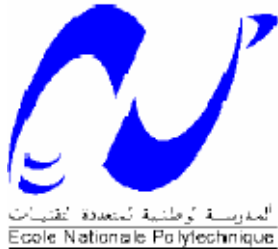


REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTRE DE L'ENSEIGNEMENT SUPERIEUR ET DE
LA RECHERCHE SCIENTIFIQUE

ECOLE NATIONALE SUPERIEURE POLYTECHNIQUE
Laboratoire Commande des processus
Département du Génie Electrique
Spécialité Automatique



Projet de fin d'études

Pour l'obtention du diplôme

d'Ingénieur d'Etat en Automatique

Intitulé

**Réalisation et Commande d'un
système à lévitation magnétique**

Proposé et dirigé par :

Pr. F. BOUDJEMA

Pr. D. BOUKHETALA

Etudié par:

OUACENE Hakim

BOULDJENIB Badreddine

Promotion : Juin 2009

Ecole Nationale Polytechnique 10, avenue Hassen Badi BP182 El-Harrach
16200 Alger (Algérie)

ملخص:

في هذا العمل سنتطرق إلى كيفية إنجاز جهاز تجريبي لنظام الحمل المغناطيسي و كذلك القيام بعملية التحكم فيه. في البداية سنعرف بالجهاز المذكور مع ذكر الإمكانيات العديدة المتاحة عند بنائه. سنقوم بعدها بإعطاء النموذج الرياضي اللاخطي لهذا النظام و الذي سيستغل لاحقا لتركيب قوانين تحكم خطية (PD) و أخرى لاختية. برنامج التحكم بالنظام سيتم تثبيته على متحكم مصغر dsPIC باستعمال لغة برمجة من المستوى العالي(C 30). من ناحية أخرى سيتم عرض المخططات الكهربائية الكاملة لدارة التحكم بكافة تفاصيلها.

الكلمات المفتاحية : حمل مغناطيسي ، متحكم dsPIC ، لغة البرمجة C30 ، تحكم رقمي ،قانون التحكم PD ، قانون التحكم اللاخطي.

Résumé :

Dans le présent travaille, nous allons entamer la réalisation et la commande d'un système à lévitation magnétique, on va d'abord présenter la maquette expérimentale et expliquer les possibilités qui existent lors de la construction de tels systèmes. Un modèle mathématique non linéaire sera donné et sera utilisé pour la synthèse d'un correcteur linéaire (régulateur PD) puis non-linéaire (commande linéarisante). Le programme de commande sera implémenté sur un microcontrôleur dsPIC en langage évolué C30, et le schéma du circuit de commande sera donné en détail.

Mots clés : lévitation magnétique, dsPIC30F3010, compilateur C30, commande numérique, régulateur PD, commande en modes glissants.

Abstract :

In this work, we are going to build and control a magnetic levitation system. First we're going to present the experimental set and explain all the possibilities that exist when building such systems. A non linear mathematical model would be given and used to calculate linear an then non-linear regulators (PD, linearising control). The control program would be implemented on a dsPIC microcontroller and written on high level language (C30). On the other hand the electrical circuits would be given in details.

key words : magnetic levitation, dsPIC30F3010, C30 compiler, digital control, PD regulator, linearising control.

Remerciements

Nous tenons à remercier en premier lieu ALLAH, le tout puissant, qui nous a donné le courage et la volonté pour bien mener ce modeste travail.

Ce travail a été effectué au sein du Laboratoire de Commande des Processus au département du Génie Electrique de l'Ecole Nationale Supérieure Polytechnique d'Alger sous la direction de nos deux promoteurs Pr. BOUKHETALA.D et Pr. BOUDJEMA. F Aux quels nous tenons à exprimer nos plus sincères remerciements pour leurs précieux conseils, leur aide et leur patience tout au long de notre travail.

Nous remercions les membres du jury qui nous ont fait l'honneur d'examiner notre travail.

Nous remercions également Mr C. LARBES pour sa précieuse aide et l'intérêt qu'il a manifesté pour notre travail.

Nous tenons à remercier Mr BAGHLI.L du laboratoire G.R.E.E.N de l'université Henry Poincaré, Nancy, France pour ses précieux conseils et sa générosité de répondre à nos questions que nous lui envoyions par e-mail.

Nous remercions nos enseignants de l'Ecole Nationale Polytechnique pour leurs efforts voués à nous transmettre le savoir, en particulier ceux des départements Génie Electrique, Electronique et Sciences Fondamentales

Nous tenons à exprimer notre très grande gratitude, et notre profonde affection à nos chers parents pour leurs encouragements, leur patience et leur grand soutien, durant toutes ces années d'études.

Nous remercions également tous nos amis et camarades, pour leurs encouragements et leur précieux soutien.

Dédicace

Je dédie ce travail à :

Mes très chers parents ...

Mes très chères sœurs et mon très cher frère ...

A toute ma famille ...

A tous mes amis proches que j'aime ...

A tous ceux qui me sont chers.

BOULDJENIB Badreddine

Dédicace

Avant tout, à la personne la plus chère, celle que j'admire et j'aime le plus au monde, ma mère, qui s'est donné corps et âme pour l'éducation de ses enfants, et sans qui je ne serais jamais devenu ce que je suis

A mon père qui m'a épaulé tout au long de ma vie, qui a été un modèle pour moi, et que j'admire

A mes frères et sœurs que j'aime très fort, et à qui je souhaite tout le bonheur et la réussite du monde

Aux membres de ma famille (oncles, tantes, cousins et cousines)

A mes amis

A mes camarades de l'école nationale polytechnique

A tous ceux qui comptent pour moi

A tous ceux qui ont de près ou de loin assisté à ma formation

OUACENE Hakim.

Table des matières

Introduction générale.....	2
CHAPITRE I : Présentation générale de la maquette expérimentale.....	5
I.1- Introduction :	6
I.2- Description de la maquette :.....	6
I.2.1- Electroaimant :.....	7
I.2.2- Le corps lévité :	7
I.2.3- Capteur de position :.....	7
I.2.4 - Circuit de commande :.....	12
I.2.4.1 - Le dsPIC 30F3010 :	12
I.2.4.2 - Double hacheur 4 quadrants L298 :.....	14
I.3 - Conception et fabrication du circuit de commande :.....	16
I.3.2 - Fabrication du circuit:.....	17
I.3.2.1 - Présentation du logiciel de développement utilisé :.....	17
I.3.2.2 - Schémas électriques sous Multisim :.....	18
I.3.2.3 - Schéma PCB sous Ultiboard :.....	24
I.4 - Conclusion :.....	27
CHAPITRE II : Modélisation du système.....	28
II.1 – Introduction :	29
II.2 – Modélisation :.....	29
II.2.1 - Calcul de l'inductance du système :	29
II.2.2 - La force magnétique d'interaction : [7].....	30
II.3 - Les équations d'état du système : [8].....	32
II.4 - Calcul des paramètres :	33
II.5 - Commandabilité et observabilité	34
II.6 - Le modèle linéaire :	36
II.7 - Simulations en boucle ouverte :.....	39
II.8 – Conclusion :.....	40

CHAPITRE III : Synthèse des correcteurs.....	41
III.1 – Introduction :.....	42
III.2 –Transposition des méthodes analogiques :.....	42
III.2.1 –Discrétisation :.....	43
III.2.2 –Approximations de la variable s.....	43
III.2.2.1 –Discrétisation avant :	44
III.2.2.2 –Discrétisation arrière :	44
III.2.2.3 –Approximation de Tustin :.....	44
III.3 -Prise en compte du bloqueur dans la synthèse : [9].....	45
III.4 –Calcul du gain du capteur :	47
III.5 -Régulateur P.D. numérique :	48
III.6 –Les simulations :	51
III.6 –Teste de performance et de robustesse :	53
III.6.1 –Teste de performance :	53
III.6.2 –Teste de robustesse.....	54
III.7 –Programmation du PD :	55
III.8 –Commande non linéaire :	56
III.8.1 –Commande linéarisante : [10]	56
III.8.1 –Régulateur d'état :	58
III.8.3 –Les simulations.....	59
III.8.4 –Test de performance et de robustesse.....	60
III.8.4.1 –Test de performance :	60
III.8.4.2 –Teste de robustesse :	61
III.9 –Conclusion :	63
CHAPITRE IV: Programmation.....	64
IV.1- Introduction :	65
IV.2 - Configuration des périphériques :	65
Le registre TRISx :	67
Le registre PORTx :.....	67
Le registre LATx :.....	67
IV.2.2 - Le module Timer1 :	68
Le registre T1CON (Time Base Register) :.....	69

IV.2.3 - Le convertisseur analogique numérique (module ADC) :.....	70
Le registre ADCON1 (A/D Control Register 1):.....	71
Registre ADCON2 (A/D Control Register 2):	73
Le registre ADCON3 (A/D Control Register 3):.....	74
Le registre ADCHS (A/D Input Channel Select Register):.....	75
IV.2.4 - Le module PWM :	76
Le registre PTPER (PWM Time Base Period register):.....	77
Le registre PTMR (PWM Time Base register):.....	78
Les registres PDC1, PDC2, PDC3 (PWM Duty Cycle registers) :	79
Le registre PWMCON1 (PWM Control register #1) :	80
Le registre PWMCON2 (PWM Control register #2) :	80
Le registre OVDCON (Override Control Register):	81
Le registre SEVTCMP (PWM Special Event Compare register):	82
Le registre PTCON (PWM Time Base Control register):	82
IV.2.5 - Le module d'interruption :.....	83
Les registres IECx (Interrupt Enable Control Register) :	83
Les registres IFSx (Interrupt Flag Status Register) :	84
Les registres IPCx (Interrupt priority Control Register) :	84
Le module change notification (CN) :.....	85
IV.2.6 - Registres de configuration du fonctionnement général :	85
Le registre FOSC (Oscillator Configuration Register) :.....	86
Le registre FWDT (Watchdog Timer Configuration Register):.....	86
Le registre FBORPOR (BOR and POR Configuration Register) :	86
Le registre FGS (General Code Segment Configuration Register):.....	88
IV.3 - Particularités de la programmation C30 :.....	88
IV.4 - Déroulement et temps d'exécution du programme :	91
IV.6 - Outils de développement :.....	93
IV.7 - Conclusion :.....	94
Conclusion et perspectives.....	96
ANNEXE A.....	101
ANNEXE B.....	109
Référence et bibliographie.....	114

Table des figures

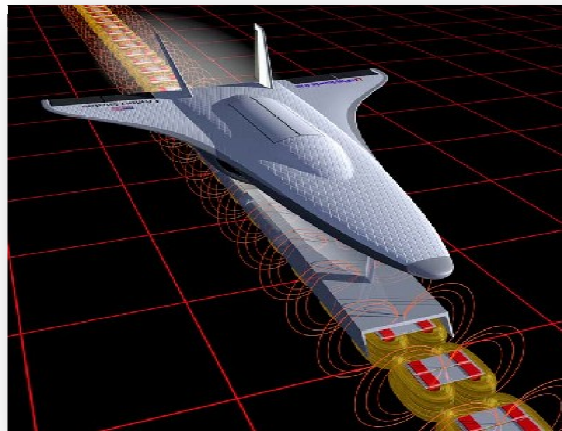
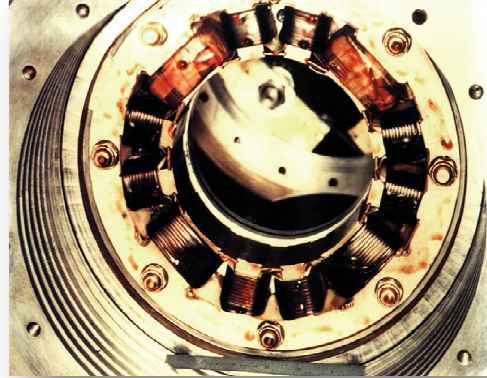
<i>Figure I.1 : Schéma général d'un système à lévitation magnétique.....</i>	<i>6</i>
<i>Figure I.2 : La solution optique.....</i>	<i>8</i>
<i>Figure I.3 : Autres solutions optiques.....</i>	<i>8</i>
<i>Figure I.4 : la solution du capteur ultrason.</i>	<i>9</i>
<i>Figure I.5 : la solution des deux aimants permanents.....</i>	<i>10</i>
<i>Figure I.6 : La solution des deux capteurs</i>	<i>10</i>
<i>Figure I.7 : Forme commerciale de capteurs à effet Hall.</i>	<i>11</i>
<i>Figure I.8: Variation de la tension de sortie en fonction de la distance (1) et du flux (2) pour le capteur UGN3503.....</i>	<i>12</i>
<i>Figure I.9 : Brochage du dsPIC 30F3010.....</i>	<i>13</i>
<i>Figure I.10 : le dsPIC 30F3010 sous forme de boîtier PDIP-28</i>	<i>14</i>
<i>Figure I.11 : Brochage et forme commerciale du L298</i>	<i>15</i>
<i>Figure I.12 : Schéma électrique interne du L298.....</i>	<i>15</i>
<i>Figure I.13 : Schéma fonctionnel de la maquette.</i>	<i>16</i>
<i>Figure I.14 : Schéma électrique d'alimentation</i>	<i>19</i>
<i>Figure I.15 : Schéma du circuit de l'horloge externe.....</i>	<i>20</i>
<i>Figure I.16 : Schéma du branchement des entrées.....</i>	<i>20</i>
<i>Figure I.17 : Schéma du branchement des entrées.....</i>	<i>21</i>
<i>Figure I.18 : Schémas électriques du branchement en parallèle du double hacheur 4 quadrants</i>	<i>22</i>
<i>Figure I.19 : Schéma électrique complet du circuit de commande.....</i>	<i>23</i>
<i>Figure I.20 : Schéma du coté composant.....</i>	<i>24</i>
<i>Figure I.21 : schéma du coté cuivre</i>	<i>25</i>
<i>Figure I.22 : Schéma du circuit supplémentaire contenant les contacts.</i>	<i>25</i>
<i>Figure I.23 : Représentation en 3D du circuit de commande (1, 2) et du module supplémentaire (3).....</i>	<i>27</i>
<i>Figure II.1 : Lieux des pôles pour le système</i>	<i>39</i>
<i>Figure II.2 : Résultat de simulation du système en boucle ouvert sans excitation.....</i>	<i>39</i>

<i>Figure II.3 : Simulation en boucle ouverte avec excitation.....</i>	<i>40</i>
<i>Figure III.1: Discrétisation d'un correcteur analogique</i>	<i>433</i>
<i>Figure III.2 : Approximation de la variable s.....</i>	<i>444</i>
<i>Figure III.3 : caractéristique du capteur (tension en fonction de la position).....</i>	<i>488</i>
<i>Figure III.4 : Lieu de racine d'Evans pour $K>0$.....</i>	<i>49</i>
<i>Figure III.5 : Lieu de racine d'Evans pour $K<0$.....</i>	<i>49</i>
<i>Figure III.6 : Résultats de simulation pour la commande linéaire (PD) (Réglages de position)</i>	<i>51</i>
<i>Figure IV.1 : brochage du dsPIC 30F301</i>	<i>66</i>
<i>Figure IV.2: Le registre TICON.....</i>	<i>69</i>
<i>Figure IV.3 : Etapes de fonctionnement du module ADC.</i>	<i>70</i>
<i>Figure IV.4 : Le registre ADCON1.....</i>	<i>71</i>
<i>Figure IV.5 : Le registre ADCON2.....</i>	<i>73</i>
<i>Figure IV.6 : Le registre ADCON3.....</i>	<i>74</i>
<i>Figure IV.7 : Le registre ADCHS.</i>	<i>75</i>
<i>Figure IV.8 : Edge Aligned PWM.....</i>	<i>76</i>
<i>Figure IV.9 : Center Aligned PWM</i>	<i>77</i>
<i>Figure IV.10 : résolution des registres PDCx par rapport au registre PTMR.....</i>	<i>79</i>
<i>Figure IV.11 : Le registre PWMCON1.....</i>	<i>80</i>
<i>Figure IV.12 : Le registre PWMCON2.....</i>	<i>80</i>
<i>Figure IV.13 : Le registre ADCON.....</i>	<i>81</i>
<i>Figure IV.14 : Le registre SEVTCMP.....</i>	<i>82</i>
<i>Figure IV.15 : Le registre PTCON</i>	<i>82</i>
<i>Figure IV.16 : Le registre FOSC.....</i>	<i>86</i>
<i>Figure IV.17 : Le registre FBORPOR.....</i>	<i>87</i>
<i>Figure IV.18 : Temps d'exécution de l'interruption de commande.....</i>	<i>91</i>
<i>Figure IV.19 : Package du programmeur ICD2.....</i>	<i>93</i>
<i>Figure IV.20 : Carte de développement EASYdsPIC4 de MikroElektronika.</i>	<i>94</i>

INTRODUCTION
GENERALE

Introduction générale

Les systèmes à lévitation (ou à sustentation) magnétique sont de plus en plus utilisés dans les différents secteurs de l'industrie et du transport, ils ont attirés l'attention des ingénieurs et des scientifiques car ils ont aidé à éliminer les pertes qui sont dues aux frottements des pièces mécaniques. Les applications majeures de ces systèmes dans l'ingénierie sont les trains à très grande vitesse, les plateformes à grande précision, les roulements magnétiques, les systèmes de suspension et d'isolation de vibrations...etc. Les avantages que présentent les systèmes à lévitation magnétique sont indiscutables : l'absence de frottements mécaniques, la rapidité, la simplicité relative de la mise en œuvre ... malgré cela ces systèmes présentent évidemment des inconvénients : une grande consommation d'énergie, des pertes électriques (par effet joule) et magnétiques considérables, mais plus important : la forte non-linéarité qui caractérise le comportement physique de ces systèmes génère de grands problèmes lors de la modélisation ce qui conduit forcément à des difficultés lors de la commande du système.



Un système classique didactique de lévitation magnétique est tout de même très facile à réaliser et à commander. Beaucoup de travaux et d'études ont été faits aux seins des universités à travers le monde en choisissant des approches de conception et de réalisation plus ou moins différentes (choix des technologies, des actionneurs, des capteurs, des méthodes de commande...).

Dans notre travail nous avons opté pour un système dont la complexité est réduite au maximum : les composants utilisés sont disponibles dans le marché et surtout pas chers, le choix de capteurs à effet Hall qui nécessitent le moins de travail d'ajustage, et finalement l'utilisation de la commande numérique basée sur un calculateur DSPIC au lieu de la solution analogique (pourtant plus rapide), celle-ci est justifiée par la réduction de la complexité de la carte de commande, les possibilités d'extension, et de révision, voir même du changement des lois de commande.

Ce mémoire est organisé comme suit :

Dans le premier chapitre nous allons donner une description générale sur la maquette expérimentale que nous avons réalisée, nous allons parler des différentes possibilités qui existent pour la réalisation du système, tout en justifiant nos choix. Puis on parlera des étapes de réalisation et présenter les schémas des circuits de commande pour expliquer le fonctionnement général de la maquette.

Dans le deuxième chapitre nous élaborons le modèle mathématique du système en citant les lois physiques qui entrent en jeu, puis nous allons effectuer la simulation du système en boucle ouverte et en tirer les remarques nécessaires.

Le troisième chapitre sera consacré à la mise en œuvre de la commande ainsi que la simulation et la vérification des performances du système en boucle fermé.

Le quatrième chapitre va parler de la programmation du calculateur en question en commençant par la présentation du calculateur (le DSPIC), le jeu de commande du compilateur utilisé, la présentation de l'environnement de développement MPLAB et finalement donner explicitement le programme de commande implémenté.

A la fin nous allons parler des perspectives et des applications ouvertes par ce travail, des possibilités d'extension de la maquette, et on va terminer par une conclusion générale.

CHAPITRE I

PRESENTATION GENERALE DE LA MAQUETTE EXPERIMENTALE

CHAPITRE I : Présentation générale de la maquette expérimentale

I.1- Introduction :

Plusieurs méthodes existent pour la réalisation d'un système à lévitation magnétique, Notre approche a été l'utilisation d'une bobine traversée par un courant électrique et qui va jouer le rôle d'un électroaimant générant une force magnétique agissant sur un corps doté d'un aimant permanent. D'autres approches existent et peuvent être trouvées dans la littérature, telles que l'utilisation d'aimants supraconducteurs, ou l'utilisation de d'objet métallique comme corps lévité. Dans ce chapitre on va donner une description générale du système et de son fonctionnement en justifiant notre approche.

I.2- Description de la maquette :

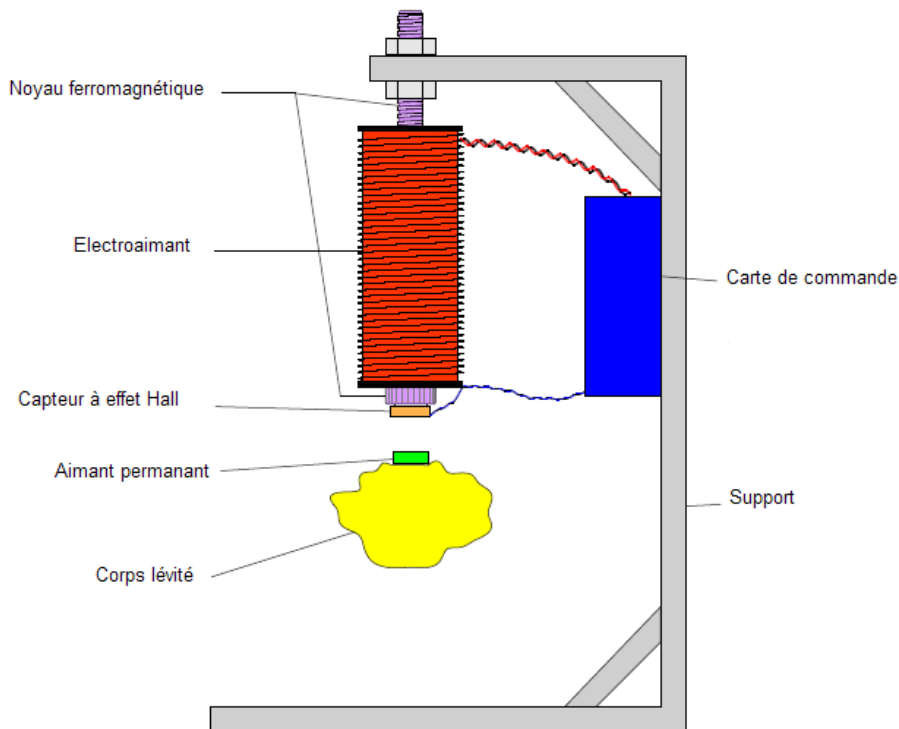


Figure I.1 : Schéma général d'un système à lévitation magnétique [1]

La maquette expérimentale comporte les éléments suivants :

I.2.1- Electroaimant :

C'est une simple bobine enroulé sur un noyau de fer (un boulon d'acier par exemple), on doit s'assurer qu'elle ne présente pas une grande impédance c'est-à-dire qu'elle doit avoir une résistance relativement petite et un grand nombre de spire N , tout de même il faut s'assurer que la résistance ne soit pas trop petite pour ne pas griller l'élément qui pilote la bobine (le hacheur)

I.2.2- Le corps lévité :

On va utiliser un corps pesant (marqueur, boulon...) sur lequel on va attacher un aimant permanent (NFeB ou NIB) pour qu'il puisse être attiré par l'électroaimant.

Les avantages de l'utilisation d'un aimant permanent à la place d'un objet métallique sont [2] :

Le problème du réchauffement de la bobine est modéré, car pour léviter un corps métallique l'électroaimant doit créer un champ magnétique assez fort, c'est-à-dire un grand courant qui va chauffer la bobine. Dans le cas de l'aimant permanent l'existence d'un champ magnétique créé par ce dernier et qui va s'ajouter au champ créé par la bobine va diminuer le courant total qui traverse la bobine.

La portée totale de la lévitation sera augmentée à cause de la même raison précédente, en fait dans la présence d'un aimant en NIB la densité du flux magnétique total est 4 à 5 fois plus grande que dans le cas d'un corps métallique.

I.2.3- Capteur de position :

Le capteur utilisé est un capteur de champs magnétique à effet Hall qui donne une tension de sortie linéaire par rapport à la grandeur de B mesurée, l'astuce est que le capteur mesure le champs magnétique créé par l'aimant permanent qui dépend de la distance qui sépare l'aimant du capteur, et retourne une tension qui est alors fonction de la distance de séparation.

D'autres solutions existent telles que l'utilisation d'un capteur optique (source lumineuse plus diode photoconductrice) mais ils nécessitent un grand travail d'ajustage avec des précisions qui reste toujours insatisfaisantes.

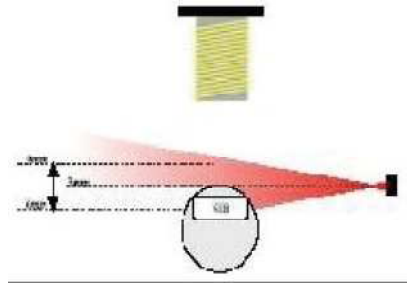


Figure I.2 : La solution optique [2]

Voici quelques configurations possibles dans le cas d'un capteur optique :

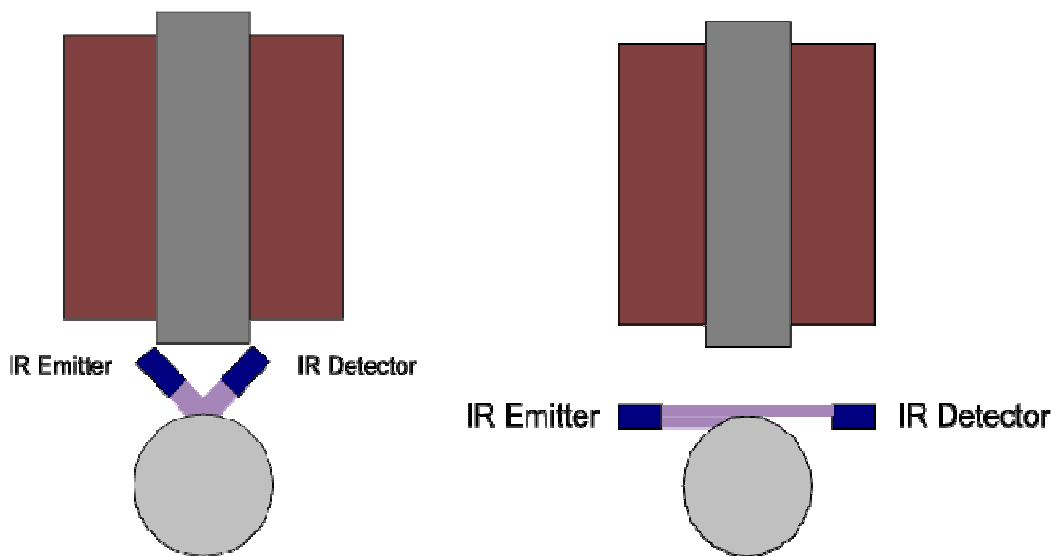


Figure I.3 : Autres solutions optiques

Une autre solution est l'utilisation de capteurs ultrasons, ces capteurs sont classiquement composés d'un émetteur ultrason et d'un récepteur, la distance sera calculer en mesurant le retard qui existe entre le signal émis et celui reçu, ces capteurs sont généralement dans les cameras pour faire le focus automatique, leur majeur problème est qu'ils ne sont pas assez rapides pour effectuer ce genre de commandes qui nécessite une réponse plus au moins rapide. C'est pour ça qu'il faut s'assurer d'abord de la rapidité d'un capteur de tel genre avant de l'utiliser.

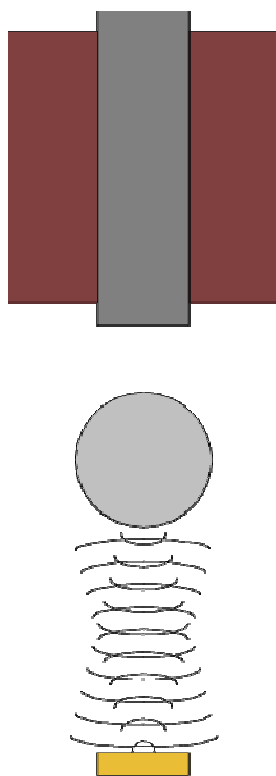


Figure I.4 : la solution du capteur ultrason.

Le problème qui se présente lors de l'utilisation d'un capteur à effet Hall est que en plaçant le capteur en dessous de la bobine il va aussi mesurer le champ magnétique créé par celle-ci ce qui va perturber la mesure de la position. En fait une régulation en position est possible même dans ce cas [3] puisque le champ créé par la bobine reste presque constant, mais lorsqu'il s'agit d'une suite de référence la qualité de la régulation reste loin du désiré.

Pour résoudre ce problème ils existent plusieurs solutions astucieuses : la première est d'utiliser un corps relativement long est deux aimants permanents, l'un sera placé au dessus du corps pour assurer la lévitation, et le deuxième sera placé en dessous pour assurer la détection de la position par un capteur à effet hall qui sera placé plus bas loin des perturbations de la bobine.

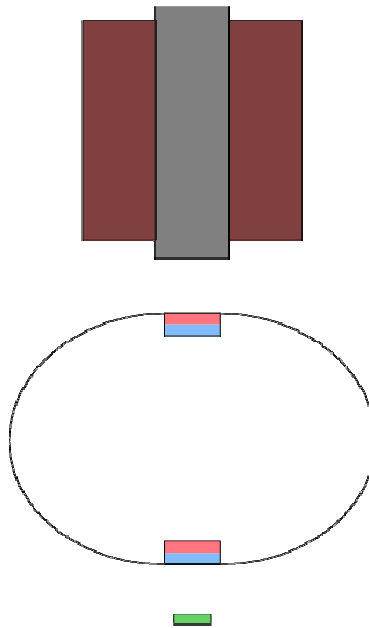


Figure I.5 : la solution des deux aimants permanents

L'autre solution est celle que nous avons opté : on a utilisé deux capteurs au lieu qu'un seul : un sera placé en dessus de la bobine, et l'autre en dessous de celle-ci, les deux signaux venants des capteurs seront passés dans un circuit soustracteur, alors on aura un signal final qui correspond à la position de l'aimant permanent puisque la bobine présente un champ magnétique uniforme ce qui permet d'annuler les perturbations créer par celle-ci.

