

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

وزارة التعليم العالي والبحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة سعد دحلب البليدة
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا
Faculté de Technologie

قسم الآليات والإلكتروني
Département D'Automatique et Electrotechnique



Mémoire de Master

Filière Automatique
Spécialité Automatique et Informatique Industrielle (AII)

Présenté par

TAHI Samy

&

LAROUCI Billel

Etude et conception d'un module de production Cyber-Physique Cas d'étude d'un robot cartésien

Proposé par: Mr. Abdelhamid BENDJELLOUL

Co-Promoteur: Pr. Boualem KAZED

Année Universitaire : 2022/2023

ملخص:

في هذه الأطروحة، ركزنا على دراسة وتصميم وحدة الإنتاج السيبراني المادي وفقا لمتطلبات الصناعة 4.0. من خلال دراسة الحالة الخاصة بنا لروبوت ديكارتي، قمنا بتصميم الجزء السيبراني بما في ذلك الإدارة والتحكم ومحاكاة الروبوت. يتم التحقق من الصحة عن طريق تقنية التكلفة الافتراضي.

الكلمات المفتاحية: الصناعة 4.0، السيبراني المادي، التكلفة الافتراضي

RESUME

Dans ce mémoire, on a s'intéresser à l'étude et conception d'un module de production cyber-physique conforme aux exigences de l'industrie 4.0. Avec notre cas d'étude d'un robot cartésien on a concevré la partie cybernétique incluant l'administration, la commande et la simulation du robot. La validation est faite par la technique du commissioning virtuel.

Mot clés : Industrie 4.0, Cyber-Physique, Commissioning Virtuel.

Summary:

In this thesis, we focused on the study and design of a cyber-physical production module in accordance with the requirements of Industry 4.0. With our case study of a Cartesian robot we designed the cybernetic part including the administration, control and simulation of the robot. The validation is done by the technique of virtual commissioning.

Key words : Industry 4.0 , Cyber-Physical, Virtual Commissioning.

Remerciements

Nous remercions avant tout, ALLAH tout puissant qui nous a donné la volonté et le courage pour accomplir ce travail.

Nous remercions Mr. Abdelhamid Bendjelloul, pour nous proposer ce thème et pour ses judicieux conseils, sa rigueur et sa persévérance et surtout pour son esprit paternel tout au long de notre cursus de formation.

Nos remerciements les plus vifs s'adressent aussi aux messieurs le président et les membres de jury d'avoir accepté d'examiner et d'évaluer notre travail.

Nous tenons à exprimer nos plus sincères remerciements et gratitude à notre promoteur Mr. Abdelhamid Bendjelloul, pour sa disponibilité, pour ses précieux conseils et ses multiples interventions constructives.

Ainsi que notre co-promoteur Pr. Boualem Kazed pour son temps, sa générosité et sa contribution sur le plan académique.

On remercie en particulier Dr. Slimane Tahi pour son temps, son soutien et ses conseils concernant la rédaction de notre mémoire.

Nous remercions aussi toutes personnes ayant contribué de près ou de loin à l'aboutissement de ce travail, qu'ils trouvent ici le témoignage de notre profonde reconnaissance pour l'aide précieuse qu'ils nous ont apportée.

Dédicaces

Je dédie humblement ce travail à mes chers parents, dont le soutien et les encouragements ont été inestimables tout au long de mon parcours académique. Je leur suis profondément reconnaissant pour tout ce qu'ils ont fait pour me permettre d'atteindre cette étape.

À ma mère, qui a été un pilier essentiel dans mon parcours éducatif, je souhaite exprimer tout mon amour et toute mon affection. Sans elle, ma réussite n'aurait pas été possible. Que cette dédicace soit le témoignage de l'amour et de la gratitude que je lui porte.

À mon père, Dr Tahi Slimane, dont l'aide précieuse a été d'une importance capitale dans l'élaboration de ce mémoire, je veux lui confirmer mon attachement sincère et mon profond respect. Sa disponibilité et son soutien sont des traits pour lesquels je serai toujours reconnaissant toute ma vie.

À mes frères jumeaux, Wassim et Aïmen et à mon petit frère Imad qui me sont très chers.

À mes proches amis, Malek, Aymen, Zaki, Rafik et Marwa, qui ont été là pour moi et que j'aime sincèrement.

À toute ma famille qui m'a encouragé,

À mon Binôme Billel et la famille Larouci,

Et tous mes amis de ma promotion.

TAKI Samy

Dédicaces

Ce n'est que grâce à l'aide d'ALLAH que j'ai finalisé cet humble Travail Que je dédie à toutes les personnes qui me sont chères et à qui je suis Enormément reconnaissant de tout le soutien qu'elles m'ont apportées Durant les moments difficiles.

À mes très chers parents, sans qui je n'aurais jamais vu la lumière Du jour, ni devenu ce que je suis. Puissent-ils trouver en ce modeste Travail Toute la gratitude et la reconnaissance d'un fils dévoué, Ma mère pour son Dévouement et ses sacrifices et Mon père pour Son écoute et ses précieux conseils.

À ma très chère femme, Ton encouragement et ton soutien étaient la bouffée d'oxygène qui me ressourçait dans les moments pénibles, de solitude et de souffrance. Merci d'être toujours à mes côtés, et deux petits bébés Allaa Eddine et Ilin Kouthar.

À mes frères ET mes amis

À mon Binôme TAHI Samy à toute sa famille.

À tous mes collègues et amis de ma promotion un par un.

À tous ceux qui m'aiment et qui me connaissent

Tous ceux pour qui je compte.

LAROUCI Billel

ACRONYMES UTILISES

- AAS** : Asset Administration Shells.
- AASX** : Format de fichier de package pour l'AAS.
- API** : Interface de programmation d'application.
- ATIL** : Analyse et Traitement Informatique de la Langue Française.
- CAO** : Conception Assisté par Ordinateur.
- CC** : Cloud Computing.
- CDTA** : Centre de Développement des Technologie Avancées.
- CPS** : Système Cyber-Physique.
- CPPS** : Système de Production Cyber-Physique.
- CPPM** : Module de Production Cyber-Physique.
- CNC** : Computer Numerical Control.
- DT** : Digital Twin.
- FBD** : Function Block Diagramme.
- HIL** : Hardware In the Loop.
- I4.0** : Industry 4.0.
- IA** : Intelligence Arttrificielle.
- IDTA** : Industrial Digital Twin Association
- IHM** : Interface Human-Machine.
- IL** : Liste d'Instruction.
- IoT** : Internet of Things.
- IIoT** : Industriel Internet of Things.
- MQTT** : Message Queuing Telemetry Transport.
- OPC-UA** : Open Protocol Communication – Unified Architecture.
- PLC** : Programmable Logic Controller.
- RAMI** : Reference Architectural Model Industry.
- RFID** : Radio Frequency Identification.
- ROS** : Robot Operating System.
- SCARA** : Selective Complaint Articulated Robot for Assembly.
- SIL** : Software In the Loop.
- SM** : Sous-Modèle.
- ST** : Structured Text.
- VC** : Commissioning Virtuel.
- VR** : Réalité Virtuelle.

SOMMAIRE

ملخص:.....	2
INTRODUCTION GENERALE.....	1
I. Chapitre I : Généralités et Définitions.....	4
I.1) Introduction.....	4
Industrie 4.0	4
I.1.a) Concept de développement pour l'industrie 4.0	5
I.1.b) Reference Architectural Model Industry 4. 0 (RAMI 4.0) [57]	12
I.1.c) Asset Administration Shell.....	16
I.1.d) Les jumeaux numériques (Digital Twin)	17
I.2) Conclusion.....	26
II. Chapitre II : L'état de l'art.....	28
II.1) Introduction	28
II.2) Systèmes Cyber-Physiques de Production	28
II.2.a) Introduction	28
II.2.b) Les Systèmes Cyber-Physiques.....	28
II.2.c) Les Systèmes de Production Cyber-Physiques	32
II.3) Le Commissioning Virtuel	39
II.3.a) Introduction	39
II.3.b) Définition du Commissioning Virtuel.....	40
II.4) Conclusion	44
III. Chapitre III : Etude et Développement du Module de production Cyber-physique.....	46
III.1) Introduction	46
III.2) Cas d'étude : Robot Cartésien à 3 axes	46
III.2.a) Historique :	47
III.2.b) Caractéristiques d'un robot.....	48
III.2.c) Définition d'un robot industriel.....	49
III.3) L'automatisme.....	56
III.3.a) Automate Programmable Industriel.....	56
III.4) Module de production cyber-physique	57
III.4.a) Généralités.....	57
III.4.b) Les exigences des systèmes cyber-physique de production.....	57
III.4.c) Architecture de principe du CPPM.....	58
III.4.d) Implémentation :	60

III.4.e) Développement du AAS.....	64
III.5) Conclusion	67
IV. Chapitre IV : Conception d'un module de Production Cyber-Physique : Cas d'étude d'un Robot Cartésien	69
IV.1) Introduction.....	69
IV.2) Partie I.....	69
IV.2.a) Le site 3Dfindit.com propulsé par CADENAS.....	70
IV.2.b) SolidWorks.....	71
IV.2.c) SolidWorks to URDF Exporter.....	72
IV.2.d) CoppeliaSim (Version Edu).....	75
IV.2.e) Utilisation d'articulation pour démarrer le robot	75
IV.2.f) Langage Lua.....	77
IV.2.g) Python.....	78
IV.2.h) RemoteAPI de CoppeiaSim.....	78
IV.2.i) Effecteur Robot.....	81
IV.2.j) La Commande du Robot par un Automate Programmable Industriel.....	82
IV.2.k) PLC Siemens S7 ET200SP - 1512SP F-1 PN.....	82
IV.2.l) Description de la CPU.....	82
IV.2.m) TIA Portal	83
IV.2.n) Configuration du PLC.....	84
IV.2.o) Communication entre PLC et Contrôleur.....	84
IV.2.p) Protocole OPC-UA	84
IV.2.q) Communication du PLC avec l'OPC-UA.....	85
IV.2.r) UA Expert.....	86
IV.2.s) Simulation du robot.....	87
IV.3) Partie II.....	90
IV.3.a) Eclipse Basyx.....	90
IV.3.b) Admin_shell_io by IDTA	91
IV.3.c) Création du fichier AAS.....	91
IV.3.d) AAS Package Explorer.....	91
IV.3.e) Docker.....	93
IV.3.f) Serveur AAS	94
IV.3.g) Registre AAS.....	95
IV.3.h) DataBridge	95
IV.4) Conclusion	97
V. CONCLUSION GENERALE	99

BIBLIOGRAPHIE ET RÉFÉRENCES.....	100
Annexe 1	108
Annexe 2	119
Annexe 3	122
Annexe 4	124
Liste IO.....	125

LISTE DES FIGURES

Figure I.1 - L'évolution de l'industrie.....	5
Figure I.2 - Piliers de la quatrième révolution industrielle (Source : BCG - Institute_Aethon.com)	6
Figure I.3 - L'architecture de RAMI 4.0.....	13
Figure I.4 - Le flux de valeur du cycle de vie.....	14
Figure I.5 - La inter connectivité de la nouvelle industrie.....	14
Figure I.6 - Les sous-couches de la couche d'architecture.....	15
Figure I.7 - Composant de fabrication intelligente - composant I4.0.....	16
Figure I.8 - Le schéma de DT (Cimino et al., 2019).....	19
Figure I.9 - Architecture de jumeau numérique en 5 couches.....	24
Figure II.1 - Domaine d'utilisation du CPS [22].....	29
Figure II.2 - Domaine d'utilisation du CPS.....	29
Figure II.3 - Différents composants d'un CPS.....	31
Figure II.4 - Modèle générique de CPPS proposé [54].....	32
Figure II.5 - Décomposition de la hiérarchie de l'automatisation avec services distribués [30].....	35
Figure II.6 - Interaction entre humains et machines dans les systèmes cyber-physiques [31].....	36
Figure II.7 - L'Ingénierie avec et sans VC (Lechler et al., 2019).....	40
Figure II.8 - Les différents types du VC (SIL, HIL)[55].....	41
Figure III.1 - Les robots SCARA.....	50
Figure III.2 - Structure du manipulateur de coude.....	51
Figure III.3 - Espace de travail du manipulateur de coude.....	51
Figure III.4 - Le robot manipulateur parallèle.....	52
Figure III.5 - Les robots cartésiens.....	53
Figure III.6 - Yamaha SXYx-C-A1-15-15-ZFL20-15.....	55
L'architecture 5C pour la mise en œuvre du CPPS[71].....	59
Figure III.7 - L'architecture du principe pour la mise en œuvre du CPPM.....	59
Figure III.8 - Le rôle du Shell d'administration d'actifs dans la numérisation [72].....	60
Figure III.9 - Diagramme décrivant l'architecture Organique.....	61
Figure III.10 - Diagramme décrit l'architecture Fonctionnelle de la solution.....	62
Figure III.11 - Diagramme décrivant le AAS du Robot.....	65
Figure IV.1 - 3Dfindit : la recherche et téléchargement du model 3D du robot.....	70
Figure IV.2 - 3Dfindit : Téléchargement des modèles 3D sous format SolidWorks.....	71
Figure IV.3 - Logiciel de modélisation 3D avec construction paramétrique.....	71
Figure IV.4 - Activation du Plugin dans le logiciel SolidWorks.....	73
Figure IV.5 - La sélection de l'option qui permet l'exportation URDF.....	73
Figure IV.6 - L'organisation et la sélection de chaque pièce du modèle pour la hiérarchie globale.....	74

Figure IV.7 - Model 3D importé de SolidWorks vers CoppeliaSim.....	74
Figure IV.8 - Notre robot sur CoppeliaSim.....	75
Figure IV.9 - Configuration des joints prismatiques.....	76
Figure IV.10 - Menu de Configuration des joints	76
Figure IV.11 - Robot se démonte et se comporte d'une manier inattendue désassemblage et comportement inattendu du Robot.....	77
Figure IV.12 - Fichier à extraire pour l'activation d'AP distant (client).....	79
Figure IV.13 - Architecture de contrôle CoppeliaSim (anciennement V-REP).....	80
Figure IV.14 - Marche convenable du Robot jusqu'aux positions finales de chaque slider.....	81
Figure IV.15 - YuMi Gripper.....	81
Figure IV.16 - La CPU 1512SP F-1 PN dans TIA Portal.....	83
Figure IV.17 - Bloc de données SCL.....	84
Figure IV.18 - L'activation du protocole OPC-UA du PLC.....	85
Figure IV.19 - Liste des variables et consigne envoyés et reçus.....	86
Figure IV.20 - La position de départ, <i>Point « A »</i>	87
Figure IV.21 - Coordonnées des sliders (x,y,z) en temps réel sur python (contrôleur).....	88
Figure IV.22 - Table de bloc de données.....	88
Figure IV.23 - Mouvement du Robot sur les trois axes.....	89
Figure IV.24 - La position d'arrivée, Point « B ».....	89
Figure IV.25 - La boucle de déclenchement d'urgence.....	90
Figure IV.26 - AAS Package Explorer : édition des données du package AAS	91
Figure IV.27 - Navigation du package AAS (Datasheet robot).....	92
Figure IV.28 - Creation des Off the Shelf Component.....	94
Figure IV.29 - Les conteneurs en état marchent	94
Figure IV.30 - Fonctionnement du DataBridge avec le serveur et l'Asset [96]	96
Figure IV.31 - Exemple d'affichage d'interface d'un AAS avec des données opérationnelles qui change en temps réel sur https://localhost:4088	96

LISTE DES TABLES

Tableau V.1 - Liste des signaux du PLC	125
Tableau V.2 - Bloc des données.....	126
Tableau V.3 - Liste des variables du contrôleur robot.....	126
Tableau V.4 - Liste des capteurs et actionneurs du robot.....	127

INTRODUCTION
GENERALE

INTRODUCTION GENERALE

Avec la montée de la quatrième révolution industrielle nommé Industrie 4.0 qui vise à intégrer les technologies de l'information et de la communication dans les processus de production et les opérations industrielles. Ce terme est même devenu incontournable pour signifier que l'entreprise a accompli son grand saut dans la quatrième révolution industrielle.

Industrie 4.0 est devenue un symbole, un logo, un marqueur d'innovation auquel on se doit d'adhérer, et se présente comme le développement logique du *Computer Integrated Manufacturing* mariant désormais l'automatisation avec l'internet. Les systèmes embarqués qui forment la base des systèmes cyber-physiques étaient déjà intégrés dans des véhicules ou dans des machines. De nombreuses industries comme l'automobile ou l'aéronautique étaient très avancées dans l'intégration de ces technologies liées à l'industrie 4.0.

La transformation induite par l'Industrie 4.0 n'est pas forcément visible. Elle a pour une large part sa source dans des codes et des algorithmes qui permettent de mettre en place des systèmes de production cyber-physiques, les systèmes cyber-physique sont l'un des technologies clés dans cette révolution industrielle. Les systèmes cyber-physiques sont des systèmes qui intègrent fortement des éléments cybernétiques et physiques. Il comprend généralement des composants physiques « capteurs, actionneurs et contrôleurs » interconnectés à un cyber infrastructure (plates-formes informatiques, réseaux de communication, algorithmes de contrôle et jumeau digital) pour permettre la surveillance et l'optimisation en temps réel des processus physiques.

Dans ce projet, nous avons considéré un module de production Cyber-Physique, conforme aux exigences de l'industrie 4.0, avec le cas d'étude d'un robot cartésien. Le modèle à concevoir est une représentation virtuelle du robot cartésien YAMAHA-SXYx-C-A1, ce modèle va prendre en charge l'intégration des métadonnées et le modèle 3D du robot cartésien, ainsi que la connexion à la partie physique du système et sa commande par un Automate Programmable Industriel.

Ce mémoire est organisé en quatre chapitres :

Le premier chapitre est dédié à l'introduction et aux généralités sur les nouvelles technologies de l'industrie 4.0.

Dans le deuxième chapitre, nous présentons un état de l'art sur les systèmes cyber-physique de production ainsi que sur le commissioning virtuel.

Le troisième chapitre sera destiné à la présentation du cas d'étude et la méthodologie adoptée pour la conception du module cyber-physique de production.

Le quatrième chapitre, est consacré à la présentation de la partie expérimentale.

Enfin, on clôture le mémoire par une conclusion générale.

CHAPITRE I :
GENERALITES
ET
DEFINITIONS

I. Chapitre I : Généralités et Définitions

I.1) Introduction

Les percées dans les technologies de l'information, les communications mobiles et la robotique ont accru l'utilisation des technologies numériques dans les entreprises du monde entier. OÙ de nombreuses technologies telles que le Cloud Computing, l'Internet des objets (IoT), le BigData et l'intelligence artificielle ont émergé.

Ces nouvelles disciplines scientifiques et technologies imprègnent l'industrie et permettent l'intégration des mondes physiques et virtuels à travers les systèmes cyber-physiques (CPS), qui marque le début de la quatrième phase de la production industrielle "Industrie 4.0". Cette dynamique devrait conduire à des changements structurels majeurs pour les entreprises, notamment en matière de fabrication et maintenance.

Ce chapitre commence par un aperçu de l'industrie 4.0 et de la fabrication dans les usines intelligentes. Ensuite, on fait un tour sur les nouvelles technologies qui caractérisent la quatrième révolution industrielle. On présente Le RAMI 4.0, Modèle d'architecture de référence Industrie 4.0, ainsi que le Shell d'administration des actifs (Asset Administration Shell) ou AAS. Enfin, on présente le concept de jumeau digital, ainsi que le jumeau digital standardisé basé sur l'AAS.

Industrie 4.0

Le terme industrie 4.0 est apparu la première fois en 2011 au Forum mondial de l'Industrie de Hanovre [58]. Le projet "Industrie 4.0" ou "Industrie du futur" correspond à une nouvelle façon d'imaginer les moyens de production. il a été inventé par un groupe de représentants de différents domaines afin d'améliorer la compétitivité de leur secteur manufacturier.

Industrie 4.0 signifie la quatrième révolution industrielle. Cette étape du processus d'industrialisation, est tout comme les trois étapes précédentes, dominée par les innovations techniques.

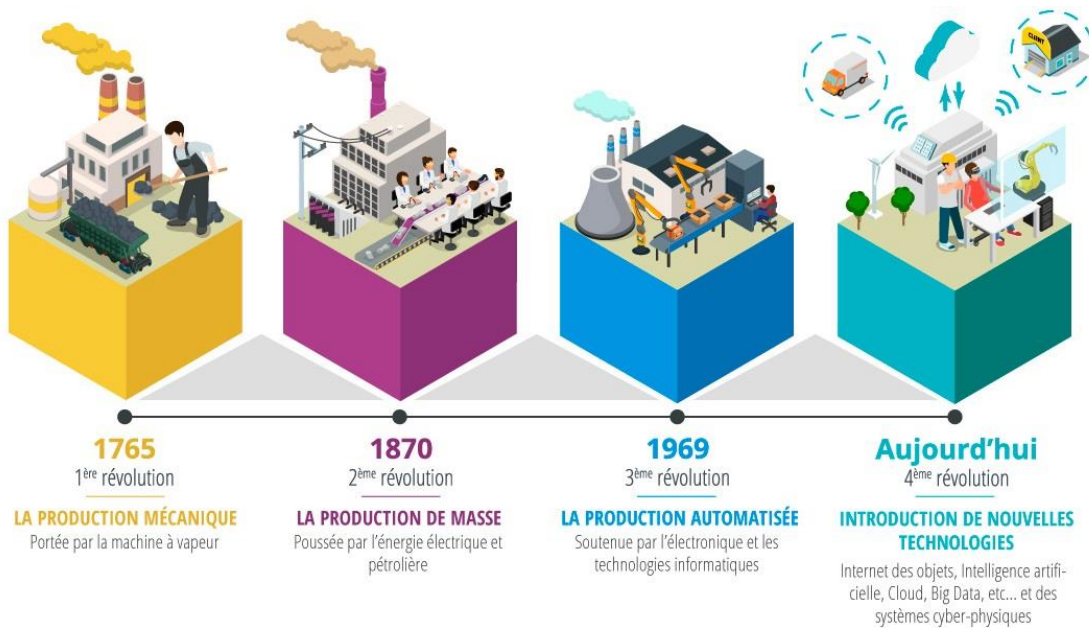


Figure I.1 - L'évolution de l'industrie.

Les premières révolutions industrielles ont été marquées par la mécanisation et l'électrification des processus de fabrication. Actuellement, la troisième étape, caractérisée par une informatisation et une automatisation croissante, se prépare doucement à céder la place à la prochaine révolution industrielle. L'industrie 4.0 se distingue par l'intégration technique des systèmes cyber-physiques dans la fabrication, ainsi que par l'Internet des objets (IoT) et les services associés aux processus industriels. Les nouvelles technologies auront un impact varié sur la création de valeur, l'organisation du travail, les services en aval et les modèles économiques des entreprises. Au cœur de tous les développements de l'industrie 4.0, le concept d'usine intelligente joue un rôle crucial dans la définition de la vision d'une nouvelle ère industrielle.

I.1.a) Concept de développement pour l'industrie 4.0

Les domaines les plus importants peuvent être liés à ce qui est appelé quatrième révolution industrielle, dont le concept repose sur plusieurs piliers principaux qui sont les suivants (Figure I.2) : Simulation de Robot autonome, intégration de systèmes, Internet des Objets, Cybersécurité, Cloud Computing, Fabrication Additive, Réalité Augmentée et Big Data.

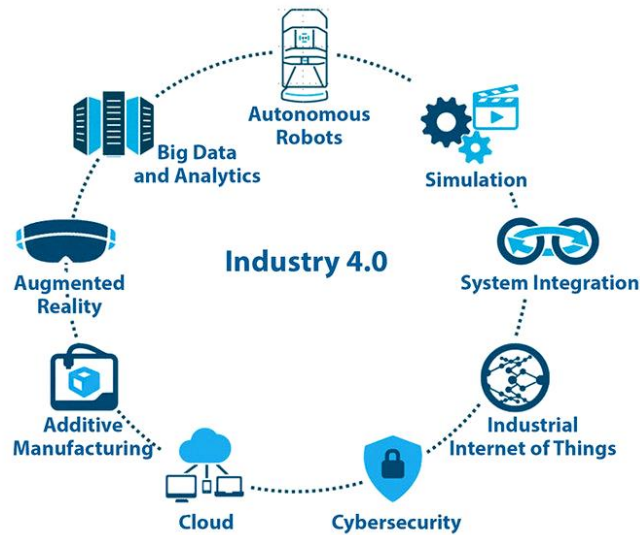


Figure I.2 - **Piliers de la quatrième révolution industrielle** (Source : BCG - Institute_Aethon.com)

✓ **La connectivité :**

La connectivité se réfère à la capacité d'un appareil, d'un réseau ou d'un système à se connecter et à communiquer avec d'autres appareils, réseaux ou systèmes. D'autre part, la communication joue un rôle crucial dans l'industrie 4.0 et la transformation industrielle. Le choix des bonnes solutions de connectivité est essentielle à l'optimisation des processus industriels. La communication fluide entre machines et systèmes d'informations permettra notamment une accélération du traitement des données et des prises de décision et aidera les opérateurs sur site pendant leurs interventions.

La connectivité d'une usine est constituée de plusieurs types de réseaux qui cohabitent pour répondre à des cas d'usage distincts : un réseau de terrain comme Modbus – par exemple - permet aux machines et automates de communiquer, quand un réseau wifi facilite les opérations en mouvement dans l'usine. Un réseau bas débit de type LoRa assure l'échange de données entre objets connectés, tandis qu'un réseau cellulaire comme la 5G, qui diminue les temps de commutation entre les antennes, est particulièrement utile pour les productions industrielles critiques nécessitant une connexion fiable et de très hauts débits.

Il existe plusieurs formes de connectivité, notamment :

1. Connectivité Internet
2. Connectivité sans fil
3. Connectivité réseau local
4. Connectivité machine à machine (M2M)
5. Connectivité sociale

La complémentarité des réseaux est le premier pilier d'une connectivité assurant la bonne transmission de la donnée dans l'usine. La circulation accélérée des données permet en effet d'examiner chaque processus à l'aide d'outils de traitement de données et d'algorithmes d'intelligence artificielle pour identifier des zones d'amélioration (rendement productif, contrôle qualité, alerte de sécurité ou maintenance prédictive).

Enfin, rassembler les données collectées aux quatre coins de l'usine nécessitera aussi une connectivité adaptée. Le recours à des protocoles de communication de machine à machine pourra croiser la donnée issue de différents appareils. Beaucoup d'usines utilisent d'ailleurs l'Open Platform Communications Unified Architecture (OPC-UA), un standard d'échange d'informations et de services qui s'adapte à plusieurs marques de machines et systèmes d'exploitation informatique. La donnée est ensuite consolidée et présentée sous forme de tableau de bord ou traitée par des algorithmes d'intelligence artificielle qui identifient des paramètres d'optimisation ou échangés entre systèmes pour une meilleure collaboration [56].

✓ **L'intelligence Artificielle :**

L'intelligence artificielle (IA) est un « ensemble de théories et de techniques mises en œuvre en vue de réaliser des machines capables de simuler l'intelligence humaine ».

Souvent classée dans le groupe des mathématiques et des sciences cognitives, elle fait appel à la neurobiologie computationnelle (particulièrement aux réseaux neuronaux) et à la logique mathématique (partie des mathématiques et de la philosophie). Il utilise les méthodes de résolution de problèmes avec quarante logiques ou algorithmes complexes. Par extension, elle comprend, dans le langage courant, les dispositifs imitant ou remplaçant l'homme dans certaines mises en œuvre de ses fonctions cognitives.

Les applications de l'IA incluent notamment les moteurs de recherche, les systèmes de recommandation, la compréhension du langage naturel, les voitures autonomes, les chatbots, les outils de génération d'images, les outils de prise de décision automatisés et les programmes compétitifs dans des jeux de stratégie.

Ses finalités et enjeux ainsi que son développement suscitent, depuis l'apparition du concept, de nombreuses interprétations, des fantasmes ou des inquiétudes qui s'inquiètent tant dans les récits ou films de science-fiction que dans les philosophiques. Si des outils pertinents d'intelligences artificielles spécialisées ont fait leurs preuves, la réalité semble encore tenir l'intelligence artificielle généraliste loin des performances du vivant ; ainsi, l'IA reste encore bien inférieure au chat dans toutes ses aptitudes naturelles [60].

✓ **La Robotique Collaborative :**

Nous vivons une révolution industrielle. Les robots sont libérés de leurs cages. Pour la première fois, il existe une réelle interaction entre l'homme et le robot sur un même espace de travail. Le robot collaboratif doit posséder à la fois les caractéristiques de performance d'un robot industriel traditionnel et une technologie sûre pour assurer la sécurité de l'opérateur [61].

La robotique collaborative est un domaine de la robotique qui se concentre spécifiquement sur le développement de robots collaboratifs, ou cobots. Il s'agit de nouveaux types de robots conçus pour travailler aux côtés de l'homme, afin de renforcer la chaîne de production et d'accroître l'efficacité. Il s'agit de robots flexibles, rentables et faciles à déployer qui rendent l'automatisation plus facile et plus accessible pour diverses entreprises. Alors que le robot d'industrie traditionnelle travaille dans un espace spécialement conçu pour lui, les cobots travaillent en collaboration avec les humains. Cela n'élimine pas le besoin de travail humain, mais le réduit. Cela permet aux employés de se concentrer sur une série d'autres tâches qui ne peuvent pas être automatisées, ce qui rend la chaîne de production plus efficace [62].

La robotique collaborative privilégie une répartition productive des tâches entre les opérateurs et les robots sur un même espace de travail. La robotique collaborative permet à la fois de gérer la pénurie de main d'œuvre et l'absentéisme mais aussi de valoriser le travail de vos opérateurs (en libérant du temps sur des postes à faible valeur ajoutée ou en les faisant monter en compétences sur la prise en main du robot) [61].

✓ **Les Jumeaux Numériques :**

Un jumeau numérique (en Anglais, Digital Twin or Device Shadow) est une réplique numérique d'un objet, d'un processus ou d'un système qui peut être utilisé à diverses fins. Intégrant la temporalisation de la conception vers l'objet tissu en précisant que la construction est l'objet initial jet réplique, cette temporalité adossée au processus de conception permet de prendre dans sa globalité la vie de l'objet bien au-delà de sa livraison et de l'extinction des garanties décennales ou trentenaires. La représentation du nombre quatre nit à la fois les éléments et la dynamique de fonctionnement d'un dispositif de l'Internet des objets sur au long de son cycle de vie.

Les jumeaux numériques intègrent l'apprentissage de l'intelligence artificielle, l'automatique et l'analyse des données avec des données pour créer des modèles de simulation numérique qui se mettent en place un jour et changent à mesure que leurs contreparties physiques changent. Un jumeau numérique apprend en permanence et se met

à jour en utilisant de multiples sources pour désigner son statut, sa condition de travail ou sa position en temps quasi réel. Ce système d'apprentissage, apprend de lui-même, en utilisant :

- Des données de capteurs qui transmettent divers aspects de son état de fonctionnement ;
- D'experts humains, tels que des ingénieurs ayant une connaissance approfondie et pertinente du domaine industriel ;
- D'autres machines sont similaires ;
- D'autres flottes de machines similaires ;
- Et de systèmes plus vastes et de l'environnement débouchent il fait partie.

Un jumeau numérique intègre également des données historiques de l'utilisation passée de la machine qu'il intègre à son modèle numérique.

Dans divers secteurs industriels, les jumeaux numériques sont utilisés pour optimiser le fonctionnement et la maintenance des actifs physiques, des systèmes et des processus de fabrication. Les jumeaux numériques constituent une technologie de formation pour l'Internet industriel des objets, où ils peuvent interagir virtuellement avec d'autres machines et personnes. Dans ce contexte de l'Internet des Objets, ils n'est pas mentionné sous l'appellation de « cyberobjets », ou encore « d'avatars digitaux » [63].

Et à travers tout cela, nous ferons valoir ce point plus tard dans ce chapitre.

✓ **L'informatique en Nuage (Cloud Computing) :**

Le Cloud Computing (CC) est un modèle à grande échelle en pleine maturation pour fournir des ressources informatiques à la demande (calcul, stockage, réseaux, plateformes et applications) sous forme de service sur Internet [1]. Avec l'évolution de la virtualisation, l'accès Internet à haut débit et surtout le soutien des principales sociétés informatiques, le rêve longtemps envisagé de "l'informatique en tant qu'utilitaire" a été réalisé, et le CC est devenu l'un des domaines qui connaissent la plus forte croissance dans l'industrie informatique. L'attrait croissant du CC résulte de son efficacité et de sa flexibilité, permettant aux clients de fournir et d'accéder rapidement à des ressources de n'importe où et à n'importe quel moment, sur la base d'un paiement à l'utilisation. Le CC permet à ses utilisateurs d'éviter les efforts d'installation et de gestion en externalisant leurs ressources matérielles et logicielles vers un environnement à grande échelle favorisant la haute disponibilité et la réduction des coûts [2].

✓ **Fabrication additive :**

L'utilisation industrielle de la technologie incrémentale permet de rationaliser le processus de production par le prototypage et la création de pièces sélectionnées de machines et d'équipements nécessaires partout et à tout moment. Grâce à la mise en place de l'impression 3D dans l'industrie [3].

L'impression 3D est un procédé de fabrication additive basé sur l'idée simple de convertir un modèle numérique en un objet solide en trois dimensions, il est possible de produire des structures plus légères et plus complexes. Au fil des années, différentes technologies d'impression 3D se sont développées dans l'industrie avec comme point commun de créer un modèle physique couche par couche.

Ce premier développement marque réellement le commencement de ce que l'on considère aujourd'hui comme la 4ème révolution industrielle, offrant des applications dans des secteurs aussi variés que le médical, l'aéronautique ou la joaillerie [4].

✓ **Réalité augmentée :**

Grâce à l'utilisation de lunettes, par exemple Microsoft HoloLens, il est possible de fournir à l'utilisateur une couche numérique supplémentaire d'informations sur des objets réels. Cela signifie que vous avez accès aux connaissances nécessaires pour accomplir les tâches. Il permet également la communication audiovisuelle sans l'utilisation d'appareils mobiles. La réalité augmentée est un support efficace pour les agents de maintenance, la planification logistique, l'assemblage et la production, ainsi que le prototypage. Pour les gestionnaires, la mise en œuvre de la technologie AR est une approche axée sur les processus pour créer de la documentation et utiliser des données pour analyser et optimiser les processus dont ils sont responsables [3].

✓ **Internet des objets industriel (IIOT) :**

L'Internet des objets (Internet of Things IoT) peut être compris comme le réseau mondial et omniprésent qui aide et fournit la fonctionnalité d'intégration du monde physique. Cela se fait par la collecte, le traitement et l'analyse des données générées par les capteurs de l'IoT, qui seront présentes dans toutes les choses et intégrées par le réseau de communication public. L'Internet des objets (IoT) peut être compris comme l'ensemble des technologies de l'information et de la communication (TIC).

Le terme IoT a été créé en 1999 par le pionnier britannique de la technologie Kevin Ashton pour développer un système dans lequel des objets physiques pourraient être

connectés à l'Internet via des capteurs [5]. Il a inventé ce terme pour déterminer l'importance des étiquettes d'identification par radiofréquence (RFID) utilisées dans les chaînes d'approvisionnement vers l'internet pour suivre les marchandises sans aucune intervention humaine. L'écosystème de IoT est constitué d'objets physiques, qui sont connectés et accessibles sur l'internet. Les "choses" de IoT pourraient être une personne ou tout objet doté d'un capteur intégré avec une adresse IP attribuée qui peut collecter et échanger les informations sur le réseau sans aucune assistance humaine (Madakam et al., 2015). La technologie intégrée dans le CPS permet sa connexion avec le monde physique pour la capture de données volumineuses et garantit une sécurité accrue d'IoT ainsi qu'une efficacité accrue du CPS. IoT prend rapidement de l'ampleur pour aider les industries à accroître la précision et l'exactitude des processus, à réduire les coûts et à percevoir les avantages de l'information en temps réel, ce qui permettrait de prendre des décisions en connaissance de cause. IoT offre la possibilité de contrôler les performances réelles et les indicateurs clés de performance d'une organisation. Cela leur permet de créer leurs systèmes de produits-services et de démêler le potentiel d'innovation du système grâce à IoT [6].

✓ **Big Data :**

Le terme "BigData" désigne une grande quantité de données structurées, semi-structurées et non structurées créées par des sources de données différentes. Le stockage et l'analyse de ces données nécessiteraient du temps et d'argent pour obtenir une valeur considérable. Par conséquent, les BigData font référence aux données massives qui ne pourraient pas être collectées, stockées, gérées, partagées, analysées et calculées par des outils de données standard dans un délai tolérable [7].

Les grandes données peuvent être définies par les caractéristiques suivantes : Volume, variété, vitesse et valeur, c'est-à-dire 4Vs [8]. En ce qui concerne le volume, l'échelle des données est très large, allant de plusieurs PB (1000TB) à ZB (un milliard de TB) [7]. Quant à la variété, elle signifie que la taille, le contenu, le format et les applications des données sont diversifiés. Par exemple, les données comprennent des données structurées (chiffres, symboles et tableaux), des données semi-structurées (arbres, graphiques et documents XML) et des données non structurées (journaux, audios, vidéos, documents et images) [8]. La vitesse signifie que la génération de données est rapide et que le traitement des données exige une grande rapidité. Face à des données massives, la vitesse est la vie des entreprises. Pour la valeur, l'importance des données n'est pas le grand volume, mais plutôt l'énorme valeur.

Comment extraire la valeur de données massives par le biais d'algorithmes puissants est la clé pour améliorer la compétitivité.

✓ **Les systèmes Cybers physiques**

Le terme "système cyber-physique" a été initialement inventé par Ellen Gill en 2006. Le CPS peut être considéré comme évolution naturelle des systèmes embarqués. Il est souvent appelé système informatique de nouvelle génération qui utilise des techniques de calcul intelligentes associées au monde physique et aux unités de calcul. Le CPS peut interagir avec les systèmes du monde réel par le biais du calcul, de la communication et des contrôles. L'interaction des unités de calcul et des unités physiques conduit à des mises en œuvre avancées de l'internet des objets (IoT).

L'IoT et les CPS sont conçus pour prendre en charge des applications en temps réel qui peuvent gérer de nombreux ensembles de données environnementales. En d'autres termes, le CPS est une combinaison de commandes numériques et de l'environnement physique [9].

Le système cyber-physique est constitué de composants cybernétiques et de composants physiques, c'est pourquoi il est appelé système cyber-physique. Le CPS est basé sur un système informatique de traitement de l'information, qui est intégré dans un produit, comme une voiture, un avion ou un autre appareil. Ces systèmes informatiques sont utilisés pour effectuer des tâches spécifiques.

Ce système informatique interagit avec l'environnement physique au moyen de capteurs et d'actionneurs. Ces systèmes embarqués ne sont plus autonomes, ils partagent leurs données via des réseaux de communication tels que l'internet et l'informatique en nuage, où les données de nombreux systèmes embarqués peuvent être collectées et traitées. Ils créent ainsi un système de systèmes. Les systèmes embarqués connectés peuvent être contrôlés et décentralisés par une unité de calcul. Les données collectées peuvent être traitées automatiquement ou par une interface homme-machine (IHM).

I.1.b) Reference Architectural Model Industry 4.0 (RAMI 4.0) [57]

Le RAMI 4.0, Modèle d'architecture de référence Industrie 4.0, a été développé par l'Association allemande des fabricants électriques et électroniques (ZVEI) pour soutenir les initiatives de l'Industrie 4.0, qui sont de plus en plus acceptées dans le monde entier.

