

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

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

جامعة سعد دحلب البلدية
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا
Faculté de Technologie

قسم الية و الالكترونقتي
Département Automatique & Electrotechnique



Mémoire de Master

Filière Automatique

Spécialité Automatique & Systèmes

Présenté par

HABIB Badr Eddine

Etude et implémentation de la commande prédictive généralisée à base d'un automate programmable industrielle Siemens S7-1200

Proposé par : Dr. BENRABAH Mohamed

Pr. KARA Kamel

Année Universitaire 2022-2023

Remerciements

Tout d'abord, je tiens à remercier mon DIEU LE TOUT PUISSANT, qui a rendu cet ouvrage possible.

Je tiens à exprimer ma profonde gratitude à mon promoteurs Dr. M. BENRABAH et Pr. K. KARA pour la proposition du sujet de cette thèse et pour ses conseils et son soutien tout au long de ce travail.

Nos remerciements à tous les membres du jury qui ont accepté de juger notre travail.

Mes remerciements aussi à :

Mes parents.

Mes frères, mes sœurs et à toutes ma famille.

Mes amies intimes : (F. Houcine – K. Abdelatif)

Un remerciement spécial aux étudiants de la 2^{ème} année Master de la spécialité d'Automatique et systèmes

Dédicace

Avant tout je tiens à remercier le bon dieu est l'unique qui m'offre le courage Et la volonté nécessaire pour affronter la difficulté de la vie Je dédié ce modeste travail à :

A la mémoire de mon cher père

A ma mer

A mes frères, mes sœurs et à toutes ma famille

A mon amie intime F. Houcine

HABIB Badr Eddine

ملخص:

إن التحكم التنبؤي أصبح من الأدوات الأساسية في العديد من المجالات، ويتم دراسته بشكل مكثف خاصة في الأنظمة الخطية في هذا العمل سنقوم بتعريف بالتحكم التنبؤي المعمم (GPC) من أجل التحكم بسرعة محرك التيار المستمر، حيث قمنا بنمذجة و تعريف المحرك قصد الحصول على نموذج CARIMA الخاص بهذا الأخير، بعد الحصول على نموذج المتحكم التنبؤي المعمم قمنا ببرمجته في المتحكم الالي (API) siemens S7-1200 بواسطة برنامج TIA Portal باستخدام لغة SCL وانشاء واجهة الة بشرية (HMI) لمراقبة تغير السرعة .

الكلمات المفتاحية: التحكم التنبؤي المعمم، محرك التيار المستمر، API، TIA Portal

Résumé :

La commande prédictive est devenue un outil essentiel dans de nombreux domaines, en particulier dans les systèmes linéaires. Dans ce travail, nous définissons la commande prédictive généralisée (GPC) pour contrôler la vitesse d'un moteur à courant continu. Nous avons modélisé et caractérisé le moteur pour obtenir son modèle CARIMA. Après avoir obtenu le modèle de commande prédictive généralisée, nous l'avons programmé l'automate programmable S7-1200 de Siemens en utilisant le langage SCL à l'aide du logiciel TIA Portal et nous avons créé une interface homme-machine (HMI) pour surveiller les changements de vitesse.

Les mots-clés : commande prédictive généralisée, moteur à courant continu, API, TIA Portal

Summary:

Predictive control has become an essential tool in many fields, especially in linear systems. In this work, we define generalized predictive control (GPC) to control the speed of a DC motor. We modeled and characterized the motor to obtain its CARIMA model. After obtaining the generalized predictive control model, we programmed it in the Siemens S7-1200 programmable logic controller using the SCL language using the TIA Portal software and created a human-machine interface (HMI) to monitor speed changes.

Keywords: generalized predictive control, DC motor, API, TIA Portal

Sommaire

Remerciements	i
Dédicace.....	ii
Résumé	iii
Introduction générale.....	1

chapitre1: commande prédictive généralisée

1.1 Introduction	3
1.2 Généralités.....	3
1.3 Principe :	3
1.4 Types d'algorithmes prédictifs	4
1.5 La Commande Prédictive généralisée (GPC)	5
1.5.1 Le modèle de prédiction	6
1.5.2 Structure du prédicteur optimal :	7
1.5.3 Forme matricielle de l'équation de prédiction :.....	9
1.5.4 Résolution récursive des équations diophantiennes :	9
1.5.5 Loi de commande :.....	10
1.6 Conclusion :	12

chapitre2 : modélisation et identification du moteur à courant continu

2.1 Introduction :	14
2.2 Présentation de la machine à courant continu :	14
2.2.1 Définition :	14
2.2.2 Constitution d'une machine à courant continu :	15
2.3 Les équations du moteur à courant continu :	15
2.3.1 Force électromotrice :	15
2.4 Modélisation d'un moteur à courant continu :	18
2.4.1 Equations électriques :	18
2.4.2 Equations mécaniques :	19

2.4.3	Equations magnétiques :	20
2.4.4	Représentation d'état d'un moteur à courant continu :	20
2.5	Simulation :	21
2.6	Conclusion :	26

chapitre3 : Implémentation de la commande et essais pratiques

3.1	Introduction :	28
3.2	Description du système :	28
3.3	Description de l'automate siemens S7-1200 :	29
3.4	Supervision :	30
3.4.1	Définition de la Supervision :	30
3.4.2	Constitutions d'un system de supervision :	30
3.5	Codeur incrémental [15]	31
3.5.1	Définition :	31
3.5.2	la différence entre un codeur incrémental et un codeur absolu :	32
3.5.3	Fonctionnement d'un codeur incrémental	33
3.6	Carte L298N :[16]	34
3.6.1	Principe de fonctionnement du L298N (pilotage PWM)	34
3.7	Carte de régulation de tension :	37
3.8	Les bouton poussoir (START, stop)	38
3.9	Description du logiciel TIA (Totally Integrated Automation) portal :	38
3.9.1	Le bloc d'organisation OB1	46
3.9.2	Les blocs supplémentaires	47
3.10	Essais pratique :	49
3.11	Conclusion :	51
	Conclusion générale.....	52
	Bibliographie	54

Liste des figures

Figure 1-2 : Principe de fonctionnement de la commande prédictive généralisée.....	5
Figure 2-1 : : Symboles de la machine à courant continu [9].....	14
Figure 2-2 : Constituants d'un MCC.	15
Figure 2-3 : Modèle équivalent de l'induit.....	16
Figure 2-4 : Schéma électrique d'une machine à excitation séparée.	18
Figure 2-5 : graph de courant	21
Figure 2-6 : graph de la vitesse.....	22
Figure 2-7 : graph de la position.....	22
Figure 2-8 : la commande u (v)	22
Figure 2-9 : graph de la vitesse.....	23
Figure 2-10 : graph de position	23
Figure 2-11 : la sortie réelle et prédite pendant la phase d'entraînement	24
Figure 2-12 : Évolution de l'erreur pendant l'entraînement du réseau.....	24
Figure 2-13 : sortie réelle versus prédite	24
Figure 2-14 : Évolution de l'erreur	25
Figure 3-1 : description du système.....	28
Figure 3-2: L'automate programmable S7-1200.....	29
Figure 3-3 HMI siemens KTP700	31
Figure 3-4 schéma d'un pont en H.	35
Figure 3-5 Pont en H les 2 configuration possible	36
Figure 3-6 circuit de régulation du tension.....	37
Figure 3-7 Vue du portal	39
Figure 3-8 Vue du projet.	40
Figure 3-9 Adressage des E/S.....	41
Figure 3-10 Créer un projet	43
Figure 3-11 Configurer un appareil	44
Figure 3-12 : Ajouter un appareil	44
Figure 3-13 Adresse IP	45
Figure 3-14 Tableau de variable.....	46
Figure 3-15 : Ajout d'un nouveau bloc	47
Figure 3-16 Programmation d'un bloc FB	48

Liste des abréviations et symboles

MPC : La Commande Prédicative à base de modèle

GPC : La Commande Prédicative généralisée

PFC : La Commande Prédicative Fonctionnelle

NMPC : La Commande Prédicative non linéaire

NGPC : La Commande non linéaire prédictive généralisée

CARIMA : Controlled Auto- Régressive Intégral Moving Average

MCC : Moteur à Courant Continu

p : le nombre de paires de pôles.

a : le nombre de paires de voies d'enroulement

\emptyset : le flux maximum à travers les spires [Wb].

Ω : la vitesse de rotation [rd/s].

N : nombre de conducteurs actif.

K : constante de la f.é.m.

f.é.m : force electrometric

w : nombre de conducteurs par section.

P_a : la puissance absorbée par le moteur.

P_u : la puissance utile.

K_m : Constante du moteur [$V / \text{rd} \cdot \text{s}^{-1}$].

V_a : Tension aux bornes de l'induit [V].

I_a : Courant aux bornes de l'induit [A].

V_f : Tension aux bornes de l'inducteur [V].

I_f : Courant aux bornes de l'induit [A].

Ω : vitesse de rotation du moteur [rd/s].

Φ : Flux d'excitation [weber].

E_a : Force contre électromotrice [V].

R_a : Résistance aux bornes de l'induit [Ω].

L_a : Inductance de l'induit [H].

R_f : Résistance aux bornes de l'inducteur [Ω].

L_f : Inductance de l'inducteur [H].

θ : Position du rotor.

C_j : Couple d'inertie [N.m].

C_f : Couple de frottement visqueux [N.m].

C_s : Couple de frottement sec [N.m].

C_{em} : Couple magnétique de la machine [N.m].

C_r : Couple résistant sur l'axe moteur [N.m].

J : Moment d'inertie de l'ensemble moteur-charge [Kg.m²].

f : Coefficient de frottement visqueux moteur-charge [N.m. rd/s].

M_{fa} : Inductance mutuelle entre l'enroulement induit et l'inducteur.

Φ_a : Flux magnétique de l'induit [weber].

Φ_f : Flux magnétique de l'inducteur [weber].

API : Automate Programmable Industriel

HMI : Human Machine Interface

Introduction générale

La Commande Prédicative a trouvé une large application dans le domaine industriel, et un grand nombre d'algorithmes d'implémentation ont été présentés dans la littérature tels que la Commande Prédicative Fonctionnelle (Predictive Functional Control : PFC) de Richalet en 1978 et la Commande Prédicative Généralisée (Generalized Predictive Control : GPC) développée par Clarke en 1987. La plupart de ces algorithmes de commande utilisent un modèle du procédé pour prédire le comportement futur du système (Model Predictive Control : MPC).

Nous nous intéressons, ici, à la Commande Prédicative Généralisée qui a montré son efficacité dans plusieurs domaines d'application. La nature non linéaire de la plupart des procédés et des applications industrielles nous oblige à concevoir des versions non linéaires de cette stratégie de commande. L'une des caractéristiques qui a contribué au succès de la technologie (MPC) est celle qui a permis d'anticiper et d'éliminer les effets des perturbations et d'avoir une meilleure poursuite du signal de référence.

Dans ce travail , nous allons contrôler la vitesse d'un moteur à courant continu en utilisation de la commande prédictive généralisée

Dans le chapitre 1 nous avons introduit la commande prédictive généralisée

Dans le chapitre 2 nous avons introduit le moteur à courant continu puis modélisée et faire identification pour la obtention de modèle CARIMA nécessaire pour contrôler la vitesse de MCC.

Dans le chapitre 3 nous avons introduit le matériel utilisé et la description de système , puis faire des essais .

En fin, nous terminons notre travail par une conclusion générale .

Chapitre 1 :

Commande prédictive généralisée

1.1 Introduction

Le concept de base de la commande prédictive consiste à utiliser un modèle pour prédire le comportement du système et à choisir la meilleure décision en termes de coût, tout en respectant les contraintes. Il s'agit d'une technique avancée de commande automatisée utilisée pour contrôler des systèmes industriels complexes. Le principe de cette technique est d'utiliser un modèle dynamique du processus intégré au contrôleur en temps réel pour anticiper le comportement futur du procédé. [1].

1.2 Généralités

La commande prédictive est une technique de commande avancée qui peut être utilisée pour contrôler des systèmes complexes et multivariés pour lesquels un simple régulateur PID est insuffisant. Cette technique est particulièrement intéressante pour les systèmes ayant des retards importants, des réponses inverses et de nombreuses perturbations. Les secteurs industriels tels que les raffineries de pétrole, l'industrie chimique, agroalimentaire, la métallurgie et l'aérospatiale sont les principaux utilisateurs de la commande prédictive.

Les principaux avantages de cette technique sont les suivants :

- L'idée générale est facile à comprendre et intuitive.
- Permet de respecter les contraintes sur les variables contrôlées et manipulées.
- Évite des variations excessives sur les variables manipulées, en conséquence la commande est plus douce, permettant une meilleure utilisation des actionneurs, tels que les vérins, vannes, moteurs, et leur durée de vie augmente.
- En cas d'anxiété mesurable, le système s'adapte automatiquement.
- Elle permet généralement des gains financiers réels.
- Le contrôle prédictif est un terme général qui englobe une variété de méthodes différentes, telles que PFC, DMC, GPC, EPSAC, NLPC, mais toutes ces méthodes partagent la même philosophie et fonctionnent selon le même principe de base.

1.3 Principe :

Le contrôle prédictif est la précision répétée de chaque pas de temps et comment passé de la situation actuelle à l'objectif dans l'optimisation en respectant les contraintes.

Voici les étapes à suivre pour appliquer la commande prédictive :

1. Calculer les prévisions des variables contrôlées sur un horizon de temps N_2 à l'aide du modèle interne.
2. Élaborer une trajectoire de référence à suivre.
3. Calculer la loi de commande future qui sera appliquée aux variables manipulées sur un nouvel horizon temporel N_u .

4. Seul le premier élément de la loi de commande calculée est appliqué au système lors du prochain cycle d'horloge. Toutes ces étapes sont ensuite répétées dans le principe de l'horizon fuyant.

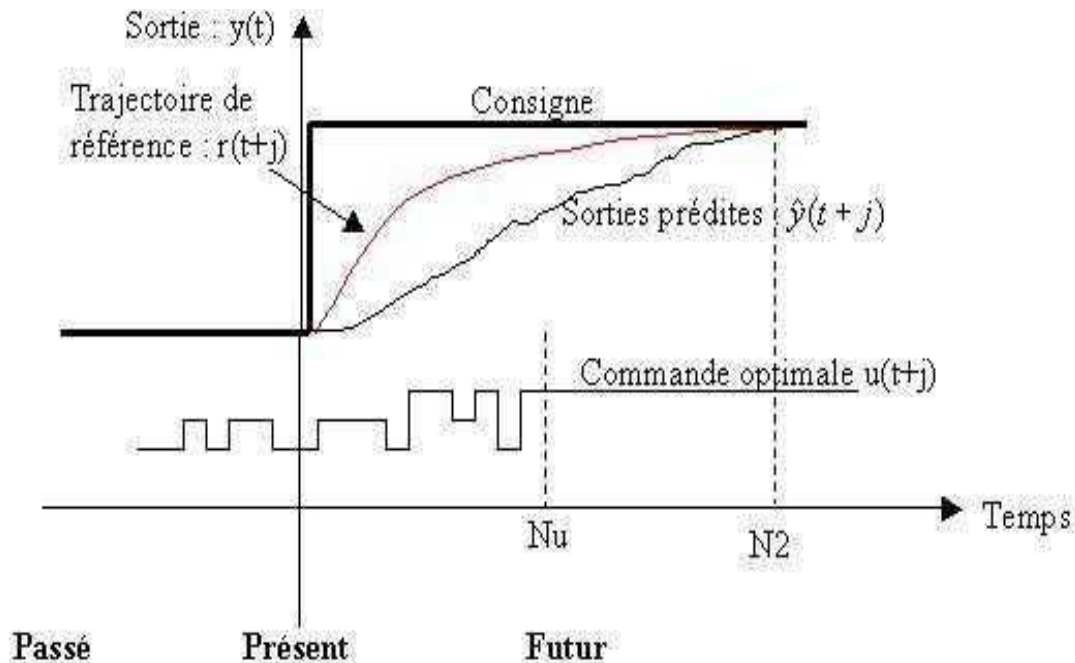


Figure 1-1 : Principe de l'horizon fuyant. [2]

1.4 Types d'algorithmes prédictifs

Les algorithmes prédictifs sont des modèles mathématiques qui permettent d'anticiper les résultats futurs en prenant en compte les résultats passés. Ils permettent d'obtenir des scores prédictifs indiquant la probabilité de répondre de manière favorable à une requête.

Il existe une grande variété d'algorithmes prédictifs, qui fonctionnent tous sur le même principe d'analyse des données et de prédiction. Parmi ces algorithmes, on peut citer :

- La Commande Prédictive à base de modèle (MPC)
- La Commande Prédictive généralisée (GPC)
- La Commande Prédictive Fonctionnelle (PFC)
- La Commande Prédictive non linéaire (NMPC)
- La Commande non linéaire prédictive généralisée (NGPC)

La commande prédictive est une méthode générale qui regroupe plusieurs techniques telles que la PFC, la DMC, la GPC, l'EPSAC et la NLPC. Bien que ces méthodes soient différentes, elles fonctionnent toutes avec la même philosophie de contrôle. Elle est principalement utilisée dans les industries impliquantes.[3]

Dans ce mémoire, nous concentrons sur l'étude de la Commande Prédictive généralisée (GPC) en tant que technique de contrôle pour moteur à courant continu.