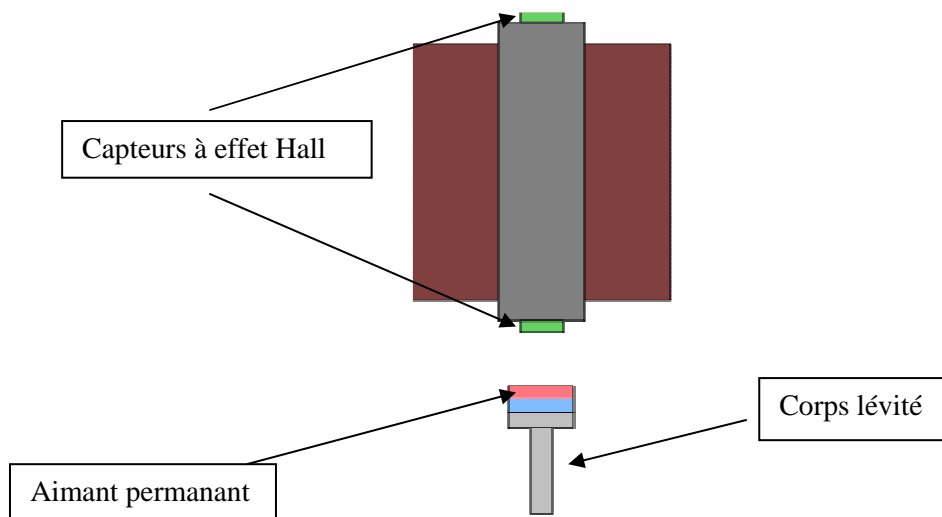


Figure I.6 : La solution des deux capteurs

Les capteurs utilisés sont des capteurs qui donnent une tension de sortie linéaire par rapport au champ mesuré. Lorsqu'ils sont alimentés en 5V, cette tension varie entre 0 V et 5 V, dans l'absence de champs (pas d'aimant) le capteur retourne une valeur de 2.5 V.

Ces capteurs sont fabriqués principalement par les firmes ALLEGRO et HONEYWELL, les meilleurs capteurs sont ceux qui possèdent une sensibilité assez grande, voici un tableau qui contient les capteurs qui peuvent être utilisés et leur grandeur de sensibilité :

Constructeur	Capteur	Sensibilité (mV/G)
Allegro	A1321 / A1315	5
Allegro	A1322	3.125
Allegro	A1323 / A1316	2.5
Honeywell	SS495 / SS490	3.125
Allegro	UGN 3503	1.5
Allegro	UGN 3501	0.7

Tableau I.1: Différents types de capteurs à effet Hall

Dans le marché nous avons trouvé que le meilleur capteur qui existe et l'UGN3503, d'autres sont malheureusement indisponibles. Malgré qu'il soit généralement utilisé pour la détection de proximité (à cause de sa sensibilité relativement faible), il peut être utilisé pour la détection de position lorsqu'il s'agit de distances relativement faibles (ce qui est notre cas). Des réalisations de système à lévitation magnétique utilisant ce même capteur ont déjà été faites [4].



Figure I.7 : Forme commerciale de capteurs à effet Hall.

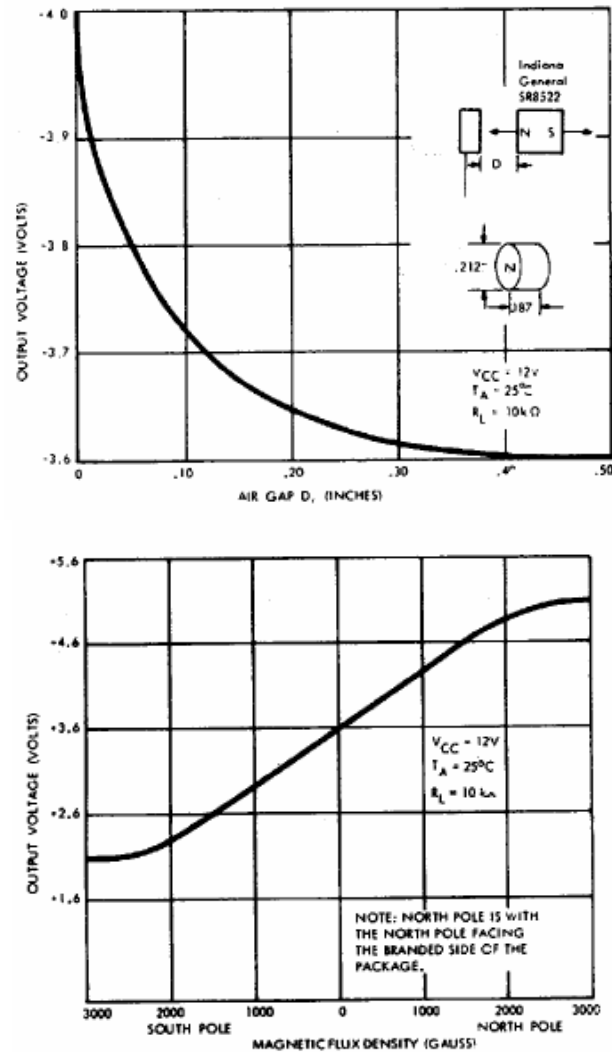


Figure I.8: Variation de la tension de sortie en fonction de la distance (1) et du flux (2) pour le capteur UGN3503

I.2.4 - Circuit de commande :

Le circuit de commande contient 2 éléments de bases qui sont le dsPIC et le Hacheur:

I.2.4.1 - Le dsPIC 30F3010 :

La famille des dsPIC est fabriquée par la firme Microchip, ce sont des DSC (Digital Signal Controller) c'est-à-dire des microcontrôleurs qui contiennent un cœur DSP (Digital Signal Processor), ce dernier permet au dsPIC la possibilité de réaliser rapidement multiplications, divisions, calculs trigonométriques... Cette puissance de calcul est d'un confort remarquable, surtout lorsqu'on programme en C.

En effet, il est facile d'écrire « $x = 45 * 658;$ » en C, or sur un microcontrôleur classique, cette seule ligne d'instruction pourrait prendre jusqu'à 1 seconde à s'exécuter. Sur un dsPIC, ça ne prendra que quelques μ secondes. Cette vitesse de calcul va assurer la commande en temps réel de notre système et justifie l'utilisation d'un DSC au lieu d'un microcontrôleur classique.

Le processeur d'un dsPIC est d'architecture Harvard (RISC) modifiée de 16-bits. Ce qui veut dire qu'il peut réaliser des opérations élémentaires (additions, soustractions) directement sur des entiers codés sur 16-bits. Par exemple, il peut additionner d'un seul coup 12352 avec 24896 (rappel : un entier sur 16 bits est compris entre 0 et 65535).

Un schéma bloc qui explique l'architecture interne du CPU peut être consulté à l'annexe.

Coté mémoire notre dsPIC 30F3010 contient 24 Ko de mémoire flash destinée à stocker le programme, plus 1 Ko de RAM et 1 Ko d'EEPROM pour stocker les variables qu'on ne veut pas qu'ils soient effacés lors de la mise hors tension du calculateur.

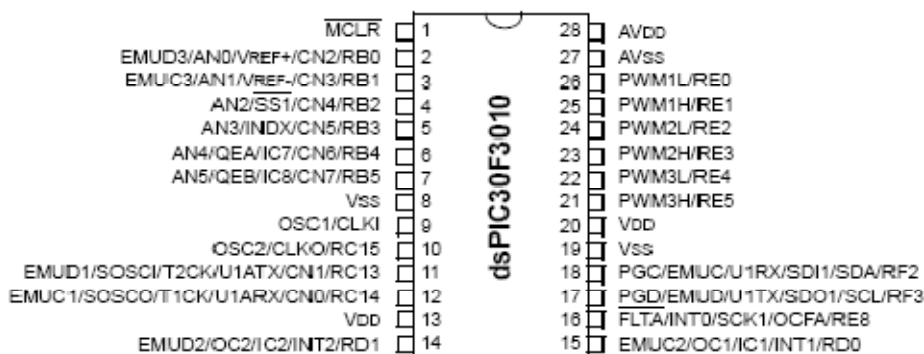


Figure I.9 : Brochage du dsPIC 30F3010 [5]

En ce qui concerne les périphériques, le dsPIC est assez fourni. Le premier périphérique est le module Entrée/Sortie, qui permet de lire où d'écrire des valeurs binaires sur 20 parmi les 28 pattes que possède le dsPIC. Il possède aussi 5 timers de 16 bits, sorte de chronomètres permettant de réaliser des temps d'attente, on d'activer un certain nombre d'actions de manière périodique.

Nous avons à notre disposition 3 modules PWM avec deux sorties pour chacun, permettant de générer des signaux en créneau à fréquence et rapport cyclique variable.

Il y a 6 entrées de convertisseur analogique-numérique 10-bits (CAN ou ADC) et 4 modules capture and compare (sert à mesurer des fréquences et à comparer des signaux logiques).

Enfin, il possède toute une gamme de modules dédiés à la communication avec l'extérieur (UART, I2C, SPI, CAN).

Le 30F3010 est peut être trouvé sous plusieurs packages (boitiers) : PDIP, SOIC. La version PDIP est la plus utilisée par les amateurs de l'électronique car elle permet une intégration plus facile du composant aux circuits imprimés. Elle existe encore une version plus professionnelle sous forme de boitier QFN avec 44 pattes.

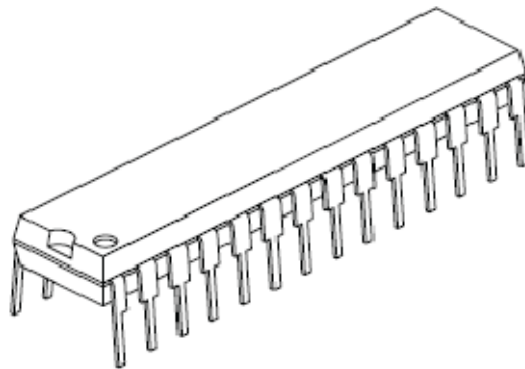


Figure I.10 : le dsPIC 30F3010 sous forme de boitier PDIP-28

I.2.4.2 - Double hacheur 4 quadrants L298 :

Le L298 est un circuit intégré qui comporte deux hacheurs 4 quadrants (pont en H), il contient 6 entrées et 4 sorties en plus des trois broches d'alimentation

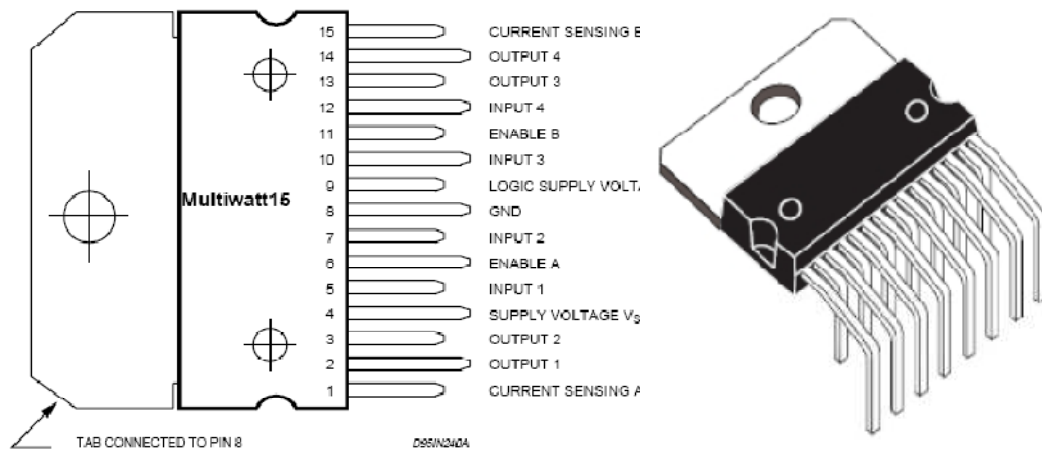


Figure I.11 : Brochage et forme commerciale du L298

Avec une tension d'alimentation maximale de 46 V le hacheur peut fournir jusqu'à 4 A de courant, en plus grâce aux entrées « enable » on peut activer ou désactiver le ponts A ou B selon le désir pour mieux comprendre le fonctionnement du composant voici le schéma électrique correspondant :

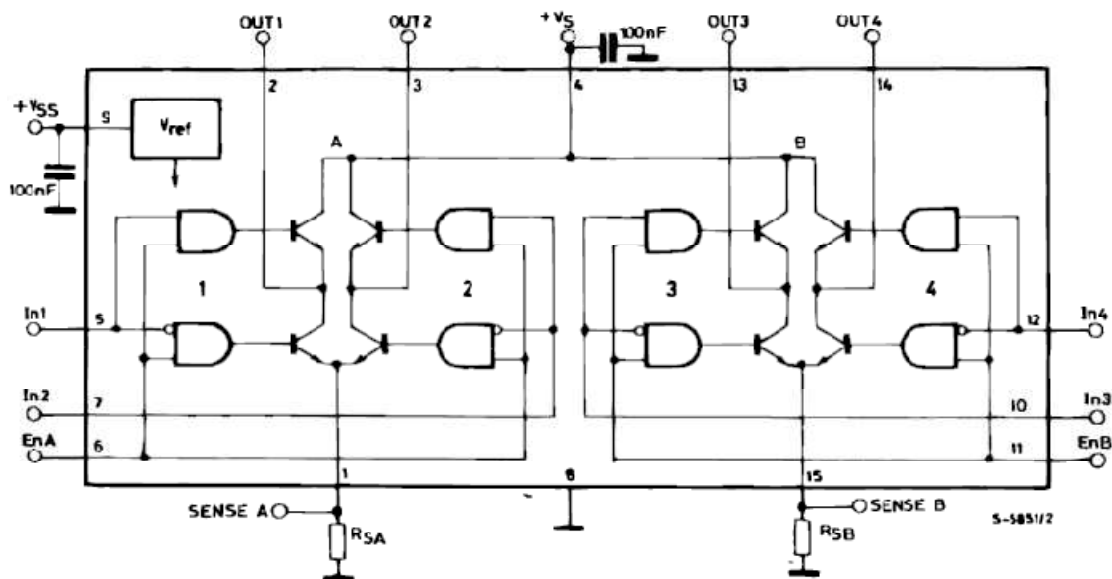


Figure I.12 : Schéma électrique interne du L298

I.3 - Conception et fabrication du circuit de commande :

I.3.1 - Schéma fonctionnel :

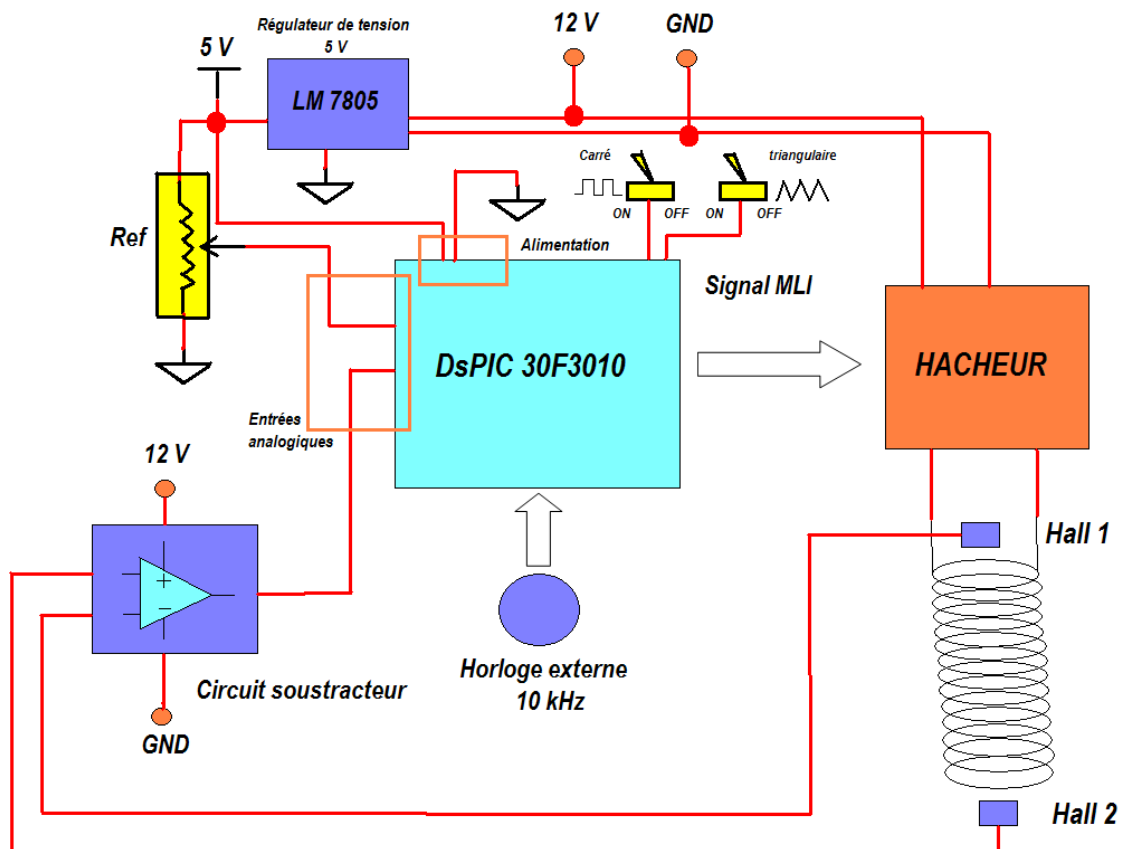


Figure I.13 : Schéma fonctionnel de la maquette.

Dans le schéma On remarque que tout le circuit de commande est basé sur le dsPIC car c'est lui qui va assurer la puissance de calcul nécessaire pour la régulation du système.

La bobine est piloté par le hacheur alimenté en 12 V, ce dernier étant lui-même piloté par le dsPIC grâce à ces sorties MLI (PWM). Le signal MLI qui sort dépend du type de la commande implémenté ainsi que de la consigne et de la position du corps lévité.

Le signal de référence (la consigne) est le signal qui donne la position désiré du corps en lévitation. Pour la régulation en position nous avons utilisé un potentiomètre qui va varier un signal analogique entre 0 et 5 V. Nous avons veillé aussi à ce que le système peut suivre des signaux de référence différents qui seront générés numériquement via programmation. Pour cela nous avons doté la carte de commande de deux contacts, chacun va déclencher la génération numérique d'un signal spécifié (carré, triangulaire)

par l'utilisation de sous programmes (voir chapitre IV : programmation). Le corps lévité va désormais suivre le nouveau signal de référence et la consigne analogique extérieur sera donc négligée.

Le retour d'information sur la position du corps lévité est assuré par les deux capteurs à effet Hall. Avant qu'ils soient injectés dans l'une des entées analogiques du dsPIC. Les signaux qui parviennent des deux capteurs sont passés dans un circuit soustracteur basé sur un ampli opérationnel : le LF356N. Ainsi les perturbations générées par la bobine seront énormément filtrées.

Comme tout processeur, pour fonctionner il doit être muni d'une horloge. Le dsPIC possède une horloge interne, mais pour une question de commodité, nous préférons utiliser une horloge externe. Ce choix est tout à fait discutable.

Coté alimentation notre circuit sera alimenté via un transformateur qui transforme la tension du réseau publique (220 V AC) en une tension de (0-12V) cette tension est utiliser pour alimenter le hacheur et l'ampli opérationnel.

D'autre part nous avons besoin d'une tension de 5V pour alimenter le dsPIC car ce dernier est fabriqué selon la technologie CMOS de 5V, pour cela on utilise un régulateur de tension : le LM7805 qui fournira une tension fixe de 5V à partir du 12V initiale. Cette tension sera à la base de la logique de toutes les entrées sorties reliées au dsPIC.

I.3.2 - Fabrication du circuit:

I.3.2.1 - Présentation du logiciel de développement utilisé :

Pour la fabrication du circuit de commande nous avons utilisé la gamme de logiciels nommé « NI Circuit Design Suite 10.0 » fabriqué par la firme *National Instrument / Electronic Workbench group (EWB)*. Cette gamme de logiciel permet une assistance totale pour la construction de circuits imprimés, d'autre logiciels existent dans le marché et peuvent être utilisés comme : Arres, OrCAD, Allegro...

La gamme de logiciels du *National Instrument* comporte deux outils de développment :

Multisim : c'est un logiciel de capture schématique et de simulation de circuits électroniques, grâce à une bibliothèque qui comporte les composants électroniques les plus courants il permet de créer des schémas électriques et de les préparer pour être construits sous formes de circuits imprimés, il donne aussi une assistance à la création de

nouveaux composants pour enrichir sa bibliothèque (car bien évidemment on ne peut pas trouver dedans tout les composants qui existent dans le marché).

Ultiboard : il utilise les schémas fournis par Multisim pour la création des schémas PCB, ces schémas (layout) sont utilisés pour la fabrication de circuits imprimés. Pour cela Ultiboard propose une multitude de tâches automatisées telles que le placement des composants, le traçage des pistes, l'optimisation du circuit... il peut effectuer aussi quelques opérations CAD mécaniques de base pour préparer les circuits à la fabrication finale.

I.3.2.2 - Schémas électriques sous Multisim :

I.3.2.2.1- Alimentation :

Le circuit est alimenté par un transformateur qui fournit une tension fixe de 12V. Cette tension sera utilisée pour alimenter le hacheur par son entrée VS (pin 4) en utilisant une capacité de 100 nF.

Pour alimenter le DsPIC on a besoin d'une tension de 5V, celle-ci est obtenue en utilisant un régulateur de tension qui est alimenté par la tension de 12V et utilisé avec deux capacités, une en entrée (100 nF) et l'autre en sortie (1 μ F). Le DsPIC est alimenté via ses pattes VDD et VSS, la première correspond au niveau logique haut (5V) et la deuxième au niveau logique bas (la terre). Des capacités de protection sont employées et reliées à la terre au niveau de la patte VDD (deux capacités en parallèle : 1 μ F et 100nF). En remarque que plusieurs pattes VDD et VSS existent sur LE dspic cela a pour but de permettre plusieurs configurations de brochage selon le besoin.

Même chose pour le hacheur, on doit lui fournir une tension de 5V pour établir les niveaux logiques. Cette tension est injectée par la patte VSS. Une capacité de protection est encore utilisée.

Pour qu'un dsPIC fonctionne correctement, il faut relier la patte MCLR (Master Clear) au 5V par une résistance de 10kohm. A ce stade, un programme peut tourner. Nous avons placé un bouton poussoir pour permettre la réinitialisation du programme à n'importe quel moment.

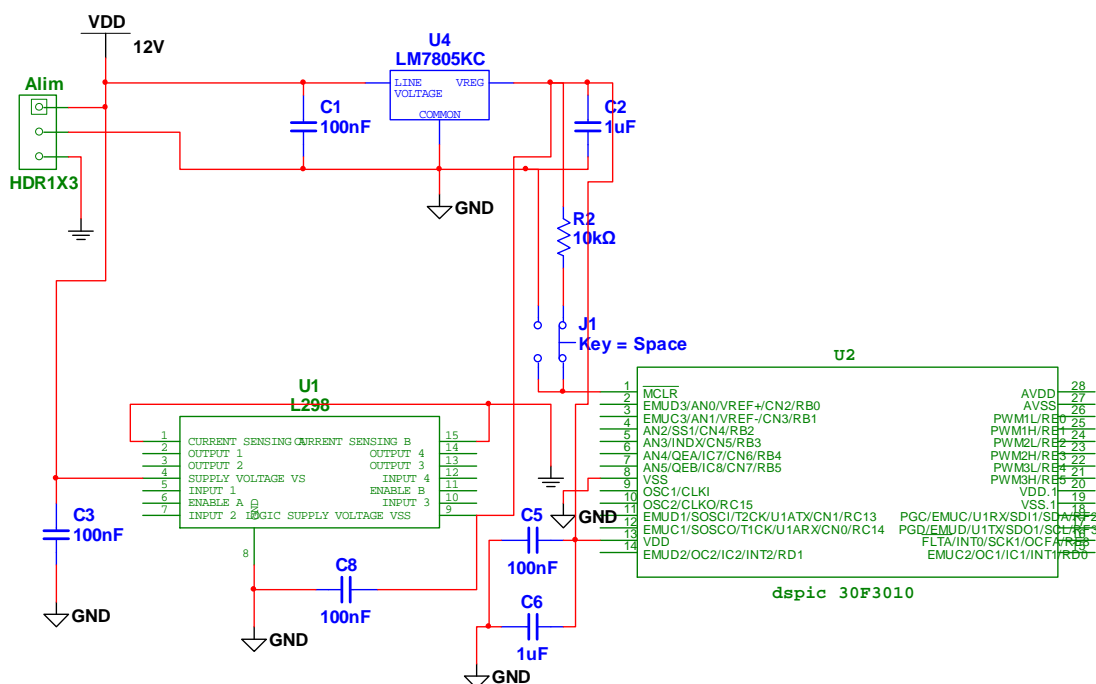


Figure I.14 : Schéma électrique d'alimentation

La terre logique (5V) et celle de puissance (12V) peuvent être couplées ou laissées indépendantes l'une de l'autre.

I.3.2.2.2 - Circuit de l'horloge externe :

Comme nous l'avons déjà mentionné nous avons choisi d'utiliser une horloge externe. L'horloge externe est un cristal de quartz (matériau piézo-électrique) de 10MHz (fréquence d'oscillation propre du matériau). Pour fonctionner correctement, il doit être relié aux entrées OSC1 et OSC2 du dsPIC, et muni de deux condensateurs de 22pF (voir illustration du chapitre II).

Il fournit une horloge à 10MHz qu'il est possible de multiplier à l'aide d'un module dit de PLL (voir la partie programmation). Nous choisissons de la multiplier par 8 (PLL x8). Or, pour exécuter une instruction, le processeur a besoin de 4 cycles d'horloge ; ce qui divise par 4 la fréquence de fonctionnement. On arrive ainsi à une fréquence de fonctionnement finale de 20MHz, soit 200ns / instruction.

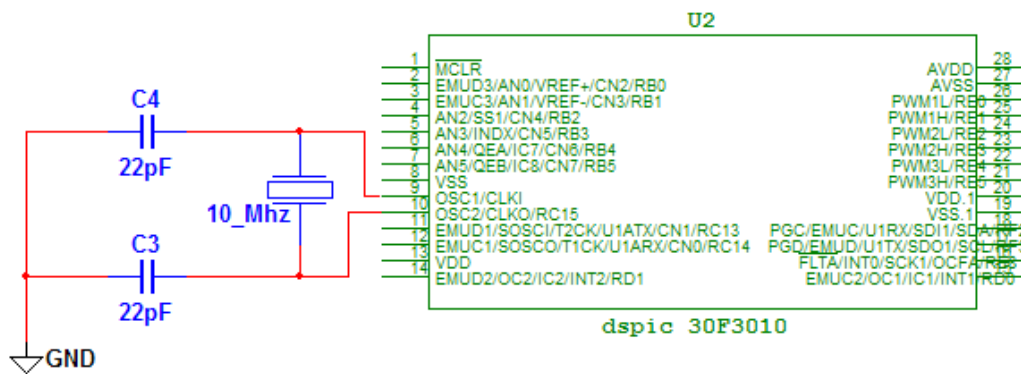


Figure I.15 : Schéma du circuit de l’horloge externe

II.3.2.2.3 - Branchement des entrées :

Les entrées analogiques sont le signal de consigne établi par le potentiomètre et le signal de retour qui vient des capteurs de position. Ces signaux sont injectés par les pattes AN2, AN3 qui peuvent être configurés comme entrées analogiques (voir partie programmation).

Les signaux qui parviennent des capteurs à effet Hall sont passés d’abord dans un circuit soustracteur. C’est un circuit amplificateur différentiel de base pour les amplis opérationnels.

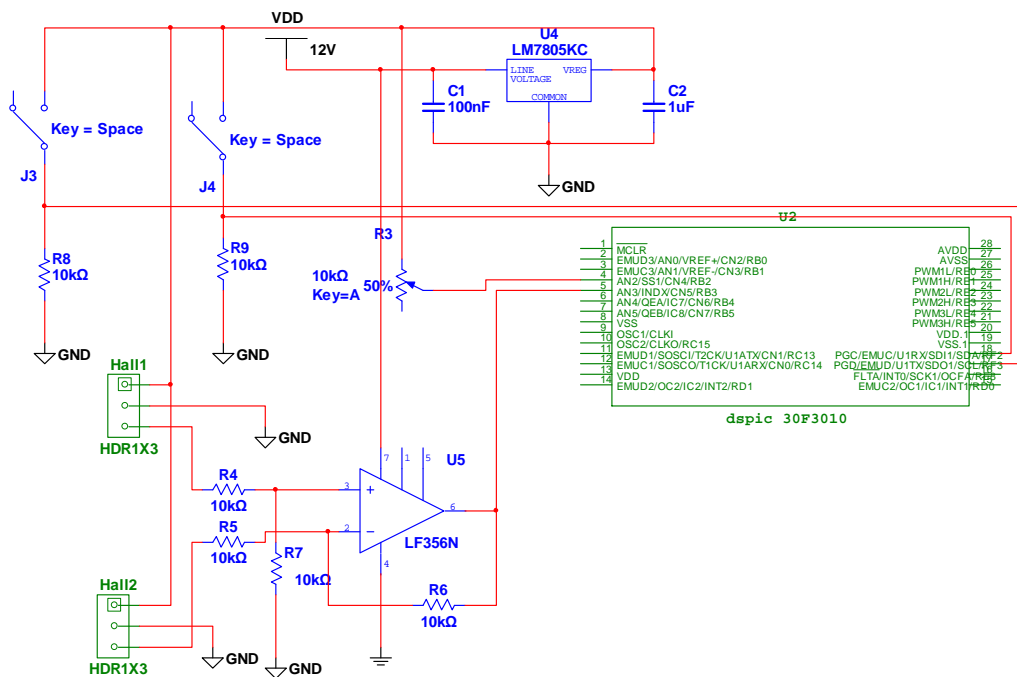


Figure I.16 : Schéma du branchement des entrées

Ils existent encore deux entrées logiques reliées aux pattes (RF2, RF3) qui seront activées par deux contacts, ces entrées seront responsables du déclenchement des signaux de consigne carrée ou triangulaire.

I.3.2.2.4 - Branchement des sorties :

Une LED est connectée à la patte RB0 en série avec une résistance de 1K ohm, son rôle sera d'indiquer si le processus de régulation est en cours de déroulement ou pas.

