ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

MÉMOIRE PRÉSENTÉ À
L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

COMME EXIGENCE PARTIELLE
À L'OBTENTION DE LA
MAÎTRISE EN TECHNOLOGIE DES SYSTÈMES
M. ING.

PAR
ZOUHAIR EL GUETIOUI

CONCEPTION, RÉALISATION ET IMPLANTATION D'UN CONTRÔLEUR POUR UN ROBOT MODULAIRE.

MONTRÉAL, LE 23 OCTOBRE 2002

CE MÉMOIRE A ÉTÉ ÉVALUÉ

PAR UN JURY COMPOSÉ DE :

- Mr Maarouf Saad, professeur au Département de génie électrique à l'École de Technologie Supérieure
- Mme Ouassima Akhrif, professeur au Département de génie électrique à l'École de Technologie Supérieure.
- M. Guy Gauthier, professeur au Département de génie de production automatisée à l'École de Technologie Supérieure.
- Mr Charles Khairallah, président Robotics Design.

IL A FAIT L'OBJET D'UNE PRÉSENTATION DEVANT UN JURY ET UN PUBLIC

LE 28 JUIN 2002

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

CONCEPTION, RÉALISATION ET IMPLANTATION D'UN CONTRÔLEUR POUR UN ROBOT MODULAIRE.

Zouhair Elguetioui

SOMMAIRE

Ce projet vise la réalisation d'un prototype pour la commande et le contrôle des robots modulaires hyper redondants. Il s'agit particulièrement de faire la conception, la réalisation et l'implantation de la partie électronique nous permettant d'interagir correctement avec le robot.

La réalisation de ce projet passe avant tout, par une étude des machines à courant continu utilisées comme actionneurs et leurs asservissements du couple qui représente la grandeur de commande ainsi qu'une étude des systèmes numériques tels que les DSP et les microcontrôleurs pour l'intégration des algorithmes développés au cours de ce projet, à savoir les protocoles de communications séries et parallèles, les boucles de régulation, les conversions analogiques numériques, l'algorithme de génération de trajectoire et les programmes de capture de position.

Ce travail nous permettra une communication plus flexible avec les robots. D'autres recherches qui sont, à ce jour, théoriques, pourront être validées expérimentalement.

CONCEPTION, RÉALISATION ET IMPLANTATION D'UN CONTRÔLEUR POUR UN ROBOT MODULAIRE.

Zouhair Elguetioui

SOMMAIRE

Ce projet vise la réalisation d'un prototype pour la commande et le contrôle des robots modulaires hyper redondants. Il s'agit particulièrement de faire la conception, la réalisation et l'implantation de la partie électronique nous permettant d'interagir correctement avec le robot.

La réalisation de ce projet passe avant tout, par une étude des machines à courant continu utilisées comme actionneurs et leurs asservissements du couple qui représente la grandeur de commande ainsi qu'une étude des systèmes numériques tels que les DSP et les microcontrôleurs pour l'intégration des algorithmes développés au cours de ce projet, à savoir les protocoles de communications séries et parallèles, les boucles de régulation, les conversions analogiques numériques, l'algorithme de génération de trajectoire et les programmes de capture de position.

Lors de ce projet, nous avons conçu les cartes électroniques d'entrées/sorties qui s'interfacent entre le processeur numérique (DSP) et le robot. Chaque carte contient le circuit encodeur nous permettant de savoir à tout moment la position du robot et le circuit PWM pour appliquer la commande aux actionneurs.

Ce travail nous permettra une communication plus flexible avec les robots. D'autres recherches qui sont, à ce jour, théoriques, pourront être validées expérimentalement.

REMERCIEMENTS

Je tiens en premier lieu à exprimer ma sincère gratitude à mon directeur de projet M. Maarouf Saad professeur au département de génie électrique. Il m'a tout le temps guidé dans mes recherches en mettant ses efforts et son temps pour aboutir ce travail. De même, j'aimerai témoigner ma reconnaissance à ma co-directrice Mme Ouassima Akhrif, professeure au département de génie électrique et à Mr Charles Khairallah.

Je remercie toute ma famille, surtout mon père et ma mère ainsi que mes frères et soeurs, qui m'ont tous encouragé et soutenu durant les années de mes études.

Un remerciement sincère est adressé à Tarik Ouahidi, Hatim Mekouar, Hassan Abdelkader et Ahmed Essfarjel pour leur aide et soutien.

Un grand merci pour tous mes professeurs et à toutes les personnes qui, d'une façon ou d'une autre, ont contribué à l'aboutissement final de mes études.

TABLE DES MATIÈRES

LISTE DES FIGURES

Page

LISTE DES ABRÉVIATIONS ET SIGLES

Bmoy	Induction magnétique moyenne
C	Couple du moteur
DSP	Digital signal processor
E	La force électromotrice
Id	Courant dans l'induit
I2C	Inter integrated communication
J	Inertie du moteur
L	Largeur utile de l'induit
n	Nombre de spire dans un encoche
N	Vitesse de rotation
P	Nombre de paires de pôle
PI	Proportionnel intégral
PIC	Unicontroleur du microship
PosC	Position courante
PosP	Position précédente
SDA	Ligne de transmission de données
SCL	Ligne d'horloge
SPI	Port de communication série synchrone
Φ	Flux magnétique

INTRODUCTION

Dès 1963, les robots industriels ont fait leur entrée dans l'industrie, aux C'est-à-dire, puis en Europe et au Japon. Aujourd'hui, ils ont pris une place importante dans les moyens de production des entreprises. L'industrie automobile est le plus grand utilisateur, par ses constructeurs, mais aussi par leurs fournisseurs d'équipements et d'accessoires. L'électroménager, les industries de la mécanique, l'électronique, l'aéronautique viennent ensuite. La pharmacie et le nucléaire s'y intéressent de près. Peu d'industries n'ont pas d'applications robotiques potentielles. Même celles travaillant en permanence y trouvent un intérêt pour des opérations de manutention lourde, de marquage ou de laboratoire. Cette évolution récente vers la presque totalité des industries est la conséquence des progrès réalisés par les robots au niveau de leurs performances et de leurs moyens de programmation avec des fonctions mieux adaptées aux applications.

Dans ce projet on s'intéresse à la partie électrique et électronique des robots, ainsi qu'au design de leurs systèmes de commande de leurs articulations en position et en vitesse.

L'architecture du système de commande est cruciale pour assurer la flexibilité et la fiabilité du contrôle d'un nombre de degrés de libertés extensibles.

1. Problématique

Supposons qu'un robot effectuant ses tâches dans un espace de travail bien déterminé est limité, en raison de l'évolution de la manière d'exécuter ces tâches, la contrainte d'augmenter le nombre de degrés de liberté du robot s'impose. En effet, au lieu de se limiter à N_1 articulations pour effectuer ces tâches, on augmentera le nombre d'articulations à N_1-x pour adapter au mieux et d'une manière flexible et fiable le nombre de degrés de liberté du robot aux nouvelles tâches imposées. Cette adaptation nous donnera le privilège d'élargir l'espace de travail du robot.

L'absence ou la rareté des pièces de rechange est un autre problème majeur rencontré fréquemment dans l'industrie. Pour y remédier, les responsables ont généralement tendance à changer radicalement l'installation et la remplacer par une autre plus récente. Prenons l'exemple d'un robot que l'on désire remplacer par un autre dont le design est différent, alors une problématique prend naissance; cette problématique, c'est la nécessité de tout changer, même sa partie de commande, ce qui alourdira le coût d'investissement et augmentera le temps d'arrêt MTD (mean time down).

La question qui se pose, est la suivante. Quelle architecture devrait avoir le système de commande pour qu'il s'adapte à n'importe quel type de robot quelque soit le nombre de ses degrés de libertés ?

2. Objectifs

L'objectif visé par ce projet est de concevoir, réaliser et implanter expérimentalement le prototype du système de commande et de contrôle d'un robot modulaire, quelque soit le nombre de degrés de liberté du robot.

La particularité fondamentale d'un tel système, c'est que son architecture possède une architecture décentralisée et autonome. En effet chaque axe est commandé par sa propre carte de contrôle, par conséquent l'ajout d'un axe supplémentaire engendrera seulement l'ajout de sa carte de commande qui sera liée à son tour à un maître qui gère l'ensemble des axes.

3. Méthodologie

La commande d'un axe est pilotée par une carte électronique dotée de deux contrôleurs esclaves, l'un assure l'acquisition de la position et l'autre s'occupe du calcul de la vitesse, de la génération de la tension de commande, de l'acquisition et de l'asservissement du couple et enfin, de la communication avec un processeur maître qui génère la trajectoire.

4. Architecture proposée

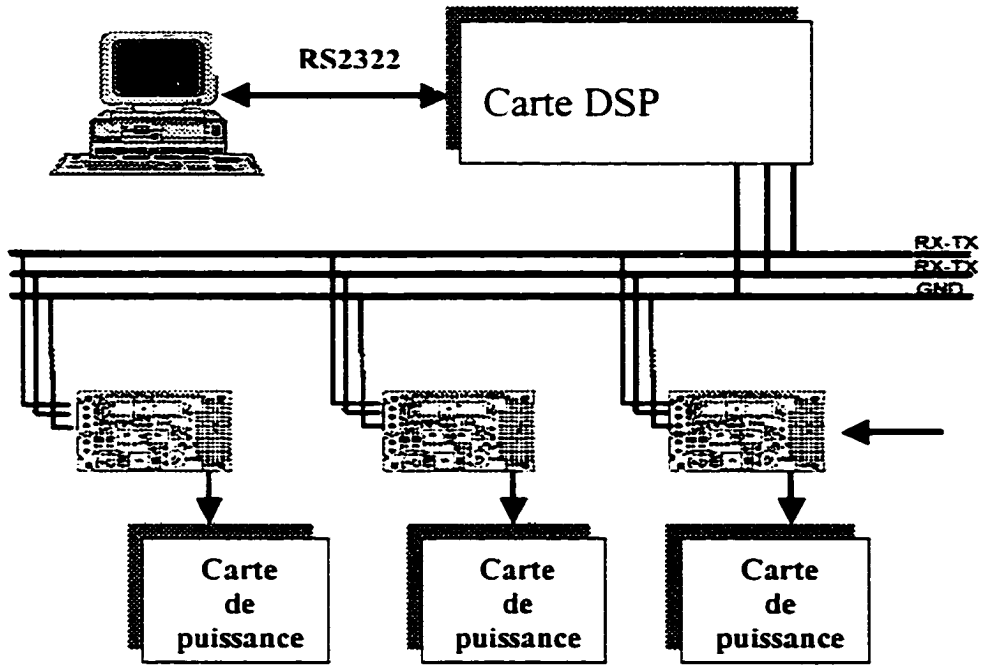


Figure 1 Architecture de la commande du robot.

Dans cette figure, on représente l'architecture qui sera choisie pour la commande du robot. Elle est constituée d'un DSP qui gère l'ensemble du réseau, génère la trajectoire sous forme de couples et communique avec le PC via le lien RS232 pour permettre l'affichage des positions, vitesses et couples des axes en temps réel. Le DSP communique aussi avec des cartes de commande dédiées à chaque articulation, celles-ci s'occupent de l'acquisition de la position, du calcul de la vitesse, de l'asservissement du couple et de la génération du signal PWM qui est amplifié par une carte de puissance pour alimenter les moteurs.

Pour mener ce projet à terme, il est primordial de passer par une étude approfondie sur les actionneurs utilisés (Moteur à courant continu), les différentes techniques d'asservissement de ces machines, ainsi qu'une revue des différents types de

communications séries nécessaires pour faire les choix appropriés à notre application.

Le premier chapitre présente les moteurs à courant continu et l'asservissement de leurs couples. Le deuxième chapitre présente les protocoles de communications, puis la modélisation du Robot utilisé pour les tests (MAKER 100) sera étudiée au troisième chapitre. Enfin, le dernier chapitre présente les différentes techniques utilisées, ainsi que les algorithmes intégrés dans la réalisation pratique. Les programmes développés au cours de ce projet, sont montrés en annexe.

CHAPITRE 1

ÉTUDE DES MOTEURS À COURANT CONTINU

La commande d'un robot est obtenue par le contrôle de ses articulations, qui sont en général actionnés par des moteurs à courant continu. Ces actionneurs assurent une conversion électromécanique réglable (position, vitesse ou couple) via la modulation de sources électriques.

Dans ce chapitre, on étudie les moteurs à courant continu, leurs constitutions, leurs principes de fonctionnement et l'asservissement du couple.

1.1 Introduction sur les moteurs à courant continu

Les moteurs à courant continu à excitation séparée sont encore largement utilisés pour l'entraînement à vitesse variable des machines. Très faciles à miniaturiser, ils s'imposent dans les applications à très faibles puissances. Ils se prêtent également fort bien à la variation de la vitesse, avec des technologies électroniques simples et peu onéreuses, pour des performances élevées et jusqu'à des puissances importantes (plusieurs mégawatts).

Leurs caractéristiques permettent également une régulation précise du couple, en moteur ou en générateur. Leur vitesse de rotation nominale est adaptable aisément par construction à toutes les applications, car elle n'est pas liée à la fréquence du réseau.

1.2 Constitution

Un moteur à courant continu est composé des éléments suivants :

Le collecteur ou stator : c'est un élément de circuit magnétique immobile sur lequel un enroulement est bobiné afin de produire un champ magnétique. L'électro-aimant ainsi réalisé comporte une cavité cylindrique entre ses pôles.

L'induit ou rotor : C'est un cylindre en tôles magnétique isolées entre elles et perpendiculaires à l'axe du cylindre. L'induit est mobile autour de son axe et est séparé de l'inducteur par un entrefer. À sa périphérie, des conducteurs sont régulièrement repartis.

Le collecteur et les balais : Le collecteur est solidaire de l'induit, les balais sont fixes. Les conducteurs de l'induit sont alimentés par l'intermédiaire de ce dispositif.

1.3 Principe de fonctionnement

Lorsque l'inducteur est alimenté, il crée un champ magnétique dans l'entrefer dirigé suivant le rayon de l'induit. Ce champ magnétique pénètre dans l'induit du côté pôle nord de l'inducteur et sort de l'induit du côté pôle sud de l'inducteur. Quand l'induit est alimenté, ses conducteurs situés sous un même pôle inducteur (d'un même côté des balais) sont parcourus par des courants de même sens et sont donc d'après la loi de Laplace, soumis à une force. Les conducteurs situés sous l'autre pôle sont soumis à une force de même intensité et de sens opposé. Les deux forces créent un couple qui fait tourner l'induit du moteur.

1.4 Expressions du couple et de la vitesse

- Si I_d est le courant fourni à l'induit et qu'il circule via 2*a voies en parallèle, l'intensité est alors de $\frac{I_d}{2a}$ dans les conducteurs. L'induit dans l'entrefer étant radial, la force tangentielle moyenne par conducteur est : $B_{moy} * L * \frac{I_d}{2a}$, B_{moy} désignant l'induction moyenne sous chaque pôle et L la longueur utile de l'induit.

S'il y a n conducteurs sur l'induit et que son diamètre est représenté par D, le couple électromagnétique est :

$$C = n * B_{moy} * \left(\frac{I_d}{2a}\right) * L * \left(\frac{D}{2}\right) \qquad (1.1)$$

Le flux utile Φ de chacun des 2p (p : paires) pôles inducteurs est le produit de B_{moy} par la surface de l'induit par le pôle :

$$\Phi = B_{moy} * (\frac{\pi * D * L}{2p}) \tag{1.2}$$

Donc

$$B_{moy} = 2p * \frac{\Phi}{\pi * D * L} \tag{1.3}$$

En reportant B_{moy} dans l'expression du couple, celle-ci devient :

$$C = \frac{1}{2 * \pi} * \frac{P}{a} * n * \Phi * I_d \tag{1.4}$$

Pour une machine donnée, le couple est proportionnel au flux inducteur Φ et au courant I_d absorbé par l'induit. Si l'on veut inverser le couple, il suffit d'inverser le flux Φ ou le courant I_d.

- Du couple, on peut passer à la puissance électromagnétique, qui est le produit du courant I_d par la force électromotrice E :

$$E * I_d = C * 2 * \pi * N = \left(\frac{p}{a}\right) * n * N * \Phi * I_d \tag{1.5}$$

$$E = \left(\frac{P}{a}\right) * n * N * \Phi$$

Or, la force électromotrice est égale à la tension U_d d'alimentation de l'induit diminuée de la chute de tension $R*I_d$ dans la résistance de celui-ci et de la chute de tension e_B aux contacts balais-collecteur.

$$E = \left(\frac{p}{a}\right) * n * N * \Phi = U_d - (R * I_d + e_B) \tag{1.6}$$

D'où l'expression de la vitesse :

$$N = \frac{(U_d - (R * I_d + e_B))}{((\frac{p}{a}) * n * \Phi)} \qquad (1.7)$$

La chute de tension $R * I_d + e_B$ étant négligeable devant U_d, on voit que la vitesse est proportionnelle à la tension d'alimentation et inversement proportionnelle au flux.

1.5 Alimentation sous tension variable

Les relations $C = k * \Phi * I_d$ et $N = \frac{(U_d - R * I_d)}{k * \Phi}$ montrent que pour faire varier la vitesse, on doit agir sur la tension U_d aux bornes de l'induit. Lorsque le moteur est alimenté sous une tension U_d constante, la vitesse N peut être augmentée en diminuant le flux Φ. Mais, l'inconvénient est une perte du couple au fur et à mesure qu'on diminue le flux.

En maintenant le flux Φ constant, on peut développer le même couple à toutes les vitesses et ainsi toujours utiliser au mieux les possibilités en courant de l'induit et de l'inducteur. Mais, varier la tension U_d est plus onéreux que varier le courant inducteur ; car on doit agir sur le circuit principal, celui de l'induit, et non plus sur un circuit auxiliaire, celui de l'inducteur. Pour notre Robot, nous avons des moteurs à aimants permanents, ce qui génère un flux constant.

1.5.1 Caractéristique de vitesse à vide

Le courant absorbé à vide par l'induit est faible, ce qui fait que la chute de tension $R * I_d + e_B$ est négligeable. Si le flux Φ est constant, alors la vitesse est proportionnelle à la tension (Figure 2). Cette zone à flux Φ constant est dite «à couple constant», car le même courant dans l'induit permet de développer le même couple à toutes les vitesses.

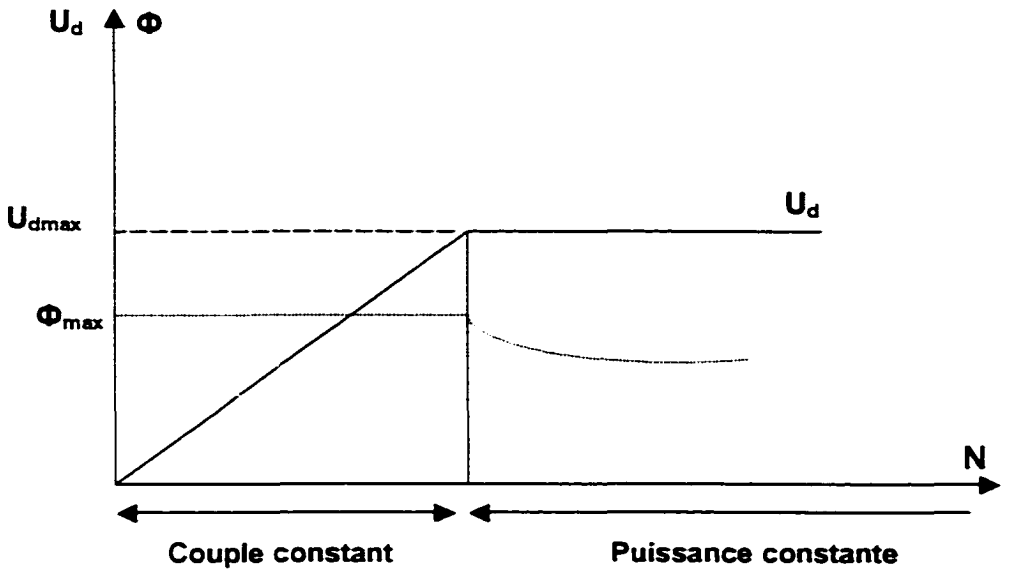


Figure 2 La vitesse en fonction de la tension.

Une fois atteinte la valeur maximale de la tension que peut donner la source alimentant l'induit, si on voulait augmenter encore N, il faudrait diminuer le flux inducteur. On travaillerait alors dans la zone dite « à puissance constante », car la puissance que pourrait développer le moteur à I_d donne, serait égale à $U_{dmax} \cdot I_d$ donc constante.

Le fonctionnement dans cette zone est rarement exploité car, quand la vitesse croit, le couple qu'on peut demander au moteur diminue.

1.5.2 Caractéristiques en charge à flux Φ constant

- Pour chaque tension U_d, quand le courant I_d absorbé par le moteur augmente, sa vitesse diminue un peu, puisque :

$$N = \frac{(U_d - (R * I_d + e_B))}{(\frac{p}{a} * n * \Phi)} \qquad (1.8)$$

En agissant sur U_d (courbes du quadrant 1 de la figure 2), on varie la vitesse tout en conservant la possibilité de développer. le couple nominal sans dépasser le courant nominal.

Le démarrage ne pose aucun problème puisqu'il s'opère par accroissement progressif de la tension U_d.

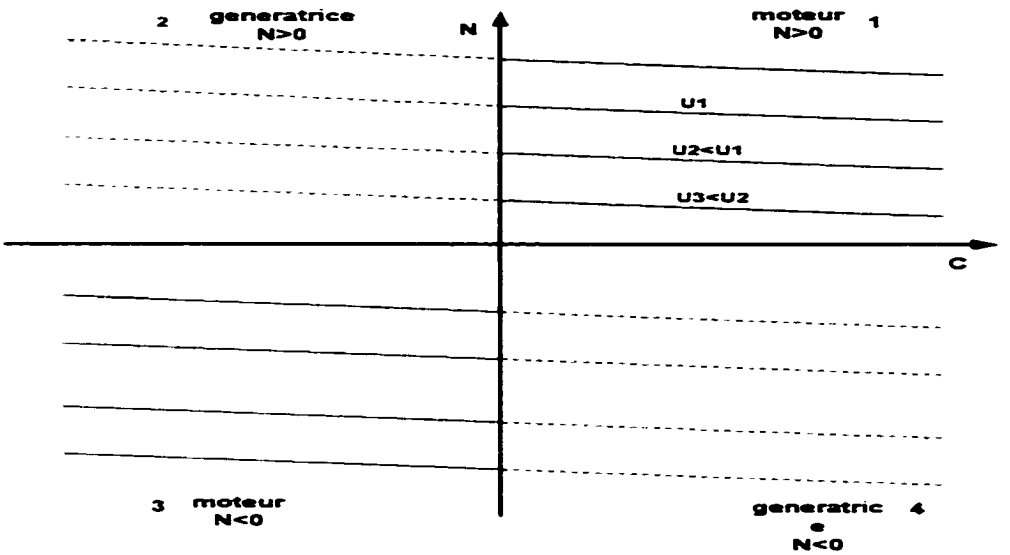


Figure 3 Caractéristiques en charge à Φ constant

- À une tension U_d donnée. si le couple sur l'arbre s'inverse. la machine fonctionne en génératrice. le courant I_d s'inverse : la machine travaille en freinage avec récupération. On peut modifier la vitesse de freinage par modification de la tension U_d (courbe du quadrant 2, figure 2).

  Sans excéder I_{dnom}, la machine peut développer un couple de freinage égal à C_{nom} quelque soit la vitesse.

- Si on inverse la tension U_d aux bornes de l'induit, on inverse le sens de rotation.

De même, comme pour le cas où U_d est positif, on peut fonctionner en moteur (quadrant 3 : $U_d<0$, $I_d<0$ donc $U_d*I_d > 0$) ou en génératrice (quadrant 4 : $U_d < 0$, $I_d > 0$ donc $U_d*I_d < 0$) et varier de la même façon la vitesse d'entraînement ou de freinage.

1.6 Étude de l'asservissement du couple de la machine à courant continu

Dans le cas de la machine à courant continu, réaliser un asservissement de couple revient à réaliser un asservissement de courant. Nous allons donc dans un premier temps étudier l'asservissement du courant dans la machine. Cet asservissement a pour schéma bloc :

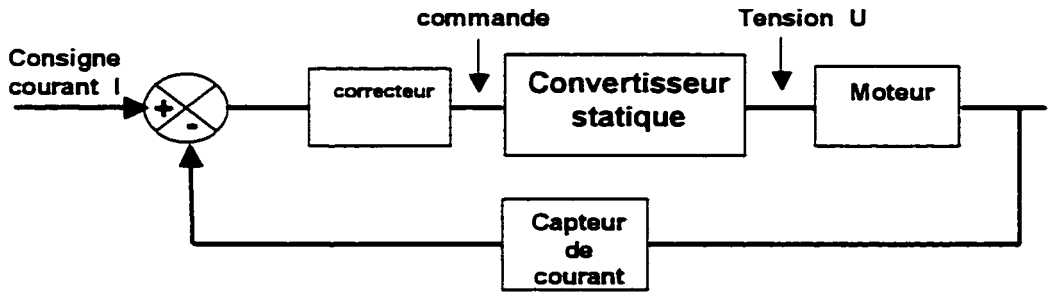


Figure 4 Schéma bloc de l'asservissement du courant dans la machine

Afin de pouvoir mener à bien la réalisation de cet asservissement, il est nécessaire de mettre en équation le comportement en courant de la motorisation, c'est-à-dire évaluer la fonction de transfert en boucle fermée :

$$C(s) = \frac{I(s)}{U(s)} \tag{1.9}$$

D'après les équations électromagnétiques, électriques et mécaniques :

Equations électromagnétiques : $C = k * \Phi * I$ (1.10)

$$E = k * \Phi * \Omega \tag{1.11}$$

Equation électrique : $U = E + R * I + f * \Omega$ (1.12)

Equation mécanique : $J * \frac{d\Omega}{dt} = C - f * \Omega$ (1.13)

Et en représentant ces équations dans le domaine de Laplace avec des conditions initiales nulles on obtient :

$$U(s) = E(s) + R * I(s) + L * S * I(s) \tag{1.14}$$

$$\Omega(s)[J * s + f] = C(s) \tag{1.15}$$

D'après ces équations on tire les expressions du courant, du couple et de la vitesse :

$$I(s) = (U(s) - E(s)) * \frac{1}{R + L * s} \tag{1.16}$$

$$C(s) = I(s) * k * \Phi \tag{1.17}$$

$$\Omega(s) = C(s) * \frac{1}{J * s + f} \tag{1.18}$$

On peut alors en déduire le schéma bloc suivant, représentatif de la fonction de transfert en courant du moteur :

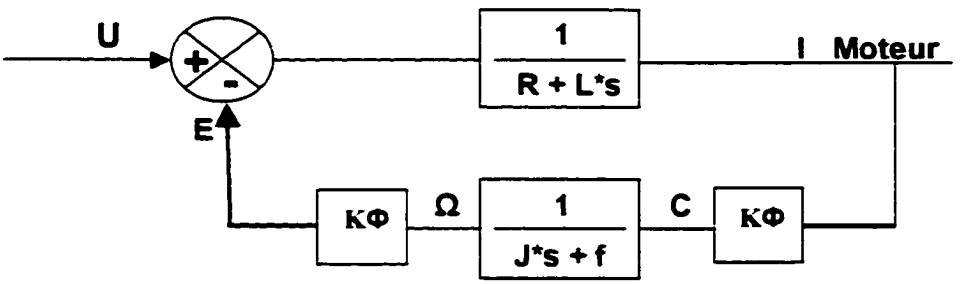


Figure 5 Schéma bloc de la fonction de transfert en courant du moteur.