1.5 La Commande Prédictive généralisée (GPC)

a été initialement présentée en 1987 par D.W. Clarke et ses collègues. Elle est considérée comme l'une des méthodes de prédiction les plus populaires, en particulier pour les procédés industriels. Elle implique la prédiction du comportement futur du processus ainsi que la commande de rétroaction.[4]

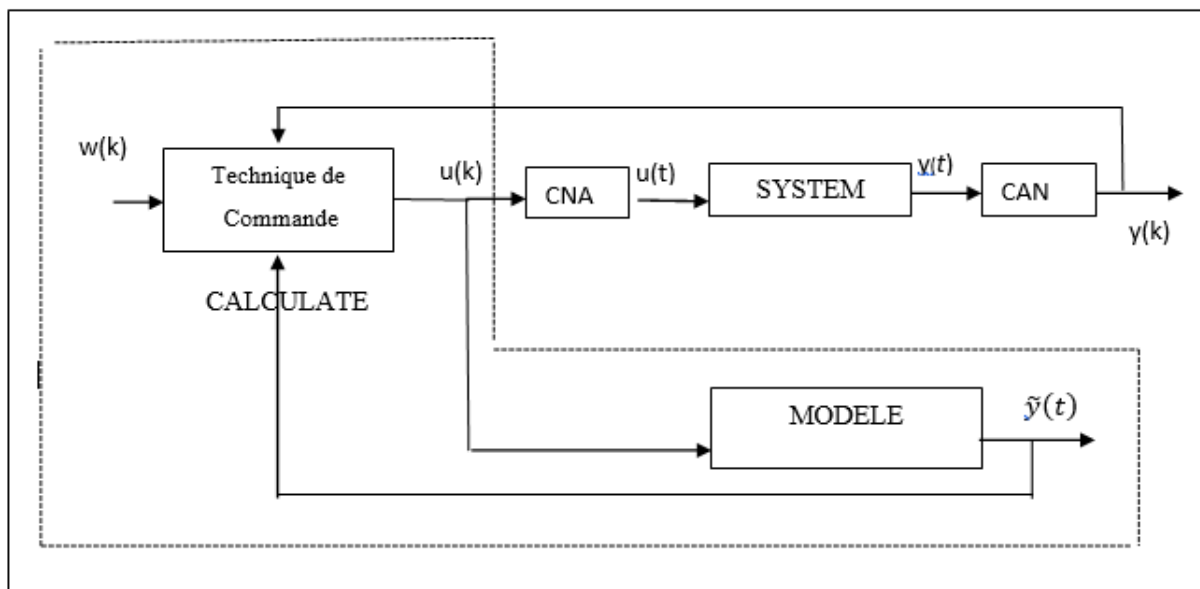


Figure 1-2 : Principe de fonctionnement de la commande prédictive généralisée

Avec :

$w(k)$: consigne.

$y(k)$: sortie du processus.

$\hat{y}(k)$: sortie prédite.

$u(k)$: commande appliqué sur le système.

La commande prédictive nécessite un modèle numérique pour le calcul de la sortie attendue sur horizon limité, elle se base sur les concepts suivants :

- Le modèle de prédiction utilisé est un modèle CARIMA (Controlled Auto- Régressive Intégral Moving Average) qui est une extension du modèle CARMA comportant une action intégrale dans le but d'éliminer l'écart permanent et l'effet des perturbations constantes.
- L'utilisation de la prédiction à horizon étendu.
- La résolution récursive de l'équation de diophantienne
- L'introduction de la pondération sur les incréments de commande à partir duquel tous les incréments sont nuls.

Le choix de l'horizon de commande à partir duquel tous les incréments sont pris nuls. La combinaison de ces concepts, issus de plusieurs algorithmes ont permis l'obtention d'une méthode de commande puissante et performante qui a donné des résultats très satisfaisants dans plusieurs applications industrielles.[5]

1.5.1 Le modèle de prédiction

La commande GPC utilise pour la prédiction du comportement un modèle entrée/sortie par fonction de transfert de type CARIMA (Controlled Auto Regressive Integrated Moving Average) [6] :

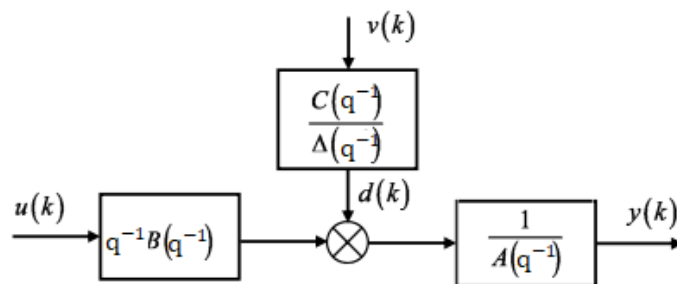


Figure 1-3 Modèle CARIMA

Dans le cas de GPC, le modèle de prédiction est donné par :

$$A(q^{-1}). y(k) = B(q^{-1}). u(k-1) + \frac{C(q^{-1})}{\Delta(q^{-1})} v(k) \quad (1-1)$$

Avec

$y(k)$: Sortie du processus.

$u(k)$: Commande appliquée au système.

$v(k)$: signal de perturbation considéré aléatoire et de moyenne nulle.

q^{-1} : Opérateur retard.

$\Delta(q^{-1})=1-q^{-1}$: Opérateur différence. (1-2)

Les polynômes : $A(q^{-1})$, $B(q^{-1})$, $C(q^{-1})$ sont définis par :

$$A(q^{-1})=1+a_1q^{-1}+\dots\dots\dots a_naq^{-na} \quad (1-3)$$

$$B(q^{-1})=b_0+b_1q^{-1}+\dots\dots\dots b_nbq^{-nb} \quad (1-4)$$

$$C(q^{-1})=1+c_1q^{-1}+\dots\dots\dots c_ncc^{-nc} \quad (1-5)$$

L'utilisation de l'expression mathématique $\Delta(q^{-1})=1-q^{-1}$ dans le modèle du bruit permet une action intégrale dans le correcteur, qui permet de supprimer toute erreur statique par rapport à une consigne ou une perturbation constante. Ce modèle est employé pour produire une séquence de prédictions de la sortie sur plusieurs pas (horizon de prédiction de la sortie), et une séquence d'entrée est ensuite calculée pour minimiser l'écart entre les sorties prédites et les sorties de référence souhaitées. Le principe de commande à horizon glissant est également mis en application.[7]

1.5.2 Structure du prédicteur optimal :

En utilisant le modèle du système, la sortie à l'instant $k+i$ est donnée par :

$$y(k+i) = +\frac{B(q^{-1})}{A(q^{-1})} u(k+i-1) + \frac{C(q^{-1})}{A(q^{-1})\Delta(q^{-1})} v(k+1) \quad (1-6)$$

on introduit la première équation diophantienne suivante :

$$\Delta(q^{-1})A(q^{-1}) Q_i(q^{-1}) + q^{-1} F_i(q^{-1}) = C(q^{-1}) \quad (1-7)$$

$$\deg[Q_i] = i-1 \Rightarrow Q_i(q^{-1}) = q_0 + q_1q^{-1} + \dots + q_{i-1}q^{-(i-1)} \quad (1-8)$$

$$\deg[F_i] = n_f = \max(n_a, n_c - i) \Rightarrow F_i(q^{-1}) = f_{i0} + f_{i1}q^{-1} + \dots + f_{in}q^{-n_f} \quad (1-9)$$

en utilisant cette équation, on obtient :

$$y(k+i) = +\frac{B(q^{-1})}{A(q^{-1})} u(k+i-1) + \frac{\Delta(q^{-1})A(q^{-1}) Q_i(q^{-1}) + q^{-1} F_i(q^{-1})}{A(q^{-1})\Delta(q^{-1})} v(k+1) \quad (1-10)$$

$$y(k+i) = +\frac{B(q^{-1})}{A(q^{-1})} u(k+i-1) + \frac{q^{-1} F_i(q^{-1})}{A(q^{-1})\Delta(q^{-1})} v(k+1) + Q_i(q^{-1})v(k+1) \quad (1-11)$$

Avec $q^{-1}F_i(q^{-1})v(k+1) = F_i(q^{-1})v(k)$, il vient :

$$y(k+i) = +\frac{B(q^{-1})}{A(q^{-1})} u(k+i-1) + \frac{F_i(q^{-1})}{A(q^{-1})\Delta(q^{-1})} v(k) + Q_i(q^{-1})v(k+1) \quad (1-12)$$

a partir des mesures disponibles à l'instant k , on peut se servir du modèle du système pour estimer le terme $v(k)$ comme suit :

$$v(k) = \frac{\Delta(q^{-1})A(q^{-1})}{C(q^{-1})} \left(y(k) - \frac{B(q^{-1})}{A(q^{-1})} u(k-1) \right) \quad (1-13)$$

on obtient alors:

$$y(k+i) = +\frac{B(q^{-1})}{A(q^{-1})} u(k+i-1) + \frac{F_i(q^{-1})}{C(q^{-1})} y(k) - \frac{F_i(q^{-1})B(q^{-1})}{C(q^{-1})A(q^{-1})} u(k-1) + Q_i(q^{-1})v(k+1) \quad (1-14)$$

On à : $u(k-1) = q^{-1} u(k+i-1)$, alors , il vient :

$$y(k+i) = \frac{F(q^{-1})}{C(q^{-1})} y(k) + \frac{B(q^{-1})}{A(q^{-1})C(q^{-1})} (C(q^{-1}) - q^{-1}F_i(q^{-1}))u(k+i-1) + Q_i(q^{-1})v(k+1)$$

De l'équation Diophantienne : $C(q^{-1}) - q^{-1} F_i(q^{-1}) = \Delta(q^{-1})A(q^{-1}) Q_i(q^{-1})$, on alors :

$$y(k+i) = \frac{F(q^{-1})}{C(q^{-1})} y(k) + \frac{B(q^{-1}) Q_i(q^{-1}) \Delta(q^{-1})}{C(q^{-1})} u(k+i-1) + Q_i(q^{-1})v(k+1) \quad (1-16)$$

On à : $\Delta(q^{-1}) u(k+i-1) = (1 - q^{-1}) u(k+i-1) = u(k+i-1) - u(k+i-2) = \Delta u(k+i-1)$

$$y(k+i) = \frac{F(q^{-1})}{C(q^{-1})} y(k) + \frac{B(q^{-1}) Q_i(q^{-1})}{C(q^{-1})} \Delta u(k+i-1) + Q_i(q^{-1})v(k+1) \quad (1-17)$$

On suppose que $Q_i(q^{-1})v(k+1) = 0$ (terme inconnu à l'instant k), alors la prédiction de la sortie à l'instant k+1 avec $i=1, \dots, N_p$ est l'horizon de prédiction , est donnée par :

$$\hat{y}(k+i) = \frac{F_i(q^{-1})}{C(q^{-1})} y(k) + \frac{B(q^{-1}) Q_i(q^{-1})}{C(q^{-1})} \Delta u(k+i-1) \quad (1-18)$$

On introduit une deuxième équation diophantienne :

$$C(q^{-1}) G_i(q^{-1}) + q^{-1} H_i(q^{-1}) = B(q^{-1}) Q_i(q^{-1}) \quad (1-19)$$

$$\deg[G_i] = i-1 \Rightarrow G_i(q^{-1}) = g_0 + g_1 q^{-1} + \dots + g_{i-1} q^{-(i-1)}$$

$$\deg[H_i] = n_h = \max(n_b - 1, n_c - i) \Rightarrow H_i(q^{-1}) = h_{i0} + h_{i1} q^{-1} + \dots + h_{in} q^{-n_h}$$

On à : $\frac{B(q^{-1}) Q_i(q^{-1})}{C(q^{-1})} = G_i(q^{-1}) + \frac{q^{-1} H_i(q^{-1})}{C(q^{-1})}$, ce qui donne :

$$\hat{y}(k+i) = \frac{F_i(q^{-1})}{C(q^{-1})} y(k) + G_i(q^{-1}) \Delta u(k+i-1) + \frac{q^{-1} H_i(q^{-1})}{C(q^{-1})} \Delta u(k+i-1) \quad (1-20)$$

Etant donné que : $q^{-1} \Delta u(k+i-1) = \Delta u(k-1)$, il vient :

$$\hat{y}(k+i) = \underbrace{\frac{F_i(q^{-1})}{C(q^{-1})} y(k) + \frac{H_i(q^{-1})}{C(q^{-1})} \Delta u(k-1)}_{\text{réponse libre (passé)}} + \underbrace{G_i(q^{-1}) \Delta u(k+i-1)}_{\text{réponse forcé (futur)}}, i=1, \dots, N_p \quad (1-21)$$

En commande prédictive, on a toujours la supposition suivante : au-delà de l'horizon de commande, la variation de la commande future est nulle, $\Delta u(k+i) = 0$ pour $i \geq N_u$. C'est-à-dire, la loi de commande est calculée en admettant que le système est remis en chaîne ouverte à partir de l'instant $k+N_u$ (commande en boucle ouverte). Cette contrainte permet de réduire considérablement le volume de calcul.

1.5.3 Forme matricielle de l'équation de prédiction :

En considérant l'ensemble des prédictions pour $i=1, \dots, N_p$, on obtient une séquence de prédictions qui peut être mise sous la forme matricielle suivante :

$$\hat{y}_f = \frac{1}{c(q^{-1})} \bar{F}y(k) + \frac{1}{c(q^{-1})} \bar{H}\Delta u(k-1) + G_f \Delta u_f \quad (1-22)$$

Les vecteurs $\hat{y}_f, \Delta u_f, \bar{F}, \bar{H}$ et la matrice G_f sont définis comme suit :

$$\hat{y}_f = \begin{pmatrix} \hat{y}(k+1) \\ \vdots \\ \hat{y}(k+N_p) \end{pmatrix} \in R^{N_p}, \Delta u_f = \begin{pmatrix} \Delta u(k) \\ \vdots \\ \Delta u(k+N_u-1) \end{pmatrix} \in R^{N_u}, \bar{F} = \begin{pmatrix} F_1(q^{-1}) \\ \vdots \\ F_{N_p}(q^{-1}) \end{pmatrix} \in R^{N_p}$$

$$\bar{H} = \begin{pmatrix} H_1(q^{-1}) \\ \vdots \\ H_{N_p}(q^{-1}) \end{pmatrix} \in R^{N_p}, G_f = \begin{pmatrix} g_0 & 0 & 0 & \dots & 0 \\ g_1 & g_0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots \\ g_{N_u-1} & g_{N_u-2} & g_{N_u-3} & \dots & g_0 \\ \vdots & \vdots & \vdots & \dots & \vdots \\ g_{N_p-1} & g_{N_p-2} & g_{N_p-3} & \dots & g_{N_p-N_u} \end{pmatrix} \in R^{N_p \times N_u}$$

$$\hat{y}_f = G_f \Delta u_f + y_L \text{ avec } y_L = \frac{1}{c(q^{-1})} \bar{F}y(k) + \frac{1}{c(q^{-1})} \bar{H}\Delta u(k-1) \quad (1-23)$$

1.5.4 Résolution récursive des équations diophantiennes :

Le modèle du système est donné par :

$$A(q^{-1}) \cdot y(k) = B(q^{-1}) \cdot u(k-1) + \frac{c(q^{-1})}{\Delta(q^{-1})} v(k) \quad (1-24)$$

$$\begin{aligned} A(q^{-1}) &= 1 + a_1 q^{-1} + \dots + a_n q^{-n} & C(q^{-1}) &= 1 + c_1 q^{-1} + \dots + c_n q^{-n} \\ B(q^{-1}) &= b_0 + b_1 q^{-1} + \dots + b_n q^{-n} & \Delta(q^{-1}) &= 1 - q^{-1} \end{aligned}$$

On considère la première équation diophantienne suivante :

$$\Delta(q^{-1}) A(q^{-1}) Q_i(q^{-1}) + q^{-1} F_i(q^{-1}) = C(q^{-1}) \quad (1-25)$$

$$\deg[Q_i] = i-1 \Rightarrow Q_i(q^{-1}) = q_0 + q_1 q^{-1} + \dots + q_{i-1} q^{-(i-1)}$$

$$\deg[F_i] = n_f = \max(n_a, n_c - i) \Rightarrow F_i(q^{-1}) = f_{i0} + f_{i1} q^{-1} + \dots + f_{in} q^{-n_f}$$

l'algorithme de calcul récursif des polynômes $Q_i(q^{-1})$ et $F_i(q^{-1})$ est donné par :

<p>Pour: $i=1$ $n_f = \max(n_a, n_c - 1)$ $q_0 = 1$ $f_{1j} = c_{j+1} - q_0(a_{j+1} - a_j), j=0, \dots, n_f$</p>	<p>Pour : $i=2, \dots, N_p$ $n_f = \max(n_a, n_c - i)$ $q_{i-1} = f_{(i-1)0}$ $f_{ij} = f_{(i-1)(j+1)} - q_{i-1}(a_{j+1} - a_j), j=0, \dots, n_f$</p>
--	---

On considère maintenant la deuxième équation diophantienne :

$$C(q^{-1}) G_i(q^{-1}) + q^{-1} H_i(q^{-1}) = B(q^{-1}) Q_i(q^{-1}) \quad (1-26)$$

$$\deg[G_i] = i-1 \Rightarrow G_i(q^{-1}) = g_0 + g_1 q^{-1} + \dots + g_{i-1} q^{-(i-1)}$$

$$\deg[H_i] = n_h = \max(n_b - 1, n_c - i) \Rightarrow H_i(q^{-1}) = h_{i0} + h_{i1} q^{-1} + \dots + h_{in} q^{-n_h}$$

L'algorithme de calcul récursif des polynômes $G_i(q^{-1})$ et $H_i(q^{-1})$ est donné par :

Pour $i=1$ $g_0 = q_0 b_0$ $h_{1j} = q_0 b_{j+1} - c_{j+1} g_0, j=0, \dots, n_h$	Pour $i=2, \dots, N_p$ $g_{i-1} = h_{(i-1)0} + q_{i-1} b_0$ $h_{ij} = h_{(i-1)(j+1)} - c_{j+1} g_{i-1} + q_{i-1} b_{j+1}, j=0, \dots, n_h$
--	--

