



# ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

## **Babyfoot Stratégie**

Projet d'ingénierie simultanée

Semestre 6

Emile Décosterd et Valentin Mottier

Sous la responsabilité de :

Christophe Salzmann

Milan Korda

Colin Jones

## Table des matières :

1. Introduction :	4
2. Matériel :	5
2.1 Moteur linéaire Servotube et boîtier Xenus :	5
2.2 Moteur rotatif Maxon et boîtier ESCON 50/5 :	5
2.3 Alimentation :	5
3. Améliorations :	6
3.1 Câblage :	6
3.2 Améliorations Mécaniques :	8
4. Base de fonctionnement du module National Instrument sur Labview :	9
4.1 Initialisation :	9
4.2 Écriture/Lecture d'informations	10
4.3 Fermeture de la tâche :	10
5. Acquisition des données et premier tests :	11
6. VI d'identification pour la calibration des contrôleurs :	14
6.1 Contrôleur PID	14
6.2 Identification des coefficients pour le contrôleur	15
7. Propositions d'améliorations	18
8. Conclusion	19
Annexes :	20
Détails des connexions pour le boîtier de connexion SCB-68 :	20
Détails de connexion pour le boîtier de commande ESCON 50/5 :	21
Détails de connexion pour le boîtier de commande Xenus :	22
Construction du port0 de la carte d'acquisition :	23
Schéma de connexion du connecteur J7 du boîtier Xenus :	24



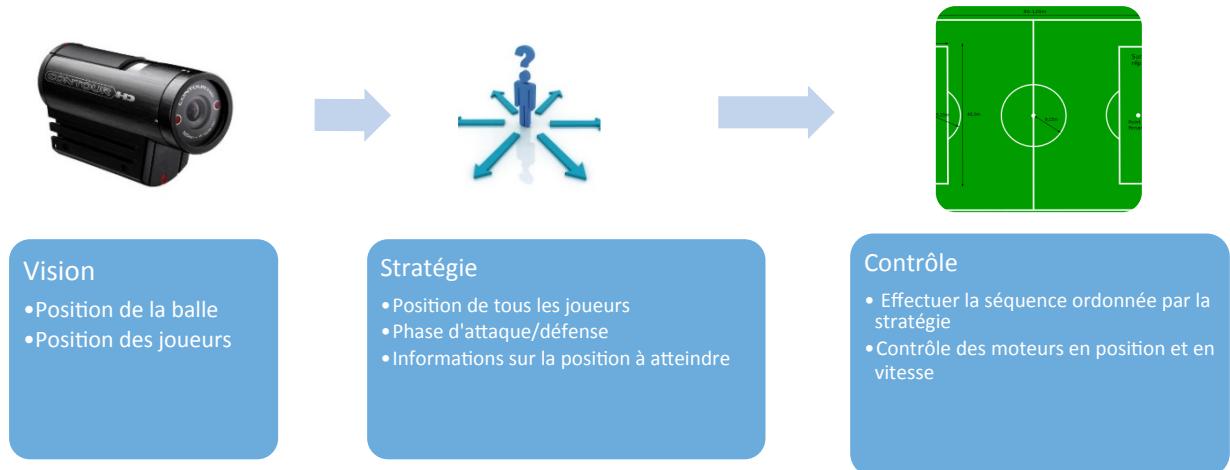
## 1. Introduction :

Le but de ce travail de Bachelor est de concevoir un babyfoot automatique qui une fois optimisé pourrait affronter un humain et idéalement le battre à plate couture. Ce n'est pas la première année que ce projet est proposé, c'est pour cela que plusieurs idées ont déjà été imaginées et réalisées. Cette année, nous avons hérité du projet de Monsieur Simon Métraller, qui lors de son projet était responsable de la mise en service d'un babyfoot basé sur un moteur linéaire pour la translation des joueurs et d'un moteur rotatif pour effectuer le contrôle de la balle. Le temps ne lui ayant pas permis de terminer, huit étudiants du Bachelor VI de la section génie mécanique prennent le relais dans le but d'améliorer son travail. Dans un premier temps nous avons procéder à un partage du travail qui consiste à répartir les tâches suivantes :

- La stratégie, c'est-à-dire l'imagination d'un algorithme d'intelligence artificielle permettant au système de pallier aux attaques du joueur adverse, ainsi que de créer des phases d'attaques.
- La vision, cette partie s'occupe d'une part de repérer la position de la balle sur le terrain à l'aide d'une caméra haute performance et aussi de déterminer la position des joueurs adverses sur le terrain à partir d'un système astucieux de lasers.
- La troisième partie, qui est le sujet de ce rapport, s'occupe de mettre ensemble toutes les informations fournies et finalement de mettre en mouvement le système en envoyant les commandes adéquates aux moteurs.

Chaque partie est effectuée par un groupe d'étudiants travaillant en parallèle, il était donc important de bien coordonner nos actions de manière à travailler dans une même direction.

Pour illustrer quelque peu la structure du projet, le schéma ci-après montre le flux d'informations et les challenges de chacun des groupes pour réaliser le babyfoot automatique.



Concernant la partie contrôle, nous avons quelque peu modifié les objectifs initiaux. En effet, avant de se préoccuper du contrôle des moteurs, nous avons décidé de nous concentrer sur quelques

améliorations plutôt pratiques à apporter à notre système qui ont pour but de faciliter la prise en main et la mise en service.

## 2. Matériel :

Le système existant se base sur une idée très simple qui se rapproche des mouvements effectués par l'humain jouant au babyfoot. Un moteur linéaire est utilisé pour déplacer les joueurs latéralement. Sur ce dernier est installé un moteur rotatif qui permet le contrôle de la balle. Le rail du moteur linéaire ainsi que les différentes alimentations sont placées sur une structure du fabricant Item 24.

Plutôt que de trop s'attarder sur les caractéristiques techniques des différents moteurs qui sont fournis dans le dossier du rapport, voyons comment fonctionnent les différents appareils.

### 2.1 Moteur linéaire Servotube et boîtier Xenus :

Le moteur linéaire fonctionne par le principe d'induction magnétique, ce qui le rend extrêmement précis et puissant. Plus particulièrement, il peut atteindre une vitesse de 8.7 m/s et possède une répétabilité de 12 microns selon les données fournisseur. Une autre particularité intéressante de cet actuateur est sa simplicité de mise en œuvre. Le fabricant fournit le logiciel CME2, qui permet de programmer simplement le boîtier de commande Xenus. Il suffit simplement de programmer des « séquences » que le moteur doit effectuer comme par exemple « va à la position P » ou encore « avance jusqu'à rencontrer un obstacle ». Dans notre cas, les séquences utiles sont la commande en position et en vitesse du moteur par des tensions analogiques, ainsi que le Homing. Ces séquences sont appelées via une commande digitale envoyée le port0 de la carte d'acquisition. En annexe se trouve une description plus précise de ce système de séquences.

### 2.2 Moteur rotatif Maxon et boîtier ESCON 50/5 :

Les boîtiers ESCON possèdent l'avantage d'être programmables, même si dans notre cas nous n'avons pas utilisé cette particularité. Un signal digital active l'amplificateur du boîtier et permet de faire tourner le moteur. En ce qui concerne le contrôle des mouvements de ce dernier, il faut implémenter un contrôleur. Plusieurs essais ont été faits, grâce à un VI d'identification décrit plus loin dans le rapport. Nous avons choisi d'utiliser un contrôleur PD pour la position alors que pour la commande en vitesse nous avons opté pour un PID. Concernant les capteurs de mesures, un encodeur est branché au boîtier ESCON qui lui communique avec la carte d'acquisition.

