

TABLE DES MATIERES

REMERCIEMENTS	i
RESUME.....	ii
ABSTRACT.....	iii
TABLE DES MATIERES.....	iv
LISTE DES FIGURES.....	x
LISTE DES TABLEAUX ET LISTE DES ABBREVIATIONS.....	xiii
LISTE DES SYMBOLES.....	xiv
INTRODUCTION GENERALE	1
Chapitre 1 : LES MACHINES A COURANT CONTINU.....	3
I. Introduction.....	3
II. Constitution du moteur.....	4
a. La partie fixe ou l'inducteur.....	4
b. La partie mobile ou l'induit.....	4
c. L'entrefer.....	4
d. Le collecteur et les balais.....	4
III. Principe de fonctionnement des moteurs à courants continu.....	5
IV. Equations générales.....	6
a. La force électromotrice.....	6
b. Le couple électromagnétique.....	7
c. Puissance électromécanique.....	7
d. La réaction magnétique d'induit.....	8

e. Réversibilité.....	8
f. Expression de la vitesse de rotation.....	9
V. Caractéristiques de fonctionnement.....	10
a. Fonctionnement à excitation constante.....	10
b. Caractéristique à vide $E : f(\Phi)$ et à vitesse constante.....	11
c. Caractéristique en charge $U : f(I)$	12
VI. Alimentation des machines à courant continu.....	13
Chapitre 2 : MODELISATION EN REGIME DYNAMIQUE DU MOTEUR A COURANT CONTINU	15
I. Introduction.....	15
II. Equation caractéristiques relatives à la modélisation.....	15
a. La tension induite généralisée.....	15
b. Le couple électromagnétique.....	16
III. Modélisation de la machine.....	16
a. Modélisation de la machine alimentée en tension.....	16
b. Modélisation de la machine à courant continu pour la régulation de courant.....	19
c. Modélisation réelle de la machine à courant continu.....	20
Chapitre 3 : PRINCIPE DE LA REGULATION DE VITESSE DES MOTEURS A COURANT CONTINU	22
I. Introduction.....	22
II. Principe général de la régulation.....	22
a. Les éléments de la chaîne de régulation.....	22

b. Concept de l'asservissement de vitesse.....	23
III. Structure de régulation d'un moteur à courant continu.....	25
a. Structure de la régulation en parallèle.....	25
b. Structure de la régulation en cascade.....	26
Chapitre 4 : MODELISATION DE LA CHAINE DE REGULATION	27
I. Introduction.....	27
II. Modélisation du moteur à courant continu.....	27
III. Modélisation de l'alimentation du moteur.....	31
IV. Synthèse du correcteur de vitesse.....	33
V. Synthèse du correcteur de courant.....	37
a. Dimensionnement du filtre de courant.....	37
b. Synthèse du correcteur de courant.....	38
Chapitre 5 : THEORIE DU STATEFLOW	41
I. Généralités.....	41
II. Introduction aux machines à états finis.....	41
III. Etudes des différents menus de l'éditeur STATEFLOW.....	42
a. Les objets graphiques.....	42
b. Les objets non graphiques.....	43
c. Le dictionnaire d'objet STATEFLOW.....	43
IV. Hiérarchie des objets dans STATEFLOW.....	43
V. Les opérateurs dans STATEFLOW.....	44
a. Les opérateurs arithmétiques et binaires.....	44
b. Les opérateurs unaires.....	44

c. Les opérateurs d’assignation.....	45
d. Les pointeurs et les adresses.....	45
e. Les types de variables.....	45
VI. Introduction aux différents objets.....	45
a. Les états.....	45
b. Les transitions.....	48
c. La transition de défaut.....	52
d. Les jonctions.....	52
e. La jonction historique.....	54
f. Les boîtes.....	54
g. Les évènements.....	54
h. Les données.....	55
i. Les actions.....	55
j. Les fonctions graphiques.....	55
VII. Utilisation des données et des fonctions MATLAB dans STATEFLOW...	56
a. Utilisation de l’opérateur ‘ml’.....	56
b. Utilisation des fonctions ml.....	56
c. Utilisation simultanée de l’expression ml et des fonctions ml...	57
VIII. Les opérateurs de logique temporelle (OLT).....	57
a. Règles d’utilisation des OLT.....	57
b. L’OLT After.....	57
c. L’OLT Before.....	58
d. L’OLT At.....	58

e. L'OLT Every.....	58
f. L'OLT In.....	58
IX. L'appel des fonctions C avec STATEFLOW.....	59
a. L'appel des fonctions C dans les transitions.....	59
b. L'appel des fonctions c dans les états.....	60
X. Utilisation des tables de vérité dans STATEFLOW.....	60
a. Introduction.....	60
b. Langage d'action des tables de vérité de STATEFLOW.....	61
c. Utilisation des tables de vérité.....	62
d. Programmation des tables de vérité.....	62
XI. Utilisation des outils de STATEFLOW.....	63
a. Le model explorer de STATEFLOW.....	63
b. Le debugger de STATEFLOW.....	63
XII. Etude d'un exemple.....	64
Chapitre 6 : APPLICATION DE STATEFLOW A LA REGULATION	
DE VITESSE	66
I. Généralités.....	66
II. Modélisation du régulateur de courant.....	66
a. Comportement du régulateur proportionnel.....	66
b. Mise en œuvre de l'algorithme pour l'action proportionnelle....	67
c. Conception du modèle STATEFLOW du correcteur proportionnel.....	68
d. Comportement du correcteur intégral.....	69
e. Mise en œuvre de l'algorithme pour l'action intégral.....	70

f. Mise en œuvre du modèle STATEFLOW du contrôleur intégral.....	72
g. Mise en œuvre du correcteur PI de courant.....	74
h. Affectation des paramètres du contrôleur PI de courant.....	75
III. Modélisation du correcteur de vitesse.....	76
a. Mise en œuvre de l'algorithme.....	76
b. Mise en œuvre du modèle STATEFLOW.....	77
c. Affectation des paramètres du contrôleur PI de vitesse.....	77
IV. Présentation de la chaîne de régulation et simulations.....	79
Chapitre 7 : REGULATION DE VITESSE DU MOTEUR A COURANT CONTINU	
DANS LE CAS REEL	83
I. Introduction.....	83
II. Présentation de la chaîne de régulation.....	83
a. Les éléments de la chaîne de régulation.....	83
b. Choix du moteur.....	84
c. Allure du couple résistant variable.....	85
d. Présentation des correcteurs.....	85
e. Introduction sur les non linéarités.....	86
f. Cycles limites et méthode du premier harmonique.....	92
g. Dimensionnement des correcteurs.....	93
III. Régulation de vitesse dans le cas réel.....	98
Chapitre 8 : MODELISATION DES CONTROLEURS DE COURANT ET DE	
VITESSE AVEC STATEFLOW	102
I. Introduction.....	102
II. Modélisation du contrôleur de vitesse.....	102

III. Modélisation de la saturation.....	103
a. Explication du diagramme.....	103
b. Mise en œuvre de l'algorithme.....	104
c. Modèle STATEFLOW du bloc saturation.....	105
IV. Modélisation du contrôleur de courant.....	106
a. Explication du diagramme.....	106
b. Présentation du contrôleur de courant.....	106
c. Mise en œuvre de l'algorithme.....	107
V. Présentation de la chaîne de régulation et simulations.....	109
Chapitre 9 : COMPARAISON DES MODELES ET INTERPRETATIONS.....	113
I. Introduction.....	113
a. Stabilité.....	113
b. Précision.....	114
c. Rapidité.....	115
d. Qualité des asservissements.....	116
II. Interprétations et conclusions.....	116
CONCLUSION GENERALE.....	118
ANNEXE	
BIBLIOGRAPHIE	

LISTE DES FIGURES

FIGURE 1.1 : Modes de fonctionnement d'une machine à courant continu

FIGURE 1.2 : Vue éclatée d'un moteur à courant continu

FIGURE 1.3 : Fonctionnement du moteur en quatre quadrants

FIGURE 1.4 : Caractéristique en charge d'un moteur à courant continu

FIGURE 1.5 : Caractéristique à vide du moteur à vitesse constante

FIGURE 1.6 : Caractéristiques en charge de la machine

FIGURE 1.7 : Alimentation d'un moteur à courant continu par un hacheur en pont

FIGURE 2.1 : Modèle du moteur à courant continu

FIGURE 2.2 : Modèle théorique du moteur à courant continu

FIGURE 2.3 : Modèle de la machine pour la régulation en courant

FIGURE 2.4 : Partie électrique de la machine

FIGURE 2.5 : Partie mécanique du moteur

FIGURE 3.1 : Réponse indicielle d'un moteur à courant continu

- FIGURE 3.2 :** Mouvement oscillatoire de la réponse
FIGURE 3.3 : Régulation en parallèle d'un moteur à courant continu
FIGURE 3.4 : Régulation en cascade d'un moteur à courant continu
FIGURE 4.1 : Modèle du moteur à courant continu.
FIGURE 4.2 : Réponse en boucle ouverte du moteur
FIGURE 4.3 : Modèle du couple résistant
FIGURE 4.4 : Allure du couple résistant
FIGURE 4.5 : Modèle macro du moteur à courant continu
FIGURE 4.6 : Réponse indicielle du moteur en boucle ouverte avec variation de C_r
FIGURE 4.7 : Montage en boucle fermée du moteur avec retour unitaire
FIGURE 4.8 : Réponse en vitesse du moteur en boucle fermée à retour unitaire
FIGURE 4.9 : Réponse en boucle fermée du moteur avec variation de charge
FIGURE 4.10 : Synthèse du correcteur par la méthode de Ziegler Nichols
FIGURE 4.11 : Régulation avec correcteur PI de vitesse
FIGURE 4.12 : Réponse indicielle du moteur pour la régulation de vitesse
FIGURE 4.13 : Oscillation de vitesse du moteur à courant continu
FIGURE 4.14 : Schéma de montage pour la régulation de courant
FIGURE 4.15 : Régulation en cascade pour l'asservissement de vitesse du moteur à courant continu
FIGURE 4.16 : Réponse indicielle du moteur lors de la régulation en cascade
FIGURE 4.17 : Allure du courant de commande
FIGURE 5.1 : Hiérarchie des objets dans STATEFLOW
FIGURE 5.2 : Hiérarchie des états
FIGURE 5.3 : Décomposition exclusive des états
FIGURE 5.4 : Décomposition parallèle des états
FIGURE 5.5 : Hiérarchie des transitions
FIGURE 5.6 : Modèle du diagramme avant l'application de la transition interne
FIGURE 5.7 : Diagramme simplifié par la transition interne
FIGURE 5.8 : Diagramme pour l'expression conditionnelle 'if-then-else'
FIGURE 5.9 : Diagramme pour l'expression conditionnelle 'if-then'
FIGURE 5.10 : Diagramme pour la boucle 'for'
FIGURE 5.11 : Diagramme pour la transition vers une source multiple
FIGURE 5.12 : Diagramme pour la transition vers une destination commune
FIGURE 5.13 : Règles de modélisation sous STATEFLOW
FIGURE 5.14 : Appel des fonctions C sur les transitions
FIGURE 5.15 : Appel des fonctions C sur les états
FIGURE 5.16 : Fonctions utilisées pour la modélisation du ventilateur
FIGURE 5.17 : Exemple de modélisation d'un ventilateur
FIGURE 6.1 : Générateur d'impulsion
FIGURE 6.2 : Organigramme pour l'action proportionnelle
FIGURE 6.3 : Fonction erreur(K_p)
FIGURE 6.4 : Modèle de l'état proportionnel
FIGURE 6.5 : Allure de la commande du trigger de l'action proportionnelle
FIGURE 6.6 : Réponse indicielle d'un correcteur intégral
FIGURE 6.7 : Organigramme pour la commande intégrale de courant
FIGURE 6.8 : Fonction decrement(step)
FIGURE 6.9 : Modèle de l'état intégral
FIGURE 6.10 : Signal de commande du trigger de l'action 'intégralcrt/'
FIGURE 6.11 : Modèle à état du correcteur Proportionnel intégral de courant
FIGURE 6.12 : Allure du courant I_a avec le modèle classique et avec STATEFLOW

- FIGURE 6.13 :** Effet des paramètres du contrôleur sur le courant I_a
- FIGURE 6.14 :** Organigramme pour la commande intégrale de vitesse
- FIGURE 6.15 :** Modèle du correcteur intégral de vitesse
- FIGURE 6.16 :** Influence de l'action Proportionnelle sur la vitesse
- FIGURE 6.17 :** Influence des paramètres du contrôleur sur la vitesse
- FIGURE 6.18 :** Allure de la vitesse avec le modèle du correcteur de vitesse
- FIGURE 6.19 :** Principe de montage pour la régulation en cascade avec correcteur STATEFLOW
- FIGURE 6.20 :** Allure de la variation du couple résistant
- FIGURE 6.21 :** Allure du courant de référence délivré par le correcteur de vitesse
- FIGURE 6.22 :** Allure de la vitesse pour la régulation en cascade avec STATEFLOW
- FIGURE 7.1 :** Chaîne de régulation en cascade
- FIGURE 7.2 :** Modèle macro du moteur
- FIGURE 7.3 :** Allure du couple résistant
- FIGURE 7.4 :** Modèle de correcteur PI
- FIGURE 7.5 :** Modèle de contrôleur de courant
- FIGURE 7.6 :** Chaîne de régulation avec un organe non linéaire
- FIGURE 7.7 :** Courbe de fonctionnement d'un organe saturation
- FIGURE 7.8 :** Lieu critique des pôles d'un organe saturation
- FIGURE 7.9 :** Courbe de fonctionnement d'un organe tout ou rien avec hystérésis
- FIGURE 7.10 :** Lieu critique des pôles d'un relais à hystérésis
- FIGURE 7.11 :** Auto-oscillation dans le plan de Nyquist
- FIGURE 7.12 :** Montage en boucle ouverte du moteur
- FIGURE 7.13 :** Réponse en boucle ouverte du moteur
- FIGURE 7.14 :** Détermination des caractéristiques de Ziegler Nichols
- FIGURE 7.15 :** Schéma du contrôleur de courant
- FIGURE 7.16 :** Schéma simplifié de l'alimentation du moteur
- FIGURE 7.17 :** Courbes théoriques de U_a et de I_a en fonction du rapport cyclique
- FIGURE 7.18 :** Réponse indicielle du moteur à charge fixe
- FIGURE 7.19 :** Allure du courant d'induit I_a pour un couple résistant constant
- FIGURE 7.20 :** Réponse indicielle du moteur à charge variable
- FIGURE 7.21 :** Allure du courant d'induit à charge variable
- FIGURE 7.22 :** Allure de la tension d'induit du moteur avec variation de charge
- FIGURE 7.23 :** Allure du couple résistant en régime nominal
- FIGURE 7.24 :** Allure du courant d'induit du moteur en régime de fonctionnement nominal
- FIGURE 7.25 :** Allure de la vitesse du moteur en régime nominal
- FIGURE 8.1 :** Fonctions graphiques STATEFLOW utilisés dans le modèle du contrôleur de vitesse
- FIGURE 8.2 :** Modèle du contrôleur de vitesse
- FIGURE 8.3 :** Organigramme du modèle saturation
- FIGURE 8.4 :** Diagramme STATEFLOW pour la saturation
- FIGURE 8.5 :** Modèle du limiteur de courant
- FIGURE 8.6 :** Bloc STATEFLOW du contrôleur de courant
- FIGURE 8.7 :** Diagramme STATEFLOW et fonction utilisée dans le contrôleur de courant
- FIGURE 8.8 :** Organigramme du modèle du relais à hystérésis
- FIGURE 8.9 :** Chaîne de régulation du moteur avec son alimentation
- FIGURE 8.10 :** Réponse indicielle du moteur avec variation de charge
- FIGURE 8.11 :** Allure de la tension d'induit du moteur avec couple résistant variable
- FIGURE 8.12 :** Allure du courant d'induit I_a pour le moteur chargé
- FIGURE 8.13 :** Allure de la vitesse du moteur en régime de fonctionnement nominal

FIGURE 8.14 : *Allure du courant d'induit du moteur en régime de fonctionnement nominal*

FIGURE 9.1 : *Réponse impulsionnelle du modèle classique*

FIGURE 9.2 : *Réponse impulsionnelle du modèle STATEFLOW*

FIGURE 9.3 : *Variation de l'erreur du modèle classique*

FIGURE 9.4 : *Variation de l'erreur du modèle avec STATEFLOW*

LISTE DES TABLEAUX

TABLEAU 1.1 : *Modes d'excitation d'une machine à courant continu*

TABLEAU 4.1 : *Fréquence de hachage du convertisseur statique*

TABLEAU 4.2 : *Tableau des paramètres d'un régulateur PID selon Ziegler Nichols*

TABLEAU 5.1 : *Liste des objets graphiques de l'éditeur STATEFLOW*

TABLEAU 5.2 : *Liste des opérateurs disponibles dans STATEFLOW*

TABLEAU 5.3 : *Hiérarchie des transitions*

TABLEAU 5.4 : *Règles de validation des transitions*

TABLEAU 5.5 : *Aperçu d'une table de vérité*

TABLEAU 5.6 : *Règle de saisie des conditions et des actions*

TABLEAU 5.7 : *Saisie des conditions*

TABLEAU 5.8 : *Saisie des actions*

TABLEAU 7.1 : *Tableau des caractéristiques de la machine*

TABLEAU 9.1 : *Caractéristiques de la courbe de démarrage du moteur*

LISTE DES ABBREVIATIONS

Sigles	Significations
BF	Boucle Fermée
BO	Boucle Ouverte
FEM	Force Électromotrice
FSM	Finite State Machine
FTBF	Fonction de Transfert en Boucle Fermée
FTBO	Fonction de Transfert en Boucle Ouverte
GTO	Gate Turn Off
I	Intégral
IAE	Integral Absolute Error
ISE	Integral Square Error
ITAE	Integral of Time-multiplied Absolute-value of Error
ITSE	Integral of Time-multiplied Square Error
MATLAB	Matrix Laboratory
MCC	Moteur à Courant Continu
OLT	Opérateur de Logique Temporelle
P	Proportionnel
PI	Proportionnel Intégral
PID	Proportionnel Intégral Dérivé
PWM	Pulse Width Modulator
SimPowerSystem	Simulink with Power System
UML	Unified Modeling Language

LISTE DES SYMBOLES

Symboles	Significations	Unités
2a	Nombre de voies d'enroulements	
2p	Nombres de pôles	
E, e	Force électromotrice	[V]
Φ	Flux d'induction magnétique	[Wb]
B	Champs d'induction magnétique	[T]
n	Nombre de conducteur d'induit	
Ω	Vitesse de rotation	[rad/s]
U, Ua	Tension d'induit	[V]
Ra, R	Résistance d'induit	[Ω]
La, L	Inductance d'induit	[H]
Rf	Résistance de l'inducteur	[Ω]

Lf	Inductance de l'inducteur	[H]
I, Ia	Courant d'induit	[A]
Laf	Inductance mutuelle induit inducteur	[H]
Cem	Couple électromagnétique	[N.m]
Cr, TL	Couple résistant	[N.m]
Pem	Puissance électromécanique	[W]
N	Vitesse de rotation	[tr/min]
F(p)	Fonction de transfert du capteur de courant	
A(p)	Fonction de transfert du correcteur de courant	
G(p)	Fonction de transfert du processus à réguler (moteur)	
C(p)	Fonction de transfert du correcteur de vitesse	
C(t), u(t)	Loi de commande	
e(t)	Entrée du système	
e(t), ε(t)	Erreur du système	
s(t)	Sortie du système	
Kp	Gain proportionnel	
Ki	Gain intégral	
Step1,step2,step	Paramètres de réglages des contrôleurs STATEFLOW	
fh	Fréquence de hachage du convertisseur statique	[Hz]
α	Rapport cyclique	
J	Moment d'inertie des parties mobiles du moteur	[kg.m ²]
f	Coefficient de frottement visqueux	[N.m.s]
N(x)	Fonction de transfert généralisé d'un organe non linéaire	
C(x)	Lieu critique d'un organe non linéaire	
Iref	Valeur du courant de référence pour la boucle de courant	[A]
D	Amplitude du premier dépassement	[%]
tr	Temps de réponse du système	[s]
tp	Temps de pic du système	[s]
tm	Temps de monrée du système	[s]
τ	Constante de temps	[s]

INTRODUCTION GENERALE

L'automatique est un domaine qui tient une place de plus en plus importante dans l'activité humaine. Elle ne se limite plus au secteur industriel, mais trouve également son application dans les produits les plus courants. Les systèmes automatisés ont été conçus, à l'origine, pour remplacer l'homme dans l'accomplissement de tâches fastidieuses, dangereuses ou dépassant ses capacités physiques. Le XIX^e siècle fut l'époque de la première révolution industrielle, on construisit des machines qui étaient le prolongement et l'amplification des muscles (locomotives, machines à vapeur,...). Puis vint le XX^e siècle, avec la seconde révolution industrielle où des machines ont été conçus pour penser et pour

commander d'autres machines. Par exemple pour le pilotage automatique d'avion ou de chaînes de montage industrielles.

La régulation quant à elle, est une technique qui consiste à la prise de contrôle d'une ou de plusieurs grandeurs physiques d'un système en vue d'en imposer son comportement. Dans notre cas le système physique sera un moteur à courant continu et le comportement à imposer sera de garder une sortie (vitesse) fixe et égale à la consigne d'entrée, quelles que soient les perturbations. Le choix s'est porté sur le moteur à courant continu à excitation séparée parce qu'il est, depuis longtemps, le type d'entraînement le plus utilisé pour les applications à vitesse variable, son côté attractif provient en grande partie de la simplicité de sa commande en raison de sa plage de tension élevée, ce qui a une incidence directe sur le niveau des performances, que l'on parle en termes d'asservissement ou en termes économiques. En dépit de cette particularité, le moteur est aussi le siège de couples résistants tendant à modifier son comportement, d'où la nécessité de contrôler son fonctionnement pour asservir sa vitesse.

Cependant face à l'évolution galopante de la technologie et la recherche de performances toujours plus élevée, l'exploration de nouvelles méthodes permettant de dépasser les limites d'autres techniques (pourtant bien maîtrisées) ne cesse de hanter l'esprit humain. L'utilisation de nouvelles approches, en l'occurrence celle des graphes d'états pour la régulation automatique est l'objet de ce mémoire, la modélisation et la simulation se fera sous l'environnement du langage textuel, MATLAB, auquel ont été ajoutés deux outils, SIMULINK et STATEFLOW. Au terme de cette étude on pourra conclure si la méthode des graphes d'états est adaptée à la régulation de vitesse des machines à courant continu.

Le mémoire est organisé en trois parties, dans la première partie nous mettrons l'accent sur les principes généraux de la régulation et la modélisation du processus à réguler c'est-à-dire de la machine à courant continu, cette modélisation se fera à partir des équations en régime dynamique de la machine, elle se soldera par la création d'un bloc macro du modèle.

La deuxième partie sera consacrée à l'analyse et la modélisation de la chaîne de régulation par la méthode classique et par la méthode des graphes d'états. La méthode classique de modélisation consiste à concevoir un modèle de la chaîne avec des éléments uniquement composé de fonctions de transfert, on synthétisera à partir de méthodes empiriques l'expression des correcteurs de vitesse et de courant. Ensuite nous entrerons à la modélisation sous STATEFLOW, le but de la modélisation est de concevoir un contrôleur délivrant le même signal de commande que celui des correcteurs, mais cette fois ci en utilisant le langage des graphes d'états.

La troisième partie quant à elle traitera de la régulation du moteur à courant continu dans le cas réel (modèle classique) et de l'application du modèle de graphe d'état à la régulation du processus réel. Le processus réel sera constitué du moteur couplé avec son alimentation ; asservir la vitesse du moteur consistera donc à modéliser le contrôleur de

vitesse et à modéliser la logique de commande de l'alimentation. On terminera par une comparaison de l'action des deux modèles.

Chapitre 1

Les Machines à Courant Continu

I. Introduction :

La machine à courant continu est un convertisseur d'énergie réversible qui peut être utilisée suivant deux modes de fonctionnement:

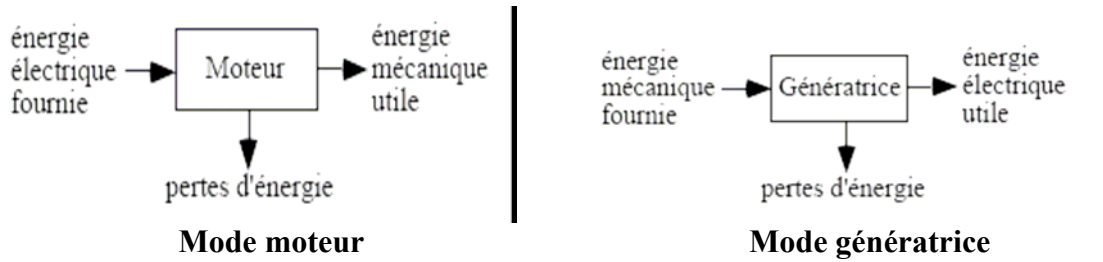


FIGURE 1.1 : Modes de fonctionnement d'une machine à courant continu

La machine à courant continu est donc un convertisseur électromécanique. Il est totalement réversible du fait de la facilité avec laquelle la machine change de mode. Les machines à courant continu peuvent être classées suivant leurs modes d'excitation :

Type d'excitation	Modèle équivalent
Machine à excitation série	
Machine à excitation indépendante	<p>Modèle équivalent :</p>
Machine à excitation composée	

TABLEAU 1.1 : Modes d'excitation d'une machine à courant continu

II. Constitution du moteur :

La machine à courant continu étant un système électromécanique, il comprend donc une partie mécanique et une partie électrique mais également un circuit magnétique qui effectue la jonction entre les deux parties précédentes. Nous allons citer ces différentes parties de la machine [4].

a. La partie fixe ou l'inducteur :

Appelée aussi stator, elle est de forme cylindrique creuse et comporte des saillies généralement constituées soit d'aimants permanents en matériaux ferromagnétiques (on les retrouve sur des machines dont les puissances varient d'une dizaine à une vingtaine de kilowatts), soit de bobines enroulées autour de noyaux polaires (puissance supérieure à 100kW), c'est cette partie du moteur qui sera à l'origine de la création du flux inducteur.

b. La partie mobile ou l'induit :

Connue sous la dénomination de rotor, elle est de forme cylindrique pleine et comporte des encoches dans lesquelles seront disposés les conducteurs d'induits ; ces conducteurs seront le siège des forces électromotrices qui créeront le mouvement rotatif du rotor.

c. L'entrefer :

Espace compris entre l'inducteur et l'induit du moteur, cet espace joue le rôle de circuit magnétique et achemine les flux de champs magnétiques allant de l'inducteur vers l'induit.

d. Le collecteur et les balais:

C'est un organe mécanique composé d'ensemble de lames de cuivre isolées latéralement les unes des autres et disposées suivant un cylindre en bout de rotor. Ces lames sont réunies aux conducteurs d'induit et réalisent ainsi une liaison électrique. Les balais portés par le stator frottent sur les lames du collecteur. L'ensemble collecteur-balais joue le rôle de redresseur mécanique en ajoutant instantanément les forces électromotrices de même signe.

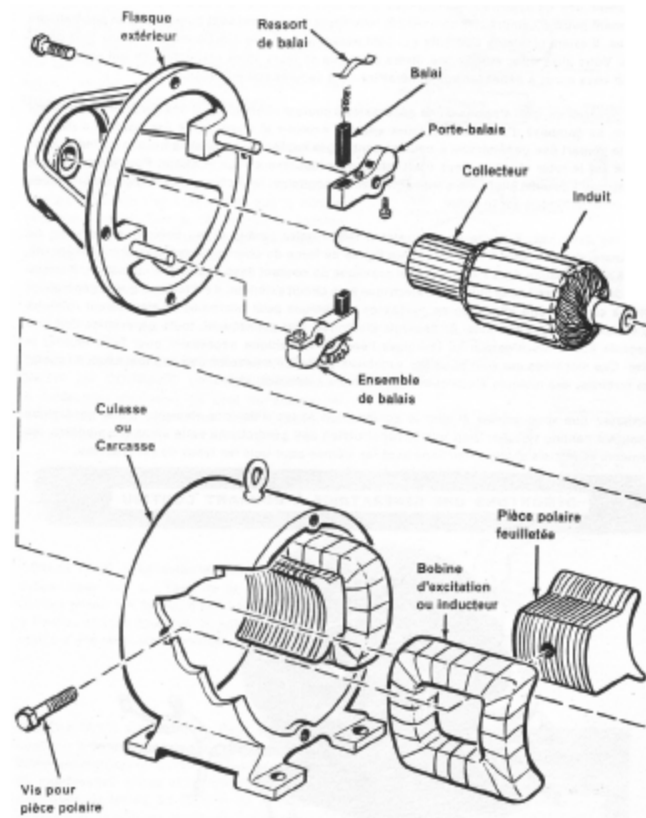


FIGURE 2.2 : Vue éclatée d'un moteur à courant continu

III. Principe de fonctionnement du moteur à courant continu :

Le moteur à courant continu est le siège de phénomènes physiques qui décrivent le fonctionnement de la machine. Ces phénomènes sont définis par la loi de Laplace et celle de Faraday [2].

Le fonctionnement du moteur à courant continu est régi par la loi de Faraday ; elle se traduit par la création d'une force électromotrice induite « \vec{e} » aux extrémités d'un conducteur de longueur l animé d'un mouvement \vec{V} et placé dans un champ d'induction magnétique \vec{B} .

$$\boxed{e = - \frac{d\phi}{dt} = l v \wedge B} \quad (1-1)$$

Simultanément au phénomène précédent on constate que le conducteur est également soumis à la loi de Laplace qui dit que toute portion de circuit $d\vec{l}$ traversée par un courant d'intensité i et placée dans un champ d'induction magnétique \vec{B} est le siège d'une force élémentaire $d\vec{f}$ tendant à le mettre en mouvement.

Cette force de Laplace est décrite par l'équation suivante :

$$\boxed{df = i dl \wedge B} \quad (1-2)$$

Conformément à ces deux lois nous allons essayer d'expliquer en quelques lignes le fonctionnement d'un moteur à courant continu. Alimenté sous un courant continu l'inducteur de la machine crée un flux magnétique qui sera acheminé vers l'induit à travers le circuit magnétique de la machine c'est-à-dire à travers l'entrefer, en supposant le rotor en mouvement, comme le circuit d'induit est un enroulement il y aura création d'une force électromotrice dans les conducteurs d'induit, les enroulements se trouvant eux même dans le champs électromagnétique créé par l'inducteur ; en suivant la loi de Laplace il y aura création d'une force F qui tend à mettre le rotor en mouvement ; et ce phénomène se répète continument.

IV. Equations générales :

Les phénomènes qui apparaissent dans un moteur à courant continu peuvent être traduits mathématiquement à l'aide d'équations permettant de mieux comprendre son fonctionnement.

a. La force électromotrice :

L'augmentation du nombre de paires de pôles diminue le pas polaire et permet de créer des forces électromotrices (f.é.m.) élémentaires sur une fraction de tour de la machine. En plaçant toutes ces f.é.m. par pôles en série, on augmente la f.é.m. totale d'un facteur de $2p$ et les conducteurs organisés en voie d'enroulements la diminue d'un facteur total de $2a$, donc relativement à la loi de Faraday, la machine à courant continu est le siège d'une f.é.m. totale E définie par la formule :

$$\boxed{E = \frac{2p}{2a} n \Phi \frac{\Omega}{2\pi}} \quad (1-3)$$

On condense cette formule en regroupant les termes constants, ce qui donne :

$$\boxed{E = K_E \Phi \Omega} \quad (1-4)$$

Où $K_E \Phi$ est la constante de la f.é.m. de la machine.

Les expressions de la tension d'alimentation de l'induit sont données ci-dessous :

En fonctionnement moteur :

$$\boxed{U = E + R_a I + e_B} \quad (1-5)$$

En fonctionnement génératrice:

$$\boxed{U = E - R_a I - e_B} \quad (1-6)$$

e_B étant la chute de tension au niveau des balais collecteurs.

b. Le couple électromagnétique :

Chaque conducteur logé dans les encoches de l'induit est le siège d'un effort créant un couple sur l'arbre moteur. La somme des efforts unitaires sur chaque conducteur donne naissance à un couple nommé couple électromagnétique. Suivant la relation issue de la loi de Laplace (1-2), le couple dépend du flux sous pôle Φ , du courant total I et du nombre de conducteur « n ». Similairement à celui de la f.é.m. le nombre de pôle augmente le couple électromagnétique tandis que le nombre de voie le diminue. On obtient finalement la relation qui suit :

$$\boxed{C_{em} = \frac{2p}{2a} n \Phi \frac{I}{2\pi}} \quad (1-7)$$

Et en regroupant les termes constants de cette relation on obtient la formule condensée suivante :

$$\boxed{C_{em} = K_{em} \Phi I} \quad (1-8)$$

Où $K_{em} \Phi$ est la constante de couple électromagnétique. On constate également que les constantes $K_{em} \Phi$ et $K_E \Phi$ sont identiques.

c. Puissance électromécanique :

Comme l'induit est le siège d'une f.é.m. E et est traversé par un courant d'intensité I , il reçoit une puissance électromécanique d'expression :

$$\boxed{P_{em} = EI} \quad (1-9)$$

Et d'après le principe de conservation de l'énergie, cette puissance est égale à la puissance développée par le couple électromagnétique C_{em} , d'où :

$$\boxed{P_{em} = C_{em} \Omega = EI} \quad (1-10)$$

d. La réaction magnétique d'induit :

En réalité il existe la présence d'un flux antagoniste qui tend à s'opposer au flux réel sous pôle donc qui tend à le diminuer ; cette perte est surtout très remarquée pour les machines non compensées. En effet lorsqu'un courant traverse le conducteur d'induit, qui est un bobinage, il y a formation d'un flux inverse de sens opposé au flux principal qui notons le lui a donné naissance, d'où le flux en charge est la différence entre le flux à vide et la chute $\Delta\Phi$.

$$\Phi = \Phi_0 - \Delta\Phi$$

On en déduit l'expression de la tension induite en charge :

$$E = \frac{2p}{2a} n \frac{\Omega}{2\pi} (\Phi_0 - \Delta\Phi) \quad (1-11)$$

La solution pour parer à ce problème est donc de compenser cette fuite en mettant en place un enroulement de compensation du côté de l'inducteur dans les pôles principaux, cet enroulement est traversé par le courant d'induit et crée ainsi un flux additionnel de même grandeur que le flux $\Delta\Phi$ mais de sens contraire, d'où son annulation.

e. Réversibilité :

La machine à courant continu est réversible. A flux constant, E ne dépend que de Ω et C_{em} ne dépend que de I . La f.é.m. de la machine et l'intensité du courant d'induit I dans le circuit sont deux grandeurs indépendantes. La machine peut donc indifféremment fonctionner en mode moteur lorsque la puissance convertie est positive ($P_{em} > 0$) ou en génératrice quand elle est négative ($P_{em} < 0$).

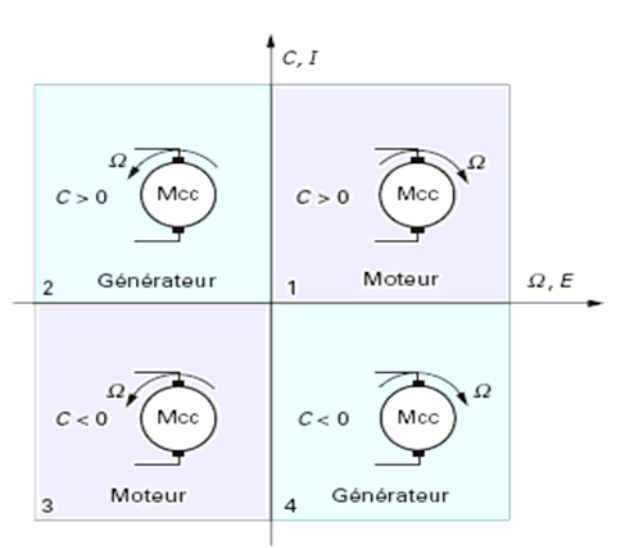


FIGURE 1.3 : Fonctionnement du moteur en quatre quadrants

On constate quatre types de fonctionnement suivant les quatre quadrants :

Quadrant I moteur : vitesse et couple positifs ;

Quadrant II génératrice –frein : vitesse positive et couple négatif ;

Quadrant III moteur : vitesse et couple négatifs ;

Quadrant IV génératrice-frein : vitesse négative et couple positif

f. Expression de la vitesse de rotation :

En supposant notre machine compensée on a :

$$\boxed{U = E + R_a I + L_a \frac{dI}{dt} + e_B} \quad (1-12)$$

e_B : étant la chute de tension du système balais collecteur

L_a : l'inductance des enroulements d'induit

R_a : la résistance d'induit

I : le courant traversant le circuit d'induit

En supposant que la variation du courant d'induit est négligeable c'est-à-dire :

$$L_a \frac{dI}{dt} \rightarrow 0$$

n : étant le nombre de conducteur actif

On a finalement l'expression de la vitesse égale à :

$$\boxed{N = \frac{U - (R_a I + e_B)}{\frac{2p}{2a} n \Phi}} \quad (1-13)$$

On constate selon cette formule que la vitesse de rotation du moteur peut être variée soit en agissant sur la tension d'alimentation U , soit en agissant sur le circuit inducteur en l'occurrence sur le courant d'excitation I_f ; cela se traduit par la variation du flux Φ .

V. Caractéristiques de fonctionnement :

a. Fonctionnement à excitation constante :

Ce type de fonctionnement est très utilisé à cause de sa simplicité d'utilisation et de la linéarité de contrôle (couple proportionnel à la vitesse). On le rencontre surtout dans les machines à aimants permanents ; par contre pour les machines à inducteur bobiné les grandeurs du circuit inducteur sont tenues constantes par la mise en alimentation de l'inducteur sous une tension constante, la variation de la vitesse n'étant plus fonction du flux, la machine est commandée uniquement par la tension d'induit U.

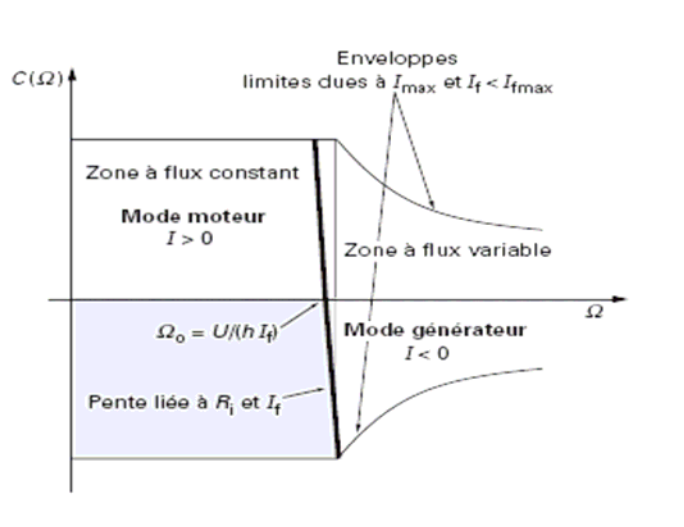


FIGURE 1.4 : Caractéristique en charge d'un moteur à courant continu

Lors du fonctionnement en excitation constante la machine est limitée en couple et en vitesse, représenté par la zone ombrée de la figure (1.4). La figure (1.3) quant à elle, représente le fonctionnement de la machine sur quatre quadrants.

- La commande à flux constant ou commande par l'induit :

Les grandeurs liées à l'inducteur en l'occurrence le courant d'excitation I_f , la tension U_f et le flux Φ sont constants, par conséquent la vitesse du moteur est proportionnelle à la tension U d'alimentation de l'induit.

On obtient, d'après les formules de base, l'équation de la vitesse en fonction du couple qui forme une droite.

Elle est donnée par :

$$\Omega = \frac{1}{K_E \Phi} \left(U - \frac{R_a C}{K_E \Phi} \right) \quad (1-14)$$

Chapitre 1 : Les machines à courant continu

Les limites de fonctionnement de la machine sont fixées par les valeurs maximales de la tension nominale U_n qui limite la vitesse et du courant nominal I_n qui limite le couple.

$$\boxed{-U_n \leq U \leq U_n} \quad (1-15)$$

$$\boxed{-C_{\max} = -K_E \Phi I_n \leq C \leq C_{\max} = K_E \Phi I_n} \quad (1-16)$$

- La commande à flux variable ou commande par l'inducteur :

La commande de la vitesse à tension d'induit U constante est possible par la variation du courant d'excitation I_f donc du flux inducteur Φ . A cause des caractéristiques magnétiques et des couplages électromécaniques, les équations de la machines sont fortement non linéaires et suivent une loi hyperbolique d'équation :

$$\boxed{C\Omega = \frac{U_n^2}{4R}} \quad (1-17)$$

Alors qu'en régime permanent la caractéristique tend vers une droite qui se traduit de Ω (1-14) avec la tension d'induit $U=U_n$.

b. Caractéristiques à vide $E=f(\Phi)$ et à vitesse Ω constante:

La caractéristique à vide définit le fonctionnement de la machine, à courant d'excitation variable et à vitesse de rotation constante, on observe sur la figure (1.5) trois zones bien distinctes :

- De 0 à A : la caractéristique est linéaire c'est-à-dire que la f.é.m. varie proportionnellement au flux inducteur Φ car elle n'est plus fonction de la vitesse qui est désormais une constante.

$$E = K_E \cdot \Phi \cdot \Omega$$

- Dans la portion AB il y a début de saturation du matériau constituant le moteur.
- Au delà du point B, il y a saturation complète du matériau donc la variation du flux inducteur n'influe plus sur la valeur de E (elle devient fixe), cette zone s'accompagne de la création de pertes qui augmentent de plus en plus avec la valeur de I_f .

La zone de fonctionnement idéale est celle qui se trouve au voisinage du point A. En réalité la variation successive du courant d'excitation I_f donne naissance à un phénomène d'hystérésis du fait de la présence du matériau ferromagnétique.

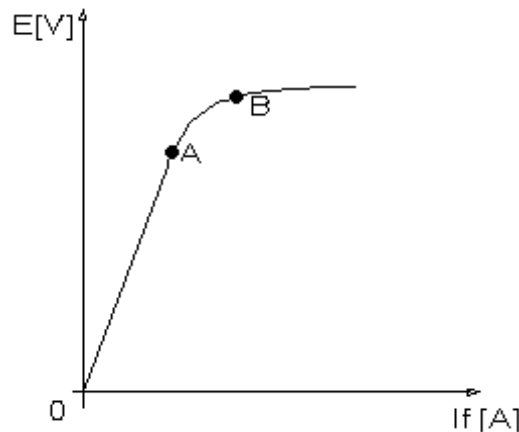


FIGURE 1.5 : Caractéristique à vide du moteur à vitesse constante

c. Caractéristiques en charge $U=f(I)$:

On observe une légère chute de tension ohmique due à la présence des résistances de bobinage de l'induit et une chute due aux contacts balais-collecteur ; cette valeur de la chute est exprimée par l'expression $R_a I + e_B$.

On remarque également la présence d'une chute de tension supplémentaire due à la réaction magnétique d'induit, elle sera notée ΔU .

On obtient les équations caractéristiques suivantes pour chaque type de fonctionnement :

- En fonctionnement moteur :

$$E = U - (R_a I + e_B) - \Delta U \quad (1-18)$$

- En mode génératrice :

$$U = E - (R_a I + e_B) - \Delta U \quad (1-19)$$

Ces deux types de fonctionnement caractérisent la réversibilité des machines à courant continu. Le moteur absorbe un courant I ($I > 0$) expression (1-18) pour alimenter l'induit, tandis qu'en génératrice, la machine restitue un courant ($I < 0$) expression (1-19).

Les caractéristiques en charge de la machine sont représentées par la figure suivante :

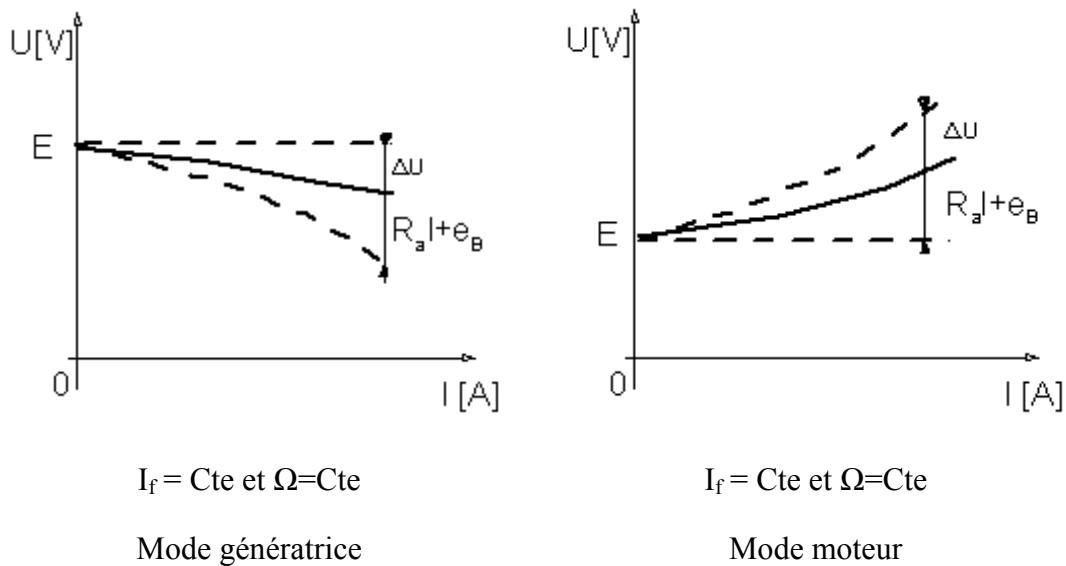


FIGURE 1.6 : Caractéristiques en charge de la machine

VI. Alimentation des machines à courant continu :

La commande de la vitesse des moteurs à courant continu s'effectue à travers l'alimentation de la machine qui joue généralement le rôle d'un interrupteur commandé associé avec un modulateur de largeur d'impulsion [7].

Les principaux convertisseurs utilisés pour la régulation des moteurs à courant continu sont :

- Les hacheurs : ce sont des convertisseurs continu-continu qui ont pour fonction de fournir une tension continue variable à partir d'une tension continue fixe. Il existe deux types de convertisseurs continu-continu ; ceux qui sont non isolés que l'on appelle les hacheurs (généralement constitué d'un pont de transistor) et ceux qui comportent un transformateur assurant l'isolation galvanique que l'on appelle les alimentations à découpage. Dans le cas de la figure (1.7) nous avons à faire à un hacheur en pont pour l'alimentation de la machine, le circuit de redressement pour l'alimentation du hacheur a aussi été représenté. Ce redresseur est nécessaire si l'alimentation est branché sur le réseau industriel, nous remarquons également la présence d'un modulateur de largeur d'impulsion qui est commandée par le correcteur de courant, ce dernier quant à lui synthétise une loi de commande pour le modulateur à partir de l'erreur.