Il est alors possible de tracer la réponse indicielle en courant de la machine, ce qui ne présente pas d'intérêt particulier au niveau de la mise en œuvre du réglage de la boucle de courant, mais qui permettra une interprétation de la réponse indicielle en boucle fermée :

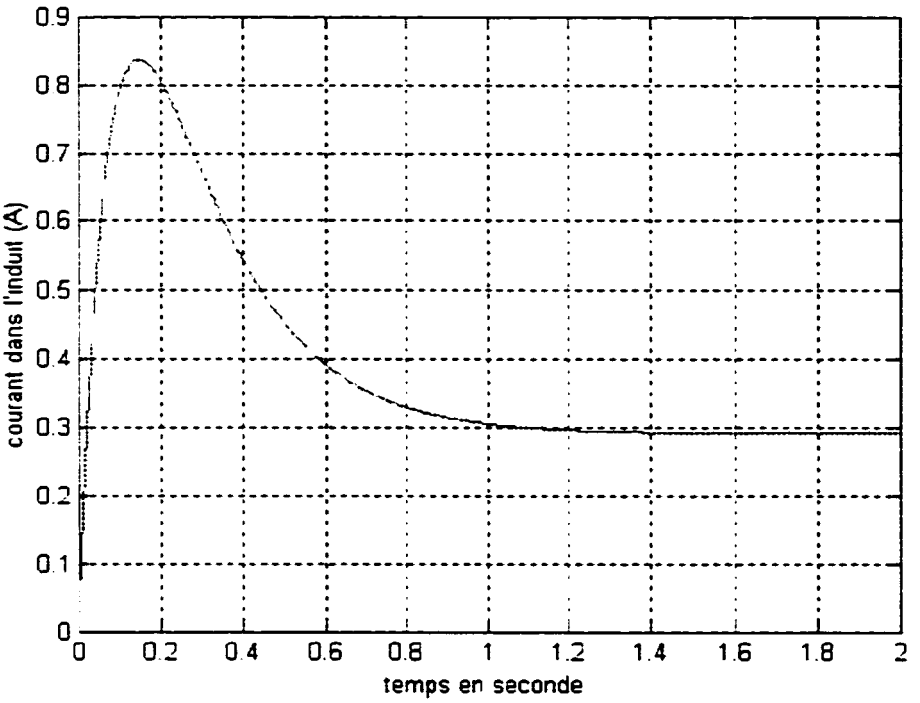


Figure 6 La réponse indicielle.

À partir du schéma bloc de la motorisation définie ultérieurement, il est possible de calculer la fonction de transfert $C(s) = \frac{I(s)}{U(s)}$, représentative du comportement en courant de la machine :

$$C(s) = \frac{\frac{1}{(R + L*s)}}{1 + \frac{k*\Phi^2}{(R + L*s)(J*s + f)}} = \frac{(J*s + f)}{(R + L*s)(J*s + f) + k*\Phi^2} \qquad (1.19)$$

Dans la figure suivante, la réponse fréquentielle de la machine est montrée.

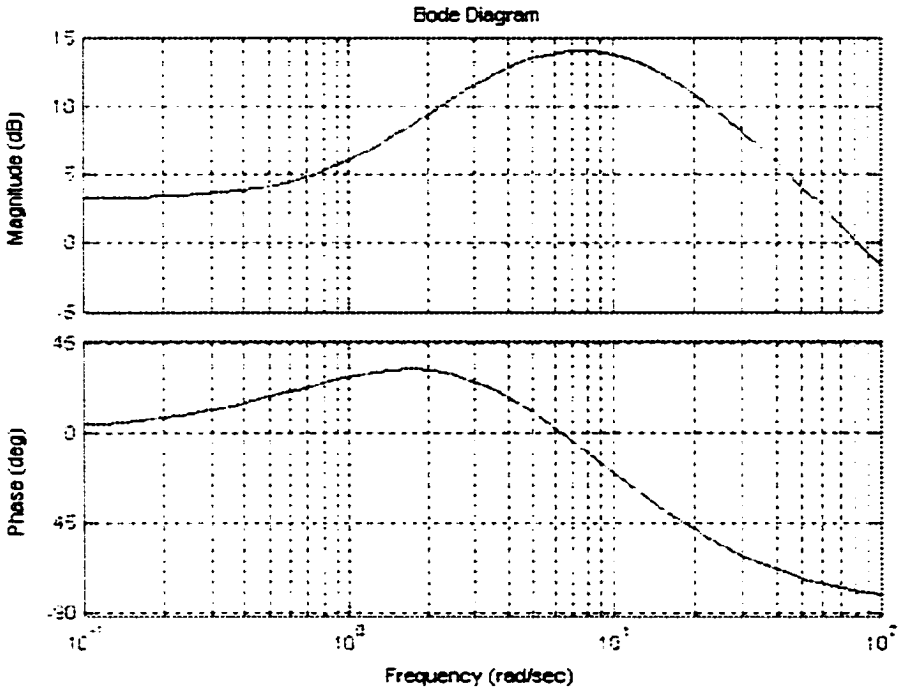


Figure 7 La réponse fréquentielle de la machine

1.7 Étude de la boucle de courant et de l'asservissement du couple

Le schéma bloc de la boucle de courant est représentée ci-dessous :

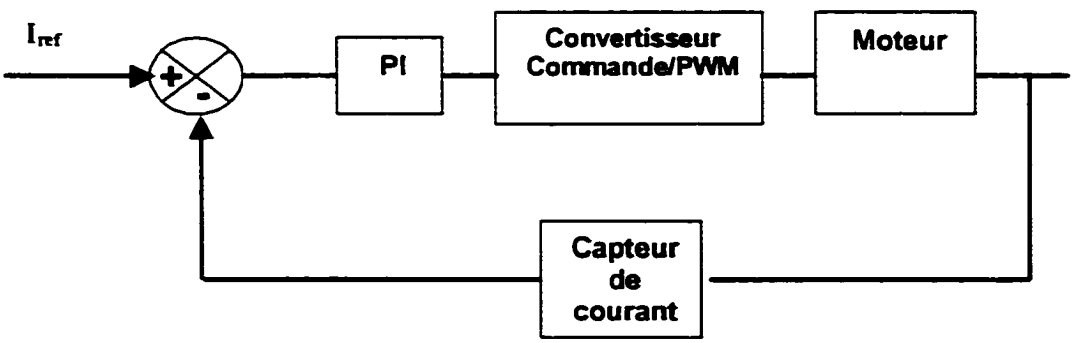


Figure 8 Le schéma bloc de la boucle de courant

Pour étudier cet asservissement, une interface de simulation avec Simulink fut réalisée. Cette interface est constituée du modèle de machine déduit des équations détaillées dans la première partie de ce chapitre, d'un générateur de signaux PWM à rapport cyclique variable, d'un régulateur PID et de dispositifs d'affichages, tels que les graphes et les afficheurs digitaux. La figure suivante présente cette interface.

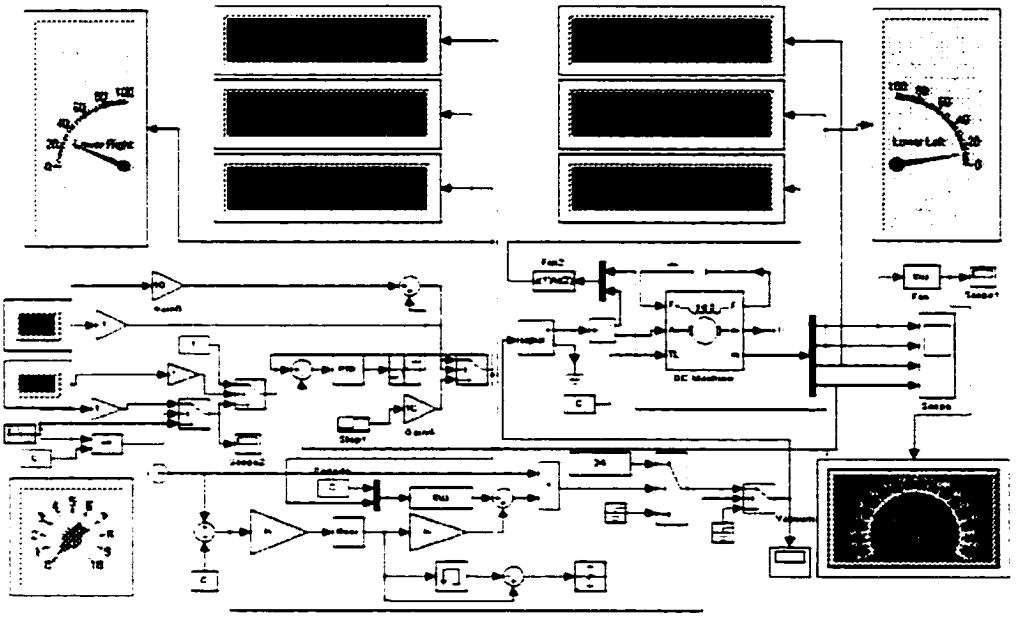


Figure 9 L'interface de l'asservissement du couple de la machine.

Dans cet asservissement du couple, on s'intéresse surtout à la rapidité de la réponse du système ainsi qu'à sa stabilité. Même, s'il y a une erreur au régime permanent, il existe une autre boucle de régulation, celle du générateur de trajectoire intégrée dans le DSP qui va compenser cette erreur en corrigeant la consigne du couple.

1.8 Les résultats de simulation

Pour effectuer cette simulation, on dispose de la fonction de transfert du couple avec la tension d'alimentation comme entrée. Cette représentation est constituée de différents paramètres du moteur, l'inertie, la constante magnétique, les coefficients de frottement. (on a pris le modèle de Matlab pour la simulation) :

$$G(s) = \frac{s+1}{0.012s^2 + 0.2s + 0.69}$$

La réponse indicielle de ce système est caractérisée par un très grand dépassement au démarrage, ce qui donne un très grand couple au départ ce qui distingue la machine à courant continu des autres telles que la machine asynchrone ou synchrone.

Ce couple élevé de démarrage pose un problème au niveau de l'utilisation du correcteur PI (proportionnel intégral), ce qui affaiblit la rapidité du système.

Dans la figure qui suit on donne la réponse du système avec un PI. D'une valeur de 400 pour le terme intégral (k_i) et 1000 pour le terme proportionnel (k_p), la consigne est de 3 N.m.

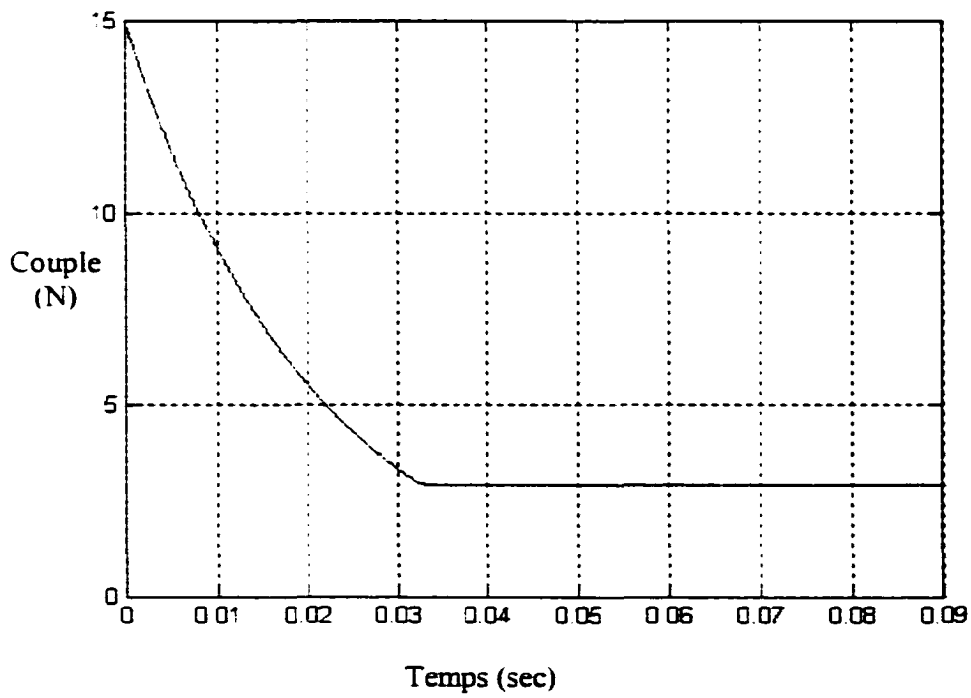


Figure 10 Asservissement du couple avec un proportionnel intégral.

D'après ce graphe on trouve que le système atteint son régime permanent avec une petite erreur après 0.032s.

Vérifiant maintenant la réponse du système avec un proportionnel pure d'une valeur de k_p égale à 1000.

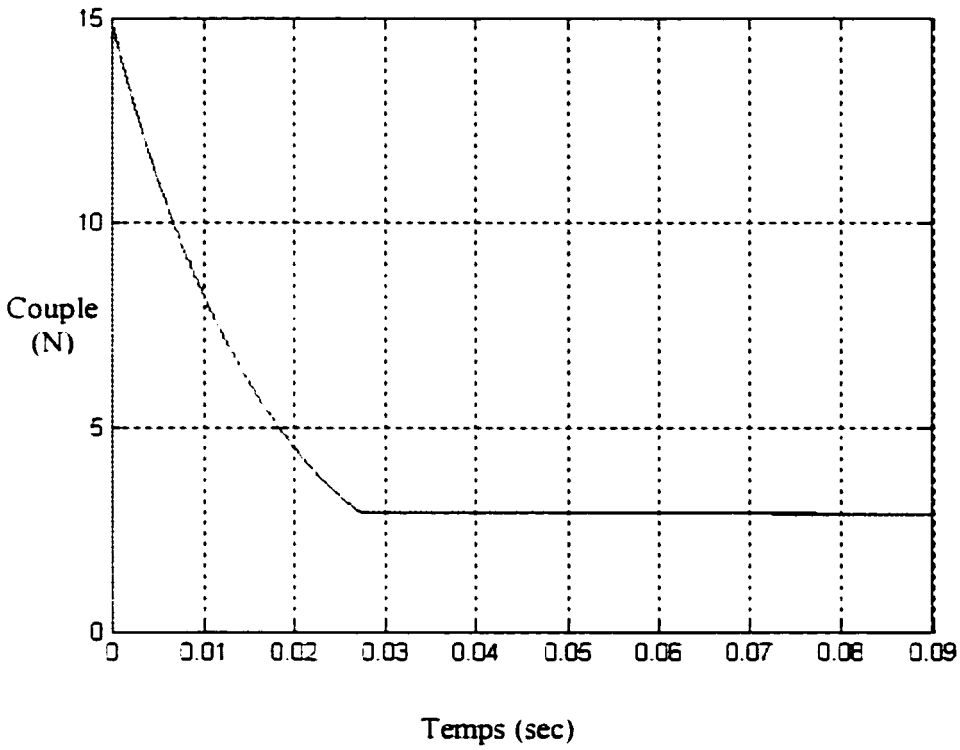


Figure 11 Asservissement du couple avec un proportionnel.

D'après ce graphe on trouve que le système atteint son régime permanent avec la même précision que le premier après 0.027s.

En analysant ces graphes, on opte pour un régulateur proportionnel car il donne une réponse satisfaisante, et de plus il est simple à intégrer dans un microcontrôleur,puis ne nécessite pas un très grand temps de traitement.

Dans ce chapitre, les machines à courant continu ainsi que l'asservissement du couple qui est envoyé par le DSP au microcontrôleur à travers une communication série ont été expliqués.

Dans le chapitre qui suit, les moyens de transfert de données et les différents protocoles de communication seront étudiés, car il faut choisir le plus approprié à notre application.

CHAPITRE 2

COMMUNICATION ET TRANSFERT DES DONNÉES

Dans ce chapitre, on étudiera en premier lieu la représentation de l'information dans des supports numériques, puisque notre application est basée essentiellement sur des systèmes numériques tel que le DSP et les microcontrôleurs. Ces derniers assument les traitements des signaux et données en provenance de l'extérieur, à savoir le courant induit de l'axe du robot, le couple, la position ..., qui sont à leur nature de base, des signaux analogiques. Pour les traiter, il faut effectuer un changement de forme, ça veut dire, il faut passer d'un niveau analogique à un autre logique ou numérique.

Après ces transformations, ces données seront transmises, soit par des liaisons séries ou parallèles selon l'importance de la rapidité et simplicité du transfert. Dans ce cadre on étudiera les différentes techniques de communications, afin de choisir et justifier les plus appropriés à notre application.

2. Représentation de l'information

2.1 Introduction

Pour être transmise, l'information doit d'abord être représentée sous une forme exploitable. Imaginons par exemple, qu'il s'agisse de transmettre une information concernant la vitesse de rotation ou de déplacement d'un axe. Il n'est pas possible, pratiquement, de transmettre une vitesse en tant que telle. Il y a fondamentalement deux manières de le faire :

- traduire l'information dans une autre grandeur physique plus facilement transmissible, tel un courant électrique au moyen d'un transducteur ou capteur : c'est la méthode analogique ;
- transmettre une information symbolisée, ici encore véhiculée par une grandeur physique appropriée, mais capable seulement de traduire l'information par un état ou

une combinaison d'états fondés sur le franchissement de valeurs, comme on le ferait avec un encodeur fonctionnant en tout ou rien. C'est la méthode logique ou numérique. Examinons plus en détail ces deux concepts.

2.2 La représentation analogique

L'information représente donc en général une grandeur physique au départ.
La grandeur physique peut être : la tension du signal, ou le courant, ou la durée de la partie haute ou basse de son cycle périodique. Des informations peuvent ainsi être véhiculées :
- par une onde modulée en amplitude : la grandeur physique utilisée est une tension ;
- par modulation de sa fréquence : on fait varier la période du signal donc un temps ;
- par des impulsions de tension ou de durée variable.

Prenons l'exemple de la lecture d'une vitesse prise par un tachymètre et visualisée par un voltmètre. La tension appliquée entraîne l'apparition d'un courant dans le galvanomètre, en vertu de la loi d'ohm. Ce courant entraîne l'apparition d'un couple qui lui est proportionnel, en vertu des principes électrodynamiques, et celui-ci, appliqué à un ressort de raideur déterminée entraîne un déplacement angulaire proportionnel de l'aiguille. L'aiguille du voltmètre évolue donc de manière analogue à la tension appliquée qui est proportionnelle à son tour à la vitesse de l'articulation.

L'avantage de ce type de traitement est qu'il met souvent en c'est-à-dire des dispositifs simples. C'est la raison pour laquelle il a d'abord été privilégié. Son inconvénient majeur est qu'à chaque niveau du traitement apparaît une erreur due aux incertitudes ou à l'instabilité des lois sur lesquelles le traitement est fondé (défaut de précision, défaut de linéarité, dérive des valeurs...). Et il n'est pas possible de s'affranchir totalement de ces défauts. On peut seulement en limiter la portée. On s'efforcera donc de « stabiliser » les montages c'est-à-dire de l'influence de la température, et de limiter les dérives des caractéristiques. On emploiera, par exemple, des amplificateurs différentiels dont la symétrie assure une certaine

compensation des dérives. Toutefois, quelque soit le soin apporté au dispositif, toute erreur introduite est irrémédiable.

2.2.1 La représentation en niveaux logiques

Dans ce cas, l'information est quantifiée. On fait correspondre de façon « discrète », un état du signal à un cas d'information. On passe d'un cas à un autre par un effet de seuil. D'une manière générale, l'information est traduite en binaire par deux états : VRAI ou FAUX, que l'on repère souvent par Haut ou Bas, 1 ou 0. Dans le cas de la mesure du couple d'un axe par un interrupteur celui-ci sera par exemple fermé si le couple de seuil n'a pas été atteint, ouvert dans le cas contraire. Le courant circulant dans le circuit de l'interrupteur traduit alors une condition logique (Vrai ou faux en réponse à la question de savoir si le couple a été atteint). L'information transmise ne traduit le couple que d'une façon sommaire, mais remarquons que si les deux cas peuvent être nettement distingués l'un de l'autre d'après la valeur du courant dans le circuit, une altération de la valeur de ce courant ne porte pas préjudice à la précision de l'information qui ne dépend que de la précision du seuil de déclenchement du détecteur.

2.2.2 La représentation numérique

Supposons que nous voulions perfectionner le dispositif précédent pour mieux connaître le couple. On serait alors amené à augmenter le nombre de niveaux détectés, et à coder le résultat par la combinaison de plusieurs variables logiques. Si le codage de ces variables se fait selon un principe numérique de pondération (binaire, octal, hexadécimal, décimal...) on parle alors de traitement numérique. Dans ce cas, la présentation du nombre obtenu est en principe conforme aux conventions en usage.

2.3 Système logique et système numérique.

Ces systèmes ne se distinguent pas fondamentalement l'un de l'autre en ce qui concerne la technologie. C'est essentiellement au niveau de la représentation des

variables que la différence est la plus évidente. Alors que dans un système logique chaque variable est représentée par un état booléen, elle est représentée dans un système numérique par un ensemble de chiffres binaires élémentaires ou bits. Cet ensemble de bits est groupé par paquet ou byte dont le format est très généralement l'octet qui est un ensemble de 8 bits. Dans de nombreux cas, une variable numérique repose sur plusieurs octets. Pour traiter ces variables, on est amené à réaliser des circuits réalisant des fonctions de calcul parfois très élaborées tels que les microprocesseurs, les microcontrôleurs ou DSP.

2.3.1 Les technologies destinées au traitement numérique de l'information

La technologie traite de la même manière les signaux numériques et logiques. Toutefois, la construction de dispositifs de calcul a fait apparaître des circuits plus particulièrement destinés au traitement numérique de l'information.

2.3.2 Circuits logiques de calcul

Ce sont les circuits tels les additionneurs, les comparateurs numériques, les compteurs. Beaucoup de ces fonctions sont actuellement intégrés dans des circuits programmables.

2.4 Circuits d'architecture de transport d'information

Ces circuits viennent en périphérie de systèmes de calcul systèmes micro-informatiques en particulier pour satisfaire aux nécessités du transport de données. L'information numérique étant fréquemment véhiculée par des mots de huit bits (octets), de nombreux composants de traitement numérique réunissent dans un même circuit intégré des éléments groupés par huit.

2.4.1 Les amplificateurs 3 états

Ils permettent de brancher ou de débrancher une source de données à un câble ou une nappe de conducteurs, que l'on appelle "Bus", selon une entrée de commande

capable de placer toutes les sorties à un état " haute impédance " pour les débrancher du bus. Les sorties sont à même de délivrer une puissance suffisante pour que le signal ne se dégrade pas. même sur des conduites de longueur conséquente.

2.4.2 Les verrous ou "latches"

Ces circuits intègrent généralement un amplificateur 3 états. mais associé à un registre qui permet de garder l'information en mémoire. La mise en mémoire peut être commandée par une entrée spéciale. La commande peut se faire par le niveau appliqué (entrée Enable). ou par son changement d'état (front montant par exemple). On parle alors d'entrée d'horloge (« clock ») dynamique.

2.4.3 Les circuits de multiplexage ou de démultiplexage

Un multiplexeur possède une logique capable d'aiguiller vers la sortie une donnée choisie par un mot de sélection. Le démultiplexeur fait le travail inverse : il permet d'aiguiller une donnée présente à l'entrée. vers l'une des sorties choisie par un mot de sélection.

L'association d'un multiplexeur et d'un démultiplexeur permet d'utiliser le même bus pour plusieurs canaux de données.

2.5 La conversion d'informations analogiques en informations logiques et réciproquement

Il est de plus en plus fréquent d'avoir à convertir des informations d'une représentation analogique à une représentation numérique ou vice versa. On a recours pour cela à des convertisseurs.

2.5.1 Les Convertisseurs Numériques Analogiques

Dans leur forme la plus simple. ils sont constitués d'un c'est-à-dire à entrées pondérées.

2.5.2 Les Convertisseurs Analogiques Numériques

La conversion analogique numérique, ce qui nous intéresse de plus dans ce projet, peut se faire par une conversion faisant intervenir des amplificateurs opérationnels en mode générateur de rampe (montage intégrateur), comparateurs et des compteurs avec des horloges dont la fréquence et réglable pour augmenter ou réduire la précision.

Une fois la grandeur physique est convertie, codée si nécessaire, on cherche des moyennes permettant de véhiculer de toute fidélité cette information d'une station à une autre, que ce soit du maître vers l'esclave ou l'inverse. Pour réaliser cette tache il existe deux façons numériques, soit en utilisant une transmission parallèle qui reste le moyen le plus robuste à cause de son faible taux d'erreur et sa rapidité, mais le seul handicap est le nombre de fils utilisés puisque il faut associer un fil à chaque bit dans l'octet à transmettre, de plus au niveau des circuits (microcontrôleurs) il faut réserver huit broches (port d'entrées/sorties) pour la communication, se qui limite la capacité du composant pour gérer d'autres tâches.

La deuxième possibilité qui nous reste, est le transfert série qui est un mode de transmission permettant d'économiser le nombre des fils en utilisant le même fil pour transmettre les bits séquentiellement les uns derrière les autres. Il s'agit donc en réalité d'un procédé de multiplexage.

2.6 Les protocoles de communication série

La communication repose essentiellement sur la représentation et l'usage d'un ensemble de symboles sur lequel plusieurs partenaires s'accordent. La première nécessité est de définir le contexte dans lequel l'échange peut avoir lieu. La plupart des échanges qui concernent les informations « système » se font sous forme parallèle, mais on a également des standards de transmission série qui s'appliquent à des composants tels que des mémoires, des circuits d'entrée-sortie, des microcontrôleurs etc... comme les standards RS232, RS485, I2C et SPI . Ces

dispositifs ne sont pas utilisés dans les micros ordinateurs. On leur préfère les bus système fonctionnant en parallèle car ils sont plus rapides. Les transmissions système de type série permettent par contre de relier un certain nombre de circuits de façon simple.

2.6.1 Les liaisons RS232 et RS485

Le protocole RS232 est destiné aux liaisons entre deux réseaux point à point. Ce n'est donc pas à proprement parler un « bus » au sens où ce terme sous-entend généralement une information distribuée à un certain nombre de partenaires. De plus, les informations sont transmises en référence commune, ce qui en limite la portée et la sécurité. On peut toutefois remédier à ces limitations en l'interfaçant avec un circuit approprié pour satisfaire aux spécifications RS485, et permettre ainsi des échanges en mode différentiel avec des points d'accès multiples. Les deux liaisons fonctionnent en mode de transmission asynchrone.

Le principe de base régissant ce protocole de transmission en série est que les données et les caractères de contrôle doivent être envoyés sur une ligne unique à raison de 1 bit à la fois.

Supposons que les bits de données circulent comme s'ils voyageaient sur un fil unique. La longueur de chaque bit est déterminée par la vitesse de transmission qui est mesurée en bits par seconde. Cette vitesse est appelée débit en bauds. Ainsi, si les données sont transmises à raison de 9600 bits par seconde, on dira que le débit de transmission est de 9600 bauds. Quand aucune donnée n'est transmise sur la ligne, la ligne est en état logique 1 ou état haut. Quand nous désirons transmettre des caractères, le premier bit envoyé s'appelle le bit de départ. Ce bit est représenté par l'état 0 sur la ligne. La durée du bit de départ est déterminée par le débit. Donc le récepteur sait qu'il va recevoir un caractère quand il détecte que la ligne passe de l'état haut à l'état bas.

Les bits de données, qui composent l'information qui est transmise, suivent immédiatement le bit de départ. Le nombre de bits de données peut varier de 5 à 8. Cependant, chaque caractère d'une même transmission doit être composé du même nombre de bits. Le nombre de bits de donnée n'est pas fixe car il arrive très souvent que le nombre de bits qui forment les caractères que nous désirons envoyer est inférieur à 8. Nous pouvons ainsi améliorer la vitesse de transmission en ne fixant pas la longueur des données transmises. Les bits de données sont transmis en commençant par le bit de poids faible et sont rassemblés à la réception pour reformer le caractère transmis.

Un bit facultatif de parité suit immédiatement les bits de données. Le type de parité choisi doit être uniforme au cours d'une même transmission. Si la parité paire est choisie, le nombre de bits 1 qui compose les données et le bit de parité doit être pair. Par contre, dans le cas de la parité impaire le nombre de bits 1 doit être impaire. Ce bit de parité nous permet de détecter certains types d'erreurs de transmission. À la fin de la trame qu'on transmet, on met un bit indiquant la fin de transmission, ce bit est appelé bit d'arrêt, ce dernier représente la durée minimale pendant laquelle la ligne doit être à l'état haut avant l'envoi d'un autre caractère.

2.6.2 La liaison I 2C

Le bus I2C *ou Inter-Integreted-Communication* a été conçu pour réaliser la liaison entre les circuits intégrés d'une même platine. Il se charge de la communication entre les périphériques qui est habituellement assurée par un bus parallèle. En développant ce bus, Philips a équipé la plupart de ses appareils électroniques destinés au grand public (appareils TV et radio, systèmes audio et radio, postes téléphoniques, systèmes électriques automobiles, appareils électroménager...). Les informations sont échangées au moyen de deux lignes bidirectionnelles. ***SDA*** qui est la ligne de transmission de données et ***SCL*** qui est une ligne de synchronisation ou d'horloge. Chaque carte dans le réseau possède une adresse unique qui la distingue des autres. Chaque partenaire peut émettre ou recevoir des informations suivant sa fonction. Le

bus est piloté par un circuit appelé **maître** qui prend l'initiative du transfert des informations, décide le sens de ce transfert et gère la ligne *SCL*. Les autres composants sont alors désignés par le terme **esclave**.