### 2.3 Alimentation :

Les boîtiers de commande Xenus fonctionnent avec une tension de 24 VDC, qui lui est fournie par une alimentation de laboratoire dont nous avons réglé la tension. Le boîtier Xenus sert ensuite d'alimentation pour le moteur linéaire. En ce qui concerne les moteurs rotatifs, ils sont alimentés par le boîtier ESCON qui nécessite une tension de 36 VDC. Un système de transformation de tension a été imaginé et mis en œuvre par un étudiant précédent, ce qui permet de n'utiliser qu'une alimentation des 24 VDC. Il faut particulièrement être prudent avec ce système de transformateur. En effet il vaut veiller à éteindre l'alimentation de laboratoire et attendre que la LED de couleur verte

soit éteinte avant toutes manipulations. L'attente est due à la décharge de la grande capacité qui prend tout de même un instant.

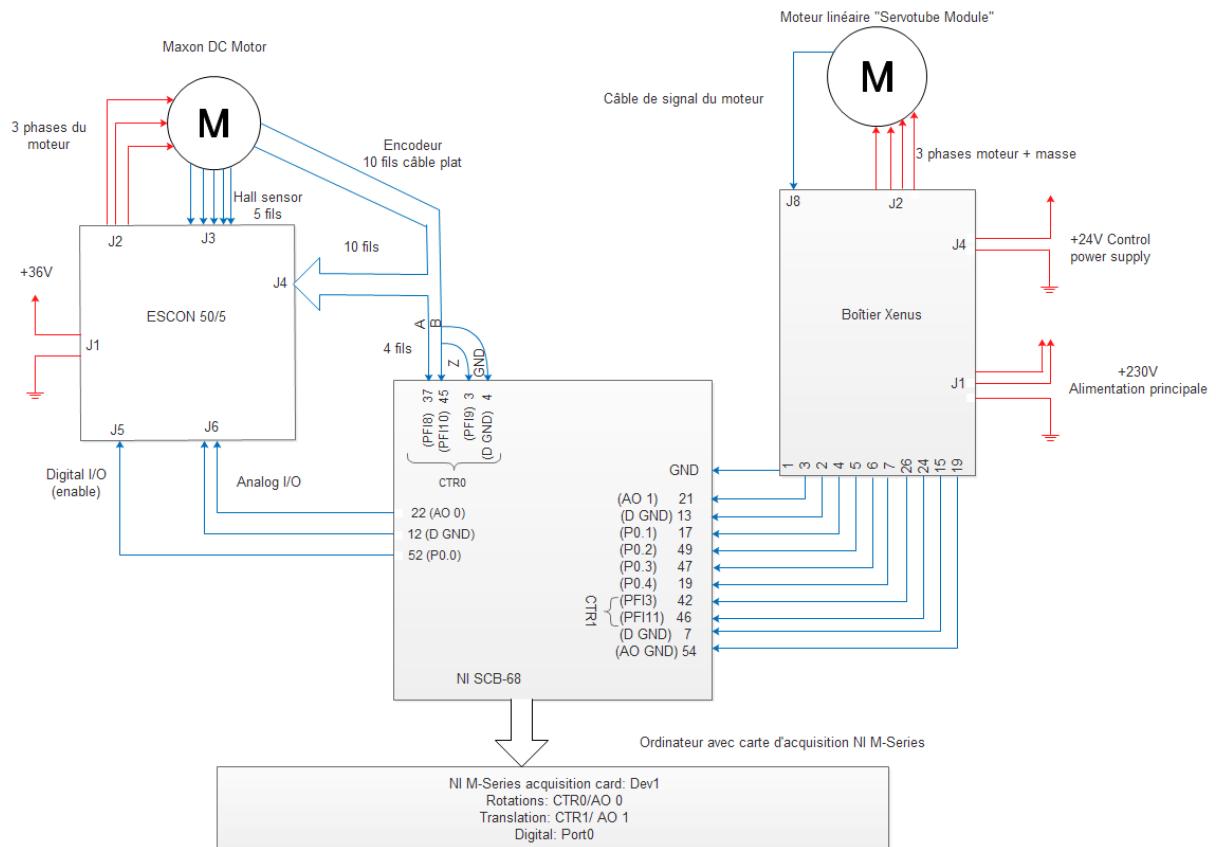
### 3. Améliorations :

Nous avons hérité d'un projet d'une année précédente, qui n'était pas totalement terminé. C'est pour cela que nous avons décidé de reprendre quelques parties du système de babyfoot, d'une part car la configuration précédente ne nous convenait pas et d'autre part pour nous familiariser avec le matériel qui nous était totalement inconnu.

#### 3.1 Câblage :

La configuration précédente ne permettait pas de fermer les blocs de connexions, ni de bien distinguer les connexions entre les différents appareils du système. Nous avons donc décidé de mieux organiser le câblage de manière à ce qu'il soit plus lisible.

Dans un premier temps, il s'agissait de repenser la manière de relier les différents appareils entre eux. Un des critères les plus importants était que le câblage devait être aussi générique que possible, afin de garantir une réutilisabilité et une duplication qui soit la plus aisée possible. En effet, le babyfoot actuel n'est motorisé que sur deux barres. Dans cette optique nous avons décidé de faire un câblage par barre, facilement reproductible. La figure ci-dessous représente le câblage pour une barre quelconque. Nous présentons ici une brève description du montage.



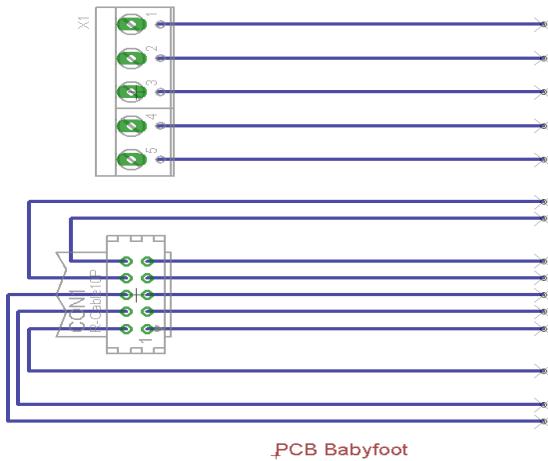
On reconnaît les deux types de moteurs qui actuent la barre : un moteur linéaire (ServoTubes Module SM2405S) pour le déplacement transversal de la barre et un moteur rotationnel (Maxon) pour la rotation de la barre. Chacun de ces moteurs est branché à un boîtier de contrôle. Pour le moteur Maxon, il s'agit d'un boîtier ESCON 50/5. Nous ne modifions pas les réglages existants sur ce boîtier, vu qu'ils ont fait leurs preuves dans les projets précédents. Du côté du moteur linéaire, ce dernier est relié à un boîtier XENUS. Celui-ci est aussi configurable. On peut notamment pré-enregistrer des séquences (pour chercher les butées par exemple). Nous avons choisi de garder celles qui ont déjà été définies lors de la première installation des moteurs. Les deux boîtiers sont ensuite branchés à un boîtier de connexion NI SCB-68. Celui-ci permet de relier tous les fils à la carte d'acquisition NI M-Series.

La partie qui a pris beaucoup de temps consistait à choisir les pins dans lesquels brancher chaque fil relayant le signal dont nous avons besoin. Il nous a d'abord fallu beaucoup de temps pour comprendre ce qui avait été fait dans le dernier projet et essayer de mettre toutes les informations en commun. Ce travail a abouti à la création de tableaux Excel décrivant les branchements (voir annexes). Nous avons fait trois tableaux différents : un pour chaque boîtier. Dans ces tableaux nous mettons non seulement l'origine et la destination de chaque fil, mais aussi sa fonction ainsi que sa couleur afin que la maintenance soit la plus aisée possible. Concernant les fils de l'encodeur, nous reprenons quatre de ces fils pour les connecter au boîtier de connexion lui-même communiquant avec la carte d'acquisition. En effet, nous devons effectuer le contrôle du moteur rotatif et avons donc besoin des signaux renvoyés par l'encodeur.

Après quelques discussions avec des experts, nous avons décidé de blinder les câbles de signaux. En effet, l'ensemble des connexions nécessaires passent dans le chemin de câbles du moteur, donc les signaux côtoient les câbles de puissance des moteurs. Cette association n'étant pas souhaitée, nous avons fait passer les cinq signaux du capteur à effet Hall dans un câble blindé à cinq pôles, et un autre de dix pôles pour l'encodeur.

