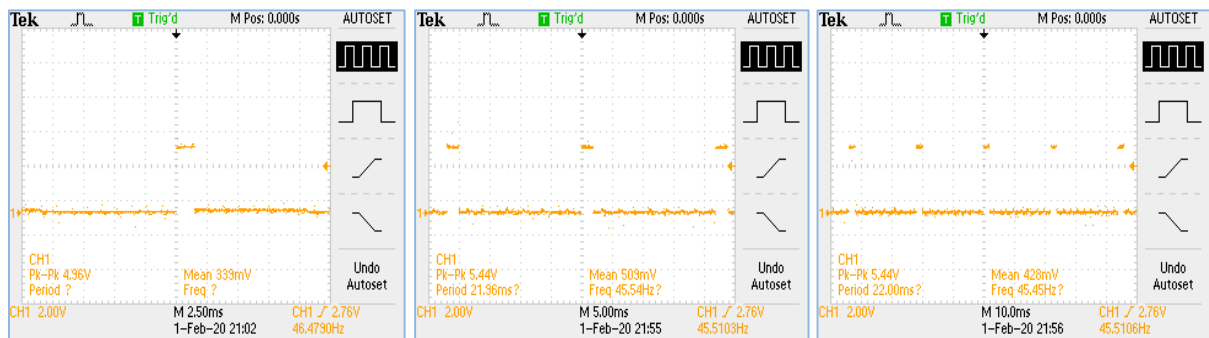


Figure 6.40. Forme du signal à la sortie commande du servomoteur en ouverture/fermeture

Les signaux de sortie de l'encodeur des deux autres moteurs présentent les mêmes signaux. Les séquences de fonctionnement des moteurs sont déterminées par l'opérateur selon la tâche à effectuer.

6.6.6 Evolution en temps réel de la position de l'effecteur du robot

En actionnant le robot pour une opération de type pick-and-place, la position de l'effecteur du robot effectuée en temps réel est donnée par la courbe 6.41.

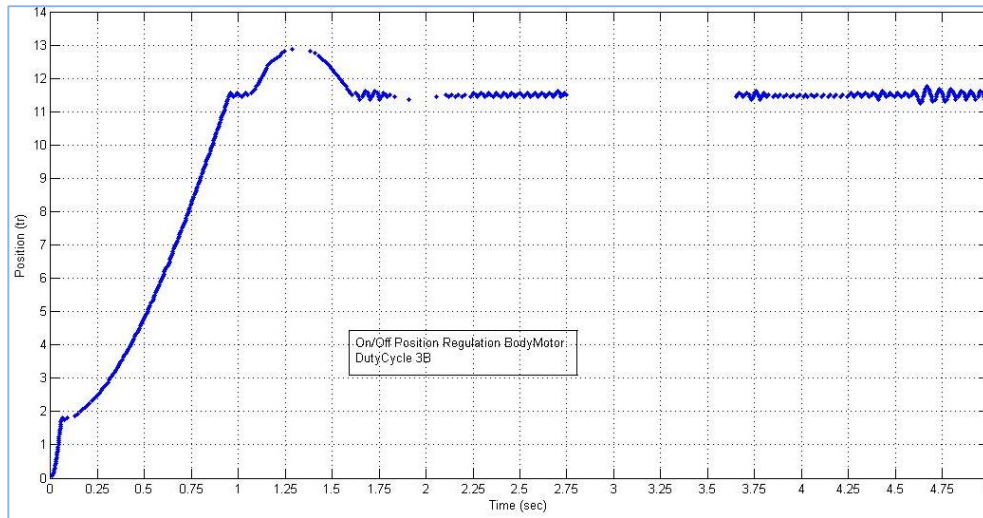


Figure 6.41. Visualisation en temps réel de la position du préhenseur pour un rapport cyclique de 59%

Les coupures sur la courbe de la figure 6.34 sont dues au jeu causé par l'instabilité du socle à cause des roulements sur lesquels l'arbre du moteur tourne.

En appliquant une régulation proportionnelle, nous constatons une amélioration (figure 6.42).

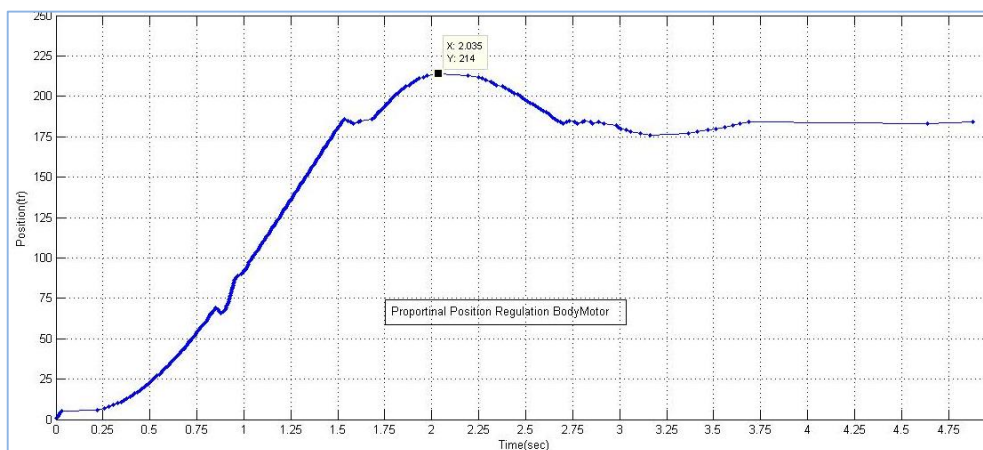


Figure 6.42. Visualisation en temps réel de la position du préhenseur pour un rapport cyclique de 59%

6.6.7 Evolution en temps réel de la vitesse du moteur du robot

La courbe de vitesse du moteur de la base montre une instabilité, ceci est dû à l'instabilité du socle, monté sur des roulements et aussi au jeu au niveau du servomoteur du préhenseur. En appliquant une régulation proportionnelle sur ce moteur, ceci améliore légèrement la courbe de vitesse (figure 6.43).

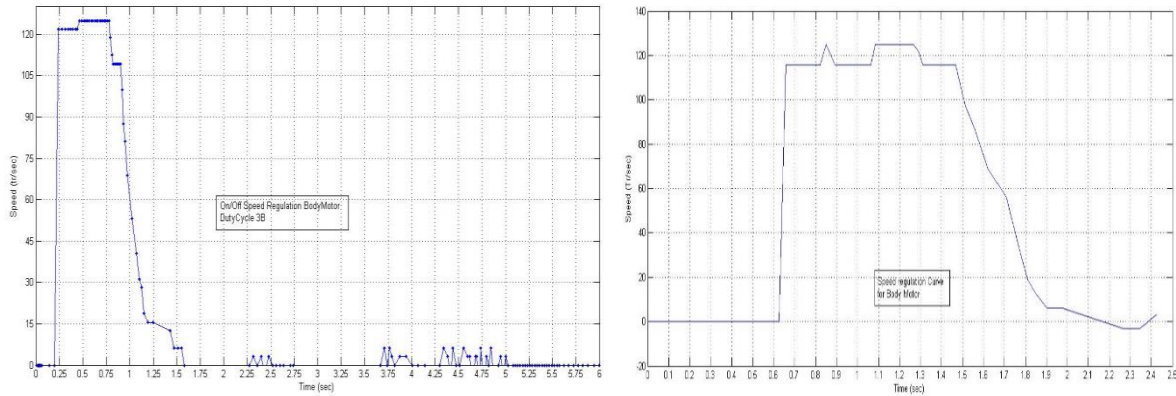


Figure 6.43. Visualisation en temps réel de la vitesse du moteur de la base pour un rapport cyclique de 59%

6.7 Comparaison des résultats pratiques avec la simulation

Les résultats obtenus théoriquement par simulation ont été comparés aux résultats pratiques, il s'agit des courbes de courant et de la forme de position de l'effecteur terminal. En exécutant sa tâche.

6.7.1 Forme du signal de courant dans les moteurs

Les courbes suivantes montrent l'évolution du courant lors du fonctionnement du moteur à vide. La figure 6.44 est celle du démarrage du moteur; la figure 6.45 montre le courant circulant dans le moteur lors du démarrage, la décroissance puis l'arrêt du moteur. En charge, le courant des moteurs ne dépasse pas 2A et celui des servomoteurs atteint 1.5A.

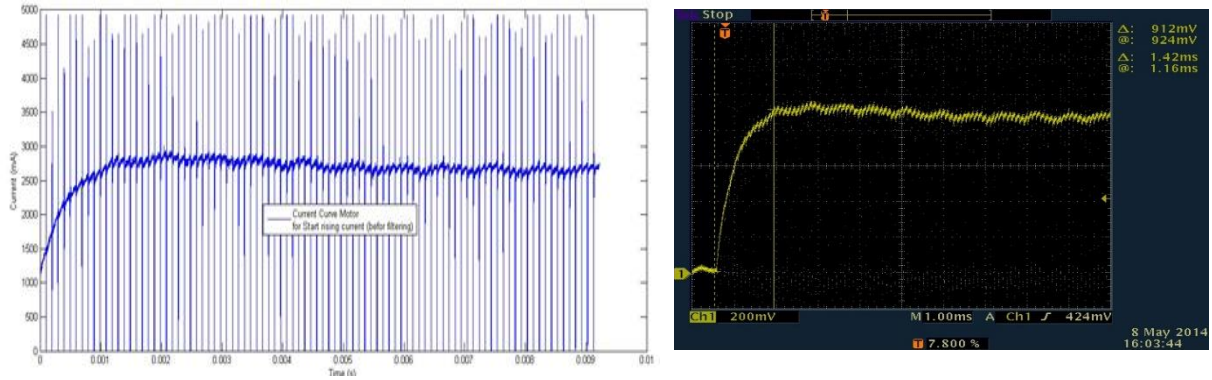


Figure 6.44. Visualisation du courant au démarrage du moteur. Courbe déterminée par Matlab, à gauche; Courbe affichée par l'oscilloscope, à droite

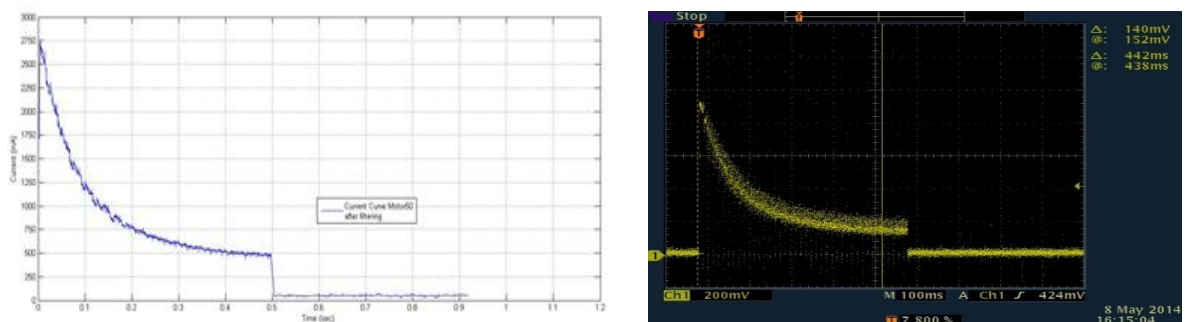


Figure 6.45. Visualisation du courant au démarrage, décroissance et arrêt du moteur. Courbe déterminée par Matlab, à gauche; Courbe affichée par l'oscilloscope, à droite

6.7.2 Forme du signal de la position de l'effecteur

La simulation peut se faire en introduisant un couple aux motorisations, pour voir l'évolution de la position et/ou de la vitesse sans et avec régulation de position (figure 6.46).

Les tests de la simulation prouvent que le schéma proposé fournit des résultats satisfaisants dans le suivi de la position dans un espace sans contrainte, et également dans le suivi de la force pour diverses situations sans connaître la rigidité de l'environnement.

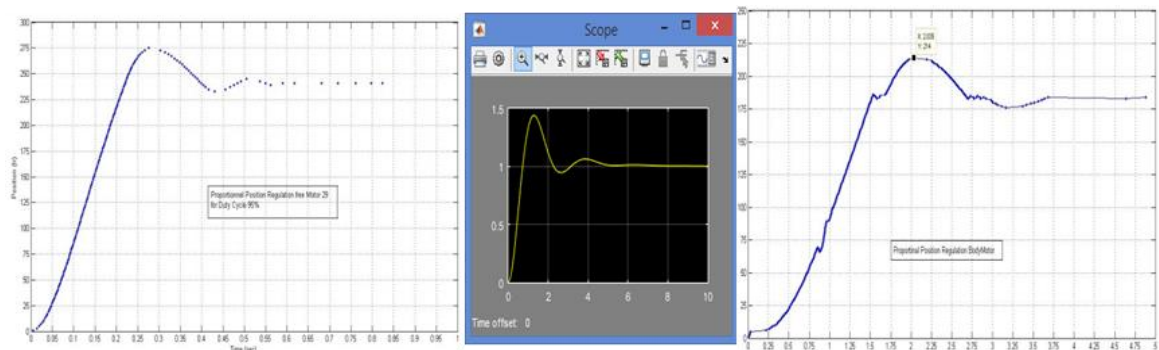


Figure 6.46. Evolution de la position du moteur actionnant la base du robot
 a) moteur à vide b) moteur en simulation c) moteur monté sur le robot

A partir des résultats de simulation réalisés sur SimMechanics, on constate une certaine similitude entre les courbes de position du moteur, en comparaison avec des essais réels sur le robot manipulateur..

Mais, pour pouvoir arriver à une conclusion finale, il est nécessaire d'effectuer plusieurs essais de modélisation testant le fonctionnement de l'ensemble des éléments du robot manipulateur.

6.8 Séquences vidéo de fonctionnement du robot manipulateur (réalisation d'une tâche de télé-opération type pick-and-place)

Un ensemble de tests a été réalisé sur le robot manipulateur. La commande utilise une carte embarquée à base du pic18F452 programmé en assembleur, connectée au PC via la liaison série RS232 à grande vitesse (115200 baudrate²⁹). Cette carte permet la réception des commandes via le PC et le contrôle et commande des moteurs ainsi que la gestion des informations des capteurs.

La commande du bras s'effectue au niveau du PC en mode télé-opération ou en mode pick-and-place en utilisant Matlab. L'écriture du programme de commande en Matlab est rendue facile en spécifiant pratiquement que la destination de la commande (moteur à opérer) et la position souhaitée. Ceci est possible grâce à l'implantation au niveau du microcontrôleur d'une routine qui permet d'envoyer les données nécessaires (temps, position,...etc.) et de recevoir et interpréter les commandes pour pouvoir les exécuter.

Pour la précision, nous avons implanté une régulation proportionnelle, et pour réduire davantage l'erreur, on compte rajouter une action intégrale.

Les figures ci-dessous représentent le fonctionnement du robot en mode pick-and-place.