Un bus peut prendre une configuration multi-maître si plusieurs composants peuvent être à tour de rôle maître ou esclave. A tout moment, un seul maître actif doit être présent sur le bus. Une procédure d'arbitrage est prévue pour éviter la perte ou la détérioration des informations lorsque plusieurs maîtres essaient de prendre la commande du bus simultanément. Mais sur la plupart des circuits, un seul composant réalise la fonction de maître étant donné la complexité de cette fonction.

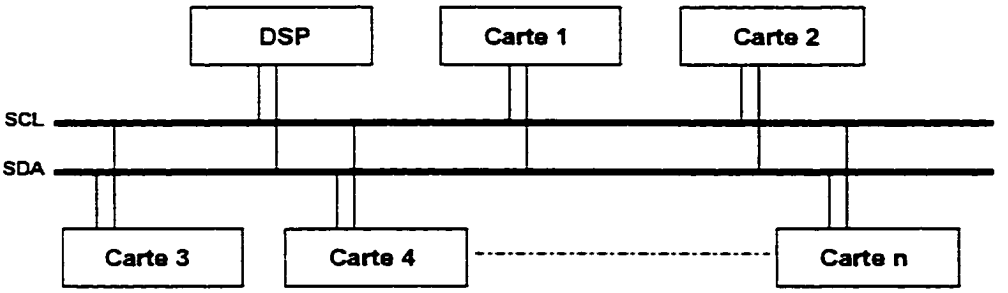


Figure 12 Connexions d'unités au bus I2C.

2.6.2.1 Le protocole I2C

Le protocole *I2C* définit la succession des états logiques possibles sur *SDA* et *SCL*, et la façon dont doivent réagir les circuits en cas de conflits.

2.6.2.2 La prise contrôle du bus

Pour prendre le contrôle du bus, il faut que celui-ci soit au repos (*SDA* et *SCL* à '1'). Pour transmettre des données sur le bus, il faut donc surveiller deux conditions particulières :

- La condition de départ. (*SDA* passe au niveau bas (0) alors que *SCL* reste au niveau haut (1).

- La condition d'arrêt. (*SDA* passe au niveau haut alors que *SCL* reste au niveau haut.

Lorsqu'un circuit, après avoir vérifié que le bus est libre, prend le contrôle de celui-ci, il en devient le **maître**. C'est lui qui génère le signal d'horloge.

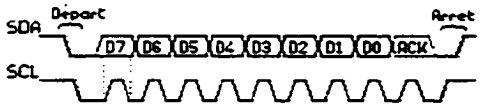


Figure 13 Exemple de condition de départ et d'arrêt.

2.6.2.3 La transmission d'un octet

Après avoir imposé la condition de départ, le maître applique sur *SDA* le bit de poids fort *D7*. Il valide ensuite la donnée en appliquant pendant un instant un niveau '1' sur la ligne *SCL*. Lorsque *SCL* revient à '0', il recommence l'opération jusqu'à ce que l'octet complet soit transmis. Il envoie alors un bit *ACK* à '1' tout en scrutant l'état réel de *SDA*. L'esclave doit alors imposer un niveau '0' pour signaler au maître que la transmission s'est effectuée correctement. Les sorties de chacun étant à collecteurs ouverts, le maître voit le '0' et peut alors passer à la suite.

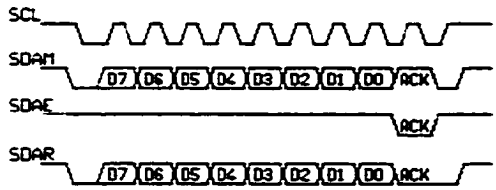


Figure 14 Exemple de transmission réussie.

Dans cet exemple :

- SCL : Horloge imposée par le maître.
- SDAM : Niveaux de SDA imposés par le maître.

- SDAE : Niveaux de SDA imposés par l'esclave.
- SDAR : Niveaux de SDA réels résultants.

2.6.2.4 La transmission d'une adresse

Le nombre de composants qu'il est possible de connecter sur un bus I2C étant largement supérieur à deux, il est nécessaire de définir pour chacun une adresse unique. L'adresse d'un circuit, codée sur sept bits, est définie d'une part par son type et d'autre part par l'état appliqué à un certain nombre de ses broches. Cette adresse est transmise sous la forme d'un octet au format particulier.

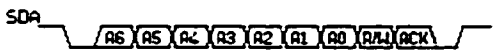


Figure 15 Exemple d'octet d'adresse.

On remarque ici que les bits *D7* à *D1* représentent les adresses *A6* à *A0*, et que le bit *D0* est remplacé par le bit de *R/W* qui permet au maître de signaler s'il veut lire ou écrire une donnée. Le bit d'acquittement *ACK* fonctionne comme pour une donnée, ceci permet au maître de vérifier si l'esclave est disponible.

2.6.2.5 Écriture d'une donnée

L'écriture d'une donnée par le maître ne pose pas de problème particulier :

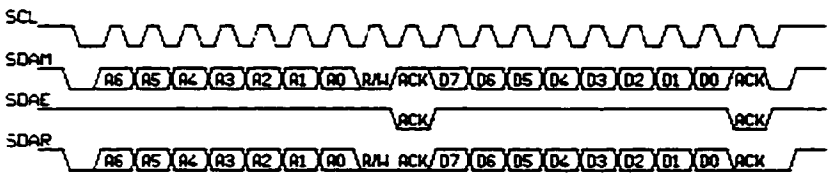


Figure 16 Exemple d'écriture d'une donnée.

2.5.2.6 Lecture d'une donnée

La lecture d'une donnée par le maître se caractérise par l'utilisation spéciale qui est faite du bit *ACK*. Après la lecture d'un octet, le maître positionne *ACK* à '0' s'il veut lire la donnée suivante ou à '1' le cas échéant. Il envoie alors la condition d'arrêt.

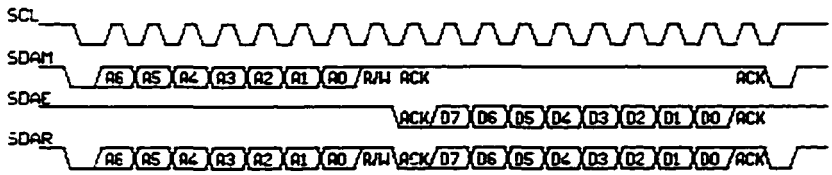


Figure 17 Exemple de lecture d'une donnée

2.6.3 La gestion des conflits

2.6.3.1 Mise en situation

La structure même du bus I2C a été conçue pour pouvoir y accueillir plusieurs maîtres. Se pose alors le problème commun à tout les réseaux utilisant un canal de communication unique : la prise de parole. En effet, chaque maître pouvant prendre possession du bus dès que celui-ci est libre, il existe la possibilité que deux maîtres prennent la parole en même temps. Si cela ne pose pas de problème sur le plan électrique grâce à l'utilisation de collecteurs ouverts, il faut pouvoir détecter cet état de fait pour éviter la corruption des données transmises.

2.6.3.2 Principe

Comme nous avons cité précédemment, pour prendre le contrôle du bus, un maître potentiel doit d'abord vérifier que celui-ci soit libre, et qu'une condition d'arrêt a bien été envoyée depuis au moins $4.7\mu s$. Mais il reste la possibilité que plusieurs maîtres prennent le contrôle du bus simultanément.

Chaque circuit vérifie en permanence l'état des lignes *SDA* et *SCL*, y compris lorsqu'ils sont eux même en train d'envoyer des données. On distingue alors plusieurs cas :

- Les différents maîtres envoient les mêmes données au même moment :

Les données ne sont pas corrompues, la transmission s'effectue normalement, comme si un seul maître avait parlé. Ce cas est rare.

- Un maître impose un '0' sur le bus :

Il relira forcément '0' et continuera à transmettre. Il ne peut pas alors détecter un éventuel conflit.

- Un maître cherche à appliquer un '1' sur le bus :

S'il ne relit pas un niveau '1', c'est qu'un autre maître a pris la parole en même temps. Le premier perd alors immédiatement le contrôle du bus, pour ne pas perturber la transmission du second. Il continue néanmoins à lire les données au cas celles-ci lui auraient été destinées.

En se basant sur cette étude, on a développé un programme en langage C qui permet la communication entre le PC et les cartes qu'on va décrire par la suite pour la commande des robots, en cas où on désire utiliser le PC pour la génération de trajectoire. Dans l'annexe vous trouvez ce programme de communication I2C.

2.7 La communication SPI

Le protocole de communication SPI est identique à celui du I2C, sauf qu'on utilise des lignes distinctes, une pour l'émission de données et l'autre pour la réception, ce qui donne la possibilité d'atteindre des vitesses très élevées d'échange qui peuvent dépasser les 5Mhz contre 400 KHz pour le I2C, ainsi que la sélection matérielle de l'esclave (SS), permet de réduire la taille de la trame émise par le maître ce qui réduit énormément le temps de scrutation des différents esclaves sur la ligne.

La figure suivante donne le schéma de câblage d'un bus SPI.

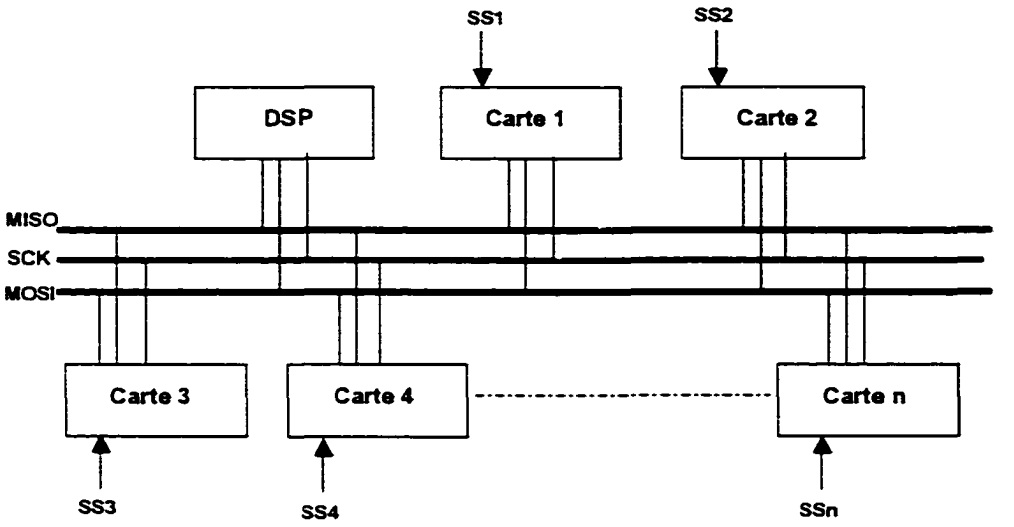
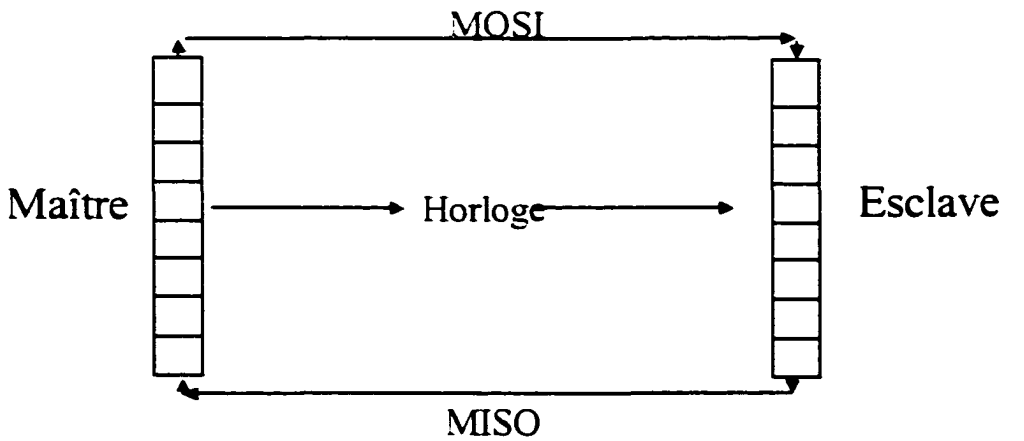


Figure 18 Schéma du câblage d'un bus SPI.

On désigne par MOSI (Master out slave in), ligne d'émission maître et réception esclave, par MISO (Mater in slave out), ligne de réception maître et émission esclave et par SCK l'horloge de synchronisation.

Le schéma suivant décrit le principe de transfert de données sur le bus :



Prenons l'exemple de deux octets à échanger entre le maître et l'esclave, le maître commence par l'appel de l'esclave par la mise à niveau bas de son entrée SS. Après cette activation, les deux stations placent les octets à émettre dans les registres de communications, puis après, le maître commence le contrôle du transfert par la ligne

d'horloge. L'échange de données est circulaire. ça veut dire qu'un bit émis est remplacé par un autre reçu.

D'après l'étude des différents protocoles de communications. on constate que le transfert via SPI est le plus approprié. à cause de sa rapidité par rapport aux autres. Ceci est un critère essentiel dans le domaine de la robotique.

Dans ce chapitre on a étudié les différentes liaisons et protocoles de communication séries. Dans le chapitre qui suit on modélise le robot sur lequel on va effectuer les tests.

CHAPITRE 3

MODÉLISATION DU ROBOT US MAKER

Dans ce chapitre, on donne l'étude du robot sur lequel on effectuera nos tests, en commençant par l'étude de la cinématique directe et inverse, après on passe à la modélisation dynamique du robot.

3.1 Cinématique directe et inverse

3.1.1 Systèmes d'axes

La figure présentée ci-dessous montre la configuration du robot Maker 100 de « United States Robots » qui possède cinq degré de liberté. Il s'agit d'axes rotatifs pour les axes 1, 2, 4 et 5 et d'un axe prismatique pour l'axe 3. A signaler que les deux derniers contrôlent l'angle de tangage (pitch) et l'angle de roulis (roll) respectivement.

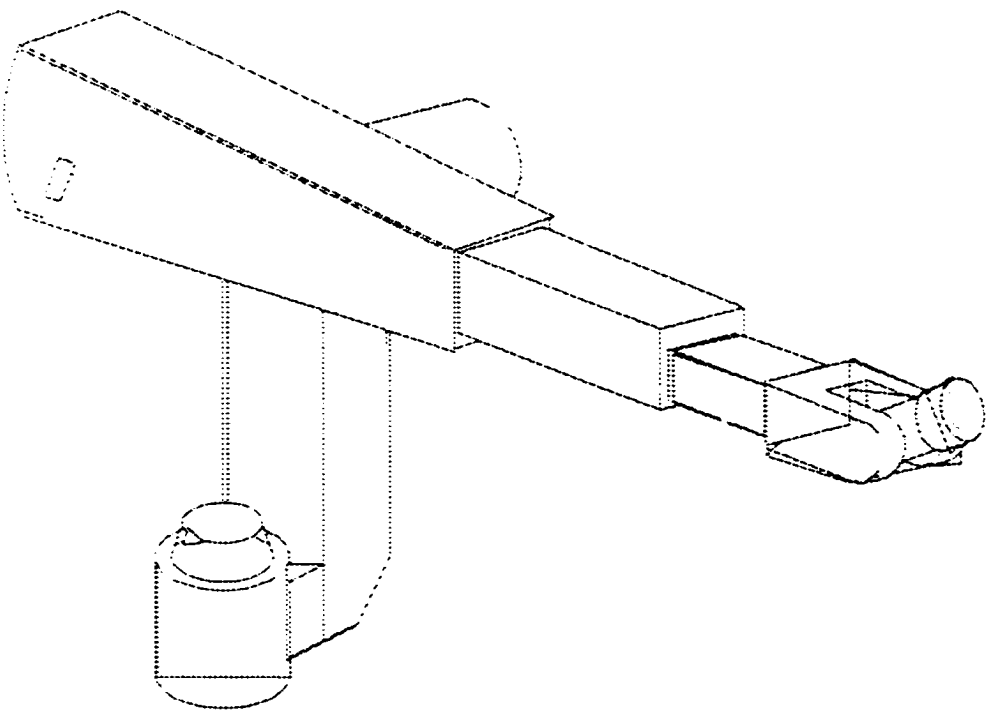


Figure 19 Le robot US MAKER 100 à cinq degrés de liberté.

Lors de l'établissement des systèmes d'axes on a suivi les conventions de Craig pour déterminer les paramètres de Denavit-Hartenberg modifiés (DH) du robot comme suit :

- On choisit un système de coordonnée fixe qui sera considéré comme référence pour tous les mouvements.
- On fixe les axes OZ des rotations (respectivement de translation) pour les joints angulaires (respectivement prismatique). Le sens positif peut être décidé d'une façon arbitraire. (voir figure 20).
- Puisque la dernière articulation est rotative, on choisit son origine (O_5) de telle sorte que d_5 soit égale à 0, donc confondu avec l'origine O_4.

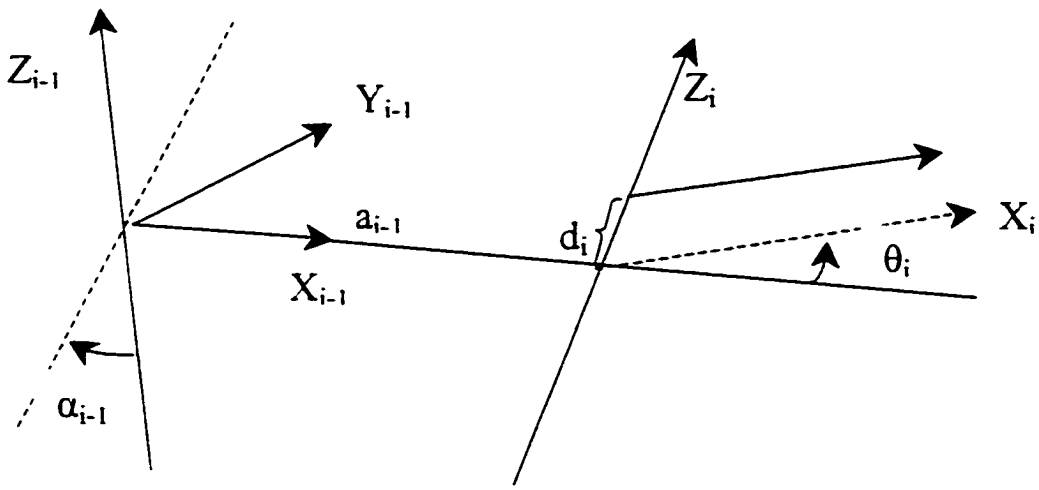


Figure 20 Convention de système d'axes.

On définit ces paramètres par :

- a_{i-1} : la distance perpendiculaire commune entre les axes i-1 et i (attachée au membre i-1).
- α_{i-1} : l'angle de torsion (fixe) : angle nécessaire pour ramener l'axe i parallèle à l'axe i-1.
- θ_i : l'angle de la rotation autour de Zi de Xi-1 vers Xi.

➢ d_i : translation selon Zi des points de rencontre avec Xi-1 et Xi.

3.1.2 Paramètres DH du robot

Le tableau I montre les différents paramètres attachés au robot MAKER 100.

Tableau I

Paramètres des différentes articulations.

joint	variable	α_i-1	A_i-1	d_i	θ_i
1	θ 1	0	0	0	θ 1
2	θ 2	90	0	0	θ 2
3	D 3	90	0	- d3	0
4	θ 4	- 90	0	0	θ 4
5	θ 5	-90	0	0	θ 5

3.1.3 Espace de travail du robot

Le tableau II résume les différentes limites de déplacement relatives à chaque articulation du robot US MAKER.

Tableau II

L'espace de travail du robot

Articulation	Type	Marge de manœuvre
# 1	Rotative	entre 0° et +350°
# 2	Rotative	entre -141° et +141°
# 3	Prismatique	entre 400 mm et 910 mm
# 4	Rotative	entre -114° et + 114°
# 5	Rotative	entre 0° et +348°

3.1.4 Matrice Globale de transformation homogène

À l'aide des outils de calcul symbolique, la matrice globale de transformation homogène du robot, calculée en détail dans l'annexe 1, est donnée par :

$$ {}^{0}_{5}T = \begin{bmatrix} C_1C_5C_{24} - S_1S_5 & -C_1S_5C_{24} - S_1C_5 & -C_1S_{24} & -C_1S_2d_3 \\ S_1C_5C_{24} + C_1S_5 & -S_1S_5C_{24} + C_1C_5 & -S_1S_{24} & -S_1S_2d_3 \\ C_5S_{24} & -S_5S_{24} & C_{24} & C_2d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (3.1) $$

D'autre part, une comparaison entre la matrice ${}^{0}_{5}T$ et celle des angles d'Euler représentée par :

$$Rot(Z,\varphi)\,Rot(Y,\phi)\,Rot(X,\gamma) = \begin{bmatrix} c\varphi\,c\phi\;\; c\gamma - s\varphi\;\; s\gamma & -c\varphi\,c\phi\;\; s\gamma - s\varphi\;\; s\gamma & -c\varphi\,s\phi \\ c\varphi\,c\phi\;\; c\gamma - s\varphi\;\; s\gamma & -s\varphi\,c\phi\;\; s\gamma + c\varphi\;\; c\gamma & -s\varphi\,s\phi \\ s\phi\;\; c\gamma & -s\phi\;\; s\gamma & c\phi \end{bmatrix} \tag{3.2}$$

Et en identifiant les éléments des deux matrices, on peut déduire les trois angles d'Euler :

$$\varphi = \theta_1; \qquad \phi = \theta_2 + \theta_4; \qquad \gamma = \theta_5 \tag{3.3}$$

3.1.5 Matrice Jacobienne

Pour déterminer la matrice jacobienne, on doit exprimer la vitesse angulaire et linéaire du poignet qui sont données par les équations suivantes (voir annexe 1 pour plus de détails du calcul de ces vitesses) :

$${}^{1}w_5 = \begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix} = \begin{bmatrix} s_1\dot{\theta}_2 - c_1 s_{24}\dot{\theta}_5 + s_1\dot{\theta}_4 \\ -c_1\dot{\theta}_2 - s_1 s_{24}\dot{\theta}_5 - c_1\dot{\theta}_4 \\ c_{24}\dot{\theta}_5 + \dot{\theta}_1 \end{bmatrix} \tag{3.4}$$

$${}^{1}v_5 = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = \begin{bmatrix} s_1 s_2 . d_3 . \dot{\theta}_1 - c_1 c_2 . d_3 . \dot{\theta}_2 - c_1 s_2 . \dot{d}_3 \\ -c_1 s_2 . d_3 . \dot{\theta}_1 - s_1 c_2 . d_3 . \dot{\theta}_2 - s_1 s_2 . \dot{d}_3 \\ -s_2 . d_3 . \dot{\theta}_2 + c_2 . \dot{d}_3 \end{bmatrix} \tag{3.5}$$

Par définition, la matrice Jacobienne lie la vitesse cartésienne et la vitesse articulaire comme suit :

$$\begin{bmatrix} V_x \\ V_y \\ V_z \\ W_x \\ W_y \\ W_z \end{bmatrix} = J \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{d}_3 \\ \dot{\theta}_4 \\ \dot{\theta}_5 \end{bmatrix} \tag{3.6}$$

Avec :

$$J = \begin{bmatrix} s_1 s_2 d_3 & -c_1 c_2 d_3 & -c_1 s_2 & 0 & 0 \\ -c_1 s_2 d_3 & -s_1 c_2 d_3 & -s_1 s_2 & 0 & 0 \\ 0 & -s_2 d_3 & c_2 & 0 & 0 \\ 0 & s_1 & 0 & s_1 & -c_1 s_{24} \\ 0 & -c_1 & 0 & -c_1 & -s_1 s_{24} \\ 1 & 0 & 0 & 0 & c_{24} \end{bmatrix} \tag{3.7}$$

3.2 Modélisation dynamique du robot par la méthode de Lagrange

On a formulé le modèle dynamique du robot, en appliquant la méthode de Lagrange choisie arbitrairement, tout en supposant que la structure mécanique est rigide. Cette méthode consiste à établir un bilan énergétique du système mécanique.

Le lagrangien $L(\theta,\dot{\theta})$ est définit comme étant la différence entre l'énergie cinétique $k(\theta,\dot{\theta})$ et l'énergie potentielle u(q) :

$$L(\theta,\dot{\theta}) = k(\theta,\dot{\theta}) - u(\theta). \tag{3.8}$$

De plus, les équations de mouvement de Lagrange pour un manipulateur sont données par :

$$\frac{d}{dt}\left[\frac{\partial L}{\partial \dot{\theta}}\right] - \left[\frac{\partial L}{\partial \theta}\right] = \tau \tag{3.9}$$

Soit encore :

$$\frac{d}{dt}\left[\frac{\partial k}{\partial \dot{\theta}}\right]-\left[\frac{\partial k}{\partial \theta}\right]+\left[\frac{\partial L}{\partial \theta}\right]=\tau \qquad (3.10)$$

Où de manière équivalente :

$$\frac{d}{dt}\left[\frac{\partial k}{\partial \dot{\theta}_i}\right]-\left[\frac{\partial k}{\partial \theta_i}\right]+\left[\frac{\partial u}{\partial \theta_i}\right]=\tau_i \quad , \quad i=1...5 \qquad (3.11)$$

Où τ_i sont les couples exercés extérieurement par les actionneurs dans l'articulation i.

Pour ce faire on va calculer les énergies cinétiques et potentielles du système mécanique puis on procédera à une dérivation pour aboutir au modèle dynamique du robot.

3.2.1 Énergie Potentielle

L'énergie potentielle totale u est la somme des énergies potentielles u_i des centres de masse des articulations i=1...5 soit alors :

$$u(\theta)=-\sum_{i=1}^{5} m_i g_i^T \cdot {}^i p_{ci} \qquad (3.12)$$

Avec :

- m_i est la masse de l'articulation i
- g_i est le vecteur de gravité de l'articulation i
- ${}^i p_{ci}$ est le vecteur de coordonnées du centre de masse de l'articulation i dans le référentiel {i}.

3.2.2 Vecteurs de gravité et des centres de masses

Les coordonnées des centres de masse des articulations i et le vecteur de gravité exprimées dans les repères relatifs {i}et 0 pour le vecteur de gravité. Sont regroupées dans le tableau suivant :

Joint	Centre de masse	Vecteur de gravité
1	$[0 \quad 0 \quad z_1]^T$	$[0 \quad 0 \quad -g]^T$
2	$[0 \quad y_2 \quad 0]^T$	$[0 \quad 0 \quad -g]^T$
3	$[0 \quad 0 \quad z_3]^T$	$[0 \quad 0 \quad -g]^T$
4	$[0 \quad y_4 \quad 0]^T$	$[0 \quad 0 \quad -g]^T$
5	$[0 \quad 0 \quad z_5]^T$	$[0 \quad 0 \quad -g]^T$

On trouve alors :

$$u = g(p_0 + (p_{12} - \frac{1}{2}.p_5).c_2 + p_6.c_{24} + p_4.c_2.d_3) \tag{3.13}$$

Avec :

$p_0 = m_1.z_1$

$p_4 = m_4 + m_5 + m_6$

$p_5 = 2.m_3.z_3$

$p_6 = m_5.z_5$

$p_{12} = m_2.y_2$

3.2.3 Énergie cinétique

L'énergie cinétique totale est la somme des énergies cinétiques de différentes articulations i :

$$k(\theta, \dot{\theta}) = \sum_{i=1}^{i=n} k_i \tag{3.14}$$

Avec : $k_i = \frac{1}{2}.m_i.{}^0v_{ci}^T.{}^0v_{ci} + \frac{1}{2}.{}^iw_i^T.{}^{ci}I_i.{}^iw_i$ pour i=1 à 5. (3.15)

Où :

- m_i est la masse de l'articulation i.
- ${}^iw_i$ est la vitesse angulaire du centre de masse de l'articulation i dans le repère i.
- ${}^0v_{ci}$ est la vitesse linéaire du centre de masse de l'articulation i dans le repère 0.
  Elle est calculée par l'équation suivante : ${}^0v_{ci} = \frac{d}{dt}[{}^0P_{ci}]$. (3.16)
- ${}^{ci}I_i$ est le tenseur d'inertie au centre de masse de l'articulation i dans le repère i. alors il est sous une forme diagonale comme suit :

$${}^{ci}I_i = \begin{pmatrix} I_{xxi} & 0 & 0 \\ 0 & I_{yyi} & 0 \\ 0 & 0 & I_{zzi} \end{pmatrix} \qquad i = 1 \text{ à } 5 \tag{3.17}$$

L'énergie cinétique k_i est en effet composée de deux termes. le premier représente l'énergie cinétique due à la vitesse linéaire du centre de masse du joint i ; et le second représente celle due à la vitesse angulaire de ce joint.