RAMI 4.0, ou Reference Architectural Model Industry 4.0, est une carte en trois dimensions montrant comment aborder la problématique d'une Industrie 4.0 de manière systématique et structurée. RAMI 4.0 est un modèle unifié pour tous les composants qui garantit à tous les participants impliqués dans un écosystème Industrie 4.0 de partager des données et des informations de manière efficace et efficiente. RAMI 4.0 cartographie tous les acteurs de l'industrie connectée selon trois axes de définition listés ci-dessous :

- **Couches d'architecture** : cette couche contient des sous-couches telles que Business, Fonctionnel, Information, Communication, Intégration et Asset.
- **Flux de valeur du cycle de vie** : cette couche contient des sous-couches telles que l'utilisation du développement, de la production et de la maintenance.
- **Niveaux hiérarchiques** : cette couche contient des sous-couches telles que Produit, Champ, Appareil, Appareil de contrôle, Station, Centres de travail, Entreprise et Monde connecté.

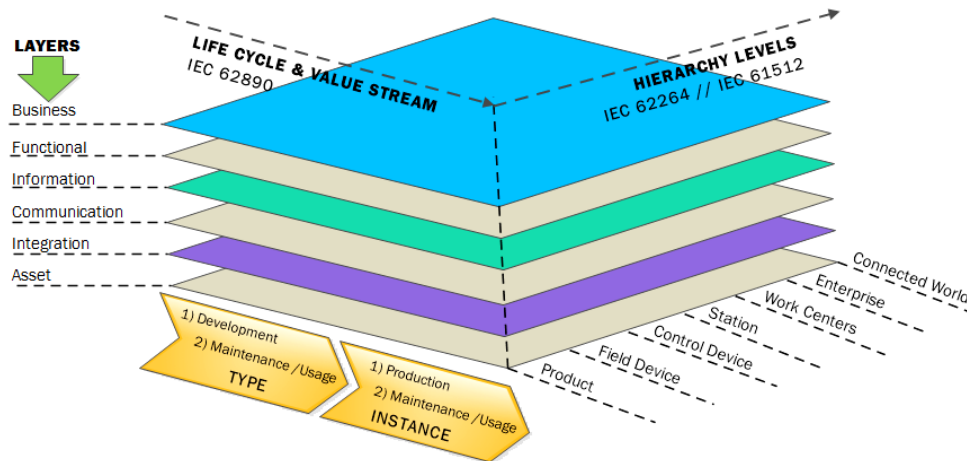


Figure I.3 - L'architecture de RAMI 4.0.

- ✓ **Flux de valeur du cycle de vie** : (Figure I.4) l'axe du flux de valeur du cycle de vie est divisé en type et instance. Le Type est divisé en Développement et Maintenance/Utilisation, tandis que l'Instance est divisée en Production et Maintenance/Utilisation. Un type représente l'idée initiale du développement du produit, tandis que chaque produit fabriqué représente une instance de ce type. Le flux de valeur dans la production entièrement numérisée peut être considéré en conjonction avec les processus à valeur ajoutée, car il permet de relier les achats, la planification de la production, la logistique, la qualité, les clients et les fournisseurs. En termes simples, si un produit est dans un état de développement, nous l'appelons un "Type". Une fois qu'il passe en production, il devient alors une "Instance". Chaque fois qu'un produit est repensé ou qu'une nouvelle fonctionnalité lui est ajoutée, son état redevient "Type".

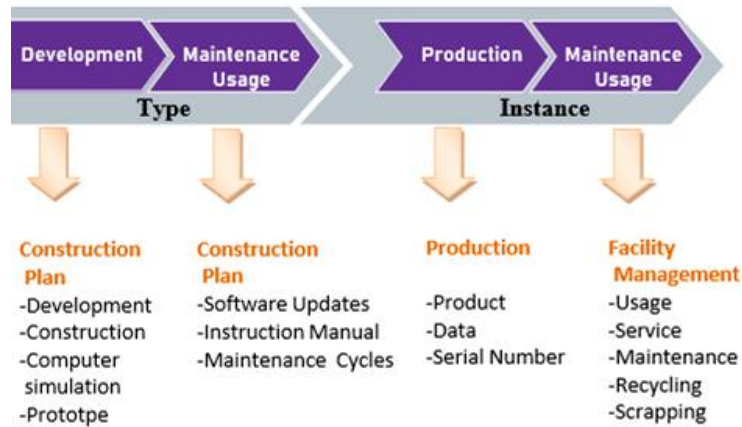


Figure I.4 - Le flux de valeur du cycle de vie.

- ✓ **Niveaux hiérarchiques :** (Figure I.5) L'architecture de l'industrie 4.0 au niveau hiérarchique montre une affectation fonctionnelle des composants. Cet axe au sein d'une entreprise ou d'une usine suit les normes CEI 62264 et CEI 61512. Le niveau au-dessus et au-dessous du domaine des normes CEI représente des étapes supplémentaires et décrit également des groupes d'usines, la collaboration au sein de sociétés d'ingénierie externes, de fournisseurs de composants et de clients. Par conséquent, les niveaux de hiérarchie sont : produit, appareil de terrain, appareil de contrôle, station, poste de travail, entreprise et monde connecté.

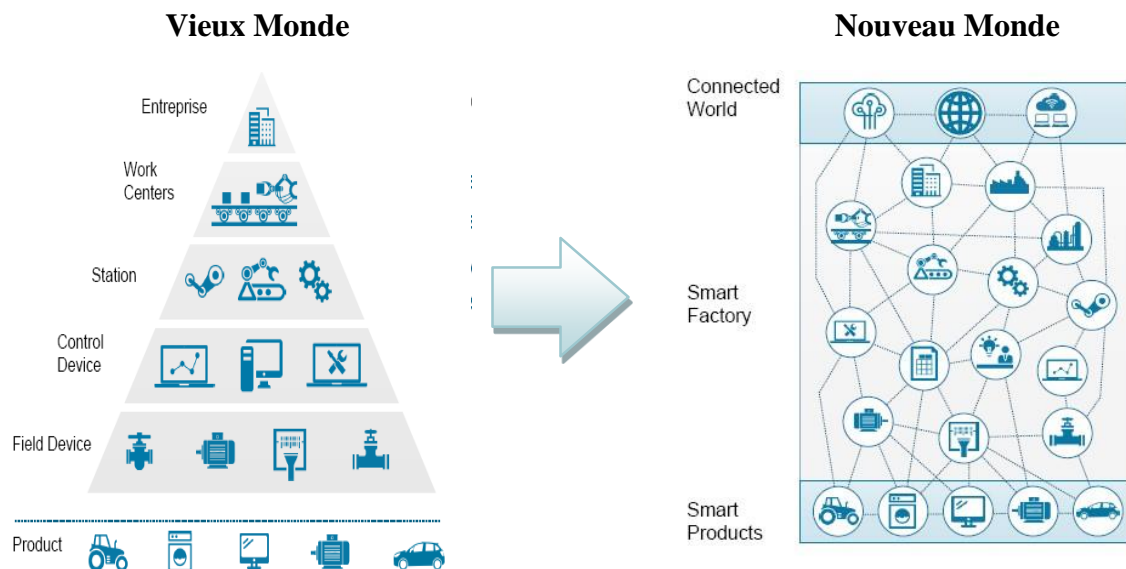


Figure I.5 - La inter connectivité de la nouvelle industrie

- ✓ **Couches d'architecture** : (Figure I.6) la couche d'architecture comprend des sous-couches comme mentionné ci-dessus. Cette couche permet le développement de solutions logicielles Industrie 4.0 de manière cohérente afin que des opérations de fabrication différentes et interdépendantes soient interconnectées en tenant compte du monde physique et numérique.



Figure I.6 - Les sous-couches de la couche d'architecture.

RAMI 4.0 décompose les processus complexes en packages, ce qui les rend facilement compréhensibles, et inclut dès la conception la confidentialité des données et la sécurité informatique. Il aborde et répond à toutes les problématiques de sémantique, d'identification, de fonctions, de normes de communication, d'internationalisation et de partenariat pour l'usine intelligente [57].

Avec l'architecture RAMI 4.0, l'usine n'est pas une superposition de couches mais un réseau d'interaction entre les produits intelligents et le monde connecté. Les fonctions sont distribuées à tous les participants permettant des opérations plus flexibles entre les systèmes et les machines. Pour connecter l'objet physique au monde de l'Industrie 4.0, la **RAMI 4.0** fournit un **shell d'administration** qui vient se placer sur le système ; le système étant un capteur, une machine, une unité ou une installation. De plus, chaque actif ayant son propre **shell d'administration**, plusieurs actifs peuvent alors former une unité avec un shell d'administration commun pour une communication et une gestion de niveau supérieur. Une fois connecté au système physique, le shell sert d'interface de communication normalisée du réseau pour partager toutes les données et informations sur l'actif [57].

Comme décrit, la fonction **RAMI 4.0** est de transformer des objets physiques en leur jumeau numérique interopérable. Amenant le monde industriel dans l' **IoT** , **RAMI 4.0** est à l'origine de la révolution de l'Industrie 4.0 [57].

I.1.c) Asset Administration Shell

L'Asset Administration Shell (AAS) est un concept clé de l'Industrie 4.0, qui consiste en une représentation numérique complète d'un Asset physique ou d'un système dans un environnement industriel. Il offre une vue unifiée et détaillée de l'Asset, incluant ses caractéristiques, fonctionnalités, historique, relations et capacités.

L'AAS fonctionne comme un "passeport numérique" pour les Asset industriels, permettant la collecte, le stockage et l'échange d'informations pertinentes tout au long du cycle de vie de l'actif. Il peut être utilisé pour décrire des machines, des équipements, des produits, des composants ou même des processus de production.

Grâce à l'AAS, les entreprises peuvent accéder en temps réel à des informations détaillées sur leurs actifs, ce qui leur permet de prendre des décisions éclairées, de planifier la maintenance, d'optimiser les performances, de garantir la qualité et d'améliorer l'efficacité opérationnelle. L'objectif ultime est de créer un écosystème industriel connecté où les actifs peuvent communiquer, collaborer et s'adapter de manière autonome pour optimiser les processus de production.

D'autre part, le terme "I4.0 Component" fait référence à un composant lié à l'Industrie 4.0, également connue sous le nom de Quatrième Révolution Industrielle. Les I4.0 Components sont les différents éléments technologiques utilisés dans le cadre de l'Industrie 4.0. Ils comprennent la plupart, ou dans certains cas, l'intégralité des technologies qui caractérisent la quatrième révolution industrielle telles que l'Internet des objets (IoT), l'intelligence artificielle (IA), la réalité virtuelle (RV), la robotique avancée, l'informatique en nuage, la fabrication additive (impression 3D), l'analyse de données, la cybersécurité, etc.

Ces composants sont intégrés dans les usines et les systèmes de production pour permettre une automatisation et une connectivité accrues entre les machines, les systèmes et les personnes. Ils visent à améliorer l'efficacité, la productivité, la flexibilité et l'agilité des processus industriels.

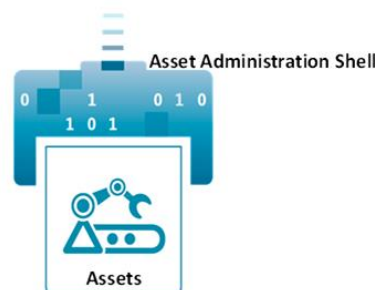


Figure I.7 - Composant de fabrication intelligente - composant I4.0.

Les Assets compatibles I4.0 ou les composants I4.0 sont des actifs qui ont au moins des capacités de communication passives et sont administrés comme une entité dans un système d'information par un Shell d'administration.

Il existe un certain nombre d'exigences pour le Shell d'administration, qui sont les suivantes [59] :

- Le Shell d'administration se compose du corps et de l'en-tête.
- Le corps contient des informations sur l'actif en question.
- L'en-tête contient des informations sur la façon dont l'actif est utilisé.
- Le Shell d'administration contient les éléments clés, le manifeste et le gestionnaire de composants.
- Les informations du Shell d'administration doivent être accessibles via une architecture orientée services (SOA) et doivent tenir compte des exigences de sécurité correspondantes.
- Le Shell d'administration représente des informations sur les aspects de l'application.
- Le Shell d'administration est structuré à l'aide de vues.
- L'administration a un identifiant unique.
- L'actif a un identifiant unique.
- Même une usine peut être un actif doté d'un Shell d'administration et accessible à l'aide de son ID. Il devrait être possible d'appliquer le concept d'imbrication.
- Les types et les instances doivent être indiqués comme tels.
- Le Shell d'administration peut contenir des références à d'autres Shell d'administration ou des informations I4.0.
- Des propriétés supplémentaires, telles que celles spécifiques au fabricant, doivent être possibles.
- Un ensemble minimum fiable de propriétés doit être défini pour chaque Shell d'administration.

I.1.d) Les jumeaux numériques (Digital Twin)

L'un des principaux concepts de l'Industrie 4.0 est le "jumeau numérique" (Digital Twin "DT") qui permet l'acquisition, le partage et l'accès à une plus grande variété de données avec une interopérabilité et une disponibilité sans précédent. Le DT est la représentation virtuelle d'un produit, utilisée pour la conception, la simulation, la surveillance, l'optimisation ou l'entretien du produit. Il offre de nombreuses possibilités pour améliorer les performances des produits, anticiper les étapes de leur cycle de vie, ou encore planifier les activités de toute une chaîne de production. Classé parmi les dix premières tendances

technologiques stratégiques du Gartner, le jumeau numérique s'impose progressivement dans le monde de l'industrie.

Le terme Digital Twin a été publié pour la première fois par la National Aeronautics and Space Administration (NASA) en 2012 [10]. Ils définissent le Digital Twin comme une simulation ultra-réaliste et à grande échelle qui utilise les meilleurs modèles physiques disponibles, les données des capteurs et les données historiques pour refléter un ou plusieurs systèmes réels. Les données pertinentes sont collectées tout au long du cycle de vie du système [10]. Les principales dimensions sont le temps et le niveau de détail. Les données pertinentes sont acquises en continu et transférées en temps réel au Digital Twin du système.

Le jumeau lui-même peut être utilisé à différents niveaux de détail. En plus de l'état actuel du système, un état historique ou des états alternatifs peuvent être utilisés pour l'analyse [11] [12]. Depuis la publication de la NASA, trois fonctions principales du Digital Twin peuvent être résumées : Prévission : exécution d'études en amont du système Sécurité : surveillance et contrôle de l'état du système en termes de prédiction continue pendant le fonctionnement du système Diagnostic : analyse des perturbations imprévues pendant le fonctionnement du système.

Aujourd'hui, Digital Twin a évolué vers un concept plus large qui fait référence à une représentation virtuelle des éléments de production tels que le personnel, les produits, les actifs et les définitions de processus, un modèle vivant qui se met à jour et change continuellement lorsque la contrepartie physique change pour représenter le statut, les conditions de travail, la géométrie des produits et les états des ressources de manière synchrone. La représentation numérique fournit à la fois les éléments et la dynamique de la façon dont une "chose" physique fonctionne et vit tout au long de son cycle de vie.

Un jumeau numérique (DT) est une représentation virtuelle qui sert de contrepartie numérique en temps réel d'un objet ou d'un processus physique. Il fonctionne comme un outil qui permet de simuler, de surveiller et de contrôler virtuellement le processus (Pérez et al., 2020). Ces copies du système doivent pouvoir interagir avec les homologues physiques de manière bidirectionnelle. Ils collectent des données du modèle physique à l'aide de capteurs, de contrôleurs PLC et d'interactions humaines ; et en recréant une représentation numérique en utilisant des technologies de visualisation 3D, telles que la réalité virtuelle et augmentée (RA). Tout ce qui précède permet de construire une représentation réalisable du modèle physique.

Pour développer un DT, il faut d'abord construire une ombre numérique (DS), qui n'est que la représentation du système réel. Le système actuel envoie des données à la DS qui les acquiert puis simule la maquette numérique. Dans ce cas, l'interaction est monodirectionnelle. Le Digital shadow devient un Digital Twin complet lorsque le flux de communication est bidirectionnel, du système réel vers le DT et inversement (Figure I.7).

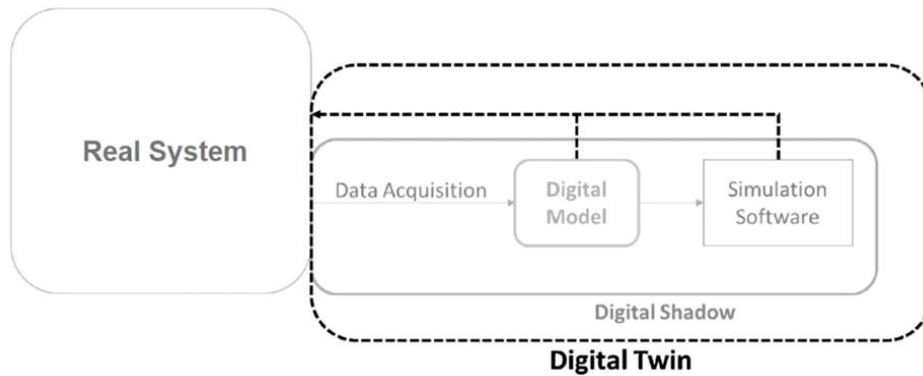


Figure I.8 - Le schéma de DT (Cimino et al., 2019)

Le concept de DT recherche la connexion et l'alimentation continue en temps réel et la surveillance des informations du système réel au système virtuel, de sorte qu'il peut être un objet d'étude et un outil d'optimisation, de débogage et de test aussi précis que le système physique (Cimino et al., 2019). Ces buts et objectifs de la technologie DT peuvent être atteints de manière fructueuse grâce à l'utilisation avancée des technologies IoT.

➤ Application

Les jumeaux numériques ont véritablement révolutionné le monde industriel en modifiant la phase de développement d'un produit ou l'utilisation d'un équipement. Les apports des jumeaux numériques peuvent être appliqués à plusieurs niveaux : à partir d'un objet, à un processus de fabrication, à une usine, voire à une ville entière, ce qui permet d'imaginer un large éventail d'applications potentielles.

Conception de produits : Le jumeau numérique a changé radicalement la phase de développement d'un produit, au lieu de créer un objet, de le tester, puis d'y apporter des améliorations. La visualisation continue de la construction d'un objet sur son modèle numérique offre l'avantage de le comparer en temps réel. Avec l'objectif final, ce développement garantit le bon déroulement de l'opération et permet de faire les bons choix en matière d'ergonomie, de matériaux et de conception. Le jumeau numérique utilise pleinement les fonctions de calcul et de simulation dans la phase de conception du produit pour permettre l'optimisation des schémas de conception et améliorer les modèles de conception. Le modèle de jumeau numérique peut également faciliter la vérification des fonctions, des structures et de la fabrication des produits [13].

Dans un modèle de référence complet basé sur le concept de Skin Model Shapes est présenté pour servir de jumeau numérique du produit physique dans la conception et la

fabrication. Dans la phase de production, l'ensemble du processus de fabrication peut être contrôlé et optimisé en temps réel par le jumeau numérique. De l'entrée des matières premières à la sortie des produits finis, les informations géométriques, les équipements, les outils et l'environnement sont gérés pour permettre de passer d'une production de masse à une production personnalisée. Dans Rosen et al. [14], Digital Twin est considéré comme la clé de la mise en œuvre de l'autonomie et de la flexibilité dans la fabrication intelligente. Dans Söderberg et al. [15], le jumeau numérique est employé pour garantir l'assurance de la géométrie du produit dans la production individualisée.

Le cycle de vie d'un produit [16] réalise une boucle fermée et une optimisation de la conception, de la production, de l'exploitation, et de la maintenance. Il permet une vue systématique des différentes phases du produit et des applications du jumeau numérique dans différents cycles de vie du produit. Pendant la phase d'exploitation, les données de performance d'un produit sont systématiquement collectées dans le cadre du jumeau numérique pour vérifier et mettre à jour les modèles existants. Les connaissances acquises à partir des conditions réelles d'exploitation peuvent être utilisées pour la modification des produits de la prochaine génération. Dans Alam et al. [17], un modèle de référence de l'architecture du jumeau numérique est proposé pour les systèmes cyber-physiques basés sur le Cloud (C2PS), et il est appliqué à un système d'aide à la conduite de véhicules basé sur le traitement des informations à distance.

Dans Abramovici et al. [17], un modèle de jumeau numérique est utilisé pour réaliser la reconfiguration de produits intelligents au stade de l'utilisation. L'acquisition de données multimodales est intégrée dans un jumeau numérique pour réaliser le système de production cyber-physique [17]. Dans la phase de service, le jumeau numérique peut fournir des services à valeur ajoutée avec le soutien de la simulation physique et de l'intelligence pilotée par les données.

Surveillance des conditions Le jumeau numérique peut être utile pour comprendre, anticiper et optimiser les performances d'un objet ou d'un système. En combinant une vue continue d'un objet et l'ajout de données de suivi des performances, le jumeau numérique permet de réaliser des analyses et d'identifier le risque d'échec à un stade précoce. Différents services, tels que le diagnostic des défauts, le dépannage, la prévision de la durée de vie restante et la maintenance peuvent être mis en œuvre pour prendre des décisions en temps utile et réduire le risque d'accident. Dans l'article de Magargle et al. [18], un modèle jumeau multi-physique est construit pour surveiller l'état du système de freinage sous

plusieurs angles. La NASA espère réaliser la gestion de la santé de l'avion et la prédiction de la durée de vie résiduelle en construisant un modèle jumeau numérique multi physique et multi-échelle [10]. Wang et al. [19] ont introduit un jumeau numérique de diagnostic de défauts pour les machines tournantes avec le processus de mise à jour utilisant un schéma basé sur l'analyse de sensibilité des paramètres pour améliorer l'adaptabilité du modèle.

➤ **Jumeau numériques et les autres technologies**

Il existe divers points de vue sur les relations entre le jumeau numérique et d'autres concepts, tels que la simulation, les systèmes cyber-physiques (CPS) et l'internet des objets (IoT). Bien que ces concepts soient étroitement liés, ils sont, par nature, différents en ce qui concerne le concept, les éléments de base et l'application.

- **Jumeau numérique et l'industrie 4.0**

L'industrie devient intelligente à tous les niveaux, du dispositif physique aux réseaux de production, en passant par la gestion d'usine, en acquérant des capacités d'apprentissage, de configuration et d'exécution grâce à l'intelligence cognitive. Cette section décrit la tendance de l'industrie intelligente et examine la connotation de la fabrication intelligente basée sur le jumeau numérique, en soulignant l'impact que le jumeau numérique pourrait avoir pour l'industrie de la future. Dans la fabrication intelligente, une " entité " physique dans une usine est connectée à l'Internet industriel via des passerelles cybernétiques standard et abstraite comme un jumeau numérique dans le cyberspace.

Chaque jumeau numérique dans le cyberspace est une abstraction de son homologue dans le monde physique en reflétant son statut physique. Le cyberspace stocke et traite les données en continu provenant des objets physiques connectés. Ces données sont utilisées pour modéliser, simuler et prédire l'état de chaque objet physique dans des conditions de travail dynamiques. L'utilisation généralisée de technologies intelligentes, telles que le traitement des données de grande taille et l'intelligence artificielle, permet d'extraire l'intelligence de fabrication à chaque instant des activités de fabrication. L'intelligence collective dans les usines connectées localement et dans le cyberspace ouvre la voie à des changements spectaculaires des aspects du fonctionnement intra-entreprise, de la collaboration inter-entreprises et du modèle de production.

- **Jumeau numérique et Internet des Objets (IoT)**

L'IoT fait référence aux connexions entre un réseau de biens physiques à travers lesquels les données peuvent circuler entre elles. Ces connexions sont rendues possibles par la mise en œuvre sécurisée de réseaux informatiques, d'Internet et de protocoles de communication. Cependant, malgré la connectivité, l'IoT n'inclut pas l'idée de modèles numériques dans le cyberspace. L'IoT est l'infrastructure de l'espace physique qui permet de connecter des biens physiques [17].

- **Jumeau numérique et BigData**

Le processus de production devient de plus en plus complexe et il est difficile d'identifier rapidement les problèmes du processus de fabrication de manière traditionnelle. Les données peuvent refléter ces problèmes visibles et invisibles. Les BigData apportent plus d'efficacité, une meilleure compréhension et plus d'intelligence à la production. En outre, le jumeau numérique permet de réaliser la corrélation et l'ajustement dynamique entre la planification et la mise en œuvre de la fabrication, ainsi que de promouvoir la prédiction, le diagnostic et la maintenance des défauts de manière numérique. Bien qu'il y ait de nombreuses différences entre le jumeau numérique et les données importantes, ils jouent des rôles complémentaires dans la fabrication [20].

Les BigData pourraient être considérées comme un élément essentiel du jumeau numérique. Sans les BigData, la plupart des fonctions du jumeau numérique seraient le château dans les airs. Et sans le jumeau numérique, l'analyse des données importantes et le processus de fabrication ne seraient pas parallèles. La convergence du jumeau numérique et des BigData peut briser les barrières entre les différentes phases du cycle de vie du produit et raccourcir le cycle de développement et de vérification du produit.

- **Jumeau numérique et les CPS**

Le jumeau numérique, CPS et l'IoT utilisent tous des réseaux et des capteurs, le jumeau numérique est un concept différent mais lié au CPS ou à l'IoT. Un CPS se caractérise par un asset physique et son jumeau numérique. En revanche, un jumeau numérique est limité au modèle numérique, et non à l'asset physique de jumeau, bien qu'un jumeau numérique ne puisse pas vivre sans son asset de jumeau dans l'espace physique. En d'autres termes, le jumeau numérique représente la condition préalable au développement d'un CPS [21].

- **Jumeau numérique et la simulation**

Les simulations et les jumeaux numériques utilisent tous deux des modèles numériques pour reproduire les différents processus d'un système, un jumeau numérique est en fait un environnement virtuel, ce qui le rend considérablement plus riche à étudier. La différence entre le jumeau numérique et la simulation est en grande partie une question d'échelle : Alors qu'une simulation étudie généralement un processus particulier, un jumeau numérique peut lui-même exécuter toutes sortes de simulations afin d'étudier plusieurs processus. Les simulations ne bénéficient généralement pas de données en temps réel. Mais les jumeaux numériques sont conçus autour d'un flux d'informations bidirectionnel qui se produit d'abord lorsque les capteurs de l'objet fournissent des données pertinentes au processeur du système, puis se reproduit lorsque les informations créées par le processeur sont partagées avec l'objet source original.

En disposant de données de meilleure qualité et constamment mises à jour, relatives à un large éventail de domaines, combinées à la puissance de calcul supplémentaire qui accompagne un environnement virtuel, les jumeaux numériques sont en mesure d'étudier davantage de questions à partir de points de vue bien plus nombreux que ne le permettent les simulations standard avec un potentiel d'amélioration des produits et des processus bien supérieur [64].

Kunath et al. [21] Définissent le jumeau numérique d'un objet physique comme la somme de toutes les données logiquement liées, c'est-à-dire les données d'ingénierie et les données opérationnelles, représentées par un modèle de données sémantique. Alors que les données d'ingénierie sont générées une fois et mises à jour si nécessaire, par exemple les modèles 3D, les modèles de simulation spécifiques ou les spécifications des matériaux, les données opérationnelles sont recueillies et traitées en temps réel. Selon cette définition, le jumeau numérique peut représenter les états historiques et en temps réel de l'objet physique. En utilisant des applications spécifiques, une visualisation en temps réel de l'objet physique peut être générée, en combinant un modèle 3D avec des données en temps réel. En outre, différentes simulations peuvent être réalisées, en utilisant des modèles de simulation existants ou en générant de nouveaux modèles qui feront partie du jumeau numérique.

Un jumeau numérique est une réplique numérique d'une " entité " du monde réel. Si cela ressemble à une simulation, le jumeau numérique est bien plus que cela. Un jumeau numérique est une représentation haute-fidélité de la dynamique opérationnelle de son homologue physique, rendue possible par une synchronisation en temps quasi réel entre le cyberspace et l'espace physique. La dynamique opérationnelle est un élément essentiel d'un jumeau numérique car le comportement d'un jumeau est basé sur des données en temps quasi réel provenant de son homologue physique. La simulation se concentre sur ce qui pourrait

se passer dans le monde réel (scénario d'hypothèses), mais pas sur ce qui se passe actuellement.

Dans le contexte de la fabrication, un jumeau numérique peut être utilisé pour la surveillance, le contrôle, le diagnostic et la prédiction, autre que la simple simulation [13].

➤ **L'architecture du jumeau numérique**

- **Architecture à 3 couches**

Le DT tridimensionnel traditionnel a été proposé pour la première fois en 2003 par M. Grieves dans le cadre de son cours de gestion du cycle de vie des produits (PLM) à l'université du Michigan [10]. Cela peut être considéré comme l'origine du DT.

Le DT en trois dimensions comprend trois parties [10] :

1. Une entité physique dans l'espace physique.
2. Une entité virtuelle dans l'espace virtuel.
3. Une connexion de données et d'informations qui relie les entités physiques et virtuelles entre elles.

- **Nouvelles exigences de l'architecture du DT**

- ✓ Des domaines d'application plus larges
- ✓ L'obligation d'adopter les nouvelles technologies de l'information
- ✓ Exigences en matière de données
- ✓ Exigences en matière de services
- ✓ Une modélisation virtuelle haute-fidélité

- **Architecture à 5 couches**

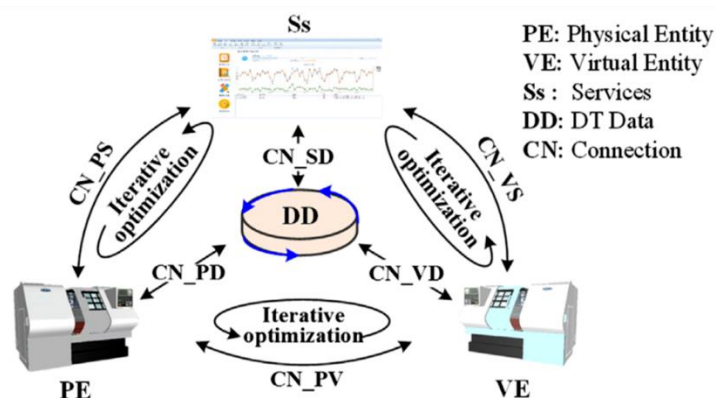


Figure I.9 - Architecture de jumeau numérique en 5 couches

- ✓ **Entité physique** : L'entité physique est constituée de divers sous-systèmes et dispositifs sensoriels.
- ✓ **Entité virtuelle** : En tant qu'image miroir fidèle de l'entité physique, l'entité virtuelle est représentée par le modèle géométrique, le modèle physique, le modèle de comportement et le modèle de règles.
- ✓ **Services** : Les services comprennent les services pour l'entité physique et l'entité virtuelle. Elles visent de faire fonctionner l'entité physique comme prévu par une régulation en temps réel et maintient une haute-fidélité de l'entité virtuelle avec l'entité physique grâce à l'étalonnage des paramètres du modèle [16].
- ✓ **Données DT** : Les données DT comprennent cinq ensembles de données :
 1. Les données de l'entité physique, comprenant principalement les états d'exploitation et les conditions de travail.
 2. Les données de l'entité virtuelle, qui comprennent les paramètres du modèle et les données d'exploitation du modèle.
 3. Les données des services, qui décrivent l'encapsulation, la combinaison, l'invocation, etc. des services.
 4. Les données de la connaissance du domaine, qui sont extraites des données collectées ou obtenues à partir des bases de données et des systèmes existants.
 5. Les données de fusion peuvent être obtenues par des algorithmes de fusion de données.
- ✓ **Connexion** : La connexion comprend la connexion entre Services et Données DT, l'entité physique et Données DT, l'entité virtuelle, Données DT, l'entité physique et Services, l'entité virtuelle et Services, et l'entité physique et l'entité virtuelle. Chacun d'eux est bidirectionnel.

➤ **Le Jumeau Digital (Digital Twin) Standardisé**

Le digital Twin standardisé est une initiative de l'association internationale des jumeaux digitaux industriels, appelée en anglais : International Digital Twin Association (IDTA). Qui se considère comme une alliance puissante qui façonne activement l'avenir du jumeau numérique de manière innovante. Ses membres actifs comprennent de nombreuses

entreprises de premier plan de l'industrie. L'expertise conjointe dans de nombreux domaines jette les bases pour connaître les exigences pratiques exactes auxquelles le jumeau numérique doit répondre et pour les remplir en termes techniques et organisationnels avec IDTA.

Le principe du digital twin standardisée est conçu sur les Shells d'administration des Ressources, appelé en anglais : Asset Administration Shell (AAS). Le « AAS » aide à l'implémentation des jumeaux digitaux pour l'industrie 4.0 et à la création de l'interopérabilité entre les solutions des différents fournisseurs impliqués dans les usines 4.0 [23].

C'est ce type de Jumeau digital qui est utilisé dans notre module Cyber physique de production présenté dans ce mémoire. En effet, l'AAS une fois unis avec son asset (ressource physique), il forme le composant I4.0 qui constitue l'unité élémentaire de l'architecture RAMI 4.0. Le composant I4.0, dans ce contexte, peut être considéré comme une mise en œuvre concrète et pratique d'un module cyber-physique de production conformément à l'architecture RAMI 4.0. Il peut représenter une unité fonctionnelle autonome, évolutive et interopérable qui intègre des éléments physiques, tel que des capteurs ou des actionneurs, des systèmes informatiques et des capacités de communication pour réaliser des tâches spécifiques dans un environnement de production.

1.2) Conclusion

Ce chapitre vise à passer en revue les généralités et les définitions modernes dans le domaine industriel d'un point de vue technologique. Les nouvelles technologies de l'industrie 4.0 ont été étudiées pour fournir des solutions dans les usines intelligentes. Cependant, ces solutions ne sont pas encore largement applicables du fait que la modélisation fidèle des données collectées à partir des équipements au niveau du cloud a été retardée. Là où le modèle « Digital Twin » a émergé pour connecter les appareils et leur affichage numérique afin d'assurer une communication continue en temps réel entre les appareils et les services.

Dans le chapitre suivant nous présentons un état de l'art sur les systèmes cyber-physiques (CPS), Les systèmes de Production cyber-physiques (CPPS) et Le commissioning virtuel (CV).

CHAPITRE II :

ETAT DE L'ART

II. Chapitre II : L'état de l'art

II.1) Introduction

Dans ce chapitre, nous abordons les systèmes cyber-physiques de production (CPPS) ainsi que leur vérification et validation utilisant le commissioning virtuel. En premiers lieux, nous donnons un bref aperçu théorique sur les CPPS, suivi d'un état de l'art sur le sujet. La seconde partie de ce chapitre traite le sujet du commissioning virtuel. Dans cette partie nous donnons la définition et les différents types cette technique de vérification et de validation, ensuite nous abordons les différents travaux qui traitent le sujet du commissioning virtuel des systèmes cyber-physiques de production.

II.2) Systèmes Cyber-Physiques de Production

II.2.a) Introduction

L'arrivée massive de nouveaux produits sur le marché, l'amélioration continue des produits existants et les changements de législation concernant l'environnement et la sécurité font que les industries doivent s'adapter pour rester compétitives [22].

Les Systèmes Cyber-Physiques (SCP) évoluent dans un environnement imprédictible et sont soumis à de nombreux changements volontaires ou non. L'adaptabilité, la connectivité et la distribution du système de commande sont des caractéristiques qui permettent d'adapter le système à l'évolution constante des produits et des besoins du consommateur [23].

II.2.b) Les Systèmes Cyber-Physiques

Un système cyber-physique en anglais « Cyber-Physical System », ou CPS en abrégé : est un système ou où des éléments informatiques collaborant pour le contrôle et la commande d'entité physique [22]. Une génération de précurseurs des systèmes cyber- physiques est très importante donc elle peut être trouvée dans plusieurs domaines comme l'automobile, aéronautique, les processus chimique, l'énergie, la santé, l'infrastructure civile, la fabrication, le transport, les divertissements et les appareils électroménagers ce qui peut être synthétisé par comme montré à la (Figure II.1).



Figure II.1 - Domaine d'utilisation du CPS [22]

Cette génération est désignée par le terme systèmes embarqués. Dans ses derniers l'accent tend à être plus, sur les éléments informatiques et moins sur les liens entre les éléments informatiques et physiques. [26]

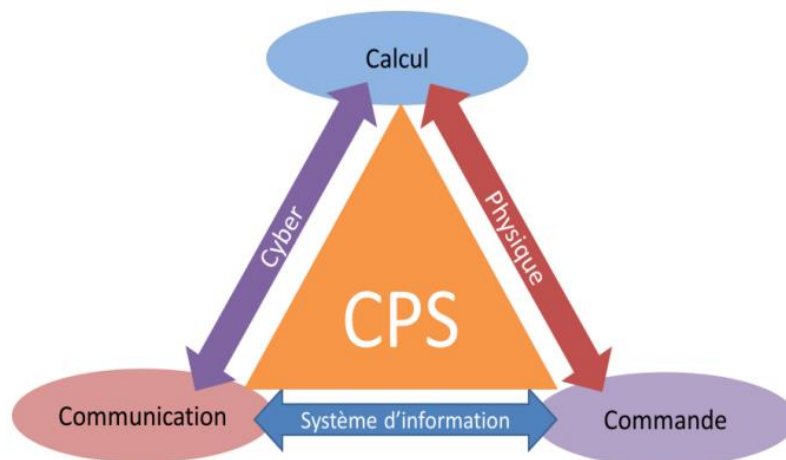


Figure II.2 - Domaine d'utilisation du CPS

À la différence des systèmes embarqués traditionnels, un système cyber-physique à part entière est généralement conçu comme un réseau d'éléments informatiques en interaction avec des entrées et des sorties physiques au lieu de dispositifs autonomes en interaction [23].

Le développement dans les sciences et l'ingénierie permettra d'améliorer le lien entre les éléments de calcul et physiques, engendrant une grande augmentation dans la capacité d'adaptation, l'autonomie, la fonctionnalité, l'efficacité, la fiabilité, la sécurité et finalement

utiliser les systèmes cyber-physiques facilement. Ceci peut améliorer les capacités des systèmes cyber-physiques dans différentes dimensions dont on peut noter [24] :

- **La réactivité** : par exemple pour l'évitement des collisions.
- **La précision** : par exemple de la chirurgie robotique
- **Travaille dans des environnements dangereux ou inaccessibles** : par exemple pour le sauvetage, la lutte contre les incendies.
- **La coordination** : par exemple pour le contrôle de la circulation aérienne dans le domaine militaire.
- **L'efficacité** : par exemple pour les bâtiments à énergie positive.
- **L'augmentation des capacités humaines** : par exemple pour les dispositifs de contrôle de santé personnels.

Les systèmes cyber-physiques sont un pilier de l'Industrie 4.0 [25].

Depuis la fin de l'année 2006, la NSF (National Science Fondation) et d'autres agences gouvernementales ont financé des ateliers sur les systèmes cyber-physiques [27] Le 1er juillet 2019 l'Union européenne a lancé le projet CPS4EU [28] qui contient les grandes entreprises (VALEO, THALES, TRUMPF, RTE, LEONARDO et SCHNEIDER ELECTRIC).

✓ **Les caractéristiques d'un CPS [29] :**

- Haut niveau d'intégration Physique/Cyber ;
- Capacités de traitement dans chaque composant physique, dû au fait que les ressources en traitement et communication sont généralement limitées ;
- Hautement connectés, via réseaux avec ou sans fil, Bluetooth, GSM, GPS etc. ;
- Adapté à des échelles temporelles et spatiales multiples ;
- Capable de reconfiguration/réorganisation dynamique ;
- Hautement automatisés, en boucles fermées ;
- Fiables, voire certifiés dans certains cas.

