الجـــــــــــــــــــمـهـوريـــــة الجـــــــــــــزائـريـة الديمـــــــقـراطيـة الـشـــــــــعبيـة

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التـــــــــــــــــــــــــــعـليـم العـــــــالي و البـــــــــحث العـــــلـمـــي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جـــــــــــــــامعة أبي بكـــر بـلـقـايد – تـلمســـــــــــــان –

Université Aboubekr Belkaïd – Tlemcen –
Faculté de TECHNOLOGIE

MEMOIRE

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

En : Electrotechnique

Spécialité : Commandes Electriques

Par : DAHAOUI YOUNES

Sujet

Réalisation d'un banc d'essai pour une régulation de position à base d'API S7-1200

Soutenu publiquement, le 11 / 06 / 2026 , devant le jury composé de :

Mme BENHADDA Yamina	MCA	Université de Tlemcen	Président
Mr LOUCIF Mourad	MCA	Université de Tlemcen	Examinateur
Mr MELIANI Sidi Mohammed	Pr	Université de Tlemcen	Encadreur

Année universitaire : 2025 / 2026

Dédicaces

C'est avec une profonde émotion que je dédie cet humble travail

A à la mémoire de mon cher oncle, Slimane Dahaoui, qui nous a quitté trop tôt en 2018. Même si j'étais encore tout petit, la bienveillance de ton regard, ton affection et tes précieux conseils m'ont marqué à jamais et ont grandement contribué à façonner la personne que je suis devenue. Toi qui portais en moi une confiance absolue et qui nourrissais de grands rêves de réussite pour mon avenir, tu as toujours été une source d'inspiration unique.

Aujourd'hui, en arrivant au terme de ce projet de fin d'études qui marque la fin d'un long parcours universitaire, mes pensées se tournent naturellement vers toi. J'espère de tout cœur que ce travail et ce chemin parcouru honoreront la confiance et l'estime que tu me portais. Ton souvenir et l'ambition que tu avais pour moi restent le moteur de ma motivation au quotidien. Ce diplôme est le fruit de la force que tu as su me transmettre.

Que Dieu te fasse miséricorde, t'accorde Son pardon et t'accueille dans Son vaste paradis. Tu resteras à jamais vivant dans mes prières et gravé dans mon cœur.

A mes chers parents auxquels je témoigne ma profonde reconnaissance. Je les remercie pour leur amour, leur soutien indéfectible, leurs sacrifices et leurs encouragements constants tout au long de mon parcours. Leur confiance en moi a toujours été une source de motivation et de persévérance. Je leur dédie ce travail en signe de respect, d'affection et de gratitude pour tout ce qu'ils ont fait pour moi.

A ma petite sœur Lina, dont la présence a toujours été une source de joie et de motivation. Je lui exprime toute mon affection et lui souhaite beaucoup de succès dans sa vie future.

Remerciements

Avant tout, je remercie Dieu le Tout-Puissant de m'avoir accordé la santé, la volonté et la patience nécessaires pour mener à bien ce travail.

Je tiens à exprimer ma profonde gratitude à mon encadreur, M. MELIANI Sidi Mohammed, pour son accompagnement, sa disponibilité, ses précieux conseils et son soutien tout au long de la réalisation de ce projet. Ses orientations et ses remarques constructives ont été d'une grande aide pour l'aboutissement de ce travail.

J'adresse également mes sincères remerciements aux membres du jury, M. Loucif Mourad et Mme Benhadda Yamina, pour l'honneur qu'ils me font en acceptant d'évaluer ce mémoire et pour l'intérêt qu'ils portent à ce travail.

Je remercie l'ensemble des enseignants qui ont contribué à ma formation et m'ont transmis les connaissances scientifiques et techniques qui m'ont permis de mener à bien ce projet.

Enfin, je remercie toutes les personnes qui ont contribué, de près ou de loin, à la réalisation de ce mémoire.

DAHAOUI Younes

Table des matières

Liste des figures ii
Liste des abréviations iv
Introduction Générale 1
Chapitre I 2
I.1 Introduction 2
I.2 Exemple de régulation dans différents domaines 2
I.2.1 Chirurgie robotisée 2
I.2.2 Stabilisation de la caméra 3
I.2.3 Système de guidage aérospatial 3
I.3 Exemple de régulation dans l'industrie à base d'API 4
I.3.1 Station de remplissage et de bouchonnage 4
I.3.2 Système de stockage et déstockage 5
I.4 Principe de régulation 7
I.5 Description de notre prototype 8
I.5.1 Organe de commande (API) et alimentation 9
I.5.2 Pré actionneur et actionneur 9
I.5.3 Capteur 10
I.5.4 Supervision (IHM) 11
I.6 Conclusion 11
Chapitre II 12
II.1 Introduction 12
II.2 Régulation de position 12
II.3 Processus et fonction de transfert 13
II.3.1 Modélisation du MCC à aimant permanent 13
II.3.2 Réducteur 16
II.4 Pont en H 17
II.5 Capteur de déplacement 18
II.5.1 Détermination de la distance parcourue 19
II.5.2 Application numérique pour notre encodeur 19
II.6 Régulateur PID 19
II.6.1 Action proportionnelle 19
II.6.2 Action intégrale 19

II .6.3 Action dérivée 20
II.6.4 PID_Compact 20
II.7 Conclusion 24
Chapitre III 25
III.1 Introduction 25
III.2 Présentation de notre banc d'essai 25
III.2.1 Partie mécanique 26
III.2.2 Partie électrique 26
III.2.3 Partie programmation 31
III.2.4 Partie supervision 46
III.3 Test et interprétation des résultats 49
III.4 Conclusion 51
Conclusion Générale 53

Liste des figures

Figure I.1 : Chirurgie Robotisée. [5] 2
Figure I.2 : Stabilisation de Caméra. [9] 3
Figure I.3 : Système de guidage aérospatial. [13] 3
Figure I.4 : Station de distribution de bouteilles. 4
Figure I.5 : Station de remplissage. 4
Figure I.6 : Station de bouchonnage. 5
Figure I.7 : Système de stockage / déstockage 5
Figure I.8 : Axe de déplacement horizontal. 6
Figure I.9 : Axe de déplacement vertical. 6
Figure I.10 : Axe pour manipuler le bras télescopique. 7
Figure I.11 : Exemple de régulation précise. [14] 8
Figure I.12 : Photo du prototype. 8
Figure I.14 : Variateur de vitesse. 9
Figure I.15 : MCC à aimant permanent. 10
Figure I.16 : Réducteur mécanique. 10
Figure I.17: Encoder incremental. [16] 10
Figure I.18 : Interface homme machine. 11
Figure II.1 : Schéma bloc pour une régulation de position. [17] 13

Figure II.2 : Phases de variation de vitesse du moteur. [18] 13
Figure II.3 : Représentation schématique d'un MCC à aimant permanent. [19] 14
Figure II.4 : Schéma de montage d'un pont en H. [20] 17
Figure II.5 : Signal PWM avec variation du rapport cyclique. [21] 18
Figure II.6 : Schéma fonctionnel d'un régulateur PID. [22] 20
Figure II.7 : Schéma fonctionnel du PID_Compact. [23] 21
Figure III.1 : Photo du banc d'essai numéroté. 25
Figure III.2 : Variateur de vitesse réalisé. 28
Figure III.3 ; Circuit PCB sur ARES-Proteus. 29
Figure III.4 : Boitier imprimé en 3D (Vues : Coté et derrière). 29
Figure III.5 : Moteur à courant continue + Réducteur. 30
Figure III.6 : Création du projet sur TIA Portal. 31
Figure III.7 : Ajout de la CPU utilisée. 32
Figure III.8 : Ajout du PC Système avec WinCC RT Advanced. 32
Figure III.9 : Ajout de l'IHM TP900 Comfort de Siemens pour la supervision et la commande. 33
Figure III.10 : Ajout du module de sortie analogique. 33
Figure III.11 : Création de la carte réseaux pour le PC Système. 34
Figure III.12 : Attribution des adresses IP. 35
Figure III.13 : Choix du HSC. 36
Figure III.14 : Mode de fonctionnement. 36
Figure III.15 : Choix d'entrée physique. 37
Figure III.16 : Adresses de début et de fin. 37
Figure III.17 : Création du bloc CTRL_HSC. 38
Figure III.18 : Fonction SCL pour le calcul de distance parcourue par le biais de l'encodeur. 39
Figure III.19 : Configuration complète du HSC. 39
Figure III.20 : Choix du générateur d'impulsions. 40
Figure III.21 : Paramétrage et adresse de sortie physique ou matérielle. 40
Figure III.22 : Adresses de début et de fin. 41
Figure III.23 : Création du bloc CTRL_PWM. 41
Figure III.24 : Normalisation et mise à l'échelle du signal de commande. 42
Figure III.25 : Création du bloc PID_Compact. 43
Figure III.26 : Type de régulation 'mm'. 44
Figure III.27 : Sélection des paramètres d'entrées. 44
Figure III.28 : Les paramètres du PID. 45

Figure III.29 : Création du Main Data Base. 45
Figure III.30 : IHM TP900 Confort. 46
Figure III.31 : Variables de l'IHM. 47
Figure III.32 : Création du bouton Marche/Arrêt. 47
Figure III.33 : Champ d'entrée. 48
Figure III.34 : Champ de sortie. 48
Figure III.35 : Vue finale de l'IHM. 49
Figure III.36 : Vue IHM représentant la régulation pour une consigne = 40 mm. 50
Figure III.37 : Réponse du système pour une consigne = 40 mm. 50
Figure III.38 : Vue IHM représentant la régulation pour une consigne = 10 mm. 51
Figure III.39 : Réponse du système pour une consigne = 10 mm. 51

Liste des abréviations

U : Tension du réseau [V]
ia : Courant de l'induit [A]
C : Couple moteur [Nm].
Cr : Couple résistant
Cem : Couple électromagnétique
E : Force contre électromotrice [V]
Ra : Résistance de l'induit [Ω]
La : Inductance de l'induit [H]
K : Gain [N.m/I]
Ω : Vitesse angulaire [tr/s]
J : L'inertie [kg m^2]
Te : Constante de temps électrique
T : Constante de temps électromécanique
R_{ed} : Coefficient de réduction
Ωr : Vitesse angulaire réduite
Θ : Position angulaire

Introduction Générale

Introduction Générale

La régulation automatique est aujourd'hui présente dans de nombreux domaines technologiques ou la précision, la rapidité et la fiabilité sont des exigences essentielles. Elle permet à un système de maintenir ou d'atteindre une valeur de consigne malgré les perturbations extérieures et les variations de fonctionnement.

Les applications de la régulation sont nombreuses et couvrent des secteurs très variés. Dans le domaine médical, elle est utilisée dans les systèmes de chirurgie robotisée afin de garantir des mouvements précis et sécurisés. Dans le domaine de l'imagerie, les dispositifs de stabilisation de caméra exploitent des boucles de régulation pour compenser les mouvements indésirables et améliorer la qualité des prises de vue. Dans le secteur aérospatial, les systèmes de guidage et de contrôle d'attitude reposent également sur des techniques de régulation permettant d'assurer la stabilité et l'orientation des véhicules spatiaux.

Dans l'industrie, la régulation joue un rôle fondamental dans l'automatisation des procédés. Elle est notamment utilisée dans les stations de remplissage et de bouchonnage afin de garantir la précision des opérations, ainsi que dans les systèmes automatisés de stockage et de déstockage ou le positionnement des équipements doit être réaliser avec une grande précision. La mise en œuvre de ces systèmes est généralement assurée par des automates programmables industriels, qui constituent aujourd'hui l'un des principaux moyens de commande des installations industrielles.

Parmi les différentes applications de la régulation, la régulation de position occupe une place importante, car elle intervient dans de nombreux systèmes nécessitant un contrôle précis du déplacement ou de l'orientation d'un organe mécanique. Ce mémoire s'intéresse à l'étude et à la réalisation d'un système de régulation de position d'un moteur à courant continue piloter par un API Siemens S7-1200.

Afin de répondre à cet objectif, ce mémoire est structuré en trois chapitres. Le premier représente une vue générale sur les systèmes automatisés et aussi le prototype réaliser. Le deuxième est consacré aux principes de régulation industriel avec la modélisation du système et à l'étude des différents composants de la boucle de régulation. Enfin, le troisième détaille la réalisation pratique, la programmation de l'automate, la mise en place de la supervision et l'analyse des résultats expérimentaux obtenus.

Chapitre I

Vue générale sur les systèmes automatisés

I.1 Introduction

Les systèmes automatisés occupent une place centrale dans tous les domaines ou on a besoin d'une précision du mouvement et surtout la coordination des différents éléments sont essentielles pour un fonctionnement précis et fiable, mais ce qui est le plus demandé est le contrôle du déplacement et le positionnement des organes mécaniques qui sont une pièce maitresse dans différents domaines et surtout l'industrie.

Ce chapitre a pour but de donner des exemples de ces systèmes qui sont présent dans plusieurs secteurs et surtout dans l'industrie qui utilise des automates programmables industriels. Nous allons clôturer ce chapitre par une présentation dans la quel il y'auras une description du prototype réaliser et surtout les principaux composants du système.

I.2 Exemple de régulation dans différents domaines

I.2.1 Chirurgie robotisée

La Figure I.1 montre une intervention chirurgicale assistée par ordinateur ou le chirurgien utilise une interface homme machine (IHM) avec un bras robotisé. Ça aide le chirurgien à être plus précis pour atteindre des zones restreintes ou de filtrer les tremblements du chirurgien et non pas à le remplacer [1], [2].

Mais pour arriver à pratiquer ces interventions, il faut tout d'abord avoir une régulation de position millimétrique. Pour réaliser cette régulation, il faut utiliser une commande robuste parce que le PID n'est pas recommandé [3], [4]. La raison que, lors de l'opération chirurgicale, pour passer d'un tissue mou a un muscle ou tout autre structure dure, le bras robotisé (l'actionneur) peut dévier de sa consigne. Pour ça il est recommandé de ne pas utiliser un PID classique.



Figure I.1 : Chirurgie Robotisée. [5]

I.2.2 Stabilisation de la caméra

Les caméras sans stabilisateur subit toutes les perturbations du monde extérieur : mouvements rapides des mains du photographe, du vents etc… [6]. Pour ça, les dernières versions de caméras sont équipées de stabilisateur, comme montré dans la Figure II.2 [7].

Le stabilisateur utilise des capteurs gyroscopes qui, détectant les mouvements, qui donnent une consigne pour un régulateur qui actionne un moteur pour fournir une force contraire pour annuler celle détecter par les gyroscopes [8].

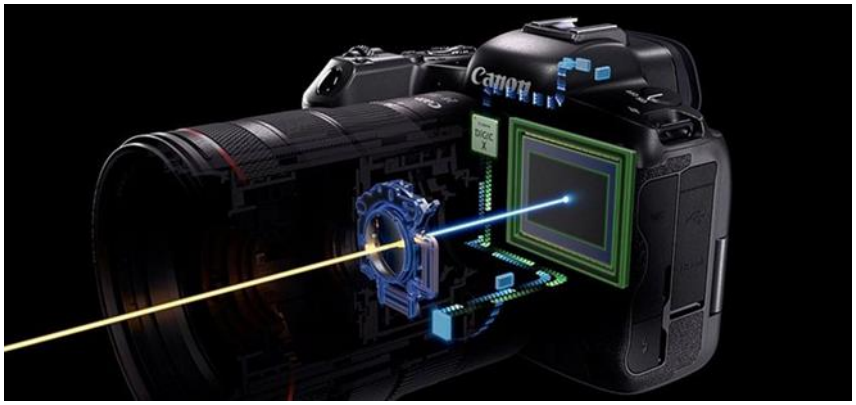


Figure I.2 : Stabilisation de Caméra. [9]

I.2.3 Système de guidage aérospatial

Dans un système aérospatial (Figure I.3), le guidage consiste à déterminer les manœuvres (forces ou accélérations) permettant de déplacer le véhicule vers un objectif (position (r_c) et vitesse (v_c)) en garantissant que la trajectoire respecte des contraintes de vol et, éventuellement, optimise un critère de performance [10].

Pour éliminer ces perturbations, il faut utiliser une commande robuste comme dans le cas de la chirurgie robotisée [11], [12].

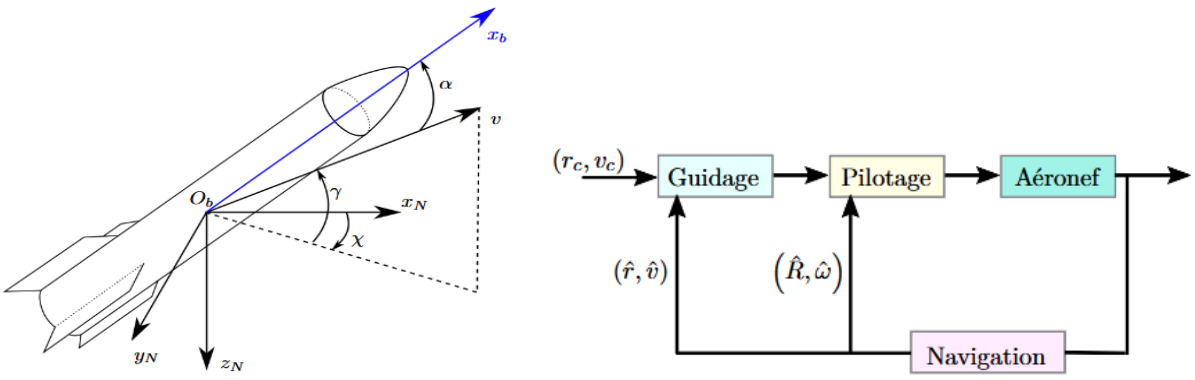


Figure I.3 : Système de guidage aérospatial. [13]

I.3 Exemple de régulation dans l'industrie à base d'API

I.3.1 Station de remplissage et de bouchonnage

Nous voyons ce système (Figure I.4) dans toutes les industries dans lesquels il y'a des flacons ou bouteilles à remplir d'un liquide comme les usines pour boisson (L'Exquise, Amazone, Belkaid et plein d'autres), usine de fabrication d'Huile que ce soit alimentaires ou non.

Pour ce système, nous avons besoin d'organe de commande principal qui est l'Automate Programmable Industriel. Sans lui, nous ne pouvons pas synchroniser les autres organes pour rendre ce système complètement automatisé.

Pour mieux concrétiser ce type de système, nous avons pris des photos du système réel qui se trouve au niveau du laboratoire de productique (MELT) à la faculté de Technologie, Université de Tlemcen (Figures I.4, I.5 et I.6).



Figure I.4 : Station de distribution de bouteilles.

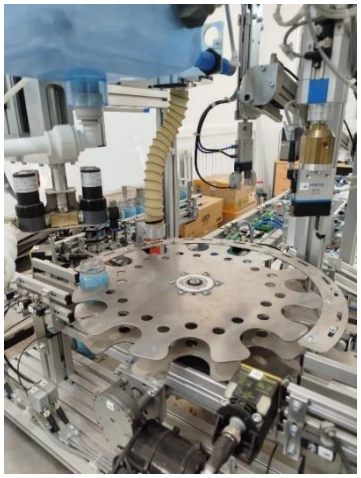


Figure I.5 : Station de remplissage.



Figure I.6 : Station de bouchonnage.

Pour faire fonctionner ce système, il a besoin :

- Un moteur pour faire tourner le tapis roulant qui sert à déplacer les bouteilles d'une opération à une autre.
- Un compresseur pour actionner les différents effecteurs.
- Des vérins pour bloquer les bouteilles lors du remplissage et le bouchonnage.
- Des capteurs qui détectent la présence des bouteilles aux différents endroits.
- Electrovanne pour contrôler le remplissage des bouteilles.

Dans l'industrie, ce système est rarement utilisé seul, il est accompagné d'un système automatisé de stockage/déstockages des bouteilles.

I.3.2 Système de stockage et déstockage

C'est un système automatisé qui est très utilisée dans l'industrie de par son gain de temps vu que le système automatisé est plus rapide par rapport à un stockage manuel par des opérateurs ce qui augmente le nombre de production journaliers.

Pour mieux visualiser ce système, nous avons pris des photos (Figures I.7, I.8, I.9 et I.10) dans le même laboratoire de productique (MELT).



Figure I.7 : Système de stockage / déstockage

Pour commander ce système, il faut jouer sur 3 axes motorisés :

- Axe 1 : Comporte un moteur qui sert à déplacer le chariot sur un plan horizontal.
- Axe 2 : Le moteur déplace verticalement le bras télescopique (Navigation entre les étages de stockage)

• Axe 3 : le troisième moteur sert à manipuler le bras télescopique pour déposer où prendre les différents produits au niveau des cases de stockage.