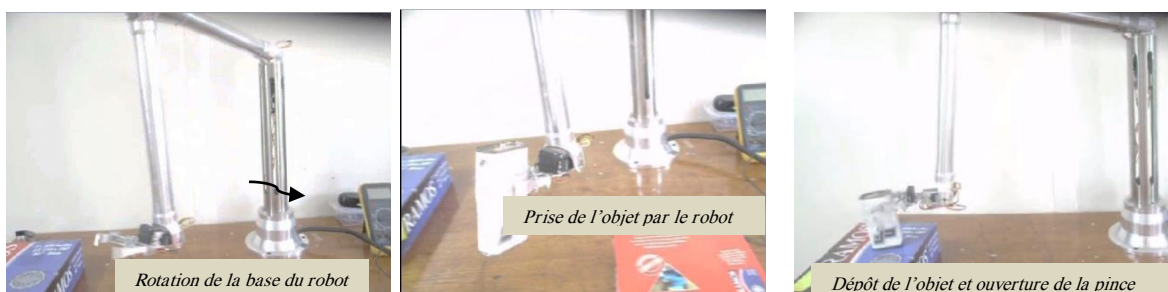


Figure 6.47. Réalisation par le robot d'une tâche pick-and-place

²⁹ Le baud rate définit la vitesse de transmission dans les liaisons séries et correspond au nombre de bits transmis par seconde (débits en bits/s ou Bauds).

6.9 Génération de trajectoire en ligne

Le problème de la planification et de la génération de trajectoires est un sujet de recherche qui dure depuis plusieurs années à cause de sa complexité. Pour comprendre le problème de planification de trajectoires, il est important de bien saisir le sens de certains concepts [107].

Le problème de la génération de trajectoire est important dans de nombreux domaines comme les machines de production industrielle et les robots. L'ouvrage de Biagiotti et Melchiorri présente une synthèse des méthodes de génération de trajectoire pour le contexte industriel [108].

Plusieurs algorithmes sont utilisés, certains sont basés sur la boîte à outils de Matlab et d'autres utilisent les fonctions dynamiques...etc. [109,110].

Kröger, dans ses ouvrages [111,112], présente un état de l'art très bien détaillé sur les différentes méthodes de génération de trajectoire en-ligne et hors-ligne. Il a montré dans un article en 2012, l'apport de la génération de trajectoire en ligne pour le contrôle visuel, en particulier pour réagir à des événements imprévus et donne les éléments essentiels sur les trajectoires en robotique [113].

Khalil et Dombre présentent également différentes méthodes de génération de trajectoire pour les robots manipulateurs [114, 115].

Katzschmann, montre qu'il est possible de prendre en compte la dynamique du robot et de la tâche pendant la génération temps réel de la trajectoire [116].

6.9.1 Effectuer une trajectoire de Robot

La génération de trajectoire pour des bras manipulateurs est largement traitée dans la littérature [117-122]. Générer une trajectoire spécifiée relative au déplacement du robot revient à constituer une série de poses très rapprochées pour l'effecteur terminal du manipulateur.

La position exacte, la trajectoire et le chemin d'un robot, le déplacement d'un point initial à un point final, sont des problèmes de la manipulation des robots. La planification repose sur des aspects plus mathématiques à savoir la modélisation comme le modèle géométrique, et le modèle cinématique. Le modèle dynamique repose beaucoup plus sur la physique de robot et la dynamique de la structure mécanique et aussi les moteurs.

Un générateur de trajectoire est conçu de manière à traduire les intentions de l'opérateur humain en trajectoires idéales que le robot doit suivre. L'organe terminal du robot parcourt les chemins nécessaires pour effectuer une tâche donnée. Par ailleurs, la trajectoire à suivre doit être générée en ligne par un traitement approprié en temps réel des informations qui lui sont fournies.

Le générateur de trajectoire a pour fonction de calculer l'évolution des positions des articulations ou de la situation du robot en fonction du temps. Il joue un rôle très important dans le schéma de commande car la qualité du mouvement produit dépend directement de la continuité de la trajectoire produite et de ses dérivées. Cette qualité a un impact sur la précision de suivi du chemin. Un chemin est une séquence de points définis soit dans l'espace des tâches (espace opérationnel) (afin de situer l'organe terminal), soit dans l'espace articulaire (espace des configurations) du robot (afin d'indiquer les valeurs des paramètres des articulations).

Différents moyens existent pour effectuer une planification de trajectoire selon que le robot soit fixe ou mobile et s'il y a ou pas des obstacles sur son chemin.

6.9.1.1 Utilisation de la boîte à outils de Matlab

La boîte à outils Matlab fournit de nombreuses fonctions nécessaires pour l'étude et la simulation des robots à bras manipulateurs, telles que la cinématique, la dynamique et la génération de trajectoires. La simulation 3D du robot est basée sur le logiciel Matlab ainsi que l'extension Robotics toolbox de Peter Corke [123,124].

La première étape de la modélisation d'un robot consiste à placer les repères de chaque articulation du robot. Pour ce faire, il faut placer les repères de la manière suivante :

- L'axe Z doit être placé de sorte à être l'axe de rotation/translation de l'articulation selon sa nature rotative ou prismatique.
- L'axe X doit être placé de sorte à faciliter le passage d'un repère à l'autre, car pour ce faire on ne peut que réaliser des rotations et translations en fonction des axes X et Z.

Voici ci-contre une représentation 3D du robot avec le placement des différents repères des articulations :

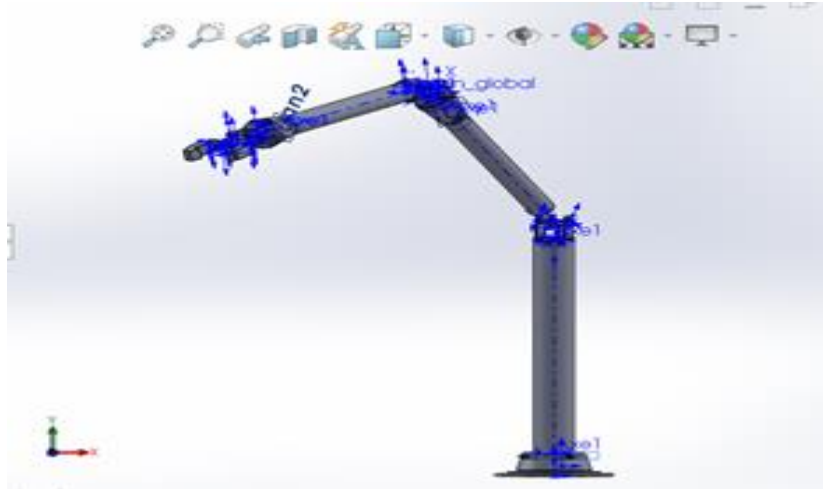


Figure 6.48. Placement des repères des articulations du Robot

Une trajectoire est définie par une équation polynomiale. Plus le degré n est haut, plus la précision est élevée et donc plus le contrôle du système est robuste.

$$q(t) = a_0 + a_1t + a_2t^2 + \dots + a_nt^n$$

Sa génération manuellement est pénible mais avec le toolbox de Peter Corke, l'opération devient simple.

Pour le cas de $n=3$ $q(t) = a_0 + a_1t + a_2t^2 + a_3t^3$

L'équation de la vitesse: $\dot{q}(t) = a_1 + 2a_2t + 3a_3t^2$

L'équation de l'accélération: $\ddot{q}(t) = 2a_2 + 6a_3t$

La Boîte à outils a toujours fourni de nombreuses fonctions utiles. Elle contient des fonctions et des classes pour représenter l'orientation et la pose en 2D et 3D. Elle fournit aussi des fonctions de manipulation et de conversion entre types de données tels que vecteurs, transformations homogènes et autres unités nécessaires pour représenter la position et l'orientation en 3 dimensions.

La boîte à outils utilise une méthode très générale de représentation de la cinématique et de la dynamique des manipulateurs de liaison série sous forme d'objets MATLAB, des objets de robot qui peuvent être créés par l'utilisateur pour tout manipulateur de liaison série et plusieurs exemples sont fournis pour des robots bien connus de Kinova ainsi que d'autres robots classiques.

La fonction "rtbdemo" permet d'explorer la boîte à outils de Matlab et affiche un panneau de contrôle (figure 6.49).



Figure 6.49. Interface de contrôle de la boîte à outils de Matlab

Chaque bouton correspond à une commande. La commande "Create a movement" permet d'exécuter un programme qui effectue le calcul d'un certains nombres de fonctions, qui permettent le mouvement du robot (figure 6.50).

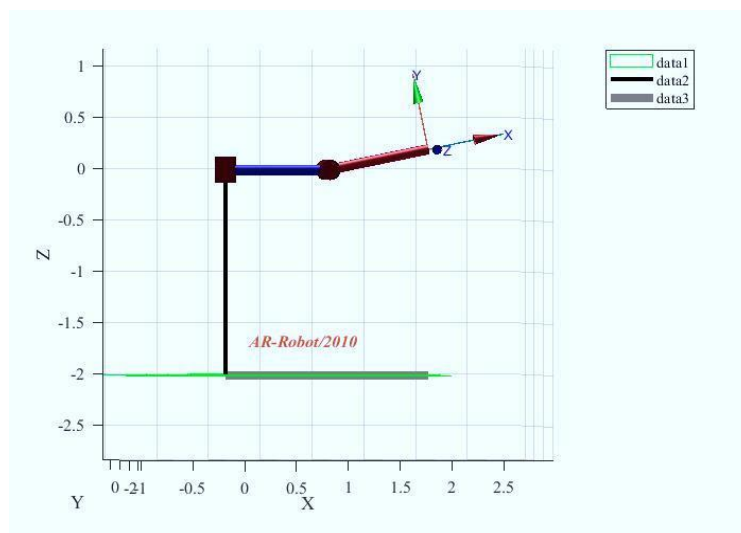


Figure 6.50. Présentation du mouvement du robot en 2D

Une autre fonction basée également sur les outils de Matlab est la fonction *WaypointTrajectory*. Cette fonction n'est disponible qu'à partir de la version Matlab 2019.

6.9.1.2 WaypointTrajectory de Matlab

La génération de trajectoire et la définition de points de cheminement représentent de nombreuses applications robotiques telles que les opérations de type pick-and-place, le calcul de trajectoires à partir d'accélération spatiales et de profils de vitesse, ou même la reproduction d'observations externes d'images clés à l'aide de caméras et de la vision par ordinateur. La figure 6.51 reproduit la trajectoire effectuée par l'effecteur du robot.

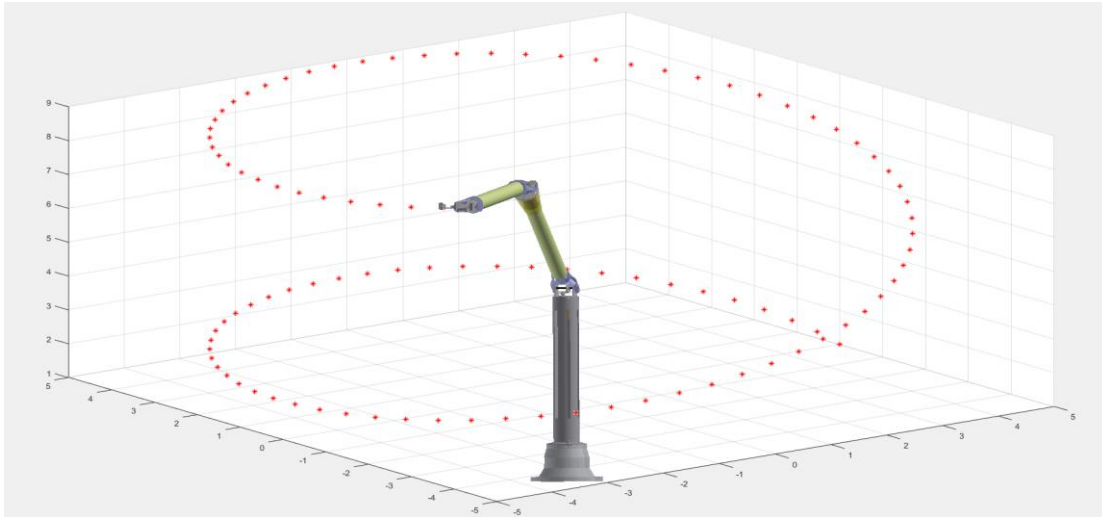


Figure 6.51. Trajectoire effectuée par l'effecteur dans la réalisation par le robot d'une tâche pick-and-place

Ceci a été réalisé en se basant sur les fonctions de Matlab tels que "WaypointTrajectory". Le waypointTrajectory objet System™ génère des trajectoires à l'aide de points de cheminement spécifiés. Lorsque l'objet Système est créé, l'heure d'arrivée, la vitesse et l'orientation à chaque point de cheminement peuvent être spécifiés. Pour générer une trajectoire à partir de points de cheminement il suffit de:

1. Créez le waypointTrajectoryobjet et définissez ses propriétés.
2. Appelez l'objet comme s'il s'agissait d'une fonction.

Qui peut être traduit par les syntaxes:

```
trajectory = waypointTrajectory  
trajectory = waypointTrajectory(Waypoints,TimeOfArrival)  
trajectory = waypointTrajectory(Waypoints,TimeOfArrival,Name,Value)
```

Une fois qu'une trajectoire est générée, le bloc 'Inverse Kinematics' de Matlab est utilisé pour la traduire en une trajectoire articulaire, qui peut ensuite être utilisée pour simuler la dynamique du manipulateur et du contrôleur.

Une autre possibilité pour générer la trajectoire du robot utilise la vidéo du mouvement qu'il effectue, c'est le logiciel de poursuite 'Tracker' dont tous les tracés sont générés en ligne.

6.9.1.3 Utilisation du logiciel de poursuite "Tracker"

L'étude de la cinématique du solide dans le plan peut être réalisée par une vraie démarche expérimentale en associant l'enregistrement vidéo et un logiciel d'analyse spécialisé. Avec la généralisation de l'utilisation d'images numériques, l'analyse du mouvement dans les vidéos s'est révélée être un outil indispensable pour des applications aussi diverses que la vidéo surveillance, la compression vidéo, l'imagerie médicale, la robotique, l'interaction homme-machine. En effet, les zones de mouvement d'une séquence d'images correspondent souvent à des événements sur lesquels un système de vision doit se focaliser [125].

Un logiciel remarquable par ses possibilités étendues et sa simplicité de mise en œuvre est le logiciel de poursuite "Tracker". Sa dénomination est liée à un de ses points forts, à savoir sa capacité, avec une grande efficacité, à faire la poursuite automatique de divers points mobiles d'une vidéo, en enregistrant les coordonnées des points dans le plan avec un repère et avec une échelle prédéfinis.

Tracker est un logiciel d'analyse vidéo et de modélisation physique basé sur l'Open Source Physics. Écrit en Java, il est un logiciel multiplateforme. Ce logiciel permet d'obtenir rapidement et facilement des graphiques de la position, de la vitesse et de l'accélération en fonction du temps à partir d'une vidéo (figure 6.52).

