

الجمهورية الجزائرية الديمقراطية الشعبية
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
- جامعة أبي بكر بلقايد - تلمسان
Université Aboubakr Belkaïd - Tlemcen

Faculté de TECHNOLOGIE



MEMOIRE

Présenté pour l'obtention du **diplôme** de **MASTER**

en Electrotechnique

Spécialité Commandes électriques

par M^r KENDOUCI MOHAMMED EL MOKHTAR

et M^r SOUR ABDENNACER

Sujet

Implémentation et supervision d'une loi de commande
Via l'API S7 1200

Soutenu publiquement, le 20 / 06 / 2023, devant le jury composé de :

M ^r . HADJ ABDELKADER Mohammed Amine	Professeur	Univ. Tlemcen	Président
M ^r . LAZREG Mohamed Haithem	MCB	Univ. Tlemcen	Examineur
M ^r . MELIANI Sidi Mohammed	Professeur	Univ. Tlemcen	Encadreur

Année universitaire : 2022 / 2023

Dédicace

Je dédie ce travail :

"À mon père, ma source de force et de sagesse, merci d'avoir été mon guide tout au long de ma vie. Tes conseils avisés et ton soutien inconditionnel ont été les fondations de ma réussite. Je te dédie ce travail avec une profonde admiration et gratitude.

À ma mère, ma source d'amour et de tendresse infinie, tu as toujours cru en moi lorsque je doutais de moi-même. Ton soutien sans faille et ta bienveillance ont nourri ma confiance et ma détermination. Je te suis reconnaissant au-delà des mots et je t'offre cette dédicace avec tout mon amour filial.

À mes chères sœurs Sameh, Imen, Wahiba et Sara et mon frère Kheireddine, votre présence joyeuse et votre soutien indéfectible ont été des rayons de lumière dans ma vie. Vous m'avez encouragé, motivé et avez été mes compagnes de joie tout au long de ce parcours. Cette dédicace est un témoignage de mon amour et de ma reconnaissance éternelle envers vous.

À mes neveux, vous êtes la joie de ma vie et le moteur de mon ambition. Votre innocence et votre amour pur ont illuminé mon chemin. Que ce travail soit un hommage à votre futur, et puissiez-vous poursuivre vos rêves avec détermination et succès.

Enfin, à tous ceux qui m'ont aidé de près ou de loin, que ce soit par des encouragements, des conseils ou un simple geste de gentillesse, je vous suis profondément reconnaissant. Votre contribution a été précieuse dans ma réussite, et je vous offre cette dédicace en témoignage de ma reconnaissance sincère.

Que cette réalisation soit le fruit de notre union, de notre amour et de notre solidarité. Merci du fond du cœur."

KENDOUCI MOHAMMED

Dédicace

Je dédie ce travail :

"À mon père et à ma mère, vous êtes les piliers de ma vie. Votre amour inconditionnel, votre soutien constant et vos sacrifices infinis m'ont permis d'atteindre ce moment de réussite. Vous êtes mes modèles et ma source d'inspiration. Je vous dédie ce travail avec une gratitude infinie.

À mes sœurs, Maoua, Bouchra et Douae, vous êtes mes complices et mes alliées. Votre présence joyeuse et votre soutien inconditionnel ont illuminé mon chemin. Ensemble, nous avons partagé des moments précieux et des rires sincères. Cette dédicace est un témoignage de notre lien indéfectible et de ma gratitude pour vous.

À mon frère Tarek, mon compagnon de vie, nous avons traversé des épreuves ensemble et nous sommes toujours restés solidaires. Ton courage et ton soutien infailible ont été mon inspiration. Cette dédicace est un hommage à notre fraternité et à notre complicité.

À ma fiancée, tu es ma muse et ma meilleure amie. Ta présence à mes côtés a rendu ma vie plus belle et plus significative. Ton amour inconditionnel et ton soutien constant ont été ma force motrice. Je te dédie ce travail en signe de mon amour profond et de ma reconnaissance pour toi.

À tous ceux qui m'ont aidé de près ou de loin, que ce soit par leurs encouragements, leur soutien ou leurs précieux conseils, je vous suis reconnaissant. Vos contributions ont joué un rôle crucial dans ma réussite. Cette dédicace vous est dédiée avec une sincère gratitude.

Que cette réalisation soit le fruit de notre union, de notre amour et de notre collaboration. Merci du fond du cœur pour votre soutien et votre amour."

SOUR ABDENNACER

Remerciements

"...الْحَمْدُ لِلَّهِ الَّذِي هَدَانَا لِهَذَا وَمَا كُنَّا لِنَهْتَدِيَ لَوْلَا أَنْ هَدَانَا اللَّهُ... (الأعراف 43)"

Ce travail a été réalisé au sein du Laboratoire de recherche de la faculté de Génie Electrique d' université Abou Bakr Belkaïd.

Nous remercions Allah Tout-Puissant de nous avoir donné le courage, la force et la volonté d'accomplir cette tâche.

Nous souhaitons exprimer notre sincère gratitude envers notre encadreur, **Mr MELIANI SIDI MOHAMMED**, professeur à l'Université Abu Bakr Belkaïd, pour sa confiance, sa disponibilité, sa patience et son aide inestimable dans notre travail. Il a été un professeur éclairant notre chemin de connaissances et un père comblant nos vies d'amour et de soutien inconditionnel. Nous avons eu l'honneur de vivre cette expérience avec lui.

Nous tenons également à exprimer notre profonde reconnaissance envers tous les membres du jury, **Mr HADJ ABDELKADER MOHAMMED AMINE** et **Mr LAZREG MOHAMED HAITHAM**, pour avoir accepté d'examiner attentivement notre travail.

Enfin, nous exprimons notre profonde gratitude à tous les enseignants du département d'électrotechnique pour leur soutien constant et leur contribution inestimable à notre formation.

Nous n'oublions pas non plus ceux qui ont joué un rôle dans la réalisation de ce travail, et nous les remercions sincèrement pour leur engagement et leur soutien précieux.

ملخص

الهدف الرئيسي من هذا المشروع هو إنشاء جهاز محاكاة وظيفي لمحرك غير متزامن باستخدام واجهة برمجة التطبيقات *Siemens S7-1200*

وقد تحققت الأهداف بالكامل وأدى المشروع إلى تعميق فهم أنظمة الصناعة وتقنيات المحاكاة وبرمجة أجهزة التحكم في المنطق القابلة للبرمجة

يحتوي جهاز المحاكاة على مثبت سرعة *PID_Compact* مدمج ومؤشر *HMI* سهل الاستخدام وذلك باستخدام *Wincc Runtime*

وقد أرسى هذا المشروع أساسا متينا للتطبيقات المستقبلية في مجال المراقبة الصناعية والبرمجة التلقائية

الكلمات المفتاحية *Wincc , Runtime HMI , PID_Compact , SiemensS7-1200*

Abstract

This graduation project successfully creates a functional simulator of an asynchronous motor using the Siemens S7-1200 API.

The objectives were fully achieved and the project deepened understanding of industry regulations, simulation techniques and programming of programmable logic controllers.

The simulator contains a PID_Compact cruise control and a user-friendly HMI using Wincc Runtime.

This project has laid a solid foundation for future applications in industrial control and automatic programming.

Keywords : *Wincc Runtime, HMI, PID_Compact, SiemensS7-1200*

Résumé

Ce projet de fin d'études crée avec succès un simulateur fonctionnel d'un moteur asynchrone à l'aide de l'API Siemens S7-1200.





Les objectifs ont été pleinement atteints et le projet a approfondi la compréhension des réglementations de l'industrie, des techniques de simulation et de la programmation des contrôleurs logiques programmables.

Le simulateur contient un régulateur de vitesse PID_Compact et une IHM conviviale à l'aide de Wincc Runtime.

Ce projet a jeté des bases solides pour de futures applications dans le contrôle industriel et la programmation automatique.

Mots-clés : Wincc Runtime, HMI, PID_Compact, Siemens57-1200



Table des matières

Liste des abréviations	iv
Liste Des figures	v
Introduction Générale	1
1 Introduction aux simulateurs et leurs constitutions	3
1.1 Introduction	4
1.2 Description et rôle des simulateurs	4
1.2.1 Introduction	4
1.2.2 Utilisations des simulateurs	4
1.2.3 Classification	5
1.2.4 Simulateur non piloté	5
1.2.5 Simulateur piloté	6
1.2.6 Exemples de simulateurs	6
1.3 Description du MAS	10
1.3.1 Introduction	10
1.3.2 Définition de la MAS	10
1.3.3 Description de moteur asynchrone triphasée	11
1.3.4 Avantages et inconvénients de du machine asynchrone	12
1.3.5 Commande du moteur asynchrone	13
1.4 Vu général sur (API) et son environnement	18

1.4.1	Introduction	18
1.4.2	Historique	19
1.4.3	Constructeurs d'API	20
1.4.4	SIMATIC S7-1200	21
1.4.5	Progiciel de développement : TIA Portal de Siemens	22
1.5	Conclusion	26
2	Modélisation et implémentation du MAS	27
2.1	Modélisation du Moteur Asynchrone	28
2.1.1	Introduction	28
2.1.2	Hypothèses Simplificatrices	29
2.1.3	Etapes de modélisation du MAS	29
2.1.4	Equations générales du moteur	31
2.2	Implémentation du modèle du MAS dans d'API	40
2.2.1	Introduction	40
2.2.2	TIA Portal V15.1	40
2.2.3	Système d'exploitation et programme utilisateur	42
2.2.4	Choix d'automate programmable	43
2.2.5	Blocs de programme	44
2.2.6	Différentes étapes pour le développement de notre projet sur Tia portal	49
2.2.7	Résultat obtenu	53
2.3	Conclusion	54
3	Régulation industrielle et création d'un simulateur à l'aide de Win CC	55
3.1	Introduction	56
3.2	Régulation industriel	56
3.2.1	Introduction	56
3.2.2	Objet technologique PID-Compact de l'API S7-1200	57
3.2.3	Configuration du régulateur PID-Compact	59
3.2.4	Validation du PID-Compact	61

3.3	Implémentation d'une interface homme-machine (IHM)	61
3.3.1	Introduction	61
3.3.2	Liaison entre l'IHM et le PLC	62
3.3.3	Tableau de liaison des variables entre l'IHM et le PLC	63
3.3.4	Comment créer une variable IHM : (on prend comme un exemple bouton de démarrage "Start")	64
3.3.5	Validation du bouton "Start" dans notre IHM	65
3.4	Présentation de notre prototype réel	67
3.4.1	Module CPU 1214C d'API S7-1200 de Siemens	67
3.4.2	Alimentation électrique Le Sitop PSU100S DC 24 V/10 A	68
3.4.3	Variateur de fréquence	68
3.4.4	Communication Ethernet	69
3.4.5	Disjoncteurs unipolaires	69
3.4.6	SIMATIC HMI	70
3.5	Résultats et validation	70
3.6	Exemple de l'application industrielle à l'aide de factory I/O	73
3.6.1	Introduction	73
3.6.2	Interface entre factory I/O, l'HMI KTP900 et l'API S7-1200	74
3.6.3	Résultat obtenu de la liaison de notre programme avec factory I/O	75
3.7	Conclusion	76
	Conclusion Générale	77
	Bibliographie	78
	ANNEXE	83

Liste des abréviations

ω_s	Pulsation statorique
p	Nombre de pair de pôle de la force magnétomotrice résultante
C_{max}	Représente le couple maximum
N_r	Vitesse de synchronisme en tr/min
V_s	La tension statorique
g	Le glissement
R_r	L'impédance rotorique référée au stator
ω	La vitesse de rotation du rotor
ω_r	La vitesse angulaire relative
RT	Runtime
MAS	Moteur Asynchrone
C_r	Couple résistant
C_{em}	Couple électromagnétique
F	Coefficient de frottement visqueux
J	Moment d'inertie de la partie tournante
σ	coefficient de fuite totale
T_s	Constante de temps statorique
API	Automate Programmable Industriel
OB	Blocs d'organisation
FB	Bloc fonctionnel
PID	Proportionnelle intégrale dérivé

Table des figures

1.1	Simulateur de Boeing 737 [1]	7
1.2	Gestion du trafic dans une ville [2]	8
1.3	Capteurs électroniques des véhicules [3]	8
1.4	Factory IO [4]	9
1.5	Constitution de La MAS [5]	11
1.6	Présentation de la commande en boucle ouverte [5]	14
1.7	Commande V/f en boucle fermée impliquant le contrôle de la vitesse et la régulation de la pulsation de glissement [6]	16
1.8	Commande scalaire en boucle fermée de la machine [7]	16
1.9	Schéma de principe de découplage de la MAS/MAC [7]	17
1.10	Commande vectorielle de la machine asynchrone[8]	18
1.11	API [9]	19
1.12	S7-1200 SIEMENS [10]	21
1.13	Simatic WinCC [10]	24
2.1	Présentation des enroulements statorique et rotorique.[7]	30
2.2	Principe de la transformation de Park [6]	33
2.3	Schéma-bloc du moteur asynchrone sous Simulink	38
2.4	Vitesse angulaire de rotation d'un MAS (rad/s).	39
2.5	Réponse du couple électromagnétique (CEM) d'un MAS	39
2.6	Tia portal V15.1 et le S7-1200 [10]	41
2.7	Interaction entre le système d'exploitation et le programme utilisateur[10]	42

2.8	S7-1200 1214C DC/DC/DC [10]	44
2.9	Schéma regroupant les différents bloc et fonctions au niveau de la CPU	45
2.10	Procédure d'appel successive des sous-programmes	46
2.11	Différents accès aux blocs de données	49
2.12	Représentation des différents blocs de notre projet	50
2.13	Bloc DB du MAS	51
2.14	Programme du Modèle MAS	52
2.15	Variable du modèle MAS	52
2.16	Fonction du MAS au niveau du bloc OB	53
2.17	Réponse sous TIA Portal de La vitesse de rotation (rad/s)	54
3.1	Schéma de principe utilisant une régulation PI[11]	58
3.2	Schéma-bloc du régulateur PI [11]	59
3.3	Régulation de vitesse par le biais du PID-Compact du S7-1200	60
3.4	Mise à l'échelle de la vitesse angulaire relative	60
3.5	Calcul de la vitesse angulaire de synchronisme (pulsation)	61
3.6	Réponses de sortie du PID-Compact lors de la régulation dans différentes conditions	62
3.7	Liaison entre l'IHM et le PLC	63
3.8	Table IHM du MAS	63
3.9	Réseaux du pilotage Start/Stop	65
3.10	"Vue de notre IHM avant appui sur bouton Star (lors de l'arrêt du MAS) "	66
3.11	" Appui sur bouton Star (lors que le moteur est en marche) "	66
3.12	simulateur de moteur asynchrone à l'aide de l'API S7-1200	67
3.13	CPU 1214C de Siemens	67
3.14	Sitop PSU100S DC 24 V/10 A de Siemens	68
3.15	Variateur de fréquence 6SL3220-3YE24-0AF0 de Siemens	68
3.16	Module de communication Ethernet industriel de Siemens	69
3.17	Schéma des 03 disjoncteurs unipolaires	69
3.18	SIMATIC HMI TP900 Comfort	70

3.19 Réponse de la vitesse de rotation du MAS (tr/min)	71
3.20 Réponse du couple électromagnétique (CEM) du MAS	71
3.21 Réponse de la vitesse de rotation avec la régulation PI	72
3.22 Réponse du CEM avec la régulation PI	73
3.23 Factory I/O	74
3.24 Factory I/O et L'API S7-1200	74

Il existe une multitude de systèmes informatiques qui sont couramment utilisés de nos jours, allant des ordinateurs personnels et des smartphones aux systèmes électroniques sophistiqués intégrés dans les voitures, les distributeurs automatiques de billets et les avions. Pour exploiter ces systèmes, il est nécessaire de développer des programmes et d'acquérir une formation appropriée, pour lesquels des simulateurs et des émulateurs peuvent être utilisés. Bien que ces deux approches soient similaires dans leur utilisation, elles diffèrent dans leur façon de fournir une représentation d'un autre système. Leur objectif est de simuler ou d'émuler un système complexe en utilisant un autre système complexe. Les simulateurs et les émulateurs sont utilisés pour diverses raisons, par exemple, un programmeur peut utiliser des émulateurs pour tester un logiciel sur différents périphériques avant de l'installer. De même, les simulateurs sont couramment utilisés pour la formation dans des situations dangereuses. Les consoles de jeux modernes peuvent également utiliser des émulateurs pour permettre aux joueurs de jouer à des jeux provenant de systèmes plus anciens sur des appareils plus récents.

Les développements technologiques dans le domaine industriel ont conduit à la généralisation de l'utilisation des simulateurs, de la modélisation et du contrôle pour améliorer les systèmes électromécaniques tels que les moteurs asynchrones. Ces avancées fournissent des solutions efficaces pour la formation, la recherche, la conception et la maintenance des systèmes dans diverses applications industrielles.

Dans ce projet, nous allons concevoir un simulateur pour un moteur asynchrone en utilisant l'API S7 1200

Le premier chapitre de ce mémoire se concentre sur l'importance des simulateurs dans différents domaines et prend les simulateurs de vol comme exemple pour une introduction clé. Il met en évidence la manière dont les simulateurs, qu'ils soient sans pilote ou interactifs, peuvent être utilisés dans divers contextes, notamment la formation, la recherche, la conception de systèmes et la maintenance. En outre, il présente les machines électriques asynchrones, également appelées moteurs asynchrones, en tant que dispositifs électromécaniques essentiels dans l'industrie, en particulier pour des applications telles que le pompage, le transport et la production d'électricité. Enfin, il se termine par une discussion sur l'automatisation programmable industrielle, avec un accent particulier sur des solutions telles que SIMATIC S7-1200, TIA Portal et WinCC de Siemens.

Le deuxième chapitre vise à modéliser un moteur à induction à l'aide du logiciel TIA Portal. Il décrit une approche de modélisation qui implique l'utilisation de Simulink et de scripts pour créer un modèle de moteur à induction, puis le répliquer dans TIA Portal.

Les résultats obtenus à partir des deux modèles ont été comparés pour s'assurer de leur cohérence et de leur précision. Cette étape de comparaison est essentielle pour valider le modèle et assurer sa fiabilité. S'il y a une différence significative entre les résultats, des ajustements sont nécessaires pour améliorer la précision du modèle.

Le troisième chapitre se concentre sur la création d'un simulateur de contrôle industriel et de moteur asynchrone sous WinCC. Le régulateur de vitesse PI de l'objet technologique PID-Compact du S7-1200 permet d'illustrer les principes de régulation et de configuration. De plus, une interface homme-machine (IHM) a été développée pour faciliter la surveillance et le contrôle des moteurs asynchrones. Les résultats obtenus à partir d'un simulateur de moteur asynchrone basé sur WinCC pour mieux comprendre le comportement d'un moteur à induction à l'aide d'outils industriels de régulation et de contrôle sont présentés et discutés.

En résumé, ce travail met en évidence l'importance des simulateurs, de la modélisation et du contrôle dans le contexte des moteurs à induction. L'utilisation de logiciels tels que TIA Portal, Simulink et WinCC augmente l'efficacité, la fiabilité et les performances de ces moteurs dans diverses applications industrielles. La comparaison et la validation des modèles entre différents outils peuvent garantir leur précision et leur fiabilité, ou identifier les ajustements nécessaires pour les améliorer. Cette combinaison de technologies ouvre de nouvelles perspectives pour l'industrie en créant des systèmes plus efficaces.

Chapitre **1**

Introduction aux simulateurs et leurs
constitutions

1.1 Introduction

Le premier chapitre donne une définition de simulateur . Comme exemple, on prendra le simulateur de vol et la simulation d'un système automatisé, sont largement utilisés dans diverses applications telles que la formation, la recherche, la conception et la maintenance des systèmes. Ensuite, la machine asynchrone, également appelée moteur asynchrone, est un dispositif électromécanique largement utilisé dans l'industrie pour diverses applications, notamment le pompage, le transport et la production d'énergie. Enfin, Le chapitre se conclut par une description généralisé sur l'automatisation programmable industrielle, en examinant spécifiquement le SIMATIC S7-1200 de Siemens, le TIA Portal et le WinCC.

1.2 Description et rôle des simulateurs

1.2.1 Introduction

Un simulateur est un dispositif technique qui permet de reproduire de manière virtuelle une situation, permettant ainsi de contrôler et d'observer l'évolution du modèle d'un phénomène ou d'un système complexe. Les simulateurs peuvent être des systèmes matériels ou logiciels qui imitent d'autres systèmes avec des niveaux variables de réalisme. Les jeux vidéos comprennent souvent des simulateurs de ville, de guerre et de vol, bien que ces derniers ne soient pas très réalistes. Les simulateurs de formation pour les compagnies aériennes, par exemple, sont très réalistes, avec des commandes matérielles et logicielles qui reproduisent fidèlement la physique et les détails du vol. Les simulateurs sont particulièrement utiles lorsque le système réel est difficilement observable pour diverses raisons telles que la dimension, la sécurité, le coût ou l'inexistence. Un simulateur est un outil qui effectue des simulations, qui présente des avantages et des inconvénients selon les situations.[12]

1.2.2 Utilisations des simulateurs

Les simulateurs ont une multitude d'applications dans divers domaines. Les entités gouvernementales les utilisent pour simuler la propagation de la maladie et les effets de catastrophes naturelles telles que les inondations et les tornades. Les scientifiques les utilisent pour modéliser des systèmes physiques, au point que l'informatique scientifique est un domaine florissant avec ses propres publications spécialisées. Les fabricants utilisent des simulations pour modéliser leurs produits avant d'engager des coûts de production.

L'intelligence artificielle utilise la simulation et émulation pour développer sa base de connaissances. En somme, la simulation et l'émulation sont devenues des outils indispensables dans cette ère moderne.[13]

1.2.3 Classification

Il existe plusieurs façons de classer les simulateurs. La classification la plus simple consiste à les distinguer en fonction de l'implication de l'homme dans la simulation :

1. Les simulateurs non pilotés, également appelés simulateurs numériques, ne nécessitent l'intervention humaine que pour le paramétrage et l'initialisation de la simulation. Comme un exemple le simulateur d'un circuit électronique ne nécessite pas d'être en temps réel. :
2. Les simulateurs pilotés (ou interactifs) impliquent la participation de l'homme à la simulation. Comme un exemple le simulateur de vole qui simule le vol d'un avion avec un pilote réel aux commandes. Ce type de simulateur nécessite une réponse en temps réel pour une expérience réaliste.
3. La réalité virtuelle est parfois considérée comme étant similaire aux simulateurs pilotés, Le modèle évolue en temps réel .
4. Dans certains cas particuliers, la simulation peut être superposée à l'environnement réel pour enrichir l'expérience. Dans ce cas, on parle de réalité augmentée et non de simulateurs. En résumé, les simulateurs peuvent être classés en fonction du degré d'implication humaine, du temps de réponse requis et de la nature de l'environnement dans lequel se déroule la simulation..[14]

1.2.4 Simulateur non piloté

Un type de simulateur qui peut fonctionner sans intervention humaine est du simulateur de conduite autonome. Il peut reproduire des situations complexes pour tester des algorithmes et des modèles de manière automatisée. Les simulateurs de trafic routier autonomes peuvent être utilisés pour évaluer l'impact de différentes politiques de mobilité du trafic et les temps de trajet, tandis que les simulateurs de marchés financiers autonomes peuvent être utilisés pour tester des stratégies d'investissement et évaluer les risques liés à divers scénarios économiques. Ces simulateurs sont souvent utilisés pour automatiser des tâches complexes et répétitives, permettant a les utilisateurs de se concentrer sur l'analyse des résultats et la prise de décisions.[15] Voici quelques exemples d'utilisation d'un simulateur autonome. Un simulateur de trafic peut être employé pour tester les répercussions de

politiques de circulation variées sur les temps de trajet et l'efficacité des systèmes d'éclairage public ou des nouvelles voies réservées aux voitures[16]. Les simulateurs de marchés financiers peuvent être utilisés pour évaluer les plans d'investissement et les dangers liés à des conditions économiques spécifiques, tels que l'impact des événements économiques sur le marché financier ou l'efficacité des plans basés sur les simulateurs d'aide à la mobilité pour des personnes handicapées qui sont très récents.[17] [14]

1.2.5 Simulateur piloté

Un simulateur de conduite ou interactif est un dispositif qui permet à un utilisateur de se placer et d'interagir avec un environnement virtuel en temps réel. Le but de tels simulateurs est de fournir une expérience immersive à l'utilisateur, lui permettant de piloter le simulateur et de voir les résultats de ses opérations. Pour cela, des technologies de réalité virtuelle sont utilisées, telles que des interfaces visuelles, acoustiques et proprioceptives. Il existe différents types de simulateurs de conduite, chacun ayant un but précis[18]. Certains sont utilisés pour la recherche et l'étude, tandis que d'autres sont utilisés pour une formation et une pratique professionnelle. Les simulateurs de vol en sont un exemple, certains sont stationnaires, c'est-à-dire que le pilote ne peut percevoir ses mouvements que visuellement.[19] Cependant, la validité du simulateur est une question fondamentale. Les environnements virtuels sont souvent très éloignés de la réalité et peuvent entraîner des écarts dans le comportement du pilote et la conscience de la situation[20]. La nature photométrique et la taille finie des systèmes de récupération de mouvement peuvent également limiter la précision des simulations. Malgré ces limites, les simulateurs habités continuent d'être largement utilisés dans des domaines aussi divers que l'aviation, la médecine et la sécurité. Les utilisateurs professionnels de simulateurs doivent être conscients des limites de la simulation et de la nécessité de valider leur expérience en le comparant à la réalité.[21] [22]

1.2.6 Exemples de simulateurs

Ça existe plusieurs types de simulateurs les plus répandus sont :

1. Simulateurs de vol

Ce sont des appareils à la pointe de la technologie qui permettent aux pilotes de s'entraîner à piloter un avion sans quitter l'aéroport. Les simulateurs modernes sont basés sur la technologie numérique, où les ordinateurs calculent les données de contrôle de vol et créent des interfaces utilisateur en temps réel. Contrairement aux simulateurs analogiques moins complexes des années 70 et 80, les nouveaux

simulateurs de vol peuvent être utilisés à diverses fins, notamment le développement d'avions, la formation des pilotes, les enquêtes sur les accidents et même les simulateurs d'aviation de combat (Figure 1.1).[23] [24]



FIGURE 1.1 – Simulateur de Boeing 737 [1]

Au cœur de tout simulateur de vol se trouvent trois composants principaux : les dispositifs d'entrée de données, les modèles et appareils de vol de compensation environnementale. Les modèles de vol utilisent des équations de vent et des paramètres de vol pour créer des simulations[25]. Dans les émulateurs professionnels, ces données sont généralement confidentielles et ne peuvent pas être utilisées dans les jeux vidéo.

2. Simulateur de système automatisé

La simulation de système d'automatisation est le processus de modélisation et de simulation virtuelle de systèmes d'automatisation réels. Il s'agit de créer une représentation virtuelle d'un système, comprenant ses composants, ses interactions, ses paramètres et son comportement, afin que son fonctionnement puisse être reproduit dans un environnement virtuel.[26]

Prenons l'exemple d'un système automatisé de gestion du trafic dans une ville. Dans cette simulation, nous modéliserons les feux de circulation, les capteurs de véhicules, les contrôleurs de signalisation et les modèles de trafic (Figure 1.2).[2]

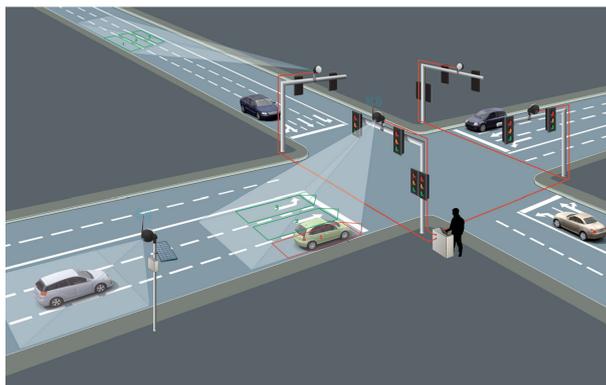


FIGURE 1.2 – Gestion du trafic dans une ville [2]

- Modélisation des feux tricolores : Nous créons une représentation virtuelle d'un feu tricolore à l'aide d'objets graphiques correspondant à des feux de différentes couleurs (rouge, vert, jaune). Nous les programmons pour changer de couleur séquentiellement en fonction des conditions météorologiques ou de circulation.
- Capteurs de véhicules : Nous ajoutons des capteurs virtuels le long des voies pour détecter la présence de véhicules. Ces capteurs envoient des informations aux contrôleurs de signaux pour régler les feux en fonction des conditions de circulation réelles.
- Signal Controller : Nous avons créé un contrôleur virtuel pour recevoir les informations des capteurs du véhicule. Sur la base de ces informations, le contrôleur détermine la durée des feux et la séquence des changements de feux à chaque intersection pour optimiser la fluidité du trafic (Figure 1.3).