Figure I.8 : Axe de déplacement horizontal.



Figure I.9 : Axe de déplacement vertical.



Figure I.10 : Axe pour manipuler le bras télescopique.

Pour automatiser ce système, l'élément principal est l'automate programmable industriel API qui est le cerveau de notre système. A travers les différentes données d'entrées fournies ou transmises par les capteurs, il commande les différents actionneurs. Sans ces capteurs, notre système travaille à l'aveuglette. Ces capteurs donnent exactement la position du chariot au niveau des 03 axes. Ainsi, l'API prend soin de positionner l'objet dans l'emplacement de stockage désiré.

Tous ces systèmes décrits précédemment applique le principe de la régulation en général et particulièrement la régulation de position.

I.4 Principe de régulation

La régulation est un principe qui a pour but de garder une situation stable face à toutes les variations de l'environnement. Pour atteindre la stabilité optimale, il faut une observation continue pour qu'on puisse ajuster et adapter notre système dès qu'une perturbation ou un changement de situation apparaît.

Mais pour bien comprendre ce principe, nous allons prendre l'exemple d'une voiture qui doit s'arrêter à une plaque de stop, comme le montre la Figure I.11. Pour arrêter la voiture au point d'arrêt, le conducteur doit gérer la pédale d'accélérateur et celle du frein de manière que la voiture s'arrête sans à-coup et de manière douce et précise.



Figure I.11 : Exemple de régulation précise. [14]

Après avoir donné différents exemples de la régulation dans différents domaines et après avoir donné aussi une description brève de la régulation en générale, le paragraphe suivant est consacré à la description superficielle de notre prototype qui est réalisé au laboratoire de recherche MELT.

I.5 Description de notre prototype

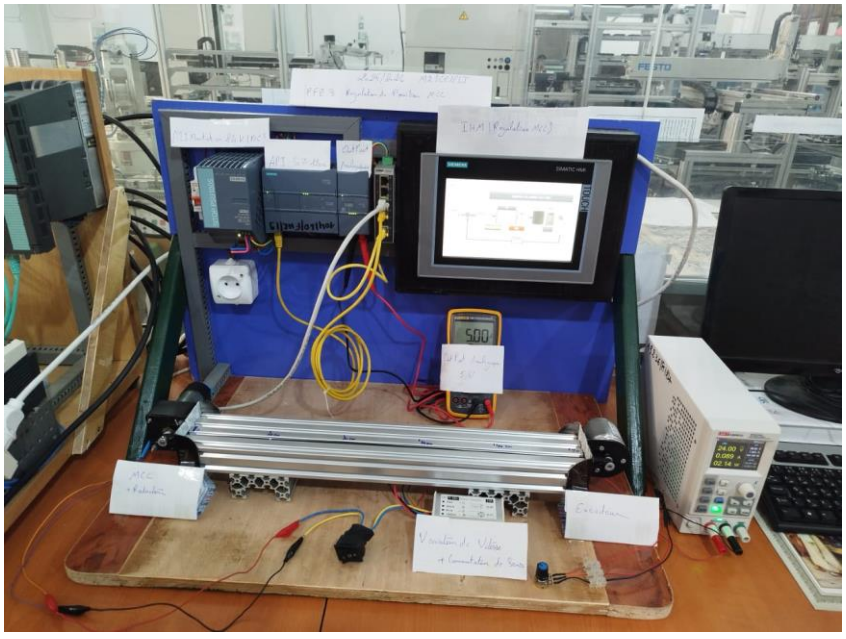


Figure I.12 : Photo du prototype.

Pour présenter de manière rapide notre prototype, nous pouvons le diviser en quatre parties, comme suit :

I.5.1 Organe de commande (API) et alimentation

Nous avons pris une alimentation DC 24V/10A de chez Siemens avec la référence 6EP1334-2BA20.

Pout notre automate programmable industriel (API) qui est notre organe de commande, nous avons choisi le S7-1200 1214C (Figure I.13) au lieu d'un S7-300 surtout pour être dans l'actualité vu que c'est une nouvelle technologie et surtout son coût qui défie toute compétitivité (moins cher) comparé à celui de son grand frère le S7-300.



Figure I.13 : API S7-1200. [15]

I.5.2 Pré actionneur et actionneur

Notre pré-actionneur (Figure I.14) est un variateur de vitesse qui se trouve chez les fournisseurs d'électronique comme DZduino à un prix imbattable et d'une performance impeccable. Ce variateur reçoit un signal PWM qui commande notre moteur.



Figure I.14 : Variateur de vitesse.

Notre actionneur est un moteur à courant continue à aimant permanent (Figure I.15) muni d'un réducteur mécanique (Figure I.16). Cet actionneur a été récupéré d'un ancien matériel défectueux. Il a besoin d'une tension d'alimentation de 24V, ce qui nous arrange convenablement.

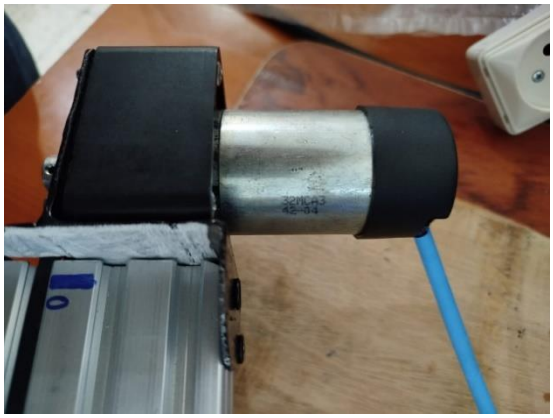


Figure I.15 : MCC à aimant permanent.

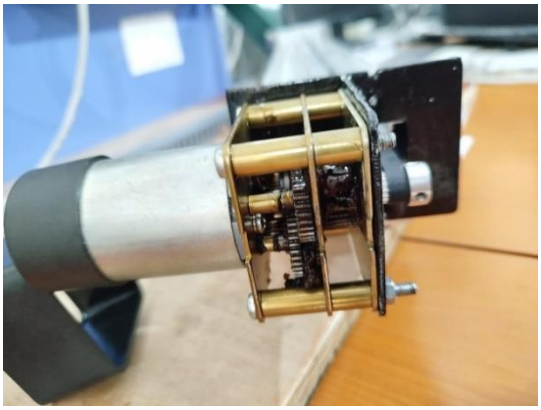


Figure I.16 : Réducteur mécanique.

I.5.3 Capteur

Nous avons utilisé, pour déterminer la position exacte de notre convoyeur, un encodeur incrémental (Figure I.17) comme capteur de retour ayant une résolution de 1024 impulsion.



Figure I.17: Encoder incremental. [16]

Pour améliorer notre système et augmenter ces performances, nous avons intégré et développer un système de supervision qui sera décrit dans le paragraphe suivant.

I.5.4 Supervision (IHM)

Pour superviser et commander système, nous avons utilisé (Figure I.18) le SIMATIC Comfort Panel Siemens TP900 Comfort de chez Siemens.

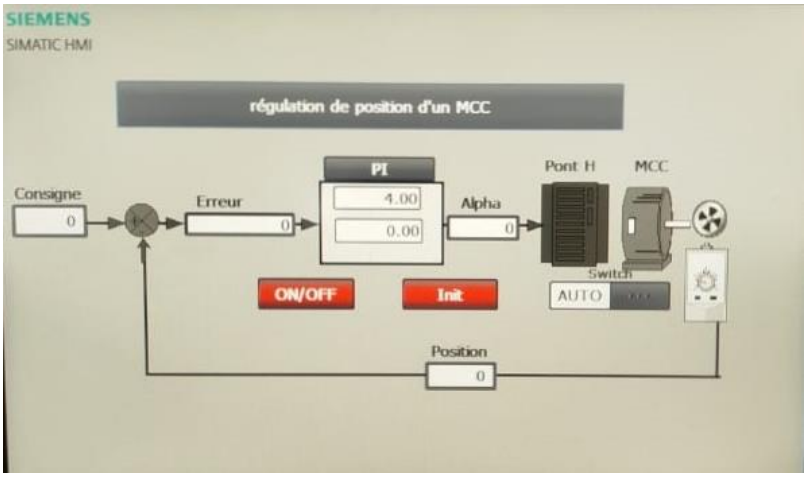


Figure I.18 : Interface homme machine.

I.6 Conclusion

Ce chapitre a permis de présenter le contexte général du projet et de montrer les éléments essentiels du système automatiser étudier. A travers les différents exemples donnés, nous avons mis en évidence l'importance de la régulation de position pour les systèmes automatise. La description du prototype nous a permis d'identifier les différents composants qui intervienne dans la régulation. A partir des explications et la présentation du prototype nous allons aborder dans les chapitres suivant toutes les étapes pour réaliser la régulation de position de notre prototype.

Chapitre II

Principe de la régulation industrielle

II.1 Introduction

Pour les systèmes industriels modernes, la régulation de position occupe une place essentielle dans toutes les applications qui nécessitent précision, rapidité et la stabilité comme les chaines de production, la robotique et les systèmes automatisés. Afin d'assurer le contrôle précis des déplacements mécaniques. Pour ça, l'utilisation d'une structure de commande associée, à des composants capables de convertir, transmettre, mesurer le mouvement et de contrôler, est primordial.

Ce chapitre présente une étude des différents éléments constituant la chaine de régulation de position pour le système étudié. Il débute par le principe de la régulation de position et le processus à commander. Ensuite la modélisation du moteur à courant continue à aiment permanent et le réducteur mécanique à travers leurs fonctions de transfert.

Par la suite, les composants de commande et de mesure sont abordés, comme le pont en H utiliser pour la variation de vitesses et le changement de sens de rotation du moteur, le capteur de déplacement qui permet la mesure de la position.

Enfin, ce chapitre traite du régulateur PID et ces différentes actions qui le compose. Le bloc PID_Compact qui est également étudié afin de préparer la mise en œuvre pratique de la régulation de position.

II.2 Régulation de position

La régulation de position repose sur le principe d'arrêt du système ou processus à une position souhaitée (consigne). Cependant, le système doit atteindre la position demandée sans erreur ou dépassement avec un minimum de temps.

Pour la régulation, nous utilisons une boucle fermée (Figure II.1). Une consigne est comparée avec la position donnée par le capteur de déplacement. La sortie du comparateur est connectée en cascade à un régulateur comme un PID, mode glissant ou logique floue. La sortie de ce régulateur nous donne un signal qui va commander notre système.

Le régulateur continue à fournir le signal de commande tant que la comparaison entre la consigne et la valeur du capteur n'est pas nulle (égale à zéro).

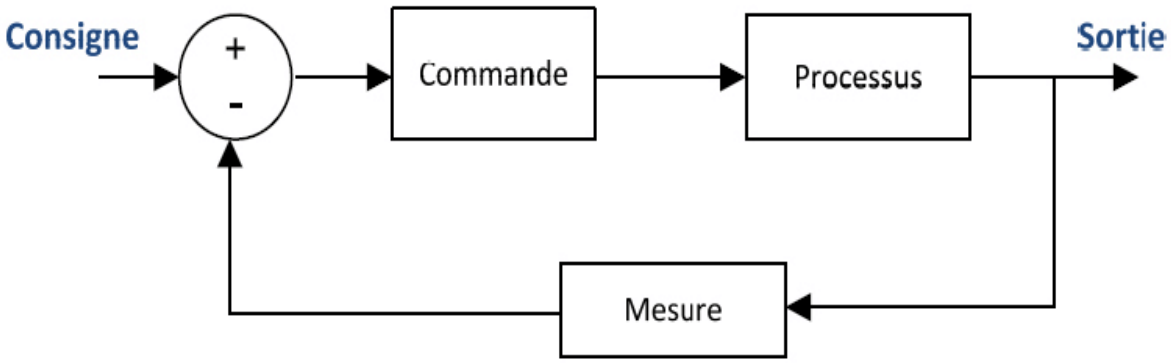


Figure II.1 : Schéma bloc pour une régulation de position. [17]

Notre processus qui est un moteur passe par trois phases (Figure II.2) :

- L'accélération jusqu'à atteindre la vitesse max lorsque la commande délivrée par le régulateur est max ;
- La vitesse reste constante tant que la commande est constante ;
- En fur et à mesure que la commande diminue (le système est très proche de la position désirée, la vitesse diminue (zone de décélération) jusqu'à l'arrêt total lorsque l'erreur est nulle, donc commande nulle.

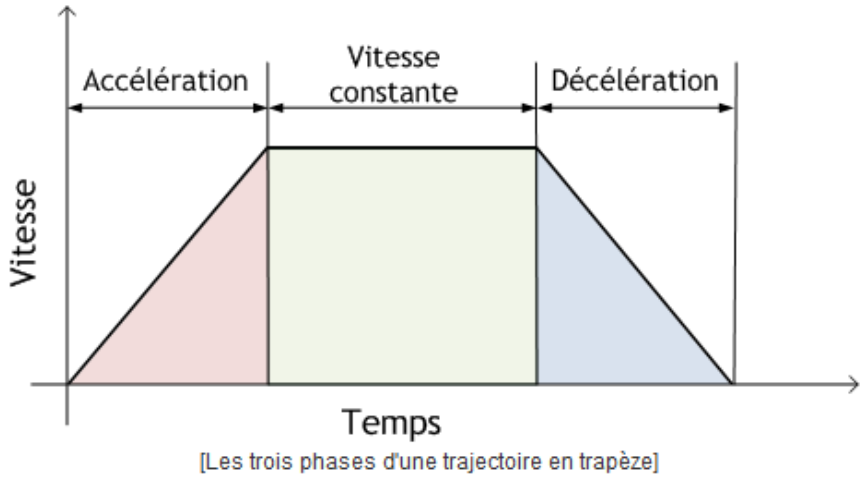


Figure II.2 : Phases de variation de vitesse du moteur. [18]

II.3 Processus et fonction de transfert

II.3.1 Modélisation du MCC à aimant permanent

Ce type de moteur n'a pas des enroulements dans le stator mais un aimant qui produit un champ magnétique (Figure II.3), c'est ce qui assure un champ magnétique constant.

Il est très utilisé en robotique vu qu'il est simple à utiliser et surtout il risque de moins chauffer vu que qu'il n'a pas de perte par effet Joule au niveau du stator.

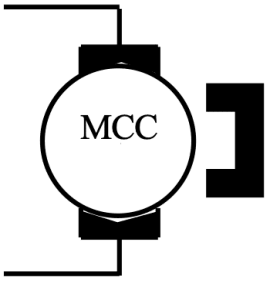


Figure II.3 : Représentation schématique d'un MCC à aimant permanent. [19]

Le moteur MCC comporte 02 parties distinctes : la partie électrique et la partie mécanique. Par conséquent, chaque partie est décrite par un modèle électrique équivalent.

II.3.1.1 Modèle électrique

1) Equation électrique : qui est une équation de la tension d'alimentation en relation le courant d'induit *ia* et la f.c.e.m *e* :

$$U(t) = Ra * ia(t) + La\frac{dia}{dt}(t) + e(t) \quad (2.1)$$

2) Force électromotrice : est une fonction qui relie la partie électrique à la partie mécanique.

$$e(t) = K_e * \Omega(t) \quad (2.2)$$

3) Equation électromagnétique : est aussi une équation reliant le couple (partie mécanique) au courant.

$$Cem(t) = K_m * i_a(t) \quad (2.3)$$

4) Equation mécanique : c'est l'équation la plus importante dans la modélisation du moteur MCC qui décrit la transformation d'énergie électrique en mécanique.

$$J\frac{d\Omega(t)}{dt} = Cem(t) - f.\Omega(t) - Cr(t) \quad (2.4)$$

$$J\frac{d\Omega(t)}{dt} + f.\Omega(t) = Km * ia(t) \quad (2.5)$$

Pour faciliter l'étude de stabilité et le régime transitoire, nous avons l'habitude de passer à la transformation de Laplace.

Après transformer de Laplace, l'eqt (2.1) sera comme suit :

$$U(p) = Ra * Ia(p) + La * p * Ia(p) + Ke * \Omega(p) \quad (2.6)$$

Aussi, la transformée de Laplace de l'équation (2.5) est comme suit :

$$(J * p + f) * \Omega(p) = Km * Ia(p) \quad (2.7)$$

A partir de l'équation (2.7), nous obtenons :

$$Ia(p) = \frac{(J*p + f) * \Omega(p)}{K_m} \quad (2.8)$$

En remplaçant l'eqt (2.8) dans l'eqt (2.6), l'équation suivante donne la fonction de tranfert générale qui nous intéresse pour notre régulation:

$$U(p) = R_a \frac{(J*p + f) * \Omega(p)}{K_m} + L_a * \frac{(J*p + f) * \Omega(p)}{K_m} + Ke * \Omega(p) \quad (2.9)$$

$$U(p) = \left(\frac{(Ra + La) * (J*p + f)}{K_m} + K_e\right) * \Omega(p) \quad (2.10)$$

$$U(p) = \left(\frac{La*J*p^2 + (Ra*J + La*f)*p + Ra*f}{K_m} + K_e\right) * \Omega(p) \quad (2.11)$$

$$\frac{\Omega(p)}{U(p)} = \frac{K_m}{La*J*p^2 + (Ra*J + La*f)*p + (Ra*f + K_e*K_m)} \quad (2.12)$$

Plusieurs paramètres importants sont tirés de l'eqt 2.12), comme :

1) Gain statique

$$K = \frac{K_m}{Ra*f + K_e*K_m} \quad (2.13)$$

2) Constante de temps électrique

$$T_e \frac{La}{Ra} \quad (2.14)$$

3) Constante de temps électromécanique :

$$T = \frac{Ra*J}{Ra*f + K_e*K_m} \quad (2.15)$$

4) Coefficient γ

$$\gamma = \frac{Ra*f}{Ra*f + K_e*K_m} \quad (2.16)$$

5) Après remplacement des constantes de temps et α dans l'équation (2.10), nous obtenons l'expression suivante :

$$\frac{\Omega(\mathrm{p})}{\mathrm{U(p)}} = \frac{\mathrm{K}}{T*Te*\,p^2 +(T+\,\gamma*Te)*p+1} \tag{2.17}$$

6) En général, la constante de temps électrique Te << T. Pour cela, nous pouvons supposer que l'action de Te sur notre système sera négligeable par rapport à la constante de temps mécanique. Par conséquent, notre fonction de transfert de notre système peut être approximée à :

$$\frac{\Omega(p)}{U(p)} = \frac{K}{T*p+1} \tag{2.18}$$

7) La position angulaire est obtenue en intégrant la vitesse.

$$\Theta(p) = \frac{1}{p} * \Omega(p)\ \frac{K}{T*p+1} \tag{2.19}$$

8) A partir de l'eqt (2.18), la fonction de transfert obtenue est comme suit :

$$\frac{\theta(p)}{U(p)} = \frac{K}{p*(1+T*p)} \tag{2.20}$$

C'est un système de second ordre, mais ce qu'il ne faut pas oublier est que l'arbre de sortie du moteur est relié à un réducteur de vitesse mécanique.

II.3.2 Réducteur

Le réducteur est un système d'engrenage mécanique qui se place à l'arbre de sortie d'un moteur. Son rôle principal est de réduire la vitesse de l'arbre, en contrepartie, il multiplie le couple du moteur suivant le coefficient de réduction. C'est l'avantage majeur d'un réducteur.

Si le rapport de transmission R_{ed}<1, il joue le rôle de réducteur. Dans le cas contraire, il aura le rôle d'un multiplicateur de vitesse.

$$R_{ed=}\ \frac{\Omega\ de\ sortie}{\Omega\ d'entrée} \tag{2.21}$$

Chaque réducteur possède un coefficient de réduction R_{ed}.

Son utilisation est indispensable parce que ces petits moteurs tournent à une grande vitesse mais il n'a pas assez de couple pour déplacer un convoyeur transportant un poids.

Par conséquent, la fonction de transfert du processus devient :

$$\Omega r(p) = \frac{1}{Red} * \Omega(p) \quad (2.22)$$

$$\frac{\theta r(p)}{U(p)} = \frac{K}{Red * p * (1+T*p)} \quad (2.23)$$

II.4 Pont en H

Le pont en H (Figure II.4) est un montage d'électronique de puissance utilisé comme variateur de vitesse et commander aussi son sens de rotation. Son nom vient du fait que les interrupteurs électroniques sont branchés en forme de H.