Par ailleurs, nous avons pensé judicieux d'imaginer un système simple et rapide pour changer le moteur à la suite d'un incident ou simplement à cause de l'usure naturelle. En effet, dans la configuration précédente, le changement d'un des moteurs rotatifs peut nécessiter beaucoup de temps et il est facile d'introduire des erreurs de connexion.

Pour cela, nous avons conçu une plaque de Veroboard qui permet à l'utilisateur de connecter très simplement les signaux du moteurs au boitier de commande ESCON. Le schéma ci-après illustre la construction de la carte.



Notre carte est composée d'un bornier à vis cinq pôles pour circuit imprimé, qui permet de relier les cinq fils du capteur à effet Hall au boîtier de commande Escon. Nous avons aussi soudé la contrepartie du connecteur plat dix pôles de l'encodeur. La numérotation des pins ainsi que la liste détaillée des fils connectés sont données en annexe. Il est nécessaire de noter que par inadvertance nous n'avons pas soudé les fils dans le même ordre pour chaque carte, même si ceci ne change en rien le principe de connexion.

En ce qui concerne le connecteur dix pôles, on peut remarquer que certains pins sont sur une même piste. Étant donné que les trous de la carte Veroboard sont reliés entre eux selon une ligne, il faut donc veiller à isoler les trous qui ne doivent pas être connectés. Ceci est fait par exemple à l'aide d'une lime.

Finalement, par mesure de sécurité, nous avons installé un bouton pressoir d'urgence sur l'alimentation de l'ensemble des moteurs. Un simple pression sur ce dernier désactive l'alimentation et peut par exemple éviter des dégâts matériels ou physiques.

### 3.2 Améliorations Mécaniques :

Après quelques discussions avec M. Norbert Crot, le responsable de l'atelier d'automatique, nous avons décidé d'apporter quelques modifications mécaniques à notre système de babyfoot automatique.

Premièrement, il fallait se poser la question de l'alignement entre l'axe du moteur et celui des barres de joueurs. Cette question est importante car un désalignement pourrait accélérer l'usure des différents composants et à long terme il faudrait remplacer certaines pièces. Dans un premier temps, nous avons enlevé un roulement du babyfoot de manière à ne pas surcontraindre l'axe du moteur lorsqu'il est quelque peu décalé avec l'axe des barres. Ensuite, nous avons décidé de fixer un entraxe définitif entre les moteurs de manière à faciliter la mise en place de la structure. Malheureusement l'utilisation d'un pied à coulisse métallique se trouvait dérangeante car l'axe des moteurs linéaire est aimanté, ce qui provoquait un choc considérable entre ce dernier et l'instrument de mesure. Nous avons d'abord pensé à réaliser des calles dans un matériau amagnétique pour fixer cet entraxe.

Après réflexion et par mesure de simplicité, nous avons fait le réglage avec un pied à coulisse en plastique.

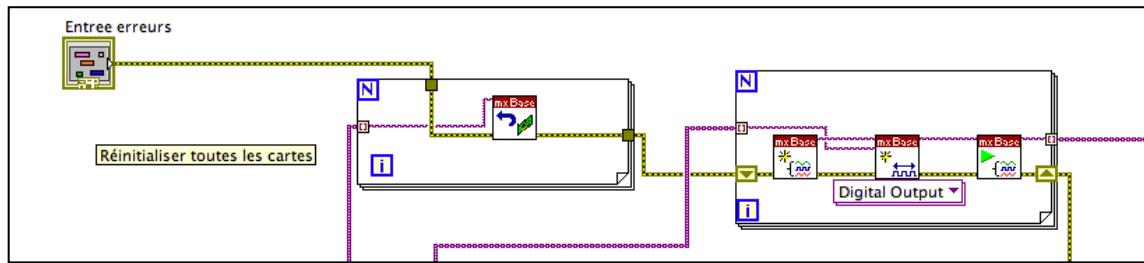
## 4. Base de fonctionnement du module National Instrument sur Labview :

Dans ce paragraphe, nous allons brièvement expliquer comment lire et écrire des données digitales ou analogiques sur les ports de la carte d'acquisition et ainsi permettre de communiquer avec les moteurs.

Une boucle typique d'acquisition de données NI-DAQmx se compose de trois étapes principales :

1. Initialisations des tâches et des « virtual channels ». (DAQ Init)
2. L'écriture ou la lecture de données. (DAQ Write/Read)
3. Fermeture de la tâche. (DAQ End)

### 4.1 Initialisation :



Sur cette capture d'écran du VI que nous avons baptisé DAQInit, on voit les étapes nécessaires à l'initialisation de la procédure. Soit de gauche à droite :

1. On commence par réinitialiser toutes les cartes avec une boucle qui effectue la tâche sur le nombre de cartes spécifié dans le fil violet.
2. Dans la deuxième boucle, on commence par créer une tâche, on crée ensuite une nouvelle « virtual channel » qu'on assigne à la tâche en cours. Pour créer la chaîne il nous suffit de spécifier le type (Digital Output dans notre cas) et de lui donner le nom, ou chemin d'accès. Finalement on démarre la tâche.

#### 4.1.1 Chemin d'accès pour une virtual channel :

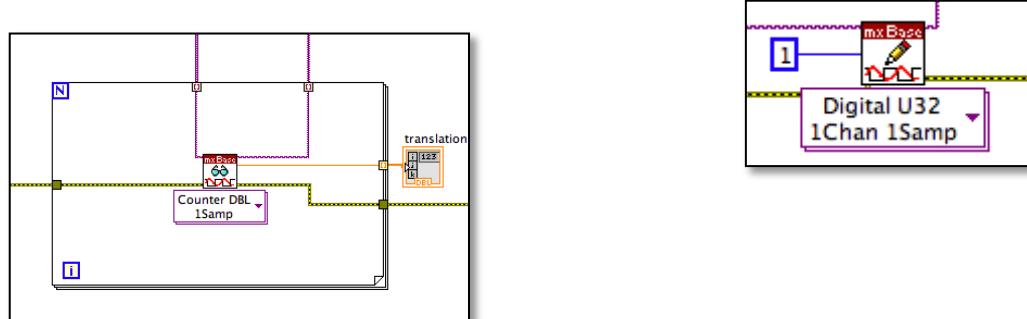
Voici ci-dessous quelques indications pour spécifier le nom de la virtual channel, qui correspond en fait au nom de la carte d'acquisition suivi du port de la carte avec lequel on souhaite communiquer au travers de la tâche.

Nom de la carte : Dev1 ou Dev2 dans le cas où nous avons deux cartes d'acquisition.

Port sur lequel écrire : Voir la description sur le boîtier de connexion d'entrée/sortie.

Par exemple :

- Si l'on veut accéder à un port digital, on peut le faire dans notre cas grâce au port0, on écrira donc : « Dev1/port0 » pour accéder au port0 de la carte numéro 1.
- Si l'on veut accéder à un compteur (pour un encodeur), on écrira : « Dev1/ctr0 » ou « Dev1/ctr1 » suivant si l'on veut accéder au compteur 1 ou 2 de la carte 1.
- Si l'on veut accéder à une sortie analogique, on écrira : « Dev1/AO0 » ou « Dev1/AO1 » suivant si l'on veut accéder à la sortie analogique 0 ou 1 de la carte 1.