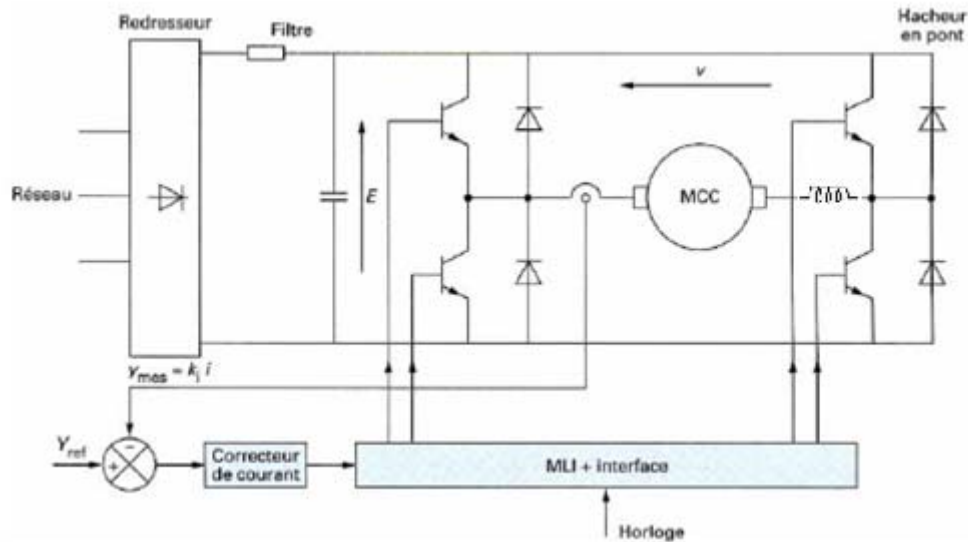


FIGURE 1.7 : Alimentation d'un moteur à courant continu par un hacheur en pont

- Le redresseur à thyristors : c'est également une solution pour l'alimentation des moteurs à courant continu, elle fournit une tension constante au moteur à chaque fois qu'un courant de gâchette apparaît. Elle joue donc le rôle d'interrupteur commandé et alimente en tension la machine à chaque fois qu'elle reçoit une impulsion de gâchette. Les types de commutateur utilisés sont des thyristors à séquence d'allumage commandable. L'allure du courant d'induit et de la tension d'induit en fonction du rapport cyclique est représentée dans la figure (1.8).

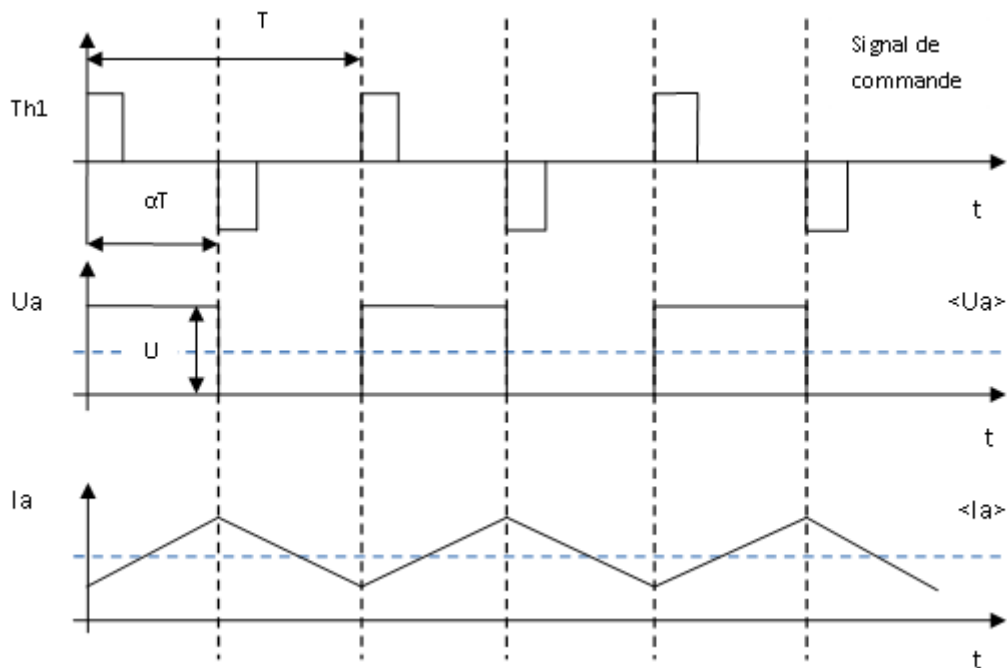


FIGURE 1.8 : Allure de la tension et du courant en fonction de l'allumage du thyristor

Chapitre 2

Modélisation en régime dynamique du moteur à courant continu

I. Introduction :

En régime de fonctionnement dynamique ou à vitesse variable, la machine est associée à des convertisseurs statiques et à des régulateurs. Il s'agira donc de déterminer les paramètres les mieux adaptés pour le contrôle du système ; que ce soit en vitesse qu'en courant. Cette synthèse ne peut se faire raisonnablement que dans le cadre des systèmes linéaires continus à coefficients constants en omettant dans un premier temps les non linéarités de la chaîne de régulation. Pour simplifier les équations, on ne va pas représenter les effets de la saturation ou de la réaction d'induit. On se ramènera donc à des équations différentielles linéaires à coefficients constants et à l'utilisation de la transformée de Laplace pour représenter le système.

II. Equations caractéristiques relatives à la modélisation :

a. La tension induite généralisée :

En prenant en compte la résistance R_a et l'inductance L_a du circuit d'induit, les résistances du collecteur, des balais et des connexions étant supposés constantes, l'équation de la tension induite s'écrit :

$$u_a(t) = R_a I(t) + \frac{d\Psi}{dt} = R_a I(t) + N_a \frac{d\Phi}{dt} \quad (2-1)$$

$$u_a(t) = R_a I(t) + L_a \frac{di_a(t)}{dt} + E(t) \quad (2-2)$$

N_a étant le nombre d'enroulements d'induit

Du coté inducteur on a l'équation suivante :

$$u_f(t) = R_f I_f(t) + \frac{d\Psi}{dt} = R_f I_f(t) + N_f \frac{d\Phi}{dt} \quad (2-3)$$

$$u_f(t) = R_f I_f(t) + L_f \frac{di_f(t)}{dt} \quad (2-4)$$

$$L_f = N_f \frac{d\Phi}{dt} \quad (2-5)$$

N_f Le nombre d'enroulements de l'inducteur

Chapitre 2 : Modélisation en régime dynamique du moteur à courant continu

On en déduit l'inductance mutuelle entre l'inducteur et l'induit qui est égale à :

$$L_{af} = \frac{\Phi - L_f I_f}{I} \quad (2-6)$$

Car d'après la définition du flux total, flux qui découle elle-même de l'inductance ; le flux total traversant un enroulement « i » est fonction du flux provenant de l'auto-inductance et de celle provenant de l'inductance mutuelle.

$$\Phi_i = \Phi = \Phi_{ii} + \Phi_{ij} = L_f I_f + L_{af} I$$

b. Le couple électromagnétique:

Comme on l'a défini auparavant l'expression du couple électromagnétique est donnée par l'équation :

$$C_{em} = K_{em} \Phi I \quad (2-7)$$

Associée à l'équation de la dynamique avec :

$J = J_m + J_{ch}$ l'inertie totale qui est la somme de l'inertie du moteur et celle de la charge.

C_{em} : le couple électromagnétique

f : le coefficient de frottement visqueux

C_r : le couple résistant

Ω : la vitesse de rotation en rad/sec

On a l'équation finale de la dynamique du système qui est égale à :

$$J \frac{d\Omega}{dt} + f\Omega = C_{em} - C_r \text{sign}(\Omega) \quad (2-8)$$

III. Modélisation de la machine :

a. Modélisation de la machine alimentée en tension :

On définit la constante électrique $K_E \Phi$ et la constante du couple $K_{em} \Phi$ comme étant égal au produit de l'inductance mutuelle et du courant d'excitation I_f .

$$K_{em} \Phi = K_E \Phi = L_{af} I_f$$

Chapitre 2 : Modélisation en régime dynamique du moteur à courant continu

Si le produit $C_r C_{em}$ est positif on a un fonctionnement en génératrice c'est-à-dire que le sens du couple résistant est le même que celui du couple électromagnétique.

Si par contre le produit $C_r C_{em} < 0$, on a un fonctionnement en moteur.

D'après les équations caractéristiques de la machine on obtient les expressions suivantes après transformation de Laplace :

$$\Omega(p) = \frac{1}{Jp} [-f\Omega(p) + K_E \Phi I(p) - C_r \text{sign}(\Omega)] \quad (2-9)$$

$$U(p) = R_a I(p) + L_a p I(p) + E(p) \quad (2-10)$$

Comme le système est un système multivariable il possède de ce fait plusieurs entrées et sorties. On sait que :

$$I(p) = \frac{U(p) - E(p)}{L_a p + R_a} \quad (2-11)$$

$$\Omega(p) = \frac{C_{em} - C_r \text{sign}(\Omega)}{Jp + f} \quad (2-12)$$

D'après les relations précédentes on obtient le schéma bloc suivant :

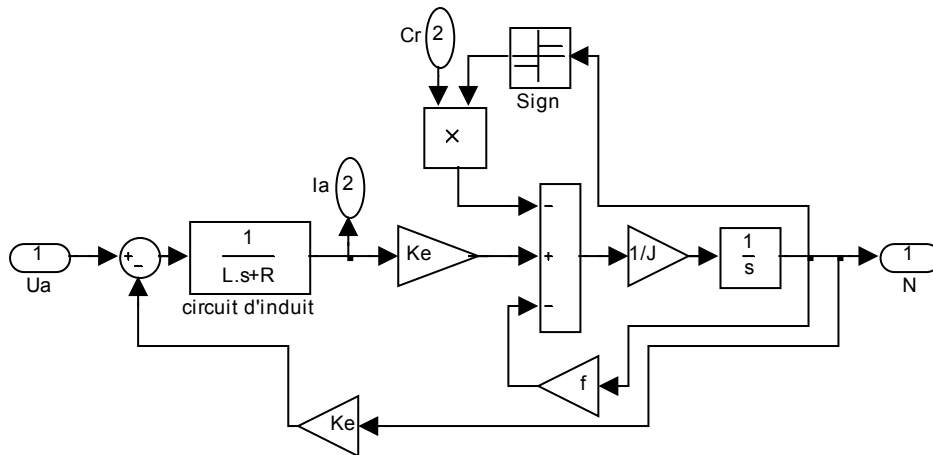


FIGURE 2.1 : Modèle du moteur à courant continu

Chapitre 2 : Modélisation en régime dynamique du moteur à courant continu

Généralement la fonction de transfert $G(p)$ d'un moteur à courant continu est définie comme suit :

$$G(p) = \frac{\Omega(p)}{U(p)} = \frac{K_m}{\tau_e \tau_{em} p^2 + (\tau_{em} + \mu \tau_e) p + 1} \quad (2-13)$$

Avec comme valeur :

- Du gain statique $K_m = \frac{K_E \Phi}{R_a f + (K_E \Phi)^2}$
- De la constante de temps électromécanique $\tau_{em} = \frac{R_a J}{R_a f + (K_E \Phi)^2}$
- De la constante de temps électrique $\tau_e = \frac{L_a}{R_a}$
- Du coefficient du moteur $\mu = \frac{R_a f}{R_a f + (K_E \Phi)^2}$

Pour l'étude proprement dite nous utiliserons le modèle plus compact suivant, il diffère du précédent par le fait que le couple résistant C_r est indépendant de la vitesse de sortie Ω . L'opérateur $\text{sign}(\Omega)$ sera donc éliminée et nous auront le modèle qui suit :

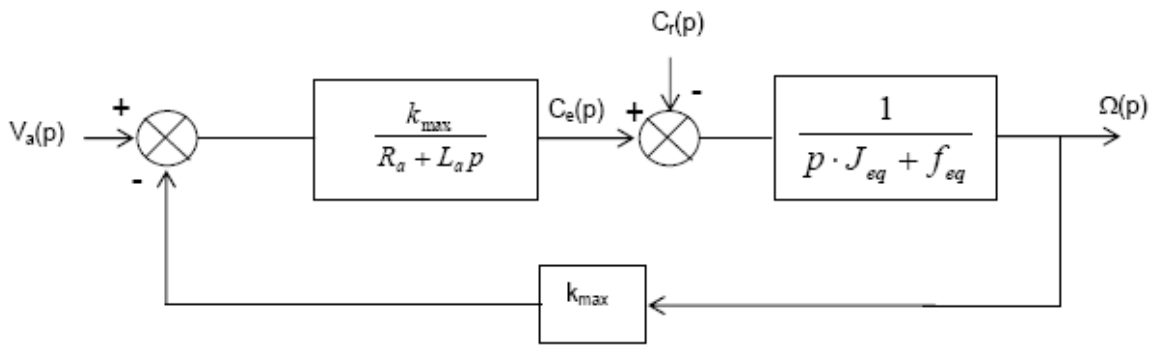


FIGURE 2.2 : Modèle théorique du moteur à courant continu

Avec $U(p) = V_a(p)$ la tension de référence et d'équation :

$$U(p) = R_a I(p) + L_a p I(p) + E(p) \quad (2-14)$$

$k_{\max} = K_E \Phi$ le coefficient de couple et également le coefficient de la f.é.m.

Chapitre 2 : Modélisation en régime dynamique du moteur à courant continu

Le couple électromagnétique est donné par la formule:

$$C_e(p) = C_{em}(p) = K_E \Phi I(p)$$

$C_r(p)$ étant le couple résistant

L'expression de la vitesse de rotation $\Omega(p)$ est :

$$\Omega(p) = \frac{C_{em} - C_r}{Jp + f} \quad (2-15)$$

b. Modélisation de la machine à courant continu pour la régulation de courant :

La détermination des correcteurs de courant utilise, dans la majorité des applications, une modélisation spécifique de la boucle de courant. Pour la régulation du courant on utilise un correcteur Proportionnel Intégral (PI) associé avec un modulateur de largeur d'impulsion (PWM) qui sera introduit à l'intérieur de la boucle.

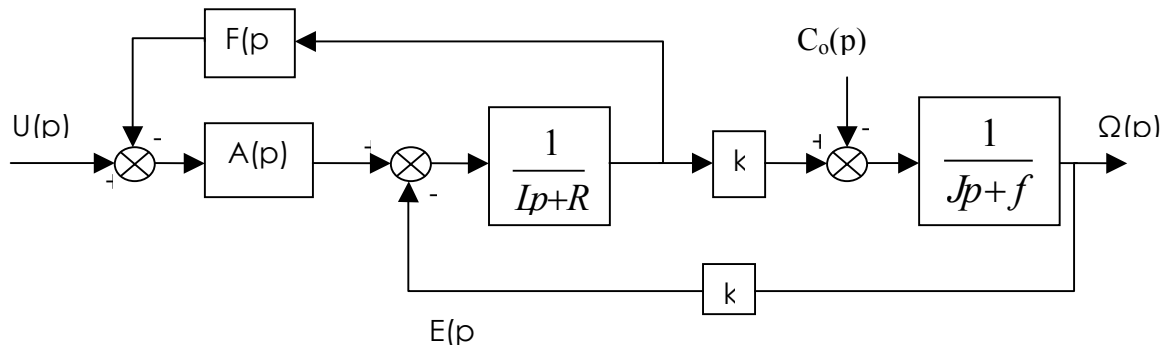


FIGURE 2.3 : *Modèle de la machine pour la régulation en courant*

La régulation de courant met en évidence la présence de capteur de courant, plus précisément une sonde à effet Hall qui aura pour fonction de transfert :

$$F(p) = \frac{V(p)}{I(p)} = k_I \quad (2-16)$$

Et il est souvent nécessaire d'introduire un filtre passe bas de premier ordre pour lisser le courant ; ce filtre aura pour fonction de transfert :

$$F(p) = \frac{k_I}{1 + bp} \quad (2-17)$$

Chapitre 2 : Modélisation en régime dynamique du moteur à courant continu

On utilise pour la régulation de courant un correcteur de courant de type PI ; de fonction de transfert $A(p)$ qui est la suivante:

$$A(p) = A \frac{1 + \tau p}{p} \quad (2-18)$$

Finalement, la chaîne de régulation sera modélisée par la fonction de transfert ci-après, avec $k = K_E \Phi$

$$\frac{\Omega(p)}{U(p)} = \frac{\frac{k}{k_I f} (1 + \tau p)}{1 + \frac{k^2 + Rf + k_I AJ + k_I A \tau f}{k_I A f} p + \frac{RJ + Lf + JA \tau k_I}{k_I A f} p^2 + \frac{LJ}{k_I A f} p^3} \quad (2-19)$$

c. Modélisation réelle de la machine à courant continu :

Le modèle réel de la machine à courant continu a été extraite du Toolbox 'SimPowerSystems' de MATLAB [16]; elle représente la machine sous sa forme la plus complète.

Comme c'est un système multivariable, il dispose de plusieurs entrées qui sont respectivement les bornes d'alimentation de l'induit notées 'A+A-', les bornes de l'alimentation de l'inducteur 'F+F-', une entrée spéciale pour la mesure du couple de charge notée 'T_L' et une sortie 'm' où seront multiplexés les valeurs des grandeurs de sortie en l'occurrence la vitesse, le courant d'induit 'i_a', le courant d'excitation 'i_f' et le couple électromagnétique 'T_e'.

• La partie électrique : Le circuit électrique du moteur comporte deux sous-circuits :

- *Le circuit de champ* : qui regroupe les éléments du circuit inducteur de la machine, elle comprend la résistance R_f de l'inducteur et son inductance L_f . Ces éléments se trouvent entre les bornes de 'F+F-'

- *Le circuit d'armature* : qui est connecté entre les bornes A+ et A- de la machine, est composé essentiellement de la résistance d'induit R_a en série avec l'inductance L_a et, est branché avec une source de tension qui représentera la f.é.m. de la machine.

Chapitre 2 : Modélisation en régime dynamique du moteur à courant continu

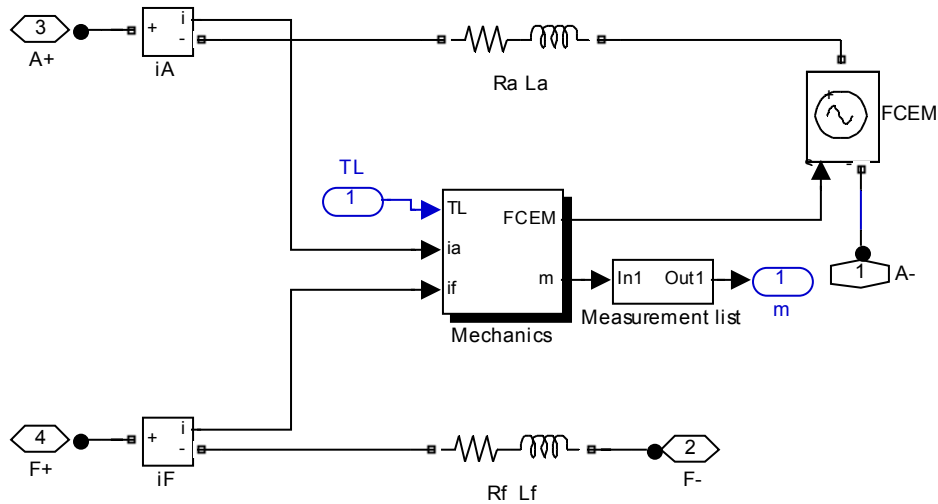


FIGURE 2.4 : Partie électrique de la machine

- La partie mécanique quant à elle transforme le couple appliqué au rotor de la machine en sa vitesse de rotation. Elle comprend surtout les expressions des frottements internes à la machine, des couples de frottements externes due à la charge ainsi que celle de l'inertie de l'arbre et celle de la charge.

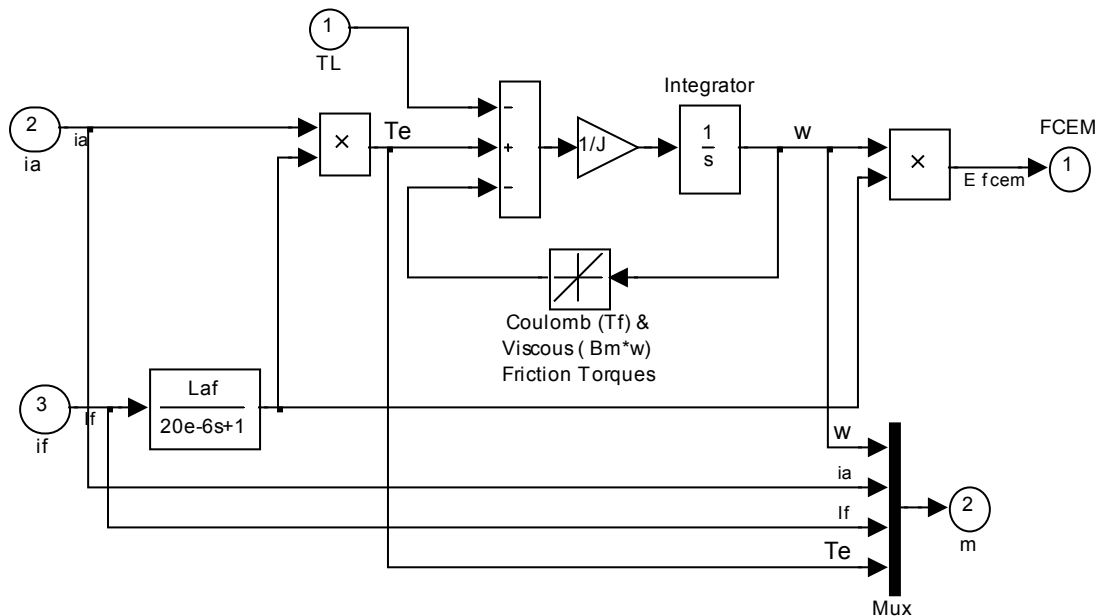


FIGURE 2.5 : Partie mécanique du moteur

Chapitre 3

Principe de la régulation de vitesse des moteurs à courant continu

IV. Introduction :

La régulation automatique est la technique qui offre les outils et les méthodes nécessaires à la prise de contrôle d'une ou de plusieurs grandeurs physiques d'un système, en vue d'en imposer son comportement. Les grandeurs physiques représentées sous formes de signaux doivent être mesurées afin de vérifier leurs états, il consistera ensuite de déterminer à l'aide d'un traitement approprié l'action à entreprendre sur le processus pour qu'elle se comporte comme souhaitée.

Une régulation contrairement à un asservissement dispose d'une entrée de référence constante ou évoluant par palier, cette entrée est appelée consigne et la sortie du processus ou sa réponse doit rester constante et égale à la consigne, quelles que soient les perturbations.

Dans notre cas la consigne est la vitesse de référence et la grandeur à réguler est la vitesse de sortie du moteur. Remarquons que le moteur à courant continu est un système physique donc il possède ses propres limites, pour cela notre régulateur aura à part sa fonction principale de régulation de vitesse, une fonction supplémentaire qui sera d'assurer la protection du processus.

V. Principe général de la régulation :

Nous nous intéresserons surtout à la régulation de vitesse, dans l'industrie on a besoin d'entraîner une charge à vitesse constante ou de positionner une pièce à un emplacement exact malgré les éventuelles perturbations qui peuvent s'exercer sur la machine. Il est donc impératif de faire en sorte que la sortie du moteur soit insensible aux distorsions et reste identique à la consigne introduite par l'opérateur.

a. Les éléments de la chaine de régulation :

La chaine de régulation est constituée des éléments suivants :

- *Un capteur* : dans notre cas nous disposons de deux types de capteurs, le premier est le capteur de vitesse qui est généralement le dynamo-tachymètre. C'est une petite génératrice à courant continu délivrant une tension proportionnelle à la vitesse mesurée ; le second est le capteur de courant qui peut être une simple sonde à effet Hall, elle consiste à détecter la valeur du courant I_a dans l'induit de la machine.

Chapitre 3 : Principe de la régulation de vitesse des moteurs à courant continu

- *Un comparateur* qui calcule la différence entre la grandeur de consigne et la grandeur délivrée par le moteur ; cette grandeur peut être tout aussi bien un courant qu'une vitesse à condition bien sûr qu'elle ait été transformée en tension, seul signal mesurable par la plupart des comparateurs, il peut être modélisé par un simple amplificateur opérationnel en montage soustracteur.
- *Un actionneur* qui est dans notre cas le moteur à courant continu.
- *Un amplificateur* qui aura pour rôle de multiplier le signal d'erreur par un gain fixe afin qu'il soit suffisamment élevé pour piloter le moteur.
- *Les correcteurs de vitesse et de courant* qui synthétisent à partir de l'erreur une loi de commande qui sera injecté au moteur.

b. Concept de l'asservissement de vitesse :

Le but de ce paragraphe est de décrire de manière intuitive le fonctionnement d'une boucle d'asservissement de vitesse. On admettra ici pour simplifier l'exposé que le retour est unitaire, c'est à dire que la génératrice tachymétrique se comporte comme un système à gain pur unitaire.

Première approche :

A accélération faible, la vitesse $\omega(t)$ augmente doucement et se rapproche de la consigne Ω_0 . On obtient une évolution du type de celle représentée figure (3.1).

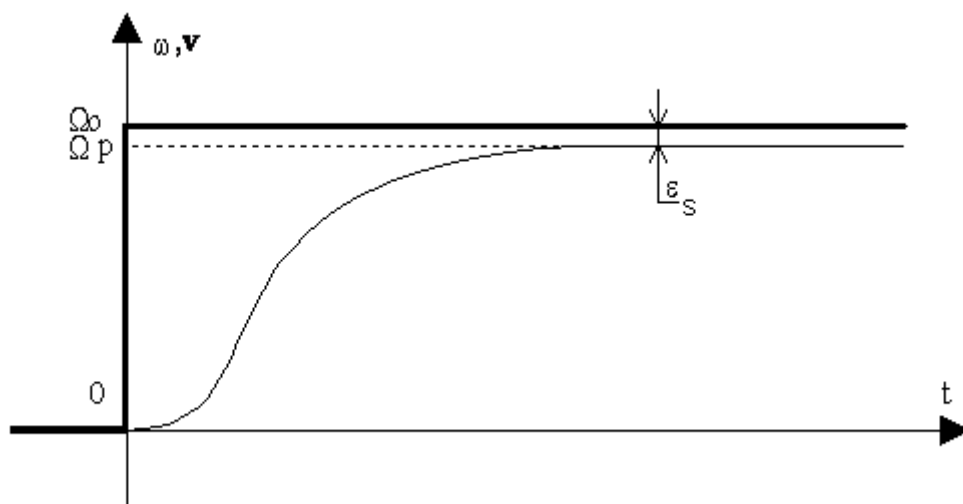


FIGURE 3.1 : Réponse indicielle d'un moteur à courant continu

On remarque que la vitesse effective $\omega(t)$ n'atteint jamais la vitesse visée Ω_0 mais tend vers une limite inférieure $\Omega_p < \Omega_0$. Il reste donc un écart en régime permanent que l'on définit comme étant l'écart statique ε_s .

Chapitre 3 : Principe de la régulation de vitesse des moteurs à courant continu

Cet écart statique reste non nul parce qu'il faut que le moteur tourne pour que la vitesse Ω puisse exister, il faut nécessairement fournir une commande de vitesse non nulle. La commande $c(t)$ étant engendrée par l'écart $\varepsilon(t)$, ce dernier ne peut pas être nul.

Lorsqu'il existe un couple résistant sur l'arbre moteur, dû aux frottements par exemple cet écart est aggravé.

Le but de l'asservissement est donc de diminuer l'écart ε_s ce qui aura pour effet de rapprocher la valeur de la vitesse effective Ω_p de celle de la valeur visée Ω_0 .

Dans notre exemple, si l'on augmente le gain de l'amplificateur d'un facteur deux fois plus important on obtiendra le même signal de commande à partir d'un signal d'écart deux fois plus petit. Et en raisonnant de la même manière on constate qu'il faudra augmenter le gain vers l'infini pour que l'erreur soit nulle ; ce qui est techniquement irréalisable.

D'autre part supposons que l'on puisse en effet maximiser le gain de la chaîne autant que possible, on constatera que cela rend le système instable et en plus le système devient très nerveux ; pour un signal d'écart ε_s très petit, le système va fournir un signal de commande $c(t)$ suffisant pour entretenir la vitesse de rotation et le problème se trouve lors du démarrage quand la différence consigne-vitesse réelle est au maximum on observera alors une commande exagérée.

- Deuxième approche :

Selon le principe fondamental de la dynamique l'accélération est importante au démarrage et la vitesse croît rapidement : on obtiendra une évolution du type de celle représentée figure (3.2).

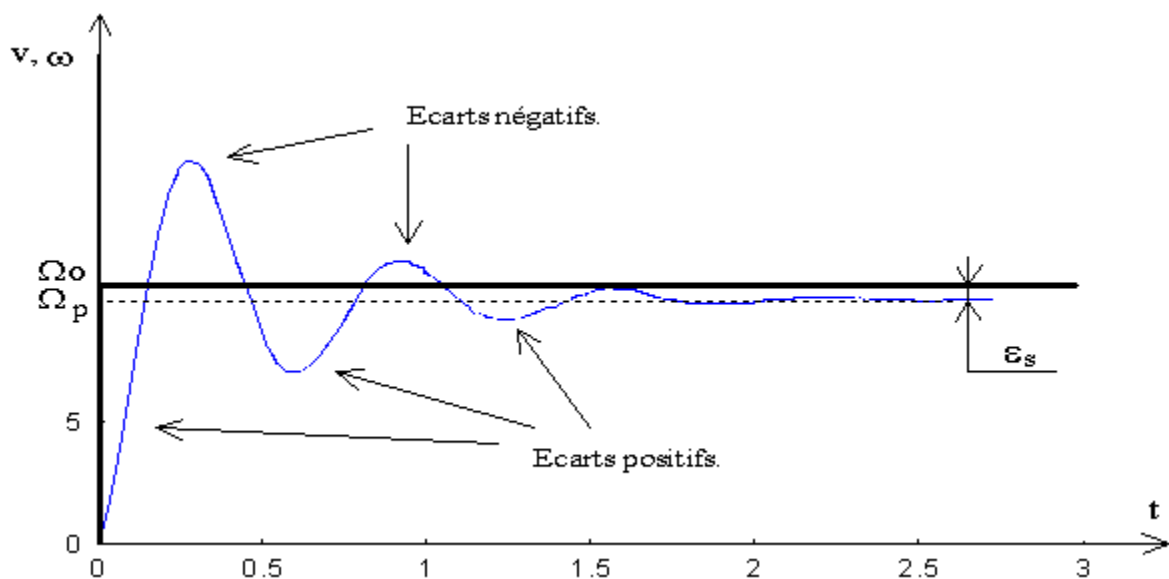


FIGURE 3.2 : Mouvement oscillatoire de la réponse

Chapitre 3 : Principe de la régulation de vitesse des moteurs à courant continu

La vitesse $\omega(t)$ augmente trop vite et dépasse la valeur visée avant que le système ne réagisse. La vitesse réelle $\omega(t)$ est alors plus grande que la valeur visée Ω_0 et la différence de vitesse ou erreur $\varepsilon(t) = \Omega_0 - \omega(t)$ devient négative.

La commande $c(t)$ du moteur change de signe, ce qui va créer un couple négatif et provoquer le passage du fonctionnement en moteur à un fonctionnement en frein.

La vitesse $\omega(t)$ diminue, repasse sous Ω_0 et l'écart redevient positif. On revient à un fonctionnement en moteur. Après une succession d'oscillations, la vitesse $\omega(t)$ tend vers une limite Ω_p qui n'est pas identique à celle du cas de l'approche 1, cette limite est fonction de la valeur du gain de la chaîne de commande.

Ce type de comportement oscillatoire est mal toléré par les organes mécaniques de la machine ; création de couple de torsion etc.....et il y aura également création de surcouple au démarrage due à la grande différence entre la vitesse réelle $\omega(t)$ et la vitesse visée Ω_0 .

Pour conclure donc un asservissement de vitesse est par nature imprécis, il existe toujours un écart statique ε non nul inévitable en l'absence de correcteur et par suite afin de minimiser cet écart il faut augmenter le gain de la chaîne de commande sans provoquer pour autant un régime oscillatoire.

VI. Structure de régulation d'un moteur à courant continu:

a. Structure de régulation en parallèle :

Le commutateur prend la valeur la plus faible entre V_{a1} et V_{a2} . Lorsque le capteur de courant détecte un courant largement différent de la valeur du courant de référence, l'erreur ε_i entrant dans le régulateur de courant est grande ; ce qui induit que la valeur de la tension de sortie V_{a2} atteindra une certaine saturation $V_{a2\max}$ et comme cette valeur est supérieure à V_{a1} le commutateur choisira V_{a1} donc le régulateur de vitesse fonctionnera seul ; le système procèdera de la même façon dans le cas contraire mais, dans ce cas là seul le régulateur de courant fonctionnera.

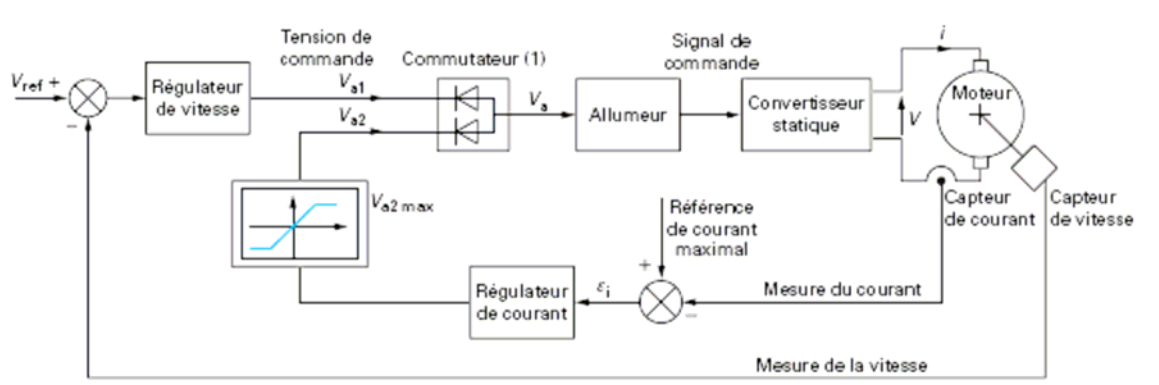


FIGURE 3.3 : Régulation en parallèle d'un moteur à courant continu

b. Structure de régulation en cascade :

Le principe de la régulation en cascade est d'insérer, à l'intérieur d'une boucle de régulation, une boucle interne permettant de compenser l'inertie du processus régulé. Ce montage n'est toutefois possible que si le processus est lui-même décomposable en deux éléments placés en cascade. Il est possible de réaliser le montage suivant avec deux régulateurs autonomes pourvu que la consigne de celui affecté à la boucle interne, c'est-à-dire le régulateur de courant, puisse être le signal de sortie du régulateur principal externe ; le régulateur de vitesse.

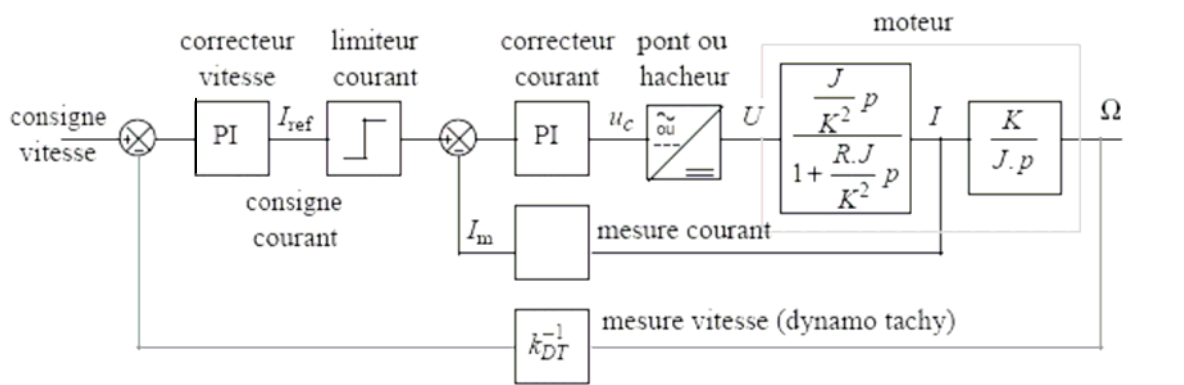


FIGURE 3.4 : Régulation en cascade d'un moteur à courant continu

Dans notre étude le processus sera le moteur à courant continu ; le régulateur de courant servira de commande au hacheur et le régulateur de vitesse associé au limiteur de courant servira de signal de référence de la boucle de courant. La boucle interne sera donc la boucle de courant et celle externe la boucle de vitesse. Pour simplifier le schéma le couple résistant a été omis volontairement mais elle n'en est pas pour autant nulle.

Chapitre 4

Modélisation de la chaîne de régulation

I. Introduction :

Nous utiliserons le montage de la figure (3.4) pour asservir la vitesse du moteur à courant continu. Les gains des capteurs de vitesse et de courant seront supposés être égaux à l'unité. Nous aurons donc à dimensionner les principaux composants suivants :

- Le moteur à courant continu
- L'alimentation du moteur
- Le correcteur de vitesse
- Le correcteur de courant

II. Modélisation du moteur à courant continu :

Nous utiliserons en premier temps le modèle classique du moteur à courant continu. D'après les caractéristiques du moteur donné en annexe (Annexe2) nous pouvons reconstituer sa fonction de transfert $G(p)$, elle est décrite comme suit :

$$G(p) = \frac{\Omega(p)}{U(p)} = \frac{K_m}{\tau_e \tau_{em} p^2 + (\tau_{em} + \mu \tau_e) p + 1} \quad (4-1)$$

Avec comme valeur :

Du gain statique $K_m = \frac{K_E}{R_a f + (K_E \Phi)^2} = 1.577$

De la constante de temps électromécanique $\tau_{em} = \frac{R_a J}{R_a f + (K_E \Phi)^2} = 0.0745 \text{ [s]}$

De la constante de temps électrique $\tau_e = \frac{L_a}{R_a} = 0.03 \text{ [s]}$

Du coefficient du moteur $\mu = \frac{R_a f}{R_a f + (K_E \Phi)^2} = 0.0377$

Chapitre 4 : Modélisation de la chaîne de régulation

D'où l'expression de $G(p)$:

$$G(p) = \frac{\Omega(p)}{U(p)} = \frac{1.577}{0.0022p^2 + 0.075p + 1} \quad (4-2)$$

Notre moteur a un couple résistant constant égal à 0.0645 [N.m] qui sera modélisé par un bloc « Constant » de valeur 0.0645.

Modélisé sous SIMULINK nous avons le schéma bloc qui suit:

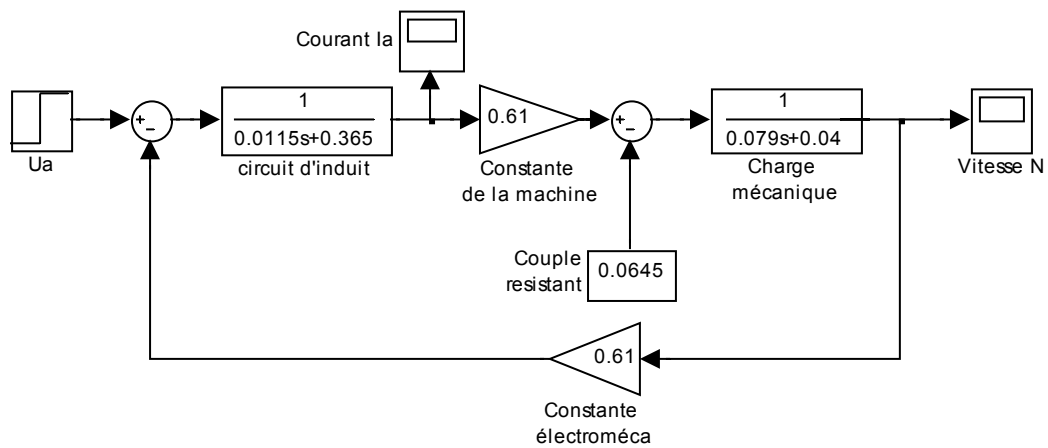


FIGURE 4.1 : Modèle du moteur à courant continu.

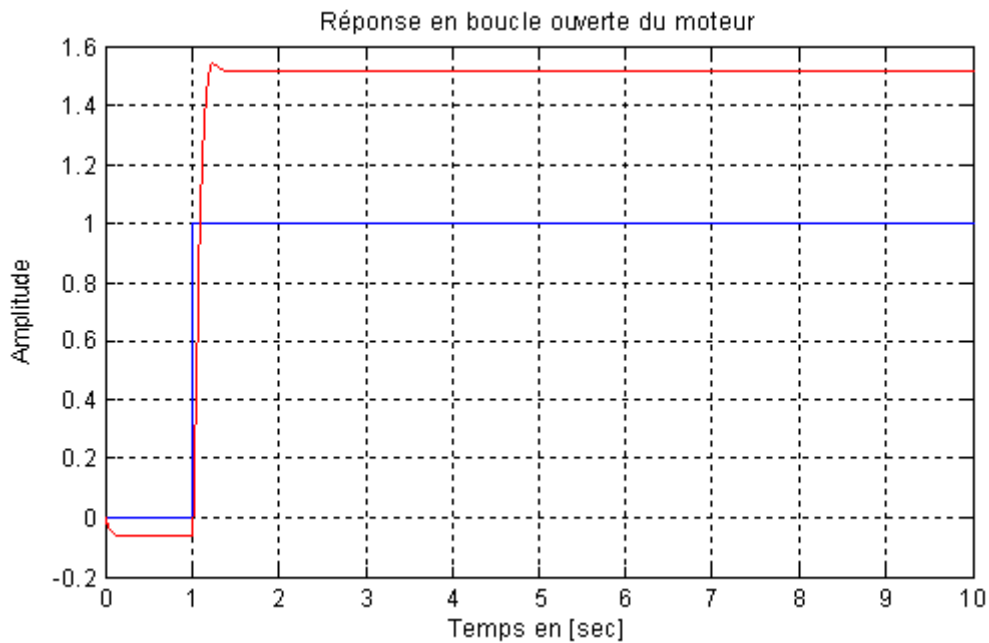


FIGURE 4.2 : Réponse en boucle ouverte du moteur

Chapitre 4 : Modélisation de la chaîne de régulation

La réponse indicielle du moteur en boucle ouverte nous permet de définir le gain de la chaîne de régulation, c'est-à-dire le gain statique du moteur car c'est le seul élément de la chaîne. On constate sur la figure (4.2) que le gain statique du moteur est égal à environ 1.5.

Pour l'étude du comportement dynamique du système nous avons fait varier le couple résistant pour observer la réponse en vitesse du moteur et son comportement vis-à-vis des perturbations. Pour simuler le couple résistant on a créé à partir d'une suite d'échelon, un bloc « Cr » modélisé comme suit:

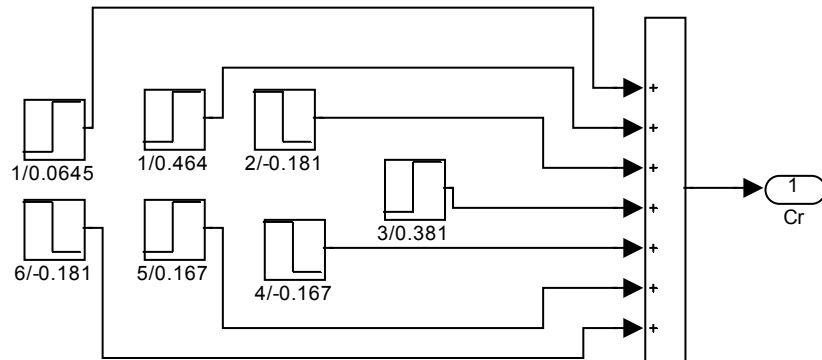


FIGURE 4.3 : Modèle du couple résistant

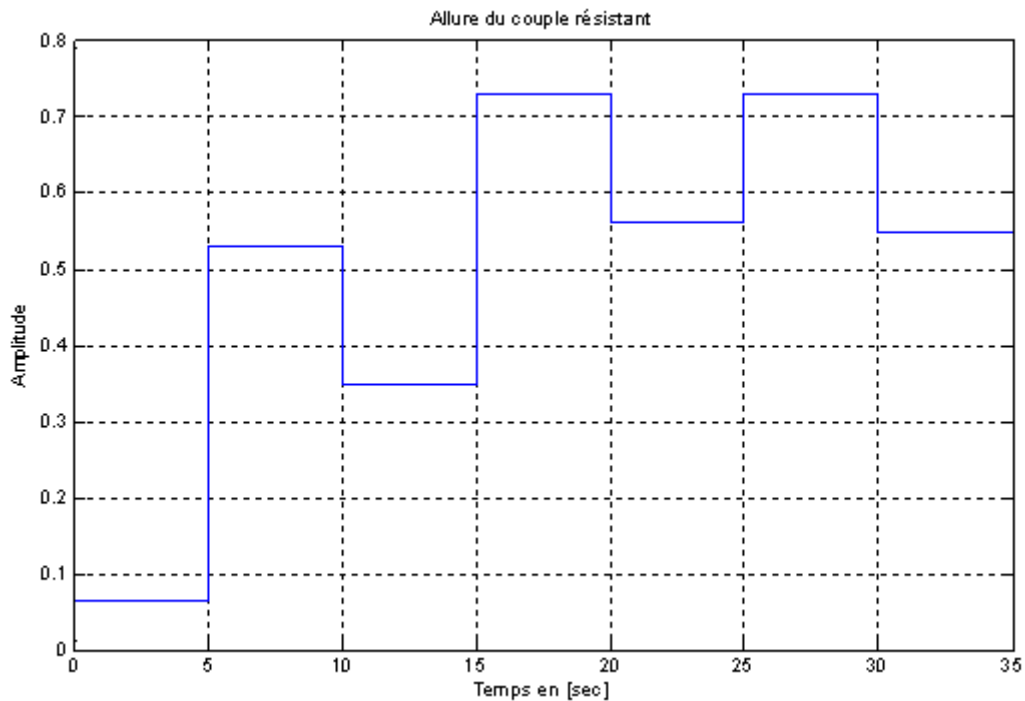


FIGURE 4.4 : Allure du couple résistant

Chapitre 4 : Modélisation de la chaîne de régulation

Le moteur est maintenant en charge et est soumis à un couple résistant variable en boucle ouverte.

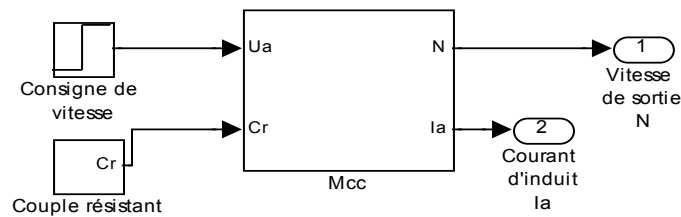


FIGURE 4.5 : Modèle macro du moteur à courant continu

On constate sur la réponse en boucle ouverte du moteur figure (4.6) que la variation de la vitesse de sortie est inversement proportionnelle à la variation du couple résistant; ceci est dû au fait que le couple résistant est un couple antagoniste, c'est-à-dire qu'il est de signe contraire au couple électromagnétique (ceci est vérifié à l'équation de la dynamique du système (2-8)).