Tracker est un logiciel d'analyse vidéo et de modélisation physique basé sur l'Open Source Physics. Écrit en Java, il est un logiciel multiplateforme.

Tracker permet d'obtenir rapidement et facilement des graphiques de la position, de la vitesse et de l'accélération en fonction du temps à partir d'une vidéo. Après avoir choisi le chemin du robot sur lequel on veut générer la trajectoire, un échantillonnage est effectué sur les différents points traversés par le robot, ainsi une trajectoire est tracée en ligne et les différentes valeurs (position, vitesse, accélération, angle, etc.) sont générées dans un tableau.

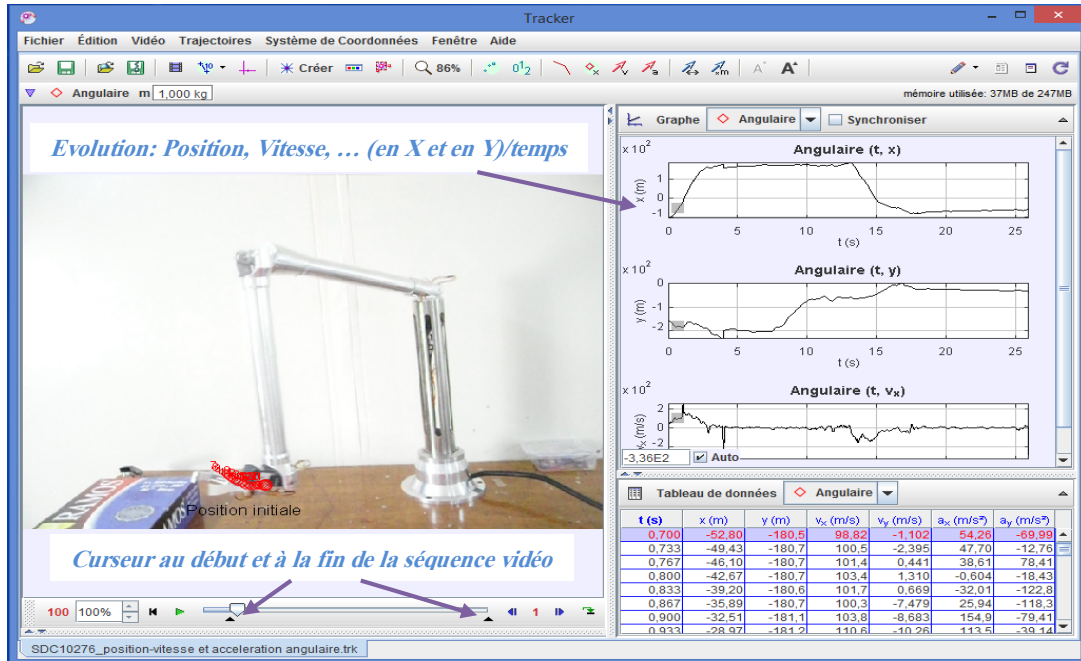


Figure 6.52. Génération de mouvement en ligne du robot manipulateur

Sur la figure ci-dessus, on constate trois fenêtres, l'une est réservée pour la réception de la vidéo du robot, une pour les graphiques déterminant la trajectoire du robot. Le nombre de graphiques, n'excédant pas 3, est déterminé par l'utilisateur suivant ce qu'il veut afficher comme paramètres (position, vitesse, etc.). La tâche de déplacement d'un robot est spécifiée en définissant un chemin que le robot doit suivre.

1. Traitement des données

La modélisation vidéo Tracker est un moyen puissant de combiner des vidéos avec la modélisation informatique.

En plus de ses possibilités de numérisation des trajectoires, Tracker dispose de nombreuses fonctions d'analyse cinématique.

2. Mise en œuvre du Tracker

Positionner le curseur au début de la séquence vidéo en déterminant un point d'arrivée; Les points de cheminement se font détecter automatiquement tout le long du mouvement du robot. Un tableau ainsi que le tracé de la composante selon X de position $X(t,x)$ sont générés en ligne. Le tableau contient en plus toutes les valeurs correspondant à la composante selon Y de la position $Y(t,y)$, les composantes selon X et Y de la vitesse et de l'accélération, la position, la vitesse et l'accélération angulaire en plus d'autres paramètres.

Etapas de la démarche expérimentale :

- ❖ Charger la vidéo du robot;
- ❖ Marquer l'objet en mouvement (la pince) ;
- ❖ Délimiter la zone à exploiter (séquence vidéo);
- ❖ Positionner le curseur au début de la séquence vidéo; Un tableau est généré contenant tous les points parcourus (temps t, vitesse, accélération, θ , ω , etc.);
- ❖ Tracer les graphes souhaités et effectuer l'analyse;
- ❖ Tracer les trajectoires des points réels ainsi que certains vecteurs (vitesse, accélération).

3. Exemples d'exploitation des résultats réels

Dans ce qui suit, un ensemble de figures caractéristiques de la trajectoire qu'effectue le robot en temps réel.

3.1 Evolution suivant X

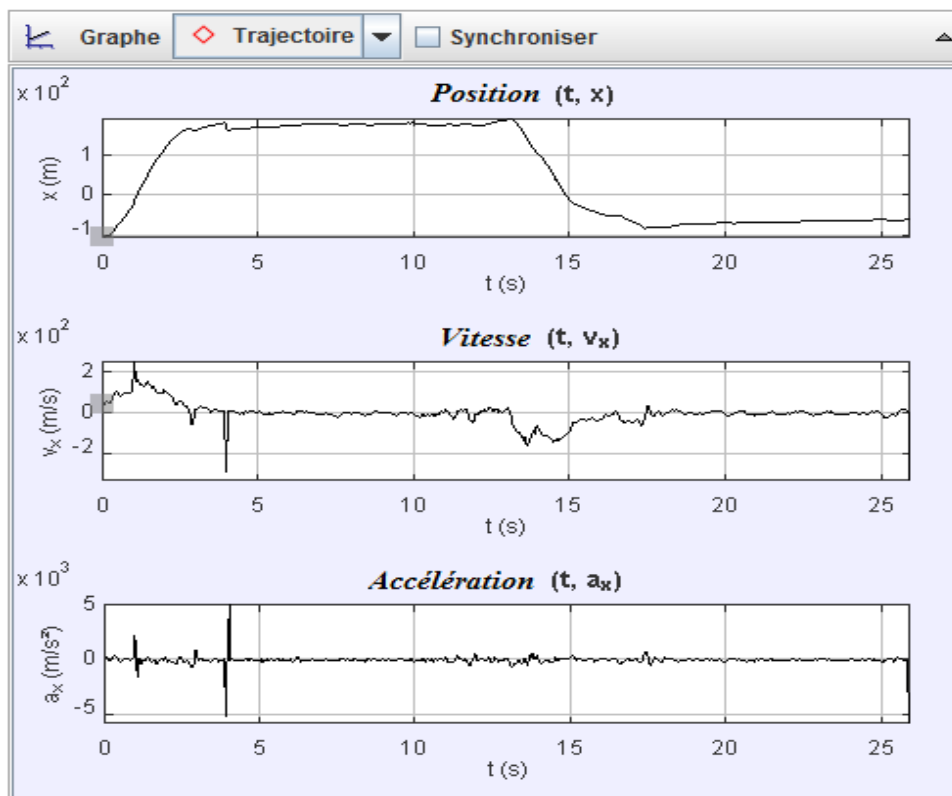


Figure 6.53. Représentation suivant X de la position du robot, sa vitesse et son accélération

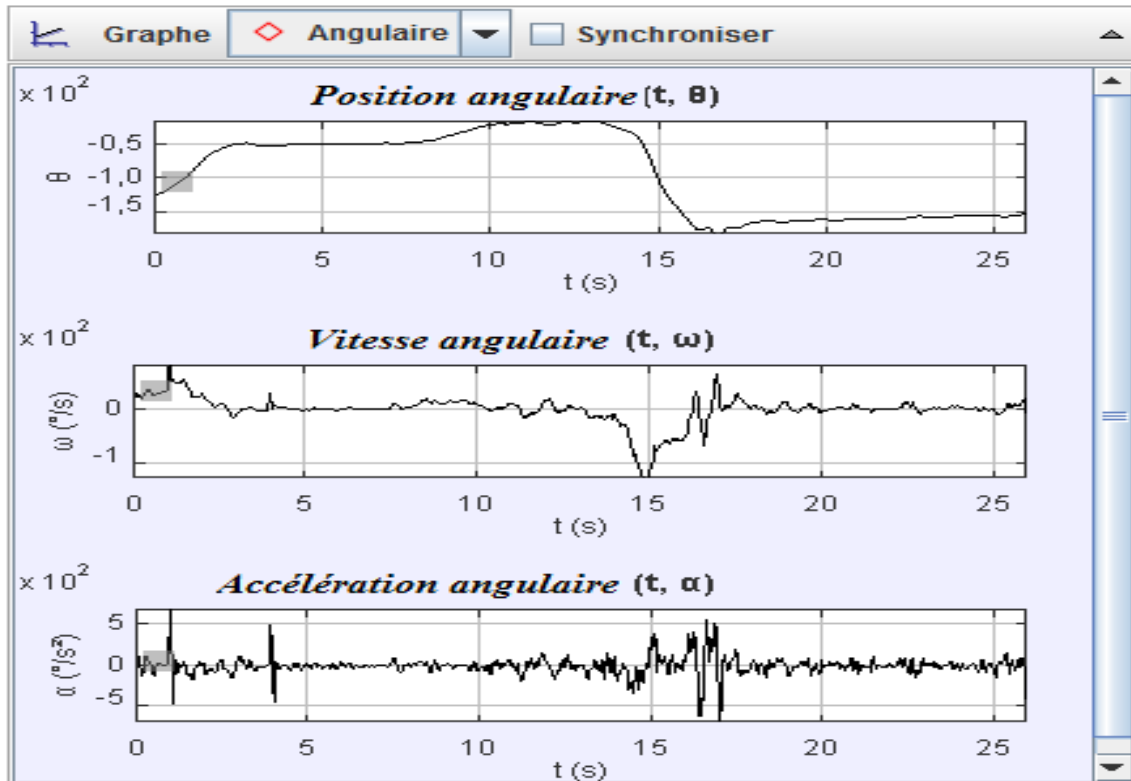


Figure 6.54. Représentation suivant X de la position, vitesse et accélération angulaire

3.2 Evolution suivant Y

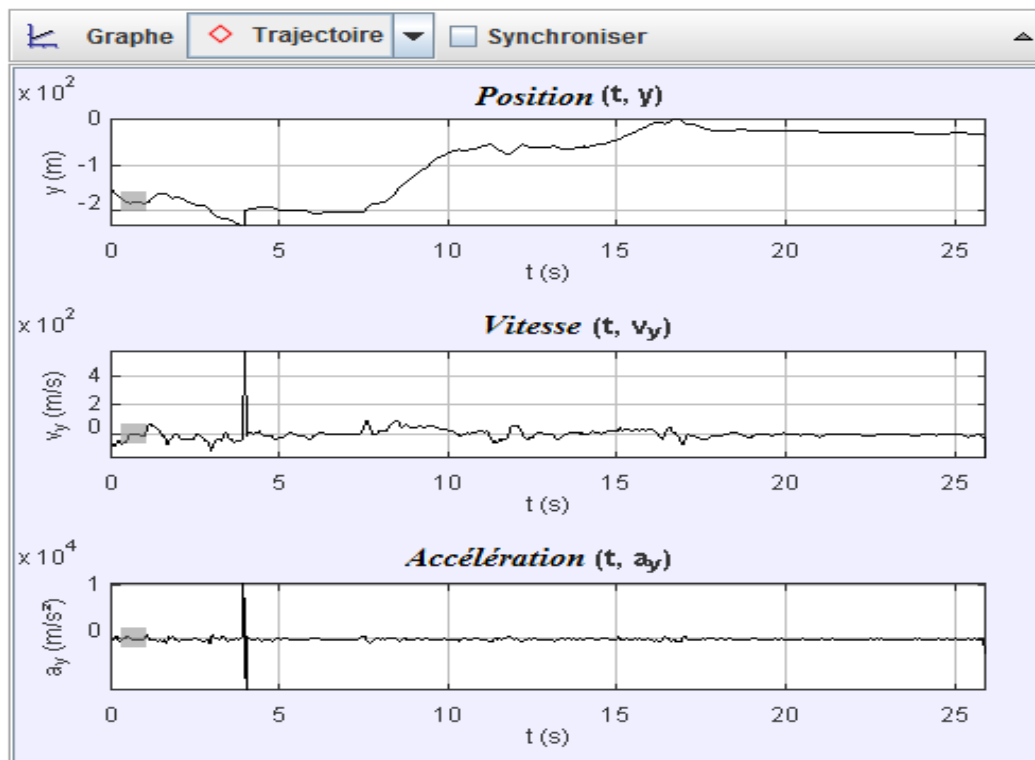


Figure 6.55. Représentation suivant Y de la position du robot, sa vitesse et son accélération

Une fois qu'une trajectoire est générée, le bloc Inverse Kinematics de Matlab est utilisé pour la traduire en une trajectoire articulaire, qui peut ensuite être utilisée pour simuler la dynamique du manipulateur et du contrôleur.

6.10 Conclusion

Notre travail avait pour but de pouvoir mettre au point un robot manipulateur pour une utilisation dans les lieux inaccessible par l'humain. Une conception et réalisation complète (mécanique et commande) d'un robot manipulateur a été accomplie. L'avantage de la réalisation du robot est qu'il nous assure une grande flexibilité et versatilité nécessaire dans l'établissement et le développement d'un environnement de réalité augmentée.

Des tests du type pick-and-place ont été opérés sur ce robot avec succès hormis quelques fluctuations dues à une légère instabilité du socle à cause des roulements sur lesquels l'arbre du moteur tourne.

Chapitre 7. Conclusion générale & Perspectives

7.1. Conclusion

Nous avons essayé, tout au long de cette thèse, d'atteindre nos objectifs à savoir l'élaboration et la mise en œuvre de systèmes robotiques opérant dans des environnements hostiles. La recherche développée dans cette thèse a été motivée par le besoin d'effectuer des opérations de manipulation et de maintenance avec plus de performance de la tâche à effectuer et plus de sécurité pour l'opérateur humain.

Dans le but de nous familiariser avec les systèmes robotiques et leur commande, nous avons fait une étude approfondie sur les systèmes de télé-opération de robots manipulateurs. Cette étude a conduit à la mise au point d'une interface graphique facilitant l'interaction homme-robot et permettant le contrôle à distance de robots manipulateurs. Nous avons utilisé pour cela un robot manipulateur expérimental à 3 d.d.l 'SEROR'. Pour sa commande, nous nous sommes basé sur les modèles géométriques inverse et direct, une commande largement suffisante pour les applications de pick-and-place.