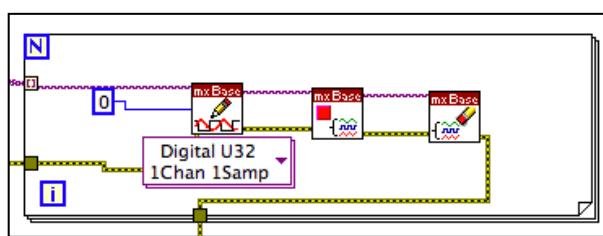


## 4.2 Écriture/Lecture d'informations

Sur l'image de gauche, on lit une information sur la tâche et donc sur le port spécifique de la carte. Pour ce faire on utilise l'icône de lecture et on fournit la tâche sur laquelle écrire ainsi que l'erreur courante. À la sortie, on récupère l'erreur ainsi que la tâche et on a accès au contenu du port en question. Il faut aussi spécifier quel genre de valeur et le format que l'on va lire. (Ici on lit un compteur de type Double)

De même, sur l'image de droite on écrit dans une tâche. Ainsi on donne la constante à écrire ainsi que la tâche et l'erreur courante en entrée, et on récupère la tâche modifiée ainsi que les erreurs qui aurait pu se passer lors de l'écriture. Ici aussi on veillera à spécifier le type de valeur à écrire.

## 4.3 Fermeture de la tâche :

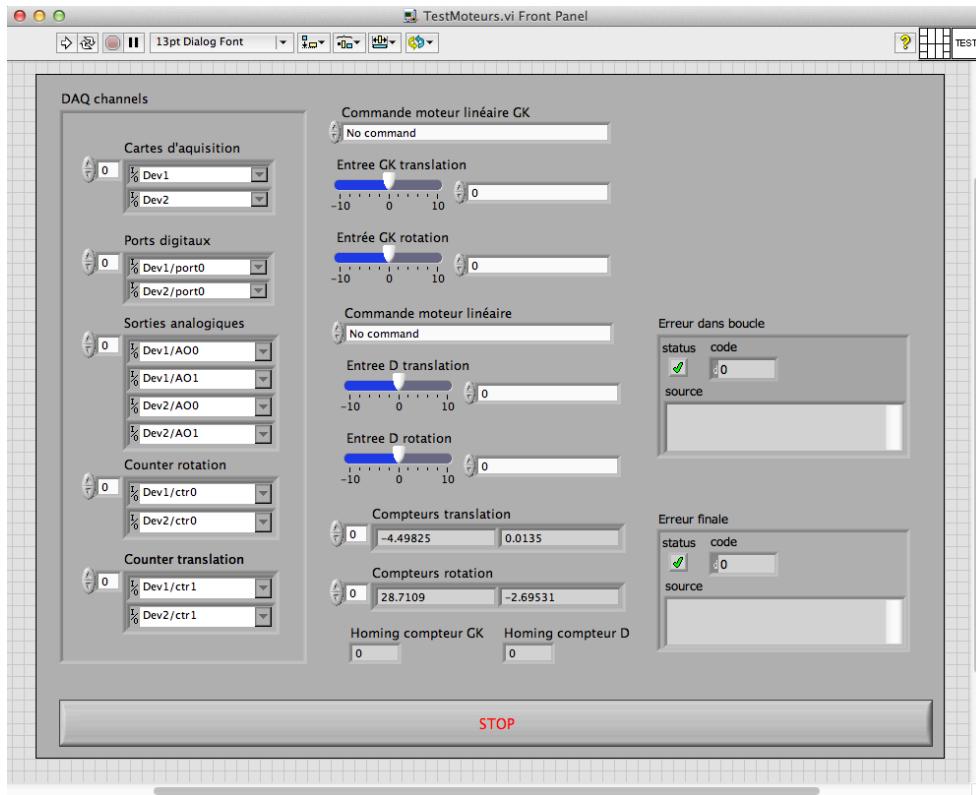


Dans cette dernière étape, on commence par mettre le port dans l'état final que l'on souhaite (dans l'image c'est 0), ensuite on arrête la tâche et finalement on la supprime.

Avec ces quelques instructions et un peu de connaissance du logiciel Labview, il est possible de créer quelques programmes simples d'acquisition. Pour de plus amples informations concernant le module NI-DAQmx, il vous suffit de vous rendre sur le site de National Instrument et de suivre les tutoriels très complets.

## 5. Acquisition des données et premier tests :

Après avoir correctement câblé notre système vient le moment de tester le fonctionnement des différents appareils installés, tels que les capteurs ainsi que les moteurs. Nous avons donc créé un VI dont la fenêtre principale se présente ainsi :



Nous présentons ici les principales fonctionnalités du VI. La première étape consiste à entrer les channels dans la partie gauche de l'interface. Voici les informations à donner au programme :

1. *Cartes d'acquisition* : ce sont les cartes d'acquisition M-Series de NI. Comme décrit plus haut, nous avons organisé le câblage par barre. Il y a donc une carte d'acquisition par barre. Pour l'instant nous ne contrôlons que deux barres, mais le VI a été conçu de manière à ce que l'on puisse facilement ajouter des barres et des carte d'acquisition. Nous avons le setup suivant :
  - a. *Dev1* : Carte d'acquisition pour la barre des défenseurs
  - b. *Dev2* : Carte d'acquisition pour la barre du gardien
2. *Ports digitaux* : ce sont les ports digitaux sur lesquels on va écrire les commandes digitales. Nous utilisons un port par barre afin d'enclencher les boîtiers de commande et lancer des séquences pour le moteur linéaire. La construction de ces ports est décrite en annexe.
3. *Sorties analogiques* : c'est sur ces sorties que le programme va écrire la commande pour les moteurs. La commande est un voltage entre  $\pm 10V$ . Chaque barre est commandée par deux sorties analogiques : une par moteur.
  - a. *AO0* : Sortie analogique 0, commande le moteur rotatif.
  - b. *AO1* : Sortie analogique 1, commande le moteur linéaire.
4. *Counters* : ce sont les compteurs. Il y en a deux par barre.
  - a. *CTR0* : compteur 0 pour le moteur rotatif
  - b. *CTR1* : compteur 1 pour le moteur linéaire

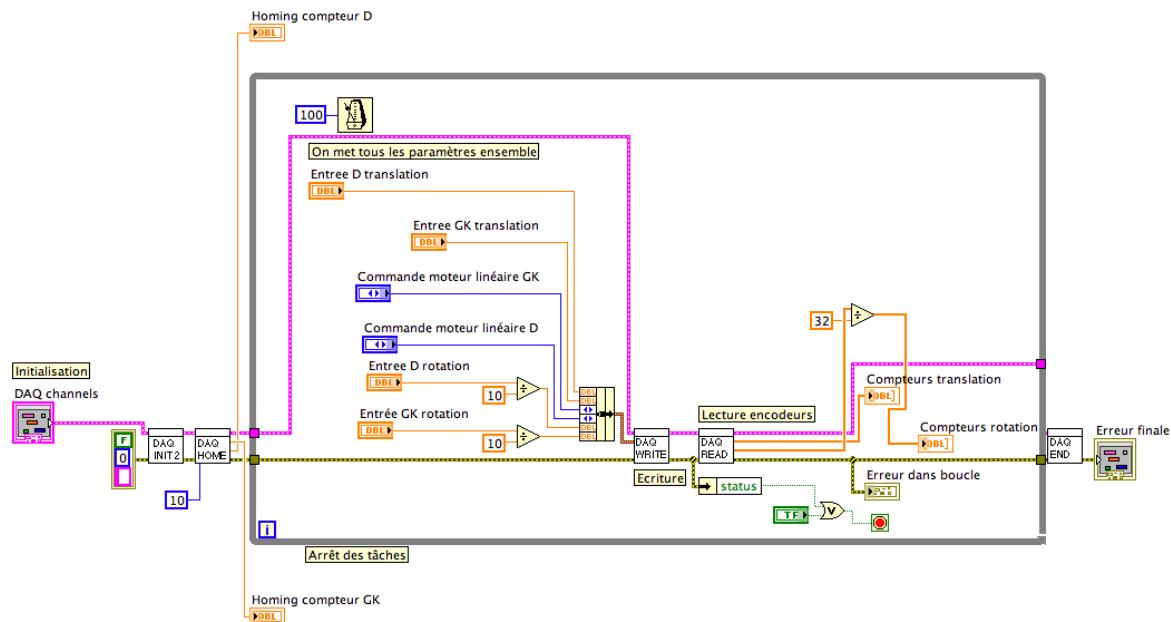
La partie centrale du programme permet de tester les moteurs dans différentes configurations. On choisit tout d'abord comment on veut commander les moteurs. Il y a ici un point essentiel : les moteurs linéaires sont commandés par les boîtiers XENUS qui contiennent déjà des contrôleurs en position et en vitesse, tandis que les moteurs linéaires tournent en boucle ouverte et doivent être régulés par notre programme. Nous avons donc défini différentes séquences pour commander les moteurs linéaires, dont deux sont commandés en position et en vitesse. Pour plus de détails concernant ces séquences, une description détaillée est disponible en annexe. Ces séquences sont sélectionnées à travers le port digital. Il faut donc un moyen de choisir quelle commande on désire. Voici les différentes options :