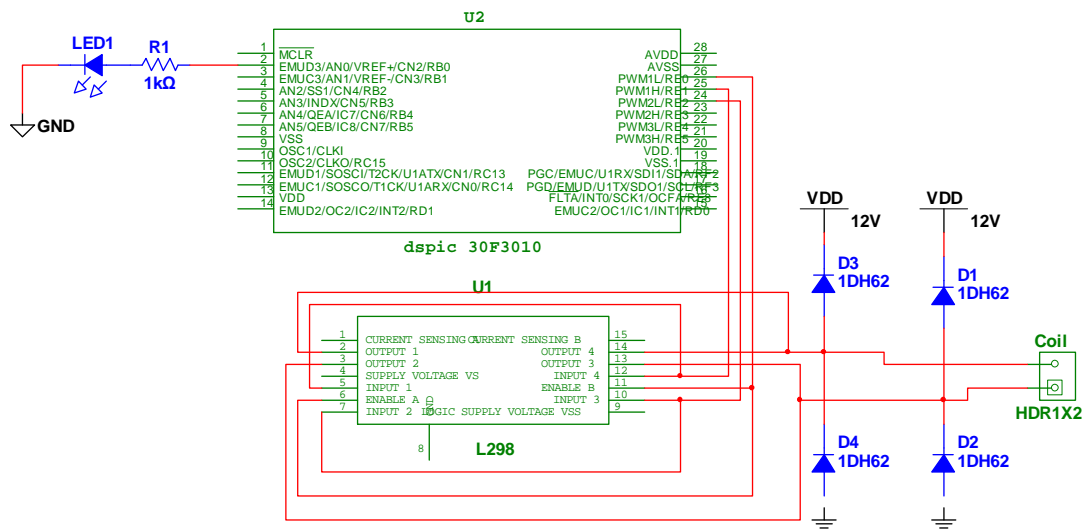
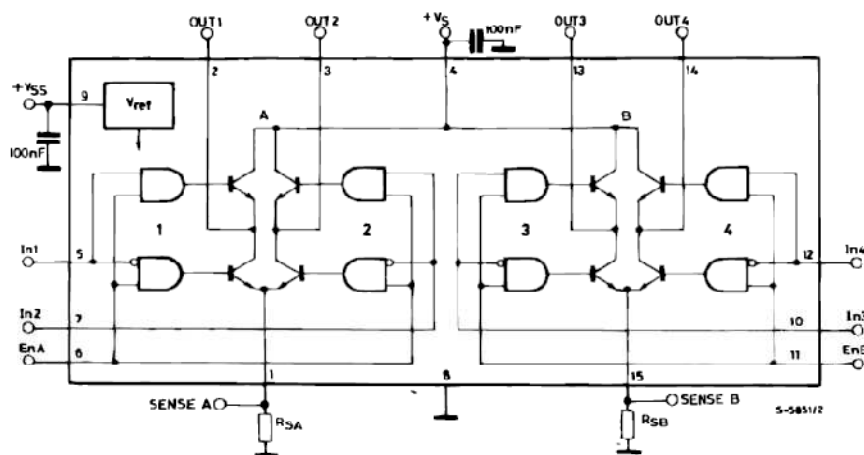


Figure I.17 : Schéma du branchement des entrées

Pour le branchement du hacheur, nous avons utilisé les deux ponts en H disponibles en les branchant parallèlement de sorte à ce qu'ils travaillent comme un seul hacheur qui supporte des courants plus puissants. Pour cela il faut s'assurer de brancher les entrées 1 et 4 ainsi que 1 et 3 en parallèle et faire la même chose pour les sorties, pour mieux comprendre voir le schéma.



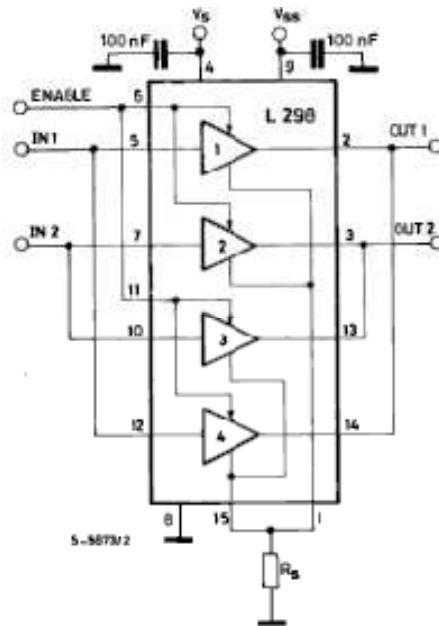


Figure I.18 : Schémas électriques du branchement en parallèle du double hacheur 4 quadrants

La sortie RE0 sera branchée aux pattes (Enable A, Enable B) du hacheur, elle sera responsable de l'activation / désactivation de l'hacheur.

Du fait que le L298 ne dispose pas de diodes de roue libre intégrées on doit brancher ces derniers à l'extérieur au niveau des sorties du hacheur. La bobine est finalement branchée elle est représentée sur le schéma par un bornier (Coil).

Nous avons ajouté un module appelé MAX3235E, qui va permettre d'effectuer la liaison RS 232 avec un micro-ordinateur, pour plus de détails voir la partie perspectives et possibilités d'extension.

Finalement, le schéma complet du circuit de commande sera comme suit :

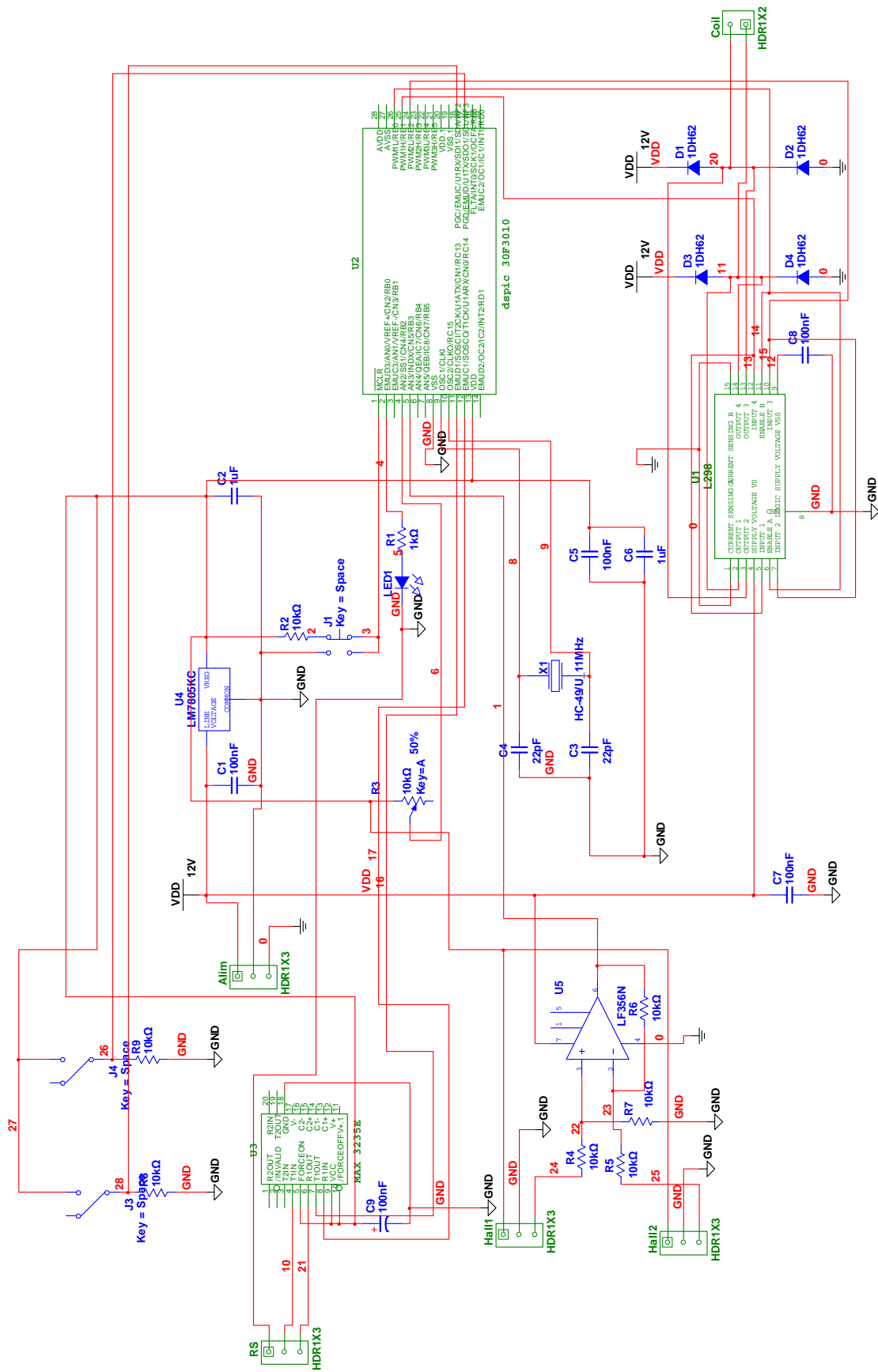


Figure 1.19 : Schéma électrique complet du circuit de commande

I.3.2.3 - Schéma PCB sous Ultiboard :

Multisim permet de transférer le schéma précédent vers le logiciel Ultiboard pour fabriquer le circuit imprimé. Il faut veiller à ce que les pistes du circuit imprimé soient suffisamment larges et que les composants soient espacés pour ne pas trouver de problèmes lors du soudage des composants. Tout de même il ne faut pas les espacer trop pour que la carte de commande ne soit pas trop grande, ce qui permet d'optimiser les coûts de fabrications.

Ultiboard possède une bibliothèque qui contient les empreintes des composants électronique les plus courants avec leurs dimensions réelles ce qui assure qu'ils seront soudés sur la carte sans difficulté.

L'optimisation des pistes est effectuée automatiquement par le logiciel.

A cause du nombre important de composants et donc de connections qui existent. Il s'est avéré indispensable que le circuit comprend des pistes sur deux faces. Voici les deux faces dites coté cuivre et coté composants.

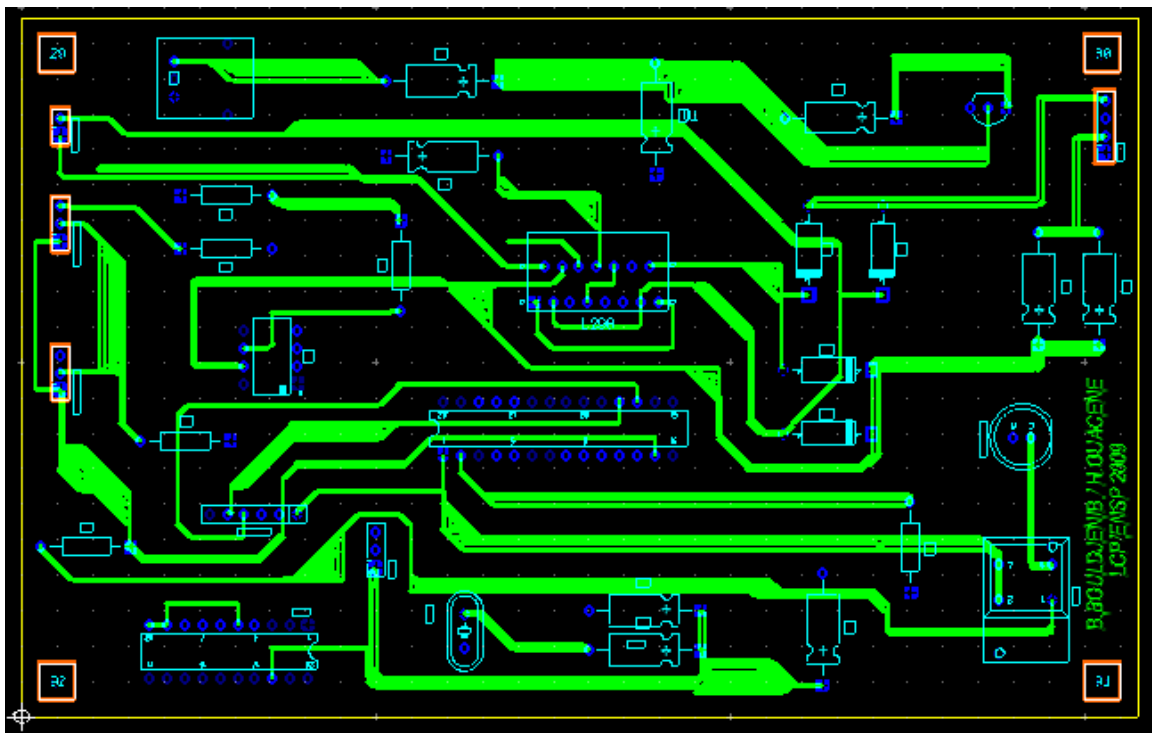


Figure I.20 : Schéma du coté composant

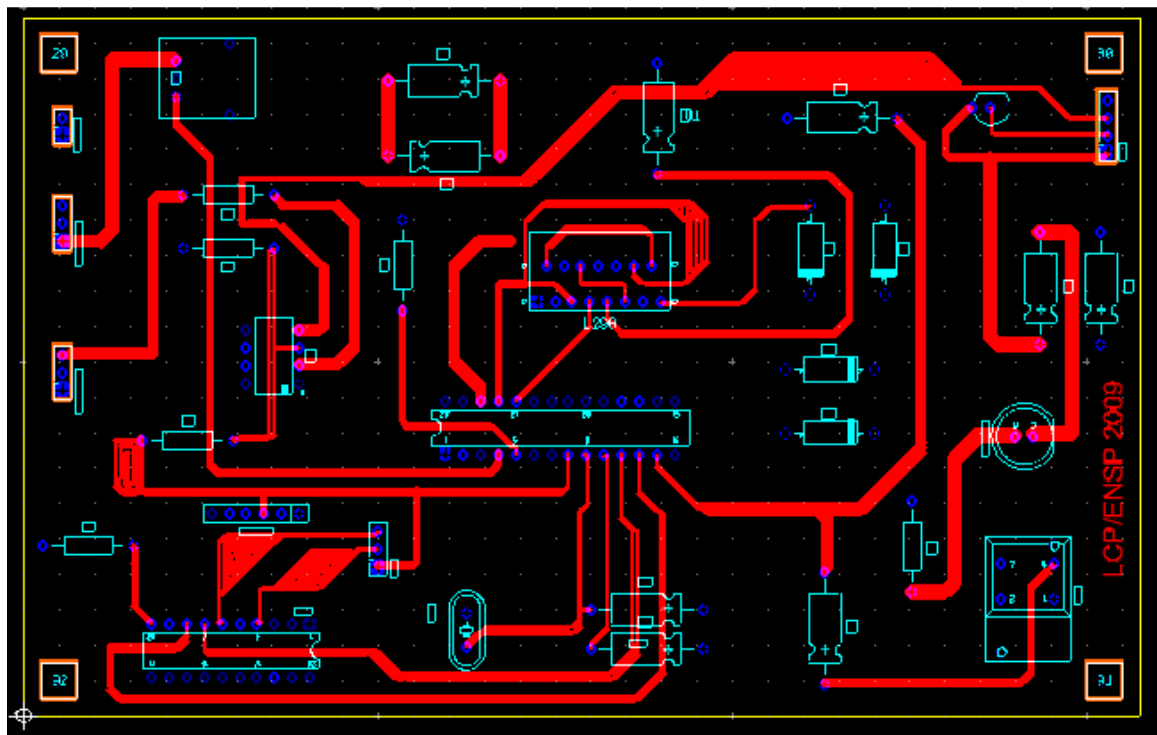


Figure I.21 : schéma du coté cuivre

Nous avons choisit que les deux boutons poussoirs qui déclenchent la suite de consigne .soit sur un module indépendant pour qu'ils soient installées à proximité de l'opérateur. Voici le circuit correspondant.

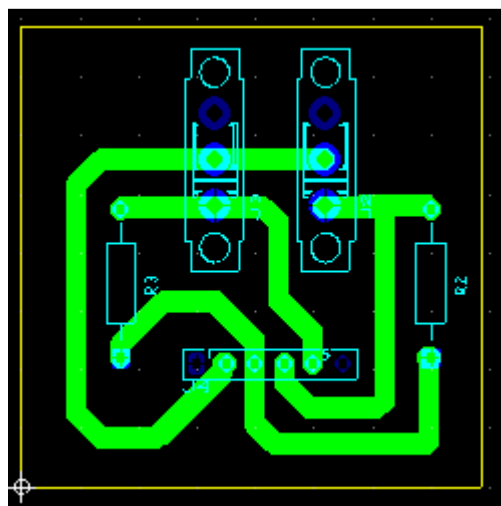
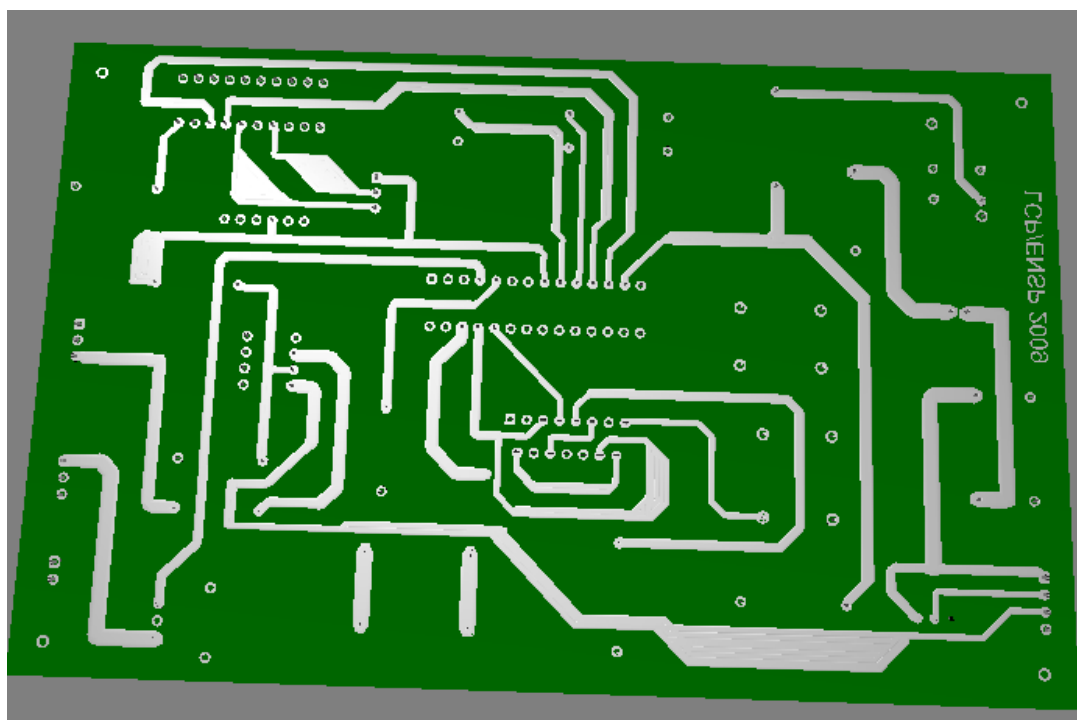
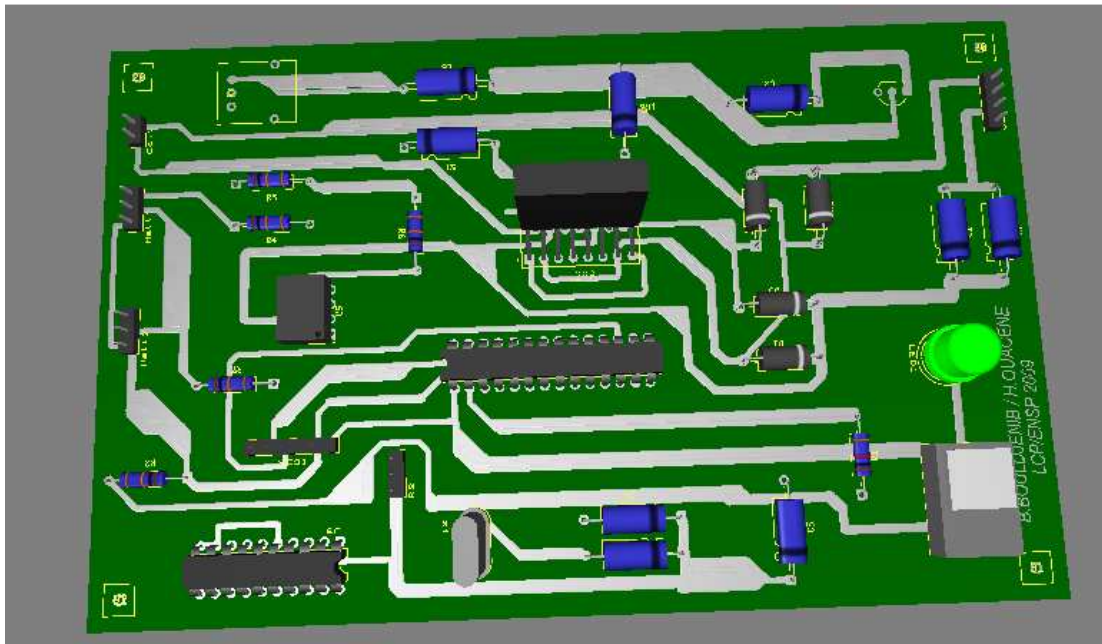


Figure I.22 : Schéma du circuit supplémentaire contenant les contacts.

D'autre part, Ultiboard possède une autre bibliothèque qui lui permet de créer des vues en 3 dimensions des circuits imprimés, ce qui permet de voir la disposition des composants avant que le circuit soit construit.

Voici les vues en 3D du circuit de commande ainsi que le module supplémentaire qui contient les contacts :



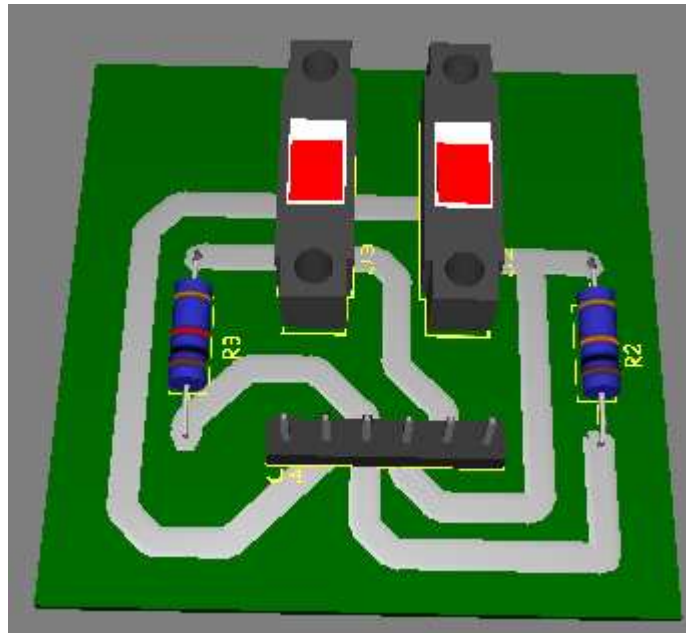


Figure I.23 : Représentation en 3D du circuit de commande (1, 2) et du module supplémentaire (3)

I.4 - Conclusion :

Dans ce chapitre nous avons entamé la présentation générale de la maquette expérimentale en commençant par les constituants du système à lévitation magnétique en donnant les choix et les possibilités technologiques qui existent lors de la construction de tels systèmes, puis nous avons parler du circuit de commande en donnant les schémas électriques correspondants en détail et en terminant par la fabrication du circuit imprimé.

Nous avons pu vérifier aussi l'importance des logiciels de développement des circuits électriques qui donnent au concepteur une assistance totale dans toutes les étapes de son projet.

CHAPITRE II

MODELISATION DU SYSTEME

CHAPITRE II : Modélisation du système

II.1 – Introduction :

Le problème de commande exige la recherche du modèle mathématique du système à commander, donc une phase de modélisation est nécessaire pour permettre l'étude en simulation.

La modélisation consiste à représenter les aspects importants du système en décrivant les relations entre ses différentes grandeurs par des équations mathématiques. L'ensemble de ces relations constitue le modèle de ce système.

Dans ce chapitre, nous commençons par une modélisation détaillée du système présenté dans le chapitre précédent, on présentera ensuite le modèle linéarisé et on terminera par une simulation en boucle ouverte du système non linéaire.

II.2 – Modélisation :

Le système à modéliser est composé d'un électroaimant, d'un objet avec un petit aimant permanent collé au-dessus de ce dernier. Le système peut être réduit à l'interaction entre un enroulement avec un noyau ferromagnétique et un aimant permanent.

Le modèle de système est établi sur l'équation du mouvement de la boule et les caractéristiques électriques de l'électro-aimant [6].

L'équation du mouvement de la boule est

$$m \frac{d^2 z(t)}{dt^2} = mg - F(i, z) \quad (\text{II.1})$$

z est la distance séparant l'électroaimant et l'aimant permanent

II.2.1 - Calcul de l'inductance du système :

L'inductance de l'enroulement est une fonction de la distance de séparation et également de la fréquence de la tension appliquée. La dépendance à l'égard de la séparation peut être approximée par [1] :

$$L(z) = L_1 + \frac{L_0}{1 + \left(\frac{z}{a}\right)} \quad (\text{II. 2})$$

Tel que $L_1 = L(\infty)$ (c-à-d l'inductance de l'électroaimant seul, en absence de l'aimant permanent)

$L_0 = L(0) - L(\infty)$ et z la distance séparant l'électroaimant et l'aimant permanent

a : est une constante caractéristique positive inconnue.

II.2.2 - La force magnétique d'interaction : [7]

Le calcul de la force d'attraction se fait en considérant qu'une variation de l'énergie magnétique dW_{mag} peut être causée par un apport d'énergie électrique dW_{elec} et/ou un apport d'énergie mécanique dW_{mec} [2]. La conservation de l'énergie permet alors d'écrire :

$$dW_{mag} = dW_{elec} + dW_{mec} \quad (\text{II. 3})$$

La variation d'énergie mécanique vaut :

$$dW_{mag} = d\left(\frac{1}{2} \cdot L \cdot i^2\right) = L \cdot i \cdot di + \frac{1}{2} \cdot i^2 \cdot dL \quad (\text{II. 4})$$

Les apports d'énergies mécanique et électrique valent :

$$dW_{mec} = -F \cdot dz \quad (\text{II. 5})$$

$$dW_{elec} = u(t) \cdot i(t) \cdot dt \quad (\text{II. 6})$$

En tenant compte de :

- La loi de Lenz

$$u(t) = \frac{d\Psi}{dt} \quad (\text{II. 7})$$

- La loi définissant l'inductance

$$\Psi = L \cdot i \quad (\text{II. 8})$$

- La variation de l'énergie magnétique

$$dW_{mag} = L \cdot i \cdot di + \frac{1}{2} \cdot i^2 \cdot dL \quad (\text{II. 9})$$

L'équation (II.6) devient :

$$\begin{aligned} dW_{elec} &= i \cdot (u(t) \cdot dt) = i \cdot d\Psi \\ &= i \cdot (L \cdot di + i \cdot dL) \\ &= L \cdot i \cdot di + i^2 \cdot dL \\ &= dW_{mag} + \frac{1}{2} \cdot i^2 \cdot dL \end{aligned}$$

D'où

$$dW_{elec} = dW_{mag} - \frac{1}{2} \cdot i^2 \cdot dL \quad (\text{II. 10})$$

Considérant les équations (II.3), (II.5) et (II.10), on voit qu'on a :

$$F \cdot dz = -dW_{mec} = \frac{1}{2} \cdot i^2 \cdot dL \quad (\text{II. 11})$$

On en déduit donc que la force dépend du carré du courant et de la manière dont l'inductance varie avec le déplacement x de l'aimant permanent collé sur l'objet :

$$F(i, z) = -\frac{dW_{mec}}{dx} = \frac{1}{2} \cdot i^2 \cdot \frac{dL(z)}{dz} \quad (\text{II. 12})$$

Se souvenant de l'expression de l'inductance $L(z)$ (équation (II.2)), il vient finalement :

$$F(i, z) = -\frac{a}{2} \cdot L_0 \cdot \left(\frac{i}{a+z} \right)^2 \quad (\text{II. 13})$$

II.3 - Les équations d'état du système : [8]

Pour le calcul de la fonction de transfert, on doit écrire les équations régissant le système, et d'après les équations (II. 1) et (II.13) on a :

$$m \frac{d^2 z(t)}{dt^2} = mg - F(i, z)$$

$$F(i, z) = -\frac{a}{2} \cdot L_0 \cdot \left(\frac{i}{a+z} \right)^2$$

Et Selon la loi de Kirchhoff, on a l'équation suivante :

$$u(t) = R \cdot i(t) + L \cdot \frac{di(t)}{dt} \quad (\text{II. 14})$$

Telle que $u(t)$ est la tension appliquée, L est l'inductance de l'enroulement, R est la résistance de l'enroulement, m est la masse de l'objet, g est l'accélération de la gravité, F est la force magnétique de commande.

Quand la pesanteur de la boule est égale à la force générée par l'électro-aimant,

On a :

$$m \cdot g = F(i_0, z_0) = -\frac{a}{2} \cdot L_0 \cdot \left(\frac{i_0}{a+z_0} \right)^2 \quad (\text{II. 15})$$

Définissons le vecteur d'état comme :

$$x^T = (x_1, x_2, x_3) = \left(i, z, \frac{dz}{dt} \right) \quad (\text{II. 16})$$

les équations d'état du système sont :

$$\begin{cases} \dot{x} = f(x) + g(x) \cdot u \\ y = h(x) = x_2 \end{cases} \quad (\text{II. 17})$$

Où

$$f(x) = (f_1, f_2, f_3) = \begin{pmatrix} \frac{-R \cdot x_1}{L} \\ x_3 \\ g + \frac{a \cdot L_0}{2 \cdot m} \cdot \frac{x_1^2}{a^2 + 2 \cdot a \cdot x_2 + x_2^2} \end{pmatrix}$$

$$g(x) = \begin{pmatrix} \frac{1}{L} \\ 0 \\ 0 \end{pmatrix}$$

Sachant que 'u' est la tension de commande, 'x1' le courant de l'électroaimant, 'x2' la position de l'objet et 'x3' la vitesse de ce dernier.

II.4 - Calcul des paramètres :

Pour synthétiser la commande on doit calculer les valeurs des différents paramètres :

- La résistance de l'enroulement est $R = 22\Omega$
- Concernant la self de l'électroaimant .Cela est fait à partir des mesures en alternatif, ils sont comme suit :

U (V)	I (A)	Z (m)
12.4	0.285	∞
12.4	0.295	0
12.4	0.289	0.005

Tableau II.1 : les valeurs expérimentales

Sachant que la fréquence est de 50 Hertz et :

$$u = \sqrt{(R^2 + L^2 \cdot \omega^2)} \cdot i \quad (\text{II. 18})$$

Avec l'application numérique, on aura comme valeur :

$$L_1 = 0.1183 \text{ H} , L(0) = 0.1127 \text{ H} , L(0.005) = 0.1158$$

Le calcul de la distance caractéristique a se fait à partir de l'équation (II.1) suivante :

$$L(z) = L_1 + \frac{L_0}{1 + \left(\frac{z}{a}\right)}$$

On faisant le calcul on aura :

$$a = 0.004 \text{ m}$$

Donc les paramètres peuvent être donnés dans le tableau suivant :

M	5*10⁻³	[kg]
G	09.810	[m/s ²]
a	0.0040	[m]
L(∞)	0.1183	[H]
L(0)	0.1127	[H]
R	22.000	[Ω]

Tableau II.2 : Les valeurs des paramètres du système

II.5 - Commandabilité et observabilité

La commandabilité s'intéresse à l'existence d'un signal de commande admissible pour le système permettant de transférer le vecteur d'état (ou de sortie) d'une position quelconque à une position désirée.

Pour ce faire on calcule le rang de la matrice suivante :

$$C_o = [ad_f^0 g(x) \quad -ad_f^1 g(x) \quad ad_f^2 g(x)] \quad (\text{II. 19})$$

Ou $ad_f^1 g(x)$ est le produit de Lie défini comme suit :

$$ad_f^0 g(x) = g(x)$$

$$ad_f^1 g(x) = [f, g] = \frac{\partial g(x)}{\partial x} f - \frac{\partial f(x)}{\partial x} g$$

$$ad_f^{k+1} g(x) = [f, ad_f^k g(x)]$$

Or :

$$f(x) = (f_1, f_2, f_3) = \begin{pmatrix} -186 * x_1 \\ x_3 \\ 9.81 - 0.0024 * \frac{x_1^2}{(0.004 + x_2)^2} \end{pmatrix}$$

$$g(x) = \begin{pmatrix} 8.45 \\ 0 \\ 0 \end{pmatrix}$$

$$C_o = \begin{bmatrix} 8.45 & 1\ 571.7 & 292\ 336.2 \\ 0 & 0 & -0.83 * \frac{x_1}{(0.004 + x_2)^2} \\ 0 & -0.83 * \frac{x_1}{(0.004 + x_2)^2} & -1309.67 * \frac{x_1}{(0.004 + x_2)^2} \end{bmatrix}$$

Comme on peut le voir le système est de rang égal à 3, Donc le système est complètement faiblement commandable

L'observabilité est un concept permettant de savoir est ce que, il est possible de restitué la trajectoire d'état à partir des mesure disponible, il existe plusieurs type d'observabilité mais on s'intéresse à l'observabilité faible.