1.5.5 Loi de commande :

La loi de commande est obtenue par minimisation d'un critère de performance. Ce critère a pour expression :

$$J = \sum_{i=1}^{N_p} (y_r(k+i) - \hat{y}(k+i))^2 + \lambda \sum_{i=0}^{N_u-1} \Delta u(k+i)^2 \quad (1-27)$$

Ce critère de performance est composé de deux termes. Le premier terme est l'accumulation, sur l'horizon de prédiction, des carrés des écarts de poursuite. Le deuxième terme représente la somme des carrés des valeurs successives de l'incrément de commande, sur l'horizon de commande, cette somme étant pondérée par un facteur λ . Pour déterminer la loi de commande, on réécrit le critère quadratique sous la forme vectorielle suivante :

$$J = (y_R - \hat{y}_f)^T (y_R - \hat{y}_f) + \lambda \Delta u_f^T \Delta u_f \quad \text{où} \quad y_R = \begin{pmatrix} y_r(k+1) \\ \vdots \\ y_r(k+N_p) \end{pmatrix}$$

Etant donné que : $\hat{y}_f = G_f \Delta u_f + y_L$, on obtient :

$$J = (y_R - G_f \Delta u_f + y_L)^T (y_R - G_f \Delta u_f + y_L) + \lambda \Delta u_f^T \Delta u_f \quad (1-28)$$

$$J = ((y_R - y_L)^T - \Delta u_f^T G_f^T) (y_R - y_L - G_f \Delta u_f) + \lambda \Delta u_f^T \Delta u_f \quad (1-29)$$

$$J = \Delta u_f^T (\lambda I + G_f^T G_f) \Delta u_f - 2 \Delta u_f^T G_f^T (y_R - y_L) + (y_R - y_L)^T (y_R - y_L) \quad (1-30)$$

La minimisation du critère par rapport à Δu_f donne :

$$\frac{\partial J}{\partial \Delta u_f} = 0 \Rightarrow 2(\lambda I + G_f^T G_f) \Delta u_f - 2G_f^T (y_R - y_L) = 0$$

$$\Delta u_f = (\lambda I + G_f^T G_f)^{-1} G_f^T (y_R - y_L) = (\lambda I + G_f^T G_f)^{-1} G_f^T \left(y_R - \frac{1}{c(q^{-1})} \bar{F}y(k) + \frac{1}{c(q^{-1})} \bar{H}\Delta u(k-1) \right)$$

Posons : $G_c = (\lambda I + G_f^T G_f)^{-1} G_f^T$ la matrice de gain et g_c^T la première ligne de cette matrice, la loi de commande à horizon fuyant (glissant) associée s'obtient alors comme suit :

$$\Delta u(k) = g_c^T (y_R - y_L) = g_c^T \left(y_R - \frac{1}{c(q^{-1})} \bar{F}y(k) + \frac{1}{c(q^{-1})} \bar{H}\Delta u(k-1) \right) \quad (1-31)$$

$$u(k) = u(k-1) + g_c^T (y_R - y_L) = u(k-1) + g_c^T \left(y_R - \frac{1}{c(q^{-1})} \bar{F}y(k) + \frac{1}{c(q^{-1})} \bar{H}\Delta u(k-1) \right) \quad (1-32)$$

Bien que l'on obtienne une séquence de commandes $\Delta u_f = [\Delta u(k) \dots \Delta u(k + N_1 - 1)]^T$ en minimisant le critère quadratique, en pratique, seule la première commande $\Delta u(k)$ est appliquée au système. À l'instant de l'échantillonnage suivant $k+1$, on réapplique cette même procédure, c'est-à-dire que l'on recalcule une nouvelle séquence de commande dont seule la première valeur est effectivement appliquée au système, et ainsi de suite (ce qui correspond au principe de la commande à horizon glissant).

1.6 Conclusion :

Dans ce chapitre, nous avons présenté un algorithme de commande prédictive généralisée. Cette commande utilise des formes récursives qui facilitent considérablement son implémentation dans un calculateur numérique. De plus, elle présente un avantage très intéressant qui réside dans l'élimination des effets de toutes perturbations en échelon de grandeur et de durée d'occurrence aléatoires grâce à l'action intégrale du modèle de CARIMA.

Chapitre 2 :

**Modélisation et identification du
Moteur à Courant Continu**

2.1 Introduction :

Une machine électrique tournante est un système électromécanique réversible qui transforme l'énergie électrique en énergie mécanique (fonctionnement moteur) et vice versa (fonctionnement générateur), sans changer la technique utilisée ni le contenu de l'énoncé. Les machines à courant continu sont très utilisées dans les domaines à faible puissance, tels que les jouets et les petites perceuses. Elles sont également largement employées dans l'équipement automobile pour faire fonctionner les essuie-glaces, les ventilateurs, les lève-vitres, les démarreurs, etc. Dans l'industrie, les moteurs à courant continu de puissance moyenne sont utilisés pour les applications à vitesse variable.

L'objectif de ce chapitre est d'introduire quelques notions de base sur les machines à courant continu

2.2 Présentation de la machine à courant continu :

2.2.1 Définition :

Une Machine à courant continu (MCC) est un convertisseur électromécanique permettant la conversion bidirectionnelle d'énergie électrique en énergie mécanique. On distingue deux types de machines à courant continu : les générateurs de courant continu, qui transforment la puissance mécanique en courant électrique continu, et les moteurs à courant continu, qui transforment le courant électrique continu en énergie mécanique .

Symbole de la machine à courant continu :

Les symboles de la machine à courant continu sont représentés par la figure 2.1

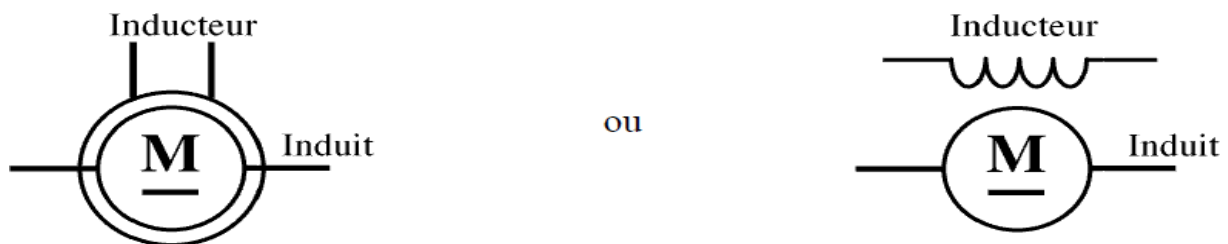


Figure 2-1 : : Symboles de la machine à courant continu [9]

2.2.2 Constitution d'une machine à courant continu :

Un moteur électrique à courant continu est constitué :

- D'un stator qui est à l'origine de la circulation d'un flux magnétique longitudinal fixe créé soit par des enroulements statoriques (bobinage) soit par des aimants permanents à stator, se trouve la partie porte balais et les balais assurant les contacts électriques avec le rotor. Il est aussi appelé inducteur.
- D'un rotor bobiné relié à un collecteur rotatif inversant la polarité dans chaque enroulement rotorique au moins une fois par tour de façon à faire circuler un flux magnétique transversal en quadrature avec le flux statorique. Les enroulements rotoriques sont aussi appelés enroulements d'induits, ou communément induit.

Le schéma de la figure 1.2 montre les parties principales d'une machine à courant continu [10]:

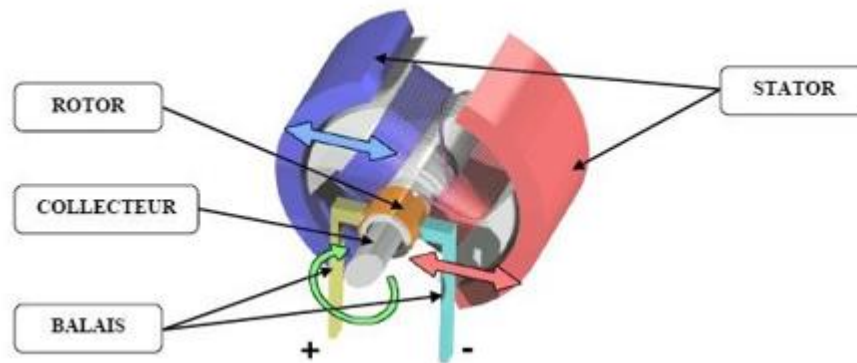


Figure 2-2 : Constituants d'un MCC.

2.3 Les équations du moteur à courant continu :

2.3.1 Force électromotrice :

La machine à courant continu est le siège d'une f.é.m. donnée par [11] :

$$E = \frac{p}{2\pi \cdot a} \cdot N \cdot \Phi \cdot \Omega \quad (2-1)$$

Avec :

p : le nombre de paires de pôles.

a : le nombre de paires de voies d'enroulement

Φ : le flux maximum à travers les spires [Wb].

Ω : la vitesse de rotation [rd/s].

N : nombre de conducteurs actif.

On peut écrire :

$$E = K \cdot \Phi \cdot \Omega \quad (2-2)$$

Avec :

$$K = \frac{p}{2\pi \cdot a} \cdot N \quad (2-3)$$

K : constante de la f.é.m.

2.3.1.1 Modèle équivalent de l'induit :

Si on néglige la chute de tensions aux contacts balais-collecteur et on considère que la réaction magnétique d'induit est totalement compensée, on obtient d'après la loi d'Ohm, avec : $\Phi = \text{Cst}$, $\Omega = \text{Cst}$.

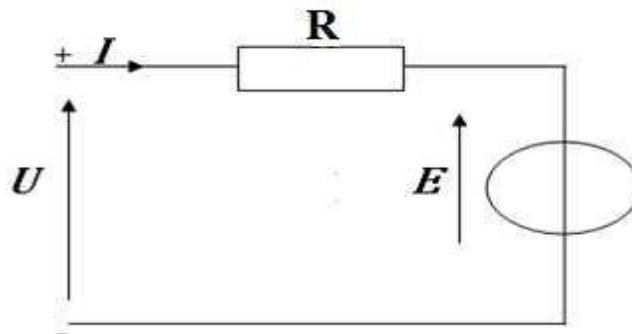


Figure 2-3 : Modèle équivalent de l'induit

$$U = R \cdot I + E \quad (2-4)$$

2.3.1.2 Vitesse du rotor :

A partir de l'expression de la force électromotrice développée aux bornes de l'induit et l'équation électrique, on peut déduire l'expression de la vitesse qui sera donnée par :

$$\Omega = \frac{U - R \cdot I}{K \Phi} \quad (2-5)$$

$$K = \frac{p}{aw} \quad (2-6)$$

Avec :

w : nombre de conducteurs par section.

2.3.1.3 Puissance et couple électromagnétiques :

Si l'induit présente une f.é.m. E et s'il est parcouru par le courant I, il reçoit une puissance électromagnétique :

$$P_{em} = E \cdot I \quad (2-7)$$

D'après le principe de conservation de l'énergie cette puissance est égale à la puissance développée par le couple électromagnétique :

$$P_{em} = C_{em} \cdot \Omega = E \cdot I \quad (2-8)$$

D'où l'expression du couple électromagnétique :

$$C_{em} = K \cdot \Phi \cdot I \quad (2-9)$$

2.3.1.4 Couple utile et puissance utile :

Les pertes mécaniques (frottement dans les paliers, action de l'aire de ventilation) et les pertes fer dans l'induit produisent un couple C_p appelé couple de pertes.

Le couple utile disponible sur l'arbre du moteur sera exprimé par :

$$C_u = C_{em} - C_p \quad (2-10)$$

La puissance utile développée par le moteur est donnée par :

$$P_u = C_u \cdot \Omega \quad (2-11)$$

2.3.1.5 Rendement :

Le rendement du moteur est donné par :

$$\eta = \frac{P_u}{P_a} \quad (2-12)$$

Avec :

P_a : la puissance absorbée par le moteur. Et P_u : la puissance utile.

2.4 Modélisation d'un moteur à courant continu :

La modélisation d'un système électromécanique consiste à écrire des équations mathématiques qui régissent son fonctionnement à partir d'une analyse physique. Cela permet de représenter le système sous forme de fonction de transfert [12].

Dans le cas du moteur à courant continu, d'un point de vue électrique, il peut être modélisé comme un système dont l'entrée est la tension de commande V_a de l'induit (rotor) et la sortie la vitesse de rotation Ω de l'arbre moteur. Le rotor est modélisé par une résistance en série avec une inductance et une force contre-électromotrice (Fig. 1.4). Les caractéristiques électriques, mécaniques et magnétiques du moteur électrique à courant continu régissent ses équations physiques.

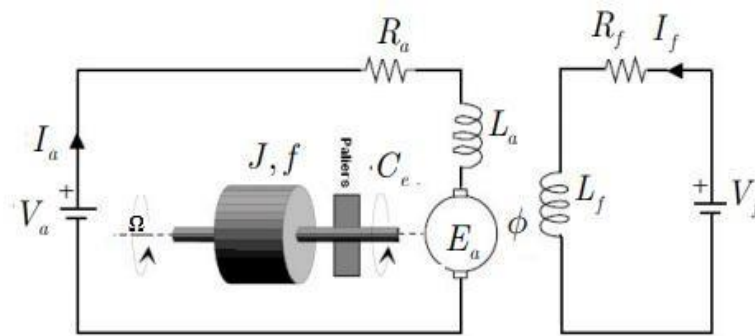


Figure 2-4 : Schéma électrique d'une machine à excitation séparée.

2.4.1 Equations électriques :

En utilise la loi des mailles et la loi d'ohm :

$$L_f \frac{dI_f}{dt} = - R_f \cdot I_f + V_f \quad (2-13)$$

$$V_a = R_a \cdot I_a + L_a \frac{dI_a}{dt} + E_a \quad (2-14)$$

$$E_a = K \cdot \Phi \cdot \Omega \quad (2-15)$$

$$K \cdot \Phi = K_m \quad (2-16)$$

$$\frac{d\theta}{dt} = \Omega \quad (2-17)$$

Avec :

K_m : Constante du moteur [$V / \text{rd} \cdot \text{s}^{-1}$].

V_a : Tension aux bornes de l'induit [V].

I_a : Courant aux bornes de l'induit [A].

V_f : Tension aux bornes de l'inducteur [V].

I_f : Courant aux bornes de l'inducteur [A].

Ω : vitesse de rotation du moteur [rd/s].

Φ : Flux d'excitation [weber].

E_a : Force contre électromotrice [V].

R_a : Résistance aux bornes de l'induit [Ω].

L_a : Inductance de l'induit [H].

R_f : Résistance aux bornes de l'inducteur [Ω].

L_f : Inductance de l'inducteur [H].

θ : Position du rotor.

2.4.2 Equations mécaniques :

L'équation mécanique du moteur est donnée par :

$$C_{em} = C_j + C_f + C_r + C_s \quad (2-18)$$

On néglige les frottements secs et on obtient :

$$C_{em} - C_r = C_j + C_f \quad (2-19)$$

$$C_f = f \cdot \Omega \quad (2-20)$$

$$C_j = J \frac{d\Omega}{dt} \quad (2-21)$$

$$C_{em} = K_m \cdot I_a \cdot \Phi = K_m \cdot I_a \quad (2-22)$$

$$K_m \cdot I_a - C_r = J \frac{d\Omega}{dt} + f \cdot \Omega \quad (2-23)$$

Avec :

C_j : Couple d'inertie [N.m].

C_f : Couple de frottement visqueux [N.m].

C_s : Couple de frottement sec [N.m].

C_{em} : Couple magnétique de la machine [N.m].

C_r : Couple résistant sur l'axe moteur [N.m].

J : Moment d'inertie de l'ensemble moteur-charge [Kg.m²].

f : Coefficient de frottement visqueux moteur-charge [N.m. rd/s].

2.4.3 Equations magnétiques :

Les équations magnétiques du moteur sont données comme suit :

$$\Phi_a = L_a \cdot I_a \quad (2-24)$$

$$\Phi_f = M_{fa} \cdot I_f \quad (2-25)$$

Avec :

M_{fa} : Inductance mutuelle entre l'enroulement induit et l'inducteur.

Φ_a : Flux magnétique de l'induit [weber].

Φ_f : Flux magnétique de l'inducteur [weber].

2.4.4 Représentation d'état d'un moteur à courant continu :

A partir des équations précédentes on peut écrire :

$$\begin{cases} L_a \dot{I}_a = -R_a I_a + K_m \Omega + V_a \\ J \dot{\Omega} = -f \Omega + K_m I_a - C_r \end{cases} \quad (2-26)$$

Donc les équations deviennent sous la forme d'état comme suit:

$$\begin{bmatrix} \dot{I}_a \\ \dot{\Omega} \end{bmatrix} = \begin{bmatrix} -\frac{R_a}{L_a} & -\frac{K_m}{L_a} \\ \frac{K_m}{J} & -\frac{f}{J} \end{bmatrix} \cdot \begin{bmatrix} I_a \\ \Omega \end{bmatrix} + \begin{bmatrix} \frac{1}{L_a} \\ 0 \end{bmatrix} \cdot V_a + \begin{bmatrix} 0 \\ -\frac{1}{J} \end{bmatrix} \cdot C_r \quad (2-27)$$

Avec les variable d'état : $x_1 = I_a$, $x_2 = \Omega$ et $u = V_a$. nous obtenons :

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -\frac{R_a}{L_a} & -\frac{K_m}{L_a} \\ \frac{K_m}{J} & -\frac{f}{J} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \frac{1}{L_a} \\ 0 \end{bmatrix} \cdot u + \begin{bmatrix} 0 \\ -\frac{1}{J} \end{bmatrix} \cdot C_r \quad (2-28)$$

2.5 Simulation :

Dans cette partie nous allons simuler le moteur à courant continu pour obtenir les paramètres de la commande prédictive généralisée pour contrôler la vitesse .