À l'aide des outils informatiques et après simplification des formules trouvées ; l'énergie cinétique totale s'écrit :

$$k = \frac{1}{2}\left(-p_5.d_3.s_2^2 + p_4 d_3^2.s_2^2 + 2.p_6 s_2 s_{24} d_3 + p_2.c_2^2 + p_7.c_{24}^2 + p_1 + p_3 s_2^2 + p_8.s_{24}^2\right)\dot{\theta}_1^2$$

$$+ \frac{1}{2}\left((-p_5.d_3. + p_9 + p_4 d_3^2. + 2.p_6.c_4 d_3)\right)\dot{\theta}_2^2 + \frac{1}{2}.p_4 \dot{d}_3^2 + \frac{1}{2}.p_{10}\dot{\theta}_4^2 + \frac{1}{2}.p_{11}\dot{\theta}_5^2 +$$

$$p_{11} c_{24}.\dot{\theta}_1.\dot{\theta}_5 - p_6 s_4 \dot{d}_3.\dot{\theta}_2 - p_6 s_4.\dot{d}_3.\dot{\theta}_4 + (p_{10} + p_6.c_4.d_3).\dot{\theta}_2.\dot{\theta}_4 \qquad (3.18)$$

avec :

$p_1 = Izz_1$

$p_2 = Izz_3 - Iyy_2$

$p_3 = m_2 y_2^2 + m_3 z_3^2 - Ixx_2 + Ixx_{33}$

$p_7 = Izz_5 - Iyy_4$

$p_8 = m_4 y_4^2 + m_5 z_5^2 + Ixx_4 + Ixx_5$

$p_9 = m_2 y_2^2 + m_3 z_3^2 + m_4 y_4^2 + m_5 z_5^2 + Ixx_5 + Iyy_3 + Izz_2 + Izz_4$

$p_{10} = m_4 y_4^2 + m_5 z_5^2 + Izz_4 + Ixx_5$

$p_{11} = Izz_5$

Alors le Lagrangien nous donne les équations dynamiques décrivant le mouvement du robot sous la forme suivante :

$$\tau_1 = M_{11}.\ddot{\theta}_1 + M_{15}.\ddot{\theta}_5 + B_{11}\dot{\theta}_1\dot{\theta}_2 + B_{12}\dot{\theta}_1\dot{d}_3 + B_{13}\dot{\theta}_1\dot{\theta}_4$$

$$+ B_{17}\dot{\theta}_2\dot{\theta}_5 + B_{17}\dot{\theta}_4\dot{\theta}_5 + G_1 + F_1\dot{\theta}_1. \qquad (3.19)$$

$$\tau_2 = M_{22}.\ddot{\theta}_2 + M_{23}\ddot{d}_3 + M_{24}.\ddot{\theta}_4 - B_{17}\dot{\theta}_1\dot{\theta}_5 + B_{25}\dot{\theta}_2\dot{d}_3 + B_{26}\dot{\theta}_2\dot{\theta}_4$$

$$- \tfrac{1}{2}B_{11}\dot{\theta}_1^2 + B_{26}\dot{\theta}_4^2 + G_2 + F_2\dot{\theta}_2. \qquad (3.20)$$

$$\tau_3 = M_{23}.\ddot{\theta}_2 + M_{33}\ddot{d}_3 + M_{23}.\ddot{\theta}_4 + B_{36}\dot{\theta}_2\dot{\theta}_4 - \tfrac{1}{2}B_{12}\dot{\theta}_1^2$$

$$- \tfrac{1}{2}B_{25}\dot{\theta}_2^2 + \tfrac{1}{2}B_{36}\dot{\theta}_4^2 + G_3 + F_3\dot{\theta}_3. \qquad (3.21)$$

$$\tau_4 = M_{24}.\ddot{\theta}_2 + M_{23}\ddot{d}_3 + M_{44}.\ddot{\theta}_4 - B_{36}\dot{\theta}_2\dot{d}_3 - B_{17}\dot{\theta}_1\dot{\theta}_5$$

$$- \tfrac{1}{2}B_{13}\dot{\theta}_1^2 - \tfrac{1}{2}B_{26}\dot{\theta}_2^2 + G_4 + F_4\dot{\theta}_4. \qquad (3.22)$$

$$\tau_5 = M_{11}.\ddot{\theta}_1 + M_{55}.\ddot{\theta}_5 + B_{17}\dot{\theta}_1\dot{\theta}_2 + B_{17}\dot{\theta}_1\dot{\theta}_4 + F_5\dot{\theta}_5. \qquad (3.23)$$

De façon générale, on écrit ce modèle dynamique sous sa forme traditionnelle comme suit :

$$\tau(\theta, p) = M(\theta).\ddot{\theta} + V_m(\theta, \dot{\theta}).\dot{\theta} + G(\theta) + F(\dot{\theta}).\dot{\theta} \tag{3.24}$$

Puis on déduit ses composantes comme ci-dessous :

$$M(\theta) = \begin{bmatrix} M_{11} & 0 & 0 & 0 & M_{15} \\ 0 & M_{22} & M_{23} & M_{24} & 0 \\ 0 & M_{23} & M_{33} & M_{23} & 0 \\ 0 & M_{24} & M_{23} & M_{44} & 0 \\ M_{15} & 0 & 0 & 0 & M_{55} \end{bmatrix} \tag{3.25}$$

$$V_m(\theta,\dot{\theta}) = \begin{bmatrix} \frac{1}{2}(B_{11}\dot{\theta}_2 + B_{12}\dot{d}_3 + B_{13}\dot{\theta}_4) & \frac{1}{2}(B_{11}\dot{\theta}_1 + B_{17}\dot{\theta}_5) & \frac{1}{2}(B_{12}\dot{\theta}_1) & \frac{1}{2}(B_{13}\dot{\theta}_1 + B_{17}\dot{\theta}_5) & \frac{1}{2}B_{14}(\dot{\theta}_2 + \dot{\theta}_4) \\ -\frac{1}{2}(B_{11}\dot{\theta}_1 + B_{17}\dot{\theta}_5) & \frac{1}{2}(B_{25}\dot{d}_3 + B_{26}\dot{\theta}_4) & \frac{1}{2}(B_{25}\dot{\theta}_2) & \frac{1}{2}B_{26}(\dot{\theta}_2 + \dot{\theta}_4) & -\frac{1}{2}(B_{17}\dot{\theta}_1) \\ -\frac{1}{2}(B_{12}\dot{\theta}_1) & \frac{1}{2}(B_{16}\dot{\theta}_4 - B_{25}\dot{\theta}_2) & 0 & \frac{1}{2}B_{16}(\dot{\theta}_2 + \dot{\theta}_4) & 0 \\ -\frac{1}{2}(B_{13}\dot{\theta}_1 + B_{17}\dot{\theta}_5) & -\frac{1}{2}(B_{16}\dot{d}_3 + B_{26}\dot{\theta}_2) & -\frac{1}{2}(B_{16}\dot{\theta}_2) & 0 & -\frac{1}{2}(B_{17}\dot{\theta}_1) \\ \frac{1}{2}B_{14}(\dot{\theta}_2 + \dot{\theta}_4) & \frac{1}{2}(B_{17}\dot{\theta}_1) & 0 & \frac{1}{2}(B_{17}\dot{\theta}_1) & 0 \end{bmatrix} \tag{3.26}$$

$$G(\theta, p) = \begin{bmatrix} G_1 \\ G_2 \\ G_3 \\ G_4 \\ 0 \end{bmatrix} \tag{3.27}$$

$$F(\theta, p) = \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \end{bmatrix} \tag{3.28}$$

avec :

$M_{11} = -s_2^2\ d_3\ p_5 + 2\ s_2\ s_{24}\ d_3\ p_6 + s_2^2\ d_3^2\ p_4 + c_2^2\ p_2 + c_{24}^2\ p_7 + p_1 + s_2^2\ p_3 + s_{24}^2\ p_8.$

$M_{15} = M_{51} = c_{24}\ p_{11}.$

$M_{22} = -\ d_3\ p_5 + 2\ c_4\ d_3\ p_6 + d_3^2\ p_4 + p_9.$

$M_{23} = M_{32} = M_{34} = M_{43} = -s_4\ p_6.$

$M_{24} = M_{42} = c_4\ d_3\ p_6 + p_{10}.$

$M_{33} = p_4.$

$M_{44} = p_{10}.$

$M_{55} = p_{11}.$

et aussi :

$B_{11} = 2(-c_2 s_2\ d_3\ p_5 + (c_2\ s_{24} - c_{24}\ s_2)d_3.p_6 + s_2\ c_2\ d_3^2\ p_4 - c_2\ s_2\ p_2 - s_{24}\ c_{24}\ p_7 + s_2\ c_2\ p_3 + c_{24}\ s_{24}\ p_8).$

$B_{12} = 2(-0.5\ s_2^2\ p_5 + s_{24}\ s_2.p_6 + s_2^2.d_3.p_4).$

$B_{13} = 2(c_{24}\ s_2.p_6 - s_{24}\ c_{24}.p_7 + c_{24}\ s_{24}.p_8).$

$B_{17} = (-s_{24}\ p_{11}).$

$B_{25} = 2(-0.5\ p_5 + c_4\ p_6 + d_3\ p_4)$

$B_{26} = -2\ d_3\ s_4\ p_6$

$B_{36} = -2\ c_4\ p_6$

$G_1 = G_5 = 0.$

$G_2 = -g(\ s_2\ p_{12} - 0.5\ s_2\ p_5 + s_{24}\ p_6 + s_2\ d_3\ p_4\ ).$

$G_3 = g(\ c_2\ p_4\ ).$

$G_4 = -g(\ s_{24}\ p_6\ ).$

$$F = [F_1 \quad F_2 \quad F_3 \quad F_4 \quad F_5] = [p_{15} \quad p_{16} \quad p_{17} \quad p_{18} \quad p_{19}]$$

CHAPITRE 4

RÉALISATION PRATIQUE

Dans ce chapitre on pressente l'architecture du système réalisé avec les détails des différentes liaisons entres les composants de la carte, les techniques développées pour la mesure de la position, la génération des signaux PWM, la conversion analogique – numérique avec une description du matériel utilisé.

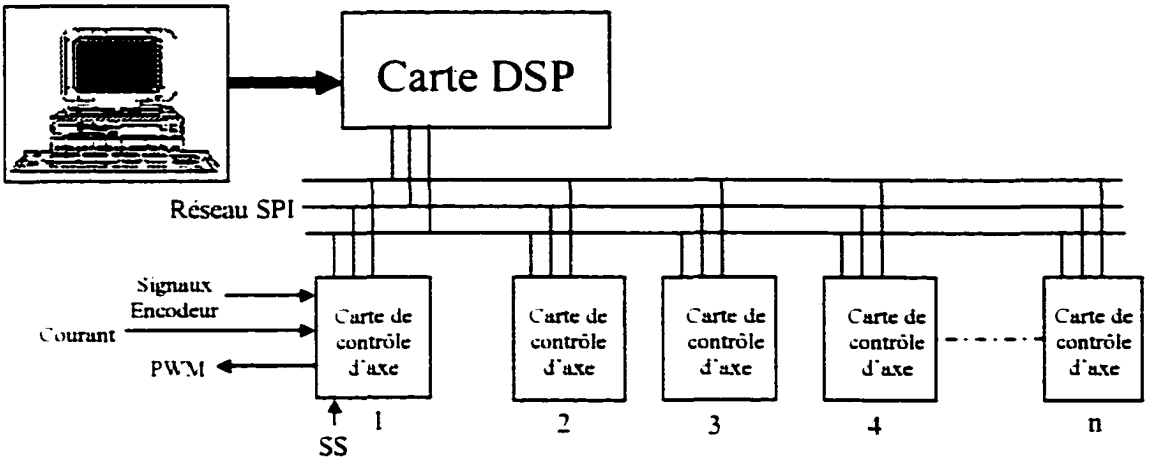


Figure 1 Architecture globale du système

Cette architecture, comme montre la figure ci-dessus, est basée sur un DSP. Celui-ci représente le maître du réseau, en établissant le lien avec le PC pour l'affichage des informations en provenance des cartes de contrôles d'axes. Ces cartes sont composées de deux microcontrôleurs, un pour l'acquisition de la position et l'autre pour la génération des signaux PWM, la conversion analogique numérique des courants et la communication avec le DSP.

Avant de donner des détails sur ce système, on commence par une description du matériel utilisé.

4.1 Présentation et description du matériel utilisé

4.1.1 Les microcontrôleurs PIC

Les PIC sont des microcontrôleurs dont le traitement s'effectue sur 8 bits. Un seul cycle d'horloge suffit pour le traitement d'une instruction. La gamme des *PIC* est très étendue et variée, ce qui permet de choisir le composant le plus adéquat à notre application.

Les PIC se présentent sous divers boîtiers : DIL (8 à 40), MQFP, PLCC, TQFP, en fonction du nombre de ports d'entrées-sorties qu'ils possèdent. Certains possèdent même une EEPROM, comme la série 16f8XX ce qui autorise une reprogrammation et une souplesse d'utilisation très utile parfois. D'autres se caractérisent par la présence de ports possédant des entrées analogiques munies de convertisseurs analogique/numérique et des ports de communications série.

Tout ceci fait du PIC, en général, un microcontrôleur simple à utiliser, flexible et efficace, qui ne nécessite que très peu de composants externes pour son fonctionnement.

Vu ces avantages offerts par les PICs de Microship, on a choisi le 16F877 qui intègre une mémoire programme flash ; ce qui permet une écriture et réécriture, un port de communication série SPI, des convertisseurs analogiques-numériques, un générateur de signaux PWM et des ports d'entrées-sorties TTL.

4.1.2 Le processeur numérique du signal

Un DSP est un type particulier de microprocesseur. Il se caractérise par le fait qu'il intègre un ensemble de fonctions spéciales. Ces fonctions sont destinées à le rendre particulièrement performant dans le domaine du traitement numérique du signal.

Comme un microprocesseur classique, un DSP est mis en œuvre en lui associant de la mémoire (RAM, ROM) et des périphériques. Un DSP typique a comme vocation de servir dans des systèmes de traitements autonomes. Il se présente donc généralement sous la forme d'un microcontrôleur intégrant, selon les marques et les gammes des constructeurs, de la mémoire, des temporisations, des ports séries synchrones rapides, des contrôleurs DMA, des ports d'E/S divers.

4.1.3 Principales distinctions entre un microprocesseur et un DSP

Après avoir été numérisé, le signal se présente sous la forme d'une suite de valeurs numériques discrètes. Cette suite de valeurs (ou échantillons) est apte à être stockée et traitée par un système informatique. Par nature, le traitement numérique du signal revient à effectuer essentiellement des opérations arithmétiques de base du type $A = (B * C) + D$.

Un microprocesseur classique va nécessiter plusieurs cycles d'horloge pour effectuer un tel calcul, par exemple, un 68000 de Motorola a besoin de :

- 10 cycles d'horloge pour effectuer une addition.

- 70 cycles d'horloge pour effectuer une multiplication.

Soit 80 cycles pour seulement calculer A. Si ce temps est admissible dans des applications informatiques courantes, il n'est pas acceptable pour faire du traitement rapide du signal. Les DSP sont donc conçus pour optimiser ce temps de calcul. À cet effet, ils disposent de fonctions optimisées permettant de calculer A beaucoup plus rapidement.

Dans la pratique, la plupart des DSP ont un jeu d'instructions spécialisé permettant de lire en mémoire une donnée, d'effectuer une multiplication puis une addition, et enfin d'écrire en mémoire le résultat, le tout en un seul cycle d'horloge.

Tous les systèmes à base de DSP bénéficient des avantages suivants :

- Souplesse de la programmation : un DSP est avant tout un processeur exécutant un programme de traitement du signal. Ceci signifie que le système bénéficie donc d'une grande souplesse de développement. De plus, les fonctions de traitements numériques peuvent évoluer en fonction des mises à jour des programmes, et cela pendant toute la durée de vie du produit incluant le système. Ainsi, il est possible de modifier des paramètres sans une nécessité de changement matériel.

- Des possibilités propres au système de traitement numérique du signal. Certaines fonctions de traitement ou de calcul sont difficiles à implanter en analogique, voire irréalisables.

- Stabilité : en analogique, les composants sont toujours plus ou moins soumis à des variations de leurs caractéristiques en fonction de la température, de la tension d'alimentation, du vieillissement. Une étude sérieuse doit tenir compte de ces phénomènes, ce qui complique et augmente le temps de développement. Ces inconvénients n'existent pas en numérique.

- Répétitivité, reproductibilité : les valeurs des composants analogiques sont définies avec une marge de précision plus ou moins grande. Dans ces conditions, aucun montage analogique n'est strictement reproductible à l'identique, il existe toujours des différences qu'il convient de maintenir dans des limites acceptables. Un programme réalisant un traitement numérique est par contre parfaitement reproductible, « à l'infini ».

De ce fait, nous avons choisi pour notre application, parmi les différents types de DSP existant, ceux possédant une mémoire *FLASH (320LF2407)* offrant une souplesse d'écriture et réécriture du programme, ainsi qu'une interface graphique conviviale qui aide à visualiser l'état des différentes zones mémoires en temps réel.

4.1.4 Les encodeurs angulaires

Tous les contrôles de mouvements mécaniques nécessitent un encodeur angulaire effectuant le lien entre l'entraînement et la commande. Les encodeurs angulaires transforment les différents mouvements de rotation en signaux électriques. Le cœur de l'encodeur optique est formé d'un disque, disposant d'un nombre de segments clairs sombres, détecté par une photo transition. La structure définit la résolution ainsi que la précision de positionnement du mouvement à contrôler.

Deux systèmes de mesures séparées optiquement génèrent deux séquences d'impulsions déphasées électriquement de 90°. Une reconnaissance du sens de rotation est ainsi rendue possible.

Les encodeurs émettent des signaux de sortie digitaux. Une période de signal peut être divisée en 4 phases de mesure lorsqu'on analyse l'écart des deux fronts CH_A et CH_B (Fig.22) on obtient de cette manière une quadruple exploitation du nombre d'impulsion utilisé ce qui augmente la résolution de capture de position.

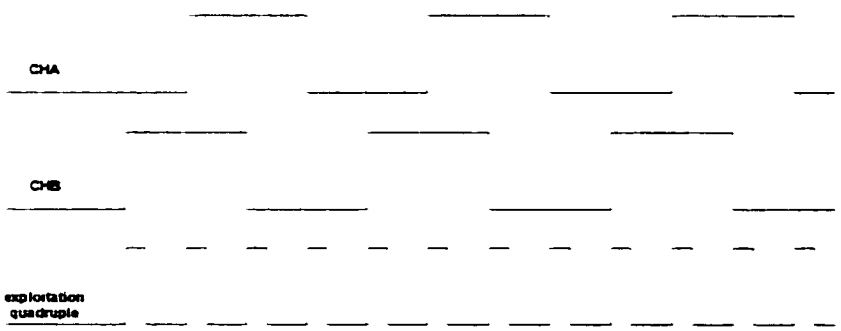


Figure 22 Exploitation quadruple

4.2 Programmes développés

4.2.1 Capture de position

Pour la mesure de la position, on a dédié un PIC (16f877). Cette détection de position se fait par comptage ou décomptage des fronts montants et des fronts descendants provenant du canal CHA et CHB de l'encodeur.

Cette capture est réalisée par génération d'interruption à l'intérieur du microcontrôleur, à savoir qu'on dispose de deux sources d'interruption matérielle dans ce PIC, une qui est générée par changement d'état ou de niveau présent sur la broche 4 du port B, celle-là ne présente pas de problème particulier, puisqu'on est capable de générer une interruption sur front montant et descendant à la fois.

Par contre pour la deuxième source (INT0), elle est configurable, ça veut dire qu'on ne peut enclencher la sous routine d'interruption que sur un front bien déterminé. Mais on a contourné ce problème en modifiant la configuration au cours de l'exécution du programme. Cela implique une commutation du front montant au front descendant de façon périodique: ainsi on est capable de détecter les deux types de front avec la même interruption.

La détection du sens de rotation se fait à l'intérieur des sous-routines d'interruptions attribués à CHA et CHB.

Une fois on est dans la sous routine générée par CHA par exemple, on fait une lecture de l'état de CHB dans le cas où CHA est sur front montant ; Si CHB a un niveau logique bas (0), on est dans le sens 1 (horaire), sinon on est dans le sens 2 (antihoraire).

Et si l'interruption est générée par un front descendant, le sens 1 correspond à un niveau haut de CHB et le sens 2 correspond à un niveau bas. La même logique s'applique pour les interruptions générées par CHB.

Dans la figure suivante on présente l'organigramme de la partie comptage. décomptage et détection du sens de rotation. Ainsi que le programme fait en assembleur en annexe 2.

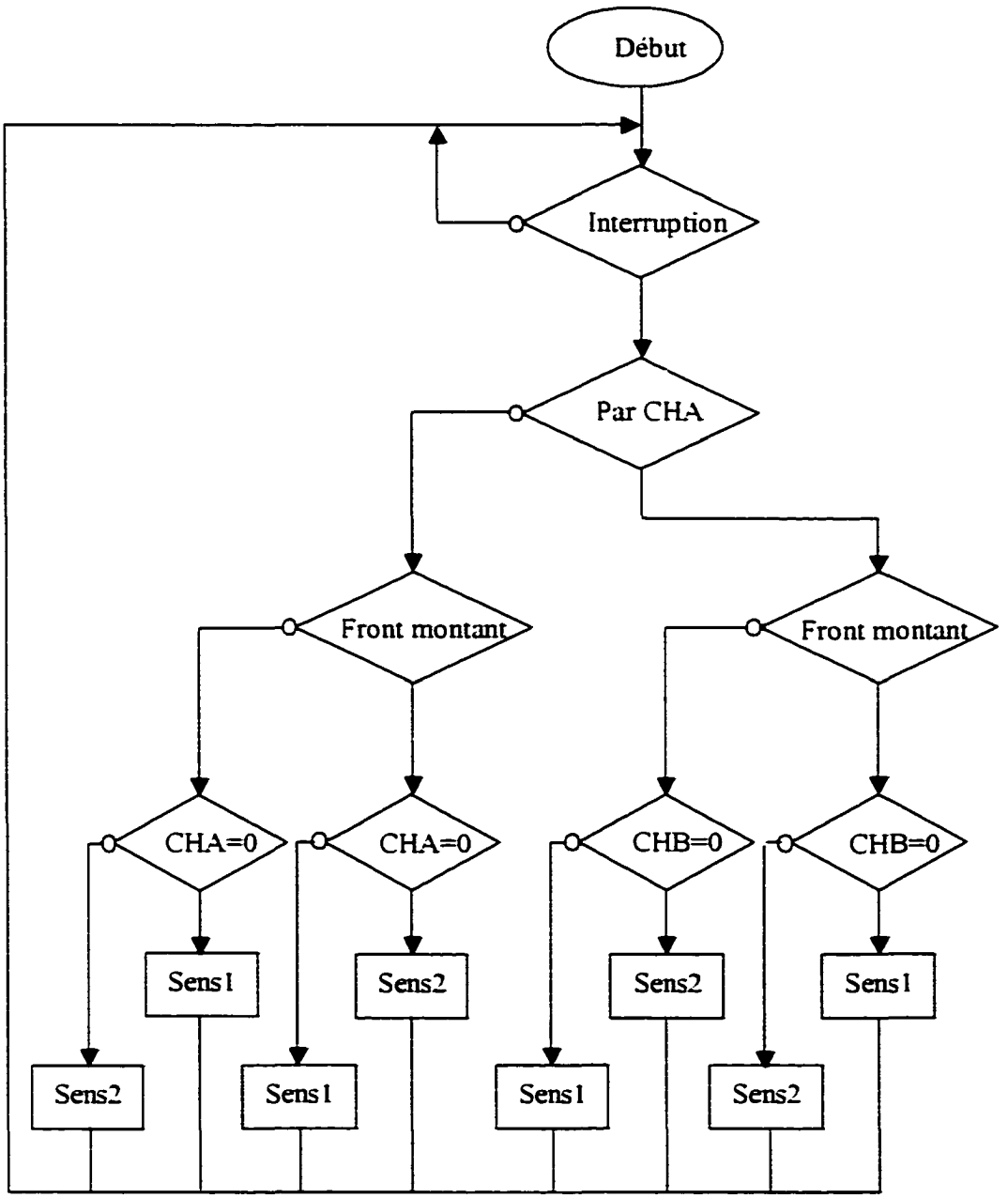


Figure 23 Organigramme de capture de position d'un axe du robot

Dans les sous programmes du sens 1 on incrémente et dans celui du sens 2 on décrémente. Pour avoir une meilleure précision, on a présenté la position sur 4 octets, ce qui donne une marge de comptage et décomptage de 0 à 16 777 253.

L'organigramme suivant illustre la procédure de comptage/décomptage. On désigne par X1, X2, X3 et X4 les cases mémoires réservées pour la position.

Sens 1 :

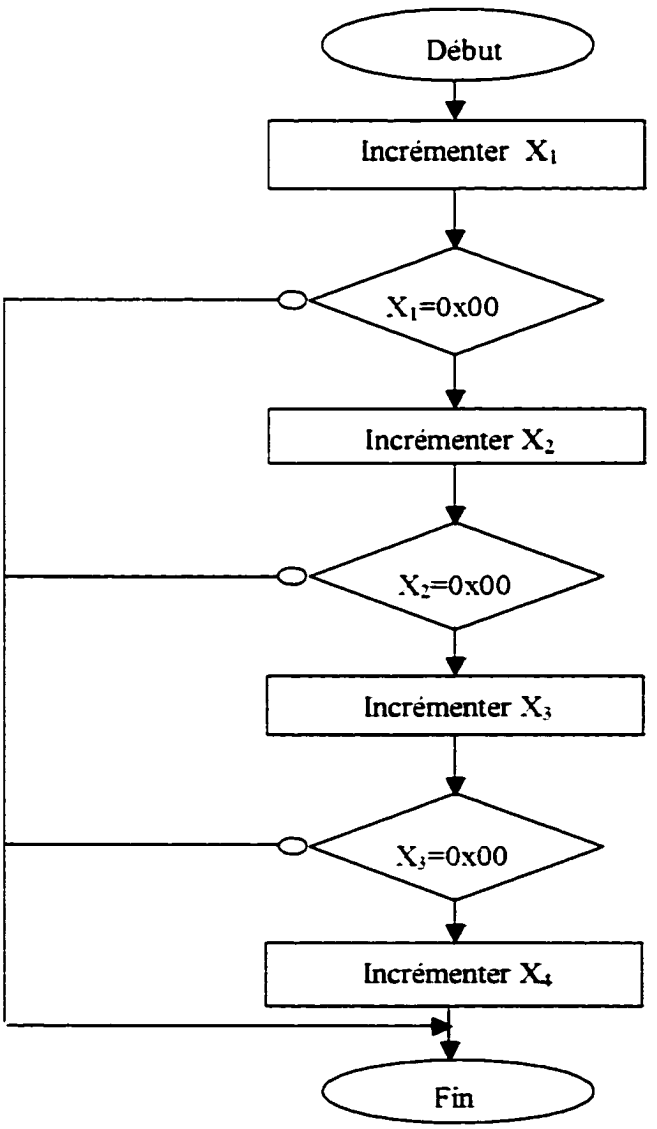


Figure 24 Organigramme incrémentation de position

Sens 2 :

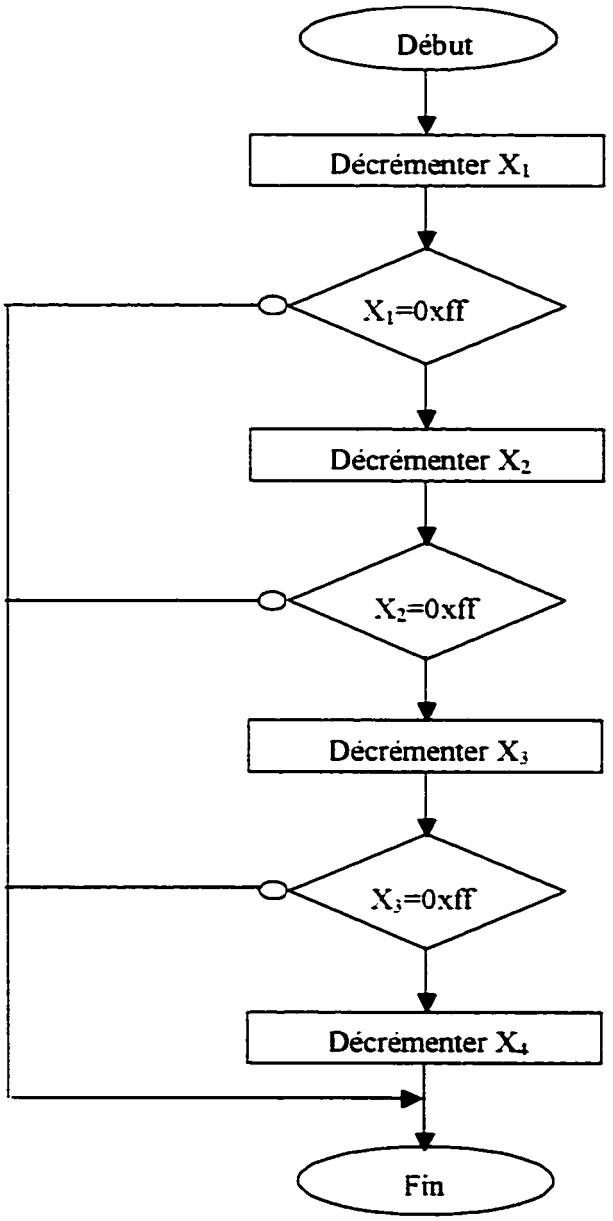


Figure 25 Organigramme décrémentation de position

4.2.2 Procédure de transfert de la position vers le PIC 1

La position une fois capturée, elle est transférée au 1er microcontrôleur via le port parallèle. Ce transfert s'effectue par une interruption générée par l'horloge du 2^{ième} microcontrôleur à chaque 3 ms. Dans le 1er microcontrôleur on profite de cette interruption pour calculer la vitesse : une fois on est dans la sous routine de réception de position, on calcule la vitesse avec un Δt=3ms. Dans les figures suivantes, on donne le schéma de câblage ainsi que les organigrammes de transfert de la position côté 1er microcontrôleur, 2^{ième} microcontrôleur et l'organigramme de calcul de vitesse.