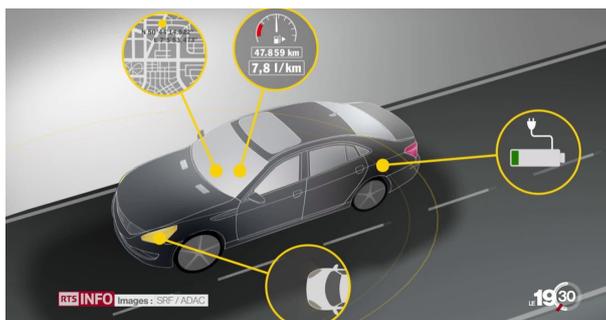


FIGURE 1.3 – Capteurs électroniques des véhicules [3]

- Modèles de trafic : nous mettons en place des modèles de trafic virtuels pour simuler différents scénarios de trafic, tels que l'heure de pointe, le volume de trafic élevé sur certaines routes, etc. Nous pouvons également introduire des événements spéciaux, tels que des accidents ou des travaux routiers, pour évaluer la réactivité des systèmes automatisés. Une fois la simulation terminée, nous pouvons exécuter le modèle et observer le fonctionnement du système automatisé

de gestion du trafic. Nous pouvons analyser les temps d'attente des véhicules, le volume de trafic, les temps de passage aux intersections, etc. En ajustant les paramètres du contrôleur de signalisation, nous pouvons optimiser les performances du système et identifier les éventuels problèmes ou goulots d'étranglement. Cette simulation permet de tester différentes configurations de contrôle des signaux, de prévoir les problèmes de congestion, de valider l'efficacité des modèles de trafic et de prendre des décisions éclairées pour améliorer la gestion du trafic dans les villes sans affecter le trafic réel ni investir dans des infrastructures physiques coûteuses.[3]

3. Factory I/O

Factory IO est un logiciel de simulation industrielle qui vous permet de créer des environnements virtuels pour les systèmes d'automatisation de la formation et de l'apprentissage. Il est principalement utilisé dans le domaine de l'automatisation industrielle pour simuler les processus de fabrication et d'assemblage.[4] Factory IO vous permet de concevoir des scénarios de production à l'aide d'une interface graphique facile à utiliser. Le logiciel fournit une bibliothèque de composants industriels tels que des capteurs, des actionneurs, des machines et des convoyeurs qui peuvent être utilisés pour créer des lignes de production virtuelles. Une fois l'environnement de production configuré, des logiciels de programmation et de contrôle tels que des automates programmables (automates programmables) et des IHM (interfaces homme-machine) peuvent être utilisés pour programmer et contrôler les différents composants. Factory IO prend en charge plusieurs langages de programmation couramment utilisés dans l'automatisation industrielle, tels que ladder logic et fonction bloc (Figure 1.4).



FIGURE 1.4 – Factory IO [4]

L'avantage de Factory IO est qu'il offre une expérience de fabrication industrielle

réaliste, permettant aux utilisateurs de tester et de déboguer des programmes de contrôle sans avoir besoin d'un équipement physique réel. Cela permet d'économiser du temps et des ressources dans la formation et le développement de systèmes d'automatisation. Factory IO est utilisé par les établissements d'enseignement, les centres de formation professionnelle et les entreprises pour former les étudiants et les professionnels à la programmation et à la mise en œuvre de systèmes d'automatisation. Vous pouvez également expérimenter différentes configurations pour tester des scénarios de production avant de les implémenter dans votre environnement réel.[4]

Dans le cas de notre travail de PFE, nous allons aborder les différentes étapes menées pour développer notre simulateur utilisant l'API de Siemens dont le but sera d'étudier le comportement du moteur asynchrone dans le domaine de la régulation industrielle. Dans le paragraphe suivant, nous commencerons par donner une description de fond du moteur asynchrone MAS.

1.3 Description du MAS

1.3.1 Introduction

La machine asynchrone est une invention qui a révolutionné le monde de l'ingénierie. Aussi appelée machine à induction, elle fonctionne grâce à un champ magnétique alternatif créé dans le stator. Le rotor, quant à lui, est constitué d'une cage d'écurie qui permet de transformer l'énergie électrique en énergie mécanique, ce qui permet à la machine de fonctionner. Le terme "asynchrone" fait référence à vitesse de rotation du rotor qui ne correspond pas exactement à la fréquence du courant alternatif traversant le stator. Cette particularité permet aux machines asynchrones d'être utilisées dans une variété d'applications, telles que les pompes, les ventilateurs, les compresseurs, les convoyeurs, et bien plus encore[27].

Les Moteurs asynchrones sont également largement utilisées dans l'industrie automobile, notamment pour les moteurs électriques des voitures hybrides et électriques. Grâce à leur grande efficacité énergétique et à leur fiabilité, les Moteurs asynchrones sont devenues un choix privilégié pour de nombreuses applications industrielles et domestiques[28].

1.3.2 Définition de la MAS

On appelle machine asynchrone toute machine qui ayant $(2p)$ pôles et étant reliée à un réseau de fréquence F_s , ne tourne pas exactement à la vitesse du champ magnétique de

fréquence (f_s) . On parle généralement de moteur d'entraînement car ces moteurs sont destinés à fournir de la puissance mécanique à partir du réseau électrique.

Parmi les machines asynchrones, on peut distinguer deux types :

- Rotor à cage d'écureuil
- Rotor bobiné

Le moteur d'induction est caractérisée par une armature non alimentée (rotor), parcourue par des courants provoqués par. l'autre armature qui est alimentée à partir d'un réseau de fréquence (stator) f_s [29].

1.3.3 Description de moteur asynchrone triphasée

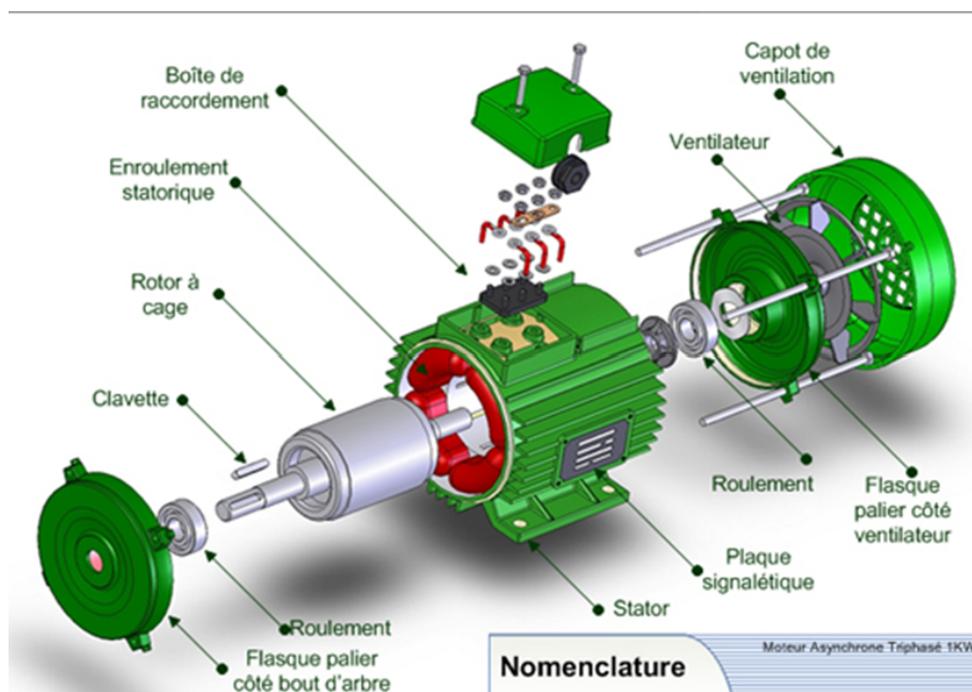


FIGURE 1.5 – Constitution de La MAS [5]

Le moteur asynchrone est constitué d'une partie fixe constituée d'un carter à l'intérieur duquel sont placés le circuit magnétique et l'enroulement du stator d'une part, et d'une partie mobile appelée rotor d'autre part (Figure 1.5). Le fonctionnement du moteur asynchrone dépend de l'interaction entre le champ magnétique rotatif du stator et le rotor qui est une cage d'écureuil. Lorsque le courant alternatif traverse le stator, Il produit un champ magnétique rotatif qui induit une tension dans les barres du rotor. Cette tension provoque un courant qui à son tour crée un champ magnétique dans le rotor. Le rotor, qui n'est pas connecté électriquement au stator, est entraîné à tourner par l'interaction

entre le champ magnétique tournant du stator et le champ magnétique créé dans le rotor. Ce mouvement de rotation est possible grâce au couple moteur produit par les forces magnétiques.[30]. Le couple moteur provoque la rotation du rotor dans la même direction que le champ magnétique tournant du stator, ce qui permet de faire fonctionner la machine asynchrone comme un moteur électrique. En exploitation normal, le rotor d'un moteur asynchrone tourne à une vitesse angulaire Ω (rad/s) et la force magnéto-motrice produite par l'enroulement statorique tournant à une vitesse angulaire de synchronisme Ω_s (rad/s) égale à :

$$\Omega_s = \frac{\omega_s}{p} \quad (1.1)$$

Avec

- ω_s : Pulsation statorique, liée à la fréquence du stator.
- p : nombre de pair de pôle de la force magnétomotrice résultante.

Le rotor est formé de façon à obtenir trois enroulements ayant un nombre de pôles identique à celui du stator. Les enroulements du rotor sont court-circuités, la vitesse de rotation du rotor est inférieure à Ω_s . On note, g , glissement de la vitesse de rotation par rapport à celle du champ tournant statorique.

$$g = \frac{\Omega_s - \Omega}{\Omega_s} \quad (1.2)$$

1.3.4 Avantages et inconvénients de du machine asynchrone

Les machines asynchrones sont couramment utilisés dans l'industrie en raison de leur fiabilité, de leur faible coût et de leur disponibilité en grandes quantités. Ils sont conçus pour fonctionner avec des courants alternatifs et leur conception simple les rend robustes et peu sujets à l'usure. Les moteurs asynchrones sont souvent utilisés dans les machines-outils, les pompes, les ventilateurs, les compresseurs et autres équipements industriels. Leur efficacité énergétique et leur durée de vie en font un choix économique pour de nombreuses applications industrielles. Pour les moteurs à puissance élevée, le démarrage sous tension réduite est une méthode courante pour éviter un appel de courant élevé lors de la mise en marche du moteur asynchrone. En revanche, La complexité opérationnelle d'un MAS est due à des non-linéarités et des caractéristiques telles que l'impossibilité de se procurer le courant rotorique , Il est donc difficile de détecter certains paramètres comme R_r , et il faut modifier l'équation vectorielle répliquée de l'équation vectorielle répliquée pour exprimer les coefficients du rotor par leur action sur le stator, malgré la simplicité de la conception[30].

1.3.5 Commande du moteur asynchrone

1. Commande scalaire

La commande scalaire est une technique de commande de la vitesse et du couple d'un moteur asynchrone. Son principe est de maintenir le rapport entre la tension et la fréquence d'alimentation (V/f) constant, ce qui permet de maintenir le flux magnétique constant. En régime permanent, le couple maximum que peut fournir la machine est directement proportionnel au carré de ce rapport [31]. Pour contrôler le couple en régime permanent, deux grandeurs de réglage sont utilisées : la pulsation rotorique (ω_r) et le flux totalisé (ϕ_s ou ϕ_r). Pour imposer la pulsation rotorique, on utilise la relation naturelle d'auto-pilotage. Pour contrôler le flux, on utilise des boucles de contrôle à partir des grandeurs électriques statoriques (courants ou tensions)[31]. Il existe plusieurs stratégies de commande scalaire, notamment la commande en V/f , qui est la plus couramment utilisée avec un onduleur de tension. Cette technique est simple à mettre en œuvre, mais sa dynamique est relativement lente et elle présente des problèmes de précision à basse vitesse ainsi qu'un contrôle moins efficace des régimes transitoires[32]. La commande scalaire, également appelée commande en V/f (tension sur fréquence), est une technique de commande de moteurs asynchrones qui vise à maintenir le flux magnétique constant tout en contrôlant le couple en agissant sur le glissement du moteur[33].

a) Principe de la commande scalaire

Dans une machine asynchrone, le stator et le champ rotorique restent en contact car, par entrée, la fréquence du courant rotorique varie automatiquement avec la vitesse du rotor pour compenser cette variation. Le champ rotorique est mobile dans le cas du stator pulsé, contrairement au cas de machine synchrone où la fréquence du courant d'excitation reste constante et nulle quelle que soit la vitesse du rotor en place, donc, le champ rotorique est fixe par rapport à celui-ci. [34] Dans une machine asynchrone, il n'y a pas de risque de passer par le couple moyen nul mais seulement de bloquer par la limite de résistance, qui est supérieure au maximum, ce qui provoque une rupture de fonctionnement et un court-circuit puisqu'il n'y a pas d'autre f.e.m. de rotation contre la tension du stator. [35] Une machine asynchrone présente une insécurité dans les états transitoires. C'est le cas lors d'une alimentation en tension continue lorsque le filtre d'entrée a des valeurs d'inductance et de capacité élevées. C'est également le cas lorsque le courant est donné à la fréquence du stator.[36]

b) Principe de la commande en boucle ouverte

La commande du moteur asynchrone en boucle par l'intermédiaire de l'onduleur s'effectue par la pulsation et la valeur efficace de la tension phase-neutre du stator pour un onduleur de tension (Figure 1.6). [31]

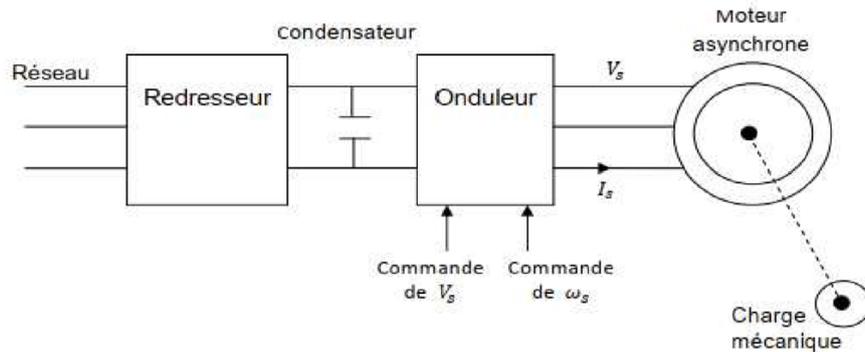


FIGURE 1.6 – Présentation de la commande en boucle ouverte [5]

c) Expression du couple

Le couple est contrôlé en action sur le glissement. En régime établi, le couple maximum est donné par la relation.

$$C_{\max} = ((3 * p) / (2 * N'r)) * (V_s / \omega_s)^2 \quad (1.3)$$

Dans cette expression :

- C_{\max} représente le couple maximum.
- p est le nombre de pôles du moteur.
- $N'r$ est la vitesse de synchronisme en tr/min ($N'r = (120 * f) / p$, où f est la fréquence statorique en Hz).
- V_s est la valeur efficace de la tension statorique en volts.
- ω_s est la pulsation statorique en rad/s ($\omega_s = 2 * \pi * f_s$, où f_s est la fréquence statorique en Hz).

Cette relation montre que le couple maximal est proportionnel au carré du rapport de la tension du stator à la fréquence du stator [37]

d) Commande en vitesse

Dans la commande scalaire en vitesse, la régulation de la vitesse de la machine se fait en reconstituant la pulsation statorique à partir de la vitesse et de la pulsation rotorique. Le régulateur de vitesse ajuste la pulsation rotorique pour fournir

le couple nécessaire en cas de charge et modifie la pulsation statorique en ajustant la fréquence de la tension d'alimentation. Cela garantit un équilibre entre le couple électromagnétique et la charge mécanique, tout en maintenant le rapport V/f constant pour un contrôle précis de la vitesse de la machine.[37] Dans ce cas le couple électromagnétique (C_e) d'une machine est proportionnel à la pulsation rotorique à flux constant et donner par :

$$C_e \approx 3p \left(\frac{V_s}{\omega_s} \right)^2 \frac{1}{R_r'} g \omega_s = a g \omega_s = a \omega_r \quad (1.4)$$

Dans cette équation :

- p : Le nombre de paires de pôles de la machine. Il représente le nombre de pôles magnétiques sur le stator et influence la vitesse de synchronisme de la machine.
- V_s : La tension statorique. C'est la tension d'alimentation appliquée au stator de la machine.
- ω_s : La vitesse de synchronisme. C'est la vitesse théorique à laquelle le champ magnétique du stator tournerait si le glissement était nul.
- g : Le glissement. Il représente la différence de vitesse entre le rotor et le champ magnétique tournant du stator. Un glissement élevé indique un écart important entre la vitesse de rotation réelle du rotor et la vitesse de synchronisme.
- R_r' : L'impédance rotorique référée au stator. Il représente l'impédance électrique du rotor vue du stator.
- a : Une constante liée aux caractéristiques de la machine. Elle dépend des paramètres de la machine, tels que sa construction et ses dimensions.
- ω_r : La vitesse de rotation du rotor. C'est la vitesse réelle à laquelle le rotor de la machine tourne.[31]

e) Commande scalaire en boucle fermée

La stratégie de commande en boucle fermée consiste à contrôler directement la pulsation de glissement w_r de la machine en utilisant l'équation $C_e = k(V/\omega_s)*\omega_r$. Pour ce faire, l'erreur de vitesse ($\Omega_{ref} - \Omega$) est utilisée pour générer une consigne de pulsation ω_r^* à travers un contrôleur. Cette consigne est ensuite ajoutée au signal de vitesse réelle pour déterminer la commande de fréquence de l'onduleur, $\omega_s^* = \omega_r^* + p\Omega$, qui est la loi interne de la machine reproduite à l'extérieur par la commande (Figure 1.7). [38]

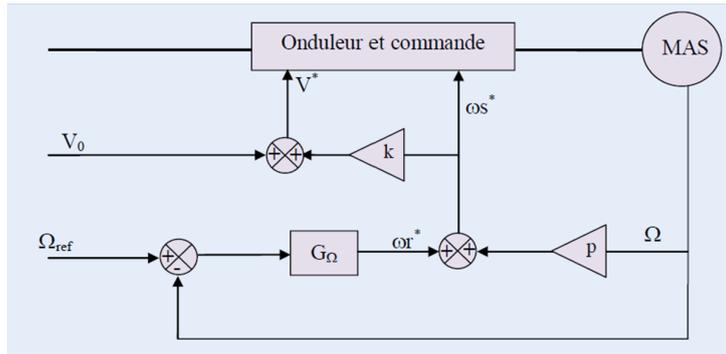


FIGURE 1.7 – Commande V/f en boucle fermée impliquant le contrôle de la vitesse et la régulation de la pulsation de glissement [6]

Le régulateur de vitesse limite la pulsation de glissement ω_r^* à une valeur qui correspond au couple maximum, ce qui permet au moteur d'accélérer jusqu'à atteindre cette consigne. À ce stade, la pulsation de glissement diminue jusqu'à atteindre la valeur correspondante au couple de charge. Ainsi, le système est capable de maintenir une vitesse de rotation constante malgré les perturbations externes (Figure 1.8). [39]

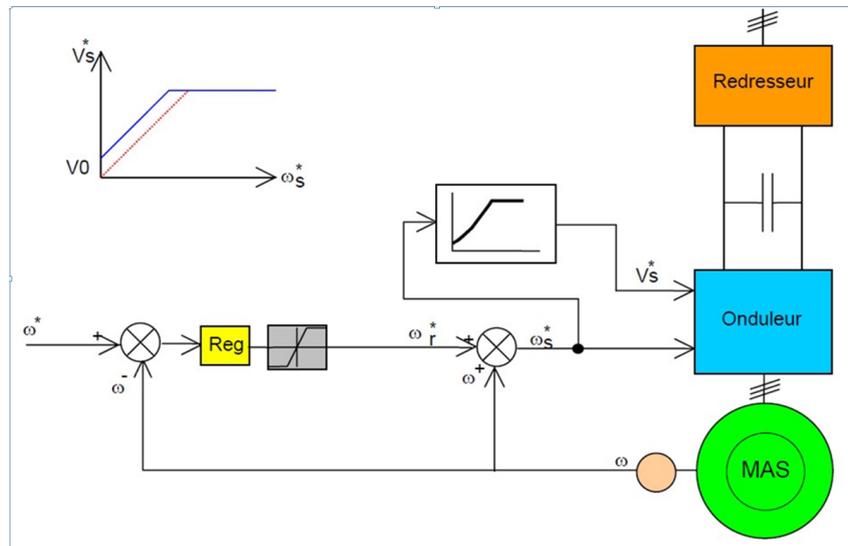


FIGURE 1.8 – Commande scalaire en boucle fermée de la machine [7]

Le schéma de commande présenté ci-dessus décrit la régulation de la vitesse de la machine en utilisant à la fois la pulsation rotorique et la vitesse pour déterminer la pulsation statorique. Le régulateur de vitesse fournit la pulsation rotorique, qui est une mesure du couple produit par la machine. En cas d'augmentation de charge sur le moteur, la vitesse peut diminuer, ce qui entraîne une augmentation du couple fourni par le régulateur (et donc un glissement plus important) pour rétablir l'équilibre. La pulsation statorique est ajustée en conséquence pour maintenir cet équilibre. Enfin,

la tension est calculée pour que le moteur fonctionne en mode de contrôle V/f. [40]

2. Commande vectorielle

La commande vectorielle (ou commande en flux orienté) est une technique de commande de moteurs électriques qui permet un contrôle précis de la vitesse et du couple en régime permanent et transitoire. Elle se base sur un modèle dynamique du moteur asynchrone et permet de contrôler le flux et le couple de manière indépendante. [41]

a) Principe de la commande vectorielle

Le principe de cette commande est de réduire l'équation de couple électromagnétique du moteur afin d'être comparable à celle d'un moteur à courant continu.

Le référentiel (d, q) lié au champ tournant est choisi de telle façon que l'axe (d) coïncide avec la direction du flux (I_{sd}) et que l'axe (q) coïncide avec la direction de la composante (I_{sq}) donnant le couple électromagnétique (Figure 1.9). [31]

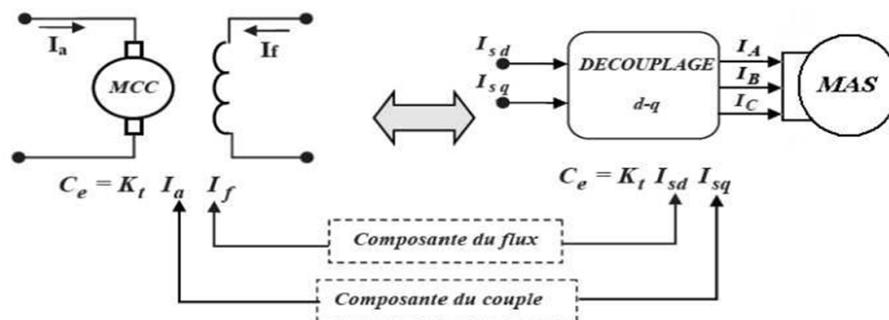


FIGURE 1.9 – Schéma de principe de découplage de la MAS/MAC [7]

Il existe trois types de commande vectorielle à flux orienté selon la nature du flux :

- Commande vectorielle à flux rotorique orienté.
- Commande vectorielle à flux statorique orienté.
- Commande vectorielle à flux de magnétisation orienté.

Le plus fréquemment utilisée est la commande vectorielle par orientation du flux rotorique, puisque dans le cas de la MAS non saturée, on obtient les équations du rotor complètement découplées. [34]

La commande vectorielle nécessite une mesure de la vitesse de rotation de la machine, généralement effectuée à l'aide d'un codeur ou d'un capteur de vitesse. Elle peut être mise en œuvre à l'aide de micro-contrôleurs ou de processeurs numériques

de signal (DSP) qui effectuent les calculs nécessaires pour le contrôle de la machine (Figure 1.10).[42]

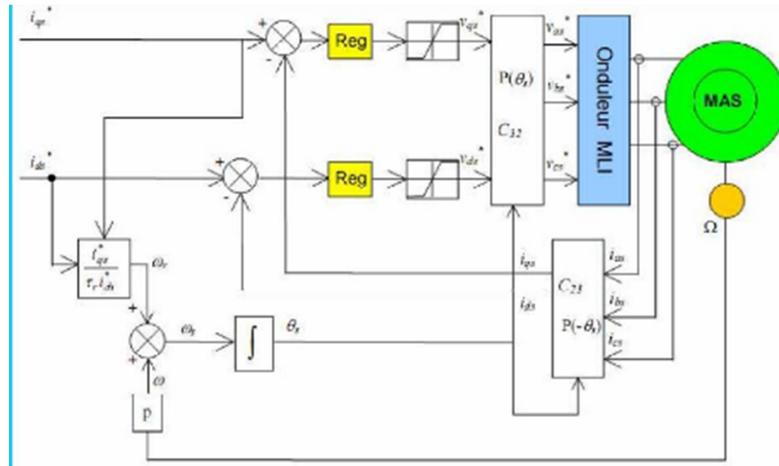


FIGURE 1.10 – Commande vectorielle de la machine asynchrone[8]

Après avoir donné un aperçu général sur les commandes du MAS et leur implémentation sur Matlab Simulink, l'étape suivante dans notre travail de PFE sera son implémentation sur l'API S7 1200. Pour concrétiser notre contribution, une validation de nos résultats obtenus par l'API sera faite en comparant avec ceux trouvés par Matlab. Pour cela, le paragraphe suivant donnera une description sur l'API et son environnement.

1.4 Vu général sur (API) et son environnement

1.4.1 Introduction

Les systèmes électroniques industriels sont devenus des outils essentiels pour améliorer les processus de fabrication et maximiser les performances de production dans les usines modernes. Les équipements proposés par Panasonic Industry Europe sont particulièrement appréciés pour leur taille compacte, leur permettant de s'adapter facilement aux machines et aux centres de commande. Ces systèmes électroniques sont soumis à des tests rigoureux pendant leur développement afin de garantir leur qualité et leur durabilité. En outre, Panasonic Industry Europe propose une garantie de trois ans pour garantir la fiabilité de ses prestataires logistiques (Figure 1.11).



FIGURE 1.11 – API [9]

Les appareils électroniques programmables pour l'environnement industriel permettent l'exécution de fonctions automatisées pour assurer le contrôle des opérateurs à travers des informations logiques, analogiques ou numériques.

1.4.2 Historique

Les Automates Programmables Industriels sont apparus à la fin des années 1960, à la demande de l'industrie automobile américaine (GM), qui réclamait plus d'adaptabilité de leurs systèmes de commande. Les coûts de l'électronique permettent alors de remplacer avantageusement les technologies actuelles s'avérant plus coûteuses ou moins performantes. Les premières applications des API étaient le remplacement des horloges de contrôle du temps des employés, puis leurs multiples utilisations industrielles en particulier sur les lignes de production des usines. De nombreux systèmes à relais ont alors cédé leur place aux API.[43]

Les premiers API n'effectuaient que la commutation ON et OFF (et vice-versa) avec la possibilité de temporisation, comme les relais. De 1970 à 1974, la technologie des microprocesseurs a ajouté une plus grande flexibilité et une « intelligence » à l'automate programmable. Les capacités d'interface avec l'utilisateur se sont améliorées et l'automate programmable a pu exécuter des opérations arithmétiques en plus des opérations logiques, manipuler les données et les adresses et effectuer la communication avec d'autres automates ou ordinateurs.[44]

La console de programmation s'est allié avec un moniteur permettant de programmer avec des symboles de relais familiers, ce qui améliore considérablement la compréhension et le dépannage parce que la logique pouvait être vue dans la même forme que les dessins à relais. Les automates programmables utilisent une mémoire non-volatile pour emmagasiner les instructions, qui accomplissent des fonctions logiques, arithmétiques, de

temporisation, de comptage et de manipulation des données.[45]

Depuis les années 1990, les API se sont développés pour intégrer des fonctionnalités telles que la communication sans fil, la connectivité à distance, la visualisation en temps réel, la collecte de données et l'analyse de données. De nos jours, les automates programmables sont devenus indispensables dans les environnements de fabrication modernes, grâce à leur capacité à assurer la commande de pré-actionneurs et d'actionneurs à partir d'informations logiques, analogiques ou numériques, et à leur compacité permettant une intégration facile dans des espaces limités.[46]

1.4.3 Constructeurs d'API

Dans le monde industriel nous allons voir les différentes gammes d'automates qui existent dans l'environnement industriel. Il existe plusieurs types d'automate programmable industriel parmi ces automates on a :

- Siemens : un des leaders dans le domaine de l'automatisation industrielle, Siemens propose une large gamme d'automates programmables, allant des modèles compacts aux modèles haut de gamme pour les applications les plus exigeantes. Elle permet aux développeurs de créer des programmes qui peuvent accéder aux données et aux fonctionnalités des produits Siemens tels que les automates programmables SIMATIC, les systèmes d'entraînement SINAMICS, les capteurs, les logiciels de supervision, etc. L'API Siemens peut être utilisé pour développer des applications personnalisées, des extensions de logiciels existants, des intégrations de système, des interfaces homme-machine (IHM) personnalisées, des outils de diagnostic, des outils de surveillance, etc. Elle permet également aux développeurs de bénéficier de la flexibilité de la programmation basée sur les normes ouvertes, telles que les protocoles OPC UA (Open Platform Communications Unified Architecture) et PROFINET, pour assurer l'interopérabilité avec d'autres équipements. L'utilisation de l'API Siemens permet de gagner du temps et de l'argent en simplifiant la conception et la mise en œuvre de projets d'automatisation industrielle, tout en augmentant l'efficacité et la précision du système.[47]
- Schneider Electric : également un acteur majeur dans l'industrie de l'automatisation, Schneider Electric propose des automates programmables sous la marque Modicon, ainsi que d'autres solutions pour la gestion de l'énergie et des bâtiments.
- Allen-Bradley (Rockwell Automation) : une autre entreprise bien établie dans le domaine de l'automatisation, Allen-Bradley propose des automates programmables de la gamme ControlLogix pour les applications les plus avancées, ainsi que des

modèles plus accessibles pour les besoins plus simples.