Soit :

$$O_b = \frac{\partial}{\partial x} \begin{bmatrix} h(x) \\ L_f h(x) \\ L_f^2 h(x) \end{bmatrix} \quad (\text{II. 20})$$

Tel que $h(x) = x_2$ et $L_f h(x)$ est la drivée de lie définie comme suit :

$$L_f h(x) = \frac{\partial}{\partial x} (h(x) * f(x))$$

$$L_f^{n+1}(h(x)) = \frac{\partial}{\partial x} (L_f^n(h(x)) * f(x))$$

Donc :

$$O_b = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0.0045 * \frac{x_1}{(0.004 + x_2)^2} & -0.0045 * \frac{x_1^2}{(a + x_2)^3} & 0 \end{bmatrix}$$

Comme on peut le voir le système est de rang égal à 3. Donc le système est complètement faiblement observable.

II.6 - Le modèle linéaire :

On réalité, tous les systèmes sont non linéaire. Il existe des limitations, des saturations ou d'autre type de non linéarités. Le traitement des systèmes non linéaire est très compliqué. On doit essaie de linéariser les systèmes non linéaires [9]

On utilise le développement Taylor :

$$\dot{x} = f(x_s) + \left. \frac{\delta f}{\delta x} \right|_{x_s} (x - x_s) \quad (II.21)$$

Le point d'équilibre autour duquel on va linéariser est le point atteint par le système en régime permanent, donc on aura :

$$\dot{x}_s = f(x_s) \quad (II.22)$$

En conséquence :

$$\dot{x} = \left. \frac{\delta f}{\delta x} \right|_{x_s, \theta_s} (x - x_s) \quad (II.23)$$

$$\text{Donc : } \begin{bmatrix} \frac{d(x_1 - x_{1s})}{dt} \\ \frac{d(x_2 - x_{2s})}{dt} \\ \frac{d(x_3 - x_{3s})}{dt} \end{bmatrix} = \begin{bmatrix} \left. \frac{\partial f_1}{\partial x_1} \right|_{x_1, x_2, x_3, u} & \left. \frac{\partial f_1}{\partial x_2} \right|_{x_1, x_2, x_3, u} & \left. \frac{\partial f_1}{\partial x_3} \right|_{x_1, x_2, x_3, u} \\ \left. \frac{\partial f_2}{\partial x_1} \right|_{x_1, x_2, x_3, u} & \left. \frac{\partial f_2}{\partial x_2} \right|_{x_1, x_2, x_3, u} & \left. \frac{\partial f_2}{\partial x_3} \right|_{x_1, x_2, x_3, u} \\ \left. \frac{\partial f_3}{\partial x_1} \right|_{x_1, x_2, x_3, u} & \left. \frac{\partial f_3}{\partial x_2} \right|_{x_1, x_2, x_3, u} & \left. \frac{\partial f_3}{\partial x_3} \right|_{x_1, x_2, x_3, u} \end{bmatrix} * \begin{bmatrix} (x_1 - x_{1s}) \\ (x_2 - x_{2s}) \\ (x_3 - x_{3s}) \end{bmatrix} +$$

$$\begin{bmatrix} \left. \frac{\partial f_1}{\partial u} \right|_{x_1, x_2, x_3, u} \\ \left. \frac{\partial f_2}{\partial u} \right|_{x_1, x_2, x_3, u} \\ \left. \frac{\partial f_3}{\partial u} \right|_{x_1, x_2, x_3, u} \end{bmatrix} * (u - u_s)$$

Soit :

$$A = \begin{bmatrix} \left. \frac{\partial f_1}{\partial x_1} \right|_{x_1, x_2, x_3, u} & \left. \frac{\partial f_1}{\partial x_2} \right|_{x_1, x_2, x_3, u} & \left. \frac{\partial f_1}{\partial x_3} \right|_{x_1, x_2, x_3, u} \\ \left. \frac{\partial f_2}{\partial x_1} \right|_{x_1, x_2, x_3, u} & \left. \frac{\partial f_2}{\partial x_2} \right|_{x_1, x_2, x_3, u} & \left. \frac{\partial f_2}{\partial x_3} \right|_{x_1, x_2, x_3, u} \\ \left. \frac{\partial f_3}{\partial x_1} \right|_{x_1, x_2, x_3, u} & \left. \frac{\partial f_3}{\partial x_2} \right|_{x_1, x_2, x_3, u} & \left. \frac{\partial f_3}{\partial x_3} \right|_{x_1, x_2, x_3, u} \end{bmatrix}$$

$$B = \begin{bmatrix} \left. \frac{\partial f_1}{\partial u} \right|_{x_1, x_2, x_3, u} \\ \left. \frac{\partial f_2}{\partial u} \right|_{x_1, x_2, x_3, u} \\ \left. \frac{\partial f_3}{\partial u} \right|_{x_1, x_2, x_3, u} \end{bmatrix}$$

On aura les matrices :

$$A = \begin{bmatrix} -\frac{R}{L} & 0 & 0 \\ 0 & 0 & 0 \\ \frac{a \cdot L_0}{m} \cdot \frac{x_1}{(a + x_2)^2} & -\frac{a \cdot L_0}{m} \cdot \frac{x_1^2}{(a + x_2)^3} & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 \\ \frac{1}{L} \\ 0 \\ 0 \end{bmatrix}$$

On considère le point d'équilibre suivant :

$$x = \begin{bmatrix} 0.750 \\ 0.019 \\ 0 \end{bmatrix}$$

On aura :

$$A = \begin{bmatrix} -185.9679 & 0 & 0 \\ 0 & 0 & 1 \\ -6.3516 & 207.1176 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 8.4531 \\ 0 \\ 0 \end{bmatrix}$$

Et comme : $y = x_2$

Donc :

$$C = [0 \quad 1 \quad 0]$$

Sachant que :

$$\begin{cases} \dot{x} = A \cdot x + B \cdot u \\ y = C \cdot x \end{cases} \quad (\text{II. 24})$$

Calcul de la fonction de transfert :

On peut calculer la fonction de transfert par la formule suivante :

$$G(s) = C \cdot [s \cdot I_4 - A]^{-1} \cdot B$$

Tel que I_3 est défini comme la matrice identité.

$$I_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Après calcul on aura la fonction de transfert suivante :

$$G(s) = -53.6907 \frac{1}{(s + 185.9679) \cdot (s^2 - 207.1176)}$$

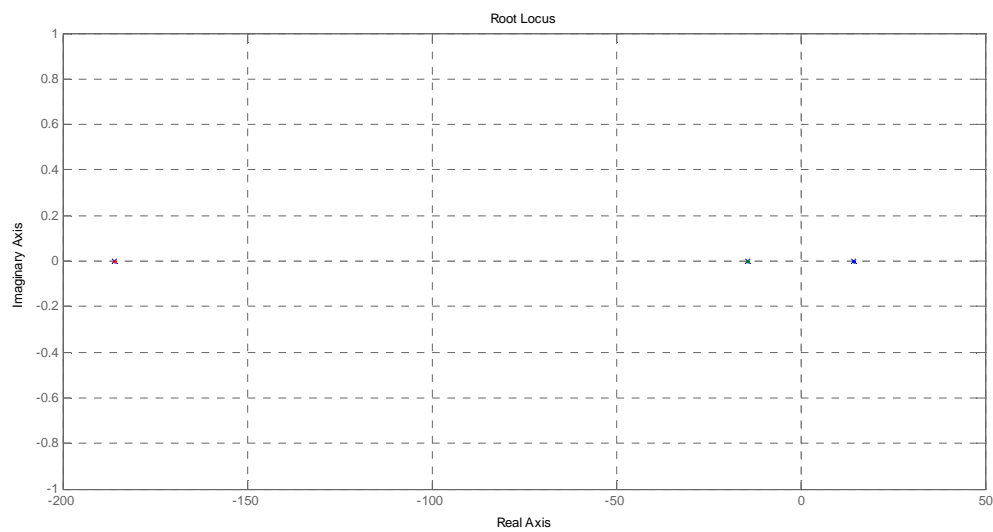


Figure II.1 : Lieux des pôles pour le système

Comme on peut voir que le système a 3 pôles et un d'entre eux est a partie réelle positive, donc le système est instable.

II.7 - Simulations en boucle ouverte :

Avant de faire la commande, soit la simulation du système en boucle ouverte :

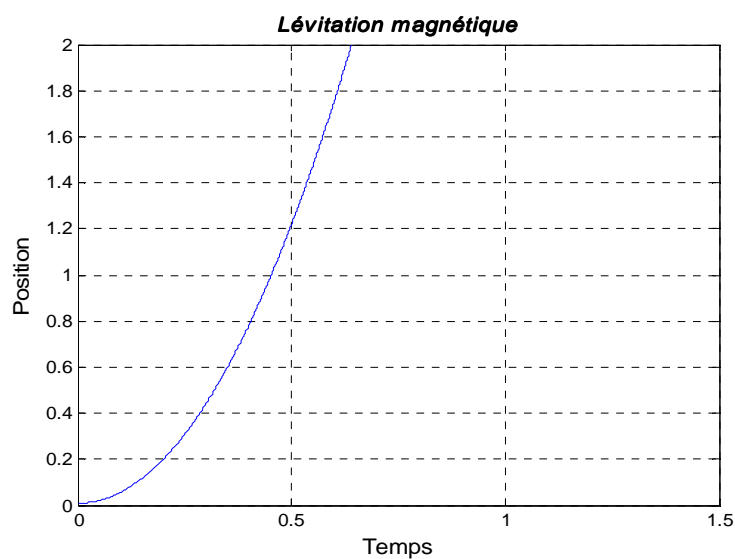


Figure II.2 : Résultat de simulation du système en boucle ouvert sans excitation

Comme on peut le voir le système est instable, tel que l'objet va tomber par terre sous l'effet de la pesanteur et en absence de la force magnétique.

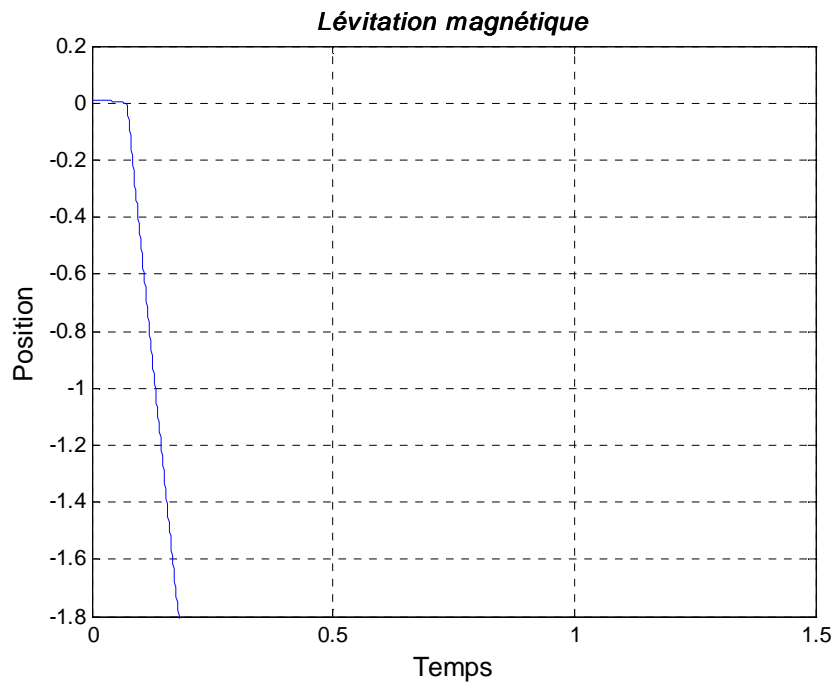


Figure II.3 : Simulation en boucle ouverte avec excitation.

Par contre ici, l'objet va se coller sur l'électroaimant sous l'effet de la force magnétique qui est assez grande comparativement à la pesanteur.

Les résultats de simulation montrent que la position choisie est un point instable. D'où une régulation doit être implémentée pour stabiliser le système

II.8 – Conclusion :

Le modèle dynamique explicite du système est établi sur l'équation du mouvement de l'objet et les caractéristiques électriques et magnétiques de l'électro-aimant. Ce modèle nous a bien montré que le système présente de fortes non-linéarités, ce qui engendre la difficulté de commande pour ce type de système.

Les simulations de la position effectuées en boucle ouverte sur système nous ont permis de constater l'instabilité de celui-ci.

CHAPITRE III

LA COMMANDE DU SYSTEME

III.1 – Introduction :

Après l'obtention du modèle mathématique qui décrit le comportement du système, ce chapitre sera consacré à la synthèse de correcteurs linéaires puis non linéaires. Nous allons commencer par la synthèse d'un régulateur linéaire du type proportionnel dérivée (PD) et qui sera à la fin programmé sur le dsPIC. D'autre part nous allons calculer un régulateur non linéaire qui présente certains avantages par rapport à son précédent tels que la robustesse et le rejet de perturbations.

III.2 – Transposition des méthodes analogiques :

La synthèse de correcteurs numériques par extension de correcteurs analogiques est une approche couramment utilisée dans le domaine industriel. Cela s'explique par le fait que les techniques d'étude des systèmes continus sont généralement bien maîtrisées et que les spécifications sont plus facilement interprétables sur des modèles continus que sur des modèles échantillonnés.

On s'intéresse à des méthodes de synthèse dans lesquelles le correcteur est conçu à partir d'un modèle prenant en compte de manière approchée l'existence du bloqueur en amont du procédé. On examine ensuite de manière détaillée la discrétisation du correcteur, le régulateur P..D.

On se place dans ce qui suit dans le cas d'un système où le théorème de SHANNON est vérifié .on désigne par f_h la fréquence la plus haute du système régulé en boucle fermée

En pratique, on prend [3] :

$$5. f_h < \frac{1}{T_e} < 25. f_h \quad (\text{III. 1})$$

Donc : $T_e = 0.001 \text{ s}$

III.2.1 –Discrétisation :

Cette approche suppose que l'on ait réalisé la synthèse d'un correcteur analogique par les méthodes d'étude des systèmes continus. On recherche alors un algorithme numérique qui se rapproche le plus possible du correcteur analogique, en faisant des approximations de la variable de Laplace s , ou sur les pôles et zéros de la fonction de transfert du correcteur analogique. Si on raisonne en termes de fonctions de transfert, on cherche à obtenir la fonction de transfert $R(z)$ d'un correcteur numérique par approximation de celle d'un correcteur analogique $R(s)$, comme c'est illustré sur la figure suivant [9] :

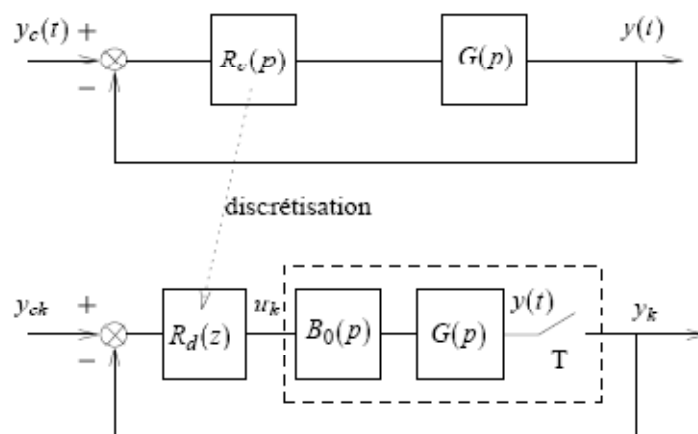


Figure III.1: Discrétisation d'un correcteur analogique

III.2.2 –Approximations de la variable s

Le principe de l'approche consiste à déduire un correcteur $R_d(z)$ du correcteur $R_c(s)$ en choisissant une approximation de la variable s , selon le schéma de la figure III.2

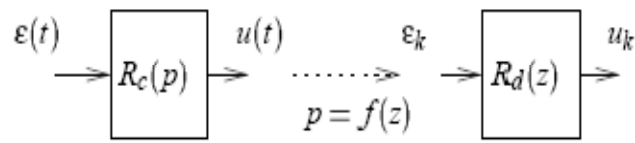


Figure III.2 : Approximation de la variable s

Les approximations les plus utilisées sont les suivantes :

III.2.2.1 –Discrétisation avant :

$$s = \frac{z - 1}{T} \quad (\text{III. 2})$$

Elle résulte de l'approximation de la dérivée d'une fonction entre deux instants d'échantillonnage par la méthode d'Euler :

$$\mathcal{L}^{-1}[s \cdot X(s)] = \frac{dx(t)}{dt} \approx \frac{x(T + t) - x(t)}{T} = Z^{-1} \left[\frac{Z - 1}{T} \cdot X(z) \right] \quad (\text{III. 3})$$

III.2.2.2 –Discrétisation arrière :

$$s = \frac{z - 1}{z \cdot T} \quad (\text{III. 4})$$

Elle résulte de l'approximation suivante de la dérivée d'une fonction entre deux instants d'échantillonnage :

$$\mathcal{L}^{-1}[s \cdot X(s)] = \frac{dx(t)}{dt} \approx \frac{x(t) - x(t - T)}{T} = Z^{-1} \left[\frac{Z - 1}{z \cdot T} \cdot X(z) \right] \quad (\text{III. 5})$$

III.2.2.3 –Approximation de Tustin :

$$s = \frac{2}{T} \cdot \frac{z - 1}{z + 1} \quad (\text{III. 6})$$

Cette approximation, connue également sous le nom de transformation bilinéaire, résulte de l'approximation de l'intégration numérique par la méthode des trapèzes. En effet, soit :

$$y(t) = \int x(t). dt \mapsto y(s) = \frac{1}{s}.X(s) \quad (\text{III. 7})$$

Par approximation entre deux instants d'échantillonnage, il vient :

$$\begin{aligned} y_k = y(k.T) &= y_{k-1} + \frac{x_{k-1} + x_k}{2}.T \\ &\quad \downarrow Z \\ (1 - z^{-1}).Y(z) &= \frac{T}{2}.(1 + z^{-1}).X(z) \end{aligned} \quad (\text{III. 8})$$

Soit la formule :

$$Y(z) = \frac{T}{2} \cdot \frac{z+1}{z-1} \cdot X(z) \quad (\text{III. 9})$$

III.3 -Prise en compte du bloqueur dans la synthèse : [9]

Si l'on reprend la logique de la figure III.2, on note qu'elle suppose que $R_d(z)$ sera une approximation de $R_c(s)$ et devrait donc satisfaire la boucle réalisée avec un opérateur $G_d(z)$ dont le comportement serait très proche de $G(s)$. Dans cette logique cela revient à ignorer le comportement du bloqueur, $B_0(s)$. Ceci peut devenir très préjudiciable si l'échantillonnage T est élevé, car dans ce cas le comportement de $Z[B_0(s).G(s)]$ se distingue fortement de celui de $G(s)$

Nous présentons ci-après des méthodes prenant en compte l'existence du bloqueur d'ordre zéro, de manière exacte ou approchée.

-Approximation du bloqueur par un retard pur

Dans ce cas, la synthèse est réalisée en analogique, mais la fonction de transfert du procédé est choisie égale à :

$$e^{-\frac{T_e}{2}.s}.G(s) \quad (\text{III. 10})$$

Ce qui revient à faire l'approximation le bloquer d'ordre zéro comme un retard pur d'une demi période d'échantillonnage et de tenir compte de cette approximation lors du calcul initial de $R_c(s)$

Pour la simulation, le terme $e^{-\frac{T_e}{2}.s}$ sera remplacé par :

$$e^{-\frac{T_e}{2}.s} = \frac{1}{1 + \frac{2}{T_e}s} \quad (\text{III. 11})$$

En tenant compte de la valeur de T_e :

$$e^{-\frac{T_e}{2}.s} = \frac{1}{1 + \frac{s}{2000}}$$

La fonction de transfert devient :

$$G(s) = -53.5 \frac{1}{(s + 185.9679).(s^2 - 207.1176).(\frac{s}{2000} + 1)} \quad (\text{III. 12})$$

III.4 –Calcul du gain du capteur :

A fin de calculer le gain du capteur, on a fait quelques mesures, tel que

- On alimente le capteur avec un signal continu de 5 volts
- on variant la distance 'z'
- en calcul a chaque fois la tension de sortie

Les résultats sont comme suit :

Position (mètre)	Tension (volte)
0	3.39
0.002	3.07
0.005	2.76
0.007	2.68
0.010	2.58
0.013	2.52
0.015	2.50
0.017	2.48
0.020	2.47
∞	2.42

Tableau III.1 : la tension de sortie du capteur en fonction de la distance

Comme notre système fonctionne autour d'un point de fonctionnement, on va se contenter d'une approximation linéaire autour de ce point.

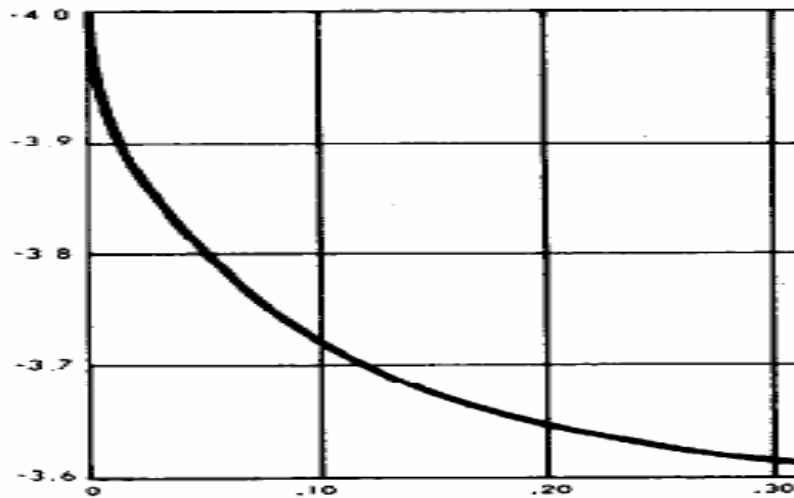


Figure III.3 : caractéristique du capteur (tension en fonction de la position)

Comme le montre la figure la caractéristique du capteur peut être assimilée à :

$$u = k * x + u_0 \quad (\text{III.13})$$

Donc : $k = -15 \text{ m/v}$ et $u_0 = 2.64 \text{ v}$

III.5 -Régulateur P.D. numérique :

Le régulateur P.D. est très répandu dans le domaine industriel. Il constitue l'outil standard de la commande de nombreux procédés industriels. Conçu initialement en technologie analogique (hydraulique, pneumatique, électronique,...), il a été transposé en numérique pour pouvoir être implémenté sur ordinateur. Cette transposition n'est rien d'autre que l'application de la méthode de discrétisation [9]

Concernant notre système, on peut montrer facilement qu'un régulateur proportionnel ne peut pas régler le système :

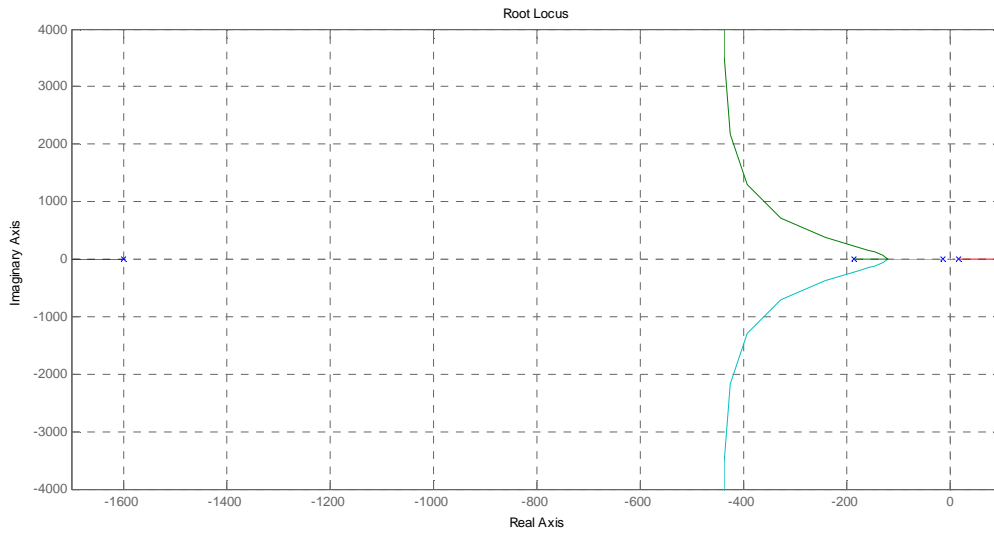


Figure III.4 : Lieu de racine d'Evans pour $K > 0$

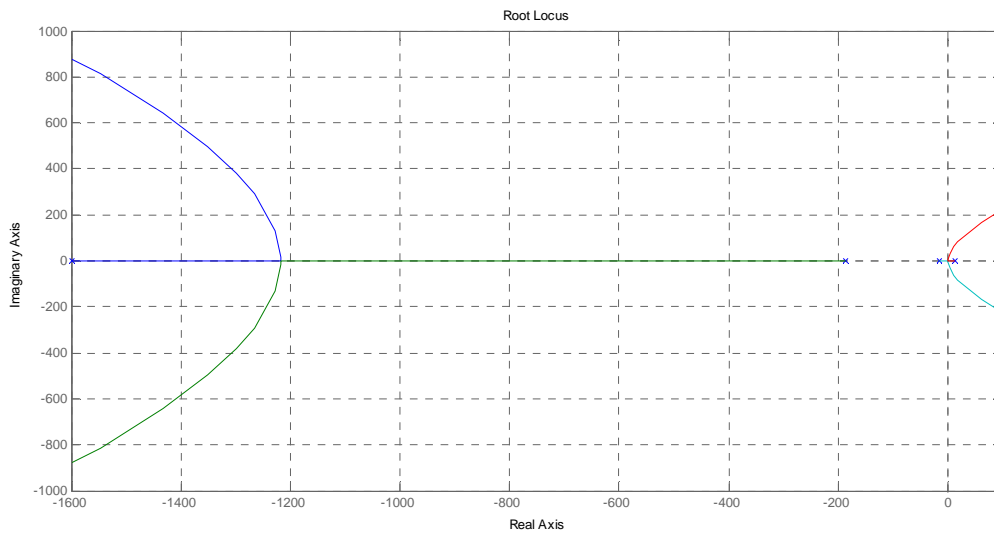


Figure III.5 : Lieu de racine d'Evans $K < 0$

Donc on ne peut pas appliquer la méthode de pompage.