1. *No command* : Les deux moteurs de la barre sont désactivés et on n'envoie aucune commande.
2. *Pass through* : Les moteurs linéaires sont désactivés, seuls les moteurs rotatifs peuvent être commandés. Cette option existe afin de commander séparément les moteurs linéaires et rotatifs.
3. *Vitesse* : Les moteurs linéaires sont commandés en vitesse et les moteurs rotatifs sont activés. Ils fonctionnent ici en boucle ouverte. L'implémentation d'un régulateur est nécessaire si on désire commander le moteur rotatif en position ou vitesse.
4. *Position* : Les moteurs linéaires sont commandés en position et les moteurs rotatifs sont activés. Même remarque que pour l'option *vitesse* pour les moteurs rotatifs.

Une fois que le mode de fonctionnement du moteur est choisi, on peut envoyer des commandes au moteurs. On choisit en fait le voltage à envoyer aux moteurs par les sorties analogiques correspondantes.

Dans la partie inférieure du VI, on peut lire la valeur des différents compteurs.

Voici le code du programme :



On distingue clairement les différentes parties du programme (de gauche à droite) :

1. *Initialisation* : démarrage des tâches et connection aux moteurs
2. *Homing* : recherche des zéros. Les moteurs linéaires font leur recherche de zéros en cherchant les butées et en se positionnant à une certaine distance de celles-ci. Cette distance est spécifique à chaque barre et doit être insérée dans la programmation des séquences sur le XENUS à l'aide du logiciel CME2. Les moteurs rotatifs ne font rien. Ceci est un point qui pourra être amélioré dans les prochaines étapes du projet : positionnement précis des joueurs en rotation au début de la partie.
3. *Boucle principale* : on boucle jusqu'à ce que l'utilisateur appuie sur le bouton stop ou qu'une erreur se produit. Le programme scanne en continu les entrées fournies par l'utilisateur et commande les moteurs en fonction avant de lire les encodeurs et d'afficher leurs valeurs.
4. *Arrêt des tâches* : on arrête toutes les tâches en fermant tous les canaux et en éteignant les moteurs.

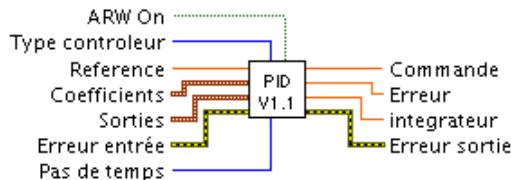
## 6. VI d'identification pour la calibration des contrôleurs :

Une fois que le fonctionnement basique du système est garanti, il convient de contrôler les moteurs en position et en vitesse. Comme dit précédemment, c'est chose faite pour les moteurs linéaires dont les boîtiers de commandes intègrent déjà des contrôleurs. Cependant, les moteurs rotatifs travaillent encore en boucle ouverte et il est impossible de les commander sans régulateur. Nous avons donc conçu dans un premier temps un programme qui ajuste la commande en fonction de la sortie actuelle du système.

### 6.1 Contrôleur PID

Afin de pouvoir choisir quel contrôleur nous voulions utiliser pour chaque commande, nous avons conçu un sous-VI qui permet de choisir non seulement le type de contrôleur mais également de le calibrer. Ce VI se nomme *ControleurMoteursRotatifs\_V1\_1.vi*. On peut choisir quel moteur on veut contrôler (gardien ou défenseur). Il est conçu de manière à ce qu'on puisse facilement ajouter deux barres. On peut aussi choisir entre plusieurs contrôleurs : aucun (boucle ouverte), P, PI, PD ou encore PID. Les coefficients sont ici à entrer manuellement, ce qui permet d'intégrer le VI dans une structure de test de paramètres par exemple.

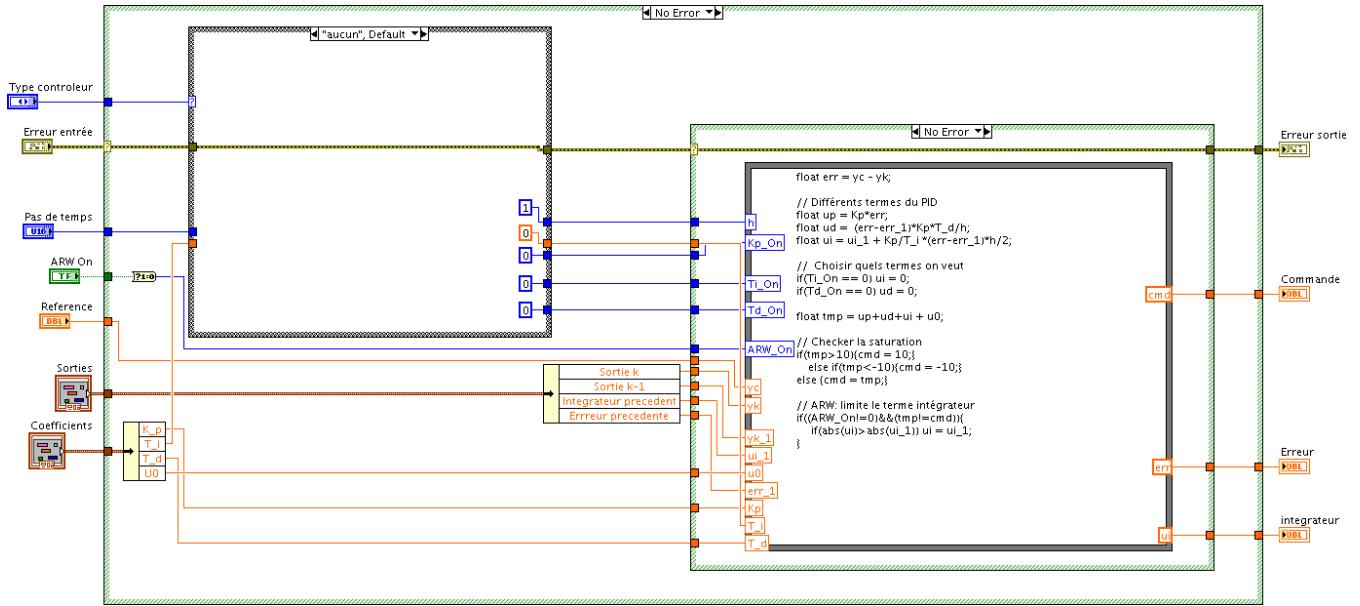
Voici les différentes entrées et sorties du VI :



On peut choisir entre différents types de contrôleurs. En fonction il faudra aussi spécifier des coefficients, qui sont regroupés dans le cluster « Coefficients ». Comme il opère dans un système en boucle fermée, on doit également lui donner les valeurs des sorties précédentes. Celles-ci sont regroupées dans le cluster « Sorties » et se présentent ainsi :

