الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieur
Et de la Recherche Scientifique
المدرسة الوطنية العليا للتكنولوجيا
Ecole Nationale Supérieure de Technologie
Département de Génie Industriel et Maintenance

Mémoire de fin d'étude en vue de l'obtention du diplôme

D'INGÉNIEUR d'État

-Filière-

ÉLECTROMÉCANIQUE

-Spécialité-

MÉCATRONIQUE

-Thème-

Conception d'un bras de tri de médicaments basée sur le système d'exploitation de robot (ROS) et de réseaux de neurones de convolution destiné à une ligne de production automatisée.

-Réalisé par-

BOUDERSAYA Aissa Zakaria - **ZENOUNE Mohamed Aimen**

Les membres de Jury :

Mr. Noureddine DJEGHRI	Président
Mr. Ikhlas MEDDOUR	Promoteur
Mr. Boualem SI-AHMED	Co-promoteur
Mr. Adel MAROC	Co-promoteur Entreprise SAIDAL-SPA
Mr. Samir KHELOUATE	Examinateur

Alger, le 26/06/2023

Année universitaire 2022 –2023

Dédicace

“ 5 ans déjà ? tout début a une fin, et pour cette fin je me permets d'écrire quelques mots sur ces 5 ans d'études, non plutôt, 17 ans d'études ! C'était un très long chapitre de ma vie, un chapitre qui n'aurait pas eu lieu ainsi que la suite des chapitres de ma vie sans Allah tout puissant et les efforts des êtres les plus chers à mon cœur, je parle bien évidemment de mon cher père ***RACHID*** *et ma chère maman* ***DJAMILA****. Ma vie, ce travail, ainsi que tous mes futurs travaux inchalah sont dédiés à vous, tout ce que je fais est pour vous. Faudrait bien que je m'arrête sinon je vais écrire un livre et des chapitres que pour vous.*

Je dédie ce travail à mon cher frère ***RASSIM****, je te souhaite un joyeux évènement canon ; Mes amis les plus proches à mon cœur* ***AMINE****,****SALAH****,* ***CHAKIB****,****RYAD****,* ***RAYAN****,****ADEL*** *et bien sûr sans oublier mes chers BlinBOYS,* ***RIDA*** *et* ***AYMEN****. Mon ami* ***YANIS*** *le futur dentiste. Vous étiez tous les meilleurs frères qu'un fils unique peut avoir, je vous remercie tous pour votre soutiens et encouragement.*

À bien sûr mes frères du sang ***NADJIB*** *et* ***HAMIDOU****, ma grande cousine* ***YASMINE*** *et le petit* ***FARES****, ainsi que toute ma famille, la famille* ***ZENOUNE*** *et* ***LAITOUTI****.*

Je n'oublie pas aussi de citer ***MALIK*** *et toute l'équipe d'El 3azawiza. Et bien sûr mon cher comité du club ENTEINS.*

Le meilleur pour la fin, je dédie notre travail à toi aussi ***ZAKARIA*** *mon binôme, au moment où j'écris cette dédicace à 4h du matin, t'es en face moi en train de stresser concernant la présentation que je sais bien que tu vas assurer.*

Je dédie ce travail à moi-même, je suis tellement fier et confiant de ce que j'ai acquis, et je serai toujours aussi confiant de moi, ce n'est que le début d'un long parcours. C'était les premiers pas vers des buts plus grands, vers la satisfaction personnelle… Je serai toujours reconnaissant d'être l'unique fils d'une fière adorable maman et un fier grand homme, ***RACHID****, le fils d'un grand et unique* ***MILOUD ZENOUNE****.*

Je serai un modeste Aimen et je donnerai la parole à mon binôme. ”

-ZENOUNE MOHAMED AIMEN-

“ *Et maintenant, la parole m'a été donnée. La scène, éclairée par l'humilité de mon binôme : le Maître de Cérémonie, me voit maintenant, prêt à raconter mon propre chapitre. Après 17 ans de voyage à travers les mers de l'éducation, toujours à la barre, tenant bon contre vents et marées, il est enfin temps de mettre pied à terre et de récolter les fruits de ce long voyage.*

Sans le bon Dieu, le Tout-Puissant, sa grâce et sa guidance, je n'aurais pas pu naviguer sur ces mers. Mes chers parents, ***Youcef*** *et* ***Souhila****, ont été les véritables architectes de ce voyage, leur amour et leur soutien inébranlables ont été mon guide vers cette destination.*

Je ne peux passer, sous silence, la présence de ma petite sœur ***Asma****, dont les mots d'encouragement ont toujours traversé la porte qui sépare nos chambres, tels des rayons de lumière.*

Je tiens également à exprimer ma profonde gratitude à mon oncle ***Samir*** *et ma tante* ***Samia****, dont l'aide et les conseils ont été une boussole précieuse dans ce voyage. À mes cousins les plus adorés* ***Walid*** *et* ***Nadjat****.*

Je dédie ce travail à mes amis les plus chers à mon cœur : ***Sidali*** *et* ***Ramy****, qui ont toujours été à mes côtés apportant leur soutien lorsque j'en avais le plus besoin.* ***Lokmane****,* ***Nassim****,* ***Abdelghani****,* ***Oussama****,* ***Allaa****,* ***Raouf****,* ***Ismail****,* ***Mounir*** *et* ***Ziko****. Sans oublier les BlinBOYS* ***Rida****,* ***Aymen****,* ***Rayane*** *et* ***Zakaria****... **Chakib**,* ***Ryad*** *et toute l'équipe d'****El 3azawiza****.*

Dans le dernier chapitre de cette reconnaissance, un hommage doit être rendu à toi ***Aimen****, mon binôme, mon bras droit, mon coéquipier dans cette odyssée académique. Nous avons brassé les mers du sacrifice, labouré le champ du travail acharné pendant ces cinq mois.* ***Aimen****, à toi et à tes honorables parents, qui m'ont toujours ouvert leur maison avec une générosité, j'adresse ma gratitude la plus profonde. L'empreinte de votre contribution sur ce travail est indélébile et ne saurait être quantifiée.*

À moi-même, qui a bravé les obstacles et a tracé son propre chemin avec résilience et détermination. Et n'oubliez jamais que : ***MOULOUDIA est toujours forte****!* ”

-ZAKARIA AISSA BOUDERSAYA-

Remerciements

Nous tenons à exprimer notre plus profonde gratitude à Allah Tout-Puissant, qui nous a accordé la force, la foi, la volonté et la patience nécessaires à l'accomplissement de ce travail.

Nos pensées se tournent vers nos parents bien-aimés, qui ont été nos piliers, nous soutenant et nous encourageant tout au long de ce long voyage.

Un sincère remerciement à nos mentors respectés, **Mr. MEDDOUR Ikhlas**, **Mr. SI AHMED Boualem**, **Mr. MAROC Adel** et **Dr. TOUZOUT Walid**. Leur orientation et leur soutien ont été inestimables pour la réalisation de ce travail.

Nous exprimons notre reconnaissance envers toute l'équipe de SAIDAL SPA qui nous a chaleureusement ouvert ses portes pendant notre stage, contribuant à enrichir notre expérience et à leurs tête **Mr. HERAOUI Noureddine**.

Nos plus sincères remerciements vont également à l'équipe de professeurs de notre département de **Génie Industriel et Maintenance**, ainsi qu'à tous les enseignants et le personnel pédagogique de l'**ENST** qui nous ont guidés et soutenus pendant ces cinq années d'étude. Nous tenons particulièrement à remercier **Mme CHEGGOU Rabéa** et **Mme FERHAH Kamila** pour leur contribution précieuse.

Enfin, nous tenons à exprimer notre gratitude envers l'équipe d'Ocran Services, qui nous a généreusement accueillis dans ses bureaux, sous la direction bienveillante de **Mr. LAIFA Amine**.

Chaque personne mentionnée ici a joué un rôle fondamental dans notre parcours et pour cela, nous restons éternellement reconnaissants.

Résumé

Ce travail porte sur le développement d'un système de tri intelligent, avec pour objectif principal d'augmenter la précision d'un système existant. L'approche combine des techniques avancées de robotique et d'intelligence artificielle, en s'appuyant sur l'utilisation d'un réseau de neurones convolutifs (CNN) pour la reconnaissance des objets et en recourant au Robot Operating System (ROS) pour la modelisation, le contrôle et la simulation. L'aspect central du projet réside dans l'emploi d'un robot SCARA AZ-2000, un choix motivé par des critères spécifiques et soutenu par une intégration réussie avec un automate programmable industriel (API) et une Interface Homme-Machine (IHM). Cette étude méthodique et détaillée offre une référence précieuse pour des initiatives futures dans ce domaine, en montrant comment les technologies de pointe peuvent être utilisées pour optimiser les systèmes de tri automatisés.

Mots clés : Robotique, Apprentissage profond, Automate Programmable Industriel, Robot Operating System.

Abstract

This work focuses on the development of an intelligent sorting system, with the primary goal of enhancing the precision of an existing one. The approach combines advanced robotics and artificial intelligence techniques, relying on the use of a convolutional neural network (CNN) for object recognition and utilizing the Robot Operating System (ROS) for simulation. The project's central aspect lies in using a SCARA AZ-2000 robot, a choice driven by specific criteria and backed by successful integration with an Industrial Programmable Logic Controller (PLC). This methodical and detailed study provides a valuable reference for future initiatives in this field, demonstrating how cutting-edge technologies can be used to optimize automated sorting systems.

Keywords : Robotics, Deep learning, Programmable Logic Controller, Robot Operating System.

ملخص

يُركَز هذا العمل على تطوير نظام فرز ذكي، بهدف رئيسي هو زيادة دقة النظام الحالي. يجمع هذا النهج بين الروبوتات المتقدمة وتقنيات الذكاء الاصطناعي، باستخدام شبكة عصبية ملتفة (CNN) للتعرف على خصائص المنتوج و كذلك نظام تشغيل الروبوت (ROS) للمحاكاة. الجانب المركزي للمشروع هو استخدام روبوت AZ-2000 ،SCARA وهو اختيار حُدِّدَ بمعايير خاصة ويدعمه تكامل ناجح مع وحدة تحكم قابلة للبرمجة الصناعية .(PLC) توفر هذه الدراسة المنهجية والمفصلة مرجعًا قيمًا للمبادرات المستقبلية في هذا المجال، مما يوضح كيفية استخدام التقنيات المتقدمة لتحسين أنظمة خطوط الانتاج الآلية.

كلمات مفتاحية :

علم الروبوتات، تعلم عميق، وحدة تحكم منطقية قابلة للبرمجة، نظام تشغيل الروبوت.

Table des matières

Dédicace . . . . . **I**

Remerciements . . . . . **III**

Résumé . . . . . **IV**

Abstract . . . . . **V**

ملخص . . . . . **VI**

Introduction générale **1**

1 État de l'art **3**
1 Introduction 4
2 Perspectives théoriques sur l'industrie et l'automatisation 4
2.1 L'évolution de l'industrie 4
2.2 L'automatisation industrielle et l'automatisme : 6
2.3 Automate Programmable industriel 6
2.4 L'industrie pharmaceutique 7
3 Etude du système de tri SAIDAL 8
3.1 La chaîne de production SAIDAL 8
3.2 Principe de fonctionnement de la MARCHESINI MB451 EVO . . . 8
3.3 Le système contrôle de qualité 9
3.4 Limitation du système de tri 10
4 Conclusion 11

2 Robotique et outils de simulation **12**
1 Introduction 13
2 Robotique 13
2.1 Types de robots 13
2.2 Types de robots spécifiques 13
2.3 La robotique dans l'industrie pharmaceutique 14
3 Intelligence artificielle 15
3.1 Définition de l'intelligence Artificielle 15
3.2 Réseau de Neurones Convolutifs : 16
4 Robot Operating System (ROS) 18
4.1 L'Espace de travail ROS (ROS Workspace) 18
4.2 Packages ROS 19
4.3 Les nœuds ROS 19
4.4 Messages ROS 20

4.5 Maître ROS . . . 20
4.6 Logiciel Rviz . . . 20
4.7 Logiciel Gazebo . . . 20
4.8 ROS et la cinématique du robot . . . 21
4.9 MoveIt . . . 21
4.10 Les contrôleurs ROS . . . 22
5 Conclusion . . . 22

3 Analyse et conception du système . . . 23
1 Introduction . . . 24
2 Analyse fonctionnelle . . . 24
2.1 Besoin général . . . 24
2.2 Spécifications du besoin . . . 25
2.3 Analyse fonctionnelle externe . . . 26
2.4 Analyse fonctionnelle interne . . . 29
2.5 Exposition de la solution . . . 30
3 Mécanique . . . 32
3.1 Choix du modèle du robot . . . 32
3.2 Description, conception et présentation du modèle SCARA AZ-2000 . . . 32
3.3 Choix des matériaux . . . 34
3.4 Etude RDM . . . 35
3.5 Modélisation Géometrique . . . 37
3.6 Modélisation Cinématique . . . 41
3.7 Modélisation Dynamique . . . 43
3.8 Simulation sur MATLAB . . . 46
4 Génération de la loi de Commande . . . 51
4.1 Equation d'état du système . . . 51
4.2 Synthèse de la loi de commande . . . 52
4.3 Contrôleur PID . . . 52
5 Actionneurs et Capteurs . . . 53
5.1 Choix de type des moteurs . . . 53
5.2 Réducteur de vitesse . . . 57
5.3 Choix de type de vérin . . . 58
5.4 Caméra Intel RealSense . . . 58
5.5 Automate programmable et module NPU : . . . 59
5.6 Alimentation et puissance totale du système : . . . 60
6 Conclusion . . . 61

4 Contrôle, Automatisation et Maintenance du système . . . 62
1 Introduction . . . 63
2 Contrôle : Intelligence Artificielle . . . 63
2.1 Le modèle proposé . . . 63
2.2 Amélioration du modèle . . . 68
3 Automate Programmable . . . 71
3.1 Principe de fonctionnement . . . 71
3.2 Grafcet du système . . . 71
3.3 Équations de Set et Reset des bascules . . . 72
3.4 Table de Variables entrée /sortie . . . 73

3.5 Programmation en langage Ladder . . . 73
3.6 Interface Homme-Machine (Ihm) . . . 76
3.7 Simulation et visualisation du programme . . . 77
3.8 Schémas de Commande et de Puissance . . . 80
4 Gestion et organisation de la maintenance . . . 81
4.1 Maintenance Préventive . . . 81
4.2 Maintenance Corrective . . . 83
4.3 Gestion des pièces de rechange : . . . 84
5 Conclusion . . . 85

5 Simulation et Discussion . . . 86
1 Introduction . . . 87
2 Simulation de tri par IA . . . 87
2.1 Scénario #1 : Échantillon de validation . . . 87
2.2 Scénario #2 : Prédiction en temps réel . . . 88
2.3 Scénario #3 : Prédiction en temps réel d'autres médicaments . . . 88
2.4 Discussion : . . . 89
3 Simulation ROS . . . 89
3.1 L'espace de travail ROS . . . 89
3.2 Le graph des noeuds . . . 89
3.3 La cinématique directe et inverse sur ROS . . . 92
3.4 Les contrôleurs ROS . . . 92
3.5 Les scénarios de la simulation . . . 93
3.6 Tests de robustesse et de perturbation . . . 94
4 Résultats et Comparaison . . . 96
4.1 Discussion des résultats . . . 97
4.2 Avantages et Inconvénients . . . 97

Conclusion générale . . . 99

Table des figures

1.1 Un Automate Programmable Industriel [10] 6

2.1 Les différentes niveaux de l'intelligence artificielle [27] 16
2.2 Une simple architecture d'un réseau de neurones convolutifs 17

3.1 Paralgan-500-mg 24
3.2 Une plaquette Paralgan-500-mg 24
3.3 Diagramme bête à cornes 26
3.4 Diagramme de pieuvre 27
3.5 Diagramme FAST 29
3.6 Diagramme SADT A-0 30
3.7 Diagramme SADT A0 30
3.8 Modèle 3D du robot SCARA AZ-2000 33
3.9 composants mécaniques clés du SCARA AZ-2000 33
3.10 Vue 3D avec course du vérin 34
3.11 Résultat des contraintes appliquées sur le robot 36
3.12 Résultat de deformation sous l'effet les forces appliquées 36
3.13 Résultat du déplacement de la structure sous la contrainte appliquée 36
3.14 Schéma cinématique de robot SCARA AZ-2000 37
3.15 Schema des parametres DH robot SCARA 38
3.16 Model dessus sur le plan x,y 42
3.17 Schéma dynamique vue de dessus 44
3.18 Modélisation de SCARA AZ-2000 avec Simulink 47
3.19 Schéma d'implémentation de la cinématique directe 47
3.20 SCARA AZ-2000 sur la 1ère position 48
3.21 SCARA AZ-2000 sur la 2ème position 48
3.22 SCARA AZ-2000 sur la 3ème position 48
3.23 Schéma de calcul de la cinématique inverse sur Simulink 49
3.24 Configuration des coordonnées du point désiré et planification de la trajectoire dans une période de temps de 2s. 49
3.25 Point de départ T=0 50
3.26 Arrivée au point désiré T=1 50
3.27 Retour au point de départ T=2 50
3.28 Moteur Nema 23 56
3.29 Moteur nema 17 56
3.30 Driver NEMA 56
3.31 Le réducteur de vitesse 57
3.32 Réducteur de vitesse avec le moteur NEMA 57
3.33 SCARA AZ-2000 vue eclatée 57
3.34 vérin avec ventouse intégrée 58

3.35 Caméra Intel RealSense 59
3.36 Automate programmable S7-1500 60

4.1 Une plaquette défaillante 63
4.2 Une plaquette non-défaillante 63
4.3 Architecture de la partie convolutive 64
4.4 Architecture de réseau de neurones 64
4.5 Évolution de la précision du modèle CNN 67
4.6 Matrice de confusion des prédictions du modèle CNN 68
4.7 Architecture du modèle CNN amélioré : VGG16 69
4.8 Evolution de la précision du modèle CNN VGG16 70
4.9 Matrice de confusion des prédictions du modèle CNN VGG16 70
4.10 Le grafcet de commande du robot SCARA AZ-2000 72
4.11 La table des variables entrée/sortie 73
4.12 Réseaux 1,2 et 3. 74
4.13 Réseaux 4, 5 et 6. 74
4.14 Réseaux 7 et 8. 74
4.15 Réseaux 9 et 10. 75
4.16 Réseaux 11 et 12 75
4.17 Réseau 13. 75
4.18 Vue racine de l'interface Homme-Machine de SCARA AZ-2000 76
4.19 Page 1 de l'interface Homme-Machine de SCARA AZ-2000 77
4.20 Page 2 de l'interface Homme-Machine de SCARA AZ-2000 77
4.21 Page 3 de l'interface Homme-Machine de SCARA AZ-2000 77
4.22 Page 4 de l'interface Homme-Machine de SCARA AZ-2000 77
4.23 Connexion API-IHM 77
4.24 Mode RESET 78
4.25 Activation X1 78
4.26 Activation Vérin + Timer 5s 78
4.27 Activation des Moteurs 79
4.28 Désactivation Vérin, Moteurs en Marche 79
4.29 Retour à la position initiale 79
4.30 arrêt d'urgence mode RESET 80
4.31 Schéma électrique du controle des moteurs à 2 sens de rotation 80
4.32 Pilotage eléctro-pneumatique du vérin à double effet 81
4.33 Démarche de la maintenance corrective 83

5.1 Une plaquette PARALGAN prédit non défaillante 87
5.2 Une plaquette PARALGANT prédit défaillante 87
5.3 Une plaquette PARALGAN prédit non défaillante en temps réel 88
5.4 Une plaquette PARALGAN prédit défaillante en temps réel 88
5.5 Une plaquette GLUCOPHAGE prédit non défaillante en temps réel 88
5.6 Une plaquette LAMIDAZ prédit défaillante en temps réel 88
5.7 Fichier URDF du robot 89
5.8 Fichier de lancement du robot 89
5.9 Les noeuds ROS utilisés 90
5.10 Graph des noeuds et topics constituans le système. 91
5.11 Le fichier de la cinématique du robot 92

5.12 Les Controleurs utilisés pour le SCARA AZ-2000 92
5.13 Visualisation du robot sur RVIZ 93
5.14 Visualisation du robot sur GAZEBO 93
5.15 Robot en mouvement position 1 93
5.16 Robot en mouvement position 2 93
5.17 Plaquette non défaillante -robot en repos- 94
5.18 Plaquette défaillante execusion du plan de mouvement 94
5.19 Retour vers le bac de recyclage 94
5.20 Les limites cartésiennes du SCARA AZ-2000 95
5.21 Génération de bruit 95
5.22 Les différentes trajectoires de test 96

Liste des tableaux

3.1 Fonctions et désignations du système 27
3.2 Cahier des charges fonctionnel et technique 28
3.3 Caractéristiques des composants du SCARA AZ-2000 34
3.4 Propriétés de l'Aluminium 6061 35
3.5 Tableau des paramètres D-H 38
3.6 Résultats de calcul 61

4.1 Architecture détaillé du modèle CNN proposé 66
4.2 Plan de la maintenance 82
4.3 Inventaire des pièces de rechange 84

Liste des sigles et acronymes

API	*Automate Programmable Indusrtiel*
PLC	*Programmable Logic Controller*
SCARA	*Selective Compliance Articulated Robot Arm*
IHM	*Interface Homme-Machine*
RGB	*Red, Green, Blue. (Rouge, Vert,Bleu)*
NPU	*Neural Processing Unit*
IA	*Intelligence Artificielle*
ANN	*Artificail Neural Network*
CNN	*Convolutional Neural Network*
VGG	*Visual Geometry Group*
ROS	*Robot Operating System*
RViz	*Robot Visualization*
URDF	*Unified Robot Description Format*
FAST	*Functional Analysis System Technique*
SADT	*(Structured Analysis and Design Technique*
AFNOR	*Association Française de NORmalisation*
AANN	*Auto-Associative Neural Network*
MDSVTF	*Modified Discrete-State Variable Transition Function*

Introduction générale

L'automatisation et l'intelligence artificielle sont désormais au cœur de nombreux débats dans l'industrie manufacturière. L'intérêt de ces technologies dans l'amélioration des systèmes de production est incontestable, mais l'intégration de ces innovations demeure un enjeu majeur et suscite des interrogations légitimes. C'est dans ce contexte que s'inscrit ce projet, visant la conception d'un système de tri intelligent basé sur l'apprentissage approfondi et ROS pour une ligne de production automatisée. Ce projet prend vie au sein de l'entreprise pharmaceutique SAIDAL SPA, un acteur majeur de l'industrie, qui offre un cadre optimal pour sa réalisation. SAIDAL SPA a généreusement ouvert ses portes, donnant l'occasion unique d'explorer de près les défis inhérents à une véritable chaîne de production. Cette immersion dans un environnement industriel réel, et l'opportunité de travailler étroitement avec des ingénieurs compétents et expérimentés, ont permis d'enrichir considérablement l'expérience et la compréhension des enjeux liés à l'implémentation de l'intelligence artificielle dans les systèmes manufacturiers.

Le projet s'articule principalement autour de deux enjeux centraux : Le premier réside dans l'illustration de la valeur ajoutée potentielle de l'intelligence artificielle dans le contexte des systèmes manufacturiers. Dans l'industrie, les solutions éprouvées et fiables, comme l'automate programmable industriel (API), sont souvent privilégiées par rapport à l'IA, perçue comme une technologie plus récente et parfois moins maîtrisée. Malgré les avancées indéniables de l'IA, la question persiste : pourquoi intégrer l'IA lorsque les API semblent accomplir la tâche de manière adéquate ?

Le second enjeu réside dans l'amélioration du système de tri des plaquettes de médicament actuel, basé sur un système Harlequin. L'objectif est d'y intégrer une capacité de prise de décision intelligente à base d'un algorithme d'Intelligence Artificielle, pour augmenter ses performances et améliorer sa précision. Ces éléments sont particulièrement cruciaux dans le domaine pharmaceutique où la marge d'erreur est quasiment inexistante.

Afin d'organiser ce travail, une structure en quatre chapitres a été établie :

Chapitre 1: État de l'art, une vue d'ensemble de l'état actuel du domaine de l'automatisation dans l'industrie manufacturière sera présentée, ainsi qu'une revue du système de production existant chez SAIDAL-SPA, une base théorique solide sera établie, sur laquelle se baseront les chapitres suivants.

Chapitre 2: Robotique et outils de simulation, un accent sera mis sur les aspects techniques et opérationnels de la robotique, ainsi que sur l'Intelligence Artificielle à travers les réseaux de neurones convolutifs (CNN). Il sera également question de comment jumeler ces deux aspects au sein d'outils de simulation, comme le Robot Operating System (ROS), pour optimiser le système de tri intelligent.

Chapitre 3: Analyse et conception du système, il sera consacré à l'identification des besoins et des exigences à la modélisation et à la conception du système en détail. Les défis techniques rencontrés et les solutions adoptées pour les surmonter seront mis en lumière.

Chapitre 4: Controle, Automatisation et Maintenance du système, Le 4ème chapitre sera entièrement dédié au controle du système à base d'Intelligence Artificielle, son automatisation, tout en gardant sa fonctionnalité selon des démarches de maintenance.

Chapitre 5: Simulation du système Le dernier chapitre sera entièrement dédié à la simulation du système sur ROS. Il décrira de façon exhaustive la manière dont le système a été simulé, les résultats obtenus, et offrira une analyse et une interprétation de ces résultats.

Chapitre 1

État de l'art

1 Introduction

Dans ce chapitre, nous explorons le domaine passionnant de l'industrie et de l'automatisation. À partir d'un aperçu global, nous décrivons l'évolution dynamique de l'industrie, avec un accent particulier sur l'automatisation industrielle et le rôle clé des Automates Programmables Industriels dans les systèmes manufacturiers. Nous plongeons ensuite dans le vif du sujet avec un focus sur l'industrie pharmaceutique, illustré par le cas de l'entreprise SAIDAL. Une exploration approfondie de leur système de tri nous permet de comprendre ses principes de fonctionnement, les aspects liés au contrôle de qualité et les limites potentielles du système.

2 Perspectives théoriques sur l'industrie et l'automatisation

L'industrie est généralement définie comme l'ensemble des activités économiques impliquées dans la production de biens. Il comprend le processus de transformation des matières premières en produits finis à l'aide de moyens de production tels que la main-d'œuvre, les machines et l'énergie. La définition inclut la fabrication et la construction, ainsi que certaines formes de services, tels que les services publics.

2.1 L'évolution de l'industrie

Le secteur industriel n'est pas à l'abri de la révolution numérique. Ainsi, le concept « d'Industrie du Futur » véhicule la vision d'une usine connectée, rendue flexible et intelligente par la mise en réseau des machines, des produits et des individus. Au-delà de cet aspect purement technique, la diffusion des outils numériques conduit à repenser les modes d'organisation, les stratégies et les modèles économiques des entreprises. Ce bouleversement profond nécessite une réponse appropriée des pouvoirs publics pour accompagner la transformation du secteur industriel.

2.1.1 Industrie 4.0

L'industrie 4.0, également connue sous le nom de quatrième révolution industrielle, fait référence à la transformation numérique de la fabrication, caractérisée par l'intégration de plusieurs technologies telles que :

a. Internet des objets (IoT)

l'IoT est un élément clé de l'industrie 4.0. Cela permet aux machines, systèmes et équipements de collecter et d'échanger des données en temps réel, améliorant ainsi l'efficacité, la productivité et la qualité de la production. [1]

b. Intelligence artificielle (IA) et apprentissage automatique

l'IA et l'apprentissage automatique permettent aux machines d'apprendre de l'expérience, d'ajuster les dépenses en fonction de nouvelles données et d'effectuer des tâches qui nécessiteraient autrement un apprentissage. Dans l'industrie 4.0, ces technologies seront utilisées

pour l'automatisation des processus, la maintenance prédictive, la qualité des produits et l'optimisation de la chaîne d'approvisionnement. [2]

c. Cybersécurité

Avec la connectivité et l'interdépendance croissantes des systèmes, la cybersécurité est devenue une préoccupation majeure. Les organisations doivent mettre en œuvre des mesures de sécurité robustes pour protéger leurs systèmes contre les cyberattaques. [3]

L'industrie 4.0 offre des avantages significatifs, notamment une productivité accrue, une efficacité accrue, des coûts réduits et la fourniture d'une personnalisation étendue. Cependant, ils sont également confrontés à des défis tels que la nécessité d'investir dans de nouvelles technologies, les risques de cybersécurité, les problèmes de confidentialité et le besoin de nouvelles compétences pour les employés. [3]

2.1.2 Industrie 5.0

L'industrie 5.0 est la prochaine étape de la révolution industrielle, après l'industrie 4.0, qui s'est concentrée sur l'automatisation, la connectivité et les systèmes cyber-physiques. L'industrie 5.0 vise à combiner l'ingéniosité humaine avec l'efficacité de l'automatisation avancée. C'est la vision de l'industrie des humains et des machines travaillant en harmonie pour construire une société plus équilibrée et durable. [4]

a. Interaction homme-machine

L'industrie 5.0 se concentre sur l'interaction et la collaboration homme-machine. Contrairement à l'industrie 4.0, qui se concentrait sur l'automatisation complète des processus, l'industrie 5.0 verra les humains et les machines travailler ensemble de manière plus intégrée. Par exemple, les cobots (robots collaboratifs) travaillent aux côtés de travailleurs humains et sont conçus pour compléter plutôt que pour remplacer les compétences humaines. lution. [5]

b. Personnalisation de masse

Avec les progrès de l'intelligence artificielle et de l'impression 3D, l'Industrie 5.0 pourrait permettre une plus grande personnalisation de masse. Au lieu de produire en masse des produits uniformes, les industries peuvent adapter les produits aux préférences individuelles des consommateurs sans sacrifier l'efficacité de la production. [6] [7]

c. Durabilité

L'industrie 5.0 attache une grande importance à la durabilité écologique. En combinant innovation technologique et conscience humaine, l'Industrie 5.0 vise à créer des processus de production plus verts, à minimiser les déchets et à promouvoir une économie circulaire. [8]

d. Amélioration de la sécurité

L'intégration des systèmes cyber-physiques, de l'intelligence artificielle et de l'Internet des objets (IoT) dans l'industrie améliorera la sécurité des travailleurs. Par exemple, les robots peuvent effectuer des tâches dangereuses à la place des humains, réduisant ainsi le risque d'accidents du travail. [4]

2.2 L'automatisation industrielle et l'automatisme :

L'ère de l'industrie moderne est caractérisée par une convergence sans précédent de la technologie et de la production. À la pointe de ce mouvement se trouve l'automatisation, une transformation qui a radicalement changé la façon dont les processus industriels sont conçus et exécutés.

2.2.1 Automatisation industrielle :

L'automatisation industrielle est l'ensemble des technologies utilisant l'électronique, l'électrotechnique, la mécanique, la télécommunication afin de concevoir des machines ou des processus automatisés qui peuvent fonctionner sans intervention humaine. Elle est largement utilisée dans l'industrie manufacturière pour augmenter la productivité et la qualité des produits, tout en réduisant les coûts et les déchets. [9]

2.2.2 L'automatisme

L'automatisme, quant à lui, est un sous-ensemble de l'automatisation industrielle et qui fait référence à la qualité ou à l'état d'être autonome. Dans le contexte industriel, il décrit le fonctionnement indépendant d'une machine, d'un processus ou d'un système qui suit un ensemble prédéterminé d'instructions sans nécessiter d'intervention humaine continue. [9]

2.3 Automate Programmable industriel

Au cœur de cette évolution des systèmes industriels, les automates programmables industriels (API) occupent une place centrale et se distinguent comme l'élément clé de l'automatisation moderne.

2.3.1 Définition

L'Automate Programmable Industriel (API), ou Programmable Logic Controller (PLC), en anglais, est un appareil électronique programmable basé sur un microprocesseur adapté à l'environnement industriel, qui utilise une mémoire programmable pour stocker des instructions et qui réalise des fonctions d'automatisme (logique séquentielle et combinatoire, réglage, calcul arithmétique, comptage, décomptage, comparaison, temporisation, asservissement, régulation ...) pour assurer la commande de pré-actionneurs et d'actionneurs à partir d'informations logique, analogique ou numérique et contrôler des machines et des processus industriels. [10]

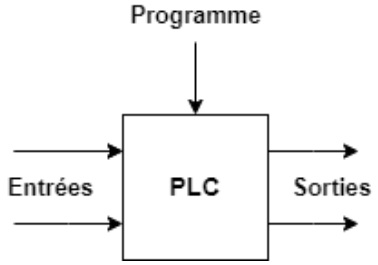


Fig. 1.1 : Un Automate Programmable Industriel [10]

Le terme logique est utilisé parce que la programmation est principalement concernée par la mise en œuvre de la logique et des opérations de commutation, par exemple :

Si A ou B se produit, allumez C ; si A et B se produisent, allumez D. Les dispositifs d'entrée (c'est-à-dire les capteurs tels que les interrupteurs) et les dispositifs de sortie (moteurs, vannes, etc.) dans le système contrôlé sont connectés à l'API. L'opérateur saisit alors une séquence d'instructions, un programme, dans la mémoire de l'API. Le contrôleur surveille ensuite les entrées et les sorties en fonction de ce programme et exécute les règles de contrôle pour lesquelles il a été programmé.

2.3.2 Avantages

Les Automates Programmables sont :

1. Robustes et conçus pour résister aux vibrations, à la température, à l'humidité et au bruit.[11]

2. Facilement programmables et disposent d'un langage de programmation facilement compréhensible qui est principalement concerné par la logique et les opérations de commutation.[11]

3. Grace à leur capacité à automatiser et à synchroniser divers processus industriels, les API peuvent augmenter considérablement l'efficacité et la productivité des opérations de production.[11]

2.4 L'industrie pharmaceutique

L'industrie pharmaceutique est un secteur majeur de l'économie mondiale dédié à la recherche, au développement, à la production et à la commercialisation de produits pharmaceutiques. Il joue un rôle important dans la promotion de la santé publique en fournissant des médicaments pour la prévention, le traitement et la guérison des maladies. [12]

2.4.1 Aspects de l'industrie pharmaceutique

L'industrie pharmaceutique a plusieurs aspects [13] :

- Recherche et développement
- Production
- Commercialisation
- Réglementation

2.4.2 L'industrie pharmaceutique en Algerie

En 2021, l'Algérie était l'un des plus grands marchés pharmaceutiques d'Afrique, et le gouvernement algérien a mis l'accent sur le développement de cette industrie comme une priorité. [14]

L'Algérie a adopté une politique de substitution aux importations dans le but de favoriser la production locale de médicaments et de réduire la dépendance vis-à-vis des importations. Plusieurs multinationales pharmaceutiques ont établi des usines en Algérie pour produire des médicaments localement. Les entreprises algériennes, comme Saidal, ont également une part importante du marché.