Des expériences ont été réalisées sur une distance de 450 km environ. Ces expériences avaient pour but d'évaluer l'interface utilisateur d'une part et d'autre part, l'architecture réseau du système en prenant en compte la distance séparant les deux sites maître et esclave. Plusieurs opérations du type pick-and-place ont été effectuées par le robot à travers le réseau informatique, se servant du protocole TCP/IP. La réussite des opérations réalisées nous pousse à conclure que notre système peut être exécuté depuis n'importe quelle machine connectée à Internet. D'autre part, grâce à l'interface homme-machine, conçue selon un design centré sur l'utilisateur "User-Centered Design", l'opérateur participe et réalise des tâches en télé- opération (saisie d'objets, perçage, télé-chirurgie, etc.). Dans ce contexte, nous avons développé et testé trois modes de contrôle qui sont: le clic de la souris "Mouse-click", le contrôle de l'image "image-based click" et le contrôle gestuel "gesture-based click".

Outre les avantages liés à ce mode de contrôle, nous avons notés certains inconvénients qui nuisent à la stabilité du système et la sécurité de l'opérateur. La plupart des problèmes rencontrés par un opérateur humain placé en télé-opération sont de nature visuelle. Ils sont surtout dus à la limitation de la taille du champ visuel, aux temps temporels trop élevés dans la transmission des images vidéo et à la mauvaise précision du système télérobotique.

Pour pallier à ces problèmes, et après une synthèse faite des différents modes de contrôle de robot : le mode prédictif, le mode de glissement, la réalité virtuelle et la réalité augmentée, la RA s'avère plus adapté au contrôle de processus avec plus d'efficacité par le renforcement de la capacité de l'opérateur humain à percevoir les aspects d'un monde réel en superposant des informations supplémentaires (visuelles ou autres). Néanmoins, sa mise en œuvre a nécessité le développement d'une plateforme pour les expérimentations de ses stratégies de contrôle. Cette plateforme a suscité, en plus de la nécessité d'avoir un robot manipulateur ayant plus de flexibilité et plus de précision, l'introduction à la vision par au contrôle en réalité augmentée à savoir, la modélisation de l'environnement de télé-opération du robot et la calibration de la caméra utilisée pour la vision de l'environnement.

Les premières expériences réalisées en laboratoire ont concerné la mise en pratique de la méthode de vision pour la mesure des grandeurs et de la profondeur d'objets dans l'image de la scène. Les tests pratiques sur la méthode de calibration de la caméra, implémentée sur Matlab, ont montré sa faisabilité dans la réalisation pratique avec des mesures obtenues avec une précision inférieures à un millimètre ce qui met en évidence l'importance et l'utilité de l'exploitation de ces méthodes pour les mesures 2D et 3D.

Dans le cinquième chapitre, nous avons présenté le robot manipulateur à 4 degrés de liberté que nous avons conçu. Sa conception faite sur Autodesk/Inventor, nous a permis d'exporter tous les éléments le constituant vers 'Matlab' pour sa modélisation et sa simulation en se basant sur les fonctionnalités SimMechanics et Simulink de Matlab. Cette étape du travail avait pour but de tester les performances du robot avant de passer à l'étape de réalisation, et ainsi détecter d'éventuels défauts de conception et les modifier.

Dans cette perspective, nous avons développé dans ce chapitre, un environnement de modélisation pour notre système robotique. Le processus s'est basé sur SimMechanics, un outil interactif de modélisation en trois dimensions des systèmes mécaniques multi-corps, dans l'environnement Simulink. L'opération a nécessité d'implanter un module supplémentaire "Add-In" dans Inventor, ce qui a donné naissance à deux actions : 'SimMechanics first génération' et 'SimMechanics seconde génération'. Nous avons utilisé cette dernière pour transférer tous les éléments du robot vers l'environnement Matlab et permettre de simuler son fonctionnement.

En sixième chapitre, nous avons présenté le robot manipulateur à 4 degrés réalisé ainsi que sa carte de contrôle à base du microcontrôleur 'PIC18F452'. Pour les articulations du robot, nous avons utilisé des motoréducteurs à encodeurs, pour permettre le contrôle de la

position et de la vitesse à tout instant.

Le robot a été commandé et contrôlé avec succès; et a exécuté différentes opérations de manipulation d'objets. Nous avons utilisé Matlab pour la programmation au niveau du PC, en le reliant à la carte de contrôle-commande via la liaison série RS232.

Pour les expériences menées, nous avons tracé l'évolution de la position du robot dans la pratique, que nous avons comparé à des simulations. Nous avons noté une certaine similitude des courbes, excepté de légères fluctuations en pratique et principalement au niveau de la pince du robot qui utilise deux servomoteurs et à un certain jeu sur le moteur de la base monté sur des roulements. Le premier problème est peut-être dû à un défaut dans le montage de la pince et le second est dû à l'instabilité du socle.

7.2. Perspectives

Nous voudrions dans le proche avenir, améliorer davantage notre système robotique en vue d'une utilisation en milieu radioactive 'Réacteur nucléaire'. Le robot doit répondre à certaines exigences de performance, notamment en termes de répétabilité, de stabilité et de précision. Le système robotique doit être capable de positionner et d'aligner avec précision les pièces 'Barre de contrôle par exemple'. Pour cela, nous avons choisi d'utiliser un moteur à courant continu (MMP S17-400C-24V GP42-169) (figure 7.1), alimenté par une tension de 24V, ayant un fort courant (5.2A le courant continu nominal) et un important couple (148 IN-lbs le couple continu nominal).

Bibliographie

1. **Baraglia, J., Cakmak, M., Nagai, Y., Rao, R. et Asada, M.** (2016), "Initiative in Robot Assistance During Collaborative Task Execution". Communication presented at The 11th ACM/IEEE International Conference on Human Robot Interaction, Nouvelle-Zélande (p. 67-74).
2. **Puran Singh, Anil Kumar, Mahesh Vashisth.** (2013), "Design of a Robotic Arm with Gripper & End Effector for Spot Welding", *Universal Journal of Mechanical Engineering*, 1(3): 92-97.
3. **John J. Craig.** (2005), "Introduction to Robotics Mechanics and control", Third edition. Pearson Prentice Hall.
4. **A. Zaatri.** "A Low Cost Experimental Telerobotic Systems", Mechanical Department Laboratory, University Mentouri of Constantine, Algeria.
5. **Mohamed Emharraf, Mohammed Saber, Mohammed Rahmoun and Mostafa Azizi.** (2016), "Control Architecture for Mobile Robot Teleoperation", Springer International Publishing Switzerland 2016 A. El Oualkadi et al. (eds.), *Proceedings of the Mediterranean Conference on Information & Communication Technologies 2015*, Lecture Notes in Electrical Engineering 381, DOI 10.1007/978-3-319-30298-0_78 .
6. **Nader Cheaib, Mouna Essabbah, Christophe Domingues, Samir Otmame.** (2012), "Software architecture for interactive robot teleoperation", *Symposium on Engineering Interactive Computing Systems (ACM EICS 2012)*, Jun 2012, Copenhagen, Denmark.
7. **J. Cui, S. Tosunoglu, R. Roberts, C. Moore and D. W. Repperger.** (2003), "A Review of Teleoperation System", *Proceedings of the Florida Conference on Recent Advances in Robotics*, FCRAR, Boca Raton, Florida.
8. **Steven Vozar and Dawn M Tilbury.** (2012), "Improving UGV teleoperation performance using novel visualization techniques and manual interfaces". *Proceedings of SPIE*, 8387:838716.
9. **Kamrani, A. Ramazani and F. Monteiro.** (2006), "Teleoperation via Internet with Time-Varying Delay", *Proceedings of the 13th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, NICE, pp. 736-739.
10. **Masmoudi Mostefa.** (2017), "Prédiction en temps réel de la qualité de service pour la téléopération à travers le réseau internet", Thèse de doctorat option Systèmes intelligents et robotique, Faculté Génie électrique, USTO M B ORAN.
11. **Ehsan Kamrani.** (2012), "Real-Time Internet-Based Teleoperation", *Intelligent Control and Automation*, 2012, 3, 356-375. DOI:<http://dx.doi.org/10.4236/ica.2012.34041>, Published Online November 2012.
12. **Mahdi Ghanbari and Ehsan Kamrani.** (2009), "Optimal Control of Robot through Internet", *World Appl. Sci. J.*, 6 (5): 702-710. ISSN 1818-4952.
13. **F. Z. Zendaoui, M.O. Mahmoudi, A. Zaatri.** (2018), "Development and Experimentation of an Articulated Mechanical System using Internet", ISSN 1392–1207. *MECHANIKA*. Volume 24(4): 462–466,

- <http://dx.doi.org/10.5755/j01.mech.24.4.19386>.
14. **M. A. Mashagbeh and M. B. Khamesee.** (2015), "Virtual performance evaluation of an industrial SCARA robot prior to real-world task", *Microsyst. Technol.*, vol. 21, no. 12, pp.2605-2609, 2015.
 15. **Costas Tzafestas.** (2006), "Virtual and Mixed Reality in Telerobotics: A Survey", *Industrial Robotics: Programming, Simulation and Application*, ISBN 3-86611-286-6, pp. 702, ARS/plV, Germany, December, Edited by: Low Kin Huat.
 16. **Ahmed R. J. Almusawi, L. Canan Dülger, Sadettin Kapucu.** (2016), "Robotic Arm Dynamic and Simulation with Virtual Reality Model (VRM)", Malta. IEEE.
 17. **D. Zhijiang, J. Zhiheng, K. Minxiu.** (2005), "Virtual Reality-based Telesurgery via Teleprogramming Scheme Combined with Semi-autonomous Control", *engineering in Medicine and Biology Society, IEEE-EMBS*, Page: 2153-2156.
 18. **Jiang Zainan, Liu Hong, Wang Jie and Huang Jianbin.** (2009), "Virtual Reality-based teleoperation with robustness against modeling Errors", *Chinese Journal of Aeronautics* 22 (2009) 325-333, Elsevier.
 19. **Eman R. Elbasiouny, T. Medhat, Amany Sarhan, Tarek E. Eltobely.** (2013), "Stepping into Augmented Reality", *International Journal of Networked Computing and Advanced Information Management (IJNCM)* doi: 10.4156/IJNCM.vol1.issue1.2 2013.
 20. **Jean-Marc Cieutat.** (2013), "Quelques applications de la réalité augmentée: Nouveaux modes de traitement de l'information et de la communication Effets sur la perception, la cognition et l'action", *Mémoire pour l'obtention du titre d'Habilitation à Diriger des Recherches en Informatique*, Université Paul Sabatier de Toulouse III.
 21. **H. C. Fang · S. K. Ong · A. Y. C. Nee.** (2014), "A novel augmented reality-based interface for robot path planning", *Int J Interact Des Manuf* 8:33–42, DOI 10.1007/s12008-013-0191-2.
 22. **H. C. Fang & S. K. Ong & A. Y. C. Nee.** (2013), "Orientation planning of robot end-effector using augmented reality", *Int J Adv Manuf Technol* (2013) 67:2033-2049 DOI 10.1007/s00170-012-4629.
 23. **J.W.S. Chong, S.K. Ong, A.Y.C. Nee, K. Youcef-Youmi.** (2009), "Robot programming using augmented reality: An interactive method for planning collision-free paths", *Robotics and Computer-Integrated Manufacturing*, 689– 701.
 24. **Oskiper, T., Chiu, H.-P., Zhu, Z., Samaresekera, S., and Kumar, R.** (2011), "Stable vision-aided navigation for large-area augmented reality", *Virtual Reality Conference (VR)*, IEEE, pages 63-70.
 25. **Steven Vozar and Dawn M. Tilbury.** (2012), "Augmented reality user interface for mobile robots with manipulator arms: Development, testing, and qualitative analysis", In *Proceedings of the Computers and Information in Engineering Conference (CIE)*.
 26. **H.C. Fang, S.K. Ong, A.Y.C. Nee,** (2012) "Robot Path and End-Effector Orientation Planning Using Augmented Reality", *45th CIRP Conference on Manufacturing Systems*; Published by Elsevier B.V, doi:10.1016/j.procir.2012.07.034.
 27. **Van Krevelen and R. Poelman.** (2010), "A Survey of Augmented Reality Technologies, Applications and Limitations", *D.W.F. The International Journal of Virtual Reality*, 9(2):1-20
 28. **Viviana Chimienti, Salvatore Iliano, Michele Dassisti, Gino Dini, Franco Failli.**

- (2014), "Guidelines for Implementing Augmented Reality Procedures in Assisting Assembly Operations", HAL Id: hal-01055687 <https://hal.inria.fr/hal-01055687> Submitted on 13 Aug 2014.
29. **Huan Hu, Xin Gao, Hanxu Sun, Qingxuan Jia, Yanheng Zhang.** (2013), "Design and Implementation of the Teleoperation Platform Based on Augmented Reality", Proc. 12th IEEE Int. Conf. on Cognitive Informatics & Cognitive Computing ©2013 IEEE.
 30. **Curtis W. Nielsen** (2006), "Using augmented virtuality to improve human-robot interactions", PhD thesis, Brigham Young University.
 31. **Siam Charoenseang and Tarinee Tonggoed.** (2011), "Human–Robot Collaboration with Augmented Reality", C. Stephanidis (Ed.): Posters, Part II, HCII 2011, CCIS 174, pp. 93–97, 2011.© Springer-Verlag Berlin Heidelberg 2011.
 32. **Scott A. Green, J. Geoffrey Chase, XiaoQi Chen, and Mark Billingham.** (2010), "Evaluating the augmented reality human-robot collaboration system", Int. J. Intell. Syst. Technol. Appl., 8(1/2/3/4):130-143
 33. **Vit Ondrousek.** (2016), "Control of the End Effector Position Based on Motion Capture", Springer International Publishing Switzerland.
 34. **F. Z. Zendaoui, M.O. Mahmoudi, A. Zaatri.** (2018), "Development and Experimentation of an Articulated Mechanical System using Internet", ISSN 1392–1207.MECHANIKA. Volume 24(4): 462–466, <http://dx.doi.org/10.5755/j01.mech.24.4.19386>.
 35. **Fatim Zohra Zendaoui, Abdelouahab Zaatri, M. Oulhadj. Mahmoudi and Rachid. Ouiguini.** (2007), "Developpement of a teleopetation System for Manipulation Processes via Internet", the third International Conference on Advances in Mechanical engineering and Machanics, (ICAMEM2006), December-17-19, 2006, Hammamet, Tunisia. Tunis.
 36. **Masmoudi Mostefa, L. Kaddour El Boudadi, J. Vareille.** (2017) "Safe and efficient mobile robot teleoperation via a network with communication delay", Int J Interact Des Manuf. DOI 10.1007/s12008-017-0370-7, springer 2017.
 37. **Lan Zhang, Xinsong Yang, Chen Xu, Jianwen Feng.** (2017), "Exponential synchronization of complex-valued complex networks with time-varying delays and stochastic perturbations via time-delayed impulsive control", Applied Mathematics and Computation, Elsevier 306 22–30.
 38. **Teréz A. Varkonyi, Imre J Rudas, Péter Pausits, Tamas Haidegger.** (2014,) "Survey on the control of time delay teleoperation systems", IEEE 18th International Conference on Intelligent Engineering Systems • July 3-5, 2014, Tihany, Hungary, 978-1-4799-4615-0/14/\$31.00 ©2014 IEEE.
 39. **Jiayi Zhu, Xiaochuan He and Wail Gueaieb** (2011), "Trends in the Control Schemes for Bilateral Teleoperation with Time Delay", University of Ottawa, Canada.
 40. **Le petit v. and Berger M. O.** (2000), "Handling Occlusions in Augmented Reality Systems", A Semi-Automatic Method, dans P RESS I. C., editor, Isar, p. 137–146, Los Alamitos, Calif.
 41. **M. Sauer, F. Leutert, and K. Schilling.** (2010), "Occlusion handling in augmented reality user interfaces for robotic systems", 41st International Symposium on Robotics, ISR 2010, and 6th Robotik.
 42. **Rodriguez-Seda, E.** (2007), "Comparative experimental study of control schemes for