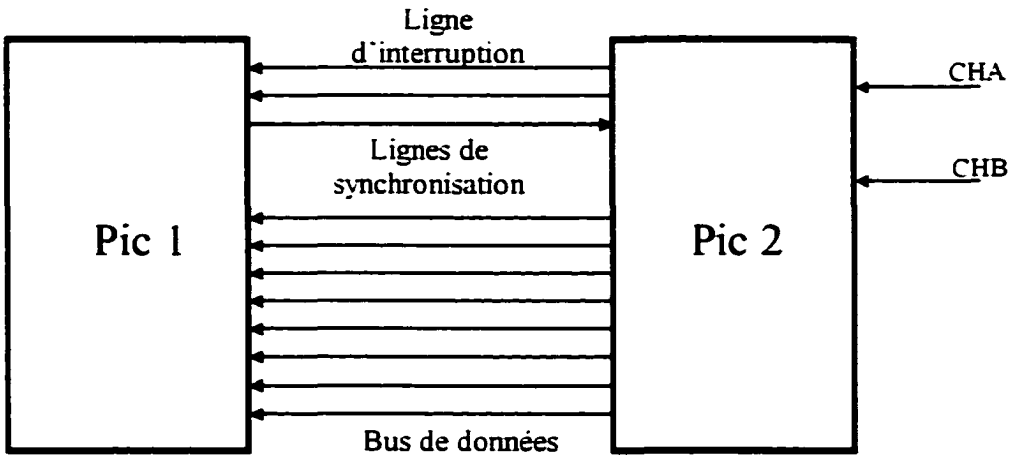


Figure 26 La liaison entre les deux microcontrôleurs

La communication entre les deux microcontrôleurs se fait par une liaison parallèle, ce qui offre un temps d'échange très réduit et un taux d'erreurs presque nul.

Alors pour faire passer la position du Pic2 vers le Pic1, le premier génère une interruption interne par son horloge, cette interruption est transformée en un signal logique sur son port A, qui active à son tour l'interruption (INT0) du PIC récepteur. Une fois on est dans la sous routine associée à cette interruption, le transfert de la position commence par une demande du premier octet en changeant l'état des lignes de synchronisation. Une fois ce changement détecté par le PIC 2, il envoie le premier octet et inverse le niveau de la ligne de synchronisation pour autoriser la lecture par

le PIC 1. Après le stockage du premier octet, le PIC 1 change l'état de la ligne de synchronisation afin de demander le deuxième octet, et la même procédure se répète jusqu'au dernier octet.

Une fois la position est transférée, le PIC 1 calcule la vitesse.

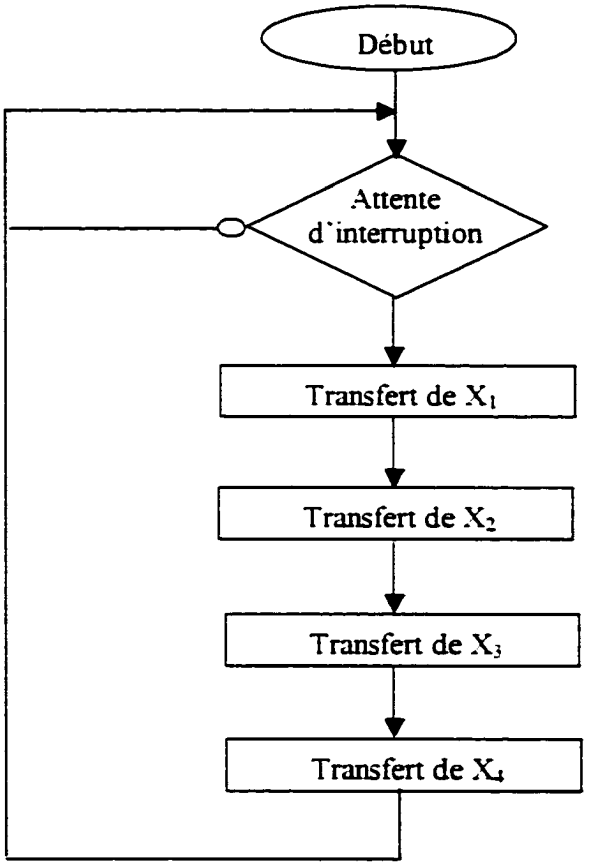


Figure 27 Organigramme de transfert de position

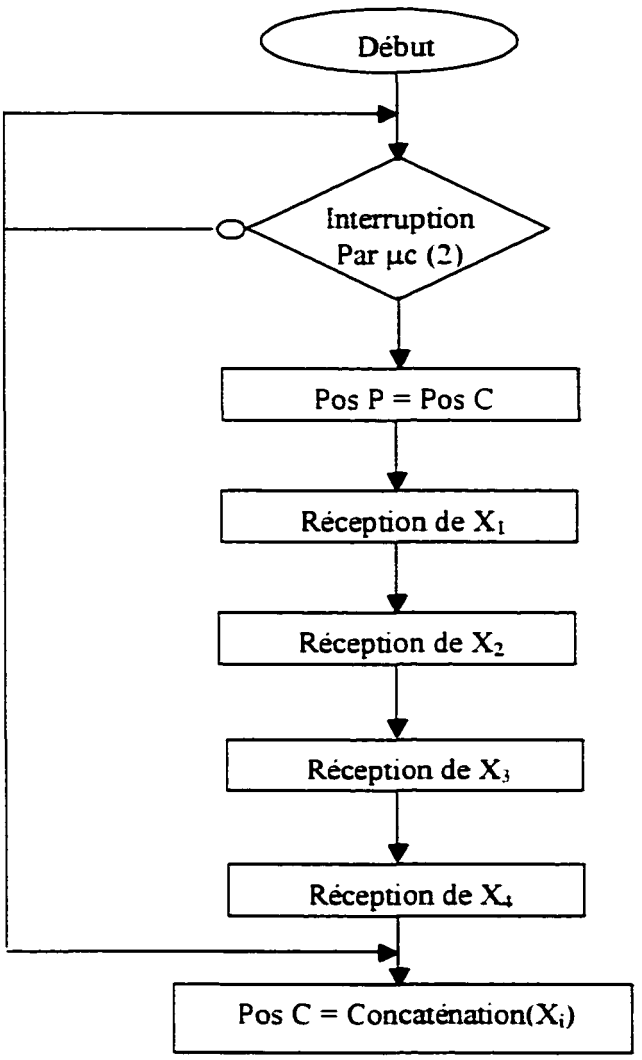


Figure 28 Organigramme de réception de position par microcontrôleur (2)

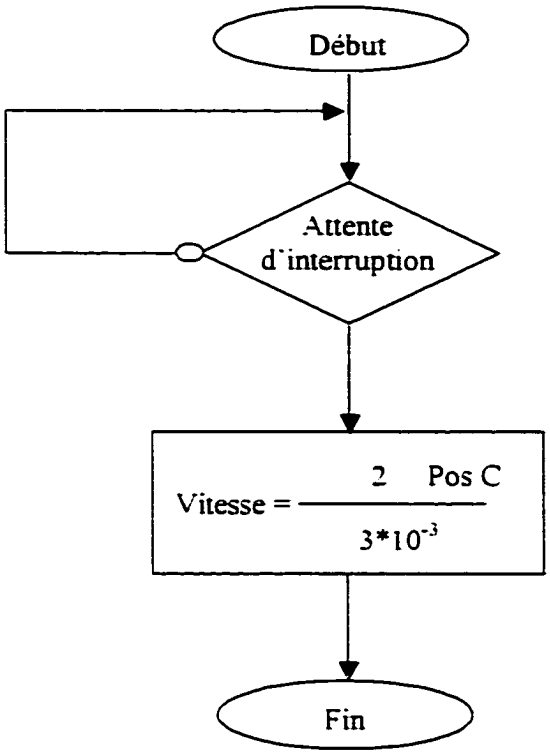


Figure 29 Organigramme de calcul de vitesse

Dans l'annexe 2, vous trouverez les programmes d'échange entre les deux microcontrôleurs et celui du calcul de vitesse.

4.3 Procédure de commande d'axes

La trajectoire étant fournie par le DSP sous forme de consignes couples, le microcontrôleur (1) transforme ces couples désirés en une tension variable. Ce calcul est celui d'une boucle de régulation avec un correcteur proportionnel.

Le couple réel ou mesuré est obtenu par lecture du courant induit et sa numérisation par des convertisseurs analogique/numérique intégrés situés dans le microcontrôleur, et enfin, on multiplie la lecture par la constante magnétique du moteur pour avoir le couple.

La commande fournie par le régulateur est convertie en un signal carré à rapport cyclique variable (PWM), qui est ensuite amplifié par un circuit de puissance.

La figure suivante montre l'organigramme de commande d'axe.

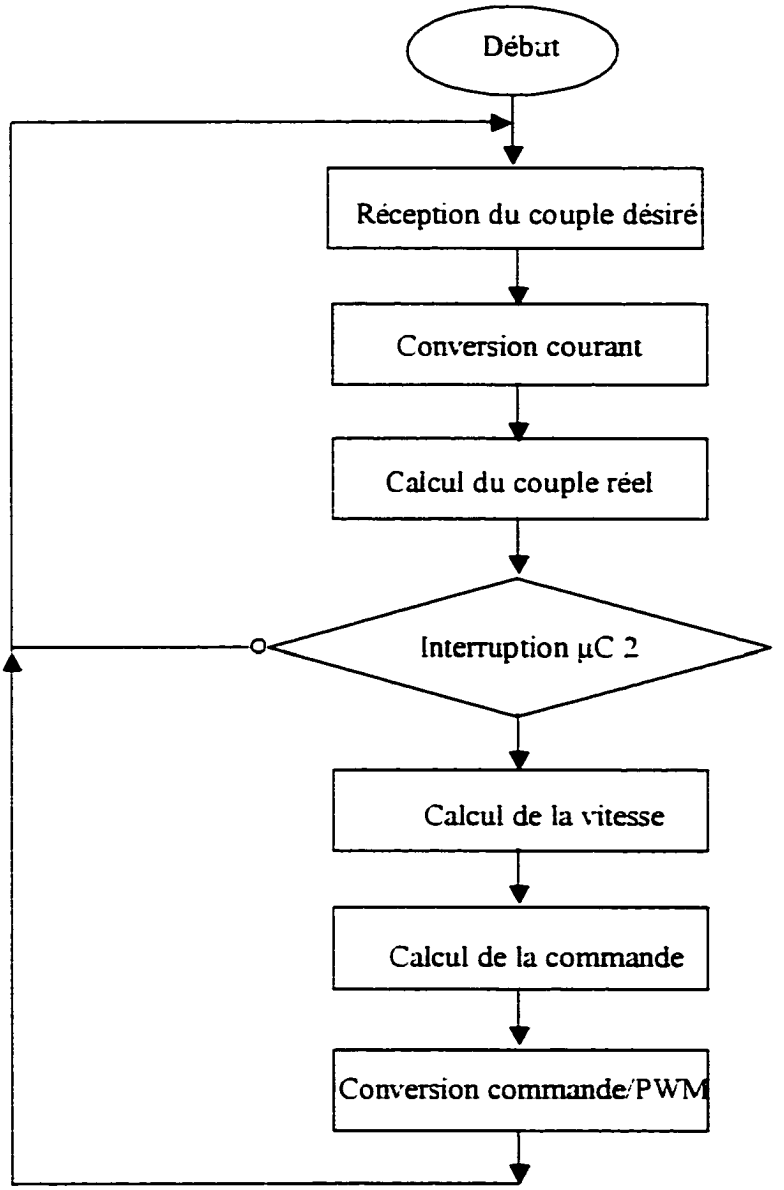


Figure 30 Organigramme de commande moteur

4.4 Communication entre le DSP et les cartes de commande

La communication entre le DSP et les cartes de commande s'établit en passant par leurs modules SPI, selon un protocole d'échange qui a été déterminé en fonction de l'ordre et la quantité des trames à transmettre.

Ces trames sont formées principalement de la position, la vitesse et le couple désiré. À chaque période d'échantillonnage, le DSP sélectionne une carte esclave et lui demande d'envoyer le premier octet de la position, puis le deuxième, le troisième et le dernier, une fois que ces octets sont reçus par le DSP, il construit la position par concaténation et conversion de l'hexadécimal au décimal. Ensuite le DSP demande la vitesse, puis il envoie le couple désiré.

L'organigramme suivant illustre ce protocole d'échange et dans l'annexe 2 vous trouverez les programmes.

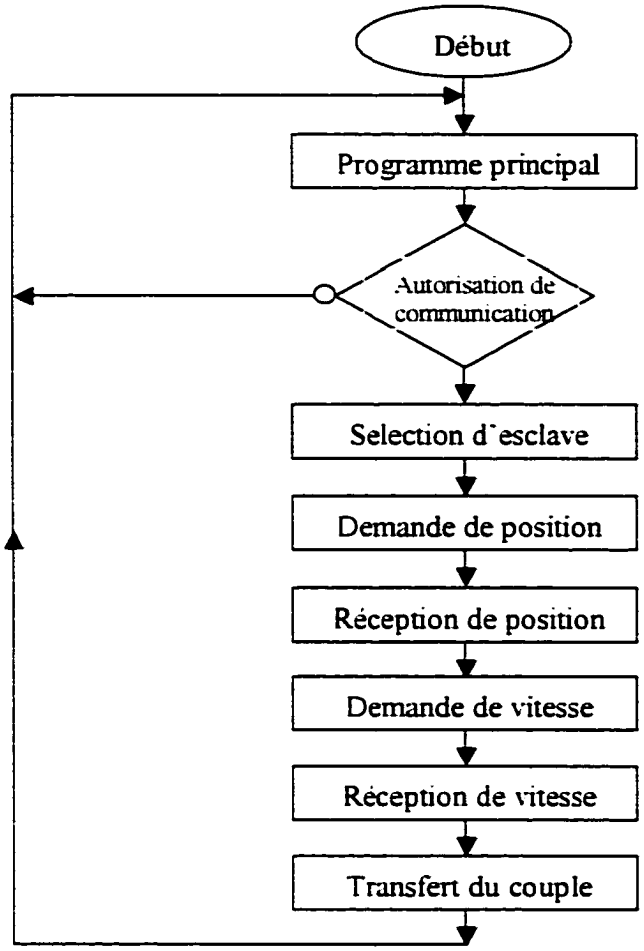


Figure 31 Organigramme du programme DSP

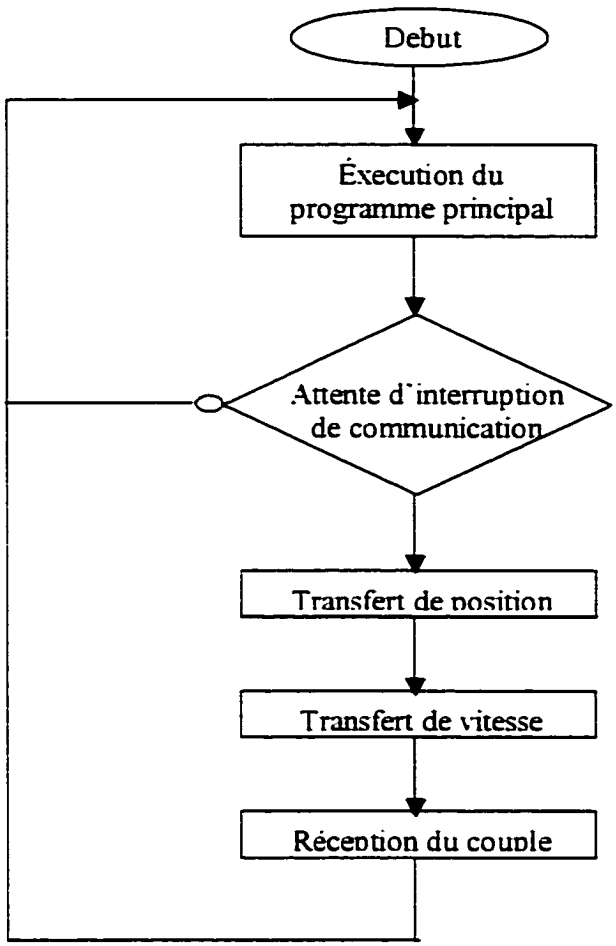


Figure 32 Organigramme de communication du microcontroleur

La sélection de l'esclave se fait matériellement en passant par une carte de démultiplexage, dont les sorties sont connectées à l'entrée SS de chaque microcontrôleur. Une fois cette entrée est à niveau bas (0), le microcontrôleur concerné entre en communication avec le DSP. Juste après la mise à zéro du SS, le DSP envoie les données et le signal d'horloge pour synchroniser la communication. A la fin de chaque transfert, le DSP positionne le SS à niveau haut, pour geler la communication afin de laisser le temps nécessaire à l'esclave pour digérer les données reçues et préparer les données de la prochaine communication.

La figure 33 donne l'allure du signal d'horloge (2) en fonction du SS (1).

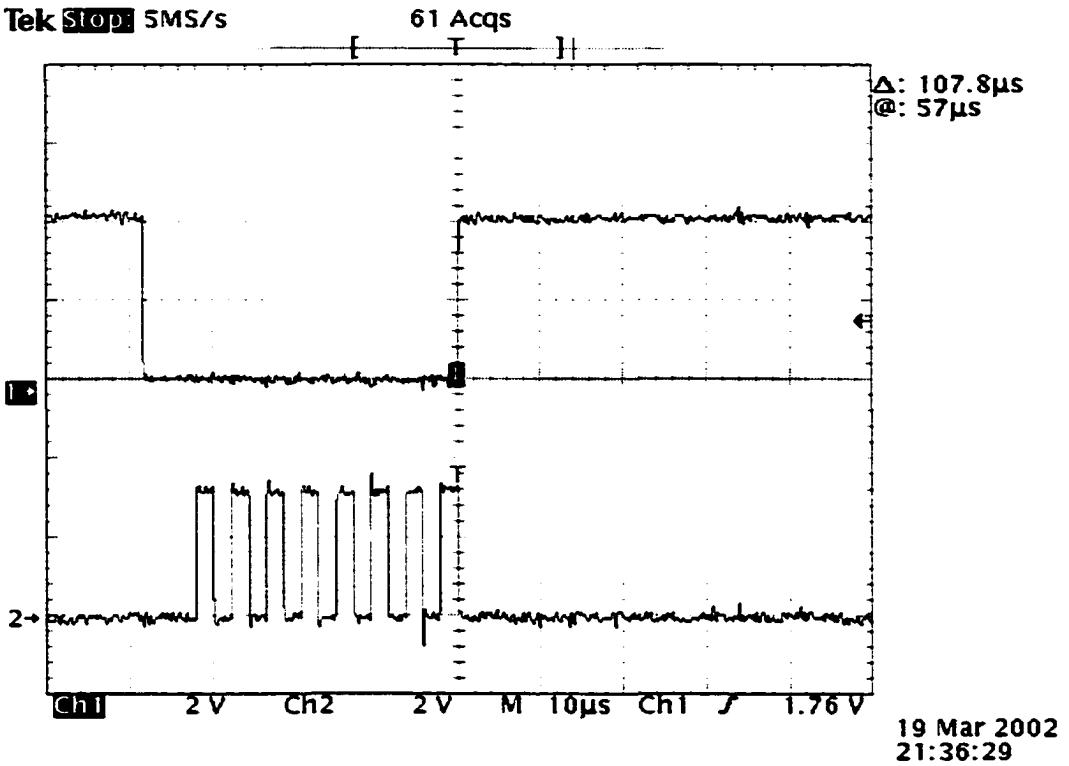


Figure 33 Les signaux SS et horloges

En utilisant ce principe, on est capable de sélectionner la carte désirée par le signal SS, et commencer l'échange de données sur les lignes MISO et MOSI, les Figures 34 et 35 donnent un exemple sur cet échange :

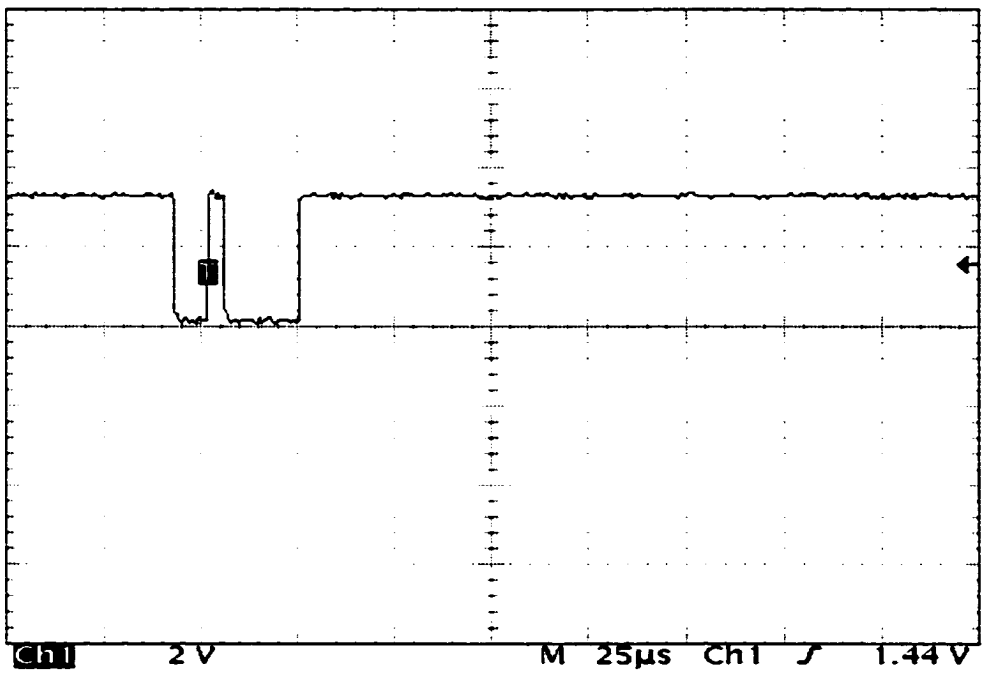


Figure 34 Émission d'un octet de position vers le DSP.

Ici on voit l'émission d'un octet de position vers le DSP.

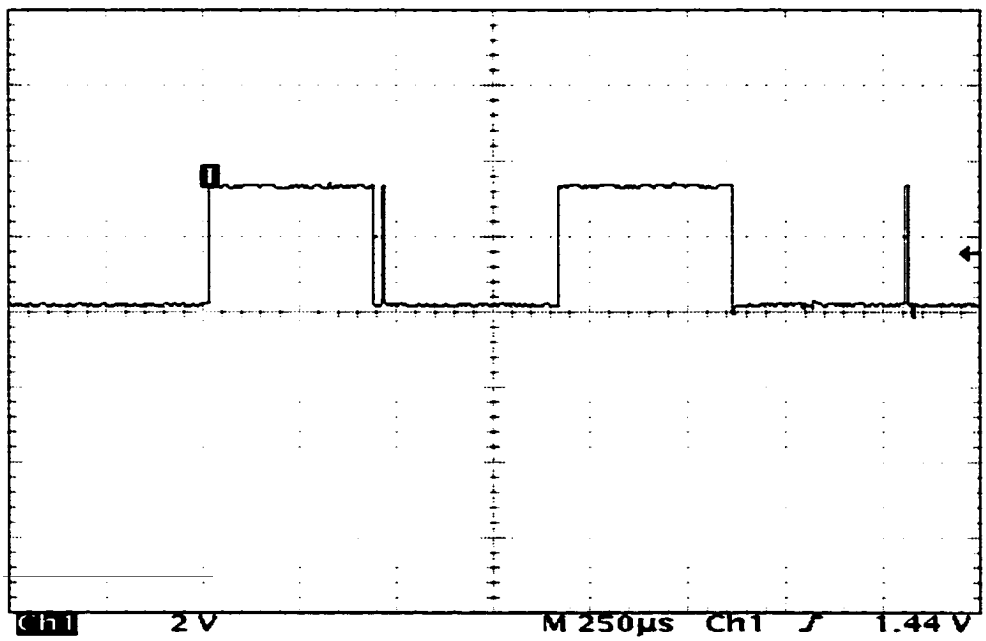


Figure 35 Demande de la position par le DSP

Dans la Figure 35, le DSP demande au PIC l'envoi du premier puis le deuxième octet de la position.

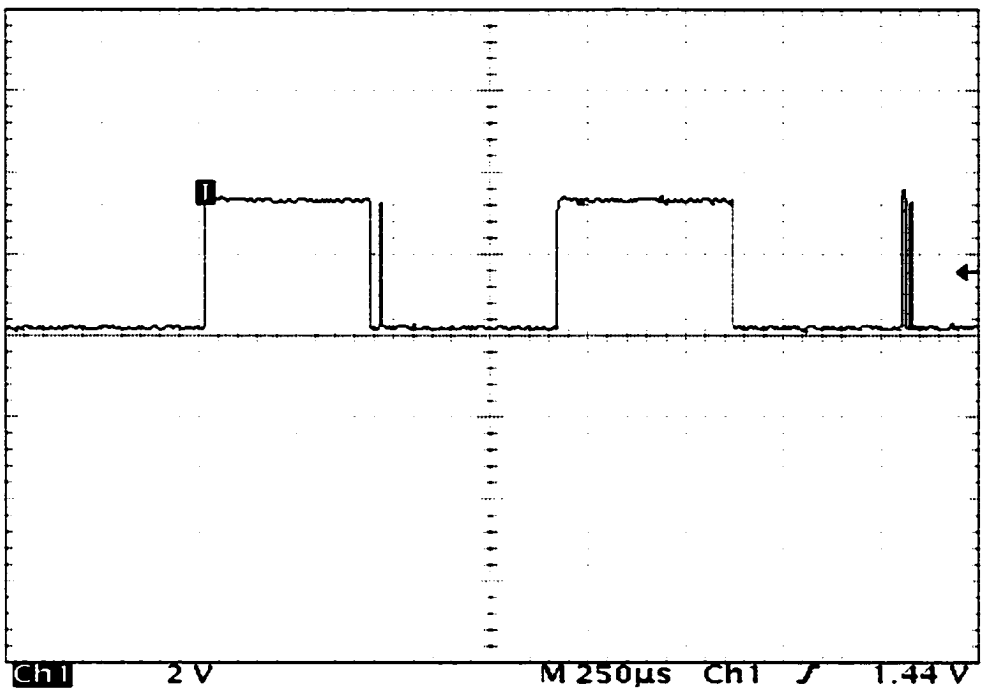


Figure 36 Émission du sens de rotation et PWM.

Dans la Figure 36, l'envoi par le DSP du sens de rotation en premier, puis le rapport cyclique.

Une fois le PIC reçoit ces deux ordres, il les applique, en modifiant l'état de la broche du sens de rotation, et en générant un signal PWM avec le rapport cyclique demandé.

Les Figures de 37 à 39, donnent le signal PWM généré, pour trois rapports cycliques désirés.