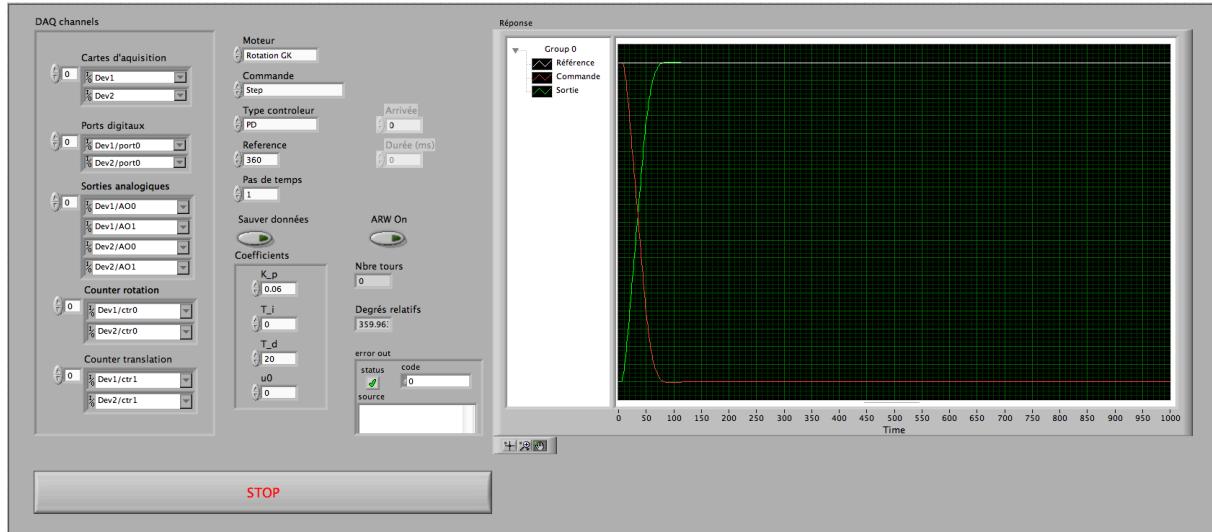
1. *Sortie k* : sortie actuelle du système, lue par les encodeurs.
2. *Sortie k-1* : sortie précédente du système (un pas d'intégration avant)
3. *Erreur précédente* : différence entre commande et sortie précédente
4. *Intégrateur précédents* : Terme intégral précédent (ui dans le code).

On peut en plus activer un filtre ARW afin de limiter l'effet du terme intégral lorsqu'il y a saturation. Celle-ci est par ailleurs gérée dans le code. Les autres vérifications que fait le VI sont celles nécessaires pour assurer qu'il n'y ait pas de division par zéro. Il va donc vérifier la valeur du pas de temps et du coefficient  $T_i$  pour le terme intégral. Si un des deux ou les deux sont nuls, le VI sort une erreur et des valeurs de commande nulles.



## 6.2 Identification des coefficients pour le contrôleur

La seconde partie pour la mise en place d'un contrôle efficace est la calibration des contrôleurs c'est-à-dire le choix des bons coefficients pour chaque type de contrôleur. Nous avons donc conçu le VI suivant pour réaliser cette tâche.



La partie de gauche est la même que pour le VI de test des moteurs : on choisit les cartes d'acquisition, les ports, les sorties analogiques ainsi que les compteurs. La partie centrale permet ensuite de choisir différents paramètres :

1. *Moteur* : on choisit sur quel moteur rotatif on choisit de faire l'identification. Il faudra faire cette identification des paramètres pour chaque moteur. En effet, les barres actuées ne sont pas toutes les mêmes au niveau de l'inertie notamment. Le comportement de l'ensemble moteur-barre va donc être différent pour chaque barre.

2. *Commande* : on a le choix entre
  - a. *Aucune* : boucle ouverte. On applique bêtement un voltage et on mesure la sortie. Aucun autre paramètre à préciser
  - b. *Step* : commande en position, boucle fermée. On doit ici préciser la référence en degrés.
  - c. *Rampe* : commande en vitesse, boucle fermée. On doit ici spécifier la position à laquelle on désire arriver et en combien de temps on veut l'atteindre. De cette manière, on simule une commande en vitesse.
3. *Contrôleur* : Si on a choisi un setup en boucle fermée (step ou rampe), on peut encore choisir le type de contrôleur que l'on désire utiliser.
  - a. *P* : commande proportionnelle
  - b. *PD* : commande proportionnelle dérivée
  - c. *PI* : commande proportionnelle intégrale
  - d. *PID* : commande proportionnelle intégrale et dérivée
4. *Coefficients* : selon le type de contrôleur sélectionné, il faut lui donner des coefficients. C'est cette partie qui est intéressante car le comportement du système va être dicté par le choix de ces coefficients. Voici les résultats auxquels nous sommes arrivés lors de notre identification.

Coefficient	Position	Vitesse
K_p	0.06	0.04
T_i	-	750
T_d	20	18

On voit ici le type de contrôleur sélectionné pour les deux commandes : PD pour la position et PID pour la vitesse. Le fait que ce soient ces contrôleurs qui conviennent le mieux découle des moteurs. En effet, pour une commande en position, le système peut être modélisé au travers d'une fonction de transfert du type

$$G(s) = \frac{K}{s(\tau s + 1)}$$

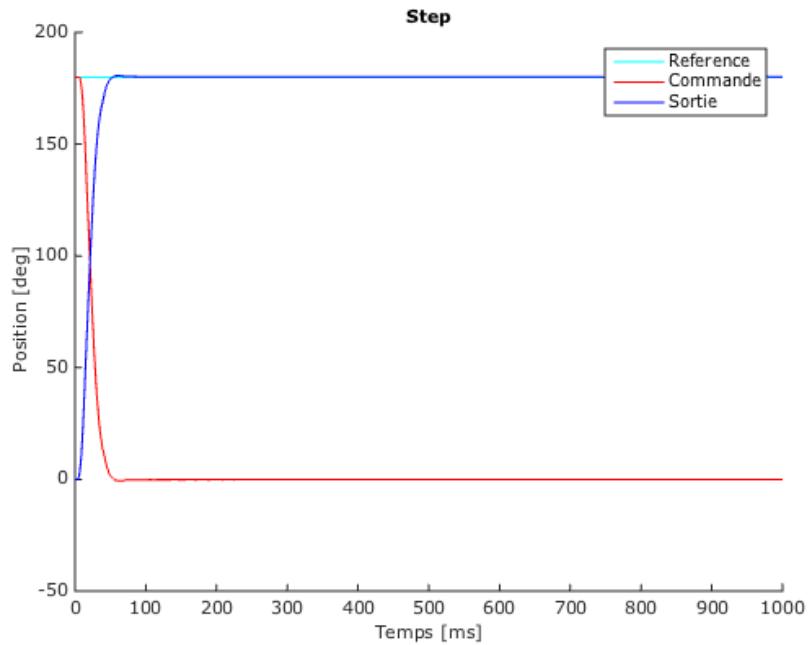
Cette fonction de transfert contient déjà un intégrateur. Il n'est donc pas nécessaire d'en rajouter un dans le contrôleur. Pour la commande en vitesse cependant, le système est modélisé par une fonction de transfert du type

$$G(s) = \frac{K}{\tau s + 1}$$

Celle-ci ne contient aucun intégrateur. Il faut donc lui en ajouter un pour ne plus avoir d'erreur par rapport à la commande.