1^{ère} étape :

On définit une fonction contenant le modèle d'état du moteur :

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -\frac{R_a}{L_a} & -\frac{K_m}{L_a} \\ \frac{K_m}{J} & -\frac{f}{J} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \frac{1}{L_a} \\ 0 \end{bmatrix} \cdot u + \begin{bmatrix} 0 \\ -\frac{1}{J} \end{bmatrix} \cdot C_r$$

Sous forme : $\dot{x} = A \cdot x + B \cdot u$

Avec :

$$A = \begin{bmatrix} -\frac{R_a}{L_a} & -\frac{K_m}{L_a} \\ \frac{K_m}{J} & -\frac{f}{J} \end{bmatrix}, B = \begin{bmatrix} \frac{1}{L_a} \\ 0 \end{bmatrix}$$

Les paramètres utilisés du moteur sont :

$$R_a = 0.32 \, \Omega$$

$$L_a = 2.61 \cdot 10^{-3} \, \text{H}$$

$$K_m = 1.05 \, \text{V} / \text{rd} \cdot \text{s}^{-1}$$

$$f = 0.008 \, \text{N} \cdot \text{m} \cdot \text{rd/s}$$

$$J = 0.068 \, \text{Kg} \cdot \text{m}^2$$

Après nous écrivons un programme qui nous donne le changement de la vitesse et de position du moteur comme le montre les figures suivantes :

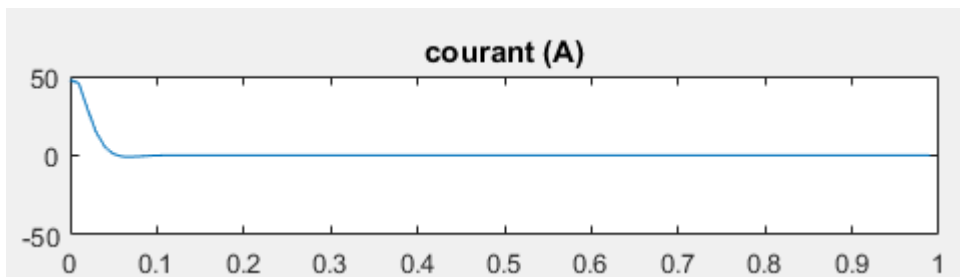


Figure 2-5 : graph de courant

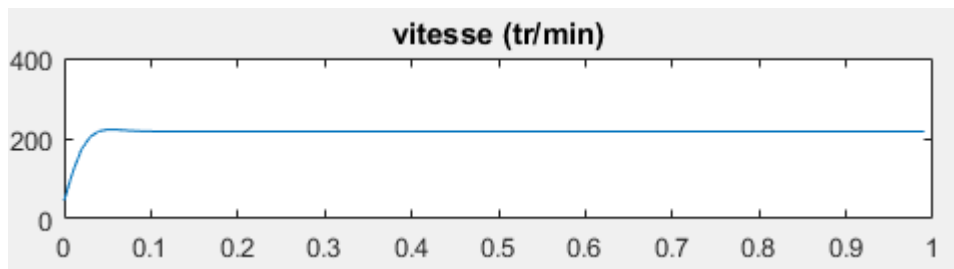


Figure 2-6 : graph de la vitesse

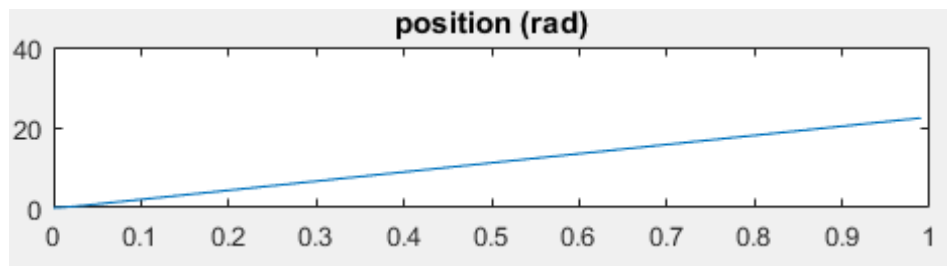
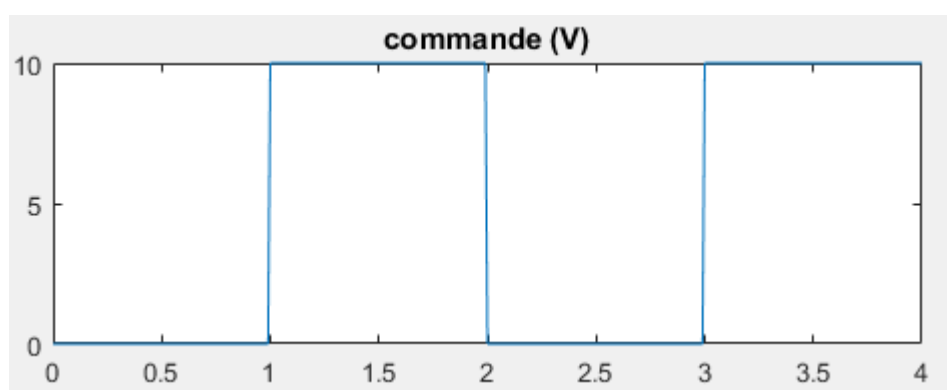


Figure 2-7 : graph de la position

2^{émé} étape :

On applique au programme précédent 400 échantillonnages. Dans les 100 premiers échantillons, la commande u est égale à 0, dans les 100 échantillons suivants, la commande u est égale à 10. Dans les 100 échantillons suivants, la commande u est égale à 0, et dans les 100 derniers échantillons, la commande u est égale à 10.

Figure 2-8 : la commande u (v)

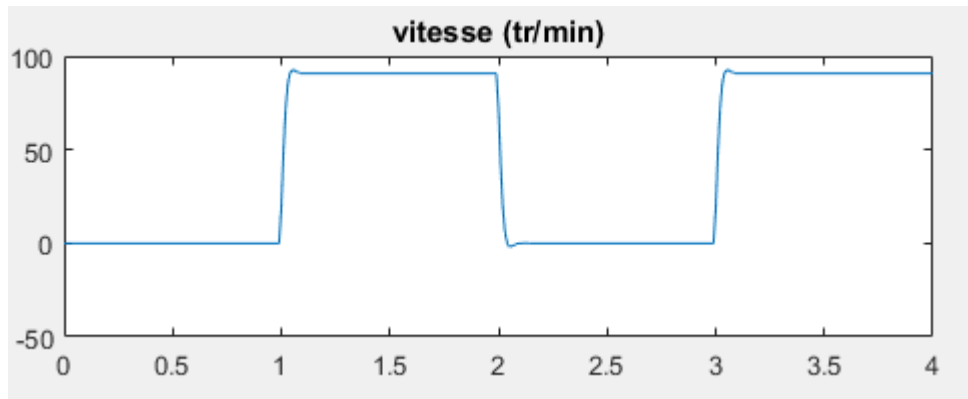


Figure 2-9 : graph de la vitesse

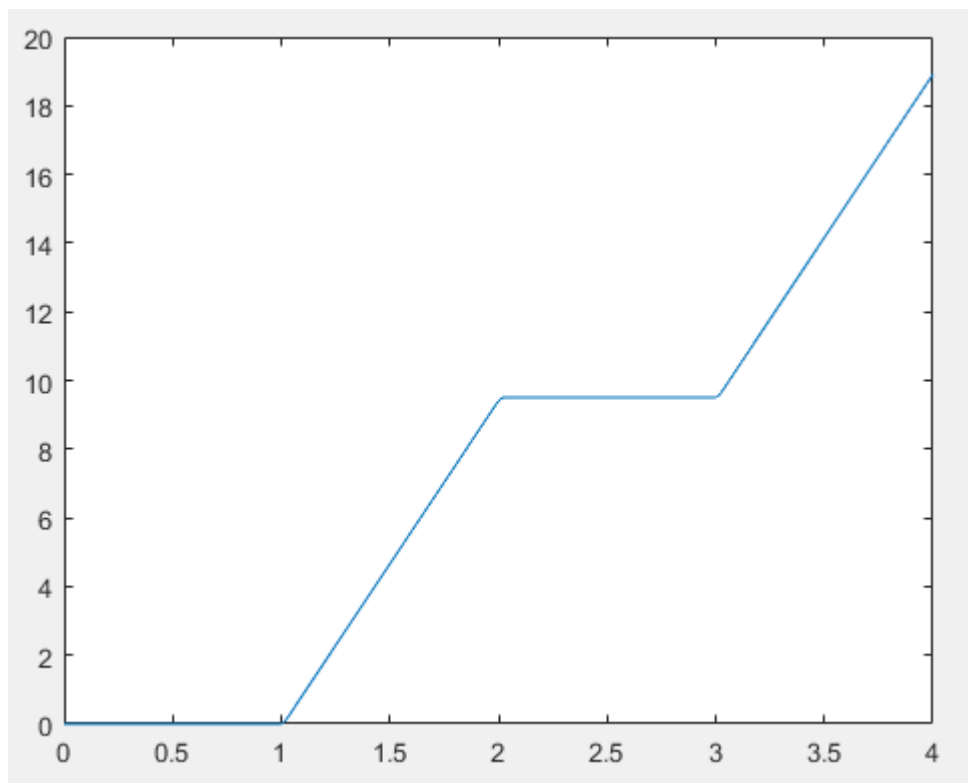


Figure 2-10 : graph de position

Nous stockons les données de programme obtenues dans la base de données .

3^{ème} étape :

Le programme met en œuvre un réseau de neurones pour prédire des valeurs à partir de données en entrée. Le programme charge des données stockées dans un fichier et initialise le réseau de neurones avec des valeurs aléatoires.

La première partie du programme implémente la phase d'entraînement du réseau de neurones. Elle utilise une boucle pour mettre à jour progressivement les poids du réseau de neurones à l'aide de la méthode de descente de gradient. La boucle se poursuit jusqu'à ce que

l'erreur entre les valeurs prédites et les valeurs réelles soit inférieure à une certaine valeur définie .

La deuxième partie du programme concerne la phase de test. Elle utilise les poids entraînés pour prédire des valeurs à partir de nouvelles données et stocke les résultats dans les tableaux .

La troisième partie du programme est une phase de validation utilisant des données stockées dans un fichier. Elle répète essentiellement la même procédure de test en utilisant ces nouvelles données .

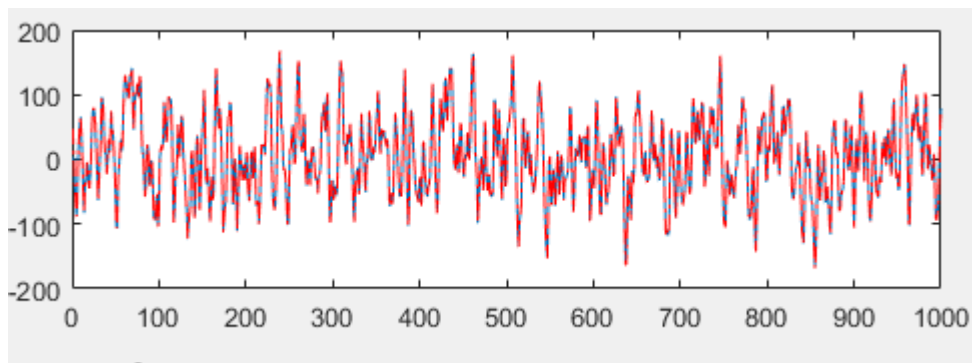


Figure 2-11 : la sortie réelle et prédite pendant la phase d'entraînement

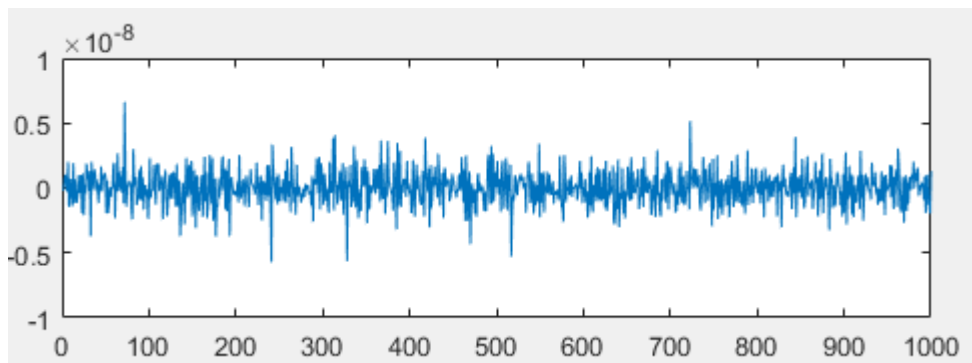


Figure 2-12 : Évolution de l'erreur pendant l'entraînement du réseau

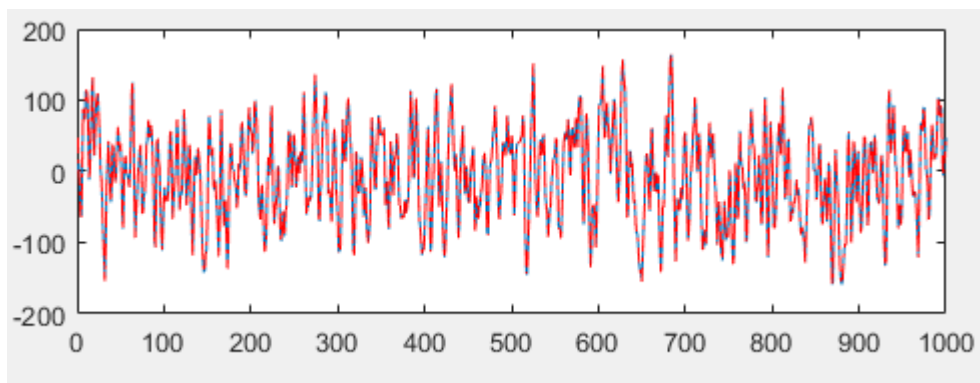


Figure 2-13 : sortie réelle versus prédite

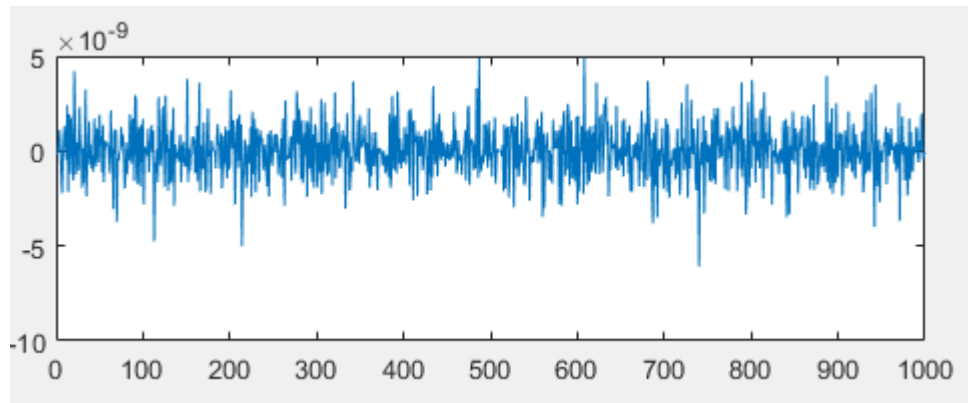


Figure 2-14 : Évolution de l'erreur

Nous obtenons également des paramètres de modèle CARIMA :

$$y(k)[1-0.952246182131270q^{-1}-0.293103317957746q^{-2}]=u(k)[1.861263826991813+1.231503242160023q^{-1}]$$

2.6 Conclusion :

Les moteurs à courant continu sont encore utilisés dans de nombreuses applications de faible à moyenne puissance. L'avantage est qu'il est simple et facile à commander. En raison de son modèle linéaire, la vitesse de rotation d'un moteur à courant continu peut être contrôlée à l'aide d'une méthode de contrôle linéaire.

Dans ce chapitre nous avons modélisé et identifié un moteur à courant continu, puis nous avons simulé le système pour obtenir les paramètres de la commande prédictive généralisée, dont nous avons présenté dans le chapitre précédent, pour contrôler la vitesse du moteur à courant continu.

Chapitre 3 :

Implémentation de la commande et essais pratiques

3.1 Introduction :

Après avoir expliqué la commande prédictive généralisée dans le chapitre 1 et on a fait la modélisation et identification du moteur à courant continu puis simulé pour obtenir un modèle CARIMA nécessaire pour calculer le prédictif optimal que nous utiliserons pour contrôler la vitesse du moteur à courant continu.

Dans ce chapitre, nous présentons le matériel et les différents circuits nécessaires au fonctionnement du système et les résultats des tests effectués.

3.2 Description du système :

L'utilisateur saisit la référence de vitesse via l'HMI, ce dernier donne le signal à l'automate programmable industrielle qui traite les informations selon le programme que nous avons mis en place pour contrôler la vitesse du moteur à courant continu selon la commande prédictive généralisée, puis donne la commande nécessaire.