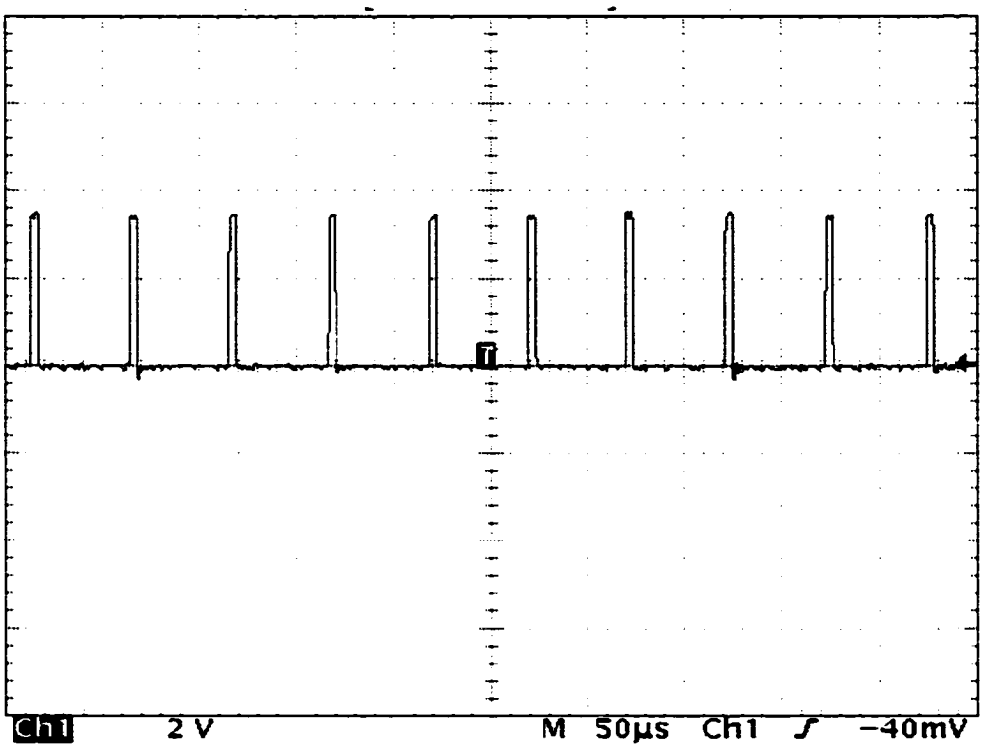


Figure 37 Signal PWM avec un rapport cyclique de 8%

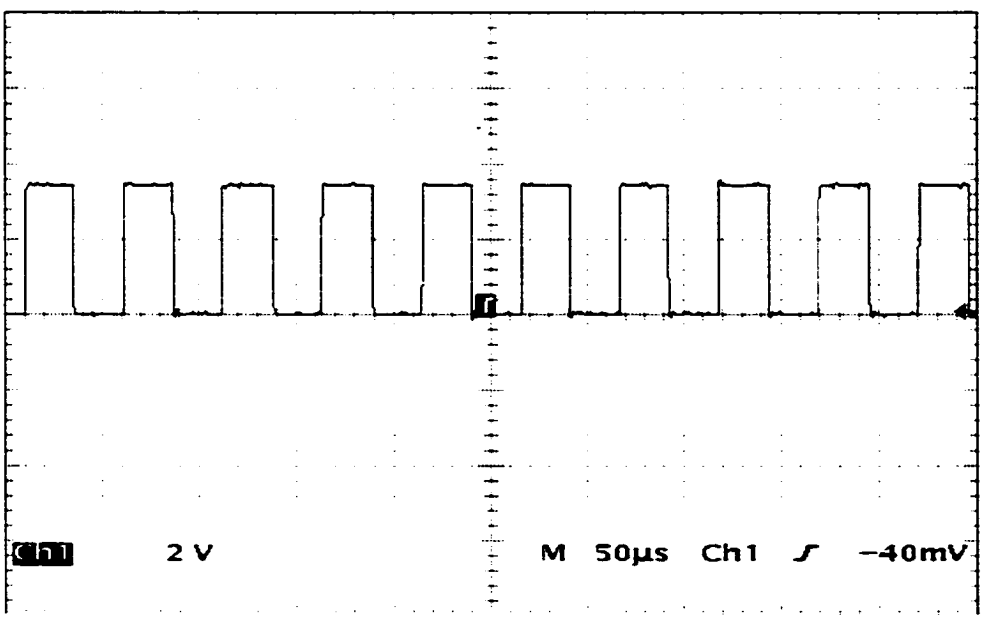


Figure 38 Signal PWM avec un rapport cyclique de 50%

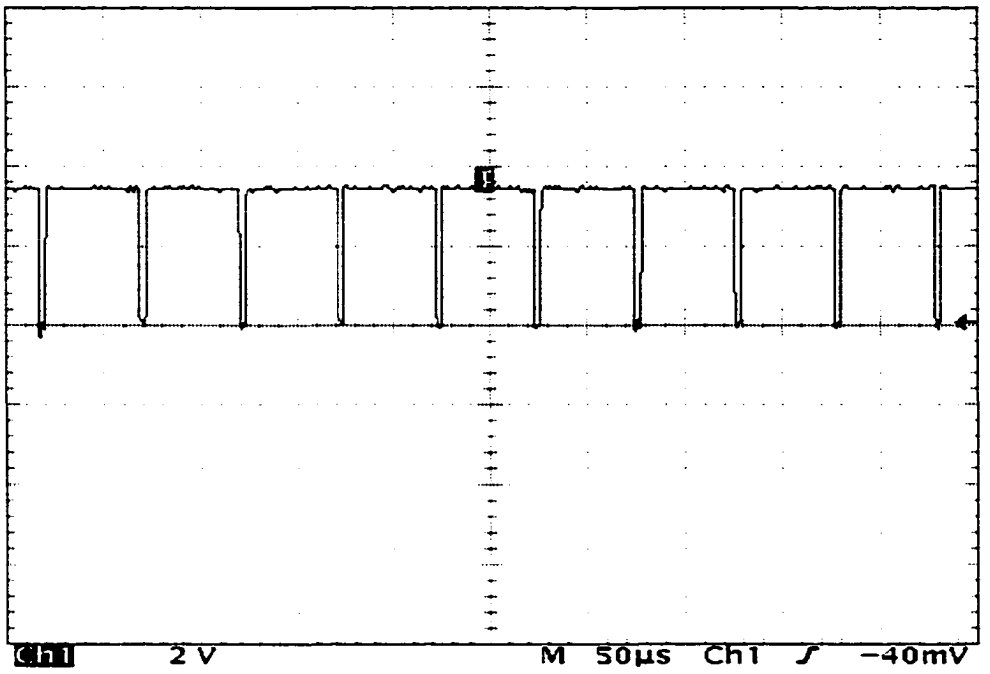


Figure 39 Signal PWM avec un rapport cyclique de 90%

Dans la Figure 40 qui suit, on donne les signaux de synchronisation entre les deux PICs pour le transfert de la position (circuit encodeur vers le deuxième microcontrôleur).

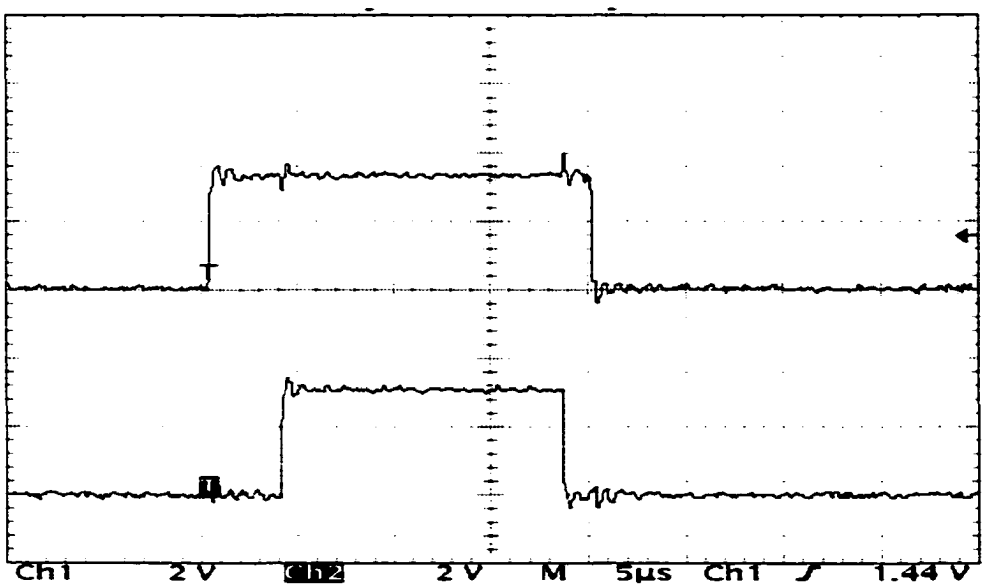


Figure 40 Signaux de synchronisations entre les deux PICs.

Par le premier signal, le circuit encodeur, génère une interruption du deuxième PIC. ce dernier, donne un acquittement indiquant qu'il est prêt à recevoir (deuxième signal sur le graphe). Une fois les deux circuits sont synchronisés, ils commencent le transfert, à la réception du dernier octet, le PIC récepteur remet à zéro le signal de synchronisation indiquant qu'il a reçu correctement la position.

Dans la Figure 41, on donne l'interface de communication et de programmation du DSP.

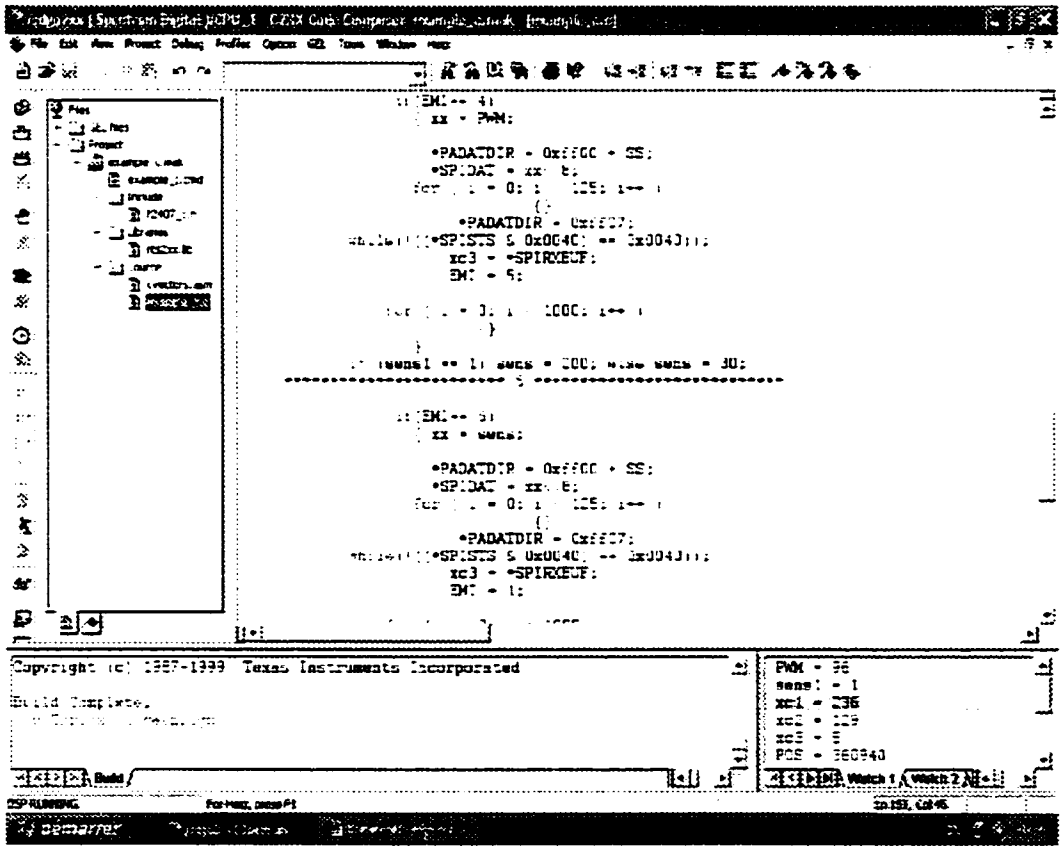


Figure 41 Signaux de synchronisations entre les deux PIC.

À partir de cette interface, on transfert le programme au DSP, ainsi qu'on est en mesure de récupérer ou modifier les données en cours d'exécution.
Dans la petite fenêtre à droite, on modifie le signal PWM, le sens de rotation et en visualise la position.

Dans les Figures 42 et 43, on affiche la position enregistrée pour les deux sens de rotation de l'axe du robot.

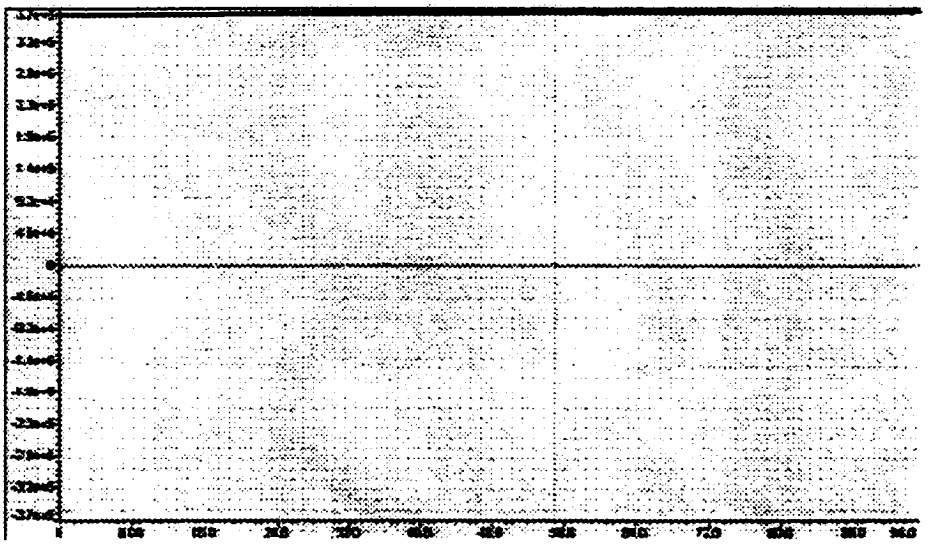


Figure 42 Position enregistrée pour le sens 1.

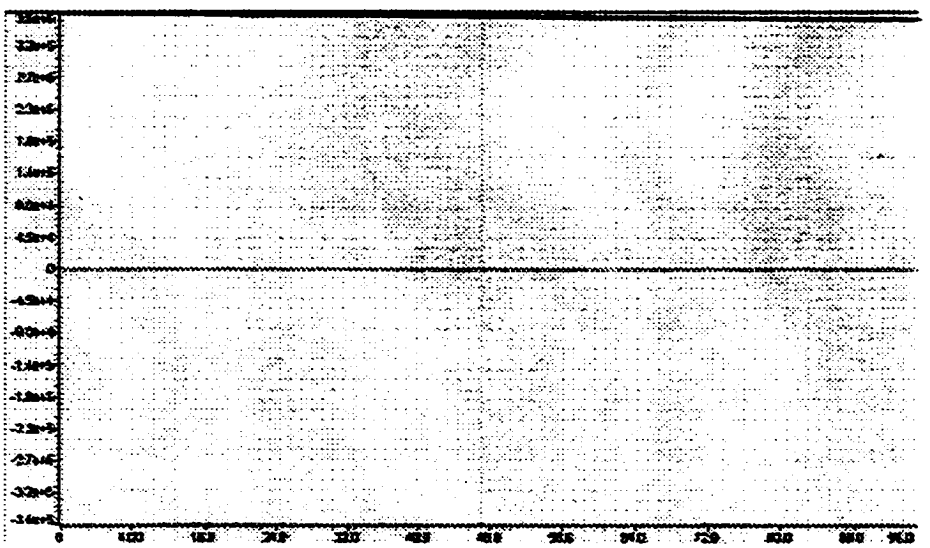


Figure 43 Position enregistrée pour le sens 0.

CONCLUSION

Dans le cadre de ce projet, nous avons pu tirer quelques conclusions concernant le système de commande d'axe sur lequel on a effectué nos travaux.

Nous avons constaté que le système ayant une architecture décentralisé permet une flexibilité en terme d'ajout et (ou) de suppression d'un et (ou) plusieurs axes d'articulation.

La rapidité de communication entre les différents composants de ce système représente un grand avantage, ce qui réduit considérablement le temps de transfert de données, et par conséquent d'augmenter la capacité de gérer un nombre maximal de degré de liberté. De même, le système ayant cette propriété (rapidité) nous encourage à implanter des algorithmes plus compliqué et plus rigoureux.

Le système étant ordiné, le coût d'acquisition et de maintenance s'avère très spéculant, par conséquent le désir de s'investir pour développer ce genre de système augmente.

Vu les caractéristiques offertes par l'architecture réalisées, on recommande son amélioration en remplaçant le DSP utilisé (TMS 320LF2407) qui est programmable en C et en assembleur par d'autres kits DSP intégrants des compilateurs de fichiers programmés avec Matlab, chose qui fera du système un outil très puissant pour la recherche dans le domaine de la robotique.

ANNEXE 1

Développement théorique du robot

A.1 Matrices de transformation homogène

En général la matrice de transformation homogène s'écrit de la façon suivante :

$$ {}^{i-1}_{i}T = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ c\alpha_{i-1}s\theta_i & c\alpha_{i-1}c\theta_i & -s\alpha_{i-1} & -d_i s\alpha_{i-1} \\ s\alpha_{i-1}s\theta_i & s\alpha_{i-1}c\theta_i & c\alpha_{i-1} & -d_i c\alpha_{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (A.1) $$

La partie qui va suivre, déterminera la matrice de transformation homogène qui correspond à chaque articulation : ${}^{0}_{1}T$; ${}^{1}_{2}T$; ${}^{2}_{3}T$; ${}^{3}_{4}T$; ${}^{4}_{5}T$.

$$ {}^{0}_{1}T = \begin{bmatrix} c\theta_1 & -s\theta_1 & 0 & 0 \\ s\theta_1 & c\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (A.2) $$

$$ {}^{1}_{2}T = \begin{bmatrix} c\theta_2 & -s\theta_2 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s\theta_2 & c\theta_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (A.3) $$

$$ {}^{2}_{3}T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & d_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (A.4) $$

$$ {}^{3}_{4}T = \begin{bmatrix} c\theta_4 & -s\theta_4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s\theta_4 & -c\theta_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (A.5) $$

$$ {}^{4}_{5}T = \begin{bmatrix} c\theta_5 & -s\theta_5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s\theta_5 & -c\theta_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (A.5) $$

A.2 Étude de la cinématique directe

Pour étudier la cinématique directe, il faut tout d'abord déterminer la matrice de transformation globale du robot, cette matrice se définit comme suit :

$$ {}^{0}_{5}T = {}^{0}_{1}T * {}^{1}_{2}T * {}^{2}_{3}T * {}^{3}_{4}T * {}^{4}_{5}T \qquad (A.6) $$

On trouve après simplification et en posant cos (θ_i) = C_i ; cos($\theta_i+\theta_j$) = C_{ij} et sin($\theta_i+\theta_j$) = S_{ij}.

$$ {}^{0}_{5}T = \begin{bmatrix} C_1C_5C_{24} - S_1S_5 & -C_1S_5C_{24} - S_1C_5 & -C_1S_{24} & -C_1S_2d_3 \\ S_1C_5C_{24} + C_1S_5 & -S_1S_5C_{24} + C_1C_5 & -S_1S_{24} & -S_1S_2d_3 \\ C_5S_{24} & -S_5S_{24} & C_{24} & C_2d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (A.7) $$

A.3 Étude de la cinématique inverse

Afin d'étudier la cinématique inverse, on va assumer qu'un point dans l'espace est définit par une matrice 4 par 4 de la même forme que la matrice obtenue par l'étude de la cinématique directe. Cette matrice définit les coordonnées (position et orientation) de ce point dans l'espace.

$$M = \begin{array}{c} \begin{matrix} m & n & o & p \end{matrix} \\ \begin{bmatrix} m_x & n_x & o_x & p_x \\ m_y & n_y & o_y & p_y \\ m_z & n_z & o_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{array} \qquad \text{(A.8)}$$

Sachant que M et $\mathcal{T}$ ont la même forme, on veut déterminer les paramètres (θ_1 , θ_2 , θ_4 , θ_5 et d_3) du robot qui permettent d'atteindre ce point.

$$p = \begin{bmatrix} P_x = -c_1 s_2 d_3 \\ P_y = -s_1 s_2 d_3 \\ P_z = c_2 d_3 \end{bmatrix} \Rightarrow \frac{p_y}{P_x} = \frac{s_1}{c_1} \Rightarrow \theta_1 = A\tan 2(P_y, P_x) \qquad \text{(A.9)}$$

$$\Rightarrow P_x^2 + P_y^2 - P_z^2 = d_3^2 \Rightarrow d_3 = \sqrt{P_x^2 + P_y^2 + P_z^2} \qquad \text{(A.10)}$$

$$\Rightarrow c_1 P_x + s_1 P_y = -s_2 d_3$$

$$\Rightarrow \frac{c_1 P_x + s_1 P_y}{-P_z} = \frac{s_2}{c_2} \Rightarrow \theta_2 = A\tan 2(c_1 P_x + s_1 P_y, -P_z) \qquad \text{(A.11)}$$

De même on a :

$$O = \begin{bmatrix} O_x = -c_1 s_{24} \\ O_y = -s_1 s_{24} \\ O_z = c_{24} \end{bmatrix}$$

$$\Rightarrow -c_1 O_x - s_1 O_y = s_{24}$$

$$\Rightarrow \frac{-c_1 O_x - s_1 O_y}{O_z} = \frac{s_{24}}{c_{24}} \Rightarrow \theta_2 + \theta_4 = A\tan 2(-c_1 O_x - s_1 O_y, O_z)$$

$$\Rightarrow \theta_4 = A\tan 2(-c_1 O_x - s_1 O_y, O_z) - \theta_2$$

$$\Rightarrow \theta_4 = A\tan 2(-c_1 O_x - s_1 O_y, O_z) - A\tan 2(c_1 P_x + s_1 P_y, -P_z) \quad (A.12)$$

On a : $\begin{bmatrix} m_z = c_5 s_{234} \\ n_z = -s_5 s_{234} \end{bmatrix}$

$$\Rightarrow -\frac{n_z}{m_z} = \frac{s_5}{c_5} \Rightarrow \theta_5 = A\tan 2(n_z, m_z) \quad (A.13)$$

A.4 Vitesses angulaires et linéaires du Robot

Dans le cas d'une articulation rotative les vitesses angulaires et linéaires sont données par les formules suivantes [1]:

$$^{i+1}W_{i+1} = {}^{i+1}_{i}R\ {}^{i}W_i + \dot{\theta}_{i+1}\ {}^{i+1}\hat{Z}_{i+1} \quad (A.14)$$

$$^{i+1}V_{i+1} = {}^{i+1}_{i}R\ ({}^{i}V_i + {}^{i}W_i \otimes {}^{i}P_{i+1}) \quad (A.15)$$

Dans le cas d'une articulation prismatique les vitesses angulaires et linéaires sont données par les formules suivantes :

$$^{i+1}W_{i+1} = {}^{i+1}_{i}R\ {}^{i}W_i \quad (A.16)$$

$$^{i+1}V_{i+1} = {}^{i+1}_{i}R({}^{i}V_i + {}^{i}W_i \otimes {}^{i}P_{i+1}) + \dot{d}_{i+1}\ {}^{i+1}\hat{Z}_{i+1} \quad (A.17)$$

Pour chaque articulation, on applique ces formules pour aboutir aux résultats montrés dans les paragraphes suivants :

A.4.1 Articulation 1 : rotative

On a :

$$ {}^{0}_{1}T = \begin{bmatrix} c\theta_1 & -s\theta_1 & 0 & 0 \\ s\theta_1 & c\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (A.18) $$

A.4.2 Articulation 2 : rotative

$$ {}^{1}W_1 = {}^{1}_{0}R \ {}^{0}W_0 + \dot{\theta}_1 \ {}^{1}\hat{Z}_1 = \dot{\theta}_1 \ {}^{1}\hat{Z}_1 = \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 \end{bmatrix} \qquad (A.19) $$

$$ {}^{1}V_1 = {}^{1}_{0}R \ ({}^{0}V_0 + {}^{0}W_0 \otimes {}^{0}P_1) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \qquad (A.20) $$

On a :

$$ {}^{2}_{1}T = \begin{bmatrix} c\theta_2 & -s\theta_2 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s\theta_2 & c\theta_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (A.21) $$

$$ {}^{1}_{2}R = \begin{bmatrix} c\theta_2 & -s\theta_2 & 0 \\ 0 & 0 & -1 \\ s\theta_2 & c\theta_2 & 0 \end{bmatrix} \Rightarrow {}^{2}_{1}R = \begin{bmatrix} c\theta_2 & 0 & s\theta_2 \\ -s\theta_2 & 0 & c\theta_2 \\ 0 & -1 & 0 \end{bmatrix} \qquad (A.22) $$

Les vitesses sont donc :

$$ {}^2W_2 = {}^2_1R\ {}^1W_1 + \dot{\theta}_2\ {}^2\hat{Z}_2 = \begin{bmatrix} c\theta_2 & 0 & s\theta_2 \\ -s\theta_2 & 0 & c\theta_2 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_1 \end{bmatrix} + \dot{\theta}_2 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} $$

$$ = \begin{bmatrix} \dot{\theta}_1 s\theta_2 \\ \dot{\theta}_1 c\theta_2 \\ \dot{\theta}_2 \end{bmatrix} \qquad \text{(A.23)} $$

$$ {}^2V_2 = {}^2_1R\,({}^1V_1 + {}^1W_1 \otimes {}^1P_2) = \begin{bmatrix} c\theta_2 & 0 & s\theta_2 \\ -s\theta_2 & 0 & c\theta_2 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \qquad \text{(A.24)} $$

puisqu'on a : $ {}^1V_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} $ (A..25)

et $ {}^1P_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} $ (A.26)

A.4.3 Articulation 3 : prismatique

On a :

$$ {}^3_3T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & d_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \text{(A.27)} $$

$$ {}^{3}_{4}R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \Rightarrow {}^{3}_{2}R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \qquad (A.28) $$

Les vitesses sont donc :

$$ {}^{3}W_3 = {}^{3}_{2}R\ {}^{2}W_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 s\theta_2 \\ \dot{\theta}_1 c\theta_2 \\ \dot{\theta}_2 \end{bmatrix} = \begin{bmatrix} \dot{\theta}_1 s\theta_2 \\ \dot{\theta}_2 \\ -\dot{\theta}_1 c\theta_2 \end{bmatrix} \qquad (A.29) $$

$$ {}^{3}V_3 = {}^{3}_{2}R\,({}^{2}V_2 + {}^{2}W_2 \otimes {}^{2}P_3) + \dot{d}_3\ {}^{3}\hat{Z}_3 \qquad (A.30) $$

$$ = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \left(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} \dot{\theta}_1 s\theta_2 \\ \dot{\theta}_1 c\theta_2 \\ \dot{\theta}_2 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ d_3 \\ 0 \end{bmatrix} \right) + \begin{bmatrix} 0 \\ 0 \\ \dot{d}_3 \end{bmatrix} $$

$$ = \begin{bmatrix} -d_3\dot{\theta}_2 \\ d_3\dot{\theta}_1 s\theta_2 \\ \dot{d}_3 \end{bmatrix} \qquad (A.31) $$

A.4.4 Articulation 4 : rotative

On a :

$$ {}^{4}_{3}T = \begin{bmatrix} c\theta_4 & -s\theta_4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s\theta_4 & -c\theta_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (A.32) $$

$$ {}^{3}_{4}R = \begin{bmatrix} c\theta_4 & -s\theta_4 & 0 \\ 0 & 0 & 1 \\ -s\theta_4 & -c\theta_4 & 0 \end{bmatrix} \Rightarrow {}^{4}_{3}R = \begin{bmatrix} c\theta_4 & 0 & -s\theta_4 \\ -s\theta_4 & 0 & -c\theta_4 \\ 0 & 1 & 0 \end{bmatrix} \qquad \text{(A.33)} $$

Les vitesses sont données par :

$$ {}^{4}W_4 = {}^{4}_{3}R\ {}^{3}W_3 + \dot{\theta}_4\ {}^{4}\hat{Z}_4 = \begin{bmatrix} c\theta_4 & 0 & -s\theta_4 \\ -s\theta_4 & 0 & -c\theta_4 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 s\theta_2 \\ \dot{\theta}_2 \\ -\dot{\theta}_1 c\theta_2 \end{bmatrix} + \dot{\theta}_4 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} $$

$$ = \begin{bmatrix} \dot{\theta}_1 s_{2+4} \\ \dot{\theta}_1 c_{2+4} \\ \dot{\theta}_2 + \dot{\theta}_4 \end{bmatrix} \qquad \text{(A.34)} $$

$$ {}^{4}V_4 = {}^{4}_{3}R\,({}^{3}V_3 + {}^{3}W_3 \otimes {}^{3}P_4) = \begin{bmatrix} c\theta_4 & 0 & -s\theta_4 \\ -s\theta_4 & 0 & -c\theta_4 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} -d_3\dot{\theta}_2 \\ d_3\dot{\theta}_1 s\theta_2 \\ \dot{d}_3 \end{bmatrix} $$

$$ = \begin{bmatrix} -d_3\dot{\theta}_2 c\theta_4 - \dot{d}_3 s\theta_4 \\ -d_3\dot{\theta}_2 s\theta_4 - \dot{d}_3 c\theta_4 \\ d_3\dot{\theta}_1 s\theta_2 \end{bmatrix} \qquad \text{(A.35)} $$

puisqu'on a : $ {}^{3}P_4 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} $