✓ **Les composants les plus importantes :**

Les systèmes cyber-physique ont plusieurs composants comme Électronique, informatique, métrologie, automate, data fusion et mécanique en peut résumer tout ça dans la figure suivante :

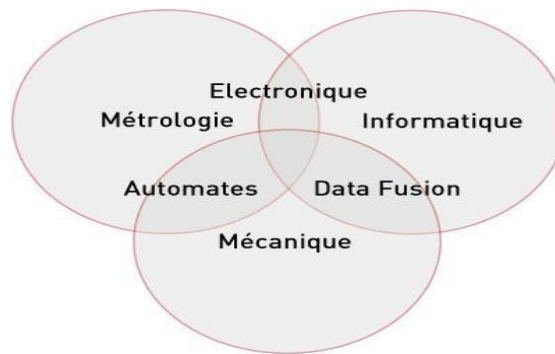


Figure II.3 - Différentes composants d'un CPS.

Chacune de ces composants est utilisée dans des cas différents :

✓ **Le domaine Électronique :**

On peut citer :

- L'intelligence interne.
- Circuits de protection (entre autres protection CEM).
- Utilisation sous conditions difficiles.

✓ **Le domaine Métrologie :**

On peut citer :

- Haute précision de signaux.
- Réduction des incertitudes de mesure.

✓ **Le domaine Informatique :**

On peut citer :

- Connexion entre le monde physique et le monde numérique.
- Interconnexion entre les systèmes et à l'extérieur.
- Intégration dans le réseau de l'entreprise et aussi technologie de l'internet.

✓ **Le domaine Automates :**

On peut citer :

- Connexion aux logiciels d'automates les plus courants.
- Transfert seule des données appréciables et Transfert de données physique.

✓ **Fusion de données :**

On cite :

- La Combinaison et le calcul de données de différentes sources.
- La Fusion de données multi-sensorielles.

✓ **Le domaine Mécanique :**

On cite :

- Boîtier robuste contre les perturbations.
- Résistance contre les chocs et l'intégration dans la machine.

II.2.c) Les Systèmes de Production Cyber-Physiques

La définition la plus communément admise est celle proposée par (Monostori, 2014). A partir de la définition de CPS, il spécifie par rapport au contexte de la production qu'un « *CPPS consiste en entités et sous-systèmes autonomes et coopérants, connectés au travers d'une relation contextualisée, au sein et au travers de tous les niveaux de la production, du process aux réseaux logistiques* ».

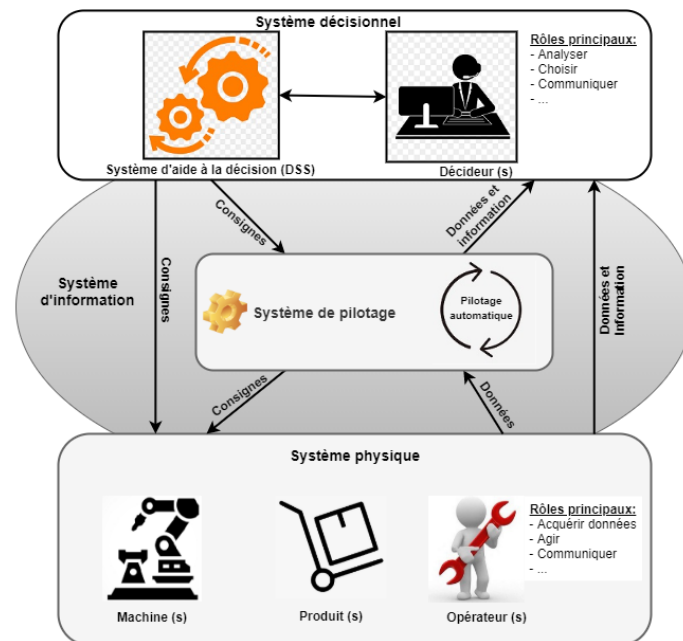


Figure II.4 - **Modèle générique de CPPS proposé [54]**

Comme l'illustre la figure II.4, notre modèle générique de CPPS (le système de production) est également caractérisé par trois sous-systèmes [54] :

- **Le système décisionnel (SD)** : il est composé de deux grandes parties qui sont :
 - Le système d'aide à la décision.
 - Le décideur (l'humain).

Il assure le pilotage du système de production (SP). Elle s'appuie sur les données et/ou informations collectées auprès de la production environnementale (technique et sociale) via des capteurs ou autres moyens d'acquisition de données et/ou d'informations. Ceux-ci n'ont pas les bases de ces types :

- Les données dites technologiques c'est-à-dire les données qui inventent des caméras, des capteurs, etc.
- Données expertes en provenance des experts humains (opérateurs / techniciens / décideurs).

Après la fusion de ces deux sources de données, des techniques de visualisation ou des algorithmes de Machine Learning, permet d'aider l'humain (le décideur) dans sa prise de décision. Son rôle est donc d'analyser les informations à sa disposition, de choisir parmi les alternatives possibles et de donner des consignes.

- **Le système d'information** : joue le rôle de passerelle entre le système de décision et le système physique. C'est à travers lui que le système de production est piloté et contrôlé. Il reçoit les données via les capteurs et les caméras installées sur les lignes de production et transmet les données et informations expertes au système de décision. D'autre part, il reçoit des consignes depuis le système de décision qu'il va répartir vers le système physique (actionneurs, automates, etc.).

- **Le système physique** : représente l'ensemble des éléments physiques du système de production (machines, produits, ...) y compris l'humain (l'opérateur). Celui-ci joue un rôle primordial dans le fonctionnement du système : il agit, il capte ou recueille des données et informations, et il en communique certaines pour le décider.

Selon (Rudtsch et al., 2014), les bénéfices pouvant être attendus de l'utilisation des CPPS sont de plusieurs ordres :

- ✓ Optimisation des processus de production, dans lesquels chaque entité autonome a des capacités cognitives, permettant de détecter l'évolution des contraintes de l'environnement et ainsi adapter sa configuration dans une zone d'application dont elle a conscience ;
- ✓ Optimisation de la personnalisation des produits par une composition intelligente de systèmes de production individuellement optimisés, sous des considérations d'objectifs tels que les propriétés des produits, le coût, la fiabilité,

- le délai, etc. ;
- ✓ Production efficiente au niveau des ressources utilisées par la minimisation des allocations de ressources imparfaites ;
- ✓ Recentrage des processus de production sur l'humain, avec des machines s'adaptant aux capacités et aux instructions de l'opérateur.

Tous ces objectifs opérationnels soulèvent de nombreuses questions et pistes de réflexion sur les prérequis à l'implémentation à large échelle des CPPS.

Outre les thématiques de technologie de réseau ou de cybersécurité, la question fondamentale est l'exploration des relations existant entre les entités, que ce soit sur des critères d'autonomie, de coopération, d'optimisation ou de réactivité (Monostori, 2014). L'utilisation d'outils de prévision du comportement émergent de l'architecture va ainsi probablement devenir de plus en plus centrale dans l'évaluation de la réponse du CPPS à un évènement inattendu (Cardin et al., 2015 ; Monostori, 2014). Une question importante est également soulevée au niveau de la gestion du Big Data (Lee et al., 2013) issu de la surveillance des procédés et de l'environnement rendue possible par la dissémination des facultés de perception dans des réseaux de capteurs, qui sont souvent la base du réseau de communication du plus bas niveau de l'architecture. Enfin, de nouveaux modes de communication entre l'humain et la machine doivent être établis (Monostori, 2014), que ce soit au niveau de la communication directe au travers d'ordres ou d'informations, ou bien la mise en forme et la présentation de données structurées d'un haut niveau afin d'augmenter la richesse et la complexité des interactions (Pirvu et al., 2015).

Les CPPS sont constitués d'éléments autonomes et coopératifs et sous-systèmes qui entrent en connexion les uns avec les autres de manières dépendantes de la situation, sur et à travers tous les niveaux de production, des processus aux machines jusqu'à réseaux de production et de logistique. La modélisation de leur fonctionnement ainsi que la prévision de leur comportement émergent, soulèvent une série des tâches de recherche fondamentale et axée sur les applications, sans oublier le contrôle de n'importe quel niveau de ces systèmes. Les questions fondamentales sont d'explorer les relations d'autonomie, de coopération, optimisation et réactivité. Intégration des analyses et on peut s'attendre à ce que les approches basées sur la simulation deviennent plus significatif que jamais. Il faut relever les défis d'exploiter des réseaux de capteurs, gérer de gros volumes de données, ainsi que les questions de recherche d'information, de représentation et d'interprétation, avec un accent particulier sur les aspects de sécurité. De nouveaux modes de communication homme-machine doivent être réalisés dans le cadre de la mise en place du CPPS.

Le CPPS rompt en partie avec la pyramide de l'automatisation traditionnelle (côté gauche de la Figure II.5). Les niveaux typiques de contrôle et de champ existent qui comprennent des automates communs proches du domaine technique processus pour être en mesure de fournir les meilleures performances pour les boucles de contrôle critiques, tandis que dans l'autre, des niveaux plus élevés de la hiérarchie et un mode de fonctionnement plus décentralisé est caractéristique du CPPS.

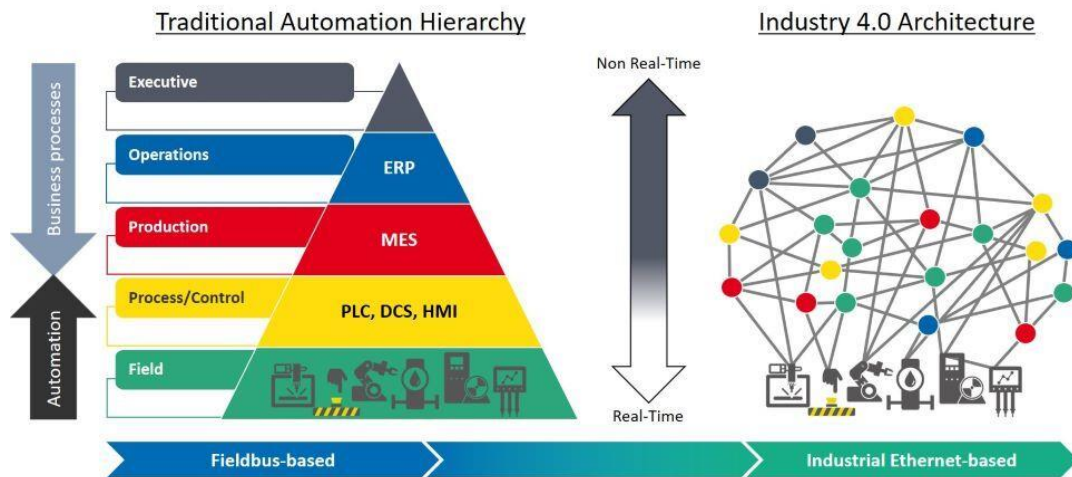


Figure II.5 - Décomposition de la hiérarchie de l'automatisation avec services distribués [30]

Le CPPS permettra et soutiendra la communication entre les humains, les machines et les produits. Les éléments d'un CPPS sont capables d'acquérir et de traiter des données, et peuvent autocontrôler certaines tâches et interagir avec les humains via des interfaces (Figure II.6).

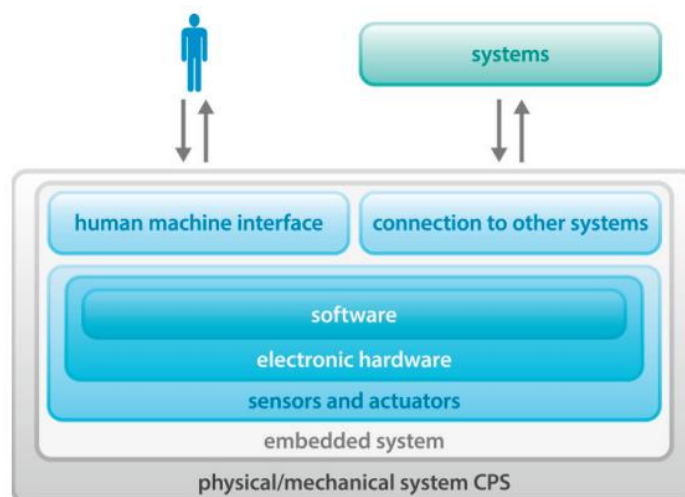


Figure II.6 - **Interaction entre humains et machines dans les systèmes cyber-physiques [31]**

▪ **Les attentes envers le CPS et le CPPS**

Les attentes sont multiples, parfois exagérées :

- ✓ Robustesse à tous les niveaux,
- ✓ Auto-organisation, auto-entretien, autoréparation, en général, auto-X,
- ✓ Sécurité,
- ✓ Télédiagnostic,
- ✓ Contrôle en temps réel,
- ✓ Navigation autonome,
- ✓ Transparence,
- ✓ Prévisibilité,
- ✓ Efficacité
- ✓ Exactitude du modèle, etc.

A travers le CPS, le développement de nouveaux business model, de nouveaux services sont attendus qui peuvent changer de nombreux aspects la vie. Les champs d'application potentiels sont nombreux : le trafic aérien et terrestre ; systèmes de production discrète et continue ; logistique ; sciences médicales, production d'énergie, etc... Par des approches cyber-physiques, ils pourraient aboutir à des villes intelligentes, production-, communication-, systèmes logistiques et énergétiques ; en outre, ils pourraient contribuer à créer une nouvelle qualité de vie. Dans ce dernier cas on peut soit parler de la société cyber-physique, qui comprend aussi bien les sphères humaines, sociales, culturelles, au-dessus des espaces physiques et cyber.

▪ **Les Racines du CPPS dans le Secteur Manufacturier**

Comme dans le cas de nombreuses révolutions, il y a des phénomènes antérieurs significatifs qui sonnent en quelque sorte dans les grands changements. Dans l'espace à venir, certains développements antérieurs dans la production seront énumérés, avec un accent particulier sur les résultats rapportés au sein de l'Académie internationale pour Production Engineering CIRP, qui peut être considéré comme racines des CPPS.

- Les systèmes de fabrication intelligents (IMS) qui devrait résoudre, dans certaines limites, sans précédent, problèmes imprévus sur la base même de documents incomplets et informations imprécises [32], [33]. Intelligence artificielle et les méthodes d'apprentissage automatique jouent ici un rôle important [34], [35].
- Systèmes de fabrication biologique (BMS) basés sur des idées d'inspiration biologique telles que l'auto-croissance, l'auto-organisation, l'adaptation et l'évolution [36].
- Systèmes de fabrication reconfigurables (RMS), où les composants sont des machines reconfigurables et contrôleurs reconfigurables [37].
- Les usines numériques (DF), la cartographie de la plupart des et les processus métier dans le monde numérique [38], [39], [40].
- Systèmes de fabrication holoniques (HMS), à base d'agents fabrication, où les deux principales caractéristiques des entités sont l'autonomie et la coopération [41], [42], [43], [44], [45].
- Méthodologies de synthèse émergentes [46].
- Réseaux de production [47].
- Structures de production évolutives [48].
- Entreprises manufacturières coopératives et réactives [49].
- Coévolution des produits, procédés et production systèmes [50].
- Gestion de la complexité dans l'ingénierie et la fabrication [50].

Toutes ces pierres angulaires abordent au moins un mais surtout plusieurs enjeux fortement liés aux enjeux de la R&D soulevées par les CPPS.

▪ **Défis R&D (La Recherche et le Développement)**

Les attentes envers le CPS et le CPPS sont polyvalents et énorme : robustesse, autonomie, auto-organisation, auto-entretien, auto-réparation, transparence, prévisibilité, efficacité, interopérabilité, suivi et traçabilité mondiale, etc... Bien qu'il y ait des développements très importants en contrôle coopératif, systèmes multi-agents (SAM), complexes systèmes adaptatifs (CAS), systèmes émergents, réseaux de capteurs, fouille de données, etc., même une réalisation partielle de ces les attentes représenteraient de vrais défis pour les chercheurs.

Dans les possibilités à venir, seuls certains des défis de R&D sont à partir de l'ensemble beaucoup plus vaste de domaines de recherche qui sont liés au CPPS :

- *Adapté au contexte et (au moins partiellement) autonome systèmes.* Méthodes pour un contexte complet et continu sensibilisation, pour la reconnaissance, l'analyse et

l'interprétation des plans et intentions des objets, systèmes et participants utilisateurs, pour la création de modèles pour le domaine et le domaine d'application et pour la conscience de soi en termes de connaissance de soi-même la situation, le statut et les options d'action doivent être développés.

- *Systèmes coopératifs de production.* De nouveaux résultats théoriques doivent être atteints et le développement d'algorithmes pour la recherche de consensus, l'apprentissage coopératif et une détection distribuée est nécessaire.
- *Identification et prédiction de systèmes dynamiques.* L'extension de l'identification et de la prédiction disponibles méthodes est nécessaire, ainsi que le développement de nouvelles qui peut être appliqué sous des hypothèses modérées sur le système dynamique, ainsi que le processus de perturbation.
- *Ordonnancement robuste.* De nouveaux résultats doivent être obtenus dans gérer les perturbations de la production dans le cadre du planning exécution.
- *Fusion de systèmes réels et virtuels.* Le développement de de nouvelles structures et méthodes sont nécessaires pour soutenir la fusion des sous-systèmes virtuels et réels afin d'atteindre un système de production intelligent et robuste dans un environnement changeant et incertain. Nouvelle référence architectures et modèles de virtuel et réel intégrés sous-systèmes de production ; la synchronisation du virtuel et les modules réels des systèmes de production et leur interaction spécifique à leur rôle ; et adaptative au contexte, ressource des algorithmes de contrôle d'atelier efficaces sont nécessaires.
- *Symbiose homme-machine (y compris homme-robot).* Le développement d'un cadre de données géométriques à fusion caractéristiques d'assemblage et mesures des capteurs et recherche rapide algorithmes pour adapter et compenser les changements dynamiques l'environnement réel est nécessaire.

Le développement parallèle de l'informatique et les technologies de l'information et de la communication sur un d'une part, et de fabrication d'autre part, a été mis en évidence, soulignant leur influence mutuelle. Le concept de systèmes de production cyber-physiques a été introduit en bref, avec de grandes attentes à leur égard. Les origines des CPPS ont également été dénombrées, principalement sur la base des contributions des membres de l'Académie internationale de la production Ingénierie (CIRP). Quelques-uns des nombreux défis de R&D dans la réalisation du CPPS ont également été soulignés. Quant au plus concret et développements récents, nous renvoyons à la littérature [51]-[52].

Sans aucun doute, le CPPS peut être considéré comme une étape importante dans le développement des systèmes de fabrication. Que cette étape serait considérée comme la

quatrième étape industrielle sera décidée par les générations à venir, mais certainement, cela se produira assurément.

II.3) Le Commissioning Virtuel

II.3.a) Introduction

La nouvelle industrie 4.0 bouscule les industries, étant cette nouvelle révolution l'union des technologies Internet et des technologies tournées vers l'avenir dans le domaine des objets « Smart » (Lasi et al., 2014). Le Commissioning Virtuel (VC) est l'une des avancées de cette nouvelle évolution de la production industrielle. Il devient un outil révolutionnaire. Même s'il a commencé lors de la dernière troisième révolution industrielle, il se trouve maintenant dans la quatrième révolution industrielle lorsqu'il est au centre des nouvelles technologies. VC est un moyen de vérification et de validation fiable qui s'est avéré être un outil formidable et adapté aux étapes de programmation, de débogage, de résolution de problèmes et de formation des travailleurs. Il est basé sur le développement d'un modèle de système et sa connexion aux données de l'usine réelle (VC en ligne) ou l'application des résultats et des données du modèle virtuel à la future usine réelle (VC hors ligne).

De cette façon, le Contrôleur Logique Programmable (PLC) et les codes de programmation du robot peuvent être testés, de nouvelles dispositions peuvent être conçues et d'autres modifications peuvent être apportées et testées avant la construction de la cellule physique réelle. Shahim & Moller (2016) concluent que l'utilisation du VC donne une modélisation et une conception de services complexes plus efficaces. Ils déclarent également que beaucoup moins d'interruptions sont subies par VC. Enfin, avec l'utilisation du VC, il y a l'avantage de réduire le délai d'exécution, car le modèle a été optimisé et développé dans sa disposition optimale. Le temps de changement, le temps d'attente, entre autres, sont réduits au maximum. Tout cela générera plus d'avantages tels que des économies de coûts et de matériaux.

Un autre progrès essentiel de l'industrie 4.0 est la Réalité Virtuelle (VR). Le Commissioning Virtuel (VC) est un pas en avant significatif dans l'industrie, mais il peut être poussé encore plus loin avec VR. La VR était, au début, utilisée à des fins récréatives. Cependant, après l'évolution des technologies VR et la diminution des coûts, la VR est entrée dans le monde de l'industrie manufacturière (Pérez et al., 2020). Allant plus loin, transformer la maquette virtuelle du système qui sera mis en service en un environnement virtuel immersif rend l'expérience du VC plus complète. Il entraîne la personne vers une recréation de la réalité pour tester la cellule et ses risques et erreurs avant la mise en œuvre physique.

Au fil du temps, la technologie évolue rapidement et, comme le monde l'a montré au cours des années 2020, la façon dont les choses sont faites peut changer radicalement en quelques

mois, voire moins. C'est une réalité que l'une des principales tendances technologiques est l'objectif de numériser tous les aspects de la vie. Aujourd'hui, la capacité de travailler, de former, de concevoir, de planifier, de valider et de contrôler sans être sur place est souhaitée. Le VR est un outil approprié pour cette tâche.

II.3.b) Définition du Commissioning Virtuel

Le commissioning signifie la conception de l'installation ou le test de l'ensemble du système et de son logiciel de contrôle. Ceci ne peut être effectué avant d'avoir terminé la partie importante du processus d'ingénierie, de l'approvisionnement et de l'assemblage. Le fait que le test intégré d'un système ne puisse être effectué avant qu'il ne soit déjà construit, pose par conséquent de nombreux problèmes et défauts de conception et de codage qui restent non détectés avant le premier démarrage du système. Cela conduit à l'une des tâches très chronophages comme la reconception du système et des logiciels (Hoffmann et al., 2010).

Dans cette optique, le VC est donc essentiel à la simulation et l'émulation du fonctionnement global de l'usine. Cette technologie peut être particulièrement bénéfique pour les fabricants dont les systèmes de production présentent un niveau élevé de complexité d'automatisation.

"VC consiste en la création d'un modèle de simulation du système de fabrication qui est utilisé pour permettre la mise en service par simulation, avant de construire le système, afin qu'il soit possible de détecter et de corriger tôt les erreurs générées lors de la planification" (Hoffmann et al., 2010).

Le commissioning traditionnel implique directement un système de fabrication réel (par exemple une machine), un contrôleur réel (par exemple un API ou d'autres méthodes de contrôle) et d'autres dispositifs nécessaires tels que des capteurs et des actionneurs (Jackson, 2019). C'est sur ces éléments physiques que s'effectue le test intégral, qui implique de nombreuses exigences liées aux installations, aux opérateurs et aux équipements. D'autre part, VC utilise des répliques virtuelles des composants du système, et ces exigences qui prennent du temps et de l'argent peuvent être réduites.

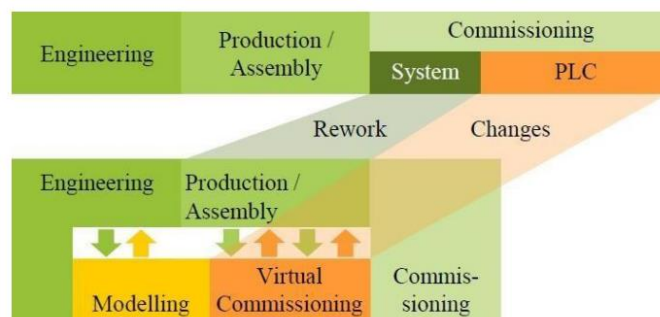


Figure II.7 - L'Ingénierie avec et sans VC (Lechler et al., 2019)

La Figure.II.7 montre la différence entre les processus d'ingénierie utilisant VC et ne l'utilisant pas. Le principal avantage est que le VC peut être réalisé en parallèle avec le processus de production et d'assemblage, permettant des changements dans l'automate et le système, le tout avant la mise en service finale. Cela permet de résoudre les erreurs avant la mise en service physique (Lechler et al., 2019).

"Au final, le but final du commissioning virtuel est d'arriver à un modèle virtuel, aussi appelé modèle de comportement, qui avec le code du système de contrôle, fournit un comportement comme celui de la machine réelle" (Lechler et al., 2019).

Pendant le processus de commissioning, nous pouvons soit utiliser le matériel, soit la simulation du système testé. Nous pourrions y associer ultérieurement soit le contrôleur matériel, soit le contrôleur virtuel. Chaque combinaison de ces éléments se traduit par une configuration de commissioning différente (Lee et Park, 2014) (Bendjelloul et al, 2022). Le test d'une représentation virtuelle du système de fabrication avec un contrôleur virtuel ou un contrôleur réel est considéré comme une configuration de mise en service virtuelle. Dans le test Software-In-the-Loop (SIL), la logique de commande est programmée selon le langage PLC standard IEC61131-3 ; cependant il est déployé sur un automate virtuel. Si le contrôleur utilisé dans les tests est un automate réel, il s'agit d'une commissioning virtuelle Hardware-In-the-Loop (HIL) (Reinhart et Wunsch, 2007) (Mayer et al., 2019).

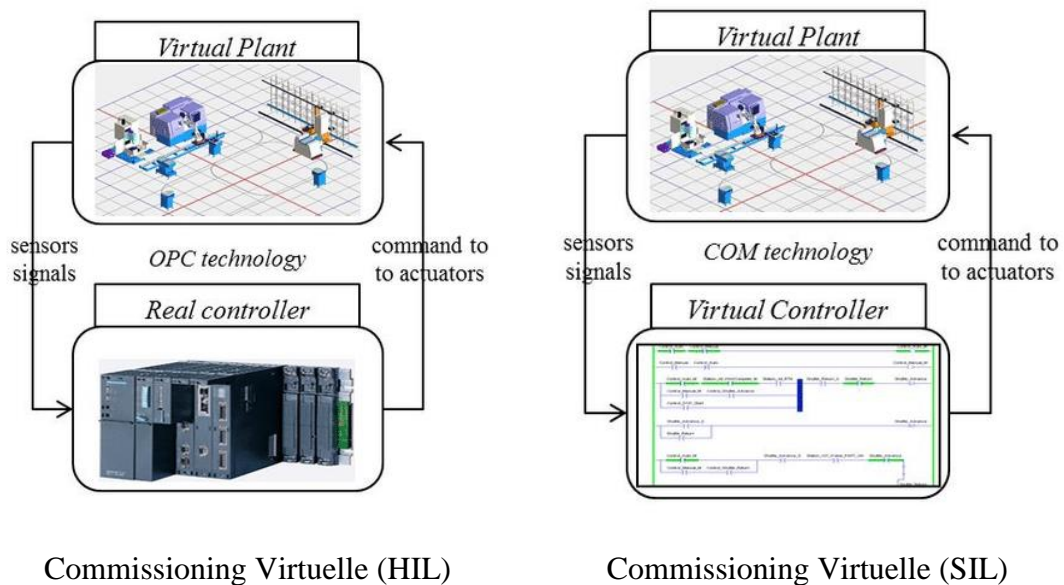


Figure II.8 - Les différents types du VC (SIL, HIL)[55]

Parmi les travaux de recherche qui ont démontré l'utilité du commissioning virtuel en production. On a sélectionné quelques-uns comme exemples de cas d'étude.

- **Cas d'étude 1 : Le commissioning virtuel d'une cellule**

Les résultats de Makris et al. (2012a), même s'il ne s'agit pas d'une des dernières revues, corroborent que VC est un moyen fiable de vérifier et de valider les opérations d'une cellule avant l'installation physique.

Les principaux avantages qu'ils décrivent sont les suivants :

1. L'augmentation de la réduction du temps d'installation de 15 à 25 %, dont 90 % correspond au temps de développement de la programmation du robot et du logiciel associé.
2. Les coûts d'investissement sont réduits au minimum, diminuant les ressources humaines nécessaires pour résoudre les problèmes pendant la phase de montée en puissance.
3. La possibilité ouverte de reconfigurer la cellule puisqu'elle est conçue dans un environnement virtuel. Il est possible de reconfigurer l'agencement et les équipements si, après avoir testé le modèle virtuel, une meilleure solution est découverte pour la cellule. Dans le même temps, il est validé la faisabilité du projet.

- **Cas d'étude 2 : Le commissioning virtuel d'une cellule robotisée**

Fernandez et al. (2019), dans une étude de cas pédagogique, prouve que les étudiants qui ont utilisé VC ont noté ses vertus. Outre la validation du logiciel, ils remarquent l'applicabilité des outils d'émulation du robot. Les étudiants ont constaté une amélioration portant sur : La compréhension du processus ; le programme PLC, qui est testé avec plus de succès ; une utilisation plus efficace de l'équipement physique après la validation dans le modèle virtuel ; un temps de mise en service réduit. Dans les programmes où un DT est utilisé pour la validation, la mise en service nécessite moins de temps et est plus facile (Fernández et al., 2019).

Une autre étude de cas qui reflète les avantages de l'utilisation du VC dans la fabrication est la recherche d'Eguti & Trabasso (2018). Là-dessus, les auteurs réfléchissent, encore une fois, que travailler avec VC, il y a un gain de temps, en parlant du processus de conception et des étapes de programmation et enfin de validation. La principale raison du gain de temps dans cette étude est que le processus peut être testé en parallèle avec la phase de conception.

Il existe d'autres études de cas où VC s'est avéré être un excellent outil, comme dans les travaux de Makris et al. (2012b), et tous conduisent à la conclusion que le processus de fabrication, qu'il soit automatique ou non, s'améliore significativement avec le VC.

- **Cas d'étude 3 : Le commissioning virtuel et réalité virtuelle**

L'introduction de la VR dans VC peut être considérée comme la prochaine étape pour VC car elle donne une visualisation plus réaliste d'un environnement virtuel et étend ses possibilités de validation. Les nouvelles possibilités de validation sont dues à l'interaction avec la production virtuelle en cours. Les concepteurs, entre autres, peuvent changer leur regard sur l'environnement d'observateurs à acteurs. (Dahl et al., 2017).

Dahl et al. (2017) plongent dans la façon dont la VR est une extension naturelle du VC, étant capable de faire l'expérience d'un système de production du monde réel qui n'a pas encore été créé. Certaines des opportunités que la réalité virtuelle offre pour la validation et la formation des utilisateurs sont le test de la sécurité et de l'équipement, la formation des opérateurs avec des IHM virtuelles ou l'inspection des pièces, et enfin, une évaluation de l'ergonomie.

- **Jumeau numérique et réalité virtuelle dans Le commissioning virtuel**

Comme Pérez et al. (2020) remarquent dans leur article que les DT sont parfaits pour les cellules robotisées. Ils peuvent être utilisés pour la prévision et la prévention des risques. Un DT permet une immersion encore meilleure dans l'environnement virtuel comme s'il s'agissait du réel, et il permet de trouver une solution optimale lors du VC. Le DT permet une surveillance en temps réel une fois que la cellule est implémentée dans l'environnement réel, en miroir avec son DT. Une bien meilleure pratique consiste à valider les DT dans le VC avant la mise en œuvre physique (Pérez et al., 2020)

- **Le commissioning virtuel des systèmes cyber-physiques de production**

Malgré l'augmentation de la popularité du VC de nos jours, la montée de la révolution de l'industrie 4.0 et des systèmes de production cyber-physiques a conduit à la décentralisation de nombreuses fonctions au niveau du terrain et aux composants d'usine intelligents, dévoilant un défi supplémentaire de validation et de test (Glock et al., 2018).

Le travail présenté dans ce mémoire de fin d'étude s'inscrit dans les travaux menés par le laboratoire des systèmes robotisés de production, au niveau de la division de productique et de robotique du Centre de Développement des Technologies Avancées.

II.4) Conclusion

Dans ce chapitre, découvrez CPS et CPPS et les avantages significatifs qu'ils offrent dans divers domaines, notamment l'industrie, les transports, les soins de santé et l'énergie. Il permet une meilleure gestion des processus physiques, une prise de décision plus intelligente et une amélioration globale de l'efficacité et de la qualité. Le commissioning virtuel est également mis en évidence, qui est une approche clé pour garantir le bon fonctionnement des CPS et des CPPS, en minimisant les risques et en maximisant l'efficacité dès le début du déploiement.

Ensuite, dans l'étude de cas du chapitre suivant « Robot cartésien 3 axes », nous nous concentrons sur les robots en cas général et leurs domaines d'utilisation avec différentes caractéristiques. Comme le robot pour notre étude avec la marque « Yamaha ». Nous ajoutons les différents digrammes.

CHAPITRE III :

ETUDE ET

DEVELOPPEMENT

DU CPPM

III. Chapitre III : Etude et Développement du Module de production Cyber-physique

III.1) Introduction

Dans ce chapitre une étude est présentée sur l'espace de travail des robots. Nous considérons notre Robot Cartésien en incluant l'étude de son espace de travail et les aspects qui lui sont liés. Ces informations sont essentielles pour la conception, la planification et le fonctionnement efficace d'un robot dans son environnement de travail. De plus, nous examinons l'automatisation et le module de production cyber-physique. Enfin, nous discutons de l'Implémentation qui contient AAS et le développement de l'AAS, Architecteur Organique et Analyse fonctionnelle.

III.2) Cas d'étude : Robot Cartésien à 3 axes

Un robot est une machine capable d'effectuer des tâches et de manipuler des objets selon un programme de façon automatique. Ils sont généralement utilisés pour remplacer les humains dans de situations où ces derniers sont incapables d'effectuer le travail, des situations plus dangereuses, de haute précision ou répétitive.

Un robot est un système mécanique poly-articulé, mû par des actionneurs et commandé par un ordinateur, qui est destiné à effectuer une grande variété de tâches [65]. Un robot est un dispositif mécatronique (alliant mécanique, électronique et informatique) accomplissant automatiquement des tâches diverses. C'est une machine intelligente fonctionnelle qui nécessite une autonomie de mouvements.

La robotique est la science et technique de la robotisation, de la conception et de la construction des robots. Il existe différents types de robots : robots industriels, robot logistiques, robots agricoles, robots militaires, robots de santé, robots ménagers, robots jouets, robots compagnons. Selon la définition normalisée ISO-8373 2012, Il existe aujourd'hui deux grands domaines dans la robotique : les robots industriels et les robots de services.

La robotique est l'ensemble des techniques permettant la conception et la réalisation de machines automatiques ou de robots.

L'ATILF donne la définition suivante du robot : « il effectue, grâce à un système de commande automatique à base de microprocesseur, une tâche précise pour laquelle il a été conçu dans le domaine industriel, scientifique, militaire ou domestique » [66].

De cette définition découlent deux interprétations : la première serait de voir le robot comme une machine, qui possède des capteurs, un système logique et des actionneurs. Il est matériel. La deuxième laisse penser qu'un robot peut aussi être virtuel [67].

La robotique actuelle trouve des applications dans différents domaines (liste non exhaustive) [67] :

- La robotique industrielle,
- La robotique domestique,
- La robotique médicale,
- La robotique militaire,
- La robotique sociale,
- La robotique scientifique, par exemple pour l'exploration de l'espace, (aérobot), des fonds marins (robots sous-marins autonomes), dans les laboratoires d'analyse (robotique de laboratoire), etc., ou encore
- La robotique de transport (de personnes et de marchandises), avec par exemple ROPITS (Robot for Personal Intelligent Transport System), Robosoft, RoboCourier, etc.

III.2.a) **Historique :**

L'histoire de la robotique s'inscrit dans celle, plus vaste, du développement et de la prolifération des machines, qui débute au XVIII^e siècle avec la Révolution industrielle. Elle commence donc avec l'automate. La différence fondamentale entre l'automate et ce que deviendra le robot au XX^e siècle tient à ce que le premier obéit à un programme strictement préétabli, que ce soit de manière mécanique ou électrique, alors que le second dispose de capteurs et d'appareillages électroniques, de sorte que ses actions découlent de ses contacts avec son environnement, ce qui - à la différence de l'automate - le rend autonome, « intelligent ». Le terme robot vient du Tchèque ; il signifie travail forcé. Le mot sous son acception moderne fut introduit par l'auteur tchèque Karel Capek dans son ouvrage R.U.R .qui date de 1921 en anglais Rossum's Universal Robots.

Le concept du Robot industriel fut breveté en 1954, par G.C. Devol (Brevet U.S. N°2988237) ; ce brevet décrit la réalisation d'un bras mécanique asservi, capable d'effectuer des tâches de caractère industriel.

- 1947 : Premier manipulateur électrique télé opéré.
- 1954 : Premier robot programmable.

- 1961 : Utilisation d'un robot industriel, commercialisé par la société Unimation (USA), sur une chaîne de montage de General Motors.
- 1961 : Premier robot avec contrôle en effort.
- 1963 : Utilisation de la vision pour commander un robot.

Les années subséquentes ont vu des améliorations de tout genre et des robots de plus en plus rapides, précis et flexibles. En 1998, par exemple, les systèmes de contrôle pouvaient gérer jusqu'à 27 axes et synchroniser le contrôle de 4 robots [69].

III.2.b) Caractéristiques d'un robot

Un robot doit être choisi en fonction de l'application qu'on lui réserve. Voici quelques paramètres à prendre, éventuellement, en compte :

- La charge maximum transportable (de quelques kilos à quelques tonnes), à déterminer dans les conditions les plus défavorables.
- Le volume de travail, défini comme l'ensemble des points atteignables par l'organe terminal. Tous les mouvements ne sont pas possibles en tout point du volume de travail. L'espace de travail (*reachable workspace*), également appelé espace de travail maximal, est le volume de l'espace que le robot peut atteindre *via* au moins une orientation. L'espace de travail dextre (*dextrous workspace*) est le volume de l'espace que le robot peut atteindre avec toutes les orientations possibles de l'effecteur (organe terminal). Cet espace de travail est un sous-ensemble de l'espace de travail maximal.
- Le positionnement absolu, correspondant à l'erreur entre un point souhaité (réel) – défini par une position et une orientation dans l'espace cartésien – et le point atteint et calculé *via* le modèle géométrique inverse du robot. Cette erreur est due au modèle utilisé, à la quantification de la mesure de position, à la flexibilité du système mécanique. En général, l'erreur de positionnement absolu, également appelée précision, est de l'ordre de 1 *mm*.
- La répétabilité, ce paramètre caractérise la capacité que le robot a à retourner vers un point (position, orientation) donné. La répétabilité correspond à l'erreur maximum de positionnement sur un point prédéfini dans le cas de trajectoires répétitives. En général, la répétabilité est de l'ordre de 0,1 *mm*.
- La vitesse de déplacement (vitesse maximum en élongation maximum), accélération.
- La masse du robot.
- Le coût du robot.
- La maintenance, ...

III.2.c) Définition d'un robot industriel

Les robots industriels sont des techniciens polyvalents, multifonctions, reprogrammables sur 2 axes ou plus. Les robots industriels sont conçus pour déplacer des matériaux, des pièces ou des outils par des mouvements variables et programmables afin d'accomplir diverses tâches. Grâce à leur flexibilité, les robots industriels sont utilisés dans tous les secteurs industriels. Le principal critère de sélection d'un robot industriel est le domaine d'application.

Les robots manipulateurs industriels, chargés de manipuler, soient :

- ✓ Des pièces :
 - ✓ Stockage – déstockage,
 - ✓ Palettisation – dé palettisation,
 - ✓ Chargement – déchargement de machine-outil,
 - ✓ Manipulation d'éprouvettes,
 - ✓ Assemblage de pièces, ...