Régulateur PD série :

Le correcteur devant effectuer un traitement de signal, il est en générale avantageux de le placer l'après ou le niveau d'énergie est le plus faible i.e. à la sortie du comparateur, il se trouve donc situé en série dans la boucle de régulation [15]

La fonction de transfert en boucle ouvert est comme suit :

$$FTBO = (K_p + K_d s) \cdot \frac{-53.5 * 2000}{(s + 185.9679) \cdot (s^2 - 207.1176) \cdot (s + 2000)} \quad (III. 14)$$

Or la fonction de transfert en boucle fermée est donnée par :

$$FTBF = \frac{FTBO}{1 + k_c * FTBO} \quad (III. 16)$$

Sachant que le gain du capteur est $k_c = -15$

$FTBF$

$$= \frac{-53.5 * 2000(K_p + K_d s)}{s^4 + 2186 s^3 + 3.717e005 s^2 - (4.528e005 - 1611000 * kd)s - (7.703e007 - 1611000 * K_p)}$$

Donc :

$FTBF$

$$= \frac{-10700 * (K_p + K_d s)}{s^4 + 2186 s^3 + 3.717e005 s^2 - (4.528e005 - 1611000K_d) s - 7.703e007 + 1611000.K_p}$$

Pour le choix de K_p et K_d , on fait un placement de pole .On aura comme valeur :

$$K_p = 70 \text{ et } K_d = 0.7$$

La fonction de transfert en boucle fermée est comme suit :

$$FTBO = \frac{-7.517e006 s - 7.517e008}{s^4 + 2186 s^3 + 3.717e005 s^2 + 6.748e005 s + 3.572e007} \quad (III. 17)$$

III.6 –Les simulations :

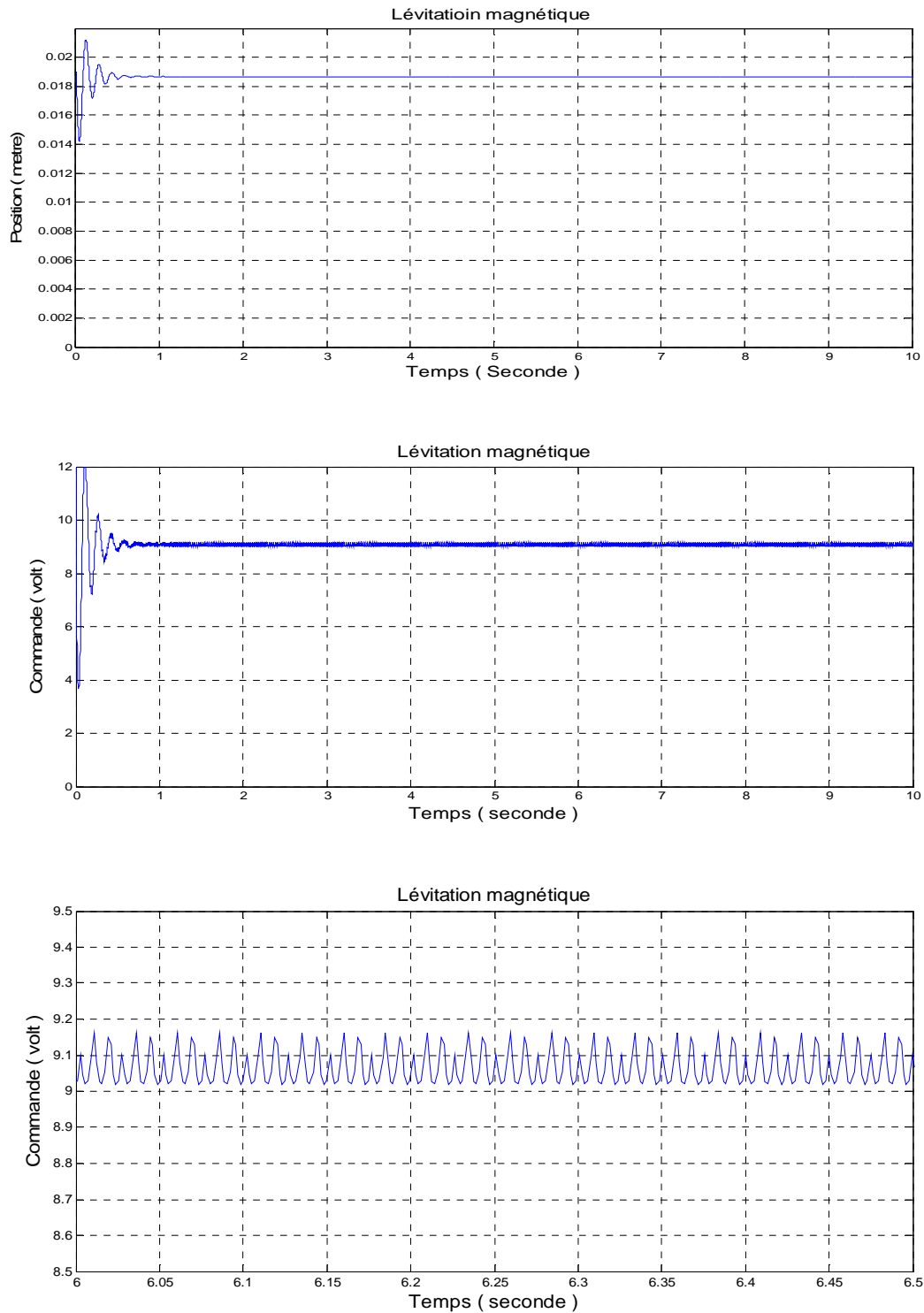


Figure III.6 : Résultats de simulation pour la commande linéaire (PD) (Réglages de position)

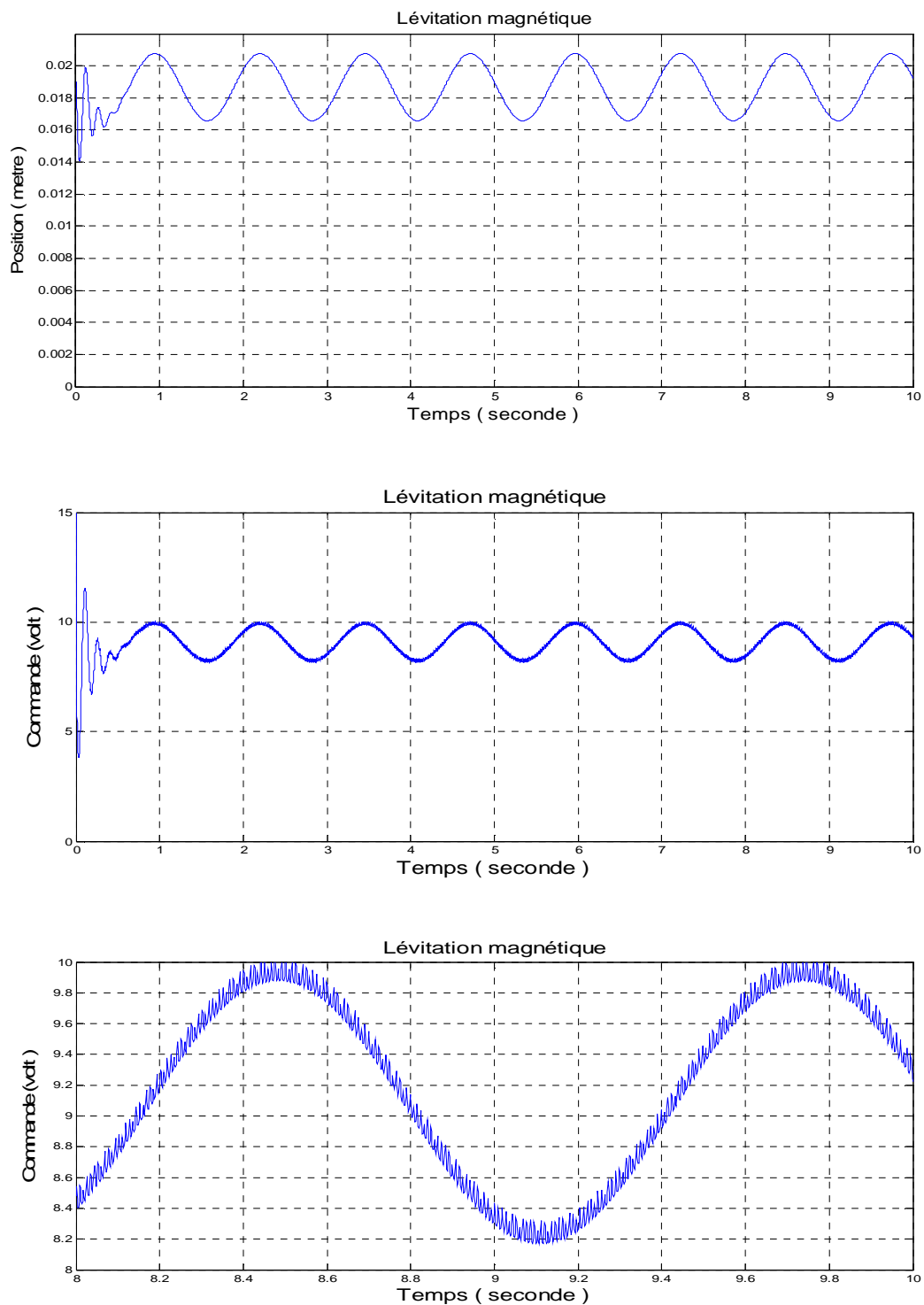


Figure III.7 : Résultats de simulation pour la commande linéaire (PD) (Suite de référence d'un signal sinusoïdale)

III.6 – Test de performance et de robustesse :

III.6.1 – Test de performance :

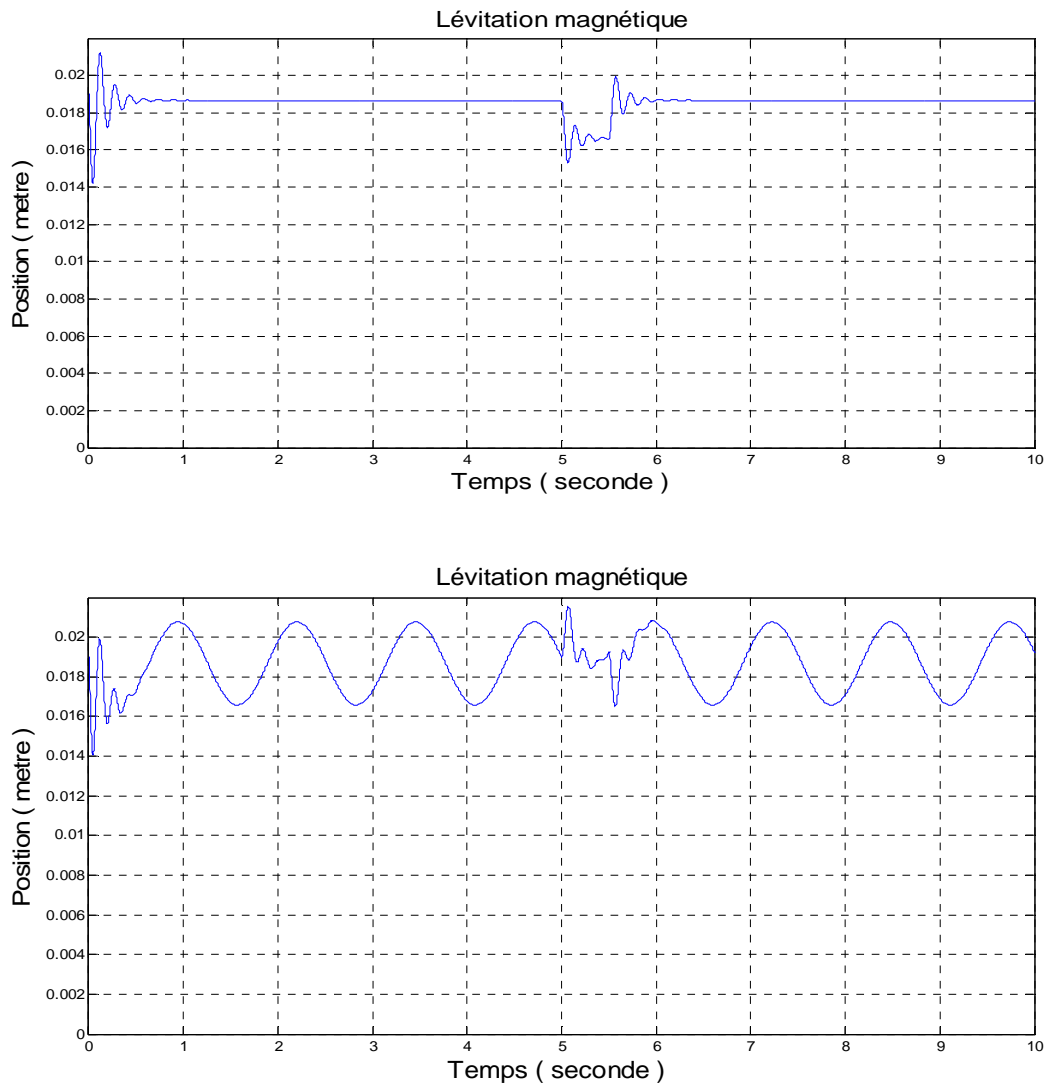


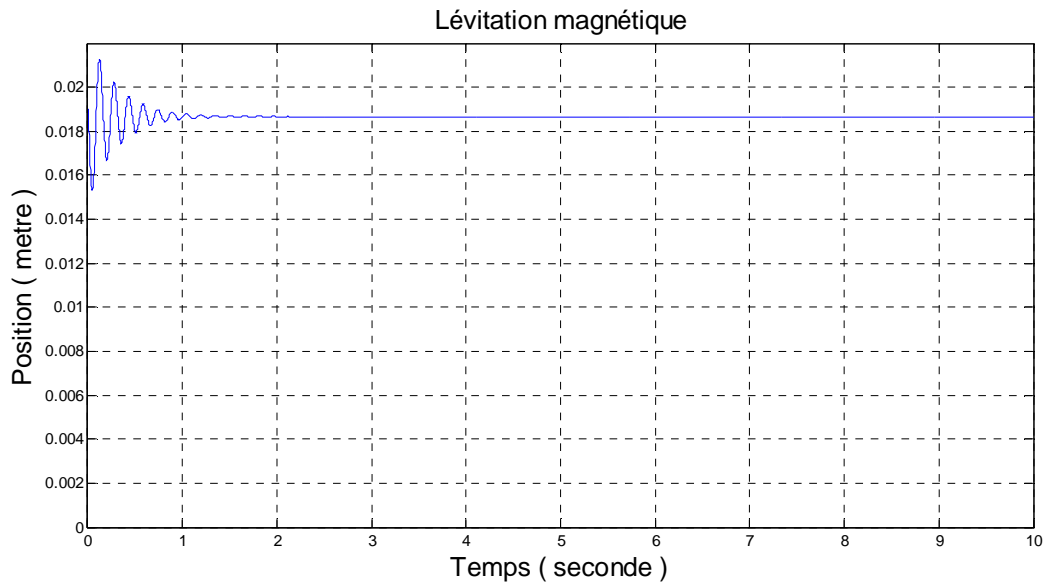
Figure III.8 : Résultats de simulation pour la commande linéaire (PD) une perturbation aléatoire, appliquée à l'instant 5 Sec, sur l'objet.

Interprétation des résultats :

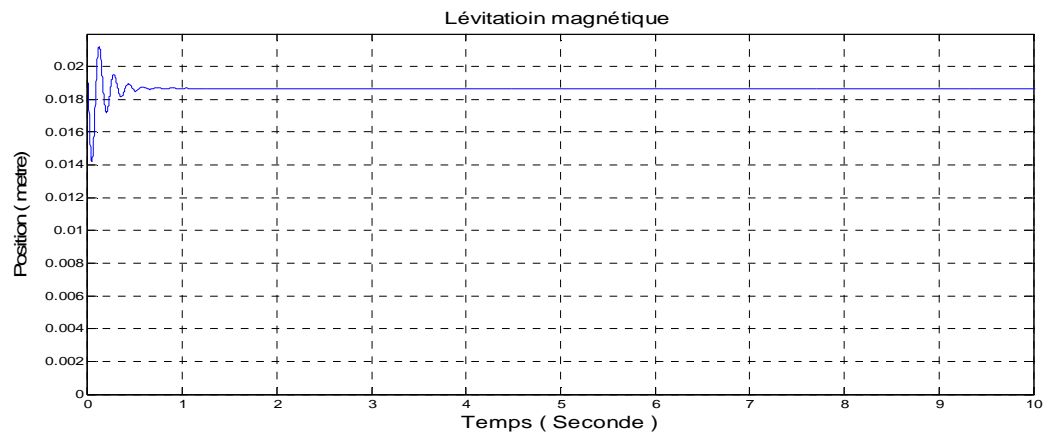
Lors des perturbations, le système répond par une oscillation très faible ou très négligeable de l'objet.

Le régulateur réagit et corrige ces perturbations d'une énergie tolérable et stabilise le système

III.6.2 – Test de robustesse



Erreurs paramétriques sur l'inductance de l'enroulement



Erreurs paramétriques sur la résistance de l'enroulement

Figure III.9 : Résultats de simulation pour la commande linéaire (PD) avec perturbation

Interprétation des résultats :

Comme on peut le voir le système n'est pas très robuste, tel que une variation de résistance à cause de l'effet joule influe sur la porté de la bobine, cependant une variation de l'inductance peut déstabiliser le système.

Donc ce type de réglage n'est pas robuste, c'est-à-dire les variations des paramètres du système ont une influence sur le comportement dynamique

III.7 – Programmation du PD :

Dans le domaine industriel, ce qui compte c'est de pouvoir reproduire facilement sans grande dispersion des caractéristiques, un système ou une commande. Dans l'état actuel des techniques, l'utilisation de processeur et d'automate répond à ces critères [8].

On retrouve en commande numérique les correcteurs classiques P, PI, PD, PID

Cas du dérivateur :

$$C(s) = K_d * s \quad (III.18)$$

On utilisant l'approximation Discrétisation arrière :

$$s = \frac{z - 1}{z \cdot T} \quad (III.19)$$

On obtient alors :

$$C(z) = \frac{K_d}{T_e} * (1 - z^{-1}) \quad (III.20)$$

Et on déduit la relation de récurrence du filtre :

$$u(n + 1) = \frac{K_d}{T_e} * [e(n + 1) - e(n)] \quad (III.21)$$

Mais, on a utilisé un régulateur PD donc :

$$u(n + 1) = K_p * e(n + 1) + \frac{K_d}{T_e} * [e(n + 1) - e(n)] \quad (III.22)$$

III.8 –Commande non linéaire :

Bien que la commande linéaire est intéressante car elle facilite la synthèse en la ramenant à une synthèse linéaire, elle souffre de deux inconvénient major :

- Par fois la paire (A, B) est non commandable.
- La stabilisation n'est que locale, donc pour les systèmes ou on doit s'éloigner beaucoup du point de fonctionnement, on ne peut utiliser cette approche.

Pour remédier ce problème on utilise la commande non linéaire.

III.8.1 –Commande linéarisante : [10]

Cette méthode consiste à trouver une nouvelle base d'état de tel sort que les équations d'état s'écrit totalement ou partiellement linéaire

Soit le système à réglé :

$$\begin{cases} \dot{x} = f(x) + g(x).u \\ y = h(x) = x_2 \end{cases} \quad (III. 23)$$

Comme le degré relatif du système est de 3, calculant la nouvelle représentation d'état :

$$\begin{cases} \dot{z} = A z + B. v \\ y = C * z \end{cases} \quad (III. 24)$$

Tel que :

$$\begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix} = \begin{pmatrix} x_2 \\ x_3 \\ 9.81 - 0.00224 * \frac{x_1^2}{(0.004 + x_2)^2} \end{pmatrix}$$

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad C = [1 \quad 0 \quad 0]$$

La commande Linéarisante est :

$$u = \frac{1}{L_g L_f^{r-1} h(x)} [-L_f^r h(x) + v] \quad (III.25)$$

Tel que r est le degré relatif du système, $L_f h(x)$ est la dérivée de lie définie comme suit :

$$L_f h(x) = \frac{\partial}{\partial x} (h(x) * f(x))$$

$$L_f^{n+1} (h(x)) = \frac{\partial}{\partial x} (L_f^n (h(x)) * f(x))$$

Après calcul on aura :

$$u = -1.20 * \frac{(0.004 + x_2)^2}{x_1} \left[-0.83 * \frac{x_1^2}{(0.004 + x_2)^2} - 0.0023 * \frac{x_1^2 * x_3}{(0.004 + x_2)^3} + v \right] \quad (III.26)$$

Or on doit prendre en compte le bloqueur et le gain du capteur donc les matrices deviennent :

$$A = \begin{bmatrix} -2000 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \quad C = [-30 \quad 0 \quad 0 \quad 0]$$

III.8.1 –Régulateur d'état :

Le régulateur d'état est un régulateur linéaire fournissant un signal de commande $v(t)$ proportionnel à chacune des variables d'état décrivant le système :

$$\begin{aligned} v(t) &= -k \cdot z + r \\ v(t) &= -(k_1 \quad k_2 \quad k_3 \quad k_4) \cdot z + r \end{aligned} \quad (\text{III.26})$$

Le vecteur-ligne k contient les gains associés à chaque variable d'état.

Par placement de pôles, K est comme suit :

$$K = [0 \quad 8000000 \quad 14000 \quad 700]$$

Donc :

$$u = 26.40 * \frac{(0.004+x_2)^2}{x_1} \left[-0.83 * \frac{x_1^2}{(0.004+x_2)^2} - 0.00225 * \frac{x_1^2 * x_3}{(0.004+x_2)^3} + v \right] \quad (\text{III.27})$$

Tel que :

$$v = -8000000 * x_2 - 1400000 * x_3 - 6867 + 1.568 * \frac{x_1^2}{(0.004+x_2)^2} + r \quad (\text{III.28})$$

III.8.3 –Les simulations

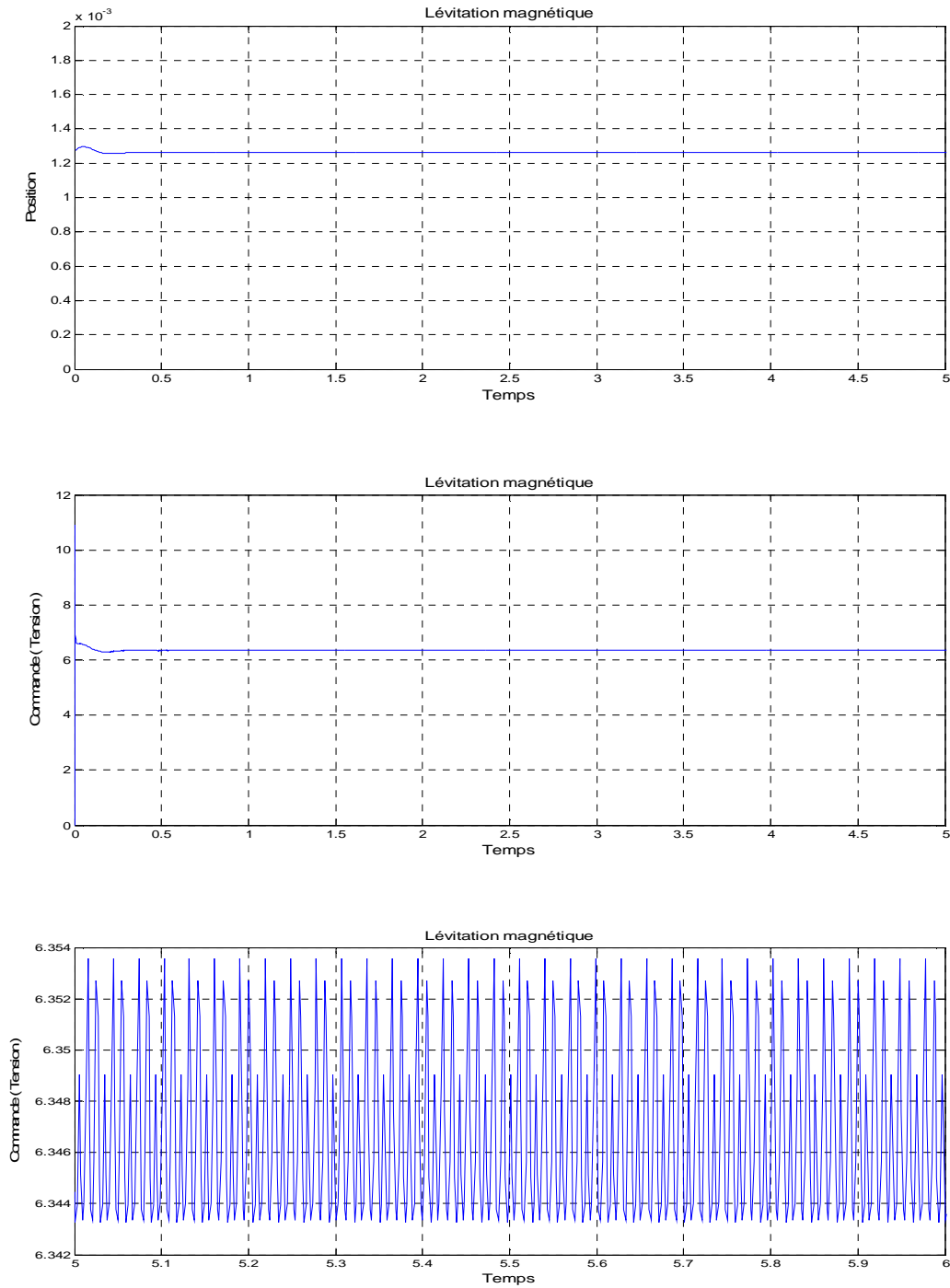


Figure III.10 : Résultats de simulation pour la commande non linéaire (Commande linéarisante)

III.8.4 – Test de performance et de robustesse

III.8.4.1 – Test de performance :

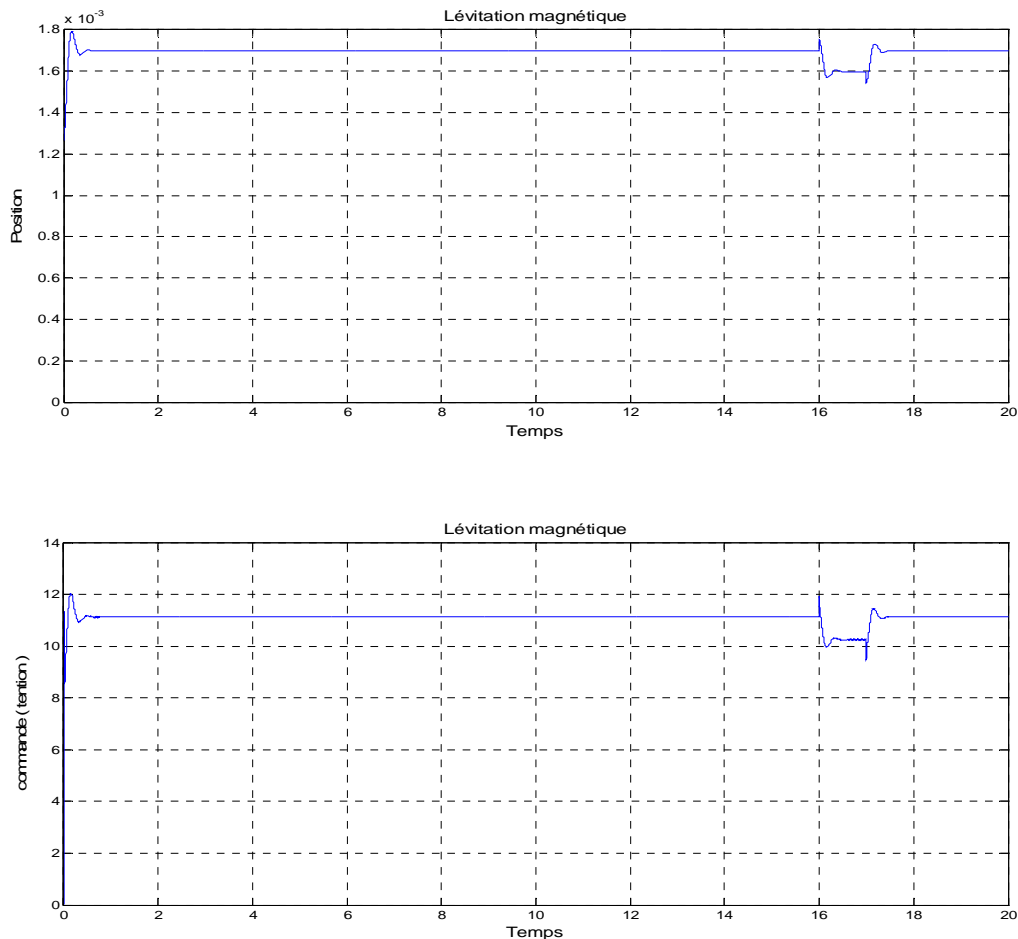
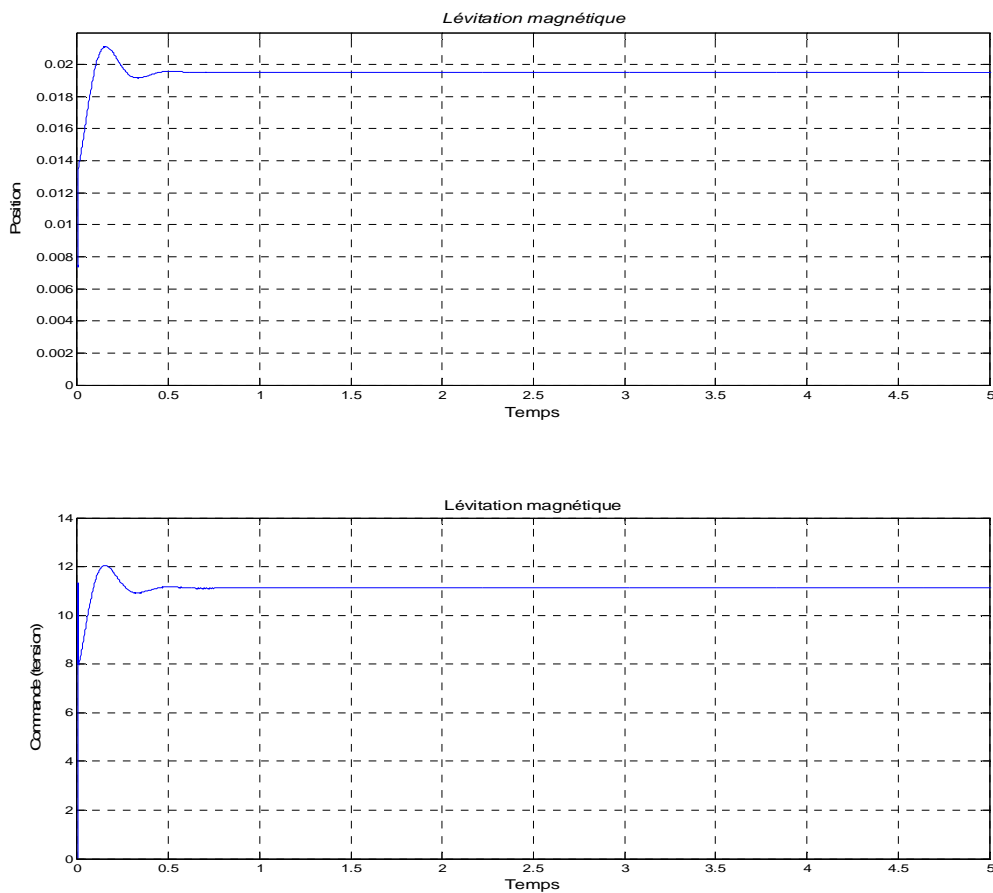


Figure III.11 : Résultats de simulation pour la commande non linéaire, une perturbation aléatoire, appliquée à l'instant 5 Sec, sur l'objet

Nous notons :

- Le rejet des perturbations sans dépassement des limites pour l'objet.
- La stabilisation d'objet avec des petits dépassements qui sont dues aux changements brusque de la position
- La commande est acceptable et réalisable pratiquement.

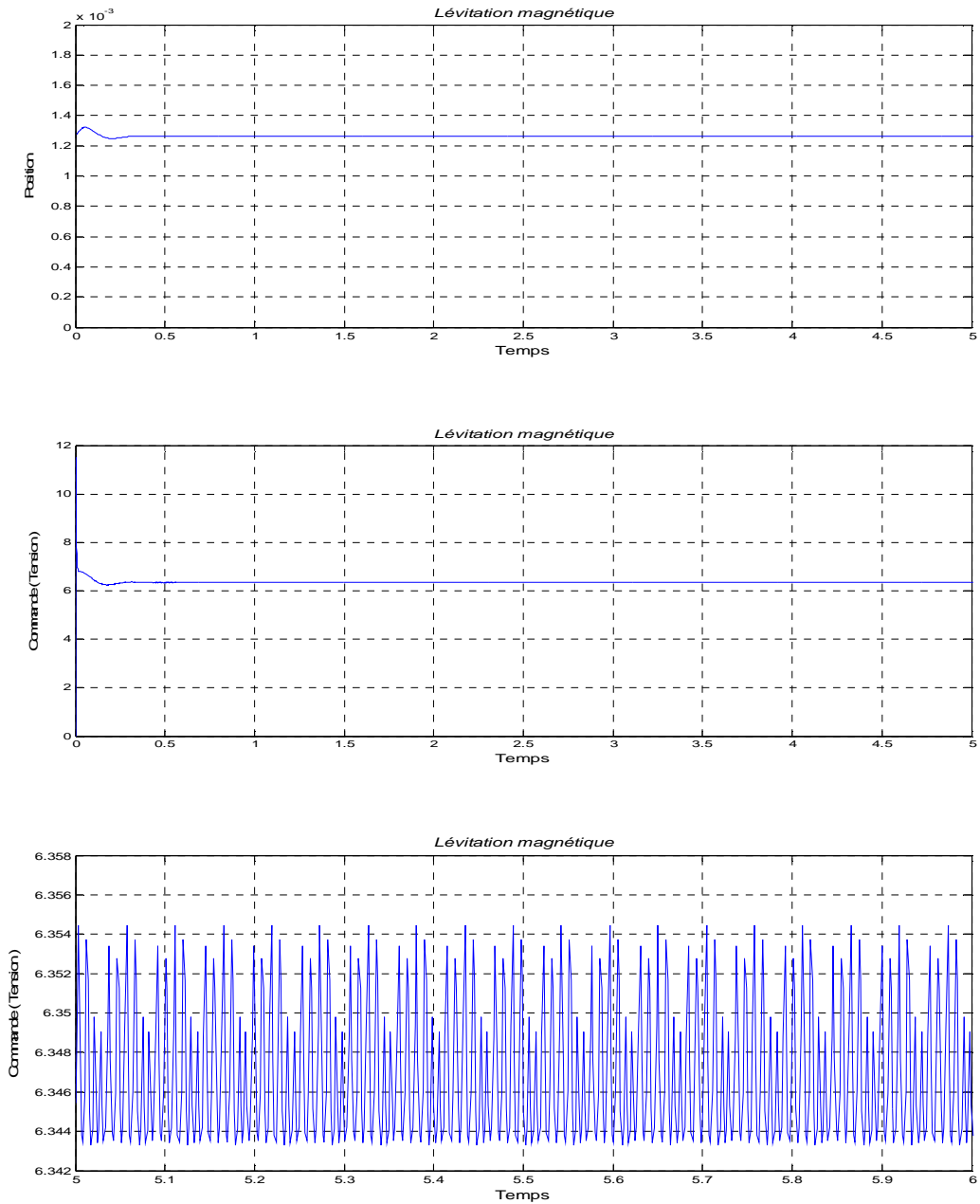
III.8.4.2 – Test de robustesse :



Erreurs paramétriques sur la résistance de l'enroulement

Figure III.12 : Résultats de simulation pour la commande non linéaire (commande linéarisante)

Le réglage non linéaire (commande linéarisante) est robuste par rapport à la variation de la résistance, tel que une variation n'introduit pas l'instabilité, mais la tension appliquée augmente pour garder la même position



Erreurs paramétriques sur l'inductance de l'enroulement

Figure III.12 : Résultats de simulation pour la commande non linéaire (commande linéarisante)

Le réglage non linéaire (commande linéarisante) est robuste par rapport à la variation de l'inductance, tel que une variation n'introduit pas l'instabilité, mais la tension appliquée très oscillatoire pour garder la stabilité du système.