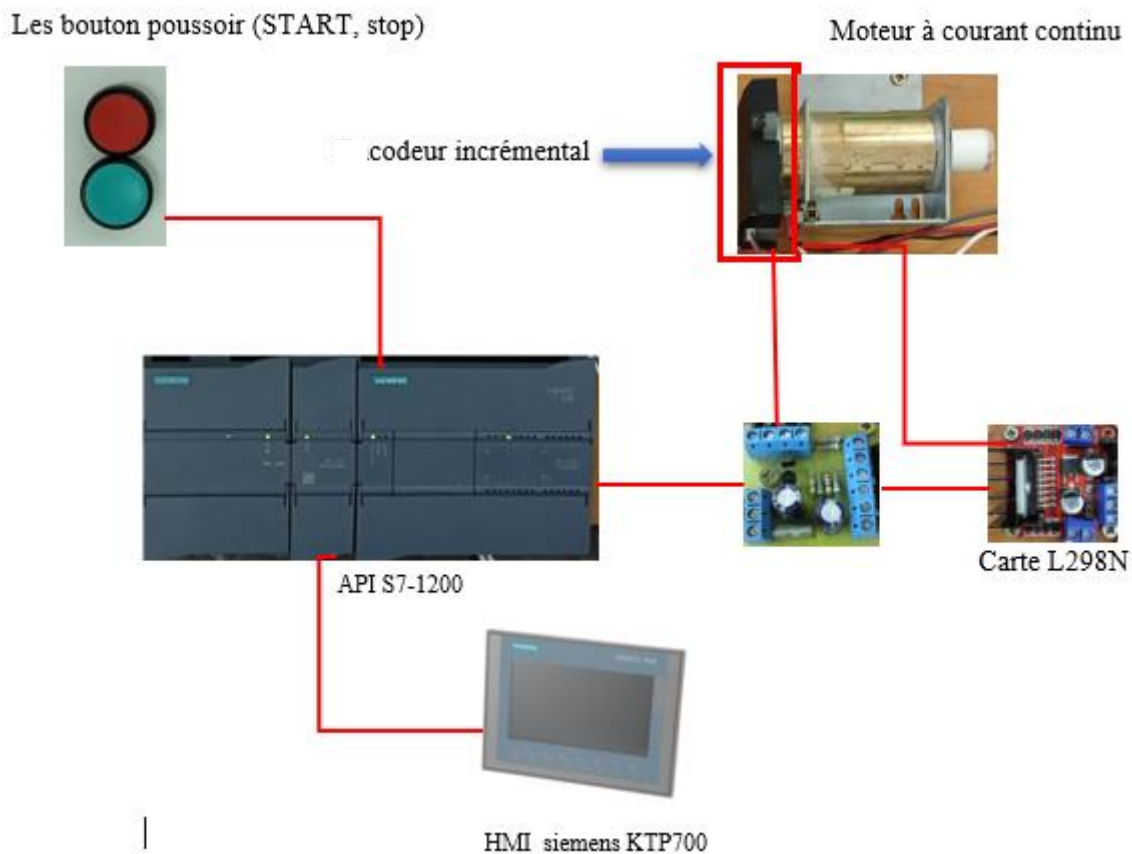


Figure 3-1 : description du système

Figure 3-1 : description du système

3.3 Description de l'automate siemens S7-1200 :

Le contrôleur S7-1200 offre la souplesse et la puissance nécessaires pour commander une large gamme d'appareils afin de répondre à vos besoins en matière d'automatisation. Sa forme compacte, sa configuration souple et son important jeu d'instructions en font une solution idéale pour la commande d'applications très variées [13]

La CPU combine un microprocesseur, une alimentation intégrée, des circuits d'entrée et de sortie, un PROFINET intégré, des E/S rapides de commande de mouvement, ainsi que des entrées analogiques intégrées dans un boîtier compact en vue de créer un contrôleur puissant. Une fois que vous avez chargé votre programme, la CPU contient la logique nécessaire au contrôle et à la commande des appareils dans votre application.

La CPU surveille les entrées et modifie les sorties conformément à la logique de votre programme utilisateur, qui peut contenir des instructions booléennes, des instructions de comptage, des instructions de temporisation, des instructions mathématiques complexes ainsi que des commandes pour communiquer avec d'autres appareils intelligents [14].

La CPU fournit un port PROFINET permettant de communiquer par le biais d'un réseau PROFINET. Des modules supplémentaires sont disponibles pour communiquer via les Réseaux PROFIBUS, GPRS, RS485 ou RS232. Pour ce projet, le S7-1200 sera programmé en l'aide du logiciel TIAPORTAL sous Windows.



Figure 3-2' automate programmable S7-1200

• **Choix de la CPU :** Les CPU du système SIMATIC S7-1200 disposent de quatre classes de puissance : CPU1211 C, CPU1212 C, CPU1214 C et CPU 1215 C, chacune pouvant être étendue en fonction des besoins de la station. Sur chaque CPU, des cartes d'extension peuvent être ajoutées pour ajouter des I/STOR supplémentaires ou analogiques

sans modifier la taille de l'automate. Des modules d'E/S supplémentaires peuvent être ajoutés du côté droit de la CPU pour étendre la capacité d'E/S TOR ou analogiques [15].

3.4 Supervision :

3.4.1 Définition de la Supervision :

La supervision est une technique industrielle de suivie et de pilotage informatique des procédés de fabrication à système automatisés. La supervision est une forme évoluée de dialogue Homme Machine. Elle consiste à surveiller l'état de fonctionnement d'un procédé et des paramètres de commande des processus généralement communiqués à des automates programmables [14].

3.4.2 Constitutions d'un system de supervision :

Les systèmes de supervision se composent généralement d'un moteur central (logiciel), à qui se rattachent des données provenant des équipements (automates). Le logiciel de supervision assure l'affichage, le traitement des données, l'archivage et la communication avec d'autres périphériques. Ayant pour fonction, la mise à la disposition de l'opérateur des données instantanées du procédé. Les modules de visualisation comportent [14]:

- a) Le module d'archivage : Ayant comme rôle la mémorisation des données (alarmes et événements) pendant une longue période et l'exploitation des données dans des applications spécifiques pour les fins de maintenance ou de gestion de production
- b) Le module de traitement : Permet la mise en forme des données afin de les présenter via le module de visualisation aux opérateurs sous une forme prédéfinie
- c) Le module de communication : Ayant pour fonctions l'acquisition, le transfert de données et la gestion de la communication avec les automates programmables industriels et autres périphériques.

Outils de supervision :

Un système de supervision et de contrôle est constitué d'une partie matérielle (automate S7-1200, différents capteur de pression et de température ...etc.) et d'une partie logicielle (traitement et affichage des donnés). La partie matérielle permet de relever les paramètres et d'interagir physiquement avec l'installation, tandis que le logiciel est le cerveau du système .

La programmation et la mise en marche d'une installation industrielle automatisée ne sont pas suffisantes, il est donc nécessaire de visualiser l'état et le mode de fonctionnement de l'installation. Il existe plusieurs configurations d'interface de contrôle/commande. La configuration la plus simple est de rassembler toutes les informations sur une Interface Homme/Machine, pour faciliter la tâche de l'opérateur [14].



Figure 3-3 HMI siemens KTP700

3.5 Codeur incrémental [15]

3.5.1 Définition :

Les codeurs incrémentaux sont des capteurs électroniques fonctionnant généralement sur un principe rotatif utilisé pour la mesure et le contrôle de la vitesse ou de la position d'un objet en utilisant des signaux électriques.

Il existe deux types de codeurs incrémentaux :

- Les codeurs à contacts, nécessitant un lien physique avec le mouvement et l'utilisation de roulements à billes pour capter le mouvement et le transformer en signaux électriques
- Les codeurs sans contact, reposant sur l'utilisation d'un aimant solidaire du mouvement d'un côté et d'une partie électronique dans un boîtier séparé et fixe.

Loin d'être limité à la mesure de distance, les codeurs incrémentaux sont également utilisés pour mesurer la vitesse d'un objet en comptant le nombre de signaux électriques produits par le capteur par unité de temps (fréquence). Cela permet de contrôler la vitesse de mouvement d'un objet avec une précision élevée.

Les codeurs incrémentaux permettent également de mesurer la position angulaire d'un élément rotatif en comptant le nombre d'impulsions électriques générées par le capteur pour chaque incrément d'angle. Tout comme pour le contrôle de la vitesse, il est possible de contrôler la rotation d'un objet avec une précision élevée.

3.5.2 la différence entre un codeur incrémental et un codeur absolu :

Les codeurs incrémentaux et les codeurs absolus permettent la mesure de plusieurs variables, mais reposent sur des fonctionnements et des caractéristiques différents. Les codeurs incrémentaux génèrent des signaux impulsions électriques à chaque incrément de rotation ou de mouvement, tandis que les codeurs absolus génèrent un code unique pour chaque position absolue de l'objet.

En effet, les codeurs incrémentaux génèrent des impulsions électriques à chaque incrément de rotation ou de mouvement de l'objet, permettant ainsi une mesure précise de la distance, de la position ou de la vitesse de l'objet. Les signaux électriques transmis par les codeurs incrémentaux peuvent être utilisés pour :

Contrôler la vitesse de mouvement d'un objet ;

Compter les incréments de rotation ;

Synchroniser des mouvements multiples ;

Détecter des erreurs de mouvement.

Les codeurs absolus, quant à eux, génèrent un code unique pour chaque position angulaire de l'objet, c'est-à-dire qu'il donne une valeur absolue de position de l'objet sur 360°, plutôt que des incréments de position. Cela signifie que les codeurs absolus ne nécessitent pas un référencement initial avant de commencer à fonctionner et qu'ils conservent leur position même si l'alimentation électrique est coupée. C'est pourquoi ils sont utilisés dans les applications qui requièrent une valeur absolue de la position sans prise de référence préalable.

En d'autres termes, les deux types de codeurs permettent la mesure de plusieurs variables, mais reposent sur des fonctionnements et des caractéristiques différents.

3.5.3 Fonctionnement d'un codeur incrémental

3.5.3.1 Composition du codeur incrémental

Les codeurs incrémentaux se composent généralement de deux parties :

- Un rotor (élément tournant), qui est relié à l'objet dont on souhaite mesurer la vitesse, la distance ou la position ;
- Un stator (élément fixe), qui contient la partie électronique de du capteur.

Dans le cadre d'un codeur à contact, le rotor est solidaire d'un disque ajouré, et le stator contient le système de lecture du disque ajouré. Lorsque le rotor tourne, les parties ajourées et pleines passent devant le système de lecture, générant ainsi les impulsions électriques qui peuvent être utilisées pour mesurer la vitesse, la distance ou la position de l'objet.

En ce qui concerne les codeurs sans contact, le champ magnétique de l'aimant permet de connaître l'information de position entre le rotor et le stator et ainsi de générer les impulsions électriques correspondantes. Le rotor est équipé d'un aimant permanent, et le stator contient des capteurs à effet Hall qui détectent les variations de champ magnétique produites par la rotation de l'aimant et les transforment en signaux électriques pour les mesures de vitesse ou de position.

3.5.3.2 Critères de choix d'un codeur incrémental

Le choix d'un codeur incrémental repose essentiellement sur 4 points :

- **La résolution** : La résolution d'un codeur incrémental se rapporte au nombre de d'impulsions électriques générées par le capteur pour une rotation complète de 360° ou pour une valeur de déplacement linéaire. Plus la résolution est élevée, plus la mesure est précise. La résolution est généralement exprimée en nombre d'impulsions électriques par tour ou par unité de distance.
- **L'interface de communication** : Les codeurs incrémentaux proposent plusieurs types d'interfaces électriques. Les impulsions peuvent être sur une base 5V (appelée sortie TTL ou RS422) ou sur une base 24V (appelée sortie HTL ou Push/Pull). Le choix du type de sortie dépend de l'application et du type d'entrée au niveau de l'automate. Il est aussi possible de disposer d'une sortie sinusoïdale dite Sin/Cos. Plus rarement utilisée, elle présente des avantages dans certaines

applications notamment ou il est nécessaire de vérifier la cohérence des signaux transmis par le codeur.

- **La programmation :** Certaines gammes de codeurs incrémentaux peuvent être programmées. Cela apporte une grande flexibilité et adaptabilité avec la possibilité de configurer divers paramètres tels que la résolution par tour, le type de sortie, etc....
- **L'interface mécanique :** pour s'adapter à l'installation, aux caractéristiques mécaniques du moteur, les codeurs incrémentaux existent sous différentes formes : arbre creux, arbre plein, axe traversant ou non traversant...

3.6 Carte L298N :[16]

3.6.1 Principe de fonctionnement du L298N (pilotage PWM)

Pour comprendre le fonctionnement du L298N, il faut tout d'abord bien comprendre qu'il y a 2 aspects à considérer :

- D'une part le fait qu'on peut faire tourner les moteurs dans un sens, ou dans l'autre (ce qui est permis par les « Ponts en H »)
- Et d'autre part, le fait qu'on peut contrôler la vitesse de rotation d'un moteur, grâce à un signal PWM envoyé sur sa broche de commande

Ce sont ces 2 aspects qui font que le L298N peut à la fois contrôler le sens de rotation des moteurs, ET, faire varier leur vitesse.

Les « Ponts en H »

Les Ponts en H ne sont en fait qu'un nom donné à une configuration particulière, de composants électroniques. Classiquement, ceux-ci sont composés de 4 transistors, disposés d'une certaine manière, formant visuellement la lettre « H ». En voici un exemple, avec quatre transistors bipolaires, comme ceux qu'on trouve en interne, dans le L298N :

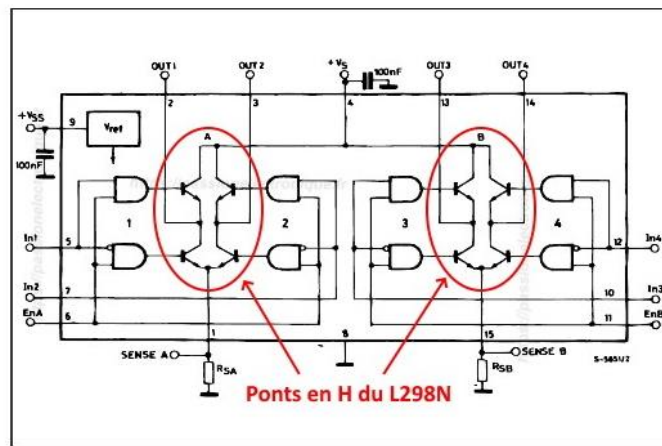
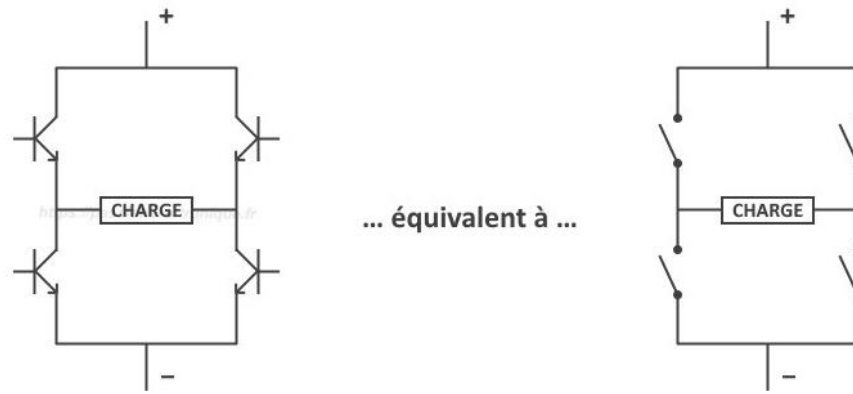


Figure 3-4 schéma d'un pont en H.

Le principe d'un pont en H est super simple : en activant les transistors deux par deux (ceux de sens opposés), on peut contrôler le sens du passage du courant dans la charge, branchée « au milieu du H ». Et c'est ce changement de sens de courant, qui permet aux moteurs à courant continu de pouvoir tourner « à l'endroit », ou « à l'envers ». Visuellement, voici comment cela se passe :

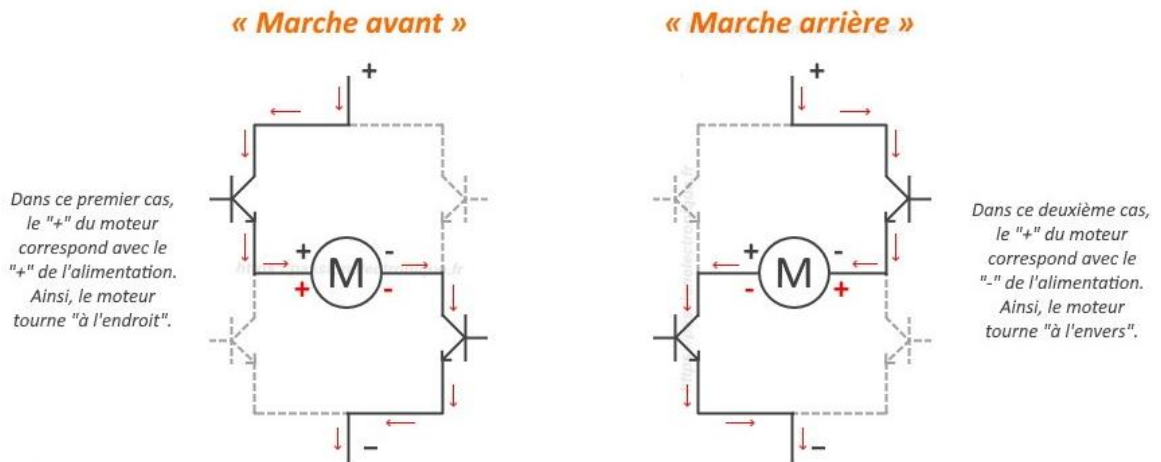


Figure 3-5 Pont en H les 2 configuration possible

Comme vous le voyez, cette configuration à 4 transistors permet :

- Dans le 1er cas : le « + » du moteur sur le « + » de l'alimentation, et le « - » du moteur sur le « - » de l'alimentation
- Dans le 2ème cas : le « + » du moteur sur le « - » de l'alimentation, et le « - » du moteur sur le « + » de l'alimentation (faisant ainsi tourner le moteur en sens inverse)

En bref, en commandant simplement 2 des 4 transistors de manière symétrique, on peut facilement faire passer le courant dans un sens, ou bien dans l'autre. Et c'est d'ailleurs exactement le principe utilisé dans le L298N, pour faire changer le sens de rotation des moteurs électriques.