3 Etude du système de tri SAIDAL

3.1 La chaîne de production SAIDAL

La production de médicaments sous forme de comprimés est un processus complexe qui doit respecter des normes de contrôle de la qualité très strictes pour garantir l'efficacité et la sécurité du produit final. Voici les étapes générales impliquées dans ce processus :

1. Préparation des comprimés
2. Pesée des ingrédients
3. Mélange de la recette à travers le mélangeur DIOSNA p800
4. Séchage du mélange à travers le Sécheur DIOSNA cap600
5. Calibrer le mélange à travers le Calibreur frewitt sm d660
6. Compression du mélange

Une fois la compression des comprimés est achevée, et les comprimés sont prêts à être emballés, ces derniers sont transportés vers la machine **Marchesini MB451 Evo**, une machine à thermoformage, qui empaquette soigneusement les comprimés dans des blisters. Cette transition fluide de la compression à l'emballage assure une efficacité maximale et une qualité constante tout au long du processus de production.

3.2 Principe de fonctionnement de la MARCHESINI MB451 EVO

Les machines Blistereuses, souvent appelés thermo-formeuses sont largement utilisées dans l'industrie pharmaceutique pour emballer des produits tels que des comprimés, des gélules et d'autres produits médicaux. Le processus d'emballage sous blister fournit une barrière protectrice contre les influences environnementales tout en garantissant un dosage précis au patient.

Dans le cadre de notre étude, nous allons mettre l'accent sur une pièce maîtresse de l'innovation et de l'efficacité dans le domaine de l'emballage pharmaceutique : la Marchesini MB451 EVO [15]. Cette machine Blistéreuse offre des solutions d'emballage sophistiquées, conçues pour répondre aux exigences rigoureuses de l'industrie pharmaceutique.

L'un des aspects les plus importants de l'emballage pharmaceutique est le processus de thermoformage, qui permet de créer les blisters dans lesquels les médicaments (comprimés) sont emballés. Le thermoformage est un processus méticuleux qui doit être exécuté avec précision pour assurer la qualité et l'intégrité du produit final. C'est à cette étape clé que notre machine à blister Marchesini MB451 EVO démontre réellement ses capacités.
Examinons de plus près les différentes étapes de ce processus :

1. Chauffage de la matière
2. Mise en contact avec le moule et moulage
3. Aspiration
4. Refroidissement
5. Démoulage

3.3 Le système contrôle de qualité

Le contrôle de qualité [16] des plaquettes de médicaments est un processus trés important dans l'industrie pharmaceutique, qui fait appel à l'utilisation du système associé à des spécifications rigoureuses en termes de qualité, un système de caméra Harlequin capable de gérer diverses tâches de contrôle. Ce processus personnalisé est conçu pour répondre aux exigences de production internes et pour assurer que chaque plaquette de médicaments respecte des critères stricts.

- **Description générale du principe du fonctionnement du système Harlequin**

Le système Harlequin [17] est un système automatisé d'inspection et de tri de plaquettes de médicaments. Ce système est composé de deux sous-systèmes principaux : une caméra d'inspection et un système de contrôle. Le fonctionnement de ce système repose sur la détection de défauts par la caméra et la prise de décision par le système de contrôle.

1. La caméra :
Elle sert à inspecter les plaquettes de médicaments qui passent devant elle et elle compare chaque plaquette avec une plaquette de référence qui est considérée comme 100% intacte. La caméra est capable de détecter divers défauts tels que l'absence de comprimés, la présence de bulles, etc. Les données de cette inspection sont ensuite transmises au système de contrôle.

2. Le système de contrôle :
Traite les informations collectées par la caméra et prend des décisions basées sur ces informations. Si une plaquette est jugée défectueuse, le système de contrôle envoie un signal à l'API, qui à son tour active un actionneur pour éliminer la plaquette défectueuse de la chaîne de production.

3. Paramétrage des Contrôles :
La programmation des différents contrôles se fait selon une approche unique. Chaque article est en fait un conteneur d'objets de mesure : caractères, codes, etc.

Pour programmer un contrôle, l'utilisateur doit créer l'objet de mesure correspondant et une fenêtre délimitant la région d'image où le contrôle doit être effectué.

La fenêtre est ensuite liée à l'objet de mesure. Il est à noter qu'un seul objet peut être associé à plusieurs fenêtres qui effectuent le même contrôle à différents endroits.

Chaque fenêtre est également dotée d'une queue de rejet qui mémorise les résultats du contrôle. Cette queue est utilisée pour retarder l'extraction de l'information.

Lorsqu'une donnée est extraite de la queue, un signal de sortie est émis pour communiquer le résultat à un dispositif externe ou au PLC interne.

4. Critères et tolérance de Contrôle :

Dans le cadre de la mise en place d'un contrôle efficace, la configuration des tolérances et d'autres paramètres de mesure est indispensable. Ces éléments permettent de définir le niveau d'acceptabilité et d'exactitude requis dans le processus. Dans le système Harlequin, ces paramètres sont gérés comme suit :

1. Chaque plaquette de médicament doit être complète avec 10 comprimés (10/10). Toute déviation de ce ratio entraîne le rejet de la plaquette.

2. Aucun comprimé ne doit être cassé ou présent dans un état dégradé. L'intégrité de chaque comprimé est essentielle pour le passage du contrôle de qualité.

3. Il ne doit y avoir aucune bavure visible sur la plaquette.

4. Absence d'humidité : Toute présence d'humidité sur la plaquette entraîne le rejet de cette dernière.

3.4 Limitation du système de tri

Malgré l'efficacité et la technologie de pointe du système Harlequin, il n'est pas exempt de certaines limitations. Comme pour tout système, certains défis peuvent affecter le rendement et les performances. Ceci comprend :

- **Temps de traitement des images :**
Bien que le système Harlequin soit avancé, le temps de traitement des images peut être relativement long. Cela peut ralentir la chaîne de production, surtout si un grand nombre de plaquettes doit être traité en continu.
- **Précision :** La précision de détection du système Harlequin est généralement comprise entre 92% et 94%. Bien qu'il s'agisse d'une valeur raisonnable dans de nombreux domaines, ce niveau de précision peut ne pas être suffisant dans l'industrie pharmaceutique, qui a des normes très élevées. Même de petites erreurs peuvent avoir un impact significatif sur la qualité des médicaments. [17]
- **Configuration complexe :** La mise en place et le fonctionnement de système Harlequin nécessitent une configuration initiale, qui peut être complexe et qui prend du temps. Les paramètres du système de détection doivent être configurés pour chaque type de produit, ce qui nécessite beaucoup de temps et d'expertise technique.
- **L'inadaptation automatique aux changements de produits :** À chaque fois qu'un nouveau type de médicament doit être produit, le système Harlequin doit être reconfiguré à zéro. Cela est du au manque de la capacité d'apprentissage automatique comme un système basé sur le Machine Learning par exemple.
- **Compatibilité Big Data limitée :** Le système Harlequin peut rencontrer des problèmes pour gérer de grandes quantités de données. Cette limitation peut compliquer l'analyse des données à long terme ou l'identification des défaillances récurrents dans les plaquettes de medicament.
- **Grande consommation d'énergie :** Le système Harlequin, interconnecté avec l'API et l'actionneur de tri, consomme une énergie considérable. En plus d'augmenter les dépenses

et les coûts de production à cause des frais d'électricité, cette forte consommation laisse une empreinte carbone importante, contrastant avec des systèmes plus écoénergétiques.

4 Conclusion

Pour conclure, ce chapitre a permis d'explorer en profondeur le monde de l'industrie et de l'automatisation, en soulignant l'importance croissante de l'automatisation dans le contexte industriel moderne. À travers l'étude du système de tri de l'entreprise pharmaceutique SAIDAL, nous avons pu comprendre comment ces principes sont appliqués dans la pratique et quels sont les défis et limites inhérents. Cette compréhension nous permettra d'appréhender plus précisément les opportunités d'innovation et d'amélioration dans le domaine de l'automatisation industrielle

Chapitre 2

Robotique et outils de simulation

1 Introduction

Ce chapitre nous fait entrer dans le monde fascinant de la robotique et des outils de simulation. Nous allons aborder les divers types de robots, en mettant un accent particulier sur leur utilisation dans l'industrie pharmaceutique. Ensuite, l'intelligence artificielle et son rôle clé dans la robotique seront discutés. Enfin, le Robot Operating System (ROS) sera étudié de manière exhaustive, incluant les divers aspects de son utilisation et son impact sur la cinématique du robot.

2 Robotique

La robotique est un domaine interdisciplinaire qui intègre des aspects de l'ingénierie mécanique, électrique et informatique pour la conception, la construction, l'exploitation et l'application de robots. Les robots sont généralement utilisés pour effectuer des tâches difficiles, dangereuses ou impossibles pour les humains. À mesure que la technologie progresse, l'impact de la robotique sur nos vies quotidiennes et nos environnements de travail continue de croître.

2.1 Types de robots

La robotique est un domaine très vaste, avec de nombreux types de robots différents conçus pour des applications spécifiques. exposons quelques-unes des principales catégories :

- Robots industriels
- Robots de service
- Robots mobiles
- Robots humanoïdes
- Robots médicaux
- Robots militaires
- Robots de recherche

2.2 Types de robots spécifiques

- Robots Cartésiens
- Robots Delta
- Robots Cylindriques
- Robot SCARA

Si nous comparons ces types de robots avec les types ci-dessus, nous pouvons voir qu'il y a un certain chevauchement. Par exemple, les robots SCARA, les robots cartésiens, les robots Delta et les robots cylindriques sont des sous-types de robots industriels, chacun avec des conceptions spécifiques adaptées à des tâches et des environnements spécifiques.

2.3 La robotique dans l'industrie pharmaceutique

L'industrie pharmaceutique utilise largement la robotique pour améliorer l'efficacité, la précision et le respect de normes de qualité strictes. Nous donnons ici un aperçu des applications robotiques, des exigences, des normes et des capacités dans ce domaine.

2.3.1 Application :

Les robots industriels sont utilisés dans divers application dont :

• **Fabrication et emballage :** Les robots sont souvent utilisés pour manipuler, assembler et emballer des produits pharmaceutiques. Ils sont capables d'effectuer des tâches répétitives avec plus de précision et de rapidité que les humains, améliorant ainsi la productivité. [18]

• **Découverte de médicaments :** Des robots sont également utilisés pour automatiser le criblage à haut débit. Le criblage à haut débit est une technique utilisée pour tester rapidement des milliers de composés dans la découverte de médicaments. Les robots peuvent manipuler des micro plaques multipuits et introduire des réactifs chimiques avec une grande précision.[19]

2.3.2 Exigences et normes

L'industrie pharmaceutique est très réglementée et la robotique doit répondre à de nombreuses normes et exigences :

• **Bonnes pratiques de fabrication (BPF) :** les robots doivent respecter les normes de bonnes pratiques de fabrication (BPF), qui définissent les exigences de fabrication des produits pharmaceutiques . Cela comprend les normes d'hygiène, de traçabilité et de contrôle de la qualité. [20]

• **Validation des robots pharmaceutiques :** avant qu'un robot puisse être utilisé pour la production pharmaceutique, il doit passer par un processus de validation rigoureux. Cela inclut la validation que le robot se comporte comme prévu et peut effectuer des tâches de manière reproductible et fiable. [21]

• **Précision :** Les robots utilisés dans l'industrie pharmaceutique doivent être très précis pour garantir la qualité et la cohérence des produits.

• **Flexibilité :** La flexibilité est également une caractéristique importante. Les robots doivent pouvoir s'adapter à différents formats de produits et de processus, car l'industrie pharmaceutique doit souvent modifier sa production pour répondre à la demande du marché.

• **Hygiène :** En raison de la sensibilité des produits pharmaceutiques, les robots doivent assurer un haut niveau d'hygiène. Cela peut inclure l'utilisation de matériaux et de conceptions spécifiques qui minimisent le potentiel de contamination. [22]

3 Intelligence artificielle

Dans un monde de plus en plus interconnecté et axé sur la technologie, l'Intelligence Artificielle (IA) est devenue un élément clé de la résolution de problématiques complexes. Elle a conduit à une transformation radicale de notre monde et a initié une nouvelle ère pour l'industrie manufacturière. Aujourd'hui, l'IA, en s'appuyant sur des approches comme les réseaux de neurones, offre des solutions inédites pour surmonter des défis qui semblaient autrefois insurmontables.

3.1 Définition de l'intelligence Artificielle

L'Intelligence Artificielle (IA) est un sous-domaine clé de l'informatique qui s'intéresse à la conception et à la création de systèmes capables de mimer les capacités cognitives humaines [23]. En termes plus techniques, l'IA aspire à concevoir des algorithmes et des modèles qui, lorsqu'ils sont exécutés par une machine, permettent cette dernière de réaliser des tâches requérant normalement l'intelligence humaine. Ces tâches peuvent inclure l'apprentissage (via l'application de fonctions d'optimisation comme la descente de gradient), le raisonnement (à travers des techniques d'inférence logique), la résolution de problèmes (par le biais d'algorithmes de recherche et d'optimisation) et la prise de décision.

L'IA a une portée vaste et significative, s'étendant à de nombreux secteurs tels que la santé, la finance, l'éducation et le commerce et également une application majeure dans la robotique, où elle optimise les opérations, améliore l'interaction homme-robot et conduit à des applications robotiques plus efficaces et innovantes, notamment l'automatisation des processus, en libérant le personnel humain de tâches répétitives et en augmentant l'efficacité opérationnelle. [24]

Par ailleurs, elle est essentielle pour la prise de décision, fournissant des analyses prédictives basées sur des techniques telles que les arbres de décision, et les algorithmes de régression. De plus, elle se distingue par sa capacité à apprendre et à s'adapter en fonction des besoins changeants de l'environnement, une caractéristique obtenue grâce à des méthodes d'apprentissage automatique et d'adaptation dynamique.

3.1.1 Machine Learning (ML) :

Le ML est un sous-domaine de l'Intelligence Artificielle qui se concentre sur le développement de systèmes informatiques qui peuvent apprendre de leurs expériences. En termes plus précis, le ML utilise des algorithmes et des modèles statistiques pour permettre aux machines d'améliorer leurs performances dans la réalisation de tâches spécifiques sans être explicitement programmées pour le faire. Cela se fait généralement en utilisant des données et des informations pour former un modèle de ML, qui peut ensuite être utilisé pour faire des prédictions ou prendre des décisions. [25]

3.1.2 Deep Learning :

Un sous-ensemble encore plus spécifique du Machine Learning est le Deep Learning. Le Deep Learning utilise des réseaux de neurones artificiels avec plusieurs couches - d'où le terme "deep". [26] Ces réseaux tentent d'imiter le fonctionnement du cerveau humain pour

traiter les données et créer des modèles pour la prise de décision. C'est une technologie clé pour de nombreuses applications d'IA, comme la reconnaissance vocale, la reconnaissance d'images, le traitement du langage naturel, et bien plus encore. Le Deep Learning a apporté d'importantes avancées et des améliorations significatives dans ces domaines.

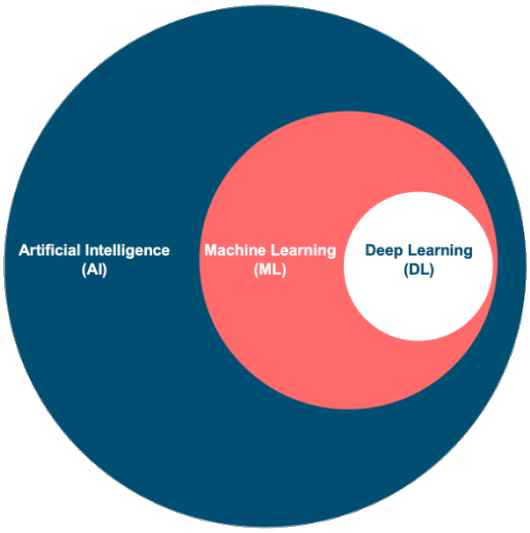


Fig. 2.1 : Les différentes niveaux de l'intelligence artificielle [27]

3.2 Réseau de Neurones Convolutifs :

Le réseau de neurones convolutifs, ou bien Convolutional Neural Network (CNNs) [28], est un type de Réseau de Neurones Artificiels (ANN) [29] conçu pour traiter des données structurées en grille, telles que les images. Les CNN sont particulièrement doués pour reconnaître les motifs dans les images, grâce à leur structure profonde et hiérarchique.

Les CNNs sont analogues aux réseaux de neurones artificiels traditionnels (ANNs) dans le sens où ils sont composés de neurones qui s'auto-optimisent par l'apprentissage. Chaque neurone reçoit toujours une entrée et effectue une opération (comme un produit scalaire suivi d'une fonction non linéaire). De l'entrée sous forme de vecteurs d'image bruts à la sortie finale du score de classe, l'ensemble du réseau exprime toujours une seule fonction de score perceptive (le poids). La dernière couche contiendra des fonctions de perte associées aux classes, et toutes les astuces régulières développées pour les ANNs traditionnels s'appliquent toujours. [30]

Architecture de CNN :

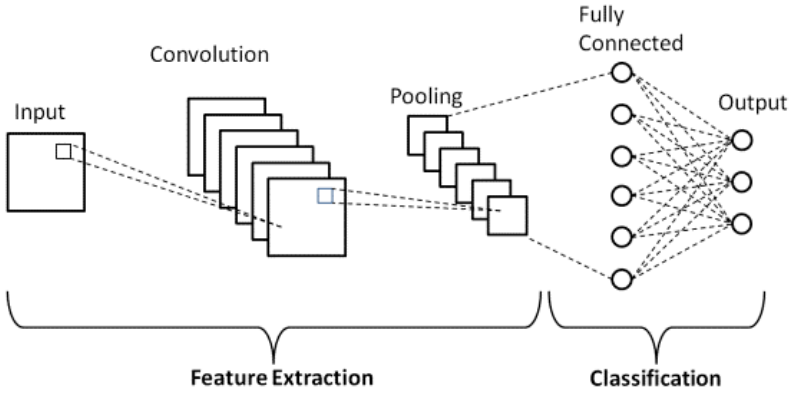


FIG. 2.2 : Une simple architecture d'un réseau de neurones convolutifs

a. Couche de Convolution (Convolutional layer) :

Cette couche effectue un produit matriciel entre deux matrices, l'une étant l'ensemble de paramètres apprenables connus sous le nom de noyau (kernel), et l'autre étant la portion limitée du champ récepteur. Le noyau est spatialement plus petit qu'une image mais est plus profond. Cela signifie que, si l'image est composée de trois canaux (RGB), la hauteur et la largeur du noyau seront spatialement petites, mais la profondeur s'étend jusqu'à tous les trois canaux. [31]

b. Couche de Pooling (Pooling layer) :

La couche de Pooling remplace la sortie du réseau à certains endroits en dérivant une statistique de synthèse des sorties proches. Cela aide à réduire la taille spatiale de la représentation, ce qui diminue la quantité de calcul et de poids requis. L'opération de Pooling est traitée sur chaque tranche de la représentation individuellement. Il existe plusieurs fonctions Pooling, comme la moyenne du voisinage rectangulaire, la norme L2 du voisinage rectangulaire, et une moyenne pondérée basée sur la distance depuis le pixel central. [31]

Cependant, le processus le plus populaire est le Max Pooling, qui rapporte la sortie maximale du voisinage.

c. Couche entièrement connectée (Fully connected layer) :

La couche entièrement connectée contient des neurones qui sont directement connectés aux neurones des deux couches adjacentes, sans être connectés à d'autres couches à l'intérieur de celles-ci. C'est analogue à la manière dont les neurones sont disposés dans les formes traditionnelles de ANN. [31]

d. Couche de Non-linéarité :

Comme la convolution est une opération linéaire et que les images sont loin d'être linéaires, des couches de non-linéarité sont souvent placées juste après la couche de convolution pour introduire de la non-linéarité dans la carte d'activation. [32]

Il existe plusieurs types d'opérations non linéaires, les plus populaires étant :

- **Sigmoïde** : La non-linéarité sigmoïde a la forme mathématique $\sigma(x) = \frac{1}{1+e^{-x}}$. Elle prend un nombre à valeur réelle et le "compresse" dans une plage entre 0 et 1.
- **Tanh** : Le Tanh comprime un nombre à valeur réelle dans la plage [-1, 1]. Comme la sigmoïde, l'activation sature, mais - contrairement aux neurones sigmoïdes - sa sortie est centrée sur zéro.
- **ReLU** : L'unité linéaire rectifiée (ReLU) est devenue très populaire ces dernières années. Elle calcule la fonction f(x)=max (0,x). En d'autres termes, l'activation est simplement seuillée à zéro.

4 Robot Operating System (ROS)

Robot Operating System, ou bien ROS est un système de logiciel robotique open source qui peut être utilisé sans frais de licence par des universités, des agences gouvernementales ou n'importe quelle entreprise commerciale. Les avantages d'un logiciel open source résident dans le fait que le code source du système est accessible et peut être modifié selon les besoins d'un utilisateur. Par conséquent, le logiciel peut être amélioré et des modules peuvent être ajoutés par les utilisateurs ou n'importe quelle équipe de recherche. [33]

ROS est un méta-système d'exploitation capable de remplir de nombreuses fonctions d'un système d'exploitation, il nécessite cependant un système d'exploitation d'ordinateur comme Linux. Les principaux objectifs de l'utilisation de ROS sont de fournir une communication entre l'utilisateur, le système d'exploitation de l'ordinateur et le matériel externe. Ce matériel peut comprendre : des capteurs, des caméras ou n'importe quel périphérique robotique. Comme avec n'importe quel système d'exploitation, l'avantage de ROS est l'abstraction matérielle et sa capacité à contrôler un robot sans que l'utilisateur n'ait à connaître tous les détails de ce robot. Par exemple, pour déplacer les bras du robot, une commande ROS est émise, ou un script Python ou C++ basé sur ROS est écrit par les concepteurs du robot pour que le robot réponde comme commandé. Les scripts peuvent à leur tour appeler divers programmes de contrôle qui provoquent le mouvement réel des bras du robot. Il est également possible de concevoir et de simuler n'importe quelle application robotique complexe à l'aide de ROS.

4.1 L'Espace de travail ROS (ROS Workspace)

Un espace de travail ROS est un espace sur votre système qui est organisé de manière spécifique et contient des packages ROS. En termes plus techniques, un espace de travail ROS est une collection de dossiers et de fichiers suivant une structure spécifique. C'est l'endroit où l'utilisateur peut créer, modifier et gérer des packages ROS. [34]

La structure typique d'un espace de travail ROS inclut les dossiers suivants :

- **Src :** Il s'agit du dossier de classement de packages ROS avant de les compiler.
- **build :** C'est là que se produit la compilation. Il est automatiquement généré lorsque vous compilez vos packages.
- **devel :** C'est là où les fichiers exécutables sont placés après la compilation. Il contient aussi des scripts qui préparent l'environnement pour l'utilisation des packages compilés.

4.2 Packages ROS

Un package ROS est le principal moyen d'organisation du logiciel dans ROS. Il peut contenir des nœuds ROS, des bases de données, des fichiers de configuration, des algorithmes, des bibliothèques externes, etc. Les packages ROS sont principalement utilisés pour organiser les fonctionnalités liées de manière à faciliter le partage et la distribution.

Un package ROS typique contient les éléments suivants :
- CMakeLists.txt : Un fichier nécessaire pour utiliser "CMake" pour la compilation.
- Package.xml : Un fichier qui fournit des métadonnées sur le package, telles que le nom du package, sa version, son auteur, sa licence et ses dépendances sur d'autres packages ROS.
- Fichiers source de code (comme .cpp pour C++ ou .py pour Python) : Ces fichiers contiennent le code qui est exécuté lors de l'exécution de nœuds ROS spécifiques.
- Fichiers de lancement (. Launch) : Ils sont utilisés pour lancer plusieurs nœuds à la fois et configurer leurs paramètres.

4.3 Les nœuds ROS

En gros, les nœuds sont des processus qui effectuent une action. Les nœuds sont en réalité des modules logiciels, mais ils ont la capacité de s'enregistrer auprès du nœud Maître ROS et de communiquer avec d'autres nœuds dans le système. L'idée derrière la conception de ROS est que chaque nœud est un module indépendant qui interagit avec les autres nœuds en utilisant les protocoles de communication de ROS. Les nœuds peuvent être créés de différentes manières : ils peuvent être générés directement à partir d'une fenêtre de terminal Linux en tapant des commandes spécifiques. En alternative, les nœuds peuvent être créés dans le cadre d'un programme écrit en Python ou en C.

- **Publication et Abonnement des nœuds**

L'une des forces de ROS est qu'une tâche spécifique, comme le contrôle d'un robot, peut être divisée en une série de tâches beaucoup plus simples. Ces tâches peuvent inclure la perception de l'environnement à l'aide d'une caméra ou d'un scanner laser, la création de cartes, la planification d'itinéraires, la surveillance du niveau de batterie, etc. Chacune de ces actions peut consister en un nœud ROS ou une série de nœuds pour accomplir les tâches spécifiques.

Un code de nœud ROS peut être exécuté indépendamment pour réaliser sa tâche ; de plus, il peut communiquer avec d'autres nœuds en envoyant ou en recevant des messages. Les messages sont organisés en catégories spécifiques appelées "topics", qui peuvent consister en des données, des commandes ou d'autres informations nécessaires à l'application en cours de développement. Par convention, lorsqu'un nœud ROS envoie des messages à un "topic", cela signifie que le nœud publie sur ce "topic". Par contre, lorsque le nœud ROS reçoit des messages d'un "topic", cela signifie qu'il s'abonne à ce "topic". Par exemple, la caméra fournie avec un robot est associée à un nœud ROS qui publie régulièrement l'image sur un "topic" particulier. Ce "topic" peut être utilisé par un autre nœud qui affiche l'image sur l'écran de l'ordinateur. On dit que le nœud qui reçoit l'information s'abonne au "topic" qui est publié.

Un concept clé dans la communication ROS est le mécanisme de publication / abonnement, qui est un modèle de communication asynchrone. Cela permet la décentralisation et

la décomposition des problèmes complexes en sous-problèmes plus simples qui peuvent être résolus indépendamment.

4.4 Messages ROS

Les messages ROS sont définis par le type de message et le format des données. Par exemple, le package ROS nommé (std msgs) contient des messages de type String qui sont constitués d'une chaîne de caractères. D'autres packages de messages pour ROS contiennent des messages utilisés pour la navigation robotique ou les capteurs robotiques. Dans notre travail, les types de messages que nous avons utilisés sont les messages d'odométrie contenant le vecteur de position de l'USV ainsi que son vecteur de vitesse.

4.5 Maître ROS

Les nœuds ROS sont généralement de petits programmes indépendants qui peuvent s'exécuter simultanément sur plusieurs systèmes. La communication est établie entre les nœuds par le Maître ROS. Le Maître ROS fournit des services de nomination et d'enregistrement aux nœuds du système ROS. Il suit les éditeurs et les abonnés des sujets. Le rôle du Maître est de permettre aux nœuds ROS individuels de se localiser mutuellement.

4.6 Logiciel Rviz

Rviz, qui est l'abréviation de ROS visualization, est un puissant outil de visualisation 3D pour ROS. Il permet à l'utilisateur de voir le modèle de robot simulé, d'enregistrer les informations des capteurs du robot et de rejouer les informations des capteurs enregistrées. En visualisant ce que le robot voit, traite et fait, l'utilisateur peut déboguer une application robotique, depuis les entrées du capteur jusqu'aux actions planifiées (ou non).

- **URDF file :**

Les modèles de robots basés sur Rviz peuvent être créés en utilisant le langage URDF ou, Unified Robot Description Language. Ce dernier est un format XML spécifiquement défini pour représenter les modèles de robots jusqu'à leur niveau de composant. Ces fichiers URDF peuvent devenir longs et encombrants sur des systèmes de robots complexes.

4.7 Logiciel Gazebo

Gazebo est un environnement de simulation robotique gratuit et open source développé par Willow Garage. En tant qu'outil multifonctionnel pour les développeurs de robots ROS, Gazebo offre les fonctionnalités suivantes [35] :

- Conception de modèles de robots.
- Prototypage rapide et test d'algorithmes.
- Test de régression utilisant des scénarios réalistes.
- Simulation d'environnements intérieurs et extérieurs.

- Simulation de données de capteurs.

L'interface utilisateur de Gazebo est similaire à celle de rviz à bien des égards. La fenêtre centrale fournit la vue pour l'environnement mondial 3D de Gazebo. La grille est généralement configurée pour être le plancher de l'environnement sur lequel tous les modèles sont maintenus à cause de la gravité dans l'environnement. [36]

4.8 ROS et la cinématique du robot

Le ROS est intégré aux mouvements robotiques, abordant les défis complexes de la cinématique, la dynamique et aussi les tests de robustesse et de perturbation. [33]

4.8.1 Cinématique directe et inverse

Dans ROS, il existe plusieurs packages qui peuvent aider à la réalisation de la cinématique directe et inverse. Un exemple est le package MoveIt ! qui est une suite logicielle pour la planification du mouvement, la cinématique, la perception 3D et bien plus encore. Il fournit une interface pour la cinématique inverse (IK) et directe (DK) pour de nombreux robots. Le package KDL, Kinematics and Dynamics Library en anglais, fournit également des outils pour la cinématique directe et inverse.

4.8.2 Dynamique du robot

La dynamique du robot concerne le mouvement du robot et les forces qui le causent. Cela comprend l'étude de l'accélération, de la vitesse, de la position, des couples et des forces. ROS offre le package Gazebo, qui est un puissant simulateur physique qui peut simuler des robots, des environnements et des capteurs. Il est capable de simuler la dynamique complexe du robot, y compris les forces et les couples, l'inertie, la friction et bien plus encore.

4.8.3 Test de robustesse et de perturbation

En ce qui concerne les tests de robustesse et de perturbation, il s'agit d'évaluer comment le système robotique répond aux perturbations et aux conditions inattendues. Ceci est particulièrement important pour garantir que le robot fonctionnera correctement dans des environnements réels, qui sont souvent imprévisibles et dynamiques.

4.9 MoveIt

MoveIt est l'une des suites logicielles les plus utilisées dans le monde du développement de logiciels pour la robotique. Il est spécialement conçu pour la manipulation de bras de robot dans l'écosystème ROS. [33] MoveIt offre des fonctionnalités essentielles pour un grand nombre de tâches robotiques, telles que la planification du mouvement, la cinématique, la perception 3D, l'analyse de la manipulation d'objets, le contrôle du robot, et plus encore. Les fonctionnalités clés comprennent :

- **Planification du mouvement et contrôle du robot.**
- **La cinématique.**

- **Gestion des collisions.**
- **Intégration avec les capteurs.**
- **Manipulation d'objets.**
- **Visualisation et interface utilisateur.**

4.10 Les contrôleurs ROS

Pour la conception de contrôleurs robustes et adaptatifs dans ROS, divers packages peuvent être utilisés en fonction des besoins spécifiques de votre application. Ces packages peuvent être utilisés pour la simulation, le prototypage, la mise en œuvre et le test de vos algorithmes de commande. [37]

4.10.1 Ros control

Ros control est un framework qui fournit une architecture de contrôleur pour les robots ROS. Il fournit des interfaces pour contrôler les joints du robot, interagir avec les capteurs et actuateurs, et intégrer des contrôleurs personnalisés. Vous pouvez développer vos propres contrôleurs adaptatifs ou robustes et les intégrer dans ce framework.

4.10.2 Control toolbox

Control toolbox est un package ROS qui fournit des outils pour le développement de contrôleurs dans ROS. Il comprend des contrôleurs PID, des filtres numériques et d'autres outils utiles pour la conception de systèmes de contrôle.

4.10.3 Gazebo ROS control

Gazebo ROS control est un package qui intègre ROS control avec le simulateur Gazebo. Il vous permet de tester vos contrôleurs dans un environnement simulé avant de les déployer sur un robot physique.

De plus, des techniques telles que la commande adaptative et la commande robuste peuvent être utilisées pour concevoir des contrôleurs qui sont robustes face aux incertitudes et aux perturbations. Ces techniques peuvent être implémentées en utilisant les packages ROS appropriés.

5 Conclusion

En somme, ce chapitre a permis de naviguer à travers l'univers de la robotique et de l'intelligence artificielle, détaillant leur rôle et leur importance dans les industries modernes, notamment la pharmaceutique. L'étude approfondie du Robot Operating System (ROS) offre un aperçu précieux des mécanismes et outils qui permettent le fonctionnement et le contrôle des systèmes robotiques. Ainsi, ces connaissances seront cruciales pour comprendre et exploiter les potentialités offertes par la robotique dans des contextes industriels variés.

Chapitre 3

Analyse et conception du système

1 Introduction

Ce chapitre présente une transition significative, marquant le passage de la phase théorique à l'application pratique. Les fondements théoriques bien établis et les besoins spécifiques du système, préalablement identifiés dans les premiers chapitres, commencent à se concrétiser. Il met en avant une analyse détaillée et minutieuse des aspects pratiques essentiels, indiquant une voie vers la mise en œuvre effective du système de tri intelligent.

2 Analyse fonctionnelle

2.1 Besoin général

SAIDAL, le principal groupe pharmaceutique en Algérie, est reconnu pour sa contribution significative dans le domaine de la santé, garantissant la disponibilité de médicaments de haute qualité pour le marché algérien. Au cœur de son fonctionnement se trouve un système de tri et de contrôle qui assure la conformité de chaque plaquette de médicament **PARALGAN - Paracétamol** sortant de sa chaîne de production.

Le système actuel, bien que fonctionnel, a ses limites et exige donc une amélioration constante pour maintenir le groupe à la pointe de la technologie.

En effet, nous cherchons à concevoir une solution capable d'identifier, de trier et de séparer les plaquettes de médicaments qui présentent des défauts. Il est essentiel que cette solution puisse non seulement identifier les défauts, mais aussi séparer les produits défectueux de ceux qui respectent les normes de qualité. Cela permettra de garantir que seuls les médicaments de la plus haute qualité atteignent les patients, renforçant ainsi la réputation du groupe SAIDAL en tant que fournisseur fiable de médicaments de haute qualité.