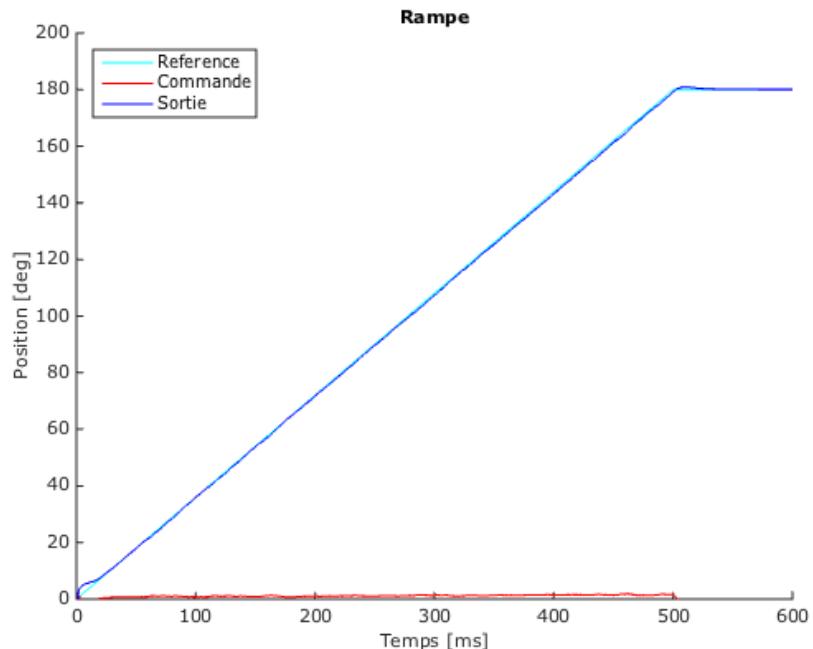
5. Il y a encore la possibilité de sauver les données dans un fichier matlab binaire afin de pouvoir travailler avec les données affichées dans le graphe de droite du « front panel ». Le format a été choisi de manière à pouvoir utiliser les outils d'identification temporelle et fréquentielle disponibles sur le site du laboratoire d'automatique ([la.epfl.ch/tp](http://la.epfl.ch/tp)). Voici les performances atteintes avec notre choix de contrôleurs :

a. Commande en position



On voit sur cette figure que la position désirée est atteinte en 50 [ms], qu'il n'y a pas d'erreur et on ne peut distinguer d'oscillation. C'est donc à première vue un bon contrôleur.

b. Commande en vitesse



Cette figure montre que nous sommes capables de suivre une rampe sans erreur. Cela a été rendu possible en ajoutant un intégrateur au contrôleur.

## 7. Propositions d'améliorations

Etant donné que nous ne sommes pas parvenus à atteindre les objectifs initiaux, nous nous permettons de proposer quelques améliorations pour les étudiants suivants.

En particulier, nous pensons qu'il serait judicieux de retravailler la plaque PCB qui permet de changer les moteurs. Notre faible expérience en matière de conception de circuit imprimé ne nous a pas permis de réaliser un travail propre et fiable. Ainsi, il serait envisageable de reconstruire une carte PCB manuellement, de manière plus appliquée de façon à éviter les courts circuits et les problèmes de connexion. Une autre solution serait de soumettre une carte de circuit imprimé conçue sur un logiciel prévu à cet effet à un atelier de l'EPFL et ainsi obtenir un produit fiable et sans trop de défauts. Ces deux approches permettront aux étudiants s'ils n'en ont pas encore la capacité soit d'apprendre la soudure électronique ou la conception de circuit imprimé par ordinateur.

Une autre amélioration qui pourrait améliorer le babyfoot automatique serait le positionnement vertical des joueurs de manière automatique lors de la phase de « homing » (recherche de zéros). En effet, le système existant ne permet pas un tel positionnement et celui-ci doit être fait manuellement, ce qui n'est pas précis.

Au niveau de l'alimentation, et plus précisément la conversion de tension, il y a une marge d'amélioration. Il faudrait modifier le système de conversion de tension dont on tire les 36V. En effet, le système actuel n'est pas isolé et un contact malencontreux peut avoir des conséquences fâcheuses. Il conviendrait donc de créer un boîtier qui faciliterait la prise en main. De plus, la fiabilité de ce système n'est pas entièrement garantie. Nous avons eu des problèmes de court-circuit. Cependant nous tenons à préciser que ce système est nécessaire et qu'il ne peut être remplacé par une simple alimentation de laboratoire.

Concernant les VI, la prochaine étape du projet consistera à mettre en commun les différentes parties faites par les différents groupes durant ce semestre. C'est un gros travail qui demandera de bien comprendre chaque partie et à trouver un bon moyen de les intégrer dans un seul VI qui commandera tout le système. L'application des améliorations précédentes pourrait en outre résoudre différents problèmes survenus lors de la calibration des contrôleurs.

La conception des VI existants permet facilement d'intégrer des barres en plus dans le système. En effet, toutes les entrées/sorties sont rassemblées dans des tableaux, facilement extensibles.

## 8. Conclusion

Hériter d'un projet mis en route depuis des années et y amener des améliorations représente un bon challenge, que nous avions à cœur de relever. La première étape de prise en main et de familiarisation avec le système existant a été l'une des plus rudes. En effet, il s'agissait de domaines d'applications dans lesquels nous n'avions peu ou pas de connaissances et la documentation existante était maigre. Cela nous a donc considérablement retardés mais cela nous a permis de nous familiariser avec les rudiments de l'électronique. Nous avons repensé la configuration du système dans l'optique de rendre aussi aisée que possible la prise en main et surtout l'amélioration du système par les étudiants qui reprendront le flambeau. Ces derniers auront également à disposition tous les outils nécessaires à la création d'un VI pour l'entier du système. En effet, ils pourront réutiliser tels quels les VIs d'identification et de contrôle

## Remerciements :

Nous tenons à remercier chaleureusement M. Christophe Salzmann et M. Norbert Crot pour leur implication et leurs conseils avisés. Merci également à Milan Korda pour le suivi hebdomadaire de nos activités. Un grand merci tout particulier à M. Jacques Epitaux (Epitaux electronics) pour le temps qu'il nous a accordé lors de la réalisation du veroboard. Finalement, nos remerciements vont encore à M. André Guignard qui nous a guidé dans les modifications de câblage.

## Annexes :

### Détails des connexions pour le boîtier de connexion SCB-68 :

Boîtier/ connecteur	Origine fil	N°Pin	Destination fil	N°Pin	Couleur fil	Fonction	Signal	Commentaire
XENUS		J7-1	SCB-68	GND	jaune	Masse	masse (boîtier lui-même fait office de masse	
XENUS		J7-2	SCB-68	13	blanc	D Gnd	$\pm 10$ Vdc Ref(-)	
XENUS		J7-3	SCB-68	21	rouge	AO 1	$\pm 10$ Vdc Ref(+)	
XENUS		J7-4	SCB-68	17	bleu	P0.1	[IN1] Digital Enable	bit 1 du Port0
XENUS		J7-5	SCB-68	49	gris	P0.2	[IN2] Entrée digitale	bit 2 du Port0
XENUS		J7-6	SCB-68	47	blanc	P0.3	[IN3] Entrée digitale	bit 3 du Port0
XENUS		J7-7	SCB-68	19	rouge	P0.4	[IN4] Entrée digitale	bit 4 du Port0
XENUS		J7-15	SCB-68	7	brun	D Gnd	masse	
XENUS		J7-19	SCB-68	54	rose	AO Gnd	masse	
XENUS		J7-24	SCB-68	46	bleu	PFI11	Signal B	encodeur
XENUS		J7-26	SCB-68	42	vert	PFI3	Signal A	encodeur
ESCON		J5-2	SCB-68	52	blanc	Port digital	D enable	bit 0 du Port0
ESCON		J6-1	SCB-68	22	vert	Entrée analogique	AO 0	
ESCON		J6-7	SCB-68	12	brun	Entrée analogique	D ground	
Encodeur		3	SCB-68	4	vert-brun	D Gnd	Masse	
Encodeur		6	SCB-68	37	blanc-jaune	PFI8	Signal A	4 fils bicolores que l'on prend de l'encodeur
Encodeur		8	SCB-68	45	jaune-brun	PFI10	Signal B	
Encodeur		10	SCB-68	3	blanc-rouge	PFI9	Signal Z (déduit)	

Câble blindé 3 pôles, sort du pin J7 XENUS  
 Câble blindé 8 pôles, sort du pin J7 XENUS