- Mitsubishi Electric : spécialisé dans les équipements électriques et électroniques, Mitsubishi Electric propose des automates programmables de la gamme MELSEC pour les applications industrielles.
- ABB : un autre acteur majeur dans le domaine de l'automatisation industrielle, ABB propose des automates programmables de la gamme AC500 pour les applications les plus exigeantes, ainsi que des modèles plus abordables pour les besoins de base.
- Omron : une entreprise japonaise qui propose une large gamme de produits pour l'automatisation industrielle, y compris des automates programmables de la gamme Sysmac
- Bosch Rexroth : spécialisé dans les systèmes d'entraînement et de commande, Bosch Rexroth propose des automates programmables de la gamme IndraLogic pour les applications industrielles.

Il existe bien sûr de nombreux autres fabricants d'API, chacun proposant des produits et des solutions adaptés à des besoins spécifiques dans divers secteurs industriels.[10]

1.4.4 SIMATIC S7-1200

SIMATIC S7-1200 est un automate programmable industriel (API) compact et modulaire fabriqué par Siemens. Il est conçu pour les applications de contrôle de petite et moyenne envergure dans les industries manufacturières et de production. L'automate SIMATIC S7-1200 est facile à utiliser, flexible et offre des fonctionnalités avancées pour la gestion des processus automatisés. Il est doté d'une gamme de modules d'extension qui permettent aux utilisateurs de personnaliser leur système en fonction de leurs besoins spécifiques (Figure 1.12).



FIGURE 1.12 – S7-1200 SIEMENS [10]

De plus, le SIMATIC S7-1200 est équipé d'une interface utilisateur conviviale qui permet une configuration et une programmation rapides et efficaces. Il est également compatible

avec la plateforme d'ingénierie TIA Portal de Siemens, qui fournit une solution complète pour la configuration, la programmation, la visualisation et la mise en service de l'automate.[48]

1.4.5 Progiciel de développement : TIA Portal de Siemens

a). Définition

Siemens TIA Portal est une plate-forme logicielle d'automatisation qui permet aux utilisateurs de configurer, programmer et commander des équipements d'automatisation et de contrôle. Il a été lancé en 2010 et est conçu pour être facile à utiliser pour les nouveaux utilisateurs ainsi que pour les utilisateurs expérimentés. La plateforme fournit des concepts de fonctionnement standardisés pour le stockage et la cohérence des contrôleurs, des interfaces homme-machine (IHM) et des lecteurs, et des données partagées. TIA Portal est conçu pour permettre aux utilisateurs de gérer tout le cycle de vie de l'automatisation numérique, de la planification numérique à l'ingénierie intégrée en passant par une exploitation transparente. Il est également compatible avec la gestion du cycle de vie des produits (PLM) et la gestion des processus industriels (MES). [48]

b). Logiciel de programmation STEP 7

Le STEP 7 est un environnement de programmation de Siemens utilisé pour programmer les automates programmables industriels (API) de la famille SIMATIC S7. Le STEP 7 fait partie de l'ensemble d'outils TIA (TotallyIntegrated Automation) Portal de Siemens, qui permet une intégration facile de l'ensemble des processus d'automatisation d'une entreprise. Le STEP 7 offre des fonctionnalités de programmation avancées pour les API de Siemens, y compris la programmation en langage graphique Ladder, en langage texte structuré (ST), en liste d'instructions (IL) et en diagramme à blocs fonctionnels (FBD). Il est largement utilisé dans l'industrie pour automatiser les processus de production et pour contrôler les machines et les équipements.[48]

c). Simulateur virtuel d'API : PLCSIM

PLCSIM est un logiciel de simulation de programmation de contrôleur programmable (PLC) développé par Siemens. Il est conçu pour simuler des programmes d'automatisation industrielle sans avoir besoin de matériel de contrôle physique. Voici quelques détails sur PLCSIM :

- Fonctionnalités : PLCSIM permet aux utilisateurs de tester des programmes PLC en mode simulation avant de les télécharger sur un équipement réel. Il peut simuler un grand nombre de périphériques tels que des capteurs, des actionneurs, des moteurs et des interfaces de communication. Les utilisateurs peuvent également créer des scénarios de test pour évaluer le comportement du programme dans des situations réelles.
- Utilisation : PLCSIM est utilisé par les programmeurs et les ingénieurs d'automatisation industrielle pour tester leurs programmes sans la nécessité d'un matériel de contrôle physique. Cela permet de réduire les coûts et les temps d'arrêt associés à l'installation, à la configuration et au dépannage d'un système de contrôle physique.
- Intégration : PLCSIM est intégré à l'environnement de développement logiciel de Siemens, appelé TIA Portal (TotallyIntegrated Automation Portal). Les utilisateurs peuvent facilement transférer des programmes entre le logiciel de programmation et le simulateur de PLC en utilisant une interface utilisateur intuitive.
- Avantages : L'utilisation de PLCSIM permet de réduire les coûts de développement et de mise en service de projets d'automatisation industrielle. Les erreurs de programmation peuvent être détectées et corrigées avant que le programme ne soit téléchargé sur un équipement réel, ce qui réduit les temps d'arrêt et les coûts associés à la maintenance. Les utilisateurs peuvent également tester différents scénarios de fonctionnement du programme pour évaluer les performances et l'efficacité du système.
- Limitations : Bien que PLCSIM soit un outil de simulation de haute qualité, il ne peut pas simuler toutes les situations réelles. Les utilisateurs doivent également s'assurer que les périphériques qu'ils utilisent dans leur programme sont disponibles dans la bibliothèque de périphériques de PLCSIM. Enfin, la simulation n'est pas toujours représentative de la réalité, ce qui peut entraîner des résultats différents de ceux attendus une fois le programme téléchargé sur un équipement réel.[48]

d. Logiciel de développement d'IHM : Simatic WinCC

L'IHM (Interface Homme-Machine) est l'ensemble des éléments permettant à un utilisateur de communiquer avec une machine, un ordinateur ou un système automatisé. L'IHM peut prendre différentes formes, telles que des écrans tactiles, des claviers, des souris, des boutons, des voyants lumineux, des haut-parleurs, des microphones, etc. L'IHM est conçue pour rendre l'utilisation du système plus intuitive, plus ergonomique et plus efficace.[49] L'IHM est particulièrement importante pour les systèmes automatisés et les machines

industrielles, car elle permet aux opérateurs de surveiller et de contrôler le système en temps réel. Les IHM peuvent être utilisées pour surveiller les performances de la machine, afficher des alarmes et des messages d'erreur, et permettre aux opérateurs de modifier les paramètres de la machine en fonction des besoins de la production.[50]

Les principaux objectifs de l'IHM sont les suivants :

- Fournir une interface utilisateur intuitive, facile à utiliser et ergonomique
- Permettre une interaction en temps réel avec les systèmes et les équipements
- Afficher les informations clés de manière concise et facilement compréhensible
- Fournir des avertissements et des alertes en cas d'urgence ou de dysfonctionnement
- Permettre la personnalisation de l'interface utilisateur en fonction des besoins de l'utilisateur.

En résumé, l'IHM est un élément clé pour améliorer l'interaction entre les machines et les utilisateurs dans les environnements industriels et automatisés. WinCC est un outil de développement d'IHM populaire qui permet aux développeurs de créer des interfaces utilisateur personnalisées pour les machines et les systèmes de contrôle, tout en offrant des fonctionnalités avancées de surveillance et de contrôle en temps réel (Figure 1.13).[51]



FIGURE 1.13 – Simatic WinCC [10]

i. WinCC Runtime (RT)

Est un logiciel de visualisation d'interface homme-machine (IHM) développé par Siemens. Il est conçu pour être utilisé avec les systèmes de contrôle-commande industriels de la gamme Siemens Simatic.

Voici les fonctionnalités principales de WinCC Runtime :

1. Interface graphique : WinCC Runtime permet de créer des interfaces graphiques personnalisées pour contrôler les systèmes de contrôle-commande industriels. Les interfaces peuvent inclure des graphiques, des boutons, des tableaux de données,

des alarmes, des courbes, etc.

2. Communication : WinCC Runtime permet de communiquer avec les systèmes de contrôle-commande industriels via différents protocoles de communication tels qu'OPC, Modbus, Profibus, etc.
3. Historique : WinCC Runtime permet de stocker l'historique des données collectées à partir des systèmes de contrôle-commande. Les données peuvent être stockées dans une base de données SQL ou dans des fichiers CSV.
4. Alarmes : WinCC Runtime permet de définir des alarmes pour avertir les utilisateurs lorsqu'un événement critique se produit dans le système de contrôle-commande.
5. Sécurité : WinCC Runtime permet de définir des niveaux d'accès pour les utilisateurs afin de protéger les données et les fonctions critiques du système de contrôle-commande.
6. Multilingue : WinCC Runtime prend en charge plusieurs langues, ce qui permet aux utilisateurs de créer des interfaces graphiques dans différentes langues pour répondre aux besoins de leur marché.
7. Redondance : WinCC Runtime prend en charge la redondance, ce qui signifie que les données peuvent être collectées à partir de plusieurs systèmes de contrôle-commande et stockées dans un serveur redondant pour garantir une haute disponibilité.[\[52\]](#)

ii. WinCC RT Advanced

Simatic WinCC Runtime Advanced est un logiciel d'interface homme-machine (HMI) développé par Siemens pour la visualisation et la supervision de processus industriels. Il permet aux utilisateurs de créer des applications de visualisation pour les systèmes de contrôle-commande et de surveillance en utilisant une interface graphique conviviale. Le logiciel prend en charge les protocoles de communication les plus courants utilisés dans l'industrie, tels que Profibus, Profinet, OPC UA, etc. Il est conçu pour être utilisé en conjonction avec les systèmes d'automatisation industrielle de Siemens et est souvent intégré dans les solutions globales de contrôle-commande de l'entreprise.

iii. WinCC RT Professionnel

SIMATIC WinCC Runtime Professional est un logiciel de visualisation de processus industriel qui permet de créer des interfaces homme-machine (IHM) pour les systèmes de contrôle-commande. Il fait partie de la gamme SIMATIC HMI de Siemens et est conçu pour fonctionner sur les systèmes d'exploitation Windows.

Avec SIMATIC WinCC Runtime Professional, les utilisateurs peuvent concevoir des IHM personnalisées pour afficher les données des processus, les alarmes et les événements, et

contrôler les systèmes automatisés. Il est également possible d'intégrer des fonctions de sécurité et de communication, telles que la connexion à des systèmes de gestion de base de données ou à des réseaux industriels.

Le logiciel est hautement flexible et peut être utilisé dans une large gamme d'applications industrielles, allant de la fabrication et la production à la gestion de l'énergie et des bâtiments. Il est souvent utilisé en combinaison avec d'autres produits Siemens tels que le système d'automatisation SIMATIC S7 et le logiciel de conception de contrôle-commande SIMATIC PCS 7.[53]

SIMATIC WinCC Runtime Professional offre également des fonctionnalités avancées telles que la possibilité de créer des scripts et des macros pour l'automatisation de tâches répétitives, ainsi que des outils de diagnostic pour faciliter la maintenance et le dépannage.[54]

1.5 Conclusion

Ce chapitre présente plusieurs technologies utilisées en ingénierie industrielle, notamment les simulateurs, la machine asynchrone MAS et les automates programmables industriels (API). Les simulateurs sont largement utilisés dans différents secteurs pour simuler des environnements complexes et pour la formation et l'entraînement des opérateurs. La machine asynchrone MAS est l'un des moteurs les plus couramment utilisés dans l'industrie pour son efficacité, sa fiabilité et sa simplicité de conception. En dernier lieu, les systèmes de commande programmables pour l'industrie sont devenus essentiels à la gestion efficace de la production. Les progrès technologiques dans ce domaine ont amélioré considérablement l'efficacité et la sécurité des processus industriels.

Chapitre **2**

Modélisation et implémentation du MAS

But

Le but de ce chapitre est de modéliser un moteur asynchrone à l'aide de TIA Portal. Au début, nous allons créer un modèle de moteur asynchrone à l'aide de Simulink et d'un script, puis nous allons utiliser ce modèle pour modéliser le même moteur dans TIA Portal. Ensuite, nous allons comparer les résultats obtenus dont le but est de valider le modèle s du moteur asynchrone de deux façons différente. Si les résultats sont cohérents et similaires entre les deux modèles, cela indique que les modèles sont valides et précis. Toutefois, si les résultats diffèrent de manière significative, cela peut indiquer qu'il y a des erreurs dans l'un ou les deux modèles et que des ajustements doivent être apportés pour améliorer leur précision. En résumé, la comparaison des résultats entre le modèle de TIA Portal et celui de MATLAB Simulink est une étape importante pour valider la modélisation et s'assurer que les deux modèles sont précis et fiables

2.1 Modélisation du Moteur Asynchrone

2.1.1 Introduction

La modélisation d'un MAS est un procédé mathématique qui permet de représenter le comportement de la machine dans différentes conditions d'exploitation. Cette modélisation est réalisée à partir des équations électriques, mécaniques et magnétiques qui régissent le fonctionnement du machine. Elle peut être réalisée dans un repère diphasé ou triphasé, selon la méthode choisie pour simplifier les calculs et donner des résultats plus faciles à comprendre. Une des étapes clés de la modélisation est la transformation du repère triphasé en un modèle diphasé, grâce à la transformation de Park. Cette transformation permet de modéliser le moteur dans un repère plus adapté pour la commande des machines électriques. Une fois que le modèle est établi, il peut être mis sous forme d'un modèle d'état pour permettre une simulation plus avancée. Ce modèle d'état permet de simuler le comportement de la machine en temps réel, en prenant en compte les interactions entre les différents composants.[55]

Cependant, pour simplifier les calculs et rendre les résultats plus accessibles, des hypothèses simplificatrices sont souvent nécessaires. Ces hypothèses peuvent inclure la supposition que la machine est parfaitement symétrique, que les circuits magnétiques ne sont pas saturés, ou que les pertes joules sont les seules pertes considérer.

Enfin, une fois le modèle de simulation est établi, il peut être utilisé pour optimiser les paramètres de la machine, concevoir des contrôleurs efficaces et prévoir les performances de la machine en temps réel. La modélisation d'un moteur asynchrone est donc un outil essentiel pour comprendre et contrôler ces machines électriques couramment utilisées dans

l'industrie.[56]

2.1.2 Hypothèses Simplificatrices

La modélisation du moteur asynchrone implique l'utilisation de plusieurs hypothèses simplificatrices pour simplifier le système et faciliter les calculs. Ces hypothèses comprennent :

1. Parfaite symétrie de la machine : La machine est supposée être parfaitement symétrique, tant au niveau du stator que du rotor. Cela signifie que les caractéristiques physiques et électriques sont identiques pour les trois phases statoriques et les trois phases rotoriques.
2. Trois phases statoriques identiques : Le stator de la machine est supposé être composé de trois bobines identiques (A, B, C) régulièrement déphasées de 120° électriques. La phase A est souvent prise comme référence angulaire.
3. Trois phases rotoriques identiques : Le rotor est considéré comme équivalent à trois bobines identiques (a, b, c) court-circuitées sur elles-mêmes, également régulièrement déphasées de 120° électriques.
4. Circuits magnétiques non-saturés : Les circuits magnétiques de la machine sont supposés ne pas être saturés, ce qui permet de définir des inductances constantes.
5. Induction sinusoïdale dans l'entrefer : L'induction dans l'entrefer est supposée être à répartition sinusoïdale.
6. Composante homopolaire du courant nulle : La composante homopolaire du courant est supposée être nulle.
7. Pertes joules uniquement : Les pertes joules dans la machine sont prises en compte, mais les pertes fer sont négligées.

Ces hypothèses simplificatrices permettent de simplifier le modèle de la machine asynchrone et de faciliter les calculs nécessaires pour simuler son comportement. Cependant, il est important de noter que ces hypothèses ne sont pas toujours valides pour toutes les machines asynchrones, et peuvent donc entraîner des erreurs dans les résultats de simulation dans certains cas.[57][58]

2.1.3 Etapes de modélisation du MAS

La modélisation des machines électriques à trois phases, comme les moteurs asynchrones, est couramment effectuée en utilisant la méthode triphasée. Cette méthode décrit le système dans un repère triphasé, dans lequel les grandeurs électriques sont représentées par

des vecteurs tournants à la vitesse de synchronisme. Les équations électriques, magnétiques et mécaniques sont exprimées en termes de grandeurs triphasées telles que les tensions et les courants des trois phases statoriques et rotoriques. Cependant, pour simplifier les calculs et l'analyse du système, il est parfois nécessaire de transformer le système triphasé en un système diphasé. Pour ce faire, la transformation de Park, également connue sous le nom de transformation dq0, est couramment utilisée. Cette transformation convertit les grandeurs triphasées en grandeurs diphasées (ou dq) en utilisant deux composantes : une composante directe (d) et une composante en quadrature (q) (Figure 2.1).[59]

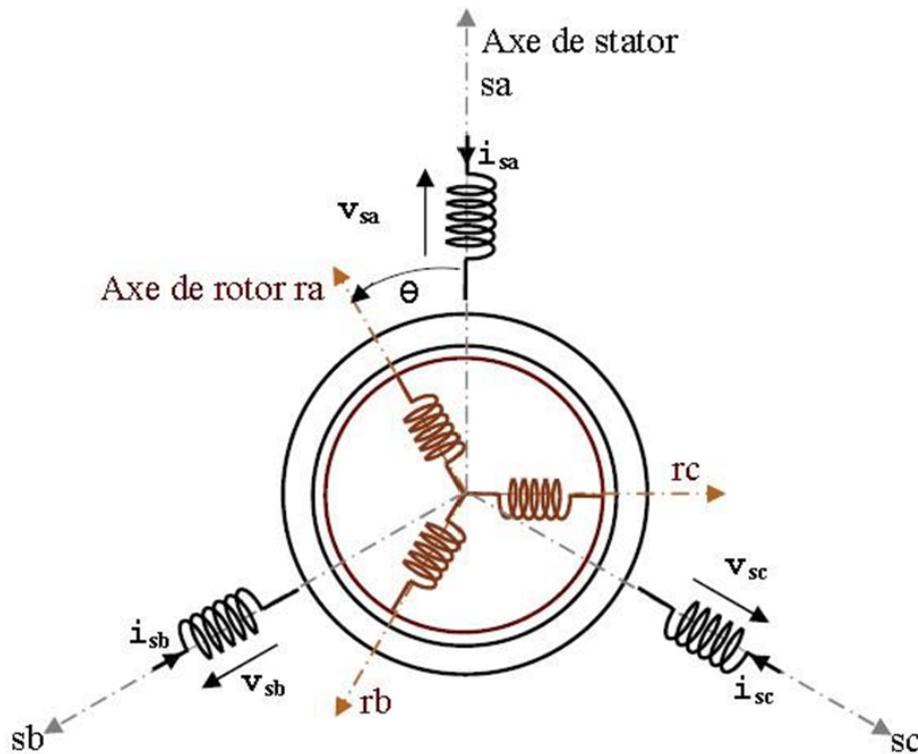


FIGURE 2.1 – Présentation des enroulements statorique et rotorique.[7]

Les moteurs asynchrones peuvent être modélisés par la méthode triphasée ou diphasée et peuvent représenter les grandeurs électriques, mécaniques et magnétiques du système. La transformation de Park est souvent utilisée pour convertir le modèle triphasé en modèle diphasé afin de faciliter les calculs. Ensuite, le modèle peut être représenté sous forme d'un ensemble d'équations différentielles, appelé modèle d'état, pour simuler le comportement de la machine dans un environnement de simulation tel que MATLAB/Simulink. Ces outils sont indispensables pour la modélisation et la simulation efficaces de la machine asynchrone.[59]

2.1.4 Equations générales du moteur

1. Équations électriques du MAS dans le repère (abc)

Les équations générales de la machine asynchrone à rotor bobiné dans le repère triphasé s'écrivent comme suit :

Stator :

$$\begin{cases} v_{as} = R_s I_{as} + \frac{d\varphi_{as}}{dt} \\ v_{bs} = R_s I_{bs} + \frac{d\varphi_{bs}}{dt} \\ v_{cs} = R_s I_{cs} + \frac{d\varphi_{cs}}{dt} \end{cases} \quad (2.1)$$

Rotor :

$$\begin{cases} v_{ar} = R_r I_{ar} + \frac{d\varphi_{ar}}{dt} \\ v_{br} = R_r I_{br} + \frac{d\varphi_{br}}{dt} \\ v_{cr} = R_r I_{cr} + \frac{d\varphi_{cr}}{dt} \end{cases} \quad (2.2)$$

La représentation adoptée est la représentation suivante qui est la représentation d'état.

$$[V_s] = [R_s] [I_s] + \frac{d}{dt} [\varphi_s] \quad (2.3)$$

$$[V_r] = [R_r] [I_r] + \frac{d}{dt} [\varphi_r] \quad (2.4)$$

Avec :

$[V_s] [V_r]$: tension simple statorique et rotorique.

$[I_s] [I_r]$: courants de phase statorique rotorique.

$[\varphi_s] [\varphi_r]$: flux statorique et rotorique.

$[R_s] [R_r]$: Matrices des résistances par phase statorique et rotorique.

2. Equations magnétiques

De la même manière, l'écriture matricielle des équations magnétiques est donnée par :

$$[\varphi_s] = [L_s] [I_s] + [M] [I_r] \quad (2.5)$$

$$[\varphi_r] = [L_r] [I_r] + [M] [I_s] \quad (2.6)$$

En remplaçant les équations (2.5) et (2.6) dans (2.3) et (2.4) respectivement, on obtient le système d'équations ci-dessous :

$$[V_s] = [R_s] [I_s] + [L_s] \frac{d}{dt} [I_s] + \frac{d}{dt} ([M] [I_r]) \quad (2.7)$$

$$[V_r] = [R_r] [I_r] + [L_T] \frac{d}{dt} [I_r] + \frac{d}{dt} ([M] [I_s]) \quad (2.8)$$

Tel que :

$$V_s = [V_{as}; V_{bs}; V_{cs}]$$

$$V_r = [V_{ar}; V_{br}; V_{cr}]$$

$$\varphi_r = [\varphi_{ar}; \varphi_{br}; \varphi_{cr}]$$

$$\varphi_s = [\varphi_{as}; \varphi_{bs}; \varphi_{cs}]$$

$$I_s = [I_{as}; I_{bs}; I_{cs}]$$

$$I_r = [I_{ar}; I_{br}; I_{cr}]$$

$$[R_s] = \begin{bmatrix} R_S & 0 & 0 \\ 0 & R_S & 0 \\ 0 & 0 & R_S \end{bmatrix}, [R_r] = \begin{bmatrix} R_r & 0 & 0 \\ 0 & R_r & 0 \\ 0 & 0 & R_r \end{bmatrix}$$

$$[L_s] = \begin{bmatrix} L_{as} & M_{as} & M_{as} \\ M_{as} & L_{as} & M_{as} \\ M_{as} & M_{as} & L_{as} \end{bmatrix}, [L_r] = \begin{bmatrix} L_{ar} & M_{ar} & M_{ar} \\ M_{ar} & L_{ar} & M_{ar} \\ M_{ar} & M_{ar} & L_{ar} \end{bmatrix}$$

$$[M_{sr}] = [M_{rs}] = \begin{bmatrix} \cos(\theta) & \cos\left(\theta + \frac{2\pi}{3}\right) & \cos\left(\theta - \frac{2\pi}{3}\right) \\ \cos\left(\theta - \frac{2\pi}{3}\right) & \cos(\theta) & \cos\left(\theta + \frac{2\pi}{3}\right) \\ \cos\left(\theta + \frac{2\pi}{3}\right) & \cos\left(\theta - \frac{2\pi}{3}\right) & \cos(\theta) \end{bmatrix}$$

$[L_s], [L_r]$: matrices d'inductances statoriques et rotoriques.

$[M]$: matrices des inductances mutuelles stator-rotor et celle des inductances mutuelles rotorstator.

M_{as} : Inductance mutuelle entre phases statoriques.

M_{ar} : Inductance mutuelle entre phases rotoriques.

L_{as}, L_{ar} : inductances propres d'une phase statorique et rotorique.

3. Equation mécanique

Le couple électromagnétique s'exprime par :

$$C_{em} = C_r + f\Omega + J \frac{d\Omega}{dt} \quad (2.9)$$

C_{em}, C_r : couple résistant et le couple électromagnétique.

F : coefficient de frottement visqueux.

J : moment d'inertie de la partie tournante.

4. Transformation de Park

La transformation de Park, également connue sous le nom de transformation dq0, est une technique mathématique souvent utilisée en électricité et en électronique pour simplifier l'analyse et les calculs des systèmes électriques. Cette transformation est particulièrement utile dans la modélisation et la commande des machines électriques à trois phases, comme la machine asynchrone. Elle permet de convertir les grandeurs triphasées, telles que les tensions et les courants statoriques et rotoriques, en grandeurs diphasées (ou dq), qui sont plus faciles à manipuler. La transformation de Park est basée sur le concept de la projection d'un vecteur sur un plan orthogonal composé des axes d et q, qui représentent respectivement la composante directe et la composante en quadrature des grandeurs diphasées (Figure 2.2). [59]

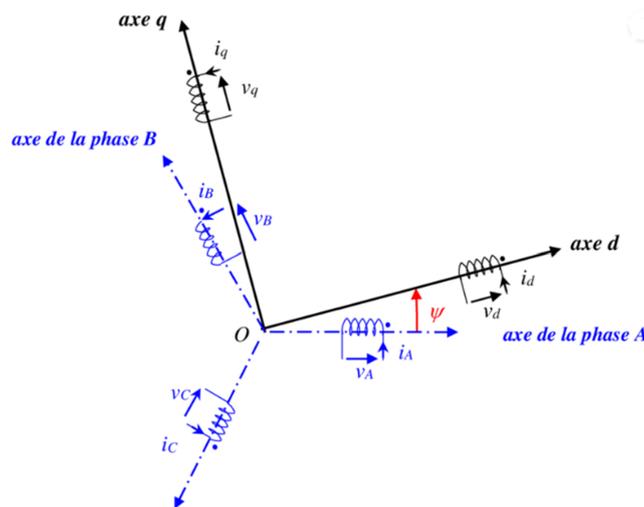


FIGURE 2.2 – Principe de la transformation de Park [6]

La transformation de Park utilise des équations trigonométriques pour calculer les composantes d et q des grandeurs triphasées. Ces équations sont basées sur les angles électriques entre les phases statoriques et rotoriques, ainsi que sur l'angle de rotation de la machine asynchrone par rapport au repère triphasé.

La transformation de Park permet donc de convertir les grandeurs triphasées en grandeurs diphasées, ce qui facilite la modélisation et la commande des machines électriques à trois phases. En utilisant cette modification de repère, il devient possible d'écrire le système d'une manière plus simple et plus intuitive et de concevoir un contrôle plus efficace du moteur asynchrone. La matrice de passage $P(\phi)$ en Ma-

trice de Park est donnée par :

$$P(\theta) = C \begin{bmatrix} \cos(\theta) & \cos\left(\theta - \frac{2\pi}{3}\right) & \cos\left(\theta + \frac{2\pi}{3}\right) \\ -\sin(\theta) & -\sin\left(\theta - \frac{2\pi}{3}\right) & -\sin\left(\theta + \frac{2\pi}{3}\right) \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}$$

$$[V_{dq0}] = [P(\theta)]^* [V_{abc}]$$

$$[V_{dq0}] = C \begin{bmatrix} \cos(\theta) & \cos\left(\theta - \frac{2\pi}{3}\right) & \cos\left(\theta + \frac{2\pi}{3}\right) \\ -\sin(\theta) & -\sin\left(\theta - \frac{2\pi}{3}\right) & -\sin\left(\theta + \frac{2\pi}{3}\right) \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} * \begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix}$$

5. Modélisation du moteur asynchrone

a) Représentations d'état du moteur asynchrone

La représentation dépend en premier lieu du référentiel choisi. Si on considère le repère tournant asynchrone, l'avantage est d'avoir des grandeurs constantes en régime permanent. Il est alors plus aisé d'en faire la régulation.