2.1.1 Le produit à trier :



Fig. 3.1 : Paralgan-500-mg



Fig. 3.2 : Une plaquette Paralgan-500-mg

2.1.2 Caractéristiques du produit à trier :

- Dimensions : 100 x 40mm.
- Poids : 30.2 grammes.
- Nombre de comprimés : 10 comprimés par plaquette.

2.2 Spécifications du besoin

2.2.1 Détection de défauts :

La solution doit être capable de détecter efficacement les plaquettes de médicaments défectueuses.

2.2.2 Critères et tolérances de contrôle :

Le Système de controle doit respecter les Critères suivants :

1. Chaque plaquette de médicament doit être complète avec 10 comprimés (10/10). Toute déviation de ce ratio entraîne le rejet de la plaquette.

2. Aucun comprimé ne doit être cassé ou présent dans un état dégradé. L'intégrité de chaque comprimé est essentielle pour le passage du contrôle de qualité.

3. Il ne doit y avoir aucune bavure visible sur la plaquette.

4. Absence de bulls, de grains et de humidité.

2.2.3 Tri et séparation :

Une fois les plaquettes défectueuses détectées, la solution doit être capable de les trier et de les séparer des plaquettes qui sont en bon état du convoyeur et dans un espace très restreint.

2.2.4 Rapidité et précision :

La solution doit offrir une vitesse et une précision de tri au moins équivalentes à celles du système de tri actuellement en place. Les paramètres précis de vitesse et de précision sont définis dans le tableau du cahier des charges.

2.2.5 Conformité aux normes de sécurité :

La solution doit être conforme aux normes de sécurité en vigueur dans les industries pharmaceutiques :

a. Normes d'hygiène :
L'industrie pharmaceutique est soumise à des normes d'hygiène très strictes pour éviter toute contamination du produit.

b. Normes de sécurité électrique :
La solution doit respecter les normes de sécurité électrique pour éviter tout risque d'électrocution ou d'incendie. Par exemple, il doit être correctement mis à la terre, et tous les composants électriques doivent être correctement isolés.

c. Normes de sécurité mécanique :
La solution doit être conçu pour éviter tout risque d'accident mécanique, comme par exemple le pincement des doigts ou l'écrasement. Cela peut impliquer des protections physiques, mais aussi des systèmes de sécurité logiciels pour éviter les mouvements dangereux.

d. Normes de sécurité opérationnelle :
Le système doit être conçu pour être sûr à utiliser. Cela peut impliquer par exemple des systèmes d'arrêt d'urgence, des systèmes de détection d'obstacles, ou encore des systèmes de formation et d'information pour les opérateurs.

e. Normes d'ergonomie :
Pour prévenir les blessures dues à des mouvements répétitifs ou à des postures inconfortables, l'équipement doit être conçu de manière ergonomique. La solution doit être à une hauteur confortable pour les opérateurs, et les commandes doivent être facilement accessibles.

2.3 Analyse fonctionnelle externe

L'analyse fonctionnelle externe d'un projet représente la définition et l'expression du besoin de conception du système

3.3.1 Diagramme bête à cornes :

La figure 3.3 illustre le diagramme de la bête à cornes, mettant en lumière les interactions clés du système avec son environnement.

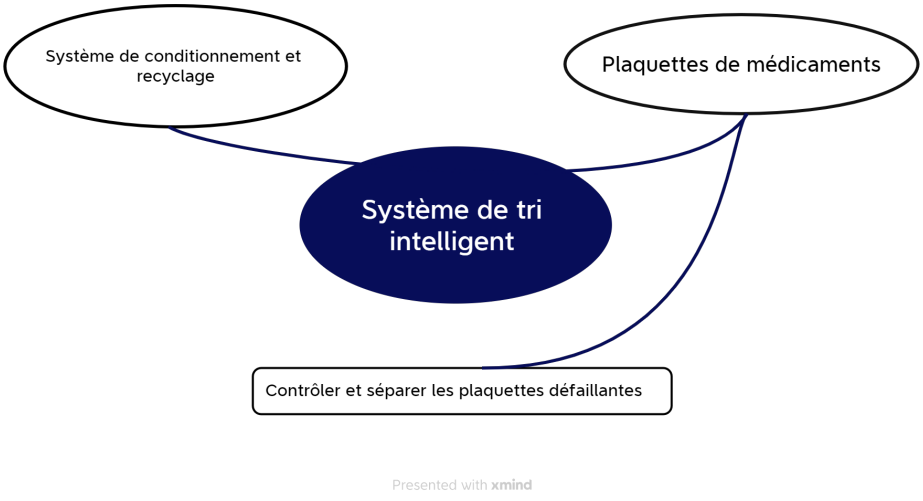


Fig. 3.3 : Diagramme bête à cornes

2-Diagramme de Pieuvre :

Le diagramme pieuvre est présenté dans la figure 3.4, accompagné d'un tableau explicatif des fonctions et désignations du système, est présenté ci-dessous, fournissant une représentation visuelle et détaillée des fonctions principales et secondaires de notre système.

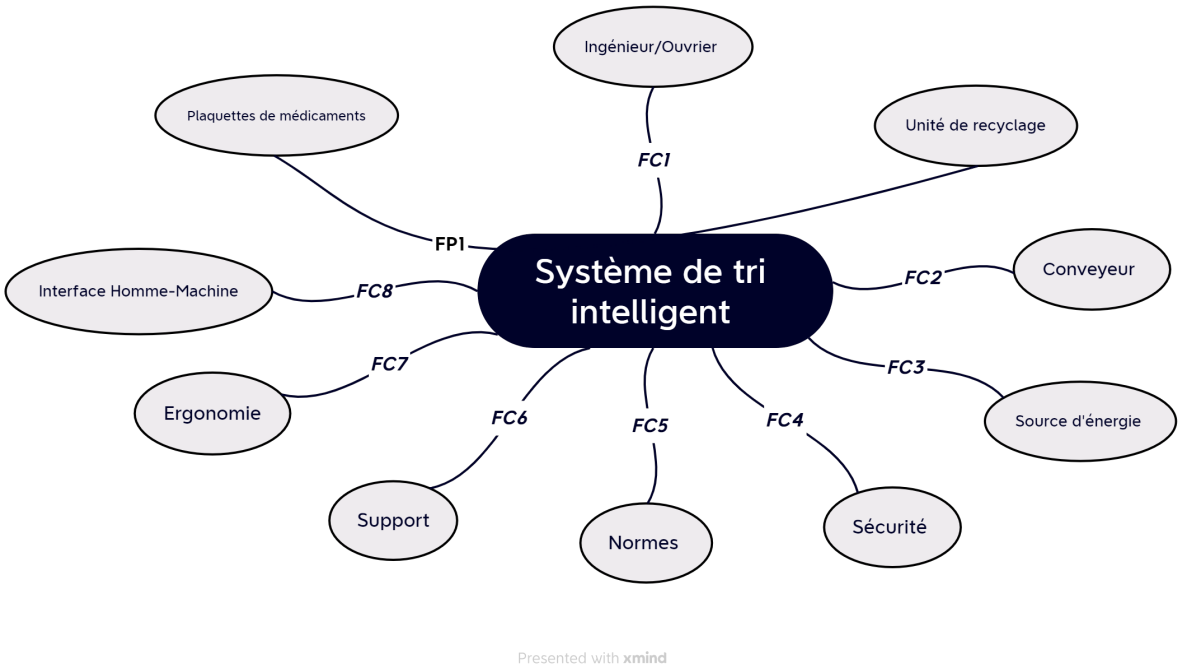


Fig. 3.4 : Diagramme de pieuvre

Fonctions	Désignation
FP	Contrôler et séparer les plaquettes défaillantes du convoyeur.
FC 1	Superviser et contrôler le système.
FC 2	Transporter les plaquettes de médicaments.
FC 3	Alimenter le système.
FC 4	Éliminer et minimiser les risques.
FC 5	Conformiser le système.
FC 6	Supporter le système.
FC 7	Adapter le système a l'environnement existant.
FC 8	Commander et donner l'état du système.

Tab. 3.1 : Fonctions et désignations du système

3-Cahier des charges Fonctionnel

Ce document de cahier des charges décrira en détail les besoins, les attentes et les spécifications de performance pour le système de détection et de tri des plaquettes de médicaments défectueuses.

Fonctions	Critères d'appréciations	Niveau d'exigences	Flexibilité
FP : Contrôler et Séparer les plaquettes défaillantes	Reconnaissance de l'objet.	Caméra RGB.	F0
	Critères et tolérances de contrôle.	Comprimés par plaquette (n=10).	F1
		Comprimés uniformes, non cassés.	F0
		Absence des grains, bavure et humidité	F0
	Tache de séparation.	Un bras manipulateur	F0
		Vérin double effet.	F1
	Coordonnées du box de tri.	(x, y,z)	F1
	Angle de rotation de moteur 1.	W= 30 degrés.	F1
	Angle de rotation de moteur 2.	W= 60 degrés.	F1
	Course du vérin.	≈ 2.5 cm	F1
	Préhension de la plaquette	Ventouse	F0
	Génération du vide.	Pompe à vide.	F0
FC1 : Superviser et contrôler le système	Interface utilisateur (HMI).	Bouton arrêt.	F0
		Etat du robot.	F1
		Interface de configuration.	F0
		Bouton marche.	F0
FC2 : Alimenter le système	Energie électrique	Alimentation DC 24V	F0
		Moteur électrique 1	F1
		Moteur électrique 2	F1
		API	F1
		NPU	F1
	Energie pneumatique.	Vérin avec ventouse.	F1
FC3 : Conformer le système	Matériau utilisé	Non corrosif	F0
	Exigences	Nettoyable, solide,	F0
FC4 : Eliminer et minimiser les risques	Sécurité électrique	Mise à la terre.	F0
		Composants électriques bien isolés.	F0
	Sécurité mécanique	Absence de mouvements mécaniques dangereux.	F0
		Présence d'un box de protection Arrêt d'urgence tout autour.	F0
		Présence d'un système de détection d'obstacles. .	F1
	Sécurité opérationnelle	Un système de QHSE solide	F0
FC5 : Supporter le système	Support solide	Matériaux non corrosif	F0
		Charge supporté : +60kg	F0
FC7 : Adapter le système a l'environnement existant	Type de robot	Scara 3DDL	F0
	Exigences	Base : ≈ 400×300 mm / ≈ 9kg	F1
		Lien 2 : ≈ 750 × 250 × 0.09 mm / ≈ 11 kg	F1
		Lien 3 : ≈ 750 × 250 × 0.05 mm / ≈ 13kg	F1
	Couleur :	Jaune	F1

TAB. 3.2 : Cahier des charges fonctionnel et technique

2.4 Analyse fonctionnelle interne

L'analyse fonctionnelle interne se concentre sur l'exploration détaillée des composantes du système, mettant en lumière leur interaction et leur contribution à la réalisation des fonctions globales du système.

1-Diagramme FAST :

Le diagramme FAST (Functional Analysis System Technique) ci-dessous illustre la séquence logique des fonctions dans le système étudié.

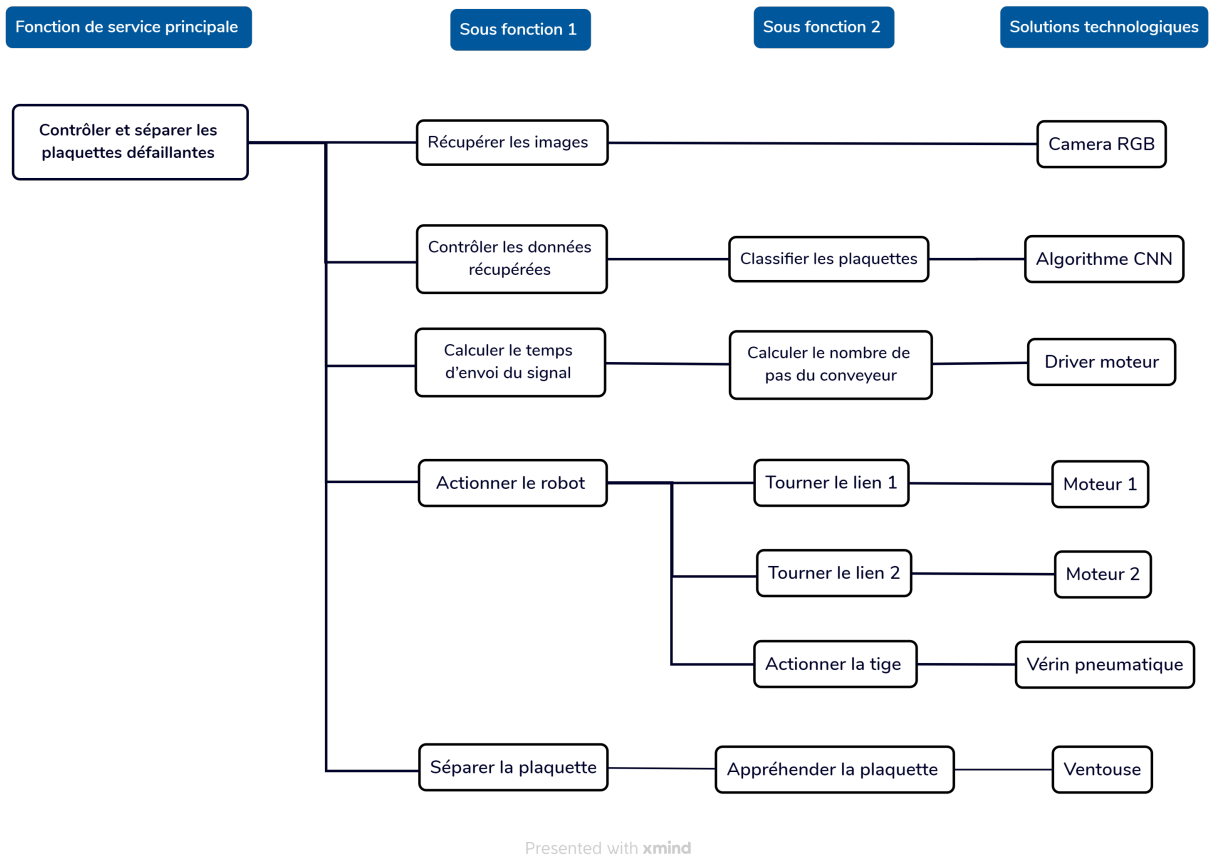


FIG. 3.5 : Diagramme FAST

2-Diagramme SADT :

Les diagrammes SADT (Structured Analysis and Design Technique) suivants, A-0 et A0, décomposent le système en sous-systèmes et décrivent leurs interactions.

Modélisation A-0 :

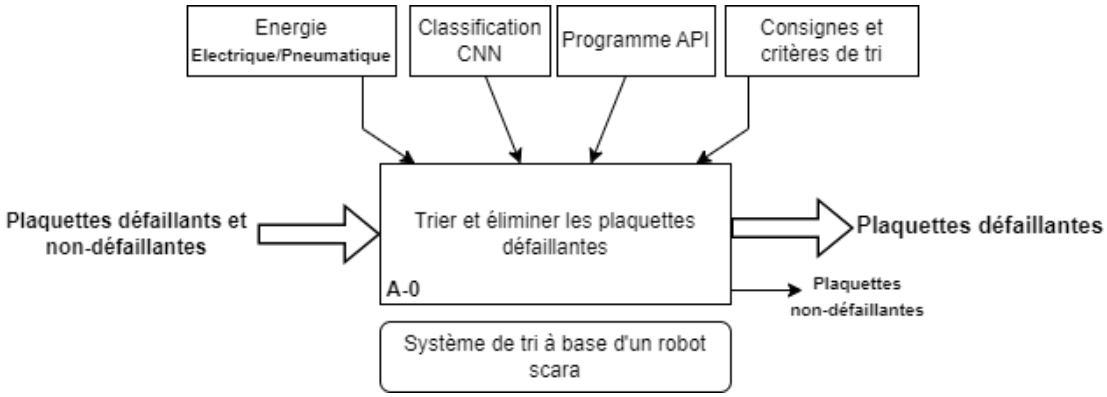


FIG. 3.6 : Diagramme SADT A-0

Modélisation A0 :

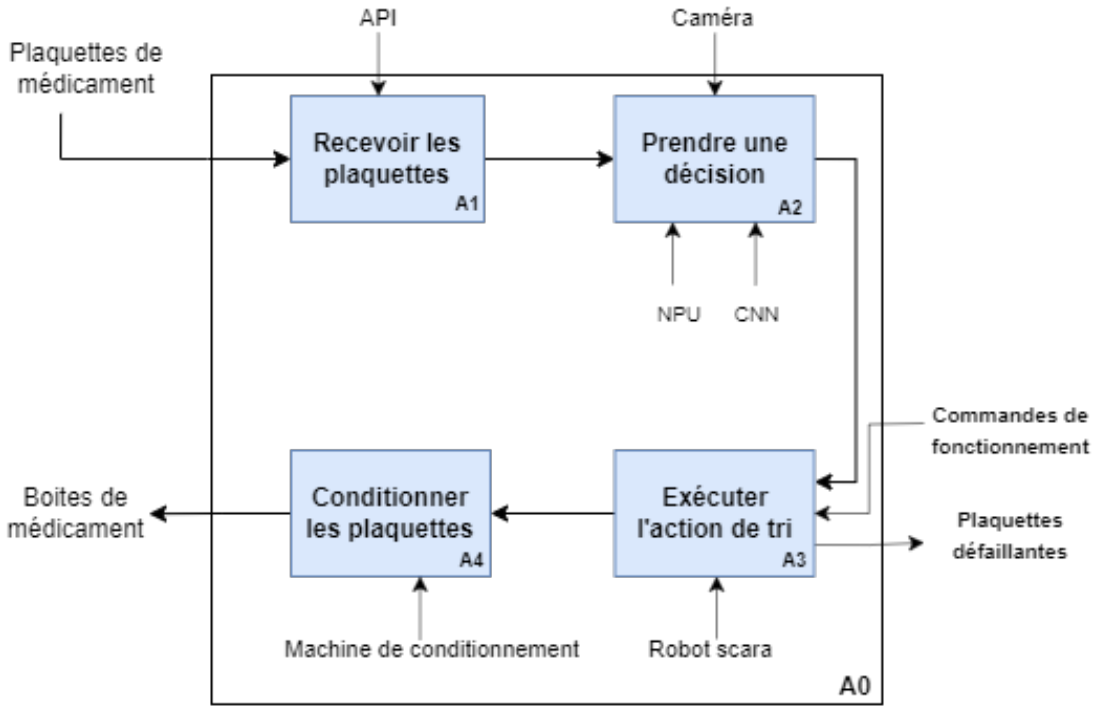


FIG. 3.7 : Diagramme SADT A0

2.5 Exposition de la solution

La système est principalement composé d'un bras manipulateur de type SCARA à trois degrés de liberté (3DDL). Doté d'un système de controle des plaquettes de médicament à base d'un algorithme des réseaux de neurones convolutifs CNN, et un système de commande à base d'un API S7-1500 avec un module de traitement neuronl NPU.

Éléments constitutifs du système

Notre système de tri est une partie de la chaine de production, et il se compose des éléments suivants :

1. Le Bras Manipulateur :

Un bras manipulateur SCARA AZ-2000 se compose des éléments suivants :

- **Moteur 1 :** Ce moteur est responsable de la rotation du premier segment (lien 2) du bras manipulateur.
- **Moteur 2 :** Ce deuxième moteur gère la rotation du second segment (lien 3) du bras manipulateur.
- **Vérin pneumatique :** Il sert à assurer un mouvement linéaire contrôlé et puissant, généralement utilisé pour la manœuvre de la ventouse.
- **Ventouse :** La ventouse joue un rôle crucial dans notre système, car elle est responsable de la préhension des plaquettes de médicaments identifiées comme défectueuses.

2. La Caméra :

Une caméra est utilisée pour capturer des images des plaquettes de médicaments sur le convoyeur. Ces images servent d'entrée pour l'algorithme d'intelligence artificielle qui classifie les plaquettes.

3. L'automate programmable

L'automate programmable est au cœur de la commande et de la coordination de l'ensemble du système. Il reçoit les informations de classification du module NPU et contrôle en conséquence les moteurs et le vérin pneumatique.

4. Module de Traitement Neuronal (NPU) :

Le NPU est un composant matériel spécifique conçu pour exécuter efficacement les opérations d'intelligence artificielle. Il est utilisé pour exécuter l'algorithme de CNN, qui analyse les images de plaquettes de médicaments capturées par la caméra et détermine si elles sont défectueuses ou non.

3 Mécanique

La conception mécanique d'un robot manipulateur est une opération complexe qui implique l'interaction harmonieuse de multiples disciplines, notamment la robotique, la mécanique et l'ingénierie de système. .

3.1 Choix du modèle du robot

Parmi les différents modèles et types de robots qui existe dans l'industrie, nous avons opté pour le bras manipulateur SCARA (Selective Compliance Assembly Robot Arm). Notre système doit faire le tri de médicaments d'une manière rapide, précise et dans un espace restreint, d'où les caractéristiques du robot SCARA :

- **Structure mécanique :**

Un robot SCARA a généralement deux bras articulés parallèles qui peuvent bouger dans un plan horizontal (x-y). Ils ont une rigidité sélective, ce qui signifie qu'ils sont rigides dans l'axe vertical (z), mais flexibles dans le plan horizontal, permettant une grande précision de mouvement dans ce plan.

- **Degrés de liberté :**

Les robots SCARA ont généralement 3 degrés de liberté : trois pour le mouvement dans le plan (deux rotations et une translation).

- **Précision :**

Ils sont connus pour leur haute précision et répétabilité, ce qui les rend idéaux pour des tâches de tri.

- **Vitesse :**

Les robots SCARA sont également rapides et efficaces, avec une haute cadence de cycle.

- **Charge utile :**

Ils sont généralement conçus pour manipuler de petites à moyennes charges utiles, typiquement de quelques kilogrammes.

- **Applications :**

Les robots SCARA sont couramment utilisés dans des applications telles que le montage de pièces, le pick-and-place, l'emballage et le vissage.

3.2 Description, conception et présentation du modèle SCARA AZ-2000

Notre système mécanique est un robot SCARA AZ-2000 à 3 degrés de liberté, avec 2 rotations et une translation. Il se compose de 3 liens et 3 articulations avec une préhension par aspiration à travers une ventouse.

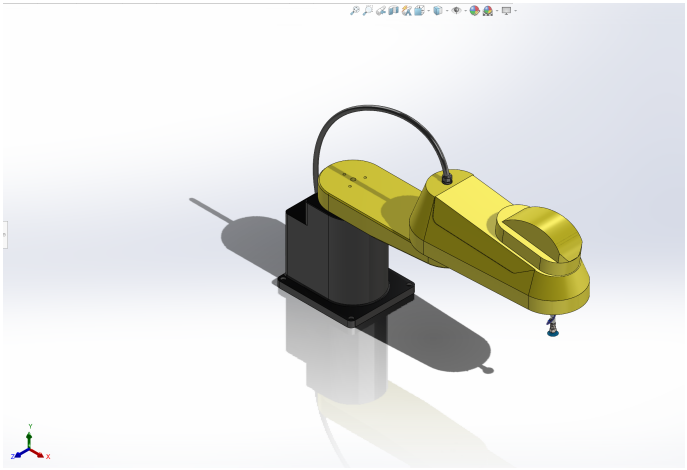
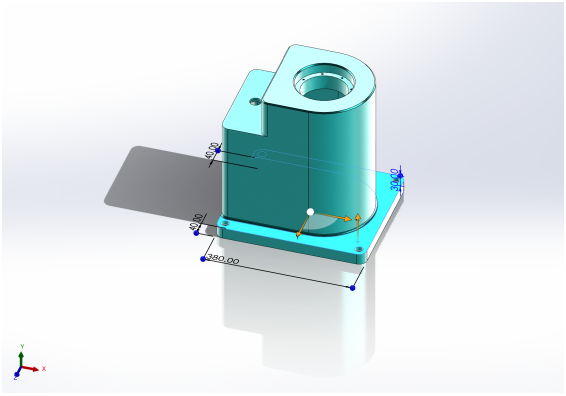


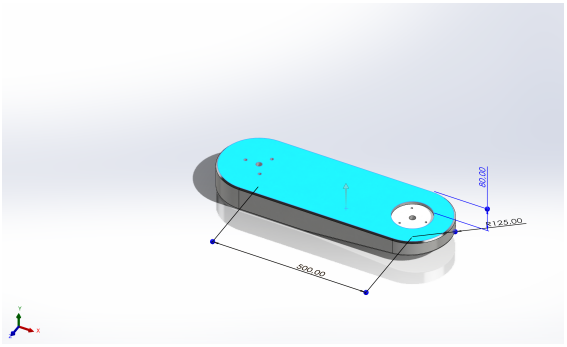
FIG. 3.8 : Modèle 3D du robot SCARA AZ-2000

3.2.1 Composants mécaniques

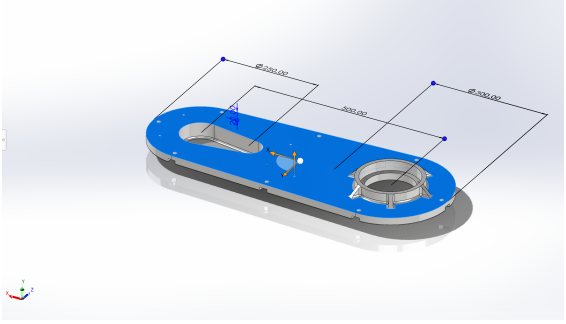
Présentés ci-aprés se trouvent les composants mécaniques clés du robot SCARA AZ-2000, aevec leurs caractéristiques détaillées dans le tableau 3.3



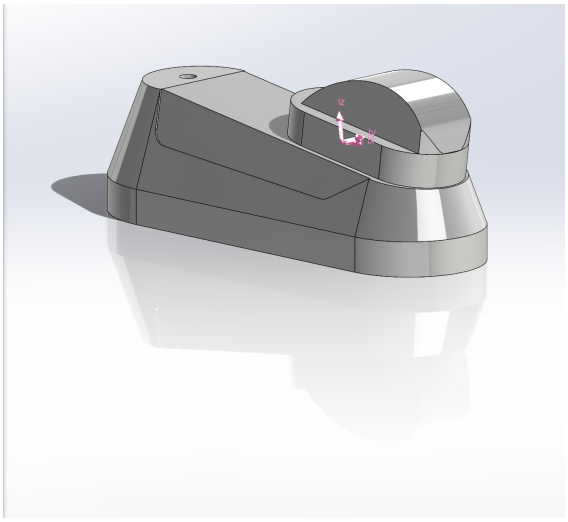
(a) Lien de base



(b) Lien 1



(c) Lien 2



(d) Cache moteur

FIG. 3.9 : composants mécaniques clés du SCARA AZ-2000

Composant	Masse (Kg)	Longuer (m)	Largeur (m)	Hauteur (m)
La base	9.348	0.38	0.35	0.4
Lien 2	11.84593	0.75	0.25	0.08
Lien 3	13.26796	0.75	0.25	0.04

TAB. 3.3 : Caractéristiques des composants du SCARA AZ-2000

3.2.2 Assemblage

C'est l'assemblage des composants illustrés dans la figure 3.8 qui donne vie au robot SCARA AZ-2000.

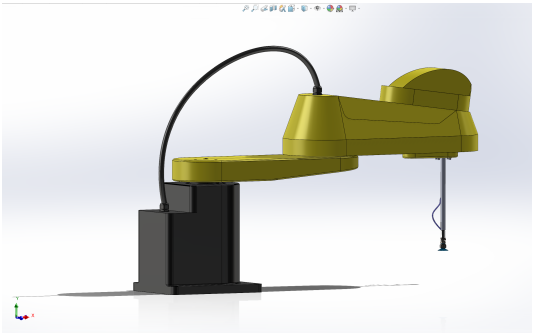


FIG. 3.10 : Vue 3D avec course du vérin

3.3 Choix des matériaux

Dans la conception d'un robot manipulateur, le choix des matériaux est un aspect essentiel qui influence de manière significative sa performance, sa durabilité et sa fonctionnalité. Et celà, est particulièrement important pour un robot destiné au tri de médicaments dans une usine pharmaceutique, où des critères tels que la résistance à la corrosion, la légèreté, la robustesse et la facilité de nettoyage sont importants.

3.3.1 Matériau choisi

- **Acier d'aluminium 6061**

L'aluminium 6061 a été choisi pour sa combinaison unique de propriétés qui le rendent idéal pour notre application spécifique.

En général, les alliages d'aluminium de la série 6000 sont alliés au magnésium et au silicium. L'alliage 6061 est l'un des alliages les plus utilisés de la série 6000 à cause des ses bonnes propriétés mécaniques (https ://material-properties.org/fr/alliage-6061-densite-resistance-durete-point-de-fusion/)

3.3.2 Propriétés

Les Propriétés suivantes ont été recueillies à partir de la bibliothèque de matériaux de **SolidWorks** :

Propriétés	Valeur	Unité
Module d'élasticité	96000	N/mm^2
Coefficient de poisson	0.33	S.O.
Module de césaillement	2600	N/mm^2
Masse volumique	2700	Kg/m^3
Limite d'élasticité	55.1485	N/mm^2
Limite de traction	124.084	N/mm^2

Tab. 3.4 : Propriétés de l'Aluminium 6061

3.3.3 Caractéristiques :

L'aluminium 6016 a été préféré pour sa gamme de caractéristiques :
• Une excellente résistance à la corrosion, ce qui est essentiel dans un environnement pharmaceutique.
• L'aluminium 6061 possède une excellente résistance à la traction, ce qui signifie qu'il peut résister à des forces importantes sans se rompre.
• Il offre une bonne résistance mécanique, une légèreté qui contribue à l'efficacité énergétique du robot.
• La facilité d'usinage qui simplifie le processus de fabrication.
• L'aluminium 6061 a une conductivité thermique élevée, ce qui aide à dissiper la chaleur générée par le robot.

3.4 Etude RDM

L'étude de la RdM peut nous aider à comprendre comment le robot réagira à diverses charges et contraintes pendant son fonctionnement. Cela peut à son tour valider la conception du robot et le choix de matériau, et nous aidant à dimensionner correctement les éléments structuraux, et à anticiper les points de défaillance potentiels.

Pour ce faire, nous allons utiliser **SolidWorks** comme outil pour analyser notre bras manipulateur.
Dans cette étape nous avons appliquer une force de 1N sur le bras manipulateur.

3.4.1 Analyse des résultats

a. Résultats : Les résultats des contraintes appliquées sur le robot, ainsi que les effets de ces forces sur la déformation et le déplacement de la structure, sont présentés ci-après.

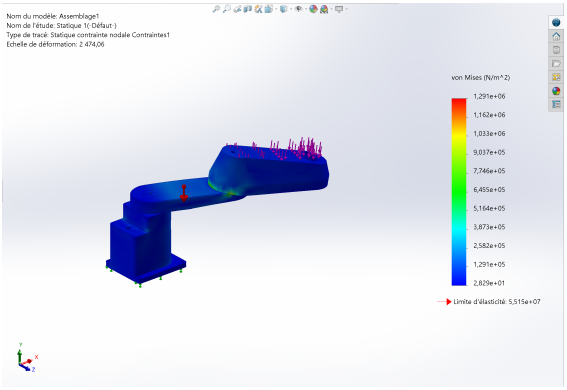


FIG. 3.11 : Résultat des contraintes appliquées sur le robot

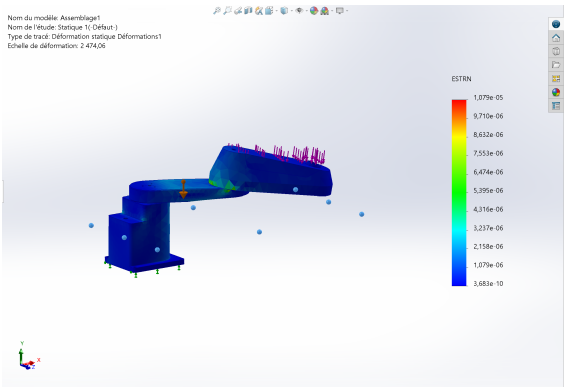


FIG. 3.12 : Résultat de deformation sous l'effet les forces appliquées

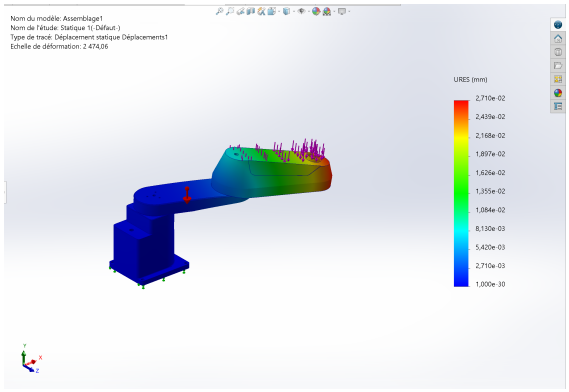


FIG. 3.13 : Résultat du déplacement de la structure sous la contrainte appliquée

b. Analyse :

1.Analyse des contraintes :

Dans l'étude RDM, nous avons examiné les contraintes subies par le robot manipulateur. En tenant compte des forces appliquées sur notre robot, nous avons identifié deux contraintes majeures :

- **Contrainte de gravité :**

Cette contrainte est due au poids propre du robot. Comme le robot est soumis à la gravité terrestre, cette contrainte est une constante et doit être prise en compte lors de la conception de la structure et des composants mécaniques du robot.

- **Contrainte de charge :**

Cette contrainte est liée à la charge que le robot doit porter. Dans notre cas, le robot est conçu pour manipuler une charge de 30.2 grammes.

2. Résistance du modèle :

En analysant les contraintes, nous avons confirmé que le modèle résiste bien à ces efforts. Ce résultat est illustré dans le diagramme de déplacement dans la figure 3.15, qui a montré que la valeur maximale de déplacement était de 2.710^{-2} mm . Cette valeur est bien à l'intérieur de notre plage de tolérance, indiquant que notre modèle mécanique et notre choix de matériaux

sont adéquats pour les contraintes prévues.

3. Analyse de déformation

Dans le diagramme de déformation illustré dans la figure 3.13, nous avons utilisé le critère de **Von Mises** pour évaluer la déformation élastique de robot sous une contrainte complexe tridimensionnelle. Selon ce critère, la contrainte équivalente était de $1.310^6\,\mathrm{N/m^2}$.