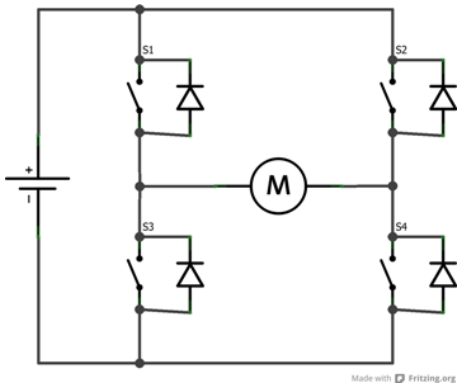


Figure II.4 : Schéma de montage d'un pont en H. [20]

Pour que le moteur tourne dans la sens directe, nous devons fermer S1 et S4. Cela va permettre au courant de traverser le moteur dans un premier sens. Ainsi, pour changer de sens, il suffit de fermer S2 et S3 au lieu de S1 et S4.

Le pont en H sert aussi de variateur de vitesse en variant la tension moyenne appliquer aux moteur grâce au rapport cyclique α de la PWM (Pulse with modulation) (Figure II.5).

Le signal PWM est le signal de commande généré par l'API. Le rôle du régulateur qui est l'objet technologique PID_Compact intégré dans notre API ayant pour but de nous fournir la commande qui est la valeur en temps réel du rapport cyclique 'α' qui varie entre 0-100%.

La tension moyenne appliquer au moteur est : $U_{moy} = \alpha * U_{alimentation}$

Donc quand α = 0% le moteur est à l'arrêt et quand α = 100% le moteur tourne à une vitesse maximale.

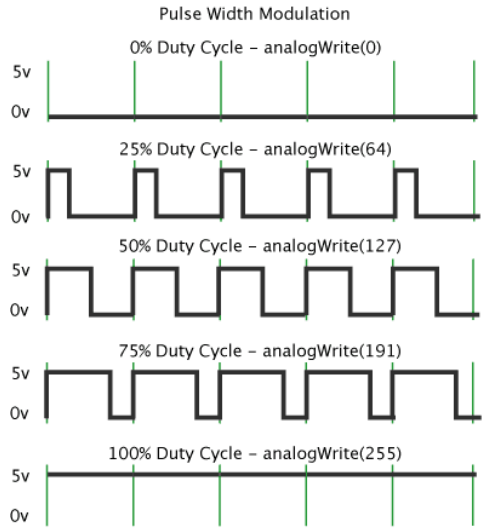


Figure II.5 : Signal PWM avec variation du rapport cyclique. [21]

II.5 Capteur de déplacement

L'encodeur est un capteur numérique. Sans lui, nous ne pouvons pas avoir une bonne régulation parce que c'est l'œil de notre système. Il est placé au bout de l'arbre du moteur qui nous donnera la position réelle qui va être comparée avec notre consigne. En d'autres termes, sans lui, nous ne pouvons rien faire.

Pour obtenir la position réelle, il faut tout d'abord connaitre le diamètre de l'arbre de l'encodeur pour que nous puissions tirer la longueur par impulsion pour en fin calculer la distance parcourue ou la position réelle.

Pour avoir une meilleure précision de l'encodeur et de savoir dans quel sens le moteur tourne, nous utilisons le mode X4 (quadruple). Ce mode consiste à surveiller les changements d'état des deux voies A et B.

Nous obtenons une valeur positive lorsque le moteur tourne dans le sens direct et une valeur négative dans le sens inverse de rotation.

Important : il est conseillé d'utiliser le mode X4 pour avoir une meilleure précision parce qu'en mode X1 nous avons 1024 points de mesure par tour. Par contre, dans le mode X4, nous avons 1024X4 se qui nous donne 4096 points de mesure par tour, d'où un pas de mesure de 0.35° à 0.08° qui est quatre fois précis.

II.5.1 Détermination de la distance parcourue

Nous avons un encodeur avec une résolution de Résolution = 1024 pts. Connaissons le diamètre 'Dpt' du pignon de transmission de la courroie, la longueur par impulsion est calculée comme suit :

*Longueur par impulsion = Diamètre * π */ Résolution.*

A partir de la longueur par impulsion et connaissons le nombre d'impulsion effectuée, nous calculons où nous déterminons la distance parcourue :

*Distance parcourue = Longueur par impulsion * Nombre d'impulsion effectuée*

II.5.2 Application numérique pour notre encodeur

Dpt = 10.0mm, Longueur par impulsion = 10* π /1024 = 0.0306 mm

*Distance parcourue = 0.0306 * Nombre d'impulsion effectuée*

Après avoir obtenu la position réelle, il reste à appliquer notre PID pour obtenir la commande adéquate suivant la distance parcourue.

II.6 Régulateur PID

Le régulateur PID est une commande de correction la plus connue et la plus utilisée en automatique ; son rôle principal est d'atteindre la consigne souhaitée. Pour y arriver, le régulateur adapte en permanence la commande (sortie du PID) suivant l'erreur entre la valeur du capteur et la consigne.

C'est cette erreur qui nous génère un signal de commande PWM (avec un rapport cyclique adapté) qui fait varier la tension d'alimentation. Pour qu'elle soit bien réglée, nous avons besoin de choisir les paramètres clés de notre régulateur qui sont :

II.6.1 Action proportionnelle

C'est une action similaire à l'accélérateur d'une voiture parce que là l'erreur est multipliée par un gain Kp choisi.

Il permet d'atteindre la consigne plus rapidement surtout quand l'erreur est grande.

II.6.2 Action intégrale

Nous pouvons assimiler Ki comme un paramètre de finition. Son rôle est de mettre à jour la valeur de la vitesse même s'il y a un infime écart en augmentant ou diminuant

la commande progressivement pour atteindre la consigne. La précision est obtenue après un choix optimal de Ki. Dans le cas contraire, il provoquera des dépassements ce qui n'est pas tolérable.

II.6.3 Action dérivée

C'est l'équivalent des freins d'une voiture, il ralentit le système quand il se rapproche trop vite de la consigne pour éviter le dépassement du système.

L'inconvénient de cette action est de provoquer l'amplification des bruits de mesure haute fréquence (les parasites).

En associant les 03 actions, nous obtenons le fameux régulateur PID, comme le montre la Figure II.6.

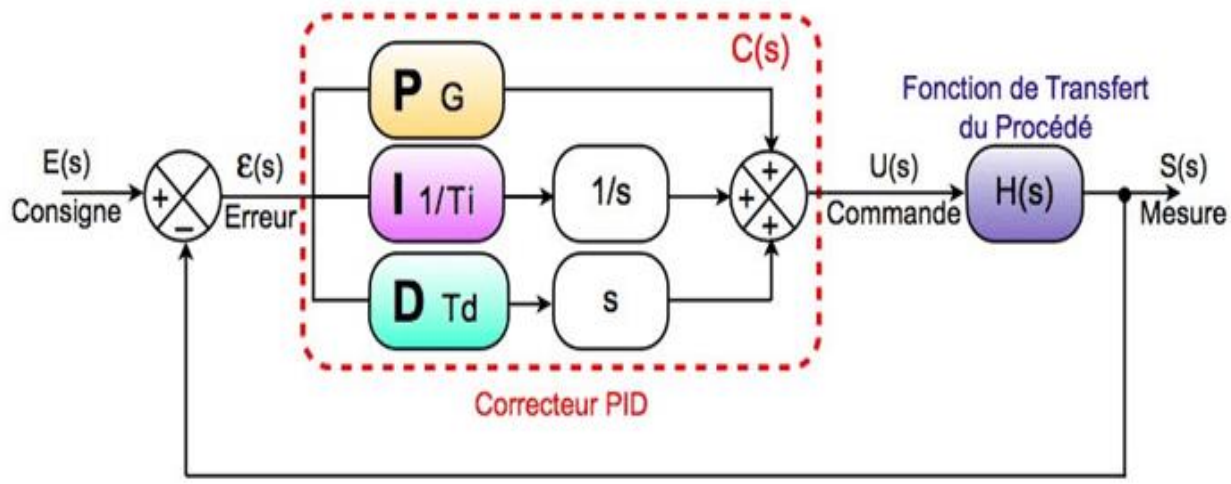


Figure II.6 : Schéma fonctionnel d'un régulateur PID. [22]

Nous avons décrit le régulateur PID en général. Dans ce qui suit, nous allons s'approfondir dans le PID_Compact intégré dans les API.

II.6.4 PID_Compact

Le PID_Compact est un bloc ou objet technologique intégré dans les API comme notre S7-1200 utilisé pour réguler notre prototype en boucle fermée.

Ce qui est avantageux avec le PID_Compact c'est que nous n'avons pas besoin de le coder juste à le configurer.

Pour mieux connaitre son fonctionnement interne, la datasheet [12] donne le schéma fonctionnel suivant, comme le montre la Figure II.7.

:

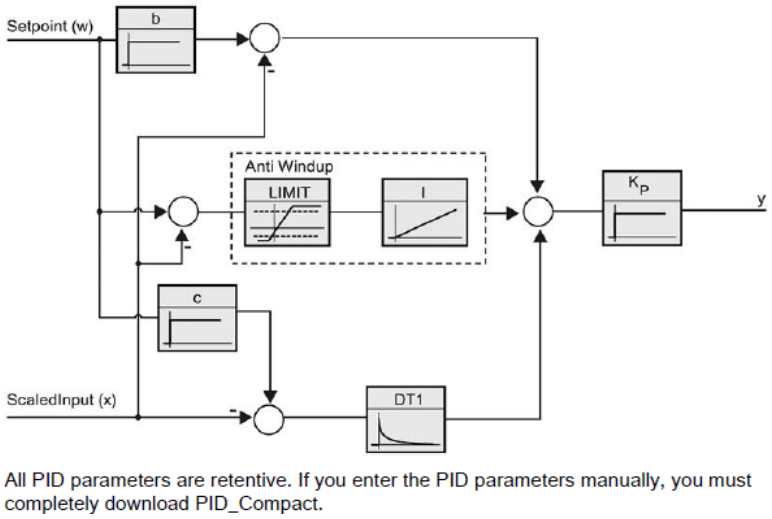


Figure II.7 : Schéma fonctionnel du PID_Compact. [23]

Le PID reçoit 02 entrées, Setpoint *w(t)* qui est la consigne et Input *x(t)* qui est la mesure du capteur et fournit une sortie qui est la commande qui peut être un signal PWM ou une tension (Output). Ce PID comporte lui aussi de 03 branches.

1) Branche proportionnelle (P)

$$P = Kp * (b * w(t) - x(t)) \quad (2.24)$$

Avec : Kp qui est le gain proportionnel et b la pondération de la consigne.

Le b sert principalement à adoucir la réaction du régulateur face au changement brutal de consigne et réduire le dépassement. Ce coefficient a une plage de variation : 0 < b < 1. Plus nous nous rapprochons du 1, plus la réponse est rapide avec plus de dépassement. Le contraire est obtenu au rapprochement de 0, c-à-d, la réponse devient lente avec moins de dépassement.

2) Branche intégrale (I)

$$I(t) = Kp * \frac{1}{Ti} \int (w - x) dt \quad (2.25)$$

L'intégrateur accumule l'erreur et supprime l'erreur statique.

Le choix de Ti est très sensible car si Ti est grand, cela va nous donner une correction trop lente. Par contre, si Ti est petit, l'intégration est rapide, ce qui nous engendrer des oscillations et un dépassement.

Cette branche travaille avec le temps d'action intégrale (Ti) et non pas avec l'intensité de l'action intégrale (Ki).

Pour rappel, l'action intégrale additionne l'erreur tant qu'elle n'est pas nulle, et cela peut causer des problèmes, par exemple le signal PWM est limité entre 0% et 100% mais l'intégrateur continue d'accumuler l'erreur même si la sortie ne peut plus augmenter ce qui va provoquer par la suite d'important dépassement et une réponse lente du système. Pour cela, le PID_compact de Siemens intègre le bloc Anti-Windup qui bloque ou limite l'action intégrale dès que le système se sature pour empêcher l'accumulation excessive de l'erreur. Par conséquent, cela améliore la stabilité et réduit le dépassement avec une réponse rapide du système.

L'activation de l'Anti-windup est automatique, il faut juste définir Output Upper Limit et Output Lower Limit pendant la configuration du PID_Compact et lui s'active quand les limites sont atteintes.

3) Branche dérivée (D)

$$D(p) = Kp * Td * \frac{d(c*w - x)}{dt} \quad (2.26)$$

Td c'est la constante du filtre passe bas et c'est la pondération sur la dérivée.

Le PID_Compact utilise le D_{T1} ce qui signifie action dérivée + filtre passe bas pour diminuer les parasites.

La branche dérivée sert à anticiper les variations et améliore la stabilité.

La pondération dérivée c sert à ne pas créer un pic de commande parce que si la consigne change brutalement la dérivée de la consigne devient énorme. Pour cette raison, la plupart du temps c est nulle.

L'addition des trois branches de commande nous fournit le signal de commande.

$$Y(t) = Kp * ((b * w - x) + \frac{1}{Ti} * \int (w - x)dt + Td * \frac{d(c*w - x)}{dt}) \quad (2.27)$$

Sachant que w est notre consigne qui peut être exprimée par : $w = \theta r^* * \frac{Dpt}{2}$, ainsi que $x = \theta r * \frac{Dpt}{2}$, dans le domaine de Laplace, l'eqt (2.27) devient :

$$Y(p) = Kp * (1 + \frac{1}{Ti\, p} + Td\, p) * (\theta r^* - \theta r) \quad (2.28)$$

Pour alléger l'expression, le terme $\frac{Dpt}{2}$ est introduit dans la valeur de Kp.

La fonction de transfert du MCC + Réducteur est donnée par l'équation suivante :

$$\frac{\theta r(p)}{U(p)} = \frac{K}{Red * p *(1+T*p)} \tag{2.29}$$

Le régulateur est en cascade avec notre processus. Réellement, la sortie du PID n'agit pas directement sur notre processus qui est le MCC + Réducteur mais agit sur notre pré-actionneur qui est un convertisseur buck. Sachant qu'on travaille avec une commande PWM de haute fréquence, la fonction de transfert de notre pré-actionneur est unitaire. Par conséquent, d'après les équations, l'équation du système devient :

$$U(p) = Y(p) \tag{2.30}$$

Avec

$$Y(p) = C(P) * e(p) \tag{2.31}$$

Donc

$$\theta r(p) = \frac{K}{Red * p *(1+T*p)} * Y(p) \tag{2.32}$$

$$C(p) = Kp * (1 + \frac{1}{Tip} + Tdp) \tag{2.33}$$

$$e(p) = \theta r^*(p) - \theta r(p) \tag{2.34}$$

En remplaçant l'eqt (2.28) dans l'eqt (2.32), nous obtenons :

$$\theta r(p) = \frac{K}{Red * p *(1+T*p)} * Kp * (1 + \frac{1}{Tip} + Tdp) * (\theta r^*(\mathrm{p}) - \theta r(p)) \tag{2.35}$$

L'équation du système devient :

$$\frac{\theta r(p)}{\theta r^*(p)} = Kp * (1 + \frac{1}{Tip} + T_{dp}) * \frac{K}{Red * p *(1+T*p)} \tag{2.36}$$

Après développement de l'eqt 2.36, nous obtenons l'équation suivante :

$$1 + \frac{1}{Tip} + T_{dp} = \frac{Ti*Td* p^2 + Ti* p +1}{Ti*p} \tag{2.37}$$

Maintenant, il nous reste à remplacer (2.30) dans (2.29) ce qui va donner l'équation finale du système :

En remplaçant (2.37) dans (2.26), l'équation finale du système sera comme suit :

$$\frac{\theta r(p)}{\theta r^*(p)} = \frac{K_p*K*(Ti*Td* p^2 + Ti* p +1)}{R_{ed}*Ti* p^2*(1+T*p)} \tag{2.38}$$

Cette équation donne la relation de l'angle de rotation en fonction de la valeur de la consigne. Cette équation sera introduite dans notre API pour obtenir le but de notre PFE qui est la régulation de position.

II.7 Conclusion

Les principaux éléments nécessaires pour la réalisation d'un système de régulation de position ont été étudiés. La modélisation du moteur à courant continue à aimant permanent et aussi celle du réducteur mécanique ont permis d'établir les fonctions de transfert.

L'étude du pont en H a mis en lumière son rôle dans la commande du sens et la variation de vitesse du moteur, tandis que le capteur de déplacement nous a permis de comprendre le principe de mesure de la position à l'aide d'un encodeur.

Enfin, l'analyse des différentes actions du régulateur PID ainsi que le bloc PID_Compact a montré l'importance de la correction dans l'amélioration des performances du système, notamment en termes de précision, de rapidité et de stabilité. Cette étude nous a permis de construire une base essentielle pour réaliser notre banc d'essai pour la régulation de position dans le chapitre suivant.

Chapitre III

Conception, réalisation et validation de notre banc d’essai

III.1 Introduction

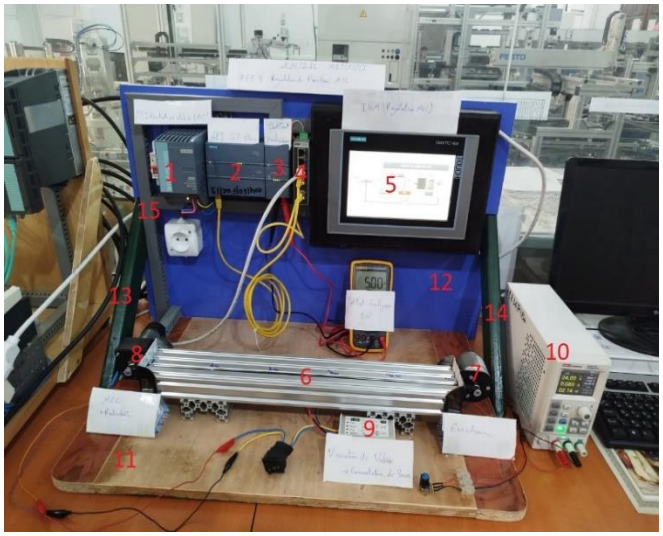
Dans les chapitres précédents, nous avons parlé des notions générales de la régulation de position et la modélisation du système étudié, notamment le moteur à courant continue, le pont en H et le régulateur PID pour commander la position. Après l'étude théorique du système, il est nécessaire de passer à la partie pratique pour valider les résultats obtenus.

Ce chapitre est dédié à la réalisation pratique du banc d'essai à base d'un automate programmable industriel (API) S7-1200. Il regroupe l'ensemble des éléments constituant le système étudier, que ce soit sur le plan mécanique qu'électrique, et aussi les étapes de configuration et de programmation sous l'environnement TIA Portal V16.

En premier, nous allons décrire les différentes parties du banc d'essai, incluant l'alimentation, l'organe de commande, le pré-actionneur, l'actionneur et le capteur pour la mesure de position. Ensuite, nous détaillerons la partie programmation, notamment la configuration du compteur rapide et le générateur PWM sans oublier la configuration du PID_Compact. En dernier, il y'aura la partie supervision dans la quel nous allons créer la vue de l'IHM pour contrôler et suivre l'évolution du système en temps réel.

Le chapitre sera clôturé par des tests expérimentaux et aussi l'interprétation des résultats obtenus afin d'évaluer les performances du banc d'essai.

III.2 Présentation de notre banc d'essai



1	Alimentation
2	Automate Programmable Industrie
3	Sortie Analogique
4	HUB Industriel
5	Interface Homme-Machine
6	Rail de guidage
7	Encodeur
8	MCC + Réducteur
9	Variateur
10	Alimentation externe
11, 12, 13 & 14	Supports en Bois
15	Goulottes

Figure III.1 : Photo du banc d'essai numéroté.

III.2.1 Partie mécanique

C'est la première partie réalisée pour la conception du banc. Il faut savoir que la précision de notre système repose sur l'efficacité de cette partie.

En premier lieu, nous avons découpé le rail de guidage linéaire pour CNC (N°6, Fugure III.1) , dans lequel le support du moteur et l'encodeur est fixé à son extrémité. La courroie est logée dans une des encoches du rail. Elle sera protégée et peut tourner librement.

Les deux supports (N° 7 & 8, Figure III.1) ont été conçus, soudés et réalisés dans l'atelier.

Après réalisations de la partie métallique, nous sommes passés à la découpe du bois. Pour le support, nous avons pris deux planches de bois qui ont été assemblés perpendiculairement ; nous les avons consolidés le tout avec deux traverses en bois (N°11, 12, 13 & 14, Figure III.1).