Les équations du modèle du moteur s'écrivent comme suit :

$$\begin{aligned} \frac{d\theta_r}{dt} &= \omega \\ \frac{d\theta_s}{dt} &= \omega_s \\ \omega_r &= \omega_s - \omega \end{aligned} \tag{2.10}$$

L'expression des tensions statoriques et rotoriques suivant les axes d et q est :

$$\begin{cases} V_{ds} = R_s I_{ds} + \frac{d\varphi_{ds}}{dt} - \varphi_{qs} \frac{d\theta_s}{dt} \\ V_{qs} = R_s I_{qs} + \frac{d\varphi_{qs}}{dt} + \varphi_{ds} \frac{d\theta_s}{dt} \end{cases} \tag{2.11}$$

$$\begin{cases} 0 = R_r I_{dr} + \frac{d\varphi_{dr}}{dt} - \varphi_{qr} \frac{d\theta_r}{dt} \\ 0 = R_r I_{qr} + \frac{d\varphi_{qr}}{dt} + \varphi_{dr} \frac{d\theta_r}{dt} \end{cases} \tag{2.12}$$

b) Equations magnétiques

L'expression des flux statoriques suivant les axes d et q est comme suite :

$$\begin{cases} \varphi_{ds} = L_s I_{ds} + M I_{dr} \\ \varphi_{qs} = L_s I_{qs} + M I_{qr} \end{cases} \quad (2.13)$$

$$\begin{cases} \varphi_{dr} = L_r I_{dr} + M I_{ds} \\ \varphi_{qr} = L_r I_{qr} + M I_{qs} \end{cases} \quad (2.14)$$

c) Couple électromagnétique

L'expression du couple électromagnétique en fonction des flux statoriques et des courants rotoriques est donnée par :

$$C_{em} = P (\varphi_{ds} I_{qs} - \varphi_{qs} I_{ds}) \quad (2.15)$$

$$C_{em} = P \frac{M}{L_r} (\varphi_{dr} I_{qs} - \varphi_{qr} I_{ds}) \quad (2.16)$$

$$C_{em} = \frac{PM}{L_s} (\varphi_{qs} I_{dr} - \varphi_{ds} I_{qr}) \quad (2.17)$$

Avec

$\sigma = 1 - \frac{M^2}{L_s L_r}$: Le coefficient de la fuite totale.

$T_S = \frac{L_s}{R_s}, T_T = \frac{L_r}{R_r}$: Constante de temps statorique et rotorique

Après arrangement des équations, on obtient :

$$\left\{ \begin{array}{l} \frac{dI_{ds}}{dt} = -\frac{1}{T_s\sigma} I_{ds} + \left(\omega + \frac{1}{\sigma}\omega_r\right) I_{qs} + \frac{M}{L_s T_r \sigma} I_{dr} + \frac{M}{L_s \sigma} \omega_r I_{qr} + \frac{1}{L_s \sigma} V_{ds} \\ \frac{dI_{qs}}{dt} = -\left(\omega + \frac{1}{\sigma}\omega_r\right) I_{ds} - \frac{1}{T_s\sigma} I_{qs} - \frac{M}{L_s \sigma} \omega_r I_{dr} + \frac{M}{L_s T_r \sigma} I_{qr} + \frac{1}{L_s \sigma} V_{qs} \\ \frac{dI_{dr}}{dt} = \frac{M}{L_r T_s \sigma} I_{ds} - \frac{M}{L_r \sigma} \omega_r I_{qs} - \frac{1}{T_r \sigma} I_{rr} + \left(\omega - \frac{M^2}{L_s L_r \sigma} \omega_r\right) I_{qr} - \frac{M}{L_s L_r \sigma} V_{ds} \\ \frac{dI_{qr}}{dt} = \frac{M}{L_r \sigma} \omega_r I_{ds} + \frac{M}{L_r T_s \sigma} I_{qs} + \left(-\omega + \frac{M^2}{L_s L_r \sigma} \omega_r\right) I_{dr} - \frac{1}{T_r \sigma} I_{qr} - \frac{M}{L_s L_r \sigma} V_{qs} \end{array} \right. \quad (2.18)$$

Mise sous forme de modèle d'état :

$$\frac{dx}{dt} = [A][x] + [B][V]$$

$$[Y] = [C][x] + [D][V]$$

Avec

[A] : matrice fondamentale qui caractérise le système.

[B] : matrice d'entrée ou la commande.

[C] : matrice de sortie.

[D] : matrice de transmission direct.

[V] : vecteur de commande.

[x] : vecteur d'état.

[Y] : vecteur de sortie.

Donc, on peut réécrire le système d'équation d'état comme suit :

$$\begin{aligned} [V] &= \begin{bmatrix} V_{ds} \\ V_{qs} \\ 0 \\ 0 \end{bmatrix} = [B][I] = \begin{bmatrix} I_{ds} \\ I_{qs} \\ I_{dr} \\ I_{qr} \end{bmatrix} = [x][R] = \begin{bmatrix} R_s & 0 & 0 & 0 \\ 0 & R_s & 0 & 0 \\ 0 & 0 & R_r & 0 \\ 0 & 0 & 0 & R_r \end{bmatrix} [L] \\ &= \begin{bmatrix} L_s & 0 & M & 0 \\ 0 & L_s & 0 & M \\ M & 0 & L_r & 0 \\ 0 & M & 0 & L_T \end{bmatrix} \end{aligned}$$

la Représentation d'état du moteur asynchrone devient comme suit :

$$[A] = \begin{bmatrix} -\frac{1}{T_s\sigma} & \left(\omega + \frac{1}{\sigma}\omega_r\right) & \frac{M}{L_s T_r\sigma} & \frac{M}{L_s\sigma}\omega_r \\ -\left(\omega + \frac{1}{\sigma}\omega_r\right) & -\frac{1}{T_s\sigma} & -\frac{M}{L_s\sigma}\omega & \frac{M}{L_s T_r\sigma} \\ \frac{M}{L_r T_s\sigma} & -\frac{M}{L_r\sigma}\omega & -\frac{1}{T_r\sigma} & \left(\omega - \frac{M^2}{L_s L_r\sigma}\omega_r\right) \\ \frac{M}{L_r\sigma}\omega & \frac{M}{L_r T_s\sigma} & \left(-\omega + \frac{M^2}{L_s L_r\sigma}\omega_r\right) & -\frac{1}{T_r\sigma} \end{bmatrix}$$

On peut réécrire la matrice A comme suit :

$$[A] = [A1] + \omega_r[A2] + \omega[A3]$$

Avec

$$[A1] = \begin{bmatrix} -\frac{1}{T_s\sigma} & 0 & \frac{M}{L_s T_r\sigma} & 0 \\ 0 & -\frac{1}{T_s\sigma} & 0 & \frac{M}{L_s T_r\sigma} \\ \frac{M}{L_r T_s\sigma} & 0 & -\frac{1}{T_r\sigma} & 0 \\ 0 & \frac{M}{L_r T_s\sigma} & 0 & -\frac{1}{T_r\sigma} \end{bmatrix}$$

$$[A3] = \begin{bmatrix} 0 & \frac{1}{\sigma} & 0 & \frac{M}{L_s\sigma} \\ -\frac{1}{\sigma} & 0 & -\frac{M}{L_s\sigma} & 0 \\ 0 & -\frac{M}{L_r\sigma} & 0 & -\frac{M^2}{L_s L_r\sigma} \\ \frac{M}{L_r\sigma} & 0 & \frac{M^2}{L_s L_r\sigma} & 0 \end{bmatrix}$$

$$[A2] = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

$$[B] = \begin{bmatrix} \frac{1}{L_s\sigma} & 0 & 0 & 0 \\ 0 & \frac{1}{L_s\sigma} & 0 & 0 \\ -\frac{M}{L_s L_r\sigma} & 0 & 0 & 0 \\ 0 & -\frac{M}{L_s L_r\sigma} & 0 & 0 \end{bmatrix}$$

Après avoir terminé de modéliser le moteur asynchrone, nous allons l'intégrer dans l'environnement MAtlab-Simulink pour vérification.

d) Schéma-bloc du moteur asynchrone sous Simulink

Le schéma bloc de la Figure 2.3 représente les différents paramètres sous forme matricielle et ainsi que les grandeurs électriques et mécaniques nécessaires pour la vérification de l'adéquation du modèle développé.

Dans notre première étape de simulation du moteur asynchrone, nous prenons les valeurs réelles de $v_d=311$ et $v_q=0$ comme tension d'alimentation du moteur. En utilisant les matrices L , A_1 , A_2 , A_3 et B fournies, nous avons calculé le flux et le couple du moteur. Après avoir calculé le flux et le couple, nous utilisons ces valeurs pour obtenir les nouveaux courants statorique et rotorique à travers les équations et matrices appropriées. Ces courants sont essentiels pour déterminer la réponse du moteur à la tension d'alimentation et pour calculer d'autres grandeurs physiques telles que la vitesse. En résumé, en utilisant la matrice de tension d'alimentation, L , A_1 , A_2 , A_3 et B et les équations de flux, de couple et de vitesse, nous sommes en mesure de simuler et d'obtenir le courant et la vitesse actualisés du moteur asynchrone.

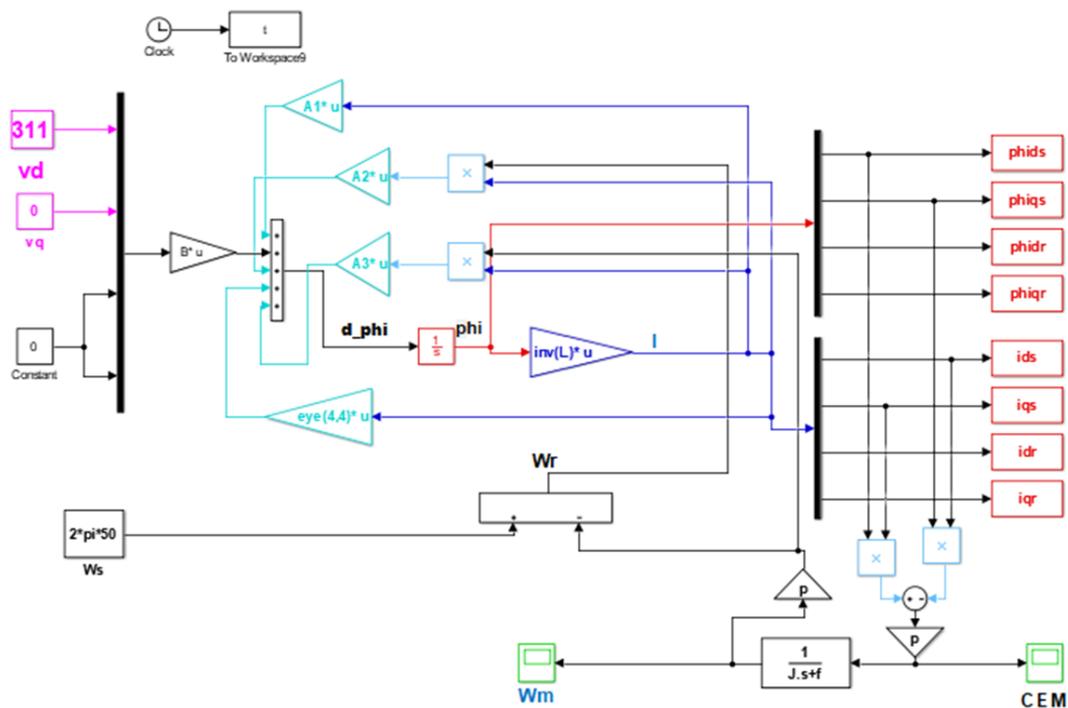


FIGURE 2.3 – Schéma-bloc du moteur asynchrone sous Simulink

Au début de la simulation, le moteur asynchrone a une vitesse mécanique de 0 rad/s, ce qui est attendu puisque le moteur est initialement arrêté.

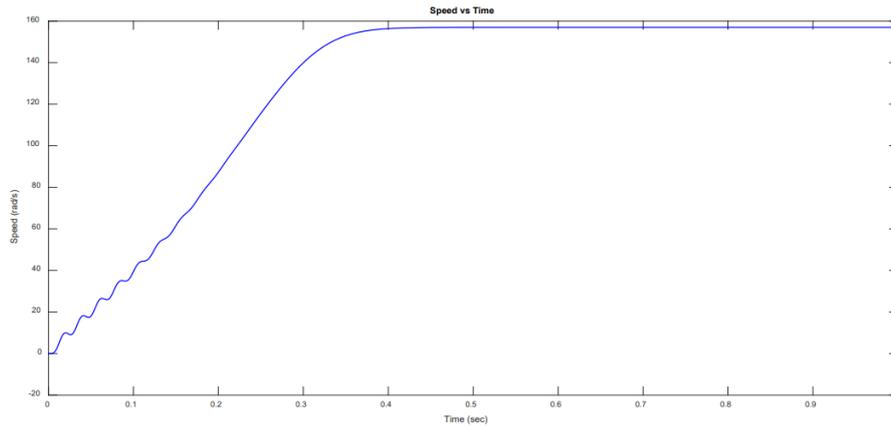


FIGURE 2.4 – Vitesse angulaire de rotation d'un MAS (rad/s).

Puis, au fur et à mesure que le temps passe à 0,4 s, la Vitesse de rotation augmente progressivement. Pendant ce temps, la Vitesse de rotation a augmenté linéairement de 0 rad/s à 156.1 rad/s. Cela montre que le moteur asynchrone accélère progressivement jusqu'à atteindre sa vitesse maximale de 156.1 rad/s. Une fois la vitesse maximale de 0,4 seconde atteinte, la courbe de vitesse de rotation se stabilise et reste constante à 156.1 rad/s. Cela signifie que le moteur asynchrone a atteint sa vitesse de fonctionnement et continue de tourner à cette vitesse constante (Figure 2.4).

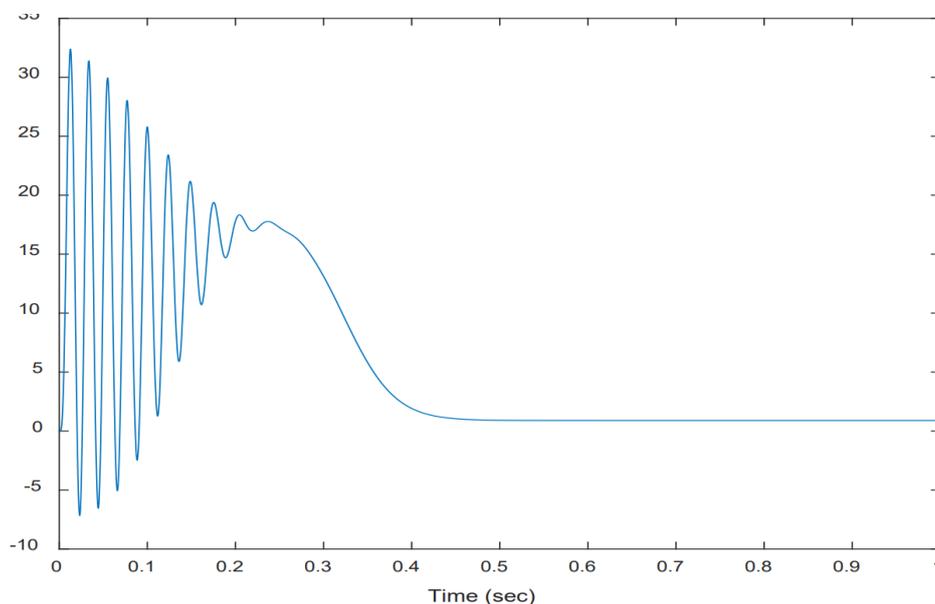


FIGURE 2.5 – Réponse du couple électromagnétique (CEM) d'un MAS

Pendant le démarrage, le couple électromagnétique peut présenter des oscillations en rai-

son des conditions transitoires du moteur. Pendant cette phase, le couple est crucial pour accélérer la vitesse de rotation de manière linéaire. Le moteur fournit un couple maximal pour assurer cette augmentation de vitesse. Cependant, une fois que la vitesse de rotation du moteur se rapproche du régime permanent, le couple électromagnétique commence à diminuer. Cela est dû aux caractéristiques de fonctionnement du moteur et à l'équilibre entre les forces électromagnétiques et les forces mécaniques. Finalement, lorsque la vitesse de rotation atteint sa valeur constante en régime permanent, le couple électromagnétique devient presque nul. (Au régime permanent, le couple électromagnétique d'un moteur n'est nul que si les pertes par frottement sont négligeables) (Figure 2.5).

2.2 Implémentation du modèle du MAS dans d'API

2.2.1 Introduction

Les dispositifs électroniques programmables connus sous le nom d'automates programmables industriels (API) sont souvent utilisées pour automatiser les processus de fabrication industrielle. Ils permettent le contrôle et la surveillance des machines, des systèmes de contrôle et des processus de production en exécutant les processus de manière autonome et fiable. Les automates programmables se déclinent en de nombreux types différents, dont la taille et la fonction varient, des petits dispositifs compacts pour les systèmes de contrôle simples aux grands dispositifs modulaires pour les installations industrielles complexes. Ils sont équipés de différentes entrées/sorties (E/S) pour répondre aux exigences spécifiques de chaque application. Les API sont programmés avec des logiciels spéciaux tels que Siemens STEP 7, TIA Portal, Rockwell Automation RSLogix et Codesys pour créer des programmes de contrôle personnalisés pour chaque application. Ces dispositifs sont utilisés dans divers secteurs industriels tels que la fabrication, l'énergie, l'automobile, la pétrochimie, la pharmacie et l'agroalimentaire, pour n'en citer que quelques-uns. Ils offrent de nombreux avantages tels que l'augmentation de la productivité, la réduction des coûts d'exploitation, l'amélioration de la qualité des produits, la réduction des risques d'accidents et une plus grande flexibilité dans les processus de production.

2.2.2 TIA Portal V15.1

TIA Portal V15.1 est une amélioration de l'outil d'ingénierie de Siemens avec de nouvelles fonctions axées sur les tests pour toutes les phases, de la planification à l'ingénierie en passant par la mise en service. Les améliorations comprennent de nouvelles options de

simulation et de mise en service virtuelle, qui permettent une meilleure conception numérique des processus de travail intégrés. Les contrôleurs Simatic S7-1200R/H, les variateurs Sinamics S210, l'ingénierie multi-utilisateur, les unités logicielles et les fonctions OPC UA sont également pris en charge.

L'utilisation de Simatic S7-PLCSIM Advanced permet de créer un jumeau numérique du contrôleur Simatic S7-1200 dans le TIA Portal pour la simulation et la mise en service virtuelle. Le Simatic Machine Simulator V1.0 combine ce jumeau numérique avec le logiciel de simulation NX Mechatronics Concept Designer (NX MCD) pour les concepts de machines mécatroniques, permettant la vérification virtuelle de machines complètes. La mise en service des contrôleurs S7-1200 est également plus rapide et plus efficace qu'auparavant grâce à l'ingénierie multi-utilisateur et à l'utilisation d'unités logicielles (Figure 2.6).[48]



FIGURE 2.6 – Tia portal V15.1 et le S7-1200 [10]

Les options d'application étendues incluent la communication contrôleur-contrôleur avec les fonctions OPC UA du contrôleur S7-1500, l'intégration du variateur de vitesse Sinamics S210 et le test d'acceptation des fonctions de sécurité du convertisseur Sinamics S120.

PLCSIM V15.1 permet de simuler et de tester des programmes de commande pour les automates programmables Simatic S7-1500 et S7-1200. Avec PLCSIM V15.1, les utilisateurs peuvent tester des programmes SPS sans avoir besoin d'un automate réel, ce qui permet de gagner du temps et de réduire les coûts de développement et de test. Le logiciel est compatible avec le logiciel de programmation Siemens TIA Portal et peut être utilisé pour simuler des applications en mode hors ligne (c'est-à-dire sans connexion à un automate réel) ou en mode connecté (avec une connexion à un automate réel)[49].

2.2.3 Système d'exploitation et programme utilisateur

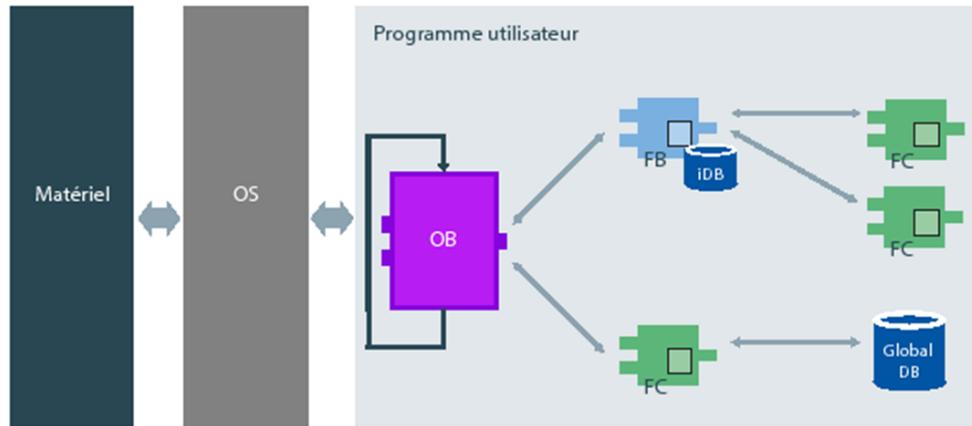


FIGURE 2.7 – Interaction entre le système d'exploitation et le programme utilisateur[10]

A. Système d'exploitation (OS) :

Chaque CPU contient un système d'exploitation qui organise toutes les fonctions et processus de la CPU n'étant pas liés à une tâche d'automatisation spécifique.

Les tâches faisant partie du système d'exploitation sont :

1. Déroulement du démarrage
2. Actualisation des 02 mémoires images : Entrées/Sorties
3. Appel du programme utilisateur
4. Acquisition des alarmes et appels des OB d'alarme
5. Détection et traitement d'erreurs
6. Gestion des zones de mémoire

Le système d'exploitation est un composant de la CPU et est déjà installé dans la CPU à la livraison.[10]

B. Programme utilisateur

Le programme utilisateur contient toutes les fonctions requises pour le traitement de tâches d'automatisation spécifiques.

Les fonctions du programme utilisateur sont :

1. Vérification des conditions préalables au démarrage à l'aide d'OB de démarrage, fin de course en bonne position ou dispositif de sécurité activé par exemple

2. Traitement des données du processus, par exemple combinaison de signaux binaires, lecture et évaluation de valeurs analogiques, détermination des signaux binaires de sortie, sortie des valeurs analogiques
3. Réaction aux alarmes, par exemple alarme de diagnostic en cas de dépassement de la limite inférieure d'un module d'extension analogique
4. Traitement des perturbations dans l'exécution normale du programme [50]

2.2.4 Choix d'automate programmable

Pour choisir un automate programmable industriel, l'automaticien doit préciser les différents besoins et contraintes de l'application à automatiser, notamment :

1. Le nombre et la nature des entrées et des sorties nécessaires pour contrôler les différents équipements de l'installation.
2. Le type de programmation souhaitée, qui peut être Ladder, Instruction List, Structuré Text ou encore Graphique Fonctionnel, selon les compétences et préférences de l'automaticien. Les besoins de traitement déterminent également la puissance de l'unité centrale et la taille de la mémoire utilisateur.
3. La nature des traitements nécessaires, tels que les temporisations, les compteurs, les comparateurs, les opérations mathématiques, etc.
4. Le dialogue avec l'automate, qui peut se faire via une console, un ordinateur ou une tablette, et qui peut nécessiter des compétences spécifiques pour programmer et interagir avec l'automate.
5. Les besoins de communication avec d'autres systèmes, tels que les capteurs, les actionneurs, les équipements de supervision et de contrôle-commande, qui peuvent nécessiter des protocoles spécifiques tels que Profibus, Modbus, Ethernet, etc.
6. La fiabilité et la robustesse de l'automate, qui doivent être adaptées aux conditions environnementales (poussière, humidité, température, etc.) et aux exigences de disponibilité et de sécurité de l'application à automatiser.[10]

CPU 1214 utilisé

Le module CPU 1214C de l'automate programmable S7-1200 de Siemens est une option populaire pour le contrôle industriel en raison de sa facilité d'utilisation, de sa polyvalence et de son coût abordable. Sa programmabilité le rend également très adaptable, permettant une personnalisation des fonctionnalités pour répondre aux besoins spécifiques de l'application (Figure 2.8).



FIGURE 2.8 – S7-1200 1214C DC/DC/DC [10]

Le module offre 14 entrées numériques et 10 sorties numériques, ainsi que 2 entrées analogiques et 1 sortie analogique. La présence de compteurs rapides (HSC) permet une mesure précise de fréquences élevées. Grâce à ses E/S, cet automate peut contrôler et surveiller divers équipements industriels tels que des moteurs, des pompes, des vannes, des capteurs, etc. Les entrées analogiques permettent la mesure de grandeurs physiques telles que la température, la pression, etc. La sortie analogique contrôle des équipements qui nécessitent une tension ou un courant variable, tels que des vannes proportionnelles ou des actionneurs hydrauliques. En somme, le CPU 1214C est une option fiable pour de nombreuses applications de contrôle industriel nécessitant des fonctionnalités E/S numériques et analogiques, ainsi que des compteurs rapides.[10]

2.2.5 Blocs de programme

Il existe plusieurs types de blocs pour exécuter les tâches dans un système d'automatisation comme le montre la Figure 2.9 . Voici une brève description de chacun d'entre eux.

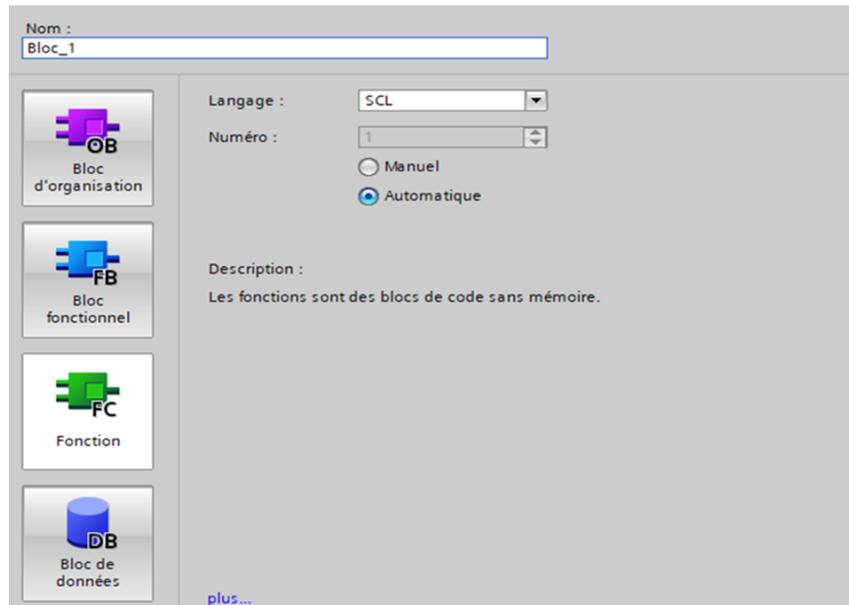


FIGURE 2.9 – Schéma regroupant les différents bloc et fonctions au niveau de la CPU

A. Blocs d'organisation (OB)

les OB (blocs d'organisation) sont des éléments logiciels qui constituent l'interface entre le système de fonctionnement et le programme utilisateur dans les systèmes d'automatisation. Ils sont responsables de plusieurs fonctions clés telles que le traitement cyclique du programme, la gestion des erreurs, la gestion des alarmes, etc. Les blocs d'organisation peuvent être programmés pour personnaliser le comportement de la CPU en fonction des besoins spécifiques du système d'automatisation. Chaque CPU possède ses propres blocs d'organisation qui peuvent être utilisés pour programmer le comportement du système. Les informations de démarrage des blocs d'organisation peuvent également être utilisées pour personnaliser le programme utilisateur en fonction des conditions de démarrage du système.

L'OB "Main" contient le programme utilisateur complet. Il adopte en général la programmation structurée.

1. Programmation structurée

La réalisation et la maintenance de tâches d'automatisation complexes sont plus simples si ces tâches sont divisées en plusieurs tâches partielles plus petites qui correspondent aux fonctions technologiques du processus d'automatisation ou qui peuvent être utilisées plusieurs fois. Dans le programme utilisateur, il y a des blocs qui représentent ces tâches partielles. Chaque bloc constitue une section indépendante du programme utilisateur.

La structuration du programme offre les avantages suivants :

- La programmation de programmes volumineux est plus claire.
- Certaines parties du programme peuvent être normalisées et être utilisées plusieurs fois avec des paramètres changeants.
- L'organisation du programme est simplifiée.
- Il est plus facile de modifier le programme.
- Le test du programme est simplifié, car il peut s'effectuer section par section.
- La mise en service est simplifiée.