- bilateral teleoperation systems", Master's thesis, Univ. Illinois.
43. **Paolo Arcara, Claudio Melchiorri.** (2002), "Control schemes for teleoperation with time delay: A comparative study", Dipartimento di Elettronica Informatica e Sistemistica (DEIS), University of Bologna, Via Risorgimento, 2-40136 Bologna, Italy.
 44. **Rodriguez-Seda, E., Lee, D. Spong, M.** (2009), "Experimental comparison study of control architectures for bilateral teleoperators", IEEE Transactions on Robotics 25(6), 1304-1318.
 45. **Slotine, J.J., Sastry, S.S.** (1983), "Tracking Control of Non-linear Systems Using Sliding Surfaces, with Application to Robot Manipulators", Int. Journal on Control 38 No. 2, p. 465-492.
 46. **Slotine J.J.E. et W. Li.** (1987), "On the adaptive control of robot manipulators", Int. Journal of Robotics Researches, Vol. 6, p.49- 59.
 47. **Niemeyer, G., Slotine.** (1991), "Stable adaptive teleoperation", IEEE Journal of Oceanic Engineering 16(1), 152-162.
 48. **E. Kamrani, H. R. Momeni and A. R. Sharafat.** (2005), "A Novel Adaptive Control System for Stable Teleoperation via Internet", Proceedings of IEEE Conference on Control Applications, Toronto.
 49. **Tanaka. H., Sumi. Y. and Matsumoto. Y.** (2012), "A Visual Marker for Precise Pose Estimation based on Lenticular Lenses", IEEE International Conference on Robotics and Automation (ICRA), pages 5222-5227.
 50. **J. R. Beveridge, P. J. Phillips, D. S. Bolme, B. A. Draper.** (2013), G. H. Givens, Y. M. Lui, M. N. Teli, H. Zhang, W. T. Scruggs, K. W. Bowyer, P. J. Flynn, and S. Cheng, "The challenge of face recognition from digital point-and-shoot cameras". IEEE Conference on Biometrics: Theory, Applications and Systems.
 51. **Ishii, H., Bian, Z., Fujino, H., Sekiyama, T., Nakai, T., Okamoto, A., Shimoda, H., Izumi, M., Kanehira, Y. and Morishita, Y.** (2007), Augmented reality applications for nuclear power plant maintenance work, in '19th Australasian conference on Computer-Human Interaction: Entertaining User Interfaces (OZCHI '07)', ACM, pp. 123-130.
 52. **Masanori. Izumi, Yoshinori. Kanehira and Yoshitsugu. Morishita.** (2008), "Augmented Reality Applications for Nuclear Power Plant Maintenance Work", International Symposium on symbiotic Nuclear Power Systems for 21st Century (ISSNP).
 53. **Hirotake Ishii1, Zhiqiang Bian1, Hidenori Fujino1, Tomoki Sekiyama1, Toshinori Nakai1, Akihisa Okamoto1, Hiroshi Shimoda1.** (2008), "Augmented Reality Applications for Nuclear Power Plant Maintenance Work"
 54. **Søren. S. Sørensen.** (2014), "Augmented Reality for Improved Communication of Construction and Maintenance Plans in Nuclear Power Plants", Progress of Nuclear Safety for Symbiosis and Sustainability, Springer, pp 269-274.
 55. **Bin Luo. And Shuangquan Ge.** (2015) "Augmented Reality for Material Processing within Shielded Radioactive Environment", 2015 8th International Congress on Image and Signal Processing (CISP 2015), 978-1-4673-9098-9/15/\$31.00 ©2015 IEEE
 56. **F.Z Zendaoui, A. Zaatri, M. Oulhadj Mahmoudi and R. Ouiguni.** (2007), "Teleoperation experiments for task manipulation through internet", World Journal of Engineering 4(2) pages.39-44. ISSN 1708-5284, ISI Web of Science.
 57. **Wan Muhamad Hanif Wan Kadir, Reza Ezuan Saminb, Babul Salam Kader Ibrahim.** (2012), "Internet Controlled Robotic Arm", Procedia Engineering 41 (2012) 1065–1071, 1877-7058 © 2012 Published by Elsevier Ltd.

doi: 10.1016/j.proeng.2012.07.28

58. **Paul Nadrag.** (2013), "Faciliter la télé-opération d'un robot mobile non-holonome", thèse de doctorat à l'Université Evry Val d'Essonne.
59. **Riccardo Muradore and Paolo Fiorini.** (2016), "A Review of Bilateral Teleoperation Algorithms", Acta Polytechnica Hungarica Vol. 13 No. 1.
60. **R. Azuma, Y. Baillet, R. Behringer, S. Feiner, S. Julier, and B. MacIntyre.** (2001) "Recent Advances in Augmented Reality", IEEE Computer Graphics and Applications, vol. 21, no. 6, pp. 34-47.
61. **Florian Leutert and Klaus Schilling.** (2012), "Support of Power Plant Telemaintenance with Robots by Augmented Reality Methods", 2nd International Conference on Applied Robotics for the Power Industry (CARPI), ETH, Zurich Switzerland.
62. **R.T. Azuma.** (1997), "A Survey of Augmented Reality" Presence: Teleoperators and Virtual Environments, vol. 6, no. 4, pp. 355-385.
63. **Manoj Gupta, Suchi Upadhyay, S.K.Singh, Ashok K. Nagawat.** (2011), Camera calibration technique using tsai's algorithm, ISSN (Online): 2230-8849, Vol. 1 Issue 2 July 2011.
64. **Feng Zheng.** (2015), "Spatio-temporal registration in augmented reality", A dissertation submitted to the faculty of the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy, Department of Computer Science.
65. **Daniel Cohen, Eddie Edwards, Ara Darzi and Guang-Zhong Yang.** (2012), "An effective visualisation and registration system for image-guided robotic partial nephrectomy", J Robotic Surg, Springer-Verlag London Ltd. doi:10.1007/s11701-011-0334-z.
66. **Tsai. R. Y.** (1987), "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf tv cameras and lenses", Aug. 1987. Camera calibration.
67. **Z. Zhang.** (1999), "A Flexible New Technique for Camera Calibration", Technical Report MSR-TR-98-71, Microsoft Research, Microsoft Corporation (www.microsoft.com).
68. **Wei Sun. and Jeremy R. Cooperstock.** (2006), "An empirical evaluation of factors influencing camera calibration accuracy using three publicly available techniques", Machine Vision and Applications, pp. 51-67.
69. **Hokayem, P., Spong, M.** (2006), "Bilateral teleoperation: An historical survey", Automatica 42, 2035-2057.
70. **M. Makarov, M. Grossard, P. Rodríguez-Ayerbe, D. Dumur** (2016), "Modeling and preview H_∞ control design for motion control of elastic joint robots with uncertainties", IEEE Transactions on Industrial Electronics, Vol. 63 No. 10, pp 6429-6438.
71. **Nader Cheaib, Mouna Essabbah, Christophe Domingues, Samir Otmame.** (2012), "Software architecture for interactive robot teleoperation", Symposium on Engineering Interactive Computing Systems (ACM EICS 2012), Jun 2012, Copenhagen, Denmark.
72. **J. Wilson, M. Charest, R. Dubay.** (2016), "Non-linear model predictive control schemes with application on a 2 link vertical robot manipulator", Robotics and Computer-Integrated Manufacturing, Vol. 41, pp.23-30
73. **Ehsan Kamrani & Mohamad Sawan.** (2012), "Wave Prediction and Delay Modeling for Teleoperation Via Internet", Global Journal of Computer Science and Technology Network, Web & Security. Volume 12 Issue 11, Global Journals Inc. (USA) ISSN: 0975-

- 4350.
74. **Stella Clarke, Gerhard Schillhuber, Michael F. Zaeh and Heinz Ulbrich.** (2008) "Prediction-based methods for teleoperation across delayed networks", *Multimedia Systems* 13:253–261 DOI 10.1007/s00530-007-0103-z springer.
 75. **Krid, M. and Ben Amar, F.** (2015), "Path tracking of fast rover based on nonlinear predictive controller", *Robotica*, Vol 33, N°3.
 76. **Sliding: Xu, J. X., et W. J. Cao.** (2000) "Synthesized sliding mode control of a single-link flexible robot", *International Journal of Control*, vol. 73, no. 3, p. 197-209.
 77. **Y. Orlov, W. Perruquetti, and J.P. Richard.** (2003), "Sliding mode control of uncertain time-delay systems", *Asian Journal of Control*, Vol. 5, No. 4, pp. 568-577.
 78. **Fareh, R., M. Saad, et M. Saad.** (2009), "Adaptive Control for a Single Flexible Link Manipulator Using Sliding Mode Technique", 6th International Multi Conference on Systems, Signals and Devices.
 79. **Farzin Piltan, N. Sulaiman, Mehdi Rashidi, Zahra Tajpeikar & Payman Ferdosali.** (2011), "Design and Implementation of Sliding Mode Algorithm: Applied to Robot Manipulator-A Review", *International Journal of Robotics and Automation (IJRA)*, Volume (2), Issue (5).
 80. **Abderrahmane Kheddar, Ee-Sian Neo, Riichiro Tadakuma, and Kazuhito Yokoi.** (2007), "Enhanced Teleoperation through Virtual Reality Techniques", AIST/CNRS Joint Japanese-French Robotics Laboratory Intelligent Systems Research Institute AIST Central 2, Umezono 1-1-1,305-305-8568,Tsukuba,Japankheddar@ieee.org.
 81. **Philippe Le Parc, , E. Pardo, A. Touil, J. Varelle** (2010), "Virtual Reality to Improve Remote Control in Presence of Delays", *IEEE International Conference on Virtual Environments, Human- Computer Interfaces, and Measurement SysteMS (VECIMS 2010)*, Tarente, Italy. pp 42-46.
 82. **Bryan C. Russell · Antonio Torralba · Kevin P. Murphy · William T. Freeman.** (2008) "LabelMe: A Database and Web-Based Tool for Image Annotation", *Int J Comput Vis* 77: 157-173 Doi 10.1007/s11263-007-0090-8. Springer.
 83. **E. Olson.** (2011), "AprilTag: A robust and flexible visual fiducial system", *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, IEEE pages 3400-3407.
 84. **M. Mairi, F. Ababsa & M. Mallem.** (2006) "Robust Fiducials Tracking in Augmented Reality", the 13th International Conference on Systems, Signals and Image Processing (IWSSIP 2006), 21-23 Budapest-Hongrie.
 85. **Katiegrace Youngsma.** (2012), "Development of a Model and Simulation Framework for a Modular Robotic Leg", Thesis, Faculty of Worcester Polytechnic Institute in partial fulfillment of the requirements for the Degree of Master of Science in Robotics Engineering.
 86. **D. J. Hartenberg R.S.** (1955), "A kinematic notation for lower pair mechanism based on matrices", *Trans. of ASME, J. of Applied Mechanics*, Vol. 22, pp. 215-221.
 87. **Zaatri and M. Oussalah.** (2003), "Integration and design of multimodal interfaces for supervisory control systems", *Information fusion journal*, pp. 135-150.E.
 88. **Mohammad Afif Ayoba, Wan Nurshazwani Wan Zakariaa, Jamaludin Jananib, Mohd Razali Md Tomari.** (2015), "Modeling and simulation of a 5-axis rv-2aj robot using simmechanics", *Jurnal Teknologi* 2015.
 89. **J. Fang and W. Li.** (2013), "Four degrees of freedom SCARA robot kinematics modeling and simulation analysis", *International Journal of Computer, Consumer and Control (IJ3C)*, vol. 2, no. 4, pp. 20 -27, 2013.
 90. **Rajeev Agrawal, Koushik Kabiraj, Ravi Singh** (2012), "Modeling a Controller for an Articulated Robotic Arm", India.