Par la suite, nous avons créé un cadre en bois comme support de notre IHM (N°5, Figure III.1). Nous avons déposé un rail de fixation pour les différents composants comme l'API, l'Alimentation, le module de sortie analogique et le HUB industriel (N° 1, 2, 3 & 4, Figure III.1).

Pour cacher les câbles de connexion, nous avons fixé des goulottes sur le support (N° 15, Figure III.1).

III.2.2 Partie électrique

III.2.2.1 Alimentation

L'alimentation utilisée est de chez Siemens qui nous fournit une tension continue de 24V stable, 10A (N°1, Figure III.1) qui servira pour alimenter : l'API, le module de sortie analogique, le HUB industriel et aussi notre variateur de vitesse.

III.2.2.2 Organe de commande

L'Automate Programmable Industriel (API) (N°2, Figure III.1) est un équipement électronique conçu pour assurer le traitement et la commande des informations dans les systèmes automatisés. Son fonctionnement est entièrement déterminé par un programme enregistré en mémoire, lui permettant d'exécuter différentes taches de contrôle et de supervision.

L'Automate Programmable Industriel S7-1200 est une gamme d'API compact conçu pour les applications d'automatisation de petite et moyenne taille. Il est très utilisé pour sa simplicité de programmation, sa modularité et surtout ses performances élevées. Il est surtout utilisé dans les systèmes de commande industriels moderne.

Dans le premier temps, L'API utilisé était un S7-1200 Avec la CPU 1214C DC/DC/RLY (RLY signifie sortie Relai) qui a été changé par la suite. La cause était purement technique. En premier temps, nous avons voulu commander la vitesse de notre moteur-réducteur avec le variateur acheté dans le commerce à un prix imbattable (Figure I.14). Malheureusement, lors de nos premiers tests, il s'est avéré qu'il ne pouvait pas supporter une tension variable fournie par notre module analogique (N°3, Figure III.1). Nous étions dans l'obligation d'utiliser l'Output_PWM de notre API qui est fournie uniquement par une CPU 1214C DC/DC/DC qui est dotée par des transistors rapides supportant des commutations à haute fréquence. Chose qui n'est pas évident pour les sorties relais (RLY).

La CPU 1214C DC/DC/DC représente une solution performante pour les systèmes de commande des moteurs, la régulation et la supervision industrielle.

La désignation DC/DC/DC signifie :

- Alimentation en courant continue (DC) : l'automate est alimenté en 24V continue.
- Entrée numérique en tension continue (DC) : les entrées TOR fonctionne en 24V continue.
- Sortie numérique en tension continue (DC) : les sorties sont de type transistor et délivrent des signaux en tension continue et signal PWM.

Cette CPU intègre plusieurs fonctions essentielles telles que :

- Des entrées et sorties numériques intégrées.
- Des entrées analogiques selon cette version.
- Des compteurs rapides (HSC) pour les applications des encodeurs.
- Des sorties PWM/PTO pour la commande des moteurs.
- Un port Ethernet/PROFINET permettant la communication et la programmation via le logiciel TIA Portal.

III.2.2.3 Pré actionneur

Les premiers tests ont été réalisés avec un variateur manuel de vitesse grâce au potentiomètre (Figure I.14), comme déjà décrit. Le système fonctionnait très bien. Nous avons aussi utilisé une alimentation stabilisée (N°10, Figure III.1) dotée par un système de limitation du courant pour alimenter uniquement le variateur et le moteur MCC. Cette procédure a été appliquée pour protéger notre matériel contre le retour du courant comme l'alimentation de Siemens, API, HUB qui sont très chers.

Lors du test final, pour modifier la vitesse automatiquement avec l'output PWM fourni par le PID_Compact, pour cette raison, nous avons réalisé notre propre variateur en assemblant des modules achetés dans le commerce de très bonne qualité et à des prix très raisonnables (riverse-engineering). C'est un point fort pour notre réalisation.

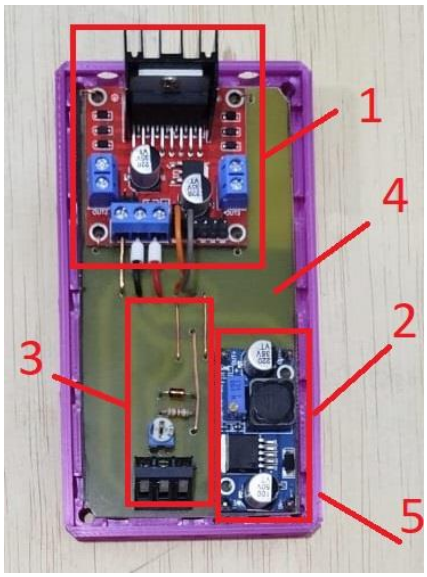


Figure III.2 : Variateur de vitesse réalisé.

Notre variateur de la Figure III.2 est composé de :

- Un hacheur Buck ''N° 2'' utilisé pour baisser la tension d'alimentation pour le driver L289N de 24V à 5V.
- Un driver L298N ''N° 1'' utilisé pour commander le moteur comme : le démarrage/arrêt et la variation de la vitesse à travers le signal PWM fourni par l'API.
- Un diviseur de tension ''N° 3'' pour réduite l'amplitude du signal PWM de 24V (signal de sortie de l'API) à une amplitude supportable par notre driver L298N qui est de 5V.

Le driver L298N est un double pont en H qui commande :

- Le sens de rotation du moteur.
- La variation de la vitesse.

Pour connecter les trois circuits (N° 4, Figure III.2), nous avons conçu notre propre circuit PCB sur ARES-Proteus, comme le montre la Figure III.3. Nous avons imprimé et percé le circuit dans le laboratoire de notre faculté. Après cela, nous avons soudé tous les composant de notre variateur en ajoutant les différents connecteurs.

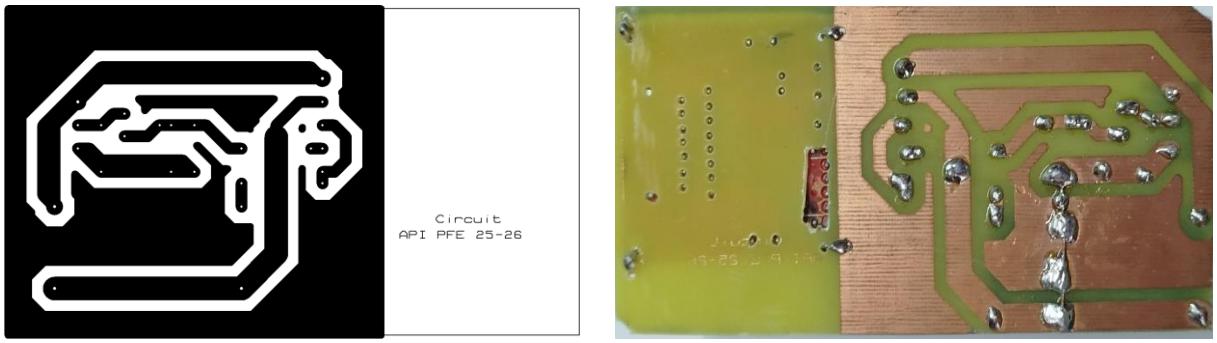


Figure III.3 ; Circuit PCB sur ARES-Proteus.

La connexion a été faite comme suit :

- Nous avons connecté en premier les deux câbles d'alimentation (+, -) et le câble de commande avec la sortie numérique de l'API Q 0.0.
- Pour connecter le moteur, nous avons deux choix : soit sur le out1 ou le out2 car le driver L298N peut commander deux moteurs en même temps.

Pour avoir une belle finition du variateur (N° 5, Figure III.2), nous avons conçu un boitier sur le logiciel « FreeCad » et imprimé en 3D par la suite par l'imprimante du laboratoire de recherche MELT. Le rendu final du variateur est dans la Figure III.4.



Figure III.4 : Boitier imprimé en 3D (Vues : Coté et derrière).

III.2.2.4 Actionneur

Comme actionneur, nous avons utilisé un moteur à courant continue à aimant permanent avec réducteur mécanique (N° 8, Figure III.1) récupéré dans le laboratoire pédagogique de notre faculté (sa modélisation est faite dans le chapitre II).

Ce qu'il faut savoir c'est que notre commande fonctionne avec la tension délivrée par la sortie du driver L298N. Tant que le driver délivre une tension, le moteur continue à tourner et s'arrête en coupant la tension aux bornes du moteur.

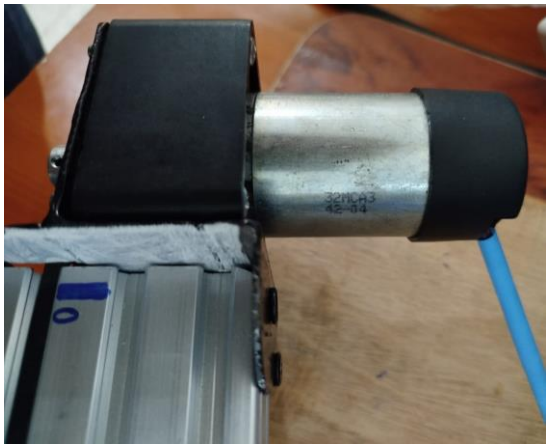


Figure III.5 : Moteur à courant continue + Réducteur.

III.2.2.5 Capteur

Le capteur est un encodeur incrémental IFM RU3500 (N° 7, Figure III.1), disponible dans le laboratoire de recherche MELT. Il est utilisé pour mesurer la position angulaire de l'arbre moteur.

Il génère des impulsions électriques proportionnelles au déplacement de l'axe. Le comptage de ces impulsions permet de déterminer précisément la position du système et assure le retour de l'information vers le comparateur d'erreur pour savoir si le moteur a atteint la bonne position.

Pour avoir plus d'information sur le fonctionnement de l'encodeur et surtout comment nous avons fait calculer la circonférence, la longueur par impulsion et comment faire pour calculer la distance en temps réel, il suffit de consulter le paragraphe

Après une description matérielle de notre prototype, nous passons maintenant à la partie programmation. C'est une étape importante où nous allons donner les différentes parties du programme en insérant différents éléments. Nous finirons cette partie par la description et le développement de notre IHM.

III.2.3 Partie programmation

Cette partie représente l'ensemble des opérations logicielles nécessaires au pilotage et à la commande du système de régulation de position.

Elle est réalisée sous l'environnement TIA Portal à l'aide de l'automate programmable Siemens S7-1200 1214C DC/DC/DC.

Ça comprend la configuration de la communication entre le PC et l'automate, le paramétrage du matériel utilisé, la configuration du compteur rapide pour l'acquisition des impulsions de l'encodeur, ainsi que la génération du signal PWM destiné à la commande du moteur à courant continue.

III.2.3.1 Configuration matériel

La configuration matérielle est pour paramétrer les différents éléments utilisés dans le système dans le logiciel TIA Portal. Ça permet de définir le matériel utiliser pour le banc d'essai.

Les étapes à suivre pour cette configuration sont :

1. Création du projet

Dans un premier lieu, nous avons créé notre projet sur TIA Portal (Figure III.6) dans laquelle nous avons donné un nom pour le projet et son chemin de sauvegarde.

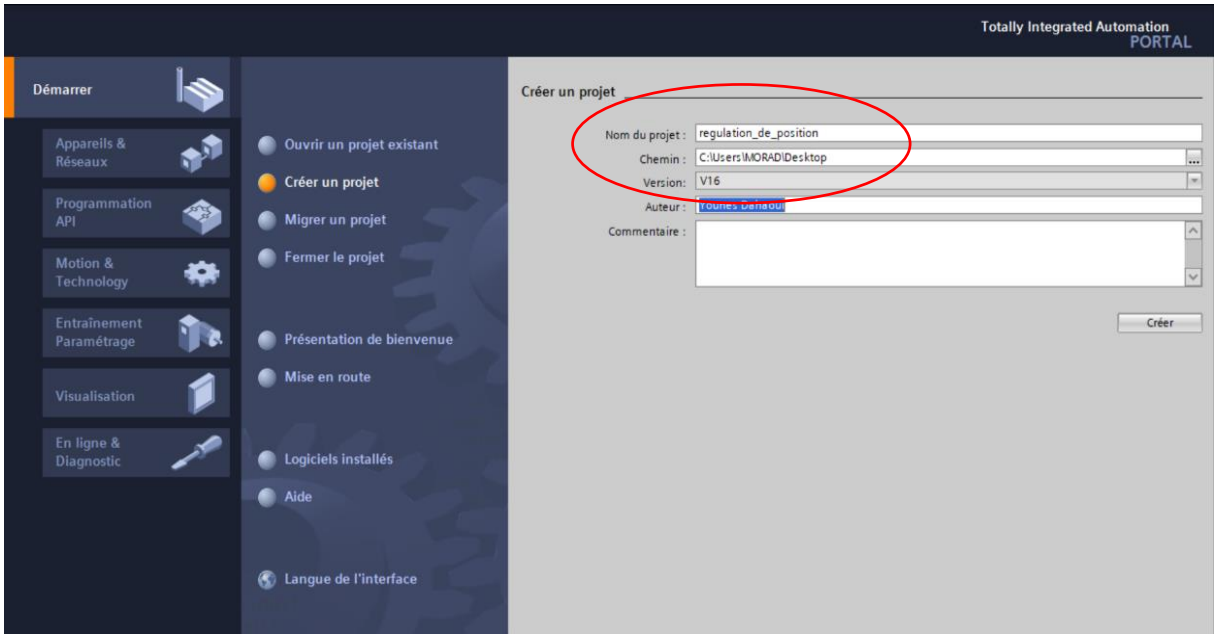


Figure III.6 : Création du projet sur TIA Portal.

2. Ajout de l'automate

Ensuite, nous avons passé dans la rubrique « Ajouter un appareil ». La Figure III.7 montre les différents CPU disponibles, ainsi que la CPU choisie pour notre projet.

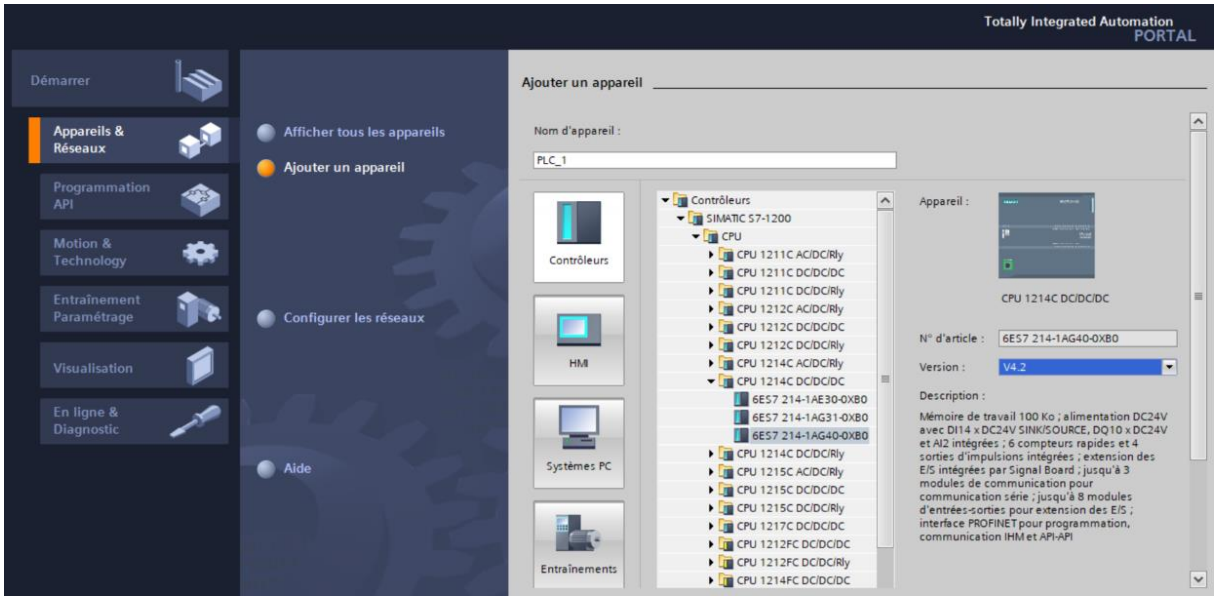


Figure III.7 : Ajout de la CPU utilisée.

3. Ajout du PC-System

La première étape était de développer et tester notre IHM dans notre PC-System avec le WinCC RT Advanced, la Figure III.8 explique comment ajouter un PC-System.

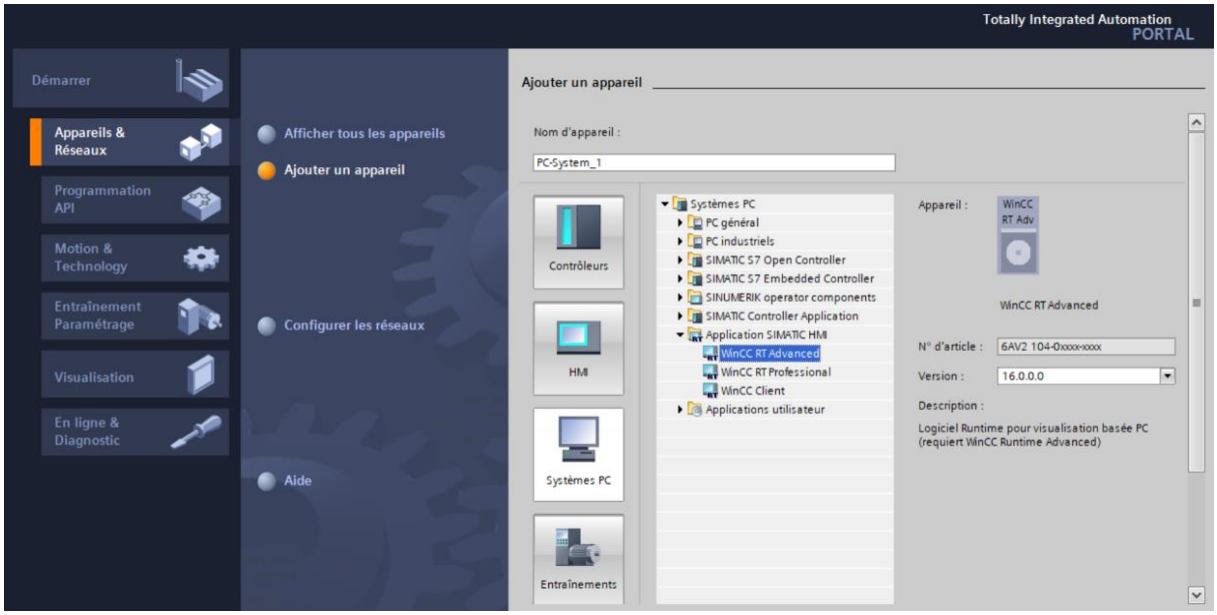


Figure III.8 : Ajout du PC Système avec WinCC RT Advanced.

4. Ajout de l'IHM TP900 Comfort

Pour rendre notre banc d'essai plus performant et transportable (pour la supervision et la commande sans passer par le biais de notre PC de programmation), nous avons ajouté un IHM TP900 Comfort. La Figure III.9 montre comment insérer ce type d'IHM dans notre projet.

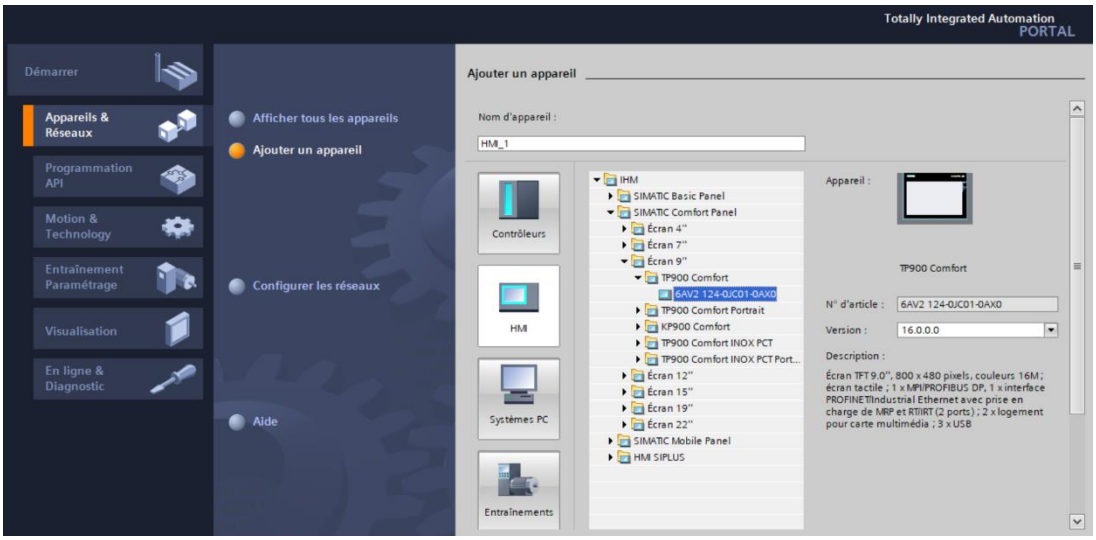


Figure III.9 : Ajout de l'IHM TP900 Comfort de Siemens pour la supervision et la commande.