Cette valeur est acceptable pour l'application et confirme que le modèle est capable de supporter les contraintes sans subir de déformation inacceptable.

3.4.2 Discussion

L'analyse RDM que nous avons réalisée valide donc notre modèle de conception du robot et justifie notre choix de matériaux. Les résultats indiquent que le robot est capable de fonctionner efficacement tout en supportant les contraintes de gravité et de charge sans subir de déplacement ou de déformation significatifs.

3.5 Modélisation Géometrique

Le modèle de robot SCARA AZ-2000 est représenté par le schéma cinématique Ci-aprés :

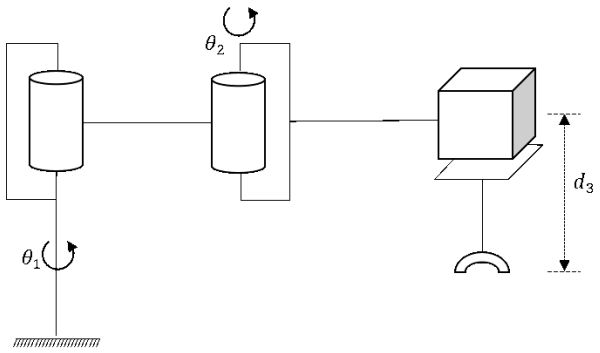


FIG. 3.14 : Schéma cinématique de robot SCARA AZ-2000

3.5.1 Modélisation Géométrique Directe (MGD)

On appelle Modèle Géométrique Direct (MGD) d'un robot, l'application f de R_n *dans* R_6.

$$X = f(q) = \begin{bmatrix} \text{Position de } l'\text{effectueur dans } R_0 \\ \text{Orientation de } R_n \text{ par rapport à } R_0 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \\ \alpha \\ \beta \\ \gamma \end{bmatrix}_{/R_0} \tag{3.1}$$

Le MGD permet donc de connaitre les mouvements de l'effecteur dans l'espace de la tâche, en fonction des mouvements des articulations du robot. Il permet notamment de déterminer l'espace de travail du robot (domaine atteignable), qui est évidemment un critère majeur du choix d'un robot.

3.5.2 Modélisation Géométrique Inverse (MGI)

On appelle modèle géométrique inverse (MGI), l'application f^{-1} de $\mathbb{R}^6$ dans $\mathbb{R}^n$ exprimant q en fonction de : $q = f^{-1}(x)$.

Le MGI est destiné à réaliser un transformateur de coordonnées, à partir des coordonnées opérationnelles souhaitées vers les consignes articulaires correspondantes. Le MGI est donc la forme explicite qui donne toutes les solutions possibles pour q, correspondantes à un x fixé.

Convention de Denavit-Hartenberg D-H :

Pour définir et résoudre les problèmes liés aux géométries directes et inverses, il est important de passer d'abord par la convention Denavit-Hartenberg (DH).

La règle DH est basée sur une définition de chaque articulation du robot et établit quatre paramètres importants, deux longueurs et deux angles, qui décrivent la relation spatiale entre les articulations adjacentes. Grâce à cette approche, un modèle précis du robot peut être développé, qui est la base pour dériver la cinématique et les équations dynamiques du robot.

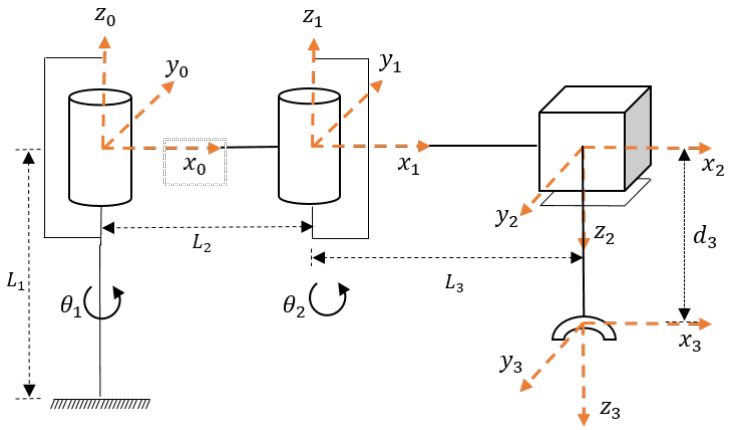


FIG. 3.15 : Schema des parametres DH robot SCARA

Articulations	θ_i	d_i	a_i	α_i	q
j_1^0	q_1^*	L_1	L_2	0	θ_1
j_2^1	q_2^*	0	L_3	π	θ_2
j_3^2	0	q_3^*	0	0	d_3

TAB. 3.5 : Tableau des paramètres D-H

Avec :

- θ_i : angle entre les axes X_{i-1} et X_i autour de l'axe Z_{i-1}
- d_i : coordonnée de O_i' le long de Z_{i-1}
- a_i : distance entre O_i et O_i'
- α_i : angle entre les axes Z_{i-1} et Z_i autour de l'axe X_i

Matrice de transformation homogène de D-H :

$$A_i^{i-1} = \begin{bmatrix} C\theta_i & -S\theta_i C\alpha_i & S\theta_i S\alpha_i & a_i C\theta_i \\ S\theta_i & C\theta_i C\alpha_i & -C\theta_i S\alpha_i & a_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} {}^0M_n & \overrightarrow{O_0O_{n/R_0}} \\ 0_{1x3} & 1 \end{bmatrix} \tag{3.2}$$

Avec :

- $\overrightarrow{O_0O_{n/R_0}}$: Vecteur qui exprime les coordonnées du point P de l'organe terminal dans le repère de base R_0
- ${}^0M_n$: Matrice de passage ${}^0M_p$ de R_0 à R_p qui exprime l'orientation de l'organe effecteur par rapport au référentiel de la base.

Joint 1 : Liaison rotoïde 1 :

$$A_1^0 = \begin{bmatrix} C\theta_1 & -S\theta_1 C\alpha_1 & 0 & 0 \\ S\theta_1 & C\theta_1 & 0 & L_2 S\theta_1 \\ 0 & 0 & 1 & L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Joint 2 : Liaison rotoïde 2 :

$$A_2^1 = \begin{bmatrix} C\theta_2 & S\theta_2 & 0 & L_3 C\theta_2 \\ S\theta_2 & -C\theta_2 & 0 & L_3 S\theta_2 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Joint 3 : Liaison prismatique :

$$A_3^2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_3^0 = A_1^0 A_2^1 A_3^2 = \begin{bmatrix} C_1C_2 - S_1S_2 & C_1S_2 + S_1C_2 & 0 & L_2C_1 + L_3(C_1C_2 - S_1S_2) \\ C_1S_2 + S_1C_2 & -C_1C_2 + S_1S_2 & 0 & L_2S_1 + L_3(C_1S_2 + S_1C_2) \\ 0 & 0 & -1 & -d_3 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

- **La position de l'organe effecteur peut être donnée suivant les coordonnées :**

$$\begin{aligned} P_x &= L_2C_1 + L_3(C_1C_2 - S_1S_2) = L_2C_1 + L_3C_{1+2} \\ P_y &= L_2S_1 + L_3(C_1S_2 + S_1C_2) = L_2S_1 + L_3S_{1+2} \\ P_z &= -d_3 \end{aligned}$$

- **L'orientation de l'organe effecteur :**

$${}^0M_3 = \begin{bmatrix} C_1C_2 - S_1S_2 & C_1S_2 + S_1C_2 & 0 \\ C_1S_2 + S_1C_2 & -C_1C_2 + S_1S_2 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

$$^0M_3 = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = \begin{bmatrix} C_\alpha C_\gamma - S_\alpha C_\beta S_\gamma & -C_\alpha S_\gamma - S_\alpha C_\beta C_\gamma & S_\alpha S_\beta \\ S_\alpha C_\gamma + C_\alpha C_\beta S_\gamma & -S_\alpha S_\gamma + C_\gamma C_\beta C_\gamma & -C_\alpha S_\beta \\ S_\beta S_\gamma & S_\beta C_\gamma & C_\beta \end{bmatrix}$$

À l'aide des angles d'Euler, on détermine les valeurs des trois angles α, β, γ de la matrice de rotation relative $^0M_3$ et à partir des éléments r_{ij}.

Par correspondance on a :

$r_{33} = C_\beta = -1 \rightarrow \beta = \pi$ Alors : $\alpha - \gamma = atan2(r_{21}, r_{11})$

Dans ce cas, α et γ sont indéterminés, et l'approche d'Euler montre une singularité de représentation.

Pour résoudre la singularité d'Euler, on procède le calcul en utilisant une représentation différente : par exemple, des angles de Cardan pour représenter l'orientation de l'effecteur. Cette représentation peut éviter les ambiguïtés et permettent de représenter l'orientation de manière plus robuste.

Nous pouvons fixer arbitrairement l'un des angles indéterminés et calculer l'autre en conséquence. En utilisant cette méthode on aura :

Calcul de β :

Comme nous l'avons déjà discuté, $r_{33} = C_\beta = -1 \rightarrow \beta = \pi$

Calcul de α :

Si nous fixons α à 0 (ce qui est arbitraire), alors $\alpha = 0$.

Calcul de γ :

Avec $\alpha = 0$ et $\beta = \pi$, nous pouvons utiliser la nouvelle matrice de rotation :

$$^0M_3 = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = \begin{bmatrix} C_\gamma & -C_\beta C_\gamma & 0 \\ C_\beta S_\gamma & C_\gamma & 0 \\ S_\beta S_\gamma & S_\beta C_\gamma & C_\beta \end{bmatrix}$$

Maintenant, γ peut être déterminé en utilisant $r_{11} = C_\gamma$ et $r_{21} = C_\beta S_\gamma$ avec $C_\beta = 1$

Comme nous avons fixé α à 0 et β à π, les équations se simplifient et on aura : $\gamma = arctan\left(\frac{r_{21}}{r_{11}}\right)$

Maintenant on peut exprimer le MGD sous la forme :

$$X = f(q) = \begin{bmatrix} x \\ y \\ z \\ \alpha \\ \beta \\ \gamma \end{bmatrix}_{R_0} = \begin{bmatrix} L_2 C_1 + L_3 C_{1+2} \\ L_2 S_1 + L_3 S_{1+2} \\ -d_3 \\ 0 \\ \pi \\ arctan\left(\frac{r_{21}}{r_{11}}\right) \end{bmatrix}$$

Pour la géométrie inverse on aura :

$$\theta_2 = atan2\left(\sqrt{1-\left(\frac{P_x^2+P_y^2-L_2^2-L_3^2}{2L_2L_3}\right)^2}\right), \left(\frac{P_x^2+P_y^2-L_2^2-L_3^2}{2L_2L_3}\right)$$
$$\theta_1 = atan2(P_y, P_x) - atan2(L_3\sin(\theta_2), L_2 + L_3\cos(\theta_2))$$
$$d_3 = -P_z$$

Avec :
$C_1 = \cos(\theta_1)$; $C_2 = \cos(\theta_2)$; $S_1 = \sin(\theta_1)$; $S_2 = \sin(\theta_2)$; $C_{1+2} = \cos(\theta_1 + \theta_1)$
$S_{1+2} = \sin(\theta_1 + \theta_1)$.

3.6 Modélisation Cinématique

La modélisation cinématique [38] s'intéresse aux relations liant les vitesses opérationnelles (c'est à dire la vitesse instantanée du point P de l'effecteur et la vitesse de rotation du repère R_p lié à l'organe terminal) en fonction des vitesses articulaires $\dot{q}$.

- **Modèle cinématique direct :**

On appelle modèle cinématique direct (MCD) d'un robot, l'application donnant les vitesses opérationnelles $\dot{X}$ en fonction des positions q et des vitesses $\dot{q}$:

$$\dot{X} = g(q, \dot{q}) \tag{3.3}$$

Au contraire du modèle géométrique, qui correspond à une relation non linéaire entre X et q, le modèle cinématique correspond à une relation linéaire entre $\dot{x}$ et $\dot{q}$. Le modèle cinématique peut s'écrire sous la forme :

$$\dot{X} = J(q)\dot{q} \tag{3.4}$$

La matrice $J(q)$ est nommée matrice jacobéenne ou simplement jacobéenne du robot. Cette matrice est de dimension (dn), où $d = 6$ au maximum et n le nombre de variable articulaire.

Par convention, les matrices jacobéennes présentées seront toujours exprimées dans le repère R_0.

En utilisant la méthode pour calculer la jacobéenne :

$$J = \begin{bmatrix} J_v \\ J_w \end{bmatrix} \tag{3.5}$$

Avec :

J_v : Jacobéen linéaire
J_w : Jacobéen rotationnel

$$J_v = \begin{cases} z_{i-1}(O_n - O_{i-1}) & \text{révolutif} \\ z_{i-1} & \text{prismatique} \end{cases}$$

$$J_w = \begin{cases} z_{i-1} & \text{révolutif} \\ 0 & \text{prismatique} \end{cases}$$

- **Calcul de la jacobéenne :**

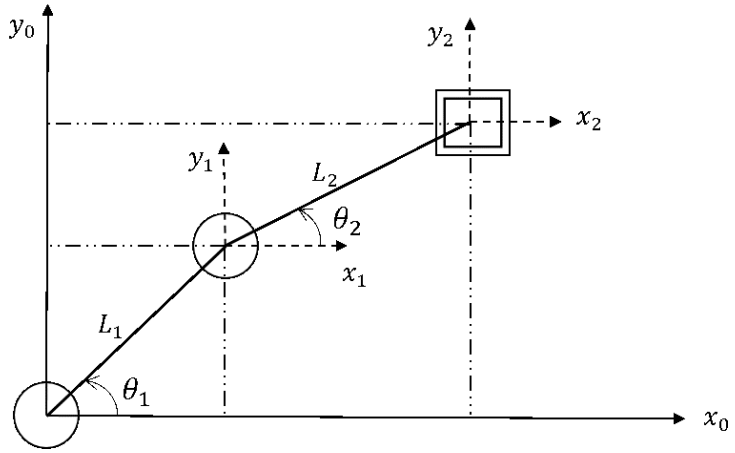


FIG. 3.16 : Model dessus sur le plan x,y

Il faut d'abord localiser tous les origines des joints O_n^0 d'aprés la figure 3.16 :

$$O_0^0 = \begin{bmatrix} 0 \\ 0 \\ L_1 \end{bmatrix}$$

$$O_1^0 = \begin{bmatrix} L_1\cos(\theta_1) \\ L_1\sin(\theta_1) \\ L_1 \end{bmatrix}$$

$$O_2^0 = \begin{bmatrix} L_2\cos(\theta_1) + L_3\cos(\theta_{1+2}) \\ L_2\sin(\theta_1) + L_3\sin(\theta_{1+2}) \\ L_1 \end{bmatrix}$$

Ecrire les vecteurs des axes Z_i sur chaque joint :

$$\begin{bmatrix} Z_0^0 \\ Z_1^0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$Z_2^0 = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix}$$

Avec une application mathématique :

$$J = \begin{bmatrix} J_v \\ J_w \end{bmatrix} = \begin{bmatrix} z_0(O_3 - O_0) & z_1(O_3 - O_1) & z_2(O_3 - O_2) \\ z_0 & z_1 & 0 \end{bmatrix} \tag{3.6}$$

$$J = \begin{bmatrix} -L_2\sin(\theta_1) - L_3\sin(\theta_{1+2}) & -L_3\sin(\theta_{1+2}) & 0 \\ L_2\cos(\theta_1) + L_3\cos(\theta_{1+2}) & L_3\cos(\theta_{1+2}) & 0 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix} \tag{3.7}$$

Alors :

Le modèle cinématique directe du robot SCARA est :

$$\dot{X} = J(q)\dot{q} \rightarrow \dot{X} = \left(\begin{bmatrix} -L_2\sin(\theta_1) - L_3\sin(\theta_{1+2}) & -L_3\sin(\theta_{1+2}) & 0 \\ L_2\cos(\theta_1) + L_3\cos(\theta_{1+2}) & L_3\cos(\theta_{1+2}) & 0 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix} \right) \dot{q} \tag{3.8}$$

Avec :

$$X = \begin{bmatrix} x \\ y \\ z \\ \alpha \\ \beta \\ \gamma \end{bmatrix}$$

$$q = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix}$$

3.7 Modélisation Dynamique

Déterminons maintenant le modèle dynamique qui est la relation entre les couples (et/ou forces) généré par les actionneurs et les positions, vitesses et accélérations articulaires[39]. Il est représenté par la relation :

$$\tau = f(q, \dot{q}, \ddot{q}, f_e) \tag{3.9}$$

Avec :

τ : Vecteur des couples ou forces des actionneurs, selon que l'articulation est rotoide ou prismatique.

q : Vecteur des positions articulaires.

$\dot{q}$: Vecteurs des vitesses articulaires.

$\ddot{q}$: Vecteurs des accélérations articulaires.

f_e : Vecteur représentant les couples ou forces des frottements.

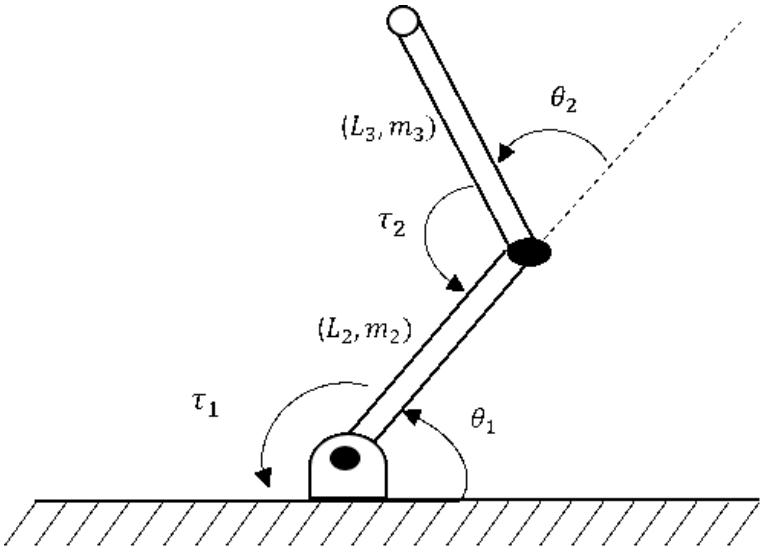


FIG. 3.17 : Schéma dynamique vue de dessus

Dans ce cas, l'organe effecteur est considéré négligeable pour plusieurs raisons :

- Masse faible par rapport au reste du robot : alors son impact sur la dynamique du système peut être considéré comme négligeable.
- Simplification du modèle : L'inclusion de l'effecteur final dans le modèle dynamique peut compliquer considérablement les équations du mouvement. En négligeant l'effecteur, on peut obtenir un modèle plus simple qui est plus facile à manipuler et à résoudre.
- Mouvement principalement dans une seule direction : Dans le cas de notre robot SCARA RRP, l'effecteur final se déplace principalement dans une seule direction (le long de l'axe prismatique). Cela signifie que l'effet de la gravité et des autres forces externes sur l'effecteur peut être assez simple et peut être facilement pris en compte sans devoir modéliser explicitement la dynamique de l'effecteur.

Procédure de calcul par la méthode de Lagrange :

Il existe plusieurs approches pour modéliser la dynamique des robots, chacune ayant ses propres avantages et défis. Parmi les méthodes les plus couramment utilisées, on trouve la méthode de Newton-Euler (NE) et la méthode de Lagrange. Dans le cadre de notre travail, nous avons choisi d'utiliser la méthode de Lagrange pour la modélisation dynamique de notre robot. Cela est dû à sa simplicité relative et à sa capacité à gérer efficacement les systèmes à

plusieurs degrés de liberté. Les équations de Lagrange opèrent à partir de l'énergie cinétique et l'énergie potentielle d'un système, le lagrangien s'écrit sous la forme :

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) - \frac{dL}{dq} = \tau \tag{3.10}$$

Avec :

$L = K - P$
K : Energie Cinétique du système.
P : Energie potentielle du système.

L'énergie cinétique du système est une fonction quadratique des vitesses articulaires :

$$K(\dot{q}, q) = \frac{1}{2}\dot{q}^T M(q)\dot{q} \tag{3.11}$$

Avec $M(q)$: la matrice d'inertie de dimensions (nxn) symétrique définie positive.

L'énergie potentielle du système est sous la forme :

$$P(q) = \sum_{i=1}^{n} P_i(q) \tag{3.12}$$

Avec :

$$P_i(q) = m_i g z_{G_i}(R_0) \tag{3.13}$$

Ceci est l'énergie potentielle du segment i avec :

- m_i : La masse du segment i
- g : La constante universelle de gravitation.
- G_i : Le centre d'inertie du segment i.

Donc :

$$L(q, \dot{q}) = \frac{1}{2}\dot{q}^T M(q)\dot{q} - P(q) \tag{3.14}$$

Le modèle obtenu peut être mis sous la forme matricielle suivante :

$$\tau = M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) \tag{3.15}$$

Avec :

- τ : vecteur des couples ou forces appliquées aux articulations.

- $M(q)$: matrice d'énergie cinétique du robot.
- $C(q, \dot{q})$: vecteur des forces de Coriolis et centrifuges.
- $G(q)$: vecteur des forces de gravité.

Par application, nous obtenons :

- Vecteur des variables articulaires : $q = (q_1, q_2)^T$

- Vecteur des couples généralisés : $\tau = (\tau_1, \tau_2)^T$

- Matrice d'inertie :

$$M(q) = \begin{pmatrix} m_3 L_3 + 2m_3 L_3 L_2 \cos\theta_2 + (m_3 + m_2)L_2^2 & m_3 L_3^2 + m_3 L_3 L_2 \cos\theta_2 \\ m_3 L_3^2 + m_3 L_3 L_2 \cos\theta_2 & m_3 L_3^2 \end{pmatrix} \tag{3.16}$$

- Matrice de Coriolis et centrifuges :

$$C(q, \dot{q}) = \begin{pmatrix} -2m_2 L_3 L_2 \sin 2\theta\dot{\theta}_2 & -m_2 L_3 L_2 \sin 2\theta\dot{\theta}_2 \\ m_2 L_3 L_2 \sin\theta\dot{\theta}_1 & 0 \end{pmatrix} \tag{3.17}$$

Vecteurs des forces de gravité :

$$G(q) = \begin{pmatrix} m_2 L_2 g \cos(1+2) + (m_2 + m_3)L_2 g \cos\theta_1 \\ m_3 L_3 g \cos(1+2) \end{pmatrix} \tag{3.18}$$

En conclusion :

Le modèle dynamique de notre SCARA est exprimé par :

$$\begin{pmatrix} \tau_1 \\ \tau_2 \end{pmatrix} = \begin{pmatrix} m_3 L_3 + 2m_3 L_3 L_2 \cos\theta_2 + (m_3 + m_2)L_2^2 & m_3 L_3^2 + m_3 L_3 L_2 \cos\theta_2 \\ m_3 L_3^2 + m_3 L_3 L_2 \cos\theta_2 & m_3 L_3^2 \end{pmatrix} \begin{pmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{pmatrix}$$

$$+ \begin{pmatrix} -2m_2 L_3 L_2 \sin 2\theta\dot{\theta}_2 & -m_2 L_3 L_2 \sin 2\theta\dot{\theta}_2 \\ m_2 L_3 L_2 \sin\theta\dot{\theta}_1 & 0 \end{pmatrix} \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \end{pmatrix} +$$

$$\begin{pmatrix} m_2 L_2 g \cos(1+2) + (m_2 + m_3)L_2 g \cos\theta_1 \\ m_3 L_3 g \cos(1+2) \end{pmatrix} \begin{pmatrix} q_1 \\ q_2 \end{pmatrix} \tag{3.19}$$

3.8 Simulation sur MATLAB

Après avoir calculé les modèles de géométrie directe et inverse, ainsi que la cinématique du robot SCARA AZ-2000, nous avons jugé essentiel de procéder à une simulation sur MATLAB grâce à ses bibliothèques spécialisées, telles que la Robotics System Toolbox et Simscape pour but de vérifier les résultats obtenus par nos calculs et d'examiner comment le robot se comporte dans divers scénarios. [40]

Les dimensions du robot pour la simulation Matlab sont les suivantes :

- Base : Un cylindre avec rayon $r = 0.05$ et hauteur $h = 0.1$
- Link 1 : Un solide rectangulaire de dimensions $x = 0.2$, $y = 0.02$ et $z = 0.02$
- Link 2 : Un solide rectangulaire de dimensions $x = 0.15$, $y = 0.02$ et $z = 0.02$
- Organe effecteur : Un solide rectangulaire de dimensions $x = 0.02$, $y = 0.02$ et $z = 0.03$

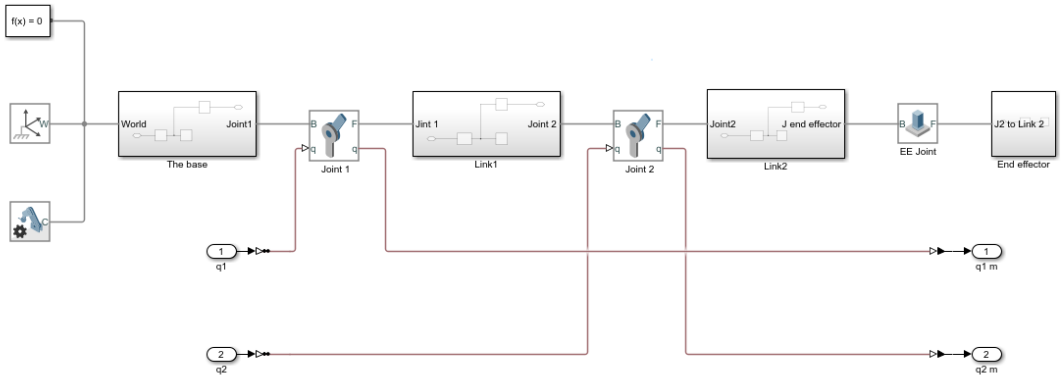


FIG. 3.18 : Modélisation de SCARA AZ-2000 avec Simulink

3.8.1 Simulation de la Cinématique Directe :

Grâce à la cinématique directe, les coordonnées (x,y,z) de l'organe effecteur ont été calculés à partir des angle θ_1 et θ_2. La précision de ces calculs, obtenue grâce à MATLAB, a permis d'obtenir des résultats fidèles à la réalité.

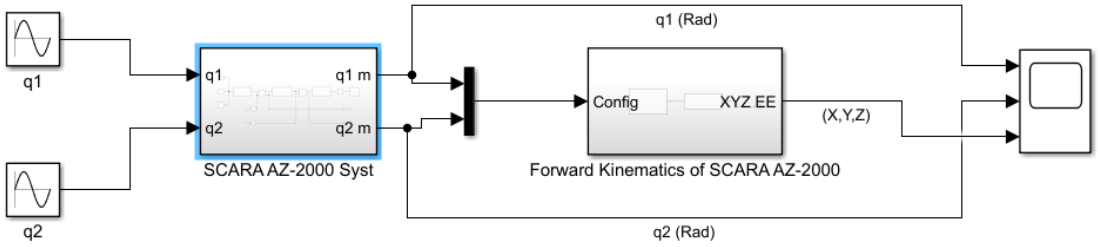


FIG. 3.19 : Schéma d'implémentation de la cinématique directe

Les scénarios et leurs résultats sont les suivants :

- **Scénario #1 :** Pour $\theta_1 = 0°$, $\theta_2 = 0$, et $Z = 0.03$, nous avons $P_x = 0.35$, $P_y = 0$, et $P_z = 0.02$.

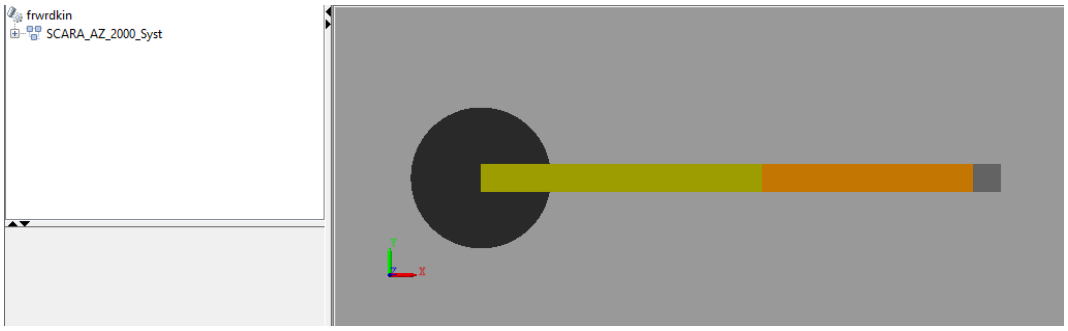


FIG. 3.20 : SCARA AZ-2000 sur la 1ère position

- **Scénario #2 :** Pour $\theta_1 = 0°$, $\theta_2 = 60°$, et $Z = 0.03$, nous avons $P_x = 0.275$, $P_y = 0.129$, et $P_z = 0.02$.

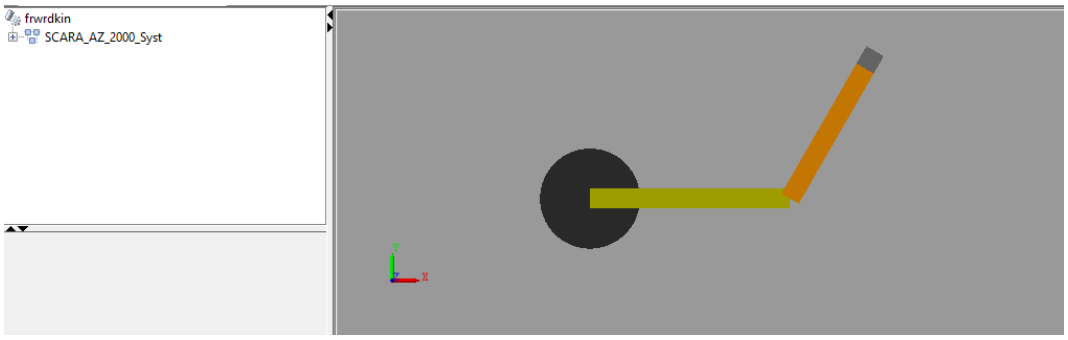


FIG. 3.21 : SCARA AZ-2000 sur la 2ème position

- **Scénario #3 :** Pour $\theta_1 = 30°$, $\theta_2 = 60$, et $Z = 0.03$, nous avons $P_x = 0.173$, $P_y = 0.25$, et $P_z = 0.02$.

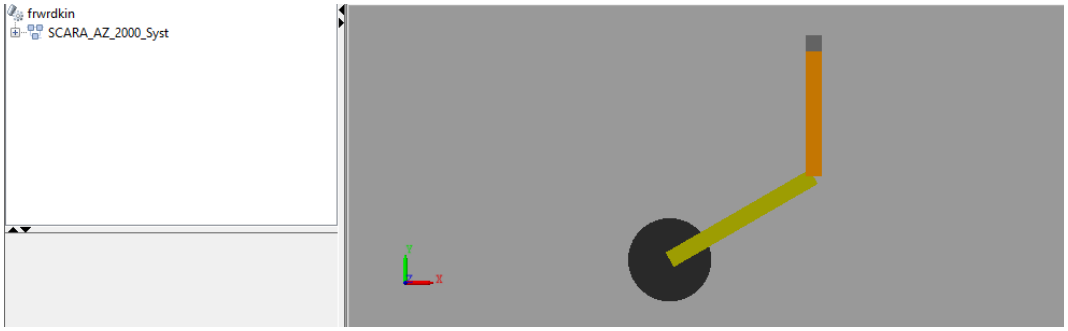


FIG. 3.22 : SCARA AZ-2000 sur la 3ème position

3.8.2 Simulation de la Cinématique inverse :

Dans le cas de la cinématique inverse, nous avons effectué l'opération réciproque, calculant les angles θ_1 et θ_2 d'articulation à partir des coordonnées x, y, z . Les résultats obtenus ont corroboré avec ceux prévus théoriquement, confirmant ainsi l'efficacité de notre approche.

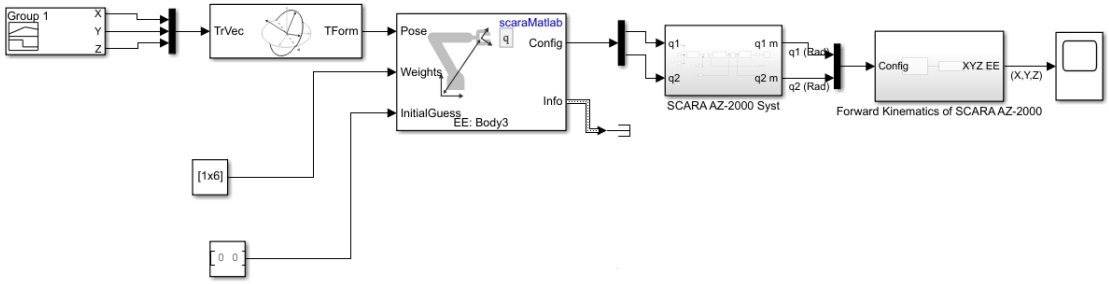


FIG. 3.23 : Schéma de calcul de la cinématique inverse sur Simulink

Scénarios et résultats :

Planification de la trajectoire dans une période de temps bien définie, vers un point désiré $(x, y, z) = (0.25; 0.2; 0.11)$, à $T = 1s$, puis le retour au point de départ à $T = 2s$.

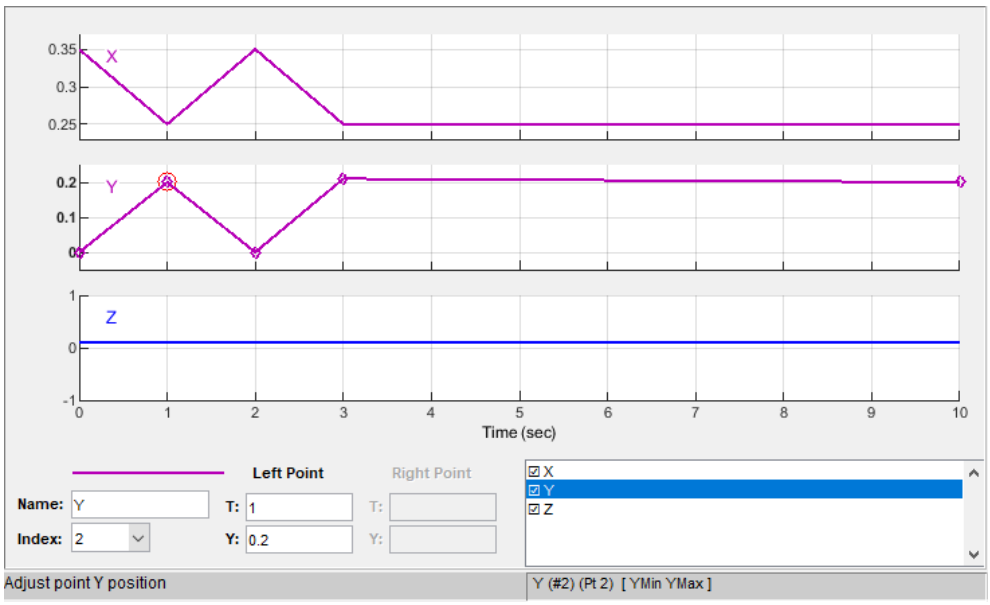


FIG. 3.24 : Configuration des coordonnées du point désiré et planification de la trajectoire dans une période de temps de 2s.