## Détails de connexion pour le boîtier de commande ESCON 50/5 :

Boîtier/ connecteur	N°Pin	Destination fil	N°Pin	Couleur fil	Fonction	Signal	Commentaire
Custom Board	-	Boîtier	N°Pin				
Custom Board	-	ESCON	J1-1	brun	Puissance	Masse	
ESCON	J2-1	Moteur	J1-2	jaune	Puissance	Vcc +36V	
ESCON	J2-2	Moteur	-	rouge	Puissance	Winding 1	
ESCON	J2-3	Moteur	-	noir	Puissance	Winding 2	
Moteur	-	ESCON	-	jaune	Puissance	Winding 3	
Moteur	-	ESCON	J3-1	blanc	Capteur de Hall	Hall Sensor 1 input	
Moteur	-	ESCON	J3-2	brun	Capteur de Hall	Hall Sensor 2 input	
Moteur	-	ESCON	J3-3	bleu	Capteur de Hall	Hall Sensor 3 input	
Moteur	-	ESCON	J3-4	noir	Capteur de Hall	Power supply (+5VDC, <30mA)	
Moteur	-	ESCON	J3-5	métal	Capteur de Hall	Masse	
Encodeur	1	ESCON	J4-1	noir	Encodeur	-	
Encodeur	2	ESCON	J4-2	rouge	Encodeur	Power supply (+5VDC, <70mA)	
Encodeur	3	ESCON	J4-3	vert-brun	Encodeur	Masse	
Encodeur	4	ESCON	J4-4	blanc	Encodeur	-	
Encodeur	5	ESCON	J4-5	bleu	Encodeur	Signal non A	
Encodeur	6	ESCON	J4-6	blanc-jaune	Encodeur	Signal A	
Encodeur	7	ESCON	J4-7	vert	Encodeur	Signal non B	
Encodeur	8	ESCON	J4-8	jaune-brun	Encodeur	Signal B	
Encodeur	9	ESCON	J4-9	brun	Encodeur	Signal non Z (déduit)	
Encodeur	10	ESCON	J4-10	blanc-rouge	Encodeur	Signal Z (déduit)	
SCB-68	52	ESCON	J5-2	blanc	Port digital	D enable	bit 0 du Porto
SCB-68	22	ESCON	J6-1	vert	Entrée analogique	AO 0	
SCB-69	12	ESCON	J6-7	brun	Entrée analogique	D ground	

**Détails de connexion pour le boîtier de commande Xenus :**

Boîtier/ connecteur	N°Pin	Destination fil	N°Pin	Couleur fil	Fonction	Signal	Commentaire
Prise secteur	- XENUS	J1	Câble noir	Alimentation	+230V		Alimentation principale du boîtier
XENUS	J2	Moteur linéaire -	Câble noir	Puissance			
Convertisseur 230/24 V	- XENUS	J4	Rouge et jaune	Alimentation	+24V		2 Câbles, alimentation du contrôleur dans le boîtier XENUS
SCB-68	GND	XENUS	J7-1	jaune	Masse	masse (boîtier lui-même fait office de masse	
SCB-68	13	XENUS	J7-2	blanc	D Gnd	±10 Vdc Ref(-)	
SCB-68	21	XENUS	J7-3	rouge	AO 1	±10 Vdc Ref(+)	
SCB-68	17	XENUS	J7-4	bleu	P0.1	[IN1] Digital Enable	bit 1 du Port0
SCB-68	49	XENUS	J7-5	gris	P0.2	[IN2] Entrée digitale	bit 2 du Port0
SCB-68	47	XENUS	J7-6	blanc	P0.3	[IN3] Entrée digitale	bit 3 du Port0
SCB-68	19	XENUS	J7-7	rouge	P0.4	[IN4] Entrée digitale	bit 4 du Port0
SCB-68	7	XENUS	J7-15	brun	D Gnd	masse	
SCB-68	54	XENUS	J7-19	rose	AO Gnd	masse	
SCB-68	46	XENUS	J7-24	bleu	PF11	Signal B	encodeur
SCB-68	42	XENUS	J7-26	vert	PF13	Signal A	encodeur
Moteur	-	XENUS	J8	Câble noir	Encodeur		

Câble blindé 3 pôles, sort du pin J7  
 Câble blindé 8 pôles, sort du pin J7

## Construction du port0 de la carte d'acquisition :

Construction du port 0							
#bit	Fonction	XENUS disable	XENUS homing	XENUS Position	Analog	XENUS Velocity	Analog
0	Digital enable ESCON	x	x	x		x	
1	Digital enable XENUS	1	1	1		1	
2	XENUS [IN2]	1	0	1		0	
3	XENUS [IN3]	1	0	0		1	
4	XENUS [IN4]	0	0	0		0	
5							
6							
7							

Par exemple, pour appeler appeler le homing on envoie à la carte Dev1 l'instruction : 111x

Le x indique que cet instruction n'est pas importante pour faire fonctionner la séquence du Xenus, en d'autre terme on peut mettre un 1 pour activer le contrôle du moteur rotatif ou un zéro pour faire l'inverse, la séquence du Xenus s'exécutera de la même façon.

Descriptions des séquences :

**Homing :**

1. On commence par un délai d'attente.
2. On avance dans une direction jusqu'au Hardstop (jusqu'à qu'une surface s'oppose au mouvement)
3. On recule d'une certaine distance pour positionner les joueurs dans la configuration de départ.
4. On règle la position finale comme « home »

**Analog Position :**

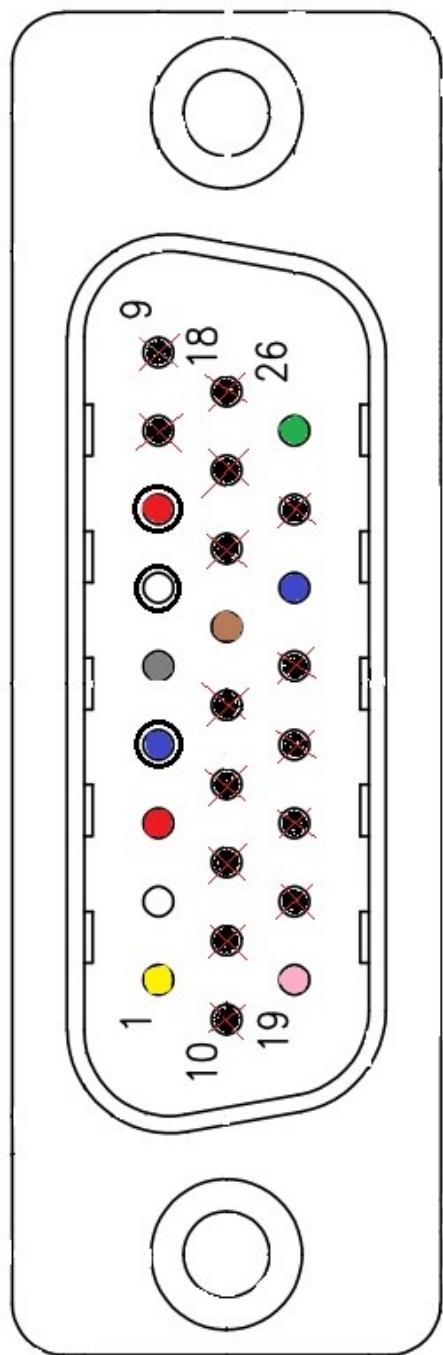
Cette séquence permet à l'utilisateur de choisir une position entre -10 Volt et +10 Volt. À noter que 0 Volt correspond à la position réglée comme « Home ». On procédera d'abord au Homing avant d'utiliser cette séquence.

**Analog Velocity :**

Comme pour la position, l'utilisateur en réglant la tension envoyée au boitier Xenus peut choisir une vitesse comprise entre -10 V et + 10V. Le signe permet de choisir la direction du déplacement.

## Schéma de connexion du connecteur J7 du boîtier Xenus :

Nous avons décidé de créer ce schéma pour faciliter la soudure des différents câbles de signaux sur ce connecteur, car cette dernière n'est de loin pas triviale et nous avons déjà dû la refaire. Pour la référence du connecteur, il s'agit d'un D-Sub High Density 26 poles.



Les fils correspondants sont contenus dans deux câbles blindés provenant de la boîte SCB-68. Les pins entourés d'un cercle noir correspondent aux fils du câble ne contenant que trois fils. (Rouge/Bleu/Blanc)

Quant aux pins de couleur noir barré, ceux-ci ne sont simplement pas connectés.