A.4.5 Articulation 5 : rotative

On a : $${}^{4}_{5}T = \begin{bmatrix} c\theta_5 & -s\theta_5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s\theta_5 & -c\theta_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^{4}_{5}R = \begin{bmatrix} c\theta_5 & -s\theta_5 & 0 \\ 0 & 0 & 1 \\ -s\theta_5 & -c\theta_5 & 0 \end{bmatrix} \Rightarrow {}^{5}_{4}R = \begin{bmatrix} c\theta_5 & 0 & -s\theta_5 \\ -s\theta_5 & 0 & -c\theta_5 \\ 0 & 1 & 0 \end{bmatrix} \qquad (A.36)$$

Les vitesses sont données par :

$${}^{4}W_5 = {}^{5}_{4}R\,{}^{4}W_4 + \dot{\theta}_5\,{}^{5}Z_5 = \begin{bmatrix} c\theta_5 & 0 & -s\theta_5 \\ -s\theta_5 & 0 & -c\theta_5 \\ 0 & 1 & 0 \end{bmatrix}\begin{bmatrix} \theta_1 s_{2+4} \\ \theta_1 c_{2+4} \\ \theta_2 + \theta_4 \end{bmatrix} + \dot{\theta}_5 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} c_5\theta_1 s_{2+4} - s_5(\theta_2 - \theta_4) \\ -s_5\theta_1 s_{2+4} - c_5(\theta_2 - \theta_4) \\ \theta_5 + \theta_1 c_{2+4} \end{bmatrix} \qquad (A.37)$$

$${}^{5}V_5 = {}^{5}_{4}R({}^{4}V_4 - {}^{4}W_4 \otimes {}^{4}P_5) = \begin{bmatrix} c\theta_5 & 0 & -s\theta_5 \\ -s\theta_5 & 0 & -c\theta_5 \\ 0 & 1 & 0 \end{bmatrix}\begin{bmatrix} -d_3\theta_2 c\theta_4 - d_3 s\theta_4 \\ -d_3\theta_2 s\theta_4 - d_3 c\theta_4 \\ d_3\theta_1 s\theta_2 \end{bmatrix}$$

$$= \begin{bmatrix} (-d_3\theta_2 c\theta_4 - d_3 s\theta_4)c_5 - (d_3\theta_1 s\theta_2)s_5 \\ -(-d_3\theta_2 c\theta_4 - d_3 s\theta_4)s_5 - (d_3\theta_1 s\theta_2)c_5 \\ -d_3\theta_2 s\theta_4 - d_3 c\theta_4 \end{bmatrix} \qquad (A.38)$$

Puisqu'on a : $${}^{4}P_5 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

A.4.6 Vitesse angulaire du poignet

Elle est donnée par :

$$
{}^{0}_{5}W_{5} = {}^{0}_{5}R\,{}^{5}W_{5}
$$

$$
= \begin{bmatrix} C_1C_5C_{24} - S_1S_5 & -C_1S_5C_{24} - S_1C_5 & -C_1S_{24} \\ S_1C_5C_{24} + C_1S_5 & -S_1S_5C_{24} + C_1C_5 & -S_1S_{24} \\ C_5S_{24} & -S_5S_{24} & C_{24} \end{bmatrix} \begin{bmatrix} c_5\dot{\theta}_1 s_{2+4} & -s_5(\dot{\theta}_2 + \dot{\theta}_4) \\ -s_5\dot{\theta}_1 s_{2+4} & -c_5(\dot{\theta}_2 + \dot{\theta}_4) \\ \dot{\theta}_5 + \dot{\theta}_1 c_{2+4} & \end{bmatrix}
$$

$$
= \begin{bmatrix} s_1\dot{\theta}_2 - c_1 s_{24}\dot{\theta}_5 + s_1\dot{\theta}_4 \\ -c_1\dot{\theta}_2 - s_1 s_{24}\dot{\theta}_5 - c_1\dot{\theta}_4 \\ c_{24}\dot{\theta}_5 + \dot{\theta}_1 \end{bmatrix} = \begin{bmatrix} W_x \\ W_y \\ W_z \end{bmatrix} \qquad \text{(A.39)}
$$

A.4.7 Vitesse linéaire du poignet

La vitesse linéaire du poignet est déterminée en appliquant la même formule avec les vitesses linéaires.

$$
{}^{0}V_5 = {}^{0}_{5}R\,{}^{5}V_5
$$

$$
= \begin{bmatrix} c_1c_5c_{24} - s_1s_5 & -c_1s_5c_{24} - s_1s_5 & -c_1s_{24} \\ s_1c_5c_{24} + c_1s_5 & -s_1s_5c_{24} + c_1c_5 & -s_1s_{24} \\ c_5s_{24} & -s_5s_{24} & c_{24} \end{bmatrix} \begin{bmatrix} (-d_3\dot{\theta}_2 c\theta_4 - \dot{d}_3 s\theta_4)c_5 - (d_3\dot{\theta}_1 s\theta_2)s_5 \\ -(-d_3\dot{\theta}_2 c\theta_4 - \dot{d}_3 s\theta_4)s_5 - (d_3\dot{\theta}_1 s\theta_2)c_5 \\ -d_3\dot{\theta}_2 s\theta_4 - \dot{d}_3 c\theta_4 \end{bmatrix}
$$

$$= \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} = \begin{bmatrix} s_1 s_2 d_3 \dot{\theta}_1 - c_1 c_2 d_3 \dot{\theta}_2 - c_1 s_2 \dot{d}_3 \\ -c_1 s_2 d_3 \dot{\theta}_1 - s_1 c_2 d_3 \dot{\theta}_2 - s_1 s_2 \dot{d}_3 \\ -s_2 d_3 \dot{\theta}_2 + c_2 \dot{d}_3 \end{bmatrix} \quad \text{(A.40)}$$

A.5 Génération de la trajectoire 1

A.5.1 Technique d'interpolation

La technique d'interpolation consiste à interpoler entre les points intermédiaires, dans le but de produire une trajectoire souple (ou lisse), c'est-à-dire, qu'elle admet au moins deux dérivées continues et finies pour éviter le cas d'une accélération infinie. Pour satisfaire à cette exigence, on peut adopter les trajectoires cubiques qui ont deux dérivées finies.

A.5.2 Trajectoires cubiques avec deux points intermédiaires

Prenons le cas le plus simple. Le point de départ q^0 et le point d'arrivée q^f doivent être reliés par une trajectoire passant les deux points intermédiaires q^1 et q^2. Cette trajectoire cubique (dans l'espace des articulations) est donnée par :

$$q(t) = a_0 + a_1.t + a_2.t^2 + a_3.t^3 \text{ avec } 0 \le t \le T \quad \text{(A.41)}$$

Où T est le temps nécessaire pour parcourir cette trajectoire.

Puisqu'on a quatre points on a besoin de trois trajectoires cubiques, l'une $q_1(t)$ pour aller du point de départ q_0 au point q_1 ; la seconde, $q_2(t)$ pour aller de q_1 au point q_2 , et la dernière trajectoire $q_3(t)$ pour aller de q_2 au point q_f :

Première trajectoire

$$\begin{cases} q_1(t) = a_{10} + a_{11}.t + a_{12}.t^2 + a_{13}.t^3 \\ \dot{q}_1(t) = a_{11} + 2.a_{12}.t + 3.a_{13}.t^2 \\ \ddot{q}_1(t) = 2.a_{12} + 6.a_{13}.t \end{cases} \quad \text{(A.42)}$$

Avec $0 \leq t \leq t1$ où t1 est le temps nécessaire pour aller de q_0 à q_1

Seconde trajectoire

$$\begin{cases} q_2(t) = a_{20} + a_{21}.t + a_{22}.t^2 + a_{23}.t^3 \\ \dot{q}_2(t) = a_{21} + 2.a_{22}.t + 3.a_{23}.t^2 \\ \ddot{q}_2(t) = 2.a_{22} + 6.a_{23}.t \end{cases} \quad (A.43)$$

Avec $0 \leq t \leq t2$ où t2 est le temps nécessaire pour aller de q_1 à q_2

Troisième trajectoire

$$\begin{cases} q_3(t) = a_{30} + a_{31}.t + a_{32}.t^2 + a_{33}.t^3 \\ \dot{q}_3(t) = a_{31} + 2.a_{32}.t + 3.a_{33}.t^2 \\ \ddot{q}_3(t) = 2.a_{32} + 6.a_{33}.t \end{cases} \quad (A.44)$$

Avec $0 \leq t \leq t3$ où t3 est le temps nécessaire pour aller de q_2 à q_f

Le temps total pour aller du point de départ au point final est alors :

$$t_{traj} = t_1 + t_2 + t_3 \quad (A.45)$$

Puisqu'on connaît la vitesse initiale v_0 qui est la vitesse au point q_0 et la vitesse finale v_f qui est la vitesse au point q_f et en assurant la continuité de vitesse et celle de l'accélération aux points intermédiaires, on peut calculer les coefficients des équations des trajectoires. En effet :

$$q_1(0) = a_{10} \Rightarrow a_{10} = q^0$$
$$q_1(t_1) = q_2(0) \Rightarrow a_{10} + a_{11}\,t_1 + a_{12}\,t_1^2 + a_{13}\,t_1^3 = q^1$$
$$\Rightarrow a_{11}\,t_1 + a_{12}\,t_1^2 + a_{13}\,t_1^3 = q^1 - q^0.$$
$$\dot{q}_1(t_1) = \dot{q}_2(0) \Rightarrow a_{11} + 2.a_{12}\,t_1 + 3.a_{13}\,t_1^2 = 0$$
$$\ddot{q}_1(t_1) = \ddot{q}_2(0) \Rightarrow a_{11} + 3.a_{12}\,t_1 - .a_{22} = 0$$

$$q_2(0) = a_{20} \Rightarrow a_{20} = q^1.$$
$$q_2(t_2) = q_3(0) \Rightarrow a_{20} + a_{21}\,t_2 + a_{22}\,t_2^2 + a_{23}\,t_2^3 = q^2$$
$$\Rightarrow a_{21}\,t_2 + a_{22}\,t_2^2 + a_{23}\,t_2^3 = q^2 - q^1.$$
$$\dot{q}_2(t_2) = \dot{q}_3(0) \Rightarrow a_{21} + 2.a_{22}\,t_2 + 3.a_{23}\,t_2^3 = 0$$
$$\ddot{q}_2(t_2) = \ddot{q}_3(0) \Rightarrow a_{21} + 3.a_{22}\,t_2 - a_{32} = 0$$

$$q_3(0) = a_{30} \Rightarrow a_{30} = q^2$$
$$q_3(t_3) = q^f \Rightarrow a_{30} + a_{31}\,t_3 + a_{32}\,t_3^2 + a_{33}\,t_3^3 = q^f$$
$$\Rightarrow a_{31}\,t_3 + a_{32}\,t_3^2 + a_{33}\,t_3^3 = q^f - q^2 \qquad \text{(A.46)}$$

$$\dot{q}_3(t_3) = v^f \Rightarrow a_{31} + 2.a_{32}\,t_3 - 3.a_{33}\,t_3^2 = v^f$$
$$\dot{q}_1(0) = v^i \Rightarrow a_{11} = v^i$$

Soit sous forme matricielle on a :

$$\begin{bmatrix} v_0 \\ q_1 - q_0 \\ 0 \\ 0 \\ q_2 - q_1 \\ 0 \\ 0 \\ 0 \\ q_f - q_2 \\ v_f \end{bmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ t_1 & t_1^2 & t_1^3 & & & & & & 0 \\ 1 & 2.t_1 & 3.t_1^2 & -1 & & & & & 0 \\ 0 & 1 & 3.t_1 & 0 & -1 & & & & 0 \\ 0 & & 0 & t_2 & t_2^2 & t_2^3 & & & 0 \\ 0 & & & 1 & 2.t_2 & 3.t_2^2 & -1 & & 0 \\ 0 & & & & 1 & 3.t_2 & 0 & -1 & 0 \\ 0 & & & & & 0 & t_3 & t_3^2 & t_3^3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2.t_3 & 3.t_3^2 \end{pmatrix} * \begin{bmatrix} a_{11} \\ a_{12} \\ a_{13} \\ a_{21} \\ a_{22} \\ a_{23} \\ a_{31} \\ a_{32} \\ a_{33} \end{bmatrix} \qquad \text{(A.47)}$$

A.5.3 Trajectoires cubiques avec plusieurs points intermédiaires

On peut généraliser les formules utilisées pour le problème de deux points intermédiaires. Supposons que l'on a f trajectoires à générer (c'est-à-dire on a f-1 points intermédiaires). L'équation de la i-ème trajectoire $q_i(t)$ qui doit aller du point q_{i-1} au point q_i est donnée par :

$$qi(t) = a_{i0} + a_{i1}.t + a_{i2}.t^2 + a_{i3}.t^3 \qquad i=1 \ldots f ; \qquad (A.48)$$

Avec $0 \leq t \leq ti$ où ti est le temps nécessaire pour parcourir cette trajectoire.
Et i allant de 1 jusqu'à f.

Alors, le temps total pour aller du point de départ au point final est :

$$t_{traj} = \sum_{i=1}^{i=f} ti \qquad (A.49)$$

On peut généraliser les équations des trajectoires ainsi que les dérivées primaires et secondaires de la façon suivante :

Pour i entre 1 et (f-1)

$$\begin{cases} q_i(0) = a_{i0} \Rightarrow a_{i0} = q^{i} \\ q_i(t_i) = q_{i+1}(0) \Rightarrow a_{i1} t_i + a_{i2} t_i^2 + a_{i3} t_i^3 = q^{i} - q^{i-1} \\ \dot{q}_i(t_i) = \dot{q}_{i+1}(0) \Rightarrow a_{i1} + 2.a_{i2} t_i + 3.a_{i3} t_i^2 = 0 \\ \ddot{q}_i(t_i) = \ddot{q}_{i+1}(0) \Rightarrow a_{i1} + 3.a_{i2} t_i - a_{(i+1)2} = 0 \end{cases} \qquad (A.50)$$

Pour la dernière trajectoire

$$q_f(t_f) = q^{f} \Rightarrow a_{f1} t_f + a_{f2} t_f^2 + a_{f3} t_f^3 = q^{f} - q^{f-1} \qquad (A.51)$$

En plus, les vitesses initiale et finale sont données par :

$$\dot{q}_f(t_f) = v^f \Rightarrow a_{f1} + 2.a_{f2}\,t_f + 3.a_{f3}\,t_f^2 = v^f \qquad \text{(A.52)}$$
$$\dot{q}_1(0) = v^1 \Rightarrow a_{11} = v^1$$

$$\begin{bmatrix} v_0 \\ q^1 - q^1 \\ 0 \\ 0 \\ \\ q^i - q^{i-1} \\ 0 \\ \dots \\ \dots \\ \dots \\ \dots \\ \dots \\ q^f - q^{f-1} \\ v^f \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & & & & & & & & & \\ t_1 & t_1^2 & t_1^3 & 0 & 0 & \dots & \dots & & \dots & & \dots & & \dots & \dots \\ 1 & 2t_1 & 3.t_1^2 & -1 & 0 & & & & & & & & & \\ 0 & 1 & 3t_1 & 0 & -1 & \dots & & & & & & & \dots & \\ & & & & & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \\ & & & & & \dots & \dots & ti & t_1^2 & t_1^2 & 0 & 0 & \dots & \dots & \dots \\ & & & & & \dots & \dots & 1 & 2ti & 3.t_1^2 & -1 & 0 & \dots & & \\ & & & & & \dots & \dots & 0 & 1 & 3ti & 0 & -1 & \dots & & \\ & & & & & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & & \\ & & & & & \dots & & & & & \dots & & & & \\ & & & & & \dots & & & & & & & & & \\ & & & & & \dots & \dots & & & \dots & \dots & \dots & \dots & \dots & \dots \\ & & & & & \dots & & & & \dots & \dots & 0 & 1 & 3tf & 0 & -1 \\ & & & & & \dots & \dots & & & \dots & \dots & 0 & 0 & tf & t_f^2 & t_f^2 \\ & & & & & \dots & & & & \dots & \dots & 0 & 0 & 1 & 2tf & 3.t_f^2 \end{bmatrix} * \begin{bmatrix} a_{11} \\ a_{12} \\ a_{13} \\ a_{21} \\ \\ a_1 \\ a_2 \\ a_3 \\ \\ \\ \\ \\ a_n \\ a_2 \\ a_3 \end{bmatrix}$$

(A.53)

Sous forme matricielle, on peut écrire que $\mathbf{B} = \mathbf{A}.\mathbf{X}$:
Où
X est le vecteur des coefficients des trajectoires de dimension 3f x 1.
A est une matrice de dimension 3f x 3f
B est un vecteur de données initiales de dimension 3fx1

Donc $\mathbf{X} = \mathbf{A}^{-1}.\mathbf{B}$.

ANNEXE 2

Programmes développés en assembleur

Dans ce qui suit on donne le code intégré dans le PIC (16f877), pour la détection de position et de sens.

```
2805    Goto    0x5     Initialisation du PIC et configuration des ports
3FFF    Addlw   0xff
3FFF    Addlw   0xff
3FFF    Addlw   0xff
2830    Goto    0x30    Branchement au programme principal
1683    Bsf     0x3,0x5
1303    Bcf     0x3,0x6
1606    Bsf     0x6,0x4
1586    Bsf     0x6,0x3
158B    Bsf     0xB,0x3
160B    Bsf     0xB,0x4
170B    Bsf     0xB,0x6
178B    Bsf     0xB,0x7
1283    Bcf     0x3,0x5
1303    Bcf     0x3,0x6
886     Movf    0x6
1683    Bsf     0x3,0x5
1303    Bcf     0x3,0x6
3000    Moviw   0x0
88      Movwf   0x8
1701    Bsf     0x1,0x6
1683    Bsf     0x3,0x5
3007    Moviw   0x7     Test de la broche génératrice d'interruption
009F    Movwf   0x1F
1085    Bcf     0x5,0x1
1105    Bcf     0x5,0x2
1605    Bsf     0x5,0x4
1285    Bcf     0x5,0x5
170B    Bsf     0xB,0x6
140C    Bsf     0xC,0x0
170B    Bsf     0xB,0x6
1283    Bcf     0x3,0x5
3067    Moviw   0x67
008E    Movwf   0xE
30C5    Moviw   0xC5
088F    Movf    0xF
3009    Moviw   0x9
90      Movwf   0x10
1683    Bsf     0x3,0x5
1787    Bsf     0x7,0x7
1283    Bcf     0x3,0x5
0       Nop
1283    Bcf     0x3,0x5
1F87    Btfss   0x7,0x7
207F    Call    0x7F    Réinisialisation du  Microcontrôleur
0       Nop
```

```
0       Nop
2829    Goto    0x29
1283    Bcf     0x3,0x5
1303    Bcf     0x3,0x6
1C0C    Btfss   0xC,0x0
284E    Goto    0x4E
100C    Bcf     0xC,0x0
3067    Movlw   0x67
008E    Movwf   0xE
3085    Movlw   0x85
088F    Movf    0xF
820     Movf    0x20,W
88      Movwf   0x8
0       Nop
0       nop
1685    bsf     0x5,0x5
0       nop
0       nop
100E    btfss   0x5,0x4
2840    goto    0x40
828     movf    0x28,W
88      movwf   0x8
0       nop
0       nop
1A05    btfsc   0x5,0x4
2846    goto    0x46
835     movf    0x25,W
88      movwf   0x8
0       nop
0       nop
1285    bcf     0x5,0x5
9       retfie
1283    bcf     0x3,0x5
1303    bcf     0x3,0x6
886     movf    0x6
1C0B    btfss   0xB,0x0
285C    goto    0x5C
100B    bcf     0xB,0x0
100E    btfss   0x6,0x4
2859    goto    0x56
1806    btfsc   0x6,0x0
2871    goto    0x71
286B    goto    0x6B
1C06    btfss   0x6,0x0
2871    goto    0x71
286B    goto    0x6F
108B    bcf     0xB,0x1
1683    bsf     0x3,0x5
1F01    btfss   0x1,0x6
2865    goto    0x65
1301    bcf     0x1,0x6
```

Branchement à la sous routine d'incrémentation de position

Test sur le bit de débordement, et branchement à la sous routine d'incrémentation du deuxième octet de la position

Branchement à la sous routine de décrémentation

```
1283      bcf     0x3,0x5
1,00E+06  btfss   0x6,0x4
2871      goto    0x71
286B      goto    0x6B      Sous routine de transfert de la position vers le pic 1
1683      bsf     0x3,0x5
1701      bsf     0x1,0x6
1283      bcf     0x3,0x5
1,00E+06  btfss   0x6,0x4
286B      goto    0x6B
2871      goto    0x71
0FA0      incfsz  0x20
9         retfie
0FA8      incfsz  0x28      Sous routine d'incrémentation
9         retfie
0AB5      incf    0x35
9         retfie
03A0      decf    0x20
820       movf    0x20,W
00C4      movwf   0x44
09C4      comf    0x44
1D03      btfss   0x3,0x2
9         retfie
03A8      decf    0x28      Sous routine de décrémentation
828       movf    0x28,W
00C4      movwf   0x44
09C4      comf    0x44
1D03      btfss   0x3,0x2
9         retfie
03B5      decf    0x35
9         retfie
1283      bcf     0x3,0x5
01A0      clrf    0x20
01A8      clrf    0x28
01B5      clrf    0x35
8         return
```

Le programme suivant, intégré dans le deuxième Pic, gère la communication série avec le DSP, la communication parallèle avec l'autre microcontrôleur, la génération du signal de commande PWM et l'asservissement du couple.

183	Clrf	0x3	Initialisation des interruptions
3001	Movlw	0x1	
008A	Movwf	0xA	
290C	Goto	0x10C	
00FC	Movwf	0x7C	
803	Movf	0x3,W	
183	Clrf	0x3	
00C8	movwf	0x48	
804	movf	0x4,W	
00C9	movwf	0x49	
080A	movf	0xA,W	
00CA	movwf	0x4A	
018A	clrf	0xA	
870	movf	0x70,W	
00D8	movwf	0x58	
871	movf	0x71,W	
00D9	movwf	0x59	
872	movf	0x72,W	
00CB	movwf	0x4B	
873	movf	0x73,W	
00CC	movwf	0x4C	
874	movf	0x74,W	
00CD	movwf	0x4D	
875	movf	0x75,W	Configuration des ports E/S
00CE	movwf	0x4E	
876	movf	0x76,W	
00CF	movwf	0x4F	
877	movf	0x77,W	
00D0	movwf	0x50	
878	movf	0x78,W	
00D1	movwf	0x51	
879	movf	0x79,W	
00D2	movwf	0x52	
087A	movf	0x7A,W	
00D3	movwf	0x53	
087B	movf	0x7B,W	
00D4	movwf	0x54	
856	movf	0x56,W	
00D5	movwf	0x55	
1C8B	btfss	0xB,0x1	
28DF	goto	0xDF	
108B	bcf	0xB,0x1	
1283	bcf	0x3,0x5	
1303	bcf	0x3,0x6	
830	movf	0x30,W	

```
00B4    movwf   0x34
831     movf    0x31,W
00B5    movwf   0x35
832     movf    0x32,W
00B6    movwf   0x36
833     movf    0x33,W
00B7    movwf   0x37
821     movf    0x21,W
00A4    movwf   0x24
822     movf    0x22,W
00A5    movwf   0x25
823     movf    0x23,W
00A6    movwf   0x26
808     movf    0x8,W
00A1    movwf   0x21
1506    bsf     0x6,0x2
01A9    clrf    0x29
01AA    clrf    0x2A
3000    movlw   0x0
022A    subwf   0x2A,W
3005    movlw   0x5
1903    btfsc   0x3,0x2
229     subwf   0x29,W
1803    btfsc   0x3,0x0
284A    goto    0x4A
0AA9    incf    0x29
1903    btfsc   0x3,0x2
0AAA    incf    0x2A
283F    goto    0x3F
808     movf    0x8,W
00A2    movwf   0x22
1106    bcf     0x6,0x2
01A9    clrf    0x29
01AA    clrf    0x2A
3000    movlw   0x0
022A    subwf   0x2A,W
3005    movlw   0x5
1903    btfsc   0x3,0x2
229     subwf   0x29,W
1803    btfsc   0x3,0x0
285A    goto    0x5A
0AA9    incf    0x29
1903    btfsc   0x3,0x2
0AAA    incf    0x2A
284F    goto    0x4F
808     movf    0x8,W
00A3    movwf   0x23
01A9    clrf    0x29
01AA    clrf    0x2A
3000    movlw   0x0
022A    subwf   0x2A,W
```

Configuration du module de communication SPI

Configuration du générateur de signal PWM

Configuration du convertisseur analogique numérique

```
3005    movlw   0x5
1903    btfsc   0x3,0x2
229     subwf   0x29,W
1803    btfsc   0x3,0x0
2869    goto    0x69
0AA9    incf    0x29
1903    btfsc   0x3,0x2
0AAA    incf    0x2A
285E    goto    0x5E
821     movf    0x21,W
00B8    movwf   0x38
01B9    clrf    0x39
01BA    clrf    0x3A
01BB    clrf    0x3B
822     movf    0x22,W
00BC    movwf   0x3C
01BD    clrf    0x3D
01BE    clrf    0x3E
01BF    clrf    0x3F
823     movf    0x23,W
00C0    movwf   0x40
01C1    clrf    0x41
01C2    clrf    0x42
01C3    clrf    0x43
00F0    movwf   0x70
841     movf    0x41,W
00F1    movwf   0x71
842     movf    0x42,W
00F2    movwf   0x72
843     movf    0x43,W
00F3    movwf   0x73
3010    movlw   0x10
120A    bcf     0xA,0x4
118A    bcf     0xA,0x3
2644    call    0x644
870     movf    0x70,W
00C4    movwf   0x44
871     movf    0x71,W
00C5    movwf   0x45
872     movf    0x72,W
00C6    movwf   0x46
873     movf    0x73,W
00C7    movwf   0x47
083C    movf    0x3C,W
00F0    movwf   0x70
083D    movf    0x3D,W
00F1    movwf   0x71
083E    movf    0x3E,W
00F2    movwf   0x72
083F    movf    0x3F,W
00F3    movwf   0x73
```

Programme principal

Partie de communication SPI avec le DSP

```
3008    movlw   0x8
120A    bcf     0xA,0x4
118A    bcf     0xA,0x3
2644    call    0x644
120A    bcf     0xA,0x4
118A    bcf     0xA,0x3
844     movf    0x44,W
04F0    iorwf   0x70
845     movf    0x45,W
04F1    iorwf   0x71
846     movf    0x46,W
04F2    iorwf   0x72
847     movf    0x47,W
04F3    iorwf   0x73
838     movf    0x38,W
470     iorwf   0x70,W
00B0    movwf   0x30
839     movf    0x39,W
471     iorwf   0x71,W
00B1    movwf   0x31
083A    movf    0x3A,W
472     iorwf   0x72,W
00B2    movwf   0x32
083B    movf    0x3B,W
473     iorwf   0x73,W
00B3    movwf   0x33
3003    movlw   0x3
00F4    movwf   0x74
01F5    clrf    0x75
01F6    clrf    0x76
01F7    clrf    0x77
834     movf    0x34,W
230     subwf   0x30,W
00F0    movwf   0x70
831     movf    0x31,W
00F1    movwf   0x71
835     movf    0x35,W
1C03    btfss   0x3,0x0
0F35    incfsz  0x35,W
02F1    subwf   0x71
832     movf    0x32,W
00F2    movwf   0x72
836     movf    0x36,W
1C03    btfss   0x3,0x0
0F36    incfsz  0x36,W
02F2    subwf   0x72
833     movf    0x33,W
00F3    movwf   0x73
837     movf    0x37,W
1C03    btfss   0x3,0x0
0F37    incfsz  0x37,W
```

```
02F3      subwf   0x73
120A      bcf     0xA,0x4
118A      bcf     0xA,0x3
2747      call    0x747
3,00E+09  movlw   0xE8
00F4      movwf   0x74
3003      movlw   0x3
00F5      movwf   0x75
3000      movlw   0x0
01F6      clrf    0x76
01F7      clrf    0x77
120A      bcf     0xA,0x4
118A      bcf     0xA,0x3
27C3      call    0x7C3
120A      bcf     0xA,0x4
118A      bcf     0xA,0x3
26ED      call    0x6ED
120A      bcf     0xA,0x4
118A      bcf     0xA,0x3
870       movf    0x70,W
00AD      movwf   0x2D
871       movf    0x71,W
00AE      movwf   0x2E
872       movf    0x72,W
00AF      movwf   0x2F
1283      bcf     0x3,0x5
1303      bcf     0x3,0x6
1D8C      btfss   0xC,0x3
2,80E+10  goto    0xE9
813       movf    0x13,W
00A0      movwf   0x20
118C      bcf     0xC,0x3
01A7      clrf    0x27
0AA7      incf    0x27
01A8      clrf    0x28
858       movf    0x58,W
00F0      movwf   0x70
859       movf    0x59,W
00F1      movwf   0x71
084B      movf    0x4B,W
00F2      movwf   0x72
084C      movf    0x4C,W
00F3      movwf   0x73
084D      movf    0x4D,W
00F4      movwf   0x74
084E      movf    0x4E,W
00F5      movwf   0x75
084F      movf    0x4F,W
00F6      movwf   0x76
850       movf    0x50,W
00F7      movwf   0x77
```