Une augmentation du couple résistant, se traduit par la diminution du couple moteur, ce qui provoque une baisse de la vitesse et inversement dans le cas contraire.

D'où l'allure de la vitesse suivante :

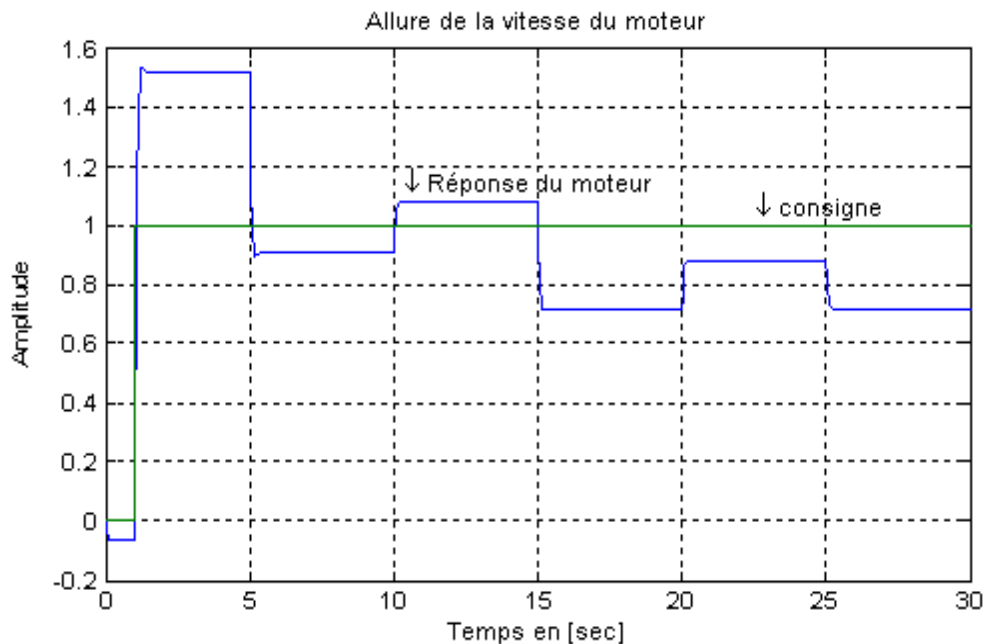


FIGURE 4.6 : Réponse indicielle du moteur en boucle ouverte avec variation de Cr

III. Modélisation de l'alimentation du moteur :

Le convertisseur est souvent modélisé par une fonction de transfert simplifiée du premier ordre [7] donnée comme suit :

$$\frac{\langle V(p) \rangle}{U(p)} = \frac{G_0}{1 + \tau p} \quad (4-3)$$

Avec : G_0 gain statique du convertisseur généralement elle est égale à 1

τ : constante de temps du convertisseur

$U(p)$: tension de commande du convertisseur

$\langle V(p) \rangle$: tension de sortie du convertisseur en valeur moyenne

Des essais expérimentaux sur différentes machines à courant continu ont permis de fixer la valeur de la fréquence de hachage du convertisseur ; elle varie selon le type de régulateur utilisé ou suivant le type de carte utilisé. On obtient le tableau suivant :

Carte d'acquisition	ISBC	TMS
Correcteur P	f=12kHz	f=15kHz
Correcteur PI	f=10kHz	f=12kHz
Correcteur P avec compensateur de retard	f=3kHz	f=5kHz
Correcteur PI avec compensateur de retard	f=2.5kHz	f=4kHz

TABLEAU 4.1 : Fréquence de hachage du convertisseur statique

Notre choix se portera sur la fréquence de 10kHz du processeur ISBC associé avec un correcteur de courant PI. On aura la fonction de transfert finale :

$$\frac{\langle V(p) \rangle}{U(p)} = \frac{1}{1 + 1e - 4p} \quad (4-4)$$

Le montage en boucle fermée du moteur avec retour unitaire et avec couple résistant variable est représenté par la figure (4.7) qui suit.

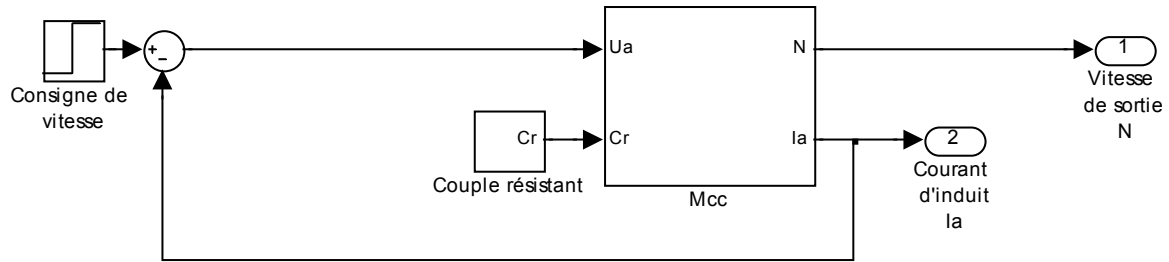


FIGURE 4.7 : Montage en boucle fermée du moteur avec retour unitaire

La réponse indicielle du moteur en boucle fermée à couple résistant constant est donnée par la figure suivante ; on constate que le gain statique de la chaîne de régulation en boucle fermée est d'environ 0.6.

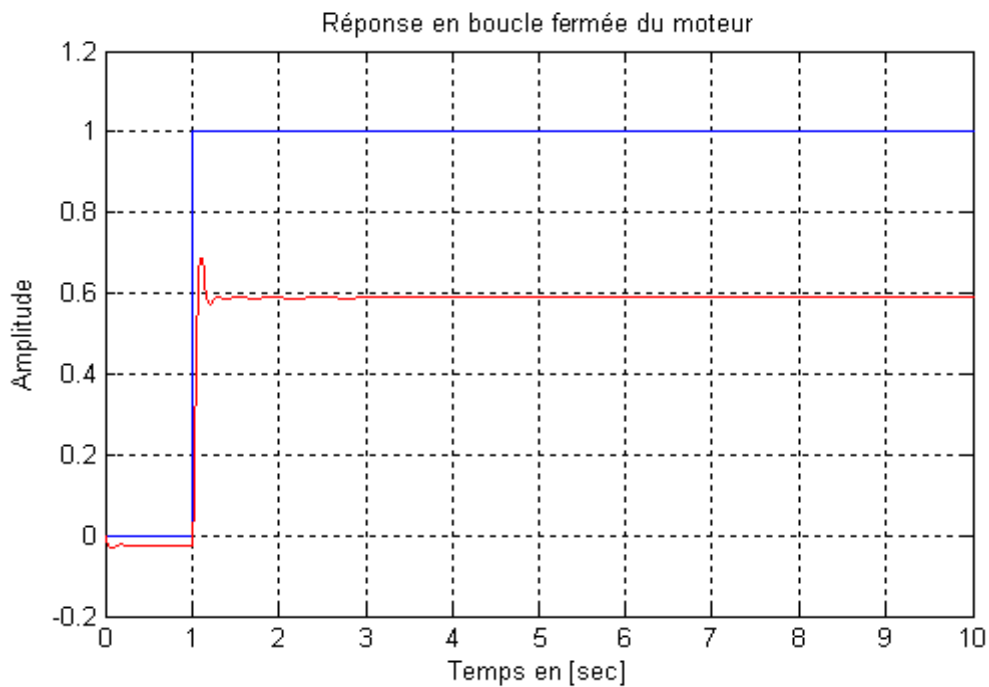


FIGURE 4.8 : Réponse en vitesse du moteur en boucle fermée à retour unitaire

La variation de la vitesse de la figure (4.9) pour le moteur chargé avec un couple résistant variable est identique à celle de la figure (4.6) mis à part le fait que le système devient plus précis (boucle fermée), mais cependant il existe toujours l'effet du couple résistant sur la vitesse. Cela se traduit par le fait que le système est encore instable et que chaque variation de la perturbation modifie l'état de la vitesse. Pour parer à ce phénomène il est nécessaire d'introduire des correcteurs.

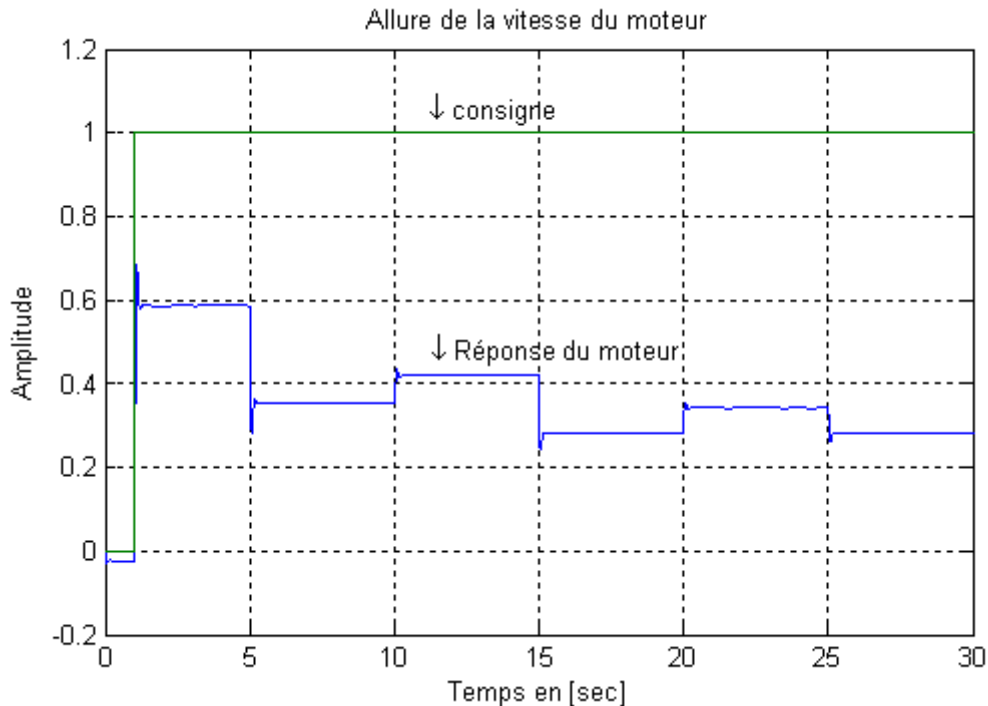


FIGURE 4.9 : Réponse en boucle fermée du moteur avec variation de charge

IV. Synthèse du correcteur de vitesse :

Le type de correcteur utilisé sera le correcteur proportionnel intégral (PI) du fait qu'un comportement intégrateur est nécessaire de façon à ce que le système soit insensible aux perturbations de couple que sont par exemple le frottement sec ou la gravité ; l'intégrateur est donc utilisé pour annuler l'erreur statique du système.

Le choix de l'action proportionnelle quand à elle, est nécessaire pour améliorer le comportement dynamique du système. Ziegler et Nichols ont conçus une méthode empirique de détermination des paramètres d'un correcteur PID, elle peut se faire soit par l'observation de la réponse indicielle du processus en boucle ouverte, soit par la mise en oscillation volontaire du système en boucle fermée pour relever sa période d'oscillation.

Nous utiliserons la mesure sur l'essai indicielle pour modéliser le correcteur de vitesse. Le processus étant modélisé par sa fonction de transfert on trace sa réponse indicielle en boucle ouverte ; on relève sur l'allure de la vitesse les caractéristiques du régulateur, c'est une méthode empirique donc la possibilité d'erreur de mesure n'est pas à exclure ; mais elle offre une méthode simple et rapide de synthèse de correcteur.

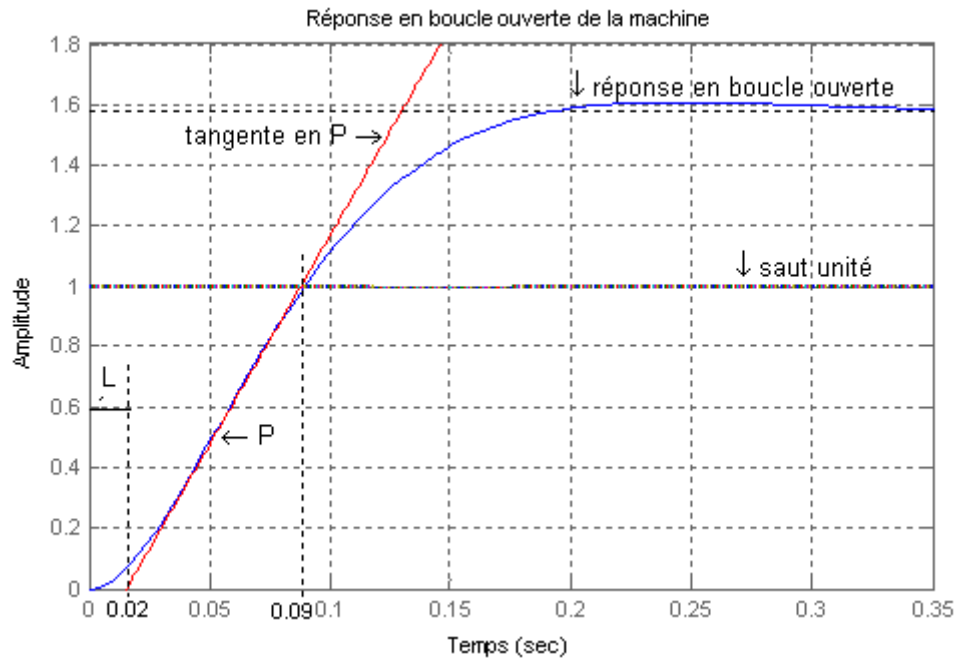


FIGURE 4.10 : Synthèse du correcteur par la méthode de Ziegler Nichols

Nous utiliserons pour réguler la vitesse du moteur à courant continu un régulateur de type Proportionnel Intégral ; elle est définie par la fonction de transfert suivante :

$$C(p) = K_p \left(1 + \frac{1}{T_i p}\right) \quad (4-5)$$

Avec K_p le gain proportionnel

Et $K_i = \frac{K_p}{T_i}$ le gain de l'intégral

Les paramètres du correcteur en fonction de son type sont regroupés dans le tableau suivant :

Type de correcteur	K_p	T_i	T_d
Proportionnel	$\frac{1}{aL}$	-	-
Proportionnel Intégral	$\frac{0.9}{aL}$	$3.3L$	-
Proportionnel Intégral Dérivé	$\frac{1.2}{aL}$	$2L$	$0.5L$

TABLEAU 4.2 : Tableau des paramètres d'un régulateur PID selon Ziegler Nichols

Chapitre 4 : Modélisation de la chaîne de régulation

La valeur de L est directement lue sur la figure (4.10) on a :

$$L=0.018[s]$$

Pour celle de a ; elle se déduit comme étant la pente de la tangente au point d'inflexion P de la réponse indicielle du moteur en boucle ouverte ; on en déduit :

$$a = 12.5$$

En tenant compte de ces dernières valeurs et des formules du tableau (4.2) nous pouvons en tirer l'expression finale du correcteur de vitesse :

$$C(p) = 3.6\left(1 + \frac{1}{0.06p}\right) \quad (4-6)$$

Nous avons le montage suivant lorsque le moteur chargé est monté avec un correcteur de vitesse :

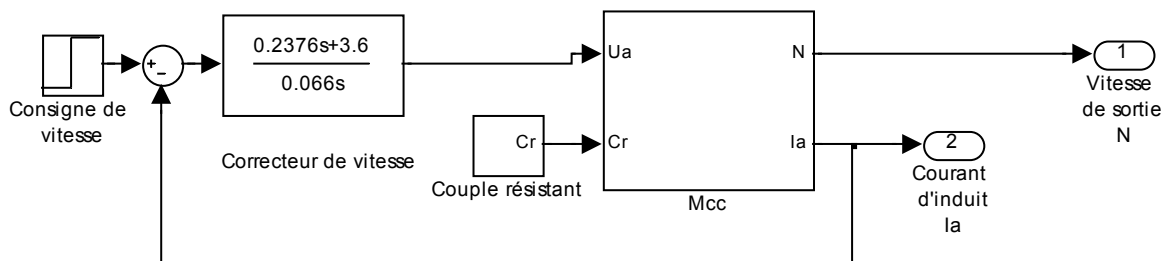


FIGURE 4.11 : Régulation avec correcteur PI de vitesse

Sur la réponse du moteur avec le correcteur de vitesse figure (4.11) on constate que la vitesse du moteur présente des oscillations et un dépassement à la limite des 50% de la valeur finale autorisée par le critère de l'asservissement, au démarrage. Les oscillations sont dues au fait que le courant d'induit n'est pas régulé, c'est-à-dire que le couple moteur est influencé par l'inertie de la partie mobile du moteur.

En conclusion : le régulateur de vitesse fonctionnant seul n'est pas adapté pour la régulation, il lui faut la mise en cascade d'un régulateur de courant.

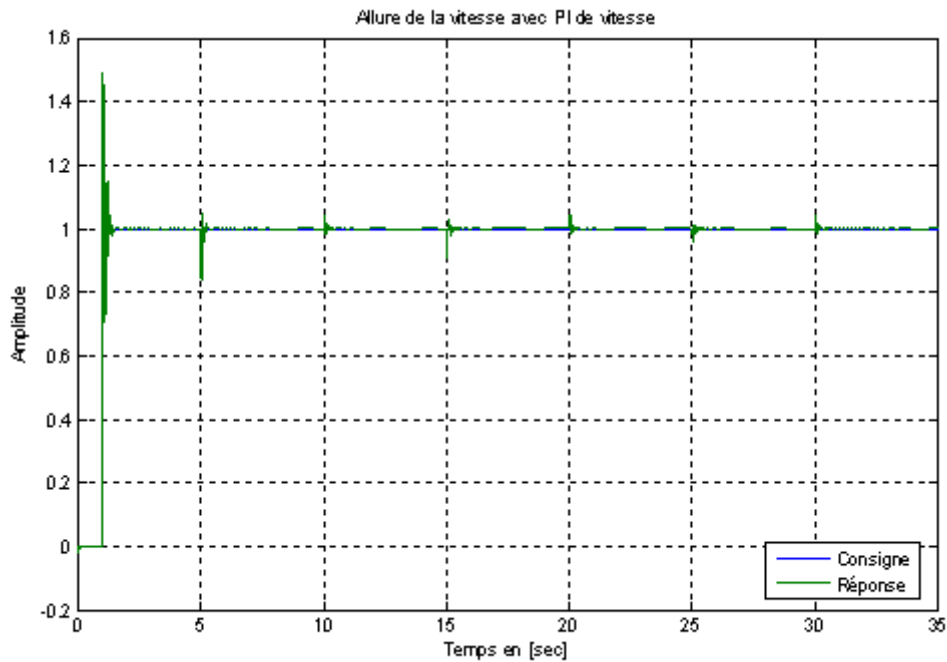


FIGURE 4.12 : Réponse indicielle du moteur pour la régulation de vitesse

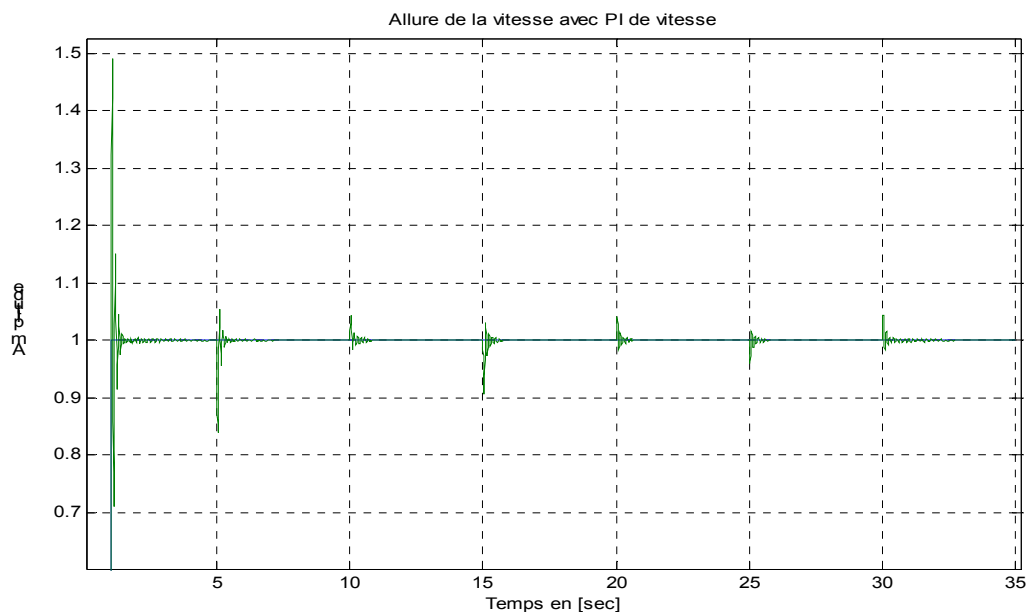


FIGURE 4.13 : Oscillation de vitesse du moteur à courant continu

En faisant un zoom entre les amplitudes de 0.6 à 1.5, on constate que des oscillations exagérées sont présents à chaque apparition d'un couple de distorsion, notons que les oscillations répétitives peuvent causer la détérioration des organes mécaniques de la machine par des efforts mécaniques de cisaillement, torsion etc., ceci est dû au fait qu'il y a création de couples de signes contraires à chaque fois qu'il existe une variation du signe de l'erreur.

V. Synthèse du correcteur de courant :

Le correcteur de courant utilisé est un correcteur de type proportionnel intégral (PI) ; il est monté à l'intérieur de la boucle de courant qui est une boucle interne à la boucle de vitesse, ce correcteur aura pour rôle de traiter l'erreur provenant du comparateur de courant, erreur qui est la différence entre le courant de référence délivré par le régulateur de vitesse et celui délivré par le moteur, et d'en tirer un signal de commande pour alimenter le moteur.

Lorsqu'on implante un correcteur de courant dans la boucle de régulation, la machine est excitée en courant. Nous avons le schéma de principe suivant :

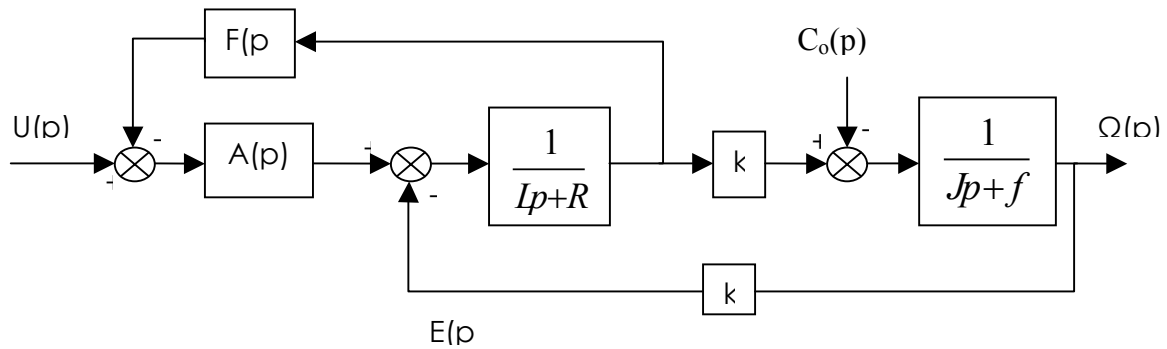


FIGURE 4.14 : Schéma de montage pour la régulation de courant

a. Dimensionnement du filtre de courant :

La constante 'b' du filtre est fonction de la fréquence de hachage du convertisseur utilisé pour l'alimentation de la machine, la fonction de transfert du filtre est donnée comme suit :

$$F(p) = \frac{k_f}{1+bp} \quad (4-7)$$

Généralement on prend comme fréquence de hachage du hacheur

$f_h = 10\text{kHz}$ et on a la constante 'b' qui est :

$$b > \frac{1}{2\pi f_h}$$

On obtient finalement l'expression du filtre.

$$F(p) = \frac{3.66}{1+1e^{-5}p} \quad (4-8)$$

b. Synthèse du correcteur de courant :

Le correcteur de courant est modélisé par sa fonction de transfert :

$$A(p) = A \frac{1 + \tau p}{p} \quad (4-9)$$

On choisira arbitrairement la constante $b = 1,6 \cdot 10^{-5}$ et l'amortissement $m = 0.7$

$$k_I \text{ étant égal } \frac{U_{\max}}{I_{\max}} = \frac{110}{30} = 3.66$$

$$\text{Le gain statique du correcteur de courant } A = \frac{R_a}{bk_I m^2} = 3180$$

$$\tau \text{ étant la constante de la machine } \tau = \frac{L_a}{R_a} = 0.03 \text{ [s]}$$

On obtient finalement l'expression du correcteur de courant :

$$A(p) = \frac{3180 + 95.4p}{p} \quad (4-10)$$

Le montage pour la régulation en cascade se présente sous la forme suivante :

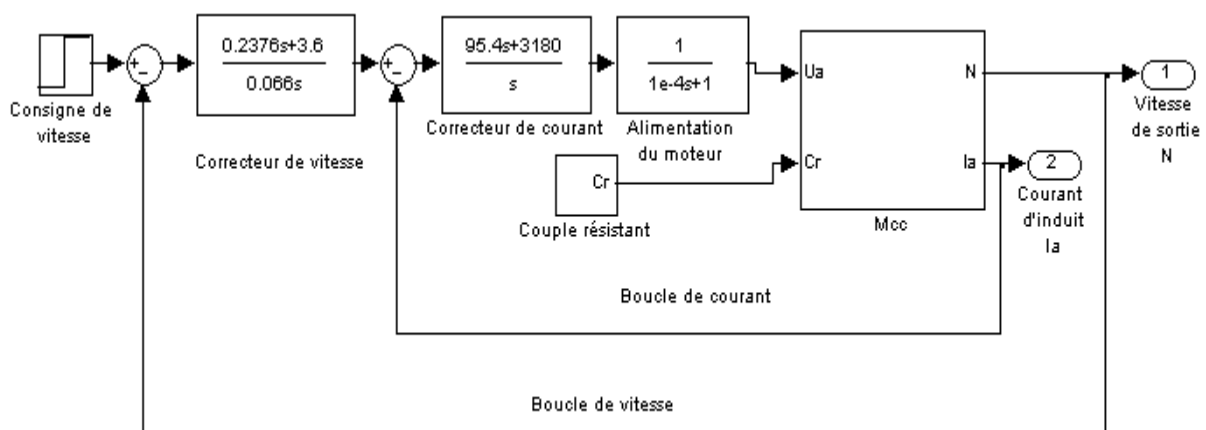


FIGURE 4.15 : Régulation en cascade pour l'asservissement de vitesse du moteur à courant continu

Chapitre 4 : Modélisation de la chaîne de régulation

Ce montage de la figure (4.15) donne la réponse indicielle suivante :

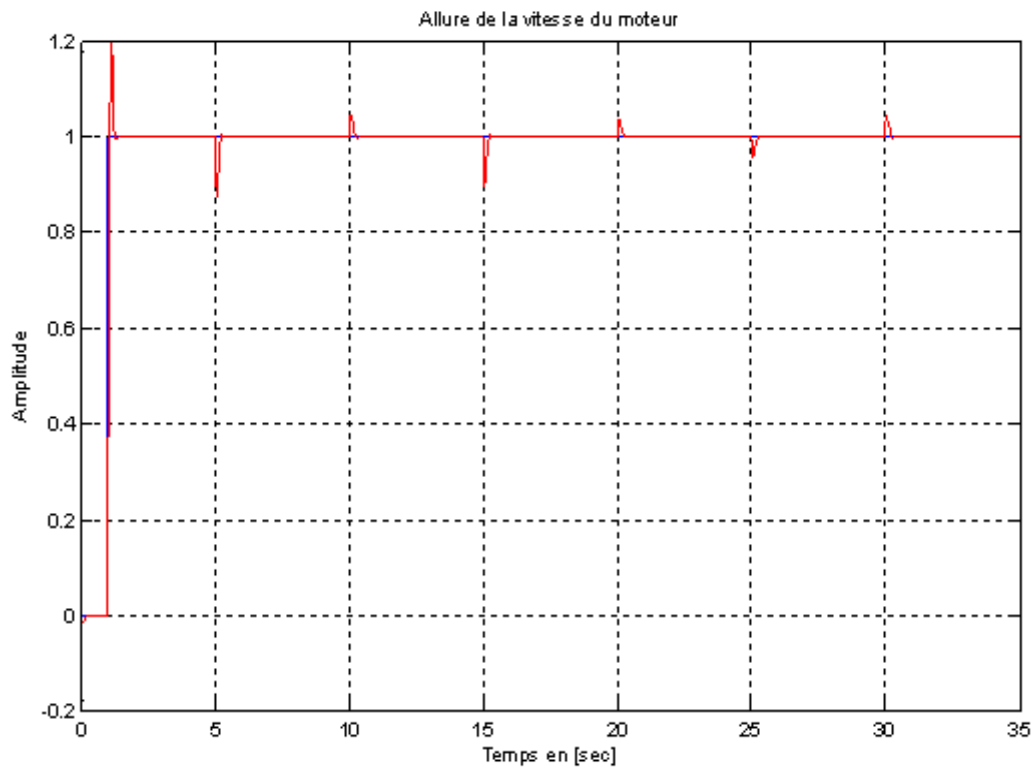


FIGURE 4.16 : Réponse indicielle du moteur lors de la régulation en cascade

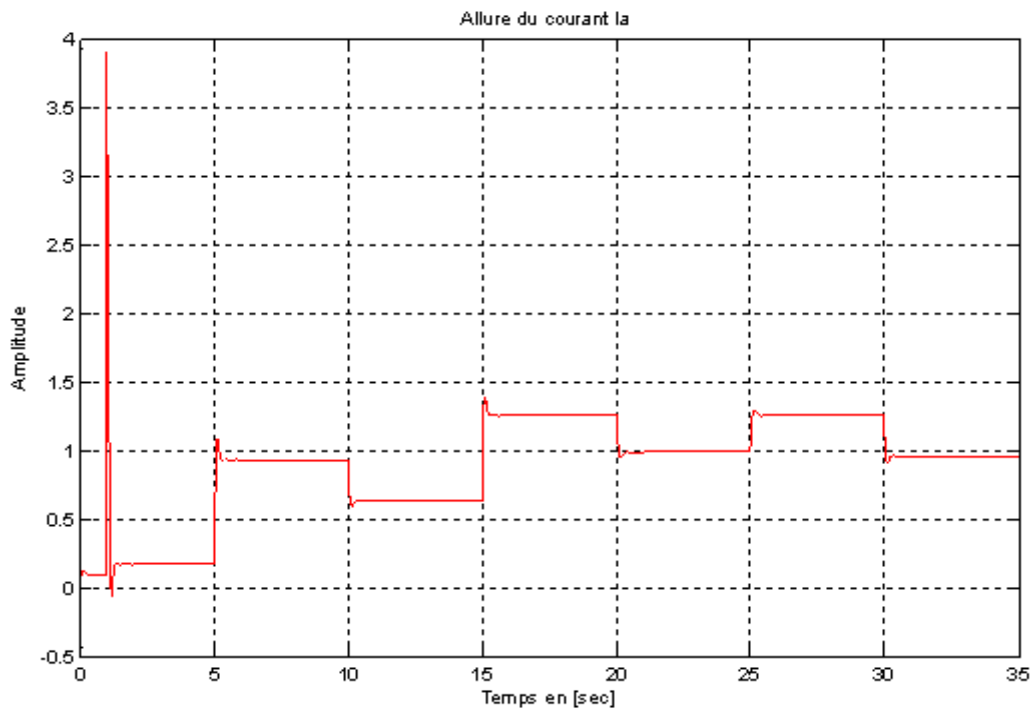


FIGURE 4.17 : Allure du courant de commande

Chapitre 4 : Modélisation de la chaîne de régulation

On constate que les oscillations de la figure (4.13) ont disparus après l'application de la boucle de courant, ils ont été amortis par le correcteur de courant.

Nous avons sur la figure (4.17) l'allure du courant de référence synthétisée à partir de l'erreur de vitesse ; cette valeur de I_{ref} sera comparée au courant d'induit puis traitée par le correcteur de courant pour créer la loi de commande pour l'alimentation de la machine.

D'après l'expression du couple électromagnétique, le courant d'induit est proportionnel au couple moteur et comme le couple résistant est inversement proportionnel à la vitesse ; l'augmentation du couple résistant entraîne la diminution de la vitesse. A puissance électromécanique constante la compensation de la chute de vitesse doit se faire par l'augmentation du couple électromagnétique, autrement dit le courant d'induit.

Chapitre 5

Théorie du STATEFLOW

I. Généralités :

STATEFLOW est un outil de construction graphique qui travaille avec SIMULINK pour modéliser et simuler des systèmes à événements [11], que l'on reconnaît aussi sous la dénomination de systèmes réactifs. C'est une interface graphique qui permet de décrire des automates de contrôle selon une sémantique proche du langage des diagrammes d'états. La sortie d'un automate permet de contrôler l'exécution des blocs SIMULINK par le biais des entrées et des triggers. Les changements de mode et l'exécution des lois de commande peuvent être contrôlés par STATEFLOW ; tandis que la mise en œuvre des lois de commande sont décrites par un graphe SIMULINK.

Les systèmes à événements sont souvent utilisés pour modéliser des logiques ou des automates en vue de contrôler un système physique comme un ventilateur, un moteur ou une pompe. Les systèmes à événements peuvent être modélisés comme une machine à états finis.

Les automates à états finis sont aussi appelés des machines séquentielles à états finis [9]. Elles sont constituées d'un ensemble finis d'états, d'un ensemble finis de signaux d'entrées, de signaux de sorties, d'une fonction de transition définissant la valeur du prochain état actif en fonction de la valeur des entrées, des sorties ; et d'une fonction de sortie de l'état courant.

II. Introduction aux machines à états finis :

STATEFLOW est un outil de simulation des machines à états finis. Une machine à états finis (FSM pour Finite State Machine) est une machine dont le comportement est défini par une succession d'état. Chaque sommet est un état et chaque arc, appelé transition, définit une condition pour passer d'un état à un autre. Les traitements sont soit attachés aux transitions (machines de Mealy), soit attachés aux états (machine de Moore). Les transitions définissent donc une séquence entre traitements. Les traitements quand à eux peuvent effectuer des lectures ou des écritures dans des variables partagées par l'ensemble du graphe.

Il existe deux types d'automates, les automates dits de Moore et ceux dits de Mealy. La différence réside dans la fonction de sortie. Pour un automate de type Mealy les sorties de l'automate sont fonction des entrées et de l'état courant, alors que dans un automate de type Moore les sorties sont fonctions uniquement des états [16]. Pour plus de simplicité dans un automate de Moore les actions sont associées à des états tandis que dans un automate de Mealy, elles sont associées à des transitions.

Par exemple un ventilateur peut être considéré comme une machine à états finis parce qu'il peut avoir plusieurs états comme: rapide, moyen, lent et éteint.

Le passage d'un état à un autre peut se faire par l'appui d'un des boutons correspondant au mode de fonctionnement, ce sont les événements.

III. Etudes des différents menus de l'éditeur STATEFLOW :

Nous présentons dans cette partie les différents objets STATEFLOW disponibles. Ils sont groupés suivants trois catégories :

- Les objets graphiques
- Les objets non graphiques
- Le dictionnaire d'objet.

a. Les objets graphiques :










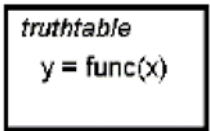

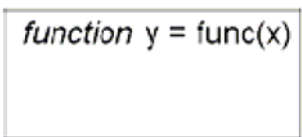

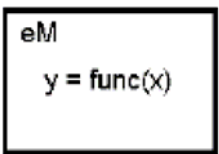



Nom	Notation	Icône
Etat		
Transition		NA
Jonction historique		
Transition de défaut		
Jonction de connexion		
Fonction table de vérité		
Fonctions graphique		
Fonctions MATLAB incorporé		
Boite		

TABLEAU 5.1 : Liste des objets graphiques de l'éditeur STATEFLOW

b. Les objets non graphiques :

STATEFLOW détermine les événements, les données, et les cibles qui n'ont pas de représentation graphique dans l'éditeur de diagramme. Cependant on peut les voir dans l'explorateur STATEFLOW.

1. Les objets événements :

Un événement est un objet qui peut activer un diagramme STATEFLOW entier ou une action individuelle dans un graphique. Parce qu'un graphique STATEFLOW s'exécute en réagissant à un événement on doit programmer et spécifier les événements dans le diagramme pour contrôler leurs exécutions.

2. Les objets 'données' :

Un diagramme STATEFLOW enregistre et récupère les données qui sont utilisés pour contrôler son exécution. Les données dans STATEFLOW résident dans leurs propres espaces de travail mais on peut accéder à des données externes SIMULINK, ou à des applications internes à la machine. Les données peuvent être de trois types : celles qui sont propres à STATEFLOW et que l'on nomme des données locales, celles qui sont des entrées venant de SIMULINK et celles qui sont des sorties vers SIMULINK. La définition des données se fait dans le menu 'add' de l'éditeur STATEFLOW.

3. Les objets 'cibles' :

On construit les cibles dans STATEFLOW pour exécuter les applications programmées dans un graphe STATEFLOW et dans le modèle SIMULINK qui les contient. On construit une cible de simulation (SFUN) pour exécuter la simulation du modèle, on construit une cible RTW (Real Time Workshop) pour exécuter le modèle SIMULINK dans un environnement de développement propre.

c. Le dictionnaire d'objet STATEFLOW :

C'est une base de données qui comprend toutes les informations à propos des objets graphiques et non graphiques. Les entrées d'objets graphiques dans le dictionnaire sont créés automatiquement dès que l'objet a été ajouté et nommé dans l'éditeur.

IV. Hiérarchie des objets dans STATEFLOW :

Un diagramme STATEFLOW organise ses objets dans une hiérarchie, la hiérarchie des objets pour tout diagramme STATEFLOW se réfère au dictionnaire de données STATEFLOW. Le schéma général du dictionnaire de données STATEFLOW est détaillé dans la figure (5.1).

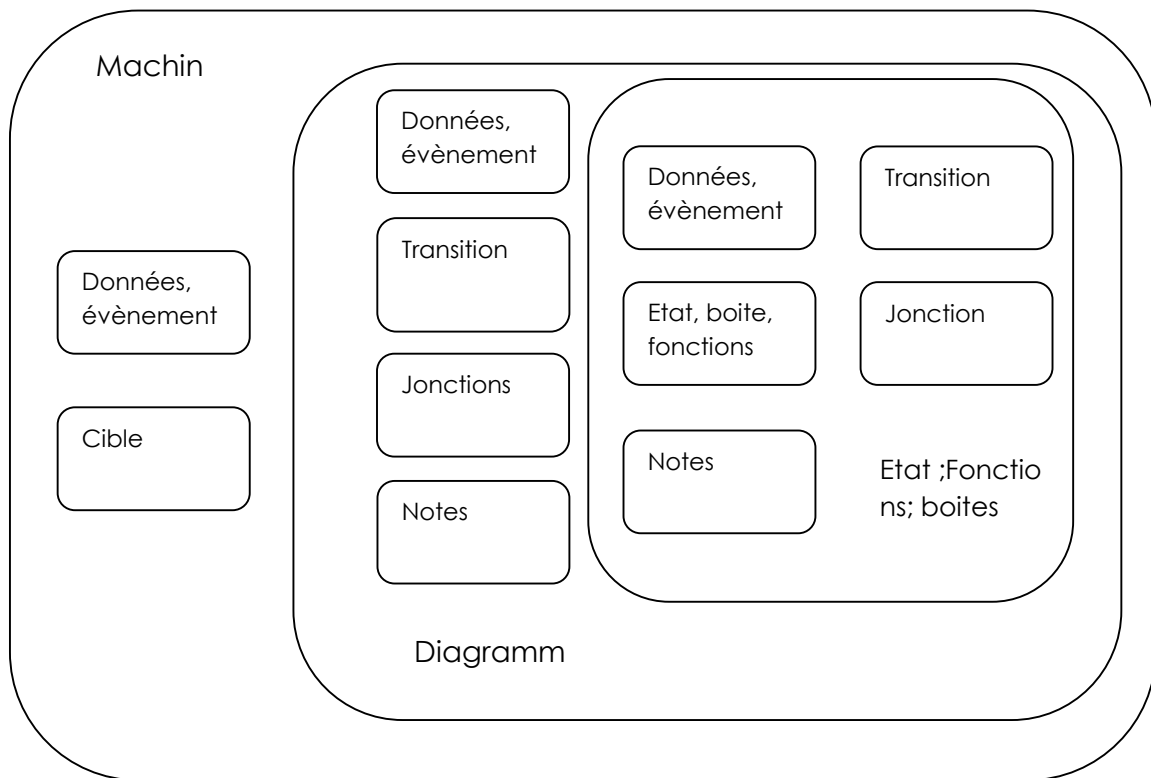


FIGURE 5.1 : Hiérarchie des objets dans STATEFLOW

V. Les opérateurs dans STATEFLOW :

STATEFLOW permet l'utilisation dans les langages d'action de quelques opérateurs arithmétiques et logiques, entre les données internes ou externes échangées. Les opérateurs rencontrés dans STATEFLOW sont généralement de quatre types :

- Les opérateurs arithmétiques, binaires et logiques
- Les opérateurs d'assignation
- Les pointeurs et les adresses
- Les types de variables

a. Les opérateurs arithmétiques et binaires :

STATEFLOW permet l'utilisation des opérateurs arithmétiques de base (+,-,/,*), des opérateurs de comparaison (<,>,>=,<=,==,<>), du produit ET logique (&), du OU logique (|), du OU exclusif (^) et du NON logique (!,~).

b. Les opérateurs unaires :

STATEFLOW supporte les expressions d'incrément (a++) et de décrémentation (a--) du langage C++.

c. Les opérateurs d'assignation :

Comme STATEFLOW permet la programmation en langage C, il offre les possibilités de syntaxes suivantes :

A = expression	Assignation simple
A := expression	Utilisé principalement pour les nombres à points fixes
A += expression	Equivalent à A = A + expression
A -= expression	Equivalent à A = A - expression
A *= expression	Equivalent à A = A * expression
A /= expression	Equivalent à A = A / expression
A = expression	Equivalent à A = A expression
A &= expression	Equivalent à A = A & expression
A ^= expression	Equivalent à A = A ^ expression

TABLEAU 5.2 : Liste des opérateurs disponibles dans STATEFLOW

d. Les pointeurs et les adresses :

On utilise généralement les déclarations en langage C pour définir les pointeurs et les adresses.

e. Les types de variables :

STATEFLOW dispose d'un opérateur qui permet la conversion de type de variable. Normalement on n'a pas besoin de convertir les données sauf si on utilise un code généré extérieurement dans un langage supporté par MATLAB. Les types de données supportées par STATEFLOW sont : type 'double', 'single', 'int32', 'int16', 'int8', 'uint32', 'uint16', 'uint8', et le type 'booléen'. La conversion d'une expression à un type et l'affectation se fait de la manière suivante :

$$variable = cast ('expression', 'data type') \quad (5-1)$$

VI. Introduction aux différents objets :

a. Les états :

1. Définition :

Un état définit un mode d'un diagramme STATEFLOW réactif. Les états dans un diagramme représentent ces modes. Un état peut être actif ou inactif. Quand un état est actif, le diagramme prend ce mode.

Quand un état est inactif le diagramme n'est pas dans ce mode, l'activité ou l'inactivité d'un état change dynamiquement en fonction des événements et des conditions sous lesquels l'état est soumis [16].

Chaque état a un parent et un état qui est lui-même son parent est appelé un diagramme racine. A chaque point de l'exécution d'un diagramme, c'est la combinaison et la succession de l'activité ou de l'inactivité d'un état.

2. Hiérarchie des états :

Un état peut contenir tous les autres objets STATEFLOW, sauf les cibles. Un état qui contient d'autres états est un 'super-état' ; et un état est appelé un 'sous-état' si il est contenu dans un autre état appelé son parent, un 'sous-état' peut être actif seulement si son parent est actif. Un état qui n'est ni un 'super-état' ni un 'sous-état' est un état qui a pour parent le diagramme STATEFLOW lui-même.

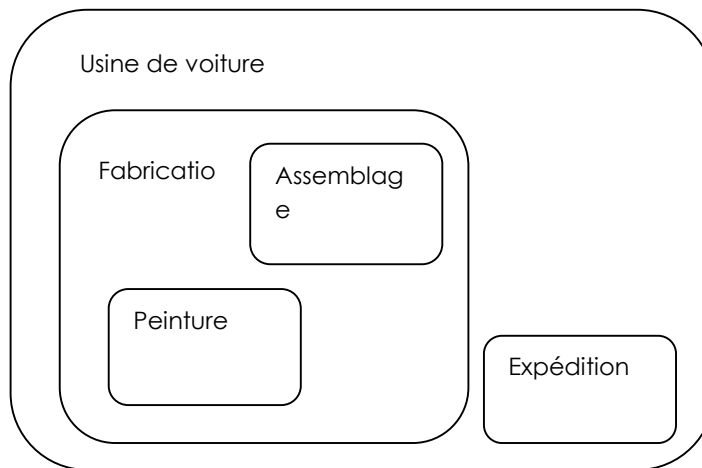


FIGURE 5.2 : *Hiérarchie des états*

Dans cet exemple le diagramme STATEFLOW est le parent de l'état 'Usine de voiture', qui est le parent des états 'Expédition' et 'Fabrication'. Le 'super-état' 'Fabrication' est le parent des sous états 'Assemblage' et 'Peinture'.

On représente cette arborescence comme suit :

- /Usine de voiture
- /Usine de voiture. Fabrication et /Usine de voiture. Expédition
- /Usine de voiture. Fabrication.Assemblage et /Usine de voiture.Fabrication.Peinture

3. Décomposition des états :

Chaque état a une décomposition qui dicte quels types de sous-états il peut contenir. Tous les sous-états d'un super état doivent être de même type. La décomposition pour un état peut

être exclusif (OR) ou parallèle (AND). Si l'on veut utiliser dans un même diagramme les deux types de décomposition, il faudra encapsuler dans un même super-état les sous états de mêmes types, et il faudra que les supers-états soient eux aussi de mêmes types.

a. La décomposition d'état exclusive (OR) :

La décomposition exclusive d'un état pour un super-état est indiqué quand ces sous-états ont des bords en traits plein. La décomposition exclusive est utilisée pour décrire les modes d'un système qui est mutuellement exclusif.

Quand un super-état a une décomposition exclusive seul un de ses sous-états peut être actif à un moment donné.