Remarque : le module L298N comporte 2 ponts en H, totalement indépendants. On peut utiliser un seul de ces ponts, si on le souhaite, tout comme les coupler ensemble, pour pouvoir délivrer deux fois plus de courant

Le pilotage PWM :

Seconde force du L298N : pouvoir être piloté via un signal à rapport cyclique variable (PWM), pour pouvoir faire varier la vitesse des moteurs. En fait, le principe est tout simple : la vitesse d'un moteur à courant continu est proportionnelle à sa tension d'alimentation, ou plus exactement, à sa tension moyenne. Ainsi, il suffit de faire varier la tension moyenne de l'alimentation fournie à un moteur, pour en faire varier sa vitesse de rotation selon ses désirs. Et pour faire varier une tension moyenne, rien de tel que la modulation en largeur d'impulsion (ou PWM, en anglais !).

En effet, en PWM, on génère un signal de fréquence fixe, et de rapport cyclique variable. Et mathématiquement, la tension moyenne sera tout simplement égale à la tension max, multipliée par ce rapport cyclique.

Du coup, pour faire varier la vitesse d'un moteur CC, il suffit d'envoyer un signal PWM au L298N. Ainsi, chaque « pont en H » hachera la tension d'alimentation des moteurs à ce rythme, et la vitesse de rotation des moteurs sera parfaitement contrôlée.

Bien évidemment, comme le L298N possède 2 ponts en H, il faudra fournir deux signaux PWM.

3.7 Carte de régulation de tension :

La figure suivante représenté circuit de régulation de tension :

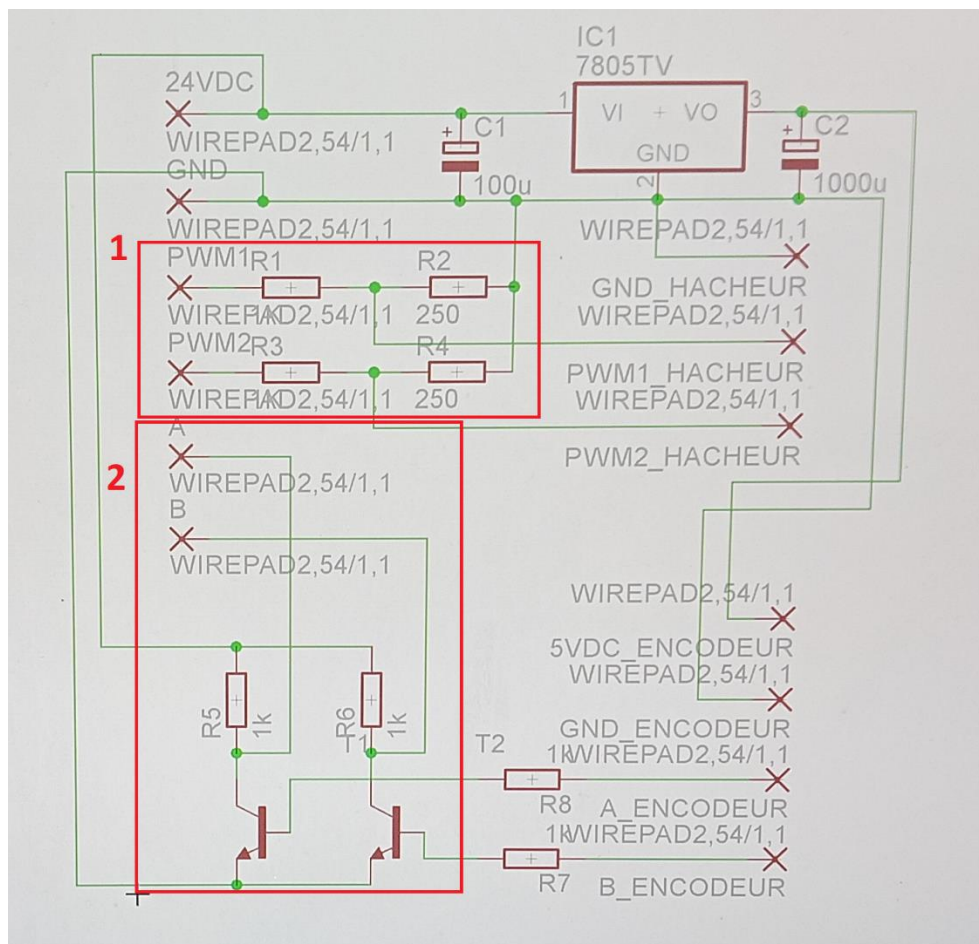


Figure 3-6 circuit de régulation du tension

L'API fonctionne sur tension 24V, alors pour contrôler MCC l'API envoie des signaux (PWM1 & PWM2) tension 24v mais la carte L298N cela fonctionne avec tension 5V, nous avons donc utilisé un diviseur de tension pour réduire la tension (cadre 1)

le codeur incrémentale donner des signaux (A & B) sur tension 5 V , pour que l'API fonctionne les signaux A ,B doivent être à une tension de 24V , nous avons donc utilisé 2 transistor bipolaire (cadre 2)

3.8 Les bouton poussoir (START, stop)

Dans notre projet, nous avons utilisée deux boutons poussoir pour commander notre système. Un bouton a contact normalement ouvert (NO) pour la commande : START
Un bouton a contact normalement fermé (NC) pour la commande : stop

3.9 Description du logiciel TIA (Totally Integrated Automation) portal :

La plateforme Totally Integrated Automation Portal est le nouvel environnement de travail Siemens qui permet de mettre en œuvre des solutions d'automatisation avec un système d'ingénierie intégré [9].

Vue du portail et vue du projet

Lorsque l'on lance TIA Portal, l'environnement de travail se décompose en deux types de vue [10] :

- **La vue du portail** : Elle est axée sur les tâches à exécuter et sa prise en main est très rapide.
- **La vue du projet** : Elle comporte une arborescence avec les différents éléments du projet. Les éditeurs requis s'ouvrent en fonction des tâches à réaliser. Données, paramètres et **Vue du portail**

Chaque portail permet de traiter une catégorie de tâche (actions).

La fenêtre affiche la liste des actions pouvant être réalisées pour la tâche sélectionné

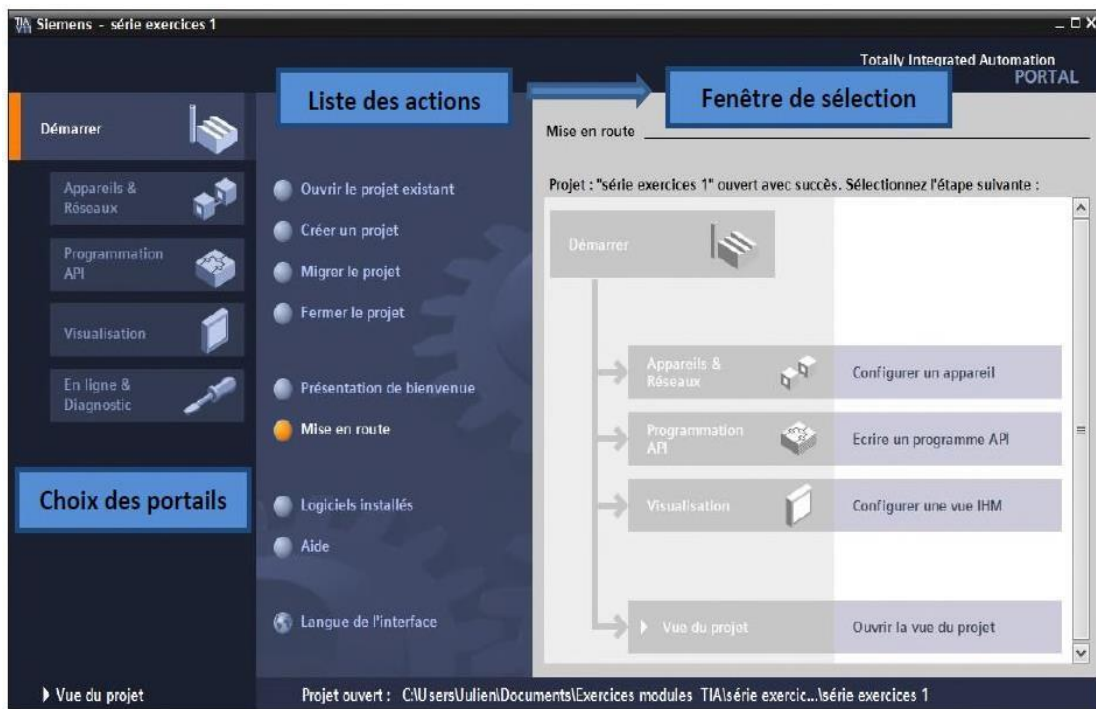


Figure 3-7 Vue du portal.

Vue du projet

L'élément « Projet » contient l'ensemble des éléments et des données nécessaires pour mettre en œuvre la solution d'automatisation souhaitée.

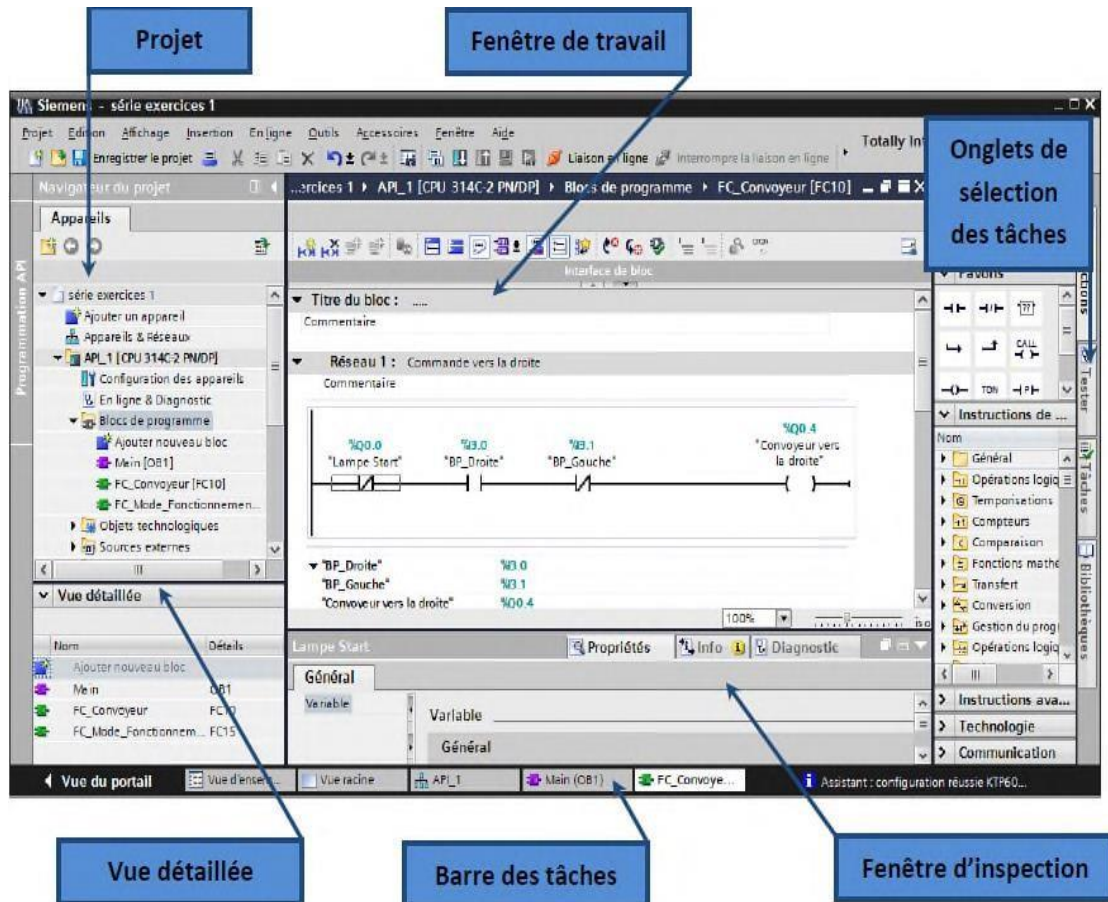


Figure 3-8 : Vue du projet.

La **fenêtre de travail** permet de visualiser les objets sélectionnés dans le projet pour être traités. Il peut s'agir des composants matériels, des blocs de programme, des tables des variables, des HMI,...

La **fenêtre d'inspection** permet de visualiser des informations complémentaires sur un objet sélectionné ou sur les actions en cours d'exécution (propriété du matériel sélectionné, messages d'erreurs lors de la compilation des blocs de programme,...).

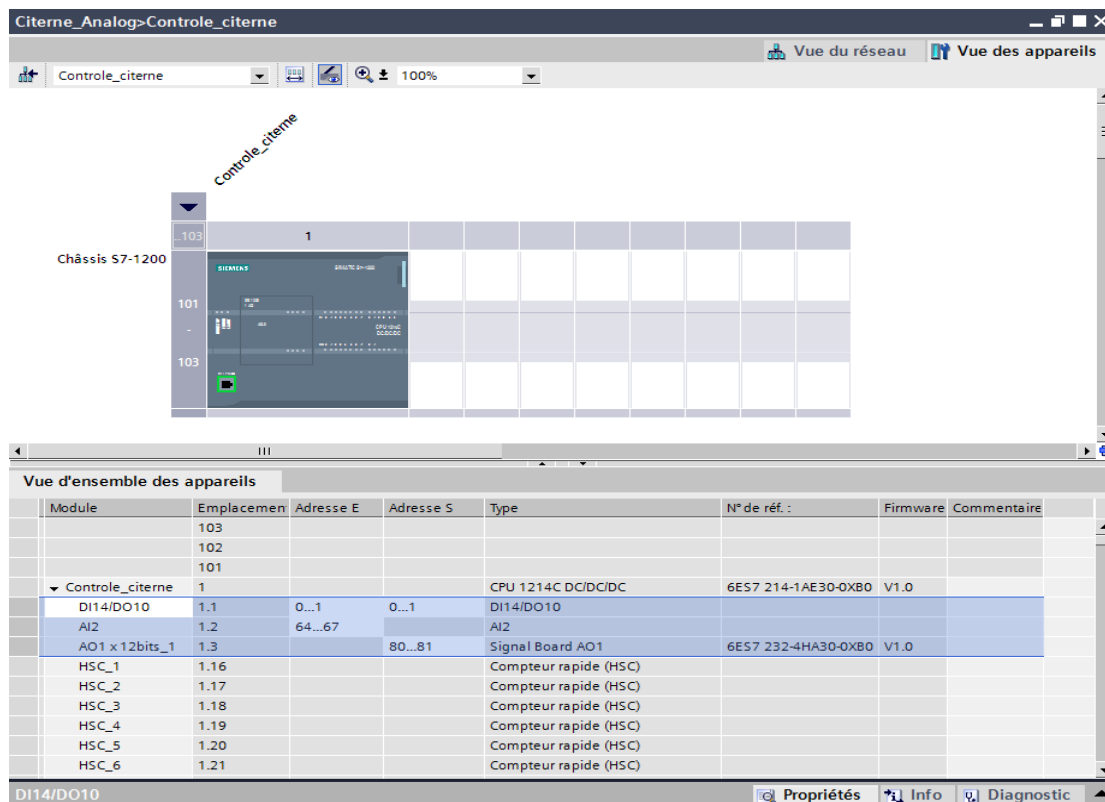
Les **onglets de sélection de tâches** sont un contenu qui varie en fonction de l'objet sélectionné (configuration matérielle, bibliothèque des composants, bloc de programme, instructions de programmation).

Cet environnement de travail contient énormément de données. Il est possible de masquer ou réduire certaines de ces fenêtres lorsque l'on ne les utilise pas.

Il est également possible de redimensionner, réorganiser, désancrer les différentes fenêtres.

Adressage des E/S.

Pour connaître l'adressage des entrées et sorties présentes dans la configuration matérielle, il faut aller dans « *appareil et réseau* » dans le navigateur du projet. Dans la fenêtre de travail, on doit s'assurer d'être dans l'onglet « *Vue des appareils* » et de sélectionner l'appareil voulu [12].



Module	Emplacement	Adresse E	Adresse S	Type	N° de réf. :	Firmware	Commentaire
	103						
	102						
	101						
▼ Controle_citerne	1			CPU 1214C DC/DC	6ES7 214-1AE30-0XB0	V1.0	
DI14/DO10	1.1	0...1	0...1	DI14/DO10			
AI2	1.2	64...67		AI2			
AO1 x 12bits_1	1.3		80...81	Signal Board AO1	6ES7 232-4HA30-0XB0	V1.0	
HSC_1	1.16			Compteur rapide (HSC)			
HSC_2	1.17			Compteur rapide (HSC)			
HSC_3	1.18			Compteur rapide (HSC)			
HSC_4	1.19			Compteur rapide (HSC)			
HSC_5	1.20			Compteur rapide (HSC)			
HSC_6	1.21			Compteur rapide (HSC)			

Figure 3-9 Adressage des E/S.

III.1.2 Les variables API Adresses symbolique et absolue

Dans TIA Portal, toutes les variables globales (entrées, sorties, mémentos,..) possèdent une **Adresse symbolique** et une **adresse absolue**.

- 1) L'**adresse absolue** représente l'identificateur d'opérande (I, Q, M,...) et son adresse et numéro de bit.

- 2) **L'adresse symbolique** correspond au nom que l'utilisateur a donné à la variable (ex : Bouton Marche). Le lien entre les adresses symbolique et absolue se fait dans **la table des variables API**.

Lors de la programmation, on peut choisir d'afficher les adresses absolues, symboliques ou encore les deux simultanément.