à T=0s : Point de départ :

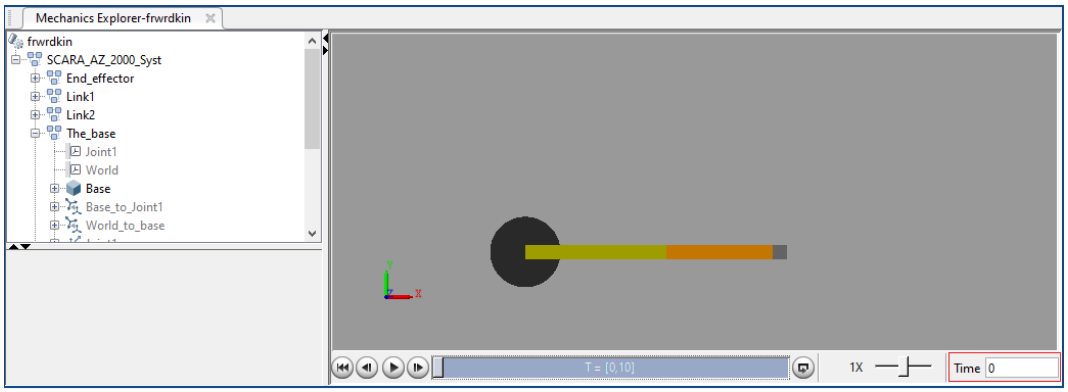


FIG. 3.25 : Point de départ T=0

à T=1,00 s : Arrivée au point désiré :

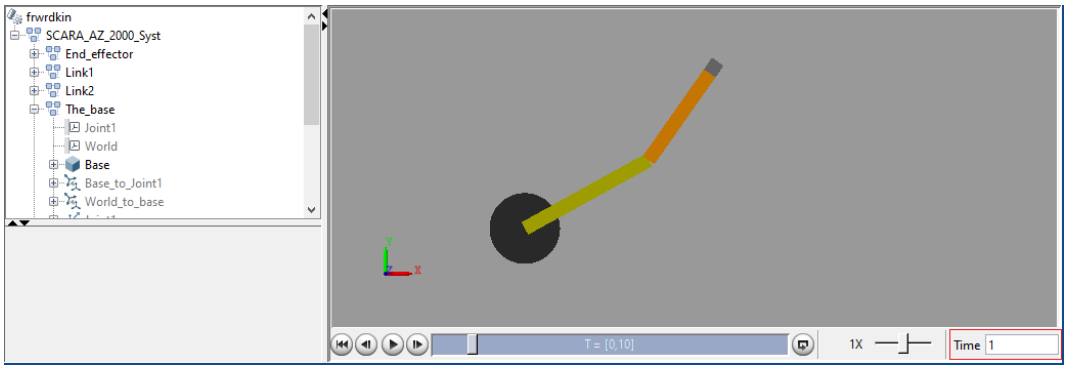


FIG. 3.26 : Arrivée au point désiré T=1

à T=2,00 s : Retour au point de départ :

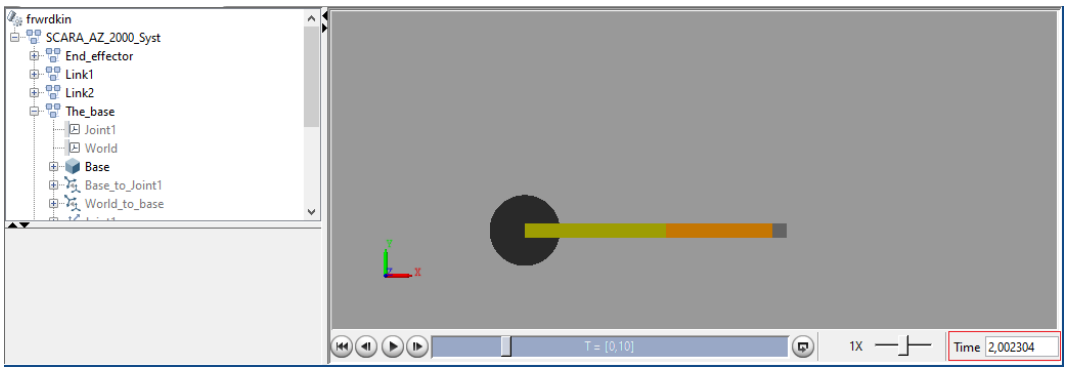


FIG. 3.27 : Retour au point de départ T=2

3.8.3 Discussion et conclusion :

La comparaison entre les résultats de nos simulations et les calculs expérimentaux précédents a confirmé la validité de notre modèle du robot SCARA AZ-2000.

En effet, pour une gamme variée de scénarios simulés, les sorties correspondant à la géométrie directe et inverse du robot étaient en adéquation avec les résultats expérimentaux attendus. Cela atteste non seulement de la justesse de nos calculs de géométrie directe et inverse, mais aussi de la précision de notre modélisation du robot avec MATLAB.
En outre, ces concordances renforcent notre confiance dans l'efficacité de nos méthodes de simulation pour reproduire le comportement réel du robot.
En conséquence, nous pouvons conclure que notre modèle du robot SCARA AZ-2000 a été correctement modélisé et offre une représentation fiable pour les futures études et simulations.

4 Génération de la loi de Commande

Suite à la détermination de l'équation dynamique du système, l'étape suivante consiste à synthétister la loi de commande. Cette loi est essentielle car elle définit la manière dont le système réagit à différentes entrées et perturbations. Elle doit donc être précise et robuste afin de garantir une performance optimale et une grande fiabilité du système dans diverses conditions de fonctionnement.

4.1 Equation d'état du système

Il est nécessaire en premier lieu de générer l'équation d'état du système qui représente une formalisation du comportement dynamique du système. Elle est obtenue à partir de l'équation dynamique préalablement calculée (3.19) en isolant $\ddot{q}_1$ et $\ddot{q}_2$. :

Pour $\ddot{q}_1$:

$$\ddot{q}_1 = \frac{\tau_1 - M[1,2] \cdot \ddot{q}_2 - C[1,1] \cdot \dot{q}_1 - C[1,2] \cdot \dot{q}_2 - G[1]}{M[1,1]} \tag{3.20}$$

Pour $\ddot{q}_2$:

$$\ddot{q}_2 = \frac{\tau_2 - M[2,1] \cdot \ddot{q}_1 - C[2,1] \cdot \dot{q}_1 - G[2]}{M[2,2]} \tag{3.21}$$

On suppose : $\quad x_1 = q_1, \quad x_2 = \dot{q}_1, \quad x_3 = q_3, \quad x_4 = \dot{q}_2$

on aura :

$$\dot{x}_1 = x_2 \tag{3.22}$$

$$\ddot{q}_1 = \frac{\tau_1 - M[1,2] \cdot \ddot{q}_2 - C[1,1] \cdot \dot{q}_1 - C[1,2] \cdot \dot{q}_2 - G[1]}{M[1,1]} \tag{3.23}$$

$$\dot{x}_3 = x_4 \tag{3.24}$$

$$\ddot{q}_2 = \frac{\tau_2 - M[2,1] \cdot \ddot{q}_1 - C[2,1] \cdot \dot{q}_1 - G[2]}{M[2,2]} \tag{3.25}$$

L'équation d'état nous permet de synthétiser la loi de commande.

4.2 Synthèse de la loi de commande

Dans notre cas ne nous sommes pas appelés à synthétiser la loi de commande manuellement, car nous utilisons un outil très puissant :

Robot Operating System (ROS) fournit une infrastructure pour le développement de logiciels pour les robots, et cela inclut plusieurs outils et bibliothèques pour le contrôle des robots et la génération de la loi de commande est prise en charge par les paquets ros control et MoveIt.

"ros control" est un paquet qui fournit une structure pour implémenter des contrôleurs. Il comprend une variété de contrôleurs prédéfinis pour le contrôle de la position, de la vitesse et de l'effort des articulations du robot. Ces contrôleurs utilisent les états actuels du robot et les commandes de l'opérateur pour générer les signaux de contrôle nécessaires. "ros control" peut lire les états du robot à partir de ses capteurs, appliquer la loi de commande, et envoyer les commandes générées aux actionneurs du robot.

MoveIt, d'autre part, est un outil de planification de mouvement. Il peut générer des trajectoires pour le robot à suivre, basées sur une tâche donnée (par exemple, déplacer un bras robotique d'un point à un autre tout en évitant les obstacles). Ces trajectoires peuvent ensuite être suivies en utilisant les contrôleurs de "ros control".

Pour la commande du SCARA AZ-2000 dans ROS, un controleur PID a été synthétisé pour plus de stabilité.

4.3 Contrôleur PID

Un contrôleur PID (Proportional, Integral, Derivative) est un type de contrôleur rétroactif couramment utilisé en contrôle automatique. Il est appelé PID parce qu'il comporte trois composants : Proportionnel, Intégral et Dérivatif.

La sortie globale du contrôleur PID est la somme des sorties proportionnelles, intégrales et dérivatives : $U = P_o + I_o + D_o$.

Dans notre cas, nous avons configuré les gains PID pour trois articulations (joint 1, joint 2 et joint 3) comme suit :

- **Le gain proportionnel (p)** :

$K_p = 100$ Cela signifie que pour chaque unité d'erreur, le contrôleur génère une commande de 100 unités pour corriger l'erreur.

- **Le gain intégral (i)** :

$K_i = 0.01$ Cela signifie que pour chaque unité d'erreur cumulée au fil du temps, le contrôleur génère une commande supplémentaire de 0.01 unités.

- **Le gain dérivatif (d)** :

$K_d = 10$ Cela signifie que pour chaque unité de changement de l'erreur par unité de

temps, le contrôleur génère une commande supplémentaire de 10 unités.

Le "iclamp" est un terme qui limite la contribution du terme intégral pour éviter une saturation excessive de la commande, souvent appelée "effet windup". Dans ce cas, la contribution intégrale est limitée à 1.

En combinant ces gains avec les termes P, I et D, le contrôleur PID génère des commandes pour chaque articulation qui aident à minimiser l'erreur entre la position actuelle de l'articulation et la position désirée.

5 Actionneurs et Capteurs

Pour assurer l'efficacité et la performance optimale de notre système, il est primordial de sélectionner les composants les plus adaptés à nos besoins spécifiques. Les actionneurs, en tant que moteurs du mouvement, et les capteurs, en tant qu'éléments fournissant une rétroaction indispensable, jouent des rôles clés dans la capacité du robot à interagir avec son environnement et à accomplir les tâches pour lesquelles il a été conçu.

5.1 Choix de type des moteurs

Les deux principaux critères de sélection des moteurs sont le couple et la puissance, qui peuvent être déterminés par des calculs basés sur les dimensions et les masses de chaque lien du robot.

5.1.1 Calcul de vitesse de rotation :

La vitesse de rotation d'un moteur est généralement mesurée en radians par seconde (rad/s).

Nous voulons que les moteurs tournent de 90 degrés en 1 seconde, cela équivaut à une vitesse de rotation de $\frac{90}{360} = 0.25$ tours/ seconde. Pour convertir cela en radians par seconde, nous multiplions par 2π :

$$\begin{aligned} \text{Vitesse de rotation} &= 0.25\text{x}2\pi \\ &\approx 1.57\,\text{rad/s} \end{aligned}$$

Donc, nous devrions chercher des moteurs qui peuvent tourner à une vitesse d'environ 1.57 rad/s pour accomplir le mouvement souhaité en 1 seconde.

5.5.2 Calcul des couples :

Ces moteurs doivent être capables de fournir le couple nécessaire pour réaliser les tâches requises. Ce choix repose sur une analyse approfondie de la dynamique du robot, qui inclut l'inertie des bras, la charge à transporter et les forces de résistance comme le frottement dans les joints rotatifs.

a. Phase d'accélération

La première étape du mouvement du robot est l'accélération à partir d'une position immobile. Durant cette phase, le moment d'inertie des bras du robot est un facteur crucial. Le moment d'inertie est calculé selon la formule suivante :

$$I = m\left(\frac{L}{2}\right)^2 \tag{3.26}$$

Où : m est la masse du bras et L est sa longueur.

Application numérique :

$$I_2 = 1.326\,\mathrm{kgm}^2, \qquad \text{et} \quad I_3 = 0.409\,\mathrm{kgm}^2.$$

- **Les couples d'accéleration :**

Le couple nécessaire à l'accélération est déterminé par la formule :

$$T = I\alpha \tag{3.27}$$

Où : $\alpha estl'acclrationangulaire.$

Application numérique :

$$t1_{\mathrm{acc}} = 2.72\,\mathrm{Nm}, \qquad \text{et} \quad t2_{\mathrm{acc}} = 0.64\,\mathrm{Nm}.$$

- **La force de friction dynamique :**

La force de friction dynamique dans les roulements est souvent calculée comme un pourcentage de la force de charge, selon la formule suivante :

$$Ff = \mu d\ m\ g \tag{3.28}$$

Tel que : $\mu d = 0.0015$.

La force de friction devient :

AN :

$$Ff = \mu d\ m\ g = 0,0015\text{x}\ 25.3\text{x}\ 9.81 = 0.37N$$

- **Le couple :**

Le couple nécessaire pour surmonter la force de friction,tel que la longueur de bras est de 1.4 m, serait :

$$Tf = F_f\ r = 0.37\text{x}\ 1.4 = 0.518Nm.$$

Par conséquent, le couple total nécessaire pour l'accélération à partir d'une position d'arrêt est :

$$t1_{\mathrm{total}} = t1_{\mathrm{acc}} + Tf = 3.24\,\mathrm{Nm}, \qquad \text{et} \quad t2_{\mathrm{total}} = t2_{\mathrm{acc}} + Tf = 1.16\,\mathrm{Nm}.$$

b. Mouvement à vitesse constante :

Une fois que le robot a atteint la vitesse de rotation désirée, le couple nécessaire pour l'accélération n'est plus requis. Le seul couple nécessaire à cette étape est celui qui est nécessaire pour surmonter le frottement déja calculé. Alors : $T_f = 0.518$ Nm.

c. Phase de décélération :

Pour arrêter le mouvement, le robot doit décélérer. Le couple nécessaire pour cette phase est le même que celui requis pour l'accélération, mais dans la direction opposée. Cela signifie que les moteurs doivent être capables de générer un couple négatif.

- **Pour le moteur 1 :**

$$t1_{\text{total}} = -3.24Nm.$$

- **Pour le moteur 2 :**

$$t1_{\text{total}} = -1.16Nm.$$

d. Retour à la position initiale :

Le mouvement de retour à la position initiale nécessitera les mêmes couples que ceux décrits ci-dessus, mais dans l'ordre inverse.

5.5.3 Calcul de puissance :

- **Puissance mécanique :**

La puissance mécanique nécessaire pour un moteur est déterminée par le produit du couple par la vitesse angulaire, comme indiqué par l'équation suivante :

$$P = T\ \Omega \tag{3.30}$$

Où : T est le couple et Ω est la vitesse angulaire. Nous avons déjà calculé les couples pour chaque phase du mouvement. La vitesse angulaire est la vitesse de rotation de 90 degrés par seconde, soit $= \frac{\pi}{2}$ rad/s.

Dans la phase d'accélération et de décélération, la puissance nécessaire est maximale.

- Pour le moteur 1 :

$$P_1 = t_{1_{\text{total}}} \left(\frac{\pi}{2}\right) = 5.09\,\text{W}$$

- Pour le moteur 2 :

$$P_2 = t_{2_{\text{total}}} \left(\frac{\pi}{2}\right) = 1.85\,\text{W}$$

- **Puissance électrique :**

Si nous supposons une efficacité de 80%, la puissance électrique requise sera supérieure à la puissance mécanique :

$$P_{1_{\text{electrique}}} = \frac{P1}{0.8} \approx 6.36\,\text{W}$$
$$P_{2_{\text{electrique}}} = \frac{P2}{0.8} \approx 2.31\,\text{W}$$

Nous avons donc besoin d'un moteur capable de tourner dans les deux sens et fournissant un couple maximal de 3.24 Nm.

Pour ce faire, nous avons choisi les moteurs Nema 23 et Nema 17, illustrés respectivement dans les figures 3.28 et 3.29. Ces moteurs développent un couple de 4.8 Nm et 1.9 Nm, ce qui est largement suffisant pour supporter les exigences précédement calculées.



Fig. 3.28 : Moteur Nema 23

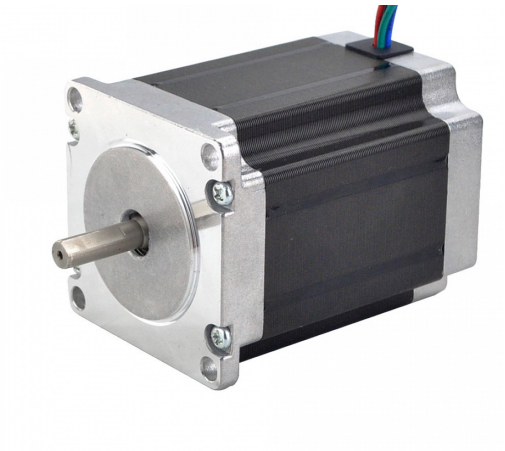


Fig. 3.29 : Moteur nema 17

Ces moteurs sont accompagnés d'un driver, présenté dans la figure 3.31.



Fig. 3.30 : Driver NEMA

Le driver permet de contrôler la vitesse, le sens de rotation et le couple du moteur. Grâce à ce dispositif, nous pouvons optimiser la performance de notre système, tout en garantissant sa sûreté et sa fiabilité.

5.2 Réducteur de vitesse

Les réducteurs de vitesse, sont des mécanismes utilisés dans les systèmes mécaniques pour réduire la vitesse de rotation (RPM) d'un moteur d'entraînement entrant. En réduisant la vitesse de rotation, ils augmentent le couple disponible pour l'application.

5.2.1 Utilité du réducteur de vitesse : Afin d'optimiser les performances de notre robot, nous avons utilisé un réducteur de vitesse pour :

- Réduction de la charge sur le moteur.
- Amélioration de la précision.

5.2.2 Le réducteur de vitesse utilisé

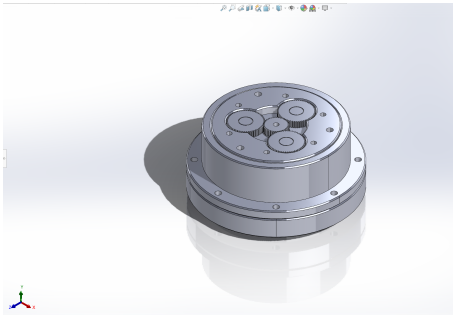


FIG. 3.31 : Le réducteur de vitesse

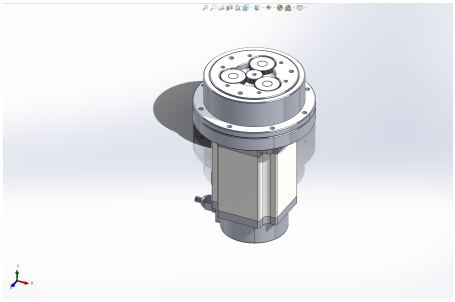


FIG. 3.32 : Réducteur de vitesse avec le moteur NEMA



FIG. 3.33 : SCARA AZ-2000 vue eclatée

5.3 Choix de type de vérin

5.3.1 Calcul de la force du vérin :

La force nécessaire dépend de la masse de l'objet à manipuler et de l'accélération souhaitée. La force est donnée par la formule suivante :

$$F = m\ a \tag{3.31}$$

Où : m est la masse et a est l'accélération.

Si nous supposons une accélération de 1 m/s², qui signifie que le vérin peut atteindre sa pleine extension en environ 1 seconde, la force requise est :

AN :

$$F = 0.0302kg\ 1m/s^2 = 0.0302N$$

C'est une force très faible, donc presque tous les vérins pneumatiques industriels seront capables de la fournir.

5.3.2 Vérin sélectionné :

Un vérin pneumatique à double effet avec une ventouse intégrée, modèle VUOTOTECNICA.



FIG. 3.34 : vérin avec ventouse intégrée

La course du vérin est de 17mm minimum, il est caractérisé par sa haute vitesse.

5.4 Caméra Intel RealSense

La caméra Intel RealSense représenté dans la figure 3.35, est un capteur de vision avancé qui combine une caméra RGB de haute résolution, une caméra de profondeur stéréoscopique et un microphone à matrice de faisceau infrarouge pour fournir une reconnaissance visuelle en trois dimensions [41]. Ce capteur peut effectuer diverses tâches, notamment la détection

des objets, la reconnaissance faciale, et bien d'autres encore.



Fig. 3.35 : Caméra Intel RealSense

Justification du choix

L'utilisation de cette caméra dans ce système est justifiée par sa capacité à lire les images et à communiquer avec le module NPU via USB 3.1. Grâce à sa compatibilité avec les algorithmes de réseaux de neurones convolutifs (CNN), elle est parfaitement adaptée à l'identification des plaquettes de médicaments défectueuses. De plus, elle est robuste et fiable pour une utilisation industrielle.

Son autre avantage majeur est sa faible consommation d'énergie, ce qui est essentiel dans les systèmes automatisés pour réduire les coûts d'exploitation et augmenter l'efficacité énergétique.

5.5 Automate programmable et module NPU :

Nous avons opté pour l'utilisation de l'API de siemens S7-1500, illustré dans la figure 3.36.

5.4.1 Description et justification du choix de cet automate :

Le SIMATIC S7-1500 est un API de Siemens réputé pour sa polyvalence dans de nombreuses applications industrielles. Il est caractérisé par ses performances élevées et sa flexibilité de programmation, adaptées aux tâches d'automatisation les plus simples aux plus complexes.

Une caractéristique essentielle du SIMATIC S7-1500 est sa capacité à intégrer un module d'extension TM NPU spécifiquement conçu pour le traitement des algorithmes d'apprentissage profond (Deep Learning), permet d'exécuter des tâches computationnelles exigeantes directement sur le contrôleur d'automatisation, offrant ainsi une solution d'Edge Computing performante.



Fig. 3.36 : Automate programmable S7-1500

5.4.2 SIMATIC S7-1500 TM NPU de Siemens :

SIMATIC S7-1500 TM NPU est un module d'extension pour les API S7-1500 de Siemens.[42] "TM" signifie Technologie Module et "NPU" pour Neural Processing Unit. Le module NPU est spécifiquement conçu pour le traitement de l'intelligence artificielle (IA) et des algorithmes d'apprentissage profond (Deep Learning) directement dans le contrôleur d'automatisation, c'est-à-dire à la périphérie du réseau (Edge Computing).[43] Il est équipé d'un processeur Intel Movidius Myriad X Vision Processing Unit (VPU) pour le calcul d'IA à haute performance.

Dans le contexte de notre système, le module S7-1500 TM NPU pourrait être utilisé pour exécuter le réseau neuronal convolutif (CNN) qui analyse les images des plaquettes de médicaments. Lorsqu'une image est capturée par la caméra, elle peut être envoyée au module NPU pour traitement. Le module NPU peut alors utiliser le CNN pour classer l'image, déterminer si la plaquette de médicaments est défectueuse ou non, et envoyer cette information au contrôleur principal S7-1500. Sur la base de cette information, le contrôleur peut alors commander le robot manipulateur pour qu'il retire la plaquette de médicaments.

5.6 Alimentation et puissance totale du système :

La détermination du type d'alimentation approprié pour un système robotique nécessite la prise en compte de plusieurs facteurs, tels que la tension requise par les composants du système et leur consommation d'énergie globale.

Supposons initialement une alimentation de 24V DC, typique pour les systèmes de ce type. Pour justifier ce choix, examinons la consommation en puissance des différents composants et calculons le courant nécessaire pour chacun en utilsant loi d'Ohm $P = VI$ .Les résultats de ces calculs seront présentés dans le tableau.

Il est important de noter que les puissances de l'API et du NPU sont obtenues à partir

de leurs caractéristiques techniques respectives. De plus, la consommation de la caméra Intel Real Sense a été considérée comme négligeable pour ces calculs.

	Puissance (W)	Courant (I)
2 Moteurs	8.7	0.325
API	60	2.5
TM NPU	5	0.208
Totale	≈ 73.7	≈ 3.033

Tab. 3.6 : Résultats de calcul

Pour garantir un fonctionnement stable et tenir compte des pics de puissance possibles, une marge de sécurité d'au moins 30% est généralement incluse, ce qui porte la puissance totale nécessaire à environ 95.77W.

Cela justifie le choix initial d'une alimentation de 24V DC capable de fournir au moins 100W, ce qui est largement disponible dans l'industrie. Ainsi, le choix initial de 24V est confirmé par ces calculs

6 Conclusion

Le Chapitre 3 a non seulement servi de base solide pour la compréhension détaillée de notre système de bout en bout, mais a également mis en lumière l'importance d'une étude approfondie avant la phase de réalisation. Ce processus a été imporatnt pour identifier les besoins spécifiques, modéliser le système mathématiquement et choisir les composants appropriés. Cette étude approfondie a permis de jeter les bases d'une mise en œuvre réussie et efficace du système.

En se basant sur cette analyse solide et exhaustive, nous sommes maintenant en mesure d'aller de l'avant et de commencer la phase de controle via un algorithme d'intelligence artificielle et faire l'automatisation et la démarche maintenece du système, sujet du chapitre suivant.

Chapitre 4

Contrôle, Automatisation et Maintenance du système

1 Introduction

Dans ce chapitre, nous allons nous intéresser aux principaux aspects du contrôle, de l'automatisation et de la maintenance du système AZ-2000. A l'aide d'un algorithme de Convolutional Neural Network (CNN), nous effectuerons le contrôle des opérations, tandis que l'automatisation sera gérée par un Automate Programmable Industriel. En outre, nous présenterons une méthodologie efficace pour la maintenance de ce système, garantissant sa fonctionnalité et sa fiabilité dans le temps.

2 Contrôle : Intelligence Artificielle

Pour répondre efficacement au défi de la détection et du tri des médicaments défectueux dans la ligne de production automatisée de SAIDAL, nous avons développé un algorithme d'apprentissage profond (Deep learning) basé sur l'architecture des réseaux de neurones convolutifs (CNN).
Cet algorithme est le cœur de notre système, permettant de traiter les images capturées par la caméra et d'identifier si les médicaments sont défectueux ou non. Grâce à la puissance du deep learning et de la capacité des CNN à apprendre les caractéristiques importantes des images, cet algorithme peut reconnaître avec précision les anomalies sur les médicaments et déclencher les actions appropriées pour les trier.

2.1 Le modèle proposé

La mise en œuvre d'un algorithme de réseau de neurones convolutifs efficace requiert une méthodologie bien définie :

2.1.1 Acquisition des données :

Les images de médicaments défectueux et non défectueux que nous avons utilisées pour entraîner et tester notre modèle CNN proviennent d'une base de données spécifiquement élaborée pour ce projet. Cette base de données a été créée à partir des déchets de la ligne de production, assurant ainsi que notre modèle est formé sur des données qui reflètent fidèlement les conditions réelles auxquelles il sera confronté.

le Nombre des images dans le dataset est 227, voici, ci-aprés, un echantillon :



Fig. 4.1 : Une plaquette défaillante



Fig. 4.2 : Une plaquette non-défaillante

2.1.2 Prétraitement des données :

Le prétraitement des données comprend la normalisation des images en divisant chaque pixel par 255.0. Cette étape est importante pour faciliter l'apprentissage du réseau de neurones.

2.1.3 Architecture de l'algorithme :

Le modèle du CNN proposé est structuré en plusieurs couches distinctes, présenté dans la figure 4.3 ET 4.4 Il comprend quatre couches de convolution (Conv2D) appliquant une fonction d'activation ReLU et une régularisation L2 pour atténuer le sur-apprentissage et améliorer la généralisation du modèle, et intercalées avec des couches de Pooling (MaxPooling2D) qui réduisent la dimensionnalité des cartes de caractéristiques, augmentant ainsi l'efficacité du modèle et sa robustesse aux petites variations de l'entrée.
Après la succession de ces couches de convolution et de Pooling, l'information est aplanie (Flatten) pour transformer la structure tridimensionnelle (RxGxB) des données en un format unidimensionnel approprié à une couche pleinement connectée (Dense).
Le modèle se termine par une série de trois couches denses (Fully connected layers), les deux premières utilisant la fonction d'activation ReLU et une régularisation L2, tandis que la dernière couche, responsable de la sortie du modèle, utilise une fonction d'activation sigmoïde pour effectuer la classification binaire finale.

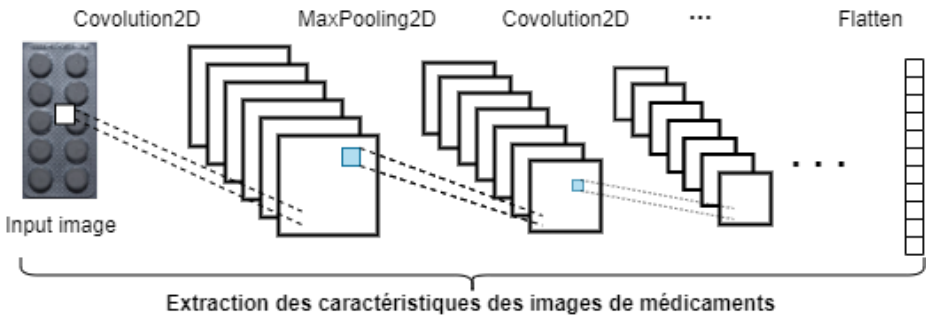


Fig. 4.3 : Architecture de la partie convolutive

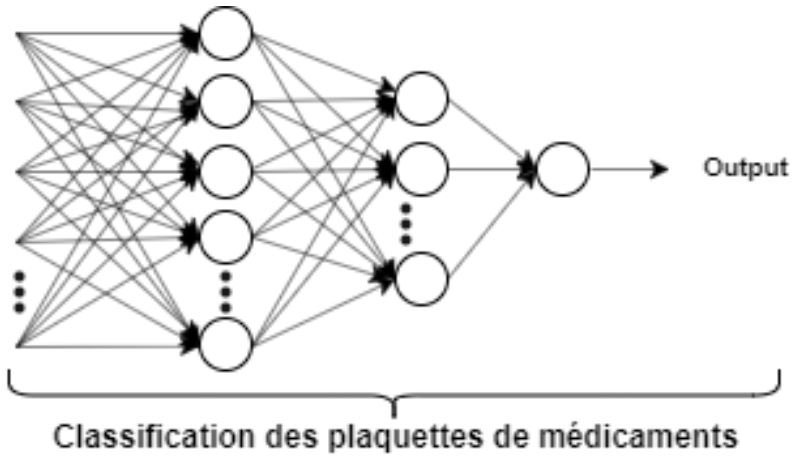


Fig. 4.4 : Architecture de réseau de neurones

Le tableu suivant résume cette architecture avec plus de détail.

Type de couche	**N° de neurones de sortie**	**Taille de noyau**	**Le pas (Stride)**	**Fonction d'activation**
Conv2D	32	3 x 3	1	ReLU
MaxPooling2D	32	2 x 2	2	/
Conv2D	64	3 x 3	1	ReLU
MaxPooling2D	64	2 x 2	2	/
Conv2D	128	3 x 3	1	ReLU
MaxPooling2D	128	2 x 2	2	/
Conv2D	256	3 x 3	1	ReLU
MaxPooling2D	256	2 x 2	2	/
Fully Connected	256	-	-	ReLU
Fully Connected	128	-	-	ReLU
Fully Connected	1	-	-	Sigmoïde

Tab. 4.1 : Architecture détaillé du modèle CNN proposé

2.1.4 Entrainement du modèle :

Dans cette partie, nous avons formé notre modèle CNN avec l'entropie croisée binaire comme fonction de perte et l'optimiseur Adam pour ajuster les poids du réseau. L'entraînement s'est déroulé sur 20 époques, avec des mises à jour des poids effectuées après chaque lot de 27 échantillons.[44] L'entropie croisée binaire est idéale pour notre tâche de classification binaire, tandis que l'optimiseur Adam gère efficacement les gradients de grande dimension. En utilisant 20 époques, nous avons permis à notre modèle d'apprendre les caractéristiques des images à travers plusieurs passages, avec la taille de lot de 27 offrant un bon équilibre entre efficacité computationnelle et qualité d'optimisation.

2.1.5 Résultats et discussion :

L'évaluation de la performance de l'algorithme s'effectue en se basant sur des métriques spécifiques telles que la précision (Accuracy), le score F1 et le rappel (Recall). Ces métriques permettent d'analyser la qualité de la classification binaire réalisée par notre modèle de Convolutional Neural Network (CNN) sur les plaquettes de médicaments défaillants et non défaillants.

- **Précision (Accuracy) :**

La précision est une mesure statistique de l'exactitude de la classification réalisée par le modèle. Elle est calculée comme le rapport entre le nombre de prédictions correctes (vrais positifs et vrais négatifs) et le nombre total de prédictions.