91. MathWorks, Simscape™ Multibody™ Getting Started Guide, 2016. Available: http://cn.mathworks.com/help/pdf_doc/physmod/sm/sm_gs.pdf. Accessed on: Mar. 16, 2016.
92. SimPowerSystems & Simulink", MathWorks, 2013.
93. "SimMechanics, User's Guide", R2015a, MathWorks.
94. **V. Fedák, F. Ďurovský, and R. Üveges.** (2014), "Analysis of Robotic System Motion in SimMechanics and MATLAB GUI Environment in MATLAB Applications for the Practical Engineer", 3rd ed., K. Bennett (ed). Rijeka, Croatia: InTech, 2014, pp. 565-581.
95. **Saravana Mohan Mariappan, Anbumalar Veerabathiran.** (2016), "Modelling and simulation of multi spindle drilling redundant SCARA robot using SolidWorks and MATLAB/SimMechanics", Revista Facultad de Ingeniería, Universidad de Antioquia, No. 81, pp. 63-72, 2016, ISSN 0120-6230. DOI: 10.17533/udea.redin.n81a06.
96. **M. E. Kütük, R. Halicioglu, and L. C. Dulger.** (2015), "Kinematics and Simulation of a Hybrid Mechanism: MATLAB/ SimMechanics", Journal of Physics: Conference Series, vol. 574, pp. 451-458.
97. **S. Umar and E. A. Bakar.** (2014) "Study on Trajectory Motion and Computational Analysis of Robot Manipulator", Jurnal Teknologi, vol. 67, no. 1, pp. 53-59, 2014.
98. **A. K. Jha, A. K. Dutta, and J. Saha.** (2014), "Analysis of Dynamics of SCORA-ER14 Robot in MATLAB", International Journal of Innovative Research in Advanced Engineering (IJRAE), vol. 1, no. 4, pp. 145-150, 2014.
99. **M. Fajar, S. S. Douglas, and J. B. Gomm.** (2013), "Modelling and Simulation of spherical inverted pendulum based on LQR control with SimMechanics", Applied Mechanics and Materials, vol. 391, pp 163-167, 2013.
100. **Yuan. Shaoqiang, Liu. Zhong. & Li. Xingshan.** (2008), "Modeling and Simulation of Robot Based on Matlab/SimMechanics", Proceedings of the 27th Chinese Control Conference, IEEE Press, pp. 161-165.
101. **Leon Žlajpah.** (2008), "Simulation in robotics", Published by Elsevier B.V, © 2008 IMACS, pp. 879-897, 0378-4754, doi:10.1016/j.matcom.2008.02.017.
102. **Woldu Zina Gebrehiwot.** (2010), "Design and Control of a Five Bar Linkage Parallel Manipulator with Flexible Arm", Thesis. Milan.
103. **L. Žlajpah.** (2001), "Integrated environment for modeling, simulation and control design for robotic manipulators", Journal of Intelligent and Robotic Systems, 32(2):219-234.
104. **Evyatar Bukai.** (2013), "Modeling and control of a four-axis control moment gyroscope", Bachelor Thesis, Czech Technical University in Prague Faculty of Electrical Engineering Department of Control Engineering.
105. **R-M. ASSOUMOU NZUE, J-F. BRETHER, E. VASSELIN, D.LEFEBVRE.** (2011), "Comparaison de la répétabilité des robots manipulateurs séries et parallèles à l'aide des ellipsoïdes stochastiques", 20ème Congrès Français de Mécanique Besançon, France.
106. **Steve Vozar and Dawn M Tilbury.** (2013), "Improving teleoperated robot speed using optimization techniques". In Proceedings of the 8th ACM/IEEE international conference on Human-robot interaction, pages 249–250. IEEE Press.
107. **C. Goerzen, Z. Kong, and B. Mettler.** (2010), "A survey of motion planning algorithms from the perspective of autonomous uav guidance", j. intell. Robotics syst., vol. 57, 2010, pp.65-100.
108. **Luigi Biagiotti and Claudio Melchiorri.** (2008). "Trajectory Planning for Automatic Machines and Robots". Springer Publishing Company Incorporated, 1st edition 2008. ISBN 3540856285, 9783540856283.
109. **Peter Frankovský, Darina Hroncová, Ingrid Delyová, Ivan Virgala.** (2013), "Modeling of Dynamic Systems in Simulation Environment MATLAB/Simulink",

- American Journal of Mechanical Engineering, Vol. 1, No. 7, 282-288.
110. **C.H. Aung, K.T. Lwin, and Y.M. Myint.** (2008), "Modeling Motion Control System for Motorized Robot Arm using MATLAB", World Academy of Science Eng. and Tech.
 111. **Torsten Kröger.** (2010), "On-Line Trajectory Generation in Robotic Systems Basic Concepts for Instantaneous Reactions to Unforeseen (Sensor) Events", volume 58 of Springer Tracts in Advanced Robotics. Springer-Verlag Berlin Heidelberg, Berlin, Heidelberg, Germany, January 2010.
 112. **Kröger 10b T. Kröger & F.M. Wahl.** (2010), "Online Trajectory Generation: Basic Concepts for Instantaneous Reactions to Unforeseen Events", Robotics, IEEE Transactions on, vol. 26, no. 1, pages 94 –111, 2010. 29, 31, 103.
 113. **Kroger, T., & Padiál, J.** (2012), "Simple and robust visual servo control of robot arms using an on-line trajectory generator". In Robotics and Automation (ICRA), 2012 IEEE International Conference on (pp. 4862-4869). IEEE.
 114. **Wisama Khalil and Étienne Dombre.** (2002), "Modeling, identification and control of robots".HPS, London, 2002. ISBN 1-903996-13-9. URL <http://opac.inria.fr/record=b1098538>.
 115. **Etienne Dombre and Wissama Khalil** (2007), "Robot Manipulators: Modeling, Performance Analysis and Control", ISTE, London,. ISBN 1-905209-10-X.
 116. **Katzschmann, R., Kroger, T., Asfour, T., & Khatib, O.** (2013, November), "Towards online trajectory generation considering robot dynamics and torque limits", In Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on (pp. 5644-5651). IEEE.
 117. **Vincent BAGAÏNI, Loïc BAUM.** (décembre 2019), "Modélisation et simulation cinématique du robot Stäubli® TX2-90 sur Matlab avec Robotics Toolbox", Ecole de technologie supérieure, département génie électrique, Canada.
 118. **S. Umar and E. A. Bakar.** (2014) "Study on Trajectory Motion and Computational Analysis of Robot Manipulator", Jurnal Teknologi, vol. 67, no. 1, pp. 53-59, 2014.
 119. **C. Urrea and J. Kern.** (2015), "Trajectory Tracking Control of a Real Redundant Manipulator of the SCARA Type", J. Electr. Eng. Technol., vol. 10, pp. 709-720, 2015.
 120. **J. yu,q. zhang, v. kroumov, s. cheng, and z. zhang,** (2008), "path planning algorithm for robot in 3d environment based on neural network", intelligent robotics and applications, c. xiong, h. liu, y. huang, and y. xiong, eds., springer berlin / heidelberg, 2008, pp. 1081-1088.
 121. **JLASSI SARRA.** (2013), "Formulation et etude des problèmes de commande en co-manipulation robotique", Thèse de doctorat, école doctorale: Sciences et Technologie de l'Information, des Télécommunications et des Systèmes, Laboratoire de signaux et systèmes, 2013 Paris.
 122. **Xavier Broquere.** (2011). "Planification de trajectoire pour la manipulation d'objets et l'interaction Homme-Robot, Thèse de doctorat de l'Univesrsité de Toulouse III Paul Sabatier (UT3 Paul Sabatier)
 123. <http://www.mathworks.com/help/toolbox/distcomp/f1-6010.html>.
 124. **Chi-Shen Tsai.** (2014). "Online Trajectory Generation for Robot Manipulators in Dynamic Environment | An Optimization-based Approach", A dissertation submitted in partial satisfaction of the requirements for the degree of Doctor of Philosophy, University of California, Berkeley.
 125. **Aurélié Bugeau.** (2011), "Détection et suivi d'objets en mouvement dans des scènes complexes, application à la surveillance des conducteurs", Thèse présentée devant l'université de rennes I pour obtenir le grade de : docteur de l'université de rennes I Mention : Traitement du signal et télécommunications.

Annexes

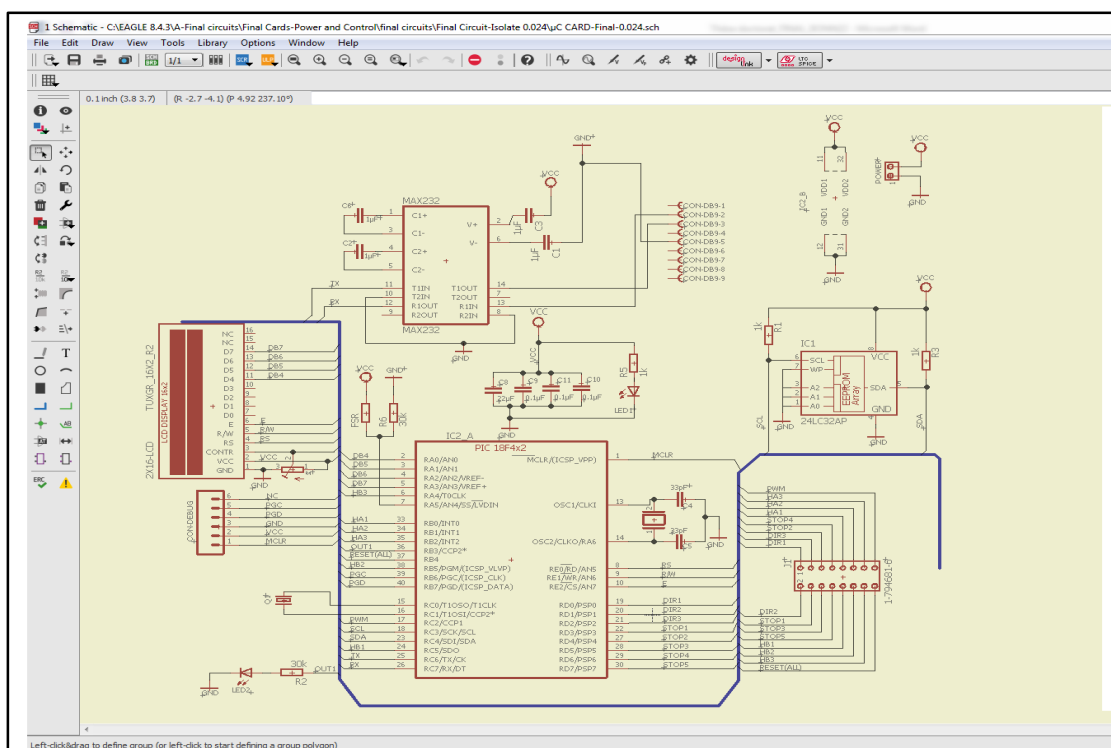
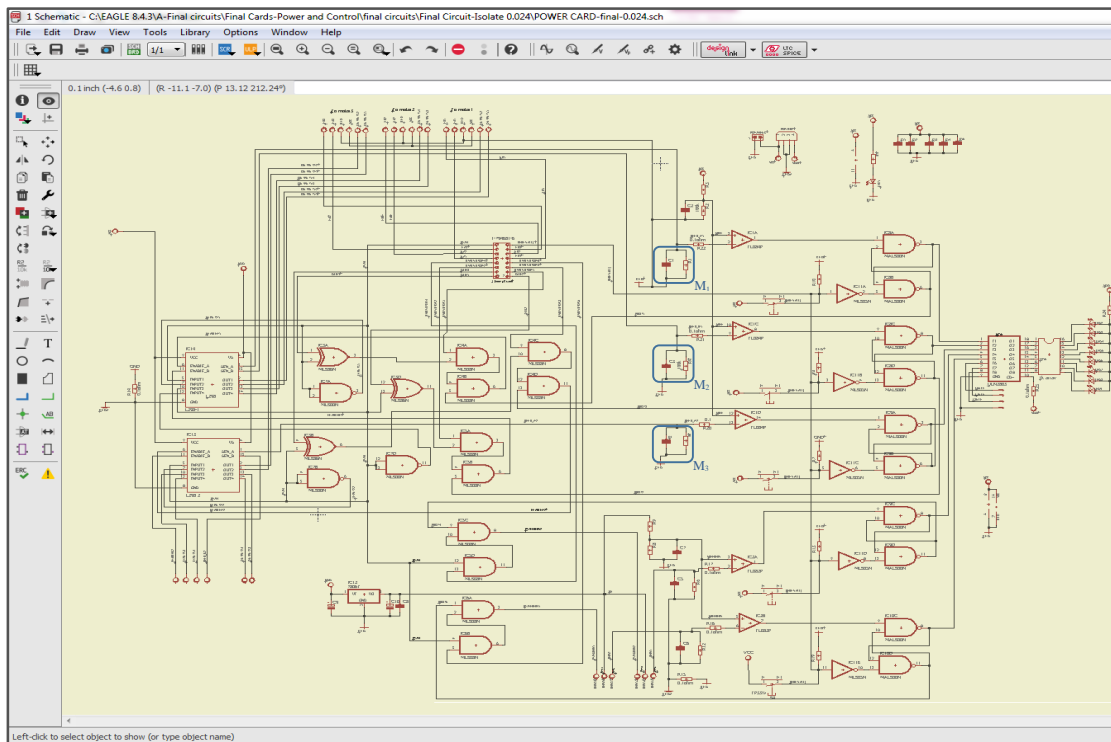
L'annexe A, affiche les cartes de commande à base du pic 18F452 et la carte de puissance, et le programmes en assembleur qui permet la programmation de toutes les fonctions liées au microcontrôleur.

L'annexe B présente un programme utilisé en Matlab pour actionner les moteurs.

L'annexe C : les schémas de conception des différents éléments constituant le robot manipulateur.

Annexes A :

1. Cartes de puissance et de commande du robot manipulateur



2. Programme en assembleur de toutes les fonctions de commande liées au microcontrôleur

```

include P18F452.inc
processor 18F452
CONFIG      OSC= HSPLL, OSCS=OFF,PWRT=ON, BOR=ON, BORV=27, WDT=OFF,
WDTPS=4, CCP2MUX=OFF, LVP=OFF

; PWM output at pin RC2 (pin 17)

file Equ      1
DIRB Equ      RD0      ; Base,pin19
DIRA Equ      RD1      ; Arm,pin20
DIRS Equ      RD2      ; Shoulder,pin21
STOPB Equ     RD3      ; StopBase,pin22
STOPA Equ     RD4      ; StopArm,pin27
STOPS Equ     RD5      ; StopShoulder, pin28
STOPL Equ     RD6      ; StopServoL,pin29
STOPH Equ     RD7      ; StopServoH,pin30
HA1 Equ       RB0      ; pin33 du Pic
HB1 Equ       RC5      ; pin24 du Pic
HA2 Equ       RB1      ; pin34 du Pic
HB2 Equ       RB5      ; pin38 du Pic
HA3 Equ       RB2      ; pin35 du Pic
HB3 Equ       RA4      ; pin6 du Pic
ResAll Equ    RB4      ; ResetAll output (pin37)
              RB3      ; PWM Servos,pin36 du Pic

Receive_Buff Equ 0x14A ; Start adress of serial data receive

PR2_Val Equ 0x3D ; fix the period of the pwm to 10 KHZ
Duty_Val Equ 0x26;3C ; max value of CCPR1_L for 95% duty cycle
Chg Equ 0 ; Chg is the bit flag for Inc/Dec position
indication
Sup Equ 1 ; bit for positif compare
Inf Equ 2 ; bit for negatif compare
Eq Equ 3 ; bit for equal compare
Tic1 Equ 0 ; 0.5 ms tic set bit
Tic2 Equ 1 ; 8 ms tic set bit
Tic3 Equ 2 ; tic bit for motor power duration
Tic4 Equ 3 ; tic bit for data sending duration
Data_Ready Equ 0 ; Data in Receive Buffer ready
chk Equ 1 ; Checksum bit
Time Equ 3 ; Timeout bit activate
Timeout Equ 4 ; status bit for time transmission duration

cblock 0x00

MotApos_L,MotApos_H,MotAPos_Flag,MotBPos_L,MotBPos_H,MotBPos_Flag
PsA_L,PsA_H,PsbA_L,PsbA_H,PsdA_L,PsdA_H
PsB_L,PsB_H,PsbB_L,PsbB_H,PsdB_L,PsdB_H
TimerP_L,TimerP_H,Duty_Buff,Timer_Flag,Receive_Flag
Command,Count_Byte,servo_L,servo_H,Timer_Ser

endc

org 00

goto main

org 0x08
btfsc INTCON,TMR0IF,0