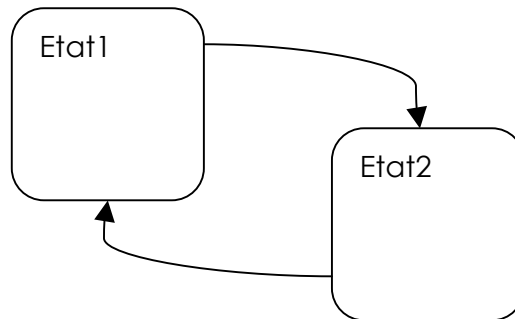


FIGURE 5.3 : Décomposition exclusive des états

Sur l'exemple de la figure (5.3) ci-dessus l'activité des états se succède.

b. La décomposition d'état parallèle :

La décomposition parallèle pour un super-état est représentée par des sous-états avec des bordures en traits interrompu.

Cette représentation est appropriée si tous les états de même niveau dans la hiérarchie sont toujours actifs en même temps.

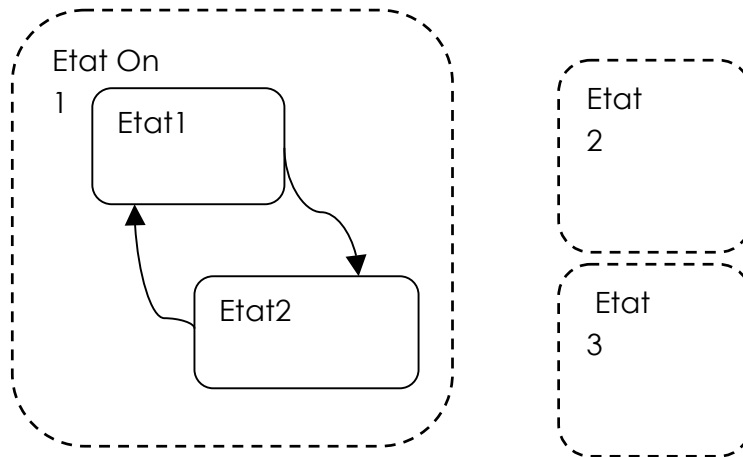


FIGURE 5.4 : Décomposition parallèle des états

Dans cet exemple les états 'Etat On', Etat2 et Etat3 sont actifs simultanément.

4. L'étiquette des états :

L'étiquette pour un état apparaît sur le coin supérieur gauche de l'état avec les indications suivantes :

Name/
Entry :entry actions
During :during actions
Exit:exit_actions
Bind:events,data
On event name: on event name actions (5-2)

a. Le nom d'état:

L'étiquette d'un état commence avec le nom de l'état suivi d'un caractère '/' optionnel, l'utilisation des hiérarchies permet une certaine flexibilité dans la nomination des états. Le nom qu'on entre doit être unique et sera par la suite enregistrée dans le dictionnaire de donnée.

b. Langage d'actions sur les états :

Après le nom on retrouve les actions optionnelles pour l'état avec un mot clé qui identifie le type d'action. Pour chaque type d'action on peut entrer plus d'une action en veillant à ce qu'elles soient séparées par un point virgule (;). On peut spécifier les actions pour différents événements en ajoutant une ligne '**on_event name**'.

- **Entry action** : Précédé par le préfixe entry (ou en) ; c'est cette action qui sera exécutée une fois que l'état sera actif.
- **During action** : Précédé du préfixe during (ou du) cette action est exécutée une fois que l'action est active et le reste pendant un certain temps.

- **Exit action** : Précédé du préfixe exit (ou ex) , l'état exécute cette action quand elle devient inactive.
- **On event_name action** : Précédé du préfixe on event_name où event_name est un évènement unique, l'action est exécutée si l'état hôte est actif et que l'évènement event_name apparaît.
- **Bind action** : Précédé du préfixe bind. Ceci veut dire que seul l'état hôte ou un de ses enfants peut changer la valeur de la variable ou de l'évènement, les autres états ne peuvent qu'utiliser cette variable.

b. Les transitions :

1. Définition :

Une transition est une ligne incurvée avec une flèche qui lie un objet graphique avec un autre. Dans la plupart des cas une transition représente le passage d'un mode à un autre d'un système; le bout d'une transition est attachée à l'objet source et l'autre à l'objet de destination suivant le sens de la flèche. La source est l'état où la transition commence et la destination est l'état où elle finit.

2. Hiérarchie des transitions :

Une transition ne peut pas contenir d'autres objets comme les états le peuvent, cependant les transitions sont contenues par les états, la hiérarchie des transitions est décrite en terme de transition parent, source et destination. Le parent est le plus bas niveau qui contient la source et la destination de la transition. Nous allons le définir par un exemple.

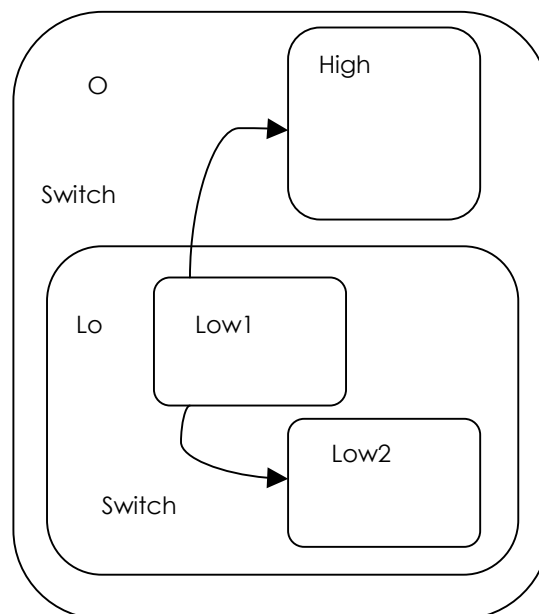


FIGURE 5.5 : Hiérarchie des transitions

Dans cette figure nous allons présenter la hiérarchie dans le tableau suivant :

Transition	Parent	Source	Destination
Switch High	/On	/On.Low.Low1	/On. High
Switch Low2	/On.Low	/On.Low.Low1	/On.Low.Low2

TABLEAU 5.3 : Hiérarchie des transitions

3. Langage d'action sur les transitions :

L'étiquette sur une transition peut être constituée d'un évènement, d'une condition, d'une action de condition et/ou d'une action de transition.

La forme générale d'une étiquette de transition est la suivante :

$$\text{[condition]\{action_de_la_condition\}/action_de_la_transition} \quad (5-3)$$

Où la syntaxe :

- **Evènement** : spécifie un évènement qui causera la prise de la transition si la condition *[condition]* est vraie. L'absence de l'évènement spécifie que la transition sera prise au dessus de l'apparition d'autres évènements.
- **Condition** : elle spécifie une expression booléenne qui, quand elle est vraie, valide la prise de la transition pour l'évènement correspondant.
- **L'action de la condition** : Elle suit la condition et elle est mise dans des accolades. L'action est exécutée immédiatement après que la condition aie été évaluée vraie et avant que la destination de la transition soit valide
- **Action de transition** : L'action de transition est exécutée une fois que la destination de la transition est valide et la condition spécifiée est vraie.

4. Validation des transitions :

Dans la plupart des cas une transition est valide quand l'état source de la transition est actif et l'étiquette de la transition est valide. Nous énumérons les cas possibles dans le tableau suivant :

Etiquette	Validité si ...
Evènement seul	L'évènement a lieu
Evènement et condition	L'évènement a lieu et la condition est vraie
Condition seule	Aucun évènement n'a lieu et la condition est raie
Action seul	Aucun évènement n'a lieu
Non spécifié	Aucun évènement n'a lieu

TABLEAU 5.4 : Règles de validation des transitions

5. La transition interne :

Une transition interne est une transition qui ne quitte pas l'état source. Pour les super-états avec une décomposition exclusive la transition interne est la plus appropriée. Elle sert grandement à la simplification des diagrammes STATEFLOW.

Nous utiliserons un exemple pour illustrer la nécessité d'une transition interne.

- Avant l'utilisation de la transition interne :

Le diagramme de la figure (5.6) comprend trois états A1/, A2 et A3/, chaque état est relié avec les autres par l'intermédiaire de deux transitions aller et retour. L'évènement pouvant causer la prise de chaque transition entre état est un évènement unique E_one, la condition quant à elle dépend de chaque transition.

On constate suivant la figure (5.6) qu'il est difficilement aisé de suivre l'évolution du diagramme, en effet il existe une redondance de l'évènement E_one, ainsi que des conditions [C_one] et [C_two].

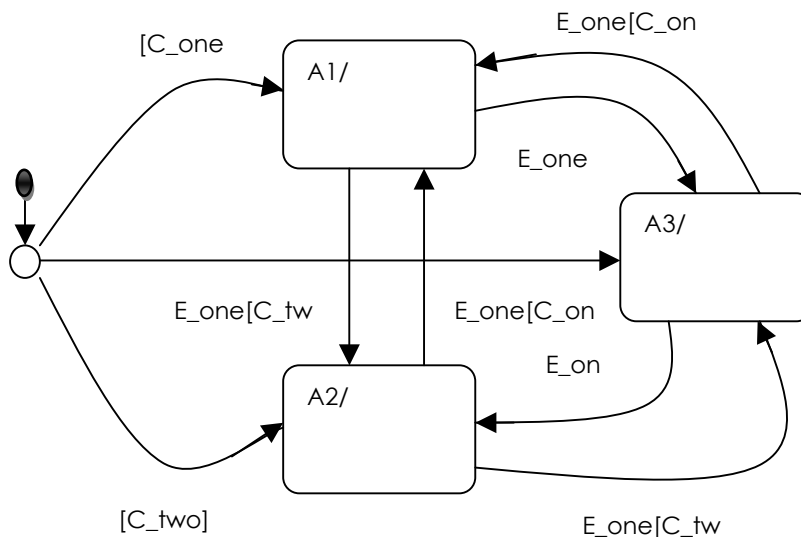


FIGURE 5.6 : Modèle du diagramme avant l'application de la transition interne

Un évènement externe apparaît et active le diagramme, la transition de défaut sur la jonction de connexion est valide. La destination de la transition est déterminée par les conditions [C_one] et [C_two]. Si [C_one] est vraie la transition vers A1 est valide, sinon si [C_two] est vraie la transition vers A2 est vraie. Si aucunes des deux conditions n'est vraie la transition vers A3 est prise, la prise des transitions entre les états A1, A2 et A3 est conditionnée par l'apparition de l'évènement E_one, sous les conditions [C_one] ou [C_two].

- Après l'utilisation de la transition interne : l'exemple de la figure (5.7) simplifie l'exemple précédent avec l'utilisation de la transition interne.

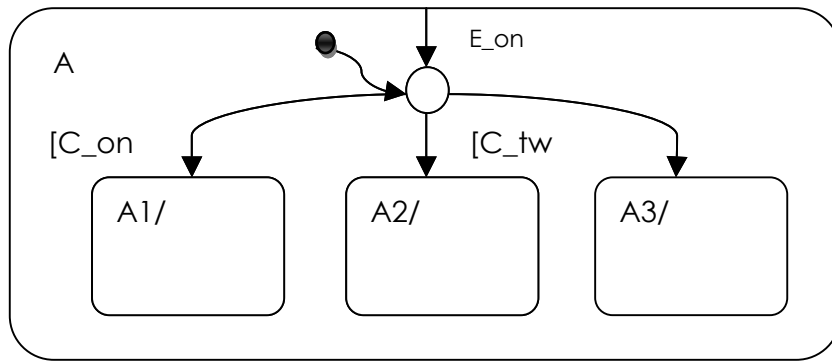


FIGURE 5.7 : Diagramme simplifié par la transition interne

Le diagramme est simplifié en utilisant une transition interne à la place des multiples transitions entre les états A1, A2 et A3 dans l'exemple original. Supposons que l'état A est déjà actif, la transition interne est utilisée pour réévaluer lequel des sous-états de A sera à son tour actif. Quand l'évènement E_one a lieu la transition interne est potentiellement valide. Le choix des sous-états se fera alors en fonction de la validité des conditions sur chaque transition de destination A1, A2 et A3

c. La transition de défaut :

La transition de défaut est principalement utilisée pour spécifier lequel des états exclusifs (ou des jonctions) doit être actif en premier, quand, il existe une ambiguïté entre deux ou plusieurs états (ou jonctions) ayant la même hiérarchie dans le diagramme. La transition de défaut a une destination mais n'a pas de source. C'est une transition avec un point noir sur l'autre bout. Elle possède aussi la possibilité d'être éditée comme toute transition.

d. Les jonctions :

1. Définition :

Les jonctions de connexion sont des points de décision dans le système. Elles servent en général de prise de décision lorsque la transition d'un état à un autre est régit par plusieurs conditions, elle donne par cela une alternative pour représenter le comportement du système.

2. Applications :

Les diagrammes de flux utilisent les jonctions pour représenter des structures de code usuels comme les boucles 'for', les conditions 'if-then-else' sans l'utilisation des états. Et en réduisant le nombre d'état dans le diagramme, la notation des diagrammes de flux permet l'optimisation de la mémoire. Une jonction permet la représentation des cas suivants :

- Construction d'un point de décision 'if-then-else' en spécifiant les conditions sur quelques un ou sur toutes les transitions sortant de la jonction. L'apparition de l'évènement E_one conditionne la prise de la transition allant de l'état A vers la

jonction ; arrivée à cette étape le choix du prochain état qui sera actif sera lié à la validité des conditions [C_one], [C_two] et [C_three].

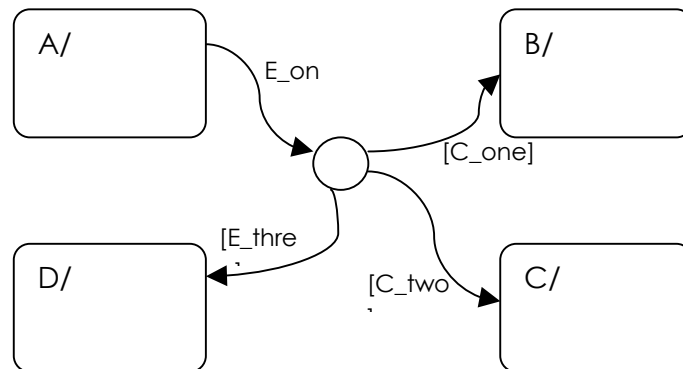


FIGURE 5.8 : Diagramme pour l'expression conditionnelle 'if-then-else'

- Une boucle de transition de retour vers l'état source si aucune des transitions de sortie de la jonction n'est valide. Dans le cas de la figure (5.9) si la condition [C_one] pas vraie la transition allant de la jonction vers l'état A sera prise et cela à chaque fois que la condition sera jugée fausse.

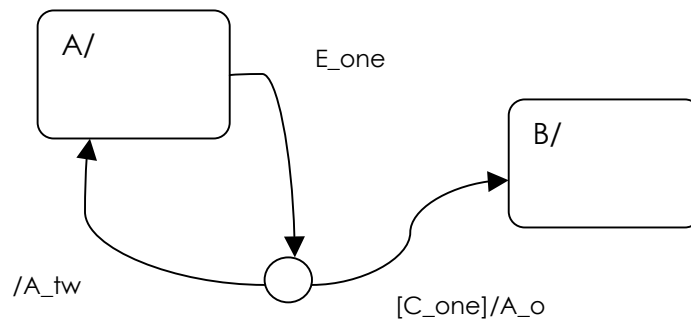


FIGURE 5.9 : Diagramme pour l'expression conditionnelle 'if-then'

- La construction d'une boucle *for* ayant comme source et destination la jonction elle-même.

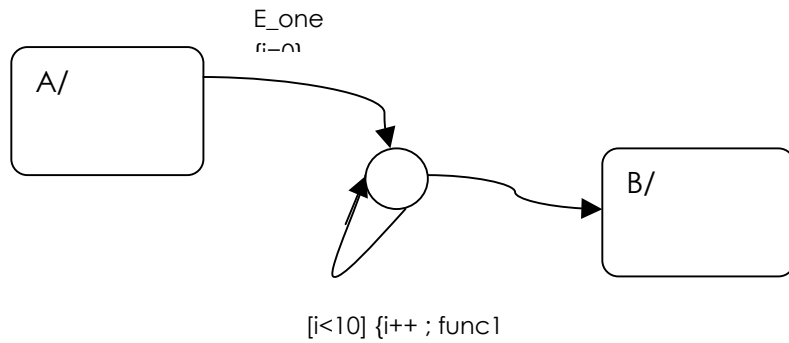


FIGURE 5.10 : Diagramme pour la boucle 'for'

La transition partant de la jonction vers l'état B ne sera prise que si la condition de la jonction sur elle-même est jugée fausse, c'est à dire que si $i \geq 10$.

- Transition d'une source commune vers de multiples destinations

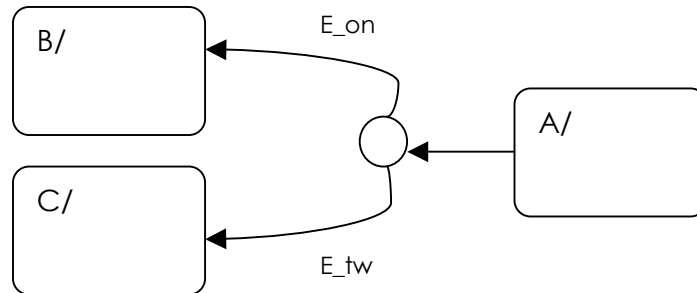


FIGURE 5.11 : Diagramme pour la transition vers une source multiple

La jonction dans la figure (5.11) sera pris comme point d'attente jusqu'à ce que l'un des évènements E_one ou E_two apparaisse.

- Transition partant de sources multiples vers une destination commune

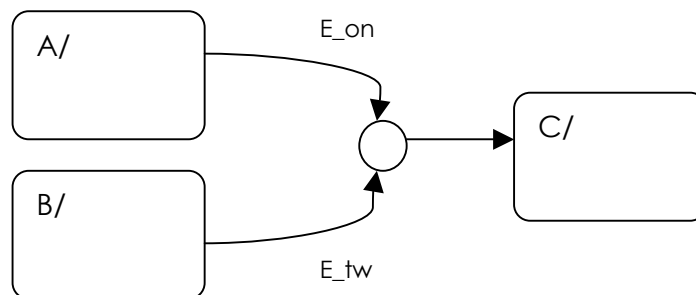


FIGURE 5.12 : Diagramme pour la transition vers une destination commune

La jonction ici sert de point de ralliement entre la transition de source A et celle de source B. L'apparition de l'évènement E_one (respectivement E_two) sera la condition de validation de la transition de A vers la jonction (respectivement de B vers la jonction).

e. Jonction historique :

Elle joue le rôle de mémoire, permettant ainsi à l'utilisateur de revenir au dernier état pris dans le diagramme. La jonction historique est utilisée pour représenter un point de décision historique dans le diagramme. Le point de décision est basé sur des données historiques relatives à l'activité d'un état. Placer une jonction historique dans un super-état indique que l'information sur l'activité de l'état est utilisée pour déterminer le prochain état à devenir actif.

f. Les boîtes :

On utilise les boîtes pour réorganiser graphiquement le diagramme, il contribue de la même façon à l'exécution du dit diagramme.

g. Les évènements :

Ce sont des objets non graphiques qui peuvent activer une grande variété d'activité incluant le réveil du diagramme. Chaque évènement affecté à un diagramme doit être défini. La diffusion d'un évènement peut causer l'apparition d'une transition ou peut causer l'exécution d'une action. Un évènement peut avoir les propriétés suivantes :

- Local au diagramme
- Entrée de STATEFLOW venant de SIMULINK
- Sortie vers SIMULINK venant de STATEFLOW
- Exportée vers une destination externe à SIMULINK
- Importée depuis un code source externe.

Le choix de cette spécificité des évènements se fait dans le menu 'add' de l'éditeur de STATEFLOW, on y accède également aux données.

h. Les données :

Ce sont des objets non graphiques utilisés pour enregistrer des valeurs numériques de référence dans le diagramme. Les données peuvent être définies de la manière suivante :

- Local au diagramme, constante, variable.
- Une entrée de SIMULINK
- Une sortie vers SIMULINK
- Une variable WORKSPACE
- Importé d'un code externe ou exporté vers un code externe à SIMULINK

i. Les actions :

On ne rencontre que deux types d'action, celles sur les états et celles sur les transitions. Une action peut être un appel de fonction, une diffusion d'un évènement ou l'assignation d'une valeur à une variable. STATEFLOW supporte en même temps les machines à états finis de Mealy et ceux de Moore.

j. Les fonctions graphiques :

Une fonction graphique est une fonction définie graphiquement par un diagramme de flux, et qui permet l'utilisation de la puissance et de la facilité des langages d'actions de STATEFLOW. Les fonctions graphiques sont similaires à de simples fonctions elles peuvent donc être utilisées dans les langages à actions (états, transitions) et acceptent des arguments qui retournent des résultats.

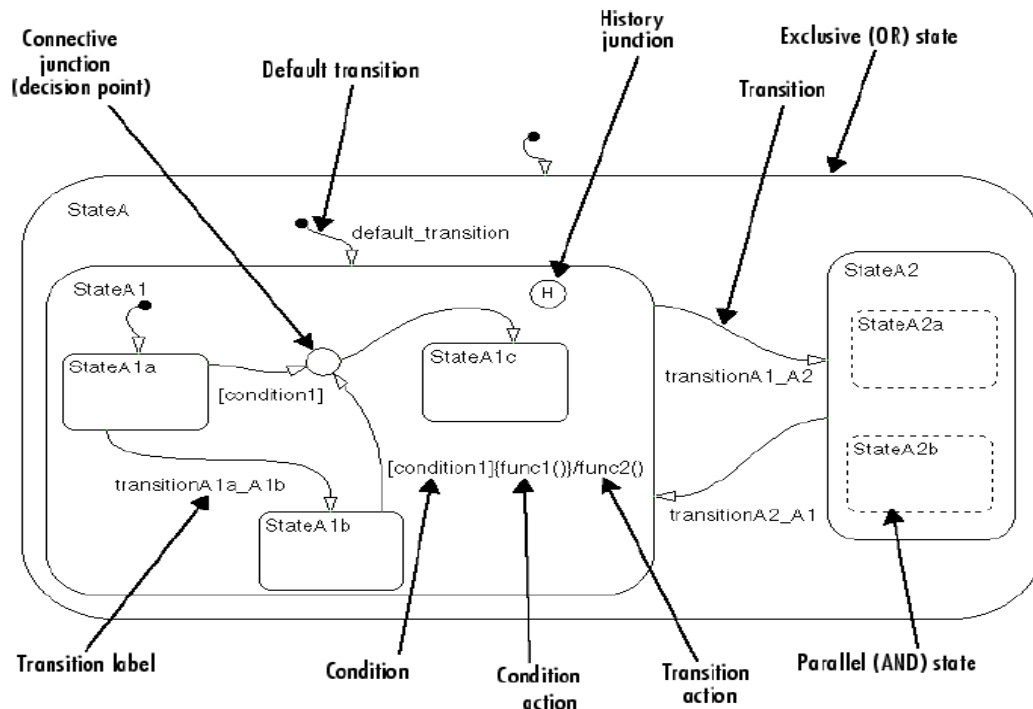


FIGURE 5.13 : Règles de modélisation sous STATEFLOW

VII. Utilisation des données et des fonctions MATLAB dans STATEFLOW :

L'appel des fonctions MATLAB et l'accès aux variables de l'environnement MATLAB est possible dans les langages d'actions en utilisant l'opérateur **'ml'** ou l'opérateur **'mlfunction'**

a. Utilisation de l'opérateur 'ml' :

L'opérateur *ml* utilise la notation standard (.) pour référencer les variables et les fonctions MATLAB dans les actions.

Par exemple la déclaration **y = ml . x** retourne la valeur de la variable WORKSPACE 'x' de MATLAB. Et la déclaration **y = ml . matlfunc (arg1, arg2)** retourne la valeur de la fonction MATLAB **matlfunc** dans la variable STATEFLOW 'y'.

Si la fonction MATLAB appelée ne requiert pas de parenthèse on doit toujours l'inclure. Si la parenthèse a été omise, STATEFLOW interprète le nom de fonction comme une variable WORKSPACE, lequel ne sera pas trouvé et générera une erreur lors de la simulation.

Par exemple dans l'expression suivante les variables 'd1' et 'd2' sont des variables STATEFLOW et les variables 'x' et 'y' des variables WORKSPACE. Et le produit de l'opération est restauré dans la variable 'x' de MATLAB.

$$\mathbf{ml.x = d1*d2/ml.y} \quad (5-4)$$

b. Utilisation des fonctions ml :

On peut utiliser les fonctions ml pour spécifier l'appel des fonctions MATLAB à travers une expression dans les langages à actions. Le format d'appel des fonctions MATLAB utilise la notation standard des fonctions :

$$\mathbf{ml (evalString, arg1, arg2,...)} \quad (5-5)$$

Par exemple dans l'expression suivante

$$\mathbf{a = ml ('sin (%f)', d1)}$$

la fonction MATLAB 'sin' évalue le sinus de d1 dans le WORKSPACE de MATLAB et assigne le résultat à la variable STATEFLOW a. Parce que d1 est une variable STATEFLOW sa valeur est insérée dans l'argument 'evalString' en utilisant le format d'expression %f.

c. Utilisation simultanée de l'expression ml et des fonctions ml :

On peut utiliser l'expression ml et les fonctions ml simultanément dans une expression d'action de STATEFLOW. L'expression qui suit calcule le carré du sinus et du cosinus d'un angle X dans le WORKSPACE et calcule leurs sommes.

$$\mathbf{ml.power (ml.sin (ml.X), 2) +ml ('power (cos(X), 2)')}$$

Le premier opérande utilise l'opérateur ml pour appeler la fonction sin. Son argument est ml.X parce que c'est une variable WORKSPACE de MATLAB. Le second opérande utilise une fonction ml. Parce que X est une variable WORKSPACE il est inclus dans l'expression

evalString . Le carré de chaque opérande est évalué avec la fonction power de MATLAB qui a deux arguments : la valeur de l'expression à évaluer et la valeur de la puissance.

VIII. Les opérateurs de logique temporelle (OLT) :

Ces opérateurs sont des opérateurs booléens qui agissent simultanément avec l'apparition des événements.

a. Règles d'utilisation des OPÉRATEURS DE LOGIQUE TEMPORELLE :

La périodicité des événements sur lesquels les OPÉRATEURS DE LOGIQUE TEMPORELLE fonctionnent sont appelés des événements de base. Chaque événement STATEFLOW peut servir d'événement de base pour un OPÉRATEUR DE LOGIQUE TEMPORELLE.

Pour un diagramme qui n'a pas d'événement d'entrée de SIMULINK, on peut utiliser l'événement **wakeup** ou **tick** pour définir un événement implicite au réveil du diagramme.

Les OPÉRATEURS DE LOGIQUE TEMPORELLE peuvent seulement apparaître sur les conditions des transitions et sur les actions des états.

b. L'OPÉRATEUR DE LOGIQUE TEMPORELLE after :

L'opérateur 'after' est valide à chaque fois qu'un événement a lieu après le temps spécifié. Il porte la syntaxe suivante :

$$\text{After (n, E)} \quad (5-6)$$

Avec E l'événement de base et n est soit une constante entière positive soit une expression qui est évaluée à une constante entière positive.

L'opérateur 'after' est vraie après n apparition de l'événement E depuis l'activation de son état associé, sinon il est évalué faux. Pour un diagramme qui n'a pas d'événement d'entrée **after (n, wakeup)** est évaluée vraie après que le diagramme ait été réveillé n fois.

c. L'OPÉRATEUR DE LOGIQUE TEMPORELLE before :

L'opérateur 'before' est évalué à chaque fois qu'un événement se produit avant le temps spécifié. Il s'écrit comme suit :

$$\text{before (n, E)} \quad (5-7)$$

L'opérateur '*before*' est vrai avant l'apparition du $n^{\text{ième}}$ évènement E, depuis l'activation de son état associé. Sinon il est faux. Un diagramme qui n'a pas d'évènement d'entrée *before (n, wakeup)* est vraie avant que le diagramme ait été éveillé n fois.

d. L'OPÉRATEUR DE LOGIQUE TEMPORELLE at :

L'opérateur 'at' est valide pour l'apparition d'un évènement spécifique. Il se formule comme suit :

$$\text{At}(n, E) \quad (5-8)$$

L'opérateur 'at' est vraie uniquement à l'apparition du $n^{\text{ième}}$ évènement de base E depuis l'activation de son état associé. Pour un diagramme qui n'a pas d'évènement d'entrée *At (n, wakeup)* est vrai quand le diagramme a été réveillé au $n^{\text{ième}}$ temps.

e. L'OPÉRATEUR DE LOGIQUE TEMPORELLE every :

L'opérateur 'every' est validé pour chaque apparition d'un évènement.

$$\text{Every}(n, E) \quad (5-9)$$

L'opérateur est vrai à chaque apparition du $n^{\text{ième}}$ évènement de base depuis l'activation de son état associé. Pour un diagramme qui n'a pas d'évènement d'entrée, *every (n, wakeup)* active le diagramme chaque fois que le $n^{\text{ième}}$ évènement apparaît.

f. L'OPÉRATEUR DE LOGIQUE TEMPORELLE in :

L'opérateur in est valide avec l'activité d'un état. Il porte la syntaxe suivante :

$$\text{In}(S) \quad (5-10)$$

Avec S un état déterminé. L'opérateur *in* est donc évalué vrai (retourne la valeur 1) quand l'état S est actif, dans le cas contraire il est jugé faux.

IX. L'appel des fonctions C avec STATEFLOW :

L'appel des fonctions C est possible à partir de STATEFLOW ; pour utiliser une fonction C dans un diagramme STATEFLOW on suit la procédure suivante :

- Dans un nouveau diagramme STATEFLOW du menu Tools, choisir une cible RTW ou Simulation.
- Quand la boîte de dialogue Open Target apparaît choisir une option
- Entrer ensuite ce qui suit :

Introduire le fichier d'entête pour la déclaration des variables dans le champ '**custom code included at the top of generated code**'. Spécifier nom de fichier source qui contient le programme dans le champ '**Custom sources files**'.

Pour utiliser la fonction C programmée il faut définir le nom de la fonction, ses arguments dans des parenthèses et un (;) optionnel. Un paramètre chaîne de caractère peut être passé à l'intérieur d'une apostrophe pour faciliter la compréhension. Une action peut contenir un appel de fonction et invoquer une fonction qui retourne une valeur scalaire.

a. L'appel des fonctions C dans les transitions :

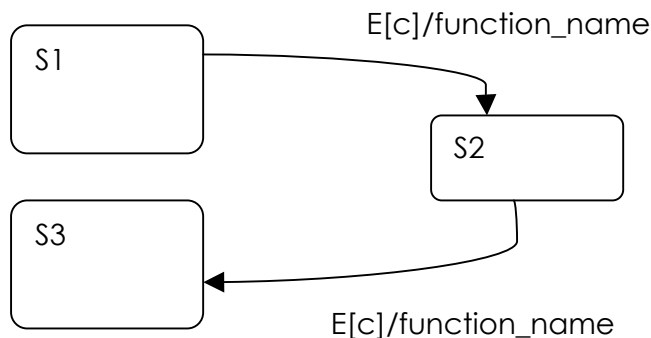


FIGURE 5.14 : Appel des fonctions C sur les transitions

Supposons que 's1' soit actif l'évènement E apparaît, la condition 'c' est vraie et la destination de la transition est déterminée, alors la fonction *function_name* avec ses arguments arg1, arg2 et arg3 est appelée. L'action de transition dans la transition de 's2' vers 's3' montre un appel de fonction implémentée avec un autre appel de fonction.

L'appel des fonctions peut se faire aussi à partir des langages d'actions des états, il peut très bien être exécuté dans les actions d'entrées, de sorties ou de durées des états.

b. L'appel de fonction dans les actions d'états :

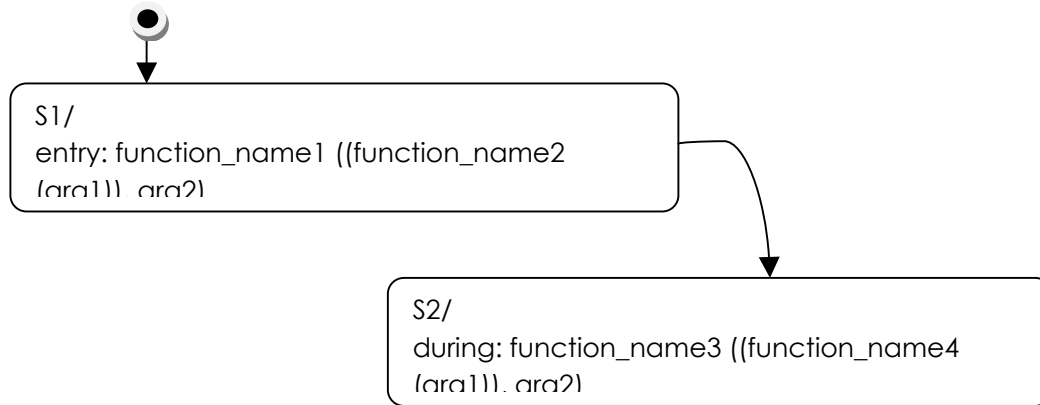


FIGURE 5.15 : Appel des fonctions C sur les états

Quand la transition de défaut sur l'état 's1' apparaît, 's1' est actif et son action d'entrée qui est un appel de fonction avec ses propres arguments est exécuté ; de même si 's2' est actif et un événement apparaît le 'during action' de l'état 's2' est pris et l'appel de fonction 'function_name3' est complété et exécuté.

X. Utilisation des tables de vérité dans STATEFLOW :

a. Introduction :

STATEFLOW utilise la fonction table de vérité pour réaliser la prise logique de décision. La table de vérité STATEFLOW contient des conditions, des décisions et des actions qui sont arrangés comme suit :

Condition	Décision 1	Décision 2	Décision 3	Décision par défaut
X==1	T	F	F	-
Y==1	F	T	F	-
Z==1	F	F	T	-
Action	t=1	t=2	t=3	t=4

TABLEAU 5.5 : Aperçu d'une table de vérité

Chaque condition entrée dans la colonne condition doit être évaluée vraie ou fausse. Cette évaluation se fera par la lettre T pour *true* F pour *false* et un (-) si ce n'est ni l'un ni l'autre. Chaque colonne de condition combine une sortie avec un ET logique de la composition des conditions, ceci est référé à une décision.

Une table de vérité évalue une décision à chaque instant en commençant par la première décision. Si une des décisions est vraie on exécute l'action correspondante et l'exécution de la table de vérité est complète.

Par exemple si les conditions 1 et 2 sont fausses et la condition 3 est vraie, la décision 3 est exécutée et à la variable t est donnée la valeur de 3.

Les décisions restantes ne sont pas testées et l'évaluation de la table de vérité est terminée. Dans la dernière décision de l'exemple précédent, la décision par défaut couvre toutes les décisions restantes possibles. Si les décisions 1, 2 et 3 sont fausses alors la décision de défaut est automatiquement vraie et l'action (t=4) est exécutée. On peut voir ce comportement quand on examine le pseudo-code équivalent suivant pour l'évaluation de la table de vérité précédente.

Description	Pseudo-code
Décision 1 Action 1	If ((x==1) & !(y==1) & !(z==1)) t = 1 ;
Décision 2 Action 2	elseif (!(x==1) & (y==1) & !(z==1)) t = 2 ;
Décision 3 Action 3	elseif (!(x==1) & !(y==1) & (z==1)) t = 3 ;
Décision par défaut Action par défaut	else t = 4 ; endif

TABLEAU 5.6 : Règle de saisie des conditions et des actions

b. Langage d'action des tables de vérité de STATEFLOW :

Les tables de vérité de STATEFLOW utilisent deux type de langage ; le langage STATEFLOW classique qui est le langage par défaut et le langage de programmation des fonctions MATLAB.

Une table de vérité classique permet de spécifier les conditions et les actions en utilisant le langage d'action de STATEFLOW, qui supporte généralement le langage C et permet d'accéder à MATLAB par la syntaxe ml ou mlfunction.

Pour les tables de vérité utilisant les fonctions implémentées de MATLAB on spécifie les conditions et les actions en utilisant le langage d'action propre de MATLAB. Les tables de vérité utilisant les fonctions propres de MATLAB offrent plusieurs avantages comparés aux tables de vérité classiques de STATEFLOW ; elles permettent en effet :

- D'utiliser plus facilement les boucles (for, while), le test (if) et les choix multiples (switch) du fait de la richesse du langage de programmation MATLAB.
- D'appeler les fonctions propres à MATLAB comme les fonctions trigonométriques, les transformées (fft) etc..... et de générer des codes pour ces fonctions en utilisant le Real-Time-Workshop.

- De créer des variables temporaires et permanents durant la simulation.

c. Utilisation des tables de vérités :

Les tables de vérité sont utilisées comme les fonctions graphiques de STATEFLOW, elles sont exécutées soit dans les langages d'action des états soit dans celles des transitions. Comme ce sont des fonctions il est nécessaire de mentionner les arguments de la table.

En chargeant une nouvelle table de vérité dans l'éditeur de STATEFLOW, on aura à introduire les arguments de la manière suivante :

$$\text{Name_var} = \text{func_name} (\text{arg1}, \text{arg2}...) \quad (5-11)$$

Name_var sera le nom de la variable utilisée dans la table et *func_name* le nom de la fonction qui sera accessible dans les langages d'action, les *argi* sont les arguments de la fonction *func_name*.

d. Programmation des tables de vérité :

On va traiter directement un exemple pour avoir une meilleure compréhension des différentes étapes à suivre. On traitera le même exemple que précédemment.

1. L'entrée des conditions :

Dans la section de la table des conditions, introduire les lignes suivantes :

Description	Condition	D1	D2	D3	D4
x is equal to 1	XEQ1 : x==1	T	F	F	-
y is equal to 1	YEQ1 : y==1	F	T	F	-
z is equal to 1	ZEQ1 : z==1	F	F	T	-

TABLEAU 5.7 : Saisie des conditions

La colonne 'description' sert à décrire une description brève de la condition, c'est une action optionnelle mais elle sera transmise comme commentaire dans le code généré par la table de vérité. La notion XEQ1 sera le nom d'une variable temporaire dans le code généré par le tableau pour l'affectation x==1. Cette dernière affectera à la valeur de l'argument x la valeur 1, c'est cette affirmation qui sera testée vraie ou fausse. Les colonnes D1 à D4 représentent les décisions prises en fonction des valeurs de x, y et z. La décision D4 est la décision par défaut elle doit être mentionnée en dernier lieu.

2. L'entrée des actions :

Pour pouvoir introduire les actions il faut cliquer dans le champ 'Action Table'. La programmation des actions pour le langage STATEFLOW classique utilise la syntaxe du tableau (5.8).

Description	Action
Set t to 1	A1 : t=1 ;
Set t to 2	A2 : t=2 ;
Set t to 3	A3 : t=3 ;
Set t to 4	A4 : t=4 ;

TABLEAU 5.8 : Saisie des actions

L'utilisation du langage MATLAB pour la programmation des actions permet d'introduire des boucles, des représentations graphiques des matrices et bien d'autres encore, nous ne commenterons pas plus cette partie.

Il se pose cependant un problème car les actions et les conditions se trouvent dans des tableaux différents il faudra donc les relier entre eux, c'est le thème du paragraphe suivant.

3. L'assignation des actions aux décisions :

Les règles d'assignations des actions aux décisions sont les suivantes :

- On spécifie les actions pour les décisions en entrant le numéro de ligne du tableau des actions ou la variable de l'action considérée dans la dernière ligne du tableau des décisions respectivement dans chaque colonne de décision.
- On doit spécifier au moins une action pour chaque décision, pour plus d'une action on utilisera des (,).
- On peut spécifier la même action pour plusieurs décisions.

XI. Utilisation des outils de STATEFLOW :

a. Le model Explorer de STATEFLOW :

Le model Explorer permet de visualiser le diagramme dans sa hiérarchie directe, c'est également une plate forme de création, d'édition et de suppression d'objet (données, événements, fonctions...), elle permet aussi de créer, de modifier ou d'effacer les objets cibles utilisés pour la génération de code durant la simulation.

b. Le debugger de STATEFLOW :

On utilise la fenêtre d'exécution de STATEFLOW pour contrôler la simulation faire en sorte de suivre le bon déroulement du système et s'assurer qu'il reproduise le comportement escompté.

En effet, le debugger permet de créer des points d'observations en milieu de simulation.

Cela permet à l'utilisateur de suivre étape par étape le comportement dynamique du diagramme afin de mieux comprendre son fonctionnement. Cette action permet de corriger les éventuelles erreurs lors de la programmation de l'automate.

XII. Etude d'un exemple :

Nous allons modéliser le comportement d'un ventilateur, ce ventilateur est défini par le cahier des charges suivant :

Le contrôleur possède les entrées suivantes :

- Un bouton poussoir On_switch/Off_switch
- La température courante d'un système physique

Le contrôleur aura une sortie qui délivre un signal réel constant.

Outre ces considérations le contrôleur sera régi par le fonctionnement suivant :

- Il doit se mettre en marche et/ou s'arrêter si le manipulateur presse sur le bouton On_switch/Off_switch.
- Il doit restituer la valeur 0 à la sortie lorsqu'il est éteint.
- Il doit restituer la valeur 1 à la sortie lorsque la température est supérieure à 100[°C].
- Il doit restituer la valeur 2 à la sortie lorsque la température est supérieure à 120[°C].

Explication du diagramme :

Au début de la simulation le système se trouve à l'état Off/, l'apparition de l'évènement On_switch active la transition de l'état Off/ vers le super-état On/. Le sous-état /on.NC/ est activé en premier lieu, puis le test [!cool()] qui vérifie si la température est supérieure à 100[°C] le diagramme passe de l'état NC/ à l'état Lo/ alors à la sortie 'speed' sera affectée la valeur 1, si le test de la transition allant de l'état Lo/ vers Hi/ est vraie, c'est-à-dire la température est supérieure à 120[°C], la sortie du contrôleur sera la valeur speed=2. Elle restera dans cet état Hi/ tant que la température sera supérieure à 120 [°C], c'est-à-dire tant que la condition [!hot()] sera fausse. L'utilisation des fonctions graphiques hot() et cool() permet d'alléger le diagramme et le rendre ainsi mieux compréhensible.



FIGURE 5.16 : Fonctions utilisées pour la modélisation du ventilateur

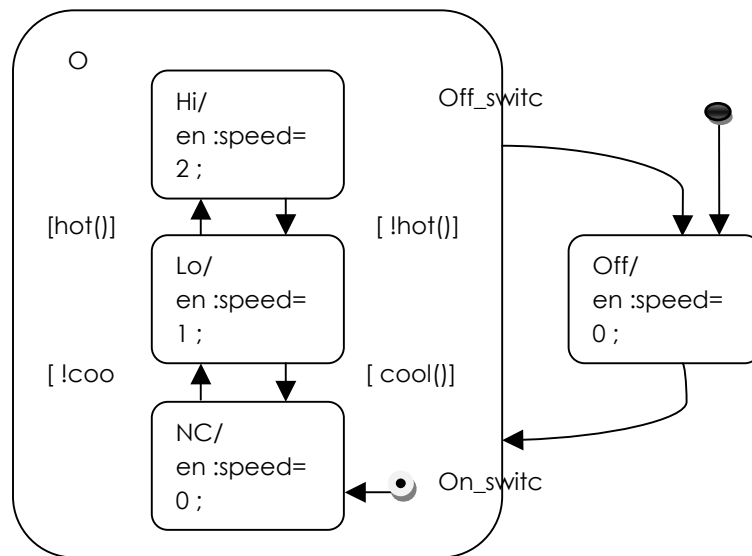


FIGURE 5.17 : Exemple de modélisation d'un ventilateur

Chapitre 6

Application de STATEFLOW à la REGULATION DE VITESSE

XIII. Généralités :

Dans cette partie de notre travail, nous essayerons de modéliser le correcteur PI de vitesse et de courant pour en tirer leurs propriétés. Les résultats précédemment obtenus dans le chapitre 4, pour la régulation du moteur à courant continu serviront de référence au modèle STATEFLOW. Notons qu'un bloc STATEFLOW n'est actif que s'il reçoit un évènement; il a aussi été vu plus haut que l'erreur est indispensable au fonctionnement de la chaîne et que sans erreur il n'y aurait plus de régulation. De ce fait, comme nous essayons de modéliser un régulateur, il est impératif qu'il fonctionne à plein temps, or comme le bloc STATEFLOW est un outil de programmation il ne peut fonctionner continuellement mais devra être réveillé par un évènement ayant une période infiniment petite (équivalence du temps). Dans notre cas, nous utiliserons une suite d'impulsion de fréquence réglable pour simuler l'évènement qui activera le diagramme STATEFLOW.

Ce type de générateur d'impulsion est disponible sous MATLAB/SIMULINK dans le « Simulink Library Browser » catégories « sources ».

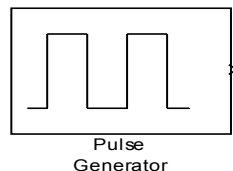


FIGURE 6.1 : Générateur d'impulsion

Le modèle de régulateur utilisé est le type Proportionnel Intégral, que ce soit pour le régulateur de vitesse que pour le régulateur de courant. Nous commencerons donc par modéliser le caractère proportionnel du régulateur du courant.

XIV. Modélisation du régulateur de courant :

a. Comportement du régulateur proportionnel :

L'action proportionnelle a pour fonction première d'assurer la rapidité du système. L'action proportionnelle P crée un signal de commande $u(t)$ proportionnel au signal d'erreur $e(t)$. Le facteur de proportionnalité K_p est la transmittance du correcteur, en l'occurrence son gain statique.

La relation liant l'entrée et la sortie est donc la suivante :

$$u(t) = K_p e(t) \quad (6-1)$$

La sortie du correcteur P est le signal d'entrée amplifié d'une constante K_p .

b. Mise en œuvre de l'algorithme pour l'action Proportionnelle :

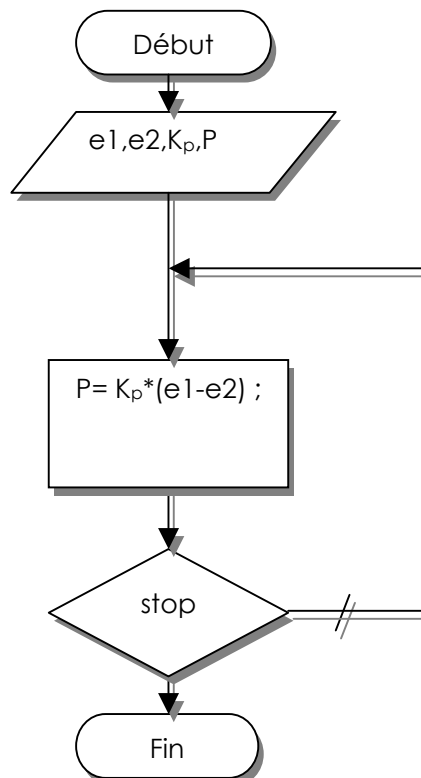


FIGURE 6.2 : Organigramme pour l'action proportionnelle

Chapitre 6 : Application de STATEFLOW à la régulation de vitesse

Explication de l'algorithme :

Les variables d'entrées 'e1' et 'e2' représentent respectivement les valeurs des grandeurs de référence de la chaîne (w_{ref} vitesse de référence définie par l'utilisateur et I_{ref} valeur du courant de référence fourni par le correcteur de vitesse) et celles des grandeurs délivrées par la machine (w_m vitesse de sortie du moteur et I_m valeur du courant d'induit).