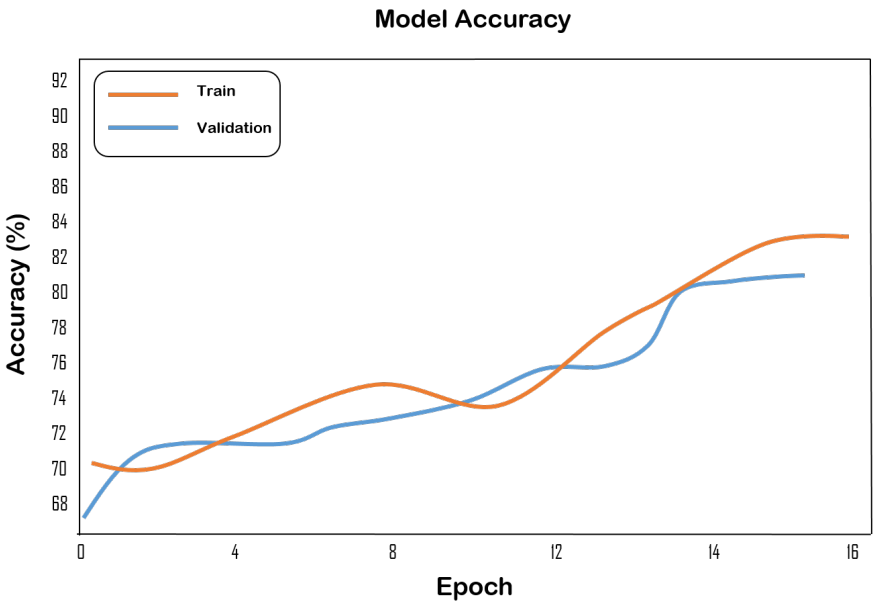


FIG. 4.5 : Évolution de la précision du modèle CNN

La figure 4.5 présente l'évolution de la précision du modèle en fonction du nombre d'epochs pour les ensembles d'entraînement et de validation. Il ressort de cette figure que la précision maximale obtenue sur l'ensemble d'entraînement est de 83.01% et celle sur l'ensemble de validation est de 81.52%.
Bien que ces résultats soient satisfaisants en général, il convient de noter qu'en raison de l'importance de la précision dans le domaine pharmaceutique, ces valeurs sont insuffisantes pour assurer une détection fiable des plaquettes de médicaments défaillantes.

- **F1 Score et Rappel :**

Le score F1 est une moyenne harmonique de la précision et du rappel, il donne donc une vue d'ensemble de ces deux mesures. Il est particulièrement utile lorsque la répartition des classes est inégale.
Le rappel, également connu sous le nom de sensibilité, est une mesure de la capacité du modèle à identifier tous les cas positifs. Il est calculé comme le rapport entre le nombre de vrais positifs et la somme des vrais positifs et des faux négatifs.
Une matrice de confusion est un outil essentiel dans l'évaluation des modèles de classification. Elle donne une vue d'ensemble de la performance du modèle en représentant les vrais positifs, les vrais négatifs, les faux positifs et les faux négatifs. Par conséquent, le score F1 et le rappel sont calculés en utilisant les informations contenues dans la matrice de confusion.

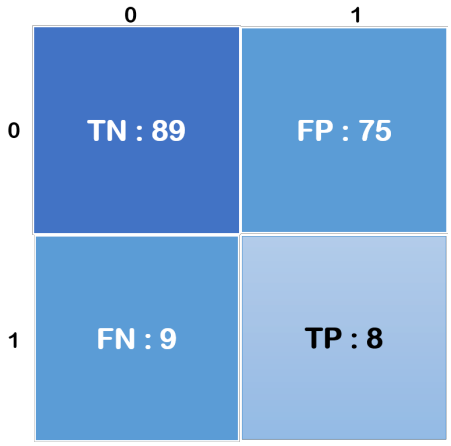


Fig. 4.6 : Matrice de confusion des prédictions du modèle CNN

Pour évaluer le score F1 et le rappel, nous nous référons à la matrice de confusion présentée dans la figure 3.6 Dans notre cas, le nombre limité de vrais positifs (8) par rapport aux faux négatifs (9) suggère que le rappel et le score F1 du modèle sont inférieurs à nos attentes. Plus précisément :

$$\text{Rappel} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{4.1}$$

Dans notre cas, cela donnerait :

$$\text{Rappel} = \frac{8}{8+9} = 0.47 = 40\% \tag{4.2}$$

$$\text{F1 Score} = 2\frac{\text{PrécisionRappel}}{\text{Précision} + \text{Rappel}} \tag{4.3}$$

Avec nos valeurs, cela donne :

$$\text{F1 Score} = 2\frac{0.540.47}{0.54 + 0.47} = 0.50 = 50\% \tag{4.4}$$

2.1.6 Conclusion :

Les résultats obtenus révèlent que le modèle actuel, malgré sa performance correcte, n'atteint pas les exigences élevées du domaine pharmaceutique en termes de précision. Il est donc essentiel de l'améliorer pour atteindre un niveau de précision d'au moins 96%, avant de le déployer dans un environnement de production, où le niveau de tolerence est trop faible.

2.2 Amélioration du modèle

Dans cette approche, nous avons d'abord repréparé notre ensemble de données, qui se compose d'images de plaquettes de médicaments. L'ensemble de données a été augmenté pour contenir 1127 images, car plus de données entraînent généralement une meilleure performance du modèle. Les images ont été chargées à partir de répertoires spécifiques, redimensionnées à une taille de (480, 720), et ont été transformées en tableaux NumPy pour une manipulation

plus facile. Les images ont également été normalisées en divisant chaque pixel par 255, ce qui améliore la vitesse de convergence de l'apprentissage. Et en vue d'améliorer les performances de notre modèle, nous avons opté pour l'utilisation du modèle VGG16 de CNN, reconnu pour sa capacité à atteindre une précision élevée dans des tâches de classification d'images

2.2.1 Architecture du modèle VGG16 :

VGG16 est une architecture de CNN pré-entraînée sur l'ensemble de données ImageNet. En raison de son apprentissage préalable sur un large ensemble de données d'images, il possède une bonne capacité à extraire des caractéristiques pertinentes des images. [45]
IL diffère des CNN traditionnels en ce qu'il a une architecture beaucoup plus profonde, avec un total de 16 couches pondérées. Cela comprend 13 couches convolutives, suivies de 3 couches entièrement connectées (Fully connected layers) préssenté dans la figure 4.4.

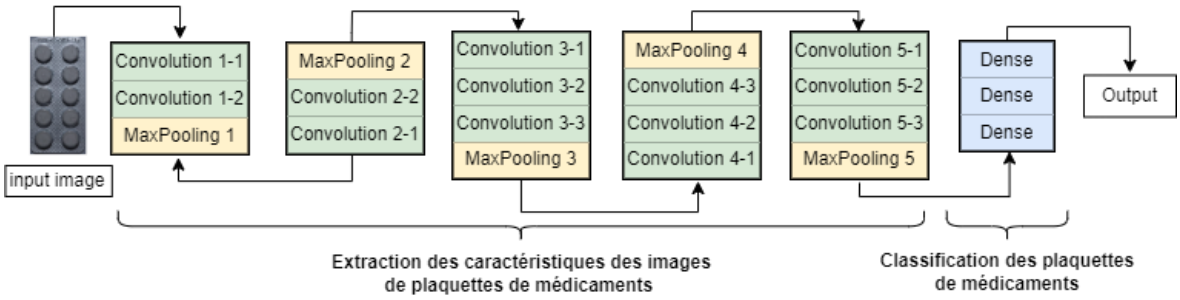


Fig. 4.7 : Architecture du modèle CNN amélioré : VGG16

2.2.2 Entraînement du modèle :

Nous avons utilisé le générateur ImageDataGenerator pour augmenter d'avantage nos données en introduisant des variations aléatoires dans les images, ce qui permet d'améliorer la généralisation du modèle. nous avons également utilisé une descente de gradient stochastique avec un taux d'apprentissage de 0.01 et un momentum de 0.9 comme optimiseur. Le modèle est compilé avec une fonction de perte de cross-entropy binaire, ce qui est approprié pour une tâche de classification binaire, et il utilise l'optimiseur SGD avec une vitesse d'apprentissage de 0.01 et un momentum de 0.9. Le modèle utilise une série de rappels, notamment EarlyStopping et ReduceLROnPlateau, pour arrêter l'entraînement si le modèle cesse de s'améliorer et pour réduire le taux d'apprentissage si la perte de validation cesse de diminuer.

2.2.3 Résultats et discussion :

Le modèle est évalué également à l'aide de la précision, du rappel, du score F1 et de la matrice de confusion

- **Précision (Accuracy) :**

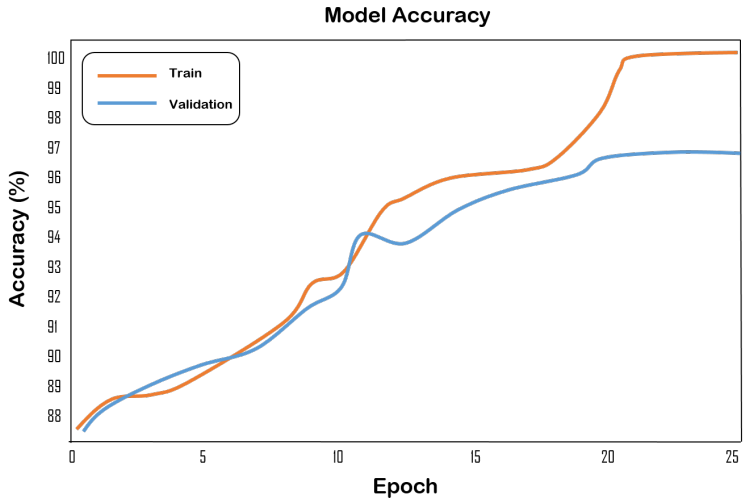


FIG. 4.8 : Evolution de la précision du modèle CNN VGG16

La figure 3.44 présente l'évolution de la précision du modèle sur les ensembles d'entraînement et de validation. Le modèle amélioré montre une performance impressionnante avec une précision d'entraînement de 99.64% et une précision de validation de 96.37%. Ces résultats démontrent l'efficacité du modèle amélioré CNN VGG16, atteignant notre objectif d'une précision d'au moins 96%.

- **F1 Score et Rappel :** La matrice de confusion présentée dans la figure 3.45 illustre l'efficacité remarquable du modèle amélioré. Le modèle n'a fait aucune erreur de classification, comme en témoignent les valeurs nulles pour les faux positifs (FP=0) et les faux négatifs (FN=0). En conséquence, le rappel et le score F1 atteignent la perfection, c'est-à-dire 1 ou 100%, ce qui indique que le modèle est capable d'identifier correctement tous les cas positifs et négatifs.

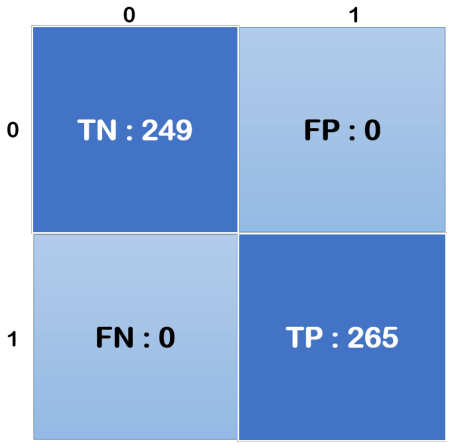


FIG. 4.9 : Matrice de confusion des prédictions du modèle CNN VGG16

2.2.4 Résultats et Discussion :

Les résultats obtenus démontrent que le modèle amélioré CNN VGG16 a atteint les exigences élevées du domaine pharmaceutique en termes de précision, de score F1 et de rappel. Par conséquent, nous pouvons valider ce modèle comme étant efficace pour la classification binaire des plaquettes de médicaments défaillants et non défaillants. L'adoption de cette approche devrait permettre d'assurer une détection fiable des plaquettes défaillantes, contribuant ainsi à garantir la qualité et la sécurité des produits pharmaceutiques.

3 Automate Programmable

Cette partie explore en détail le processus de commande du robot manipulateur de tri SCARA AZ-2000 via un automate programmable industriel (API). L'automate, interfaçant avec le module NPU, commande les actions du robot, assurant une interaction fluide entre l'intelligence artificielle et les composants mécaniques du robot.

3.1 Principe de fonctionnement

L'automate programmable reçoit un signal du module NPU, qui déclenche la séquence de d'élimination de la plaquette défaillante. À l'état initial, le robot est situé sous le convoyeur et prêt à interagir avec les plaquettes de médicaments. Dès la réception du signal, le vérin pneumatique s'actionne en premier. Suite à un délai de 5 secondes, nécessaire pour assurer la précision du placement, les moteurs M1 et M2 sont activés. Le moteur M1 effectue une rotation de 30 degrés pour aligner le bras du robot avec le bac de tri de médicament sélectionnée, et simultanément, le moteur M2 effectue une rotation de 60 degrés. Cette coordination permet au robot de déplacer la plaquette défaillante vers le bac de tri correspondant. Une fois l'élimination effectuée, une seconde rotation est réalisée : M1 et M2 reviennent à leur état initial, repositionnant ainsi le robot sous le convoyeur et prêt pour le prochain cycle de tri.

3.2 Grafcet du système

Le GRAFCET (GRAphe Fonctionnel de Commande Étape-Transition), est un outil de modélisation de systèmes de commande séquentiels. Il s'appuie sur une représentation graphique pour décrire de manière détaillée la séquence des opérations à effectuer par le système de commande. Dans le contexte de bras manipulateur de tri, le GRAFCET montré ci-aprés sert à décrire la séquence de fonctionnement, du déclenchement initial du vérin à la rotation des moteurs et à leur retour à l'état initial.

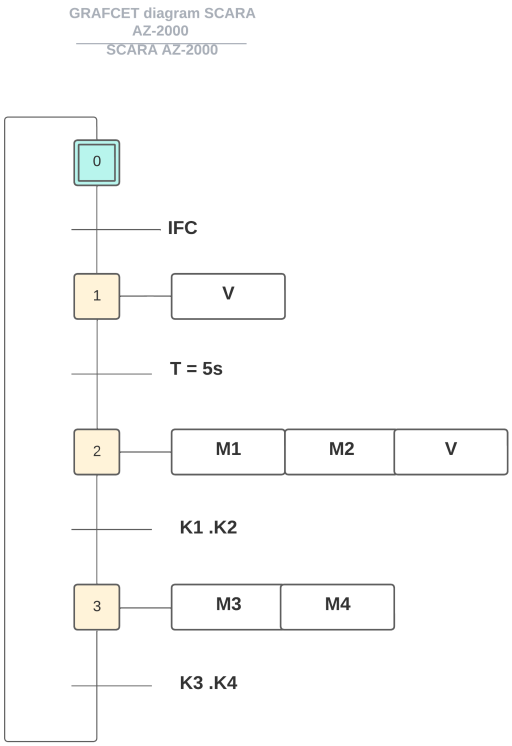


Fig. 4.10 : Le grafcet de commande du robot SCARA AZ-2000

3.3 Équations de Set et Reset des bascules

Les équations de Set (mise à 1) et de Reset (mise à 0) sont utilisées pour contrôler l'état des bascules dans un système automatisé. Une bascule est un type de circuit qui a deux états stables et peut être utilisé pour stocker l'état d'un bit d'information. L'équation de Set est utilisée pour mettre la bascule dans un état, tandis que l'équation de Reset est utilisée pour la remettre dans l'autre état. Les équations ci-aprés permettent de contrôler l'état des différents composants mécaniques, assurant que chaque partie du système fonctionne en synchronisation avec les autres.

$$\begin{cases} S_0 = X_3 \cdot K_3 \cdot K_4 + \text{Reset} \\ R_0 = X_0 \cdot \text{IFC} \end{cases}$$

$$\begin{cases} S_1 = X_0 \cdot \text{IFC} \cdot \overline{\text{Reset}} \\ R_1 = X_1 \cdot T + \text{Reset} \end{cases}$$

$$\begin{cases} S_2 = X_1 \cdot T \cdot \overline{\text{Reset}} \\ R_2 = X_2 \cdot K_1 \cdot K_2 + \text{Reset} \end{cases}$$

$$\begin{cases} S3 = X_2 \cdot K_1 \cdot K_2 \cdot \overline{\text{Reset}} \\ R3 = X_3 \cdot K_3 \cdot K_4 + \text{Reset} \end{cases}$$

3.4 Table de Variables entrée /sortie

La table suivante présente une liste détaillée des variables d'entrée et de sortie utilisées dans la programmation Ladder. Cette table permet de clarifier la fonction de chaque variable pour une meilleure compréhension du programme :

Table de variables standard_1

	Nom	de données	Adresse	Réma...	Acces...	Ecritu...	Visibl...	Surveillan...	Commentaire
1	IFC		%I0.0		☑	☑	☑		Info camera
2	KM1		%M1.1		☑	☑	☑		Relai Moteur 1
3	KM2		%M1.2		☑	☑	☑		Relai Moteur 2
4	KM3		%I0.2		☑	☑	☑		Relai Moteur 1 sens 2
5	KM4		%I0.3		☑	☑	☑		Relai Moteur 2 sens 2
6	KR		%Q0.2		☑	☑	☑		Relai Verin
7	LT1		%M1.3		☑	☑	☑		Relai Timer
8	M1		%Q0.0		☑	☑	☑		Moteur 1
9	M2		%Q0.1		☑	☑	☑		Moteur 2
10	M3		%Q0.4		☑	☑	☑		Moteur 1 sens 2
11	M4		%Q0.5		☑	☑	☑		Moteur 2 sens 2
12	RESET		%I0.1		☑	☑	☑		Reset
13	T	er	%T1		☑	☑	☑		Timer
14	V		%Q0.3		☑	☑	☑		Verin
15	X0		%M1.4		☑	☑	☑		Etat 0
16	X1		%M1.5		☑	☑	☑		Etat 1
17	X2		%M1.6		☑	☑	☑		Etat 2
18	X3		%M1.7		☑	☑	☑		Etat 3

FIG. 4.11 : La table des variables entrée/sortie

3.5 Programmation en langage Ladder

Dans le cadre de ce projet, nous avons choisi d'utiliser le langage de programmation Ladder pour programmer l'API S7-300 qui commande le mouvement de robot. Le langage Ladder, avec ses origines dans les schémas de relais électromécaniques, offre une représentation graphique de la logique booléenne qui est intuitivement compréhensible par les techniciens et les ingénieurs.

Le programme LADDER :

Le langage Ladder est idéal pour décrire les opérations logiques séquentielles et les interrelations entre différents systèmes et sous-systèmes dans le SCARA AZ-2000.

Les images ci-après illustrent les 13 réseaux de programmation qui ont été conçus pour orchestrer les diverses fonctionnalités .

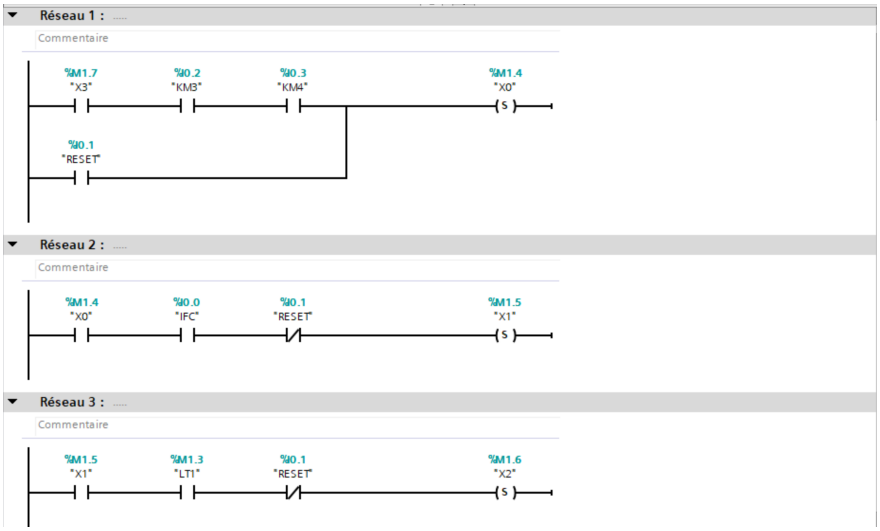


FIG. 4.12 : Réseaux 1,2 et 3.

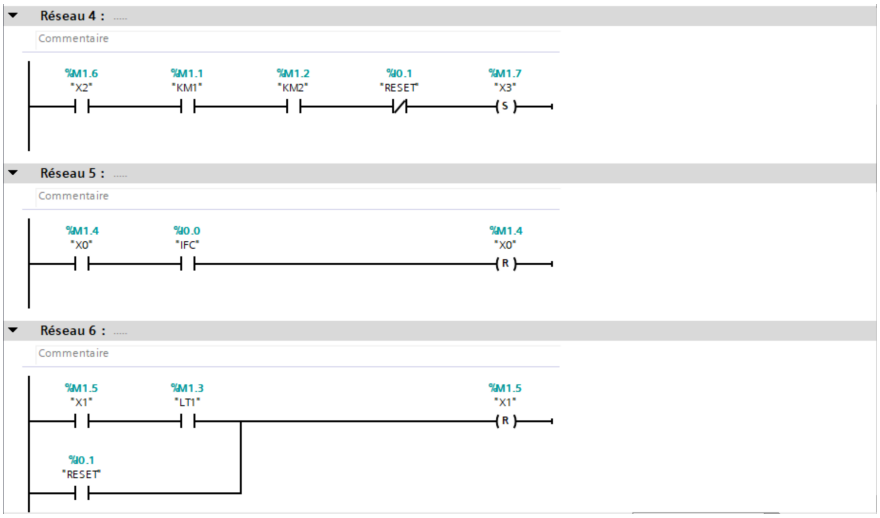


FIG. 4.13 : Réseaux 4, 5 et 6.

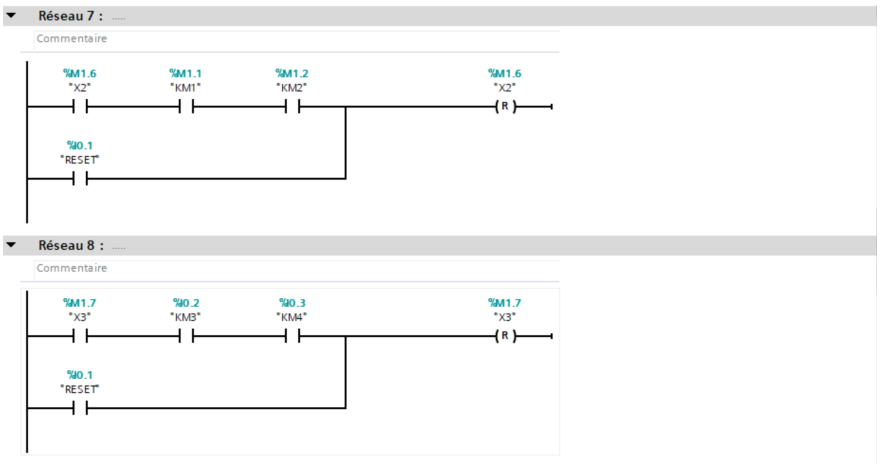


FIG. 4.14 : Réseaux 7 et 8.

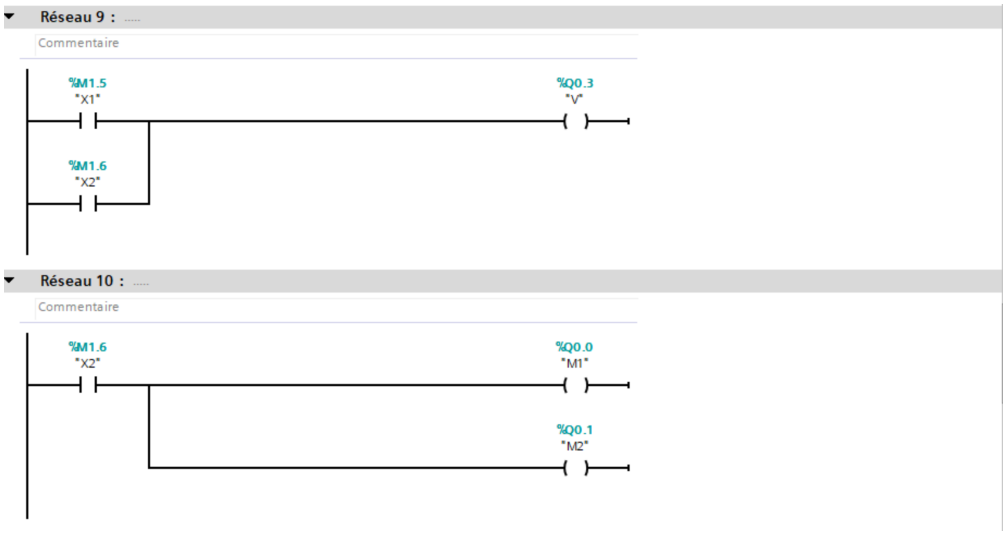


FIG. 4.15 : Réseaux 9 et 10.

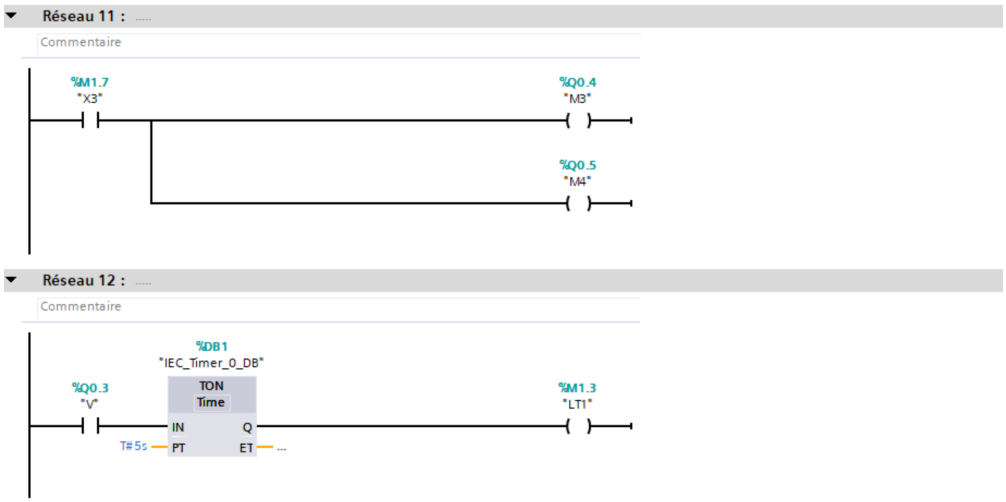


FIG. 4.16 : Réseaux 11 et 12

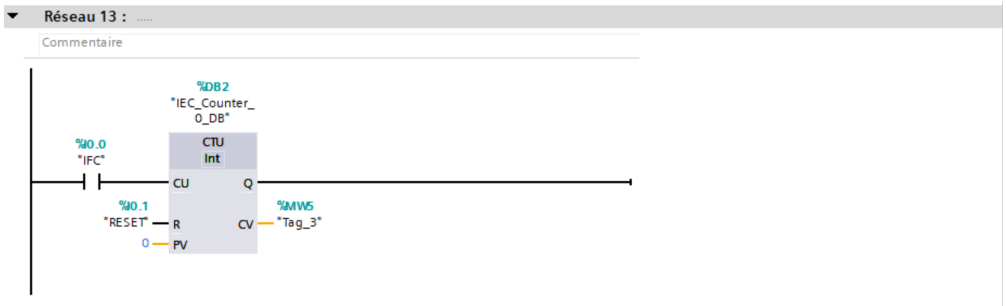


FIG. 4.17 : Réseau 13.

3.6 Interface Homme-Machine (Ihm)

L'interface homme-machine (HMI) est une composante essentielle du système, elle permet aux opérateurs de contrôler et de visualiser le fonctionnement du robot en temps réel. C'est la principale interface entre l'utilisateur et le robot manipulateur de tri.

L'HMI est en contact direct avec l'API, recevant et envoyant des informations pour surveiller et contrôler le statut du système. Il donne aux opérateurs la possibilité de démarrer, d'arrêter, de surveiller et de régler les opérations du bras à tout moment.

Le design de l'interface a été réalisé en tenant compte de l'ergonomie et de la facilité d'utilisation. Des indicateurs visuels permettent de connaître en un coup d'œil l'état actuel du système, incluant l'état des moteurs et du vérin, la position du robot, ainsi que l'état du processus de tri. Des boutons de commande sont également disponibles pour démarrer, arrêter ou modifier le fonctionnement du robot.

L'interface Homme-Machine SCARA AZ-2000

Les images qui suivent offrent un aperçu détaillé de l'Interface Homme-Machine (IHM) du système, illustrant l'efficacité de notre approche en matière de conception d'interface pour une interaction utilisateur intuitive et fluide

a. Vue racine : c'est la partie initiuale de notre HMI, à partir d'ici nous pouvons accéder aux différents pages de notre système.

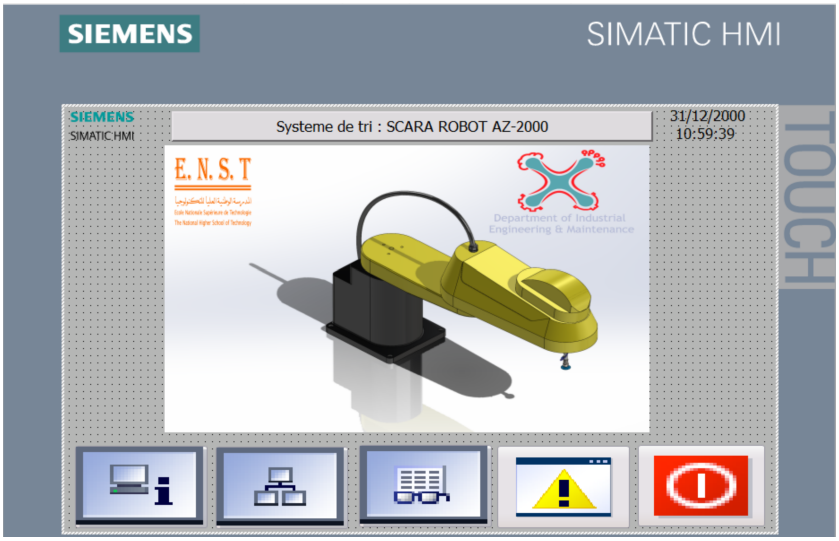


FIG. 4.18 : Vue racine de l'interface Homme-Machine de SCARA AZ-2000

b. Page 1 :

Dans cette page nous aurons la possibilité de visualiser les défaillances détectées et le nombre en temps réel. Le bouton d'arrêt d'urgence est présent dans toutes les pages.

c. Page 2 : Dans cette page nous aurons accès aux types de défaillances détectées et éliminées par le système, avec la date et l'heure de détection de la défaillance.

d. Page 3 : Dans cette page nous aurons la possibilité de visualiser l'états des actionneurs du robot en marche (La direction des moteurs 1 et 2 ainsi que l'actionnement du vérin)

e. Page 4 : Lorsqu'on clique sur le bouton d'arrêt d'urgence, une page indiquant que le système d'arrêt d'urgence est actif sera affichée.

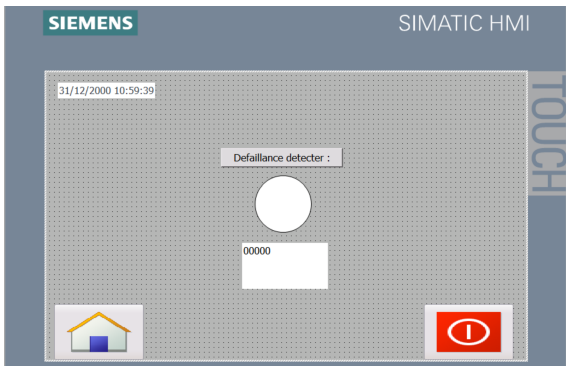


FIG. 4.19 : Page 1 de l'interface Homme-Machine de SCARA AZ-2000

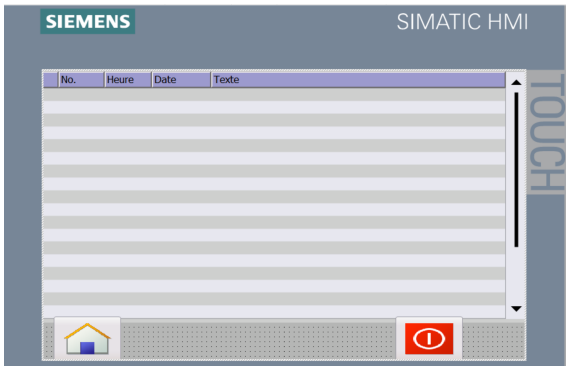


FIG. 4.20 : Page 2 de l'interface Homme-Machine de SCARA AZ-2000

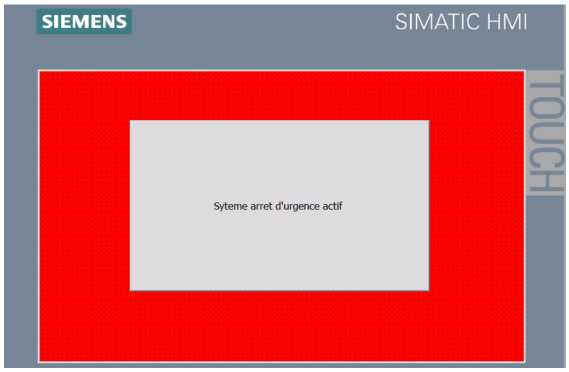


FIG. 4.21 : Page 3 de l'interface Homme-Machine de SCARA AZ-2000

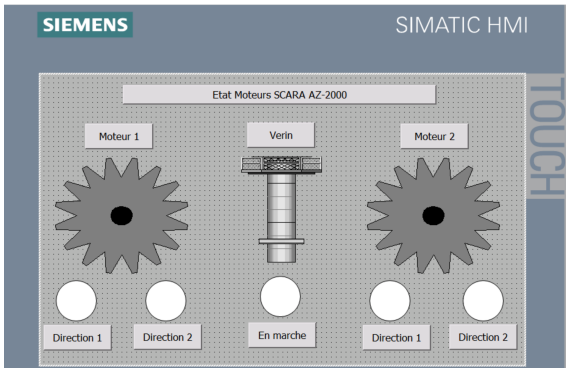


FIG. 4.22 : Page 4 de l'interface Homme-Machine de SCARA AZ-2000

3.7 Simulation et visualisation du programme

Dans cette partie nous allons simuler notre programme LADDER, ainsi que sa connexion avec l'IHM.

Le standard de connexion entre API et IHM est MPI.

