

EPFL

Projet d'ingénierie simultanée

Babyfoot : Vision et Calibration

Etudiants : Leila AFILAL, Kenza BENABDERRAZIK, David NIGRI

Professeur : Christophe SALZMANN

Spring 2019

Table des matières

1. Introduction	3
2. Organisation générale du Vi.....	4
3. Calibration de la vision.....	7
a. Diagnostic et Mesure des erreurs.....	7
b. Correction par Vision Assistant.....	8
c. Implémentation dans le programme.....	17
d. Etablissement d'une procédure automatique et améliorations futures	17
4. Correction des translations.....	19
a. Attaquants : <i>joueurs 1, 2 et 3.</i>	20
b. Milieu : <i>joueurs 4, 5, 6, 7, 8.</i>	20
c. Défenseurs : <i>joueurs 9, 10.</i>	22
5. Rotation	23
6. Contrôle	24
7. Conclusion.....	24

1. Introduction

Dans le cadre du projet d'ingénierie simultanée au laboratoire d'automatique de l'EPFL, nous avons été amenés à améliorer certains aspects d'un babyfoot automatisé, conçu en 2012 par des étudiants. Ce semestre, nous nous sommes divisés en 2 groupes, l'un se concentrant sur la calibration, l'autre sur la stratégie.

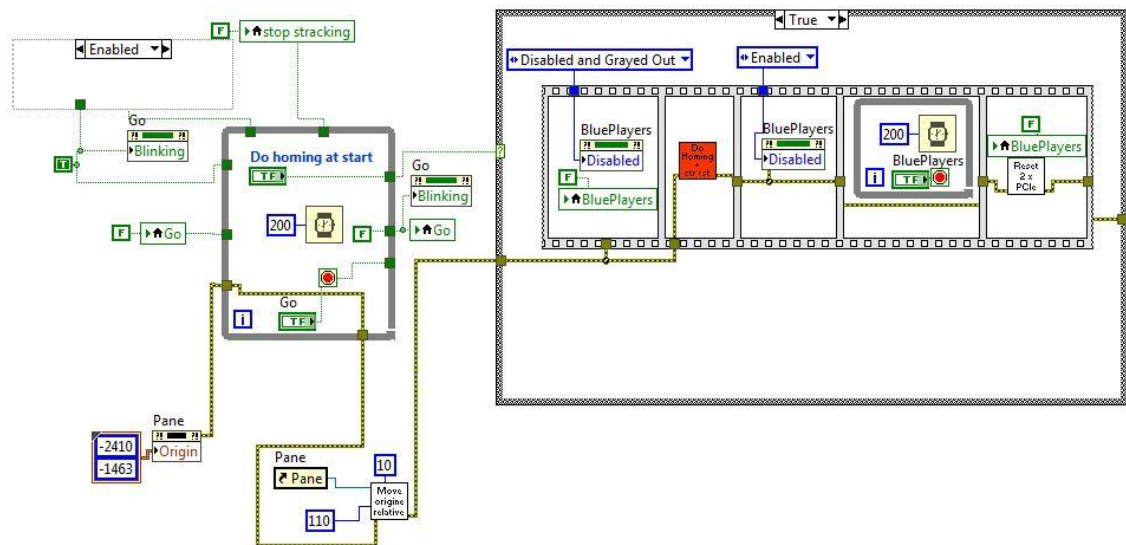
Le but de ce projet fut donc de procéder à l'amélioration de la calibration de l'image obtenue par la caméra et à la correction des commandes de moteurs linéaires et de rotation.

En effet, en testant la table automatisée, plusieurs problèmes nous sont apparus. Les joueurs ne répondaient pas d'une manière sensée au mouvement de la balle. Ainsi après avoir étudié le code LabVIEW pendant plusieurs semaines afin de le maîtriser, nous avons réalisé la difficulté d'améliorer un projet sur lequel au moins une vingtaine de personnes avaient déjà travaillé séparément. C'est pourquoi nous avons consacré du temps à rajouter des commentaires sur certains des VIs afin qu'ils soient plus lisibles pour les prochaines personnes qui travailleront dessus.