5. Ajout du module de sortie analogique

Comme déjà expliqué dans le paragraphe III.2.2.2, nous étions dans l'obligation d'ajouter un module analogique. Ce dernier a été ajouté comme solution technologique pour fournir une tension de sortie variable.

La Figure III.10 montre la façon d'ajouter ce module.

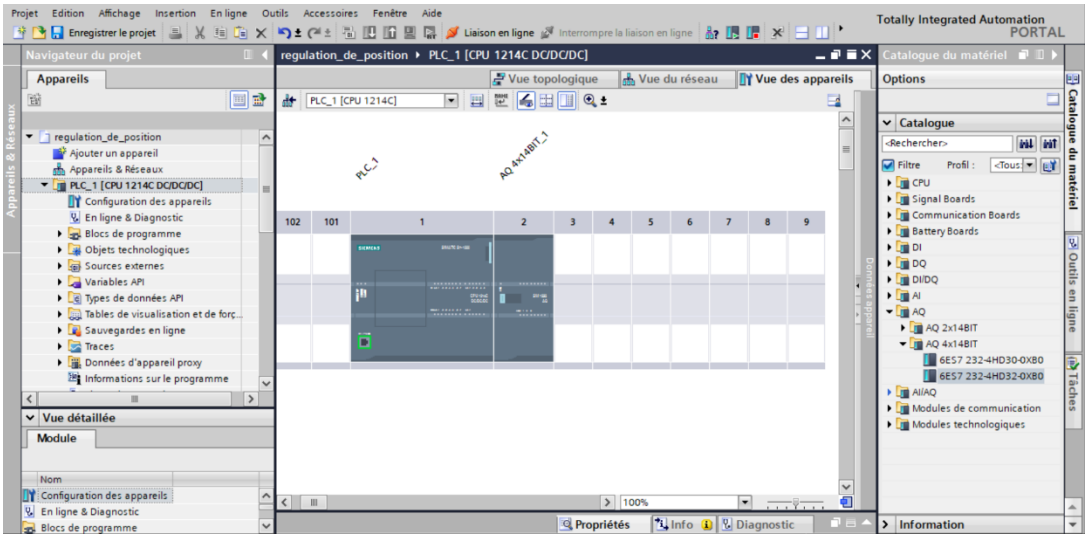


Figure III.10 : Ajout du module de sortie analogique.

La partie configuration du matériel est terminée, nous allons passer à la configuration PG/PC dans la partie suivante pour permettre la liaison de notre PC avec les différents appareils ajoutés (API, PC-system, HUB industriel et IHM TP900 Comfort).

III.2.3.2 Configuration PG/PC

La configuration PG/PC (Programming Device / Personal Computer) permet d'établir la communication entre l'ordinateur et l'automate programmable. Cette liaison est indispensable pour le téléchargement du programme, la supervision et le diagnostic du système.

Dans notre projet, la communication entre le PC et l'automate S7-1200 est réalisée via un réseau Ethernet/PROFINET en utilisant un HUB industriel permettant l'interconnexion des différents équipements du banc d'essai.

Le HUB industriel assure la communication entre :

- L'automate.
- L'ordinateur de programmation.
- L'IHM TP900 Comfort.

Les étapes pour la configuration PG/PC sont comme suit :

1. Création de carte réseaux pour le PC-System

Cette Figure nous montre comment ajouter un système de communication pour notre PC-System, cela nous permet d'ajouter des adresses IP pour le PC-Système.

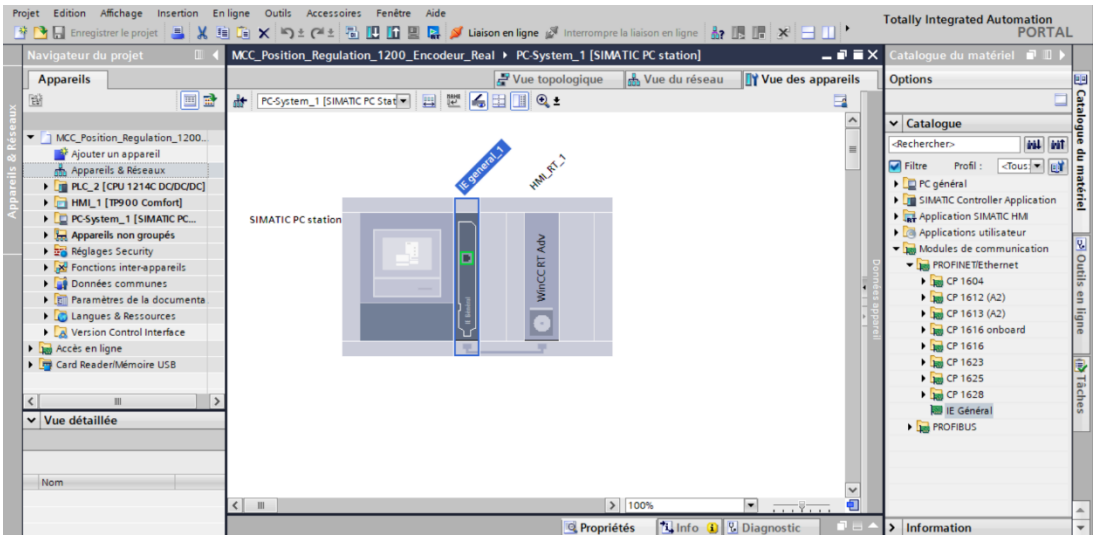


Figure III.11 : Création de la carte réseaux pour le PC Système.

2. Création des adresses IP

La Figure III.12 montre comment insérer les différentes adresses IP pour éliminer tout conflit.

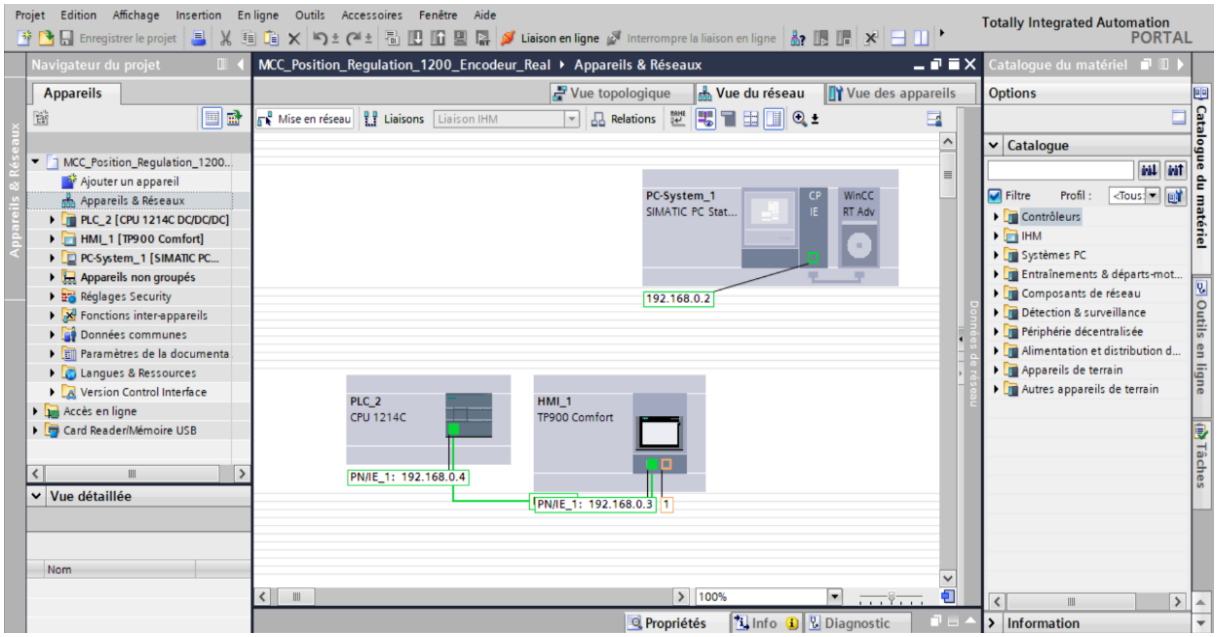


Figure III.12 : Attribution des adresses IP.

Maintenant le PC est connecté à l'automate réel et aussi à l'IHM, il nous reste à les configurer et les programmer.

III.2.3.3 Configuration du compteur rapide et obtention de la mesure capteur

Le compteur rapide (High Speed Counter) est utilisé pour acquérir les impulsions délivrées par l'encodeur incrémental. Contrairement aux entrées numériques classiques.

Le HSC permet de compter les impulsions à haute fréquence générées par l'encodeur sans perte d'information, c'est ce qui le rend adapté aux applications de mesure de position.

Pour configurer le HSC, nous allons passer par plusieurs étapes :

1. Choix du compteur rapide

Nous avons choisi le compteur rapide HSC_1 ayant un nom 'local_HSC_1', comme le montre la Figure III.13.

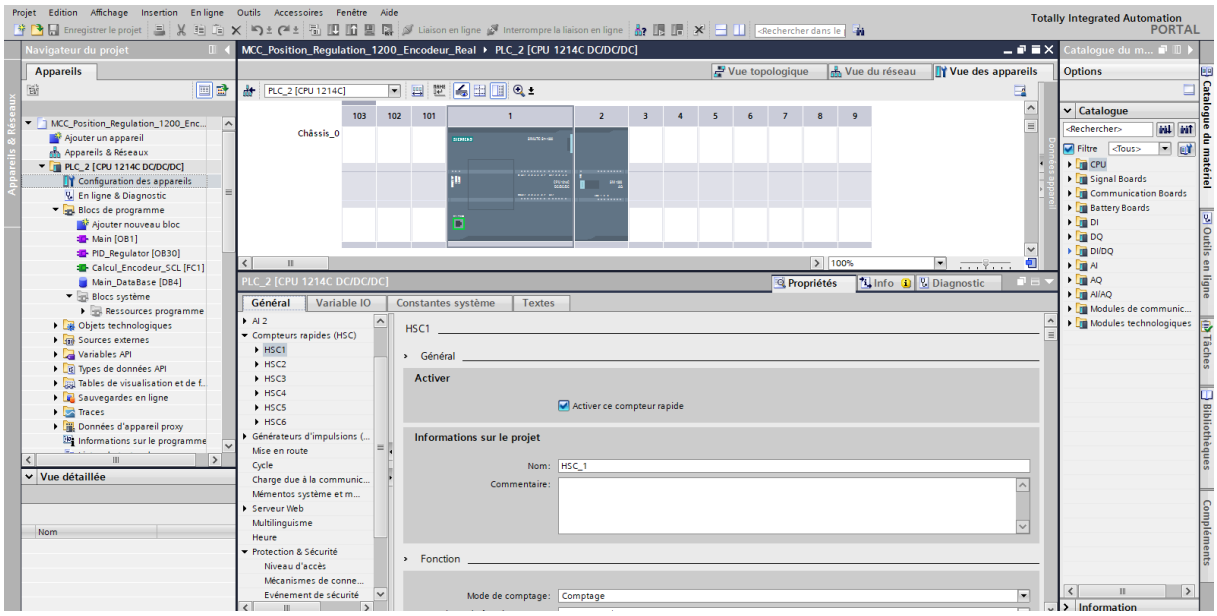


Figure III.13 : Choix du HSC.

2. Sélection du mode de fonctionnement

Nous avons sélectionné le mode de comptage A/B, pour identifier le sens de rotation du moteur (Figure III.14).

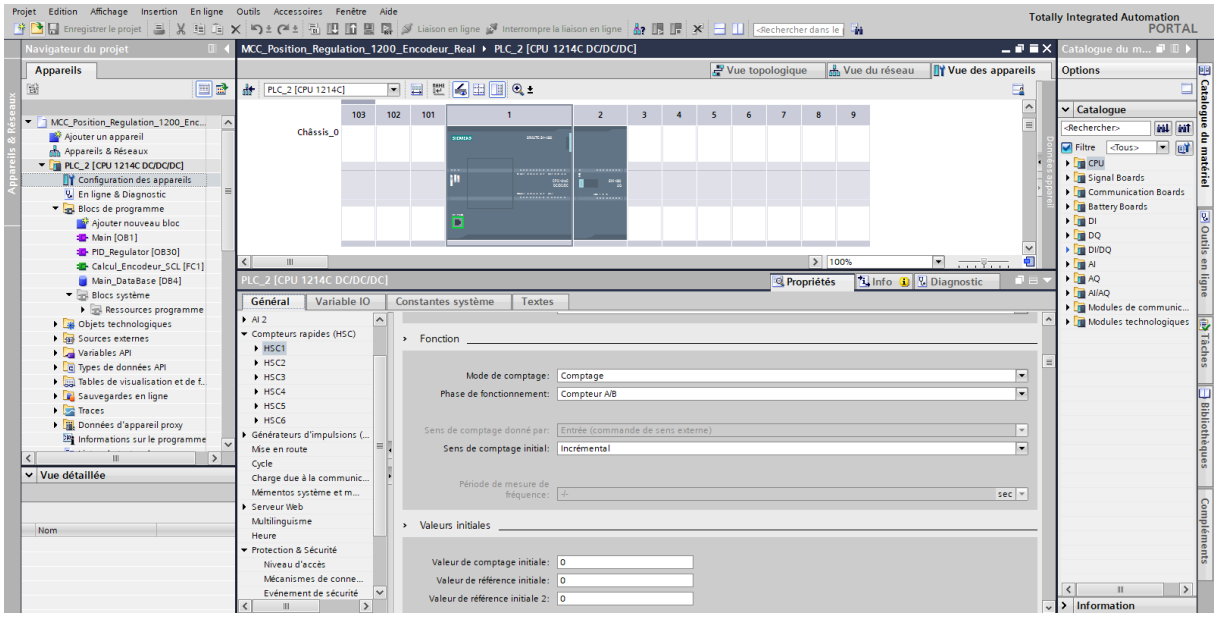


Figure III.14 : Mode de fonctionnement.

3. Affectation des entrées

Nous avons choisi un mode de fonctionnement Comptage A/B. Pour ce comptage, il faut une entrée physique '%I0.0' pour la voie A et une autre '%I0.1' pour la vois B (Figure III.15).

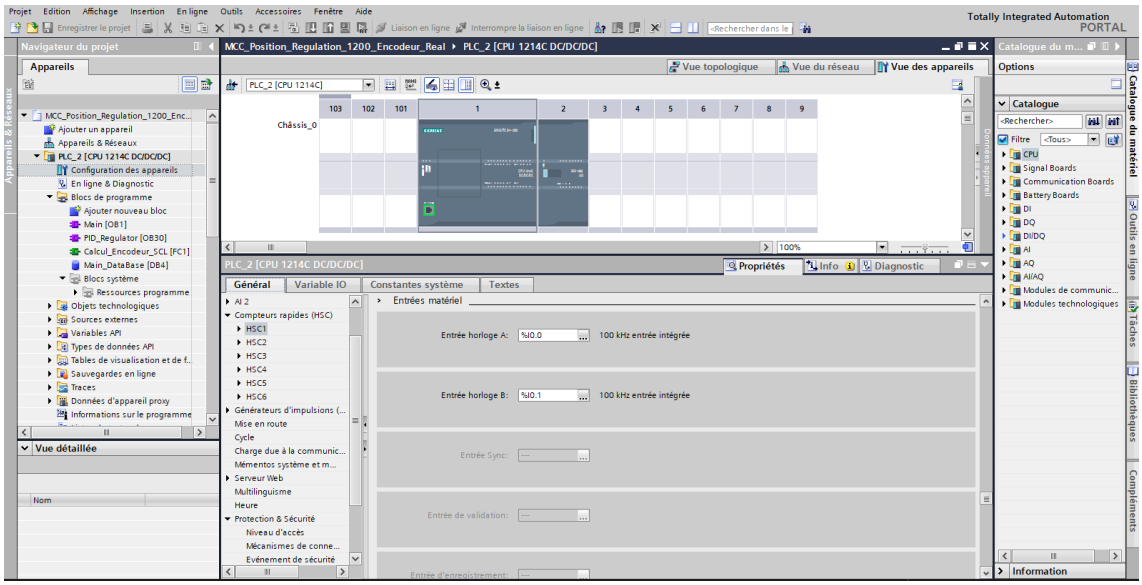


Figure III.15 : Choix d'entrée physique.

4. Adresses d'Entrée

Pour le HSC du S7-1200, les adresses de début et de fin (Figure III.16) représentent la zone mémoire utilisée par l'automate pour stocker les données du compteur rapide.

Elles permettent au programme d'accéder aux informations du HSC comme :

- La valeur du comptage.
- Le sens de rotation.
- L'état du compteur.
- Les commandes de démarrage ou de remise à zéro.

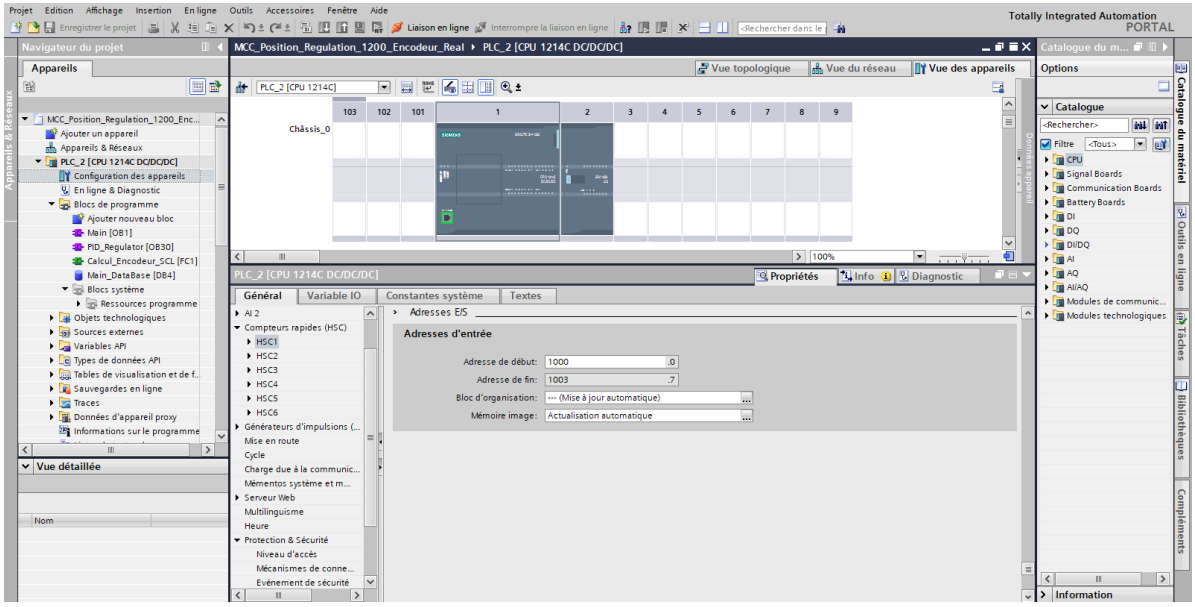


Figure III.16 : Adresses de début et de fin.

5. Programme Ladder

En premier, nous allons dans les instructions de base à Droite de la fenêtre (Figure III.17) pour prendre un bloc CTRL_HSC et le mettre dans un réseau Ladder.

Comme adresse d'Entrée de ce bloc, elle est fournie automatiquement 'Adresse 257' qui correspond à l'adresse mémoire de départ attribuée par TIA Portal au compteur rapide.