Nous notons que même pour des erreurs paramétriques considérables sur L ou R , nous enregistrons les mêmes robustesses telles que stabilité, non dépassement des limites et la commande tolérable et non énergétique.

III.9 –Conclusion :

Dans ce chapitre nous nous sommes intéressées, à la synthèse de la commande linéaire de type proportionnel dérivée. Cette méthode était facile à synthétiser mais elle souffre de quelque limitation tel que la robustesse

Nous avons vu la partie théorique de la commande non linéaire. Ainsi, nous avons cité les avantages et les inconvénients de cette commande, tel que l'amélioration des performances et robustesse.

D'après les résultats obtenus, nous constatons les avantages qu'elle apporte cette commande par la rapidité du temps de réponse, la commande moins énergétique, la meilleure limitation du déplacement de l'objet et la très bonne robustesse vis-à-vis des erreurs paramétriques tel que, la commande garde ses robustesses et performances sans dépasser les limites de déplacement de l'objet.

Après avoir procédé à l'exposé des principaux aspects de la commande linéaire, l'objectif du prochain chapitre est de développer le programme de la commande du dsPIC.

CHAPITRE IV

PROGRAMMATION

CHAPITRE IV : Programmation

IV.1- Introduction :

Après avoir synthétisé des correcteurs pour la commande du système, il est temps de les implémenter sur notre calculateur. Dans cette partie nous allons donc envisager la programmation du dsPIC en expliquant les particularités du langage évolué que nous avons utilisé (le C30). La plus grande partie de ce chapitre sera destinée à expliquer la configuration du module entrées / sorties et des autres périphériques (CAN, PWM, TIMER...). Pour chaque périphérique on va expliquer brièvement le fonctionnement, quoique pour quelqu'un qui cherche de compréhension meilleure, le retour au manuel et au datasheet du composant est primordial car les explications que nous citons ici ne seront pas suffisants et ne donne que des résumés que nous avons jugé d'être importants afin d'avoir une idée de base et pouvoir comprendre le programme final qui sera à la fin donné et doté de lignes de commentaires pour faciliter la tâche de la lecture du code.

IV.2 - Configuration des périphériques :

IV.2.1 - Le module D'E/S :

Le premier module que nous allons aborder est le module d'entrées / sorties. Ce module est le plus simple de tous, et les concepts qui vont être développés ici s'appliqueront à tous les autres modules.

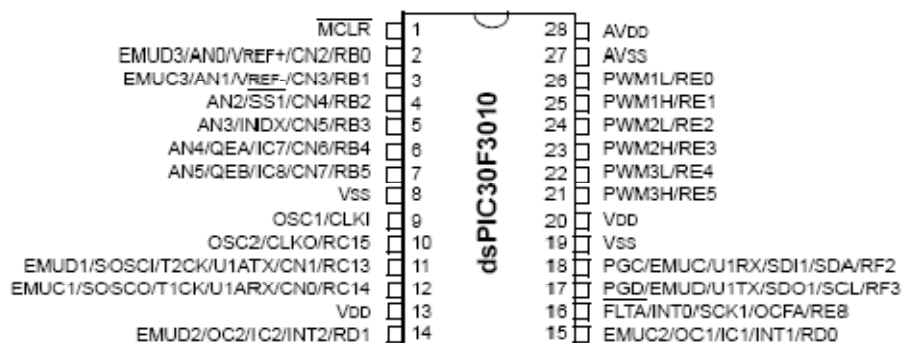


Figure IV.1 : brochage du dsPIC 30F301

Cette illustration se trouve dans la section « Pin Diagrams » du datasheet du dsPIC, elle est démonstrative du brochage de ce composant. Nous observons que la majorité des pattes possèdent un nom du type Rxy où x est une lettre allant de B à F, et y un chiffre allant de 0 à 15. Cette dénomination décrit la possibilité de ces pattes à être utilisées comme entrées/sorties.

Sur les 28 pattes qui composent un dsPIC, une vingtaine d'entre eux peuvent être utilisées comme entrées / sorties logiques. Généralement une patte peut être configurée pour assurer deux ou trois... jusqu'à cinq fonctions différentes

Toutes les pattes du microcontrôleur sont organisées en port, c'est la lettre x. Sur ce port, chaque patte à son propre numéro y. Ainsi, la patte 26, « RE0 » est la patte 0 du port E. Tous les ports ne sont pas complets (aucun même sur ce microcontrôleur), c'est-à-dire qu'ils n'ont pas tous les chiffres possibles. Si on avait utilisé le dsPIC30F ayant plus de pattes (un autre modèle de la même famille), on aurait eu tous les ports complets.

Pour utiliser une patte en entrée/sortie il faut définir 2 choses :

- Est-ce une entrée ou une sortie ?
- Quelle est sa valeur (de sortie ou d'entrée) ?

Pour répondre à ces questions, nous allons utiliser des registres de configuration. Un registre est une zone de la mémoire qui est directement utilisée par le matériel pour piloter son fonctionnement. C'est une sorte de variable, mais qui possède un impact direct et intrinsèque. Chaque périphérique utilise des registres pour fonctionner. Pour le module I/O, ces registres se nomment TRISx, PORTx et LATx

Le registre TRISx :

Le registre TRISx sert à donner la direction d'une patte : entrée ou sortie. D'après le manuel, si on met le bit TRISE0 du registre TRISE à 0, alors la patte 26 (RE0) est défini comme étant une sortie. De même, si on met le bit TRISF2 du registre TRISF à 1, alors la patte 18 (RF2) sera défini comme étant une entrée. Une méthode facile pour s'en souvenir et que le « 1 » est semblable à un « I » (Input), tandis que le 0 est semblable à un O (Output).

Le registre PORTx :

Les données d'une patte E/S sont accessibles via le registre PORTx. Une lecture de ce dernier donne la valeur de la patte E/S, alors qu'une opération d'écriture dans ce registre écrit la valeur voulue dans une bascule de données associée au port à qui appartient la patte. Il faut avoir attention lors de l'utilisation d'une commande de lecture / écriture sur ce registre pour des pattes configurées en entrée. Si une patte est configurée en entrée est changée pour fonctionner comme sortie plus tard, une valeur inattendu va apparaître en sortie de la patte. Cela est causé par le fait que l'utilisation d'une commande de lecture / écriture lit la valeur instantanée sur la patte puis écrit cette valeur sur la bascule associée au port (port data latch)

Le registre LATx :

Le registre LATx élimine le problème précédent, une lecture / écriture de LATx se fait toujours dans la data latch, ce qui élimine la possibilité d'écrire la valeur d'entrée de la patte sur la data latch.

Ce qui va suivre est la méthode d'utilisation optimale des registres LATx et PORTx pour éviter toutes sortes de conflit.

Lorsqu'une patte est défini en sortie, on lui affecte une valeur en utilisant le registre LATx. Par exemple, si on veut mettre la patte 26 (RE0) à 5V, il suffit de positionner le bit LATE0 du registre LATE à 1.

Si la patte est définie en entrée, on lit sa valeur en regardant la valeur du registre PORTx. Par exemple si la patte RF2 est à 0V, alors le bit RF2 du registre PORTF sera à 0.

Pour comprendre mieux le fonctionnement de ces trois registres (TRISx, PORTx, LATx) voir Annexe.

Il faut toujours garder à l'esprit que le PORTB est couplé en entrée avec les convertisseurs analogiques numériques. Et que par défaut, si on passe une patte du PORTB en entrée, elle est configurée comme une entrée analogique. Il ne faut pas oublier d'aller modifier le registre ADPCFG du module CAN (voir la configuration du module CAN)

IV.2.2 - Le module Timer1 :

Ce module est un compteur. Dans son utilisation la plus basique, il compte le nombre de cycles d'horloge. Mais on peut aussi le configurer pour compter le nombre de changements d'états sur une entrée.

Si on regarde le manuel de la famille 30F, on voit qu'il existe plusieurs types de timer (A, B et C). Or, le datasheet du 30F3010 nous informe que le Timer1 est un timer de type A. Nous allons détailler le fonctionnement du Timer1 comme compteur d'horloge \Leftrightarrow chronomètre.

On observe que ce module utilise 3 registres.

TMR1 : c'est la valeur actuelle du compteur.

PR1 : c'est la valeur max du compteur : lorsque $TMR1 = PR1$, le compteur se remet à 0 et recommence à compter. Le module peut alors générer une interruption s'il en a l'autorisation.

T1CON : c'est le registre de configuration du module. Son utilisation est décrite succinctement comme suit :

Le registre T1CON (Time Base Register) :

Upper Byte:							
R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
TON	—	TSIDL	—	—	—	—	—
bit 15				bit 8			

Lower Byte:							
U-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	U-0
—	TGATE	TCKPS<1:0>		—	TSYNC	TCS	—
bit 7				bit 0			

Figure IV.2: Le registre T1CON

TCS : permet de choisir ce qu'il faut compter. Pour compter les cycles de l'horloge interne, et donc le temps, Si on positionne TCS à 0 (c'est ce qu'on doit faire généralement) on va utiliser l'horloge interne (Internal Clock ($F_{osc} / 4$)). ($F_{osc} / 4$ fait référence au fait qu'il faut 4 cycles pour réaliser une instruction). Si TCS=1, on utilisera l'horloge externe qui correspond à la patte TxCK

TSYNC : c'est pour activer / désactiver la synchronisation avec l'hologe externe. Ce bit n'est pas à utiliser. En effet, il est indiqué d'ignorer ce bit si TCS = 0.

TCKPS : c'est un groupe de 2 bits servant à définir au bout de combien d'évènements à compter, la valeur de TMR1 est incrémentée de 1 (valeur de l'échelle). Ici l'évènement à compter est « une période d'horloge ». A priori, on veut donc que TMR1 augmente de 1 à chaque période d'horloge. En réalité, tout dépend du temps que l'on veut compter. Car il ne faut pas oublier que TMR1 est un registre 16 bits, et donc, il ne peut prendre que des valeurs entre 0 et 65535. Si TCKPS est à 00 (1 période \Leftrightarrow une incrémentation de TMR1), alors on ne pourra compter que 6.5535 ms (en considérant que 1 instruction \Leftrightarrow 100 ns). On peut alors choisir de perdre en précision, mais de gagner en longévité. Pour cela on pourra par exemple positionner TCKPS à 1 (8 périodes d'horloge \Leftrightarrow 1 incrémentation de TMR1). La précision sera alors de 800ns, mais on pourra compter jusqu'à 52.428 ms. On peut ainsi régler ce prescaler (changement d'échelle) pour un rapport de 1:1, 1:8, 1:64 ou 1:256, qui correspond aux valeurs (00, 01, 10, 11)

TGATE : Activer / désactiver le fonctionnement en mode GTA (Gated Time Accumulation). Ça ne sera pas utilisé ici.

TSIDL : permet de dire si le module doit continuer de compter ou non lorsque le microcontrôleur passe en mode veille. En mettant TSIDL à 0, on pourra ainsi mettre le dsPIC en veille, et le réveiller via une interruption au bout d'un certain temps.

TON : sert à activer ou non le comptage. Il est préférable d'activer le timer après avoir configuré tout le reste.

Cette description succincte du Timer1 n'illustre qu'une façon de l'utiliser, mais c'est une des plus courantes.

IV.2.3 - Le convertisseur analogique numérique (module ADC) :

Le module ADC est un convertisseur analogique numérique de 10 bits qui possède 4 canaux de conversions désignés CH0...CH4 et qui peuvent fonctionner tous en même temps ou alternativement (l'un après l'autre). Le schéma bloc qui explique le fonctionnement du module ADC peut être consulté dans l'annexe.

Les entrées associées au module ADC sont celles du port B (il peut être configuré comme entrée logique ou analogique). Nous avons donc 6 pattes (RB0...RB5) qui peuvent être utilisées pour ce module.

La figure suivante démontre une séquence de conversion de base ainsi que les termes que nous allons utiliser.

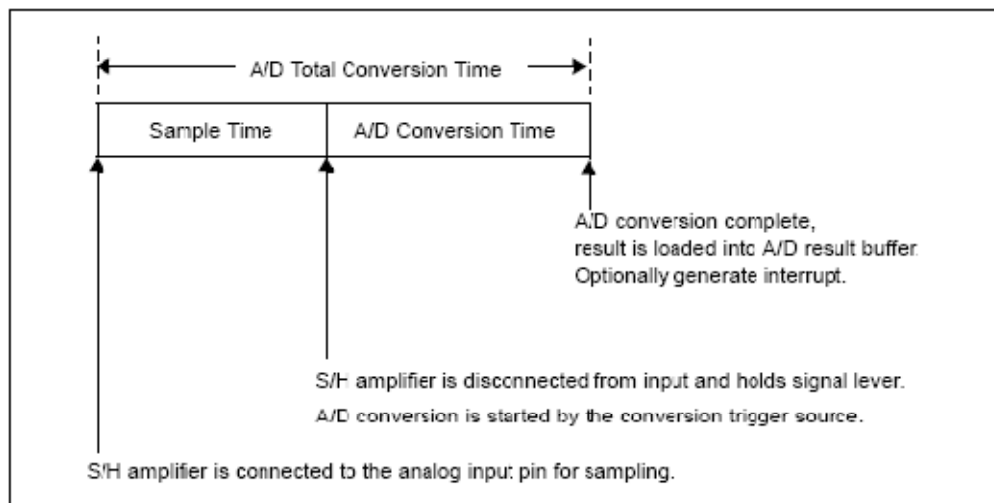


Figure IV.3 : Etapes de fonctionnement du module ADC.

On remarque qu'une séquence de conversion totale est composée d'une phase d'échantillonnage (sampling) et une phase de conversion. On va utiliser les

expressions ‘temps de la phase d’échantillonnage’ et ‘temps de conversion’ pour désigner le temps que va prendre chaque étape. Le ‘temps totale de conversion’ est la somme des deux.

Les registres de configurations qui sont associés à ce module sont les suivants :

Le registre ADPCFG (A/D Port Configuration Register):

C’est un registre de 16 bit, mais les six premiers bits seulement sont utilisables (0 à 5), car chaque bit est associé à une entrée du port B. la mise à un d’un bit correspond à une entrée analogique de la patte du port B, tandis qu’une mise à zéro correspond à une entrée logique.

Le registre ADCON1 (A/D Control Register 1):

Upper Byte:							
R/W-0	U-0	R/W-0	U-0	U-0	U-0	R/W-0	R/W-0
ADON	—	ADSIDL	—	—	—	FORM<1:0>	
bit 15						bit 8	

Lower Byte:							
R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0 HC, HS	R/C-0 HC, HS
SSRC<2:0>			—	SIMSAM	ASAM	SAMP	DONE
bit 7						bit 0	

Figure IV.4 : Le registre ADCON1

ADON : Permet d’allumer (mise à 1) ou d’arrêter (mise à 0) le module ADC. Il est préférable de l’activer après la configuration totale de tous les registres.

ADSIL :

1 = Le module ADC s’arrête en mode veille.

0 = Le module ADC continue à fonctionner en mode veille.

FORM :

Ces deux bits désignent la forme du codage de la donnée convertit dans le buffer du convertisseur.

00 : entier non signé (0000 00dd dddd dddd).

01 : entier signé (ssss sssd dddd dddd).

10 : fractionnel non signé (dddd dddd dd00 0000).

11 : fractionnel signé (sddd dddd dd00 0000).

SSRC :

Mode de déclenchement de la phase de conversion (codage sur trois bit). (011) correspond à une synchronisation avec le module PWM (mode utilisé dans notre cas)

SIMSAM :

Mode de fonctionnement des canaux de conversions CH0, CH1, CH2, CH3.

1 = fonctionnement simultané.

0 = fonctionnement en séquence.

ASAM :

Mode du fonctionnement de l'échantillonneur.

1 = Mode automatique : l'échantillonnage commence automatiquement après avoir terminé la dernière conversion.

0 = Mode manuel (l'échantillonnage commence dès la mise à 1 du bit SAMP)

SAMP :

Indique l'état de l'échantillonneur (marche / arrêt). Ce bit est mis à un (ou à zéro) par le matériel lorsque l'échantillonneur fonctionne en mode automatique ASAM = 1

Si : ASAM = 0, l'échantillonneur est déclenché manuellement par logiciel (programmation)

Si : SSRC = 000, la mise à zéro de ce bit déclenche l'échantillonnage juste après la phase de conversion précédente.

Registre ADCON2 (A/D Control Register 2):

Upper Byte:								
R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	
VCFG<2:0>			reserved	—	CSCNA	CHPS<1:0>		
bit 15								bit 8

Lower Byte:							
R-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
BUFS	—	SMPI<3:0>				BUFM	ALTS
bit 7						bit 0	

Figure IV.5 : Le registre ADCON2

VCFG :

Sélection des entrées établissant les références du voltage High et Low pour la conversion. Pour la valeur 000 : VrefH = AVDD, VrefL= AVSS.

CSNA :

Configuration du mode de fonctionnement du multiplexeur, ça ne sera pas utilisé.

CHPS :

Sélection des canaux utilisés pour la conversion.

1x = CH0, CH1, CH2, CH3 utilisés.

01 = CH0, CH1 utilisés.

00 = CH0 seulement utilisé.

BUFS :

Indique l'état du buffer. Utilisé quand BUFM = 1 pour indiquer l'octet du buffer utilisé.

SMPI :

Mode du déclenchement de l'interruption liée au module ADC. Pour SMPI = 000, il y aura interruption à la fin de chaque phase de conversion.

BUFM :

Mode d'utilisation du buffer. 0 pour un fonctionnement en 16 bits, 1 Pour 8 bits.

ALTS :

Configure le mode de sélection entre les entrées du module ADC via les deux multiplexeurs qui existent. Ce bit ne sera pas utilisé dans notre cas.

Le registre ADCON3 (A/D Control Register 3):

Upper Byte:								
U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
—	—	—	SAMC<4:0>					
bit 15								bit 8

Lower Byte:							
R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADRC	—	ADCS<5:0>					
bit 7							bit 0

Figure IV.6 : Le registre ADCON3

SAMC :

Configuration de la période d'échantillonnage.

SAMC = 00000 dans notre cas

ADRC :

Choix de l'horloge.

1 = horloge RC interne destiné au module ADC.

0 = horloge dérivée de l'horloge du système.

ADCS :

Choix de l'horloge pour le convertisseur (T_{AD}).

Ce registre ne sera plus pris en compte du moment où on choisit de mettre ADRC = 1.

Le registre ADCHS (A/D Input Channel Select Register):

Upper Byte:							
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CH123NB<1:0>		CH123SB	CH0NB	CH0SB<3:0>			
bit 15						bit 8	

Lower Byte:							
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CH123NA<1:0>		CH123SA	CH0NA	CH0SA<3:0>			
bit 7						bit 0	

Figure IV.7 : Le registre ADCHS.

Ce registre est utilisé pour déterminer les entrées analogiques qui seront couplées aux quatre canaux de conversion. Ce registre se compose de deux octets un octet pour la configuration du Mux A et le deuxième comporte les mêmes bits mais pour Mux B. dans notre cas on utilisera seulement l'octet du poids faible (Mux A)

CH123NA :

Sélection de l'entrée négative pour les canaux de CH1, CH2, CH3. Pour CH123NA = 0x l'entrée sera VrefL (= AVSS).

CH123SA :

Sélection des entrées positives pour les canaux de CH1, CH2, CH3.

1 → CH1 = AN3, CH2 = AN4, CH3 = AN5. C'est ce qu'on va utiliser.

0 → CH1 = AN0, CH2 = AN1, CH3 = AN2.

CH0NA :

Sélection de l'entrée négative pour le canal CH0. Pour CH0NA = 0 l'entrée négative sera VrefL (= AVSS).

CH0SA :

Sélection de l'entrée positive pour le canal de CH0, si CH0SA = 0010, l'entrée sera AN2.

Il existe un dernier registre (ADCSL) pour la configuration du fonctionnement en mode séquentiel et qui ne sera pas utilisé dans notre cas.

IV.2.4 - Le module PWM :

Le module PWM assure la liaison extérieure avec le monde analogique car il facilite la tâche de la génération de signaux modulés en largeurs d'impulsions (MLI). Ce module est utilisé en général pour la commande de plusieurs types de moteurs électriques. Dans notre cas c'est lui qui va piloter le hacheur L298 pour faire varier la tension à sa sortie et donc le courant qui passe dans la bobine.

Un schéma bloc qui explique le fonctionnement du module peut être consulté dans l'annexe.

Sur notre dsPIC ils existent 3 canaux indépendants (channels) chaque canal possède deux sorties : high et low. Les pattes associées à ces sorties sont appelées PWMxH et PWMxL, x étant le numéro du canal (x = 1, 2, 3). Aussi on peut configurer chaque paire de sortie PWM pour travailler en mode complémentaire ou en mode indépendant.

Le module permet plusieurs solutions de générations de signaux MLI, ces méthodes sont divisées en deux groupes selon le signal utilisé pour la génération du PWM : les méthodes du signal en dent de scie (dite Edge Aligned PWM), et les méthodes du signal triangulaire (Center Aligned PWM).

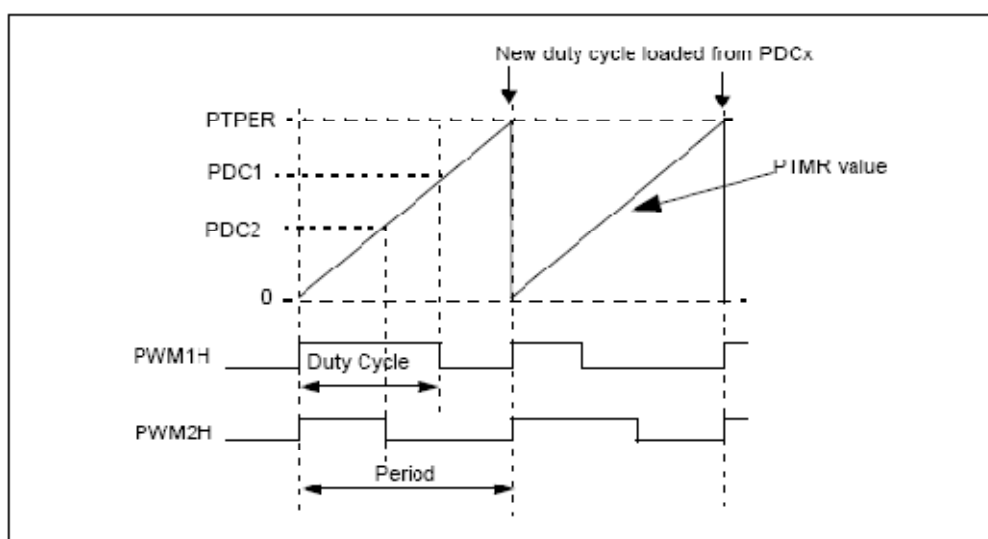


Figure IV.8 : Edge Aligned PWM

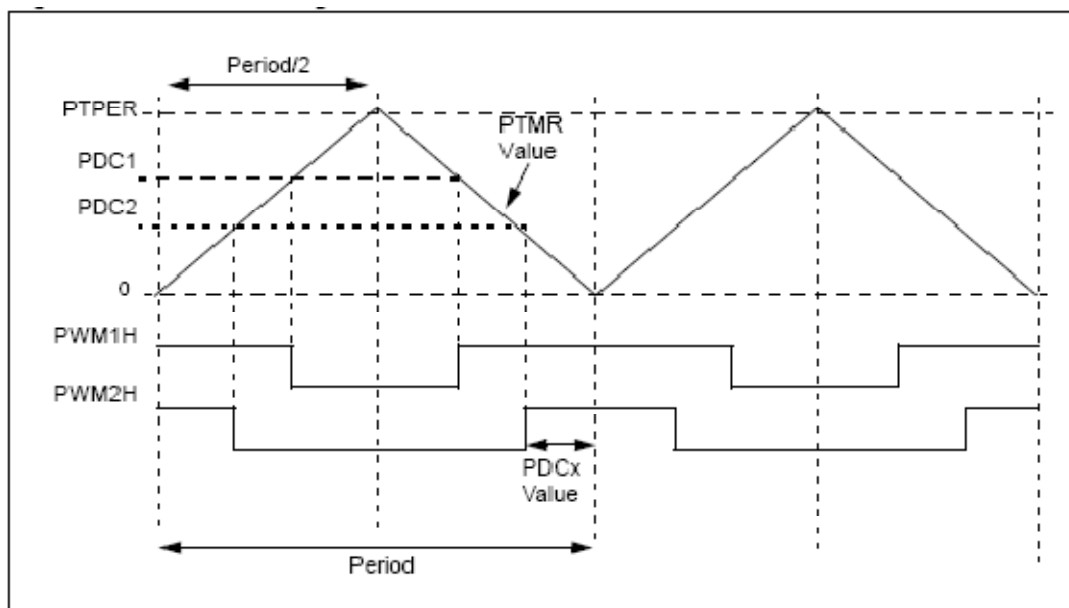


Figure IV.9 : Center Aligned PWM

Les principaux registres de control associés à ce module sont :

Le registre PTPER (PWM Time Base Period register):

Dans ce registre on met la valeur de la période à la quelle on veut que notre signal MLI fonctionne. La valeur qu'il faut prendre le registre PTPER dépend sur la fréquence voulue du signal MLI et la méthode de la génération de celle-ci.

Pour un MLI généré à partir d'un signal en dents de scie la valeur du registre PTPER est donnée par :

$$PTPER = \frac{FCY}{FPWM \cdot (PTMR \text{ Prescaler})} - 1$$

Dans le cas où on utilise un signal triangulaire la valeur de PTPER sera divisée sur deux :

$$PTPER = \frac{FCY}{FPWM \cdot (PTMR \text{ Prescaler}) \cdot 2} - 1$$

Où :

(PTMR Prescaler) : est une valeur numérique qui indique la mise en échelle de la base de temps du module PWM par rapport à celle de l'horloge interne (cycle rate). Cette valeur peut être configurée dans le registre PTCON.

F_{PWM} : est la fréquence à laquelle on veut que notre signal MLI fonctionne.

F_{CY} : est la fréquence de l'horloge interne (cycle rate) à laquelle fonctionne le dsPIC

Exemple :

Dans le cas où on a un signal triangulaire, et on a les valeurs suivantes :

$$F_{CY} = 20 \text{ MHz}$$

$$F_{PWM} = 16 \text{ kHz}$$

$$(\text{PTMR Prescaler}) = 1 : 1$$

$$PTPER = 20\,000\,000 / 16\,000 * 2 - 1 = 624$$

Le registre PTMR (PWM Time Base register):

Ce registre est un compteur qui compte toujours de zéro jusqu'à la valeur PTPER. L'incrémentatation de ce registre est synchronisée par la période du cycle rate T_{CY} selon une échelle (PTMR Prescaler). Quand la valeur PTPER est atteinte il y a deux possibilités :

- Dans le cas du fonctionnement en signal en dent de scie le registre PTMR est remis à zéro
- Dans le cas du signal triangulaire le registre PTMR commence à décompter (compter en inverse), jusqu'à ce qu'il arrive à la valeur zéro

Dans les deux cas le cycle de comptage recommence quand le compteur atteint la valeur zéro.

Les registres PDC1, PDC2, PDC3 (PWM Duty Cycle registers) :

Chaque canal du module PWM peut être considéré comme un générateur de signal MLI indépendant. Les registres PDCx sont utilisés pour mémoriser les périodes des cycles de fonctionnement de chaque générateur (duty cycle). Ces registres sont comparés aux registres PTMR pour effectuer le passage entre les deux niveaux logiques (0 ou 1) du signal PWM en sortie.

Les registres PDCx sont dits à doubles précision par rapport au registre PTMR contrairement au registre PTPER. Cette différence est expliquée par la figure suivante :

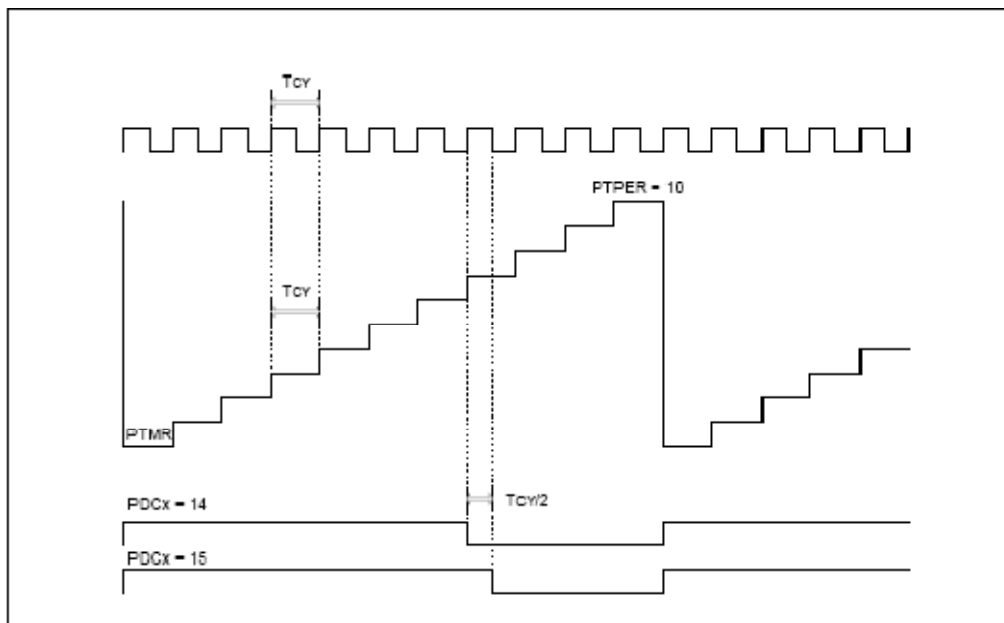


Figure IV.10 : résolution des registres PDCx par rapport au registre PTMR

Le registre PWMCON1 (PWM Control register #1) :

Upper Byte:							
U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	PMOD4	PMOD3	PMOD2	PMOD1
bit 15				bit 8			

Lower Byte:							
R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
PEN4H	PEN3H	PEN2H	PEN1H	PEN4L	PEN3L	PEN2L	PEN1L
bit 7				bit 0			

Figure IV.11 : Le registre PWMCON1

PMOD_x :

Chaque bit PMOD_x (x=1, 2, 3) détermine le mode de fonctionnement de chaque paire (PWM_xH, PXM_xL)

1 = fonctionnement en mode indépendant.