III.1.3 Table des variables API

C'est dans la table des variables API que l'on va pouvoir déclarer toutes les variables et les constantes utilisées dans le programme. Lorsque l'on définit une variable API, il faut définir

- Un nom : c'est l'adressage symbolique de la variable.
- Le type de donnée : BOOL, INT,...
- L'adresse absolue : par exemple Q1.5

On peut également insérer un commentaire qui nous renseigne sur cette variable. Le commentaire peut être visible dans chaque réseau utilisant cette variable.

III.1.4 Configuration de l'API S7-1200

Les étapes ci-dessous montrent comment créer un projet pour SIMATIC S7-1200 est

1. Les programmes pour SIMATIC S7-1200 sont gérés sous forme de projets. Nous allons maintenant créer un nouveau projet via la vue portail (« **Créer un projet** > **Nom : Citerne_Analog**> **Créer** »).

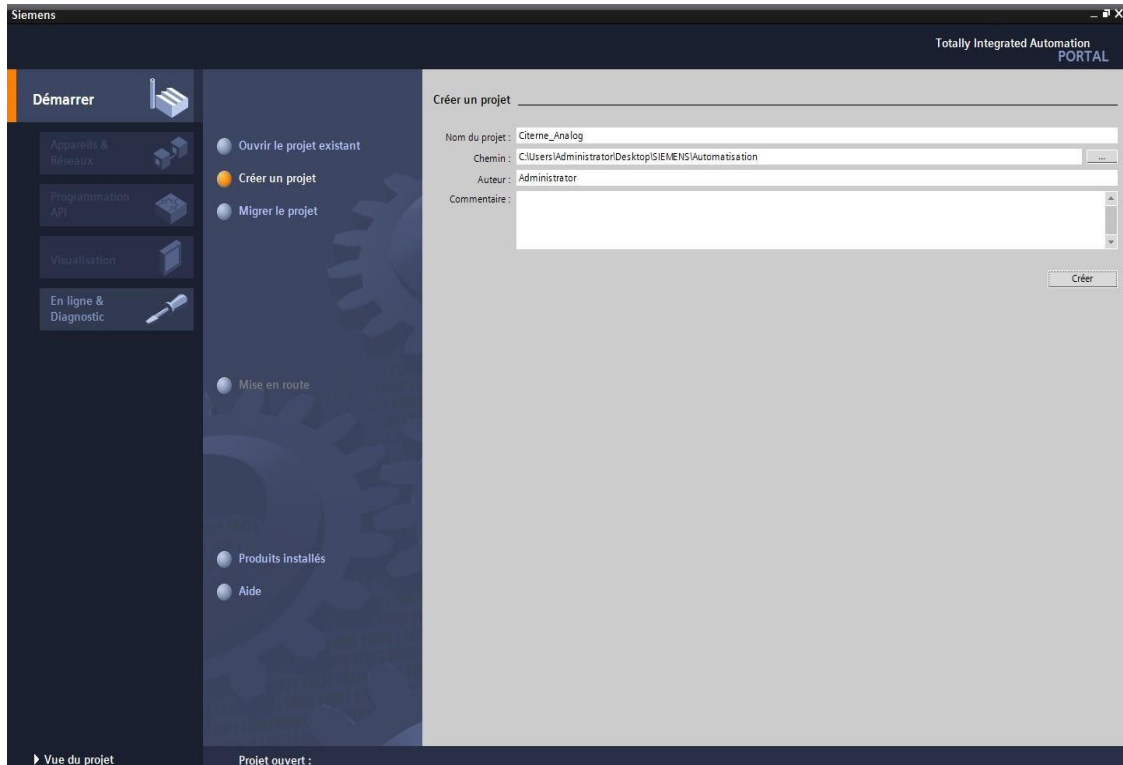


Figure 3-10 Créer un projet

1. « Mise en route » est recommandée pour le début de la création du projet. Premièrement, nous voulons « **Configurer un appareil** » (« **Mise en route** > **Configurer un appareil** »).

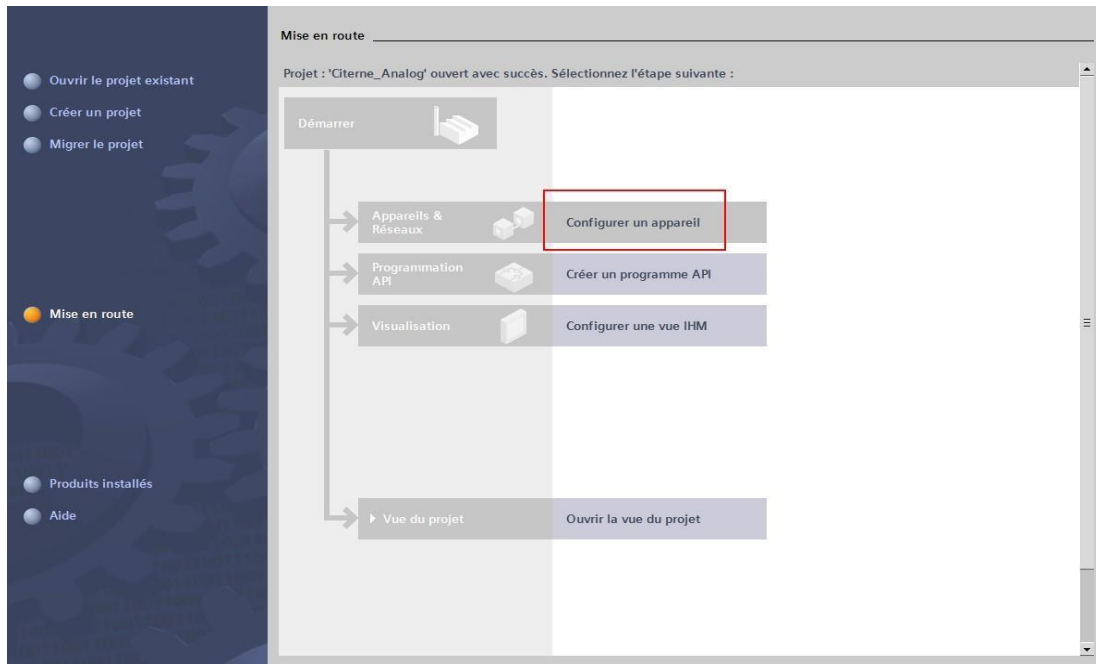


Figure 3-2 : Configurer un appareil

Figure III.5. Configurer un appareil

1. Puis « **Ajouter un appareil** » avec le nom d'appareil : *Controle_citerne*. Choisissez alors dans le catalogue la « **CPU 1214DCDCDC** » avec la bonne combinaison de lettres derrière. « **Ajouter un appareil** > SIMATIC PLC > CPU 1214DCDCDC > 6ES7 214-1AE30-0XB0>**Ajouter** »)

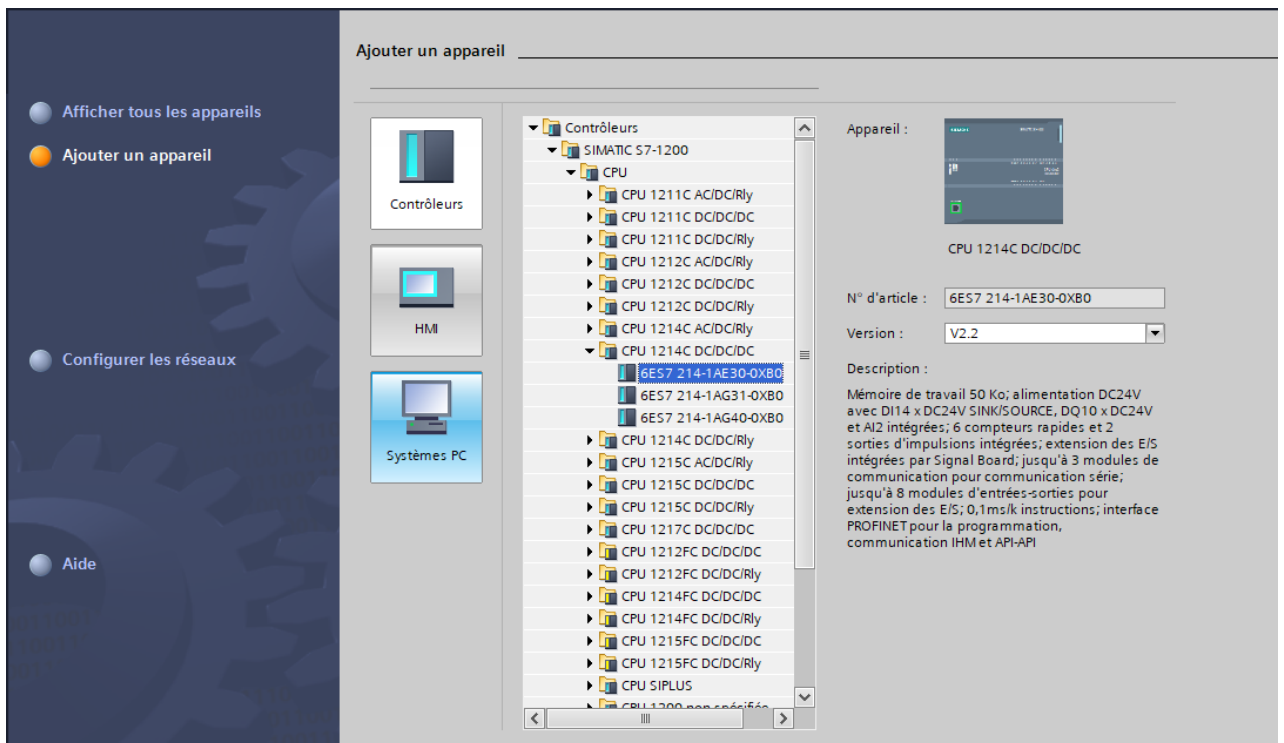


Figure 3-11: Ajouter un appareil

- 2 Le logiciel bouge automatiquement vers la vue du projet avec la configuration matérielle ouverte. Ici, on peut ajouter des modules supplémentaires depuis le **Catalogue du matériel** (fenêtre de droite). A l'aide d'un glisser-déposer on ajoute le Signal Board pour une sortie analogique. (« **Catalogue > Signal Board> A01x12bits > 6ES7 232-4HA30-0XB0** »)
- 3 Afin que le logiciel puisse accéder dans la suite à la bonne CPU, son adresse IP et le masque de sous-réseau doivent être paramétrés (« **Propriétés > Général > Interface PROFINET > Adresses Ethernet > Adresse IP : 192.168.0.1 et Masq. s/rés. : 255.255.255.0** »).

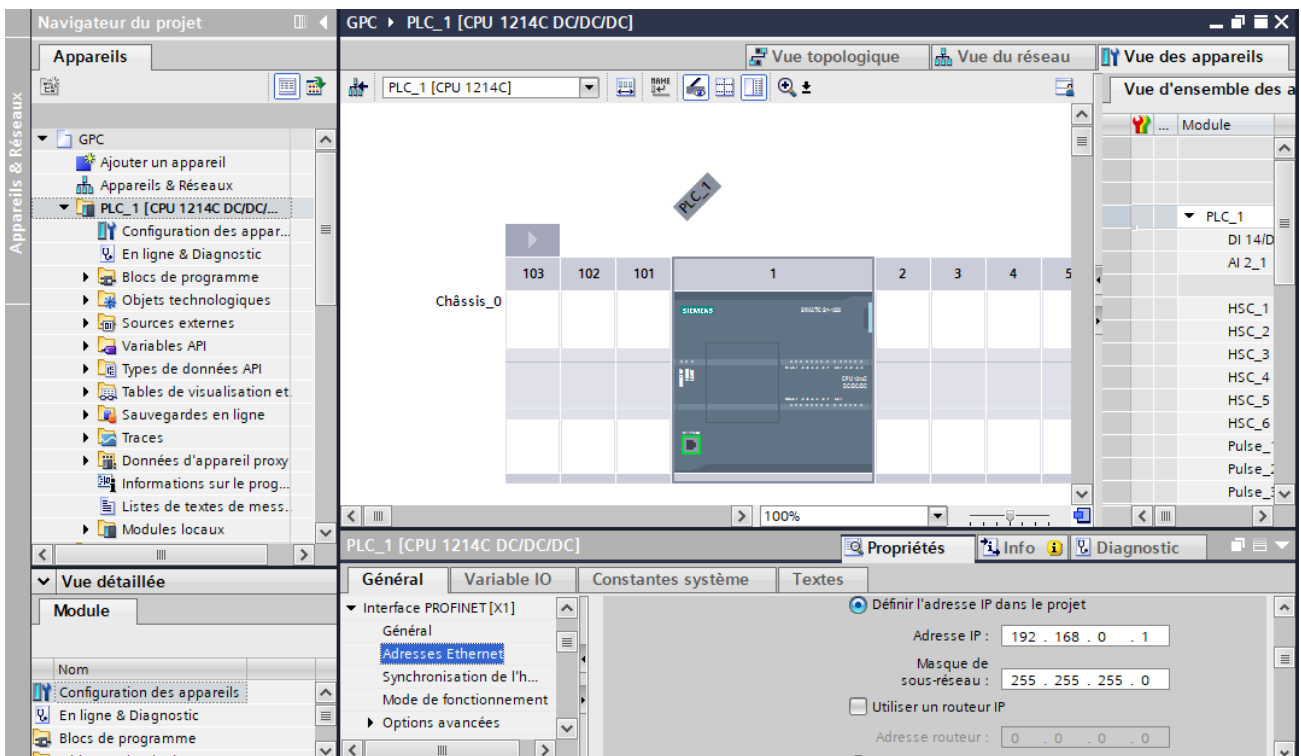


Figure 3-12 Adresse IP

- 4 Puisque de nos jours on programme avec des variables plutôt qu'avec des adresses absolues, on doit spécifier les **variables globales de l'API**. Avec un double-clic, ouvrez la table des variables API et entrez, comme montré ci- dessous, les noms des entrées et des sorties.

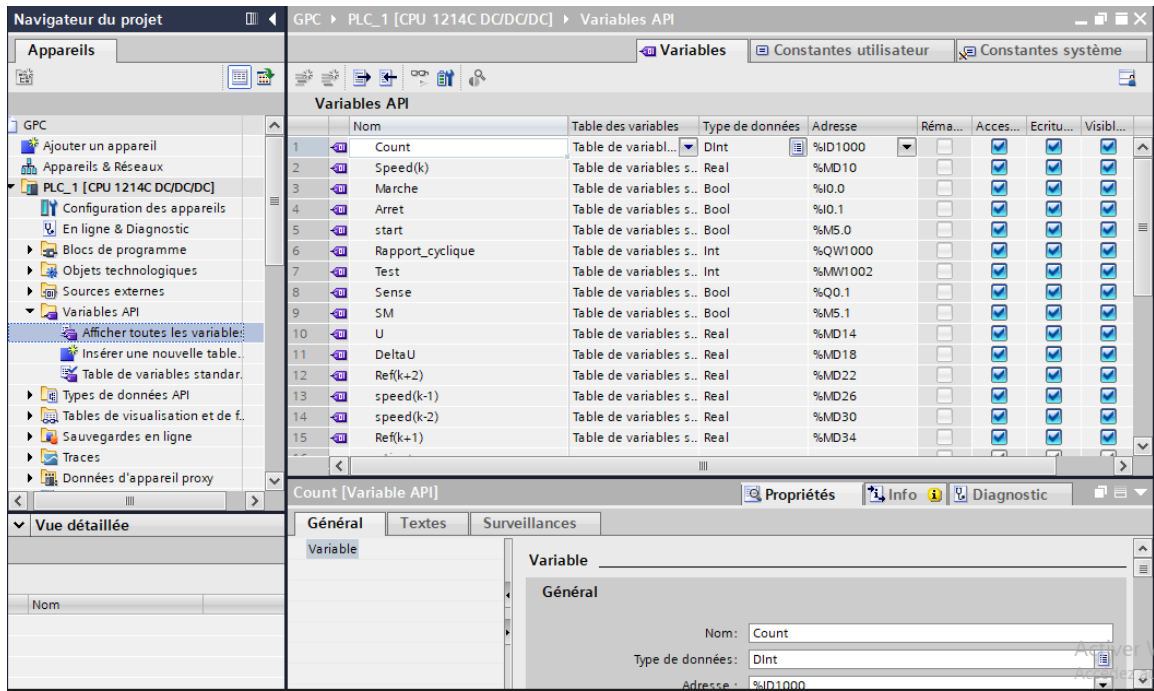
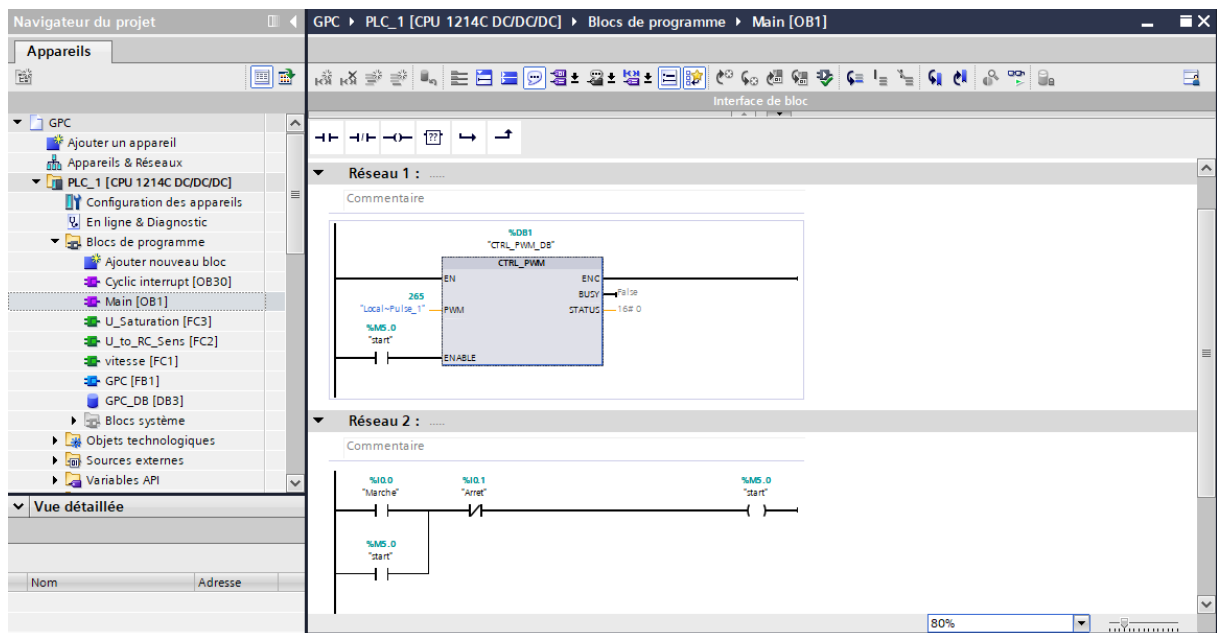


Figure 3-13 Tableau de variable

3.9.1 Le bloc d'organisation OB1

Le programme main pour un automate SIEMENS est dans OB1, où on doit mettre notre programme principal. Pour y accéder, on sélectionne Main [OB1] dans «bloc de programmation»



3-14 : Bloc OB1

3.9.2 Les blocs supplémentaires

Il existe différents types de bloc qu'on peut ajouter pour écrire notre programme avec des langages différents. On doit aller dans « blocs de programme », puis cliquer sur « Ajouter nouveau bloc ».