La figure 2.10 représente un programme structuré schématique. L'OB de cycle "Main" appelle successivement des sous-programmes exécutant des tâches partielles définies.

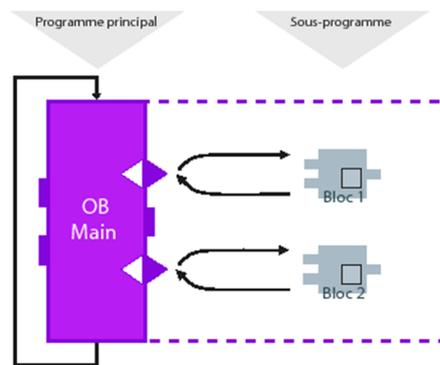


FIGURE 2.10 – Procédure d'appel successive des sous-programmes

B. Fonctions (FC)

1. Définition

Les fonctions (FC) sont des blocs de code sans mémoire. Elles n'ont pas de mémoire de données dans laquelle il est possible d'enregistrer les valeurs de paramètres de bloc. C'est pourquoi des paramètres effectifs doivent être fournis à tous les paramètres formels lors de l'appel d'une fonction.

Pour enregistrer les données durablement, les fonctions disposent de blocs de données globaux.

2. Domaine d'application

Une fonction contient un programme qui est exécuté lorsque la fonction est appelée par un autre bloc de code. Les fonctions peuvent par exemple servir dans les cas suivants :

- Retourner des valeurs de fonction au bloc appelant, par ex. pour les fonctions mathématiques
- Exécuter des fonctions technologiques, par ex. commandes uniques avec combinaisons binaires

Une fonction peut être appelée plusieurs fois à différents endroits d'un programme. Ainsi vous simplifiez la programmation de fonctions utilisées fréquemment.

C. Blocs fonctionnels (FB)

1. Définition

Les blocs fonctionnels sont des blocs de code qui mémorisent durablement leurs paramètres d'entrée, de sortie et d'entrée/sortie dans des blocs de données d'instance afin qu'il soit possible d'y accéder même après le traitement de blocs. C'est pourquoi ils sont également appelés "Blocs avec mémoire".

Les blocs fonctionnels peuvent aussi travailler avec des variables temporaires. Cependant, les variables temporaires ne sont pas enregistrées dans la DB d'instance mais disponibles uniquement tout le temps d'un cycle.

2. Domaine d'application

Les blocs fonctionnels contiennent des sous-programmes qui sont exécutés lorsqu'un bloc fonctionnel est appelé par un autre bloc de code. Un bloc fonctionnel peut être appelé plusieurs fois à différents endroits d'un programme. Ainsi vous simplifiez la programmation de fonctions utilisées fréquemment.

3. Instances de blocs fonctionnels

Un appel d'un bloc fonctionnel est désigné par le terme "instance". Pour chaque instance d'un bloc fonctionnel, il faut un bloc de données d'instance dans lequel sont mémorisées des valeurs spécifiques à l'instance pour les paramètres formels déclarés dans le FB.

Le bloc fonctionnel peut mémoriser ses données spécifiques aux instances dans un bloc de données d'instance dédié ou dans un bloc de données d'instance du bloc appelant.

4. Types d'accès

S7-1200 et S7-1500 offrent deux possibilités d'accès différentes pour les blocs de données d'instance rattachés à un bloc fonctionnel lors de l'appel :

- Des blocs de données avec un accès optimisé
Les blocs de données avec accès optimisé n'ont pas de structure de mémoire fermement définie. Dans la déclaration, les éléments de données ne reçoivent qu'un nom symbolique et pas d'adresse fixe dans le bloc.

- Blocs de données à accès standard (compatible avec S7-300/400)
Les blocs de données avec accès standard ont une structure de mémoire fixe. Dans la déclaration, les éléments de données reçoivent aussi bien un nom symbolique qu'une adresse fixe dans le bloc.

D. Blocs de données d'instance

1. Définition

L'appel d'un bloc fonctionnel est une instance. Les données avec lesquelles opère l'instance sont mémorisées dans un bloc de données d'instance.

La taille maximale des blocs de données d'instance varie selon la CPU. Les variables déclarées au bloc fonctionnel définissent la structure du bloc de données d'instance.

2. Types d'accès

S7-1200 et S7-1500 offrent deux possibilités d'accès différentes pour les blocs de données d'instance rattachés à un bloc fonctionnel lors de l'appel :

- Des blocs de données avec un accès optimisé
Les blocs de données avec un accès optimisé n'ont pas de structure fermement définie. Dans la déclaration, les éléments de données ne reçoivent qu'un nom symbolique et pas d'adresse fixe dans le bloc.

- Blocs de données à accès standard (compatible avec S7-300/400)
Les blocs de données avec accès standard ont une structure ferme. Dans la déclaration, les éléments de données reçoivent aussi bien un nom symbolique qu'une adresse fixe dans le bloc.

E. Blocs de données globaux

1. Définition

Les blocs de données servent à mémoriser les données de programme. Les blocs de données contiennent donc des données variables qui sont utilisées dans le programme utilisateur. Les blocs de données globaux enregistrent des données qui peuvent être utilisées par tous les autres blocs.

La taille maximale des blocs de données varie selon la CPU. Vous pouvez définir la

structure des blocs de données globaux à votre gré.

Vous avez également les possibilités d'utiliser des types de données API (UDT) comme modèle pour créer des blocs de données globaux.

2. Blocs de données globaux dans le programme utilisateur

Chaque bloc fonctionnel, chaque fonction et chaque bloc d'organisation peuvent lire les données d'un bloc de données global et écrire des données dans un bloc de données global. Ces données sont conservées dans le bloc de données, même à la fermeture du bloc de données. Un bloc de données global et un bloc de données d'instance peuvent être ouverts simultanément. (Figure 2.11)

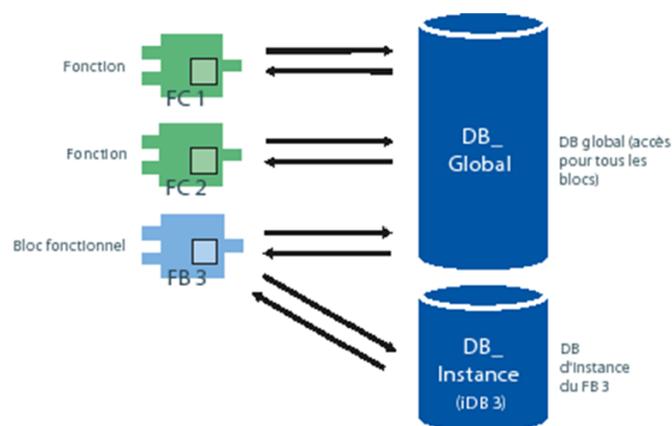


FIGURE 2.11 – Différents accès aux blocs de données

Ces différents types de blocs permettent une structuration efficace du programme utilisateur, ce qui facilite la compréhension et la maintenance du système d'automatisation.

2.2.6 Différentes étapes pour le développement de notre projet sur Tia portal

La figure suivante reprend l'architecture de notre projet comportant différents blocs nécessaires pour l'implémentation de notre modèle, sans oublier la régulation de vitesse du MAS (Figure 2.12).



FIGURE 2.12 – Représentation des différents blocs de notre projet

A. Blocs de données globaux (DB)

1. Introduction

Lorsque nous créons des variables pour notre projet, nous pouvons les stocker dans une base de données DB plutôt que dans une table de variables. Cela peut être plus simple et plus efficace car cela permet de stocker les données de manière structurée et facilement accessible. De plus, cela peut offrir plus de flexibilité et de sécurité pour gérer les variables de notre projet.

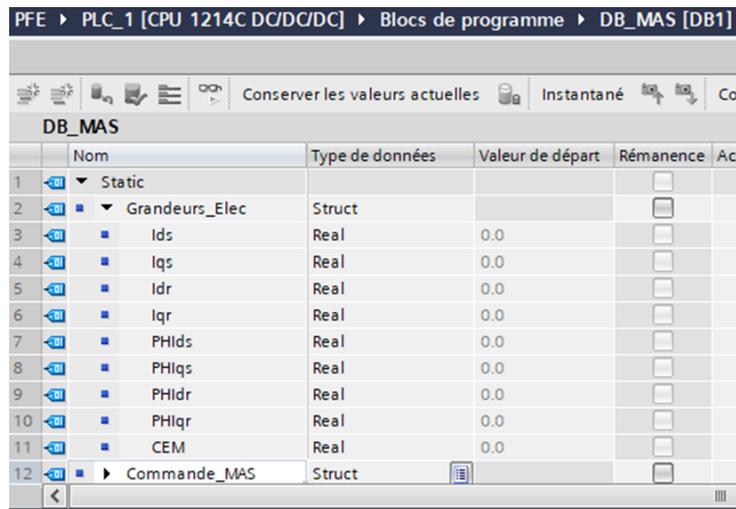
2. Création d'une base de données DB pour le modèle du MAS

Créer des Données globales représente les différentes grandeurs électriques, les commandes du moteur, la vitesse de rotation, la vitesse de consigne et les paramètres d'initialisation pour le modèle du moteur asynchrone. Les variables sont toutes de type réel (Real) ou type entier (Int).

Les variables sont stockées dans une structure (Struct) pour la commodité et l'organisation du code. Les variables sont également marquées comme étant statiques (Static), ce qui signifie qu'elles sont conservées entre les appels de fonction et sont accessibles dans l'ensemble du programme. Les variables I_{ds} , I_{qs} , I_{dr} , I_{qr} , Φ_{Hds} , Φ_{Hqs} , Φ_{Hdr} , Φ_{Hqr} et CEM représentent les grandeurs électriques telles que les courants statoriques, les flux statoriques et rotoriques et le couple électromagnétique.

La structure Commande MAS contient les variables v_d et v_q qui représentent les tensions statoriques qui sont des signaux de commande pour le moteur.

La structure Vitesse MAS contient la vitesse actuelle du moteur (ω), la vitesse de synchronisation (ω_s), et la vitesse de référence (ω) (Figure 2.13).



	Nom	Type de données	Valeur de départ	Rémanence	Ac
1	Static			<input type="checkbox"/>	
2	Grandeurs_Elec	Struct		<input type="checkbox"/>	
3	Ids	Real	0.0	<input type="checkbox"/>	
4	Iqs	Real	0.0	<input type="checkbox"/>	
5	Idr	Real	0.0	<input type="checkbox"/>	
6	Iqr	Real	0.0	<input type="checkbox"/>	
7	PHIds	Real	0.0	<input type="checkbox"/>	
8	PHIqs	Real	0.0	<input type="checkbox"/>	
9	PHIdr	Real	0.0	<input type="checkbox"/>	
10	PHIqr	Real	0.0	<input type="checkbox"/>	
11	CEM	Real	0.0	<input type="checkbox"/>	
12	Commande_MAS	Struct		<input type="checkbox"/>	

FIGURE 2.13 – Bloc DB du MAS

B. Fonctions (FC)

1. Créer une fonction FC pour le modèle du MAS

Pour écrire la fonction du moteur asynchrone dans le bloc FC, voici les étapes à suivre :

- Définir les variables nécessaires pour la fonction, comme les constantes et les variables d'état.
- Écrire la fonction en utilisant les équations mathématiques et les variables définies dans l'étape 1.
- Ajouter des commentaires pour expliquer chaque ligne de code et pour aider à la compréhension de la fonction.
- Tester la fonction en utilisant différentes valeurs d'entrée pour vérifier son bon fonctionnement (Figures 2.14 2.15)

Chapitre 2 : Modélisation et implémentation du MAS

```

IF... CASE... FOR... WHILE... (*...*) REGION
OF... TO DO... DO...

1 #Idso := #Ids;
2 #Iqso := #Iqs;
3 #Idro := #Idr;
4 #Iqro := #Iqr;
5 #W := #P * #Wm;
6
7 #PHIds := #L11 * #Idso + #L12 * #Iqso + #L13 * #Idro + #L14 * #Iqro;
8 #PHIqs := #L21 * #Idso + #L22 * #Iqso + #L23 * #Idro + #L24 * #Iqro;
9 #PHIdr := #L31 * #Idso + #L32 * #Iqso + #L33 * #Idro + #L34 * #Iqro;
10 #PHIqr := #L41 * #Idso + #L42 * #Iqso + #L43 * #Idro + #L44 * #Iqro;
11 #CEM := #P * (( #PHIds * #Iqso) - (#PHIqs * #Idso));
12
13
14 #Ids:=((((#A111+(#A211*(#Ws-#W))+(#A311*#W))*dt)+1)*#Idso+((((#A112+(#A212*(#Ws-#W))+(#A312*#W))*dt)+0)*#Iqso+((((#A113+(#A213
15 #Iqs:=((((#A121+(#A221*(#Ws-#W))+(#A321*#W))*dt)+0)*#Idso+((((#A122+(#A222*(#Ws-#W))+(#A322*#W))*dt)+1)*#Iqso+((((#A123+(#A223
16 #Idr:=((((#A131+(#A231*(#Ws-#W))+(#A331*#W))*dt)+0)*#Idso+((((#A132+(#A232*(#Ws-#W))+(#A332*#W))*dt)+0)*#Iqso+((((#A133+(#A233
17 #Iqr:=((((#A141+(#A241*(#Ws-#W))+(#A341*#W))*dt)+0)*#Idso+((((#A142+(#A242*(#Ws-#W))+(#A342*#W))*dt)+0)*#Iqso+((((#A143+(#A243
18
19 #Wm:=((#dt*(#CEM-#Cr-(#Fr*#Wm)))/#J)+#Wm;
20
21 IF (#Wm < 0) THEN
22     #Wm := 0;
23 END_IF;
24
25 #Wm_per:=REAL_TO_INT(#Wm*(27648/180));
26 #N := REAL_TO_INT(#Wm *30/#pi);

```

FIGURE 2.14 – Programme du Modèle MAS

	Nom	Type de données	Valeur par déf.	Commentaire
1	Input			
2	vq	Real		
3	vd	Real		
4	Ws	Real		
5	Output			
5	PHIds	Real		
7	PHIqs	Real		
8	PHIdr	Real		
9	PHIqr	Real		
10	CEM	Real		
11	W_prf	Int		
12	InOut			
13	Ids	Real		
14	Iqs	Real		
15	Idr	Real		
16	Iqr	Real		

FIGURE 2.15 – Variable du modèle MAS

Le modèle implique plusieurs équations (équations de la modélisation) pour calculer différents paramètres liés à la machine, notamment les flux statoriques et rotoriques, le couple électromagnétique, les courants statoriques et rotoriques, ainsi que la vitesse de la machine.

Les entrées du modèle sont les tensions statoriques et rotoriques (vd et vq) ainsi que la vitesse de la machine (Ws). Les sorties du modèle comprennent les flux statoriques et rotoriques (PHIds et PHIqs pour le stator, PHIdr et PHIqr pour le rotor), le couple électromagnétique (CEM) et la vitesse de la machine (W).

Le modèle inclut également plusieurs variables internes, notamment les courants statoriques et rotoriques (Ids, Iqs, Idr, Iqr), la vitesse de la machine (W) et certaines variables

temporaires (Idso, Iqso, Idro, Iqro). Le modèle comprend également plusieurs constantes, notamment diverses inductances (Ls, Lr), résistances (Rs, Rr) et autres paramètres de la machine (M, sig, Ts, Tr, etc. . .).

C. Blocs d'organisation (OB)

1. Création d'un Blocs d'organisation (OB) pour le modèle du MAS

On va lier la fonction FC du MAS Model à l'organisation block OB :

Cela implique de programmer l'appel de la fonction FC dans le bloc d'organisation OB, afin que celle-ci soit exécutée en réponse à certains événements ou à intervalles réguliers. Cette liaison entre la fonction et l'organisation block permet d'intégrer la fonction dans le programme de contrôle global, ce qui permet une automatisation efficace du système (Figure 2.16).

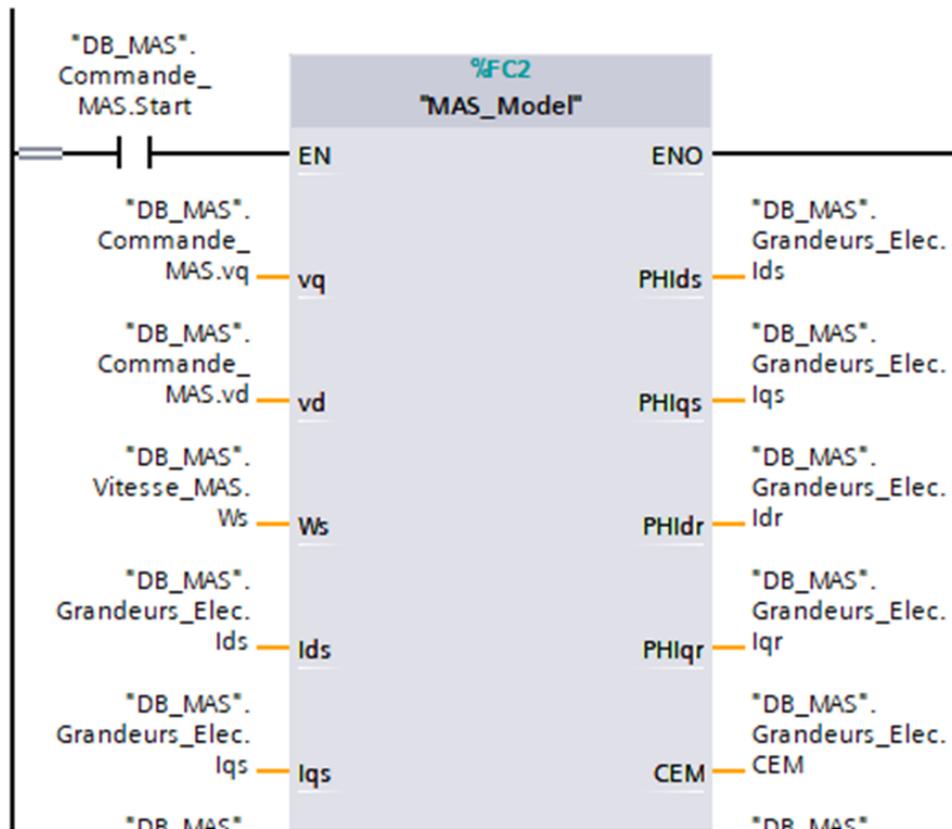


FIGURE 2.16 – Fonction du MAS au niveau du bloc OB

2.2.7 Résultat obtenu

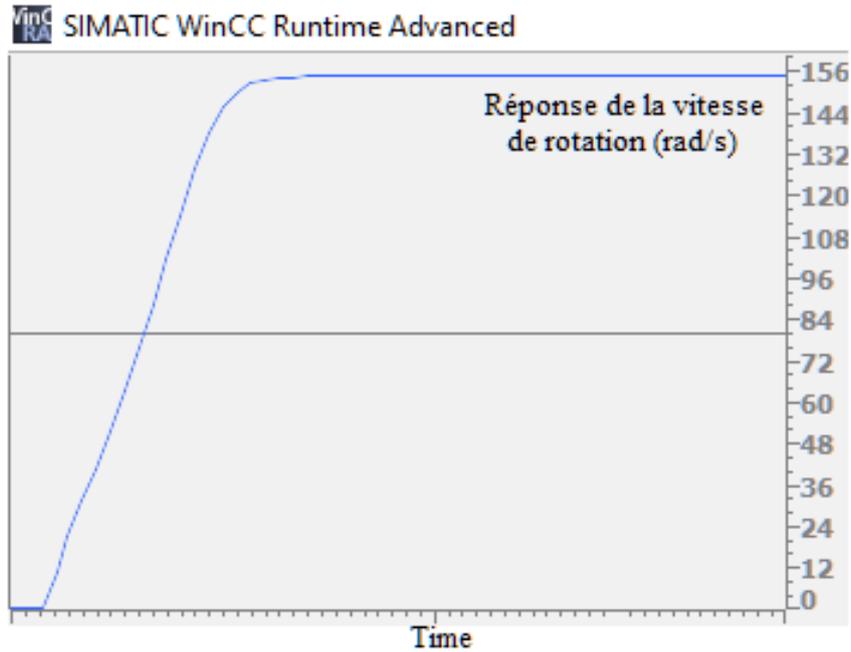


FIGURE 2.17 – Réponse sous TIA Portal de La vitesse de rotation (rad/s)

Nous constatons que les résultats obtenus dans TIA Portal sont identiques avec ceux de MATLAB.

2.3 Conclusion

Les résultats obtenus dans TIA Portal sont les mêmes que ceux obtenus dans MATLAB Simulink, cela confirme la validité et la précision des deux modèles d'un moteur asynchrone. Cela indique que les modèles sont fiables et peuvent être utilisés pour simuler le comportement du moteur dans différents scénarios de fonctionnement. Cependant, il est important de souligner que cette comparaison ne doit pas être considérée comme une étape unique et finale dans le processus de modélisation. D'autres tests et ajustements peuvent être nécessaires pour affiner la précision des modèles en fonction des objectifs de simulation spécifiques. En résumé, la comparaison des résultats entre les modèles de TIA Portal et MATLAB Simulink est une étape importante pour confirmer la validité des modèles et s'assurer qu'ils sont fiables pour simuler le comportement du moteur asynchrone.

Chapitre 3

Régulation industrielle et création d'un simulateur à l'aide de Win CC

3.1 Introduction

Le chapitre 3 est consacré au contrôle industriel et à la réalisation du simulateur du moteurs asynchrone sous Win CC. L'objet technologique PID-Compact sera utilisé comme régulateur PI de vitesse de rotation, notamment son principe de régulation et sa configuration. Nous créons également une Interface Homme Machine (IHM) pour faciliter la surveillance et le contrôle du MAS. Enfin, les résultats de notre simulateur du moteur asynchrone basé sur Win CC seront présentés et discutés. Nous allons donner plus de valeur à notre travail en utilisant le logiciel Factory I/O pour valider notre application en simulant un environnement réaliste d'usine. Ce chapitre permettra de mieux comprendre comment utiliser les outils industriels pour réguler et contrôler le comportement des moteurs à induction

3.2 Régulation industriel

3.2.1 Introduction

La réglementation industrielle est une discipline fondamentale dans l'industrie visant à surveiller et à stabiliser les processus de production. Il est conçu pour maintenir les paramètres de production à un niveau constant en régulant les variables de processus telles que la température, la pression, le débit, la vitesse, etc. Les contrôleurs électroniques sont des appareils qui mesurent les valeurs de ces variables et les comparent à des valeurs de référence (ou points de consigne), puis ajustent les paramètres de processus en conséquence. Les principaux objectifs du contrôle industriel sont d'optimiser les performances des processus de production, d'améliorer la qualité des produits et de minimiser les temps d'arrêt et les coûts de production. Les contrôleurs modernes comportent souvent des algorithmes de contrôle avancés tels que les contrôleurs PID (Proportionnel, Intégral, Dérivé) qui fournissent des performances de contrôle précises et stables. La réglementation industrielle s'applique à de nombreux secteurs industriels tels que la pétrochimie, l'alimentation, l'automobile, l'aviation et la production d'énergie. Il s'agit d'un élément essentiel de l'industrie 4.0 et est souvent associé à la maintenance prédictive et à l'analyse des données pour une optimisation continue des processus de production[60]

3.2.2 Objet technologique PID-Compact de l'API S7-1200

1. Introduction

Le régulateur PID-Compact est un objet technologique introduit dans les APIs Siemens pour contrôler ou réguler divers grandeurs physiques comme la température, la hauteur et la vitesse de rotation de différents équipements industriels tels que des moteurs, des ventilateurs ou des pompes, etc... Ce régulateur utilise un algorithme de contrôle de type PID (proportionnel Intégral Dérivé) qui permet de réguler la grandeur désirée en fonction de la différence entre la consigne souhaitée et la mesure. Le PID-Compact est configurable en fonction de la grandeur physique à réguler, du sens de régulation, du comportement au démarrage après un reset, ainsi que de la source de consigne, de mesure et de valeur de réglage. Les paramètres P, I et D du régulateur peuvent également être ajustés pour optimiser son comportement. Le PID-Compact peut fonctionner en mode manuel ou automatique en fonction des besoins de l'application.[61]

2. Principe de la régulation par le biais du PID-Compact

D'abord, il est important de comprendre le principe de régulation du PID-Compact. Il faut savoir aussi que ce régulateur sera utilisé dans notre travail à réguler la vitesse de rotation utilisant uniquement le régulateur PI de l'élément technologique PI-Compact.. Ce régulateur fonctionne selon un mode de contrôle proportionnel et intégral, également connu sous le nom de régulation PI. Le régulateur compare la mesure de la grandeur physique (telle que la vitesse) avec la consigne de référence et calcule la valeur de réglage nécessaire pour atteindre cette consigne. Pour ce faire, le régulateur utilise deux termes : le terme proportionnel (P) et le terme intégral (I). Le terme proportionnel détermine la réponse du régulateur en fonction de l'écart entre la mesure et la consigne. Plus l'écart est grand, plus la réponse sera forte. Le terme intégral agit sur la réponse du régulateur en fonction de la durée pendant laquelle l'écart entre la mesure et la consigne persiste. Ainsi, plus la durée de l'écart est longue, plus la réponse sera forte. Il est important de régler correctement les paramètres P et I pour obtenir une régulation efficace. Le paramètre P doit être réglé de manière à obtenir une réponse rapide et précise aux changements de mesure, tandis que le paramètre I doit être réglé de manière à éliminer les erreurs de régulation persistantes. En résumé, le régulateur PI utilise un mode de régulation proportionnel et intégral pour comparer la mesure de la grandeur physique avec la consigne de référence et calculer la valeur de réglage nécessaire pour atteindre cette consigne. Les paramètres P et I doivent être réglés correctement pour obtenir une régulation efficace.[62]

A. Action proportionnelle (P)

L'action proportionnelle (P) d'un régulateur représente la réponse instantanée du système en fonction de la différence entre le point de consigne et la valeur mesurée. Elle est directement proportionnelle à la différence entre la consigne et la valeur mesurée à un instant donné.

Plus précisément, l'action proportionnelle consiste à multiplier la différence entre la consigne et la valeur mesurée par le gain proportionnel K_p , qui est un paramètre réglable du régulateur. Plus la différence est grande, plus la correction appliquée est importante. Par conséquent, l'action proportionnelle peut permettre au système d'obtenir une réponse rapide, mais si le gain K_p n'est pas ajusté correctement, cela provoquera également une oscillation ou une instabilité.[61]

B. Action intégrale (I)

Dans le cas d'un régulateur PI compact, l'action intégrale (I) est utilisée pour réduire l'écart constant de régulation qui peut persister en présence d'une action proportionnelle seule. Elle ajoute une correction proportionnelle à l'intégrale de l'écart entre la consigne et la mesure, ce qui permet d'éliminer les erreurs de régulation persistantes. Contrairement à l'action proportionnelle qui ne prend en compte que l'erreur instantanée, l'action intégrale se base sur l'erreur cumulée au fil du temps. Pour réagir aux variations de l'écart de régulation, l'action intégrale dépend d'un paramètre appelé temps d'intégration. Si ce temps est trop court, l'action intégrale sera trop sensible aux variations rapides de l'écart, ce qui peut entraîner une oscillation de la sortie du régulateur. En revanche, si le temps d'intégration est trop long, l'action intégrale sera trop lente pour corriger l'écart, ce qui peut entraîner un temps de réponse lent et une instabilité du système (Figure 3.1).[11]

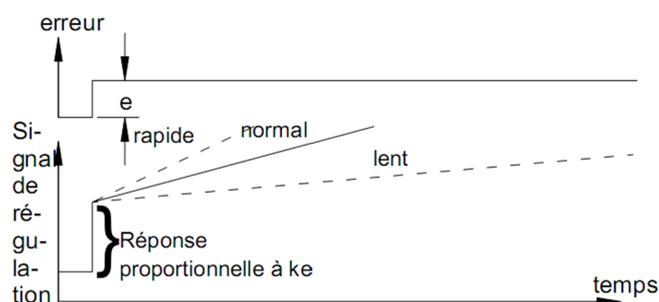


FIGURE 3.1 – Schéma de principe utilisant une régulation PI[11]

La formule mathématique associée à l'action intégrale d'un régulateur est la suivante : la sortie régulée est affectée par une correction proportionnelle à l'intégrale de l'écart de régulation au fil du temps :

$$Y = K_i \int e dt \text{ Avec } K_i = \frac{1}{T_n}$$

K_i : Le gain intégrateur

T_n : La constante d'intégration de système

y : La valeur de sortie

La croissance ou la diminution de la valeur réglée est fonction de l'écart de régulation et de la durée d'intégration.