La variable 'P' calcule l'erreur et le multiplie par un facteur ' K_p ' qui est le gain statique du correcteur proportionnel. La condition 'stop' est fausse tant que l'on n'appuiera pas sur le bouton stop de la simulation (ou qu'elle ne se termine par elle-même) ; la seule action qui n'est pas représentée sur l'organigramme est que la fréquence d'exécution de la boucle est de 100Hz.

c. Conception du modèle STATEFLOW du correcteur Proportionnel :

1. Les données:

On utilise deux types de données qui sont généralement les entrées et les sorties ; les entrées sont des variables de SIMULINK et les sorties des données STATEFLOW qui ont été traitées à l'intérieur de diagramme et qui seront restituées à SIMULINK. Le format des données peut être fixé à partir du 'Model Explorer' de STATEFLOW, comme on traite différents types de signaux on choisit l'option type de donnée '*inherit*'. Dans notre cas les variables d'entrées sont e1 et e2 et la variable de sortie est P.

2. Les événements:

Ce sont les événements qui activent le diagramme ; dans notre cas comme nous modélisons un régulateur qui doit fonctionner constamment l'événement est le temps de simulation. Mais comme le temps est un système continu il a fallu activer le diagramme STATEFLOW à une cadence plus élevée, donc notre choix s'est porté sur une suite d'impulsion de fréquence élevée.

Les paramètres du générateur d'impulsion sont définis de la manière suivante:

- Type d'impulsion : fonction temporelle
- Durée : utilisation du temps de simulation
- Amplitude : 1
- Période : 0.01 [sec]
- Largeur d'impulsion : 50%

3. Les fonctions utilisées:

Ce sont des blocs graphiques qui regroupent une liste de syntaxe pour l'exécution d'une tâche bien précise, nous utiliserons la fonction 'erreur(K_p)' pour évaluer le caractère proportionnel du régulateur.

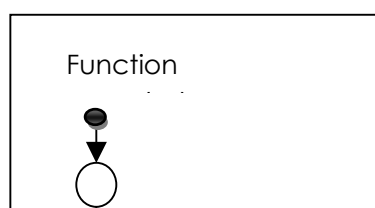


FIGURE 6.3 : Fonction erreur(K_p)

L'exécution de cette fonction se fera dans un état que l'on nommera 'Proportionnel/

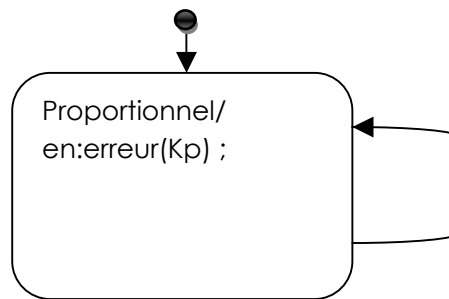


FIGURE 6.4 : Modèle de l'état proportionnel

On remarque également que le trigger du diagramme est de type 'either' c'est-à-dire qu'il est réveillé lorsque l'impulsion est à front montant et à front descendant.

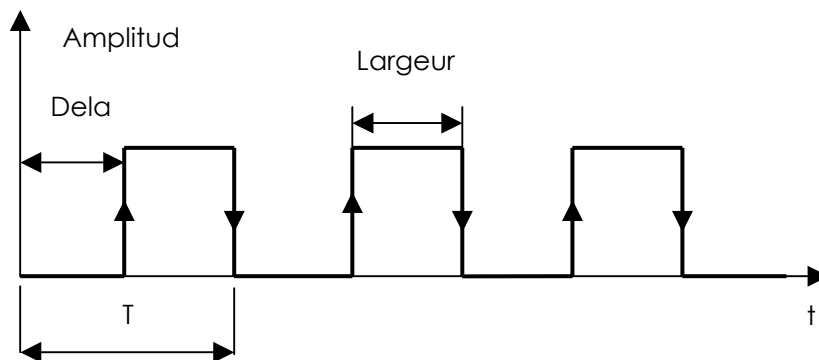


FIGURE 6.5 : Allure de la commande du trigger de l'action proportionnelle

d. Comportement du correcteur intégral :

Chapitre 6 : Application de STATEFLOW à la régulation de vitesse

Le correcteur intégral a pour fonction d'éliminer l'erreur statique du système c'est-à-dire d'améliorer la précision de la régulation. L'action Intégrale (I) crée un signal de commande $u(t)$ qui est l'intégrale du signal d'erreur $e(t)$. L'équation du correcteur est la suivante :

$$u(t) = K_I \int_0^t e(t) dt \quad (6-2)$$

L'intervalle de temps d'intégration $[0, t]$ sera la durée de la simulation ou alors le temps au bout duquel le manipulateur arrête la simulation. K_I étant le gain de l'intégrateur.

Où K_I est le gain permanent du correcteur et $T_i = \frac{1}{K_I}$ sa constante de temps d'intégration.

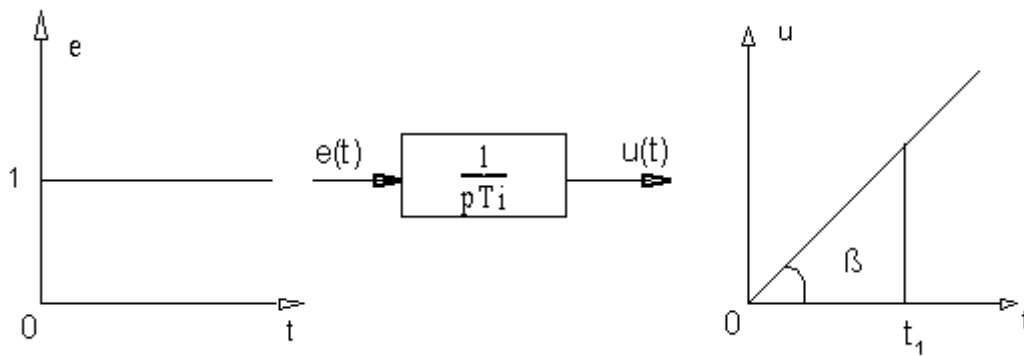


FIGURE 6.6: Réponse indicielle d'un correcteur intégral

La réponse indicielle du correcteur montre qu'un correcteur à action purement intégral est lent et de plus le signal de commande $u(t)$ reste constant lorsque le signal d'erreur $e(t)$ s'annule. On a :

$$\tan(\beta) = \frac{1}{T_i}$$

On constate sur la figure (6.6) que la réponse d'un intégrateur à une consigne indicielle est un signal rampe de pente β . Si la consigne indicielle est positive la rampe est croissante, par contre si la consigne est négative, la réponse décroît avec une pente de $-\beta$. On utilisera ce comportement pour modéliser le correcteur. Avec comme entrée du contrôleur intégral l'erreur entre la référence et la grandeur délivrée par le moteur, et la sortie, le signal de commande.

Chapitre 6 : Application de STATEFLOW à la régulation de vitesse

Ce type de correcteur à action purement intégral n'est donc jamais utilisé seul à cause de sa lenteur, mais on fait appel à l'action intégrale chaque fois qu'une erreur permanente doit être annulée.

e. Mise en œuvre de l'algorithme pour l'action Intégral :

Comme on l'a vu précédemment sur la réponse indicielle de l'intégrateur, elle délivre une droite de pente β si le saut est positif et une pente de $-\beta$ si elle est négative, nous allons donc essayer de créer une réponse identique en supposant que dans notre cas la consigne d'entrée $e(t)$ est la différence des deux entrées ' $e_1 - e_2$ ' et la sortie $u(t)$ est la variable ' s '.

L'organigramme se présente comme suit :

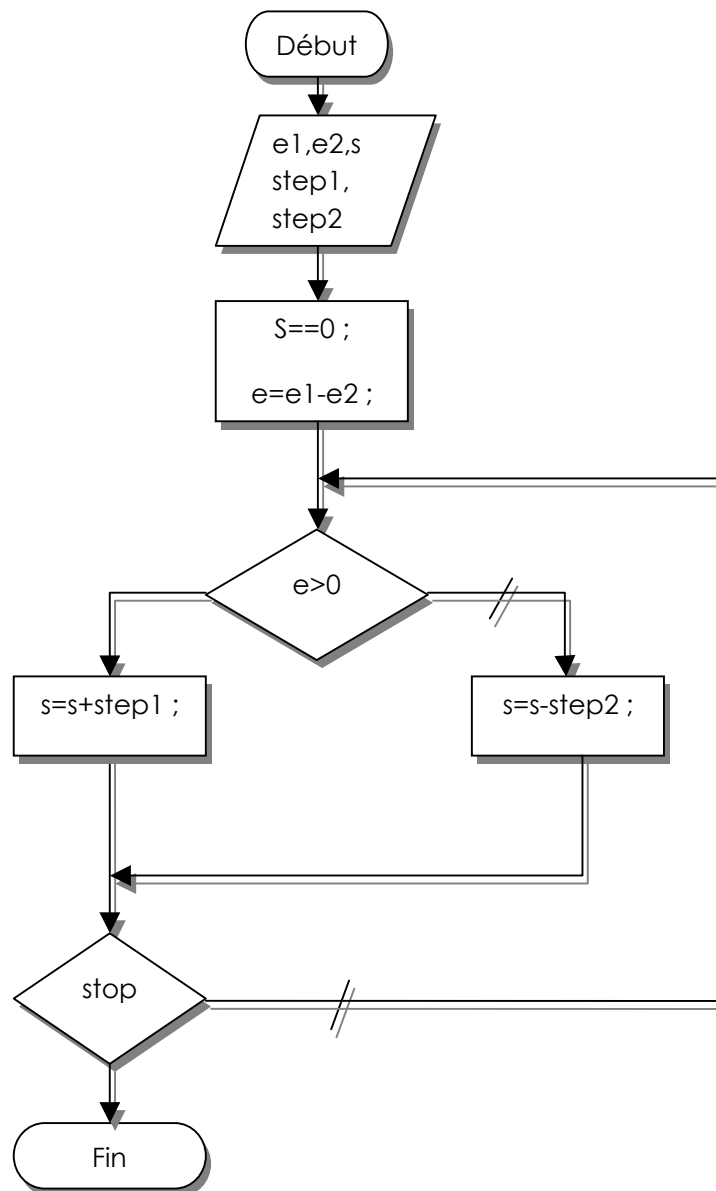


FIGURE 6.7 : Organigramme pour la commande intégrale de courant

Explication de l'algorithme :

Dans cet organigramme les entrées sont notées 'e1' et 'e2' représentent respectivement les valeurs des grandeurs de référence de la chaîne (I_{ref} valeur du courant de référence) et celles des grandeurs délivrées par la machine (I_m valeur du courant d'induit), 's' étant la sortie du contrôleur, les valeurs de 'step1' et de 'step2' constituent les paramètres de réglage du régulateur.

La variable 'e' calcule l'erreur qui est la différence entre la valeur de référence et celle de la valeur à réguler. La condition est basée sur le signe de l'erreur. Si celle ci est positive,

Chapitre 6 : Application de STATEFLOW à la régulation de vitesse

cela signifie que la grandeur à réguler est inférieure à la référence et nécessite d'être augmentée dans le cas contraire elle sera diminuée.

La sortie 's' étant initialisée à 0 (condition initiale) elle représente la courbe de variation de l'action intégrale. La condition 'stop' est vraie si le bouton stop de la simulation est enfoncée (ou s'arrête automatiquement) ; la fréquence d'exécution de la boucle est de 100Hz.

f. Mise en œuvre du modèle STATEFLOW du correcteur Intégral :

1. Les données et les évènements:

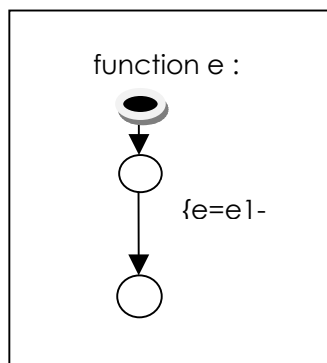
Les données et les évènements ont les mêmes caractéristiques que pour celles pour le correcteur proportionnel.

2. Les fonctions utilisées :

Ici la fonction utilisée consiste à incrémenter ou décrémenter une variable 's1' d'un pas 'step' en fonction de la valeur de l'erreur calculée par la différence entre l'entrée 'e1' et l'entrée 'e2'.

- La fonction 'error()' :

Elle sert à calculer la différence entre la grandeur de référence fixée et celle fournie par le moteur à courant continu



- La fonction 'decrement(step)' : elle consiste à augmenter ou à diminuer la variable 's1' d'une valeur 'step' en fonction de 'e' et la sortie 's2' affiche 1 lorsque la référence 'e1' est supérieur à 'e2' et inversement, cette valeur de 's2' n'a pas été représenté sur le diagramme mais elle permet de suivre l'évolution de l'erreur. La fonction *decrement(step)* est représentée par la figure (6.8).

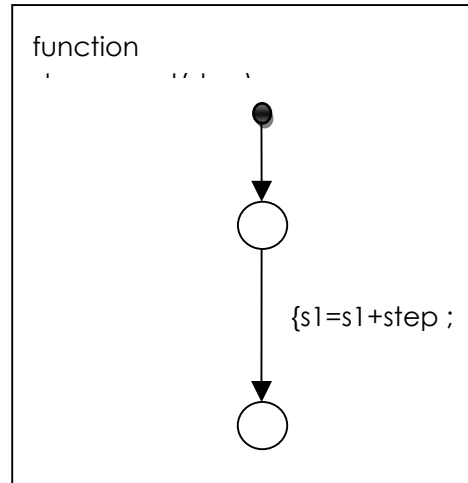


FIGURE 6.8 : Fonction *decrement(step)*

Ces deux fonctions s'appliquent dans le diagramme comme suit :

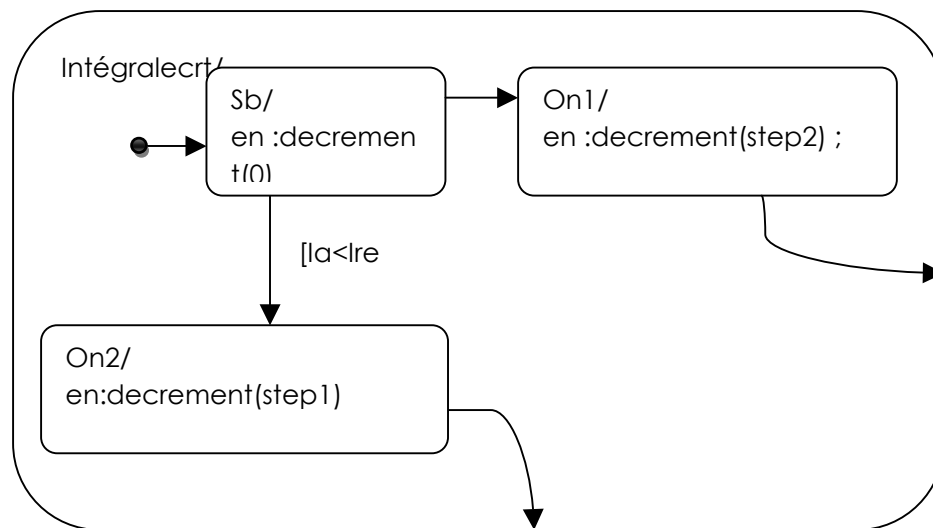


FIGURE 6.9 : Modèle de l'état intégral

Dans le cas suivant l'évènement est à front montant, le diagramme est réveillé lorsque l'impulsion du trigger monte. Cette particularité diffère de celui de la figure (6.5) en ce que le diagramme est réveillé deux fois moins pour une même fréquence.

Le *delay* du trigger est nul, l'amplitude du niveau haut est de 1 et la valeur de la période est de 0.01 seconde, nous avons un rapport cyclique de 50% de la période pour le trigger de l'action '*integralcrt/*'. Lors de la création du modèle PI de courant de la figure (6.11) ; nous avons choisit le signal de commande de la figure (6.10) pour réveiller le diagramme entier.

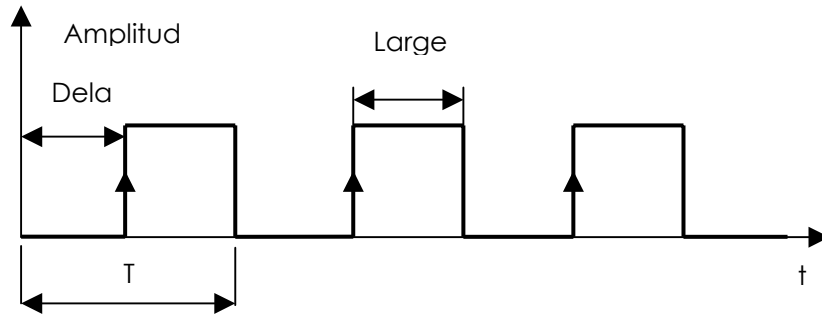


FIGURE 6.10 : Signal de commande du trigger de l'action 'intégralcrt/'

g. Mise en œuvre du correcteur Proportionnel Intégral de courant :

Le correcteur PI est la somme des correcteurs proportionnel et intégral, nous avons essayé de grouper ces deux correcteurs dans un seul bloc STATEFLOW ce qui a aboutit au modèle suivant :

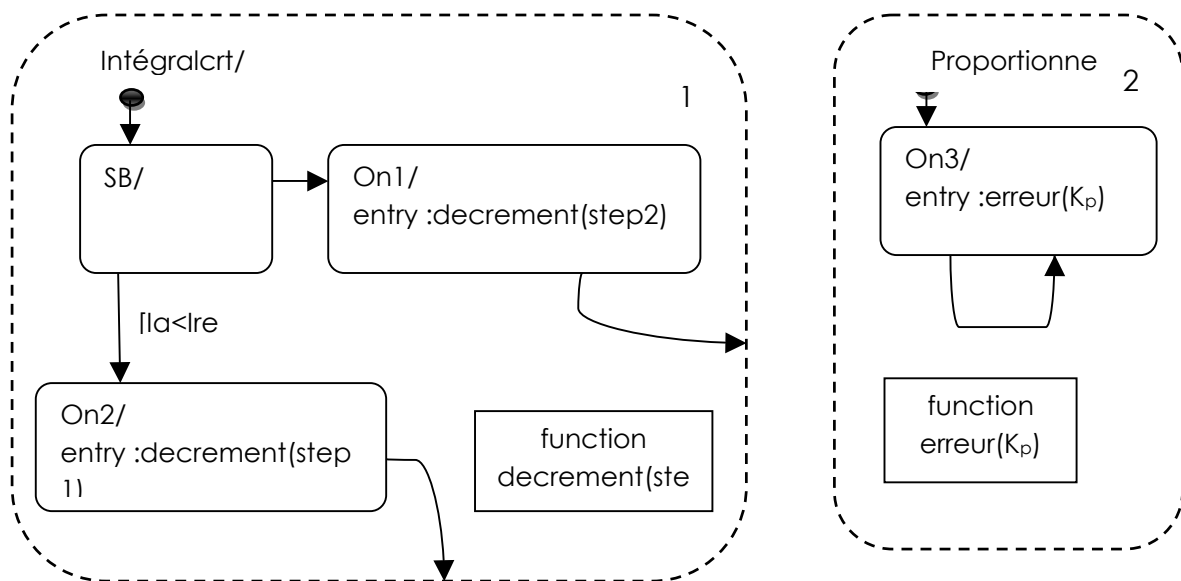


FIGURE 6.11 : Modèle à état du correcteur Proportionnel intégral de courant

On a utilisé la décomposition en parallèle pour simuler le correcteur PI, ceci est la seule façon de représenter un système fonctionnant simultanément car l'action proportionnelle et intégrale doit se faire conjointement. La fonction erreur(Kp) a été modifiée pour restituer à sa sortie chaque fois qu'il est appelé la somme de l'action Intégrale et celle de l'action proportionnelle.

h. Affectation des paramètres du contrôleur PI de courant :

Les paramètres caractéristiques régissant le comportement du régulateur PI sont les valeurs de step1 , step2 et K_p . Ces valeurs ont été choisies après diverses simulations, elles dépendent généralement de la valeur de la vitesse de référence, la valeur de K_p pour le correcteur de courant a été choisie pour que la valeur du courant d'induit du moteur corresponde à celui du modèle classique (fonction de transfert). On constate qu'un coefficient proportionnel trop élevé entrainerait un courant de démarrage important, donc il y aurait risque d'endommagement de la machine.

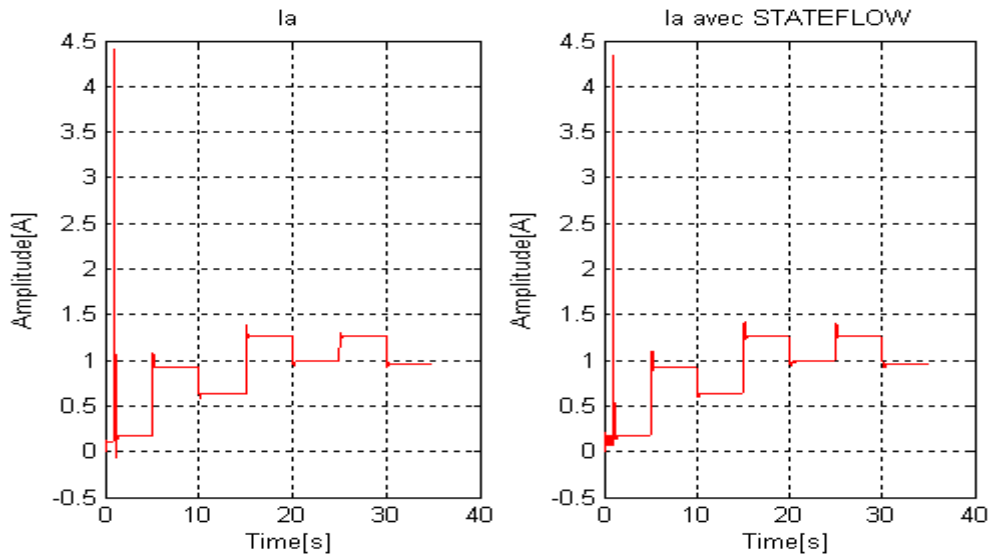


FIGURE 6.12 : Allure du courant I_a avec le modèle classique et avec STATEFLOW

Par contre la valeur de step1 et step2 permet de contrôler l'oscillation du courant, elles devront être le plus faible possible pour permettre d'avoir un courant d'induit I_a , ayant de faibles oscillations voire même avoir un courant I_a constant.

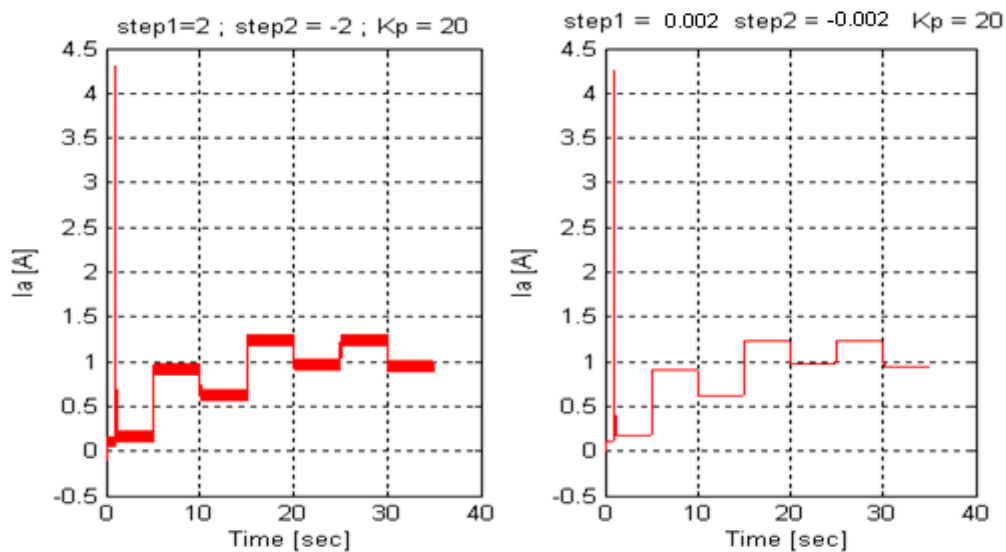


FIGURE 6.13 : Effet des paramètres du contrôleur sur le courant I_a

XV. Modélisation du correcteur de vitesse :

Le modèle du contrôleur de vitesse diffère de celui du contrôleur de courant par le fait qu'on a créé une zone, en fonction de l'amplitude de la vitesse du moteur 'wm' par rapport à la référence 'wref', les coefficients de variation étant différents pour chaque zone. Le modèle de l'état pour la commande proportionnelle étant la même que celle du correcteur de courant, nous avons introduit par contre un état supplémentaire dans le modèle intégral.

a. Mise en œuvre de l'algorithme

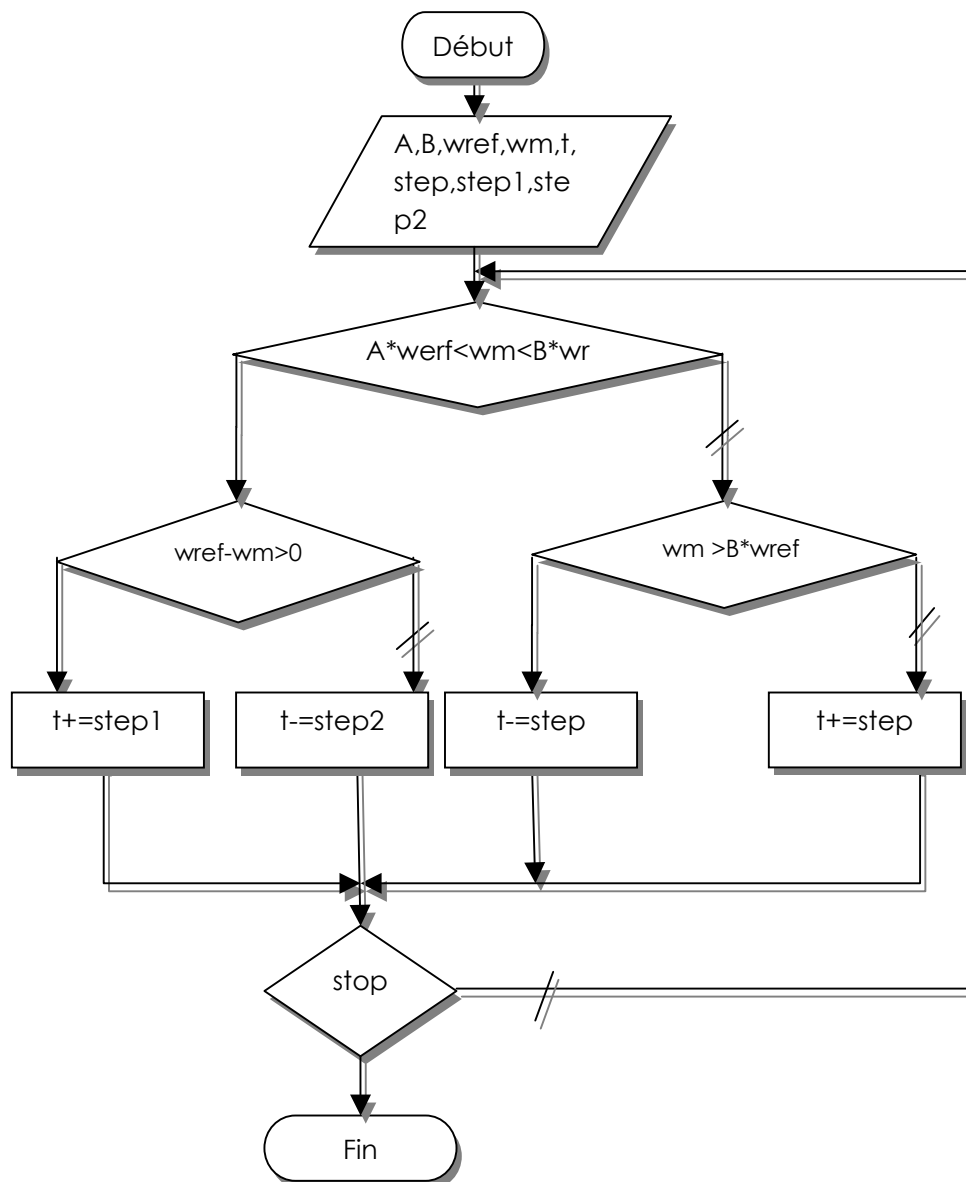


FIGURE 6.14 : Organigramme pour la commande intégrale de vitesse

Chapitre 6 : Application de STATEFLOW à la régulation de vitesse

Le contrôleur modélisé par l'organigramme de la figure (6.14) possède comme paramètre de fonctionnement les grandeurs A, B, step, step1 et step2. Le choix de ces paramètres permettent de contrôler la valeur de la sortie du régulateur soit la valeur de Iref.

Explication de l'algorithme :

Lorsque la valeur de la vitesse réelle du moteur 'wm' est inférieure à une certaine valeur finale 'A*wref', la fonction de sortie incrémente d'une valeur 'step1' dans le cas contraire si 'wm' est supérieure à 'B*wref', la fonction de sortie du diagramme décrémente d'une valeur 'step2'. Par contre si la vitesse se trouve entre ces deux limites, il existe deux cas possibles ; soit l'erreur est positive soit elle est négative, dans le cas où l'erreur est positive la sortie incrémente d'une valeur 'step' dans le cas contraire elle décrémente de cette même valeur.

b. Mise en œuvre du modèle STATEFLOW :

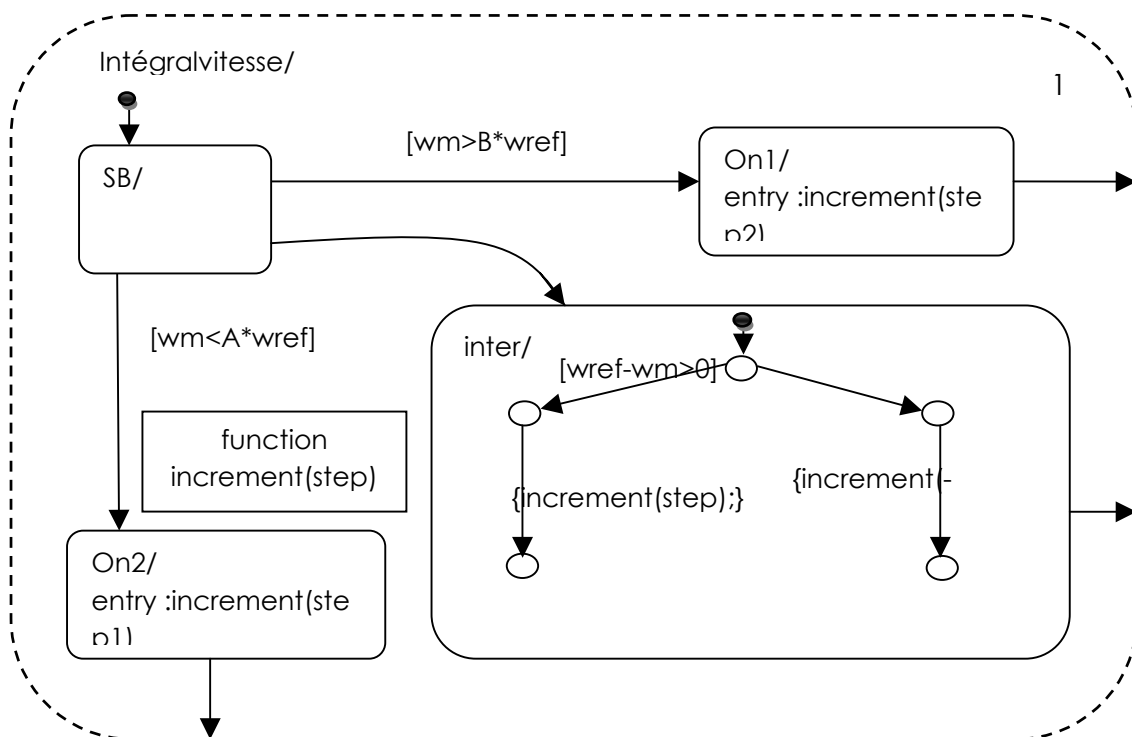


FIGURE 6.15 : Modèle du correcteur intégral de vitesse

Le modèle du contrôleur de vitesse est identique à celle de la figure (6.11) avec à la place de l'état 'intégralcrt/', le modèle 'intégralvitesse/' de la figure (6.15).

c. Affectation des paramètres du contrôleur PI de vitesse :

Le modèle du contrôleur de vitesse est paramétré par les valeurs de step1, step2, step, Kp, A et B.

Chapitre 6 : Application de STATEFLOW à la régulation de vitesse

Les grandeurs 'Awref' et 'Bwref' étant les valeurs limite de l'oscillation de la vitesse en régime permanent, elles permettent de créer une zone à l'intérieur duquel la vitesse du moteur ne varie plus mais devient constante.

Ce couloir est généralement inférieur à 10% de la consigne réelle, ceci est fait pour limiter les oscillations de la grandeur de sortie.

Le choix de K_p se fait en fonction de la réponse en boucle fermée du moteur avec un correcteur proportionnel, la valeur de la vitesse de sortie du moteur dans le modèle STATEFLOW doit être sensiblement identique à celui de la vitesse du modèle classique. Nous obtenons la figure suivante dans les deux cas de simulation.

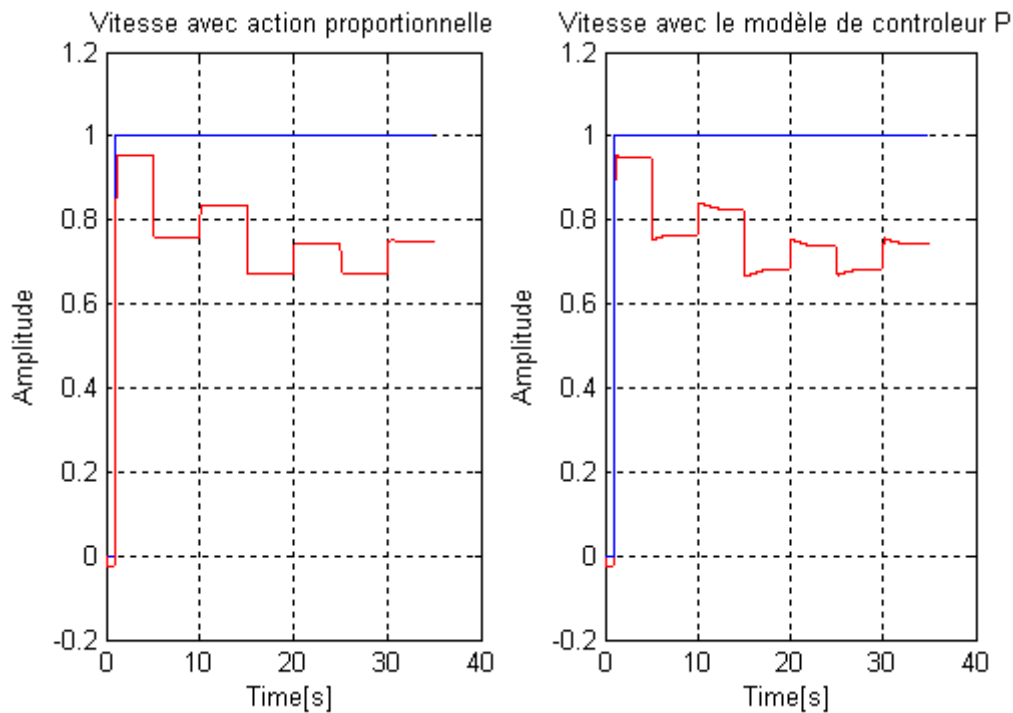


FIGURE 6.16 : Influence de l'action Proportionnelle sur la vitesse

Le choix du paramètre *step* définit l'amplitude des oscillations de la vitesse dans la zone [Awref, Bwref].

Le choix de la valeur de *step1* définit le temps de montée du système, si cette valeur est élevée le système est rapide et il est possible de constater l'apparition d'oscillation de vitesse due a cette trop grande rapidité.

Le choix de *step2* définit la rapidité du système à décroître la valeur de la vitesse, une valeur de *step2* trop élevée provoquerait lors des variations de couples, des baisses de vitesse de grandes amplitudes et cela malgré l'effet du correcteur de courant, des oscillations peuvent aussi se produire autour de la valeur de référence de la vitesse.

Par contre une valeur de 'step1' et 'step2' trop faible entrainerait un large régime transitoire, c'est-à-dire que la vitesse tend trop lentement vers celle de la référence ce qui entrainerait une mauvaise réaction de système aux perturbations car la correction serait trop lente.

En modifiant ces paramètres nous avons effectué une série de mesures pour mieux comprendre leur incidence sur la sortie.

Nous avons créé une interface pour pouvoir facilement accéder aux paramètres du contrôleur de vitesse, après diverses simulations nous avons obtenus les résultats suivants :

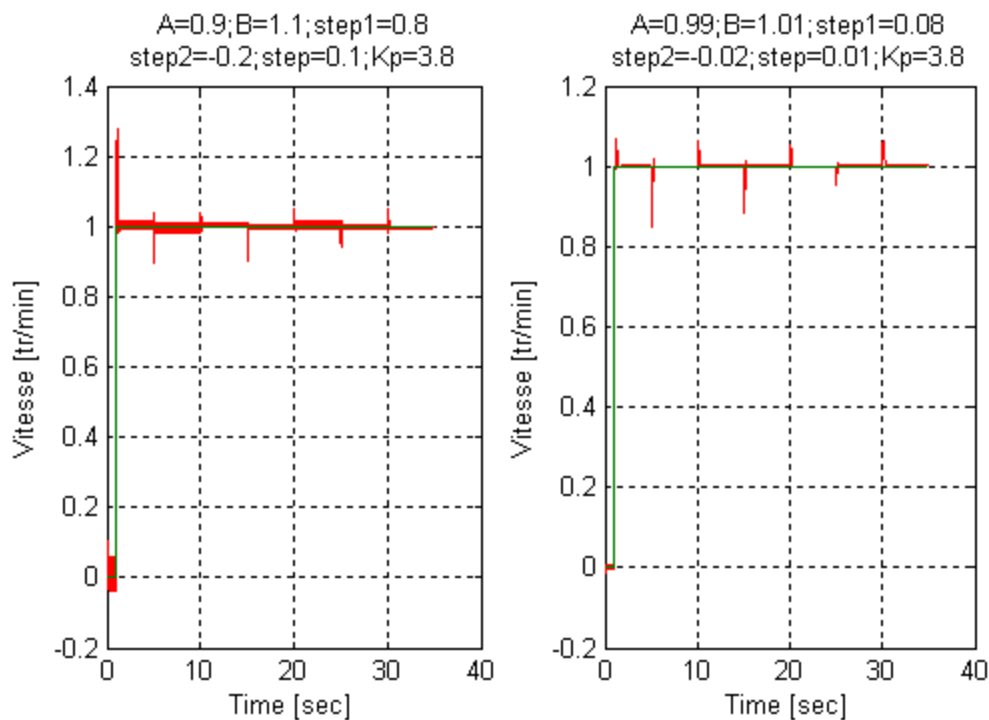


FIGURE 6.17 : Influence des paramètres du contrôleur sur la vitesse

XVI. Présentation de la chaîne de régulation et simulations:

Elle présente le cas d'une régulation classique en cascade, il existe une boucle de courant qui est implémentée à l'intérieur d'une boucle de vitesse. En remplaçant les correcteurs de vitesse et de courant par leurs modèles STATEFLOW équivalent, on obtient le schéma de montage de la figure (6.19).

On constate sur la figure (6.18) qui suit la présence d'oscillations non amorties dans le cas de l'utilisation du correcteur de vitesse seul, on observe également que l'amplitude du premier dépassement est exagéré approximativement égal à 45% de la valeur finale.

Par contre lorsque le montage en cascade est effectué on constate sur la figure (6.22) que les oscillations ont disparus et que l'amplitude du premier dépassement est d'environ 5% de la valeur finale.

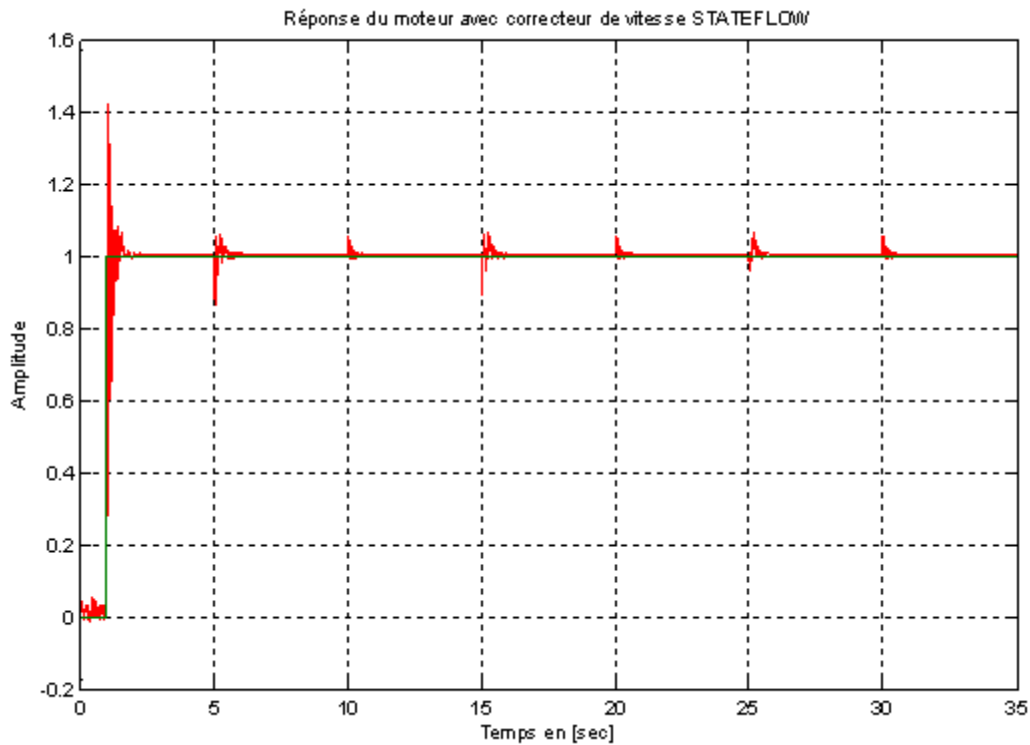


FIGURE 6.18 : Allure de la vitesse avec le modèle du correcteur de vitesse

On observe sur la figure (6.21) l'allure du courant de commande délivré par le correcteur de vitesse, on constate que des oscillations de courants de faibles amplitudes existent au début de la simulation ceci est dû au fait que la réponse en vitesse du moteur passe de part et d'autre de la valeur de référence (qui est nulle) créant ainsi tantôt un couple positif tantôt un couple négatif ; ce phénomène ne disparaît pas mais, elle est moindre en régime permanent.