```

```

    call TMR0_ISR          ;ISR: Interrupt Service Routine
    btfsc PIR2,CCP2IF,0   ;Timer module
    call CCP2_ISR         ;Compare module
    btfsc PIR1,RCIF       ;RCIF Receive interrupt file
    call Receive_ISR      ;RS232
    btfsc INTCON,INT0IF,0 ;Encoder Bras
    call MotBPos_ISR
    btfsc INTCON3,INT1IF,0 ;Encoder Arm
    call MotAPos_ISR
    btfsc INTCON3,INT2IF,0 ;Shoulder Motor
    call MotSPos_ISR      ;Encoder Motor3

    retfie

; ***** Main Program *****

    org 100h

main
    call init_Port
    call init_PWM
    call init_Compare2
    call init_TMR0
    call init_Interrupt

                                ;initialisations
    call init_Serial
    call init_MotB          ; MotB is the Body Motor
    call init_MotA          ; MotA is the Arm Motor
    call init_Servo
    bsf  PORTB,RB4          ; reset all

repm call Receive_Process
    call MotorA_Control
    call MotorB_Control
    ;call ServoH_Control
    ;call ServoL_Control
    goto repm

init_MotA
    clrf MotAPos_H,0
    clrf MotAPos_L,0
    clrf MotAPos_Flag
    clrf PsA_H
    clrf PsA_L
    return

init_MotB
    clrf MotBPos_H,0
    clrf MotBPos_L,0
    clrf MotBPos_Flag
    clrf PsB_H
    clrf PsB_L
    return

init_Servo
    movlw 0x21              ;corresponding at 1msec.
                                ;The max value (2ms) is fixed with 0x41
    movwf Servo_L          ;servo opening the gripper

```



```

        clrf Servo_H                ;servo rotating the gripper
        return

init_Port                ;***** Configure µC peripherals operations
*****

        movlb 0x0F                ;move literal to bank
        select register BSR<3:0>
        ;Setting input/output Ports

        clrf TRISD                ; Set PORTD in output Mode (Direction and
start/STOP)
        bcf TRISB,RB4              ; ResetAll output
        bcf TRISB,RB3              ; set compare output (RB3)
        bsf TRISB,RB0              ; HA1 input
        bsf TRISB,RB1              ; HA2 input
        bsf TRISB,RB2              ; HA3 input
        bsf TRISC,RC5              ; HB1 input
        bsf TRISB,RB5              ; HB2 input
        bsf TRISA,RA4              ; HB3 input

        bcf TRISE,RE0              ; set RS to output

; Set Output port state

        call ResetM
        bcf PORTD,STOPA            ; STOP MotArm
        bcf PORTD,STOPB            ; STOP MotBody
        bcf PORTD,STOPS            ; STOP MotShoulder
        bsf PORTD,STOPL            ; STOP Servo Low
        bcf PORTD,STOPH            ; STOP Servo High
        bcf PORTD,DIRB              ; Set Direction MotA (Arm motor)
        bcf PORTD,DIRA              ; Set Direction MotB (Body motor)
        bcf PORTD,DIRS              ; Set Direction MotS (Shoulder
motor)
        return

init_PWM                ; ***** Configure PWM Module *****

        movlw PR2_Val
        movwf PR2                  ;fix the period of the pwm to 10
KHZ
        movlw 0x0C
        movwf CCP1CON              ;enable pwm module
        movlw Duty_Val
        movwf CCP1L                ;fix the duty cycle to 95%( CCP1L ou Dutyval fixé
à 3B)
        movlw 0x07
        movwf T2CON                ;enable TMR2 and prescaler to 16
        bcf TRISC,2                ;set pwm output
        movlw Duty_Val
        movwf Duty_Buff
        return

init_Compare2            ;Configure Compare2 module

        movlw 0x8B
        movwf T3CON                ;configure TMR1 as CCP1 source
clock
        movwf T1CON
        movlw 0x02

```

```

    movwf CCP2CON                ;configure CCP2 to toggle mode
    bcf   PIR2,CCP2IF
    movlw 0x8F
    movwf CCPR2L
    movlw 0x02
    movwf CCPR2H
    bcf   PIR2,CCP2IF
    return

init_TMR0                        ; Configure TMR0 for Time base (0.5
ms)

    movlw 0x80
    movwf T0CON                ;configure TMR0 to 16 bit mode & 1:2
prescale
    movlw 0xF6
    movwf TMR0H
    movlw 0x41
    movwf TMR0L                ;fix base time to 0.5ms
    clrf TimerP_H
    clrf TimerP_L
    return

init_Interrupt
; ***** Enabling needed Interrupts *****
; INTO INT1 INT2 CCP2 (compare) to enable

    bsf   INTCON,GIE           ;enable global interrupt
    bsf   INTCON,PEIE         ;enable peripheral interrupts
    bcf   RCON,IPEN           ;disable priority levels on
interrupts
    bsf   PIE2,CCP2IE         ;enable CCP2 interrupt
    bcf   PIE1,CCP1IE         ;enable CCP1 interrupt
    bsf   INTCON,TMR0IE       ;enable TMR0 interrupt
    bsf   INTCON,INT0IE       ;enable QEncoder motor interrupt
    bsf   INTCON3,INT1IE
    bsf   INTCON3,INT2IE
    bsf   PIE1,RCIE           ;active the interruption
    return

init_Serial                    ; initialise the serial port (**Configure UART
Peripheral **)

    bcf   TRISC,RC6           ;set transmit pin to output
    (Tx)
    bsf   TRISC,RC7           ;set receive pin to input
(Rx)
    movlw B'00100100'
    movwf TXSTA                ;asynchronous mode, 8 bit transmission
    movlw B'10010000'         ;enable reception
    movwf RCSTA
    movlw 0x15
    movwf SPBRG                ;fix the speed to 115200 baud (error
1.3%)
    lfsr  0,Receive_Buff      ;load FSR0 with adress receive data
store
    clrf  Count_Byte
    clrf  Receive_Flag
    return

ResetM
    bsf   PORTB,RB4

```

```

        bcf    PORTB, RB4
        return

CCP2_ISR
    movlw 0
    movwf TMR3H
    clrf  TMR3L
    bcf  PIR2, CCP2IF
    btfss PORTB, RB3                ;test state of compare output
    call FixLow
    btfsc PORTB, RB3                ;set the PWM Servo
    call FixHigh
    return

FixLow                                ; fix compare register to low value
(SERVOS)
    movlw 0x8F
    movwf CCPR2L
    movlw 0x02
    movwf CCPR2H
    return

FixHigh
    movff Servo_L, CCPR2L
    movff Servo_H, CCPR2H
    return

TMR0_ISR                                ;fix tic time to 0.5 ms
    movlw 0xF6
    movwf TMR0H
    movlw 0x41
    movwf TMR0L
    incf TimerP_L, F, 0
    btfsc Status, Z
    incf TimerP_H, F, 0
    bsf  Timer_Flag, Tic1

    btfss Receive_Flag, Time
    goto nextm
    incf Timer_Ser, F
    btfsc Timer_Ser, 2
    bsf  Receive_Flag, Timeout
    btfsc Timer_Ser, 2
    clrf Timer_Ser

nextm
    bcf  INTCON, TMR0IF
    return

Receive_ISR
    btfss Receive_Flag, Time
    clrf Timer_Ser
    btfss Receive_Flag, Time
    bsf  Receive_Flag, Time
    movff RCREG, Postinc0
    incf Count_Byte, F
    btfsc Count_Byte, 2
    bsf  Receive_Flag, Data_ready

    bcf  PIR1, RCIF, 0                ;receive ISR
    return

MotAPos_ISR

```

```

        btfsc PORTB,HB2
        goto deca
        infsnz      MotAPos_L,F,0
        incf MotAPos_H,F,0
        goto stpa

deca
        movf MotAPos_L,W
        iorwf MotAPos_H,W
        btfsc Status,Z
        goto stpa
        decf MotAPos_L,F,0           ;dec for decrement
        btfss status,c
        decf MotAPos_H,F,0

stpa bcf      INTCON3,INT1IF
        bsf MotAPos_Flag,Chg       ;signal position change
        return

MotBPos_ISR
        btfsc PORTC,HB1
        goto decb
        infsnz      MotBPos_L,F,0
        incf MotBPos_H,F,0
        goto stpb

decb movf MotBPos_L,W
        iorwf MotBPos_H,W
        btfsc Status,Z
        goto stpb

        decf MotBPos_L,F,0           ;dec for decrement
        btfss status,c
        decf MotBPos_H,F,0

stpb bcf      INTCON,INT0IF
        bsf MotBPos_Flag,Chg
        return

MotSPos_ISR
        bcf INTCON3,INT2IF
        return                               ; Shoulder Motor not used

Transmit_Data
        movff TimerP_L,TXREG
        btfss TXSTA,TRMT
        goto $-2

        movff TimerP_H,TXREG
        btfss TXSTA,TRMT
        goto $-2

        movff MotAPos_L,TXREG
        btfss TXSTA,TRMT
        goto $-2

        movff MotAPOS_H,TXREG
        btfss TXSTA,TRMT
        goto $-2
        bcf Timer_flag,Tic2
        return

```

```

Receive_Process
    btfsc RCSTA,FERR ; Test for errors (Frame errors and Overrun
                        ; errors)
    call Fix_Receive
    btfsc RCSTA,OERR
    call Fix_Receive
    btfsc Receive_Flag,Time

    call Test_Timeout ; Depassement de
temps
    btfsc Receive_Flag,Data_ready
    call Checksum ; verify data integrity
                ; (Verification checksum)

    btfss Receive_Flag,Data_ready
    goto EP
    call Process_Message

EP
    return

Process_Message

    lfsr FSR0,Receive_Buff ; point to the first byte of
message
    movlw 'a'
    cpfseq INDF0
    goto next1
    call set_PosA
    goto nextF
next1 movlw 'b'
    cpfseq INDF0
    goto next2
    call set_PosB
    goto nextF
next2 movlw 'h'
    cpfseq INDF0
    goto next3
    call set_PosH
    goto nextF
next3 movlw 'l'
    cpfseq INDF0
    goto nextF
    call set_PosL

nextF call Fix_Receive
    return

set_PosA
    movff PREINC0,PsA_L
    movff PREINC0,PsA_H
    bsf MotAPos_Flag,Chg
    return

set_PosB
    movff PREINC0,PsB_L
    movff PREINC0,PsB_H
    bsf MotBPos_Flag,Chg
    return

set_PosH
    bcf PORTD,STOPL
    bsf PORTD,STOPH
    movff PREINC0,Servo_L

```

```

        movff PREINC0,Servo_H
        return

set_PosL
        bcf   PORTD,STOPH
        bsf   PORTD,STOPL
        movff PREINC0,Servo_L
        movff PREINC0,Servo_H
        return

Checksum
        lfsr  0,Receive_Buff
        clrf  Count_byte
        movlw 0
chk1    addwf Postinc0,W
        incf  Count_Byte,F
        btfss Count_byte,2
        goto  chk1
        movwf Count_byte
        movf  Count_byte,F                ; to verify if result is
zero    btfsc  Status,Z
        bsf   Receive_Flag,chk          ; Data OK
        btfss Status,Z
        call  Fix_Receive

        return

Test_Timeout
        btfsc Receive_Flag,Timeout
        call  Fix_Receive
        return

Fix_Receive
        lfsr  0,Receive_Buff
        clrf  Count_byte
        bcf   RCSTA,RCEN,0              ; clear error (FERR,OERR)
        bsf   RCSTA,RCEN,0
        bcf   RCSTA,ADDEN,0
        clrf  Receive_Flag
        clrf  Timer_Ser
        return

MotorA_Control
        btfss MotAPos_Flag,Chg
        goto  MAE                        ;put end for Motor A
        btfsc PORTD,STOPB
        goto  MAE

        movlw 0x3C
        movwf Duty_Buff

        call  substA                    ;comparison of motor position "Ps and
Pr"
        btfsc MotAPos_Flag,Sup
        call  DirectA_Sup
        btfsc MotAPos_Flag,Inf
        call  DirectA_Inf
        btfsc MotAPos_Flag,Eq
        call  DirectA_Eq
MAE     Return

```

```

MotorB_Control
    btfss MotBPos_Flag,Chg
    goto MBE
    btfsc PORTD,STOPA
    goto MBE

    movlw 0x26
    movwf Duty_Buff

    call substB          ;comparison of motor position "Ps and
Pr"
    btfsc MotBPos_Flag,Sup
    call DirectB_Sup
    btfsc MotBPos_Flag,Inf
    call DirectB_Inf
    btfsc MotBPos_Flag,Eq
    call DirectB_Eq
MBE    Return

ServoL_Control

    return

ServoH_Control

    return

    PosA_Sens          ;NegA_Sens
    bcf    PORTD,DIRA
    movf   Duty_Buff,w
    movwf  CCPR1L
    return

    NegA_Sens          ;PosA_Sens
    bsf    PORTD,DIRA
    movf   Duty_Buff,w
    subwf  PR2,w
    movwf  CCPR1L
    return

    PosB_Sens
    bsf    PORTD,DIRB
    movf   Duty_Buff,w
    subwf  PR2,w
    movwf  CCPR1L
    return

    NegB_Sens
    bcf    PORTD,DIRB
    movf   Duty_Buff,w
    movwf  CCPR1L
    return

    PosS_Sens
    bsf    PORTD,DIRS
    movf   Duty_Buff,w
    subwf  PR2,w
    movwf  CCPR1L
    return

    NegS_Sens
    bcf    PORTD,DIRS

```

```

    movf Duty_Buff,w
    movwf CCP1L
    return

substA                                     ;subtract of 2 words
(16 bits) for Arm Motor
    movff PsA_L,PsbA_L
    movff PsA_H,PsbA_H
    movf MotAPos_L,w
    subwf PsbA_L,f
    movf MotAPos_H,w
    subwfb PsbA_H,f

    btfsc status,N ;test for Negatif/positif compare
    bsf MotAPos_Flag,Inf
    btfss status,N
    bsf MotAPos_Flag,Sup
    movff PsbA_L,Psda_L ;create copies of position difference
    movff PsbA_H,Psda_H ;to be used in Update_PWM
                        (proportional regulation)
    movf PsbA_L,w ;test for 0 compare
    iorwf PsbA_H,f
    btfsc status,z
    bsf MotAPos_Flag,Eq
; call Update_PWM_Arm ;calculate the output gain
    bcf MotAPos_Flag,Chg
    return

substB                                     ; subtract of 2 words (16 bits) for
Body Motor
    movff PsB_L,PsB_L
    movff PsB_H,PsB_H
    movf MotBPos_L,w
    subwf PsbB_L,f
    movf MotBPos_H,w
    subwfb PsbB_H,f

    btfsc status,N ;test for Negatif/positif compare
    bsf MotBPos_Flag,Inf
    btfss status,N
    bsf MotBPos_Flag,Sup
    movff PsbB_L,PsdB_L ;create copies of position
difference
    movff PsbB_H,PsdB_H ;to be used in
                        Update_PWM(proportional regulation)
    movf PsbB_L,w ;test for 0 compare
    iorwf PsbB_H,f
    btfsc status,z
    bsf MotBPos_Flag,Eq
; call Update_PWM_Body ;calculate the output gain
    bcf MotBPos_Flag,Chg
    return

DirectA_Sup
    call PosA_Sens
    bsf PORTD,STOPA
    bcf MotAPos_Flag,Sup
    return

DirectA_Inf
    call NegA_Sens