Sous routine de communication avec le pic d'acquisition de position

```
851     movf    0x51,W
00F8    movwf   0x78
852     movf    0x52,W
00F9    movwf   0x79
853     movf    0x53,W
00FA    movwf   0x7A
854     movf    0x54,W
00FB    movwf   0x7B
855     movf    0x55,W
00D6    movwf   0x56
849     movf    0x49,W
84      movwf   0x4
084A    movf    0x4A,W
008A    movwf   0xA
848     movf    0x48,W
83      movwf   0x3
0EFC    swapf   0x7C
0E7C    swapf   0x7C,W
9       retfie
3020    moviw   0x20
84      movwf   0x4
3048    moviw   0x48
2117    call    0x117
183     clrf    0x3
120A    bcf     0xA,0x4
118A    bcf     0xA 0x3
2E4F    goto    0x64F
604     xorwf   0x4,W
180     clrf    0x0
0A84    incf    0x4
604     xorwf   0x4,W
1D03    btfss   0x3,0x2
2914    goto    0x114
3400    retlw   0x0
0F4     movwf   0x74
0AF4    incf    0x74
03F4    decf    0x74
1903    btfsc   0x3,0x2
3400    retlw   0x0
1003    bcf     0x3,0x0
0DF0    rlf     0x70
0DF1    rlf     0x71
0DF2    rlf     0x72
0DF3    rlf     0x73
200E    goto    0x646
1683    bsf     0x3,0x5
1303    bcf     0x3,0x6
1406    bsf     0x6,0x0
1586    bsf     0x6,0x3
1106    bcf     0x6,0x2
30FF    moviw   0xFF
```

Sous routine de convertion de courant

```
88      movwf   0x8
1701    bsf     0x1,0x6
178B    bsf     0xB,0x7
160B    bsf     0xB,0x4
300C    movlw   0xC
1283    bcf     0x3,0x5
97      movwf   0x17
30FF    movlw   0xFF
1683    bsf     0x3,0x5
92      movwf   0x12
1107    bcf     0x7,0x2
1283    bcf     0x3,0x5
1092    bcf     0x12,0x1
1012    bcf     0x12,0x0
1512    bsf     0x12,0x2
1294    bcf     0x14,0x5
3004    movlw   0x4
94      movwf   0x14
1694    bsf     0x14,0x5
170B    bsf     0xB,0x6
1683    bsf     0x3,0x5
158C    bsf     0xC,0x3
3040    movlw   0x40
94      movwf   0x14
1287    bcf     0x7,0x5
1587    bsf     0x7,0x3
1607    bsf     0x7,0x4
3004    movlw   0x4
009F    movwf   0x1F
1685    bsf     0x5,0x5
1283    bcf     0x3,0x5
01A7    clrf    0x27
0AA7    incf    0x27
01A8    clrf    0x28
320     decf    0x20,W
3000    movlw   0x0
1903    btfsc   0x3,0x2
3001    movlw   0x1
00F0    movwf   0x70
01F1    clrf    0x71
827     movf    0x27,W
05F0    andwf   0x70
828     movf    0x28,W
05F1    andwf   0x71
871     movf    0x71,W
470     iorwf   0x70,W
1903    btfsc   0x3,0x2
2E8B    goto    0x68B
821     movf    0x21,W
93      movwf   0x13
01A7    clrf    0x27
```

Sous routine de convertion couple en un signal PWM

```
01A8    clrf    0x28
01AB    clrf    0x2B
01AC    clrf    0x2C
01A7    clrf    0x27
0AA7    incf    0x27
01A8    clrf    0x28
820     movf    0x20,W
3A02    xorlw   0x2
3000    movlw   0x0
1903    btfsc   0x3,0x2
3001    movlw   0x1
00F0    movwf   0x70
01F1    clrf    0x71
827     movf    0x27,W
05F0    andwf   0x70
828     movf    0x28,W
05F1    andwf   0x71
871     movf    0x71,W
470     iorwf   0x70,W
1903    btfsc   0x3,0x2
2EA3    goto    0x6A3
822     movf    0x22,W
93      movwf   0x13
01A7    clrf    0x27
01A8    clrf    0x28
01AB    clrf    0x2B
01AC    clrf    0x2C
01A7    clrf    0x27
0AA7    incf    0x27
01A8    clrf    0x28
820     movf    0x20,W
3A03    xorlw   0x3
3000    movlw   0x0
1903    btfsc   0x3,0x2
3001    movlw   0x1
00F0    movwf   0x70
01F1    clrf    0x71
827     movf    0x27,W
05F0    andwf   0x70
828     movf    0x28,W
05F1    andwf   0x71
871     movf    0x71,W
470     iorwf   0x70,W
1903    btfsc   0x3,0x2
2EBC    goto    0x6BC
823     movf    0x23,W
93      movwf   0x13
01A7    clrf    0x27
01A8    clrf    0x28
01AB    clrf    0x2B
0AAB    incf    0x2B
```

01AC	clrf	0x2C
032B	decf	0x2B,W
042C	iorwf	0x2C,W
1D03	btfss	0x3,0x2
2ECC	goto	0x6CC
327	decf	0x27,W
428	iorwf	0x28,W
1D03	btfss	0x3,0x2
2ECC	goto	0x6CC
820	movf	0x20,W
95	movwf	0x15
01A7	clrf	0x27
01A8	clrf	0x28
01AB	clrf	0x2B
01AC	clrf	0x2C
823	movf	0x23,W
93	movwf	0x13
1C06	btfss	0x6,0x0
2,00E+73	goto	0x673
1506	bsf	0x6,0x2
1506	bsf	0x6,0x2
1106	bcf	0x6,0x2
2,00E+73	goto	0x673
884	movf	0x4
1903	btfsc	0x3,0x2
3400	retlw	0x0
872	movf	0x72,W
80	movwf	0x0
384	decf	0x4
871	movf	0x71,W
80	movwf	0x0
384	decf	0x4
870	movf	0x70,W
80	movwf	0x0
3400	retlw	0x0
1FF3	btfss	0x73,0x7
2EED	goto	0x6ED
1478	bsf	0x78,0x0
09F0	comf	0x70
09F1	comf	0x71
09F2	comf	0x72
09F3	comf	0x73
0AF0	incf	0x70
1903	btfsc	0x3,0x2
0AF1	incf	0x71
1903	btfsc	0x3,0x2
0AF2	incf	0x72
1903	btfsc	0x3,0x2
0AF3	incf	0x73
2EEE	goto	0x6EE
01F8	clrf	0x78

```
308E    movlw   0x8E
00F6    movwf   0x76
184     clrf    0x4
08F3    movf    0x73
1903    btfsc   0x3,0x2
2F25    goto    0x725
1003    bcf     0x3,0x0
0CF3    rrf     0x73
0CF2    rrf     0x72
0CF1    rrf     0x71
0CF0    rrf     0x70
0AF6    incf    0x76
2EF1    goto    0x6F1
01F8    clrf    0x78
0D71    rlf     0x71,W
0D72    rlf     0x72,W
00F6    movwf   0x76
3800    iorlw   0x0
1903    btfsc   0x3,0x2
2F06    goto    0x706
0DF8    rlf     0x78
01F2    clrf    0x72
17F1    bsf     0x71,0x7
8       return
01F0    clrf    0x70
01F1    clrf    0x71
01F2    clrf    0x72
3400    retlw   0x0
10F8    bcf     0x78,0x1
0D74    rlf     0x74,W
0D75    rlf     0x75,W
00F7    movwf   0x77
3800    iorlw   0x0
1903    btfsc   0x3,0x2
2F16    goto    0x716
1803    btfsc   0x3,0x0
14F8    bsf     0x78,0x1
01F5    clrf    0x75
17F4    bsf     0x74,0x7
8       return
01F7    clrf    0x77
01F3    clrf    0x73
01F4    clrf    0x74
01F5    clrf    0x75
8       return
876     movf    0x76,W
1903    btfsc   0x3,0x2
2F06    goto    0x706
13F1    bcf     0x71,0x7
0CF8    rrf     0x78
0C76    rrf     0x76,W
```

```
00F2    movwf   0x72
1803    btfsc   0x3,0x0
17F1    bsf     0x71,0x7
2ED2    goto    0x6D2
08F6    movf    0x76
1D03    btfss   0x3,0x2
2F39    goto    0x739
01F0    clrf    0x70
01F1    clrf    0x71
01F2    clrf    0x72
2ED2    goto    0x6D2
372     decf    0x72,W
1903    btfsc   0x3,0x2
0AF0    incf    0x70
1903    btfsc   0x3,0x2
0AF1    incf    0x71
1903    btfsc   0x3,0x2
0AF2    incf    0x72
1003    bcf     0x3,0x0
0CF2    rrf     0x72
0CF1    rrf     0x71
0CF0    rrf     0x70
0AF6    incf    0x76
2F25    goto    0x725
08F2    movf    0x72
1D03    btfss   0x3,0x2
2F2C    goto    0x72C
870     movf    0x70,W
471     iorwf   0x71,W
1903    btfsc   0x3,0x2
2F28    goto    0x728
1BF1    btfsc   0x71,0x7
2F1B    goto    0x71B
03F6    decf    0x76
1003    bcf     0x3,0x0
0DF0    rlf     0x70
0DF1    rlf     0x71
2F40    goto    0x740
184     clrf    0x4
1283    bcf     0x3,0x5
1303    bcf     0x3,0x6
01D6    clrf    0x56
01F8    clrf    0x78
01F9    clrf    0x79
01FA    clrf    0x7A
01FB    clrf    0x7B
874     movf    0x74,W
475     iorwf   0x75,W
476     iorwf   0x76,W
477     iorwf   0x77,W
1903    btfsc   0x3,0x2
```

Transfert du rapport cyclique au module de génération du signal pwm

```
2F58    goto    0x758
30C0    movlw   0xC0
05D6    andwf   0x56
2F63    goto    0x763
01F0    clrf    0x70
01F1    clrf    0x71
01F2    clrf    0x72
01F3    clrf    0x73
3400    retlw   0x0
1003    bcf     0x3,0x0
0DF4    rlf     0x74
0DF5    rlf     0x75
0DF6    rlf     0x76
0DF7    rlf     0x77
0AD6    incf    0x56
1FF7    btfss   0x77,0x7
2F5D    goto    0x75D
0AD6    incf    0x56
1003    bcf     0x3,0x0
0DF8    rlf     0x78
0DF9    rlf     0x79
0DFA    rlf     0x7A
0DFB    rlf     0x7B
877     movf    0x77,W
273     subwf   0x73,W
1D03    btfss   0x3,0x2
2F79    goto    0x779
876     movf    0x76,W
272     subwf   0x72,W
1D03    btfss   0x3,0x2
2F79    goto    0x779
875     movf    0x75,W
271     subwf   0x71,W
1D03    btfss   0x3,0x2
2F79    goto    0x779
874     movf    0x74,W
270     subwf   0x70,W
1C03    btfss   0x3,0x0
2F8A    goto    0x78A
1478    bsf     0x78,0x0
874     movf    0x74,W
02F0    subwf   0x70
875     movf    0x75,W
1C03    btfss   0x3,0x0
0F75    incfsz  0x75,W
02F1    subwf   0x71
876     movf    0x76,W
1C03    btfss   0x3,0x0
0F76    incfsz  0x76,W
02F2    subwf   0x72
877     movf    0x77,W
```

Transfert de la position au DSP

```
1C03    btfss   0x3,0x0
0F77    incfsz  0x77,W
02F3    subwf   0x73
1003    bcf     0x3,0x0
0CF7    mf      0x77
0CF6    mf      0x76
0CF5    mf      0x75
0CF4    mf      0x74
03D6    decf    0x56
856     movf    0x56,W
393F    andlw   0x3F
1D03    btfss   0x3,0x2
2F66    goto    0x766
1F56    btfss   0x56,0x6
2FA3    goto    0x7A3
1003    bcf     0x3,0x0
30FF    movlw   0xFF
07F0    addwf   0x70
09F0    comf    0x70
1C03    btfss   0x3,0x0
07F1    addwf   0x71
09F1    comf    0x71
1C03    btfss   0x3,0x0
07F2    addwf   0x72
09F2    comf    0x72
1C03    btfss   0x3,0x0
07F3    addwf   0x73
09F3    comf    0x73
870     movf    0x70,W
00F4    movwf   0x74
871     movf    0x71,W
00F5    movwf   0x75
872     movf    0x72,W
00F6    movwf   0x76
873     movf    0x73,W
00F7    movwf   0x77
878     movf    0x78,W
00F0    movwf   0x70
879     movf    0x79,W
00F1    movwf   0x71
087A    movf    0x7A,W
00F2    movwf   0x72
087B    movf    0x7B,W
00F3    movwf   0x73
1FD6    btfss   0x56,0x7
2FF1    goto    0x7F1
1003    bcf     0x3,0x0
30FF    movlw   0xFF
07F0    addwf   0x70
09F0    comf    0x70
1C03    btfss   0x3,0x0
```

```
07F1    addwf   0x71
09F1    comf    0x71
1C03    btfss   0x3,0x0
07F2    addwf   0x72
09F2    comf    0x72
1C03    btfss   0x3,0x0
07F3    addwf   0x73
09F3    comf    0x73
2FF1    goto    0x7F1
184     clrf    0x4
870     movf    0x70,W
00F8    movwf   0x78
871     movf    0x71,W
00F9    movwf   0x79
872     movf    0x72,W
00FA    movwf   0x7A
873     movf    0x73,W
00FB    movwf   0x7B
01F2    clrf    0x72
01F3    clrf    0x73
01F0    clrf    0x70
01F1    clrf    0x71
1003    bcf     0x3,0x0
0CFB    rrf     0x7B
0CFA    rrf     0x7A
0CF9    rrf     0x79
0CF8    rrf     0x78
1C03    btfss   0x3,0x0
2FE5    goto    0x7E5
874     movf    0x74,W
07F0    addwf   0x70
875     movf    0x75,W
1803    btfsc   0x3,0x0
0F75    incfsz  0x75,W
07F1    addwf   0x71
876     movf    0x76,W
1803    btfsc   0x3,0x0
0F76    incfsz  0x76,W
07F2    addwf   0x72
1003    bcf     0x3,0x0
0DF4    rlf     0x74
0DF5    rlf     0x75
0DF6    rlf     0x76
0DF7    rlf     0x77
2FD0    goto    0x7D0
884     movf    0x4
1903    btfsc   0x3,0x2
3400    retlw   0x0
873     movf    0x73,W
80      movwf   0x0
384     decf    0x4
```

```
872     movf    0x72,W
80      movwf   0x0
384     decf    0x4
871     movf    0x71,W
80      movwf   0x0
384     decf    0x4
870     movf    0x70,W
80      movwf   0x0
3400    retlw   0x0
```

ANNEXE 3

Programme de communication I2C développé en C

Ici on donne le programme de communication I2C développé en langage C, pour commander le même système, en utilisant le PC comme maître et générateur de trajectoire.

```
#include<stdio.h>
#include<conio.h>
#include<dos.h>
const StatAdr =889,
      CtrlAdr =890,
      SDA_LOW =0xFE,
      SDA_HIGH =0x01,
      SDA_STAT =0x80,
      SCL_LOW =0xFD,
      SCL_HIGH =0x02,
      RD     =1,
      WR     =0,
      Tempo   =9000;
unsigned char Control;

unsigned char ReadStat(void)
{return(inportb(StatAdr) & 0xF8);
}
void  WriteCtrl(unsigned char control)
{control=control & 15;
 outportb(CtrlAdr,control);
 return;
}
int ReadSDA(void)
   {if ((ReadStat() & SDA_STAT)==SDA_STAT)
                          return(0);
                          else
                          return(1);
   }
void ATTENTE(int T)
{int i;
 for (i=1;i<=T;i++){}
 return;
}
void SDA(int Etat )
{ if (Etat)
    {Control = Control & SDA_LOW;
     WriteCtrl(Control);
    }
    else
    {Control = Control | SDA_HIGH;
```

```
    WriteCtrl(Control);
   }
   return;
}
void SCL(int Etat )
{ if (Etat)
   {Control = Control & SCL_LOW;
    WriteCtrl(Control);
   }
   else
   {Control = Control | SCL_HIGH;
    WriteCtrl(Control);
   }
   return;
}
void START(void)
{ SDA(1);
 SCL(1);
 ATTENTE(Tempo);
 SDA(0);
 ATTENTE(Tempo);
 SCL(0);
 ATTENTE(Tempo);
 return;
 }
void STOP(void)
{ SDA(0);
 ATTENTE(Tempo);
 SCL(1);
 ATTENTE(Tempo);
 SDA(1);
 return;
}
int ACKSL(void)
{ SDA(1);
 ATTENTE(Tempo);
 SCL(1);
 ATTENTE(Tempo);
 if(! ReadSDA()) {SCL(0);
  ATTENTE(Tempo);
  return(1);}
  else return(0);
 }
void ACKMA(void)
{ SDA(0);
 ATTENTE(Tempo);
 SCL(1);
```

```
 ATTENTE(Tempo);
 SCL(0);
 SDA(1);
 return;
 }
void NOACKMA(void)
{ SDA(0);
 ATTENTE(Tempo);
 SCL(1);
 return;
 }
 void Send(unsigned char D)
 {unsigned char TransmitData,i;
 TransmitData=D;
 for(i=1;i<=8;i++)
 {if(TransmitData>=128) SDA(1);
  else SDA(0);
  ATTENTE(Tempo);
  SCL(1);
  ATTENTE(Tempo);
  TransmitData=TransmitData<<1;
  SCL(0);
  ATTENTE(Tempo);
  }
  return;
 }
 unsigned char Receive(void)
 {unsigned char ReceiveData,i;
 ReceiveData=0;
 for(i=1;i<=8;i++)
 {ReceiveData=ReceiveData<<1;
  SCL(1);
  ATTENTE(Tempo);
  if(ReadSDA()) ReceiveData++;
  SCL(0);
  }
 return(ReceiveData);
 }
 void main(void)
 {int i;
 textbackground(0);
 clrscr();
 START();
 Send(0x02);
   if (! ACKSL()) printf("!!!!Erreur de transmission d'adresse!!!!");
 while(1)
 { Send(0x80);
```

```
  if (! ACKSL()) printf("!!!!Erreur de transmission de donn_e!!!!");}
STOP();
}
```

ANNEXE 4

Programme intégré dans le DSP

Dans cet annexe on donne le programme de communication coté DSP.

```
#include      "f2407_c.h"

int i,PWM,RECEPTION,SS,xx,xc1,xc2,xc3,v1,v2,v3,v4,EMI,rien,xd1,xd2,xd3;

long unsigned int   x1=0,x2=0,x3=0,POS=0,VIT=0,vt1,vt2,vt3,vt4,POSd=15000;

void main(void)
{


   *SCSR1 = 0x00FD;
   *SCSR2 = (*SCSR2 | 0x000B) & 0x000F;

   *WDCR  = 0x00E8;
    WSGR = 0x0040;


  *SPICCR = 0x0000;
  *SPICCR = 0x0007;
  *PCDATDIR = 0x1400;
  *MCRB = 0xFFFC;


  *SPICTL = 0x0006;
  *SPIBRR = 0x007f;
  *SPICCR = 0x0087;

  *MCRA = 0x0000;
  *PADATDIR = 0xff07;

   PWM = 0x0060;
   EMI = 1;
   SS = 0;
  *PADATDIR = 0xff00 + SS;
  *SPIDAT = EMI<<8;
    for ( i = 0; i < 125; i++ )
              {}
       *PADATDIR = 0xff07;
   rien = *SPIRXBUF;


 while(1)
    {
```

```
if(EMI== 1)
    { xx = EMI;

     *PADATDIR = 0xff00 + SS;
     *SPIDAT = xx<<8;
    for ( i = 0; i < 125; i++ )
            {}
      *PADATDIR = 0xff07;
while(!((*SPISTS & 0x0040) == 0x0040));
      rien = *SPIRXBUF;
      EMI = 2;

  for ( i = 0; i < 1000; i++ )
        {}
   }
   if(EMI== 2)
    { xx = EMI;

     *PADATDIR = 0xff00 + SS;
     *SPIDAT = xx<<8;
    for ( i = 0; i < 125; i++ )
            {}
      *PADATDIR = 0xff07;
while(!((*SPISTS & 0x0040) == 0x0040));
      xc1 = *SPIRXBUF;
      EMI = 3;

  for ( i = 0; i < 1000; i++ )
        {}
    }


     if(EMI== 3)
    { xx = EMI;

     *PADATDIR = 0xff00 + SS;
     *SPIDAT = xx<<8;
    for ( i = 0; i < 125; i++ )
            {}
      *PADATDIR = 0xff07;
while(!((*SPISTS & 0x0040) == 0x0040));
      xc2 = *SPIRXBUF;
      EMI = 4;

  for ( i = 0; i < 1000; i++ )
        {}
    }
```

```
    if(EMI== 4)
    { xx = EMI;

      *PADATDIR = 0xff00 + SS;
      *SPIDAT = xx<<8;
    for ( i = 0; i < 125; i++ )
            {}
       *PADATDIR = 0xff07;
while(!((*SPISTS & 0x0040) == 0x0040));
       xc3 = *SPIRXBUF;
       EMI = 5;

  for ( i = 0; i < 1000; i++ )
        {}
    }


   if(EMI== 5)
    { xx = EMI;

      *PADATDIR = 0xff00 + SS;
      *SPIDAT = xx<<8;
    for ( i = 0; i < 125; i++ )
            {}
       *PADATDIR = 0xff07;
while(!((*SPISTS & 0x0040) == 0x0040));
       v1 = *SPIRXBUF;
       EMI = 6;

  for ( i = 0; i < 1000; i++ )
        {}
    }


   if(EMI== 6)
    { xx = EMI;

      *PADATDIR = 0xff00 + SS;
      *SPIDAT = xx<<8;
    for ( i = 0; i < 125; i++ )
            {}
       *PADATDIR = 0xff07;
while(!((*SPISTS & 0x0040) == 0x0040));
       v2 = *SPIRXBUF;
       EMI = 7;

  for ( i = 0; i < 1000; i++ )
```

```
        {}
    }


  if(EMI== 7)
    { xx = EMI;

      *PADATDIR = 0xff00 + SS;
      *SPIDAT = xx<<8;
    for ( i = 0; i < 125; i++ )
            {}
       *PADATDIR = 0xff07;
while(!((*SPISTS & 0x0040) == 0x0040));
       v3 = *SPIRXBUF;
       EMI = 8;

  for ( i = 0; i < 1000; i++ )
        {}
    }


  if(EMI== 8)
    {
      *PADATDIR = 0xff00 + SS;
      *SPIDAT = xd1<<8;
    for ( i = 0; i < 125; i++ )
            {}
       *PADATDIR = 0xff07;
while(!((*SPISTS & 0x0040) == 0x0040));
       v4 = *SPIRXBUF;
       EMI = 9;

  for ( i = 0; i < 1000; i++ )
        {}
    }



  if(EMI== 9)
    {
      *PADATDIR = 0xff00 + SS;
      *SPIDAT = xd2<<8;
    for ( i = 0; i < 125; i++ )
            {}
       *PADATDIR = 0xff07;
while(!((*SPISTS & 0x0040) == 0x0040));
       rien = *SPIRXBUF;
```

```
            EMI = 10;

        for ( i = 0; i < 1000; i++ )
              {}
          }



         if(EMI== 10)
          {
           *PADATDIR = 0xff00 + SS;
           *SPIDAT = xd3<<8;
          for ( i = 0; i < 125; i++ )
                 {}
            *PADATDIR = 0xff07;
      while(!((*SPISTS & 0x0040) == 0x0040));
            rien = *SPIRXBUF;
            EMI = 1;

        for ( i = 0; i < 1000; i++ )
              {}
          }

         x1 = xc1;
         x2 = xc2;
         x3 = xc3;
         vt1 = v1;
         vt2 = v2;
         vt3 = v3;
         vt4 = v4;

         POS = (x1|x2<<8|x3<<16);
         VIT = (vt1|vt2<<8|vt3<<16|vt4<<24);


             xd1 = POSd & 0x000000ff;
        xd2 = (POSd & 0x0000ff00)>>8;
        xd3 = (POSd & 0x0000ff00)>>16;

    }

}
```

BIBLIOGRAPHIE

[1] Craig, J.J. (1986). Introduction to Robotics- Mecanics and Control. *Reading (Mass.) : Allison-Wesley*, 2e edition.

[2] Schiling, R. J. (1990). Fundamentals of Robotics – Analysis and Control. *Englewood Cliffs (N.J.) : Prentice-Hall.*

[3] Asada, H. et Slotine, J.-J.E. (1986). Robot Analysis and Control. *New York (N.Y.) : John Wiley & Sons.*

[4] Dessaint, L.-A, Saad, M., Hebert, B. et Al-Haddad, K. (1992). An Adaptive Controller for a Direct-Drive Scara Robot. *IEEE Transcations on industrial Electronics*. 17 (22), 105-111.

[5] Slotine, J.-J. E. et Li, W. (1991). Applied non Linear Control. *Englewood Cliffs (N.J.) : Prentice-Hall.*

[6] Saad, M., Dessaint, L. A., Bigras, P., Al-Haddad K. (1994). Adaptive Versus Neural Adaptive Control : Application to Robotics. *International Journal of Adaptative Control and Signal Processing*, 8, 223-236.

[7] Slotine, J.-J. E. et Li, W. (1987). Adaptative Manipulator Control : A Case Study. *IEEE Transcations on Automatic Control*, 33(11), pp. 995-1003

[8] Koivo, A. J. (1989). Fundamentals for Control of Robotic manipulators. *New York (N.Y.) : John Wiley & Sons.*

[9] Brady, M. (1982). Trajectory Planning. Robot Motion : Planning and Control. *M. Brady et al. Cambridge (Mass.) : MIT Press.*

[10] Taylor, R. H. (1979). Planning and Execution of Stright-Line Manipulator Trajectories. *IBM Journal of Research and Development*, 23, 424-436

[11] Whitney, D. E. (1972). The Mathematics of Coordinated Control of Prosthetics Arms and Manipulators. *Journal of Dynamic Systems, Measurement, and Control*, 303-309.

[12] Paul, R. P. C. (1979). Manipulator Cartesian Path Control. *IEEE Transactions on Systems, Man, and Cybernetics*, 9, 702-711.

[13] Slotine, J.-J. E. et Li, W. (1987). Adaptative Strategies in Constrained Manipulation. *IEEE International Conference on Robotics and Automation*, 595-601.

[14] Latombe, J.-C. (1991). Robot Motion Planning. *Boston (Mass.) : Kluwer Academic Publishers.*

[15] Chernousko, F. L., Bolotnik, N. et Gradetsky, V. (1994). Manipulation Robots : Dynamics, control and Optimization. *Boca Raton (Flor.) : CRC Press.*

[16] MacKerrow, P. J. (1995). Introduction to Robotics. *Reading (Mass.) : Allisson-Wesley.*

[17] Saad, Y. (1992). Numerical Methods for Large Eigenvalue Problems : Theory and Algorithms. *Manchester University Press.*

[18] Goldberg, D. S. (1991). *Matrix* Theory with Applications. *New York (N. Y.) : McGraw-Hill.*

[19] Watkins, D. S. (1991). Fundamentals of Matrix Computations. *New York (N. Y.) : John Wiley & Sons.*

[20] Press, W. H., Teukolsky, S. A., Vetterling, W. T., Flannery, B. P., Numerical Recipes in C. *New York (N. Y.) : Cambridge University Press.*

[21] Golub, G. et Ortega, J. M. (1993). Scientific Computing : An Introduction with Parallel Computing. *Sandiego (Calif.) : Academic press.*