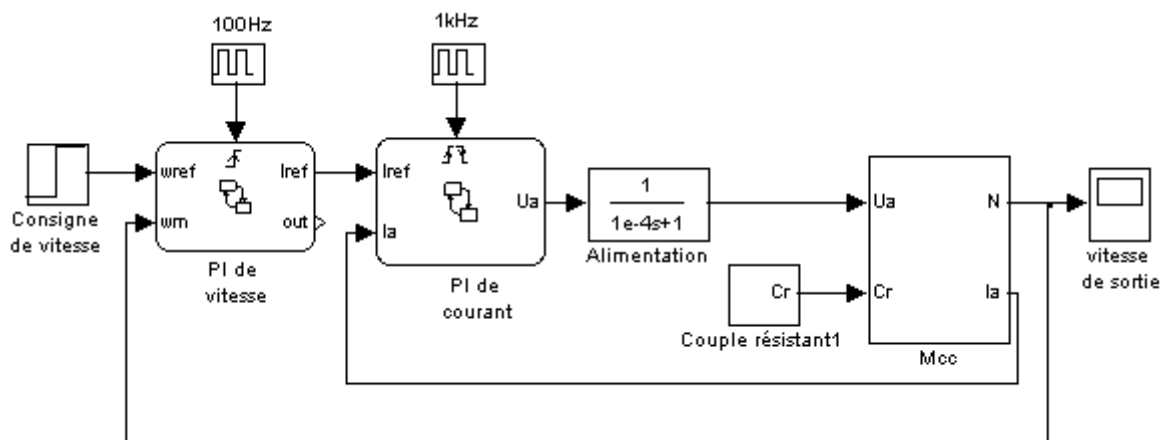


FIGURE 6.19 : Principe de montage pour la régulation en cascade avec correcteur STATEFLOW

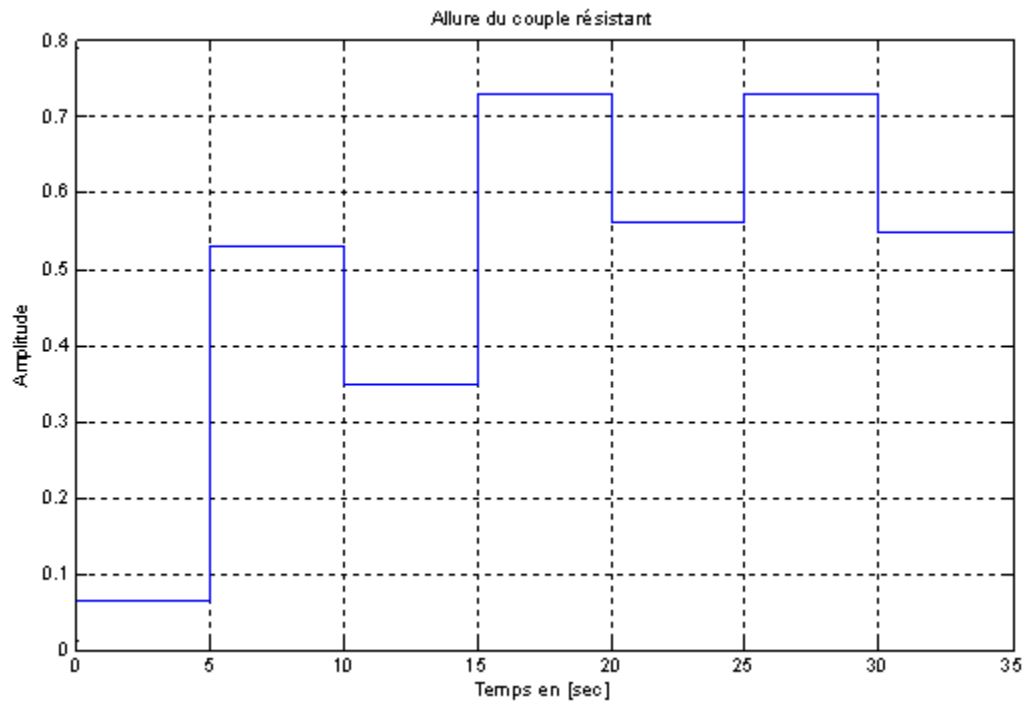


FIGURE 6.20 : Allure de la variation du couple résistant

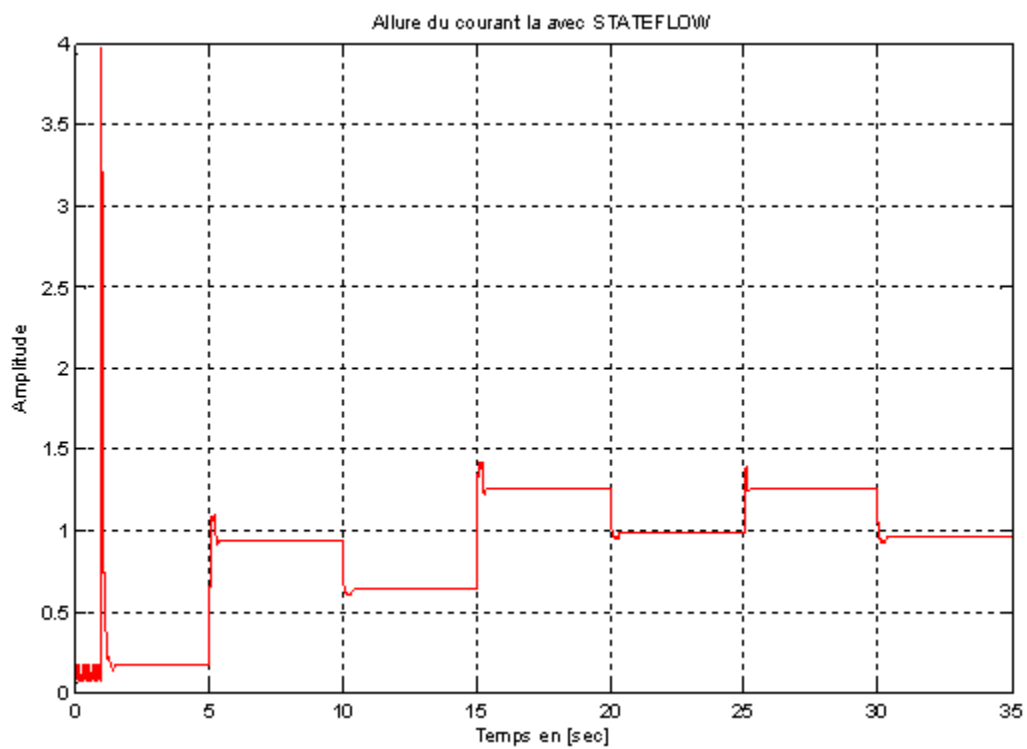


FIGURE 6.21 : Allure du courant de référence délivré par le correcteur de vitesse

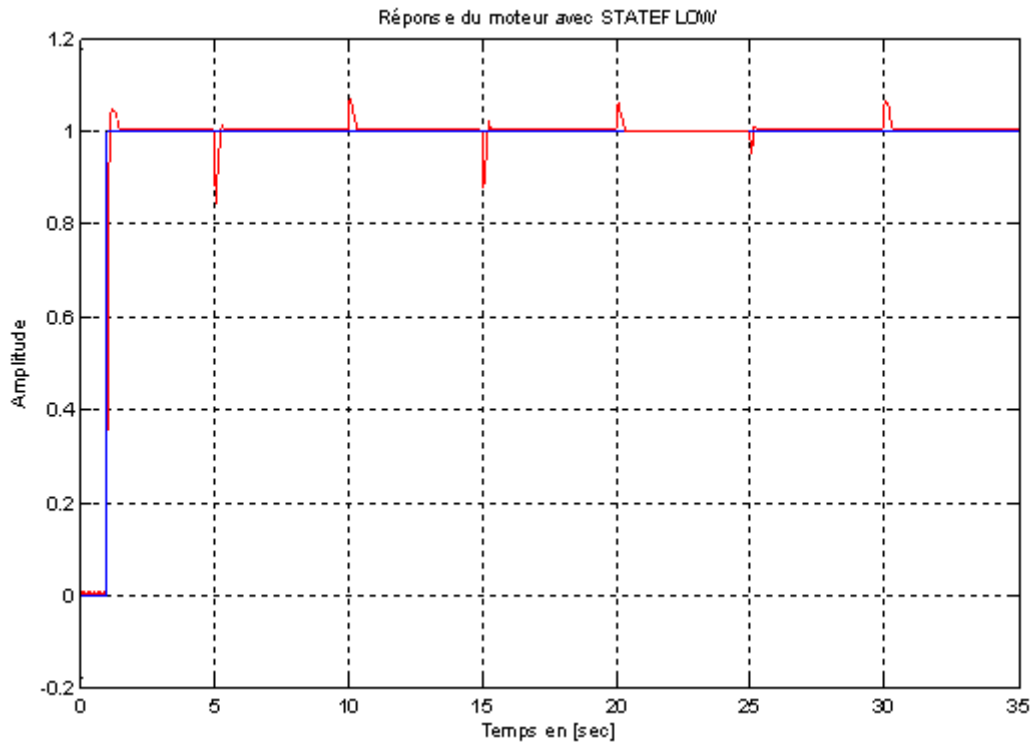


FIGURE 6.22 : Allure de la vitesse pour la régulation en cascade avec STATEFLOW

On constate sur la courbe de la figure (6.21) qu'une augmentation du couple résistant se traduit par l'accroissement de la valeur du courant de référence, cela peut se traduire par le fait que lorsque le couple résistant augmente, la vitesse diminue et le moteur réagit à ce phénomène en créant un surplus de la valeur du couple moteur, comme le couple électromagnétique est directement proportionnel au courant d'induit ; une augmentation de la valeur du couple se traduit par une augmentation du courant I_a . Le phénomène se reproduit mais dans le sens inverse dans le cas de la diminution du couple résistant. D'où l'on constate pour le temps $t = 5$ [sec] qu'une augmentation de la valeur de C_r se traduit sur la figure (6.21) par une augmentation du courant d'induit I_a , et pour $t = 10$ [sec] une diminution de C_r se traduit dans la figure (6.21) par une diminution de I_a .

Chapitre 7

Régulation de vitesse des moteurs à courant continu dans le cas réel

I. Introduction :

Le but de ce chapitre est de simuler le comportement réel du moteur à courant continu corrigé en vitesse lorsqu'il est alimenté par un convertisseur statique. On utilisera pour cela la boîte à outils 'SimPowerSystems' de MATLAB [16] pour étudier le comportement de la machine. Le problème se pose ici dans le fait de commander l'alimentation de la machine pour qu'il puisse fonctionner dans les quatre quadrants ; le paramétrage du correcteur de vitesse, le choix du convertisseur statique et la modélisation de la logique de commande sera donc les objectifs de ce chapitre.

II. Présentation de la chaine de régulation :

a. Les éléments de la chaine de régulation :

Pour la régulation de la vitesse du moteur à courant continu dans le cas réel le montage utilisé est la suivante:

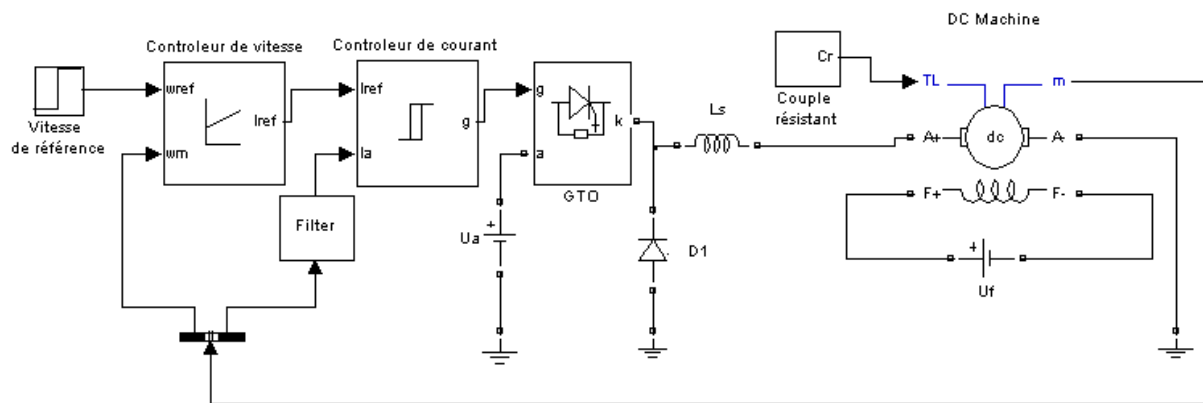


FIGURE 7.1: Chaîne de régulation en cascade

La chaîne de régulation comprend les éléments suivants

- Le moteur à courant continu
- Le contrôleur de vitesse
- Le contrôleur de courant
- L'alimentation du moteur.

Chapitre 7 : Régulation de vitesse des moteurs à courant continu dans le cas réel

b. Choix du moteur :

Nous utiliserons pour la simulation le moteur de type 5 HP/240 [V] avec une vitesse nominale de 1500 [tr/min] ; les caractéristiques du moteur sont énumérées dans le tableau suivant :

Paramètres nominaux	Valeurs numériques
Puissance nominale	$P_n = 3[\text{kW}]$
Tension nominale d'induit	$U = U_{an} = 240[\text{V}]$
Tension d'excitation	$U_{fn} = 280[\text{V}]$
Courant d'induit	$I_{an} = 30[\text{A}]$
Courant nominal de l'inducteur	$I_{fn} = 1.3 [\text{A}]$
Vitesse de rotation	$N=1500[\text{tr/min}]$
Paramètres du circuit d'induit	$R_a=0.5 [\Omega] ; L_a=0.01 [\text{H}]$
Paramètres du circuit inducteur	$R_f=240[\Omega] ; L_f=120[\text{H}]$
Moment d'inertie totale du moteur	$J=0.05 [\text{kg.m}^2] ;$
Couple résistant	$C_r = 5[\text{N.m}]$
Coefficient de frottement visqueux	$f=0.02 [\text{N.m.s}]$
Constante du couple	$K_E=k\Phi=1.23$

TABLEAU 7.1 : Tableau des caractéristiques de la machine

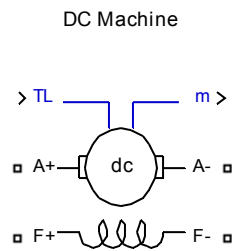


FIGURE 7.2 : Modèle macro du moteur

Le moteur est constitué de trois parties :

- le circuit d'armature qui est un circuit RL constituant le circuit d'induit,
- le circuit de champs qui est lui aussi composé de la résistance R_f et de l'inductance L_f de l'inducteur
- la partie mécanique qui a pour rôle de créer la fonction suivante :

$$\boxed{C_m = J \frac{d\omega}{dt} + f\omega + C_r \text{sign}(\omega)} \quad (7-1)$$

c. Allure du couple résistant variable :

Le couple résistant a été conçu de la façon suivante dans le but d'observer le comportement du moteur dans le cas d'une variation la charge.

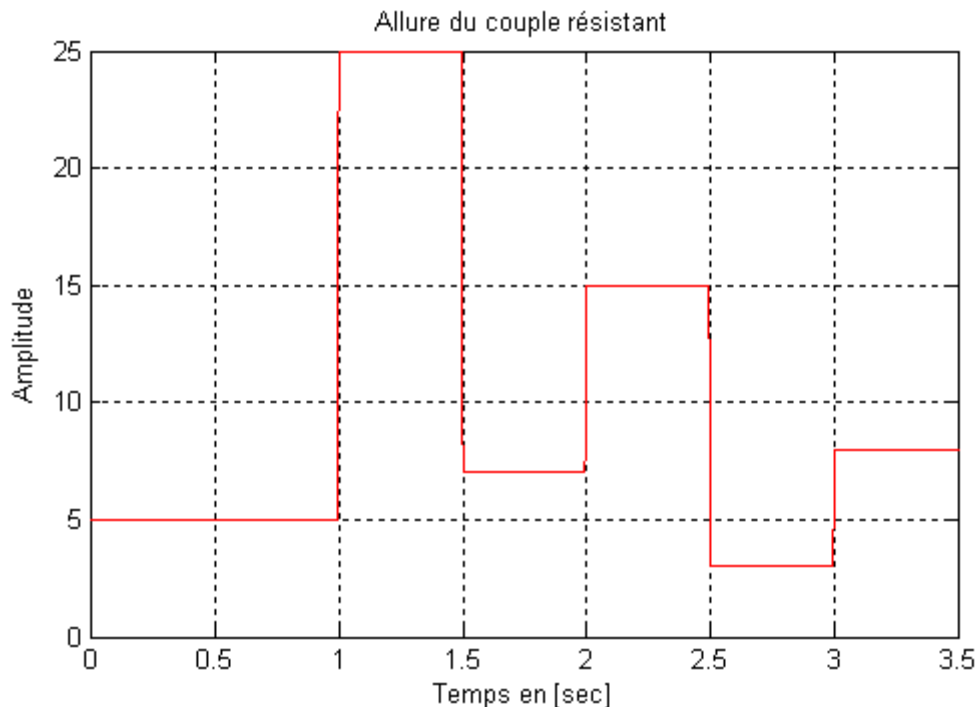


FIGURE 7.3 : Allure du couple résistant

d. Présentation des correcteurs :

1. Présentation du contrôleur de vitesse :

Le contrôleur de vitesse est un régulateur proportionnel intégral (PI) modélisé par le schéma bloc suivant :

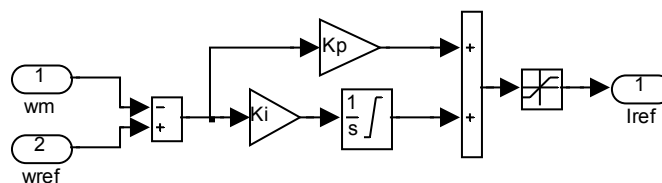


FIGURE 7.4 : Modèle de correcteur PI

L'action du régulateur est limitée par un organe non linéaire « bloc saturation ». Cet organe à pour effet de laisser passer le signal I_{ref} , si il se trouve entre deux limites supérieures et inférieures, par contre au delà de ces limites, le bloc sature la valeur.

Chapitre 7 : Régulation de vitesse des moteurs à courant continu dans le cas réel

Cette action permet de protéger le système lors du démarrage et pendant les variations importantes du couple de distorsion.

2. Présentation du contrôleur de courant :

Le contrôleur de courant quant à lui est un générateur d'impulsion, on la représente par la figure suivante :

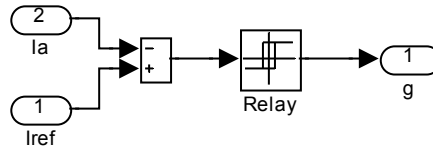


FIGURE 7.5 : Modèle de contrôleur de courant

Le contrôleur de courant sert généralement à la gestion de l'alimentation du moteur à courant continu, il consiste à créer une séquence logique de commande des thyristors ou éventuellement des transistors du convertisseur statique alimentant le moteur. Comme on l'observe sur la figure (7.5), c'est également un élément non linéaire. Nous allons faire un bref aperçu sur les éléments non linéaires dans la suite de notre étude.

e. Introduction sur les non linéarités :

1. Généralités :

Pour observer le comportement réel d'un système afin de prendre en compte une réalité plus complexe, il est utile de retenir dans la modélisation du système physique des éléments non linéaires qui sont difficiles à modéliser et que l'on ne peut approximer. Du fait de la puissance des outils développés dans le domaine linéaire, il est avantageux de linéariser le modèle non linéaire autour d'un point de fonctionnement et d'utiliser le modèle ainsi obtenu afin d'en extraire le plus d'information.

2. Système non linéaire :

Dans la suite de l'étude nous considérons uniquement les systèmes asservis possédant un élément non linéaire invariant dans le temps, dans la chaîne de régulation. Nous obtenons le schéma simplifié de régulation suivant :

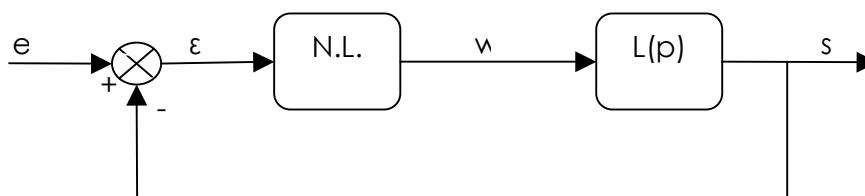


FIGURE 7.6 : Chaîne de régulation avec un organe non linéaire

Chapitre 7 : Régulation de vitesse des moteurs à courant continu dans le cas réel

Nous supposons que le bloc N.L regroupe les organes non linéaires de la chaîne de régulation et le bloc L(p) celui des éléments linéaires.

3. Critère de stabilité au sens de Lyapunov :

Par définition si un système est dans un état d'équilibre, il restera dans cet état pour t variant dans le temps. L'étude de la stabilité au sens de Lyapunov consiste en l'étude des trajectoires du système quand l'état initial est près d'un état d'équilibre. Au sens de Lyapunov [3] un état d'équilibre x_e est dit stable si :

$$(\forall \varepsilon > 0) (\forall \alpha > 0) (||x(0) - x_e|| < \alpha \rightarrow ||x(t) - x_e|| < \varepsilon, \forall t \geq 0 ;)$$

Cela signifie que la trajectoire peut être gardée arbitrairement près de x_e , si l'on prend une condition initiale suffisamment proche de x_e .

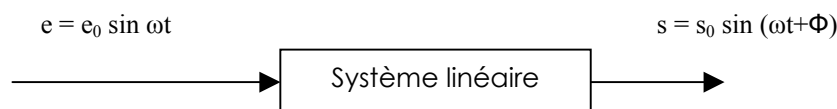
4. Etude des systèmes non linéaires par la méthode de l'approximation du premier harmonique :

a. Introduction :

L'idée de cette méthode est de réaliser une linéarisation du système non linéaire dans le domaine fréquentiel, de façon à généraliser la notion de fonction de transfert, il s'agit donc d'une méthode approchée qui s'applique à toutes les formes de caractéristiques non linéaires, à condition que le système contienne un élément qui joue le rôle d'un filtre passe-bas à bande passante suffisamment étroite.

b. Principe de la méthode :

Généralement pour un système linéaire d'entrée $e = e_0 \sin \omega t$ la réponse est également sinusoïdale et de même pulsation $s = s_0 \sin (\omega t + \Phi)$;



La fonction de transfert est alors définie par :

- Son amplitude $A = \frac{s_0}{e_0}$
- Sa phase Φ

Ainsi pour un système linéaire on aboutit à une fonction de transfert qui ne dépend que de ω .

Chapitre 7 : Régulation de vitesse des moteurs à courant continu dans le cas réel

Etendons le cas à un organe non linéaire supposé isolé d'entrée $x = x_1 \sin \omega t$ et de sortie $w(t)$; cette fois en régime permanent la réponse $w(t)$ n'est pas sinusoïdale mais seulement, en général, périodique de même période que l'entrée. On peut la décomposer en série de Fourier, ce qui donne, si la caractéristique d'amplitude entrée/sortie de la non linéarité est symétrique par rapport à l'origine (valeur moyenne nulle):

$$w(t) = \sum_{n=1}^{+\infty} w_n \cdot \sin(n\omega t + \psi_n)$$

L'approximation du premier harmonique consiste à considérer le premier harmonique (rang $n=1$) et à oublier les autres harmoniques supposés filtrés par un système linéaire situé en aval de la non linéarité :

$$w(t) = w_1 \cdot \sin(\omega t + \psi_1)$$

c. Fonction de transfert généralisée, gain complexe équivalent et lieu critique :

Par analogie avec le cas linéaire, on peut définir la fonction de transfert généralisée $N(x_1, \omega)$:

$$w(t) = a_1 \cdot \sin \omega t + b_1 \cdot \cos \omega t$$

a_1 et b_1 étant les coefficients de Fourier tel que :

$$a_1 = \frac{\omega}{\pi} \int_0^t w(t) \sin(\omega t) dt$$

$$b_1 = \frac{\omega}{\pi} \int_0^t w(t) \cos(\omega t) dt$$

Ce qui permet d'écrire $w(t) = x_1 \cdot q(x_1, \omega) \cdot \sin \omega t + x_1 \cdot q'(x_1, \omega) \cdot \cos \omega t$

$$\boxed{q'(x_1, \omega) = \frac{\omega}{x_1 \pi} \int_0^t w(t) \cos(\omega t) dt} \quad (7-2)$$

$$\boxed{q(x_1, \omega) = \frac{\omega}{x_1 \pi} \int_0^t w(t) \sin(\omega t) dt} \quad (7-3)$$

$$\boxed{N(x_1, \omega) = B(x_1, \omega) \cdot e^{j\Phi(x_1, \omega)} = \frac{w_1}{x_1} e^{j\psi_1}} \quad (7-4)$$

Avec $\Phi(x_1, \omega) = \psi_1(x_1, \omega) = \arctan\left(\frac{q'}{q}\right)$ $B(x_1, \omega) = \sqrt{q^2 + q'^2} = \frac{w_1}{x_1}$

La manipulation de $N(x_1, \omega)$, fonction de deux variables étant assez lourde, on a l'habitude de considérer le cas particulier où N est indépendante de ω et ne dépend que de l'amplitude x_1 .

Chapitre 7 : Régulation de vitesse des moteurs à courant continu dans le cas réel

$$\boxed{N(x_1) = B(x_1) \cdot e^{j\Phi(x_1)} = \frac{W_1}{x_1} e^{j\psi_1(x_1)}} \quad (7-5)$$

Cette simplification a lieu si la caractéristique de l'organe non linéaire est indépendante du temps (non linéarité sans inertie). C'est le cas de tous les éléments-types (courbures, seuil, saturation etc....). La fonction de transfert généralisée $N(x_1)$ est alors nommée 'gain complexe équivalent' ou 'fonction d'amplitude' qui correspond à un lieu unique gradué en x_1 .

Plutôt que de tracer directement dans le plan de Nyquist ou de Black le lieu des points $N(x_1)$, on préfère tracer le lieu critique qui est le lieu des points complexes donnés par :

$$\boxed{C(x_1) = -\frac{1}{N(x_1)}} \quad (7-6)$$

Avec pour module $1/B(x_1)$ et pour argument $\pi - \Phi(x_1)$

d. Gain complexe équivalent et lieu critique de la saturation :

Nous avons une linéarité impaire et symétrie sur l'axe des abscisses

$$w_0 = 0 \text{ et } q'_1 = 0$$

Non linéarité statique

$$N(x_1, \omega) = N(x_1) = B(x_1)$$

On a l'expression du gain statique équivalent :

$$\boxed{N(x_1) = \frac{2k}{\pi x_1} \left[x_1 \arcsin \left[\frac{x_M}{x_1} \right] + x_M \sqrt{1 - \frac{x_M^2}{x_1^2}} \right]} \quad (7-7)$$

Pour $x_1 < x_M$ on $N(x_1) = k$

L'expression du lieu critique

$$\boxed{N(x_1) = -\frac{1}{N(x_1)} = -\frac{\pi x_1}{2k} \left[x_1 \arcsin \left[\frac{x_M}{x_1} \right] + x_M \sqrt{1 - \frac{x_M^2}{x_1^2}} \right]^{-1}} \quad (7-8)$$

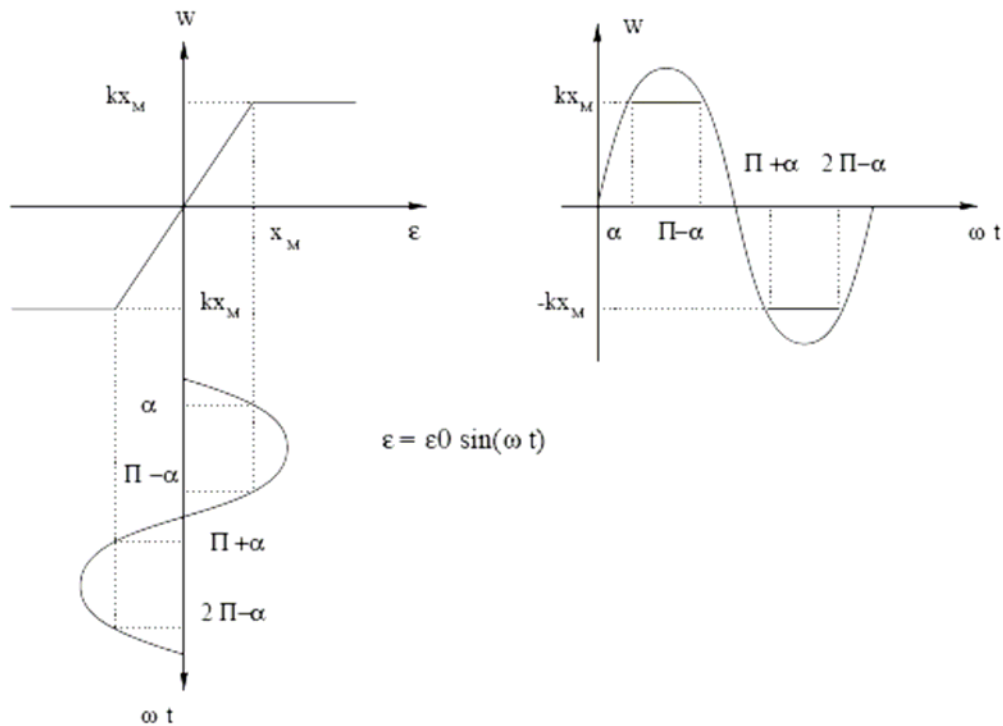


FIGURE 7.7 : Courbe de fonctionnement d'un organe saturation

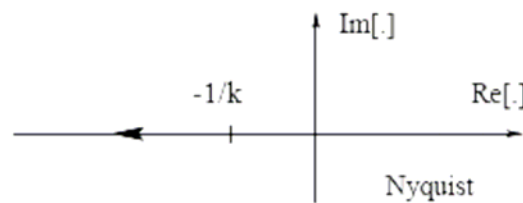


FIGURE 7.8 : Lieu critique des pôles d'un organe saturation

e. Gain complexe et lieu critique d'un plus ou moins avec hystérésis :

Nous sommes en présence de non linéarité ni paire ni impaire mais présentant une symétrie par rapport à l'axe des abscisses.

$$w_0 = 0.$$

Non linéarité statique :

$$N(x_1, \omega) = N(x_1) ; \quad B(x_1) = \sqrt{q^2 + q'^2} \quad \text{et} \quad \Phi(x_1) = \psi_1(x_1) = \arctan\left(\frac{q'}{q}\right)$$

Chapitre 7 : Régulation de vitesse des moteurs à courant continu dans le cas réel

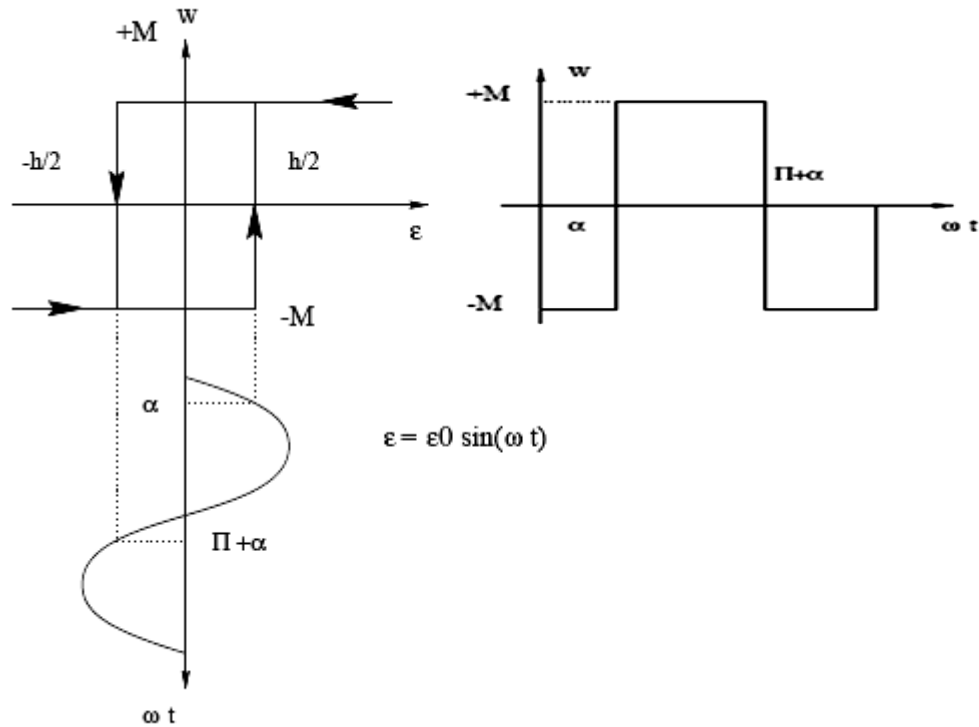


FIGURE 7.9 : Courbe de fonctionnement d'un organe tout ou rien avec hystérésis

On a après calcul l'expression du gain statique équivalent :

$$B(x_1) = \frac{4M}{\pi x_1} \text{ et } \Phi(x_1) = -\arcsin\left(\frac{h}{2x_1}\right)$$

$$\boxed{N(x_1) = \frac{4M}{\pi x_1} [\cos(\alpha) - j \sin(\alpha)]} \quad (7-9)$$

Le lieu critique est donné par :

$$\boxed{C(x_1) = -\frac{\pi x_1}{4M} e^{j\alpha}} \quad (7-10)$$

Avec l'argument $\alpha = \pi + \arcsin\left(\frac{h}{2x_1}\right)$

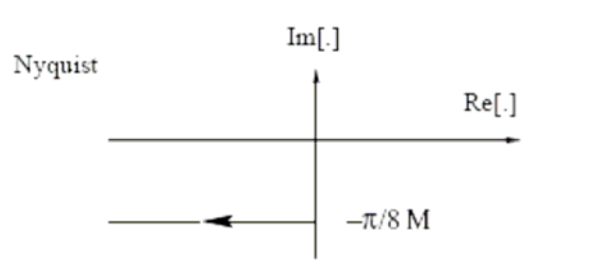


FIGURE 7.10 : Lieu critique des pôles d'un relais à hystérésis

f. Cycles limites et méthode du premier harmonique :

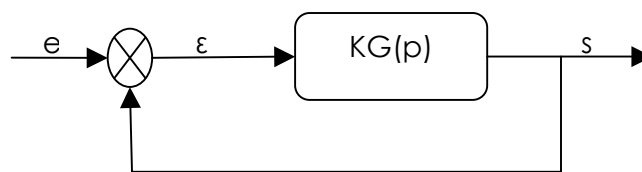
La méthode du premier harmonique peut être utilisée afin de prévoir l'existence de cycles limites dans les asservissements comportant un élément non linéaire et d'en déterminer approximativement l'amplitude et la fréquence. Le principe est fondé sur l'utilisation du critère du revers développé dans le cadre des asservissements linéaires.

1. Définitions :

Soit la fonction de transfert en boucle ouverte (FTBO) $G(p)$; la fonction de transfert en boucle fermée (FTBF) s'écrit :

$$H(p) = \frac{KG(p)}{1 + KG(p)}$$

- ❖ Stabilité : Un asservissement est stable si le polynôme caractéristique du système asservi (dénominateur de la FTBF) a des racines à parties réelles négatives. Et selon le critère du revers si le lieu de Nyquist de $G(j\omega)$ (FTBO) parcouru dans le sens des ω croissants laisse à gauche le point critique $-1/K$ le système est stable.



- ❖ Cycle limite : les systèmes non linéaires peuvent être le siège d'oscillations d'amplitude et de fréquence fixées, indépendantes des conditions initiales et sans excitation extérieures, dénommées (auto-oscillations) cycles limites.

2. Conditions d'auto-oscillations et de stabilité :

Nous supposons que le système asservi de la figure (7.6) soit le siège d'une oscillation d'amplitude ε_1 et de pulsation de ω_0 avec $e \equiv 0$ et $\varepsilon = \varepsilon_1 \sin \omega_0 t$.

Chapitre 7 : Régulation de vitesse des moteurs à courant continu dans le cas réel

Nous avons alors $\varepsilon(t) = -s(t)$; $W = N(\varepsilon_1) \cdot \varepsilon$ et $S = L(j\omega_0) \cdot W$ on en tire $S[1 + L(j\omega_0)N(\varepsilon_1)] = 0$ comme $S \neq 0$ seule l'expression $[1 + L(j\omega_0)N(\varepsilon_1)]$ est nulle; le cycle limite est alors caractérisé par son amplitude ε_1 et sa pulsation ω_0 qui doivent vérifier la condition d'existence suivante :

$$\boxed{L(j\omega_0) = -\frac{1}{N(\varepsilon_1)}} \quad (7-11)$$

Quand ε_1 varie $-\frac{1}{N(\varepsilon_1)}$ parcourt le lieu critique, donc à l'amplitude ε_1 , la stabilité du système varie en fonction de la position de ce point par rapport au lieu $L(j\omega)$. Le lieu critique se trouve ainsi partagé en région d'amplitude de stabilité et en région d'amplitude d'instabilité.

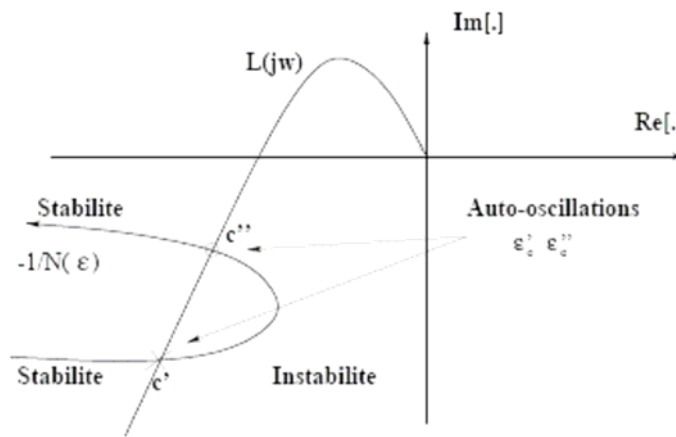


FIGURE 7.11 : Auto-oscillation dans le plan de Nyquist

Pour $\varepsilon_1 < \varepsilon'_c$ il y a stabilité les auto-oscillations vont décroître.

Pour $\varepsilon'_c < \varepsilon_1 < \varepsilon''_c$ instabilité les auto-oscillations vont croître $\varepsilon_1 \uparrow \varepsilon''_c$

Pour $\varepsilon''_c < \varepsilon_1$ il y a stabilité les auto-oscillations vont décroître. $\varepsilon_1 \downarrow \varepsilon''_c$

g. Dimensionnement des correcteurs :

1. Dimensionnement du correcteur de vitesse :

Le dimensionnement du contrôleur de vitesse se fera par la méthode de Ziegler Nichols temporelle, elle s'effectuera par le traçage de la réponse indicielle du moteur, en boucle ouverte, puis par la mesure des paramètres du régulateur sur la figure (7.13).

Chapitre 7 : Régulation de vitesse des moteurs à courant continu dans le cas réel

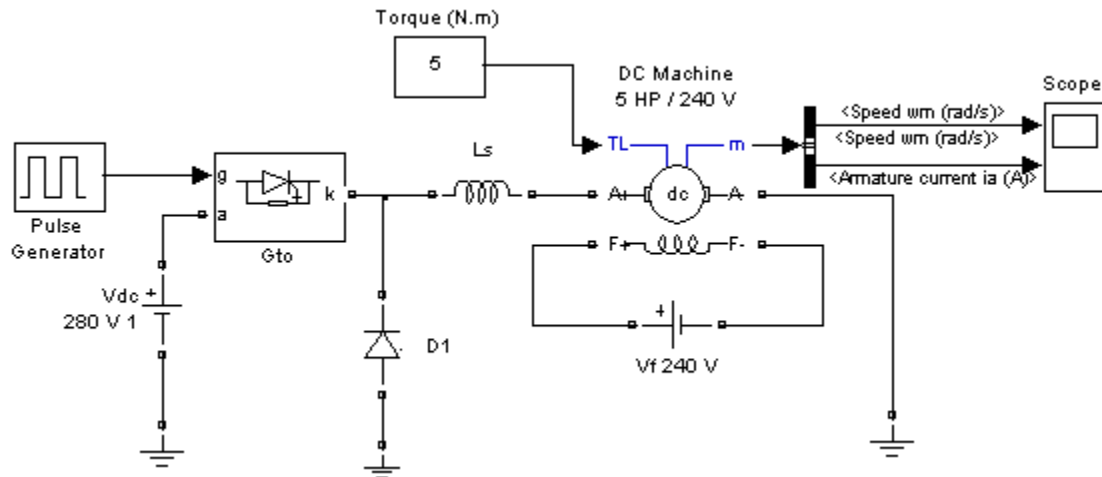


FIGURE 7.12 : Montage en boucle ouverte du moteur

Le schéma de montage adopté pour l'obtention de la réponse indicielle en boucle ouverte est donnée par la figure (7.12) ci-dessus.

Nous avons mis comme signal de commande de gâchette une impulsion unité et la consigne indicielle du moteur est donnée par la tension d'alimentation de l'induit soit 280[V] ; la réponse en boucle ouverte du moteur est la suivante :

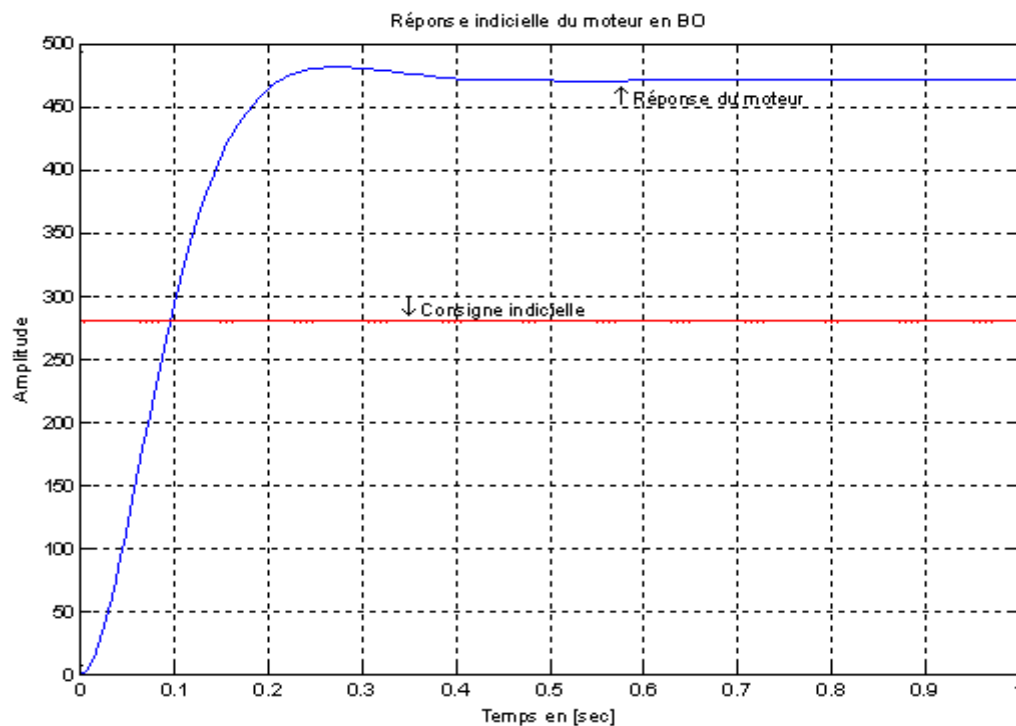


FIGURE 7.13 : Réponse en boucle ouverte du moteur

En utilisant la méthode de Ziegler Nichols temporelle on obtient les caractéristiques du régulateur ; la mesure directe sur la figure (7.14) nous donne les valeurs de :

$$L = 0.03 \text{ [sec]} \text{ et celle de la pente } a = 18.75$$

Chapitre 7 : Régulation de vitesse des moteurs à courant continu dans le cas réel

D'après ces valeurs et d'après le tableau (4.2) nous obtenons les valeurs des gains proportionnel et intégral

$$K_p = 1.6 \text{ et } K_i = 16$$

La limitation du courant d'induit à la valeur de 30[A] est effectuée par l'organe non linéaire à saturation ; cette saturation du courant d'induit permet de protéger les organes de la machine et d'empêcher les pertes par hystérésis dues à la saturation du matériau ferromagnétique.

Le correcteur de vitesse donne donc comme expression :

$$C(p) = \frac{1.6p + 16}{p} \quad (7-12)$$

La mesure des paramètres de Ziegler Nichols s'est faite sur la figure qui suit :

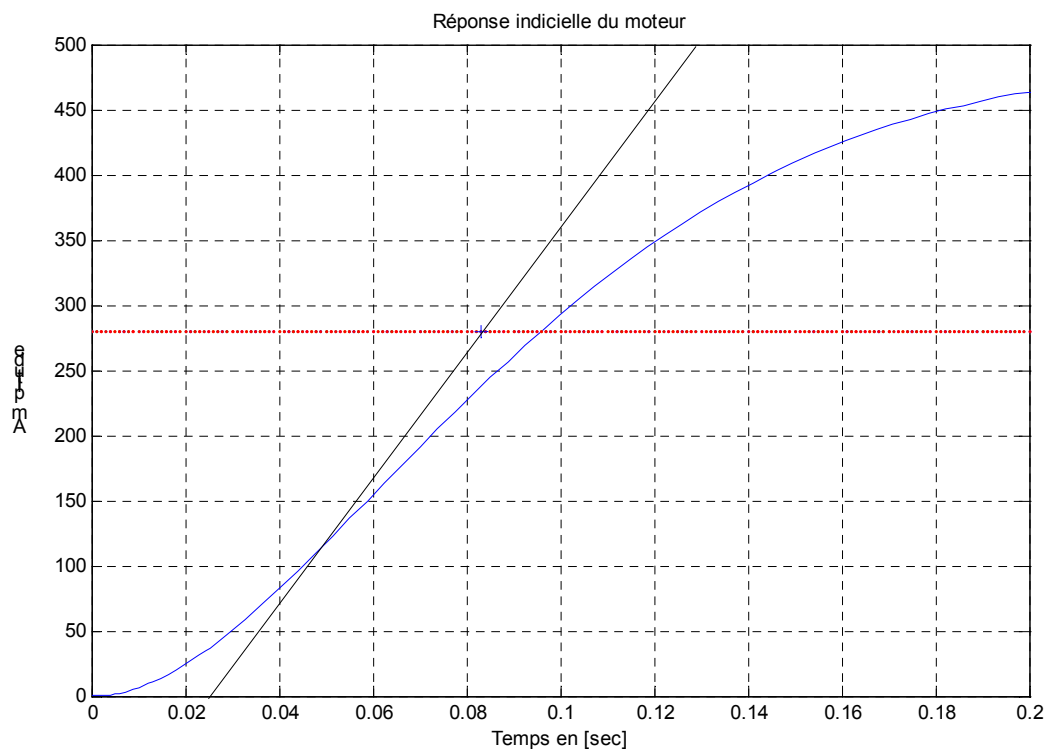


FIGURE 7.14 : Détermination des caractéristiques de Ziegler Nichols

2. Dimensionnement du correcteur de courant :

Le correcteur de courant est un organe purement non linéaire, un relais avec hystérésis plus précisément.

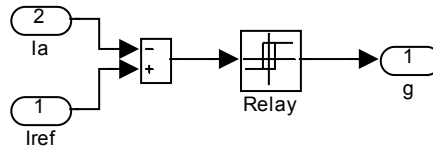


FIGURE 7.15 : Schéma du contrôleur de courant

Elle permet la génération d'un signal modulé en largeur d'impulsion (PWM) et est destiné à la commande de la gâchette du thyristor. Elle délivre un signal binaire qui est régit par le comportement du relais à hystérésis ; la valeur de l'hystérésis (h) est fixée à 2 ce qui fait que le seuil de l'hystérésis est fixé à $(-h/2, h/2)$ soit $(-1, 1)$; la sortie du relais lorsqu'il est au niveau haut est 1, dans le cas contraire le relais délivre un signal nul. Lorsque l'entrée dépasse le seuil limite maximal 1 le correcteur envoie une indice 1 qui commande la fermeture du thyristor et provoque ainsi l'alimentation de l'induit du moteur sous une tension U , quand l'erreur est inférieure au seuil limite minimal à '-1' le relais envoie un signal nul qui entrainera l'ouverture du circuit d'alimentation du moteur.

3. Présentation de l'alimentation :

Le schéma suivant montre une représentation simplifiée de l'alimentation du moteur à courant continu ; le moteur étant remplacé par sa résistance et inductance d'induit (R_a, L_a) en série avec sa f.é.m. E . Une diode à roue libre a été insérée dans le circuit pour assurer la continuité du courant I_a .

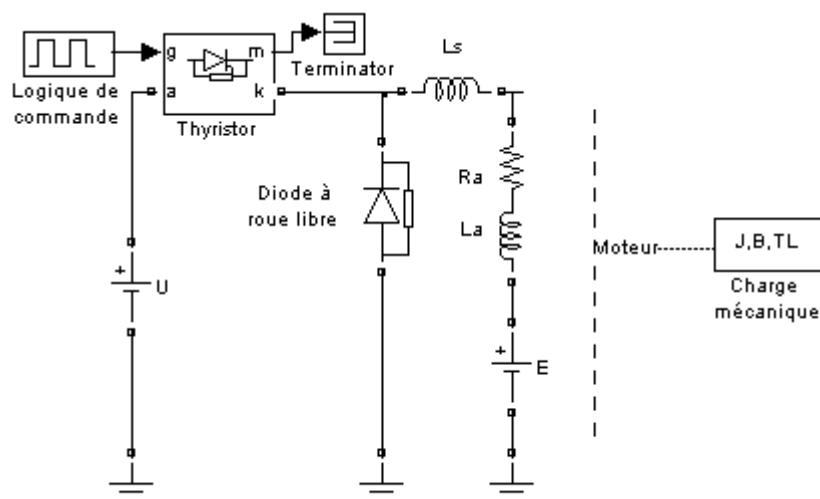


FIGURE 7.16 : Schéma simplifié de l'alimentation du moteur

L'alimentation du moteur est un pont à thyristor GTO, le thyristor est commandé par un signal PWM qui est simplifié par le bloc 'logique de commande' pour contrôler la tension en valeur moyenne du moteur. On a mis en place une inductance (L_s) en série avec l'alimentation pour le lissage du courant d'induit. Les caractéristiques de L_s , du GTO et de la diode à roue libre sont données en annexe (Annexe1).

Chapitre 7 : Régulation de vitesse des moteurs à courant continu dans le cas réel

L'expression de la tension induite en valeur moyenne est fonction du rapport cyclique α :

$$\langle U_a \rangle = \alpha U$$

Le rapport cyclique est défini comme étant le rapport entre le temps de conduction et la période de hachage.