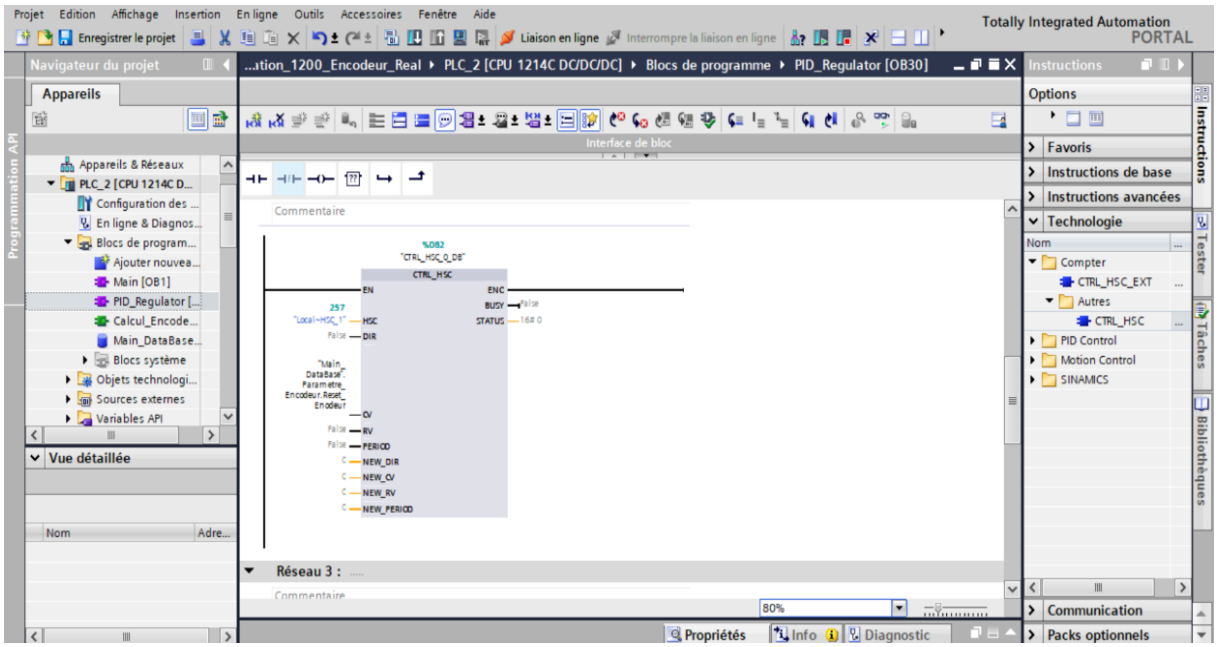


Figure III.17 : Création du bloc CTRL_HSC.

6. Obtention de la mesure capteur (création de la fonction SCL)

Une fonction SCL est un bloc de programmation utilisé dans les automates Siemens permettant d'écrire des programmes sous forme textuelle.

Le langage SCL est utilisé dans l'environnement TIA Portal pour réaliser des traitements complexes, des calculs mathématiques, des algorithmes de régulation et manipulation de variables.

Une fonction SCL reçoit des données d'entrée, exécute des instructions programmées puis retourne un résultat ou agit sur des variables du système.

Notre fonction SCL 'Calcul_Encodeur_SCL' est codée comme le montre la Figure III.18.

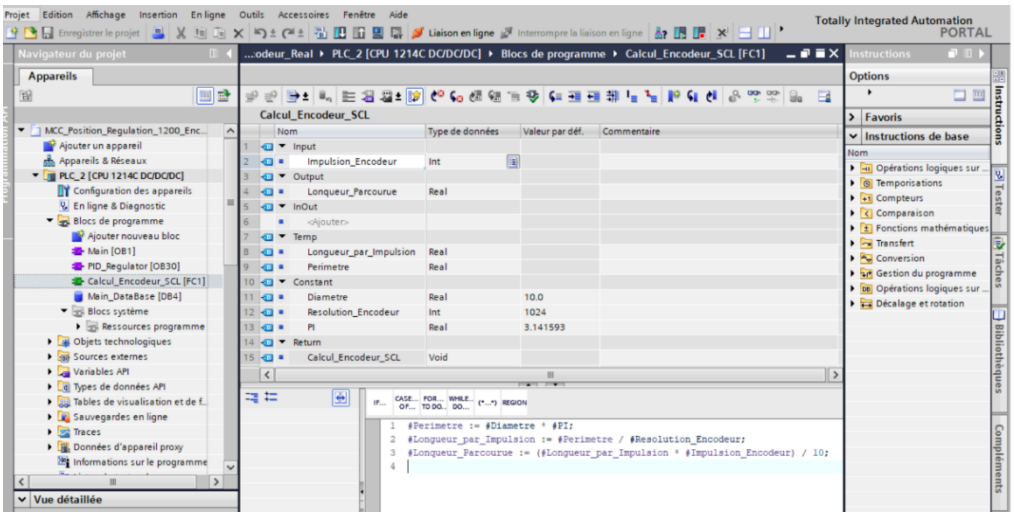


Figure III.18 : Fonction SCL pour le calcul de distance parcourue par le biais de l'encodeur.

Nous allons glisser le bloc de la fonction SCL de l'encodeur, qui se trouve avec les différents blocs, dans un réseau Ladder au niveau de l'OB cyclique 'PID_Regulateur (OB30)'. Pour obtenir la longueur parcourue par le biais de cette fonction, nous devons introduire l'adresse de début du HSC (%ID1000) à l'entrée du bloc (Figure III.19).

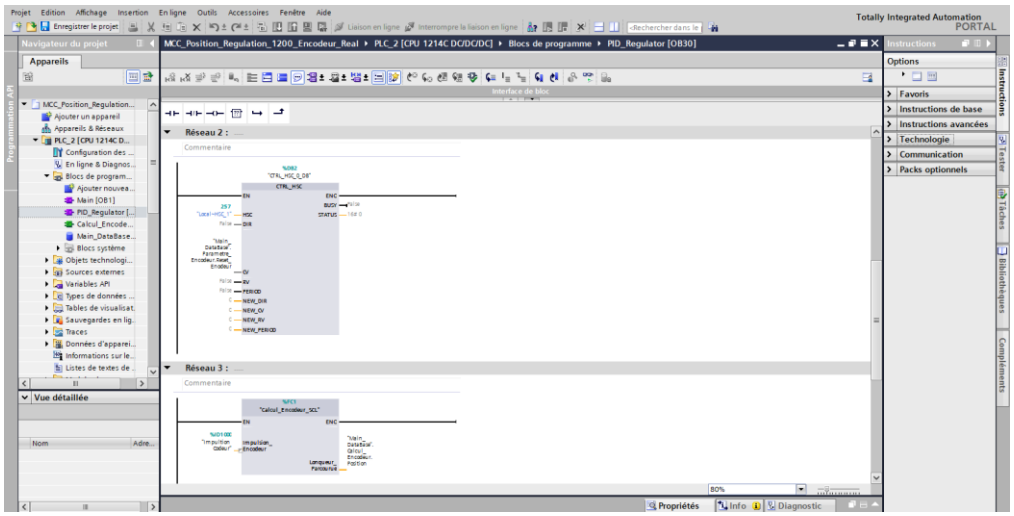


Figure III.19 : Configuration complète du HSC.

III.2.3.4 Configuration du générateur PWM

Le générateur PWM (Pulse Width Modulation) est utilisé pour commander le moteur à courant continue à travers le pont en H du driver L298N.

La modulation de largeur d'impulsion permet de faire varier la tension moyenne appliquée au moteur en modifiant le rapport cyclique du signal PWM.

Pour notre banc d'essai, le signal de PWM est généré par un générateur d'impulsions interne.

Nous allons configurer le générateur d'impulsions dans les étapes suivantes :

1. Choix du générateur d'impulsions

Le Pulse_3 est choisi et non pas le Pulse_1, ni Pulse_2 car ils nous ont créé une confusion entre les adresses de début et fin du compteur rapide (Figure III.20).

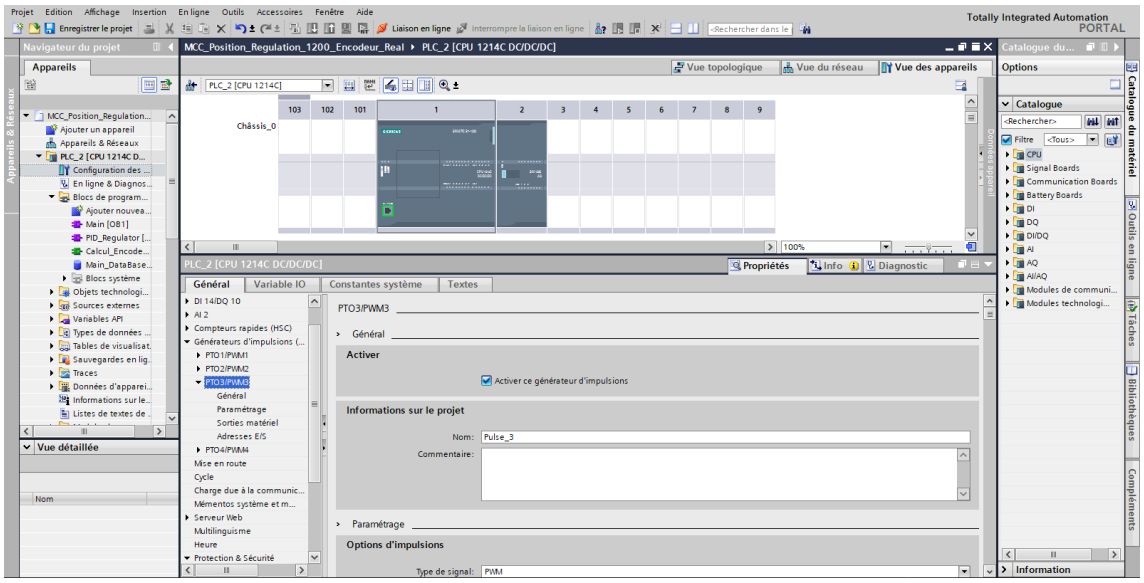


Figure III.20 : Choix du générateur d'impulsions.

2. Paramétrage et adresse de la sortie physique

La sortie physique est la %Q0.0 de notre automate dans la quel nous allons brancher le câble de commande du variateur de vitesse. A travers cette sortie, l'API va nous fournir le signal de commande.

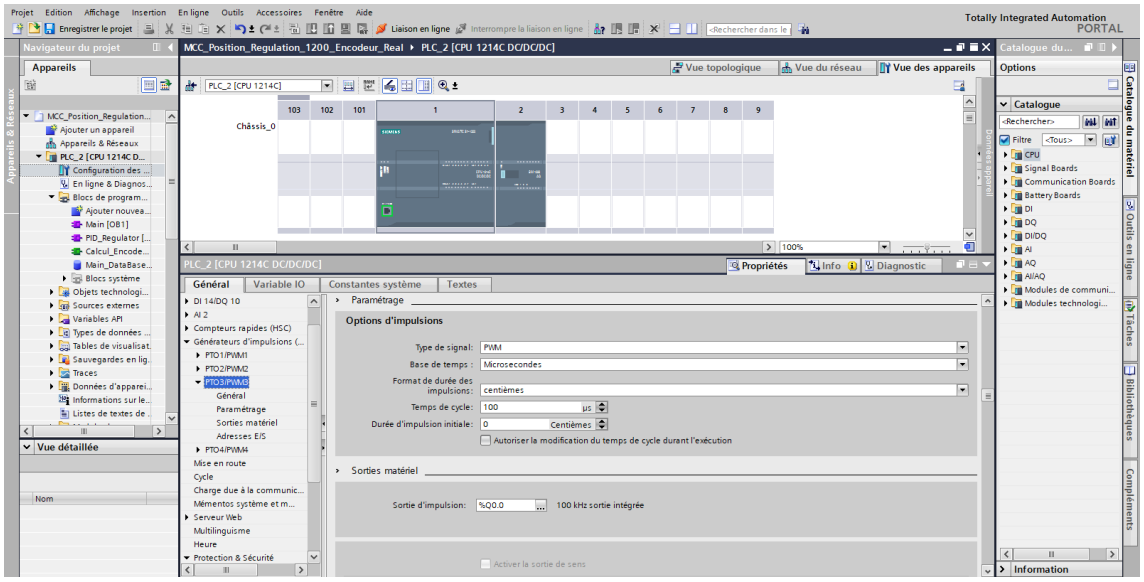


Figure III.21 : Paramétrage et adresse de sortie physique ou matérielle.

3. Attribution automatique des adresses du début et de la fin

Pour le fonctionnement de ce générateur, il faut bien récupérer les adresses du début et de la fin fournie automatiquement par la CPU (Figure III.22).

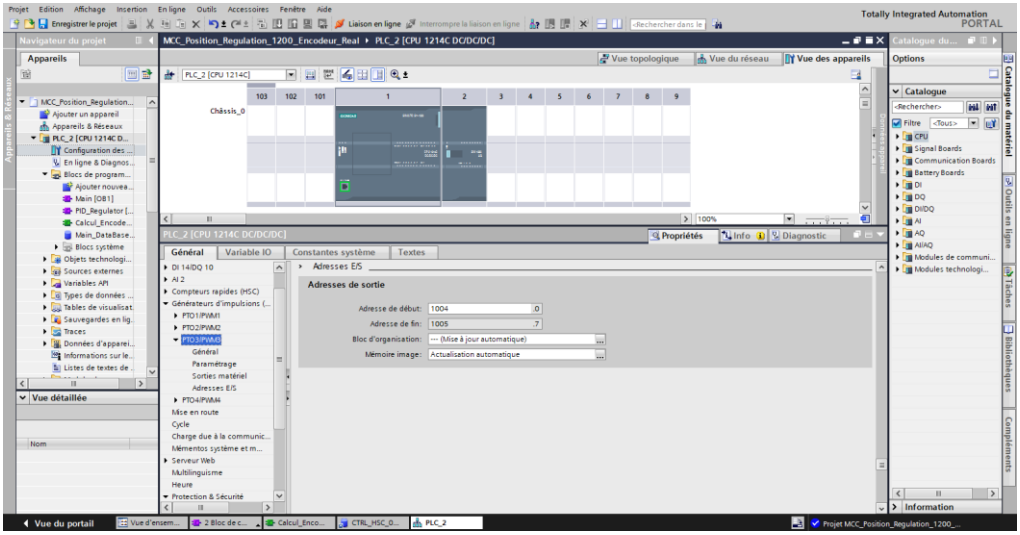


Figure III.22 : Adresses de début et de fin.

Maintenant il reste juste une dernière étape qui est la configuration du Pulse_3 dans un réseau Ladder en suivant presque la même procédure que le compteur HSC.

4. Programme Ladder

La première chose, nous allons ajouter un bloc CTRL_PWM avec l'adresse 276, qui correspond à l'adresse mémoire interne attribuée au local_Pulse_3.

Il faut ajouter un bouton sur l'entrée ENABLE du bloc CTRL_PWM pour permettre à l'utilisateur d'activer ou désactiver la génération du signal PWM (Figure III.23).

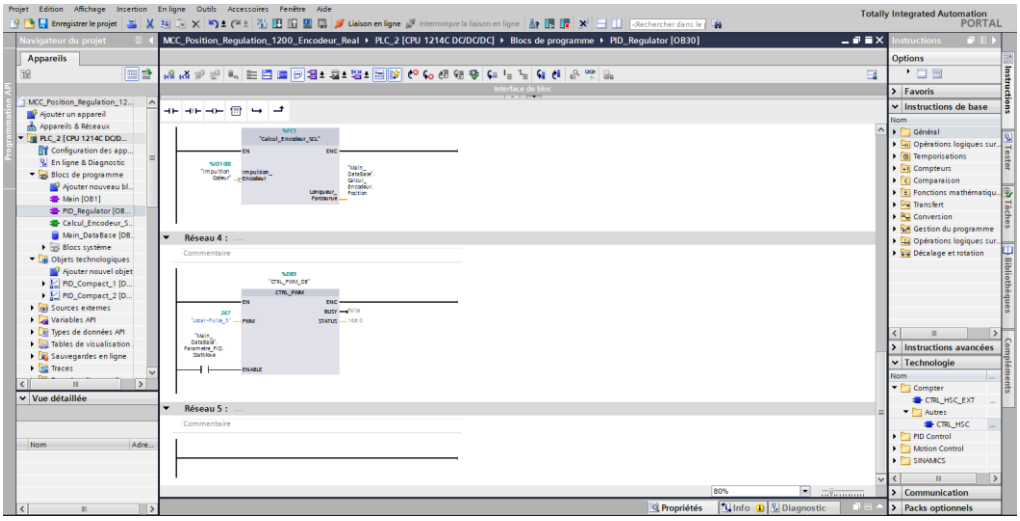


Figure III.23 : Création du bloc CTRL_PWM.

C'est impossible de générer un signal PWM sans avoir la valeur du rapport cyclique de notre signal de commande qui doit être affectée à l'adresse de sortie de notre générateur 'Pulse_3' qui doit varier de « 0 à 100 » de type 'Int'.

La valeur de notre rapport cyclique est fournie par le biais de la sortie normale (Output) du régulateur PID_Compact qui est une valeur de type 'Real' comprise entre [0.0 - 1.0]. Pour la convertir en type 'Int', il faut passer, en premier par le bloc 'Norm', puis le bloc 'SCALE_X' pour obtenir la plage la plus adaptée au fonctionnement de notre moteur qui est comprise entre [40-100].

Cette méthode permet d'adapter correctement la commande du PID au rapport cyclique appliqué au moteur tout en imposant des limites de fonctionnement sécurisées.

La limite inférieure (trouvée après essai) du rapport cyclique pour le fonctionnement de notre moteur est 40%. C'est à partir de cette valeur que le moteur commence à tourner. Au-dessous de cette valeur, la tension est très faible pour faire tourner les moteurs.

Pour améliorer notre système, nous avons ajouté un réseau Ladder qui nous permet de forcer une valeur de 70% pour le rapport cyclique du signal PWM pour remettre le tapis roulant à la position de départ (voir Figure III.24).

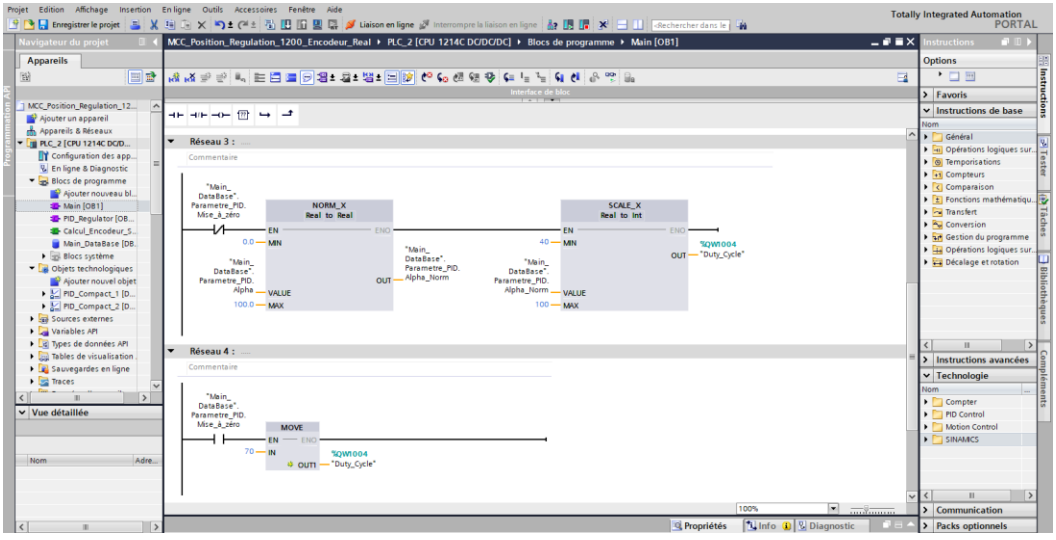


Figure III.24 : Normalisation et mise à l'échelle du signal de commande.

III.2.3.5 Configuration du PID_Compact

Le bloc PID_Compact de TIA Portal est utilisé pour réaliser la régulation de position du système. Il compare en permanence la consigne de position à la position réelle mesurée par l'encodeur afin de calculer la commande à appliquer au moteur.

Pour configurer ce régulateur, nous allons passer par plusieurs étapes qui sont dans les Figures suivantes :

Dans la Figure III.25, nous avons pris un bloc PID_Compact dans la partie technologique (à droite de la fenêtre) avec la version 1.2 qui doit déposer sur un réseau Ladder au niveau d'un 'OB cyclique' (dans notre cas, nous avons choisi pour ce bloc un temps d'échantillonnage de 1ms) dans lequel il faut donner :

- Le 'Setpoint' : qui correspond à la variable de notre consigne,
- L'Input : qui la longueur mesurée par notre encodeur
- L'Output : est la variable nommée 'Alpha' qui correspond au rapport cyclique du signal de commande PWM.