- ✓ Des outils :
 - ✓ Soudure en continu ou par points,
 - ✓ Peinture,
 - ✓ Collage,
 - ✓ Ebavurage, ...

Selon la tâche ultime du robot, il existe quatre principaux types de robots parmi lesquels choisir :

- ✓ Robots SCARA.
- ✓ Robots Articulé.
- ✓ Robots Parallèle.
- ✓ Robots Cartésien.

Chacun de ces robots à ses propres avantages et contraintes tels que le nombre d'axes, la charge maximale (appelée charge utile) et la portée.

Chacun de ces critères vous permettra d'affiner le choix.

▪ Les robots SCARA

Le robot SCARA (Selective Complaint Articulated Robot for Assembly) travaille le long d'axes verticaux en tournant sur le même plan. Ils sont plus rapides et plus flexibles que les robots cartésiens.

Ils se distinguent notamment par leur capacité à effectuer des opérations d'insertion verticale en utilisant uniquement le mouvement de l'axe Z.

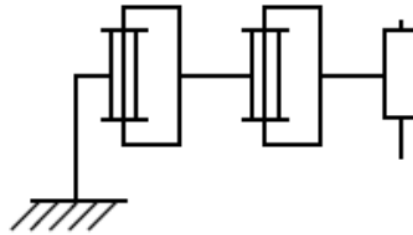


Figure III.1 - Les robots SCARA

La configuration de ce robot permet des structures simples et peu coûteuses. La plupart des robots SCARA sont fabriqués avec seulement quatre degrés de liberté et sont utilisés pour des opérations d'assemblage simples qui ne nécessitent pas une capacité d'orientation complète des pièces. Pour ces tâches, les robots SCARA seront plus précis (répétabilité des tâches) et moins coûteux que les robots articulés.

Cependant, si vous avez besoin de six degrés de liberté pour l'orientation complète de la pièce, un robot articulé serait une option plus pratique car il offrira une plus grande flexibilité opérationnelle.

▪ Le bras robotique articulé

Est construit sur le principe d'un bras humain. Il est constitué d'une série d'articulations, chaque articulation lui conférant un degré de liberté (Jusqu'à 6 DDL, ou plus en cas de redondance). Cette structure très souple lui permet d'atteindre n'importe quelle position et orientation dans son enveloppe de travail, malgré les obstacles.

Cette structure de robot est utilisée pour une large gamme d'applications :

- ✓ Soudage à l'arc
- ✓ Soudage par points
- ✓ Manutention
- ✓ Alimentation/Chargement de la machine

Ils sont cependant plus difficiles à programmer que les autres robots et sont également plus chers. Ils sont de plus en plus déclinés en version collaborative (cobot).

Ce type de manipulateur est connu sous le nom de manipulateur coudé. La structure et la terminologie associées au manipulateur coudé sont illustrées à la Figure III.2. Son espace de travail est illustré à la Figure III.3 [68].

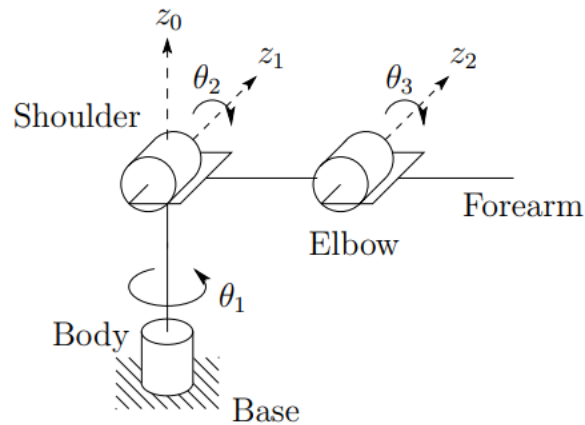


Figure III.2 - Structure du manipulateur de coude.

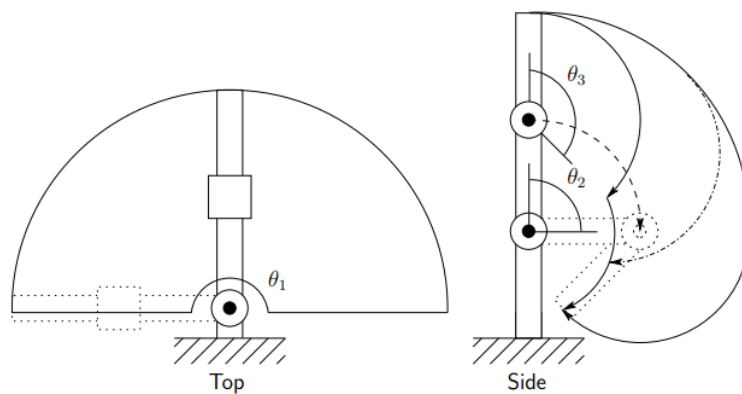


Figure III.3 - Espace de travail du manipulateur de coude.

Le manipulateur tournant offre une liberté de mouvement relativement grande dans un espace compact. La liaison parallélogramme, bien que généralement moins habile que le manipulateur coudé, présente néanmoins plusieurs avantages qui en font une conception attrayante et populaire. La caractéristique la plus notable du manipulateur à liaison parallélogramme est que l'actionneur de l'articulation est situé sur la liaison. Étant donné que le poids du moteur est supporté par la liaison, les liaisons et peuvent être rendues plus légères et les moteurs eux-mêmes peuvent être moins lourds. Puissant. De plus, la dynamique du

manipulateur à parallélogramme est plus simple que celle du manipulateur à coude, ce qui le rend plus facile à contrôler [68].

▪ Les Robots Parallèles

Ils sont construits à partir de parallélogrammes articulés reliés à une base commune.

Dans la catégorie des robots parallèles, vous trouverez principalement des robots Delta, également appelés robots Spider. Cette configuration de robot est capable de mouvements délicats et précis.

Ils offrent des mouvements très dynamiques et sont la solution idéale pour les tâches d'assemblage et de conditionnement à faible charge utile, ou pour les opérations de pick and place sur des objets légers (de 10 g à 1 Kg). Ces robots industriels sont fréquemment utilisés dans l'industrie alimentaire ainsi que dans les industries pharmaceutiques et électroniques.

Un manipulateur parallèle est un manipulateur dans lequel un sous-ensemble de maillons forme une chaîne fermée. Plus spécifiquement, un manipulateur parallèle a deux ou plusieurs chaînes cinématiques indépendantes reliant la base à l'effecteur terminal. La Figure III.4 montre l'Architecture robotique dont le robot hybride à chaîne cinématique parallèle [68].

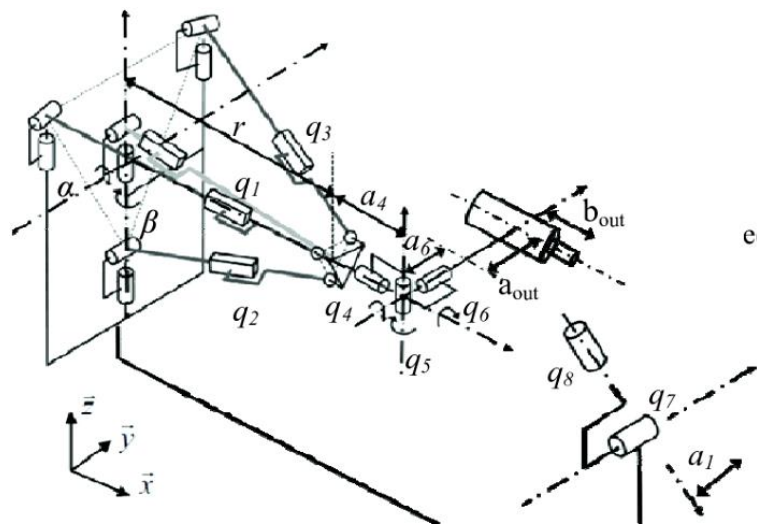


Figure III.4 - Le robot manipulateur parallèle.

La cinématique à chaîne fermée des robots parallèles peut entraîner une plus grande rigidité structurelle, et donc une plus grande précision, que les robots à chaîne ouverte. La description cinématique des robots parallèles est fondamentalement différente de celle des robots à liaison série et nécessite donc des méthodes d'analyse différentes.

▪ **Les robots cartésiens :**

Ne peuvent se déplacer que le long **d'axes linéaires**, ce qui les rend très rentables et **faciles** à programmer. Ils sont parfaitement adaptés aux tâches répétitives, telles que le déchargement des machines.

Leur principal avantage est qu'ils permettent une grande précision de positionnement dans toute la zone de travail. Par exemple, la plupart des machines à mesurer tridimensionnelles sont fabriquées avec cette structure. De plus, la modularité de la structure permet de réaliser de gros robots pouvant, entre autres, contrôler toutes les machines d'un îlot de production.

Quelques exemples d'applications incluent l'utilisation de robots cartésiens pour des opérations d'assemblage qui nécessitent un positionnement très précis, pour le contrôle de machines-outils (chargement/déchargement de pièces et changement d'outils) et des opérations de pick-and-place pour des presses à injecter.

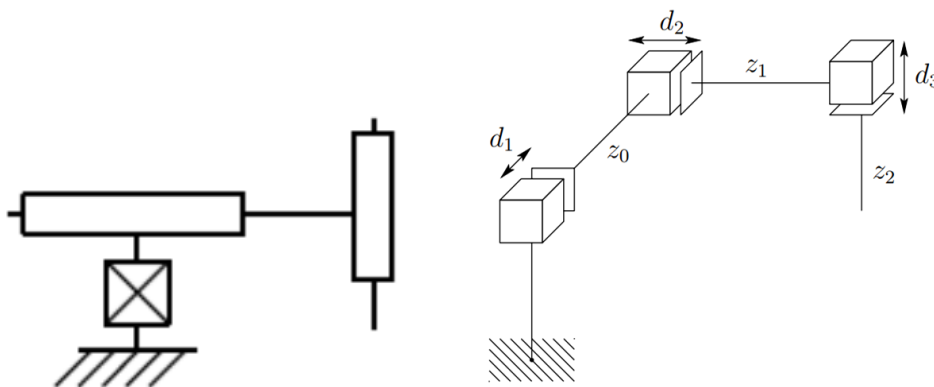
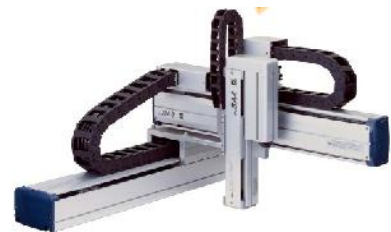


Figure III.5 - **Les robots cartésiens.**

Dans le cas des charges plus lourdes que la moyenne, on doit envisager des robots portiques cartésiens. Ce type de robot est plus grand et a une charge utile et une portée plus élevée.

- **Les types des robots cartésiens [70] :**

- ✓ **Robot linéaire :** est un robot industriel dont les trois principaux axes de contrôle sont linéaires (c'est-à-dire qu'ils se déplacent en ligne droite plutôt qu'en rotation) et sont à angle droit les uns par rapport aux autres. Les trois articulations coulissantes correspondant au déplacement le

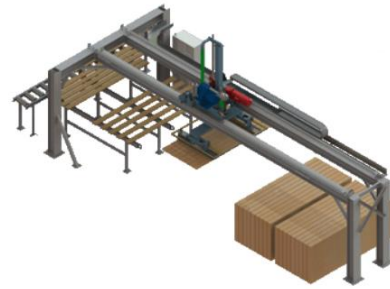


poignet de haut en bas, dedans-dehors, d'avant en arrière. Entre autres avantages, cette disposition mécanique simplifie la solution du bras de commande du robot.

- ✓ 3 axes linéaires de liberté.
- ✓ Orienté perpendiculairement.

En raison de leur structure rigide, ce type de robots peut généralement offrir de bons niveaux de précision et de répétabilité.

- ✓ **Robots portiques** : Les robots à coordonnées cartésiennes avec l'élément horizontal soutenu aux deux extrémités. Ils sont souvent assez grands. À structure du portique comprend une série de poutres ou d'entretoises qui fournissent au robot une stabilité et une précision pour se déplacer le long des axes X, Y et Z. et situés sur



les plus grandes zones de déplacement. Les robots peuvent être configurés pour fonctionner dans divers environnements, y compris les salles blanches, les environnements dangereux et d'autres applications spéciales.

- ✓ 3 axes linéaires de liberté.
- ✓ Orienté perpendiculairement.

L'enveloppe de travail est similaire avec d'autres robots de type cartésien, cependant, un robot portique enferme généralement son enveloppe de travail de l'extérieur.

- ✓ **Machine CNC** : Une application populaire pour ce type de robot est une machine à commande numérique par ordinateur. L'application la plus simple est utilisée dans les machines de fraisage et de dessin où un stylo ou un routeur se déplace sur un plan XY tandis qu'un outil est soulevé et abaissé sur une surface pour créer une conception précise.

✓ **Avantages :**

- ✓ Haute précision et vitesse ;
- ✓ Moins de coûts ;
- ✓ Procédures d'exploitation simples ;
- ✓ Charges utiles élevées ;
- ✓ Travail très polyvalent ;
- ✓ Simplifie les systèmes de contrôle du robot et du maître.

✓ Inconvénients :

Ils nécessitent un grand volume d'espace pour fonctionner

Dans le cas de notre étude des robots cartésiens, nous nous sommes référés à une étude sur une grande marque « YAMAHA » de Mode « SXYx-C»

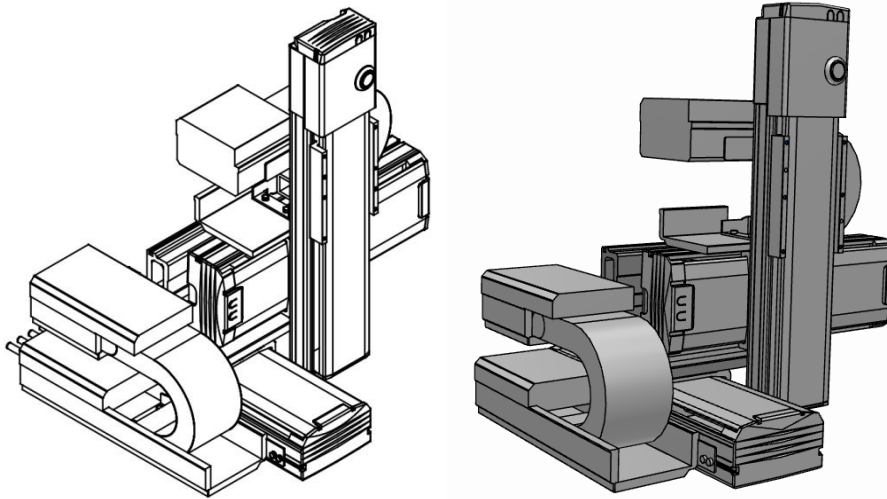


Figure III.6 - Yamaha SXYx-C-A1-15-15-ZFL20-15

Caractéristiques de Robots :

- MODEL (Model - Cable) : SXYx-C
- COMB (Combination) : A1
- Répétabilité (mm) : +/- 0.01
- Vitesse maximum (mm/sec) : 1200
- Charge utile maximale (Kg) : 8
- Système de conduite : Vis à billes $\Phi 15$
- Pas de vis à billes (Rapport de décélération) (mm) : 20
- Plage de déplacement (mm) :
 - X_STROKE (X-axis Stroke / mm) 150
 - Y_STROKE (Y-axis Stroke / mm) 150
 - Z_STROKE (Z-axis Stroke / mm) 150

- Effecteur Robot

L'effecteur robot s'agit d'une pince électrique conçue pour des tâches de manipulation précise et délicate. Sa polyvalence lui permet d'être utilisé dans divers domaines tels que l'assemblage électronique, la manipulation d'objets fragiles dans l'industrie pharmaceutique et l'emballage alimentaire.



Grâce à son contrôle précis, sa conception compacte et ses fonctionnalités de sécurité intégrées, il est un outil performant pour les manipulations qui requièrent une grande précision et une interaction en douceur avec les objets [78].

III.3) L'automatisme

L'automatisme est la discipline traitant d'une part la caractérisation des systèmes automatisés et d'autre part le choix de la conception et de la réalisation de la partie commande. Il s'agit donc d'étudier les systèmes en :

- Réalisant leurs fonctions en relative autonomie.
- Assurant un contrôle des performances par la mise en place possible d'une chaîne de retour.

L'automatisation de la production consiste à transférer toutes ou une partie des tâches de coordination, auparavant exécutées par l'humain.

III.3.a) Automate Programmable Industriel

Les automates programmables industriels peuvent être défini comme une forme spéciale de microprocesseur qui utilise une mémoire programmable pour stocker des instructions et mettre en œuvre des fonctions telles que la logique, le séquençement, la temporisation, le comptage et résoudre les opérations arithmétiques aussi que la fonction de communication afin de contrôler des machines et des processus industrielles [69].

Ils sont utilisés pour accomplir diverses tâches au sein des processus industriels, notamment la surveillance des capteurs, le contrôle des actionneurs, la collecte de données, le traitement des signaux, la gestion des alarmes et la coordination des opérations dans des systèmes complexes. Ils permettent de mettre en place des logiques de contrôle, des séquences d'opérations, des temporisations, des inter-verrouillages et d'autres fonctions essentielles à l'automatisation industrielle.

Ces automates sont programmés à l'aide de langages spécifiques tels que le langage ladder, le langage de liste d'instructions (IL), le langage de blocs fonctionnels (FBD) ou le langage structuré à contact (ST). Ces langages permettent de définir la logique de contrôle et les séquences d'opérations de l'automate, offrant ainsi une flexibilité dans la programmation et l'adaptation aux besoins spécifiques de l'automatisation industrielle.

III.4) Module de production cyber-physique

III.4.a) Généralités

Aujourd'hui, grâce à CPS, on constate une progression. Les niveaux de contrôle et de terrain typiques sont toujours présents, comprenant des automates programmables industriels communs proches des processus techniques pour assurer des performances optimales dans les boucles de contrôle critiques. D'autre part, les niveaux supérieurs de la hiérarchie se caractérisent par un fonctionnement plus décentralisé, propre au CPPS [71].

En général, on suppose qu'un CPPS est composé de deux principaux éléments fonctionnels. Le niveau inférieur est responsable de la connectivité avancée, permettant l'acquisition en temps réel de données provenant du monde physique ainsi que le retour d'informations provenant du cyberspace. Le niveau supérieur intègre quant à lui la gestion intelligente des données, l'analyse et les capacités informatiques pour construire le cyberspace [71].

III.4.b) Les exigences des systèmes cyber-physique de production

Les systèmes de production cyber-physiques (CPPS) sont des systèmes complexes qui intègrent des composants physiques et numériques pour créer un environnement de fabrication plus efficace et réactif. Les exigences d'un CPPS varient en fonction de l'application spécifique, mais certaines des exigences les plus courantes sont les suivantes :

- **Scalabilité** : les CPPS doivent être évolutifs pour s'adapter aux évolutions des besoins de production. Cela nécessite l'utilisation de systèmes modulaires et flexibles pouvant être facilement étendus ou réduits.
- **Interopérabilité** : les CPPS doivent pouvoir interagir avec d'autres systèmes, tels que des systèmes de gestion de la chaîne d'approvisionnement et des systèmes de planification des ressources de l'entreprise. Cela nécessite l'utilisation d'interfaces et de protocoles standardisés.

- **Traçabilité** : les CPPS doivent être capables de suivre le mouvement des matériaux et des produits tout au long du processus de production. Cela nécessite l'utilisation de systèmes de suivi et de traçabilité.
- **Surveillance et contrôle en temps réel** : les CPPS doivent être capables de surveiller et de contrôler les processus physiques en temps réel. Cela nécessite l'utilisation de capteurs et d'actionneurs pour collecter et transmettre des données, ainsi que des algorithmes de contrôle en temps réel.
- **Intégration des données** : Afin de permettre l'intégration de données provenant de diverses sources telles que capteurs, actionneurs, machines et opérateurs humains, les CPPS doivent se doter de normes et de protocoles de données adéquats, garantissant ainsi un échange fluide et sécurisé entre les différents systèmes.
- **Fiabilité** : les CPPS doivent être fiables pour garantir que la production n'est pas interrompue. Cela nécessite l'utilisation de systèmes tolérants aux pannes qui peuvent continuer à fonctionner même en cas de défaillances.
- **Flexibilité** : les CPPS doivent être capables de s'adapter aux évolutions des besoins de production. Cela nécessite l'utilisation de systèmes de fabrication flexibles qui peuvent être facilement reconfigurés.

Les exigences des CPPS évoluent constamment à mesure que la technologie se développe. Avec l'émergence de nouvelles technologies, de nouvelles exigences seront identifiées. Il est important de rester à jour sur les dernières exigences afin de concevoir et mettre en œuvre des CPPS à la fois efficaces et efficaces.

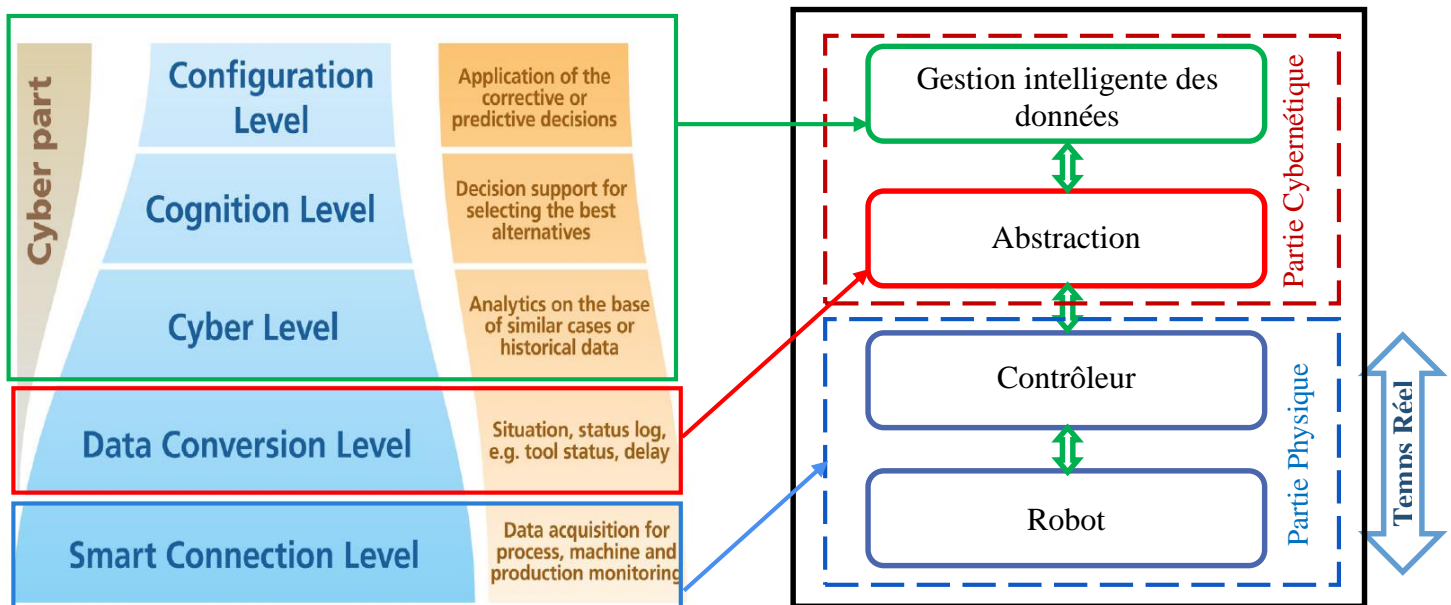
III.4.c) Architecture de principe du CPPM

Pour la mise en œuvre de notre CPPM, on a adapté l'architecture 5C de monostori et al, [71] à notre cas d'utilisation. L'architecture du CPPM se compose de différentes couches, chacune ayant des responsabilités spécifiques.

La première couche représente le robot cartésien. La deuxième couche c'est la couche qui est responsables des fonctions de contrôle de bas niveau, le rôle du contrôleur robot dans le CPPM est de garantir la transmission des données physique vers la partie cybernétique, sans avoir à passer par le PLC et le SCADA. Pour la communication entre la partie physique et part cybernétique, le protocole OPC-UA est le protocole le plus approprié. La troisième couche est la couche d'abstraction, elle est responsable de la transformation des données brute en informations utilisable par la couche des gestions intelligente des données et vice-versa. La quatrième est la couche des gestions intelligente des données, dans cette couche les données opérationnels sont analysés sur la base des données historique du robot, elle

assure aussi la pris de décision pour assurer un fonctionnement optimal du robot, ainsi que la génération des configurations et groupes de paramètres correspondant aux meilleurs cas.

La couche d'abstraction contient le AAS passif qui est un fichier statique ou un package de fichiers utilisé pour l'échange des informations entre partenaires, principalement dans le but de fournir les données du catalogue d'un asset. Les informations des assets sont stockées dans le AAS dans un format de données uniforme. En cas le AAS est un AAS proactif, il contient une compétence de prise de décision qui peut assurer un comportement autonome [Ocker2021], en d'autre terme le AAS peut assurer la gestion intelligente des données. Durant cette présente étude on se focalise sur l'implémentation de AAS passif.



L'architecture 5C pour la mise en œuvre du CPPS[71].

Notre CPPM

Figure III.7 - L'architecture du principe pour la mise en œuvre du CPPM

Un module de production cyber-physique (CPPM) est une unité autonome de production chargée d'une tâche ou d'un processus spécifique. Les CPPM sont généralement composés de composants physiques tels que des capteurs, des actionneurs et des machines, ainsi que de composants cybernétiques tels que des logiciels, des contrôleurs et des réseaux [72].

III.4.d) Implémentation :

- AAS :

L'implantation utilisant les AAS satisfait les exigences quand a l'implémentation d'un CPPM ce qui nous a poussez à utiliser les AAS car ils sont conçus sur la base d'architecture microservice modulaire, interopérable et évolutive (scalable). (Voir Chapitre.1 pour plus d'information sur les AAS).

- Choix et Motivation :

Les AAS ont été choisis comme moyen d'implémenter la composante cybernétique de notre module, car ils sont conçus sur une architecture qui répond aux exigences des CPPS. L'objectif fondamental de cet effort est de rendre la complexité déjà énorme des systèmes, qui est susceptible de continuer à croître à l'avenir, plus gérable, voire totalement gérable, en passant d'une approche centralisée à une approche décentralisée. Par conséquent, on peut s'attendre à une désintégration progressive de la pyramide d'automatisation traditionnelle en un réseau distribué et décentralisé de participants au service [73].

- Composant I4.0 :

Le composant I4.0 est un cas spécifique des systèmes cyber-physiques [19]. Il est constitué d'une ressource et de son AAS. Plusieurs I4.0 peuvent se former pour constituer un composant I4.0 plus grand. Le AAS a deux interfaces principales : une interface pour la communication entre l'AAS et l'actif ainsi qu'une interface conforme I4.0 pour la communication entre les différents AAS. La vision d'I4.0 est de construire un système I4.0 organisé de manière décentralisée contenant plusieurs AAS où les participants autonomes et intelligents du système interagissent dans le but d'atteindre ses objectifs individuels.

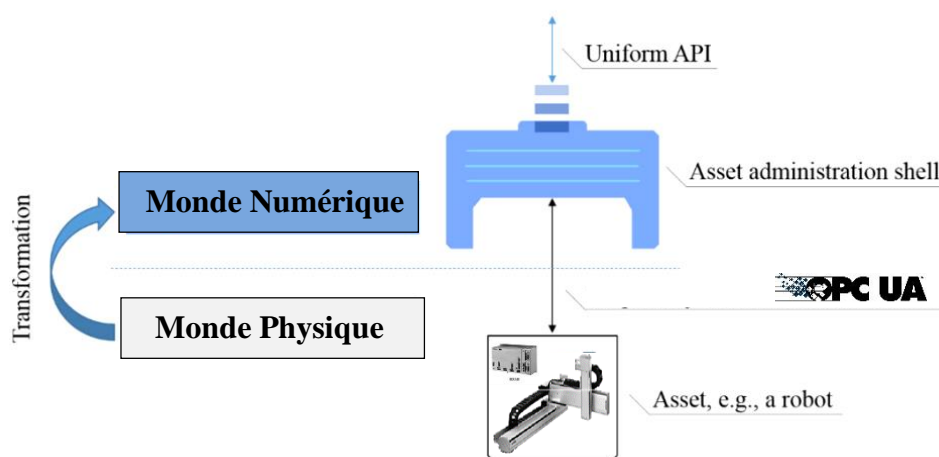


Figure III.8 - Le rôle du Shell d'administration d'actifs dans la numérisation [72]

- Architecture Organique de la solution

L'avantage de ce diagramme est qu'il permet de représenter des relations plus dynamiques. Cela peut faciliter la compréhension des systèmes complexes et favoriser l'analyse des interactions entre les différents éléments du réseau. L'objectif principal de la (figure III.8) est de représenter visuellement les relations entre les différentes composantes de notre implémentation à vocation expérimentale, en mettant l'accent sur l'interconnexion plutôt que sur la hiérarchie.

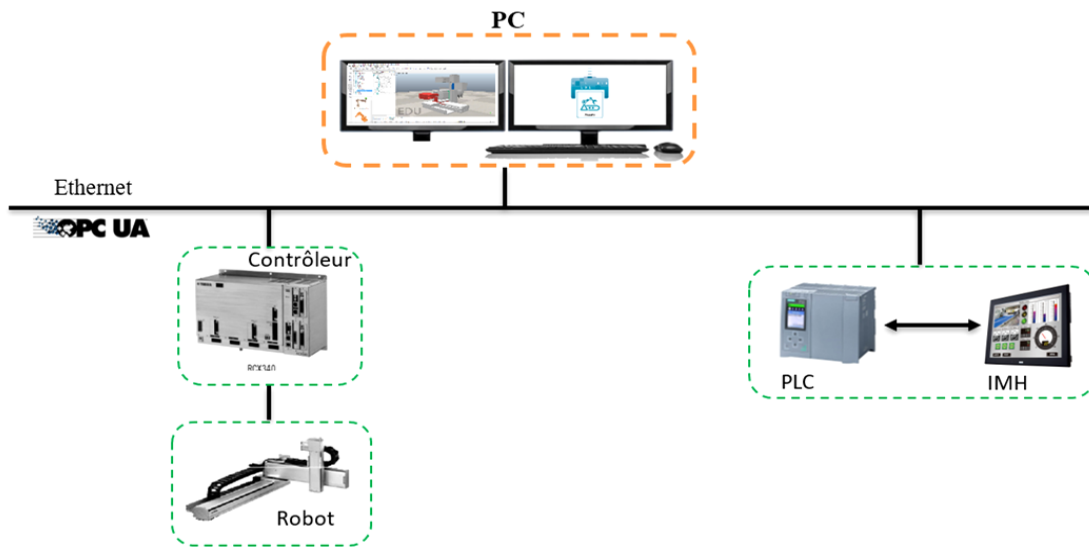


Figure III.9 - Diagramme décrivant l'architecture Organique

- Analyse fonctionnelle

Le deuxième diagramme (Figure III.10) est une représentation visuelle des différentes fonctionnalités et composants. Il met l'accent sur les interactions et les relations entre ces composants pour atteindre les objectifs spécifiques de la solution. Ce diagramme est couramment utilisé dans le domaine de l'ingénierie logicielle et des systèmes d'information pour décrire l'architecture d'une application, d'un système ou d'une solution. Il permet de visualiser comment les différentes fonctionnalités sont organisées et comment elles interagissent les unes avec les autres.

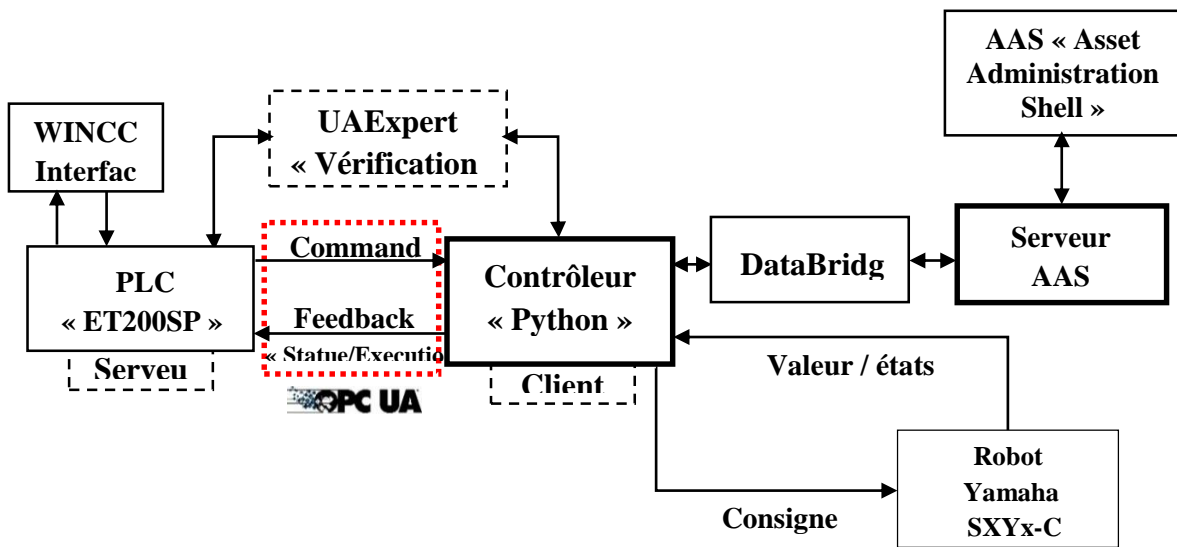


Figure III.10 - Diagramme décrit l'architecture Fonctionnelle de la solution

✓ Objectif

L'objectif principal est de fournir une vue d'ensemble claire et structurée des fonctionnalités de la solution, en identifiant les principales dépendances et les interactions clés entre les différents composants. Cela permet de comprendre comment la solution est organisée et comment les différentes parties interagissent pour fournir la fonctionnalité globale et les données opérative en temps réel.

✓ Cas d'utilisation

Le module crée est pour faire une opération Pick and Place. L'opération de prélèvement et de placement peut être décomposée en plusieurs étapes :

- ✓ Localisation de l'objet : Le système de vision du robot détermine l'emplacement de l'objet (d'une façon préprogrammée ou via un système d'identification).
- ✓ Préhension de l'objet : La pince du robot saisit l'objet.
- ✓ Transport de l'objet : Le robot déplace l'objet vers l'emplacement souhaité.
- ✓ Placement de l'objet : Le robot place l'objet à l'emplacement souhaité.

✓ **Description du fonctionnement général**

L'automate est chargé de contrôler les actions du robot en fonction des données d'entrée qu'il reçoit. La communication s'effectue en envoyant des commandes au contrôleur du robot, qui à son tour reçoit des retours d'informations (feedback). Ensuite, le DataBridge facilite l'échange de données entre les différents composants du système, notamment le serveur AAS qui envoie l'AAS, ainsi que le contrôleur qui transmet les données au contrôleur automate. Un serveur AAS est responsable de la création et de la mise à disposition des AAS, qui représentent numériquement le jumeau numérique.

✓ **Rôle de chaque entité :**

Les rôles décrits et la répartition des fonctions et des responsabilités de notre module sont comme suite :

- ✓ **Robot** : Un robot est un dispositif mécanique ou électromécanique conçu pour effectuer des tâches automatisées ou répétitives. Dans ce cas d'étude le robot représente la ressource physique objet de cette étude, son rôle est d'effectuer des tâches de déplacement d'objets de la table vers un convoyeur et vice versa.
- ✓ **Contrôleur** : est un appareil programmable qui gère des moteurs et des variateurs électroniques connectés pour exécuter des fonctions spécifiques pour des machines d'automatisation et de commande [75]. Il reçoit les consignes de tâches à effectuer émises par le PLC, et les transforme en consignes de commande (ex : position, vitesse...). Il reçoit les mesures en temps réel à partir du robot, et gère les différentes protections de ce dernier.
- ✓ **PLC « ET200SP »** : Son Rôle est d'envoyer des ordres vers le contrôleur robot en fonction de la logique de commande et des données qu'il reçoit du contrôleur du robot, de capteurs, ou bien des autres entités de la cellule de production (interface de supervision et de commande, autres contrôleurs...). Il gère en partie la protection du robot (alarmes, défauts, boucle d'arrêt d'urgence).
- ✓ **DataBridge** : Le DataBridge prend en charge l'intégration de divers protocoles avec Asset Administration Shells. Les données peuvent être acquises à partir de divers points de terminaison, être transformées et poussées dans SubmodelElements. En étant fourni en tant que composant prêt à l'emploi facile à utiliser sur DockerHub, il peut facilement être intégré dans ses propres cas d'utilisation [77].

- ✓ **AAS « Asset Administration Shell »** : Un AAS est la représentation numérique d'un actif physique qui contient des informations détaillées et contextuelles sur cet actif.
- ✓ **Serveur AAS** : Le composant serveur AAS fournit un serveur AAS vide qui peut être utilisé pour héberger plusieurs AAS et sous-modèles.

✓ **Condition préalable :**

Les conditions préalables au démarrage d'un robot sont les critères qui doivent être remplis pour garantir une utilisation sûre et fiable du robot. Ces critères incluent généralement une programmation adéquate du robot, son état de sécurité, ainsi qu'un environnement sécurisé.

Les conditions préalables spécifiques varient en fonction du type de robot et de la tâche à accomplir. Néanmoins, tous les robots doivent respecter certaines conditions préalables de base pour être utilisés de manière sûre et fiable.

Il est important de respecter les conditions préalables afin de garantir que le robot est sûr à utiliser et qu'il fonctionnera comme prévu.

✓ **Alarme et défauts :**

Les alarmes et les défauts sont une partie importante de la sécurité et de la fiabilité des robots et des API. Ils contribuent à garantir que ces systèmes fonctionnent de manière sûre et prévisible. Ils augmentent la sécurité en jouant un rôle essentiel dans la prévention des accidents en informant les opérateurs des problèmes ou des dangers potentiels, et amélioration de la fiabilité en s'assurant que les robots et les API fonctionnent de manière sûre et prévisible.

Ils réduisent les temps d'arrêt en facilitent l'identification et la résolution des problèmes avant qu'ils ne provoquent des interruptions de production. Ils optimisent productivité en défauts contribuent à maintenir les robots et les API à leur plein rendement, garantissant ainsi une productivité maximale.

III.4.e) Développement du AAS

Le diagramme dans la figure III.11 offres des notations adaptées pour représenter les concepts et les relations de l'Asset Administration Shell (AAS) et peut inclure les éléments (Asset, Submodel, Interface, Relation, ...).