Le régulateur PI est une combinaison des fonctions proportionnelle et intégrale. L'action proportionnelle permet de corriger rapidement les écarts de régulation, tandis que l'action intégrale assure la stabilité à long terme en éliminant l'écart de régulation constant. L'utilisation conjointe de ces deux fonctions peut réduire de manière significative les temps de montée et de régulation tout en améliorant la précision du contrôle. En raison de ces avantages, le régulateur PI est largement utilisé dans les applications industrielles qui nécessitent une régulation précise en temps réel (Figure 3.2).[62]

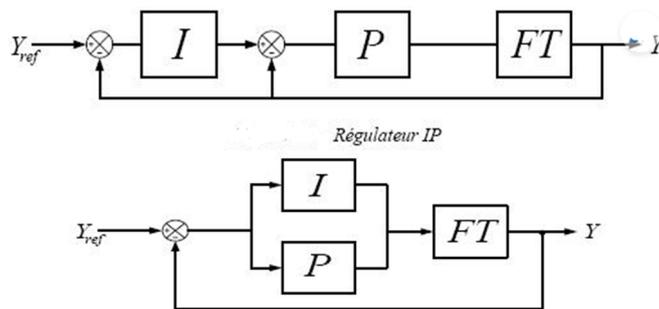


FIGURE 3.2 – Schéma-bloc du régulateur PI [11]

3.2.3 Configuration du régulateur PID-Compact

La configuration d'un régulateur PID-Compact pour la régulation de la vitesse se fait en plusieurs étapes. Tout d'abord, il est nécessaire de définir la grandeur physique à réguler et le sens de régulation. Ensuite, il faut configurer la consigne, la mesure et la valeur de réglage. Pour cela, il est possible de les configurer soit à partir du DB d'instance, soit via une instruction interconnectée.

Il est également important de configurer les paramètres P, I du régulateur pour optimiser son comportement. Pour cela, une optimisation préalable peut être réalisée pour calculer ces paramètres de manière autonome. Une optimisation fine peut ensuite être réalisée pour ajuster ces paramètres en fonction des caractéristiques de l'application.

Enfin, il est possible de configurer le régulateur pour fonctionner en mode manuel ou automatique. En mode manuel, la valeur de réglage est définie par l'utilisateur, tandis

Chapitre 3 : Régulation industrielle et création d'un simulateur à l'aide de Win CC

qu'en mode automatique, elle est calculée par le régulateur en fonction de la mesure et de la consigne (Figure 3.3).

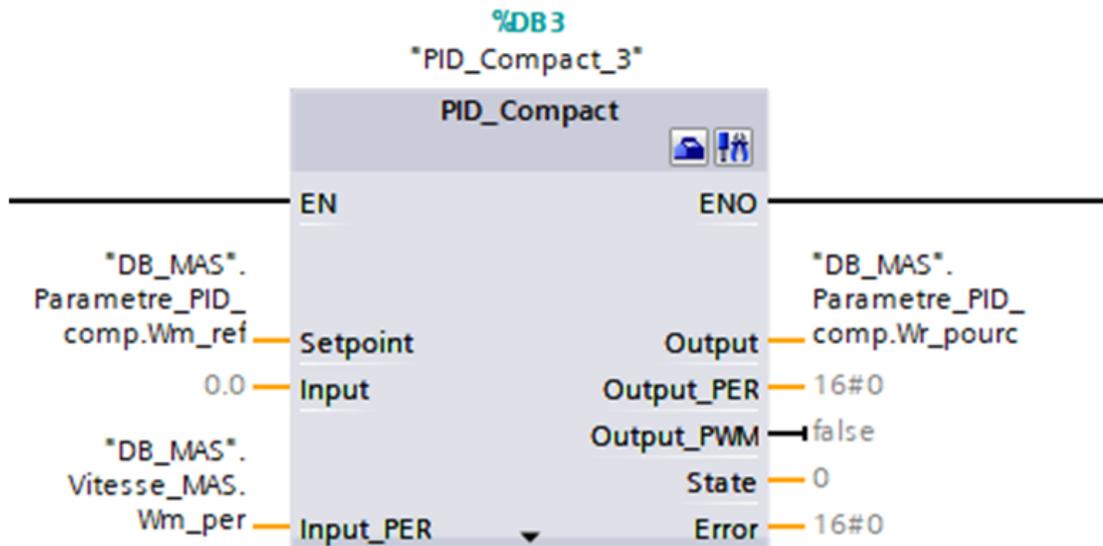


FIGURE 3.3 – Régulation de vitesse par le biais du PID-Compact du S7-1200

Et après cette étape on a ajouté deux réseaux :

Réseaux numéro 1 :

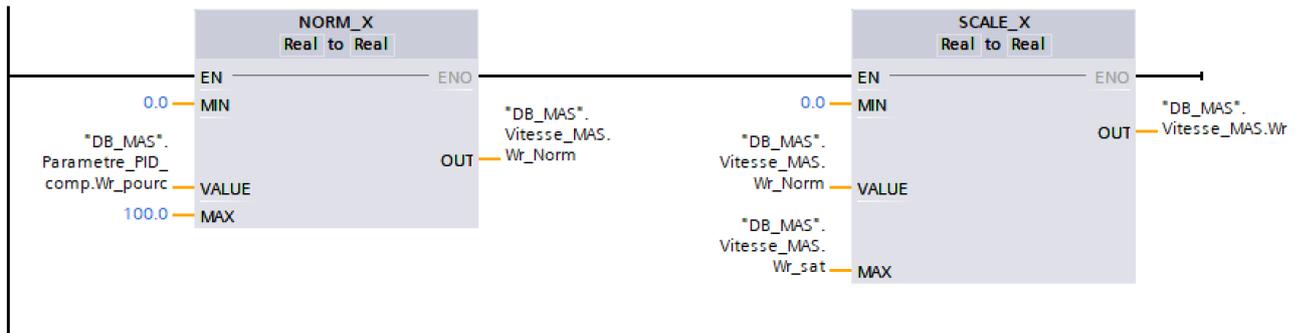


FIGURE 3.4 – Mise à l'échelle de la vitesse angulaire relative

Réseaux numéro 2 :

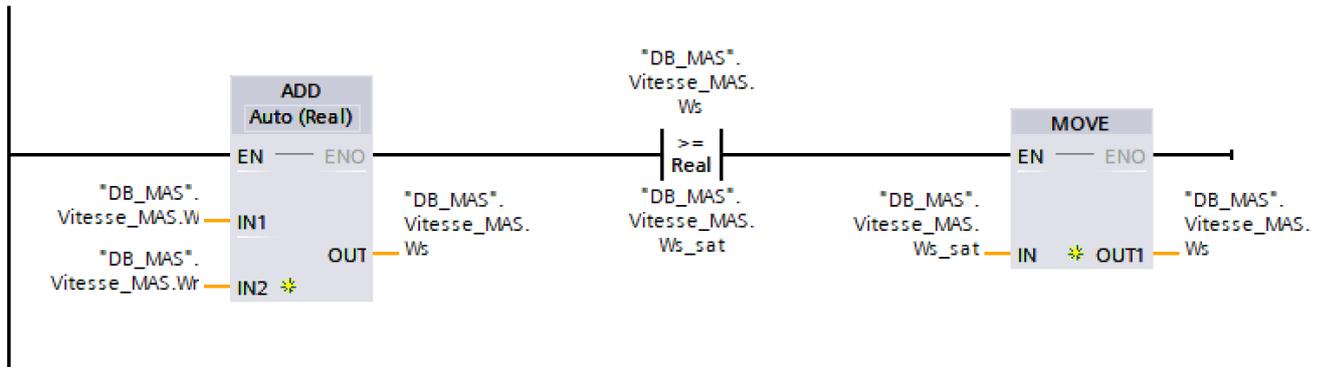


FIGURE 3.5 – Calcul de la vitesse angulaire de synchronisme (pulsation)

Pour déterminer la vitesse angulaire de synchronisme (pulsation), on na utiliser la relation naturelle d'auto pilotage. Ainsi, un réseau peut être conçu pour calculer cette pulsation en utilisant les valeurs de ω (Vitesse angulaire électrique) et de ωr (Vitesse angulaire relative)

3.2.4 Validation du PID-Compact

Pour la validation du régulateur PID, nous réaliserons des tests en utilisant le régulateur pour contrôler la vitesse du moteur dans différentes conditions tout en vérifiant la sortie de régulateur.

Il est essentiel de mener ces tests de validation afin de garantir que le PI répond aux exigences de performance et de contrôle attendues pour la vitesse du moteur.

3.3 Implémentation d'une interface homme-machine (IHM)

3.3.1 Introduction

Il est important d'avoir une interface IHM pour surveiller et contrôler notre PID-Compact. Cette interface peut afficher des informations clés telles que la température du système, la vitesse, le courant et la tension. En utilisant des éléments interactifs tels que des boutons et des widgets, l'utilisateur peuvent facilement contrôler le système en démarrant, en arrêtant ou en modifiant la vitesse des moteurs. En utilisant une IHM, il est possible de faciliter l'interaction entre l'utilisateur et le système, de réduire les erreurs de manipulation et d'améliorer l'efficacité du contrôle, ce qui est essentiel pour une gestion optimale du PID-Compact (Figure 3.6).

Chapitre 3 : Régulation industrielle et création d'un simulateur à l'aide de Win CC

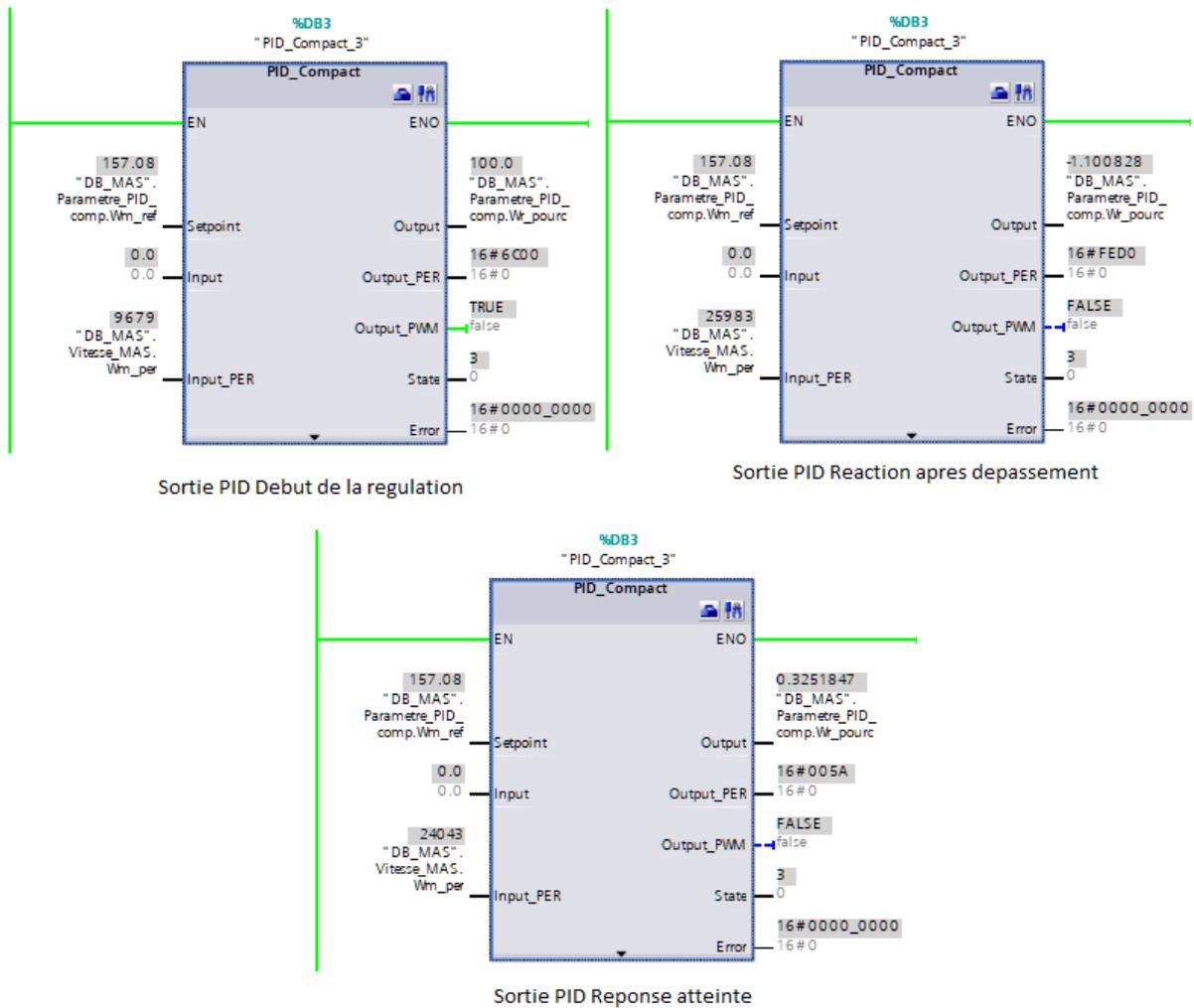


FIGURE 3.6 – Réponses de sortie du PID-Compact lors de la régulation dans différentes conditions

3.3.2 Liaison entre l'IHM et le PLC

Le lien entre IHM (Interface Homme Machine) et PLC (Automate Programmable) est essentiel au bon fonctionnement d'un système d'automatisation. En effet, l'IHM permet de visualiser et de contrôler les variables du système via une interface utilisateur graphique, tandis que l'automate gère les entrées et les sorties du système et exécute le programme de contrôle. Le lien entre IHM et automate peut être établi via une communication de type PROFINET TCP/IP ou encore PROFIBUS, selon les besoins et les caractéristiques du système. Le choix du mode de communication dépend entre autres de la distance entre l'IHM et l'automate, de la vitesse de transmission des données et de la fiabilité requise (Figure 3.7). [10]

Il est également important de définir une table de liaison des variables entre l'IHM et

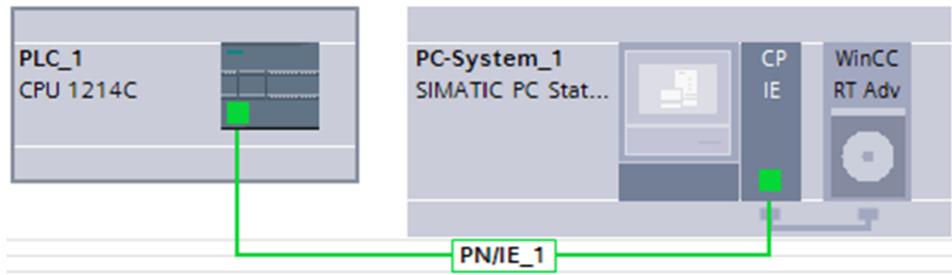


FIGURE 3.7 – Liaison entre l'IHM et le PLC

l'automate afin de pouvoir définir les adresses mémoire des variables système accessibles depuis l'IHM

3.3.3 Tableau de liaison des variables entre l'IHM et le PLC

Un document de référence appelé "tableau de correspondance des variables" permet d'associer les variables utilisées dans le programme de l'IHM aux variables correspondantes dans le programme du PLC. Cette table permet de garantir que les informations échangées entre l'IHM et le PLC sont correctement identifiées et interprétées par chacun des systèmes. Le tableau peut contenir des informations telles que le nom de la variable, son type de données, son emplacement dans le programme et son accès symbolique. C'est un élément crucial pour garantir une communication fiable et efficace entre les deux systèmes dans les systèmes de contrôle industriels (Figure 3.8).^[10]

Variables IHM						
	Nom ▲	Table des variab..	Type de données	Connexion	Nom API	Variable API
[-]	H_Conv	Table_IHM_Mas	Bool	HMI_Liaison_1	PLC_1	DB_MAS.Commande_M...
[-]	H_Graph_speed	Table_IHM_Mas	Real	HMI_Liaison_1	PLC_1	DB_MAS.Vitesse_MAS.Wm
[-]	H_initialisation	Table_IHM_Mas	Bool	HMI_Liaison_1	PLC_1	DB_MAS.Commande_M...
[-]	H_Mas	Table_IHM_Mas	Bool	HMI_Liaison_1	PLC_1	DB_MAS.Commande_M...
[-]	H_PID_Compact_Mode_...	Table_IHM_Mas	Int	HMI_Liaison_1	PLC_1	PID_Compact_3.sRet.i_M...
[-]	H_Speed	Table_IHM_...	Real	HMI_Liaison...	PLC_1	DB_MAS.Vitesse_MAS...
[-]	H_Speed_Des	Table_IHM_Mas	Real	HMI_Liaison_1	PLC_1	DB_MAS.Parametre_PID_...
[-]	H_Start	Table_IHM_Mas	Bool	HMI_Liaison_1	PLC_1	DB_MAS.Commande_M...
[-]	H_Stop	Table_IHM_Mas	Bool	HMI_Liaison_1	PLC_1	DB_MAS.Commande_M...

FIGURE 3.8 – Table IHM du MAS

3.3.4 Comment créer une variable IHM : (on prend comme un exemple bouton de démarrage "Start")

Dans ce contexte, nous allons décrire les étapes à suivre pour créer une variable IHM "H Start" qui sera liée au bouton de démarrage (Start) de la machine. Cette variable permettra de contrôler le démarrage de la machine depuis l'IHM. Les étapes décrites ci-dessous sont générales et peuvent varier en fonction du logiciel de programmation de l'IHM que vous utilisez.

Pour créer la variable IHM "H Start" pour le bouton Start, vous pouvez suivre les étapes suivantes :

1. Ouvrez le logiciel de programmation de l'IHM.
2. Créez un nouveau projet.
3. Créez une nouvelle vue où vous souhaitez ajouter le bouton Start.
4. Ajoutez un bouton à la vue en sélectionnant l'outil de bouton dans la barre d'outils et en cliquant sur la vue où vous voulez que le bouton apparaisse.
5. Dans la fenêtre de propriétés du bouton, recherchez les propriétés pour lier le bouton au programme PLC. Les propriétés peuvent varier selon le logiciel de programmation de l'IHM que vous utilisez.
6. Dans la propriété "Variable", entrez le nom de la variable que vous voulez lier au bouton. Dans ce cas, entrez "H Start".
7. 7. Dans la propriété "Type de variable", sélectionnez "Bool" car il s'agit d'une variable booléenne.
8. Dans la propriété "Liaison de données", sélectionnez la table de liaison des variables IHM/PLC.
9. Dans la propriété "Liaison de programme IHM", sélectionnez la liaison IHM que vous utilisez.
10. Dans la propriété "Liaison de programme PLC", sélectionnez le programme PLC que vous utilisez.
11. Dans la propriété "Adresse de variable PLC", entrez l'adresse de la variable dans le programme PLC. Dans ce cas, entrez "DB MAS.Commande MAS.BP Start".
12. Dans la propriété "Accès symbolique", entrez un texte symbolique qui permettra de se référer à la variable.
13. Dans la propriété "Intervalle de rafraîchissement", définissez la fréquence à laquelle la variable doit être mise à jour. Dans ce cas, entrez "100 ms".

14. Dans la propriété "Initialisation", définissez l'état initial de la variable. Dans ce cas, sélectionnez "False" pour que le bouton ne soit pas enfoncé au démarrage de l'IHM.
15. Enregistrez les modifications et téléchargez le programme sur l'IHM.

Maintenant, le bouton Start est lié à la variable H Start dans le programme PLC, ce qui permet de contrôler le démarrage de la machine à partir de l'IHM.

Dans notre cas, la variable H-Start est lié au contact "DB-MAS".Commande-MAS.BP-Start se trouvant au niveau du réseau de pilotage Start/Stop du bloc OB (Figure 3.9)

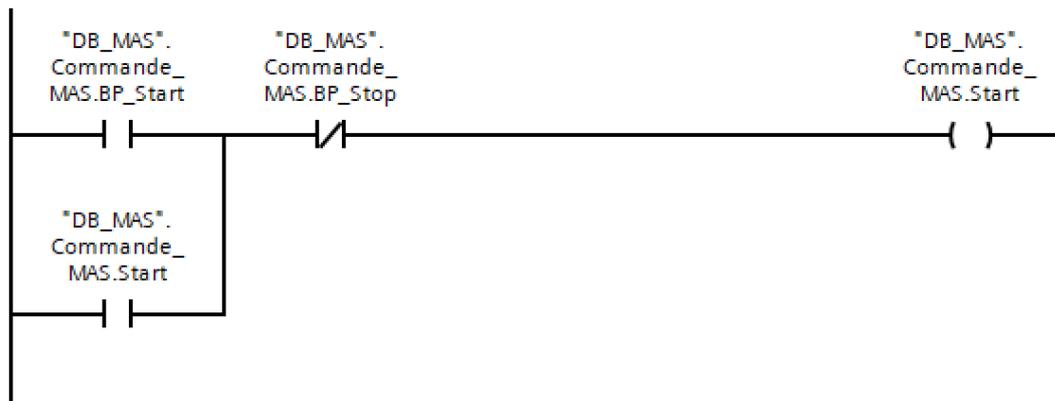


FIGURE 3.9 – Réseaux du pilotage Start/Stop

Ce réseau ladder "Start/Stop" est un réseau de commande utilisé pour démarrer et arrêter le moteur. Il est basé sur deux boutons-poussoirs, l'un pour démarrer le moteur (bouton "Start") et l'autre pour l'arrêter (bouton "Stop"). Le réseau se compose généralement d'une série de contacts normalement ouverts reliés au bouton "Start" et d'une série de contacts normalement fermés reliés au bouton "Stop". Ces contacts sont ensuite reliés à une bobine de commande qui active le contacteur qui alimente le moteur. Lorsque le bouton "Start" est enfoncé, les contacts normalement ouverts se ferment, permettant ainsi à la bobine de commande d'être alimentée et d'activer le contacteur pour démarrer le moteur. Si le bouton "Stop" est enfoncé à tout moment, les contacts normalement fermés s'ouvrent, interrompant ainsi l'alimentation de la bobine de commande et arrêtant le moteur .

3.3.5 Validation du bouton "Start" dans notre IHM

La validation du bouton "Start" inclut la validation de son bon fonctionnement dans le système. Pour cela, nous avons réalisé des tests expérimentaux pour nous assurer que le

Chapitre 3 : Régulation industrielle et création d'un simulateur à l'aide de Win CC

bouton "Start" déclenchant bien le démarrage du moteur. Observez que le moteur répond de manière appropriée en atteignant la vitesse souhaitée et en maintenant un fonctionnement stable. Cette vérification est essentielle pour s'assurer que le bouton "Start" remplit son rôle d'activation du moteur de manière fiable et sûre (Figure 3.10).

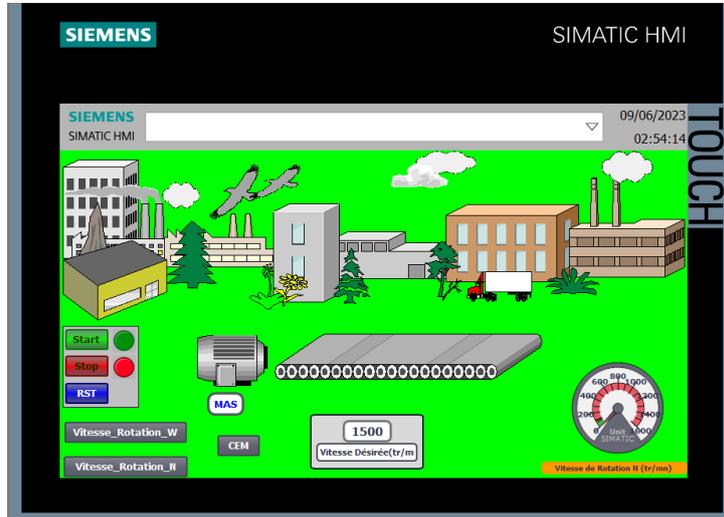


FIGURE 3.10 – "Vue de notre IHM avant appui sur bouton Star (lors de l'arrêt du MAS)

"

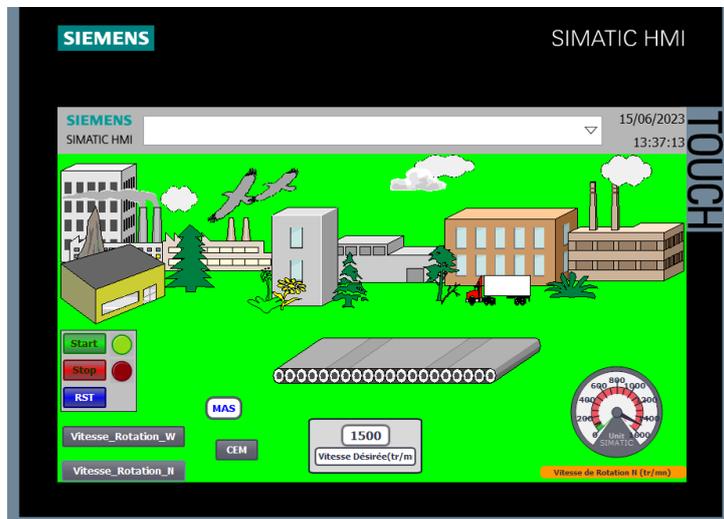


FIGURE 3.11 – " Appui sur bouton Star (lors que le moteur est en marche) "

Après avoir fini le développement et la vérification de notre IHM sur notre PC-system, nous allons passer à la description de notre prototype réel où on va installer notre IHM en utilisant un HMI de marque Siemens(Figure 3.11).

3.4 Présentation de notre prototype réel

Dans la réalisation de notre prototype, nous allons utiliser différents composants et éléments pour assurer son bon fonctionnement. 3.12

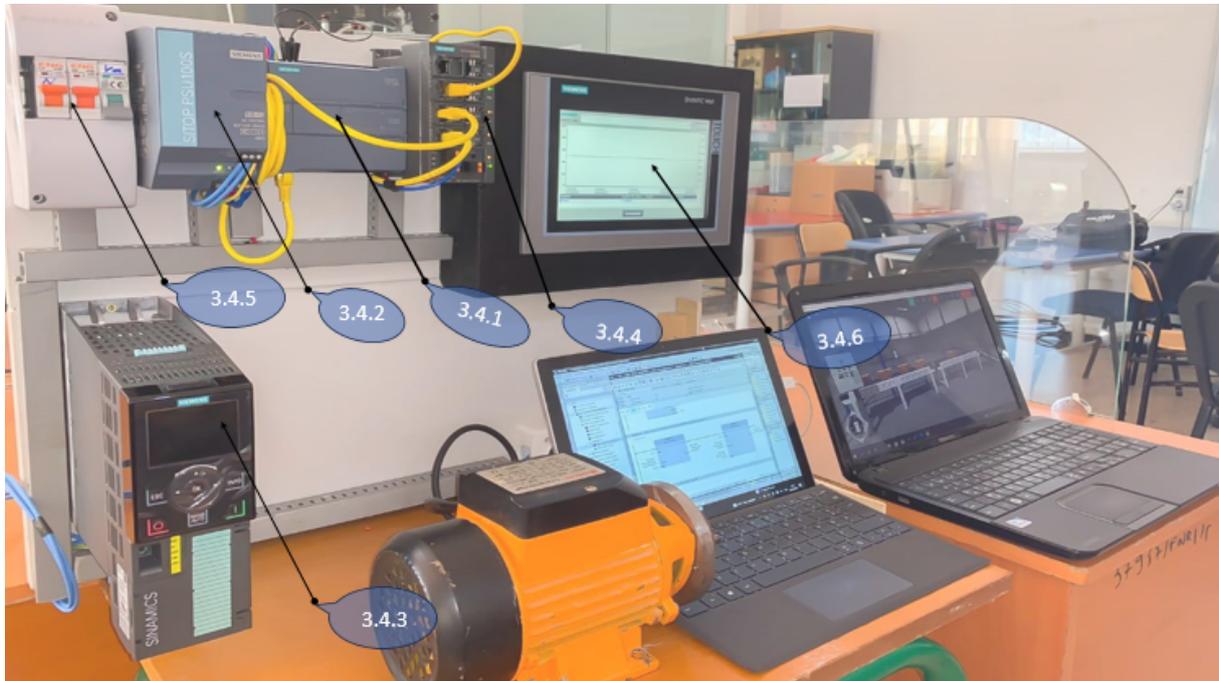


FIGURE 3.12 – simulateur de moteur asynchrone à l'aide de l'API S7-1200

Voici quelques-uns des éléments que nous allons utiliser :

3.4.1 Module CPU 1214C d'API S7-1200 de Siemens



FIGURE 3.13 – CPU 1214C de Siemens

Le module CPU 1214C de l'automate programmable S7-1200 de Siemens est une unité centrale de traitement puissante et polyvalente nous utilisée dans notre systèmes automa-

tisés pour contrôler et exécuter le programme d'automatisation. Il constitue un élément clé dans la mise en œuvre de solutions de contrôle industriel efficaces et fiables (Figure 3.13).

3.4.2 Alimentation électrique Le Sitop PSU100S DC 24 V/10 A



FIGURE 3.14 – Sitop PSU100S DC 24 V/10 A de Siemens

Le Sitop PSU100S DC 24V/10A de Siemens est une alimentation fiable et robuste conçue pour fournir une tension continue stable de 24 Volts et un courant maximum de 10 Am-pères. Il offre des fonctionnalités de protection, une large plage de tension d'entrée et une robustesse pour notre système (Figure 3.14).