Le réseau qui est en parallèle avec le PID_Compact est un réseau où nous avons calculé l'erreur avec un bloc SUB dans lequel on compare la consigne à la position donner par l'encodeur. Ce qu'il faut souligner est que ce réseau a été ajouté, non pas parce que le PID ne le calcule pas, mais c'est juste pour afficher la valeur de l'erreur sur notre vue dans l'IHM.

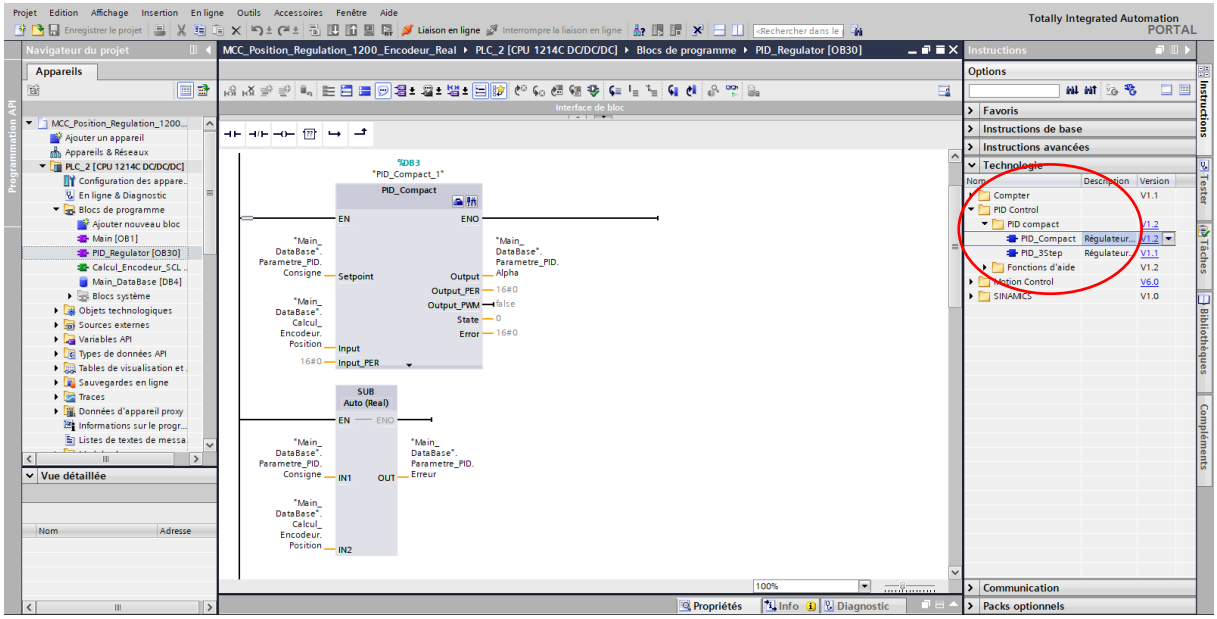


Figure III.25 : Création du bloc PID_Compact.

Dans les figures qui suivent, nous allons montrer les étapes de configuration interne du PID (Type de régulation, les paramètres d'entrée et aussi les paramètre du PID).

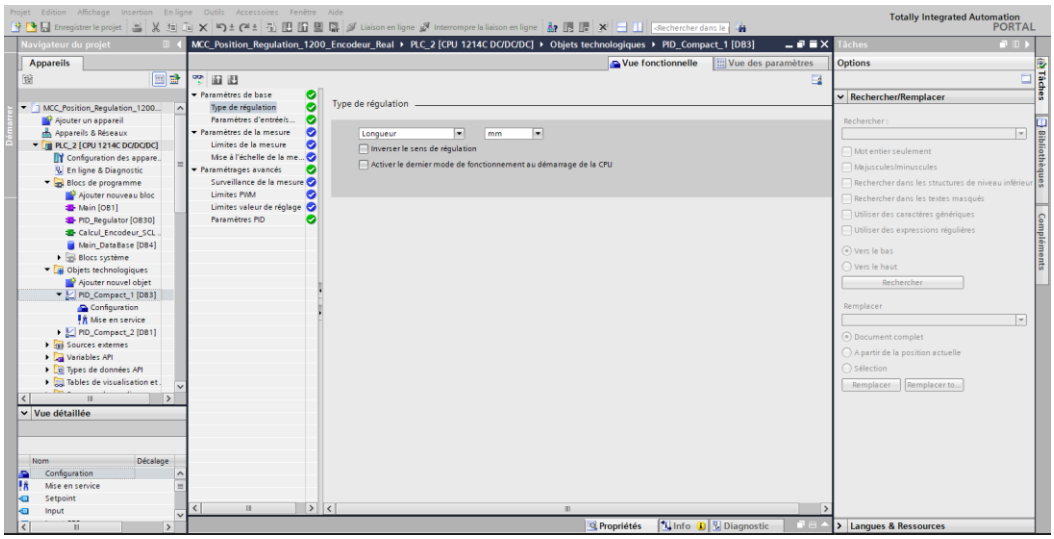


Figure III.26 : Type de régulation 'mm'.

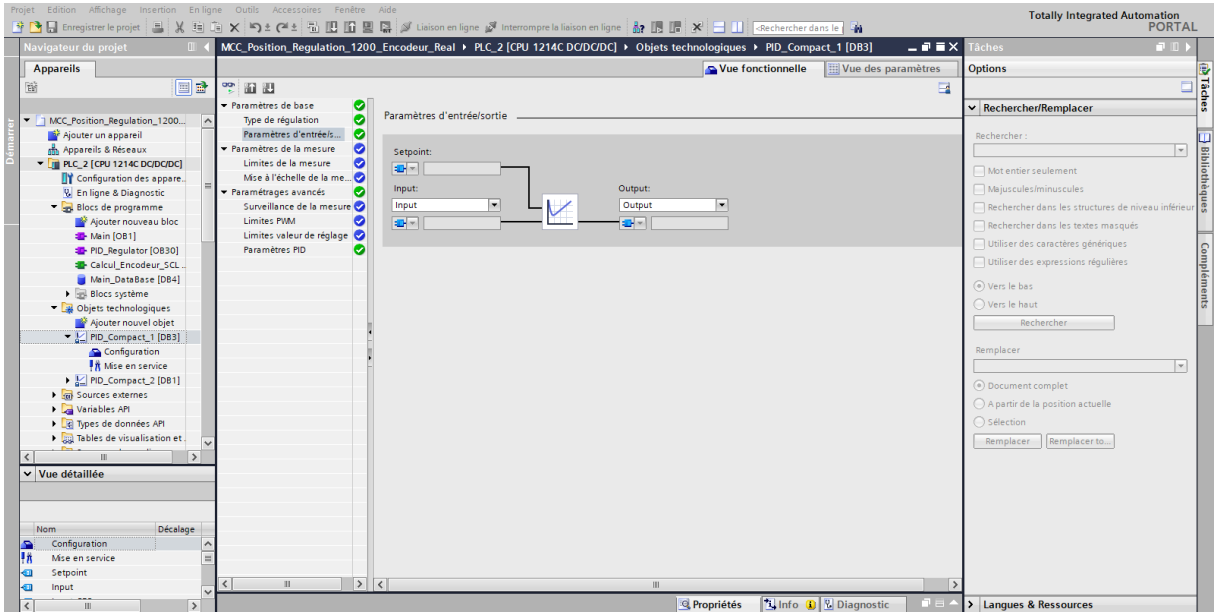


Figure III.27 : Sélection des paramètres d'entrées.

Dans la configuration du bloc PID_Compact, seul l'action proportionnelle a été utilisée. L'action intégrale a été désactivée afin d'éviter l'introduction d'un comportement intégrateur supplémentaire dans la boucle de régulation. En effet, la position est obtenue à partir d'un encodeur incrémental dont les impulsions sont comptées par l'automate pour déterminer le déplacement angulaire du moteur. Ce comptage correspond à une accumulation des impulsions au cours du temps, ce qui permet d'obtenir directement la position à partir du mouvement du moteur. D'un point de vue dynamique, la boucle de position comporte déjà un effet intégrateur puisque la position est l'intégrale de la vitesse angulaire.

L'ajout d'une action intégrale dans le régulateur n'était donc pas nécessaire et aurait pu augmenter le risque d'oscillations ainsi la difficulté de réglage du système. Pour cette raison, seul le gain Kp a été réglé expérimentalement.

Les paramètre du PID de la Figure III.28 ne sont pas ceux utilisés pour le fonctionnement final du banc d'essai. Pour le bon fonctionnement de cet objet, nous devons donner une valeur défaut.

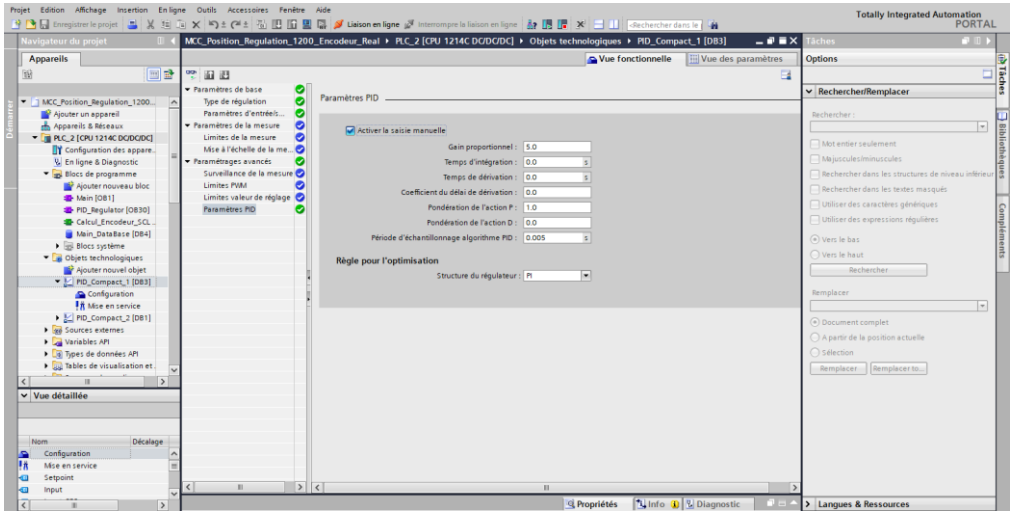


Figure III.28 : Les paramètres du PID.

III.2.3.6 Configuration de la 'main Data Base'

Afin de centraliser les données utilisées par le programme, une Main Data Base (DB principal) a été créé. Cette base de données contient l'ensemble des variables nécessaires au fonctionnement du projet (Figure III.29). L'avantage de cette procédure est de fournir des données structurées et sans conflits d'adresse.

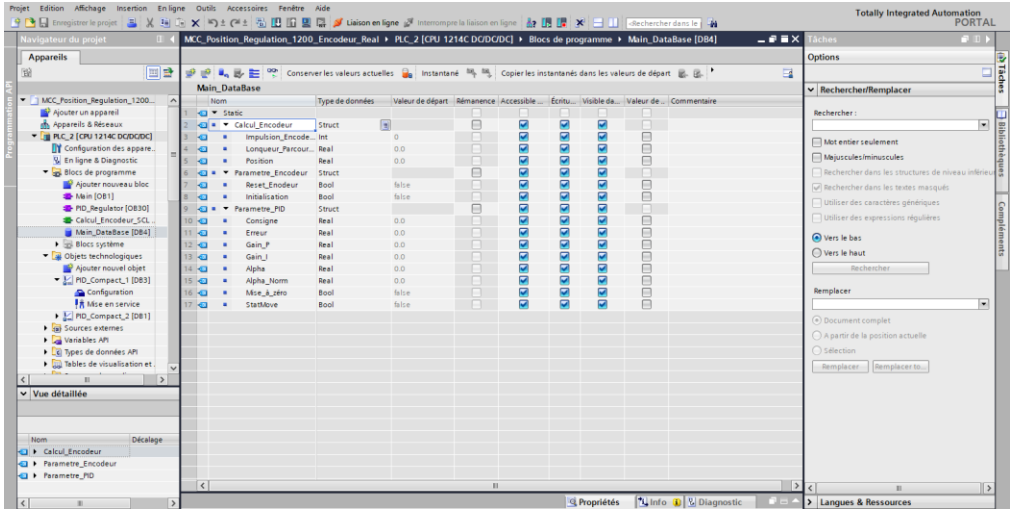


Figure III.29 : Création du Main Data Base.

III.2.4 Partie supervision

La supervision permet à l'opérateur de surveiller et de piloter le système à travers une interface Homme-Machine (IHM). Cette interface offre une visualisation en temps réel des grandeurs importantes du procédé ainsi qu'un accès simple aux commandes du système.

III.2.4.1 Outil de supervision IHM

Une interface Homme-Machine (IHM) est un dispositif permettant d'assurer la communication entre l'opérateur et le système automatisé. Elle constitue une interface conviviale à travers laquelle l'utilisateur peut surveiller l'état du procédé, visualiser les grandeurs importantes et agir sur le fonctionnement du système.

Dans le cadre de ce projet, la supervision est assurée par un pupitre opérateur Siemens SIMATIC HMI TP900 Confort équipé d'un écran tactile couleur de 9 pouces (Figure III.30). Ce pupitre offre une interface graphique performante permettant à l'utilisateur d'interagir facilement avec le système de régulation de position. Grace à son écran tactile, il est possible de modifier la consigne, de visualiser les variables du procédé et de suivre en temps réel l'évolution du système.



Figure III.30 : IHM TP900 Confort.

III.2.4.2 Création des variables IHM

La première étape consiste à créer les variables de communication entre l'IHM et l'automate dans une table de variable IHM, montré dans la Figure III.31.

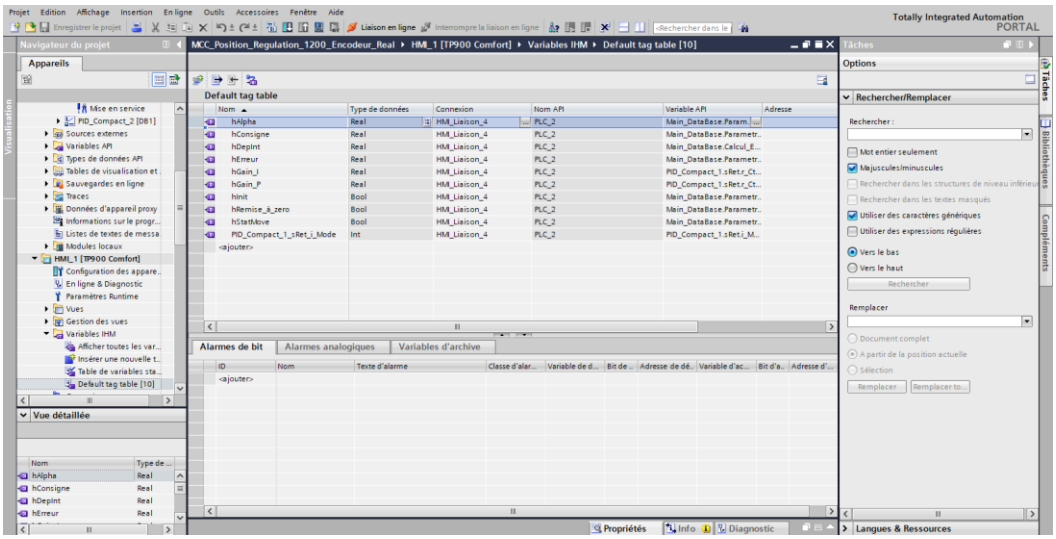


Figure III.31 : Variables de l'IHM.

III.2.4.3 Création de la vue IHM

Après la création des variables de communication, une vue de supervision a été développée à l'aide de WinCC intégré à TIA Portal. Cette étape consiste à concevoir l'interface graphique permettant à l'opérateur d'interagir avec le système de régulation de position.

La vue de l'IHM a été réalisée en utilisant différents objets graphiques tels que des boutons, des champs d'entrée, des champs d'affichage numériques ainsi que les indicateurs d'état. Chaque objet a été associé à une variable de l'automate afin d'assurer l'échange d'informations entre l'IHM et le programme de commande.

Les boutons de commande, tel que marche/arrêt et initialisation, sont configurés comme suit :

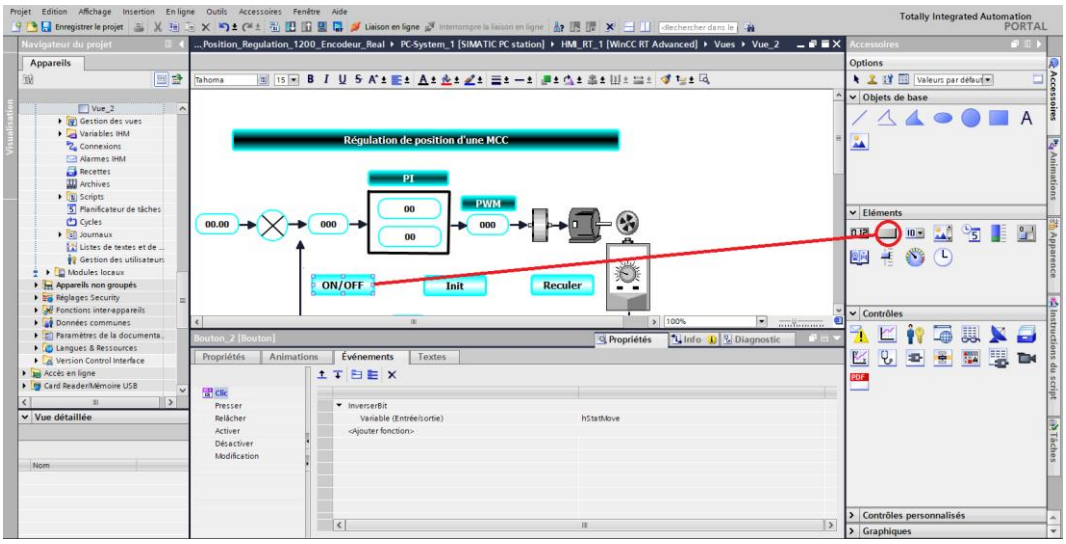


Figure III.32 : Création du bouton Marche/Arrêt.

Les champs d'entrée/sortie sont associés aux variables de consigne afin de permettre à l'opérateur de saisir directement la position souhaitée depuis l'écran tactile. De la même manière, nous avons ajouté les champs d'affichage comme l'erreur, la position mesurée par l'encodeur et les gains du régulateur.

Les deux Figures III.33 & III.34 expliquent comment faire pour créer un champ d'entrée et de sortie dans une vue.

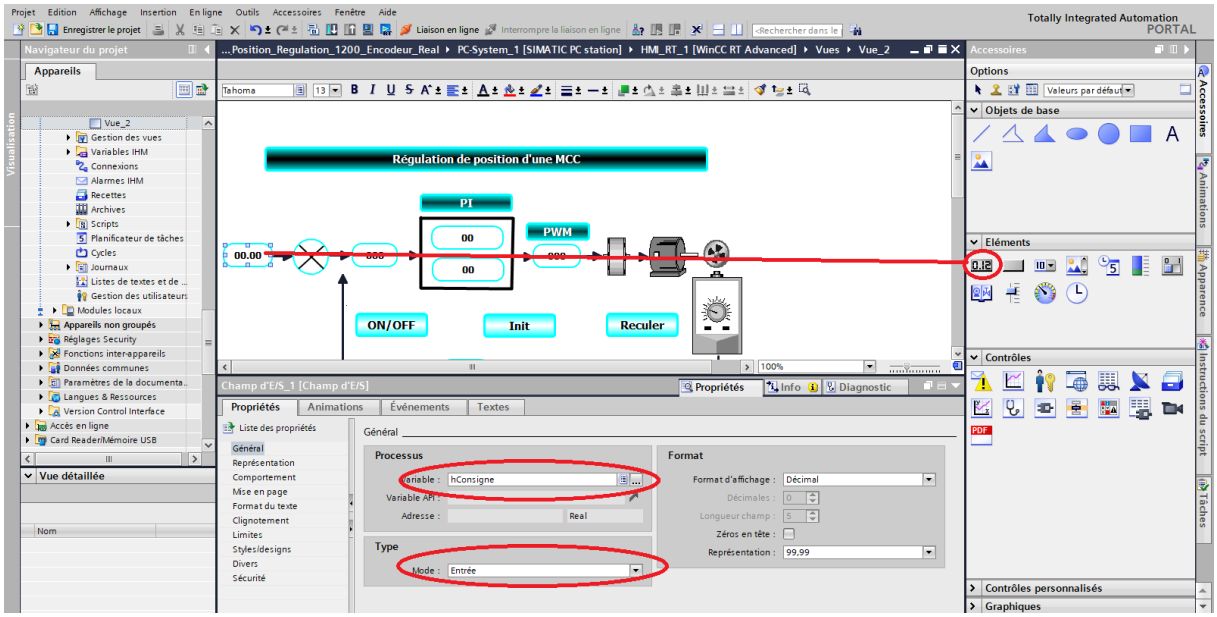


Figure III.33 : Champ d'entrée.