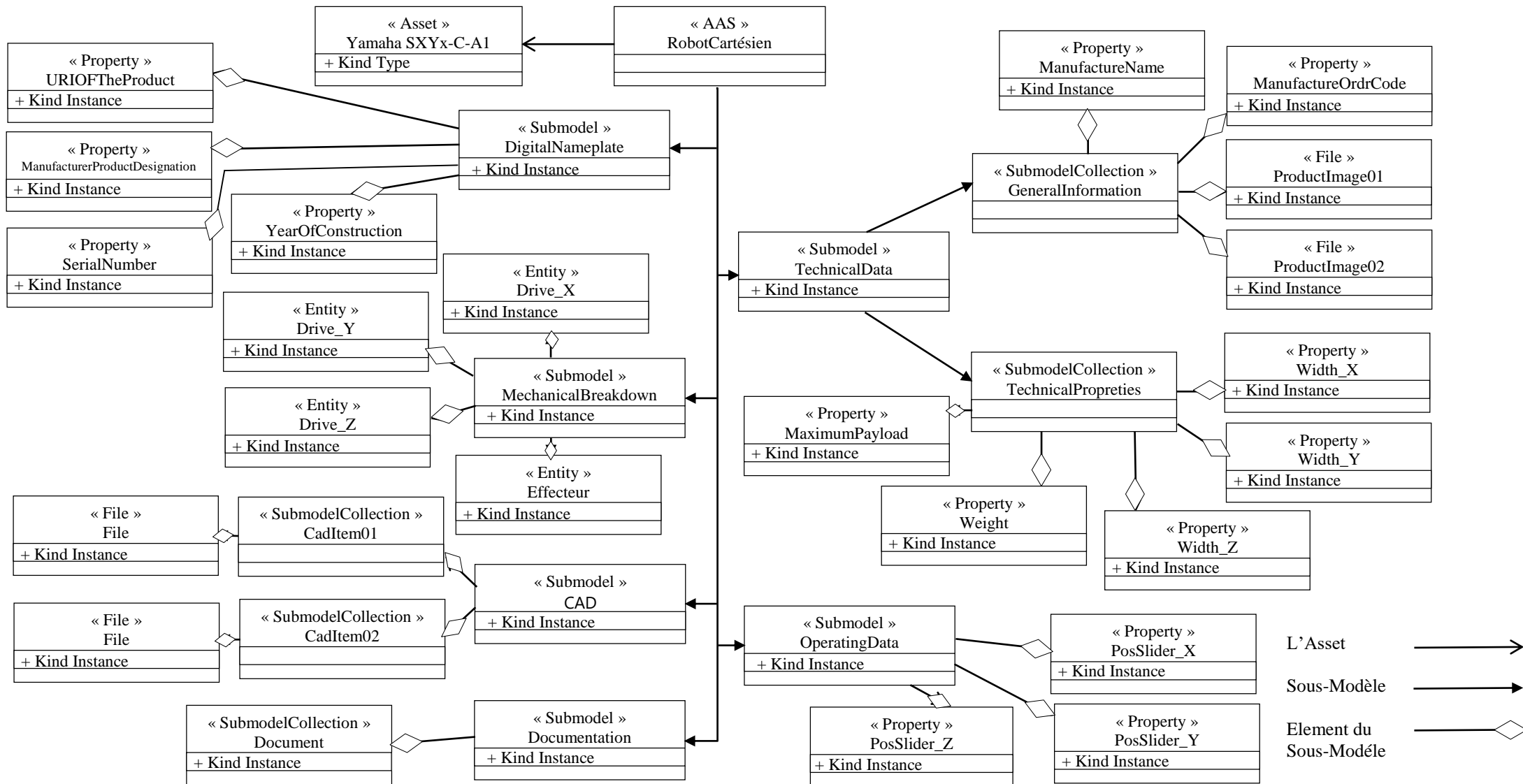


Figure III.11 - Diagramme décrivant le AAS du Robot.

Dans notre AAS on a créé plusieurs sous-modèle qui acte comme une source d'information sur le module :

- **DigitalNameplate** : est la plaque signalétique du robot, il existe plusieurs éléments propriétés qui sont des éléments sous-modèle qui donne le nom de constructeur, numéro de série du robot, l'année de construction et le pays d'origine.
- **MecanicalBreakdown** : il s'agit des pièces et composants qui constituent le robot, nous nous sommes basés sur les grandes pièces les plus importantes comme les entraînements moteurs et les moteurs ainsi que l'effecteur.
- **CAD** : on trouve les model 3D du robot en format (.stp)
- **TechnicalData** : ce sont les données techniques du robot qui se déploie dans deux collections des éléments sous-modèle, Information General et les Propriétés Technique qui sont le poids, la hauteur, la longueur, la charge maximum, etc.
- **OperationalData** : les données qui sont reliée avec le changement d'états ou des valeurs en temps réel. Notre AAS suit le changement de position de chaque slider sur son propre axe et renvoie ces coordonnées dans le fichier AAS à partir du contrôleur robots (via OPC-UA), et après les afficher dans une interface utilisateur graphique « GUI » connectée à l'AAS.
- **Documentation** : Répertoire des documents, PDFs et manuels portant sur le robot.

✓ **Modélisation d'un AAS : [14]**

Les concepts clés de la modélisation AAS sont présentés dans le but de décrire l'interaction des différents éléments du modèle pour une tâche spécifique.

Au niveau supérieur, l'Asset Administration Shell (AAS) contient des informations sur les actifs, les sous-modèles et les vues. Chacune de ces unités d'information possède des attributs permettant une description plus détaillée.

Chaque AAS comprend des sous-modèles spécifiques tels que la plaque signalétique numérique, la documentation et les données techniques. Les sous-modèles de plaque signalétique et de documentation sont basés sur les modèles de sous-modèles correspondants fournis par l'IDTA (International Digital Twin Association). Des informations plus détaillées peuvent être trouvées dans ces sous-modèles, décrivant la station dans son ensemble, y compris les valeurs de raccordement électrique et les dimensions géométriques. Cependant, pour répondre aux besoins du planificateur, des détails techniques sur les composants sont également nécessaires.

✓ Runtime :

Puisque les AAS sont basés sur le concept des micro-services, L'Exécution du serveur AAS, entraîne la création d'un micro-service qui charge la configuration du module du AAS à partir d'un fichier (.aasx) et exécute les services inclus et les met à disposition ainsi que les métadonnées du robot sous formes de ressources accessibles via des interfaces API REST.

III.5) Conclusion

Dans ce chapitre, nous avons détaillé les tâches effectuées par les-robots et de leurs types. Nous nous sommes étalés en particulier sur celui qui nous intéresse à savoir le robot cartésien « Yamaha SXYx-C-A1-15-15-ZFL20-15 » et abordé une étude de son espace de travail, ses limites, sa capacité à se déplacer, et à d'autres aspects liés à ses caractéristiques. L'achèvement est également mis en évidence sur l'automatisme et les systèmes de production physique cybernétique qui à son tour, nous amène aux CPPS et leur exigences et Architecture avec l'examinassions des CPPM. Enfin, il a été mentionné l'Implémentation qui contient AAS et le développement de l'AAS, Architecteur Organique et Analyse fonctionnelle.

Dans l'étude de cas présentée au chapitre suivant « Conception d'un module de Production Cyber-Physique », nous nous concentrons sur la démarche et sur les méthodes à suivre qui nous permettent d'accomplir. Le but que nous visons.

CHAPITRE IV :

CONCEPTION D'UN CPPM :

CAS D'ETUDE D'UN

ROBOT CARTESIEN

IV. Chapitre IV : Conception d'un module de Production Cyber-Physique : Cas d'étude d'un Robot Cartésien

IV.1) Introduction

Dans ce chapitre, nous nous concentrons sur le développement et la conception d'un module de production cyber-physique, avec un robot cartésien comme cas d'étude. Nous allons concevoir la partie cybernétique (cyber-part) du robot, qui englobera l'administration, la commande et la simulation. La validation du module se fera à l'aide de la technique du commissioning virtuel.

La conception du module que nous allons développer s'effectue en deux parties, la première partie est dédiée au développement du modèle virtuel de notre équipement, notamment le robot Yamaha SXYx_C_A1, son contrôleur, la programmation de la logique de commande sur l'automate programmable industriel SIEMENS ET200SP doté d'une CPU 1512SP-1 PN. L'établissement de la communication entre les différents composants et la vérification et validation de la commande via la technique SIL (software-in-the-loop).

La deuxième partie est dédiée à la création d'un AAS, le serveur qui va l'héberger, le registre et le pont de données qui transmettent les variables en temps réel vers le AAS comme des données opérationnelles, avec des outils récemment développés. Finalement assurer et garantir l'interconnexion et vérifie que tous ces éléments fonctionnent de manière cohérente et harmonieuse

IV.2) Partie I

La première partie du travail consiste à préparer le cas d'étude sur lequel nous allons illustrer notre travail.

L'objectif du travail est d'utiliser le robot comme cas d'étude durant le développement et la validation de la partie cybernétique de notre CPPS, et non la création du robot lui-même. Pour cela, nous avons opté pour l'utilisation d'un modèle 3D existant du robot industriel afin d'éliminer tous les problèmes résultants d'éventuelles erreurs d'une conception manuelle rapide.

IV.2.a) Le site 3Dfindit.com propulsé par CADENAS

3Dfindit.com est le moteur de recherche visuel de nouvelle dimension qui explore des milliards de modèles CAO 3D à partir de centaines de catalogues de fabricants du monde entier pour l'ingénierie mécanique, électrique et électronique. Avec ses fonctions de recherche intelligentes et ses données CAO gratuites, 3Dfindit.com est la plateforme indispensable pour les ingénieurs et les concepteurs [80].

The screenshot shows the 3Dfindit.com interface. At the top, there is a search bar with the text '5536 Browse 3D CAD & BIM catalogs...'. Below the search bar, there are navigation options: Filter, 3D Sha..., 2D Ske..., Color, Grid, Catalogs, and Compare. The breadcrumb trail indicates the search path: Yamaha Motor Co., Ltd > XY-X SERIES > Arm Type > SXYx (2,3,4axes) > SXYx-C-ZFL20 (3axes). The main content area shows a table with columns: Model - Cable MODEL, Combination COMB, X-axis Stroke XSTROKE [mm], Y-axis Stroke YSTROKE [mm], and Z-axis Stroke ZSTROKE [mm]. A single row is visible with values: SXYx-C, A1, 150, 150, 150. Below the table, there is a 3D model viewer with a '3D Shape' button and a 'Download CAD' button. To the right, there is a 'Similar parts' section showing a 'YAMAHA SXYx-C-ZF (3axes) - SXYx-C-A1-15-15-ZF-15' model with a 90.3% match score. The footer contains links for Contact, Imprint, Privacy policy, GTC, Cookie settings, Terms of Use, and version information: V12.6.1 Build 1423 ©2023 | CADENAS.

Model - Cable MODEL	Combination COMB	X-axis Stroke XSTROKE [mm]	Y-axis Stroke YSTROKE [mm]	Z-axis Stroke ZSTROKE [mm]
SXYx-C	A1	150	150	150

Figure IV.1 - 3Dfindit : la recherche et téléchargement du model 3D du robot

3Dfindit.com les prend en charge avec des données CAO natives et des fonctions de recherche intelligentes pour les composants techniques tels que la recherche de formes 3D, la recherche de croquis 2D, la recherche de couleurs, la recherche de texte et la recherche de topologie. Les modèles CAO 2D & 3D peuvent être exportés en plusieurs formats (CAD, STEP, etc.) et accélèrent ainsi le processus de conception.

✓ Avantages de 3Dfindit.com [80]

- Des milliards de modèles CAO 2D et 3D gratuits ;
- Les modèles CAO sont disponibles dans des formats natifs et neutres ;
- Les ingénieurs gagnent du temps dans le processus de conception ;
- Modèles CAO intelligents avec de nombreuses métadonnées (Digital Twins) disponibles dans de nombreux catalogues.

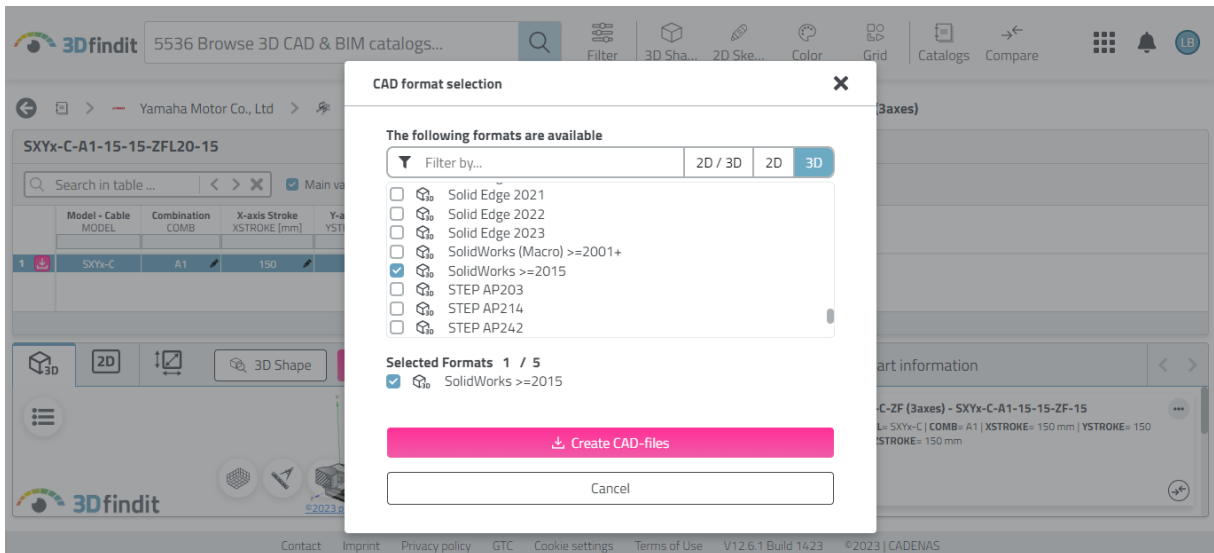


Figure IV.2 - 3Dfindit : Téléchargement des modèles 3D sous format SolidWorks

IV.2.b) SolidWorks

SolidWorks est un logiciel de concept assistée par ordinateur (CAO) appartenant à Dassault Systèmes. Il utilise le principe du concept paramétrique et produit trois types de fichiers interconnectés : la pièce, l'assemblage et le dessin. Par conséquent, toute modification de l'un de ces trois fichiers qui se répercutera sur les deux autres.

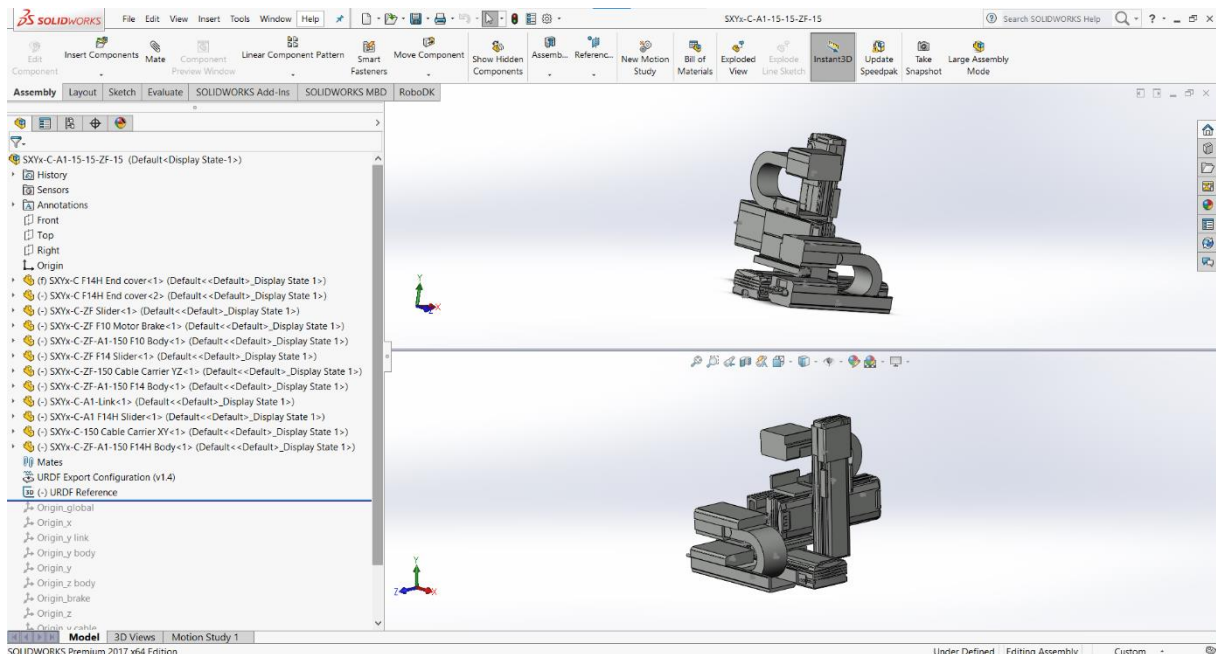


Figure IV.3 - Logiciel de modélisation 3D avec construction paramétrique.

Pour plus d'informations, veuillez-vous reporter à la page dédiée à la description des fichiers pour impression 3D avec SolidWorks [81].

SolidWorks aide à effectuer des modélisations 2D et 3D, ce logiciel de CAO est connu pour sa facilité d'utilisation et son intuition. SolidWorks est utilisé pour :

- Conception des objets 3D précis ;
- Développer des produits ;
- Revérifier la conception d'un fichier ;
- Maintenir une bibliothèque de fichiers 3D ;
- Créer des dessins 2D ;
- Créer des images et des animations d'objets 3D ;
- Estimer le coût de fabrication d'objets 3D.

Le logiciel SolidWorks prend en charge les formats de fichier 3D suivants (3D XML, ACIS, Adobe Illustrator, Adobe Photoshop, Adobe Portable Document Format, Autodesk Inventor, CADKEY, CATIA Graphics, fichiers DXF/DWG, DXF 3D, SolidWorks eDrawings, graphiques fortement compressés, HOOPS, IDF 2.0, IDF 3.0, IDF 4.0, IFC, IGES, JPEG, Mechanical Desktop, PADS, Parasolid, PDF, Pro/ENGINEER, ProStep EDMED, Rhino, ScanTo3D, Solid Edge, STEP, STL, TIFF, U3D, Unigraphics, VDAFS, Viewpoint, VRML et XPS) [81].

Dans notre cas, le format d'on on a besoin au niveau du logiciel CoppeliaSIM n'est pas disponible dans la liste des formats téléchargeables pour le Robot. De ça, il est nécessaire d'utiliser le Plug-in qui aide à convertir le modèle 3D du format SolidWorks vers format (.urdf).

IV.2.c) SolidWorks to URDF Exporter

L'exportateur SolidWorks vers URDF est un « plug-in » pour SolidWorks qui permet l'exportation pratique de pièces et d'assemblages SW dans un fichier URDF. L'exportateur créera un package de type ROS contenant un répertoire pour les maillages, les textures et les robots (fichiers urdf). Pour les pièces SolidWorks uniques, l'exportateur de pièces extrait les propriétés du matériau et crée un lien unique dans l'URDF. Pour les assembler, l'exportateur crée les liens et crée une arborescence basée sur la hiérarchie de l'assemblé logiciel [82].

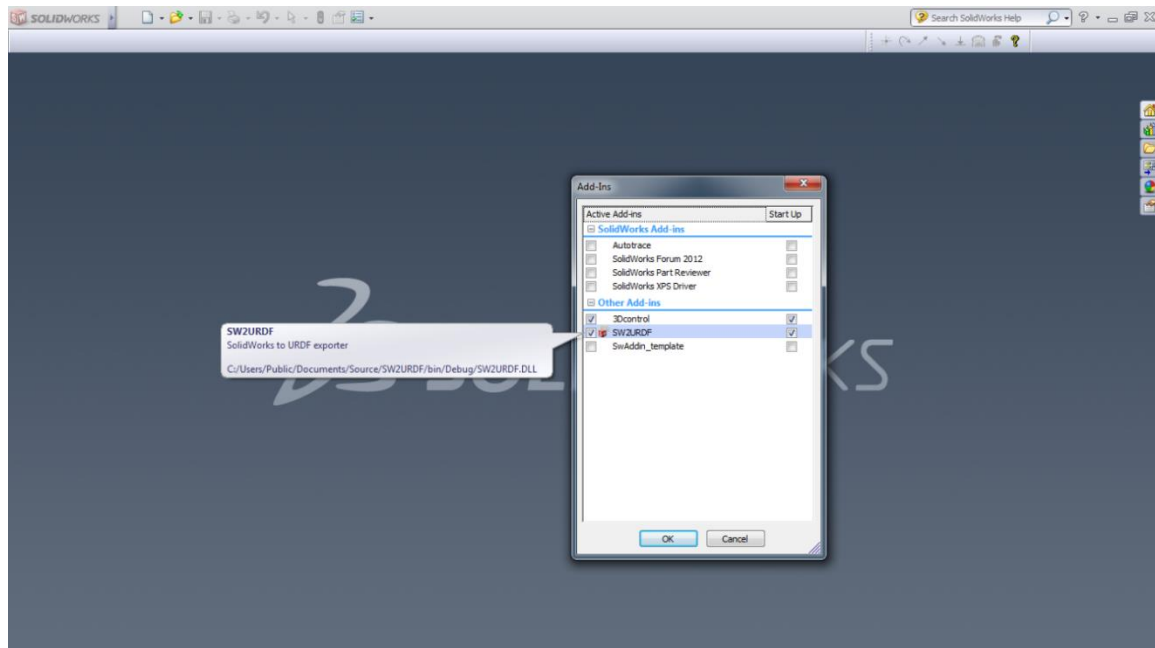


Figure IV.4 - Activation du Plugin dans le logiciel SolidWorks.

L'étape qui suit l'ajout du Plugin (Figure.IV.4), est la conversion du model 3D dans SolidWorks vers le format désirer (Figure.IV.5) :

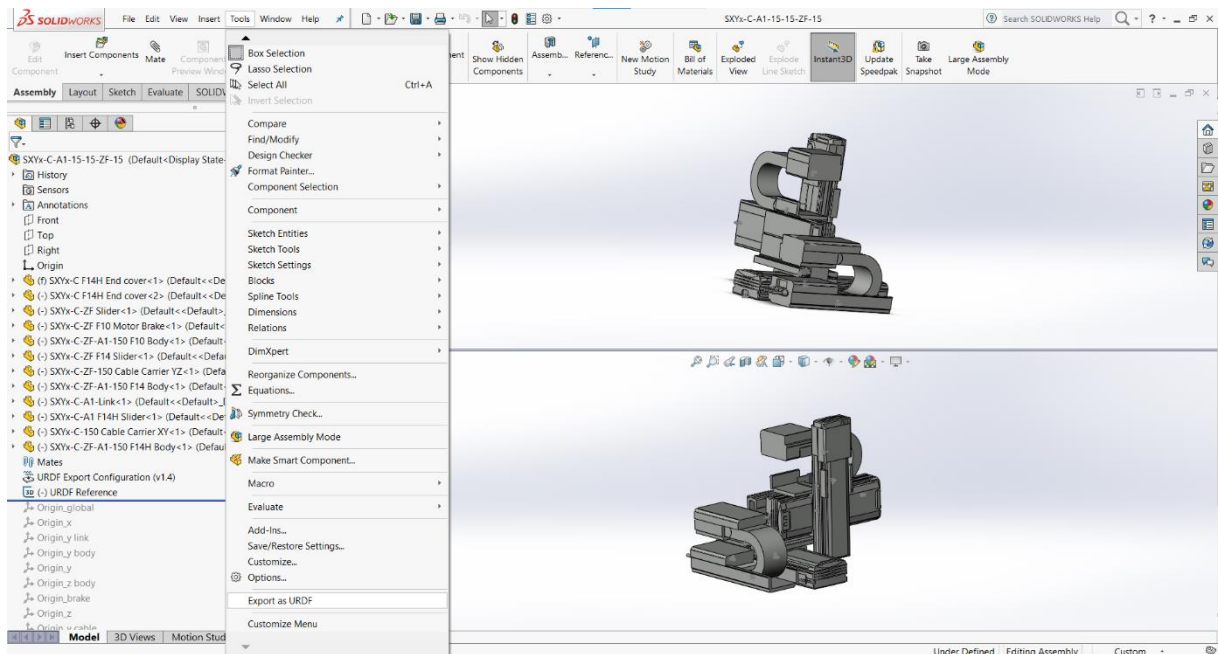


Figure IV.5 - La sélection de l'option qui permet l'exportation URDF.

Ensuite, il faut sélectionner chaque pièce avec sa propre liaison et les organiser dans une hiérarchie structurée du robot (Figure.IV.6) pour le bon fonctionnement du modèle 3D après la conversion vers le format (.urdf)

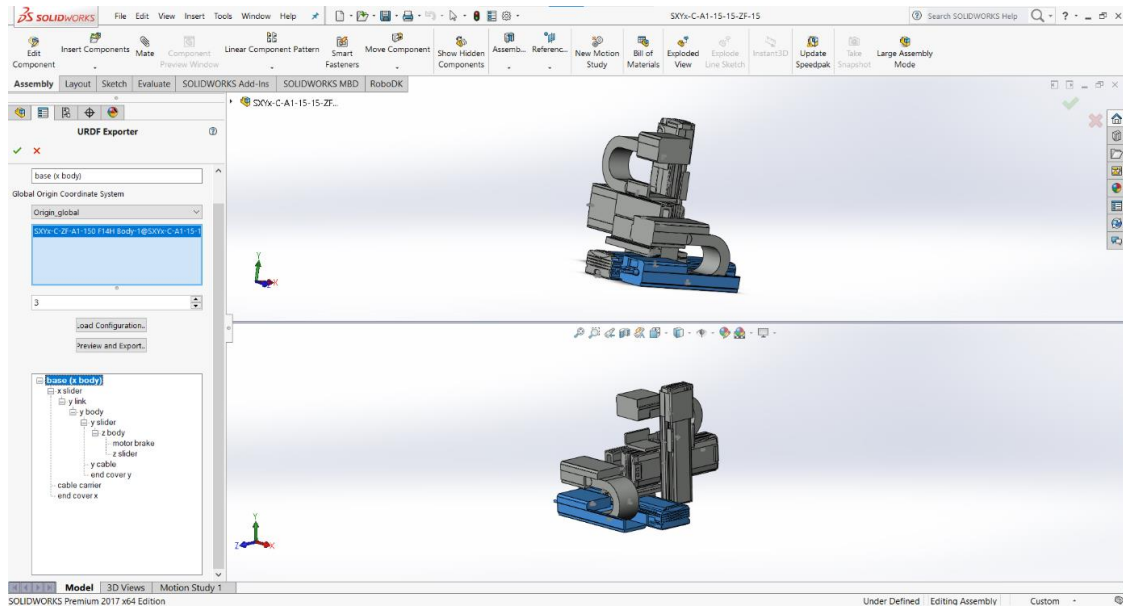


Figure IV.6 - L'organisation et la sélection de chaque pièce du modèle pour la hiérarchie globale.

Quand la conversion est terminée, le robot est prêt à être transféré, importé et chargé dans le logiciel de simulation 3D, CoppeliaSim.

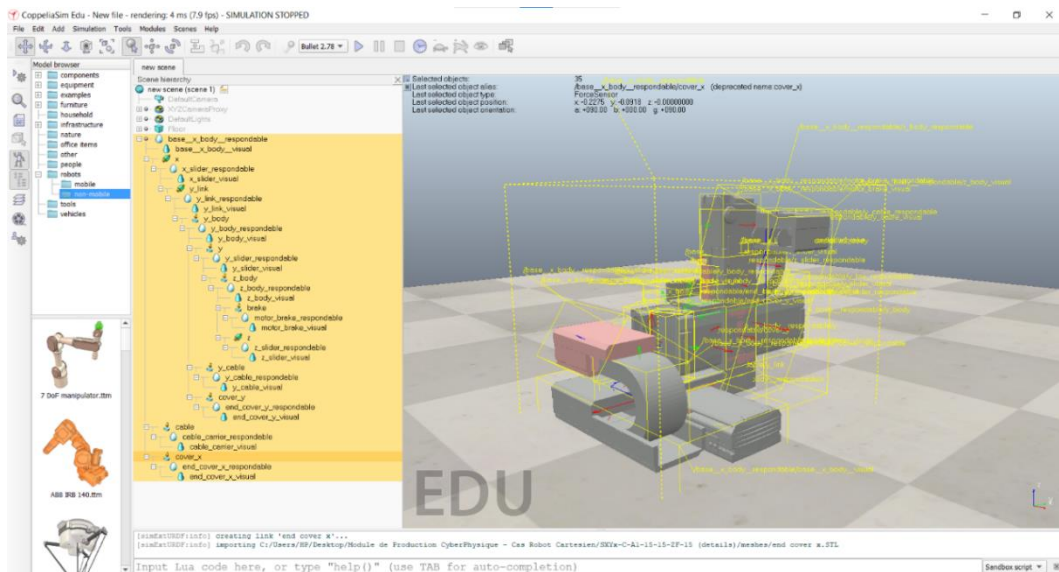


Figure IV.7 - Model 3D importé de SolidWorks vers CoppeliaSim

IV.2.d) CoppeliaSim (Version Edu)

Le simulateur robotique CoppeliaSim (anciennement V-REP), avec environnement de développement intégré, est basé sur une architecture de contrôle distribué : chaque objet/modèle peut être contrôlé individuellement via un script embarqué, un plugin, un nœud ROS, un client API distant, ou une solution personnalisée. Cela rend CoppeliaSim très polyvalent et idéal pour les applications multi-robots. Les contrôleurs peuvent être écrits en C/C++, Python, Java, Lua, Matlab ou Octave [83].

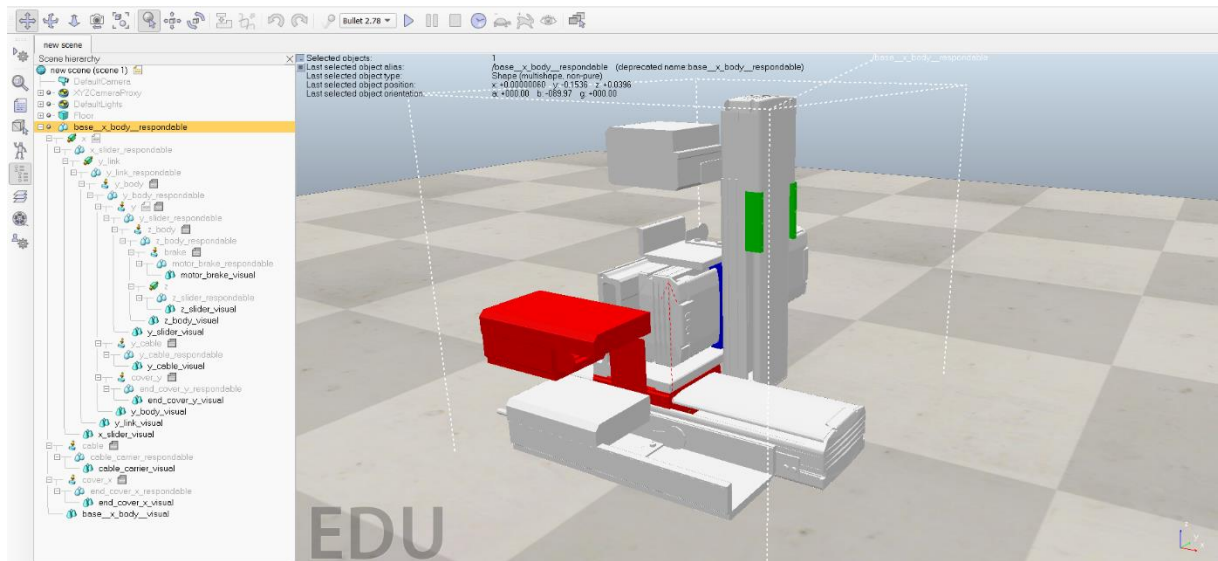


Figure IV.8 - Notre robot sur CoppeliaSim

CoppeliaSim est utilisé pour le développement rapide d'algorithmes, les simulations d'automatisation d'usine, le prototypage et la vérification rapides, la formation liée à la robotique, la surveillance à distance, la double vérification de sécurité, en tant que jumeau numérique, et bien plus encore [83].

IV.2.e) Utilisation d'articulation pour démarrer le robot

- **Les joints dans CoppeliaSim :**

CoppeliaSim propose plusieurs types de joints qu'on peut utiliser pour simuler des systèmes mécaniques complexes, Joint rotatif, Joint sphérique, Joint prismatique. Comme notre robot est supposé cartésien, nous avons des Joints Prismatiques pour simuler le mouvement sur les trois axes.

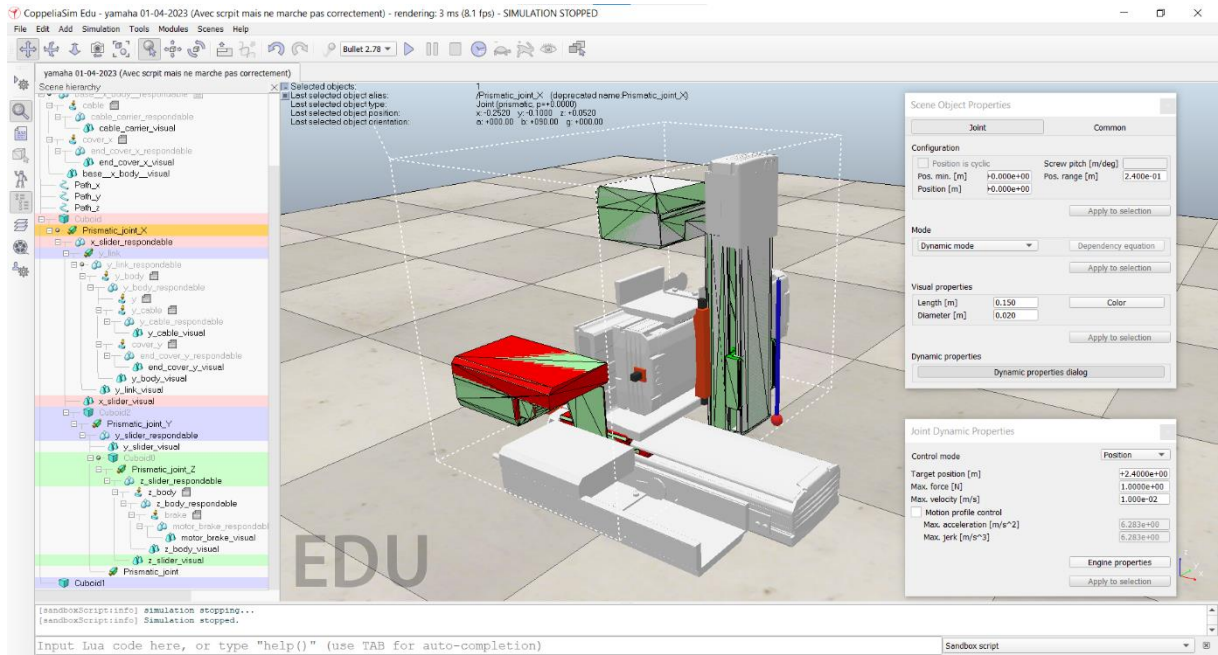


Figure IV.9 - Configuration des joints prismatiques.

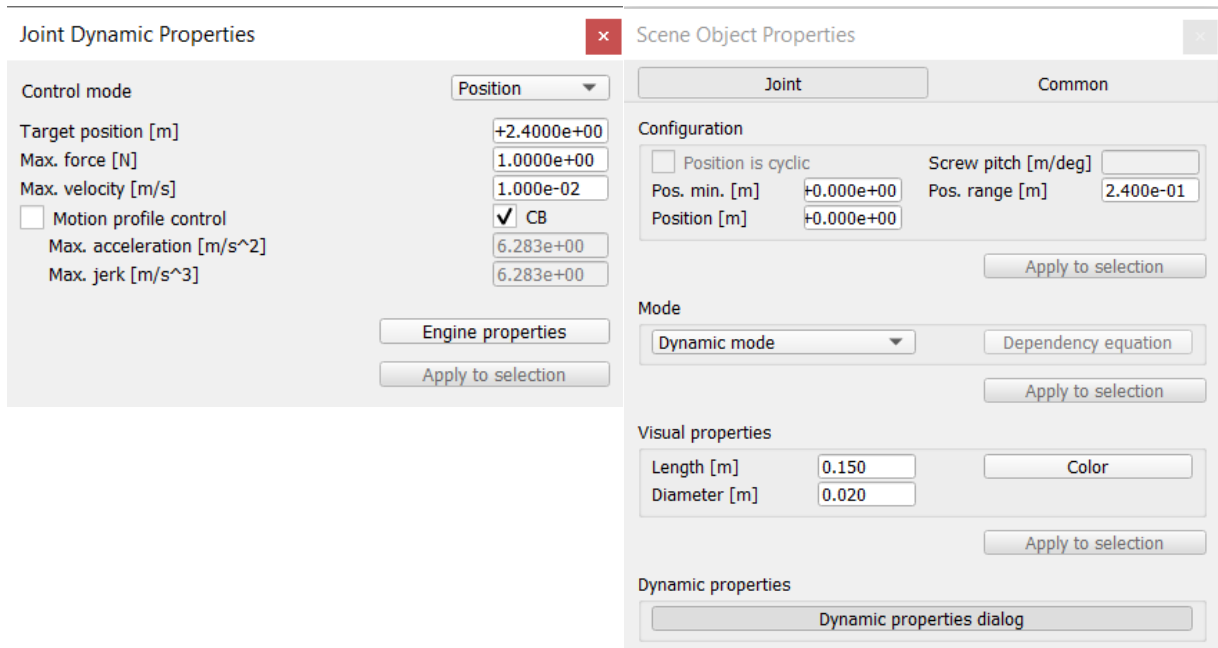


Figure IV.10 - Menu de Configuration des joints

Les joints prismatiques, permettant de simuler le robot, sont considérés comme des moteurs. Lorsqu'on a voulu tester le comportement du modèle 3D importé par CoppeliaSim, le robot s'est comporté de manière inattendue et non conforme à nos attentes.

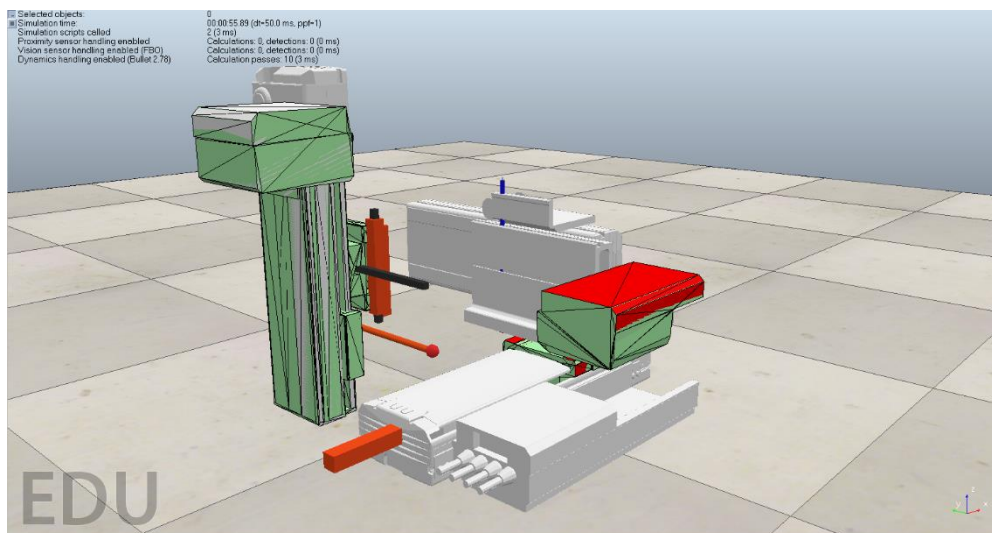


Figure IV.11 - **Robot se démonte et se comporte d'une manière inattendue désassemblage et comportement inattendu du Robot.**

Suite au problème rencontré durant plusieurs tests, nous avons pris la décision d'abandonner l'option avec le mouvement dynamique moyennant des joints, et porter notre choix plutôt sur une route plus cinématique dans le mouvement en utilisant le script Lua et python.

IV.2.f) Langage Lua

Lua est un langage procédural extensible avec de puissantes fonctions de description de données, Conçu pour être utilisé comme langage d'extension à usage général. Lua est né de la fusion de deux langages descriptifs, conçu pour la configuration de deux applications spécifiques : un des deux est pour la saisie des données scientifiques [84] script qui se distingue par sa légèreté, son extensibilité et sa puissance. Il est largement utilisé pour étendre les fonctionnalités des applications existantes ou pour créer des scripts indépendants. Lua est apprécié pour sa simplicité, sa portabilité et ses performances élevées, ce qui en fait un choix privilégié dans divers domaines, notamment, la méta-programmation, les systèmes embarqués et les jeux vidéo.