3.4.3 Variateur de fréquence



FIGURE 3.15 – Variateur de fréquence 6SL3220-3YE24-0AF0 de Siemens

Le variateur de fréquence 6SL3220-3YE24-0AF0 de Siemens offre une solution efficace pour la commande de moteurs. Il permet un contrôle précis de la vitesse et du couple, ainsi

jouent un rôle crucial dans la sécurité électrique en interrompant automatiquement le courant en cas de conditions anormales, préservant ainsi l'intégrité du système et réduisant les risques d'accidents ou de dommages (Figure 3.17).

3.4.6 SIMATIC HMI

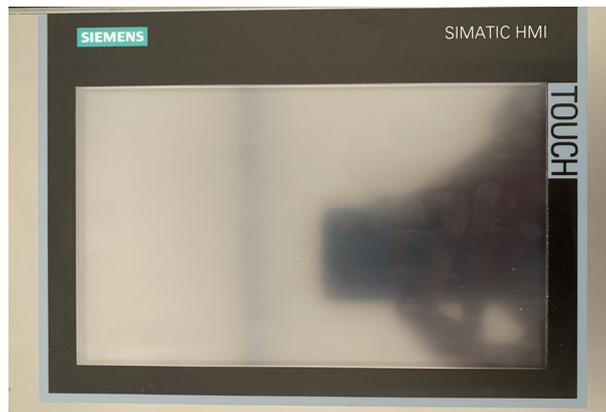


FIGURE 3.18 – SIMATIC HMI TP900 Comfort

Le SIMATIC HMI TP900 Comfort offre aux opérateurs une interface intuitive et conviviale pour interagir avec notre système automatisé, surveiller les processus en temps réel, effectuer des réglages et des contrôles, et visualiser les données essentielles pour le moteur(Figure 3.18).

3.5 Résultats et validation

Afin de vérifier les résultats obtenus lors du réglage, nous avons comparé les données obtenues à partir de notre IHM, notamment à l'aide de l'objet "Graphics" de WinCC, avec les résultats obtenus à partir de MATLAB Simulink.

Résultats à partir de Matlab Simulink :

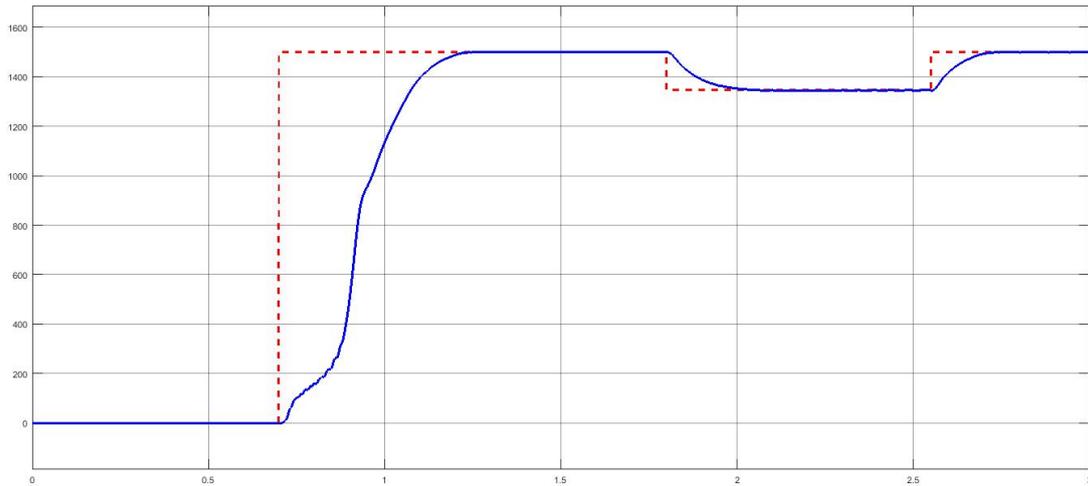


FIGURE 3.19 – Réponse de la vitesse de rotation du MAS (tr/min)

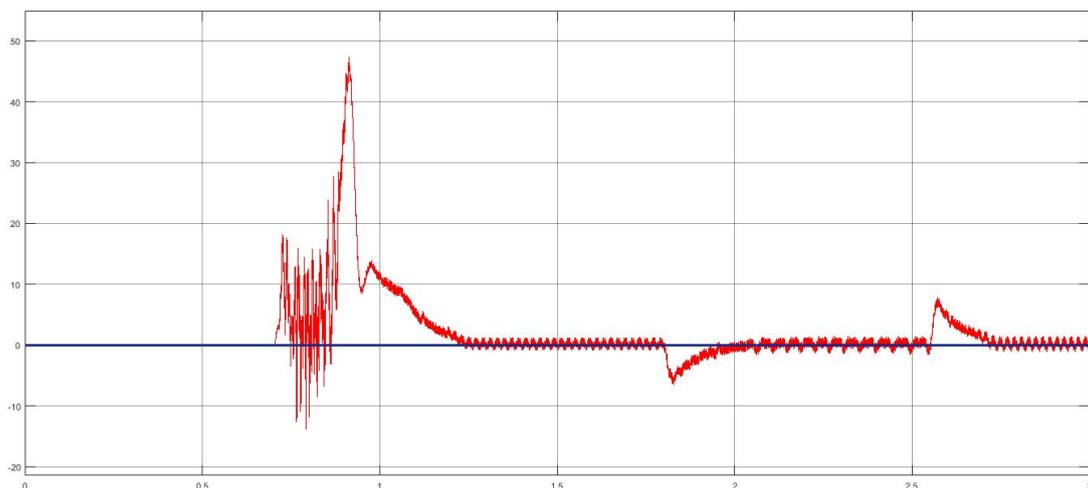


FIGURE 3.20 – Réponse du couple électromagnétique (CEM) du MAS

D'après les résultats obtenus, nous pouvons tirer les interprétations suivantes :

1. Variation de la vitesse : Lorsque nous avons appliqué une consigne de vitesse de 1500 tr/min à l'instant $t=0.7$ minute, la vitesse de rotation a augmenté progressivement jusqu'à atteindre cette valeur à 1.25 minute. Cette augmentation de la vitesse s'est faite de manière régulière et sans dépassement. De même, lorsque nous avons modifié la consigne à 1300 tr/min à l'instant $t=1.75$ minute, la vitesse a diminué progressivement jusqu'à atteindre cette nouvelle valeur à 2 minute. Enfin, lorsque nous avons réappliqué une consigne de 1500 tr/min à l'instant $t=2.6$ minute, la vitesse a augmenté progressivement jusqu'à atteindre cette valeur à 3.8 minute.
2. Comportement du couple électromagnétique : À l'instant $t=0.7$ minute, le couple

Chapitre 3 : Régulation industrielle et création d'un simulateur à l'aide de Win CC

électromagnétique peut présenter des oscillations en raison de la variation de la vitesse. Cependant, ces oscillations se stabilisent rapidement. Lorsque la consigne de vitesse est passée à 1300 tr/min à l'instant $t=1.75$ minute, le couple a diminué progressivement jusqu'à atteindre une valeur de -4 Nm. Enfin, lorsque la consigne est de nouveau passée à 1500 tr/min à l'instant $t=2.6$ minute, le couple a augmenté progressivement jusqu'à atteindre une valeur de un pic de 8Nm. Ces variations du couple électromagnétique sont en corrélation avec les changements de vitesse du moteur, démontrant ainsi la réaction du système dans le contrôle du couple en fonction de la consigne de vitesse.

Les résultats de la simulation montrent que la vitesse de rotation peut suivre la consigne donnée. Le couple électromagnétique peut présenter des oscillations avant de se stabiliser en réponse aux changements de la consigne de vitesse. Ces observations soulignent l'importance d'une conception et d'un réglage appropriés du régulateur pour obtenir des performances de vitesse et de couple optimales dans des conditions de fonctionnement variables.

Résultats tracés à l'aide de l'objet "Graphics" de WinCC

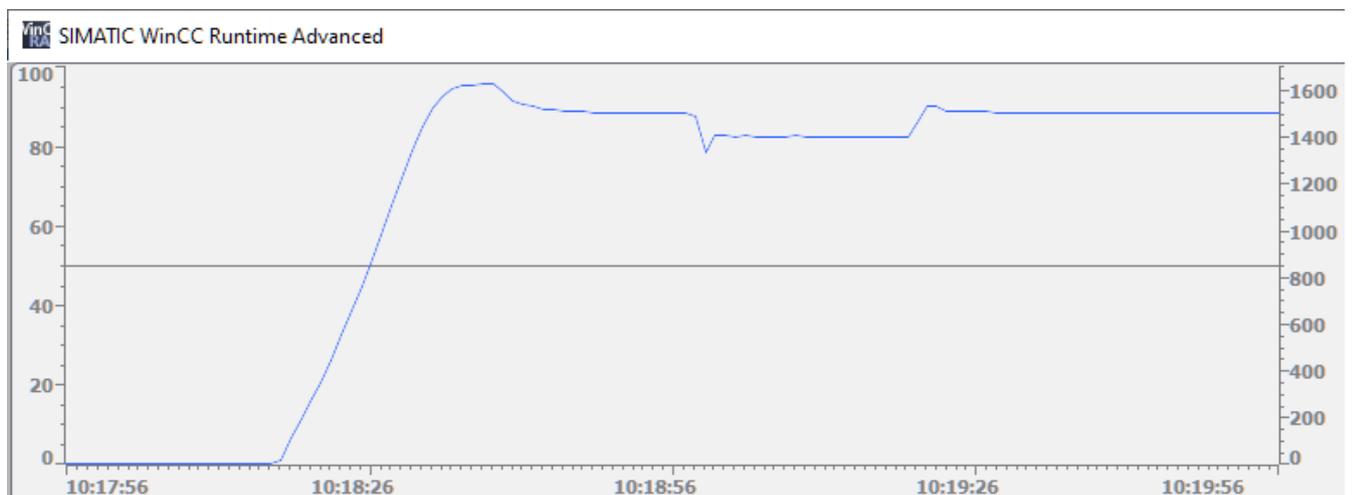


FIGURE 3.21 – Réponse de la vitesse de rotation avec la régulation PI



FIGURE 3.22 – Réponse du CEM avec la régulation PI

Les résultats obtenus lors des tests expérimentaux confirment la qualité et le bon fonctionnement de notre régulateur. Les performances du prototype sont satisfaisantes et prometteuses, montrant que le régulateur PI de l'objet technologique PID-Compact peut contrôler efficacement la vitesse du moteur dans différentes conditions. La comparaison des résultats obtenus avec le régulateur PID-compact à ceux simulés avec MATLAB/Simulink permet de vérifier la cohérence et la précision du régulateur. Les réponses en vitesse et en couple électromagnétique étaient conformes aux attentes et spécifiées pour l'application (Figure 3.21 3.22).

3.6 Exemple de l'application industrielle à l'aide de factory I/O

3.6.1 Introduction

En conclusion, notre régulation du moteur peut être appliquée dans de nombreuses applications industrielles et l'utilisation pratique de ce moteur peut être réalisée en utilisant Factory I/O, un logiciel de simulation d'usine. Par exemple, considérez une application où notre contrôleur PID-compact est utilisé pour contrôler la vitesse d'un moteur électrique dans une ligne de production automatisée. En utilisant Factory I/O, nous pouvons créer des simulations réalistes d'environnements de production, y compris des capteurs de vitesse, des actionneurs et d'autres dispositifs (Figure 3.23).



FIGURE 3.23 – Factory I/O

Factory I/O nous permettra de visualiser les performances du régulateur en temps réel en affichant la valeur de consigne et la vitesse réelle du moteur. Nous pouvons également observer d'autres paramètres tels que le couple électromagnétique et le courant du stator pour évaluer l'efficacité du régulateur à maintenir la vitesse souhaitée. En utilisant Factory I/O, nous pouvons effectuer des tests et des simulations pour évaluer différentes configurations de contrôle, régler les paramètres du contrôleur et optimiser les performances du système. Cela permet d'assurer un contrôle précis et stable de la vitesse du moteur dans des conditions industrielles simulées. En résumé, l'application pratique de la régulation de notre moteur dans des environnements industriels peut être réalisée en créant des simulations d'usine réalistes à l'aide de factoryI/O. Cela nous permet d'évaluer et d'optimiser les performances du contrôleur avant sa mise en œuvre dans des applications réelles, garantissant que le moteur fonctionne de manière fiable et efficace dans les environnements de production industrielle.

3.6.2 Interface entre factory I/O, l'HMI KTP900 et l'API S7-1200



FIGURE 3.24 – Factory I/O et L'API S7-1200

Pour interfacer Factory I/O avec l'HMI KTP900 et le PLC S7-1200 :

1. Pour établir la communication entre Factory I/O et le PLC S7-1200, nous devons utiliser un pilote approprié. Dans Factory I/O, cliquez sur "File > Driver Configuration" pour ouvrir la fenêtre de configuration du pilote.
2. Sélectionnez le pilote "Siemens S7-1200/1500" dans la liste déroulante du pilote.
3. Dans la fenêtre de configuration du pilote, spécifiez l'adresse IP du PLC S7-1200 et d'autres paramètres de connexion nécessaires. Assurez-vous que le modèle "S7-1200" est sélectionné.
4. Une fois la configuration terminée, cliquez sur "Connect" pour établir la connexion entre Factory I/O et le PLC S7-1200. Si la connexion est réussie, une icône verte s'affichera à côté du pilote dans la fenêtre de configuration du pilote.
5. Maintenant, nous pouvons créer des scènes dans Factory I/O et associer les entrées/sorties de l'HMI KTP900 avec les entrées/sorties du PLC S7-1200. Utilisez les fonctionnalités de Factory I/O pour configurer les actions et les réactions en fonction des interactions avec l'HMI.
6. Testez votre configuration en exécutant Factory I/O et en interagissant avec l'HMI KTP900. Vous devriez voir les actions et les réactions correspondantes dans Factory I/O et sur le PLC S7-1200 (Figure 3.24).

Ces étapes devraient nous permettre d'interfacer Factory I/O avec l'HMI KTP900 et le PLC S7-1200.

3.6.3 Résultat obtenu de la liaison de notre programme avec factory I/O

Après la liaison réussie entre Factory I/O, notre HMI KTP900 et le PLC S7-1200, lorsque nous appuyons sur le bouton "Start" dans Factory I/O, cela démarrera le moteur à la fois sur l'interface HMI KTP900, dans la réalité et dans Factory I/O. De même, lorsque nous appuyons sur les boutons "Start" ou "Stop" sur l'HMI KTP900, la régulation sera effectuée simultanément dans Factory I/O et sur le moteur réel. Cette synchronisation des actions entre les différents systèmes (Factory I/O, HMI KTP900 et moteur réel) montre que notre modèle de régulation et notre programme peuvent fonctionner efficacement dans un environnement industriel.

De plus, nous obtenons les mêmes résultats de vitesse à la fois sur l'HMI KTP900 et dans Factory I/O, ce qui confirme que les mesures de vitesse sont correctement transmises depuis le moteur réel via le PLC S7-1200 et sont affichées avec précision sur les deux

interfaces.

Cette preuve de concept démontre que nos modèles, nos réglementations et nos procédures peuvent fonctionner de manière fiable et cohérente dans des environnements industriels réels. Cela confirme que notre solution est prête à être mise en œuvre et utilisée pour le contrôle et la régulation d'équipements industriels.

3.7 Conclusion

Ce chapitre de notre projet a été couronné de succès en créant un simulateur de régulation de la vitesse d'un moteur asynchrone à l'aide d'API S7-1200 de SIEMENS. Grâce à l'utilisation du régulateur PID-Compact, qui est intégré à l'API S7-1200 de SIEMENS, nous avons pu contrôler efficacement la vitesse du moteur en ajustant la puissance fournie. Le principe de régulation proportionnelle-intégrale (PI) a été appliqué avec succès, permettant d'obtenir une réponse rapide et précise du système, assurant ainsi une régulation stable et fiable de la vitesse du moteur. L'interface homme-machine (IHM) a joué un rôle crucial en fournissant une interface conviviale pour surveiller et contrôler le moteur. Le logiciel Factory I/O a été utilisé pour valider notre application en fournissant un environnement de simulation réaliste d'une usine. En résumé, ce chapitre a démontré la faisabilité et l'efficacité de la création d'un simulateur de régulation de la vitesse d'un moteur asynchrone à l'aide de l'API S7-1200 de SIEMENS. Cette réalisation ouvre la voie à de nombreuses applications pratiques dans le domaine de l'industrie, en permettant la régulation précise et fiable de la vitesse des moteurs asynchrones.

Ce projet de fin d'étude a été couronné de succès en réalisant la création d'un simulateur de moteur asynchrone en utilisant l'API S7-1200 de marque SIEMENS. Les objectifs initiaux ont été pleinement atteints, et ce travail a contribué à approfondir notre compréhension des régulations industrielles, des technologies de simulation et de la programmation des automates programmables industriels.

Dans le premier chapitre, nous avons exploré différentes technologies utilisées en ingénierie industrielle, telles que les simulateurs, les machines asynchrones et les automates programmables industriels. Nous avons souligné l'importance des simulateurs pour la formation des opérateurs et la simulation d'environnements complexes, ainsi que les avantages des machines asynchrones en termes d'efficacité et de fiabilité. De plus, nous avons souligné le rôle essentiel des automates programmables industriels dans la gestion efficace de la production.

Le deuxième chapitre s'est concentré sur la comparaison des résultats obtenus dans TIA Portal et MATLAB Simulink pour valider les modèles de moteur asynchrone. Les résultats concordants entre les deux outils ont confirmé la validité et la précision des modèles, soulignant ainsi leur fiabilité pour la simulation du comportement du moteur dans différents scénarios de fonctionnement.

Le troisième chapitre a été couronné de succès en créant un simulateur de régulation de la vitesse d'un moteur asynchrone à l'aide d'API S7-1200 de SIEMENS. Nous avons démontré l'efficacité du régulateur PID-Compact pour contrôler la vitesse du moteur en ajustant la puissance fournie, tandis que l'HMI a fourni une interface conviviale pour surveiller et contrôler le moteur. La validation de notre application à l'aide de Factory I/O a permis de tester différentes situations et de vérifier la cohérence des résultats avec les attentes.

Ce projet a été une expérience enrichissante qui a combiné des connaissances théoriques, des compétences techniques et une approche pratique pour réaliser un simulateur fonctionnel d'un moteur asynchrone. Il constitue une base solide pour de futures recherches et applications dans le domaine de la régulation industrielle et de l'automatisation.

Bibliographie

- [1] “European flight simulator,” <https://www.groupon.be/deals/european-flight-simulator-2>, consulté le 28 avril 2023.
- [2] A. Technologies, “Intelligent traffic systems,” <https://agtkw.com/project/intelligent-traffic-systems/>, 2023.
- [3] RTS, “Les capteurs électroniques des autos, utiles à la police et à la justice,” *RTS.ch*, 2019. [Online]. Available : <https://www.rts.ch/info/sciences-tech/10590051-les-capteurs-electroniques-des-autos-utiles-a-la-police-et-a-la-justice.html>
- [4] “Factory IO,” <https://factoryio.com>, consulté le [date].
- [5] “node47,” Web Page, 2023. [Online]. Available : <http://relatifetincongru.free.fr/node47.html>
- [6] “Chapitre 3 partie ii,” PDF, 2023. [Online]. Available : http://elearning.centre-univ-mila.dz/pluginfile.php/91372/mod_resource/content/1/Chapitre%203%20partie%20II.pdf
- [7] “Saleak mohamed ch’beil,” PDF, 2019. [Online]. Available : <https://biblio.univ-annaba.dz/ingeniorat/wp-content/uploads/2019/09/Saleak-Mohamed-ch%E2%80%99beil.pdf>
- [8] D. T. YAPTA. (2012) Commande vectorielle à flux rotorique orienté de la machine asynchrone :simulation et expérimentation. [Online]. Available : <https://www.memoireonline.com/04/12/5669/Commande-vectorielle--flux-rotorique-orienté-de-la-machine-asynchrone-simulation-et-experimentation.html>
- [9] Nom de l’auteur ou de l’organisation (si disponible), “Instrumentation automates programmables industriels (api),” <https://cabinetnpm.com/instrumentation-automates-programmables-industriels-api/>, date, consulté le 23 juin 2023.

- [10] S. France. (2023) Simatic controllers – prenez le contrôle du futur. [Online]. Available : <https://new.siemens.com/fr/fr/produits/automatisation-entrainements/systemes-automatisation/industrial/plc.html>
- [11] K. J. Astrom and R. M. Murray, *Feedback systems : An introduction for scientists and engineers*. Princeton University Press, 2010.
- [12] O. Dutailis, *Le simulateur*. Gallimard, 1987.
- [13] R. Narasimhan, “Applications of simulation and analysis tools in engineering and science,” *Procedia Engineering*, vol. 97, pp. 1–8, 2014.
- [14] M. K. Watfa, *Simulation and Modelling : Concepts and Tools*. Springer Science Business Media, 2013.
- [15] S. Djavadian, M. Rezaei, K. Hassani, A. AghaKouchak, and H. Rakha, “Autonomous vehicle simulators : A comprehensive survey,” *Transportation Research Part C : Emerging Technologies*, vol. 111, pp. 312–337, 2020.
- [16] K. Nagel and M. Schreckenberg, “Traffic simulation : State of the art,” *Proceedings of the IEEE*, vol. 80, no. 12, pp. 1675–1696, 1992.
- [17] L. Tesfatsion and K. L. Judd, *Handbook of computational economics, Volume 2 : Agent-based computational economics*. Elsevier, 2006.
- [18] D. L. Fisher, M. Rizzo, J. Caird, and J. D. Lee, Eds., *Handbook of Driving Simulation for Engineering, Medicine, and Psychology*. CRC Press, 2011.
- [19] S. Kearns and D. MacKenzie, Eds., *Flight Simulation : Virtual Environments in Aviation*. Ashgate Publishing, 2012.
- [20] J. S. Graf and J. Hinkelbein, “The role of simulation in training for crisis management in aviation,” *International Journal of Emergency Medicine*, vol. 7, no. 1, pp. 1–5, 2014.
- [21] D. M. Gaba and S. K. Howard, Eds., *Simulation-Based Training in Healthcare : Delivery in High-Stakes Environments*. Springer, 2012.
- [22] D. S. Williamson and A. Johri, *Virtual Reality and Augmented Reality Applications in Manufacturing*. CRC Press, 2020.
- [23] D. Fisher, M. Rizzo, J. Caird, and J. Lee, “Handbook of driving simulation for engineering, medicine, and psychology,” 2011.
- [24] S. Kearns and D. MacKenzie, *Flight Simulation : Virtual Environments in Aviation*. Ashgate Publishing, Ltd., 2012.
- [25] D. Gaba and S. Howard, *Simulation-Based Training in Healthcare : Delivery in High-Stakes Environments*. Springer Science Business Media, 2012.

- [26] “Efficacité énergétique : nos conseils et expertise en chauffage, climatisation, énergies renouvelables,” https://conseils.xpair.com/consulter_parole_expert/moteurs-de-ventilation-gisement-economies.htm, accessed : 2023-05-12.
- [27] J. Gupta and R. Anand, *Induction Motors and their Applications*. S.K. Kataria & Sons, 2008.
- [28] T. Wildi, *Electrical Machines, Drives, and Power Systems*. Pearson, 2013.
- [29] J. Lutz and W. Roduner, *Asynchronous Machines : Construction, Basic Principles and Applications*. Springer, 2009.
- [30] H. Toliyat and S. Campbell, *Asynchronous Motors : Applications, Design and Optimization*. CRC Press, 2019.
- [31] “Modélisation et commande de la machine asynchrone.” [Online]. Available : http://www.baghli.com/dl/courscmde/cours_cmde_MAS.pdf
- [32] W. H. Yeadon and A. W. Yeadon, *Handbook of Small Electric Motors*. CRC Press, 2011.
- [33] B. de Fornel, “Machines asynchrones - commande par contrôle scalaire,” *Techniques de l’Ingénieur*. [Online]. Available : <https://www.techniques-ingenieur.fr/base-documentaire/energies-th4/machines-electriques-tournantes-conception-construction-et-commande-42252210/machines-asynchrones-d3622/>
- [34] “Les techniques de commande du moteur asynchrone.” [Online]. Available : <https://eduscol.education.fr/sti/sites/eduscol.education.fr.sti/files/ressources/techniques/688/688-gt-commande-mot.pdf>
- [35] S. Sassi and A. Abdelkafi, *Commande des machines électriques*. Springer, 2015.
- [36] F. Meibody-Tabar, *Commande des machines électriques et de l’électronique de puissance*. Dunod, 2016.
- [37] L. Baghli, “Modélisation et commande de la machine asynchrone,” 2020. [Online]. Available : http://www.baghli.com/dl/courscmde/cours_cmde_MAS.pdf
- [38] X. Kestelyn, *Machines électriques : Commande des machines à courant alternatif*. De Boeck Supérieur, 2013.
- [39] J.-P. Louis, *Commande vectorielle des machines électriques : Régulation en flux orienté*. Dunod, 2012.
- [40] F. Wurtz, *Commande des machines asynchrones*. Ellipses, 2007.
- [41] “Commande vectorielle,” Wikipédia. [Online]. Available : https://fr.wikipedia.org/wiki/Commande_vectorielle

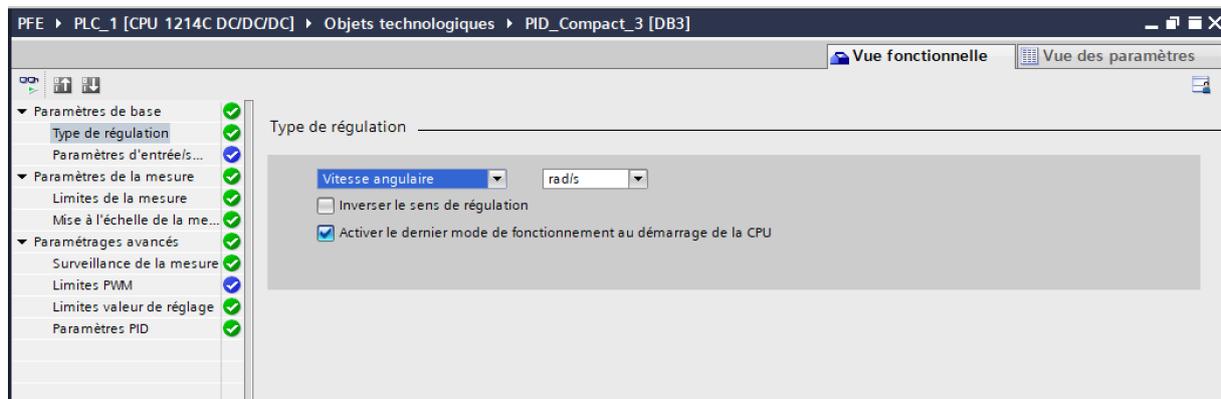
- [42] “Commande vectorielle de la machine asynchrone sous la carte ds1104.” [Online]. Available : https://www.academia.edu/19689514/Commande_Vectorielle_de_la_Machine_Asynchrone_sous_la_Carte_DS1104
- [43] “Automate programmable industriel,” Wikipédia. [Online]. Available : https://fr.wikipedia.org/wiki/Automate_programmable_industriel
- [44] “Automation industrielle,” Wikipédia. [Online]. Available : https://fr.wikipedia.org/wiki/Automation_industrielle
- [45] “Les automates programmables industriels (api).” [Online]. Available : <https://www.technologuepro.com/cours-automate-programmable-industriel/Les-automates-programmables-industriels-API.htm>
- [46] “Exposé sur l’automates programmables industriels.” [Online]. Available : <https://www.cours-et-exercices.com/2014/05/expose-sur-lautomates-programmables.html>
- [47] “Systèmes d’automatisation.” [Online]. Available : <https://new.siemens.com/fr/fr/produits/automatisation-entrainements/systemes-automatisation.html>
- [48] Siemens, “Simatic s7-1200 - simatic controller - global,” <https://www.siemens.com/global/en/products/automation/systems/industrial/plc/s7-1200.html>, consulté le [date de consultation].
- [49] LeBigData.fr, “Interface homme-machine : tout savoir sur les ihm,” <https://www.lebigdata.fr/interface-homme-machine-tout-savoir-sur-les-ihm>, consulté le 28 avril 2023.
- [50] COPA-DATA, “Qu’est-ce qu’une ihm? interface homme-machine,” <https://www.copadata.com/fr/produits/zenon-software-platform/visualisation-controle/quest-ce-qu-une-ihm-interface-homme-machine-copa-data/>, consulté le 28 avril 2023.
- [51] Pro-face, “Definition pratique ihm (interface homme machine),” <https://www.proface.com/fr/ihm>, consulté le 28 avril 2023.
- [52] Siemens Industry Online Support, “WinCC Runtime Professional v16 (TIA Portal),” [https://support.industry.siemens.com/cs/document/109745178/wincc-runtime-professional-v16-\(tia-portal\)?dti=0&lc=en-US](https://support.industry.siemens.com/cs/document/109745178/wincc-runtime-professional-v16-(tia-portal)?dti=0&lc=en-US), 2021, accessed : Apr 28, 2023.
- [53] “Siemens Industry Catalog - SIMATIC WinCC (Runtime Advanced and Runtime Professional),” <https://mall.industry.siemens.com/mall/en/WW/Catalog/Products/10000309>.
- [54] J. Moreno, J. Fernández de Canete, and F. Gómez-Cambronero, “OPC UA implementation in a SCADA system based on WinCC runtime advanced and S7-1500,”

- in *2018 IEEE International Conference on Industrial Technology (ICIT)*, 2018, pp. 1005–1010.
- [55] M. T. Alnaser and A. A. Ibrahim, “Modeling and simulation of induction motors using matlab/simulink environment,” *2009 International Conference on Computer Engineering Systems, ICCES*, pp. 6–11, 2009.
- [56] M. Kolcun and M. Guzan, “Dynamic model of induction motor in stationary reference frame,” *Acta Polytechnica*, vol. 57, no. 3, pp. 166–173, 2017.
- [57] J. Mativo, “Modeling of an induction motor,” *International Journal of Engineering Research and Development*, vol. 10, no. 1, pp. 01–10, 2014.
- [58] A. Kumar and P. Ranjan, “Induction motor modeling : A review,” *International Journal of Innovative Research in Science, Engineering and Technology*, vol. 3, no. 6, pp. 12 832–12 839, 2014.
- [59] N. Mohan, “Electric machines and drives : principles, control, modeling and simulation,” 2009.
- [60] “Inrs - institut national de recherche et de sécurité pour la prévention des accidents du travail et des maladies professionnelles,” <https://www.inrs.fr/>, 2023, consulté le 29 avril 2023.
- [61] Y. Zhu, “Tuning of pid controllers and pi controllers,” *Journal of Process Control*, vol. 24, no. 3, pp. 255–266, 2014.
- [62] E. Guglielmino, A. Rizzo, and S. Strano, “Comparative analysis of different tuning methods for pi controller in frequency domain,” in *2015 IEEE International Symposium on Industrial Electronics*. IEEE, 2015, pp. 872–877.