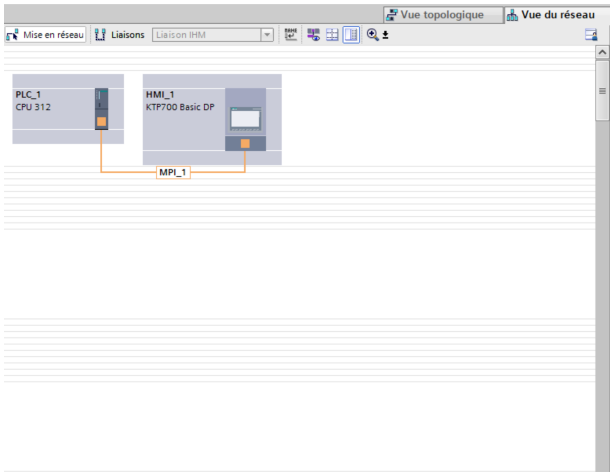


FIG. 4.23 : Connexion API-IHM

- **Mode RESET :**

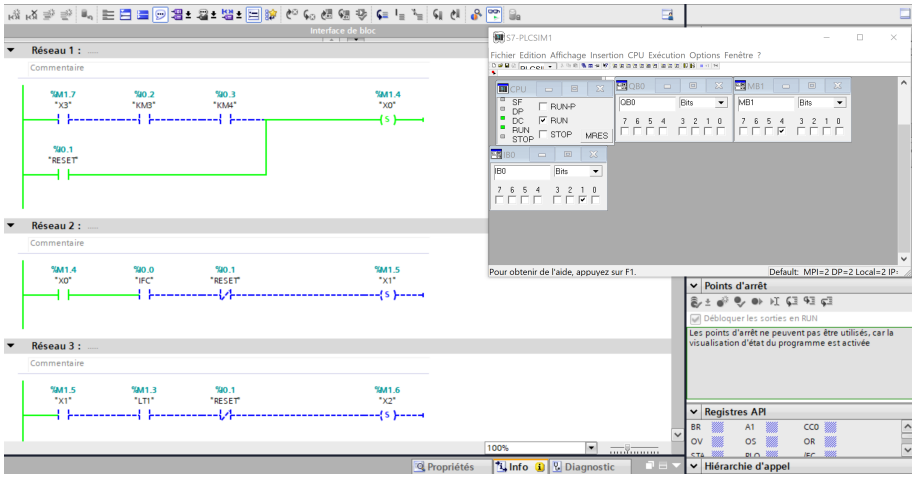


Fig. 4.24 : Mode RESET

- **Mode SET : Activation Verin**

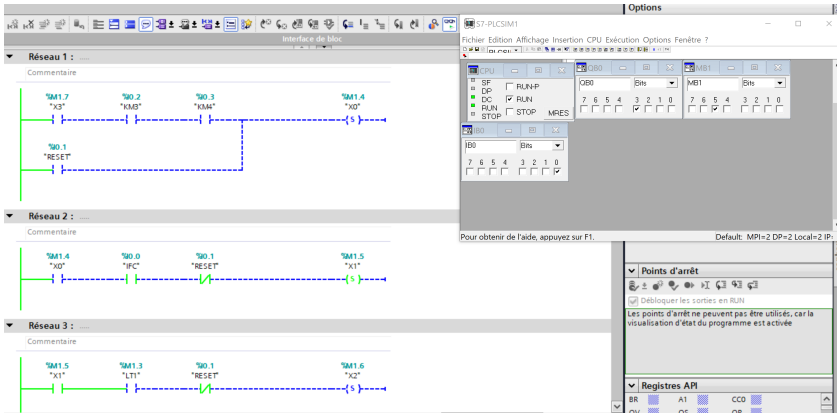


Fig. 4.25 : Activation X1

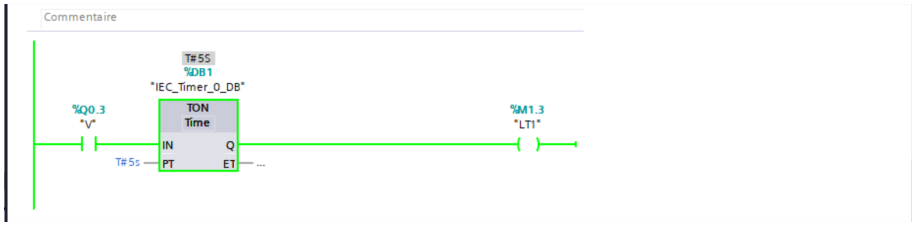


Fig. 4.26 : Activation Vérin + Timer 5s

- **Activation des moteurs**

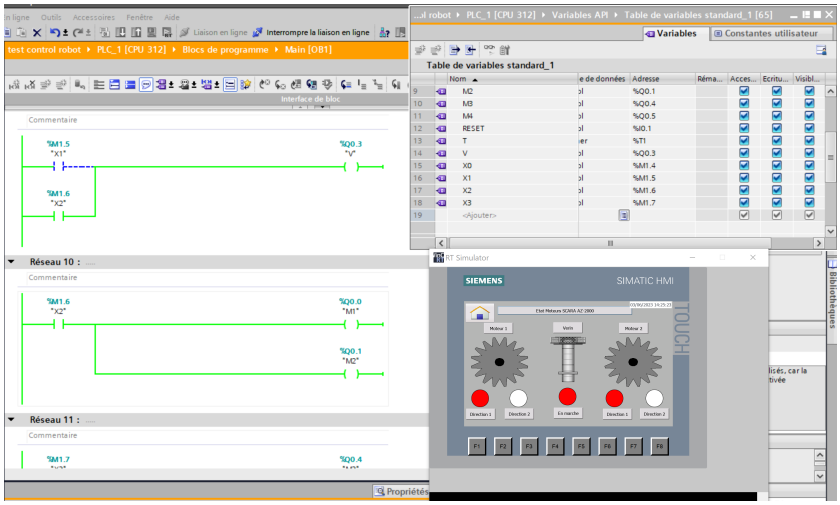


FIG. 4.27 : Activation des Moteurs

- **Désactivation Vérin, Moteurs en Marche**

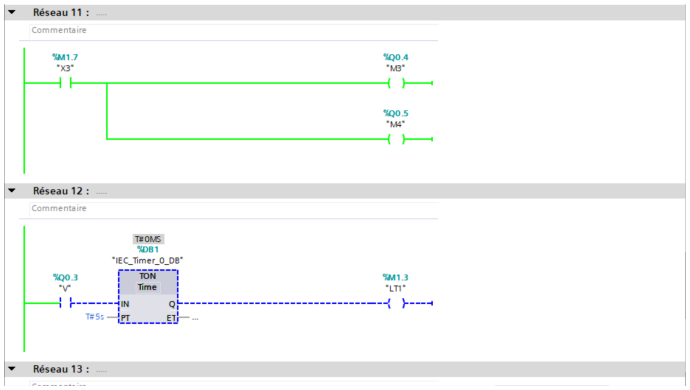


FIG. 4.28 : Désactivation Vérin, Moteurs en Marche

- **Retour à la position initiale**

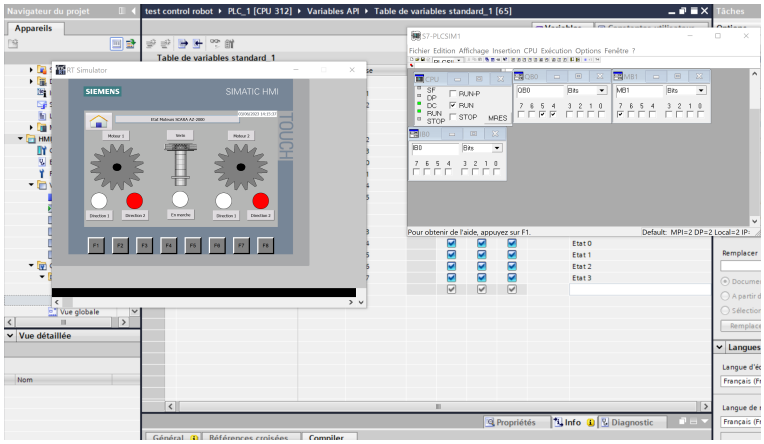


FIG. 4.29 : Retour à la position initiale

Arret d'urgence

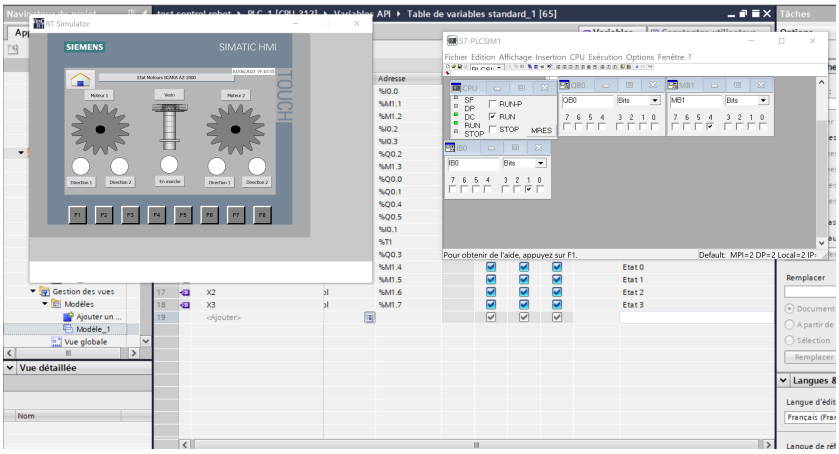


FIG. 4.30 : arrêt d'urgence mode RESET

3.8 Schémas de Commande et de Puissance

Les schémas de commande et de puissance montrent comment l'énergie est distribuée dans le système pour réaliser chaque action. Ils mettent en évidence les relations entre le contrôleur et le robot.

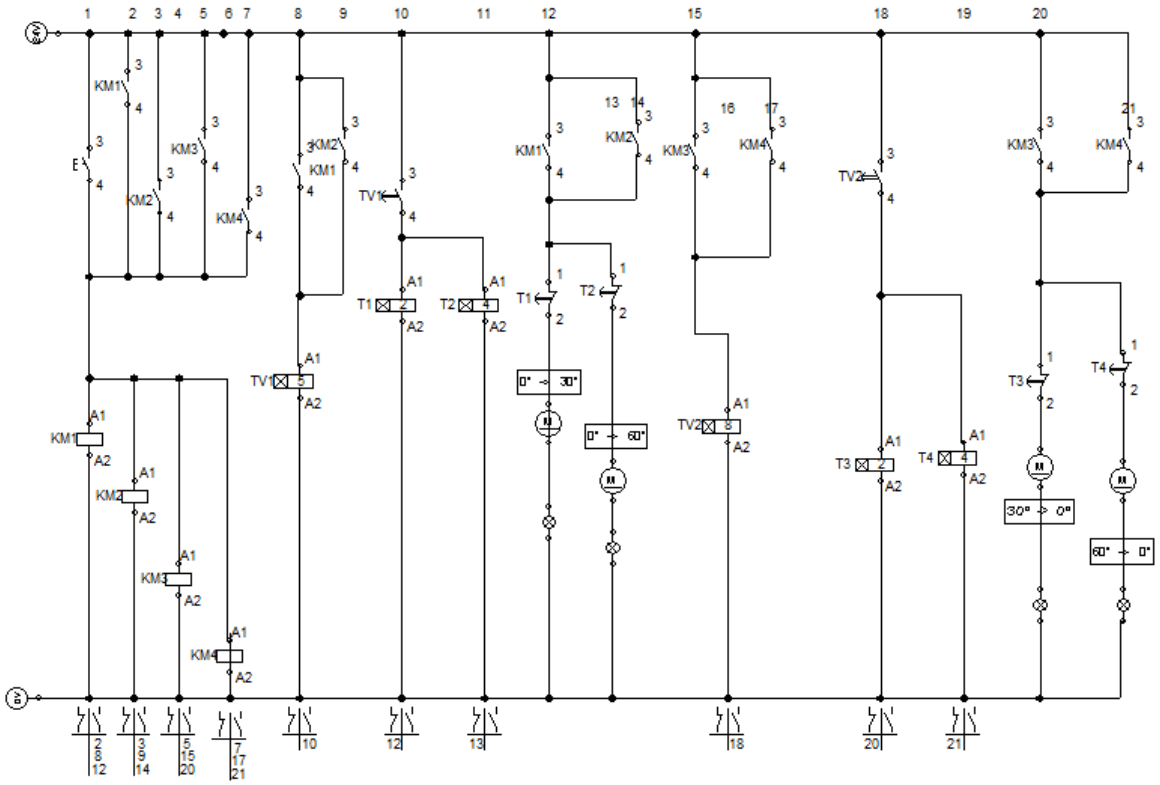


FIG. 4.31 : Schéma électrique du controle des moteurs à 2 sens de rotation

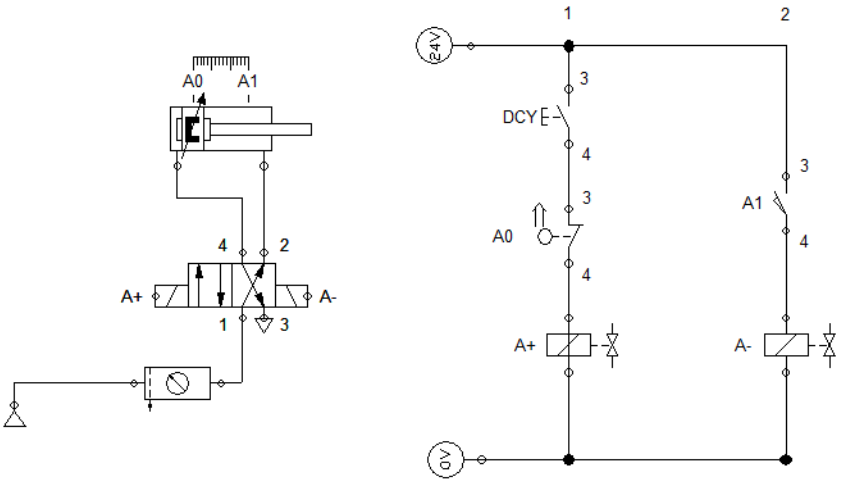


FIG. 4.32 : Pilotage eléctro-pneumatique du vérin à double effet

4 Gestion et organisation de la maintenance

Selon la définition de l'AFNOR, la maintenance industrielle est un ensemble d'actions permettant de maintenir ou de rétablir un bien dans un état spécifié ou en mesure d'assurer un service déterminé. Cette définition s'applique également aux systèmes automatisés intelligents robotisé, où la maintenance assume une importance cruciale. Elle consiste à mettre en œuvre une série de mesures techniques et organisationnelles visant à garantir la performance optimale et la longévité d'un système robotique.

Ce système intelligent de tri est un assemblage complexe de composants matériels et logiciels qui doivent travailler en harmonie pour atteindre les objectifs fixés. Tout dysfonctionnement de ces composants peut sérieusement entraver la performance du système.

Par conséquent, une stratégie de maintenance bien conçue et rigoureusement appliquée est essentielle, et pour cela, nous suivrons une approche méthodique, combinant des techniques de maintenance préventive et corrective pour maintenir l'efficacité du robot SCARA AZ-200, améliorer sa fiabilité et prolonger sa durée de vie.

4.1 Maintenance Préventive

C'est une maintenance exécutée à des intervalles prédéterminés ou selon des critères prescrits ou déterminés et destinée à réduire la probabilité de défaillance ou la dégradation du fonctionnement d'un bien.

Nous avons opté pour un plan de maintenance préventive pour notre système robotique SCARA AZ-2000. comprend une série d'activités de maintenance prédéctive et systématique régulières pour chaque composant du système.

4.1.1 Maintenance systématique :

Un plan de maintenance préventive systématique détaillé dans le tableau 3.8, pour le système SCARA AZ-2000 a été mise en place. Il comprend une série d'activités de maintenance régulières pour chaque composant du système.

PLAN DE MAINTENANCE		
Installation : SYTEME DE TRI	Référence : AZ-2000	**Type : Systématique**

Désignation/Action	**Périodicité**	**Remarque**
Vérification mécanique		
Nettoyage/Lubrification des moteurs	1/ mois	
Vérifier les points à graisser	1 /semaine	
Vérifier corrosion	1/semaine	
Contrôle état du tapis de convoyeur	1/mois	
Inspecter les articulations pour détecter tout signe d'usure.	1/mois	
Vérifier et ajuster le serrage des boulons et des fixations	1/semaine	
Nettoyer les composants pour enlever la poussière et les débris.	1/semaine	
Vérification Pneumatique		
Nettoyage filtre pompe à vide	1/semaine	
Contrôle état de ventouse	1/semaine	
Contrôle filtre à air	1/semaine	
Contrôle pression d'air (8 bars)	1/jour	
Contrôle état des tuyaux d'airs	1/semaine	
Contrôle de lubrification des joints	1/semaine	
Contrôle de lubrification de la tige de piston	1/semaine	
Vérification des raccords pneumatiques	1/jour	
Contrôle de ressort de rappelle	1/jour	
Vérifier le bon fonctionnement des capteurs de pression.	1/jour	
Vérification Electrique		
Contrôle des capteurs	1/semaine	
Essai arrêt d'urgence/automatique	1/semaine	
Contrôle intensité moteurs	1/mois	
Vérifier mise à la terre	1/jour	
Contrôle état des câbles électriques	1/mois	
Contrôle des voyants et boutons	1/semaine	
Contrôle de l'armoire électrique	1/semaine	
Vérifier réglage protection	1/jour	
Vérification système informatique		
Nettoyer et vérifier le bon fonctionnement de la caméra et la qualité de l'image capturée	1/mois	
Vérifier la sécurité du réseau et du système	1/semaine	
Tester régulièrement les performances de l'algorithme	1/mois	
Sauvegarder régulièrement les données et les paramètres du système.	1/jour	

	Vérificateur	Approbateur
Nom		
Signature		

Tab. 4.2 : Plan de la maintenance

4.1.2 Maintenance prédictive :

Dans le but de renforcer la maintenance préventive, l'adoption d'une maintenance prédictive a été envisagée. [46] Cette approche se manifeste par la surveillance en temps réel de l'état du matériel, en se reposant sur des techniques avancées comme la méthode hybride : MDSTVTF+AANN. [47]

La méthode MDSTVTF, ou la fonction de transition d'état variable discret modifiée, est une technique statistique qui exploite des modèles de transition d'états pour anticiper les

variations dans le comportement du système. Cela permet d'isoler les dysfonctionnements potentiels dans le processus de fabrication en scrutant les changements dans les données de performance du système.

Quant aux réseaux neuronaux auto-associatifs (AANN), il s'agit d'une forme de réseau neuronal artificiel qui peut apprendre une représentation des données d'entrée en l'absence de données de sortie supervisées. Grâce à cette approche, il est possible de modéliser le comportement normal du système et d'identifier les anomalies en comparant les nouvelles données aux modèles établis.

La synergie MDSTVTF+AANN tire parti des forces de ces deux techniques, utilisant les réseaux neuronaux auto-associatifs pour discerner les anomalies par rapport au comportement du système, et les modèles de transition d'états pour prédire les évolutions dans ce comportement. Cette démarche permet une détection et un diagnostic plus précis des défaillances, contribuant ainsi à une gestion de la maintenance plus proactive et plus efficace.

4.2 Maintenance Corrective

Pour compléter notre stratégie de maintenance, nous avons également élaboré une maintenance corrective, qui sera exécuté après détection d'une panne et destinée à remettre un bien dans un état dans lequel il peut accomplir une fonction requise. en cas de panne.

Il est bien évident que la maintenance corrective ne peut pas être planifiée car elle est imprévisible et imprévue. Mais en se basant sur le plan de la maintenance préventive déjà élaboré, nous avons mis en place une stratégie de réponse à la maintenance des pannes présenté dans le diagramme suivant :

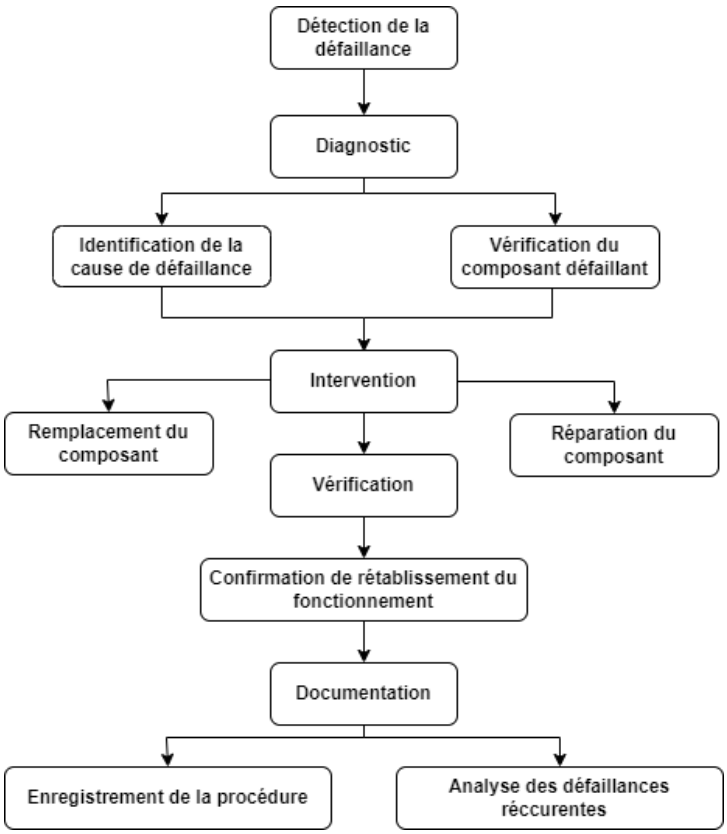


Fig. 4.33 : Démarche de la maintenance corrective

4.3 Gestion des pièces de rechange :

L'une des considérations essentielles pour maintenir la performance et la fiabilité d'un système est la gestion efficace des pièces de rechange. Une bonne gestion des pièces de rechange peut réduire considérablement le temps d'arrêt en cas de panne et contribuer à prolonger la durée de vie utile du système.

Cela comprend l'identification des pièces qui seront probablement nécessaires et la quantité des pièces de rechange pour être précis et à jour.

Planification des achats de pièces de rechange :

Une fois l'inventaire établi, nous avons planifié les achats futurs de pièces de rechange. Pour chaque type de pièce, nous avons tenu compte de plusieurs facteurs, dont la durée de vie prévue de la pièce, son coût et le délai d'approvisionnement. Par exemple, si une pièce est particulièrement coûteuse ou si elle nécessite un long délai d'approvisionnement, nous pourrions choisir de la stocker en plus grande quantité. Nous avons également tenu compte de l'obsolescence potentielle des pièces. Si une pièce risque de devenir obsolète dans le futur, nous avons prévu de stocker une quantité supplémentaire pour garantir que nous aurons toujours accès à cette pièce lorsque nous en aurons besoin.

Pièce	Composants	Durée de vie (h)	Nombre de pièces nécessaire pour chaque type de maintenance		Remarque pour la gestion de stock
			Préventive	Corrective	
Moteurs pas à pas	Roulements	20000	2	2	Pas moins de 4 pièces
	Ressorts	10000	1	1	Pas moins de 2 pièces
	Bobines	30000	-	2	Pas moins de 2pieces
	Connecteurs	10000	-	5	Pas moins de 3 pièces
Système pneumatique	Joints	20000	2	2	Pas moins de 4 pièces
	Piston vérin	30000	-	2	Pas moins de 1 pièces
	Tige de vérin	30000	-	2	Pas moins de 1 pièces
	Ressorts	10000	-	3	Pas moins de 2 pièces
	Tuyaux	20000	2	3	Pas moins de 4 pièces
	Raccord	10000	5	5	Pas moins de 6 pièces
	Compresseur	30000	-	1	Au moins 1
Ventouse	Soufflets	15000	2	2	Pas moins de 3 pièces
	Joins	10000	2	3	Pas moins de 4 pièces
	Connecteurs	10000	-	5	Pas moins de 3 pièces
Composants électrique	Capteurs	20000	2	2	Pas moins de pièces
	Câbles	10000	1	4	Pas moins de 3 pièces
	Connecteurs	10000	-	5	Pas moins de 3 pièces

TAB. 4.3 : Inventaire des pièces de rechange

5 Conclusion

Au cours de ce chapitre, nous avons discuté en détail du rôle important du contrôle, de l'automatisation et de la maintenance dans le fonctionnement optimal de notre système de tri de médicaments. Grâce à l'implémentation de l'algorithme CNN et de l'API S7-1200, nous avons pu assurer une efficacité et une précision sans faille. La stratégie de maintenance présentée assure non seulement la longévité de notre système, mais aussi sa performance constante, garantissant ainsi une production de médicaments de qualité supérieure.

Chapitre 5

Simulation et Discussion

1 Introduction

Passer de la théorie à la pratique est un pas crucial dans tout projet d'ingénierie. Après avoir établi les bases théoriques et pratiques du système de tri intelligent dans les chapitres précédents, ce chapitre se consacre à la simulation du système en utilisant ROS (Robot Operating System). Ce cadre logiciel flexible et robuste va permettre de tester et de valider le système conçu avant son déploiement réel. La simulation joue un rôle clé, non seulement dans la validation du design, mais aussi dans l'identification et la résolution des défis potentiels. Le voyage à travers ce chapitre sera un mélange d'explorations théoriques et de mises en pratique concrètes, illustrant l'efficacité du système de tri intelligent dans un environnement contrôlé.

2 Simulation de tri par IA

Nous allons voir l'efficacité de l'algorithme d'Intelligence Artificielle, avec un test sur des échantillons qui ne font pas partie des ensembles de test et d'apprentissage (PARALGAN), ainsi que des cas d'échantillons en temps réel avec la camera du PC. L'algorithme va être notamment testé sur d'autres médicaments (LAMIDAZ, GLUCOPHAGE) afin d'analyser sa précision de généralisation.

2.1 Scénario #1 : Échantillon de validation

Le test est fait sur un échantillon de la base de donées.

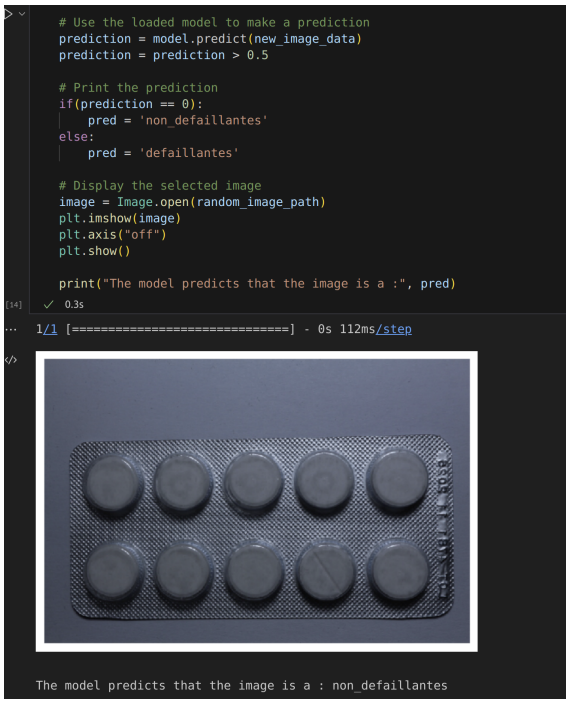


Fig. 5.1 : Une plaquette PARALGAN prédit non défaillante

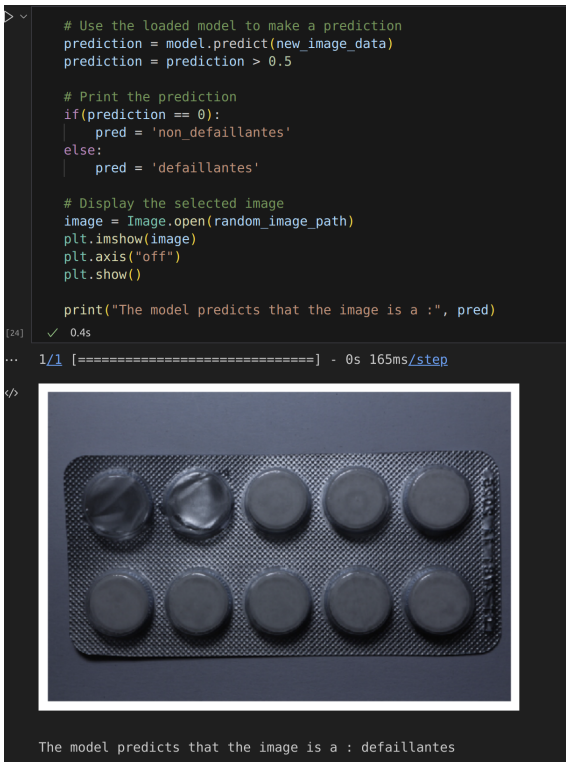


Fig. 5.2 : Une plaquette PARALGANT prédit défaillante

Il montre un résultat satisfaisant, le modèle est capable de prédire correctement dans $0s112ms$ seulement.

2.2 Scénario #2 : Prédiction en temps réel

Le modèle a été testé en temps réel, en utilisant la caméra de l'ordinateur portable.



Fig. 5.3 : Une plaquette PARALGAN prédit non défaillante en temps réel

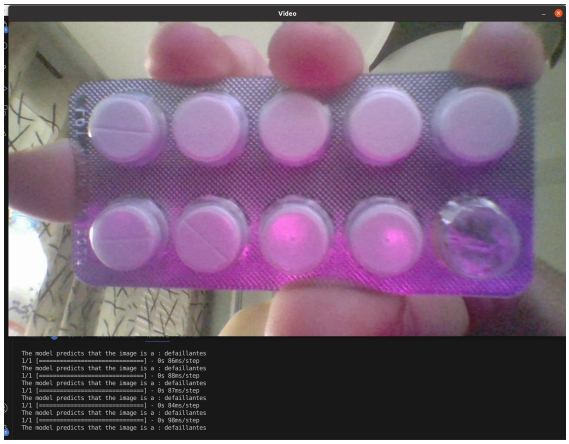


Fig. 5.4 : Une plaquette PARALGAN prédit défaillante en temps réel

Il montre aussi un bon résultat : une correcte prédiction dans $0s98ms$ seulement.

2.3 Scénario #3 : Prédiction en temps réel d'autres médicaments

Un test sur 2 autres médicaments : GLUCOPHAGE LAMIDAZ



Fig. 5.5 : Une plaquette GLUCOPHAGE prédit non défaillante en temps réel



Fig. 5.6 : Une plaquette LAMIDAZ prédit défaillante en temps réel

Fait impressionnant, le modèle a également démontré sa capacité à prédire efficacement les résultats pour 2 différents types de médicaments.

2.4 Discussion :

Le modèle, après avoir été formé sur des échantillons de PARALGAN, a démontré son efficacité et sa précision tant sur les données de validation que sur les données en temps réel. De plus, il a réussi à démontrer sa précision sur des échantillons non entraînés, en temps réel, tels que LAMIDAZ et GLUCOPHAGE, ce qui souligne sa polyvalence et son potentiel pour une large gamme d'applications.

3 Simulation ROS

Cette partie va aborder la simulation de l'algorithme avec le robot SCARA AZ-2000 en utilisant Robot Operating System (ROS), la simulation va se deviser en 2 scenarios :

Le premier : prédiction de plaquette non défaillante, Où le robot ne va pas bouger et la plaquette va passer.

Le deuxième : prédiction de plaquette défaillante et donc le robot bouge (pour la séparer de la chaine de production, autrement dit, le convoyeur)

Mais avant de passer à la simulation, il important d'explorer les différents éléments de l'environnement ROS qui ont permis de tester notre modèle, ainsi que simuler différents scénarios de tests de robustesse et de perturbation.

3.1 L'espace de travail ROS

Le fichier URDF du robot :

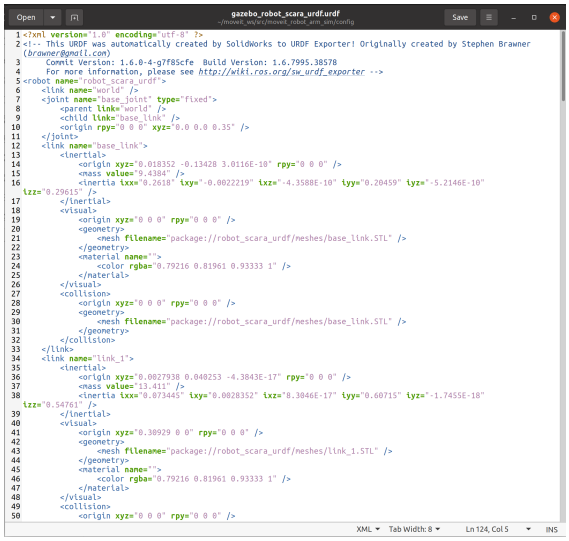


FIG. 5.7 : Fichier URDF du robot

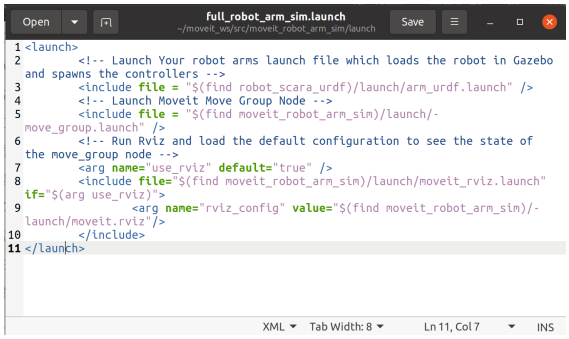


FIG. 5.8 : Fichier de lancement du robot

3.2 Le graph des noeuds

3.2.1 Les noeuds ROS

Nous avons utilisé 3 noeuds : un neuds pour le contrôle automatique du robot, un noeud pour la prédiction et un noeud pour la visualisation par la caméra.

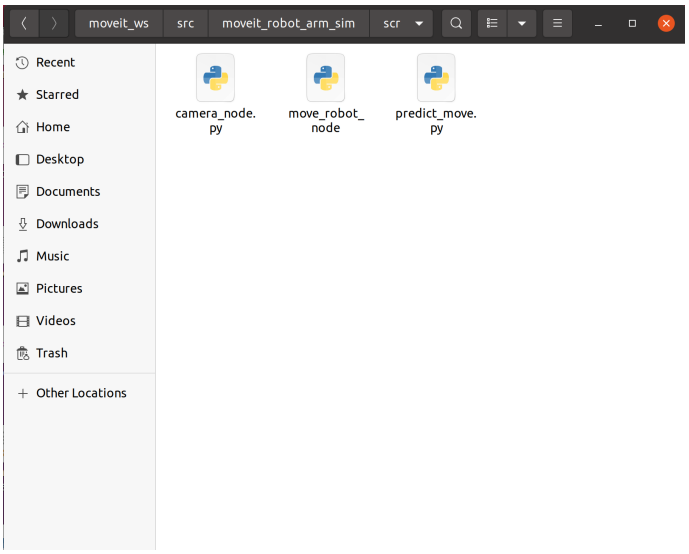


FIG. 5.9 : Les noeuds ROS utilisés

3.2.2 Graph des noeuds et topics

Le graph des noeuds et topics est une représentation visuelle de l'architecture générale logicielle du système robotique. Il montre les différents nœuds, qui sont des entités logicielles autonomes, ainsi que les topics, qui sont les canaux de communication utilisés pour l'échange d'informations entre les nœuds.