La valeur du courant induit moyen est donnée par :

$$\boxed{\langle I_a \rangle = \frac{\langle U_a \rangle - E}{R_a}} \quad (7-13)$$

On obtient le pic de courant par la formule :

$$\boxed{\Delta i = \frac{U(1 - e^{-\alpha r} + e^{-r} - e^{(1-\alpha)r})}{R_a(1 - e^{-r})}} \quad (7-14)$$

Avec r le rapport entre la période de hachage et la constante de temps du moteur. Théoriquement les formes d'ondes du courant et de la tension d'induit dans le cas d'un hacheur sont données par la suite :

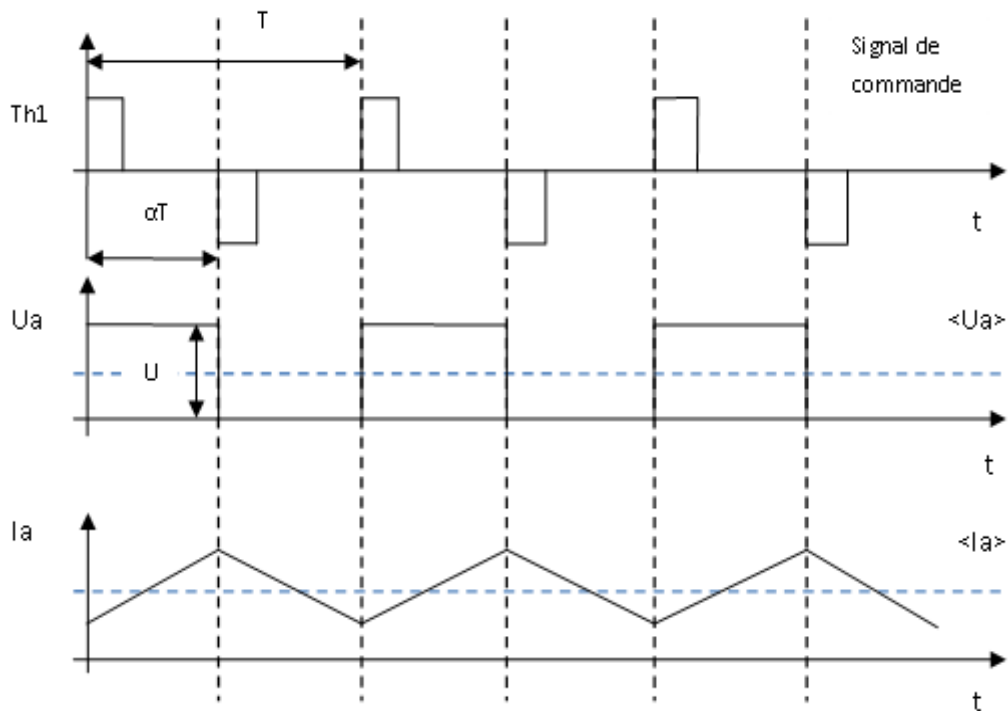


FIGURE 7.17 : Courbes théoriques de U_a et de I_a en fonction du rapport cyclique

III. Régulation de vitesse dans le cas réel :

Le montage de la figure (7.1) est celui utilisé pour la régulation de vitesse d'un moteur à courant continu dans le cas réel. On a fixé la vitesse de rotation du moteur à environ de 120 [rad/s]. Avec la limitation du courant d'induit à 30[A] et un couple résistant constant à la valeur de 5[N.m] nous avons les courbes de fonctionnement du moteur suivant :

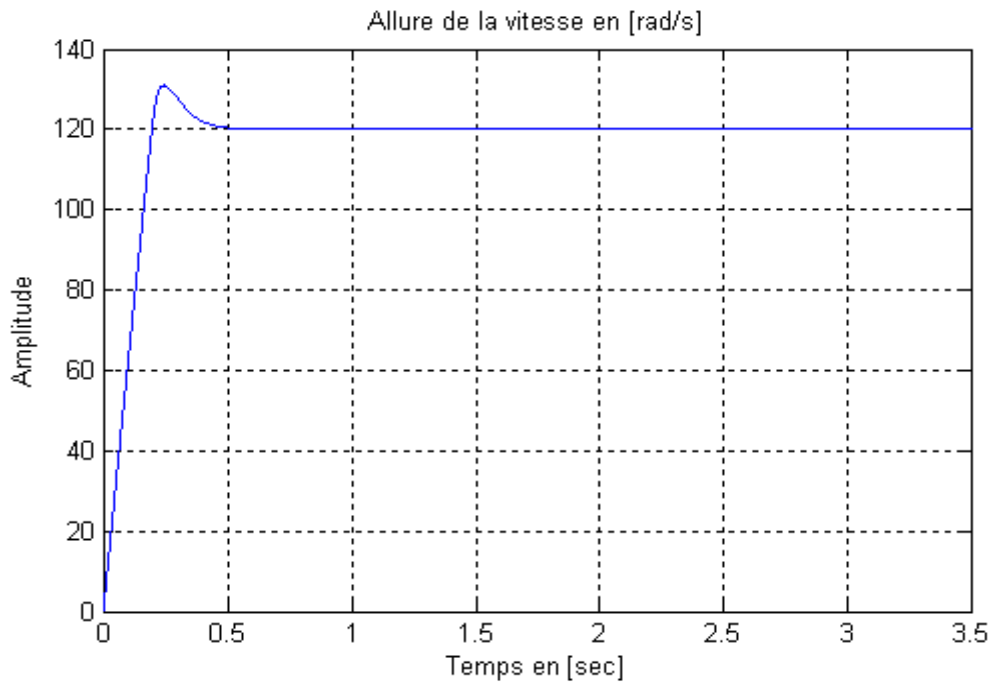


FIGURE 7.18 : Réponse indicielle du moteur à charge fixe

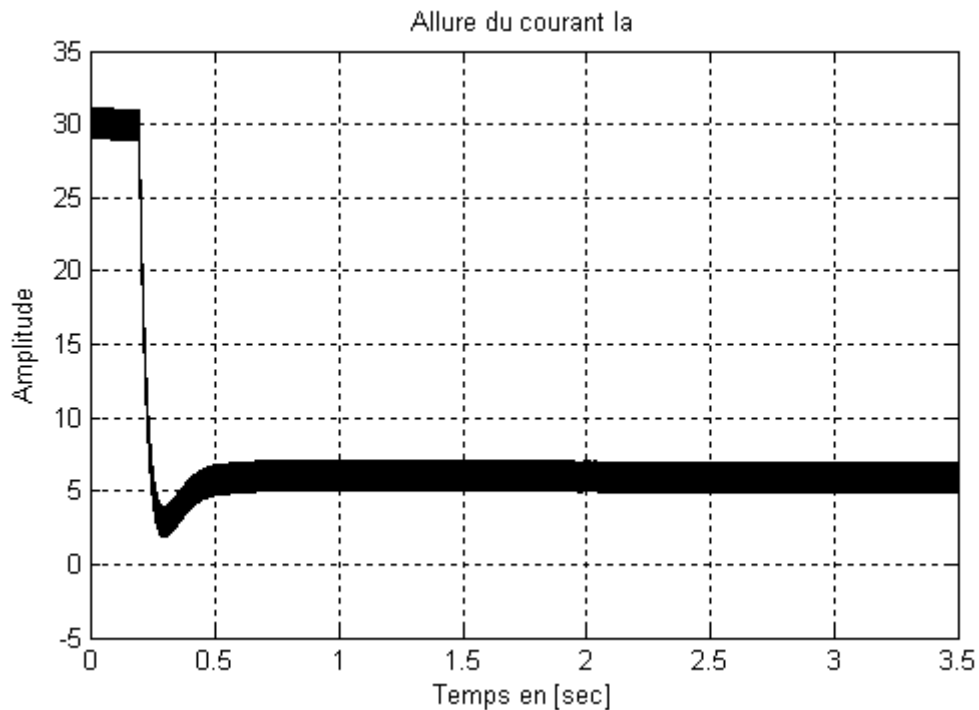


FIGURE 7.19 : Allure du courant d'induit I_a pour un couple résistant constant

Chapitre 7 : Régulation de vitesse des moteurs à courant continu dans le cas réel

Nous obtenons un premier dépassement de l'ordre de 8% de la valeur de la consigne et nous constatons qu'en régime permanent le courant d'induit est à la valeur de 5[A].

Nous pouvons également étudier la performance dynamique du système en appliquant une variation du couple de charge ; nous obtenons les courbes de démarrages suivantes :

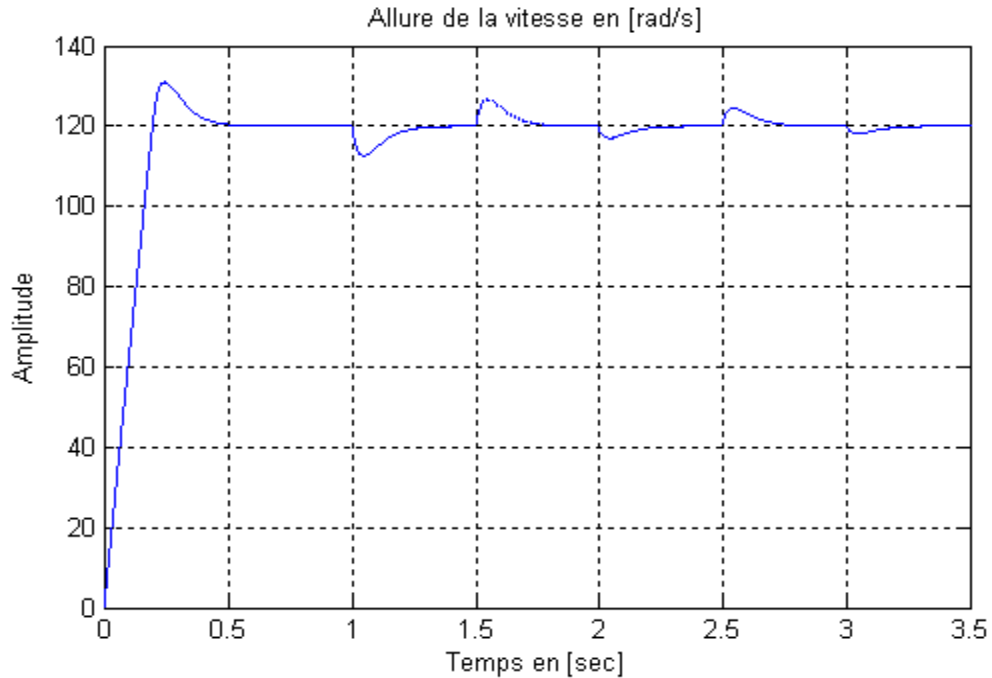


FIGURE 7.20 : Réponse indicielle du moteur à charge variable

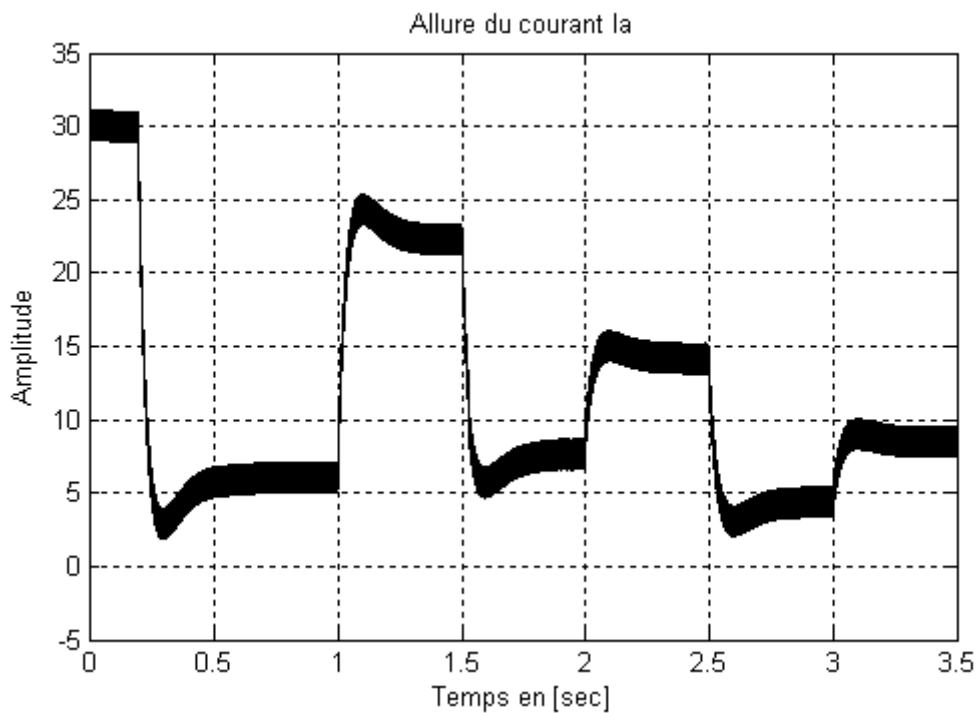


FIGURE 7.21 : Allure du courant d'induit à charge variable

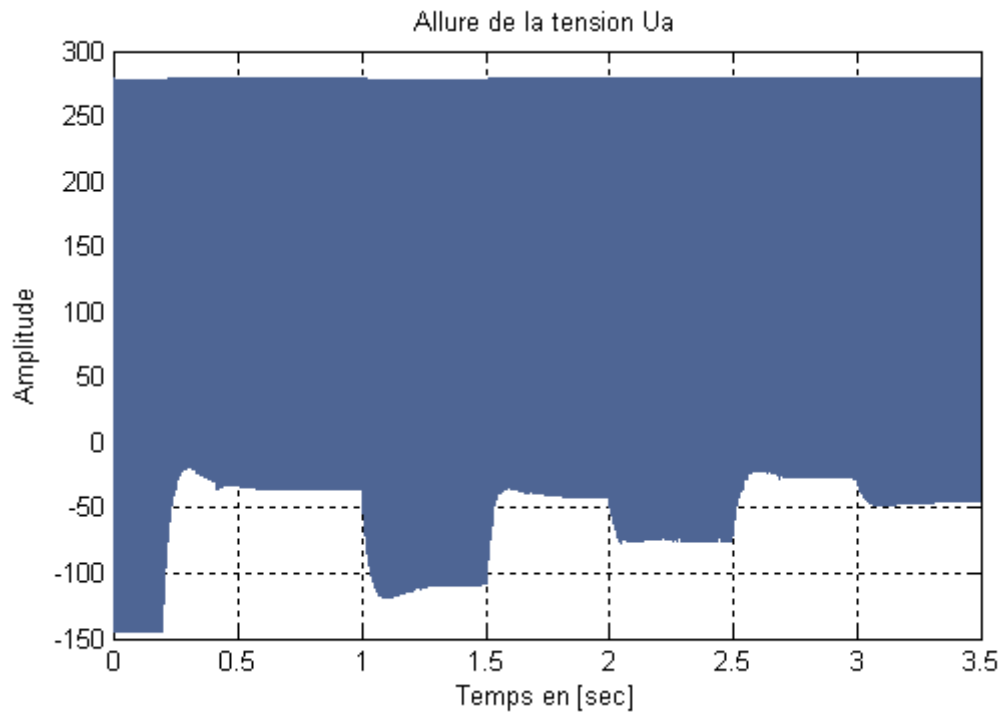


FIGURE 7.22 : Allure de la tension d'induit du moteur avec variation de charge

Sans la limitation du courant d'induit ; le pic de courant au démarrage peut atteindre une valeur quatre à cinq fois supérieure à la valeur nominale du courant d'induit soit 160[A].

Pour vérifier l'efficacité du correcteur de courant nous allons étudier le moteur en régime nominal en lui appliquant le couple de charge de la figure (7.23) qui suit :

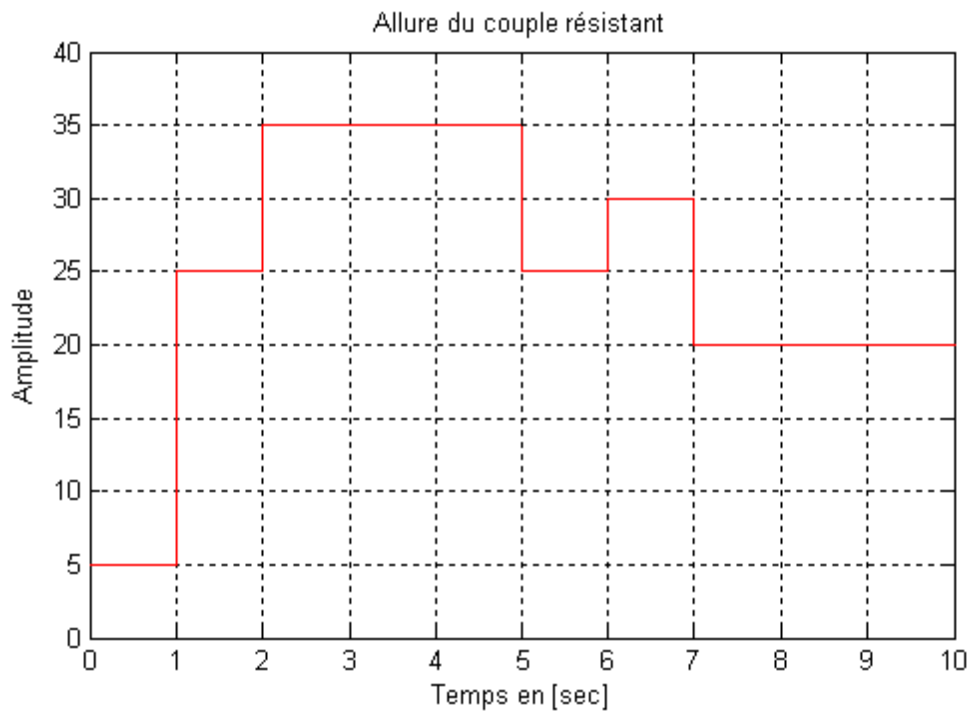


FIGURE 7.23 : Allure du couple résistant en régime nominal

Chapitre 7 : Régulation de vitesse des moteurs à courant continu dans le cas réel

Nous obtenons les courbes de démarrage en vitesse et en courant suivants :

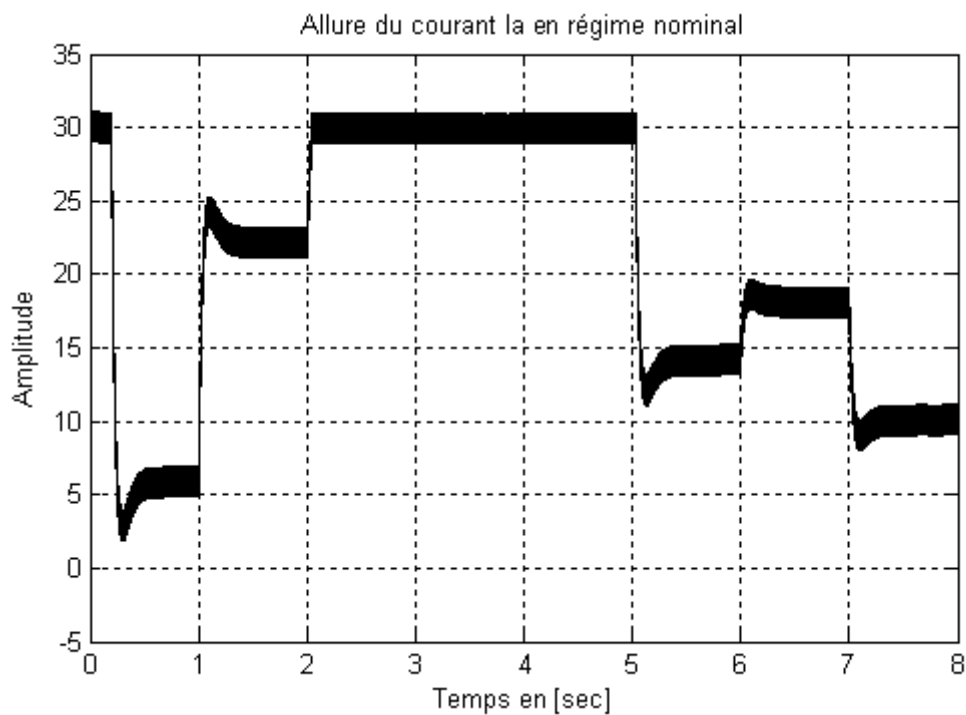


FIGURE 7.24 : Allure du courant d'induit du moteur en régime de fonctionnement nominal

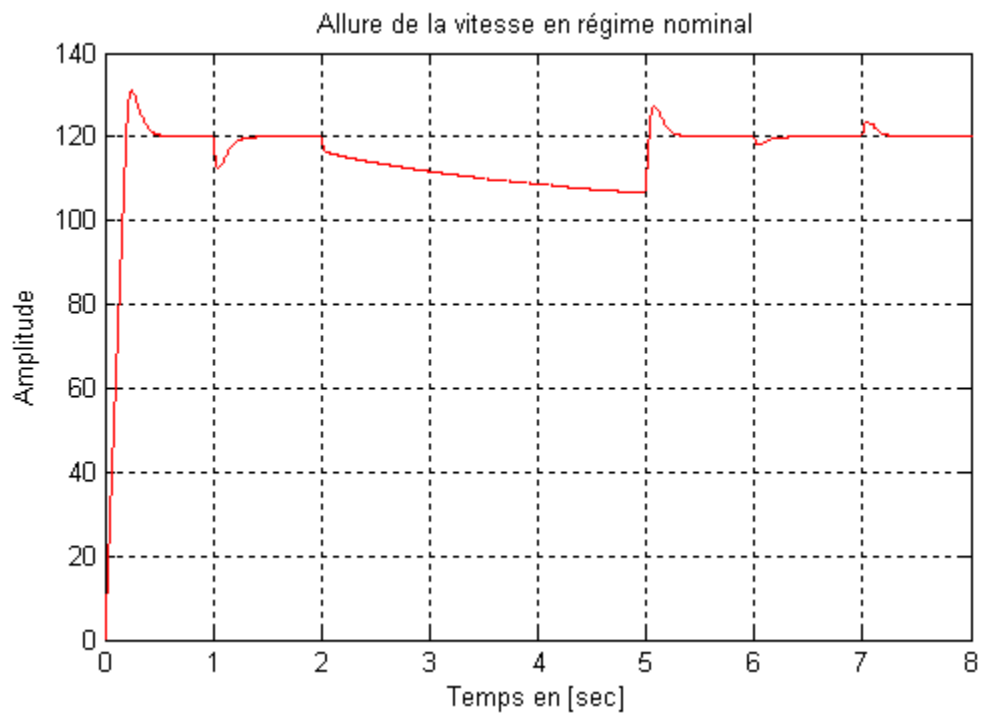


FIGURE 7.25 : Allure de la vitesse du moteur en régime nominal

Chapitre 8

Modélisation des contrôleurs de courant et de vitesse avec STATEFLOW

I. Introduction :

STATEFLOW est un outil de programmation graphique puissant, elle permet de modéliser différents types de contrôleur. Dans notre cas nous aurons à modéliser un correcteur de vitesse et un contrôleur de courant ; le correcteur de vitesse sera modélisé par un seul diagramme et l'action proportionnelle intégrale sera associée à ce diagramme. Pour le correcteur de courant nous utiliserons une séquence de plusieurs états associés à des fonctions propres au diagramme.

Nous tenons à remarquer que la régulation de vitesse comporte deux boucles en cascade. La boucle interne de courant qui est une boucle rapide, elle permet de maîtriser le couple, tandis que la boucle externe de vitesse possède une dynamique plus lente. A cause de la dynamique différente des deux boucles, il est utile de choisir deux fréquences d'échantillonnage différentes. Ces fréquences d'échantillonnages sont modélisées par les triggers des deux diagrammes STATEFLOW.

II. Modélisation du contrôleur de vitesse :

Le contrôleur de vitesse est un régulateur proportionnel intégral (PI), il est modélisé de la même façon que le correcteur de vitesse du chapitre 6 .

Nous avons créé une fonction qui calcule l'erreur entre la valeur de la vitesse de consigne et celle de la vitesse de sortie du moteur ; cette grandeur sera traitée par le correcteur pour délivrer le courant de référence de la boucle de courant. Le diagramme est activé à une fréquence de 100[Hz] par le trigger.

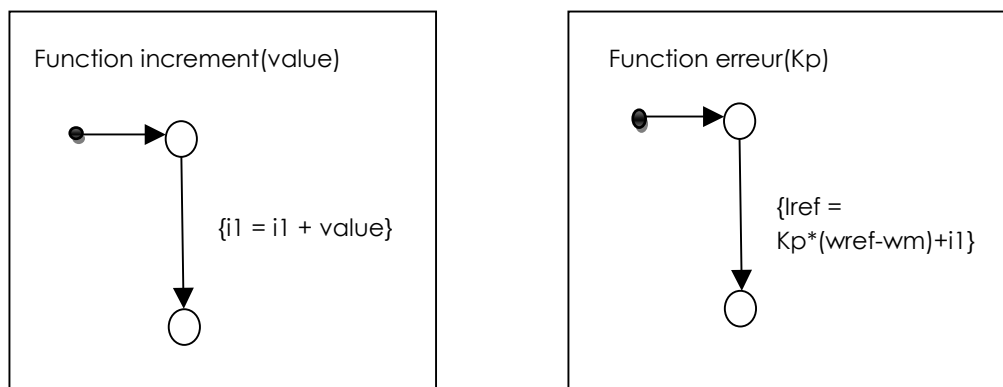


FIGURE 8.1 : Fonctions graphiques STATEFLOW utilisés dans le modèle du contrôleur de vitesse

Chapitre 8 : Modélisation des contrôleurs de courant et de vitesse avec STATEFLOW

La liste des fonctions est représentée par la figure (8.1); la fonction graphique 'increment (value)' effectue une incrémentation et/ou une décrémentation d'une valeur 'value' ceci en fonction du signe de cette valeur; tandis que la fonction 'erreur(Kp)' multiplie la différence entre la vitesse de référence et celle de la vitesse réelle par une constante Kp puis ajoute la valeur de l'incrément à chaque échantillon pour la restituer finalement à la sortie 'Iref'.

On constate par la figure (8.2) qui suit que la décomposition des états est une décomposition parallèle de fait à ce que les deux actions fonctionnent simultanément.

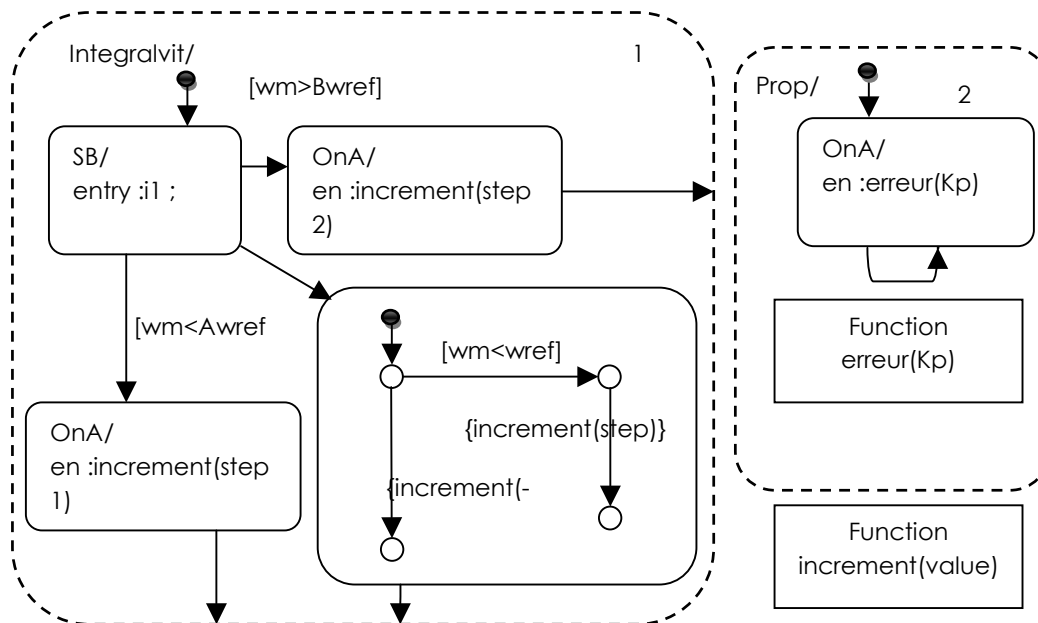


FIGURE 8.2 : Modèle du contrôleur de vitesse

III. Modélisation de la saturation :

a. Explication du diagramme :

Le modèle de la saturation a été construit à partir de la courbe de fonctionnement de la figure (7.7) du bloc saturation pour la limitation du courant d'induit à la valeur de 30[A]. Le diagramme a une fréquence de 500[Hz]. Celle-ci a été choisie pour être supérieure à la fréquence du contrôleur de vitesse pour pouvoir traiter tous les échantillons de la grandeur de sortie du correcteur PI.

Nous avons pour cela mis en œuvre une fonction qui donne une valeur de 30[A] (respectivement -30[A]) à la sortie du diagramme si la valeur de la commande débitée par le contrôleur de vitesse est supérieure (resp. inférieure) à cette valeur, et laissant passer la valeur de l'entrée du diagramme dans le cas contraire.

b. Mise en œuvre de l'algorithme :

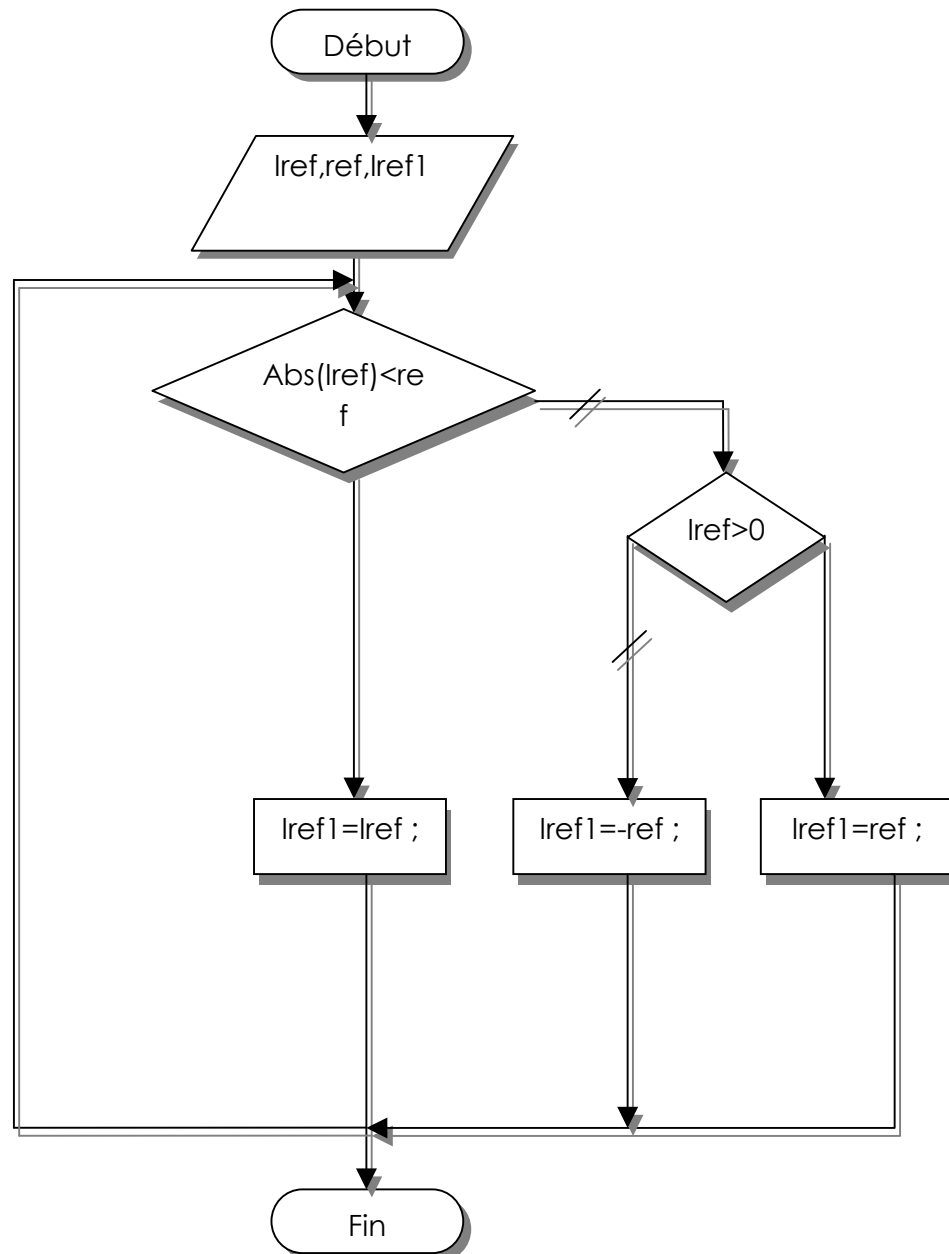


FIGURE 8.3 : Organigramme du modèle saturation

Cet organigramme sature la sortie du correcteur de vitesse à la valeur de 30[A] (ou -30[A]) lorsque le courant au démarrage commence à atteindre une valeur supérieure à celle admise par la machine; elle sert également à saturer le courant lors des variations brusques de la consigne.

c. Modèle STATEFLOW du bloc saturation :

Le bloc saturation a été représenté par le diagramme suivant :

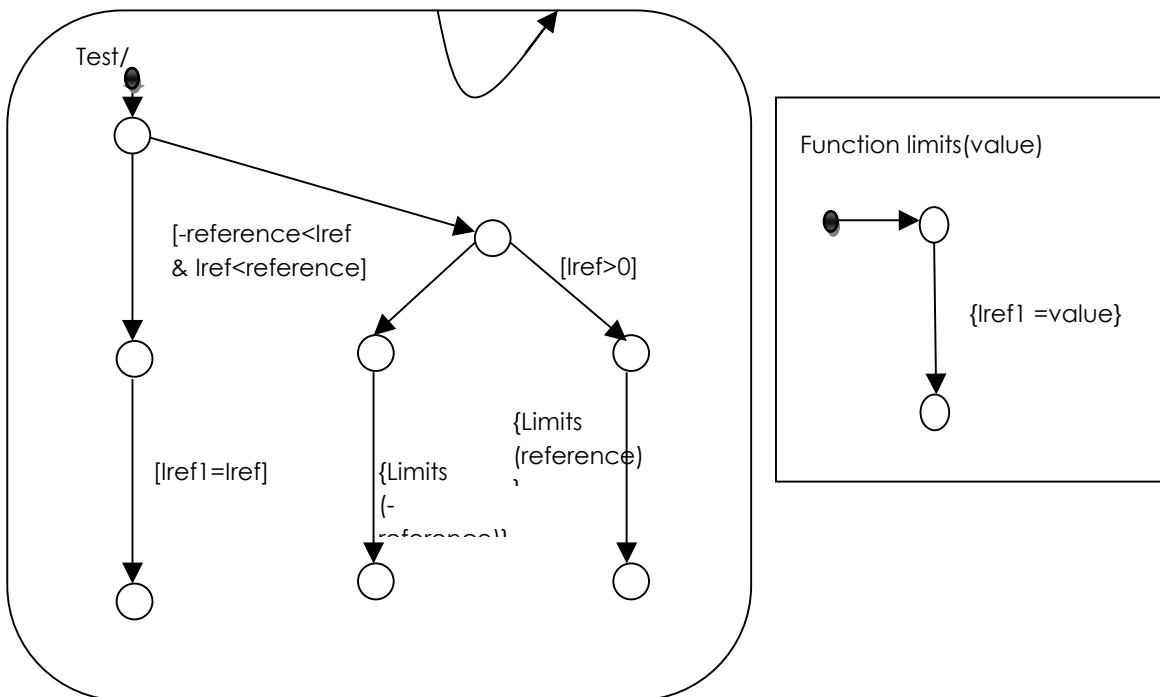


FIGURE 8.4 : Diagramme STATEFLOW pour la saturation

Une fonction de limitation ‘limits (value)’ a été créée pour pouvoir être utilisé dans l’état ‘Test/’ ; elle servira de valeur de sortie du diagramme lorsque le courant sera trop élevé.

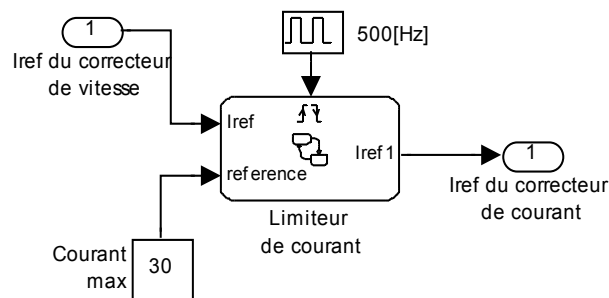


FIGURE 8.5 : Modèle du limiteur de courant

Nous allons maintenant entrer dans la modélisation du contrôleur de courant ; elle sera le modèle d’un relais à hystérésis.

IV. Modélisation du contrôleur de courant :

a. Explication du diagramme :

Le contrôleur de courant est un modèle de relais à hystérésis, c'est un bloc non linéaire dont on a essayé de reconstituer le comportement. Ce modèle restitue une valeur binaire en fonction de la grandeur d'entrée du diagramme c'est-à-dire l'erreur entre la valeur du courant de référence et celle débitée par le moteur à courant continu.

La variation de la sortie du contrôleur de courant est fonction de l'entrée du contrôleur. Elle suit généralement le comportement de l'organe non linéaire de la figure (7.9). Une fonction mémoire a été introduite pour pouvoir comparer la valeur de l'échantillon 'n+1' du courant I_a avec celle de l'échantillon 'n' de ce courant d'induit I_a . Ceci a été fait pour vérifier si l'erreur est croissante ou décroissante ; c'est-à-dire si le courant d'induit I_a s'éloigne de la référence ou s'en approche.

L'organigramme de la figure (8.8) représente le fonctionnement du contrôleur, il calcule la différence entre la valeur de référence délivré par le contrôleur de vitesse et celle délivrée par la machine, puis vérifie dans quelle zone se trouve l'entrée du diagramme, et enfin en déduit la sortie 'g' pour la commande de la gâchette.

b. Présentation du contrôleur de courant :

Le contrôleur est représenté par le schéma bloc de la figure (8.6) ; nous pouvons voir comment est représentée le diagramme du contrôleur de courant ; une fonction graphique '**turn_reg (mode)**' a été créée pour définir la sortie du régulateur en fonction de la valeur des entrées ' I_{ref} ' et ' I_a '. Cette fonction a été employée dans le diagramme d'état 'Cmd' pour restituer une valeur de sortie 'regulate' en fonction des états actifs.

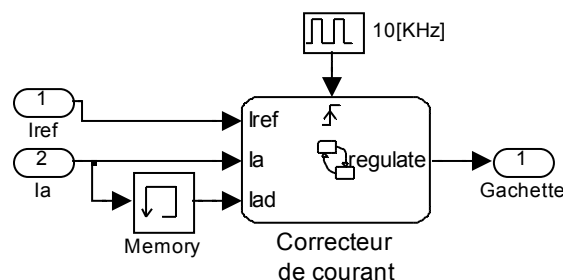


FIGURE 8.6 : Bloc STATEFLOW du contrôleur de courant

La fonction et l'état utilisé pour la modélisation du contrôleur de courant est représentée par le diagramme de la figure (8.7).

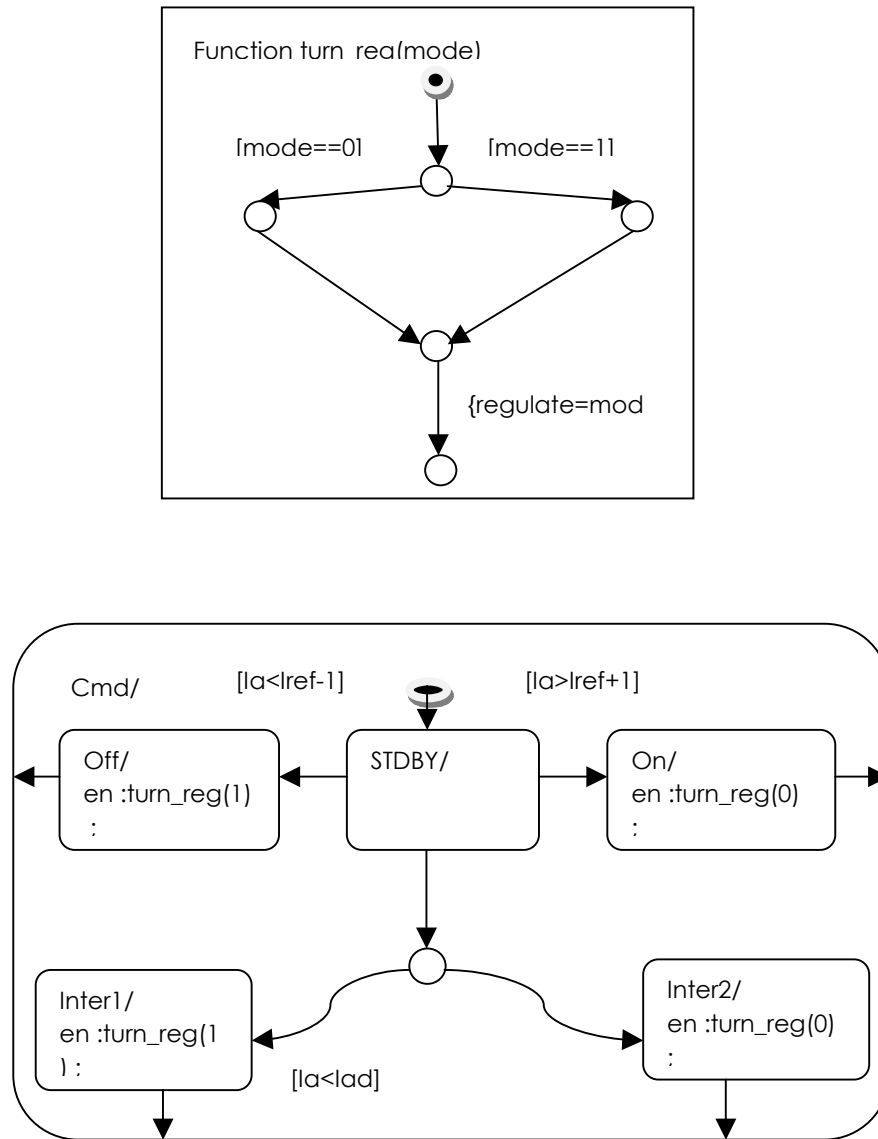


FIGURE 8.7 : Diagramme STATEFLOW et fonction utilisée dans le contrôleur de courant

c. Mise en œuvre de l'algorithme :

L'organigramme de la figure (8.8) modélise le comportement d'un relais à hystérésis, avec comme valeur de l'hystérésis h égal à 2, cette valeur est la limite au bout duquel le modèle change la valeur de la sortie.

Une première fonction calcule la valeur de l'erreur, différence entre la valeur du courant I_{ref} du contrôleur de vitesse et le courant I_a du moteur. Puis viens le premier test, si la valeur de l'erreur se trouve dans la largeur de l'hystérésis, le programme teste si la valeur de l'erreur est croissante ou non, si oui ($I_a > I_{ad}$) le contrôleur envoie la valeur 0 à la sortie, ce qui provoque qu'un courant nul arrive à la gâchette du thyristor.

Le thyristor ne s'amorcera pas et le circuit d'armature ne sera pas alimenté, ce qui se traduit par une diminution du couple électromagnétique donc une chute de la valeur du courant. Dans le cas où l'erreur diminue le contrôleur sortira la valeur 1 à la sortie ce qui est synonyme de la fermeture du circuit d'induit d'où l'alimentation du moteur.