0 = fonctionnement en mode complémentaire.

PEN_xH :

Chaque bit PEN_xH (x=1, 2, 3) est utilisé pour activé (mise à 1) ou désactivé (mise à 0) la sortie PWM_xH correspondante.

PEN_xL :

Chaque bit PEN_xL (x=1, 2, 3) est utilisé pour activé (mise à 1) ou désactivé (mise à 0) la sortie PWM_xL correspondante.

Le registre PWMCON2 (PWM Control register #2) :

Upper Byte:							
U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	—	SEVOPS<3:0>			
bit 15				bit 8			

Lower Byte:							
U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	—	—	—	—	IUE	OSYNC	UDIS
bit 7				bit 0			

Figure IV.12 : Le registre PWMCON2

SEVOPS :

Utilisé pour déterminer l'échelle de la base du temps (postscale) des événements spéciaux (déclenchement d'interruption, synchronisation avec le module ADC...)

IUE :

Méthode de la mise à jour de la valeur des registres PDCx.

1 = mise à jour immédiate.

0 = mise à jour synchronisée avec la base du temps du PWM.

OSYNC :

Méthode de synchronisation des sorties lors du fonctionnement en mode manuel (PWM désactivé, les sorties PWMxx sont forcées manuellement grâce au registre OVDCON), ce bit ne sera pas utilisé dans notre cas.

UDIS :

Ce bit sert à la désactivation (mise à 1) ou l'activation (mise à 0) de la mise à jour des sorties PWM via les registres PDCx et PTPER.

Le registre OVDCON (Override Control Register):

Ce registre est utilisé pour la configuration du mode de fonctionnement dit 'Override' dans lequel les sorties PWM sont forcées à une valeur voulue.

Upper Byte:							
R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
POVD4H	POVD4L	POVD3H	POVD3L	POVD2H	POVD2L	POVD1H	POVD1L
bit 15						bit 8	

Lower Byte:							
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
POUT4H	POUT4L	POUT3H	POUT3L	POUT2H	POUT2L	POUT1H	POUT1L
bit 7						bit 0	

Figure IV.13 : Le registre ADCON

Les bits POVDxx servent à l'activation de la commande PWM (mise à 1) ou de la commande manuelle (mise à 0) dans ce cas le PWM est désactivée et la valeur de la sortie est celle qui est écrite dans le bit POUTxx correspondant.

Le registre SEVTCMP (PWM Special Event Compare register):

Upper Byte:							
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SEVTDIR	SEVTCMP <14:8>						
bit 15							bit 8

Lower Byte:							
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SEVTCMP <7:0>							
bit 7							bit 0

Figure IV.14 : Le registre SEVTCMP

Ce registre est utilisé pour configurer la synchronisation entre le module PWM et le module ADC. Une telle opération permet de minimiser les retards qui existent entre le temps d'acquisition des résultats de la conversion analogique numérique et le temps de la mise à jour des registres PDCx.

Dans les bits SEVTCMP on doit écrire la valeur du temps à laquelle on veut avoir le déclenchement de la conversion. Pour cette raison cette valeur est toujours comparée à la valeur du compteur PTMR. Le bit SEVTDIR sert à spécifier si le déclenchement aura lieu dans la période de comptage (mise à 0) ou de décomptage (mise à 1).

Le registre PTCN (PWM Time Base Control register):

Upper Byte:							
R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
PTEN	—	PTSIDL	—	—	—	—	—
bit 15							bit 8

Lower Byte:							
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PTOPS<3:0>			PTCKPS<1:0>		PTMOD<1:0>		
bit 7							bit 0

Figure IV.15 : Le registre PTCN

PTEN :

Sert à l'activation (mise à 1) ou la désactivation (mise à 0) de la base de temps PWM.

PTSIDL :

1 = désactivation de la base du temps PWM quand le CPU entre en mode veille.

0 = la base du temps reste active PWM quand le CPU entre en mode veille.

PTOPS :

Sélection de la mise à l'échelle postscale de la base du temps PWM. 0000 correspond à une échelle 1 : 1

PCKPS :

Sélection de la mise à l'échelle prescale de la base du temps PWM par rapport à T_{CY} .

11 = base de temps PWM = $64 T_{CY}$. (1 : 64)

10 = base de temps PWM = $16 T_{CY}$. (1 : 16)

01 = base de temps PWM = $4 T_{CY}$. (1 : 4)

00 = base de temps PWM = T_{CY} . (1 : 1)

PTMOD :

Sert à déterminer le mode de la base de temps utilisé. La valeur (10) correspond à un signal triangulaire avec une mise à jour simple des registres PDC. (D'autres modes existent)

IV.2.5 - Le module d'interruption :

A priori, un microcontrôleur n'est capable d'exécuter qu'un seul programme à la fois. Or, dans le cadre d'applications en temps réel il est nécessaire de pouvoir réagir de façon dynamique à des changements de l'environnement. C'est là le rôle du module d'interruption.

Plusieurs modules peuvent générer des interruptions sous plusieurs conditions (PWM, ADC, TIMER ...). La famille 30F possède un masque d'interruption de 7 niveaux qui permet de régler le degré de priorité de chaque interruption. Elles existent aussi des interruptions système ou 'Trap Routine' qui sont utilisées pour exécuter des sous-programmes lorsqu'on a des erreurs dans le système.

Les registres de control associés au module d'interruption sont :

Les registres IECx (Interrupt Enable Control Register) :

Ces registres servent à définir quelles sont les interruptions qui sont autorisées (déclenchement matériel). Vu qu'il y a beaucoup d'interruptions possibles, un seul

registre ne suffit pas, il y en a donc 3 (IEC0, IEC1, IEC2). Pour autoriser l'interruption du module ADC, il faut positionner le bit ADIE du registre IEC0 à 1

Les registres IFSx (Interrupt Flag Status Register) :

Ces registres indiquent au processeur quelle interruption il doit traiter. Vu qu'il y a beaucoup d'interruptions possibles, un seul registre ne suffit pas, il y en a donc 3 (IFS0, IFS1, IFS2) Par exemple si le bit ADIF du registre IFS0 est à 1, alors le processeur doit exécuter le programme d'interruption correspondant au module ADC. Ce bit ne se remet pas à zéro tout seul, il faut donc systématiquement le remettre à 0 à la fin du programme d'interruption.

Les registres IPCx (Interrupt priority Control Register) :

Ce registre sert à définir la priorité d'une interruption. Cette priorité permet de résoudre les conflits d'interruptions (plusieurs interruptions en même temps). La priorité la plus haute, sur une échelle de 0 à 7, passe en premier (0 ⇔ jamais exécuté). Si on veut mettre une priorité de 4 pour le Change Notification (interruption liée au changement d'état d'une entrée logique), il faut écrire la valeur 4 sur les 3 bits CNIP du registre IPC3. En total ils existent 12 registres IPCx (IPC0...IPC11) pour couvrir toutes les interruptions possibles.

Le processeur possède aussi des interruptions non masquée appelées 'traps'. Elles ont pour rôle de fournir à l'utilisateur le moyen de corriger les erreurs qui surviennent lors du débogage ou lors le fonctionnement ordinaire de l'application. Le dsPIC possède 4 sources d'interruption non masquée pré-implémentées :

- Défaillance de l'oscillateur.
- Erreur sur le pointeur de pile. (stack pointer).
- Erreur d'adressage.
- Erreur sur les opérations arithmétiques.

Ces interruptions sont chargées par l'utilisateur qui peut employer un sous programme qui corrige l'erreur, ou faire la réinitialisation du système voir même son arrêt.

Le module change notification (CN) :

Le module change notification donne au dsPIC la possibilité de faire des demandes d'interruption au processeur à la réponse d'un changement d'état sur l'une des pattes CN_x (x = 0 ... 7).

Ils existent quatre registres de control associés au module CN. Les registres CNEN1 et CNEN2 contiennent les bits CN_xIE (x = 0 ... 7), ces bits sert à activer (mise à 1) ou désactiver (mise à 0) les interruptions associés à chaque entrée CN.

Les registres CNPU1 et CNPU2 contiennent les bits CN_xPUE qui sont utiliser por activer le 'weak pull up device' qui agit comme une source de courant connectée à la patte pour éliminer le besoin d'utilisation de résistances externes. Ces registres ne seront pas utilisés dans notre cas.

Remarque :

Sur le dsPIC 30F3010, seulement les registres CNEN1 et CNPU1 sont utilisables car on se dispose de 8 entrées CN_x. Les registres CNEN2 et CNPU2 n'ont aucun rôle et seront simplement négligés.

IV.2.6 - Registres de configuration du fonctionnement général :

Les registres de configuration généraux sont des registres de 24 bits qui servent à configurer différents aspects sur le fonctionnement général du dsPIC comme le choix de l'horloge, le watchdog timer, les options de réinitialisation ... etc.

Ils existent 4 registres de configuration de ce type, ces registres peuvent être configurés à l'entête du programme, donc il ne sera pas nécessaire d'écrire les valeurs de configurations directement dans ces bits. (Voir particularités du langage C30).

Ces registres sont :

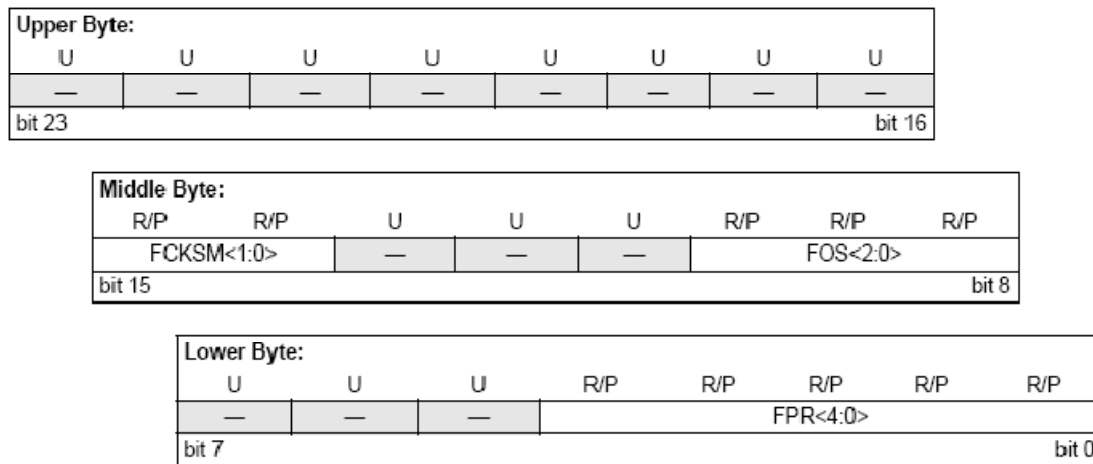
Le registre FOSC (Oscillator Configuration Register) :

Figure IV.16 : Le registre FOSC

Ce registre est utilisé pour configurer le fonctionnement général de l'horloge du processeur.

Les bits FOS sont les plus importants car ils permettent le choix de l'horloge désirée. Nous avons le choix entre 4 horloges internes de fréquences différentes et l'horloge externe (que nous avons décidé d'utiliser). La valeur 011 correspond à cette dernière.

Les bits FPR correspondent au choix de la valeur PLL. La valeur (00110) correspond à une multiplication (*8) de la valeur de l'horloge externe (PLLx8).

Le registre FWDT (Watchdog Timer Configuration Register):

Ce registre sert à la programmation du module watchdog, ce module est une sorte de compteur qui observe l'exécution du temps d'exécution des tâches du système et évite le blocage du processeur. Dans notre cas ce module ne sera pas utilisé et sera en état d'arrêt pendant l'exécution du programme.

Le registre FBORPOR (BOR and POR Configuration Register) :

Ce registre contient principalement les options de réinitialisation du processeur ainsi que d'autres options comme l'établissement des polarités des sorties PWM.

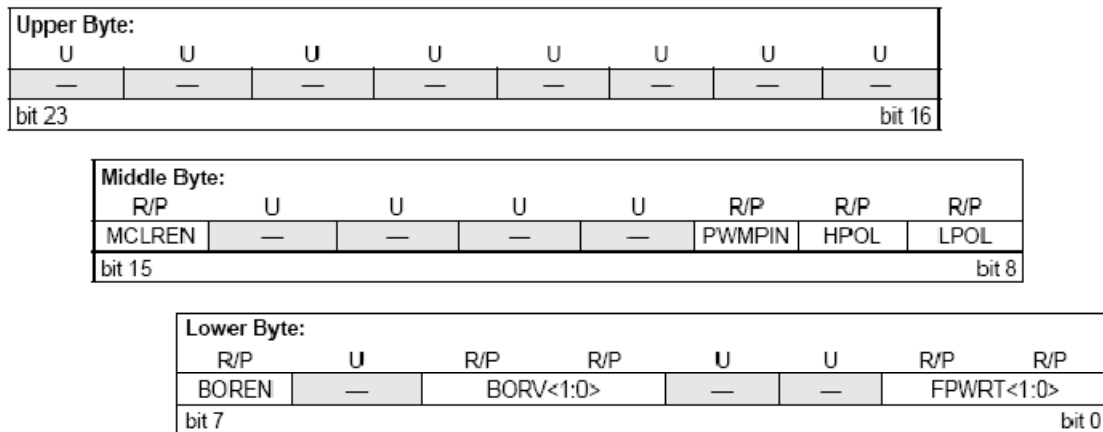


Figure IV.17 : Le registre FBORPOR

MCLEREN :

1 = La patte MCLR est activée

0 = la patte MCLR est désactivée

PWMPIN :

1 = Les pattes associées au module PWM sont configurées en tant E/S logiques après une réinitialisation du processeur.

0 = Les mêmes pattes sont configurées comme sorties PWM après une réinitialisation.

HPOL : configuration de la polarité des sorties PWMxH.

1 = les sorties PWMxH sont actives en état haut.

0 = les sorties PWMxH sont actives en état bas.

LPOL : configuration de la polarité des sorties PWMxL.

1 = les sorties PWMxL sont actives en état haut.

0 = les sorties PWMxL sont actives en état bas.

BORV:

Configuration du voltage qui correspond à une chute de tension de l'alimentation et qui va causer la réinitialisation du processeur.

11 = 2.0V

10 = 2.7V

01 = 4.2V

00 = 4.5V

FPWRT:

Réglage du temps d'attente (PWRT / Power-on Reset Timer) après le rétablissement de la tension d'alimentation usuelle et avant la réinitialisation du système.

11 = PWRT = 64 ms

10 = PWRT = 16 ms

01 = PWRT = 4 ms

00 = Power-up timer arrêté.

Le registre FGS (General Code Segment Configuration Register):

Ce registre est utilisé pour configurer les options sur le code de protection du programme.

IV.3 - Particularités de la programmation C30 :

Le langage utilisé pour programmer les dsPIC est le C30. C'est du C classique auquel viennent se rajouter quelques petites particularités que nous allons détailler. Le logiciel utilisé pour programmer se nomme MPLAB, il est fourni gratuitement par Microchip. Pour le compilateur C30, il est payant, mais il peut être fourni gratuitement aux étudiants.

Voici la liste non exhaustive des spécificités du langage C30 :

```
1/ #include "p30f3010.h"
```

Cette ligne permet d'inclure un fichier qui indique au compilateur C la correspondance entre les noms de registres que nous allons utiliser, et leurs emplacements dans la mémoire du microcontrôleur.

2/

Les lignes de configuration du fonctionnement général du dsPIC qui seront utilisées dans notre programme sont :

```
_FOSC (XT_PLL8);
```

```
_FWDT (WDT_OFF);
```

```
_FBORPOR (RST_PWMPIN & PWMxH_ACT_LO & PWMxL_ACT_HI &  
PBOR_OFF & BORV_27 & PWRT_16 & MCLR_EN);
```

```
_FGS (CODE_PROT_OFF);
```

La ligne la plus intéressante est la ligne « `_FOSC(XT_PLL8)` ». Elle configure le microcontrôleur comme utilisant une horloge externe (XT) avec une PLL x8 (`_PLL8`).

La deuxième ligne configure le module watchdog en état d'arrêt « `_FWDT (WDT_OFF)` »

La troisième ligne configure l'état des sorties PWM après la réinitialisation du système (`RST_PWMPIN`).

Dans la même ligne on remarque que les sorties PWMxH sont configurés pour qu'ils soient actives en état bas (`PWMxH_ACT_LO`) pendant que les sorties PWMxL sont configurés pour qu'ils soient actives en état haut (`PWMxL_ACT_HI`). Les polarités des sorties PWMxH et PWMxL sont inverses l'une par rapport à l'autre ce qui permet de piloter le hacheur à quatre quadrants. On remarque donc que cette ligne de programme est primordiale pour le fonctionnement du système.

Toujours dans la même ligne on choisit la tension BORV = 2.7 à laquelle correspond la réinitialisation du système (`BORV_27`). On remarque aussi la configuration de la période PWRT (`PWRT_16`) qui correspond à 16 ms. Finalement la patte MCLR est configurée en état active (`MCLR_EN`).

La quatrième ligne configure la protection par code comme désactivée « `_FGS (CODE_PROT_OFF)` ».

3/ LATEbits.LATE0 = 1;

Cette ligne positionne le bit LATE0 du registre LATE à 1. D'une manière générale, il convient de lire le nom des bits et des registres dans les Register Map de la datasheet, et d'opérer ainsi pour accéder à la valeur : « [Nom du registre]bits.[Nom du bit] ».

Pour lire la valeur de la patte RF2 configurée comme une entrée, il faut faire :

```
« int x = PORTFbits.RF2 ; ».
```

Pour activer le Timer1 il faut faire :

```
« T1CONbits.TON = 1 ; »
```

4/ La déclaration des sous programmes d'interruption :

```
void __attribute__((interrupt, auto_psv)) _ADCInterrupt ( )
{
    /* Code du programme d'interruption */

    IFS0bits.ADIF = 0 ;
}
```

La commande « __attribute__ » est complètement spécifique au langage C30. Elle permet d'attribuer à la fonction qui s'en suit (ici ADCInterrupt) les propriétés qu'on veut. Ces propriétés sont mises entre des parenthèses doubles. A ce stade pour déclarer une fonction comme interruption on doit lui affecter la propriété ((interrupt)). D'autre coté pour que cette interruption prend en compte les variables globales déclarées dans le programme principal on doit utiliser la syntaxe ((auto_psv)). ADC ici réfère au module convertisseur analogique numérique. La syntaxe générique est la suivante : « void __attribute__((interrupt, auto_psv)) _ [Nom d'interruption] Interrupt(void) ».

Pour connaître la valeur de [Nom d'interruption] il faut se référer à la datasheet (voir annexe).

Il convient ensuite de placer dans ce programme d'interruption le corps du programme, puis de ne surtout pas oublier de remettre le Interrupt Status Flag à 0 pour rendre la main au programme principal à la fin de la fonction.

5/ while(1) ;

Sur un PC, lorsqu'un programme a fini de s'exécuter, il quitte. Or, on ne peut pas « quitter » sur un microcontrôleur. Il faut donc impérativement que lorsque le programme principal se termine (main) il arrive sur une boucle infinie (while (1)) : au lieu de quitter, on lui demande simplement de ne rien faire indéfiniment. Si vous oubliez cette boucle infinie, le microcontrôleur va se mettre à faire n'importe quoi de façon aléatoire.

IV.4 - Déroulement et temps d'exécution du programme :

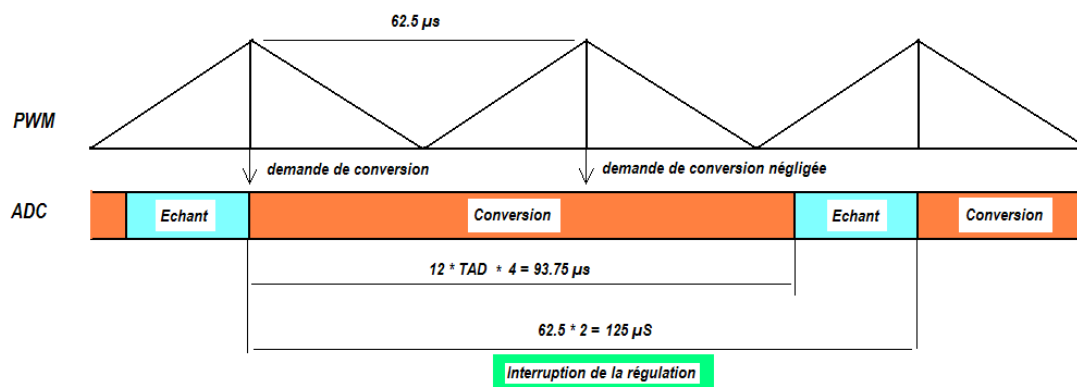


Figure IV.18 : Temps d'exécution de l'interruption de commande.

Le déroulement du programme de commande sera synchrone avec l'interruption de régulation. Cette interruption est déclenchée par le module ADC chaque $125 \mu\text{s}$ à la fin de deux périodes de conversion successives. une telle valeur est justifiée par la synchronisation du déclenchement de l'étape de conversion du module ADC avec le signal PWM (de période $62.5 \mu\text{s}$) et le fait que l'étape de conversion prend un temps de $93.75 \mu\text{s}$ qui est supérieur à la période du signal PWM, une demande de conversion sera donc négligée à chaque fois parce que l'opération de conversion en cours n'est pas encore achevée . On utilisant des compteurs de valeurs distinctes on peut aussi synchroniser le programme de régulation PD ($T_e = 1 \text{ ms} = 8 * 125 \mu\text{s}$) et les sous programmes de la génération du signal carré et triangulaire.

Pour bien illustrer les choses voici les points qui entre en jeu à l'établissement du temps d'exécution de chaque sous-programme :

- L'horloge interne du dsPIC est dérivée à partir d'un oscillateur externe de 10MHz et on utilisant le module PLLx8 on aura $F_{CY} = 10 * 8 / 4 = 20$ MIPS.

- La période du signal PWM est de 16 kHz (62.5 μ s). le signal modulant est de type triangulaire.

- Le convertisseur analogique numérique fonctionne avec une horloge RC interne de fréquence $F_{AD} = 512$ kHz et qui indépendante de l'horloge du processeur. Aussi ce module fonctionne avec les 4 canaux qu'il se dispose (bits CHPS = 10 du registre ADCON2) mais deux seulement seront utilisés.

- Le convertisseur analogique numérique fonctionne sur deux étapes: l'étape d'échantillonnage et l'étape de conversion. Le déclenchement de cette dernière est synchrone avec le signal PWM (bits SSRC = 011 du registre ADCON1) et se fait à chaque valeur max du signal triangulaire (registre SEVTCMP=PTPER).

- L'étape de conversion prend un temps qui est égale à ($12 T_{AD} * 4 = 93.75$ μ s). ce temps est supérieur à la période du signal PWM, c'est pour ça qu'il y aura des demandes de conversion qui seront négligées par le module ADC entre deux périodes PWM. Ce qui résulte une étape de conversion chaque deux périodes ($62.5 * 2 = 125$ μ s). d'autre coté l'étape d'échantillonnage suivante va prendre tout le temps entre la fin de la conversion précédente et la nouvelle conversion.

- L'interruption de commande qui est déclenchée par la fin de chaque conversion contient tout le programme de commande. Un compteur de valeur 4 est utilisé pour établir la fréquence d'échantillonnage ($4 * 125$ μ s = 0.5 ms). C.-à-d. que le programme de régulation PD est exécuté chaque quatre interruptions.

- La génération des signaux carré et triangulaire se fait en utilisant aussi des compteurs de la même méthode que la précédente. On peut jouer sur la fréquence d'un signal en modifiant la valeur du compteur.

Pour lire le programme de commande complet voir l'annexe A

IV.6 - Outils de développement :

Pour la mise en mémoire du programme (flashage du dsPIC) il ya plusieurs gammes d'outils de développement suggéré par Microchip ou par d'autres firmes.

Microchip propose le programmeur / débogueur ICD2 (In-Circuit Debugger), qui permet de faire le flashage du dsPIC sans avoir être enlevé du circuit imprimé. Un autre programmeur est le Pickit 2 qui très recommandé pour les amateurs de l'électronique. En effet, il coûte 6 fois moins cher pour les mêmes fonctionnalités. Il a de plus un connecteur qui est au format SIL – qui est très courant – au lieu de la contraignante fiche RJ11 de l'ICD2. Les membres de la famille PIC et dsPIC, que ces deux modules permet de programmer, s'est largement étoffée et couvre un très grand nombre de cibles.

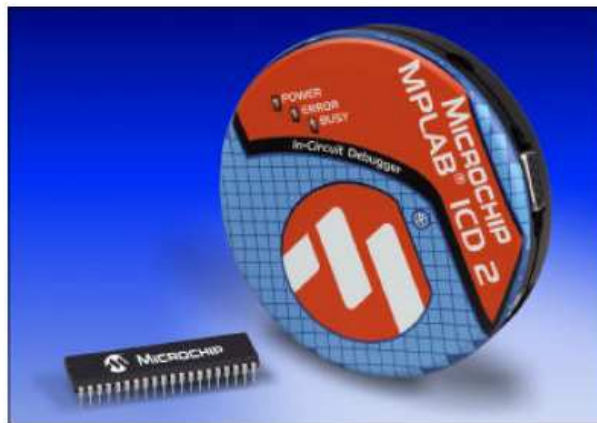


Figure IV.19 : Package du programmeur ICD2

L'environnement de développement intégré (IDE) MPLAB regroupe tous les outils nécessaires à la mise au point d'une application avec un cœur de microcontrôleur dsPIC, entre autres, de Microchip :

- Editeur de texte interactif.
- Compilateur C30 (et assembleur).
- Simulateur.
- Debugger si on dispose de l'équipement nécessaire.

D'autre part MikroElektronika propose sa propre gamme de programmeurs. Les systèmes de développements EASYdsPIC3 et EASYdsPIC4 permettent la programmation, le débogage ainsi que la simulation des différentes familles dsPIC.

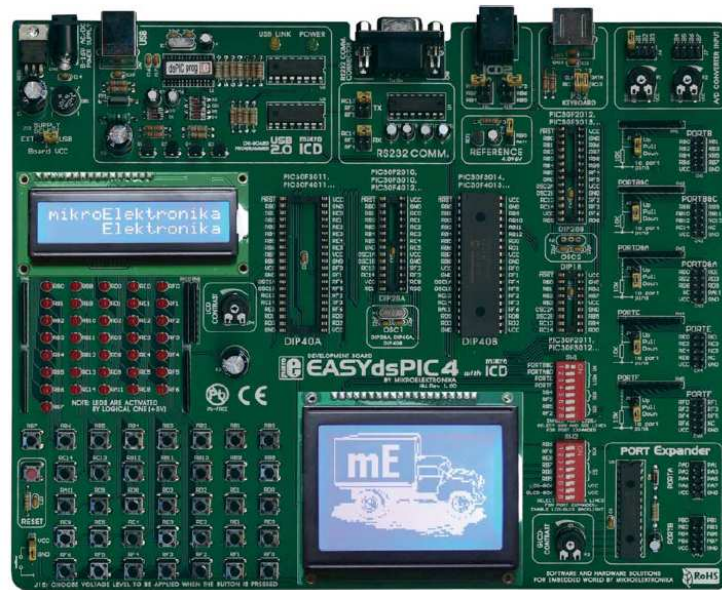


Figure IV.20 : Carte de développement EASYdsPIC4 de MikroElektronika.

Ces cartes de développement sont munies d'entrées logiques et analogiques, d'écrans LCD et de LED pour permettre de vérifier les sorties mais aussi d'interfaces qui permettent de relier d'autre périphériques, tout cela a pour but de faciliter la simulation des programmes. MikroElektronika propose aussi son propre logiciel de développement avec possibilité de programmer en C et en assembleur. (Mikro C, Mikro Assembler).

IV.7 - Conclusion :

La programmation du dsPIC est une tâche relativement facile, surtout avec l'utilisation de langage évolué approprié comme le C30. L'architecture interne du dsPIC à encore faciliter cette tâche grâce à sa structure en registre ce qui permet de configurer chaque périphérique séparément.

Le dsPIC offre une multitude de solutions pour un problème donné : choix de différents types d'horloges, de timers, plusieurs modes de fonctionnement du module ADC et du PWM, cela rend la tâche plus facile sur les développeurs et permet d'adapter et d'étendre leurs programmes selon les contraintes auxquelles ils seront faces.

***CONCLUSION ET
PERSPECTIVES***

Conclusion générale

L'objectif de ce travail était la réalisation et la commande d'un système à lévitation magnétique. Nous avons choisi de léviter un objet en aimant permanent avec l'utilisation de capteurs à effet Hall comme détecteurs de position, et pour l'implémentation de la commande, notre choix s'est posé sur la solution numérique basée sur le dsPIC 30F3010 qui intègre le plus de périphériques possibles.

La construction de la maquette était la tâche la plus importante et aussi celle qui nous a demandé le plus de temps. En fait, la recherche des composants s'est avérée éprouvante, surtout avec l'absence de capteurs à effet hall efficaces dans le marché et de fournisseurs de bobine qui peuvent être utilisées dans ce genre de réalisation.

D'autre coté nous avons remarqué l'importance d'avoir un savoir faire et une main experte lors des différents étapes de réalisation. A titre d'exemple, le soudage des composants peut s'avérer très complexe pour un débutant dans ce domaine surtout pour les composants qui présentent beaucoup de pattes (dsPIC, Hacheur) où les pistes se rapprochent et passent sous les composants.

Nous avons constaté aussi l'importance que prends les logiciels de conception des circuits électroniques pour diminuer l'effort et les délais de la conception et la fabrication de telles maquettes. Une connaissance sur l'utilisation de tels logiciels est indispensable pour la formation d'un ingénieur automatique, surtout que ces logiciels sont devenus très disponibles et accessibles aux ingénieurs ainsi qu'aux étudiants.

Le système à régulé est un système instable et qui présente une grande non-linéarité. Pour obtenir le modèle du système nous avons choisi une approche de modélisation classique basée sur l'écriture des lois physique qui concerne la force magnétique et des lois générales de la mécanique et de l'électricité. Il est très clair qu'une telle approche présente des limites, surtout quand on travail avec des hypothèses simplificatrices qui négligent de prendre en compte les phénomènes physiques secondaires. En fait dans ce genre de système, l'identification donne une plus grande précision sur les coefficients et les constantes du modèle mathématique en prenant en compte plus de dynamiques.

Pour la synthèse des correcteurs, nous avons donné un régulateur PD classique, qui malgré tout, prouve toujours son efficacité et reste employé dans tous les secteurs industriels. Les avantages d'un tel correcteur et des correcteurs linéaires en général sont la facilité de mise en œuvre et de l'implémentation et la réduction du temps de calcul. D'autre part ces commandes se présentent très sensibles à la variation paramétrique. Ce qui implique le passage aux régulateurs non linéaires pour obtenir des commandes plus robustes.

Pour la partie programmation, l'utilisation d'un langage évolué tel que le C est la meilleure solution pour implémenter les lois de commande et pour éviter la complexité du langage assembleur. La programmation temps réelle présente des particularités liées aux temps de calcul et de la période d'échantillonnage qui doivent être respectés après avoir être calculés soigneusement.