IV.2.g) Python

Python est un langage de programmation interprété, de haut niveau et polyvalent, apprécié pour sa syntaxe claire et compréhensible. Il est particulièrement recommandé pour les débutants en programmation en raison de sa facilité d'apprentissage. Python offre la possibilité de programmer selon différents styles, tels que la programmation orientée objet, fonctionnelle et impérative.

IV.2.h) RemoteAPI de CoppeliaSim

La connexion API à distance permet de contrôler le logiciel CoppeliaSim sans interagir réellement avec l'interface de l'application. Dans notre cas, Python a été utilisé pour communiquer entre CoppeliaSim et le PLC.

Cette approche facilite la connexion d'une application externe à CoppeliaSim grâce à l'utilisation de commandes d'API à distance. Elle comprend des serveurs d'API à distance et des clients d'API à distance. Le côté client peut être intégré sous forme de code compact (C/C++, Python, Java, Matlab et Urbi) dans pratiquement n'importe quel matériel, y compris les robots réels.

- **Activation de l'API distante - côté serveur :** Pour activer l'API distante côté serveur (c'est-à-dire côté CoppeliaSim), on doit s'assurer que le plug-in d'API distant a été chargé avec succès au démarrage de CoppeliaSim.

Un service de serveur peut être démarré à partir d'un script. L'utilisateur a le contrôle lorsque le service est démarré ou arrêté. Cependant, lorsqu'un service de serveur d'API distant temporaire est démarré à partir d'un script de simulation, le service sera automatiquement arrêté à la fin de la simulation. Un service de serveur d'API distant temporaire peut être démarré ou arrêté avec les 2 fonctions Lua suivante : **simRemoteApi.start**, **simRemoteApi.stop**. on peut collecter des informations sur tout service de serveur API distant avec la fonction **simRemoteApi.status** [92].

- **Activation de l'API distante - côté client :**

Python client : Pour pouvoir effectuer cette connexion, il faut d'abord extraire certains fichiers et les mettre dans un dossier avec le programme python :

- sim.py
- simConst.py
- remoteApi.dll, remoteApi.dylib ou remoteApi.so (selon la plate-forme cible).

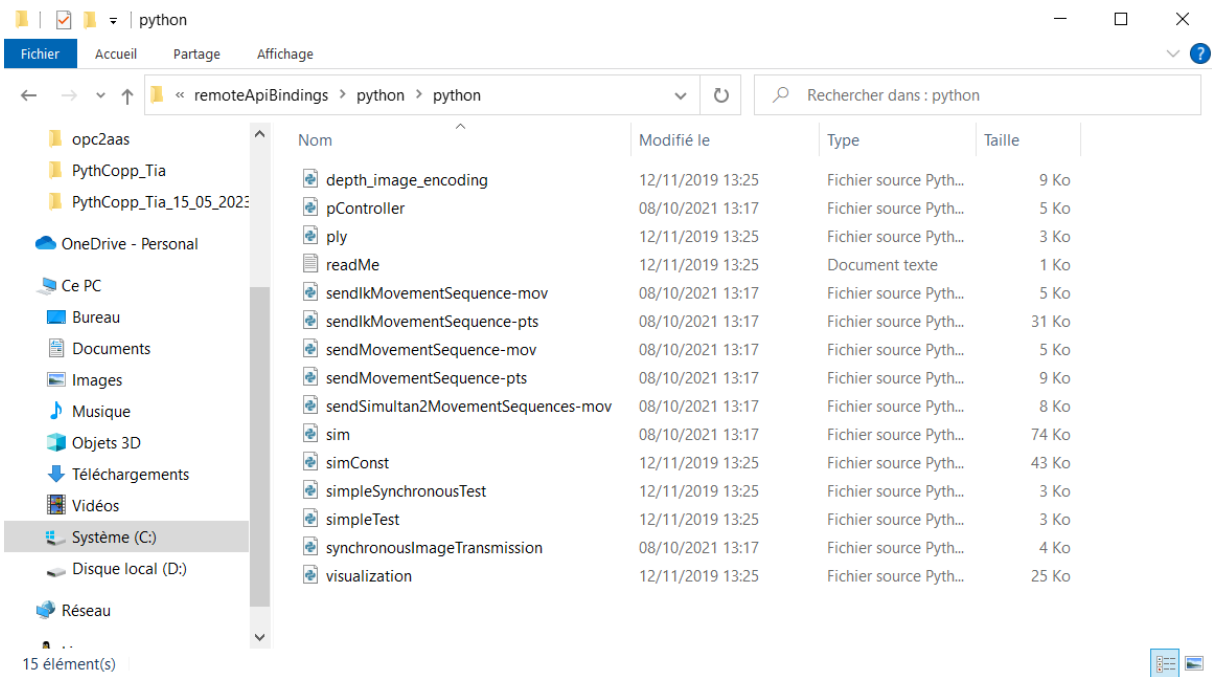


Figure IV.12 - Fichier à extraire pour l'activation d'AP distant (client).

Ces fichiers peuvent être trouvés dans le chemin d'accès suivant :
 CoppeliaRobotics>CoppeliaSimEdu>programming>legacyRemoteApi>remoteApi
 Bindings>python>python

Après avoir placé les éléments mentionnés ci-dessus dans un répertoire Python connu, vous pouvez charger la bibliothèque en utilisant l'instruction "import sim". Pour activer l'API distante côté client (c'est-à-dire votre application), appelez la fonction "**sim.simxStart**".

ROS nodes : CoppeliaSim implémente un nœud ROS avec un plug-in qui permet aux ROS d'appeler des commandes CoppeliaSim via des services ROS ou de transmettre des données [92].

NoeudBlueZero : Ce procédé permet à une application externe (située par exemple sur un robot, une autre machine, etc.) de se connecter à CoppeliaSim via BlueZero [92].

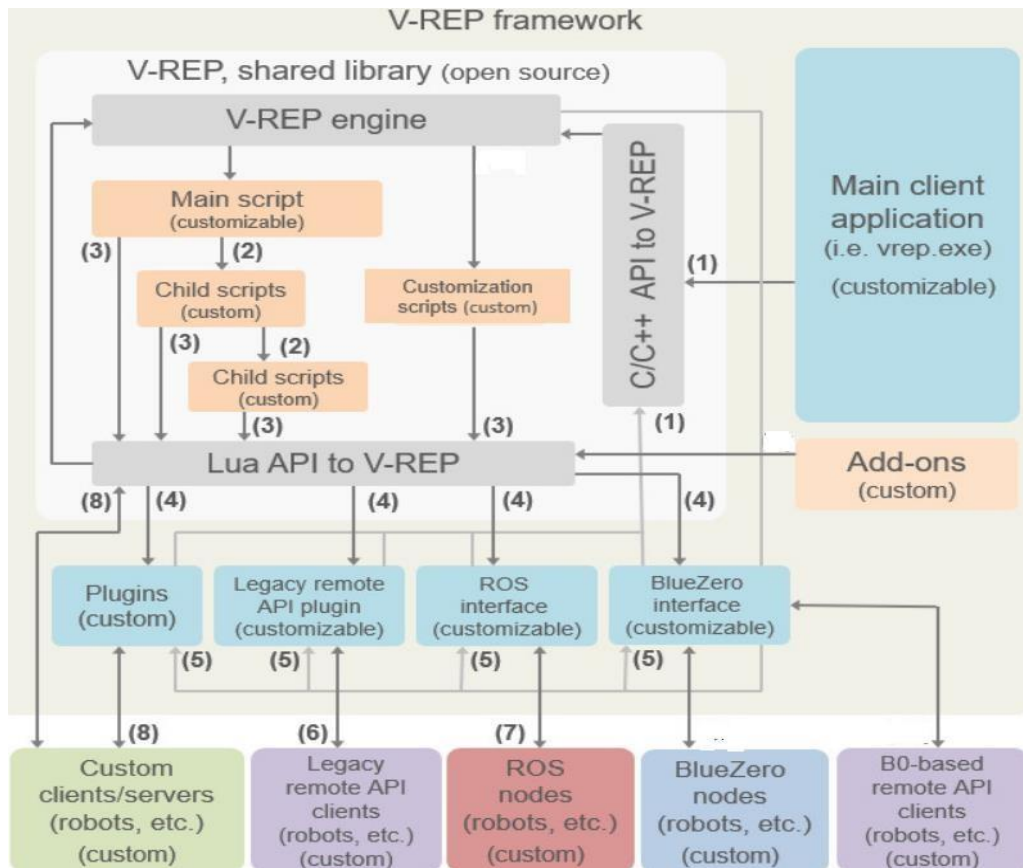


Figure IV.13 - Architecture de contrôle CoppeliaSim (anciennement V-REP)

Dans la figure IV.13 on trouve :

(1) appels API C / C ++, (2) exécution de script enfant en cascade, (3) appels API Lua, (4) rappels personnalisés de Lua API, (5) rappels d'événements CoppeliaSim, (6) appels de fonction API distants, (7) ROS transit, (8) communication personnalisée (prise, série, tuyaux, etc.) [93].

En utilisant un fichier programmé en python comme contrôleur robot, nous avons été en mesure de faire fonctionner le robot correctement et faire déplacer chaque sliders sur son axe respectif.

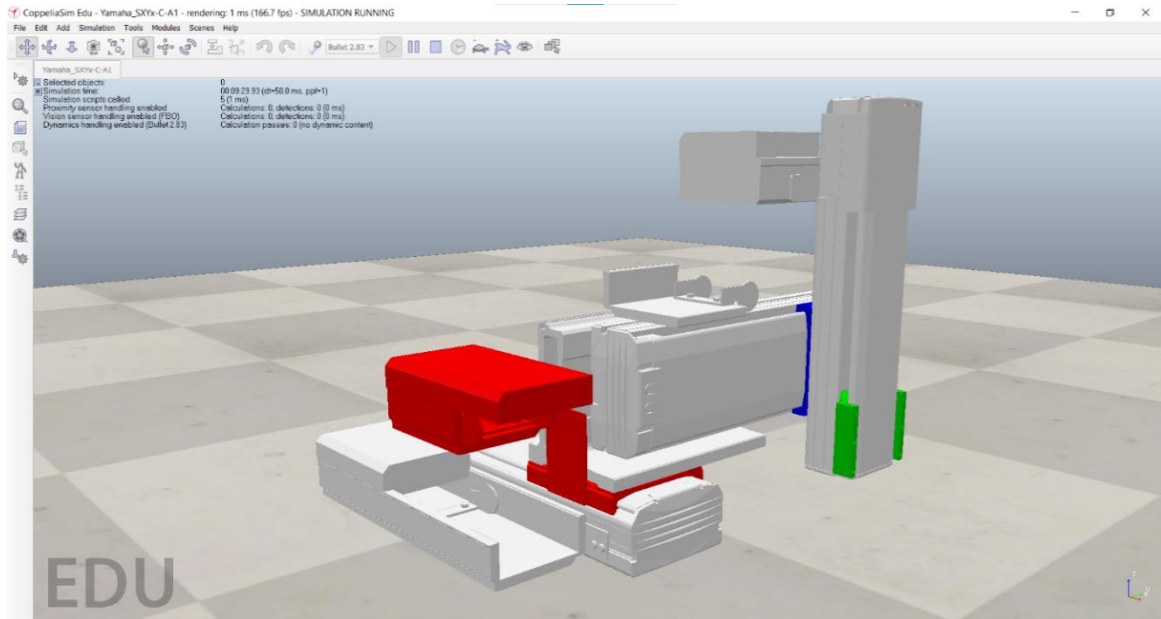


Figure IV.14 - Marche convenable du Robot jusqu'aux positions finales de chaque slider.

IV.2.i) Effecteur Robot

Pour notre Robot Yamaha SXYx_C_A1, nous avons choisi l'outil effecteur YuMi-Gripper qui est disponible dans la bibliothèque des effecteurs de Coppeliasim. Ce choix est juste pour des fins de simulation et n'a pas de relation avec l'étude menée, et il peut être remplacé par un autre effecteur lors de la phase de réalisation selon le besoin.

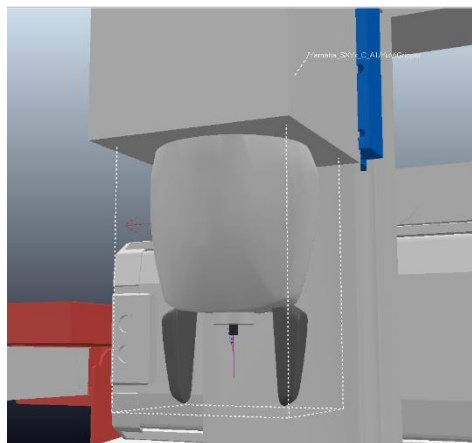


Figure IV.15 - YuMi Gripper

Le YuMi Gripper est un effecteur robotique spécifiquement pour le robot collaboratif YuMi. Il s'agit d'une pince électrique conçue pour des tâches de manipulation précise et délicate.

Pour que l'effecteur fonctionne correctement, nous avons ajouté un capteur proximité qui détecte les objets à prendre et programmé l'effecteur avec un script Lua. Le programme Lua de l'effecteur YuMi Gripper, référer à l'annexe 2.

IV.2.j) **La Commande du Robot par un Automate Programmable Industriel**

Afin d'intégrer notre robot dans une cellule de production, on a souvent besoin de relier son contrôleur à un Automate Programmable industriel. Dans notre cas, nous avons opté pour l'utilisation du PLC SIEMENS ET200SP, étant donné qu'il est disponible au niveau du Laboratoire SRP (CDTA) dans lequel nous effectuons notre Stage PFE.

IV.2.k) **PLC Siemens S7 ET200SP - 1512SP F-1 PN**

Le CPU 1512SP F-1 PN est un module de contrôle hautement performant et polyvalent spécialement conçu pour les automates programmables SIMATIC S7-1500.



Il se distingue par ses capacités de communication avancées, une mémoire généreuse, des performances exceptionnelles et une grande flexibilité de programmation. Pour les applications avec des exigences moyennes en termes de portée du programme et de vitesse de traitement, pour les configurations distribuées via PROFINET IO ou PROFIBUS DP [87].

En raison de ces caractéristiques, il est largement apprécié dans le domaine de l'automatisation industrielle et trouve une multitude d'applications dans différents secteurs.

IV.2.1) **Description de la CPU**

CPU avec mémoire de travail 200 KB code et 1 Mo de données ; Temps de fonctionnement de 48 ns bits ; Concept de protection en 4 étapes, fonctions technologiques : contrôle de mouvement, contrôle en boucle fermée, comptage et mesure ; traçage. Options d'exécution ; protocole de transport TCP/IP, communication utilisateur ouverte sécurisée,

communication S7, transfert IP, serveur Web, client DNS, OPC UA : DA serveur, DA client, méthodes, spécifications associées ; Contrôleur PROFINET IO, prend en charge RT / IRT, mise à niveau des performances PROFINET V2.3, 3 ports, I-Device, MRP, MRPD, mode isochrone, firmware V2.9 [88]

IV.2.m) TIA Portal

L'outil que nous avons utilisé pour la programmation de la logique de commande du PLC est TIA Portal. C'est un progiciel et un ensemble d'outils développé par Siemens, qui vise à intégrer plusieurs outils de développement pour les dispositifs d'automatisation à partir de l'unification et du remodelage de logiciels préexistants tels que Simatic Step 7, Simatic WinCC et Sinamics Starter [89].

Les environnements sont responsables de la programmation, du développement et de la configuration des automates, des IHM et des variateurs de fréquence Siemens. La logique de programmation de l'utilisateur dans TIA Portal suit une structure de blocs, un agent facilitant le développement, la maintenance et le diagnostic des machines et des processus industriels lorsqu'ils sont développés de manière structurée et organisée [89].

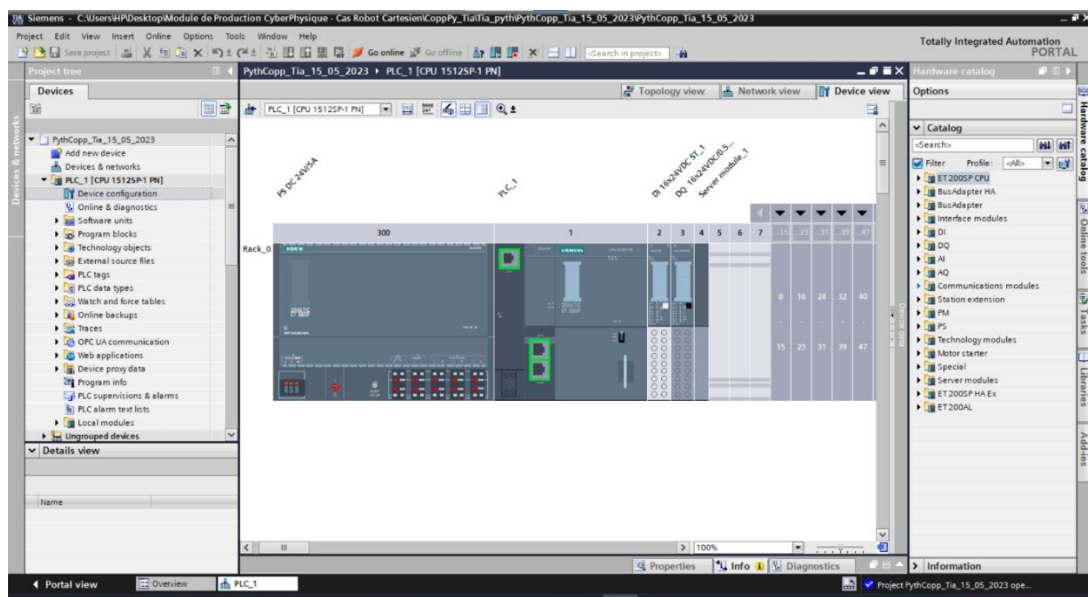


Figure IV.16 - La CPU 1512SP F-1 PN dans TIA Portal

IV.2.n) Configuration du PLC

La CPU choisit dans le projet est la CPU 1512SP F-1 PN, avec une alimentation PS DC 24V/5A et deux modules, le DI 16x24VDC ST_1 pour les entrées, et le DQ 16x24VDC/0.5A BA_1 pour les sorties.

Après la finition de la table des entrées et sorties, on démarre avec la programmation de notre PLC avec le langage Ladder et le SCL. Se référer à l'Annexe 3 et l'Annexe 4 pour les programmes en Ladder et SCL.

Le bloc des données est comme suit :

	Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	Setpoint	...
1	Input			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
2	task_over	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
3	err_comm	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
4	Pos_X	Real	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
5	Pos_Y	Real	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
6	Pos_Z	Real	0.0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
7	Trip_X	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
8	Trip_Y	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
9	Trip_Z	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
10	Output			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
11	task_sp	String	"	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
12	reset	Bool	false	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
13	vitesse	Int	0	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
14	InOut			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
15	Static			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Figure IV.17 - Bloc de données SCL

IV.2.o) Communication entre PLC et Contrôleur

Le protocole choisi pour la communication entre le PLC et notre Simulateur de Contrôleur Robot est le protocole OPC UA. Ce protocole est le protocole le plus utilisé pour communication en Industrie 4.0.

IV.2.p) Protocole OPC-UA

Le protocole OPC-UA est un standard d'échange d'informations et de services, indépendant et sécurisé, véritable esperanto du système d'information de l'usine, quel que soient les machines, les langages et les systèmes d'exploitation utilisés. Il est particulièrement dédié à l'Internet Industriel des Objets (IIoT ou I4.0) [90]. Son objectif

principal est de permettre la connectivité et l'interopérabilité entre des équipements et des systèmes variés tels que des capteurs, des actionneurs, des machines, ainsi que des logiciels de supervision et de contrôle.

L'architecture d'OPC-UA est flexible et évolutive, ce qui lui permet de transmettre des données en temps réel de manière sécurisée et fiable. De plus, il offre des fonctionnalités avancées pour la gestion des alarmes, des événements et des historiques. Il est compatible avec une large gamme de plates-formes matérielles et logicielles, et prend en charge différents protocoles de communication, notamment Ethernet, TCP/IP et même des protocoles sans fil.

IV.2.q) Communication du PLC avec l'OPC-UA

La nouvelle gamme des PLC support le protocole OPC-UA, il faut l'activer dans les paramètres de configuration du PLC dans TIA portal. L'automate peut fonctionner comme un serveur OPC-UA ainsi qu'un client.

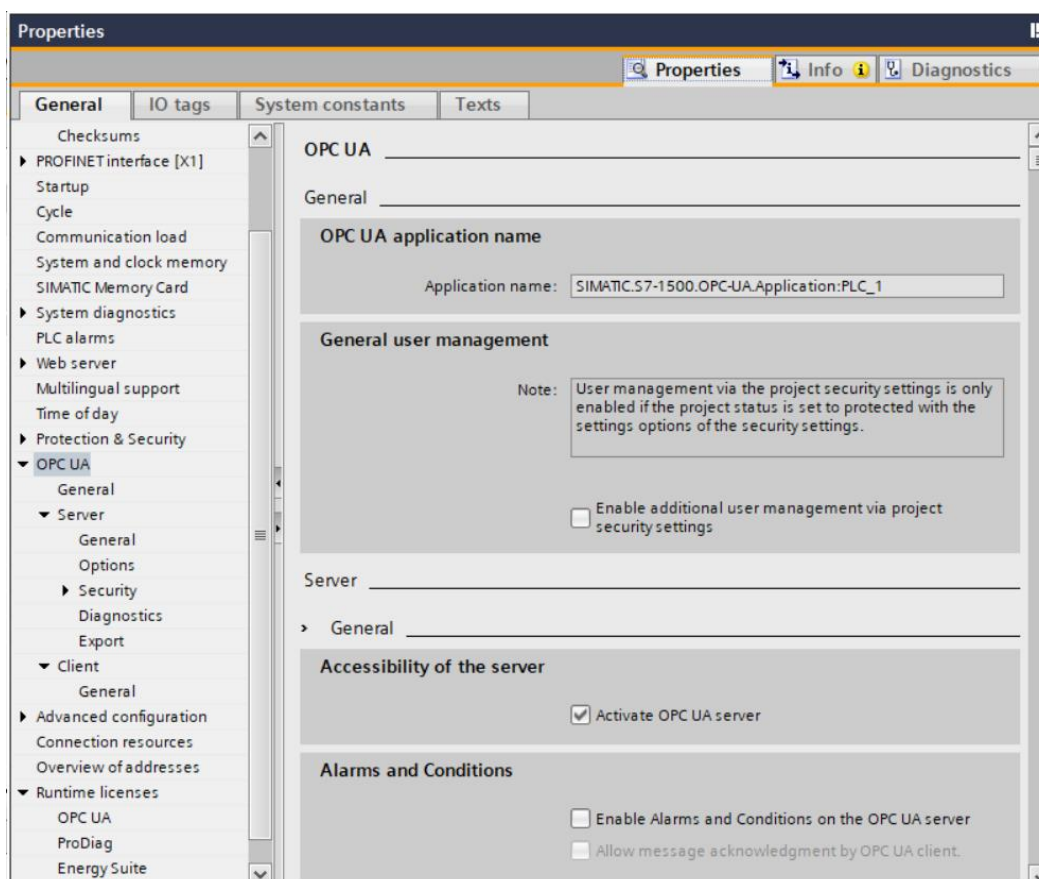


Figure IV.18 - L'activation du protocole OPC-UA du PLC

IV.2.r) UA Expert

Afin de tester notre communication, et vérifier et valider l'interface avec notre logique de commande sur le PLC SIEMENS, nous avons utilisé le logiciel UA Expert de Unified Automation.

UA Expert est un client OPC UA complet développé par Unified Automation. C'est une application multiplateforme qui peut être utilisée sur Windows, Linux et macOS. Il est conçu comme un client de test à usage général prenant en charge les fonctionnalités OPC UA telles que DataAccess, Alarmes et conditions, Accès historique et appel de méthodes UA. Il utilise la bibliothèque graphique sophistiquée QT de Nokia (anciennement Trolltech) formant le cadre de base qui est extensible par des plugins [91]. Il offre une interface graphique (GUI) permettant de naviguer et d'interagir avec les serveurs OPC UA et comprend également plusieurs plugins qui peuvent être utilisés pour effectuer des tâches spécifiques, telles que la surveillance des alarmes et des conditions, la visualisation des tendances historiques et l'enregistrement des données.

Nous utilisons UA Expert on peut voir et vérifier les données qui traverse entre le PLC et le contrôleur robot. On peut constater sur la Figure IV.19 les valeurs des différents signaux qui sont affichés.

#	Server	Node Id	Display Name	Value	Datatype	urce Timestar	iver Timestan	Statuscode
1	SIMATIC.S7-1500.OPC-UA.Application:PLC_1	NS3 String "Blo...	reset	false	Boolean	01:59:34.285	01:59:34.285	Good
2	SIMATIC.S7-1500.OPC-UA.Application:PLC_1	NS3 String "Blo...	task_sp	tacheA	String	01:59:34.285	01:59:34.285	Good
3	SIMATIC.S7-1500.OPC-UA.Application:PLC_1	NS3 String "Blo...	vitesse	1	Int16	01:59:34.285	01:59:34.285	Good
4	SIMATIC.S7-1500.OPC-UA.Application:PLC_1	NS3 String "Blo...	Pos_X	-0.0706187	Float	01:59:34.285	01:59:34.285	Good
5	SIMATIC.S7-1500.OPC-UA.Application:PLC_1	NS3 String "Blo...	Pos_Y	0.175303	Float	01:59:34.285	01:59:34.285	Good
6	SIMATIC.S7-1500.OPC-UA.Application:PLC_1	NS3 String "Blo...	Pos_Z	1.21173	Float	01:59:34.285	01:59:34.285	Good
7	SIMATIC.S7-1500.OPC-UA.Application:PLC_1	NS3 String "Blo...	Trip_X	false	Boolean	01:59:34.285	01:59:34.285	Good
8	SIMATIC.S7-1500.OPC-UA.Application:PLC_1	NS3 String "Blo...	Trip_Y	false	Boolean	01:59:34.285	01:59:34.285	Good
9	SIMATIC.S7-1500.OPC-UA.Application:PLC_1	NS3 String "Blo...	Trip_Z	false	Boolean	01:59:34.285	01:59:34.285	Good
10	SIMATIC.S7-1500.OPC-UA.Application:PLC_1	NS3 String "Blo...	err_comm	false	Boolean	01:59:34.285	01:59:34.285	Good
11	SIMATIC.S7-1500.OPC-UA.Application:PLC_1	NS3 String "Blo...	task_over	false	Boolean	01:59:34.285	01:59:34.285	Good
12	SIMATIC.S7-1500.OPC-UA.Application:PLC_1	NS3 Numeric 5...	Icon	89504e470d...	ByteString	01:59:34.285	01:59:34.285	Good
13	SIMATIC.S7-1500.OPC-UA.Application:PLC_1	NS3 String "tac...	tacheA_Recopie	true	Boolean	01:59:34.285	01:59:34.285	Good
14	SIMATIC.S7-1500.OPC-UA.Application:PLC_1	NS3 String "tac...	tacheB_Recopie	false	Boolean	01:59:34.285	01:59:34.285	Good
15	SIMATIC.S7-1500.OPC-UA.Application:PLC_1	NS3 String "de...	demarraqe	true	Boolean	01:59:34.285	01:59:34.285	Good

Figure IV.19 - Liste des variables et consigne envoyés et reçus.

Lorsque la validation des variables envoyées du PLC à l'OPC-UA est effectuée avec des valeurs correctes, le fichier Python du contrôleur est programmé en utilisant la bibliothèque OPC en tant que client OPC. En parallèle, il se connecte au RemoteAPI via le port de CoppeliaSim avec le contrôleur robot pour recevoir les instructions envoyées par le PLC sous forme de variables, puis les transmet au contrôleur robot.

IV.2.s) Simulation du robot

Le démarrage du robot est effectué en appuyant sur le bouton marche, un des boutons qui choisit une des tâches voulues et le bouton qui choisit la vitesse.

En utilisant un fichier programmé en python comme contrôleur robot, nous avons pu faire fonctionner le robot correctement et faire déplacer chaque sliders sur son axe respectif.

Notre programme est composé de plusieurs parties qui se connectent simultanément à différents côtés. Il est conçu pour jouer le rôle de client en recevant les instructions de l'automate, ainsi que le rôle de client pour se connecter au robot en tant que contrôleur, afin de le mettre en marche et de le contrôler. Nous recevons les coordonnées sur les trois axes (x, y, z) et les transmettons à l'automate en tant que client OPC-UA, tout en les utilisant en tant que données opérationnelles pour l'Asset Administration Shells (AAS) en tant que serveur OPC-UA. Pour plus d'informations sur le programme contrôleur, se référer à l'Annexe 1.

L'effecteur est programmé séparément du contrôleur avec son propre capteur de proximité qui transmette la détection d'un objet à l'effecteur pour fermer les pinces.

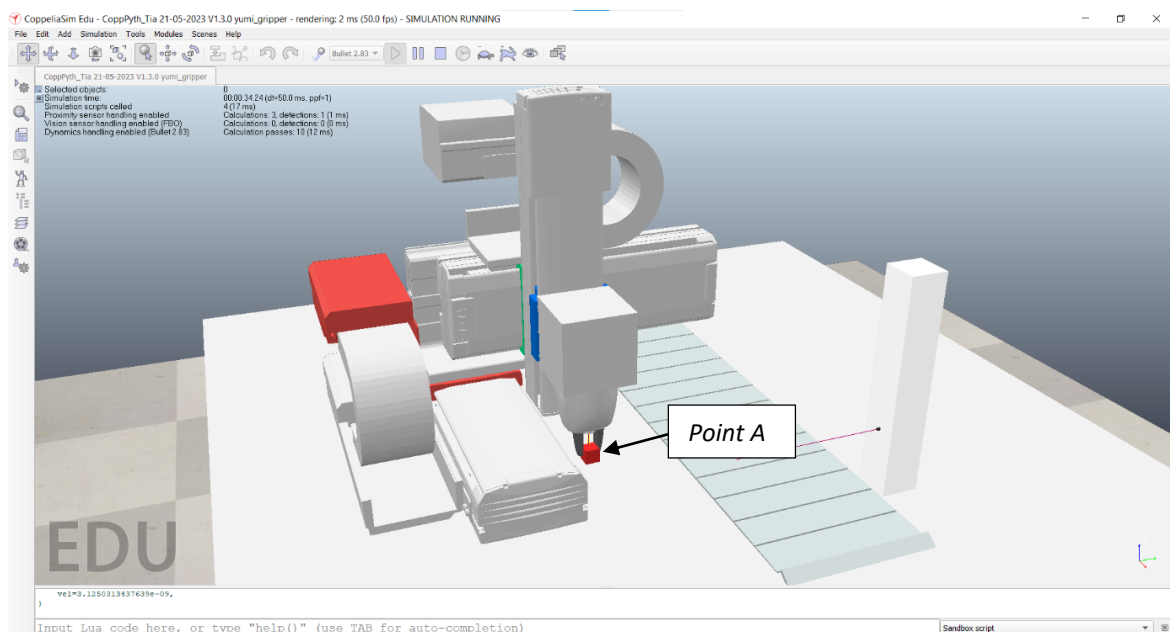


Figure IV.20 - La position de départ, *Point « A »*

Le déplacement des sliders est faite avec le changement de position de chaque 'un d'eux avec une vitesse déjà choisit. Il démarre du point « A » (Figure IV.20) pour se déplace au point « B »

```

Connected to remote API server
La Position de x : -0.0701248049736023
La Position de y : 0.17717516422271729
La Position de z : 1.2103095054626465
+-----+-----+-----+
|           x           |           y           |           z           |
+-----+-----+-----+
| -0.0701248049736023 | 0.17717516422271729 | 1.2103095054626465 |
+-----+-----+-----+
    
```

Figure IV.21 - Coordonnées des sliders (x,y,z) en temps réel sur python (contrôleur).

Le contrôleur robot sur python relève les coordonnées des sliders et les transmet au PLC ainsi que le AAS (Figure IV.21).

	Name	Data type	Start value	Monitor value	Retain	Accessible f...	Writa...	Visi...
1	Input							
2	task_over	Bool	false	FALSE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3	err_comm	Bool	false	FALSE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
4	Pos_X	Real	0.0	-0.07061872	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
5	Pos_Y	Real	0.0	0.1753028	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
6	Pos_Z	Real	0.0	1.211732	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
7	Trip_X	Bool	false	FALSE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
8	Trip_Y	Bool	false	FALSE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
9	Trip_Z	Bool	false	FALSE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
10	Output							
11	task_sp	String	"	'tacheA'	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
12	reset	Bool	false	FALSE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
13	vitesse	Int	0	1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
14	InOut				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
15	Static				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figure IV.22 - Table de bloc de données.

La position de chaque slider est transmise du robot vers le serveur OPC-UA ainsi que le PLC (Figure IV.22).

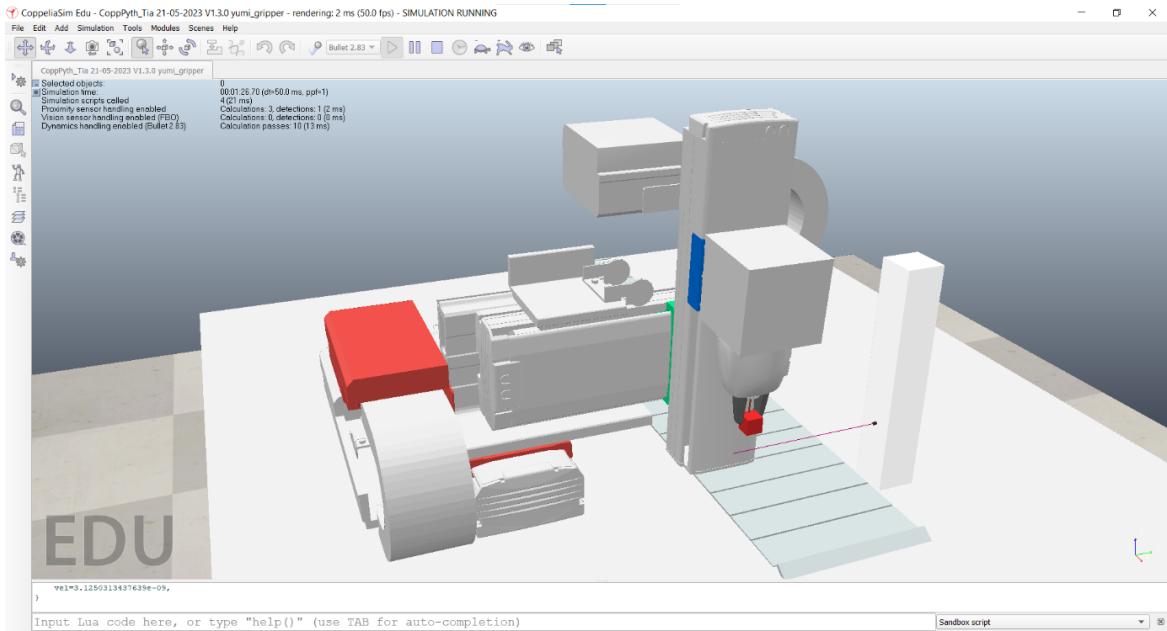


Figure IV.23 - Mouvement du Robot sur les trois axes

Lorsque le robot arrive à la position désirée, il commence à descendre jusqu'à le capteur de proximité le détecte, où en ce point (point « B ») il relâche l'objet et attende la consigne suivante (Figure IV.24).

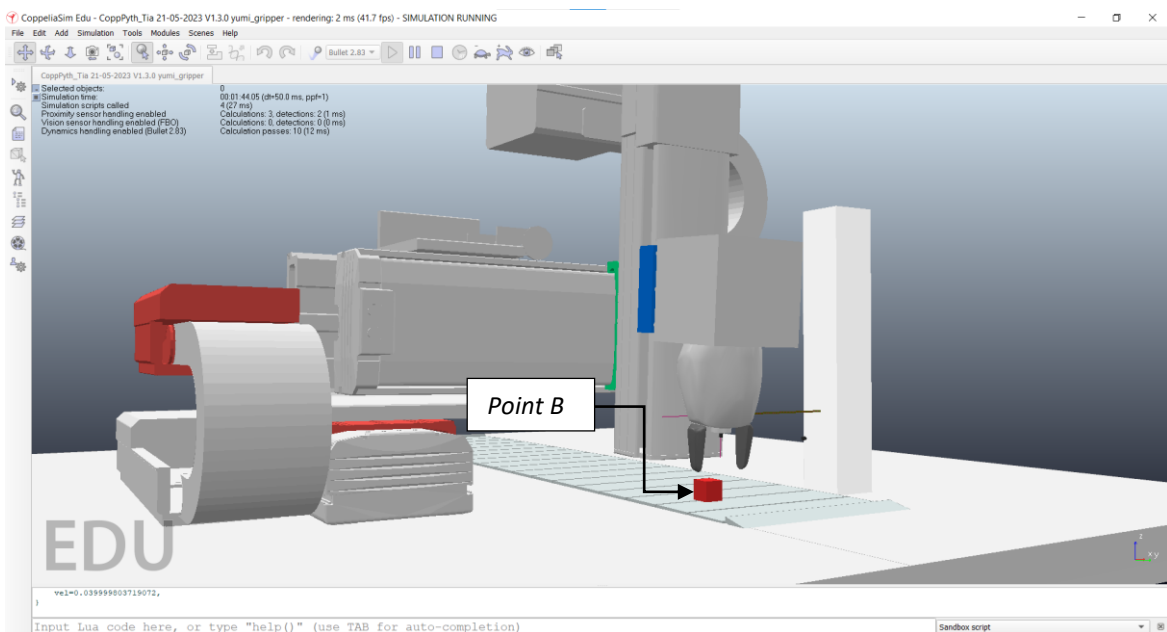


Figure IV.24 - La position d'arrivée, Point « B »

Un convoyeur démarre pour le transport de l'objet après que la valeur du capteur de proximité se change de Faux à Vrai. Si le bouton Reset est activé, le robot revient à sa position initiale.

Si un des sliders se rapproche dangereusement à la fin de sa course, une alarme d'urgence se déclenche et coupe la consigne envoyée par le PLC.