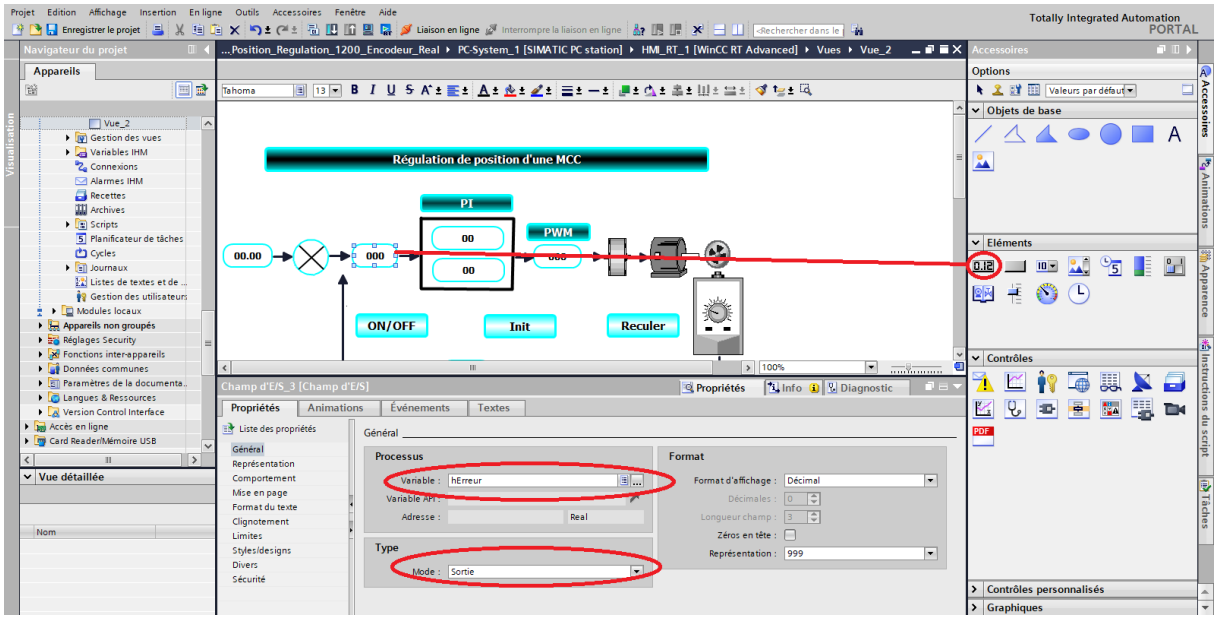


Figure III.34 : Champ de sortie.

La Figure III.35 donne la vue finale de notre IHM après avoir introduits tous les éléments nécessaires pour le bon fonctionnement de notre banc d'essai

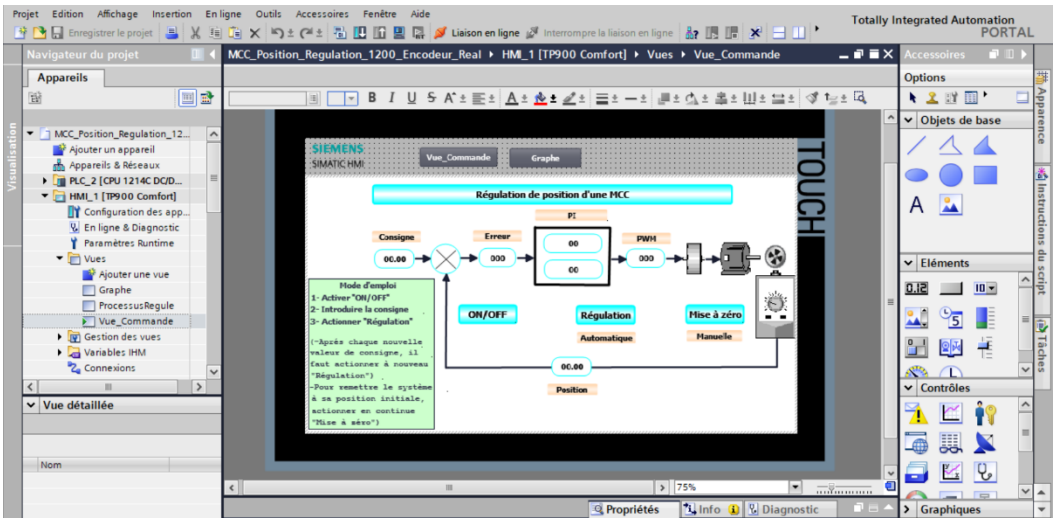


Figure III.35 : Vue finale de l'IHM.

III.2.4.4 Téléversement de la vue

Après la conception et validation de la vue de supervision, il est nécessaire de la transférer vers le pupitre Siemens HMI TP900 Confort afin de le rendre opérationnelle.

Le téléversement de la vue est réalisé depuis l'environnement TIA Portal à travers une connexion Ethernet établie entre l'ordinateur de programmation et l'IHM. Avant le transfert, une compilation logicielle et matérielle du projet est effectuée afin de vérifier l'absence d'erreurs de configuration ou de communication.

Une fois la compilation terminée, l'adresse IP de l'IHM est vérifiée, ensuite le chargement dans l'appareil (chargement complet) est sélectionné dans TIA Portal. Le logiciel établit la connexion avec le TP900 Confort.

Après avoir sélectionné l'IHM, le projet de supervision est transféré vers la mémoire de l'IHM. A l'issue du téléchargement, l'IHM redémarre automatiquement et charge la nouvelle vue de supervisassions. La vue téléchargée est celle dans la Figure III.30.

III.3 Test et interprétation des résultats

Après la mise en service du système, plusieurs essais expérimentaux ont été réalisés afin de déterminer la valeur optimale du gain proportionnel Kp du régulateur.

Les tests ont débuté avec une valeur initiale de Kp = 4. Cette valeur produisait une réponse stable, mais relativement lente avec une précision insuffisante. Afin d'améliorer les performances dynamiques du système, le gain proportionnel a été augmenté progressivement.

La valeur Kp = 60 a été retenue car elle offre le meilleur compromis entre rapidité, précision et stabilité. Pour cette valeur, le moteur atteint rapidement la position demandée tout en gardant un comportement stable et sans oscillation.

Les 04 Figures III.36, III.37, III.38 et III.39 montrent que notre banc d'essai atteint la consigne sans aucun dépassement et surtout sans oscillations.

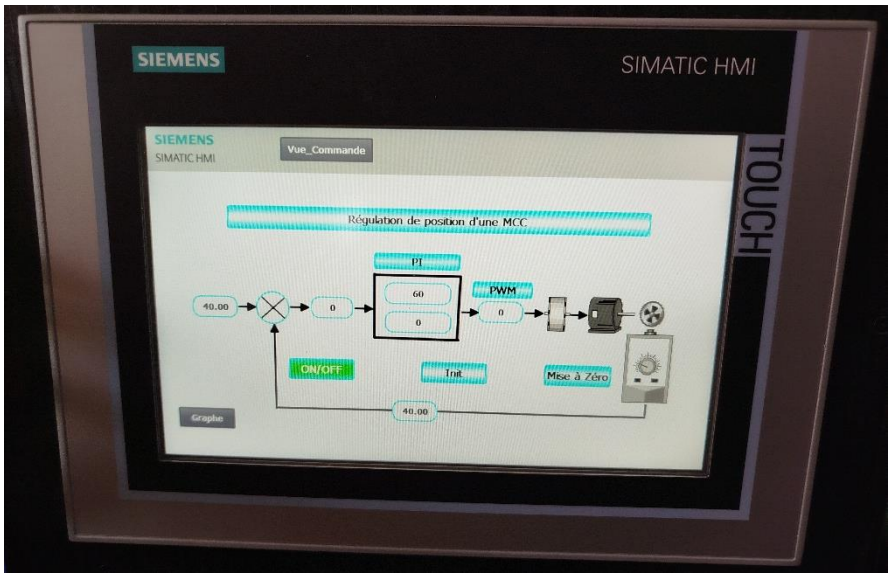


Figure III.36 : Vue IHM représentant la régulation pour une consigne = 40 mm.

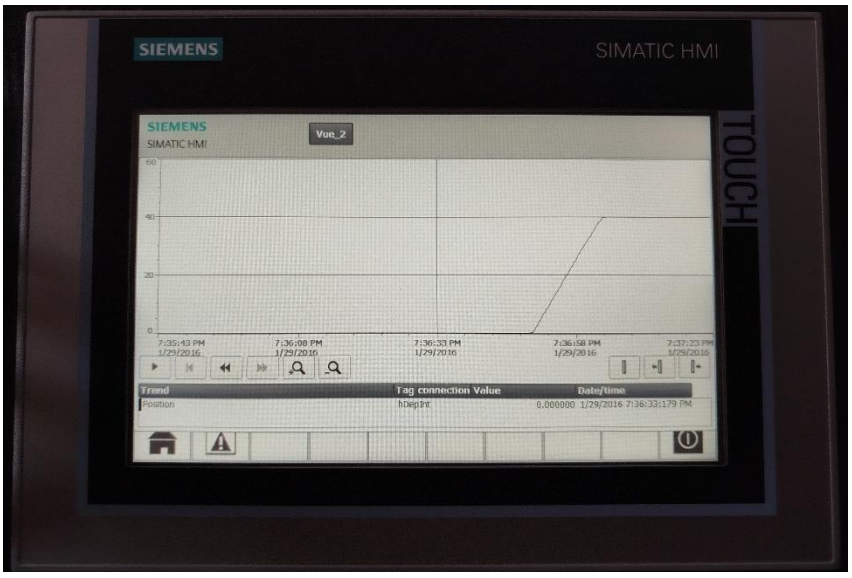


Figure III.37 : Réponse du système pour une consigne = 40 mm.

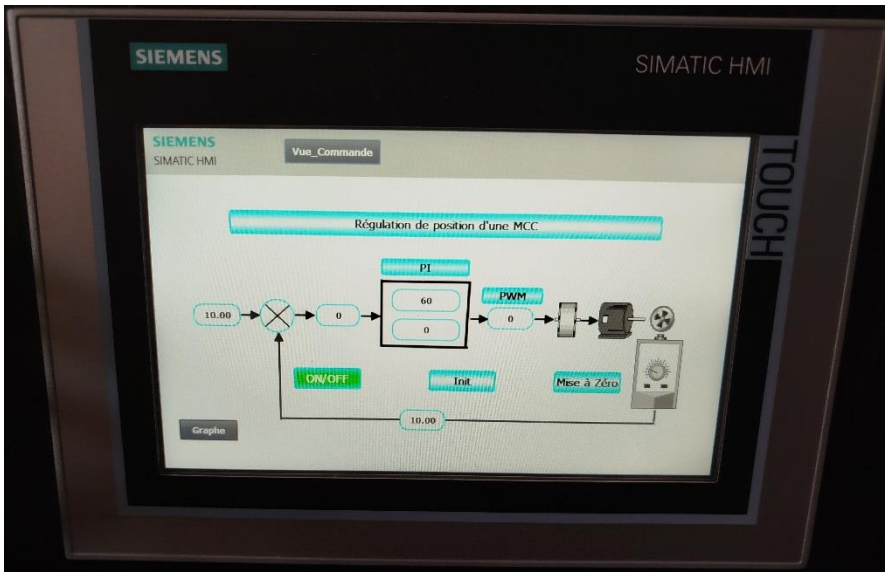


Figure III.38 : Vue IHM représentant la régulation pour une consigne = 10 mm.

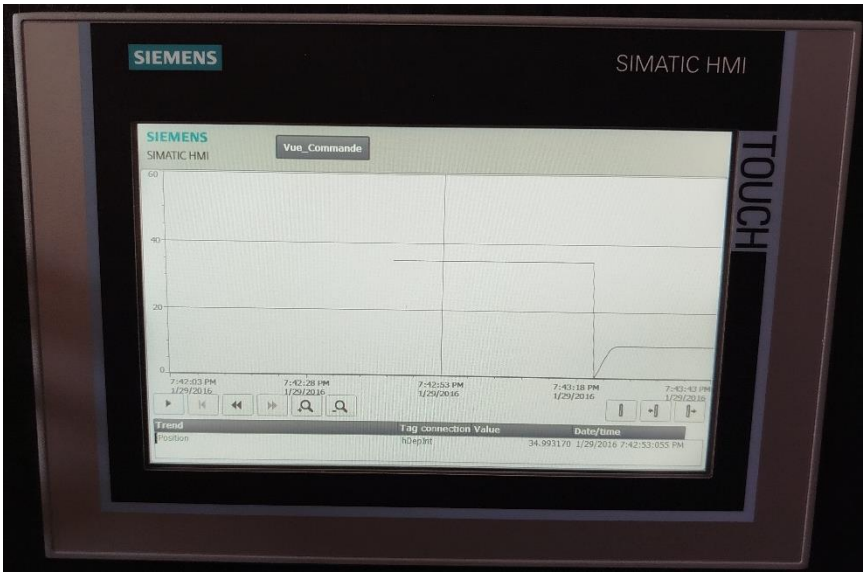


Figure III.39 : Réponse du système pour une consigne = 10 mm.

III.4 Conclusion

Dans ce chapitre, nous avons présenté la réalisation pratique du système de régulation de position basé sur un automate programmable S7-1200. Après la description du banc d'essai et de ses différents composants mécaniques et électriques, les étapes de programmation ont été détaillées, notamment la configuration de l'encodeur, générateur de PWM et du régulateur PID_Compact. La mise en place de la supervision à l'aide de l'IHM TP900-Comfort permettant le suivi de la commande du système en temps réel a été développée.

Les essais expérimentaux ont permis d’ajuster le gain proportionnel du régulateur. Les résultats obtenus ont montré que le système est capable de suivre correctement la consigne tout en conservant un fonctionnement stable.

Ainsi, l’ensemble des objectifs fixés pour ce projet a été atteint. Le banc d’essai réalisée constitue une solution efficace pour la régulation de position d’un moteur à courant continue.

Conclusion Générale

Conclusion Générale

Dans le cadre de ce travail, nous nous sommes intéressés à la réalisation d'un système de régulation de position basé sur un automate programmable industriel S7-1200. Après avoir présenté les principes généraux de régulation et les différentes applications qui en découlent, une étude détaillée du système a été menée afin de comprendre le fonctionnement des différents éléments constituant la chaine de commande.

La modélisation du moteur à courant continue, l'étude de réducteur, Du pont en H, l'encodeur incrémental ainsi que du régulateur PID ont permis d'établir les bases théoriques nécessaires à la conception du système. Cette étude a ensuite été complétée par la réalisation pratique sous TIA Portal, comprenant la configuration de l'encodeur, générateur PWM, la programmation de l'automate et le bloc PID_Compact sans oublier le développement d'une interface de supervision sur un IHM Siemens TP900 Confort.

Les essais expérimentaux réalisés ont permis de valider le bon fonctionnement du système. Le réglage du gain proportionnel par la méthode essai-erreur a conduit au choix d'une valeur de Kp = 60, offrant un compromis satisfaisant entre rapidité de réponse, précision et stabilité. Les résultats obtenus ont montré que le système est capable de suivre correctement les consignes tout en conservant un comportement stable.

Comme perspectives, nous envisageons l'intégration d'un module Wi-Fi afin de remplacer la liaison Ethernet pour la communication avec l'automate.

Nous proposons également l'ajout d'un capteur de référence permettant au système de revenir automatiquement à sa position de départ.

Enfin, l'utilisation d'un système de vision basé sur une caméra pourrait remplacer l'encodeur et offrir de nouvelles possibilités de mesure de position.

Références bibliographiques

[1] P. Gomes, “Medical robotics: Minimally invasive surgery,” IEEE Robotics & Automation Magazine, vol. 13, no. 2, pp. 94–102, 2006.

[2] G. S. Guthart and J. K. Salisbury, “The Intuitive™ telesurgery system: Overview and application,” in Proc. IEEE Int. Conf. Robotics and Automation, 2000.

[3] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, Robotics: Modelling, Planning and Control. Springer, 2009.

[4] M. W. Spong, S. Hutchinson, and M. Vidyasagar, Robot Modeling and Control. Wiley, 2005.

[6] B. E. Boser and R. T. Howe, “Surface Micromachined Accelerometers,” IEEE Journal of Solid-State Circuits, vol. 31, no. 3, pp. 366–375, 1996.

[7] Canon Inc., “Image Stabilizer Technology.” [Online]. Available: https://global.canon/en/technology/canon_tech/explanation/is.html

[8] K. Ogata, Modern Control Engineering, 5th ed. Prentice Hall, 2010.

[10] J. R. Wertz, Spacecraft Attitude Determination and Control. Springer, 1978.

[11] K. Ogata, Modern Control Engineering, 5th ed. Prentice Hall, 2010.

[12] H. K. Khalil, Nonlinear Systems, 3rd ed. Prentice Hall, 2002.

[13] https://theses.hal.science/tel-04095485v1/file/HDR_HERISSE_BRUNO_v1.pdf

[23] Support Siemens : https://support.industry.siemens.com/dl/files/036/108210036/att_916450/v4/s71500_pid_control_function_manual_en-US_en-US.pdf

Références sitographiques

[5] https://www.gustaveroussy.fr/fr/chirurgie/chirurgie-robotique

[9] https://fr.canon-cna.com/pro/infobank/image-stabilisation-lenses/

[14] https://www.wikihow.com/Stop-at-a-STOP-Sign

[15] https://sieportal.siemens.com/en-qa/products-services/detail/6AG1214-1HG40-4XB0?tree=CatalogTree#overview

[16] https://www.ifm.com/fr/fr/product/RU3500

[17] https://www.researchgate.net/figure/Boucle-fermee-pour-une-poursuite-a-base-des-capteurs_fig4_319741560

[18] https://www.developpez.net/forums/d1898443/general-developpement/algorithme-mathematiques/mathematiques/courbe-bezier-profil-trapezoidal-vitesse/

[19] https://www.nomadeducation.fr/revisions/prepa-pc-2e-annee-pc-lb6f2768/la-machine-a-courant-continu-cyf29d07

[20] https://zestedesavoir.com/forums/sujet/2514/pont-h-4-mosfet/

[21] https://igm.univ-mlv.fr/~dr/XPOSE2013/arduino/plateforme.html

[22] https://www.researchgate.net/figure/Schema-de-bloc-du-regulateur-PID-Le-signal-de-sortie-est-soustrait-de-la-valeur-de_fig24_281956558

[23] https://support.industry.siemens.com/dl/files/036/108210036/att_916450/v4/s71500_pid_control_function_manual_en-US_en-US.pdf

الهدف من هذا العمل هو تصميم وتنفيذ نظام التحكم في الموضع لمحرك تيار مستمر ذو مغناطيس دائم باستخدام وحدة تحكم منطقية صناعية قابلة للبرمجة Siemens S7-1200. يستخدم النظام وحدة تحكم PID_Compact، ومشفرًا تدريجيًا لقياس الموضع، ووحدة تحكم PWM لتشغيل المحرك. وتسمح النتائج التي تم الحصول عليها بتقييم أداء النظام من حيث الدقة والاستقرار وسرعة الاستجابة.

الكلمات المفتاحية: محرك التيار المستمر، التحكم في الموضع،

PID_Compact، PWM، PLC Siemens S7-1200

Abstract

The objective of this work is to design and implement a position control system for a permanent magnet DC motor using a Siemens S7-1200 Programmable Logic Controller (PLC). The system employs a PID_Compact controller, an incremental encoder for position feedback, and PWM control for motor actuation. The obtained results are used to evaluate the system performance in terms of accuracy, stability and response speed.

Keywords : Position Control, Siemens S7-1200 PLC, DC motor, PID_Compact, PWM.

Résumé

L’objectif de ce travail est de concevoir et de réaliser un système de régulation de position d’un moteur à courant continu à aimant permanent à l’aide d’un automate programmable industriel Siemens S7-1200. Le système utilise un régulateur PID_Compact, un encodeur incrémental pour la mesure de position et une commande PWM pour piloter le moteur. Les résultats obtenus permettent d’évaluer les performances du système en termes de précision, de stabilité et de rapidité de réponse.

Mots-clés : Régulation de position, API Siemens S7-1200, moteur à courant continue, PID_Compact, PWM.