```



```

        bsf    PORTD,STOPA
        bcf    MotAPos_Flag,Inf
        return

DirectA_Eq
        bcf    PORTD,STOPA
        bcf    MotAPos_Flag,Eq
        return

DirectB_Sup
        call   PosB_Sens
        bsf    PORTD,STOPB
        bcf    MotBPos_Flag,Sup
        return

DirectB_Inf
        call   NegB_Sens
        bsf    PORTD,STOPB
        bcf    MotBPos_Flag,Inf
        return

DirectB_Eq
        bcf    PORTD,STOPB
        bcf    MotBPos_Flag,Eq
        return

Inv_SignA
        comf   PsdA_L,file
        comf   PsdA_H,file
        movlw 1
        addwf  PsdA_L,f
        movlw 0
        addwfc        PsdA_H,f
        return

Inv_SignB
        comf   PsdB_L,file
        comf   PsdB_H,file
        movlw 1
        addwf  PsdB_L,f
        movlw 0
        addwfc        PsdB_H,f
        return

Update_PWM_Arm
        movlw low genarm
        movwf tblptrl                ;table read pointeur low
        movlw high genarm
        movwf tblptrh                ;table read pointeur high
        movlw upper genarm
        movwf tblptru                ;table read pointeur upper
        btfscl MotAPos_Flag,Inf
        call   Inv_SignA
        movf   PsdA_L,w
        addwf  tblptrl,f
        movf   PsdA_H,w
        addwfc        tblptrh,f      ;addition with carry
        tblrd*+                ;table read
        movff  tablat,Duty_Buff
        return

Update_PWM_Body

```

```
    movlw low genbody
    movwf tblptrl                ;table read pointeur low
    movlw high genbody
    movwf tblptrh                ;table read pointeur high
    movlw upper genbody
    movwf tblptru                ;table read pointeur upper
    btfsc MotBPos_Flag,Inf
    call Inv_SignB
    movf PsdB_L,w
    addwf tblptrl,f
    movf PsdB_H,w
    addwfc      tblptrh,f        ;addition with carrey
    tblrd*+                      ;table read
    movff tablat,Duty_Buff
    return

    org 1000 ;Conversion table for proportional gain K for arm motor
genarm
    include regul4.txt

    org 1400 ;Conversion table for proportional gain K for shoulder
(motor not in use now)
genshoulder
    ;include regul4.txt

    org 1800 ;Conversion table for proportional gain K for base motor
genbody
    include regul5.txt

    END
```

Annexes B: Programme de commande de moteurs

Permet la télé-opération du robot en faisant fonctionner ses actionneurs 'moteurs'.

```
% *****
pos=1;

while pos >0
    pos=input('Give Position: ')
    Id='b';
    position=int16(pos)
    chk=0;
    pos=typecast(position,'uint8');
    posl=int16(pos(1,1));
    posh=int16(pos(1,2));
    chr=int16(Id);

    sum=-(chr+posl+posh);
    res=typecast(sum,'uint8')
    chk=res(1,1);

    Mes=[uint8(chr) uint8(posl) uint8(posh) chk]
    fwrite(s,Mes)
end
% *****
% pause
% *****
% Id='B';
% position=int16(230);
% chk=0;
% pos=typecast(position,'uint8');
% posl=int16(pos(1,1));
% posh=int16(pos(1,2));
% chr=int16(Id);
%
% sum=-(chr+posl+posh);
% res=typecast(sum,'uint8')
% chk=res(1,1);
% Mes=[uint8(chr) uint8(posl) uint8(posh) chk]
% fwrite(s,Mes)
% % *****
% pause
% % *****
% Id='A';
% position=int16(55);
% chk=0;
% pos=typecast(position,'uint8');
% posl=int16(pos(1,1));
% posh=int16(pos(1,2));
% chr=int16(Id);
%
% sum=-(chr+posl+posh);
% res=typecast(sum,'uint8')
% chk=res(1,1);
% Mes=[uint8(chr) uint8(posl) uint8(posh) chk]
```

```

% fwrite(s,Mes)
% % *****
% pause
% % *****
% Id='H';
% position=int16(49);
% chk=0;
% pos=typecast(position,'uint8');
% posl=int16(pos(1,1));
% posh=int16(pos(1,2));
% chr=int16(Id);
%
% sum=-(chr+posl+posh);
% res=typecast(sum,'uint8')
% chk=res(1,1);
% Mes=[uint8(chr) uint8(posl) uint8(posh) chk]
% fwrite(s,Mes)
% % *****
% pause
% % *****
% Id='L';
% position=int16(50);
% chk=0;
% pos=typecast(position,'uint8');
% posl=int16(pos(1,1));
% posh=int16(pos(1,2));
% chr=int16(Id);
%
% sum=-(chr+posl+posh);
% res=typecast(sum,'uint8')
% chk=res(1,1);
% Mes=[uint8(chr) uint8(posl) uint8(posh) chk]
% fwrite(s,Mes)
% % *****
% pause
% % *****
% Id='B';
% position=int16(90);
% chk=0;
% pos=typecast(position,'uint8');
% posl=int16(pos(1,1));
% posh=int16(pos(1,2));
% chr=int16(Id);
%
% sum=-(chr+posl+posh);
% res=typecast(sum,'uint8')
% chk=res(1,1);
% Mes=[uint8(chr) uint8(posl) uint8(posh) chk]
% fwrite(s,Mes)
% % *****
% pause
% % *****
% Id='H';
% position=int16(33);

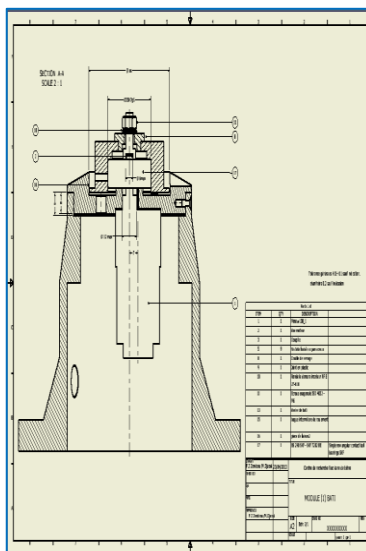
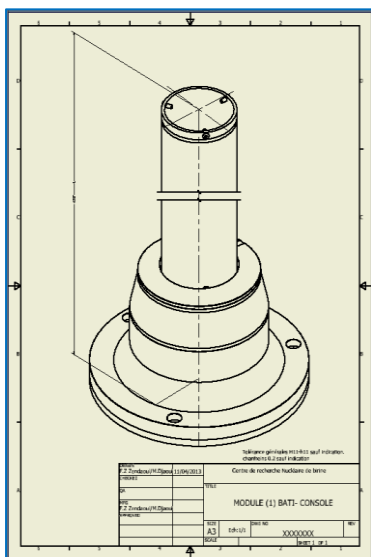
```

```
% chk=0;
% pos=typecast(position,'uint8');
% posl=int16(pos(1,1));
% posh=int16(pos(1,2));
% chr=int16(Id);
%
% sum=-(chr+posl+posh);
% res=typecast(sum,'uint8')
% chk=res(1,1);
% Mes=[uint8(chr) uint8(posl) uint8(posh) chk]
% fwrite(s,Mes)
% % *****
% pause
% % *****
% Id='L';
% position=int16(49);
% chk=0;
% pos=typecast(position,'uint8');
% posl=int16(pos(1,1));
% posh=int16(pos(1,2));
% chr=int16(Id);
%
% sum=-(chr+posl+posh);
% res=typecast(sum,'uint8')
% chk=res(1,1);
% Mes=[uint8(chr) uint8(posl) uint8(posh) chk]
% fwrite(s,Mes)
% % *****
% pause
% % *****
% Id='B';
% position=int16(33);
% chk=0;
% pos=typecast(position,'uint8');
% posl=int16(pos(1,1));
% posh=int16(pos(1,2));
% chr=int16(Id);
%
% sum=-(chr+posl+posh);
% res=typecast(sum,'uint8')
% chk=res(1,1);
% Mes=[uint8(chr) uint8(posl) uint8(posh) chk]
% fwrite(s,Mes)
```

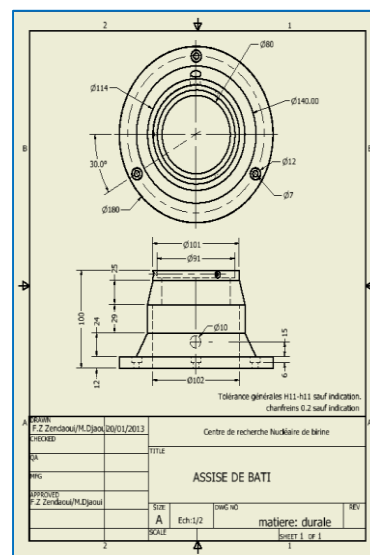
Annexes C : Schémas représentant la conception de quelques pièces constituant le robot manipulateur.

Nous représentons quelques pièces de conception du robot manipulateur.

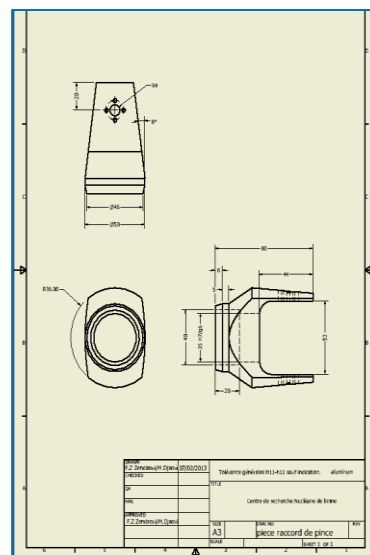
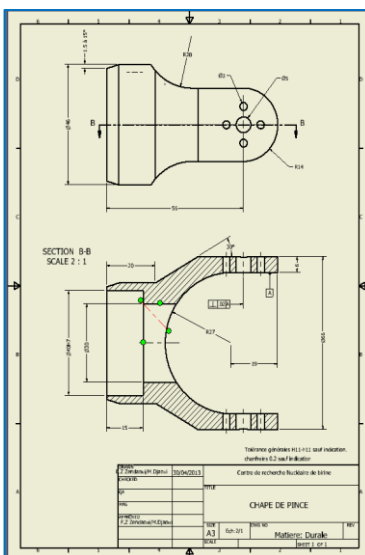
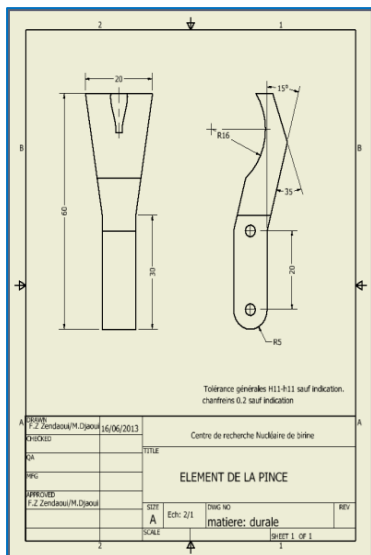
1. Modules de bati console



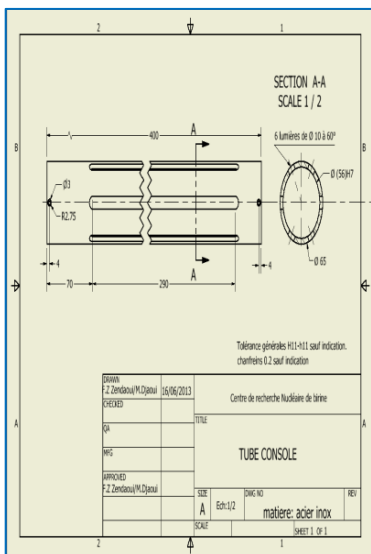
2. Assise de bati



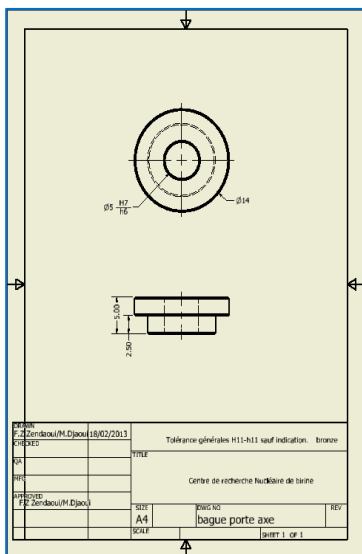
3. Quelques éléments de la pince



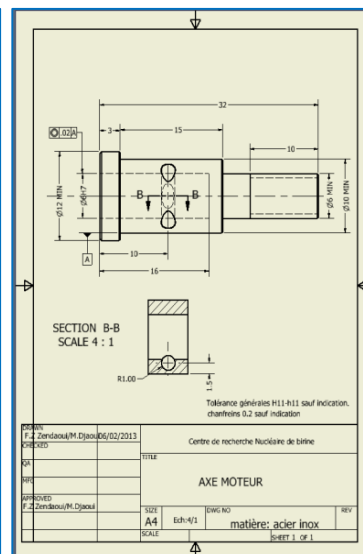
4. Tube console



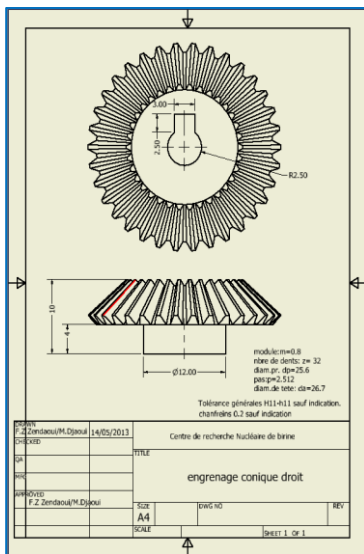
5. Bague porte_axe



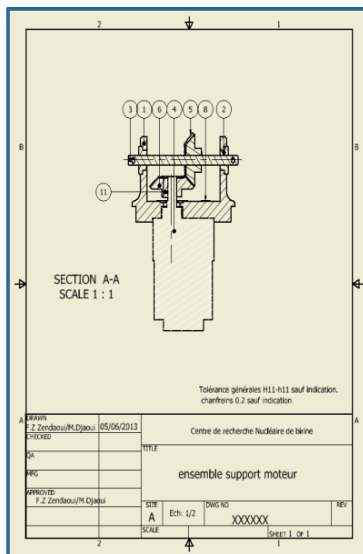
6. Axe moteur



7. Engrenages coniques



9. Ensemble support moteur



10. Axe d'entraînement