Les blocs fonctionnels (FB)

Les FB ont la particularité de pouvoir mémoriser les variables d'entrées, de sorties et d'entrées/sorties durablement dans des DB d'instances. Par ailleurs des variables temporaires peuvent être utilisées mais ces dernières ne sont disponibles que le temps d'un cycle.

Les fonctions (FC)

A la différence des FB, les fonctions (FC) sont des blocs de code sans mémoire, il est donc impératif de déclarer des variables locales avant d'entamer la programmation.

Le bloc de donnée (DB)

Les DB sont des zones de données pour le stockage des données utilisateur. En plus des données affectées à un bloc fonctionnel, les données partagées peuvent également être définies et utilisées par n'importe quel bloc.

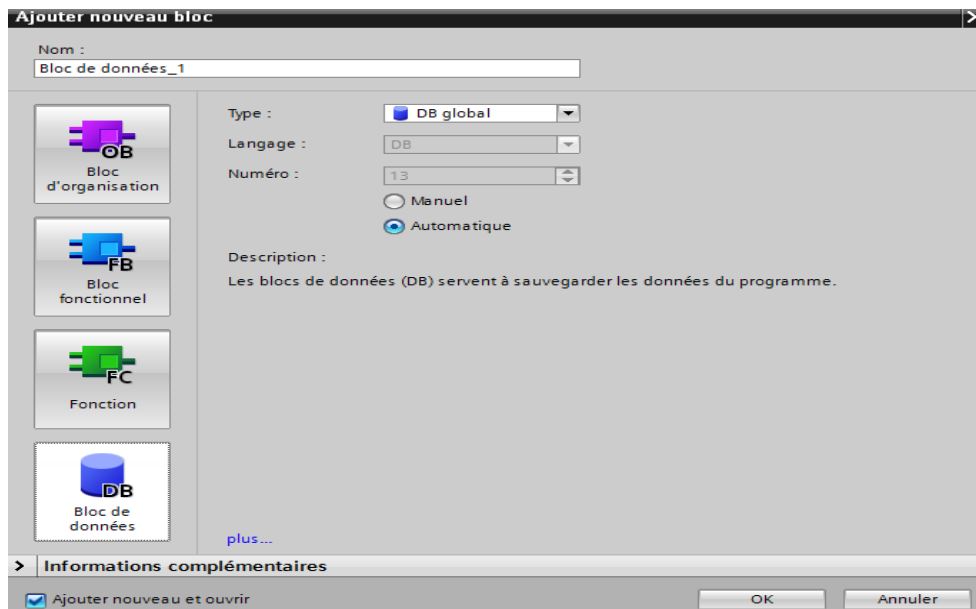


Figure 3-15 : Ajout d'un nouveau bloc

Si on veut choisir de créer un bloc FB, on clique d'abord sur l'icône où se trouve

« bloc fonctionnel »

Une fois le bloc choisi, ensuite on choisit le langage de programmation avec qui en souhaite écrire notre programme. Après on procède à la déclaration des variables (entrées, sorties, les entrées sorties) et on définit leurs types (Bool, Real..) avant de passer à la partie programmation qui se trouve dans la fenêtre en dessous :

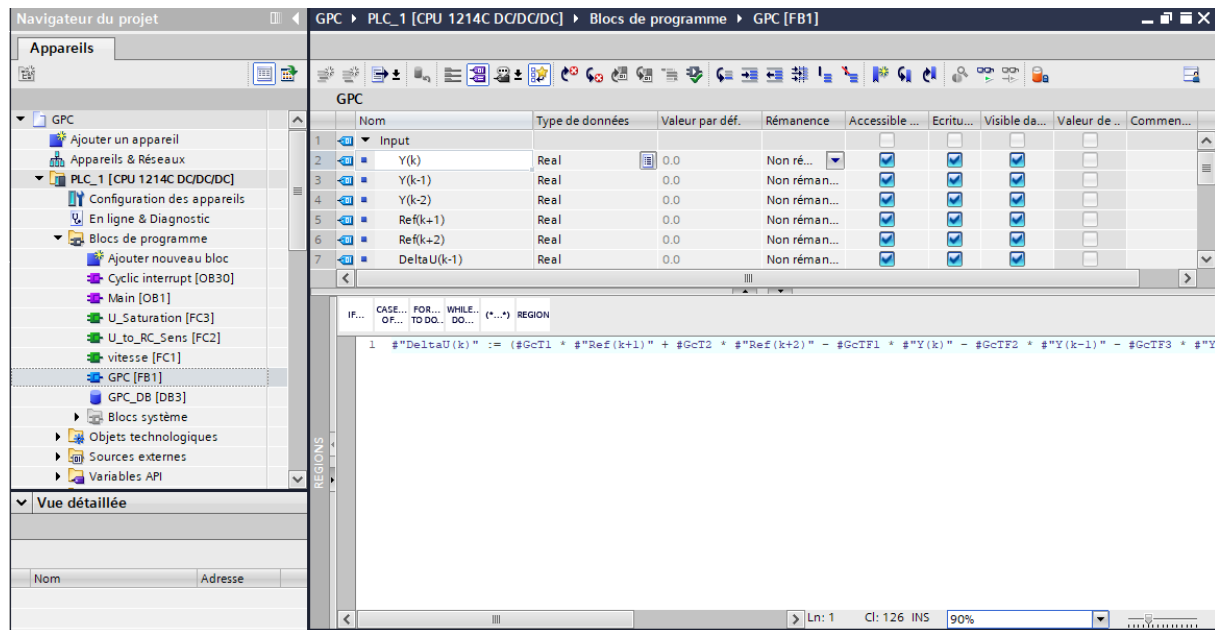


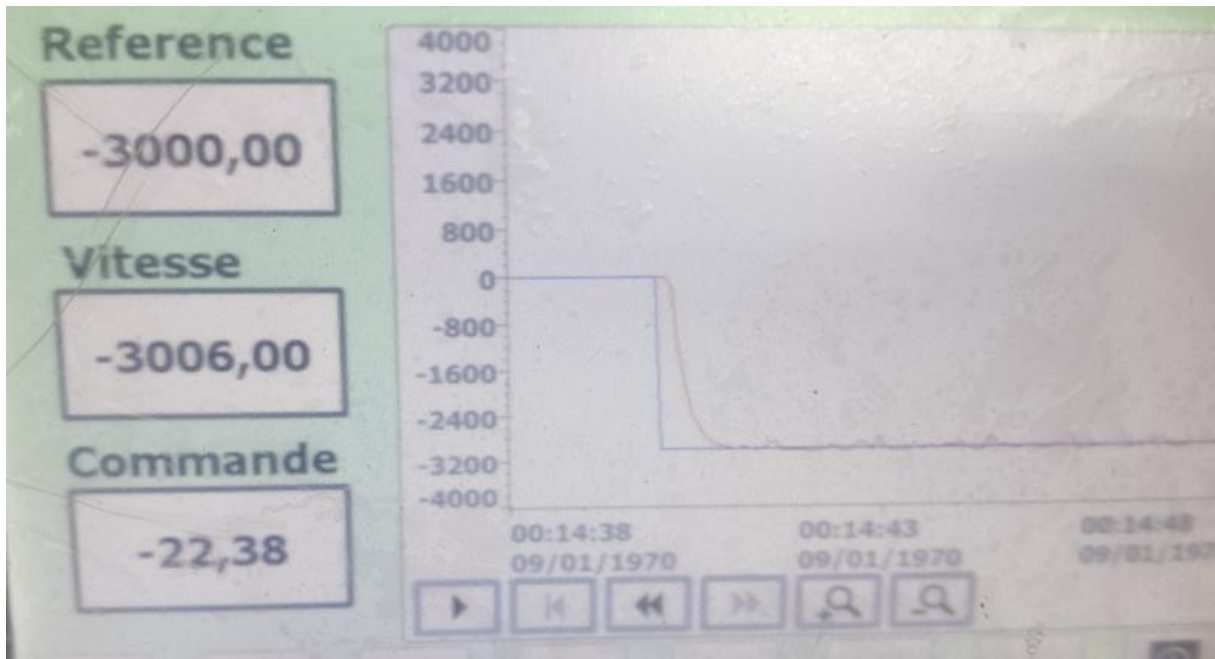
Figure 3-16 Programmation d'un bloc FB

Utilisation des fonctions dans le programme :

Une fois que les fonctions sont créés et compilés on peut les utilisé si besoin dans le main (OB1): le bloc de fonctions nécessite un bloc de données, et la définition de ces entrées sorties.

3.10 Essais pratique :

Essai 1 :



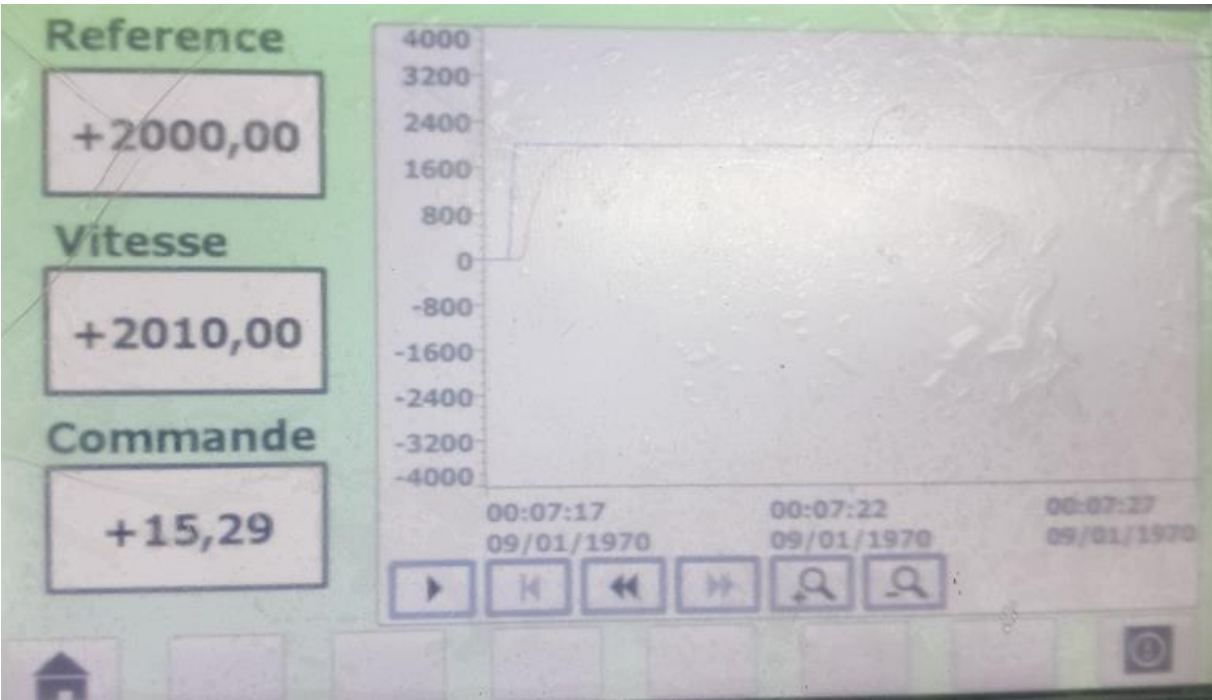
Essai 2 :



Essai 3 :



Essai 4 :



3.11 Conclusion :

L'utilisation de la commande prédictive généralisée à base d'un automate programmable industriel Siemens S7-1200 pour contrôler la vitesse d'un moteur à courant continu présente des avantages en termes de performance et de précision. Cette mise en œuvre nécessite également des compétences techniques et une compréhension approfondie des modèles prédictifs et des automates programmables industriels. La surveillance continue et la maintenance régulière sont également nécessaires pour assurer un fonctionnement efficace et éviter les défaillances potentielles.

Conclusion générale

La commande prédictive est une technique de commande avancée qui peut être utilisée pour contrôler des systèmes complexes et multivariés pour lesquels un simple régulateur PID est insuffisant. Cette technique est particulièrement intéressante pour les systèmes ayant des retards importants, des réponses inverses et de nombreuses perturbations. Les secteurs industriels tels que les raffineries de pétrole, l'industrie chimique, agroalimentaire, la métallurgie et l'aérospatiale sont les principaux utilisateurs de la commande prédictive.

Dans ce mémoire nous allons étudier la commande prédictive généralisée pour l'appliquer sur un moteur à courant continu à base d'un automate programmable industriel Siemens S7-1200.

L'utilisation de la commande prédictive généralisée basée sur un automate programmable industriel Siemens S7-1200 pour contrôler la vitesse d'un moteur à courant continu offre des avantages en termes de performances et de précision. Cependant, sa mise en œuvre nécessite des compétences techniques et une compréhension approfondie des modèles prédictifs et des automates programmables industriels. Une surveillance continue et une maintenance régulière sont également nécessaires pour assurer un fonctionnement efficace et éviter les défaillances potentielles.

Bibliographie

- [1]. KLAOUA Fares, commande prédictive généralisée Mémoire de Fin d'Etudes En vue de l'obtention du diplôme master en Automatique, Université Mohamed Kheder de Biskra 2014
- [2]. BOUTAINA Laredj, Commandes prédictives appliquées à un système hydraulique : étude expérimentale Mémoire de Fin d'Etudes En vue de l'obtention du diplôme master en Automatique, Université de Jijel 2021
- [3]. FERDJOUKH Yacine Commande Prédictive Généralisée : Application Au Moteur A Excitation Séparée , Mémoire de Fin d'Etudes En vue de l'obtention du diplôme master en Automatique, Université Mohamed Boudiaf – M'sila
- [4]. BENBRAHIM Amel, commande prédictive généralisée d'Une machine synchrone à Aimants Permanents, Mémoire de Fin d'Etudes En vue de l'obtention du diplôme magistère en Electrotechnique, Université de Batna, 2009
- [5]. SEKIIOU Ibrahim MEDFOUNI Hanan, Commande prédictive généralisée conception et application, Mémoire de Fin d'Etudes En vue de l'obtention du diplôme master en Automatique industrielle, Université l'Arbi Ben M'Hidi Oum Elbouaghi 2011
- [6]. KARA Kamel, Contrôle intelligent et systèmes autonomes (application a la robotique), Pour l'obtention du Diplôme de doctorat d'état en Electronique Université Ferhat Abbas– Sétif, 2006
- [7]. YACIA Sofiane. TAOUINET M_ Rafik. Commande prédictive d'un redresseur à MLI Triphasé, Mémoire de Fin d'Etudes En vue de l'obtention du diplôme master en électrotechnique, Université Abderrahman Mira Bejaia , 2013
- [8]. YACIA Sofiane. TAOUINET M_ Rafik. Commande prédictive d'un redresseur à MLI Triphasé, Mémoire de Fin d'Etudes En vue de l'obtention du diplôme master en électrotechnique, Université Abderrahman Mira Bejaia , 2013
- [9]. DEROUICH.A, "Machine a courant continu fonctionnement en moteur", Maroc : Université Sidi Mohamed Ben Abdellah, cours : chapitre 6, 2013, 11p.
- [10]. FRANÇOIS.B, "La machine à courant continu", France : université de Lille, cours, 2009, 69p.
- [11]. PIOU.M, "La machine à courant continu ", France, [En ligne]. Iutenligne, cours, 2015, 39p

[12]. REMADI Fouad et SIOUANE Abdelaziz, Modélisation et commande de la machine à courant continu, Mémoire de Fin d'Etudes En vue de l'obtention du diplôme master en Informatique Industrielle , Université Larbi Ben M'hidi, Oum El Bouaghi, 2012.

[13] : 01 1200 Initiation au programme du SIMATIC S7-1200.pdf

[14] : Automatisation et supervision de l'unité de traitement des eaux par osmose inverse du complexe CEVITAL via le logiciel TIA portal v13 de SIEMENS.pdf

[15] : Automatisation et supervision d'une station de Thermo laquage par un automate S7-1200 complet.pdf

[16] [Codeur incrémental : Fonctionnement, applications et produits SICK | SICK](#)
(consulter le 15/06/2023 à 14 :42)

[17] [Tutoriel L298N : fonctionnement, branchement, \(passionelectronique.fr\)](#)(consulter le 19/06/2023 à 16h)