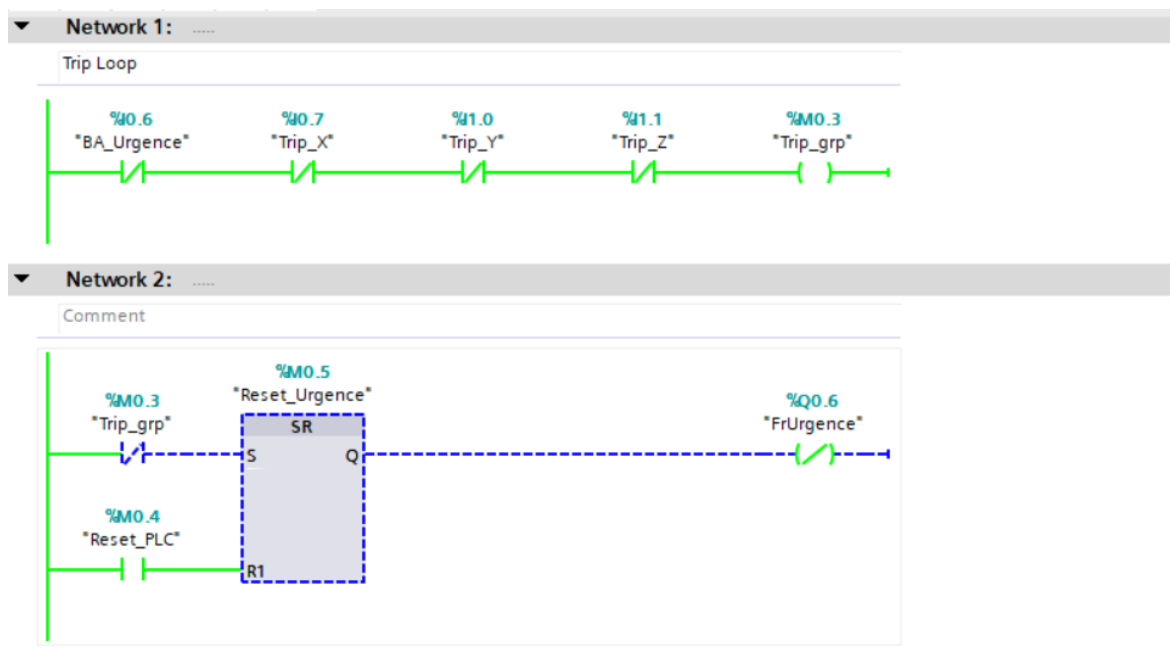


Figure IV.25 - La boucle de déclenchement d'urgence

IV.3) Partie II

IV.3.a) Eclipse Basyx

Eclipse BaSyx est la plate-forme open source pour la nouvelle génération de l'automatisation. C'est une plate-forme logicielle libre qui permet à toutes les parties intéressées, grandes et petites industries, instituts de recherche, universités et personnes intéressées, de participer et de façonner la quatrième révolution industrielle [95].



Il fournit des composants communs et réutilisables de l'Industrie 4.0 et un SDK qui prend en charge le développement de nouveaux composants logiciels Industrie 4.0 pour permettre le développement rapide de solutions Industrie 4.0 [95].

BaSyx fournit plusieurs Off-the-Shelf-Componant facile et prêt à l'utilisation. Ils peuvent être utilisés par programmation, comme un fichier jar exécutable ou comme un conteneur docker.

IV.3.b) Admin_shell_io by IDTA

C'est un processus d'enregistrement et une collection d'identifiants des asset administration shells actifs gérés par IDTA. Ce référentiel agit comme une solution rapide et légère pour gérer les éléments de l'espace de noms <https://admin-shell.io/>. On notera qu'il n'existe actuellement aucun site Web servant l'URL de l'espace de noms [94].

IV.3.c) Création du fichier AAS

Pour la création des Asset Administration Shells on peut utiliser l'Eclipse Basyx pour le programmer ou on peut utiliser l'AAS Package Explorer développé par IDTA.

IV.3.d) AAS Package Explorer

Le AAS Package Explorer est un outil application logicielle conçu pour explorer et interagir avec les packages AAS (Asset Administration Shell).

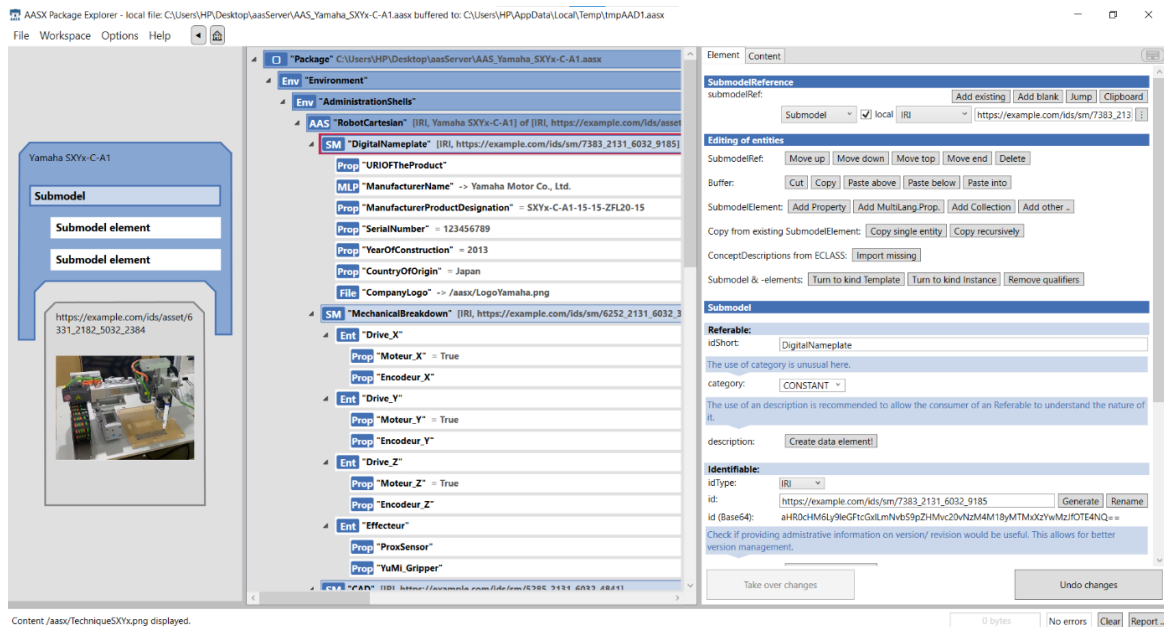


Figure IV.26 - AAS Package Explorer : édition des données du package AAS

Les packages AAS sont des structures de données standardisé qui sont utilisées pour représenter et échanger des informations sur les actifs industriels dans le contexte de l'industrie 4.0 et de l'Internet industriel des objets (IIoT).

On a créé le fichier package qui contient le AAS de notre robot à l'aide de l'interface fournie dans le Package explorer, en ajoutant notre AAS avec ses différents sous-modèles.

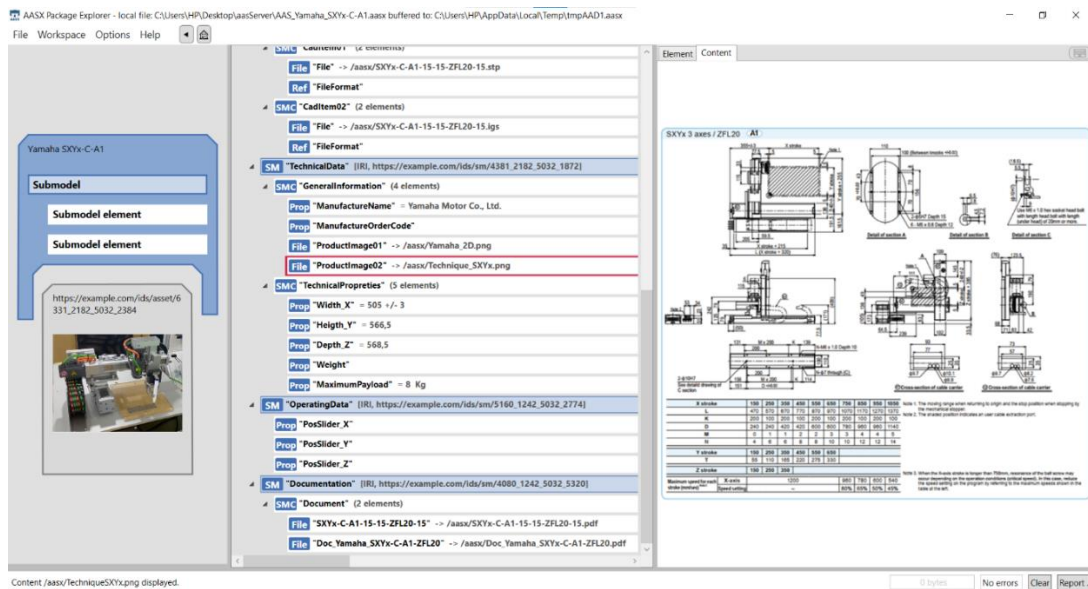
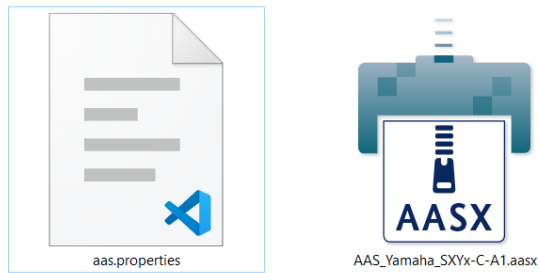


Figure IV.27 - Navigation du package AAS (Datasheet robot)

AAS Package Explorer permet aux utilisateurs de parcourir, de naviguer et d'inspecter le contenu des packages AAS et l'enregistrer format (.aasx). Il fournit des fonctionnalités telles que l'affichage de la structure AAS, l'accès aux métadonnées des ressources, l'exploration des relations entre les ressources et la gestion des données dans le package. Il permet de voir en temps réel les changements des données opérationnelles du robot (Figure IV.27).

Pour charger le fichier AAS dans un serveur, la création du fichier (.aasx) nécessite un fichier « propriétés » programmer d'une façon qui charge le AAS dans une source qui lui-même sera charger dans le serveur AAS.



Le fichier propriétés contient cette petite configuration si dessous :

```
# AAS Server configuration file
# #####
# Specifies the backend that loads the AAS and Submodels
# InMemory - does not persist AAS or submodels
aas.backend=InMemory
# #####
# Source
# Possible to load an AAS Environment from a file
aas.source=/Usr/share/config/AAS_Yamaha_SXYx-C-A1.aasx
```

IV.3.e) Docker

Docker est une plate-forme ouverte pour le développement, l'expédition et l'exécution d'applications. Il offre la possibilité d'empaqueter et d'exécuter une application dans un environnement peu isolé appelé conteneur.

L'isolation et la sécurité vous permettent d'exécuter plusieurs conteneurs simultanément sur un hôte donné. Les conteneurs sont légers et contiennent tout le nécessaire pour exécuter l'application, vous n'avez donc pas besoin de compter sur ce qui est actuellement installé sur l'hôte [95].

On crée les « Off the Shelves Components » avec la fenêtre de commande Windows « cmd » comme dans la figure si dessous :

```

C:\Windows\System32\cmd.exe
Microsoft Windows [version 10.0.19045.3086]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\HP\Desktop\basyx-applications-main\opc2aas>docker-compose up -d
[+] Running 6/6
  Container registrydb   Healthy              0.5s
  Container webui        Running              0.0s
  Container opcserver    Running              0.0s
  Container aasserverdb  Healthy              0.5s
  Container databridge  Running              0.0s
  Container opc2aas      Running              0.0s

C:\Users\HP\Desktop\basyx-applications-main\opc2aas>
    
```

Figure IV.28 - Creation des Off the Shelf Component

Après qu'ils s'affichent dans le Docker, on peut les démarrer, arrêter, supprimer ou voir ce que ces conteneurs abritent comme fichiers :

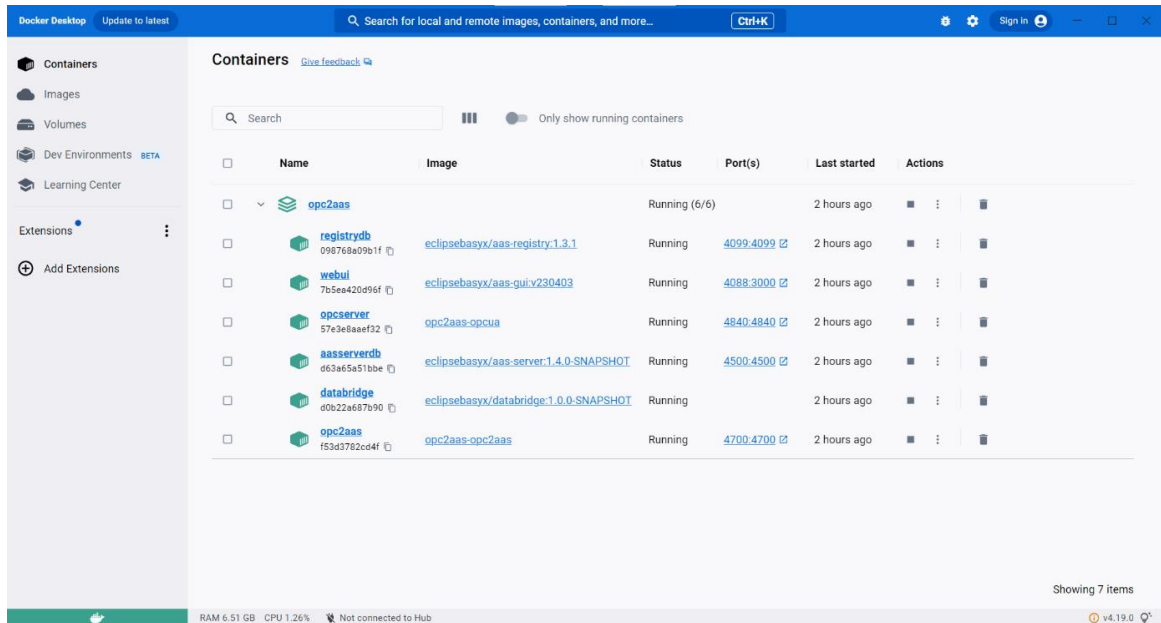


Figure IV.29 - Les conteneurs en état marchent

IV.3.f) Serveur AAS

Un serveur AAS (Asset Administration Shell) est un composant logiciel qui fournit un serveur AAS vide qui peut être utilisé pour héberger plusieurs AAS et sous-modèles. Il est utilisé dans le cadre Asset Administration Shell (AAS), qui est un ensemble de normes pour l'échange d'informations sur les actifs dans l'automatisation industrielle.

Il fournit une API REST qui peut être utilisée pour accéder à l'AAS et aux sous-modèles. L'API vous permet d'interroger, de créer, de mettre à jour et de supprimer des AAS et des sous-modèles. Il prend également en charge les protocoles OPC UA et MQTT.

Pour pouvoir créer un serveur AAS qui héberge notre fichier AAS, il faut tout d'abord tirer l'image du serveur dans un Docker avec la commande `docker pull eclipsebasyx/aas-server:1.4.0-SNAPSHOT` exécuté dans le cmd.

IV.3.g) **Registre AAS**

Un registre AAS (Asset Administration Shell) est une base de données qui stocke des informations sur les AAS. Le registre peut être utilisé pour :

- Découvrir les AAS : le registre peut être utilisé pour découvrir les SAA disponibles dans un réseau. Cela peut être utile si vous recherchez un AAS spécifique ou si vous souhaitez trouver tous les AAS associés à un actif particulier.
- Accéder aux AAS : le registre peut être utilisé pour accéder aux AAS. Cela peut être fait en utilisant l'API du registre ou en parcourant l'interface Web du registre.
- Gérer les AAS : le registre peut être utilisé pour gérer les AAS. Cela inclut des tâches telles que la création, la mise à jour et la suppression d'AAS.

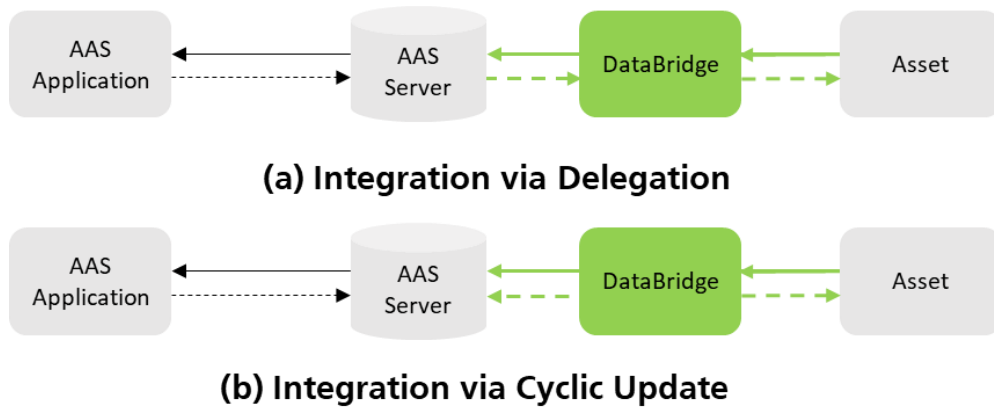
Le tirage du registre dans le Docker est fait par cette commande **`docker pull eclipsebasyx/aas-registry:1.3.1`**

IV.3.h) **DataBridge**

DataBridge prend en charge l'intégration de divers protocoles avec Asset Administration Shells. Les données peuvent être acquises à partir de différents points de terminaison, transformées et poussées dans SubmodelElements. En étant fourni en tant que composant prêt à l'emploi facile à utiliser sur DockerHub

Il prend en charge l'intégration de données avec une variété de protocoles tels que OPC UA, http / REST, MQTT, ActiveMQ ou Kafka. De plus, il permet la transformation des données fournies via les protocoles. Par exemple, les fichiers JSON peuvent être convertis en JSONata - un langage de transformation pour JSON - à l'aide d'expressions simples. Plus précisément, il prend en charge deux modèles d'intégration:

- Intégration cyclique : les données sont interrogées, transformées et intégrées dans l'AAS à intervalles réguliers ou en fonction des événements (-> Push).
- Délégation à la demande : à la demande du serveur AAS, un point de données est interrogé, transformé et renvoyé (-> Pull).



Flux des Données

Flux de Contrôle

Figure IV.30 - Fonctionnement du DataBridge avec le serveur et l'Asset [96]

Le DataBridge peut être tiré dans le docker avec la commande `docker pull eclipsebasyx/databridge:1.0.0-SNAPSHOT` dans le cmd.

Le DataBridge[97] qu'on a configuré est sensé de communiquer avec un serveur OPC-UA dans le contrôleur robot pour acquérir les données du robot notamment les coordonnées cartésiennes et les envoyer sur une route de fichiers de configuration du DataBridge interconnecté, ces données sont envoyées à travers le pont vers le fichier AAS dans le serveur AAS et les affiches dans un GUI sur <https://localhost:4088>.

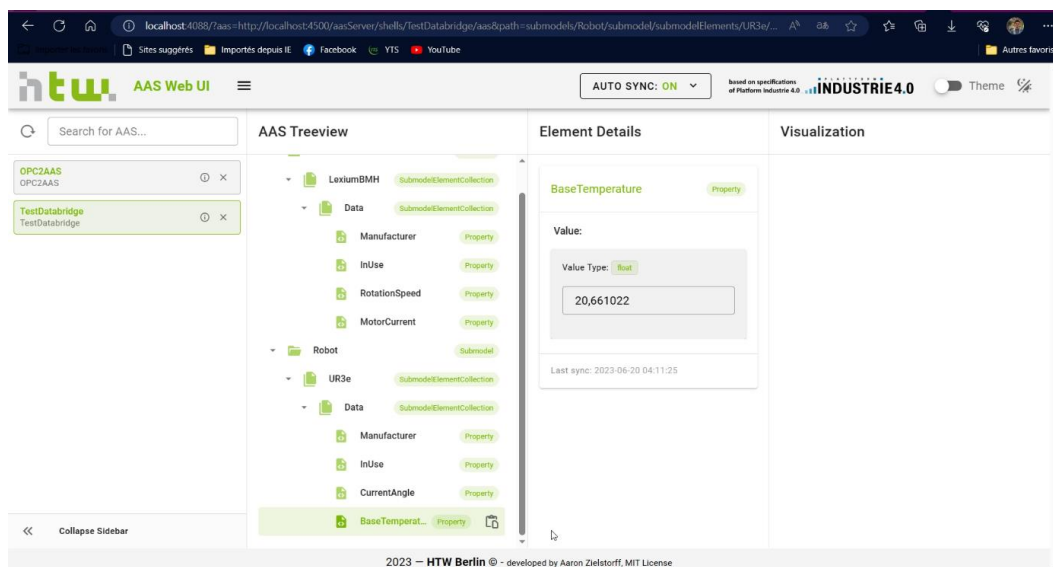


Figure IV.31 - Exemple d'affichage d'interface d'un AAS avec des données opérationnelles qui change en temps réel sur <https://localhost:4088>

IV.4) Conclusion

Dans ce chapitre nous avons développé un module de production cyber-physique, à l'aide de plusieurs programmes et interfaces développés. On a réussi à développer un contrôleur de robot qui communique avec un automate programmable virtuel, permettant ainsi une interaction fluide et synchronisée entre les deux. En prenant en considération la collection des données pour une autre utilisation. Pour cela, nous avons eu recours à plusieurs outils de développement, à savoir les outils de conception 3D, les automates programmables industriels, les protocoles de communications industrielles telles que les sockets TCP/IP, l'OPC UA ainsi que différents langages de programmation comme le « Lua » au sein du CoppeliaSim, et le « python » pour la programmation du contrôleur robot.

Le commissioning virtuel via la technique SIL s'est déroulé avec succès et nous a permis de rectifier les erreurs dans la simulation du modèle 3D, de mettre au point le code python qui contrôle le robot, ainsi que la logique PLC.

La seconde partie a été dédiée à la création du jumeau digital basé sur l'AAS. On a pu créer un AAS de type 1 (statique) transmissible en tant que fichier de format « .aasx » et de charger les informations et données nécessaires durant le cycle de vie de produit, partant de la plaque signalétique numérique (Digital Nameplate), jusqu'au chargement des fiches techniques et manuels nécessaires à la mise en service et d'opération du Robot, dont la disponibilité est d'une grande importance durant le cycle de vie de l'équipement.

Afin de profiter pleinement de la puissance d'un jumeau digital, il est primordial de mettre à disposition les données opérationnelles aux différentes applications de traitement et d'analyse de ces données, notamment via l'intelligence artificielle. Ceci a un impact important sur l'optimisation des conditions opératoires de l'équipement, ainsi qu'à sa maintenance. Pour cela on a procédé à la création d'un sous-modèle dédiée aux données opératoires au niveau de notre AAS, malheureusement, on a rencontré des difficultés quand-a-la connexion a notre ressource.

L'objectif était de construire et de configurer un DataBridge qui collecte à partir du serveur OPC-UA les valeurs opérationnelles du robot qui les envoie à son propre AAS pour créer un Digital Twin qui fonctionne. Cela n'a pas été possible, dans le cadre de notre travail de fin d'études, à cause de la limitation du temps imparti, et des difficultés rencontrés durant les tests.

CONCLUSION

GENERALE

V. CONCLUSION GENERALE

Au cours de notre projet dont le but était d'étudier et de concevoir d'un module de production cyber-physique – cas d'étude d'un Robot Cartésien, nous avons abordé la quasi-totalité des aspects liés à ce type de travail.

En fait, nous avons décidé de diviser le travail en deux parties, la première partie est basée sur la préparation d'une étude dont le but était d'utiliser un modèle 3D du robot industriel comme cas d'étude lors du développement et de la validation de la partie cyber du CPPS, pour cela un robot cartésien linéaire de type "YAMAHA-SXYx-C-A1" qui a été choisi et son contrôleur, où nous nous sommes appuyés sur python. Afin d'intégrer le robot, nous devons connecter son contrôleur à l'Automate Programmable industriel "SIEMENS ET200SP PLC". Au niveau logiciel, ce projet nous a mis en contact direct avec des langages de programmation puissants tels que Python, ainsi que les propres langages Lua de Robot et Ladder du PLC. Nous incluons le protocole de communication OPC UA qui est le protocole de communication le plus utilisé dans l'Industrie 4.0. A la fin de cette section, nous effectuons le processus de mise en service par défaut, qui réunit chaque fournisseur au préalable. En ce qui concerne la deuxième section, à travers laquelle nous nous concentrons sur la création d'un jumeau numérique sur la base d'AAS, qui nous permet de télécharger les informations et les données nécessaires pendant le cycle de vie du produit. Afin d'utiliser pleinement la puissance du jumeau numérique.

De tout cela, nous constatons que les systèmes de production cyber-physiques (CPS) représentent une avancée majeure dans le domaine de la fabrication et de l'automatisation industrielle. Ces systèmes combinent des éléments physiques, tels que des machines et des robots, avec des composants numériques, tels que des logiciels, des algorithmes et des réseaux de communication. Il vise à améliorer l'efficacité, la flexibilité et la qualité des processus de production en exploitant les capacités dans le domaine de l'Industrie 4.0.

Malgré les difficultés et les défis, les systèmes de production cyber-physiques offrent un énorme potentiel pour améliorer l'efficacité et la compétitivité des entreprises. Avec une mise en œuvre appropriée et une gestion efficace, CPS peut transformer les processus de production, inaugurant une nouvelle ère de fabrication intelligente et durable.

BIBLIOGRAPHIE ET RÉFÉRENCES

- [1] Sunyaev, A. (2020). *Cloud computing*. In *Internet computing*, pages 195–236. Springer.
- [2] Marcon, P., Zezulka, F., Vesely, I., Szabo, Z., Roubal, Z., Sajdl, O., Gescheidtova, E., and Dohnal, P. (2017). *Communication technology for industry 4.0*. In 2017 Progress In Electromagnetics Research Symposium-Spring (PIERS), pages 1694–1697. IEEE.
- [3] Abdel-Basset, M., Mohamed, M., and Chang, V. (2018). Nmcda: A framework for evaluating cloud computing services. *Future Generation Computer Systems*, 86:12–29.
- [4] <https://www.3dnatives.com/technologies-3d/#!>
- [5] Madakam, S., Lake, V., Lake, V., Lake, V., et al. (2015). *Internet of things (IOT): A literature review*. *Journal of Computer and Communications*, 3(05):164.
- [6] Dhall, R. and Solanki, V. (2017). *An IOT based predictive connected car maintenance*. *International Journal of Interactive Multimedia & Artificial Intelligence*, 4(3).
- [7] Canizo, M., Onieva, E., Conde, A., Charramendieta, S., and Trujillo, S. (2017). *Realtime predictive maintenance for wind turbines using big data frameworks*. In 2017 IEEE International Conference on Prognostics and Health Management (ICPHM), pages 70–77. IEEE.
- [8] Wan, J., Tang, S., Li, D., Wang, S., Liu, C., Abbas, H., and Vasilakos, A. V. (2017). *A manufacturing big data solution for active preventive maintenance*. *IEEE Transactions on Industrial Informatics*, 13(4):2039–2047.
- [9] Mueller, E., Chen, X.-L., and Riedel, R. (2017). *Challenges and requirements for the application of industry 4.0: a special insight with the usage of cyber-physical system*. *Chinese Journal of Mechanical Engineering*, 30(5):1050–1057.
- [10] Glaessgen, E. and Stargel, D. (2012). *The digital twin paradigm for future NASA and us air force vehicles*. In 53rd AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics and materials conference 20th AIAA/ASME/AHS adaptive structures conference 14th AIAA, page 1818.
- [11] Bengio, Y., Lamblin, P., Popovici, D., and Larochelle, H. (2007). *Greedy layer-wise training of deep networks*. In Schölkopf, B., Platt, J. C., and Hoffman, T., editors, *Advances in Neural Information Processing Systems 19*, pages 153–160. MIT Press.
- [12] Bhardwaj, S., Jain, L., and Jain, S. (2010). *Cloud computing: A study of infrastructure as a service (iaas)*. *International Journal of engineering and information Technology*, 2(1) :60–63.

- [13] Schleich, B., Anwer, N., Mathieu, L., and Wartzack, S. (2017). *Shaping the digital twin for design and production engineering*. CIRP Annals, 66(1):141–144.
- [14] Rosen, R., Von Wichert, G., Lo, G., and Bettenhausen, K. D. (2015). *About the importance of autonomy and digital twins for the future of manufacturing*. IFAC-PapersOnLine, 48(3):567–572.
- [15] Söderberg, R., Wärmefjord, K., Carlson, J. S., and Lindkvist, L. (2017). *Toward a digital twin for real-time geometry assurance in individualized production*. CIRP Annals, 66(1):137–140.
- [16] Tao, F., Zhang, M., Liu, Y., and Nee, A. (2018b). *Digital twin driven prognostics and health management for complex equipment*. CIRP Annals, 67(1) :169–172.
- [17] Jana AL HAJ ALI – Doctorant, *Jumeau numérique pour l'interopérabilité cognitive des CPS, Ingénierie des Systèmes Eco-Techniques*, Campus sciences - Bât. Henri Poincaré.2023
- [18] Magargle, R., Johnson, L., Mandloi, P., Davoudabadi, P., Kesarkar, O., Krishnaswamy, S., Batteh, J., and Pitchaikani, A. (2017). *A simulation-based digital twin for modeldriven health monitoring and predictive maintenance of an automotive braking system*. In Proceedings of the 12th International Modelica Conference, Prague, Czech Republic, May 15-17, 2017, number 132, pages 35–46. Linköping University Electronic Press.
- [19] Wang, X. V. and Wang, L. (2019). *Digital twin-based weee recycling, recovery and remanufacturing in the background of industry 4.0*. International Journal of Production Research, 57(12):3892–3902.
- [20] Qi, Q., Tao, F., Zuo, Y., and Zhao, D. (2018). *Digital twin service towards smart manufacturing*. Procedia Cirp, 72:237–242.
- [21] M. Kunath and H. Winkler, “*Integrating the Digital Twin of the manufacturing system into a decision support system for improving the order management process*,” Procedia CIRP, vol. 72, pp. 225-231, 2018
- [22] BNP PARIBAS , « *Les Etats-Unis misent gros sur la cyber-physique* », Paris, L'Atelier, 3septembre 2008.
- [23] Edward Lee, *Systèmes Cyber physiques : Conception défis*,23 janvier, 2008.
- [24] Gilles Lasnier, Janette Cardoso, Claire Pagetti, Pierre Siron, « *Environnement de coopération de simulation pour la conception de systèmes cyber-physiques* », Rennes,

- Modélisation des systèmes réactifs (MSR), 2013 (consulté le 1er mai 2014).
- [25] Claus Hilger, « *La 4ème révolution industrielle est en marche* », Tec news (consulté le 1er mai 2014).
- [26] <https://www.clicours.com/les-systemes-cyber-physiques/> (consulté le 19 juin 2023).
- [27] https://fr.wikipedia.org/wiki/Syst%C3%A8me_cyber-physique (consulté le 19 juin 2023).
- [28] Sylvain Andre, « *MDA (Model Driven Architecture) principes et états de l'art* », 25p, Mémoire d'ingénieur en informatique, C.N.A.M. Centre d'enseignement de Lyon, 2004.
- [29] Shi, J., Wan, J., Yan, H., Suo, H., 2011. *A survey of cyber-physical systems*, in: *Wireless Communications and Signal Processing (WCSP)*, 2011 International Conference on. IEEE, pp. 1–6.
- [30] VDI/VDEGesellschaft Mess und Automatisierungstechnik (GMA). *Systèmes cyber-physiques : Chancen und nutzen aus sicht der Automation*, Thesen und Handlungsfelder, avril 2013 : 9.
- [31] Brettel M., Friederichsen N., Keller M., Rosenberg N. *Comment la virtualisation, la décentralisation et la création de réseaux modifient le paysage de la fabrication : une perspective de l'industrie 4.0*. Revue internationale des sciences, de l'ingénierie et de la technologie 2014 ; 8/1 : 3744.
- [32] Hatvany J., *L'utilisation efficace des informations déficientes*, CIRP Annals – Manufacturing Technology 2013, 32/1 : 423425.
- [33] Hatvany J. *Intelligence et coopération dans les systèmes de fabrication hétérarchiques*. Robotique et fabrication intégrée par ordinateur 1985 2/2 : 101104.
- [34] Monostori L. *Un pas vers la fabrication intelligente : Modélisation et surveillance des processus de fabrication par le biais de réseaux de neurones artificiels*. CIRP Annales Technologie de fabrication 1993 ; 42/1 : 485488.
- [35] Monostori L., Márkus A., Van Brussel H., Westkämper E. *Approches d'apprentissage automatique de la fabrication*. CIRP Annales Technologie de fabrication 1996 ; 45/2 : 675712.
- [36] Ueda K., Vaario J., Ohkura KH. *Modélisation des systèmes de fabrication biologique pour la reconfiguration dynamique*. CIRP Annales Technologie de fabrication 1997 ; 46/1 : 343–346.
- [37] Koren Y., Heisel Z., Jovane F., Moriwaki M., Pritschow G., Ulsoy G., Van Brussel H.

- Systèmes de fabrication reconfigurable*. CIRP Annales Technologie de fabrication 1999 ; 48/2 : 527–540.
- [38] Maropoulos PG. *Technologie d'entreprise numérique—Définir les perspectives et les priorités de recherche*. Proc. Du 1er séminaire CIRP sur la technologie d'entreprise numérique, Durham, RoyaumeUni, 2002 ; Partie V : 3–12.
- [39] Kádár B., Lengyel A., Monostori L., Suginishi Y., Pfeiffer A., Nonaka Y. *Amélioration du contrôle des structures de production complexes par un couplage étroit des mondes numérique et physique*. CIRP Annales Technologie de fabrication 2010 ; 59/1 : 437440.
- [40] Kádár B., Terkaj W., Sacco M. *Usine virtuelle sémantique prenant en charge la modélisation et l'évaluation interopérables des systèmes de production*. CIRP Annales Technologie de fabrication 2013 ; 52/1 : 443446.
- [41] Van Brussel H., Wyns J., Valckenaers P., Bongaerts L., Peeters P. *Architecture de référence pour les systèmes de fabrication holonique : PROSA*. Ordinateurs dans l'industrie 1998 ; 37 :255–274.
- [42] Bongaerts L., Monostori L., McFarlane D., Kádár B. *Hiérarchie dans le contrôle d'atelier distribué*. Ordinateurs dans l'industrie 2000 ; 43/2 : 123–137.
- [43] Valckenaers P., Van Brussel H. *Holonic manufacturing execution systems*. CIRP Annales Technologie de fabrication 2005 ; 54/1 : 427432.
- [44] Monostori L., Váncza J., Kumara SRT. *Systèmes à base d'agents pour la fabrication*. Annales CIRP Technologie de fabrication 2006 ; 55/2 : 697720.
- [45] Monostori L., Ueda K. *Conception de systèmes adaptatifs complexes : Introduction*. Informatique d'ingénierie avancée 2006 ; 20/3 : 223225.
- [46] Ueda K., Márkus A., Monostori L., Kals HJJ., Arai T. *Méthodologies de synthèse émergente pour la fabrication*. CIRP Annales Technologie de fabrication 2001 ; 50/2 : 535551.
- [47] Wiendahl HP., Lutz S. *Production en réseaux*. CIRP Annales Technologie de fabrication 2002 ; 51/2 : 573–586.
- [48] Wiendahl HP., ElMaraghy HA., Nyhuis P., Zaeh MF., Wiendahl, HH., Duffie N., Brieke M. *Fabrication variable Classification, conception et fonctionnement*. CIRP Annales Technologie de fabrication 2007 ; 56/2 : 783–809.

- [49] Váncza J., Monostori L., Lutters E., Kumara SR., Tseng M., Valckenaers P., Van Brussel H. *Coopérative, entreprises manufacturières réactives*. CIRP Annales Technologie de fabrication 2011 ; 60/2 : 797820.
- [50] Tolio T., Ceglarek D., ElMaraghy HA., Fischer A., Hu SJ., Laperrière L., Newman ST, Váncza J. ESPÈCES – *Coévolution des produits, procédés et systèmes de production*, CIRP Annals – Manufacturing Technology 2010 ; 59/2 : 672–693.
- [51] Monostori L., Csáji, BCs. *Contrôle dynamique stochastique de la production par programmation neurodynamique*. CIRP Annales Technologie de fabrication 2006 ; 55/1 : 473–478.
- [52] Givehchi O., Jasperneite J. *Services d'automatisation industrielle dans le cadre du Cloud : premières expériences*. Actes du Jahreskolloquium Kommunikation in der Automation – KommA, Magdebourg, 1314 novembre 2013, Magdebourg, Allemagne : 10.
- [53] Bendjelloul, A., Mihoubi, B., Gaham, M., Moufid, M., & Bouzouia, B. (2022). *A framework for an effective virtual commissioning of agent-based cyber-physical production systems integrated into manufacturing facilities*.
- [54] Adama Arama; Laura Laguna Salvadó; Éric Villeneuve; *Christophe Merlo*. *Modélisation de CPPS : proposition d'un modèle générique*. Conférence : Conférence Internationale Génie Industriel QUALITAAt, Grenoble (France). April 2021.
- [55] Chi G. Lee, Sang C. Park. *Survey on the virtual commissioning of manufacturing systems*. (DOI: 10.7315/JCDE.2014.021). July 2014.
- [56] <https://www.orange-business.com/fr/magazine/connectivite-levier-industrie-40> (consulté le 24 juin 2023).
- [57] <https://industry40.co.in/rami-reference-architecture-model-industry-4-0/> (consulté le 24 juin 2023).
- [58] Kagermann, H., Helbig, J., Hellinger, A., and Wahlster, W. (2013). *Recommendations for implementing the strategic initiative INDUSTRIE 4.0: Securing the future of German manufacturing industry*; final report of the Industrie 4.0 Working Group. Forschungsunion.
- [59] Plamen Vasilev, Vasil Metodiev. *I4.0 Component Models based on IEC/EN 62264*. INTERNATIONAL SCIENTIFIC JOURNAL "INDUSTRY 4.0". YEAR VI, ISSUE 5, P.P. 181-184 (2021).