Dans ce projet, le graph des nœuds prsenté dans la figure 4.10 ci-aprés, met en évidence les composants clés du système en présentant les nœuds et les topics qui les relient, offrant ainsi un aperçu global de la structure et de la coordination des différentes parties de l'application.

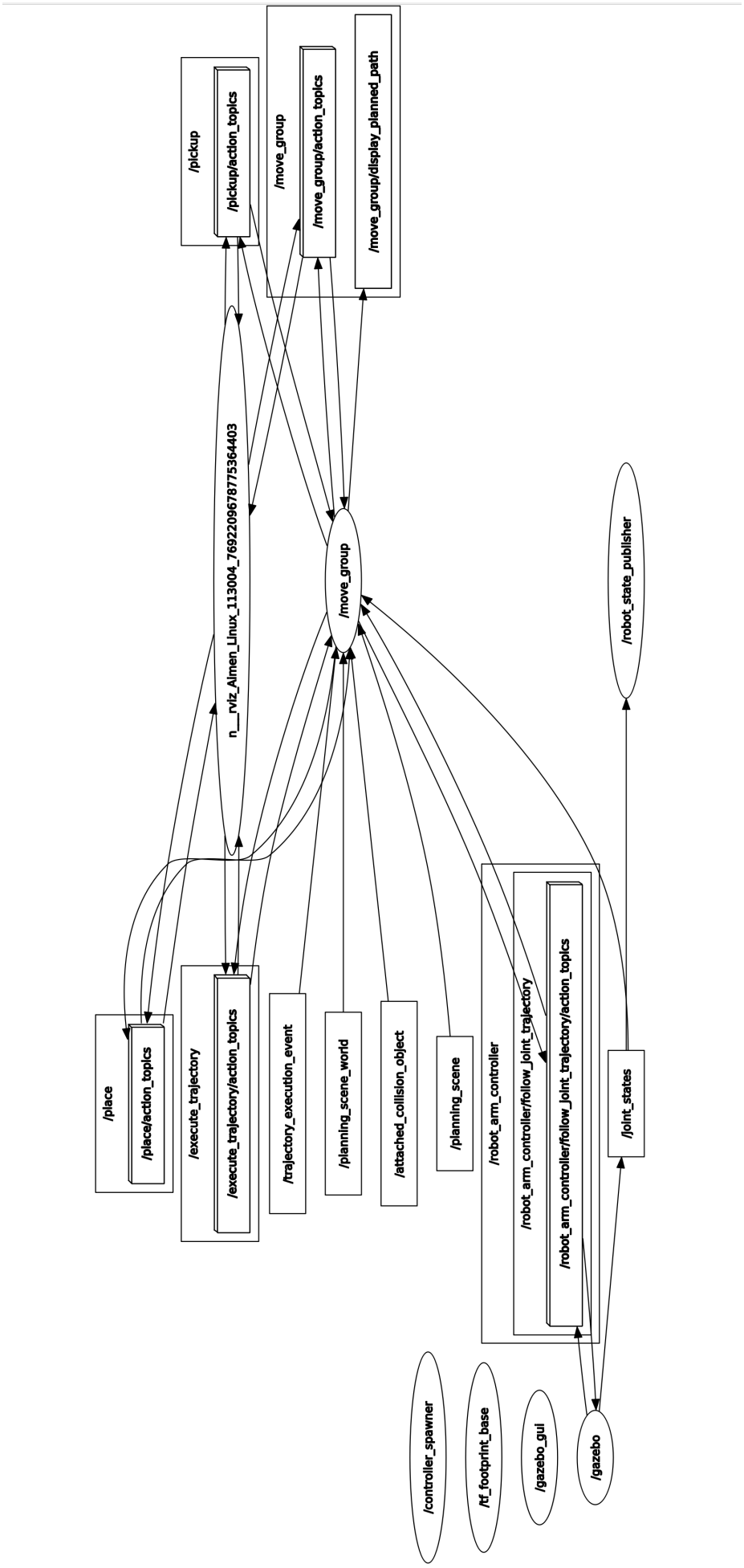


FIG. 5.10 : Graph des noeuds et topics constituans le système.

3.3 La cinématique directe et inverse sur ROS

Dans ROS et MoveIt, les problèmes de la cinématique sont résolus à l'aide de plugins de résolution de cinématique. Par exemple, le KDLKinematicsPlugin que nous avons utiliser utilise la " Kinematics and Dynamics Library (KDL) " pour résoudre les problèmes de cinématique. Ces plugins sont utilisés par MoveIt pour planifier et exécuter des mouvements pour notre robot.

Dans cette simulation nous utilisons la cinématique inverse pour déterminer comment déplacer le robot pour atteindre un certain point dans l'espace, et la cinématique directe pour vérifier où se trouve l'effecteur final de notre robot après avoir effectué ses mouvements planifiés.

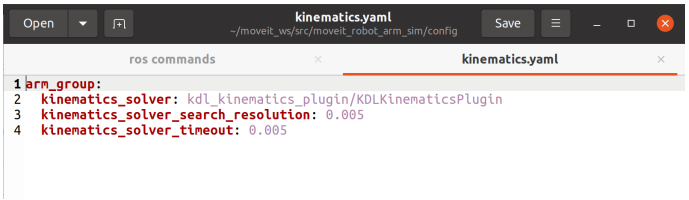


FIG. 5.11 : Le fichier de la cinématique du robot

3.4 Les contrôleurs ROS

Les contrôleurs dans ROS sont des entités qui génèrent les commandes de mouvement pour chaque composant du robot. Ils prennent en compte les informations sensorielles et l'état actuel du robot, puis génèrent une commande qui est envoyée aux actionneurs du robot.

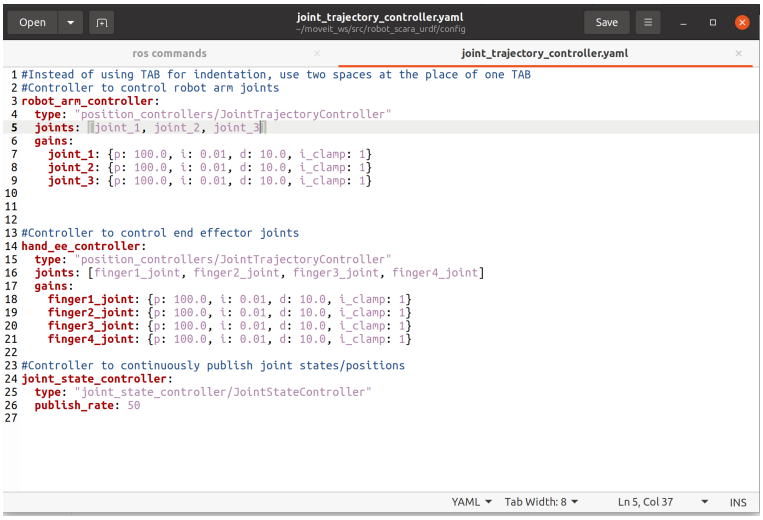


FIG. 5.12 : Les Controleurs utilisés pour le SCARA AZ-2000

- **Contrôleur du bras du robot (robot arm controller) :**

Il contrôle les articulations du bras de notre robot en utilisant un contrôleur de type position controllers/JointTrajectoryController. Ce type de contrôleur est couramment utilisé pour des tâches telles que le suivi de trajectoires spécifiques des articulations.

- **Contrôleur de l'état des articulations (jointnstate controller) :**

Il publie les informations de l'état des articulations du robot à un taux défini (ici, 50Hz).

- **Les contrôleur PID :**

L'utilité d'un contrôleur PID est de régler efficacement et précisément le mouvement des articulations du robot, ainsi que les problèmes de vibration.

3.5 Les scénarios de la simulation

- **Scénario #1 : Robot en mouvement**

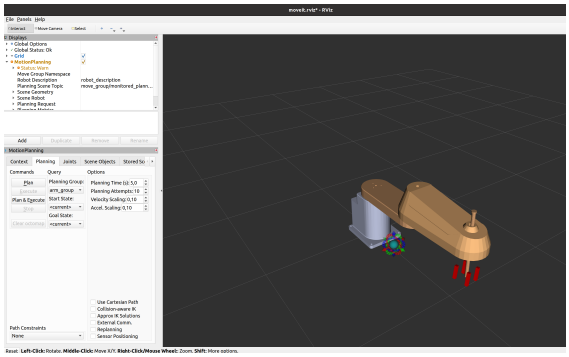


FIG. 5.13 : Visualisation du robot sur RVIZ

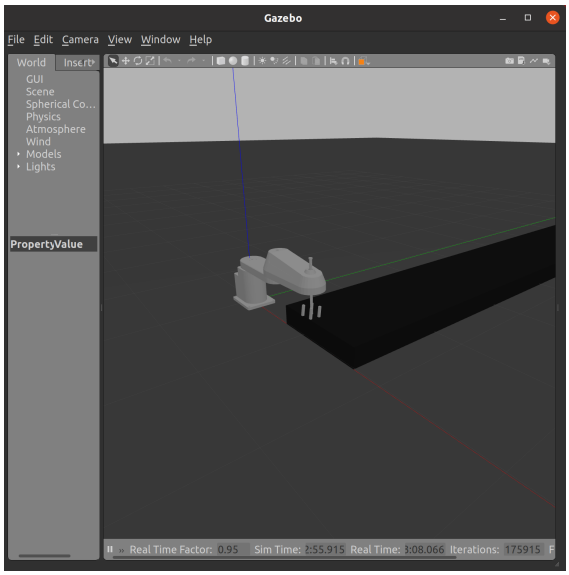


FIG. 5.14 : Visualisation du robot sur GAZEBO

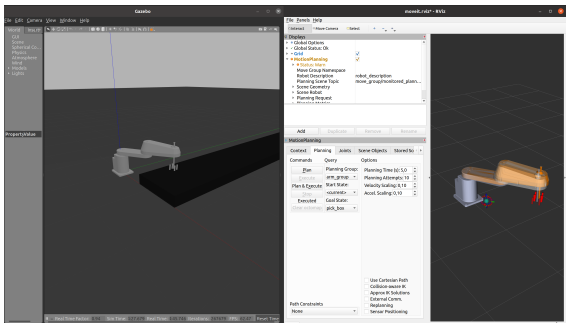


FIG. 5.15 : Robot en mouvement position 1

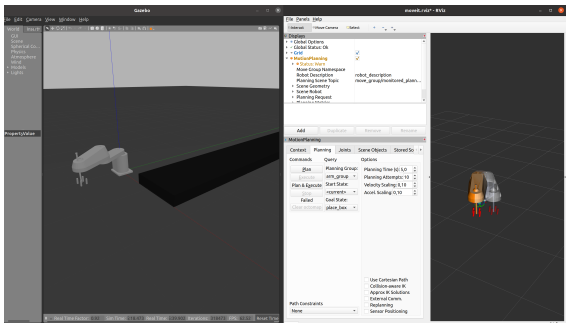


FIG. 5.16 : Robot en mouvement position 2

- **Scénario #2 : Plaquette non défaillante -robot en repos-**

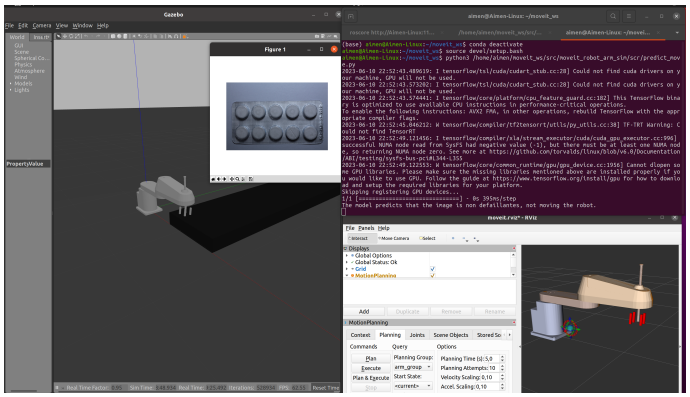


FIG. 5.17 : Plaquette non défaillante -robot en repos-

- **Scénario #3 : Plaquette défaillante -robot en mouvement-**

Plaquette défaillante execusion du plan de mouvement vers la position de la plaquette dans le conveyeur et plan de retour vers le bac de recyclage.

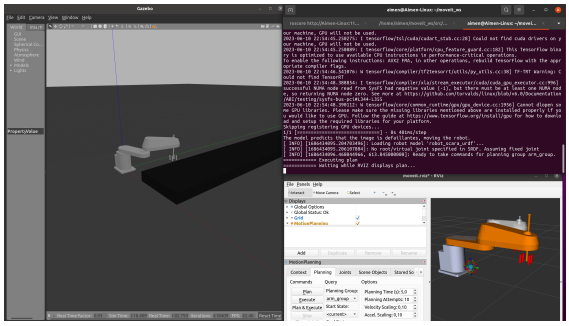


FIG. 5.18 : Plaquette défaillante execusion du plan de mouvement

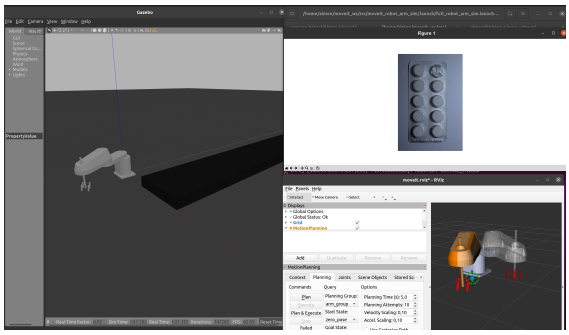


FIG. 5.19 : Retour vers le bac de recyclage

3.6 Tests de robustesse et de perturbation

Les tests de robustesse et de perturbation sont essentiels pour s'assurer que le robot peut fonctionner de manière fiable et sécuritaire dans le monde réel. La robustesse se réfère à la capacité du robot à maintenir sa performance en présence de variations dans les conditions d'entrée et les perturbations. Par exemple, il doit être capable de suivre un plan de mouvement même si le mouvement initial est légèrement différent de ce qui était prévu. De plus, un robot robuste doit être capable de réagir de manière appropriée à des situations imprévues, telles que des obstacles non planifiés, les vibrations et le bruit.

Afin d'évaluer la robustesse de notre robot, nous avons simulé une variété de scénarios d'exploitation.

Les limites cartésiennes



FIG. 5.20 : Les limites cartésiennes du SCARA AZ-2000

Ces paramètres sont utilisés pour contrôler comment le robot se déplace le long d'une trajectoire dans l'espace cartésien. En limitant la vitesse de translation, la vitesse de rotation, l'accélération et la décélération, nous assurons que le robot ne se déplace pas trop rapidement ou n'accélère pas trop rapidement, ce qui pourrait l'endommager.

Nous avons testé le robot à une vitesse de rotation de 1.57 rad/s (comme définit sur le cahier des charges), cela nous a permis de voir si notre robot resiste à une telle vitesse.

3.6.1 Le bruit

Le générateur de bruit (noise generator) ajoute du bruit aux trajectoires générées, ce qui peut permettre à l'algorithme de trouver des trajectoires qui sont robustes face à des perturbations de petite taille.



FIG. 5.21 : Génération de bruit

La génération du bruit a été varié pour pouvoir analyser le comportement de notre robot.

3.6.2 Variation de trajectoires

Nous avons planifier par MOVEIT différentes trajectoires pour le robot, ensuite tester les performances sur ces variations.

```
   the joints that are parents to included links are also included. This me
   link are included in the group-->
11     <!--SUBGROUPS: Groups can also be formed by referencing to already d
12     <group name="arm_group">
13         <joint name="joint_1"/>
14         <joint name="joint_2"/>
15         <joint name="joint_3"/>
16     </group>
17     <group name="hand">
18         <joint name="finger1_joint"/>
19         <joint name="finger2_joint"/>
20         <joint name="finger3_joint"/>
21         <joint name="finger4_joint"/>
22     </group>
23     <!--GROUP STATES: Purpose: Define a named state for a particular gro
   'folded arms'-->
24     <group_state name="zero_pose" group="arm_group">
25         <joint name="joint_1" value="0"/>
26         <joint name="joint_2" value="0"/>
27         <joint name="joint_3" value="0.0765"/>
28     </group_state>
29     <group_state name="pick_pose" group="arm_group">
30         <joint name="joint_1" value="1.2367"/>
31         <joint name="joint_2" value="0"/>
32         <joint name="joint_3" value="-0.0563"/>
33     </group_state>
34     <group_state name="place_pose" group="arm_group">
35         <joint name="joint_1" value="0.0334"/>
36         <joint name="joint_2" value="0"/>
37         <joint name="joint_3" value="0.0599"/>
38     </group_state>
39     <group_state name="pick_pose_2" group="arm_group">
40         <joint name="joint_1" value="0.9"/>
41         <joint name="joint_2" value="0"/>
42         <joint name="joint_3" value="0"/>
43     </group_state>
44     <group_state name="pick_box" group="arm_group">
45         <joint name="joint_1" value="0.5867"/>
46         <joint name="joint_2" value="0"/>
47         <joint name="joint_3" value="-0.0295"/>
48     </group_state>
49     <group_state name="place_box" group="arm_group">
50         <joint name="joint_1" value="-1.0166"/>
51         <joint name="joint_2" value="0"/>
52         <joint name="joint_3" value="-0.0295"/>
53     </group_state>
54     <group_state name="take_box" group="hand">
55         <joint name="finger1_joint" value="0.021"/>
56         <joint name="finger2_joint" value="-0.021"/>
57         <joint name="finger3_joint" value="0.021"/>
58         <joint name="finger4_joint" value="-0.021"/>
59     </group_state>
```

FIG. 5.22 : Les différentes trajectoires de test

4 Résultats et Comparaison

Une analyse comparative détaillée est faite entre le système de tri automatisé Harlequin et notre système de tri par algorithme d'intelligence artificielle (IA). Pour une comparaison juste et équilibrée, nous examinerons plusieurs critères, notamment la précision, le temps de tri, le temps d'exécution de la tâche d'élimination et la consommation d'énergie.

Système / Métriques	Sytème Harlequin	Système de tri intelligent
Précision	$\approx 93\%$	$\approx 96\%$
Temps de tri	≈ 60 à $70ms$	≈ 80 à $100ms$
Temps d'élimination	$\approx 3.5s$	$\approx 2s$
Consommation d'énergie	$\approx 3kW$	$\approx 2.4kW$

4.1 Discussion des résultats

Alors sur la base de :

- **La précision** : Le système de tri intelligent a montré une précision de 96%, supérieure à celle du système Harlequin de 93%.
- **Temps de tri** : Le temps de tri a révélé que le système Harlequin a une meilleure performance avec un temps de traitement entre 60 et 70 ms, contre 100 à 180 ms pour système intelligent lors des tests de validation des données. Cependant, pour le traitement de données en temps réel, notre système a montré un temps de réponse compétitif de 80 à 100 ms.
- **Temps d'exécution** : Le SCARA AZ-2000 utilisé a été conçu pour la réaliser en 2 secondes, ce qui est $42,8\%$ plus rapide que le robot à 1 degré de liberté utilisé par le système Harlequin, qui nécessite $3,5$ secondes.
- **Consommation d'énergie** : par une cosommation de 2.4 kW par jour, notre système utilise environ 20% moins d'énergie que le système Harlequin qui consomme 3 kWh pour Harlequin. Ce facteur est bénéfique tant pour le retour sur investissement que pour la responsabilité environnementale.

4.2 Avantages et Inconvénients

4.2.1 Système de tri Harlequin

- Avantages :

- **Efficacité** : Les systèmes automatisés peuvent fonctionner en continu sans se fatiguer ni ralentir, ce qui peut augmenter la productivité.
- **Réduction des erreurs** : Les erreurs de tri dues à des facteurs humains sont considérablement réduites.
- **Standardisation** : Les systèmes automatisés offrent une standardisation dans le processus de tri, garantissant une qualité constante.
- **Fiabilité** : En raison de sa longue existence sur le marché, le système Harlequin a démontré une grande fiabilité.
- **Temps de tri rapide** : Il excelle dans le tri rapide des médicaments, ce qui peut être particulièrement avantageux dans les situations où le temps est essentiel.

- Inconvénients :

- **Moins de flexibilité** : Les systèmes automatisés ont une capacité limitée à s'adapter aux changements et nécessitent souvent une reprogrammation ou une révision matérielle pour gérer de nouvelles tâches ou des variations dans les tâches existantes.
- **Coût d'investissement initial élevé** : L'achat, l'installation et la maintenance d'un système automatisé peuvent nécessiter un investissement initial significatif.

- **Précision** : Bien que respectable, la précision de 93% est inférieure à celle de notre système de tri par algorithme d'IA.
- **Consommation d'énergie** : Avec une consommation de 3 kWh par jour, il est plus énergivore que notre système.
- **Temps d'exécution de la tâche d'élimination** : Plus lent lors de l'élimination des plaquettes de médicaments défaillantes.

4.2.2 Système inteligent SCARA AZ-2000

- Avantages :

- **Précision accrue** : Les algorithmes d'IA, lorsqu'ils sont bien entraînés, peuvent surpasser les systèmes traditionnels en termes de précision.
- **Flexibilité** : Les algorithmes d'IA sont capables d'apprendre et de s'adapter aux changements. Ils peuvent gérer de nouvelles tâches sans nécessiter une reprogrammation ou une révision matérielle.
- **Amélioration continue** : L'IA a la capacité d'apprendre de ses erreurs et de s'améliorer au fil du temps grâce à l'apprentissage continu.
- **Économie d'énergie** : En consommant seulement 2,4 kWh par jour, notre système est moins énergivore que le système Harlequin.
- **Rapidité de la tâche d'élimination** : système effectue la tâche d'élimination plus rapidement que Harlequin.

- Inconvénients :

- **Nécessite des données de qualité pour l'entraînement** : La performance de l'IA dépend fortement de la qualité des données utilisées pour l'entraînement. De mauvaises données peuvent entraîner de mauvaises performances.
- **Complexité et compétences requises** : Le développement, l'entraînement et la maintenance d'un algorithme d'IA nécessitent des compétences spécialisées et peuvent être complexes.
- **Temps de tri** : Pour les tests de validation des données, notre système a besoin de plus de temps pour trier les médicaments par rapport au système Harlequin.
- **Nouveau sur le marché** : Le système étant nouveau sur le marché, sa fiabilité à long terme reste à établir.

Conclusion générale

En conclusion, ce projet a mis en évidence la valeur incontestable de l'intégration de l'intelligence artificielle, y compris le module NPU pour une accélération dédiée des tâches de l'IA, et de la robotique dans l'industrie manufacturière. Cela a été clairement illustré dans le contexte de l'entreprise pharmaceutique algérienne SAIDAL, où le but principal a été de concevoir un système de tri intelligent utilisant l'apprentissage approfondi et le Robot Operating System (ROS) pour une ligne de production automatisée, exploitant le potentiel du robot SCARA AZ-2000 dans une application industrielle réelle.

Ce projet a non seulement illustré le potentiel énorme de ces technologies pour améliorer l'efficacité et la précision des systèmes de production, mais a également aidé à éliminer les réticences préexistantes chez les ingénieurs algériens concernant l'intégration de l'intelligence artificielle et de la robotique dans les systèmes industriels. La réalisation réussie de ce système a marqué un pas significatif dans l'intégration harmonieuse de l'IA, avec l'aide du module NPU, la robotique et les automates programmables industriels (API), améliorant ainsi les performances de l'API grâce à l'intégration de l'IA. Cette valeur ajoutée s'étend au-delà de la conception de systèmes de tri, couvrant également la maintenance et la gestion des installations industrielles.

La familiarisation et l'utilisation de ROS, reconnu mondialement comme un outil puissant pour la robotique, ont été des aspects essentiels de ce projet. Son utilisation a non seulement renforcé les compétences techniques du projet, mais a également renforcé sa crédibilité et sa pertinence dans le domaine de la robotique moderne.

Le système de tri intelligent conçu a brillé par une précision exceptionnelle de 96% et une consommation d'énergie optimisée. Toutefois, une analyse constructive met en évidence certaines limitations, telles que la capacité du système à traiter de grands volumes de produits rapidement et à s'adapter à des changements dans l'environnement de production.

En regardant vers l'avenir, il serait opportun de se concentrer sur l'amélioration de la capacité de traitement du système, sa résilience face aux coupures de courant, ainsi que la simplification de la maintenance et de la mise à jour de l'IA. Par ce projet, un pas important a été franchi vers l'adoption de l'industrie 4.0 en Algérie, ouvrant ainsi la porte à une transformation technologique de l'industrie manufacturière algérienne.

Le système intelligent de tri répond à plusieurs caractéristiques clés de l'Industrie 4.0. C'un exemple d'automatisation, tandis que l'utilisation de l'IA pour détecter les défauts et classer les médicaments.

Ce projet illustre également certains principes de l'Industrie 5.0. Le système de bras robotisé est conçu pour travailler en collaboration avec des humains, et cela correspond à l'aspect

de collaboration homme-machine. De plus,il vise à augmenter l'efficacité du processus de production de médicaments et à minimiser les déchets (par le système de recyclage des déchets, un futur projet avec l'ajout d'une ligne de recyclage),cela correspond à l'aspect de durabilité de l'Industrie 5.0.

Ce projet englobe des éléments de l'Industrie 4.0 et de l'Industrie 5.0, ce qui reflète l'évolution continue de la technologie industrielle, où les concepts de l'Industrie 4.0 et 5.0 se chevauchent et coexistent.

Pour finir, ce travail a démontré la possibilité et les bénéfices de combiner les technologies traditionnelles avec l'IA et la robotique pour améliorer l'efficacité et la précision dans un environnement de production réel. Il est donc impératif de poursuivre la recherche et l'innovation dans ce domaine pour assurer une optimisation continue des systèmes de production.

Références

[1] M. Hermann, T. Pentek et B. Otto. "Design Principles for Industrie 4.0 Scenarios". In : *49th Hawaii International Conference on System Sciences (HICSS)*. 2016.

[2] M. Rüssmann et al. "Industry 4.0: The Future of Productivity and Growth in Manufacturing Industries". In : *Boston Consulting Group* (2015).

[3] R. Roman, P. Najera et J. Lopez. "Securing the Internet of Things". In : *Computer* 44.9 (2011), p. 51-58.

[4] A. Stocker et S. Shaheen. "Industry 5.0 - A Human Centric Solution". In : *Proceedings of the International Conference on Business Excellence* (2019).

[5] Xun Xu et al. "Industry 4.0 and Industry 5.0—Inception, conception and perception". In : *Journal of Manufacturing Systems* 61 (2021), p. 530-535. issn : 0278-6125. doi : 10.1016/j.jmsy.2021.10.006.

[6] L. Benhamou, V. Giard et P. Fénies. "Un outil de conception et de production intelligent permettant la personnalisation d'une production continue de masse". In : *Revue Française de Gestion Industrielle* 36.1 (2022), p. 07-26.

[7] N. Jazdi. "Cyber Physical Systems in the Context of Industry 4.0". In : *IEEE 2014 International Conference on Automation, Quality and Testing, Robotics*. 2014.

[8] R. Rosen. "Towards Industry 5.0: Empowering Humans in Industry and Manufacturing Systems". In : *IFAC Journal of Systems and Control* (2020).

[9] Stamatios Manesis et George Nikolakopoulos. *Introduction to Industrial Automation*. CRC Press, Taylor Francis Group, 2018.

[10] W. Bolton. *Programmable Logic Controllers*. Newnes, Elsevier, 2009.

[11] L. A. Bryan et E. A. Bryan. *Programmable Controllers*. Second Edition. Industrial Text Company, 1988.

[12] J. Goodman. "Pharmaceutical industry". In : *Medicine in the Twentieth Century*. Taylor & Francis, 2020, p. 141-154.

[13] J. Karamehic et al. "Financial aspects and the future of the pharmaceutical industry in the United States of America". In : *Materia socio-medica* 25.4 (2013), p. 286.

[14] T. Alsamara, G. Farouk et A. Adel. "Public health and the legal regulation of the pharmaceutical industry in Algeria". In : *The Pan African Medical Journal* 41 (2022).

[15] *Marchesini Group Linea blister MB451 + MA305 Blister machine*. Manuel d'utilisation, non publié, confidentiel. Marchesini Group, Italy. 2020.

[16] L. Peltonen. "Practical guidelines for the characterization and quality control of pure drug nanoparticles and nano-cocrystals in the pharmaceutical industry". In : *Advanced drug delivery reviews* 131 (2018), p. 101-115.

[17] *HARLEQUIN hq manual control*. Manuel d'utilisation, non publié, confidentiel. Halequin HQ, Marchesini Group. 2019.

[18] C. Mavroidis, V. N. Dubey et M. L. Yarmush. "Robotics in the pharmaceutical industry". In : *Biomechatronics*. Sous la dir. de C. Mavroidis et V. N. Dubey. Springer, 2011.

[19] R. Macarron, M. N. Banks, D. Bojanic et al. "Impact of high-throughput screening in biomedical research". In : *Nature Reviews Drug Discovery* 10.3 (2011), p. 188-195.

[20] European Medicines Agency. *Guidelines of 21 September 2020 on Good Manufacturing Practice*. 2020.

[21] FDA. *Process Validation : General Principles and Practices*. U.S. Food and Drug Administration. 2011.

[22] Shimon Y. Nof. *Handbook of Industrial Robotics*. Wiley, 1999.

[23] Iaia Vincenzo. "To Be, or Not to Be … Original Under Copyright Law, That Is (One of) the Main Questions Concerning AI-Produced Works". In : *GRUR International* 71.9 (août 2022), p. 793-812.

[24] Spyros Makridakis. "The forthcoming Artificial Intelligence (AI) revolution : Its impact on society and firms". In : *Futures* 90 (2017), p. 46-60.

[25] Vincenzo Falbo et al. "Analyzing machine learning on mainstream microcontrollers". In : *Applications in Electronics Pervading Industry, Environment and Society : APPLEPIES 2019 7*. Springer. 2020, p. 103-108.

[26] Vivienne Sze et al. "Efficient processing of deep neural networks : A tutorial and survey". In : *Proceedings of the IEEE* 105.12 (2017), p. 2295-2329.

[27] PhData. *Les différentes niveaux de l'intelligence artificielle*. [Capture d'écran]. Consulté le 15/03/2023. s.d. URL : https://www.phdata.io/blog/data-science-terms-you-should-know-the-difference-between-ai-ml-and-dl.

[28] Andrew G Howard et al. "Mobilenets : Efficient convolutional neural networks for mobile vision applications". In : *arXiv preprint arXiv :1704.04861* (2017).

[29] Keiron O'Shea et Ryan Nash. "An Introduction to Convolutional Neural Networks". In : (2015).

[30] Y. Hagiwara J.H. Tan M. Adam A. Gertych T.R. San U.R. Acharya S.L. Oh. "A deep convolutional neural network model to classify heartbeats". In : (2017). DOI : 10.1016/j.compbiomed.2017.08.022.

[31] Abhishek Samanta Arohan Ajit Koustav Acharya. "A Review of Convolutional Neural Networks". In : (2020).

[32] Tianmei Guo et al. "Simple Convolutional Neural Network on Image Classification". In : *Department of Computer Science and Technology Shandong Provincial Key Laboratory of Network based Intelligent Computing University of Jinan* (2021).

[33] Morgan Quigley et al. "ROS : an open-source Robot Operating System". In : *ICRA workshop on open source software*. T. 3. 3.2. Kobe, Japan. 2009, p. 5.

[34] ROS-Industrial Consortium. *ROS-Industrial*. https://rosindustrial.org/.

[35] Lenka PITONAKOVA et al. “Feature and performance comparison of the V-REP, Gazebo and ARGoS robot simulators”. In : *Annual Conference Towards Autonomous Robotic Systems*. Springer. 2018, p. 357-368.

[36] Nathan KOENIG et Andrew HOWARD. “Design and use paradigms for gazebo, an open-source multi-robot simulator”. In : *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*. T. 3. IEEE. 2004, p. 2149-2154.

[37] Carol FAIRCHILD et Thomas L HARMAN. *ROS Robotics By Example : Learning to control wheeled, limbed, and flying robots using ROS Kinetic Kame*. Packt Publishing Ltd, 2017.

[38] A. Wahab Mawadah Jlaa Yahyh IMAN SALIH KAREM Talal A. Jabbar. “Design and Implementation for 3-DoF SCARA Robot based PLC”. In : (2017). DOI : 10.22153/kej.2017.01.002.

[39] Brahim FERNINI. “La modélisation cinématique et dynamique du robot SCARA en utilisant SOLIDWORKS et vérification par MATLAB/SIMULINK”. In : (Septembre 2011).

[40] Yaru XU et Rong LIU. “Dynamic modeling of SCARA robot based on Udwadia-Kalaba theory”. In : Vol. 9(10) (2017), p. 1-12. DOI : 10.1177/1687814017728450.

[41] Monika RYBCZAK, Natalia POPOWNIAK et Krystian KOZAKIEWICZ. “Applied AI with PLC and IRB1200”. In : *Applied Sciences* 12.24 (2022), p. 12918.

[42] SIEMENS. *SIMATIC S7-1500 TM NPU : Operating Manual*. 2022.

[43] Anupa KELKAR et Chris DICK. “A GPU Hyperconverged Platform for 5G vRAN and Multi-Access Edge Computing”. In : *2021 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*. IEEE. 2021, p. 1-6.

[44] Yuehui Zhao Peng Wang Jun Zhang TIANYI LIU Shuangsang Fang. “Implementation of Training Convolutional Neural Networks”. In : (2015). DOI : 10.48550/arXiv.1506.01195.

[45] Srinivas TAMMINA. “Transfer learning using VGG-16 with deep convolutional neural network for classifying images”. In : *International Journal of Scientific and Research Publications (IJSRP)* 9.10 (2019), p. 143-150.

[46] Suresh ANNAMALAI, R UDENDHRAN et S VIMAL. “Cloud-based predictive maintenance and machine monitoring for intelligent manufacturing for automobile industry”. In : *Novel practices and trends in grid and cloud computing*. IGI Global, 2019, p. 74-89.

[47] “A novel automata and neural network based fault diagnosis system for PLC controlled manufacturing systems”. In : *Computers Industrial Engineering* 139 (2020), p. 106188. ISSN : 0360-8352. DOI : https://doi.org/10.1016/j.cie.2019.106188.