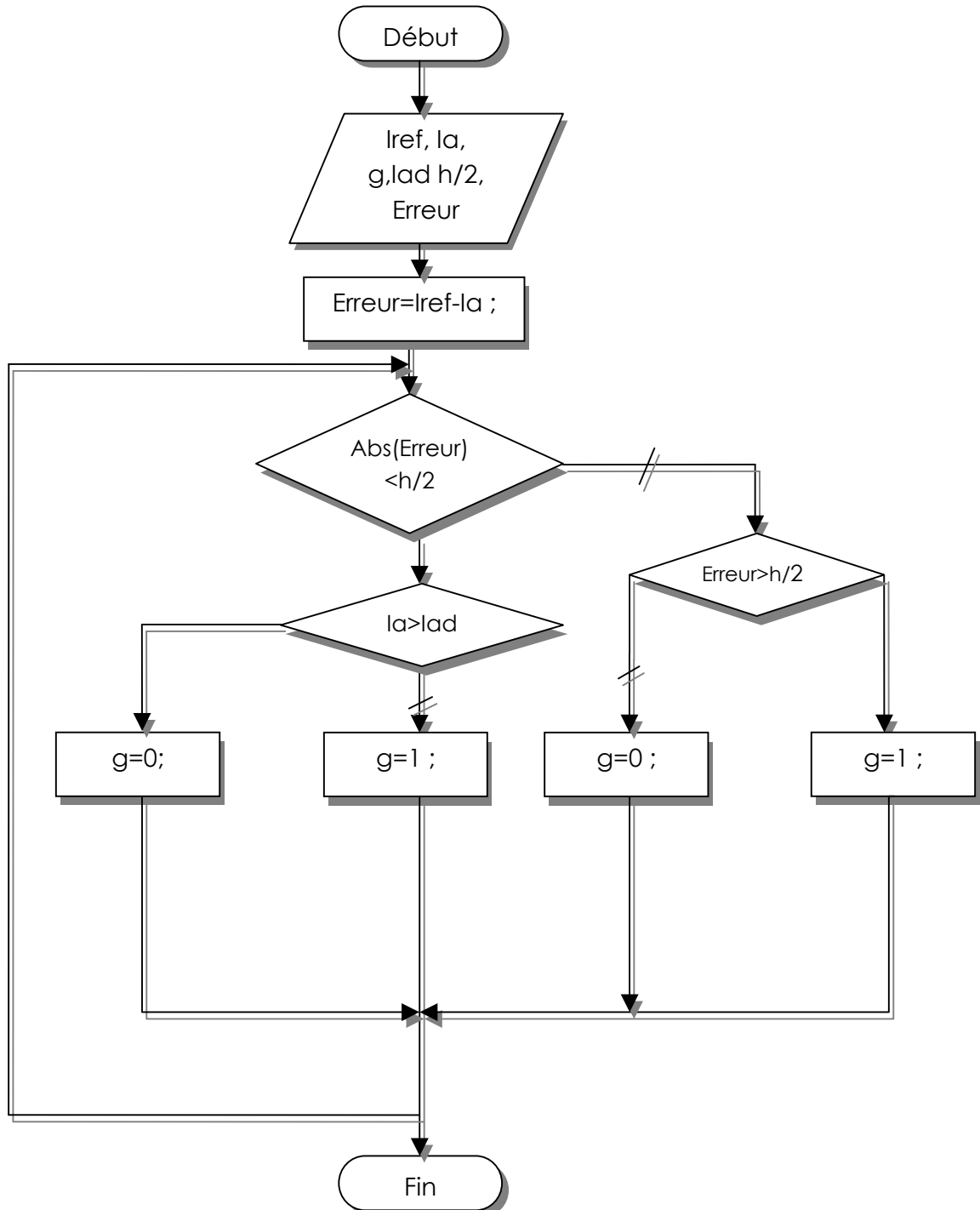


FIGURE 8.8 : Organigramme du modèle du relais à hystérésis

V. Présentation de la chaine de régulation et simulations:

Le couple résistant utilisé pour cette simulation est le même que celui de la figure (7.3). La chaine de régulation possède une disposition en cascade suivant la figure suivante :

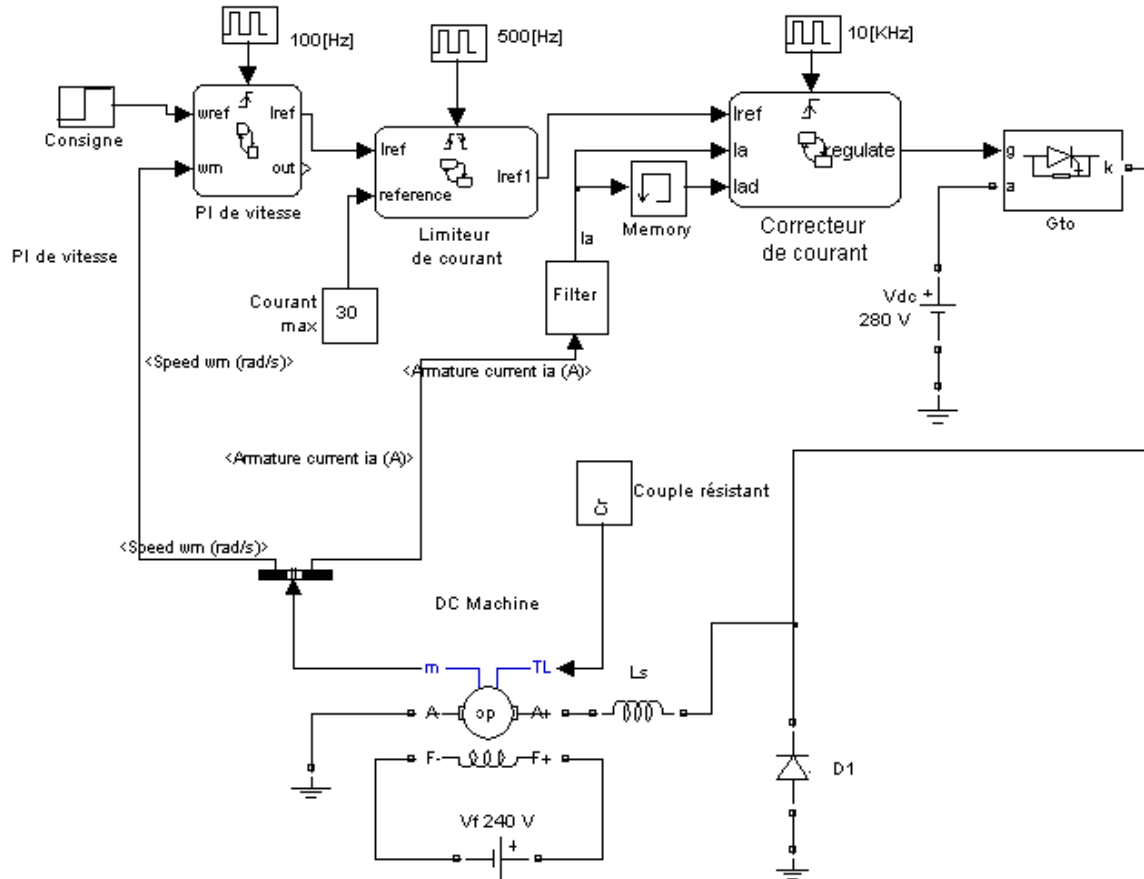


FIGURE 8.9 : Chaine de régulation du moteur avec son alimentation

On constate que le correcteur PI de vitesse délivre la référence de courant qui sera ensuite limité par le modèle de la saturation 'limiteur de courant', la valeur de sortie 'Iref1' du limiteur de courant sera pris comme courant de référence du contrôleur de courant ; ce dernier délivre un signal binaire qui sera la commande de la gâchette du thyristor, lorsque le contrôleur de courant délivre un signal niveau haut il commande la fermeture du circuit d'alimentation de l'induit et crée ainsi l'augmentation du couple du moteur, dans l'autre cas il ouvre le circuit d'alimentation du moteur et diminue le couple moteur. Une succession de 1 et de 0 provenant de la sortie du contrôleur de courant sera à l'origine d'une succession de montée et de descente de la valeur du couple du moteur mais l'inertie du moteur élimine ces oscillations, pour que la vitesse soit uniformément croissante la fréquence d'apparition des 1 doit être supérieure à celle des 0, et inversement

Nous obtenons les courbes de démarrage suivantes de la machine :

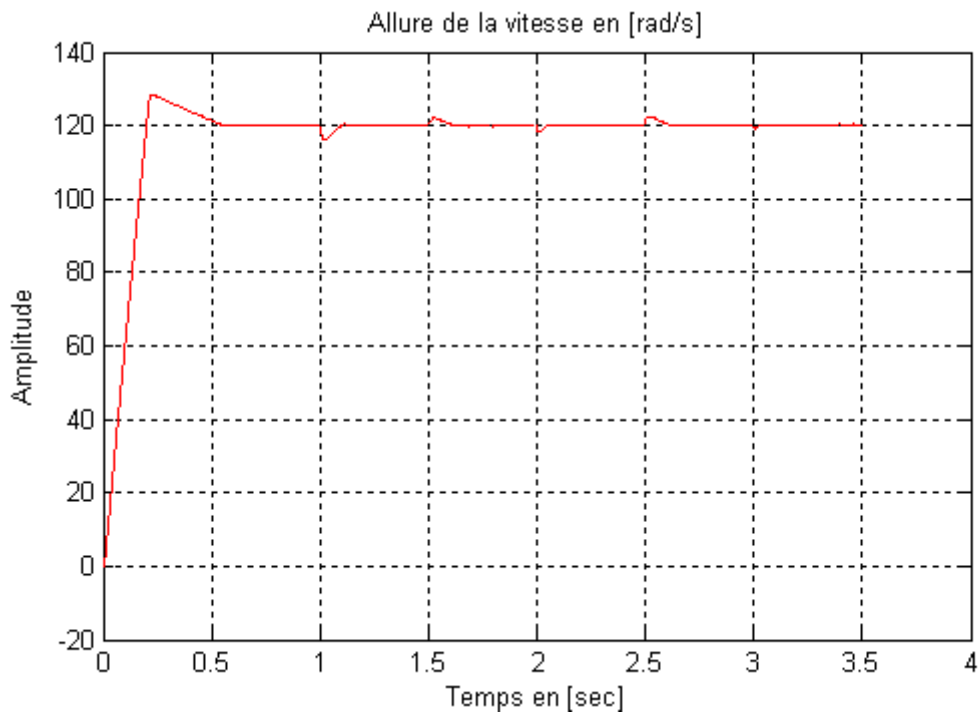


FIGURE 8.10 : Réponse indicielle du moteur avec variation de charge

Nous constatons que les effets du couple résistant sont amortis, la vitesse revient à la consigne après un certain temps, l'amplitude du premier dépassement est de 8% de la valeur finale c'est-à-dire de la consigne qui est à 120[rad/s].

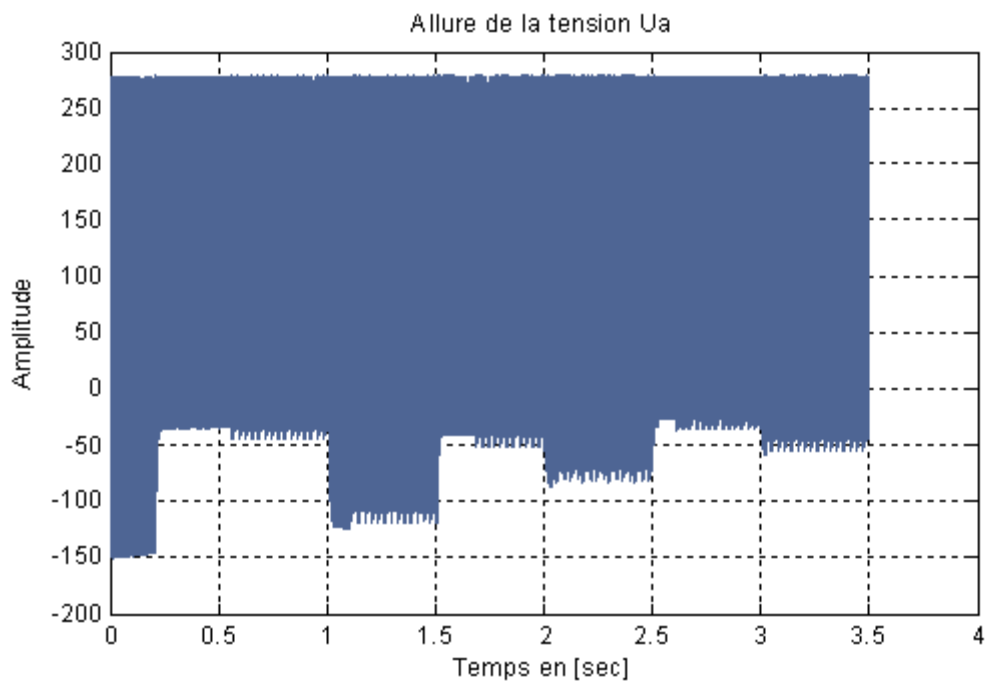


FIGURE 8.11 : Allure de la tension d'induit du moteur avec couple résistant variable

On constate d'après l'allure de la tension d'induit de la figure (8.11) que la variation maximale de l'amplitude se trouve à la valeur de 280[V], c'est la valeur de la tension U d'alimentation de l'induit, elle balaye une surface d'après la figure (8.11) ceci est du au fait que le courant de commande de la gâchette du thyristor ouvre puis ferme le circuit d'alimentation d'induit, lorsque le circuit est fermé on a la tension de 280[V], dans le cas inverse la tension est fonction de la vitesse de sortie du moteur.

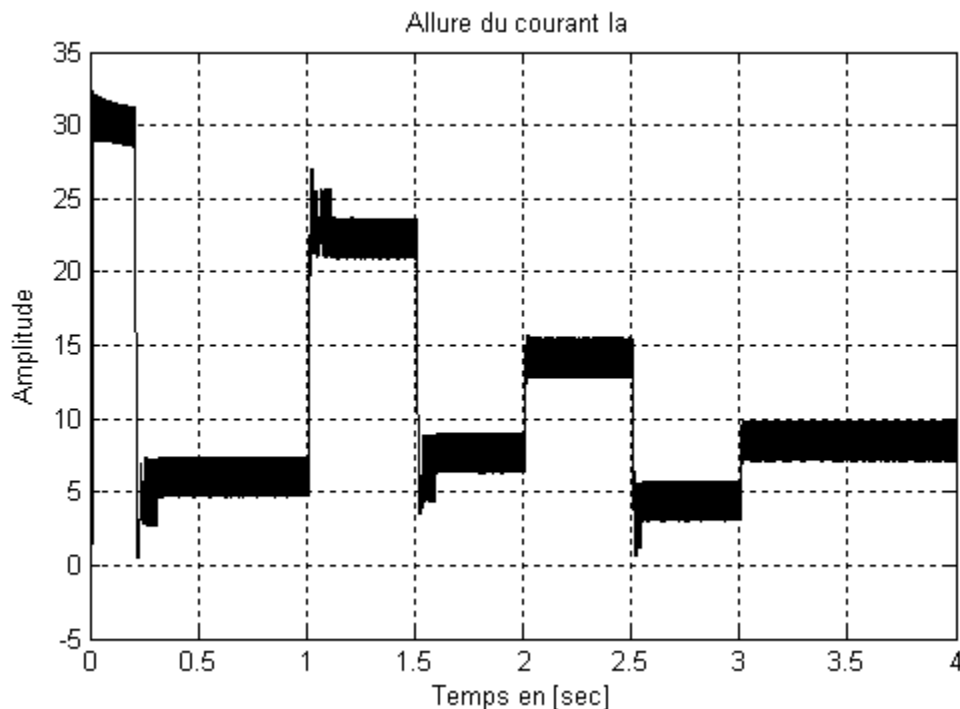


FIGURE 8.12 : Allure du courant d'induit I_a pour le moteur chargé

On observe également une oscillation de la valeur du courant I_a ceci est du au fait de l'oscillation de la vitesse de part et d'autre de la consigne provoque une création de couple de sens opposés, mais comme cette oscillation est faible elle n'est pas visible sur l'allure de la vitesse de la figure (8.10), le création de couple tendant à réduire la vitesse quand celle-ci est supérieure à la consigne entraine une baisse de la valeur du courant et une vitesse de sortie inférieure à la consigne entraine une augmentation de la valeur du courant I_a donc du couple moteur. L'amplitude des oscillations est due en partie à l'adaptation d'un système échantillonné (contrôleur STATEFLOW) pour l'action sur un système continu.

Pour tester l'efficacité des contrôleurs modélisés sous STATEFLOW, nous avons fait fonctionner le moteur en régime de fonctionnement nominal, ce mode de fonctionnement permet de valider la performance du limiteur de courant. Pour cela nous avons chargé le moteur avec le couple résistant de la figure (7.23), une modification des paramètres du contrôleur de vitesse a été effectuée pour qu'il puisse fonctionner dans ce régime, en effet la valeur de `step1` a été diminuée pour empêcher un dépassement trop important lors des variations de charges.

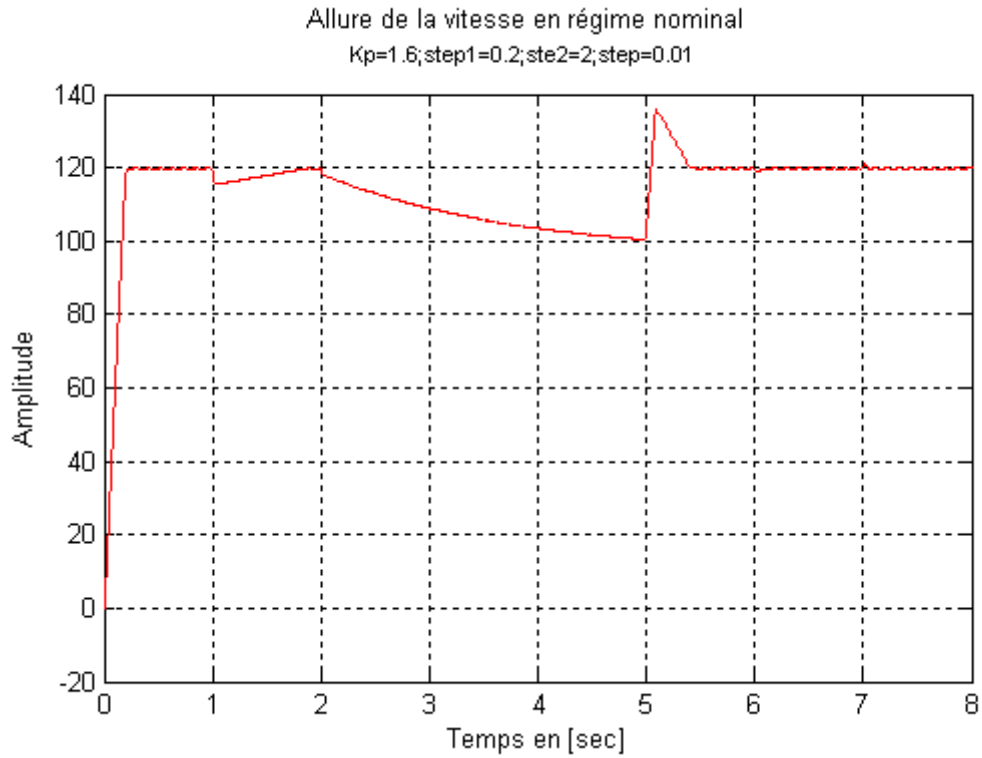


FIGURE 8.13 : Allure de la vitesse du moteur en régime de fonctionnement nominal

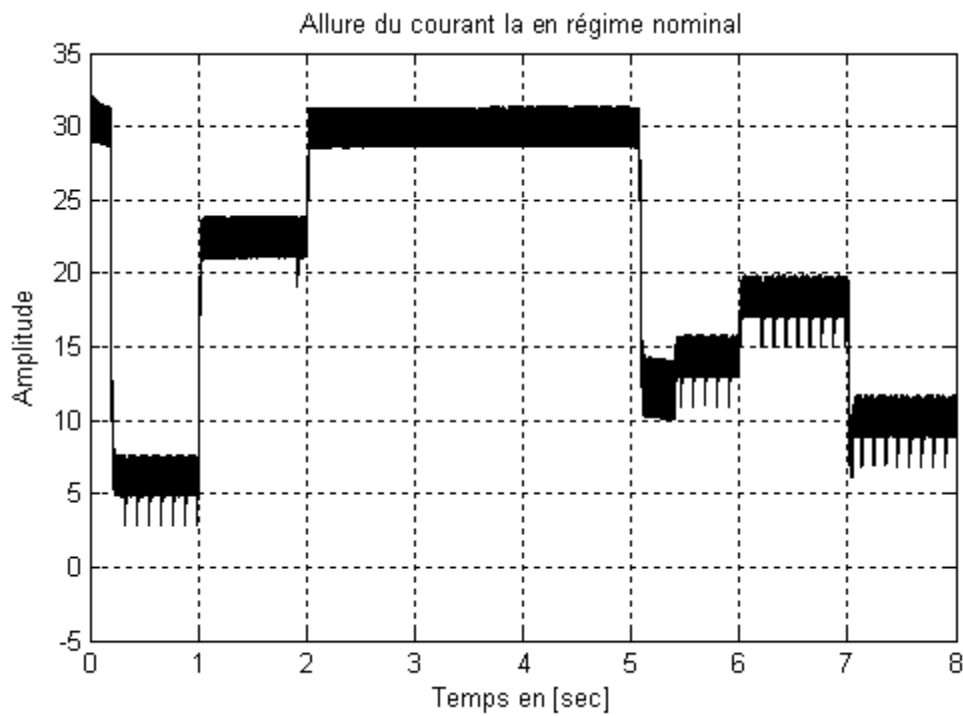


FIGURE 8.14 : Allure du courant d'induit du moteur en régime de fonctionnement nominal

Chapitre 9

Comparaison des modèles et interprétations

VI. Introduction :

L'étude des performances des systèmes asservis consiste à évaluer et comparer les systèmes suivant les 4 critères suivants :

- La stabilité
- La précision
- La rapidité
- La qualité de l'asservissement

Pour l'étude de ces critères nous nous limiterons au cas du système théorique. Nous présenterons sous forme de tableau les résultats obtenus.

a. La stabilité :

Un système dynamique linéaire est stable si et seulement si, écarté de sa position d'équilibre par une sollicitation extérieure, le système revient à cette position d'équilibre lorsque la sollicitation a disparue.

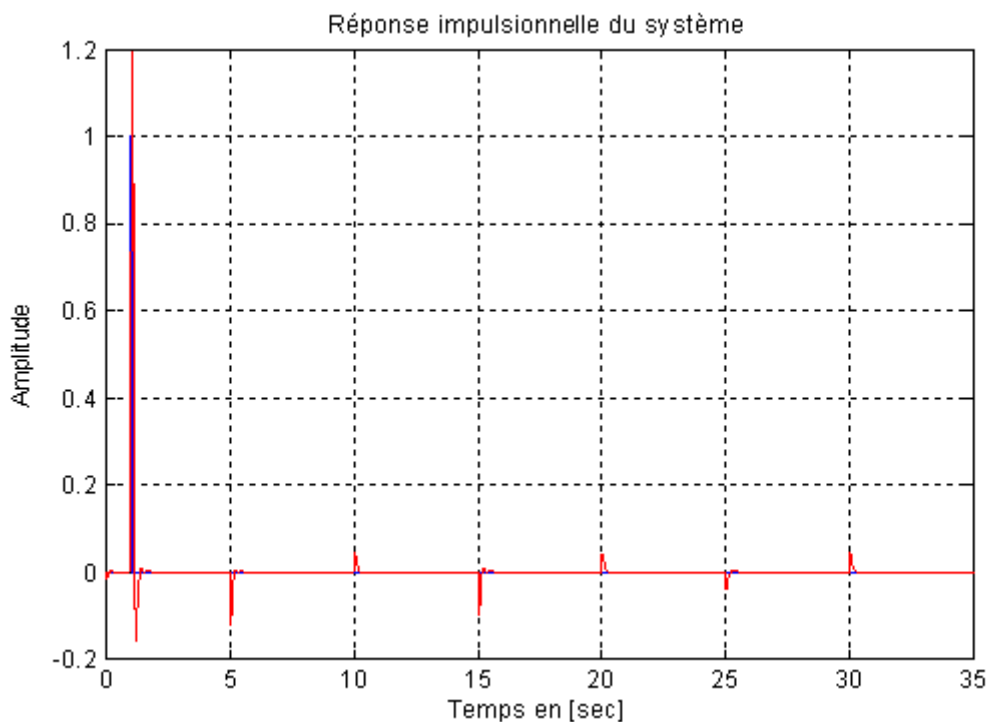


FIGURE 9.1 : Réponse impulsionnelle du modèle classique

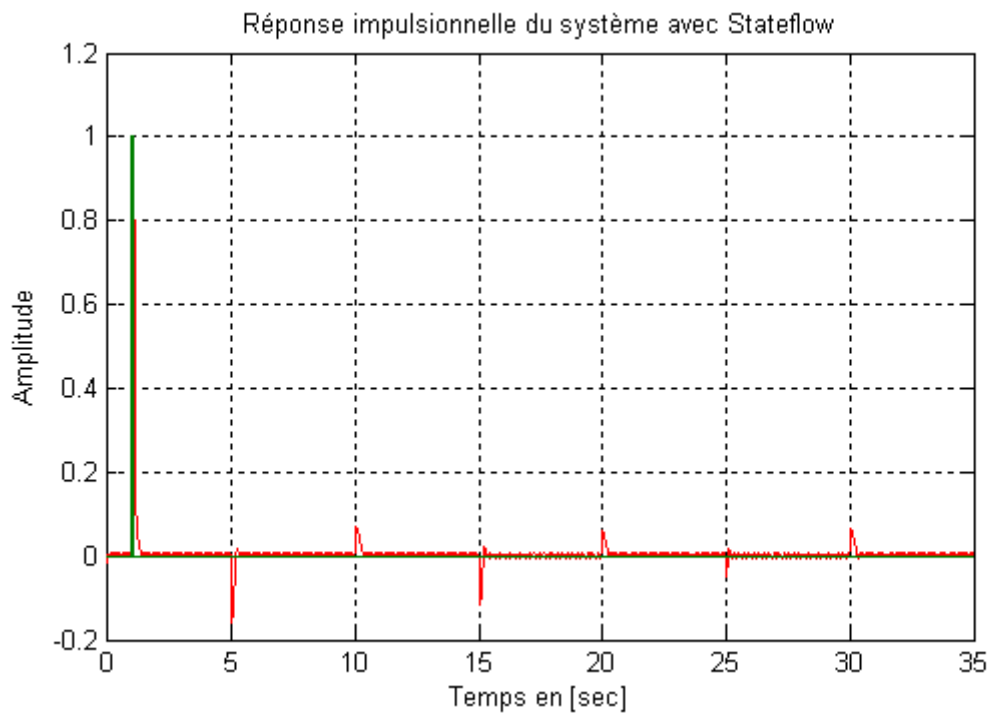


FIGURE 9.2 : Réponse impulsionnelle du modèle STATEFLOW

b. La précision :

La précision d'un système de régulation automatique est définie par les valeurs de ses signaux d'erreurs, en valeurs relatives, en régime transitoire et en régime permanent. L'étude de la précision statique peut se faire sur la base de la réponse indicielle en régime permanent et par l'observation de l'erreur de sortie du comparateur.

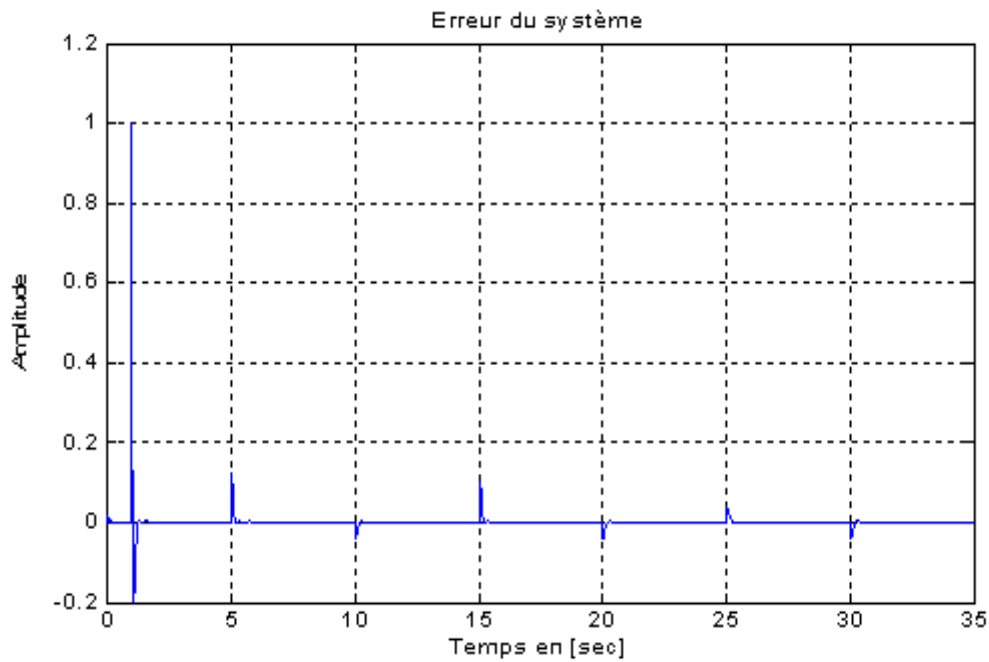


FIGURE 9.3 : Variation de l'erreur du modèle classique

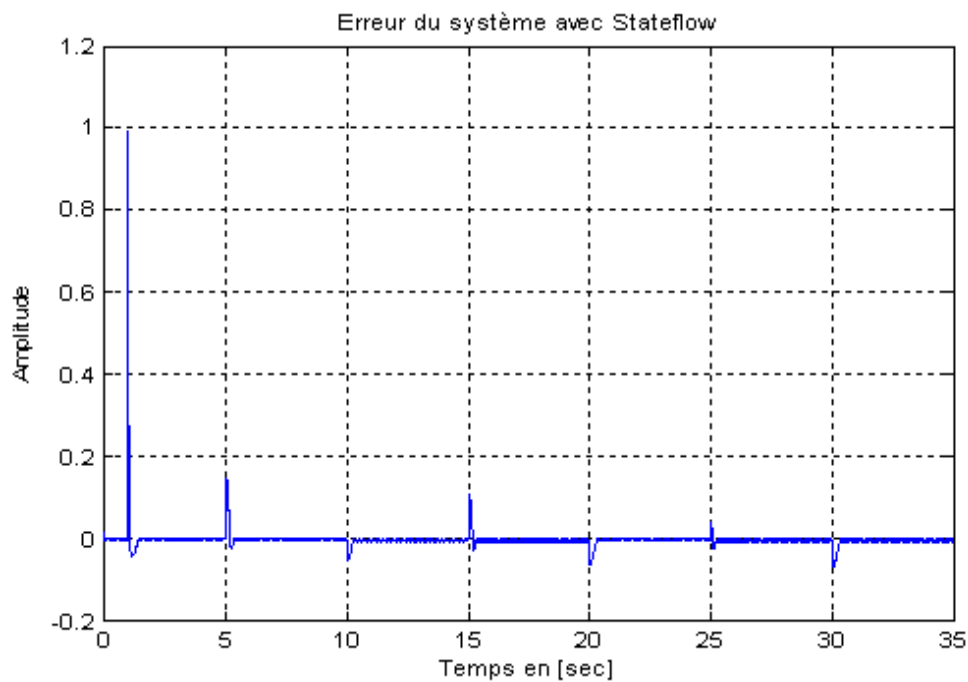


FIGURE 9.4 : Variation de l'erreur du modèle avec STATEFLOW

c. La rapidité :

La rapidité d'un système de régulation peut être évaluée sur la base du régime transitoire de sa réponse indicielle en boucle fermée.

1. Le temps de réponse ou la durée de réglage:

Chapitre 9 : comparaison des modèles et interprétations

La durée de réglage est l'intervalle de temps compris entre l'instant d'une variation en échelon d'un signal d'entrée et l'instant où la variation corrélative du signal de sortie ne s'écarte plus au-delà d'une tolérance spécifiée, en général un couloir de $\pm 5\%$ de sa valeur de régime finale.

2. Le taux de dépassement :

Pour une réponse indicielle, le taux de dépassement est l'écart transitoire maximal entre la réponse et la valeur du régime final exprimée en [%] de la différence entre la valeur de régime finale et la valeur de régime initiale.

3. Temps de montée et temps de pic :

Pour une réponse indicielle, le temps de montée est l'intervalle de temps compris entre l'instant où le signal de sortie, partant de la valeur zéro, atteint une fraction spécifique et faible, en général 10% de sa valeur en régime finale et l'instant où il atteint, pour la première fois, une fraction spécifiée élevée, en général 90%, de cette même valeur de régime.

Le temps de pic est l'intervalle de temps compris entre l'instant d'une variation en échelon d'un signal d'entrée et l'instant où le signal de sortie, partant de la valeur zéro, atteint sa valeur maximale qui est égale à 100% de sa valeur de régime finale augmentée du taux de dépassement.

Critère \ Système	Temps de réponse	Taux de dépassement	Temps de montée	Temps de pic
SIMULINK seul	0.2	20	0.04	0.109
STATEFLOW et SIMULINK	0.07	5	0.06	0.17

TABLEAU 9.1 : Caractéristiques de la courbe de démarrage du moteur

d. Qualité des asservissements : cette particularité sera juste citée à titre d'information, elle sera cependant plus amplement détaillée en annexe (Annexe2).

VII. Interprétations et conclusions :

Nous nous baserons sur les trois premiers critères de performance pour juger de l'efficacité et de la performance des systèmes de régulation étudiés.

D'après les courbes de la figure (9.1) et (9.2) présentant la réponse à une impulsion du système régulé dans les deux cas de modélisation (classique et STATEFLOW), nous pouvons en déduire que les deux systèmes possèdent approximativement la même réponse impulsionnelle on peut donc dire que le système est stable dans les deux cas, mais nous remarquons que la figure (9.2) présente un pic d'amplitude plus faible que celui de la figure (9.1), ceci peut se traduire par le fait que le système modélisé avec STATEFLOW réagit plus robustement aux variations extérieures du fait de ses paramètres jouables.

Pour la précision le système modélisé avec STATEFLOW présente des oscillations de faibles amplitudes, ces oscillations sont dues au fait qu'en régime permanent la valeur de la vitesse du moteur ne tend pas vers celle de la consigne mais y approche sensiblement, l'effet de la correction par le modèle STATEFLOW agit périodiquement mais pas continuellement. L'observation des courbes d'erreur de la figure (9.4) montre qu'une annulation de l'erreur en régime permanent, en réalité l'erreur n'est jamais nulle car elle génère le signal de commande, mais on peut constater que la valeur de l'erreur en régime permanent tend suffisamment vers zéro pour qu'elle soit acceptable.

Concernant la rapidité des deux systèmes, les valeurs du tableau (9-1) montrent que pour le taux de dépassement et le temps de réponse, le système modélisé avec STATEFLOW est avantageux, ceci se traduit du fait que le système STATEFLOW ne présente qu'une seule entrée dans le couloir de 5% et n'en sort plus, d'où le retard de la réponse indicielle du système STATEFLOW lorsqu'il est comparé à celui du modèle SIMULINK. Ce retard se traduit par les valeurs du temps de pic et du temps de montée du modèle STATEFLOW par rapport à celui du modèle SIMULINK.

Pour conclure le modèle STATEFLOW est adapté pour la régulation de vitesse des moteurs à courant continu, cependant il est nécessaire de résoudre le problème de la fréquence d'échantillonnage ; l'utilisation d'un élément discret (bloc STATEFLOW avec trigger) dans une chaîne de régulation continue pose un problème de fréquence d'échantillonnage. Le choix de la fréquence d'échantillonnage sera donc l'une des étapes à définir pour le fonctionnement optimum de la chaîne. Une fréquence trop élevée provoquera la nécessité d'un temps de calcul important par contre si la fréquence d'échantillonnage est trop faible, le problème de reconstitution du signal se posera (Théorème de SHANNON). Une possible extension du travail pourra se faire dans le domaine de la commande par ordinateur numérique, le problème d'échantillonnage peut être résolu par l'implantation de convertisseur analogique-numérique (CAN) et numérique-analogique (CNA) dans la chaîne de régulation, cela permettra de cadencer le fonctionnement de la chaîne. Le programme développé avec STATEFLOW pourra être implanté dans la mémoire de l'ordinateur qui sera le réseau correcteur pour développer une loi de commande numérique.

CONCLUSION GENERALE

Le travail présenté dans ce mémoire concerne la modélisation et la commande des moteurs à courant continu par des approches relativement linéaires. La méthode utilisée dans cette étude étant une méthode peu abordée en termes de régulation, elle s'est voulue pragmatique mais aussi méthodologique de manière à réduire la complexité d'application des technologies de conceptions.

Ainsi l'objectif du mémoire était double, d'une part offrir au concepteur de la commande une nouvelle méthode et des outils lui permettant une aide à l'intégration de la commande et d'autre part d'appliquer une méthodologie obtenue à partir de plusieurs exemples de conception dans le domaine de la régulation de vitesse.

La première phase de ce travail a porté sur l'étude des techniques de modélisation des moteurs à courant continu, nous nous sommes limités à un modèle linéaire du moteur, puis nous avons explicité les principes généraux de la régulation. Le premier chapitre rappelle les principes physiques entrant en jeu dans la conversion électromécanique, l'exploitation des équations constitutives de la machine ont permis la mise en œuvre d'un modèle linéaire du moteur.

La deuxième phase de l'étude est consacrée à la modélisation de la chaîne de régulation. Elle se compose de trois chapitres, le premier s'est étalé sur la mise en œuvre d'un modèle linéaire de la chaîne de régulation et sur la synthèse des correcteurs de vitesse et de courant par le modèle continu de Laplace, la synthèse des coefficients des correcteurs de vitesse et de courant par la méthode de Ziegler Nichols donne un modèle exploitable pour la régulation de vitesse d'un moteur à courant continu, le deuxième chapitre a essayé de résumer en quelques paragraphes la théorie des graphes d'états et a énoncé les outils de bases nécessaires à la

modélisation sous l'environnement MATLAB/SIMULINK/STATEFLOW. Le dernier chapitre s'est efforcé de mettre en œuvre un modèle de contrôleur de vitesse et de courant, le modèle a été conçu pour être ni trop complexe, pour pouvoir générer des lois de commandes exploitables, ni trop simples, pour être valable en régime transitoire. La simulation de l'effet des contrôleurs modélisés avec STATEFLOW sur la machine à courant continu ont permis d'effectuer une régulation efficace de la vitesse du moteur, que ce soit sur le plan qualité que sur le plan énergétique.

La dernière phase de notre travail s'est portée sur l'application au processus de régulation dans le cas réel, entre autre la commande de l'alimentation et la régulation du moteur proprement dite. La modélisation du contrôleur de vitesse et de courant dans la boîte 'SimPowersystems' de MATLAB s'est vu remarqué la présence d'éléments non linéaires, nous avons introduit un bref rappel sur ces éléments particulièrement sur la saturation et l'élément tout ou rien avec hystérésis. La modélisation de ces éléments non linéaires avec STATEFLOW a permis d'exécuter une régulation convenable de la machine et de son alimentation.

Au terme de cette étude nous avons constaté que notre première approche dans la modélisation avec STATEFLOW pour la régulation de vitesse était assez concluante, et que des travaux futurs devraient être entrepris dans le domaine de la régulation numérique, en effet les systèmes numériques sont actuellement en pleine expansion, ils ont l'avantage d'être puissants. L'utilisation d'un outil de modélisation pratique, STATEFLOW, à la régulation numérique aurait pour conséquence d'éveiller un nouvel aspect de l'asservissement.

Caractéristiques du moteur à courant continu

c. Paramètres nominaux	Valeurs numériques
Puissance nominale	$P_u = 3 \text{ [kW]}$
Tension nominale d'induit	$U = U_{an} = 110 \text{ [V]}$
Tension d'excitation	$U_{fn} = 110 \text{ [V]}$
Courant d'induit	$I_{an} = 30 \text{ [A]}$
Courant nominal de l'inducteur	$I_{fn} = 1.3 \text{ [A]}$
Vitesse de rotation	$N = 1500 \text{ [tr/min]}$
Paramètres du circuit d'induit	$R_a = 0.365 \text{ } [\Omega] ; L_a = 0.0115 \text{ [H]}$
Paramètres du circuit inducteur	$R_f = 85 \text{ } [\Omega] ; L_f = 1.01 \text{ [H]}$
Moment d'inertie totale du moteur	$J = 0.079 \text{ [kg.m}^2\text{]} ;$
Couple résistant	$C_r = 0.0645 \text{ [N.m]}$
Coefficient de frottement visqueux	$f = 0.04 \text{ [N.m.s]}$
Constante du couple	$K_E = k\Phi = 0.61$

CARACTERISTIQUES DES COMPOSANTES DE PUISSANCES UTILISEES DANS LA CHAINE DE REGULATION

Caractéristiques de la diode à roue libre

Caractéristiques	Valeurs
Resistance R_{on} (Ω)	0.05
Inductance L_{on} (μH)	5
Forward voltage (V)	0.7
Snubber resistance R_s (Ω)	inf
Snubber capacitance C_s (μF)	0.1

Caractéristiques du pont de thyristor GTO

Caractéristiques	Valeurs
Resistance R_{on} (Ω)	0.05
Inductance L_{on} (μH)	5
Forward voltage (V)	1
Current 10% fall time T_f (μs)	1
Current tail time T_t (μs)	1
Initial current I_c (A)	0
Snubber resistance R_s (Ω)	100
Snubber capacitance C_s (μF)	0.1

Qualité des asservissements :

Pour évaluer la qualité d'une régulation automatique, il faut aussi connaître l'évolution du signal d'erreur en régime transitoire. Cette évaluation peut se faire à partir de critères de qualité, dont seuls les plus connus sont présentés. Dans ce cas il s'agira toujours de l'étude de l'erreur transitoire $e(t)$ apparaissant dans la réponse indicielle à l'excitation échelon-unité $\varepsilon(t)$.

On chiffre la qualité d'une régulation au moyen de l'indice de performance J qui est une fonction de l'intégrale de $e(t)$. Deux cas peuvent être rencontrés :

- L'erreur statique $E_{\infty} = 0$, on intègre alors de 0 à l' ∞ .
- L'erreur statique $E_{\infty} \neq 0$: on intègre alors de 0 à $2T_r$.

Dans notre cas nous utilisons celui où l'erreur statique n'est jamais nulle car la chaîne fonctionne en régulation et c'est ce signal d'erreur qui élabore le signal de commande, autrement dit si il n'y a pas d'erreur il n'y aura pas de sortie.

Le choix d'un critère dépend essentiellement du cahier des charges. Si on veut minimiser l'énergie de réglage, on choisira le critère Integral Absolute Error (IAE) et même mieux le critère Integral of Time-multiplied Absolute-value of Error (ITAE). Si l'on veut privilégier le temps de montée on prendra le critère Integral of Time-multiplied Square Error (ITSE) et enfin si on veut maîtriser les coûts on a intérêt à prendre le critère Integral Square Error (ISE).

a. Le critère de performance ISE (Integral Square Error) :

Appelé encore critère de l'erreur quadratique ou critère de Hall-Satorius il est défini par :

$$J = \int_0^{2Tr} e^2(t) dt$$

Ce critère met en évidence les écarts transitoires de forte amplitude par rapport à ceux de faible amplitude (élévation au carré). Il est très significatif lorsqu'il s'agit de minimiser l'énergie de réglage.

b. Le critère de fonctionnement ITAE (Integral of Time-multiplied Absolute-value of Error) :

Ce critère s'écrit :

$$J = \int_0^{2Tr} t |e(t)| dt$$

Il sert à mettre en évidence les valeurs de l'erreur en fin de régime transitoire. La minimisation du critère J conduit à une réponse indicielle bien amortie avec de faibles dépassements.

c. Le critère ITSE (Integral of Time-multiplied Square Error) :

Lorsqu'on veut pondérer la valeur de l'erreur en fin de régime transitoire, on utilise le critère ITSE. Elle est donnée par l'expression :

$$J = \int_0^{2Tr} t e^2(t) dt$$

d. Le critère IAE (Intégral Absolute Error) :

Ce critère est surtout intéressant lorsqu'on a affaire à des systèmes à amortissement moyen et à réponse transitoire peu oscillante.

$$J = \int_0^{2Tr} |e(t)| dt$$

Généralement le choix du critère pour un problème donné dépend essentiellement des contraintes imposées dans le cahier des charges. On a donc omis volontairement cette partie, nous l'avons juste citée à titre d'information.

BIBLIOGRAPHIE

[1] : ANDRIAMITANJO SOLOFOMBOAHANGY, *Cours d'asservissement linéaire continu* ; Quatrième année Génie Industriel, Ecole Supérieure Polytechnique d'Antananarivo, 2006.

[2] : ANDRIANAHARISON Yvon, *Cours de machine à courant continu* ; Troisième année Génie Industriel, Ecole Supérieure Polytechnique d'Antananarivo, 2005.
Cours d'automatique numérique ; Quatrième année Génie Industriel, Ecole Supérieure Polytechnique d'Antananarivo, 2006.
Cours de commande numérique ; Cinquième année Génie Industriel, Ecole Supérieure Polytechnique d'Antananarivo, 2007.

[3] : ARZELIER D., PEAUCELLE D., *Systèmes et asservissements non linéaires* ; Cours du Conservatoire National des Arts et Métiers (CNAM), 2003. Version PDF.

[4] : BERNOT F., *Machines à courant continu. Constitution et fonctionnement* ; Techniques de l'ingénieur, Traité de Génie Electrique, D3555.pdf.
Machines à courant continu. Construction ; Techniques de l'Ingénieur, Traité de Génie Electrique, D3556.pdf.

[5] : ETIQUE M., *Entraînements réglés* ; Cours de la Haute Ecole d'Ingénierie et de Gestion du canton de Vaud (HEIG-Vd), 2006. Version PDF.
Régulation automatique ; HEIG-Vd, 2006. Version PDF.

- [6] : LOUIS J.P., MULTON B., BONNASSIEUX Y., LAVABRE M. , *Commande des machines à courant continu à vitesse variable* ; Techniques de l'Ingénieur, Traité de Génie Electrique, D3610.pdf.
- [7] : LOUIS J.P., BERGMANN C., *Convertisseur-moteur à courant continu* ; Techniques de l'Ingénieur, Traité de Génie Electrique, D3641.pdf.
Commande numérique des machines-Evolution des commandes ; Techniques de l'Ingénieur, Traité de Génie Electrique, D3640.pdf
- [8] : MARET Louis, *Régulation automatique. Système analogique* ; Presses polytechniques romandes, 1987.
- [9] : MARTIN R.C., *UML Tutorial : Finite State Machine*; Engineering Notebook Column, 1998. Version PDF.
- [10] : PASQUALINI G., *Alimentation par convertisseur statique* ; Techniques de l'Ingénieur, Traité de Génie Electrique, D3562.pdf.
- [11] : PHILIPPOT A., *Contribution au diagnostic décentralisé des systèmes à évènements discrets : Application aux systèmes manufacturiers* ; Thèse de doctorat en génie Informatique, Automatique et Traitement du Signal, Université de Reims Champagne Ardenne, 2006. Version PDF.
- [12] : RAJAONARISOA Ando Nomenjanahary, *Conception et réalisation d'un banc d'essai de servomécanisme* ; Mémoire d'ingénieur en Génie Industriel, Ecole Supérieure Polytechnique d'Antananarivo, 2004.
- [13] : RAKOTOARISOLO Andry, *Influence de la non linéarité sur la régulation automatique d'un processus. Application à un moteur à Courant continu* ; Mémoire d'ingénieur en Génie Industriel, Ecole Supérieure Polytechnique d'Antananarivo, 2006.
- [14] : RAOELISON Hery Zolalaina, *Asservissement de vitesse d'un moteur à Courant continu par correcteur Tout ou Rien et par correcteur à logique floue* ; Mémoire d'ingénieur en Génie Industriel, Ecole Supérieure Polytechnique d'Antananarivo, 2004.
- [15] : SEGUIER G., NOTELET F., *Electrotechnique industrielle* ; Techniques et Documentation, Lavoisier, 1977.
- [16] : *STATEFLOW and STATEFLOW Coder6 , Help MATLAB R14*; The Mathworks Incorporation, 2004. Version PDF.
- [17] : TRUONG TRUNG N., MODIGUY C., *Méthode Support pour le bon usage de MATLAB/SIMULINK/STATEFLOW*; Communication & Systèmes, Ingénierie Scientifique, 2002. Version PDF.

Titre : MISE EN ŒUVRE D'UN ALGORITHME EVENEMENTIEL SUR LA
REGULATION AUTOMATIQUE.

Auteur : Mr. RAMAROSON Anthony

Contact : 0331277725 - 0324414694

Directeur de mémoire : Pr. Yvon ANDRIANAHARISON.

RESUME

Les machines électriques jouent un rôle important dans l'industrie, elles sont généralement utilisées pour générer une puissance électrique (génératrice) ou pour produire un travail mécanique (moteur). Le cas des moteurs à courant continu concerne principalement les actionneurs en automatique, l'asservissement de vitesse des moteurs à courant continu est donc une partie indispensable dans les systèmes automatisés. Le but de ce travail est d'introduire un nouvel outil graphique de modélisation et d'utiliser cet outil pour la simulation de la régulation de vitesse des moteurs à courant continu. L'utilisation de l'outil STATEFLOW de MATLAB pour cette modélisation permettra la mise en œuvre d'un modèle de simulation en vue de contrôler le comportement dynamique du moteur et celles des interactions souvent non directement apparent dans la théorie. Ainsi nous proposons un modèle flot de données des contrôleurs de vitesse et de courant. Afin d'effectuer une comparaison nous avons étudié différentes lois de commandes, notamment celle de la commande par correcteur PI. Ce type de commande s'avère donner de bons résultats de simulation. En effet elles ont permis de répondre à plusieurs critères ; par exemple le temps de réponse, dépassement, etc. Nous proposons une extension du travail dans le domaine de la commande par calculateur numérique.

Mots clés : automatique, simulation, modélisation, flots de données, contrôleurs.

ABSTRACT

Electric machines play important role in industry, they are used to generate electrical power (generator) or provide mechanical work (motor). The case of DC machine concern firstly automatic like robotic, then speed regulation of DC motor is a very important part of automatic system. The aim of this work is to provide a new modeling tool for the simulation of DC motor speed regulation. The use of MATLAB/STATEFLOW for this simulation, allows creating a model in sight to control the dynamic motor behavior and interactions that are not directly visible in theory. We introduce a new data flow model of speed and current controller; in order to carry out a comparison we studied various linear controls, in particular, the control by a PI corrector. This type of order proves to give good results. Indeed, they made it possible to answer several criteria; for example the response time, going beyond, etc. In this context we propose one extension of the work, about the order with numeric calculator.

Key words: automatic, simulation, modeling, data flow, controllers.

Nombre de pages : 119.

Nombre de figures : 117

Nombre de tableaux : 13