- [60] https://fr.wikipedia.org/wiki/Intelligence_artificielle (consulté le 24 juin 2023).
- [61] <https://www.humarobotics.com/blog/la-robotique-collaborative/> (consulté le 25 juin 2023).
- [62] <https://safetyculture.com/fr/themes/robot-collaboratif/> (consulté le 25 juin 2023).
- [63] https://fr.wikipedia.org/wiki/Jumeau_num%C3%A9rique (consulté le 25 juin 2023).
- [64] <https://www.ibm.com/fr-fr/topics/what-is-a-digital-twin> (consulté le 25 juin 2023).
- [65] https://fr.wikipedia.org/wiki/Robot#cite_note-1 (consulté 21 juin 2023).
- [66] <https://www.atilf.fr/ressources/grand-public/mots-de-la-science/cadres/mots/tlfi/robot.htm> (consulté 12 avril 2019)
- [67] <https://fr.wikipedia.org/wiki/Robotique> (consulté 21 juin 2023).
- [68] Mark W. Spong, Seth Hutchinson, and M. Vidyasagar. *Robot Modeling and Control*. First Edition
- [69] Dr. A. Hadjaissa, n book: *Automates Programmables Industriels Description et programmation*, Publisher: UATL-Dpt-ELN, 2019
- [70] <http://www.robotpark.com/academy/all-types-of-robots/stationary-robots/cartesian-gantry-robots/> (consulté 21 juin 2023)
- [71] L. Monostori et al. *Cyber-physical systems in manufacturing / CIRP Annals - Manufacturing Technology* 65 (2016) 621–641.
- [72] Xuan Wu · VirginieGoepp · A. Siadat, «*Cyber-Physical Production Systems: A Review of Design and Implementation Approaches*»
- [73] <https://www.i40.ovgu.de/en/Asset+Administration+Shell.html>.(Consulté 26 juin 2023)
- [74] Alexander Belyaev, Christian Diedrich,AAS Reference Modelling. VWS-Referenzmodellierung - *ExemplarischeModellierung einer fertigungstechnischen Anlage mit AASX Package Explorer auf Basis des VWS-Metamodells"*, publié par Platt form Industrie 4.0 en avril 2021.
- [75] <https://www.kollmorgen.com/fr-fr/blogs/quelle-est-la-fonction-dun-contr%C3%B4leur-de-mouvement> (consulté 26 juin 2023).
- [76] https://fr.wikipedia.org/wiki/Automate_programmable_industriel (consulté 26 juin 2023).
- [77] https://wiki.eclipse.org/BaSyx/_/Documentation/_/Components/_/DataBridge

- (consulté 26 juin 2023).
- [78] <https://webshop.robotics.abb.com/fr/catalog/product/view/id/396/s/onrobot-2fg-gripper/> (16/06/2023 -21:30).
- [79] VDI/VDE. *ZVEI (2015) Reference Architecture Model Industrie 4.0 (RAMI4.0)*. VDI/VDE, ZVEI, Dusseldorf, Germany, Status Report.
- [80] <https://solidedge.siemens.com/es/partner/cad-models/> (consulté 14 juin 2023).
- [81] <https://www.sculpteo.com/en/glossary/solidworks-definition/> (consulté 14 juin 2023).
- [82] http://wiki.ros.org/sw_urdf_exporter (consulté 16 juin 2023).
- [83] <https://www.coppeliarobotics.com/> (consulté 14 juin 2023).
- [84] L. H. Figueiredo, C. S. Souza, M. Gattass et L. C. Coelho, « *Geração de interfaces para captura de dados sobre desenhos* », V SIBGRAPI, 1992, pp. 169–175.
- [85] <https://webshop.robotics.abb.com/fr/catalog/product/view/id/396/s/onrobot-2fg-gripper/> (consulté 16 juin 2023).
- [86] Dr. A. Hadjaissa, « *Automates Programmables Industriels Description et programmation* », Publisher: UATL-Dpt-ELN, 2019
- [87] <https://mall.industry.siemens.com/mall/en/WW/Catalog/Products/10239949> (consulté 16 juin 2023).
- [88] Logiciel TIA Portal v17 – Description de la CPU 1512SP-1 PN
- [89] Letácio Lins, *The Complete Practical Guide to Siemens Tia Portal Programming*, solisplc.com
- [90] Integral System, *QU'EST-CE QUE LE PROTOCOLE OPC-UA ? UN PROTOCOLE DANS L'AUTOMATISM*
- [91] <https://www.unified-automation.com/products/development-tools/uaexpert.html> (consulté 19 juin 2023).
- [92] Alem Abdelbasset & Rahali Hichem, *Implémentation des modules de marche d'un robot bipède NAO sur V-rep*. (2019)
- [93] <http://www.coppeliarobotics.com/helpFiles/index.html> (consulté 18 juin 2023).
- [94] <https://github.com/admin-shell-io> (consulté 19 Mai 2023).
- [95] <https://docs.docker.com/get-started/overview/> (consulté 21 juin 2023).
- [96] https://wiki.eclipse.org/BaSyx/_/Scenarios/_/Device_Integration (consulté le 21 juin 2023)

Annexe

Des Programmes

Annexe 1.

```
# Python programme for Master 2 PFE 2023
# developement d'un module cyber-physique
# TAHI_Samy / LAROUCI_Billel
# coding=utf-8
# Insert in a script in Coppelia
# simRemoteApi.start(19999)

try:
    import sim
except:
    print ('error')
import math
import time
import sys
import struct
import threading
from opcua import Client
from opcua import Server
from opcua import ua
from prettytable import PrettyTable

sys.path.insert(0, "..")
print('')
print ('Program started')
print('')

marche_rc = True

if __name__ == "__main__":

    #connect to PLC Tia_portal
    client = Client("opc.tcp://192.168.0.1:4840")

    #connect to Coppeliasim
    client.connect()

    #just in case, close all opened connections
    sim.simxFinish(-1)

    # Get the client ID
    clientID = sim.simxStart('127.0.0.1',19999,True,True,5000,5)

    if clientID != -1:

        while marche_rc == True:
            #marche_rc = input()

            # Client has a few methods to get proxy to UA nodes that should always be in
            # address space such as Root or Objects
            root = client.get_root_node()
            print("Objects node is: ", root)
```

```

#node objects have methods to read and write node attributes as well as browse or
  populate address space
print("Children of root are: ", root.get_children())

#Receiving data from Tia Portal
var_X = client.get_node('ns=3;s="Block_1_DB"."Pos_X"')
Pos_X = var_X.get_value()
print('La valeur de la position x est : ',Pos_X)

var_Y = client.get_node('ns=3;s="Block_1_DB"."Pos_Y"')
Pos_Y = var_Y.get_value()
print('La valeur de la position y est : ',Pos_Y)

var_Z = client.get_node('ns=3;s="Block_1_DB"."Pos_Z"')
Pos_Z = var_Z.get_value()
print('La valeur de la position z est : ',Pos_Z)

#Var du trip déclanchement
var_tripX = client.get_node('ns=3;s="Block_1_DB"."Trip_X"')
Trip_X = var_tripX.get_value()

var_tripY = client.get_node('ns=3;s="Block_1_DB"."Trip_Y"')
Trip_Y = var_tripY.get_value()

var_tripZ = client.get_node('ns=3;s="Block_1_DB"."Trip_Z"')
Trip_Z = var_tripZ.get_value()

#Get permission for tacheA
tachA_perm = client.get_node('ns=3;s="tacheA_Recopie"')
PermissionA_PLC = tachA_perm.get_value()

#Get permission for tacheB
tachB_perm = client.get_node('ns=3;s="tacheB_Recopie"')
PermissionB_PLC = tachB_perm.get_value()

#Get task_sp variable from Tia Portal
task_sp = client.get_node('ns=3;s="Block_1_DB"."task_sp"')
pos_sp = task_sp.get_value()
print('La tâche sélectionner est : ',pos_sp)

#Get reset variable from Tia Portal
reset = client.get_node('ns=3;s="Block_1_DB"."reset"')
rst = reset.get_value()
print(' le reset est : ',rst)

#Get task_over variable from Tia Portal
over = client.get_node('ns=3;s="Block_1_DB"."task_over"')
task_over = over.get_value()

#Get err_comm variable from Tia Portal
err_var_comm = client.get_node('ns=3;s="Block_1_DB"."err_comm"')
err_comm = err_var_comm.get_value()

#Get vitesse variable from Tia Portal
var_vitesse = client.get_node('ns=3;s="Block_1_DB"."vitesse"')
vit = var_vitesse.get_value()

```

```

print('La vitesse selectionner est : ',vit)

print ('Program coppelia started')

#Get robot and sliders handles

Robotname = 'Yamaha SXYx-C-A1'
position_x = 'x'
position_y = 'y'
position_z = 'z'

sim.simxStartSimulation(clientID, sim.simx_opmode_one-shot_wait)
print ('Connected to remote API server')

sim.simxAddStatusBarMessage(clientID, 'Fonctionnement...',sim.simx_opmode_one-shot_
                             wait)
time.sleep(0.02)

#collect handles
[err, Robot] = sim.simxGetObjectHandle(clientID, Robotname,
                                       sim.simx_opmode_one-shot_wait)
[err, on_x] = sim.simxGetObjectHandle(clientID, position_x,
                                       sim.simx_opmode_one-shot_wait)
[err, on_y] = sim.simxGetObjectHandle(clientID, position_y,
                                       sim.simx_opmode_one-shot_wait)
[err, on_z] = sim.simxGetObjectHandle(clientID, position_z,
                                       sim.simx_opmode_one-shot_wait)

# Create data stream
[err, PositionRobot] = sim.simxGetObjectPosition(clientID, Robot, -1,
                                                  sim.simx_opmode_streaming)
[err, Position_slider_x] = sim.simxGetObjectPosition(clientID, on_x, -1,
                                                      sim.simx_opmode_streaming)
[err, Position_slider_y] = sim.simxGetObjectPosition(clientID, on_y, -1,
                                                      sim.simx_opmode_streaming)
[err, Position_slider_z] = sim.simxGetObjectPosition(clientID, on_z, -1,
                                                      sim.simx_opmode_streaming)

time.sleep(2)

# Collect robot and target data
[err, [xt,yt,zt]] = sim.simxGetObjectPosition(clientID, Robot, -1,
                                              sim.simx_opmode_buffer)
[err, [xx,yx,zx]] = sim.simxGetObjectPosition(clientID, on_x, -1,
                                              sim.simx_opmode_buffer)
[err, [xy,yy,zy]] = sim.simxGetObjectPosition(clientID, on_y, -1,
                                              sim.simx_opmode_buffer)
[err, [xz,yz,zz]] = sim.simxGetObjectPosition(clientID, on_z, -1,
                                              sim.simx_opmode_buffer)

#sim.simxAddStatusBarMessage(clientID, 'La Position de x : '+str(xt) + ' La
Position de y : '+str(yt) + ' La Position de z : '+str(zt),
sim.simx_opmode_one-shot_wait)

```

```

sim.simxAddStatusbarMessage(clientID, 'La Position de x : '+str(xx) ,
                             sim.simx_opmode_onehot_wait)
sim.simxAddStatusbarMessage(clientID, 'La Position de y : '+str(yy) ,
                             sim.simx_opmode_onehot_wait)
sim.simxAddStatusbarMessage(clientID, 'La Position de z : '+str(zz) ,
                             sim.simx_opmode_onehot_wait)

#print position du slider X sur axe X
print('La Position de x : ', str(xx))

#print position du slider Y sur axe Y
print('La Position de y : ',str(yy))

#print position du slider Z sur axe Z
print('La Position de z : ',str(zz))

#Define the table structure
table = PrettyTable()
table.field_names = ["x","y","z"]

# Add some data to the table
table.add_row([str(xx), str(yy), str(zz)])

#Print the table
print(table)

#Get the handles of the objects to move
_, x = sim.simxGetObjectHandle(clientID, 'x', sim.simx_opmode_onehot_wait)
_, y = sim.simxGetObjectHandle(clientID, 'y', sim.simx_opmode_onehot_wait)
_, z = sim.simxGetObjectHandle(clientID, 'z', sim.simx_opmode_onehot_wait)
_, y_body = sim.simxGetObjectHandle(clientID, 'y_body_respondable',
                                     sim.simx_opmode_onehot_wait)

#Set the condition for the desired positions
if pos_sp == "tacheA":
    Cont_x = 0.15494
    Cont_y = 0.35129
    Cont_z = 1.0770
    Cont_z_init = 1.2106

if pos_sp == "tacheB":
    Cont_x = 0.03
    Cont_y = 0.27627
    Cont_z = 1.1356

#Set the condition for the desired positions
if vit == 1 :
    Pas = 0.001
    Pas_rst = 0.001
if vit == 2 :

```

```

    Pas = 0.01

# Move the objects incrementally until they reach the desired positions
Pas = 0.001
Pas_rst = 0.001
Pas_inv = -0.001
_, pos_x = sim.simxGetObjectPosition(clientID, x, -1,
                                     sim.simx_opmode_oneshot_wait)
_, pos_y = sim.simxGetObjectPosition(clientID, y, -1,
                                     sim.simx_opmode_oneshot_wait)
_, pos_z = sim.simxGetObjectPosition(clientID, z, -1,
                                     sim.simx_opmode_oneshot_wait)
_, pos_y_body = sim.simxGetObjectPosition(clientID, y_body, -1,
                                          sim.simx_opmode_oneshot_wait)

if Pas != 0 and Cont_x !=0 and PermissionA_PLC == True :
    permission = True
else :
    permission =False

#deplacement slider_z
while pos_z[2] >= Cont_z and permission == True:

    _, pos_z = sim.simxGetObjectPosition(clientID, z, -1,
                                         sim.simx_opmode_oneshot_wait)

    if Pas_inv != 0 or Cont_z !=0 and permission :
        sim.simxSetObjectPosition(clientID, z, -1, (pos_z[0], pos_z[1],
                                                    pos_z[2]), sim.simx_opmode_oneshot)

    if pos_z[2] >= Cont_z and Pas != 0 and Cont_z !=0 :
        sim.simxSetObjectPosition(clientID, z, -1, (pos_z[0], pos_z[1], pos_z[2]
                                                    + Pas_inv), sim.simx_opmode_oneshot)

time.sleep(5)
if pos_z[2] == Cont_z :
    break

#deplacement slider_z_haut
while pos_z[2] <= Cont_z_init and permission == True :

    _, pos_z = sim.simxGetObjectPosition(clientID, z, -1,
                                         sim.simx_opmode_oneshot_wait)

    if Pas_inv != 0 or Cont_z_init !=0 :
        sim.simxSetObjectPosition(clientID, z, -1, (pos_z[0], pos_z[1],
                                                    pos_z[2]),sim.simx_opmode_oneshot)

    if pos_z[2] <= Cont_z_init and Pas != 0 and Cont_z_init !=0 :
        sim.simxSetObjectPosition(clientID, z, -1, (pos_z[0], pos_z[1], pos_z[2]+
                                                    Pas_rst), sim.simx_opmode_oneshot)

```

```

#deplacemnt slider_x
while pos_x[0] <= Cont_x and permission ==True :

    _, pos_x = sim.simxGetObjectPosition(clientID, x, -1,
                                         sim.simx_opmode_oneshot_wait)

    if Pas != 0 or Cont_x !=0 :
        sim.simxSetObjectPosition(clientID, x, -1, (pos_x[0] , pos_x[1],
                                                    pos_x[2]), sim.simx_opmode_oneshot)

    if pos_x[0] <= Cont_x and Pas != 0 and Cont_x !=0 :
        sim.simxSetObjectPosition(clientID, x, -1, (pos_x[0] + Pas, pos_x[1],
                                                    pos_x[2]), sim.simx_opmode_oneshot)

#deplacement slider_y
while pos_y[1] <= Cont_y and permission == True :
    _, pos_y = sim.simxGetObjectPosition(clientID, y, -1,
                                         sim.simx_opmode_oneshot_wait)

    if Pas != 0 or Cont_y !=0 :
        sim.simxSetObjectPosition(clientID, y, -1, (pos_y[0], pos_y[1],
                                                    pos_y[2]), sim.simx_opmode_oneshot)

    if pos_y[1] <= Cont_y and Pas != 0 and Cont_z !=0:
        sim.simxSetObjectPosition(clientID, y, -1, (pos_y[0], pos_y[1] + Pas,
                                                    pos_y[2]), sim.simx_opmode_oneshot)

#deplacement slider_z_bas
Cont_z_sens = 1.1116
while pos_z[2] >= Cont_z_sens and permission == True:

    _, pos_z = sim.simxGetObjectPosition(clientID, z, -1,
                                         sim.simx_opmode_oneshot_wait)

    if Pas_inv != 0 or Cont_z_sens !=0 and permission :
        sim.simxSetObjectPosition(clientID, z, -1, (pos_z[0], pos_z[1],
                                                    pos_z[2]), sim.simx_opmode_oneshot)

    if pos_z[2] >= Cont_z_sens and Pas != 0 and Cont_z !=0 :
        sim.simxSetObjectPosition(clientID, z, -1, (pos_z[0], pos_z[1], pos_z[2]
                                                    + Pas_inv), sim.simx_opmode_oneshot)

time.sleep(3)

#collect handles
[err, Robot] = sim.simxGetObjectHandle(clientID, Robotname,
                                       sim.simx_opmode_oneshot_wait)
[err, on_x] = sim.simxGetObjectHandle(clientID, position_x,
                                       sim.simx_opmode_oneshot_wait)
[err, on_y] = sim.simxGetObjectHandle(clientID, position_y,
                                       sim.simx_opmode_oneshot_wait)
[err, on_z] = sim.simxGetObjectHandle(clientID, position_z,
                                       sim.simx_opmode_oneshot_wait)

```

```

# Create data stream
[err, PositionRobot] = sim.simxGetObjectPosition(clientID, Robot, -1,
                                                sim.simx_opmode_streaming)
[err, Position_slider_x] = sim.simxGetObjectPosition(clientID, on_x, -1,
                                                    sim.simx_opmode_streaming)
[err, Position_slider_y] = sim.simxGetObjectPosition(clientID, on_y, -1,
                                                    sim.simx_opmode_streaming)
[err, Position_slider_z] = sim.simxGetObjectPosition(clientID, on_z, -1,
                                                    sim.simx_opmode_streaming)

time.sleep(0.05)

# Collect robot and target data
[err, [xt,yt,zt]] = sim.simxGetObjectPosition(clientID, Robot, -1,
                                             sim.simx_opmode_buffer)
[err, [xx,yx,zx]] = sim.simxGetObjectPosition(clientID, on_x, -1,
                                             sim.simx_opmode_buffer)
[err, [xy,yy,zy]] = sim.simxGetObjectPosition(clientID, on_y, -1,
                                             sim.simx_opmode_buffer)
[err, [xz,yz,zz]] = sim.simxGetObjectPosition(clientID, on_z, -1,
                                             sim.simx_opmode_buffer)

#Ajout d'un message live dans coppeliasim
sim.simxAddStatusBarMessage(clientID, 'La Position de x : '+str(xx) ,
                            sim.simx_opmode_oneshot_wait)
sim.simxAddStatusBarMessage(clientID, 'La Position de y : '+str(yy) ,
                            sim.simx_opmode_oneshot_wait)
sim.simxAddStatusBarMessage(clientID, 'La Position de z : '+str(zz) ,
                            sim.simx_opmode_oneshot_wait)

#print position du slider X sur axe X
print('La Position de x : ', str(xx))

#print position du slider Y sur axe Y
print('La Position de y : ',str(yy))

#print position du slider Z sur axe Z
print('La Position de z : ',str(zz))

# Define the table structure
table = PrettyTable()
table.field_names = ["x","y","z"]

# Add some data to the table
table.add_row([str(xx), str(yy), str(zz)])

# Print the table
print(table)

if pos_x[0] > 0.17 or pos_x[0] < -0.09 :
    print("Default Slider x")

else :

```

```

    print ("posistion sur x atteint")

if pos_y[1] > 0.37 or pos_y[1] < 0.15 :
    print("Defaut Slider Y")
else :
    print("position sur y atteint")

if pos_z[2] > 1.23 or pos_z[2] < 0.9 :
    print("Defaut Slider Z")
else :
    print("position sur Z atteint")

    #pack x,y,z coordonnates in Pos to send to Tia Portal
Pos_X=xx
Pos_Y=yy
Pos_Z=zz
#Sending data to Tia Portal

D_Val_X = ua.DataValue(ua.Variant(Pos_X,ua.VariantType.Float))
var_X.set_data_value(D_Val_X)
Pos_X = var_X.get_value()
print('La Nouvelle Position de X est : ',Pos_X)

D_Val_Y = ua.DataValue(ua.Variant(Pos_Y,ua.VariantType.Float))
var_Y.set_data_value(D_Val_Y)
Pos_Y = var_Y.get_value()
print('La Nouvelle Position de Y est : ',Pos_Y)

D_Val_Z = ua.DataValue(ua.Variant(Pos_Z,ua.VariantType.Float))
var_Z.set_data_value(D_Val_Z)
Pos_Z = var_Z.get_value()
print('La Nouvelle Position de Z est : ',Pos_Z)

task_over = True
task_end = ua.DataValue(ua.Variant(task_over,ua.VariantType.Boolean))
over.set_data_value(task_end)
task_over = over.get_value()
print('Task is over')

task_over = False
task_end = ua.DataValue(ua.Variant(task_over,ua.VariantType.Boolean))
over.set_data_value(task_end)
task_over = over.get_value()

#Rest au position initial position
rst = True
if rst == True :

    #position initial
    Cont_x = -0.07

```



```

Cont_y = 0.17627
Cont_z = 1.2106

Pas_rst = 0.001
Pas_rst_inv = -0.001

_, pos_x = sim.simxGetObjectPosition(clientID, x, -1,
                                     sim.simx_opmode_oneshot_wait)
_, pos_y = sim.simxGetObjectPosition(clientID, y, -1,
                                     sim.simx_opmode_oneshot_wait)
_, pos_z = sim.simxGetObjectPosition(clientID, z, -1,
                                     sim.simx_opmode_oneshot_wait)

while pos_x[0] >= Cont_x and permission == True :

    _, pos_x = sim.simxGetObjectPosition(clientID, x, -1,
                                         sim.simx_opmode_oneshot_wait)
    #deplacemnt slider_x
    if Pas != 0 or Cont_x !=0 :
        sim.simxSetObjectPosition(clientID, x, -1, (pos_x[0], pos_x[1],
                                                    pos_x[2]), sim.simx_opmode_oneshot)
    if pos_x[0] >= Cont_x and Pas != 0 and Cont_x !=0 :
        sim.simxSetObjectPosition(clientID, x, -1, (pos_x[0] - Pas_rst,
                                                    pos_x[1], pos_x[2]), sim.simx_opmode_oneshot)

while pos_y[1] >= Cont_y and permission == True:

    _, pos_y = sim.simxGetObjectPosition(clientID, y, -1,
                                         sim.simx_opmode_oneshot_wait)
    #deplacement slider_y
    if Pas != 0 or Cont_y !=0 :
        sim.simxSetObjectPosition(clientID, y, -1, (pos_y[0], pos_y[1],
                                                    pos_y[2]), sim.simx_opmode_oneshot)
    if pos_y[1] >= Cont_y:
        sim.simxSetObjectPosition(clientID, y, -1, (pos_y[0], pos_y[1] -
                                                    Pas_rst, pos_y[2]), sim.simx_opmode_oneshot)

while pos_z[2] <= Cont_z and permission == True :

    _, pos_z = sim.simxGetObjectPosition(clientID, z, -1,
                                         sim.simx_opmode_oneshot_wait)
    #deplacement slider_z
    if Pas_inv != 0 or Cont_z !=0 :
        sim.simxSetObjectPosition(clientID, z, -1, (pos_z[0], pos_z[1],
                                                    pos_z[2]), sim.simx_opmode_oneshot)
    if pos_z[2] <= Cont_z and Pas != 0 and Cont_z !=0 :
        sim.simxSetObjectPosition(clientID, z, -1, (pos_z[0], pos_z[1],
                                                    pos_z[2]+ Pas_rst), sim.simx_opmode_oneshot)

time.sleep(0.05)

#collect handles
[err, Robot] = sim.simxGetObjectHandle(clientID, Robotname,
                                       sim.simx_opmode_oneshot_wait)

```

```

[err, on_x] = sim.simxGetObjectHandle(clientID, position_x,
                                     sim.simx_opmode_oneshot_wait)
[err, on_y] = sim.simxGetObjectHandle(clientID, position_y,
                                     sim.simx_opmode_oneshot_wait)
[err, on_z] = sim.simxGetObjectHandle(clientID, position_z,
                                     sim.simx_opmode_oneshot_wait)

# Create data stream
[err, PositionRobot] = sim.simxGetObjectPosition(clientID, Robot, -1,
                                                  sim.simx_opmode_streaming)
[err, Position_slider_x] = sim.simxGetObjectPosition(clientID, on_x, -1,
                                                      sim.simx_opmode_streaming)
[err, Position_slider_y] = sim.simxGetObjectPosition(clientID, on_y, -1,
                                                      sim.simx_opmode_streaming)
[err, Position_slider_z] = sim.simxGetObjectPosition(clientID, on_z, -1,
                                                      sim.simx_opmode_streaming)

time.sleep(0.05)

# Collect robot and target data
[err, [xt,yt,zt]] = sim.simxGetObjectPosition(clientID, Robot, -1,
                                              sim.simx_opmode_buffer)
[err, [xx,yx,zx]] = sim.simxGetObjectPosition(clientID, on_x, -1,
                                              sim.simx_opmode_buffer)
[err, [xy,yy,zy]] = sim.simxGetObjectPosition(clientID, on_y, -1,
                                              sim.simx_opmode_buffer)
[err, [xz,yz,zz]] = sim.simxGetObjectPosition(clientID, on_z, -1,
                                              sim.simx_opmode_buffer)

#Ajout d'un message live dans coppeliasim
sim.simxAddStatusBarMessage(clientID, 'La Position de x : '+str(xx) ,
                             sim.simx_opmode_oneshot_wait)
sim.simxAddStatusBarMessage(clientID, 'La Position de y : '+str(yy) ,
                             sim.simx_opmode_oneshot_wait)
sim.simxAddStatusBarMessage(clientID, 'La Position de z : '+str(zz) ,
                             sim.simx_opmode_oneshot_wait)

#print position du slider X sur axe X
print('La Position de x : ', str(xx))

#print position du slider Y sur axe Y
print('La Position de y : ',str(yy))

#print position du slider Z sur axe Z
print('La Position de z : ',str(zz))

# Define the table structure
table = PrettyTable()
table.field_names = ["x","y","z"]

# Add some data to the table
table.add_row([str(xx), str(yy), str(zz)])

# Print the table
print(table)

```

```

#pack x,y,z coordonnates in Pos to send to Tia Portal
Pos_X=xx
Pos_Y=yy
Pos_Z=zz

#Pause simulation
#sim.simxPauseSimulation(clientID,sim.simx_opmode_oneshot_wait)

#Stop simulation:
#end = sim.simxStopSimulation(clientID, sim.simx_opmode_oneshot_wait)

#Now close the connection to V-REP:
#sim.simxAddStatusbarMessage(clientID, 'Pause Program', sim.simx_opmode_blocking)
#sim.simxFinish(clientID)

#Sending data to Tia Portal

D_Val_X = ua.DataValue(ua.Variant(Pos_X,ua.VariantType.Float))
var_X.set_data_value(D_Val_X)
Pos_X = var_X.get_value()
print('La Nouvelle Position de X est : ',Pos_X)

D_Val_Y = ua.DataValue(ua.Variant(Pos_Y,ua.VariantType.Float))
var_Y.set_data_value(D_Val_Y)
Pos_Y = var_Y.get_value()
print('La Nouvelle Position de Y est : ',Pos_Y)

D_Val_Z = ua.DataValue(ua.Variant(Pos_Z,ua.VariantType.Float))
var_Z.set_data_value(D_Val_Z)
Pos_Z = var_Z.get_value()
print('La Nouvelle Position de Z est : ',Pos_Z)

else:
    err_comm = True
    print('Failed to connect to PLC')

else:
    err_comm_robot = True
    print ('Failed connecting to remote API server')
    print ('Fin de Program !')

client.disconnect()

```

Annexe 2

```
function sysCall_init()
    -- Get Object Handles
    x = sim.getObjectHandle('x')
    y = sim.getObjectHandle('y')
    z = sim.getObjectHandle('z')
    gripperName=sim.getObjectAlias(sim.getObject('.'),4)
    openCloseMotorHandle=sim.getObject('./openCloseJoint')
    centerJointHandle=sim.getObject('./centerJoint')
    sensor=sim.getObject('./proxSensor')
    detachSensor=sim.getObject('./detachSensor')
    obj=sim.getObject('./obj')
    attachPt=sim.getObject('./attachPoint')
    detachPt=sim.getObject('./detachPoint')
    conveyor=sim.getObject('./conveyor')
    closed = false
end

function sysCall_actuation()
    -- Start scanning the sensors for the attach and detach
    local scan,dist,Pt,Handle=sim.handleProximitySensor(sensor)
    local
Dscan,Ddist,DPt,DHandle=sim.handleProximitySensor(detachSensor)
    print('scan result :',scan)
    print('Detach scan result :',Dscan)
    -- Start condition for closing the gripper
    if closed == false then
        if(scan == 1 and Dscan == 0) then
            sim.setObjectParent(obj,attachPt,true)
            sim.setJointTargetVelocity(openCloseMotorHandle,-0.02)
            closed = true
            print('closed')
            print('attach', attach)
        end
    -- Start condition for opening the gripper
```

```

else
    if closed == true then
        if Dscan == 1 then
            sim.resetProximitySensor(sensor)
            print('scan result :',scan)
            sim.setJointTargetVelocity(openCloseMotorHandle,0.02)
            sim.setObjectParent(obj,detachPt,true)
            print('open')
        end
    end
end

-- Start condition for starting the convoyer
Dscan,Ddist,Dpt,DHandle=sim.handleProximitySensor(detachSensor)
    if Dscan == 1 then
        sim.writeCustomTableData(conveyor,'__ctrl__',{vel=0.1})
    else
        sim.writeCustomTableData(conveyor,'__ctrl__',{vel=0})
    end
    local data=sim.readCustomTableData(conveyor,'__state__')
    print(data)

end

function sysCall_joint(inData)
    if inData.handle==centerJointHandle then
        local desired=sim.getJointPosition(openCloseMotorHandle)/2
        local error=desired-inData.currentPos
        local ctrl=error*20

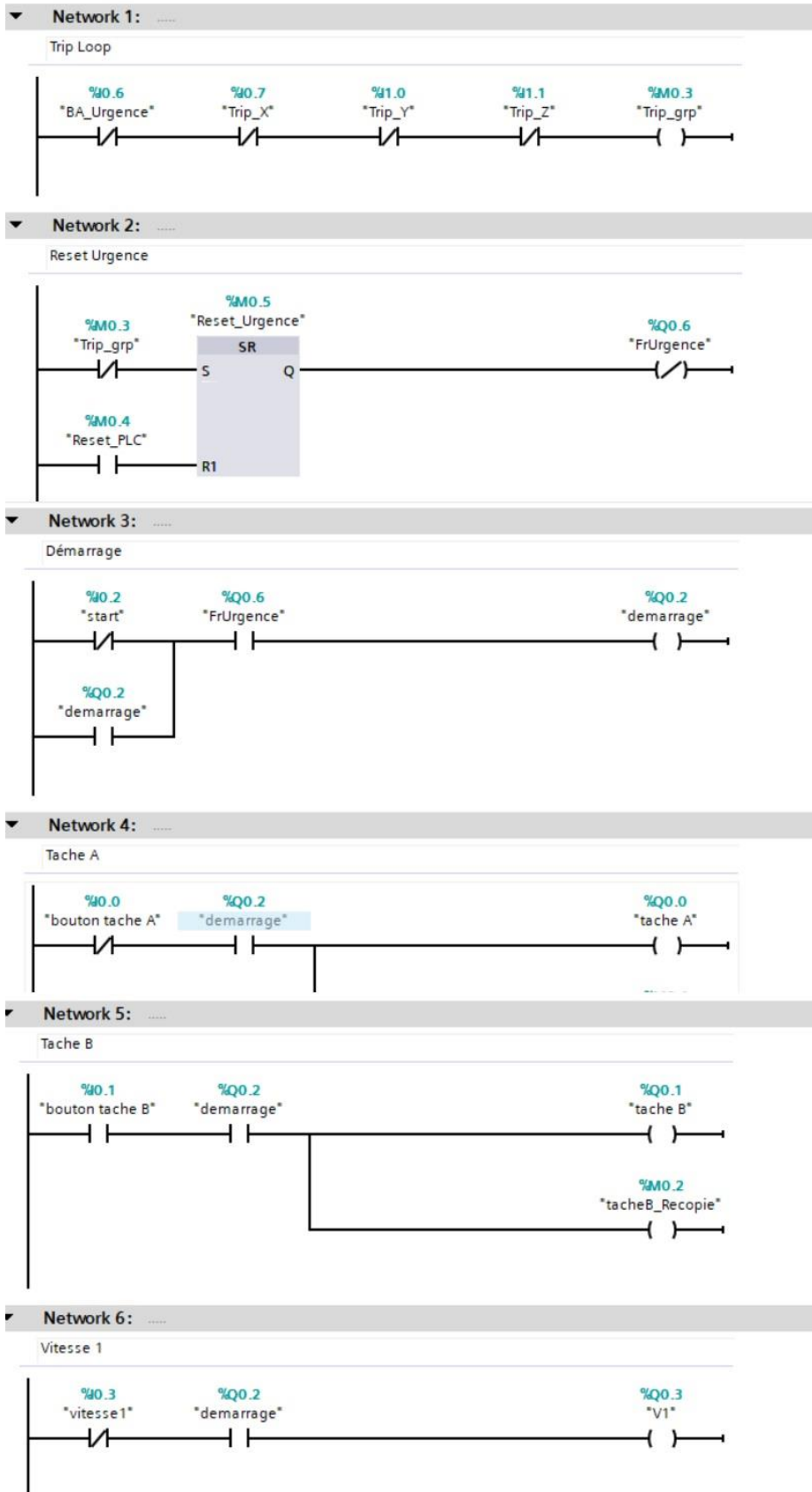
        local maxVelocity=ctrl
        if (maxVelocity>inData.maxVel) then
            maxVelocity=inData.maxVel
        end
    end
end

```

```
    if (maxVelocity<-inData.maxVel) then
        maxVelocity=-inData.maxVel
    end
    local forceOrTorqueToApply=inData.maxForce

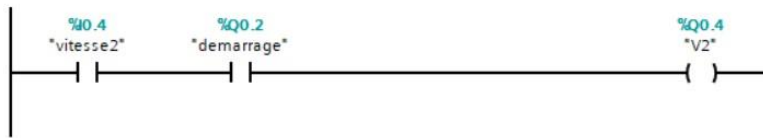
    local outData={vel=maxVelocity,force=forceOrTorqueToApply}
    return outData
end
end
```

Annexe 3



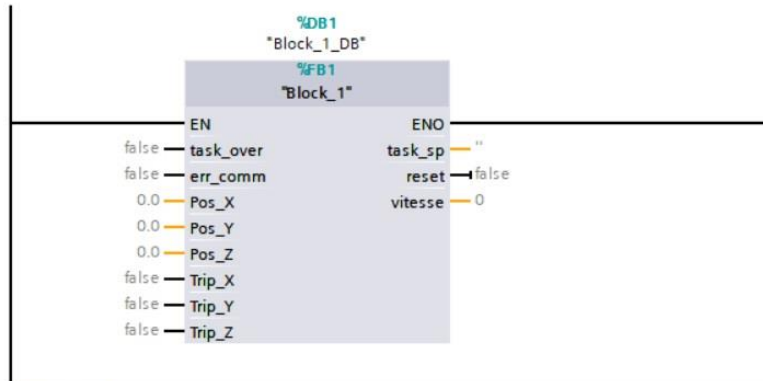
▼ Network 7:

Vitesse 2



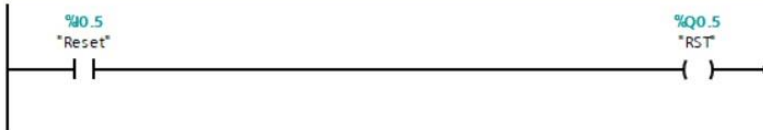
▼ Network 8:

Data Block



▼ Network 9:

Reset Robot



Annexe 4

```
IF "demarrage" = TRUE THEN
  IF "tache A" = TRUE THEN
    #task_sp := 'tacheA';
  END_IF;

  IF "tache B" = TRUE THEN
    #task_sp := 'tacheB';
  END_IF;
;
IF "V1" = TRUE THEN
  #vitesse := 1;
END_IF;

IF "V2" = TRUE THEN
  #vitesse := 2;
END_IF;

IF "RST" = TRUE THEN
  #reset := TRUE;
END_IF;

  IF #Trip_X = TRUE THEN
    "Trip_X" := FALSE ;
  END_IF;
IF #Trip_Y = TRUE THEN
  "Trip_Y" := FALSE;
END_IF;
IF #Trip_Z = TRUE THEN
  "Trip_Z" := FALSE;
END_IF;

END_IF;
```

Liste I/O

Liste des signaux du PLC

PLC TAGs	DESCRIPTION	TYPE	ADRESSE I/O	TABLE TAG
bouton tache A	Button choix tache A	Bool	%I0.0	Default tag table
bouton tache B	Button choix tache B	Bool	%I0.1	Default tag table
tache A	Sortie tache A	Bool	%Q0.0	Default tag table
tache B	Sortie tache B	Bool	%Q0.1	Default tag table
start	Button demarrage	Bool	%I0.2	Default tag table
demarrage	Sortie demarrage	Bool	%Q0.2	Default tag table
vitesse1	Button choix Vitesse 1	Bool	%I0.3	Default tag table
vitesse2	Button choix Vitesse 2	Bool	%I0.4	Default tag table
V1	Sortie Vitesse 1	Bool	%Q0.3	Default tag table
V2	Sortie Vitesse 2	Bool	%Q0.4	Default tag table
Reset	Button reset du robot	Bool	%I0.5	Default tag table
RST	Sortie reset du robot	Bool	%Q0.5	Default tag table
FrUrgence	Frein Urgence	Bool	%Q0.6	Default tag table
BA_Urgence	Button Arrêt d'urgence	Bool	%I0.6	Default tag table
Trip_X	Déclanchement défaut sur X	Bool	%I0.7	Default tag table
Trip_Y	Déclanchement défaut sur Y	Bool	%I1.0	Default tag table
Trip_Z	Déclanchement défaut sur Z	Bool	%I1.1	Default tag table
tacheA_Recopie	Permission de demarrage de la tache A	Bool	%M0.1	Default tag table
tacheB_Recopie	Permission de demarrage de la tache B	Bool	%M0.2	Default tag table
Trip_grp	Groupage des déclanchement	Bool	%M0.3	Default tag table
Reset_PLC	Reset du PLC	Bool	%M0.4	Default tag table
Reset_Urgence	Reset Urgence	Bool	%M0.5	Default tag table

Tableau V.1 - Liste des signaux du PLC

Bloc des données

DATA BLOCK TAGS	DESCRIPTION	TYPE	I/O	TABLE TAG
task_over	tâche terminer	Bool	Input	Data block table
err_comm	erreur communication	Bool	Input	Data block table
Pos_X	Position du slider x	Real	Input	Data block table
Pos_Y	Position du slider y	Real	Input	Data block table
Pos_Z	Position du slider z	Real	Input	Data block table
Trip_X	Déclanchement défaut sur x	Bool	Input	Data block table
Trip_Y	Déclanchement défaut sur y	Bool	Input	Data block table
Trip_Z	Déclanchement défaut sur z	Bool	Input	Data block table
task_sp	nom de tâche	String	Output	Data block table
reset	reset robot	Bool	Output	Data block table

Tableau V.2 - Bloc des données

Liste des variables du contrôleur robot

VARIABLE	DESCRIPTION	TYPE
pos_sp	nom de tâche	String
task_sp	consigne position	String
Pos_Y	Valeur position du slider y	Real
Pos_Z	Valeur position du slider z	Real
Pos_X	Valeur position du slider x	Real
PermissionA_PLC	Permission d'exécution de la tâche A	Bool
PermissionB_PLC	Permission d'exécution de la tâche B	Bool
rst	Reset Robot	Bool
task_over	tâche terminer	Bool
vit	choix de vitesse	Int
Cont_x	position prédefinis sur x	Real
Cont_y	position prédefinis sur y	Real
Cont_z	position prédefinis sur z	Real
Cont_z_init	position initiale sur z	Real
Pas	valeur du pas "vitesse"	Real
Pas_inv	valeur du pas "vitesse" direction inverse	Real
Pas_rst	valeur du pas "vitesse" dans le reset	Real
position_x	prendre la position d'object x	String
position_y	prendre la position d'object y	String
position_z	prendre la position d'object z	String
err_comm	erreur communication PLC	Bool
err_comm_robot	erreur communication robot	Bool

Tableau V.3 - Liste des variables du contrôleur robot

Liste des capteurs et actionneurs du robot

OBJECT	DESCRIPTION	TYPE	MOUVEMENT
YAMAHA SXYs_C_ZF	Robot cartésien	Robot	Dynamique
Slider_x	pièce marche sur l'axe "x"	Actionneur	Dynamique
Slider_y	pièce marche sur l'axe "y"	Actionneur	Dynamique
Slider_Z	pièce marche sur l'axe "z"	Actionneur	Dynamique
Drive_x	Pousseur slider x	Actionneur	Dynamique
Drive_y	Pousseur slider y	Actionneur	Dynamique
Drive_z	Pousseur slider z	Actionneur	Dynamique
Moteur1	Moteur Drive x	Actionneur	Dynamique
Moteur2	Moteur Drive_y	Actionneur	Dynamique
Moteur3	Moteur Drive z	Actionneur	Dynamique
Encodeur1	Encodeur Drive_x	Capteur	Dynamique
Encodeur2	Encodeur Drive_y	Capteur	Dynamique
Encodeur3	Encodeur Drive_z	Capteur	Dynamique
YuMi Gripper	Effecteur Robot	Effecteur	Dynamique
proxSensor	Capteur detection d'effecteur	Capteur	Dynamique

Tableau V.4 - Liste des capteurs et actionneurs du robot