Perspectives :

Ce travail, par sa nature pratique, présente un issu à d'autres travaux d'ordres pratique ou théorique qui peuvent être réalisés pour l'accomplir, car nous croyons qu'un travail humain n'est jamais parfait et qu'il présente toujours des lacunes quelque soit sont degré de perfection.

En réalisant la maquette, nous avons toujours gardé un esprit évolutif, cela ce manifeste par le fait que nous avons pris en considération les possibilités d'extension et de développent de la maquette, en laissant de l'espace pour que d'autres composants peuvent être ajoutés au circuit de commande afin d'améliorer son fonctionnement. Parmi les possibilités d'extension du circuit de commande on peut citer :

- Un connecteur SIL5 peut être ajouté afin de permettre la programmation du dsPIC sans l'avoir enlever de son circuit de commande en utilisant un programmeur du type ICD2 ou Pickit2.
- Nous avons laissé une place pour ajouter un composant MAX3235E et un connecteur SIL4, cela va permettre d'établir la liaison série RS232 entre le dsPIC et un PC en utilisant le module UART du dsPIC et le port série (DB9) du PC.

D'autre coté, on suggère que cette maquette soit utilisée par d'autres étudiants pour effectuer des TP, voir même des PFE, et sur laquelle ils peuvent valider les lois de commandes qu'ils ont développé, et vérifier la validité des notions théoriques développées aux cours.

Pour obtenir un modèle mathématique plus performant, nous suggérons d'effectuer une identification sur le système, au lieu de la modélisation analytique que nous avons entamé. Autre possibilité est d'utiliser des logiciels de modélisation comme le logiciel FEMM qui permet de modéliser les phénomènes électromagnétiques avec une grande précision. Cela va permettre de synthétiser des lois de commandes plus fiables.

Une fois une liaison est établie entre la maquette et le PC d'autres possibilités peuvent être envisagées :

- La création d'une interface logicielle sous l'environnement Windows qui donne la possibilité de permuter entre plusieurs lois de commande, plusieurs consignes, de pouvoir changer les paramètres d'un régulateur donné ...etc.
- Utiliser l'environnement Matlab / Simulink pour effectuer une commande temps réelle en utilisant une carte d'acquisition appropriée et les Toolbox que propose Simulink pour la commande en temp réel (Stateflow, Real Time Workshop, Real Time Windows Target, xPC Target, Virtual Reality Toolbox...)

Finalement cette maquette constitue le point de départ pour la construction d'autres maquettes en relation avec le phénomène de la lévitation magnétique, par exemple : maquette d'un train à lévitation magnétique, plate forme à suspension magnétique à plusieurs degrés de liberté...etc. On peut aussi entamer les phénomènes de lévitation magnétique reliés au phénomène de supraconductivité, un domaine qui reste négligé et mal envisagé dans les universités de notre pays.

ANNEXE



ANNEXE A

Programme de commande

```
//-----
//B.Badreddine & H.OUACENE
//Ecole Nationale Supérieure Polytechnique
//Département génie électrique, laboratoire commande des processus.
//5ème Année Automatique
//Commande de bobine de levitation magnétique
// dsPIC 30F3010, capteur ugn 3503
// 2 Hacheurs 4Q dans le L298, mis en //
//horloge externe en quartz de 10 Mhz
// 20 MIPS, PWM 16kHz, ADC/ISR 8 kHz ie 125 us
//-----
#include "p30F3010.h"

//Configuration générale
_FOSC(CSW_FSCM_OFF & XT_PLL8); //10Mhz *8 = 80 MHz /4 = FCY = 20 MIPS maxi pour
ce pic

_FWDT(WDT_OFF);

_FBORPOR(RST_PWMPIN & PWMxH_ACT_LO & PWMxL_ACT_HI & PBOR_OFF & BORV_27 &
PWRT_16 & MCLR_EN);

_FGS(CODE_PROT_OFF);

//Constantes globales
#define FPWM 16000 //freq PWM*2

#define HalfDUTY 624 // set the pwm half duty = PTPER = 624=FCY/FPWM/2-1;

#define FullDUTY HalfDUTY*2 // car PDCx a une double resolution /PTPER

#define samp 4 // Compteur pour la régulation 4* 125us = 0.5 ms
```

```

#define RunningLED                LATBbits.LATB0 // Output

#define EnableL8                   LATEbits.LATE0 // la pin PWM1 low utilise en Enable
du L298

#define Vh 740 // limite haute des signaux carre et triangulaire

#define Vl 540 // limite basse des signaux carre et triangulaire

#define carre 16000 // compteur pour le signal carre 16000*125us= 2s

#define triangle 40 // compteur pour signal triangle 40*125us=5ms.. 5ms*(Vh-Vl)=1s

int Ref; // reference (via pot 10kOhm), lu sur l ADC AN2/RB2 ch0

int Mes; // distance mesure par le capteur, lu sur l ADC AN3/RB3 ch1

int Vref; // tension de sortie = rapport cyclique

int Vreflimiteur, Vrefmin, Vrefmax; // limiteur de tension de sortie par limitation du
rapport cyclique

int Kp, Ki, Kd, e, Olde, xe_reg, countReg;

int choice, dir, countsq, counttr ;

//-----
//   initialisation des variables
//-----

void InitVar()

{
    Kp=70000;

    //Ki=0; // regulateur PD, Ki pour illustrer seulement.

    Kd=700000;

    //xe_reg=0; // lorsque Ki non nul

    countReg=0;

    Olde=0;

    Vrefmin=0; // limiteur de tension bas

    Vrefmax=FullDUTY; // limiteur de tension haut

    RunningLED=0; // LED run = On

```

```

dir=-1;    // direction initialise vers le bas

countsq=0;

counttr=0;

}

//-----

//    Configuration des ports

//-----

void setup_ports()

{

    ADPCFG = 0xFFF3; // all PORTB = Digital(1) ;RB2 to RB3 = analog(0) ie 1111 0011

    // Clear All Ports Prior to defining I/O

    PORTB=0;    //Initialize LED pin data to off state

    PORTC=0;

    PORTD=0;

    PORTE=0;

    // Now set pin direction registers

    TRISB = 0x0006;    // RunningLED RB0 out, RB2-3 inputs ADC.

    TRISC = 0xFFFF;    // RC13 et RC14 inputs.

    TRISD = 0x0000;    // not connected.

    TRISE = 0x0000;    // RE1, RE2 are the PWMs -> outputs, EnableL298 RE0 out.

}

//-----

//    config du module ADC :

//    2 entrees de conversion

//    PWM trigger commence la conversion

//    l'interrupt ADC est activee

//-----

```



```

void InitADC10()
{
    ADCON1 = 0x006F;    // PWM declenche la conversion, Echant simultanes, ADC
Off pour configuration

    ADCON2 = 0x0200;    // 4 canaux de conversion, ADC INTerrupt a chaque EOC=100
us (End Of Conversion).

    ADCHS = 0x0022;    // AN2/RB2 Ch0 Ref, AN3/RB3 Ch1 Mes, AN4/RB4 Ch2 NC,
AN5/RB5 Ch3 NC

    ADCON3 = 0x0080;    // Tad = horloge RC interne

IFS0bits.ADIF = 0;    // Adc int flag Off

IEC0bits.ADIE = 1;    // Adc int On

    ADCON1bits.ADON = 1; // turn ADC ON
}

//-----
// InitMCPWM, config du PWM comme suit:
// FPWM = 16 khz voir en haut
// PWMs Independants
// La conversion ADC est synchrone avec le PWM
//-----

void InitMCPWM()
{
    PTPER = HalfDUTY;    // set the pwm period register, ne pas oublier la
double precision

    PWMCON1 = 0x0312;    // mode indep pour pwm1 et pwm2, activer PWM1H
& PWM2L.

    EnableL298=0;        // disable L298

    PDC1=HalfDUTY; PDC2=HalfDUTY; // init sans rien, apres une regul ca change

    OVDCON = 0xFFFF;    // Cmde MLI, aucun effet de OVDCON

    SEVTCMP = PTPER;    // set ADC to trigger at PTPER value.

```

```

    PWMCON2 = 0x0000;          // mise a jour synchrone avec la base du temps

    PTCON = 0x8002;           // cnfig PWM signal modulant.
}

//-----
//    L'iterrupt ADC se declenche apres lecture des entrees analog .
// tt est synchrone % a cette int de ADC int =2*PWMperiod=2*62.5 us=125 us
//-----

void __attribute__((interrupt, auto_psv)) _ADCInterrupt ()
{
    RunningLED=1;

    IFS0bits.ADIF = 0;

    Mes=ADCBUF1;

if (PORTCbits.RC13==1) //signal carre
{
    countsq=countsq+1;

    if(countsq<=carre) Ref=Vh; // niveau haut

    if(countsq>carre && countsq<=2*carre) Ref=Vl; //niveau bas

    if(countsq>2*carre) // reinitialisation
    {
        countsq=1;

        Ref=Vh;

    }

if (PORTCbits.RC13==0 && PORTCbits.RC14==1) ;// signal triangulaire
{
    if(++counttr>triangle) //copmteur d increment°
    {
        counttr=0;

```

```

    Ref=Ref+dir; //incr decr de 1
    if(Ref==Vh) dir=-1; // direction bas
    if(Ref==Vl) dir=1 ;//direction haut
    }
    }

else    Ref=ADCBUF0; // referece du potentiometre

// reg PD

// execute toutes les 4 * 125 us = Ts= 0.5 ms

    if ( ++countReg==samp )
        {
            countReg=0;
            // regul angle
//          e=Ref-Mes;          // erreur a forcer a zero
//          e=-Ref+Mes;        // erreur a forcer a zero
            Vref =Kp*e + xe_reg + Kd*(e-Olde) + HalfDUTY;

// les lignes qui suivent c'est dans le cas de Ki non nul
//          xe_reg += Ki*e;
//          if (xe_reg<-FullDUTY)      xe_reg=-FullDUTY;
//          if (xe_reg>FullDUTY)      xe_reg=FullDUTY;
//.....
            if (Vref<Vrefmin)    Vref=Vrefmin; //limitation
            if (Vref>Vrefmax)    Vref=Vrefmax;

            Olde=e;

            // mise a jour du rapport cyclique

            PDC1=Vref;          // alim de la bobine via L298
            PDC2=Vref;          // alim de la bobine via L298
        }

```

```

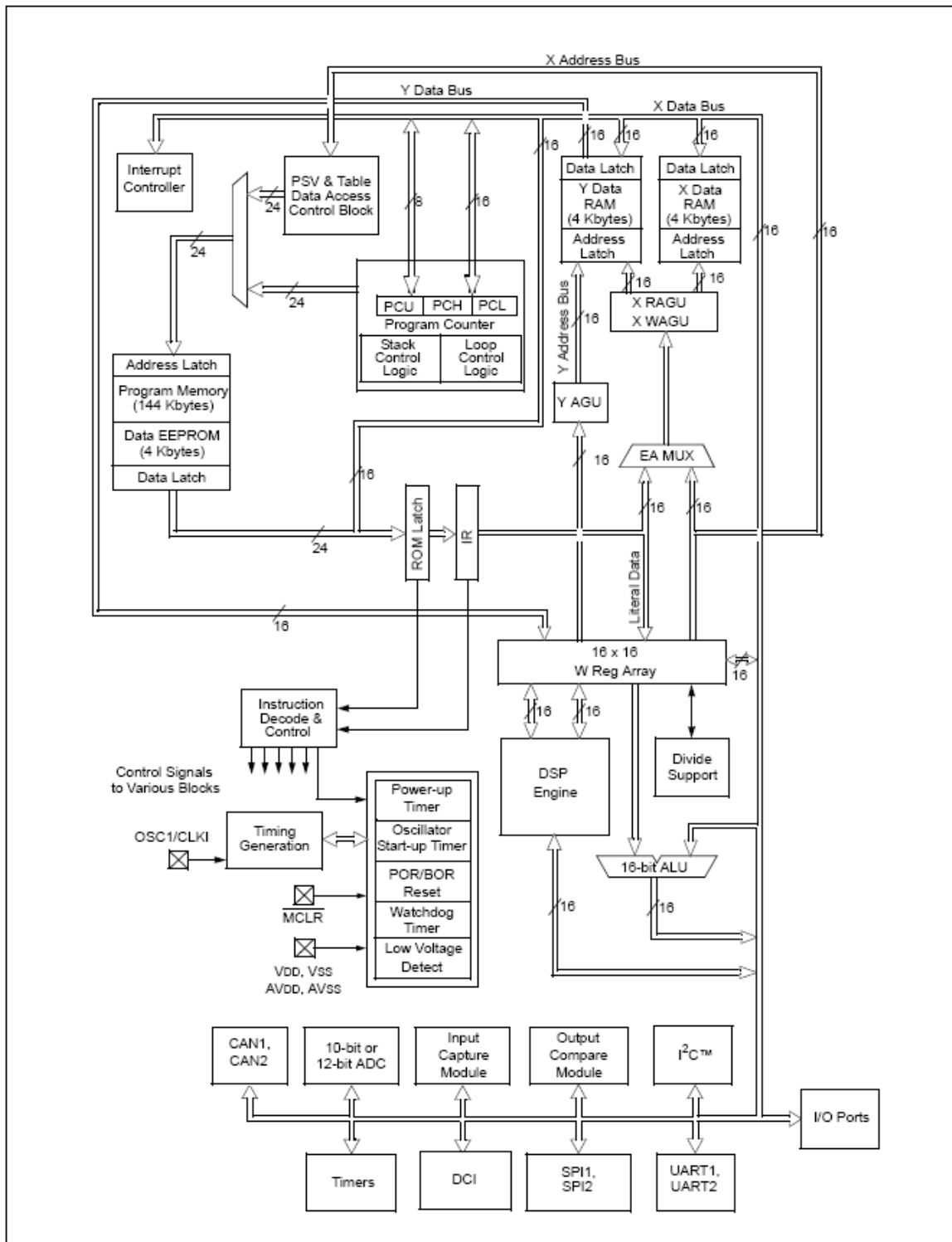
}
//-----
// Programme principal
int main()
{
    setup_ports();
    InitVar();
    InitUART();
    InitMCPWM();
    InitADC10();
    RunningLED=0;
    while(1)
        {
            EnableL298=1; // enable L298 principal
            RunningLED=1; // LED run = On
        } // end of while (1)
} // end of main
//-----
//Interruptions d'erreurs
//-----
//Oscillator Fail Error trap routine
void __attribute__((interrupt, auto_psv)) _OscillatorFail(void)
{
    EnableL298=0; // disable L298
    RunningLED=0;
    while(1); //Wait forever
}

```

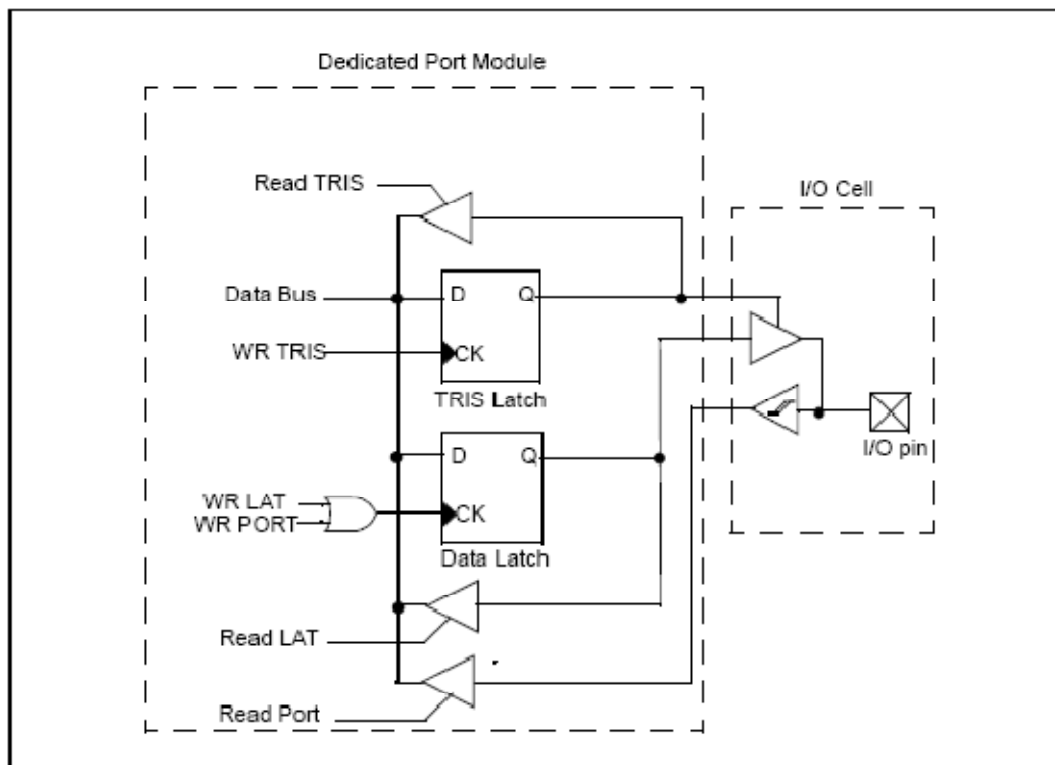
```
//-----  
//Address Error trap routine  
void __attribute__((interrupt, auto_psv)) _AddressError(void)  
{  
    EnableL298=0; // disable L298  
    RunningLED=0;  
    while(1); //Wait forever  
}  
//-----  
//Stack Error trap routine  
void __attribute__((interrupt, auto_psv)) _StackError(void)  
{  
    EnableL298=0; // disable L298  
    RunningLED=0;  
    while(1); //Wait forever  
}  
//-----  
//Math (Arithmetic) Error trap routine  
void __attribute__((interrupt, auto_psv)) _MathError(void)  
{  
    EnableL298=0; // disable L298  
    RunningLED=0;  
    while(1); //Wait forever  
}
```

ANNEXE B

Architecture interne du F3010



Utilisation des registres LATx, PORTx et TRISx :



TRIS Registers

The TRISx register control bits determine whether each pin associated with the I/O port is an input or an output. If the TRIS bit for an I/O pin is a '1', then the pin is an input. If the TRIS bit for an I/O pin is a '0', then the pin is configured for an output. An easy way to remember is that a '1' looks like an I (input) and a '0' looks like an O (output). All port pins are defined as inputs after aReset.

PORT Registers

Data on an I/O pin is accessed via a PORTx register. A read of the PORTx register reads the value of the I/O pin, while a write to the PORTx register writes the value to the port data latch. Many instructions, such as BSET and BCLR instructions, are read-modify-write operations. Therefore, a write to a port implies that the port pins are read, this value is modified, and then written to the port data latch. Care should be taken when read-modify-write commands are used on the PORTx registers and when some I/O pins associated with the port are configured as inputs. If an I/O pin configured as an input is changed to an output at some later time, an unexpected value may be output on the I/O pin. This effect occurs because the read-modify-write instruction reads the instantaneous value on the input pin and loads that value into the port data latch.

11.2.3 LAT Registers

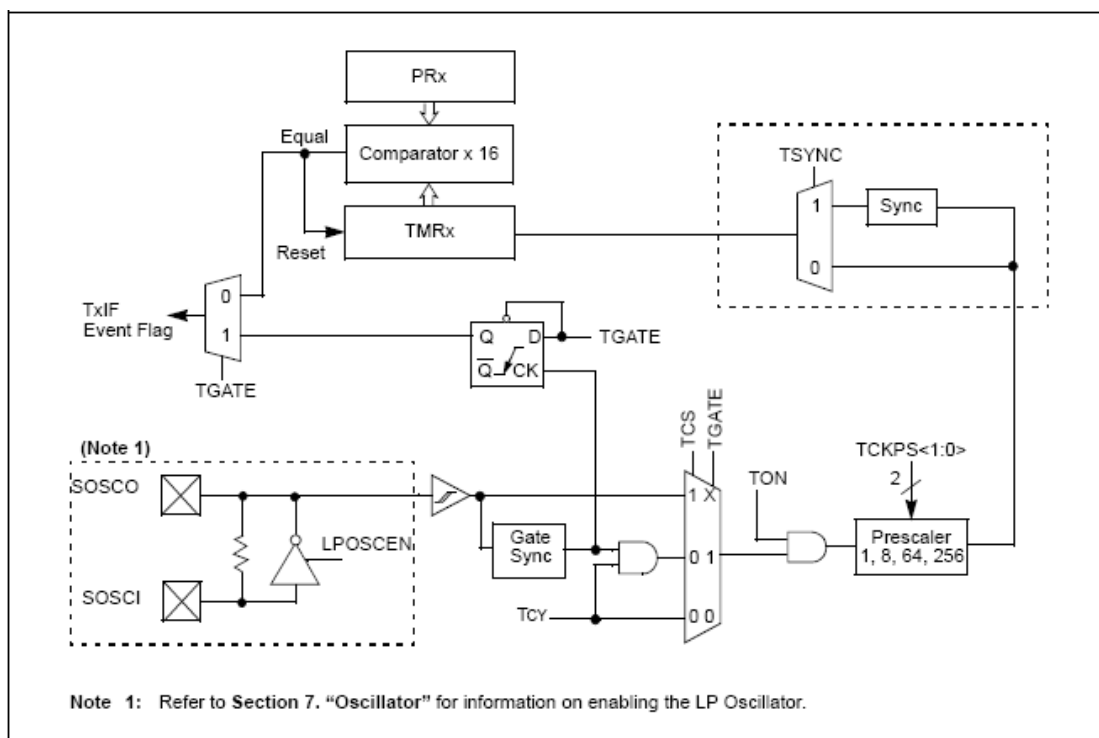
The LATx register associated with an I/O pin eliminates the problems that could occur with

read-modify-write instructions. A read of the LATx register returns the values held in the port output latches, instead of the values on the I/O pins. A read-modify-write operation on the LAT register, associated with an I/O port, avoids the possibility of writing the input pin values into the port latches. A write to the LATx register has the same effect as a write to the PORTx register.

The differences between the PORT and LAT registers can be summarized as follows:

- A write to the PORTx register writes the data value to the port latch.
- A write to the LATx register writes the data value to the port latch.
- A read of the PORTx register reads the data value on the I/O pin.
- A read of the LATx register reads the data value held in the port latch.

Architecture interne du timer1 :



Architecture interne du module PWM :

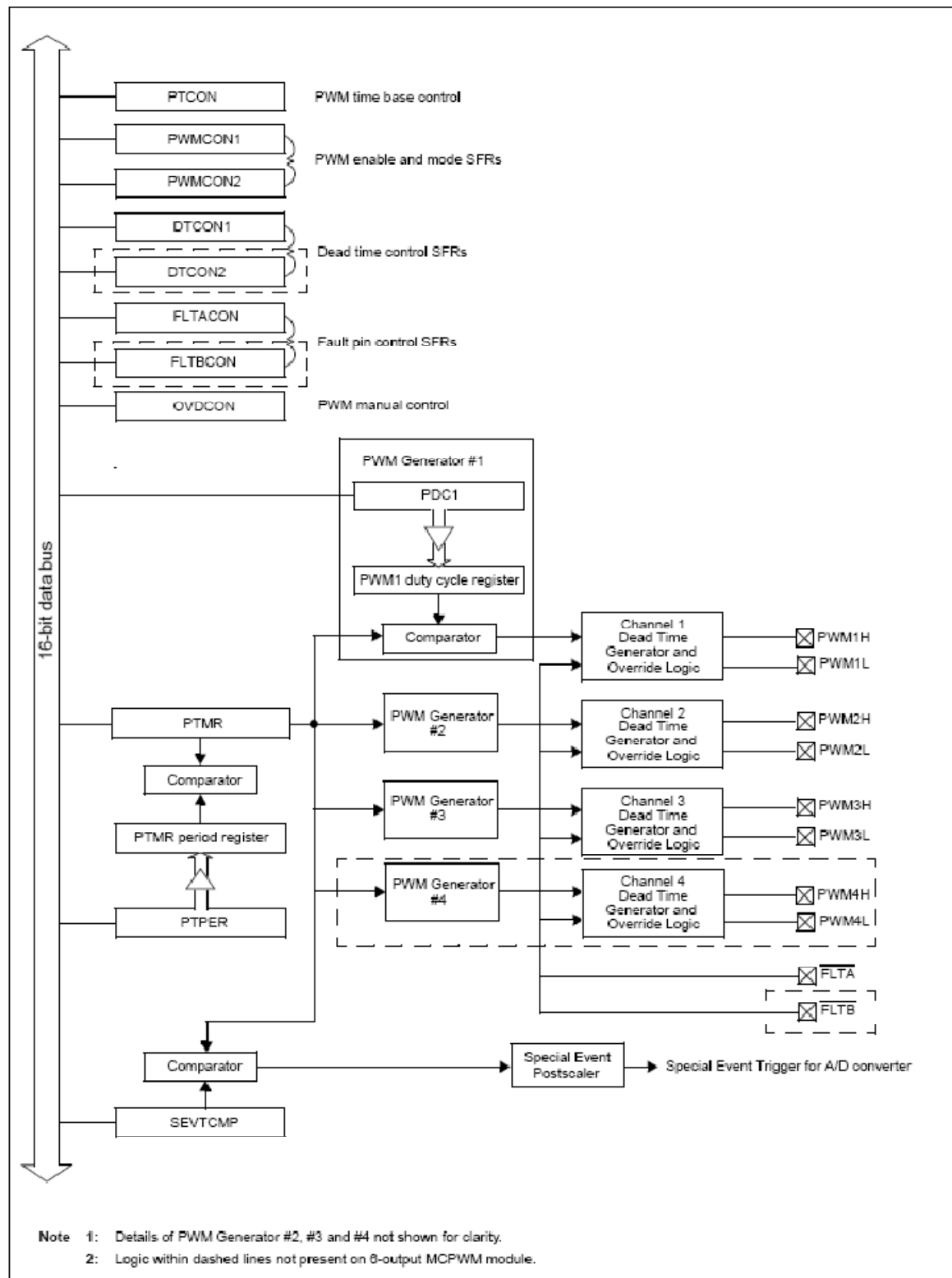


Table des interruptions

INT Number	Vector Number	Interrupt Source
Highest Natural Order Priority		
0	8	INT0 – External Interrupt 0
1	9	IC1 – Input Capture 1
2	10	OC1 – Output Compare 1
3	11	T1 – Timer 1
4	12	IC2 – Input Capture 2
5	13	OC2 – Output Compare 2
6	14	T2 – Timer 2
7	15	T3 – Timer 3
8	16	SPI #1
9	17	U1RX – UART1 Receiver
10	18	U1TX – UART1 Transmitter
11	19	ADC – ADC Convert Done
12	20	NVM – NVM Write Complete
13	21	SI2C – I ² C™ Slave Interrupt
14	22	MI2C – I ² C Master Interrupt
15	23	Input Change Interrupt
16	24	INT1 – External Interrupt 1
17	25	IC7 – Input Capture 7
18	26	IC8 – Input Capture 8
19	27	OC3 – Output Compare 3*
20	28	OC4 – Output Compare 4*
21	29	T4 – Timer 4
22	30	T5 – Timer 5
23	31	INT2 – External Interrupt 2
24	32	U2RX – UART2 Receiver*
25	33	U2TX – UART2 Transmitter*
26	34	Reserved
27	35	Reserved
28	36	Reserved
29	37	Reserved
30	38	Reserved
31	39	Reserved
32	40	Reserved
33	41	Reserved
34	42	Reserved
35	43	Reserved
36	44	Reserved
37	45	Reserved
38	46	Reserved
39	47	PWM – PWM Period Match
40	48	QEI – QEI Interrupt
41	49	Reserved
42	50	Reserved
43	51	FLT \bar{A} – PWM Fault A
44	52	Reserved
45-53	53-61	Reserved
Lowest Natural Order Priority		

* Available on dsPIC30F3011 only

Références et Bibliographie

Références :

[1] Adam Kumpf : Magnetic Levitation System, Massachussets institute of technology (MIT) USA, décembre 2003.

[2] Zhenyu Yang, Gerulf K.M. Pedersen and Jørgen H. Pedersen : Model-Based Control of a Nonlinear One Dimensional Magnetic Levitation with a Permanent-Magnet Object -*Esbjerg Institute of Technology, Aalborg University Denmark.*

[3] Lotfi BAGHLI, Abderrezak REZZOUG : Lévitiation magnétique, une approche objet-projet, université de Nancy, Bruxelles CETISIS'2008.

[4] I.Hassanzadeh, S.Mobayen,G.Sedaghat :Design and implementation of controller for magnetic levitation system using genetic algorithms,University of Tebriz,Iran.

[5] Dspic 30F3010/3011 datasheet , Microchip Technology Inc, 2008.

[6] Trent M. Gues and David G. Alciatore Model Development and Control Implementation for a magnetic Levitation Apparatus, Dept of Mechanical eng. Colorado state University,1995.

[7] E.lambert, Cours d'électromagnétisme, phénomènes quasi-statique, eivd ,1999.

[8] MICHEL Pinard, commande électronique des moteurs électrique, Dunod ; paris 2004.

[9] Dimitri Peaucelle, système à temps discret, Commande numérique des procédés, Cours destiné à des élèves ingénieurs. Université Ibn Zohr, Agadir, Maroc. 2003.

[10] Bharathwaj Muthuswamy and Kevin Peterson, Design of Magnetic Levitation Controllers Using Jacobi Linearization, Feedback Linearization and Sliding Mode Control, ME237 SPRING 2008.

[11] Q. H. Guo, D. S. Li, Z. Y. Lu, W. Wang, and L. Z. Ye, Research on a Maglev Ball Control System Based on DSP2812, PIERS Proceedings, Hangzhou, China, March 24-28, 2008.

[12] Freddy Mudry, Signaux et systèmes : 7ème partie Modélisation et régulation, Ecole d'ingénieurs de Canton de Vaud, ei.vd 2008.

[13] A. E. Hajdjaji and M. Ouladsine, "Modeling and nonlinear control of magnetic levitation systems," IEEE Transactions on Industrial Electronics, August 2001.

[14] Hansruedi Buhler , conception de systèmes automatique, Presses Polytechniques Romandes,1988.

[15] Louis Maret, Régulation automatique Systèmes analogiques, Presses Polytechniques Romandes, Paris 1988.

Bibliographie :

[1] Harris Benson, Physique 2 : Electricité et Magnétisme, De Boeck 2004

[2]Gérard Meunier, modèles et formulations en électromagnétisme, bermes, Lavoisier paris 2002

[3] R. Langlois Berthelot, étude Electromagnétique générale des machines électriques, Edition EYROLLES paris 1945

[4] M. Mokhtari, M. Marie, Application de MATLAB 5 et SIMULINK 2, Springer-Verlag, France 1998

[5] Claude Foulard, Sylviane Gentil, Jean-Paul Sandraz, commande et régulation par calculateur numérique, EYROLLES paris 1984.

[6] Marwan K. Abbadi, Modeling and Control of a Magnetic Levitation System, Department of Electrical and Computer Engineering-Bradley University, 2004

[7] Michel de Mathelin, Iulia Bara et Jacques Gangloff, Commande Numérique des Systèmes Cours 2ème partie, Ecole Nationale Supérieure de Physique de Strasbourg

[8] H Boudali, R D Williams and T C Giras, A Simulink simulation framework of a MagLev model, School of Engineering and Applied Science, University of Virginia, Charlottesville, Virginia, USA