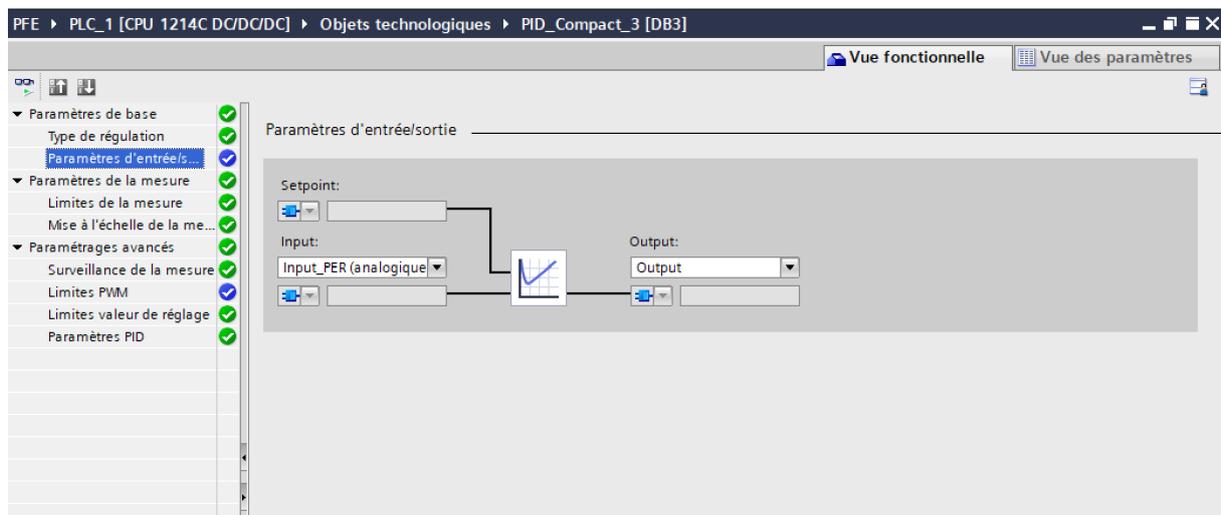
ANNEXE

➤ Paramètre de régulateur PID_Compact

1. Dans cette étape, nous choisissons le type de régulation, en l'occurrence la régulation de la vitesse angulaire en rad/s. De plus, nous activons le dernier mode de fonctionnement au démarrage de l'unité centrale (CPU).

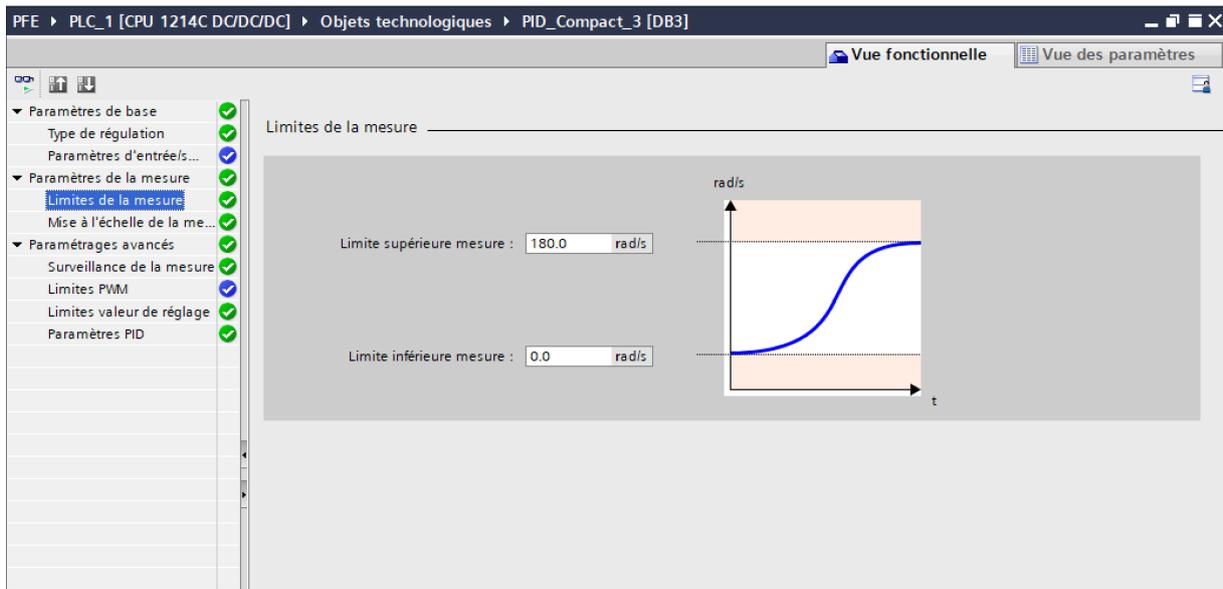


2. Dans les paramètres d'entrée et de sortie du régulateur PID Compact, nous avons choisi les signaux appropriés pour l'entrée (input) et la sortie (output) du régulateur.

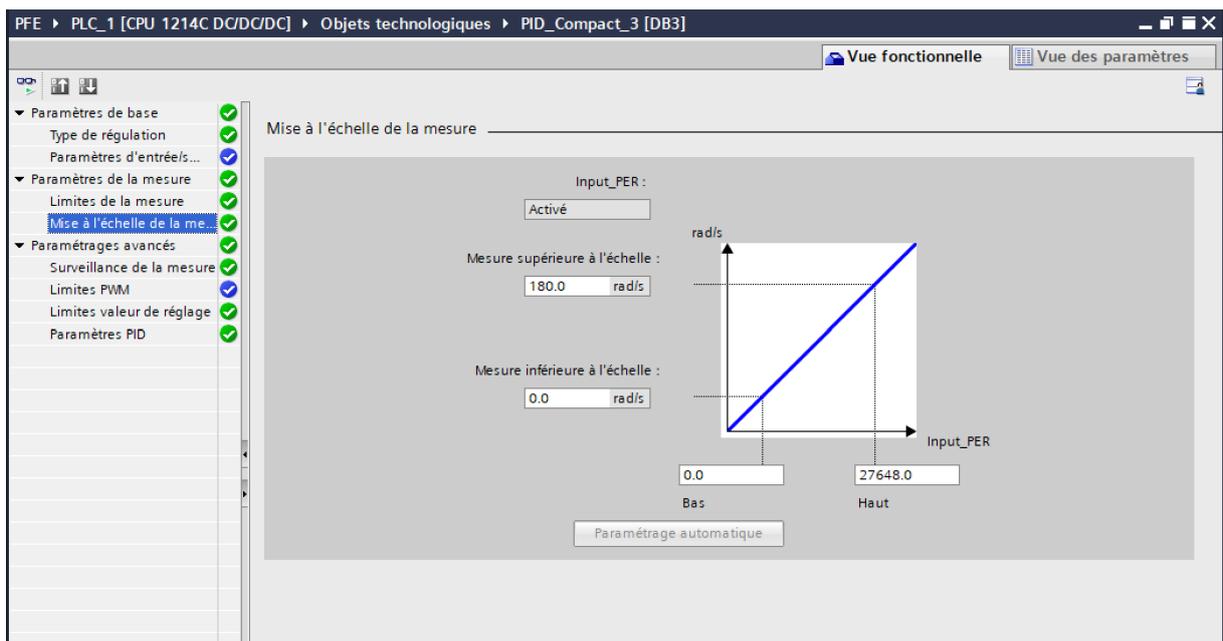


3. Dans les limites de mesure, nous avons défini une limite supérieure de 180 rad/s et une limite inférieure de 0 rad/s pour la vitesse angulaire du moteur.

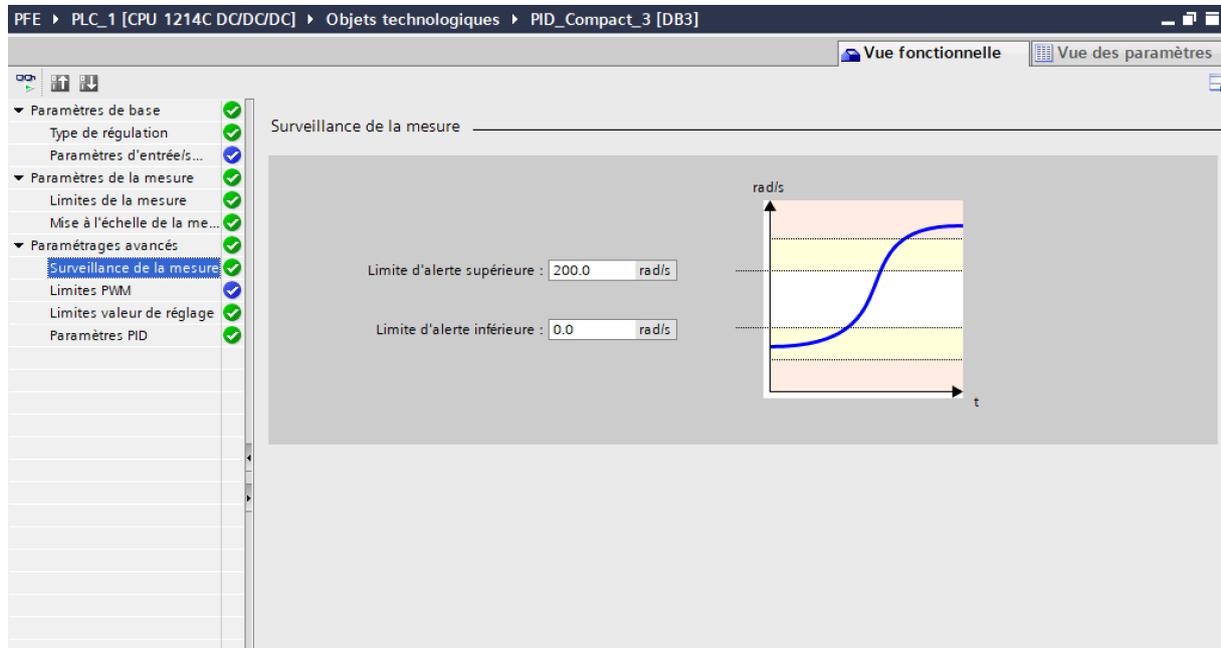
Cela signifie que le régulateur PID Compact s'attend à ce que la vitesse mesurée du moteur reste dans cette plage de valeurs.



4. Dans ce cas, une vitesse angulaire de 180 rad/s correspondrait à une valeur numérique d'environ 27 658 sur 16 bits.

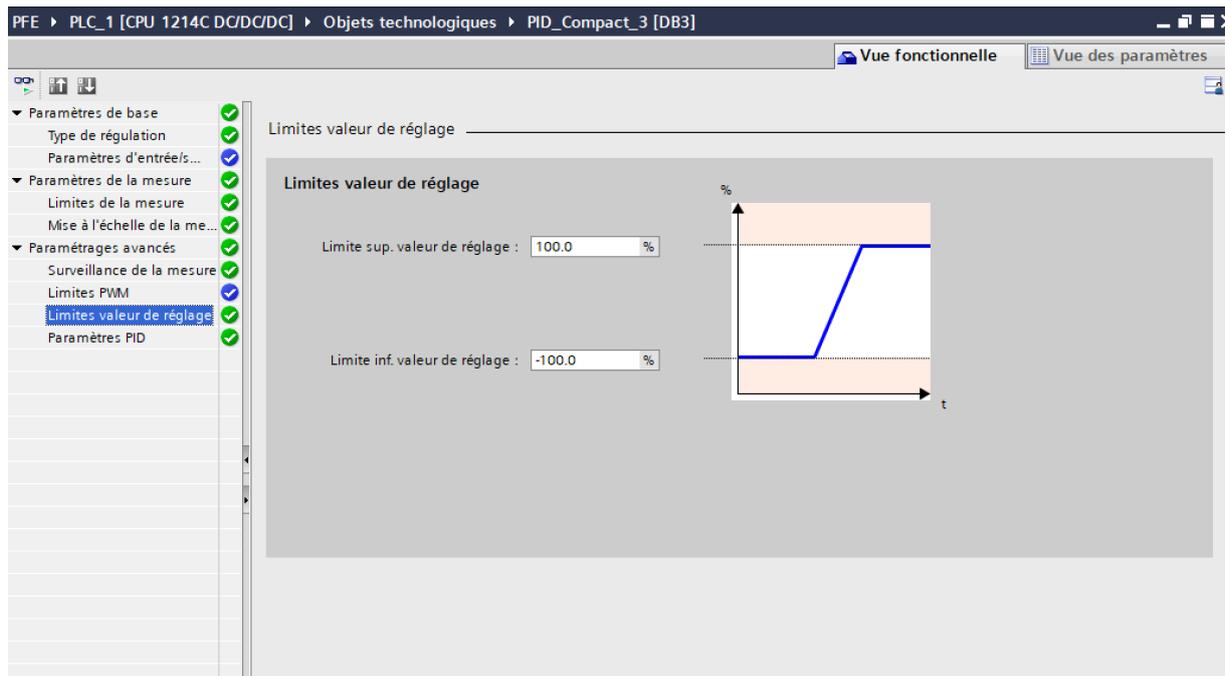


5. Dans la surveillance de la mesure, nous avons choisi une limite d'alerte supérieure de 200 rad/s. Cela signifie que si la vitesse angulaire mesurée dépasse cette limite, une alerte sera déclenchée pour signaler un dépassement de seuil.



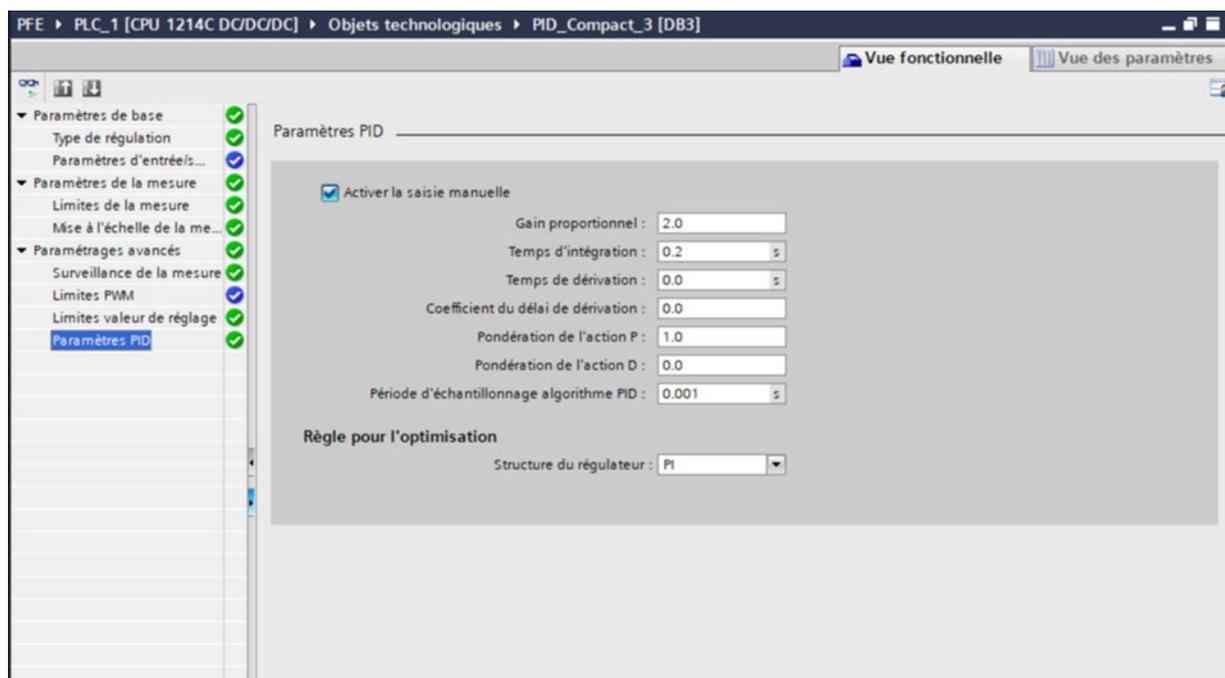
6. Dans la limite de la valeur de réglage, nous avons choisi une limite supérieure de 100% et une limite inférieure de 0%. Cela signifie que la valeur de réglage de la régulation peut varier entre 0% et 100% de la plage totale disponible.

La limite supérieure de 100% indique que la régulation peut être réglée à son maximum, ce qui permet de fournir la pleine puissance ou l'action de contrôle maximale au système régulé. Cela peut être utile lorsque nous souhaitons atteindre une vitesse maximale ou un certain niveau de performance.



7. Dans les paramètres de notre régulateur PID, nous avons choisi un gain proportionnel de 2 et un temps d'intégration de 0.2 Le temps d'intégration est utilisé pour corriger l'erreur cumulée au fil du temps.

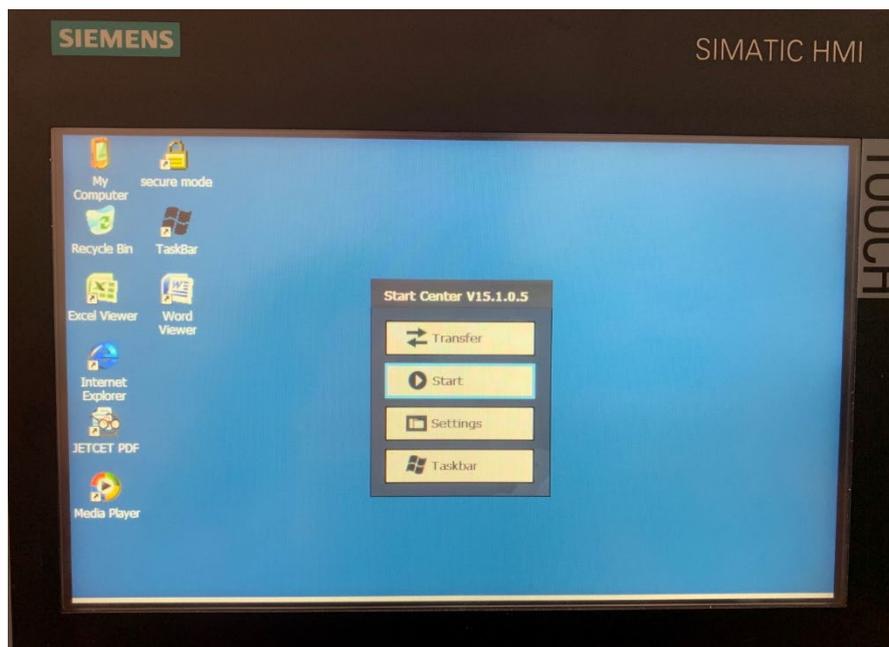
En activant la saisie manuelle, nous donnons la possibilité de régler manuellement la valeur de sortie du régulateur. Cela peut être utile pour effectuer des réglages initiaux ou pour contourner temporairement le contrôle automatique



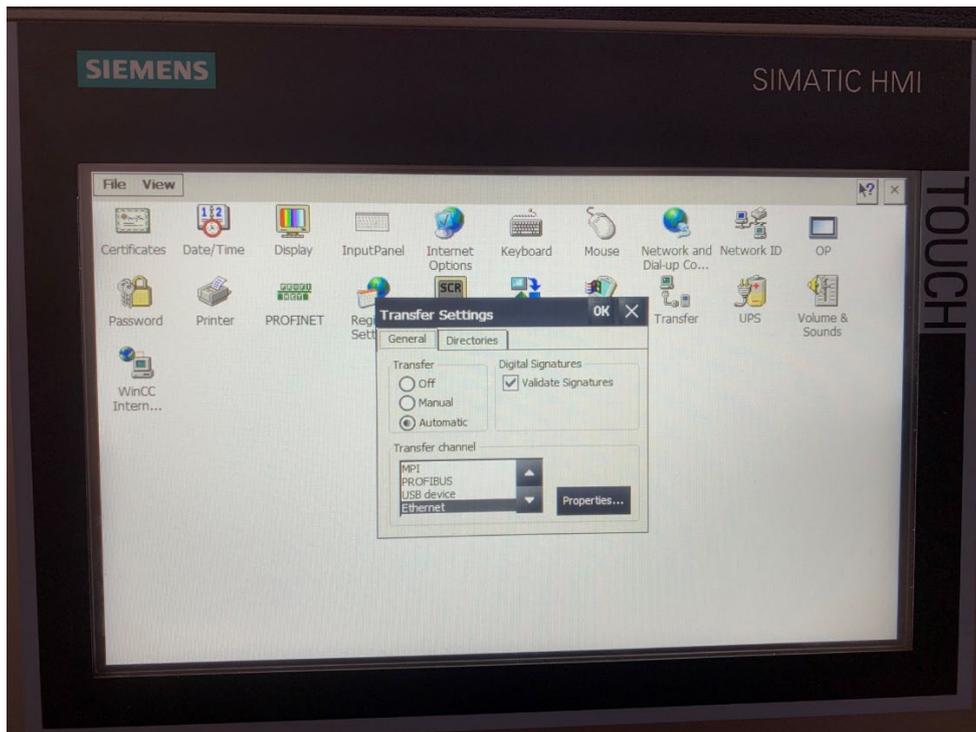
➤ Configuration de HMI KTP 900

Pour charger et connecter un HMI KTP 900 à TIA Portal, suivez les étapes suivantes :

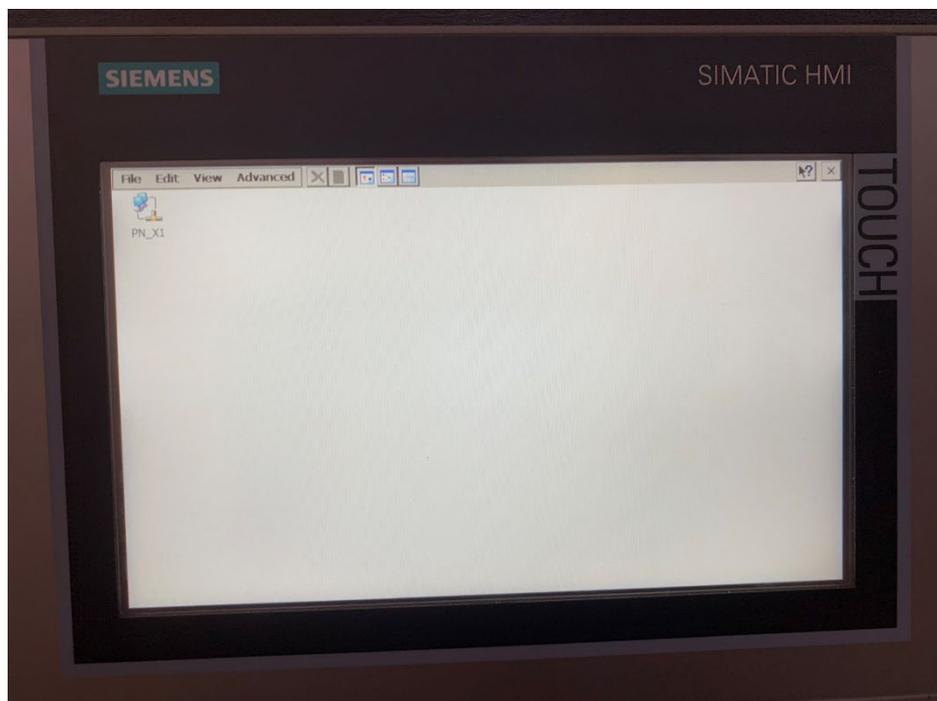
1. Alimentation du HMI : Assurez-vous que le HMI KTP 900 est correctement alimenté en utilisant une source d'alimentation appropriée.
2. Configuration du transfert : Dans HMI, accédez aux paramètres de transfert en cliquant sur l'icône "Settings" dans la barre d'outils.



3. Transfert automatique et canal de transfert Ethernet : Sélectionnez l'option de transfert automatique pour permettre une mise à jour automatique des données entre le HMI et TIA Portal. Assurez-vous également de sélectionner le canal de transfert Ethernet pour établir une connexion réseau.

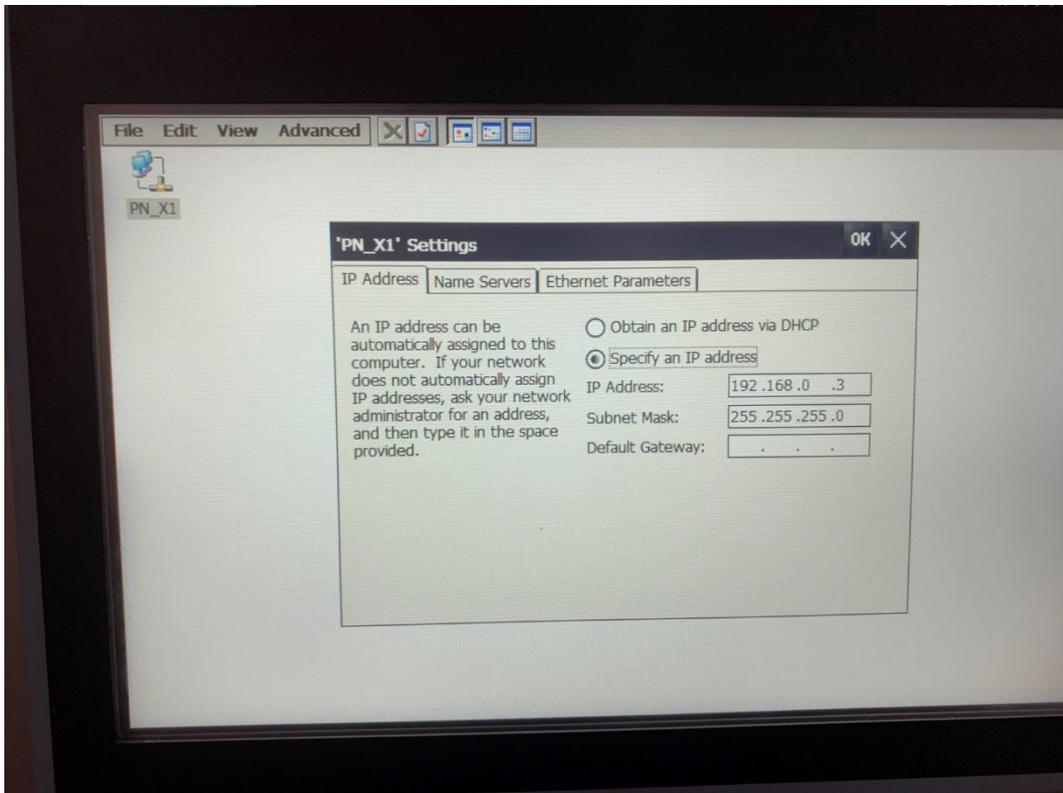


4. Configuration de la connexion réseau et du numéroteur : Dans les paramètres réseau et numéroteur du HMI, sélectionnez "PN_X1" pour configurer la connexion Ethernet.



5. Attribution d'une adresse IP au HMI :

Donnez une adresse IP unique au HMI KTP 900. Assurez-vous que cette adresse IP est différente de celle de l'automate CPU et de l'ordinateur, et changé le aussi dans le TIA Portal.



Assurez-vous également de configurer les autres paramètres nécessaires dans TIA Portal, tels que les tags de communication et les écrans HMI, pour établir une communication fonctionnelle entre le HMI et le système de contrôle.