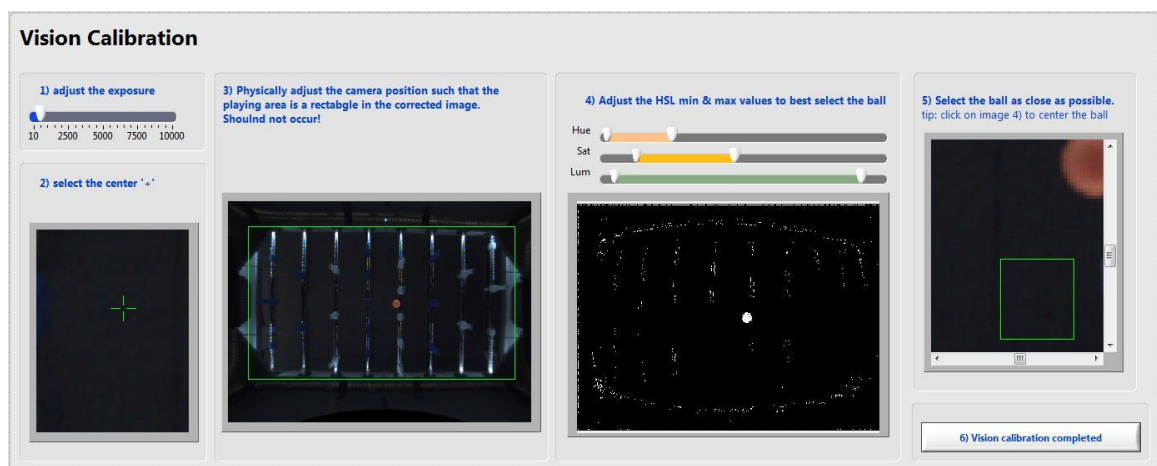
Ainsi, après avoir effectué quelques tests et mesures, nous avons repéré un problème de calibration d'image. Après avoir réglé ce souci de vision, le positionnement de la balle était certes précis à quelques millimètres près, mais il était toujours évident que des problèmes persistaient par rapport au jeu des joueurs. Nous nous sommes donc penchés sur la partie du code *Stratégie*, et plus précisément sur les commandes envoyées aux moteurs en translation. Après avoir corrigé les translations pour les 4 rangées de joueurs, le jeu du babyfoot s'en est trouvé grandement amélioré.

Nous avons aussi entamé plusieurs pistes d'amélioration que nous n'avons pas pu mener à bien mais que nous avons tout de même documenté.

2. Organisation générale du Vi

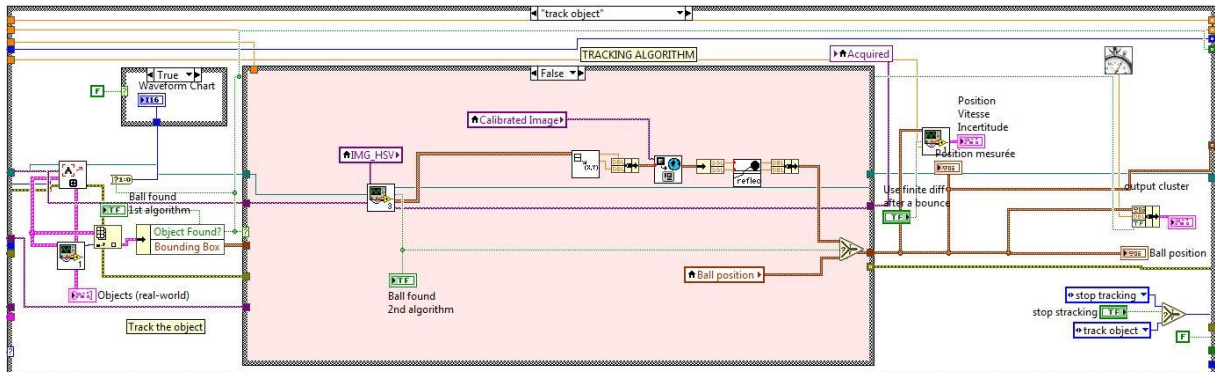


En premier lieu, les moteurs linéaires sont activés, ce qui permet de réaliser le homing des joueurs.

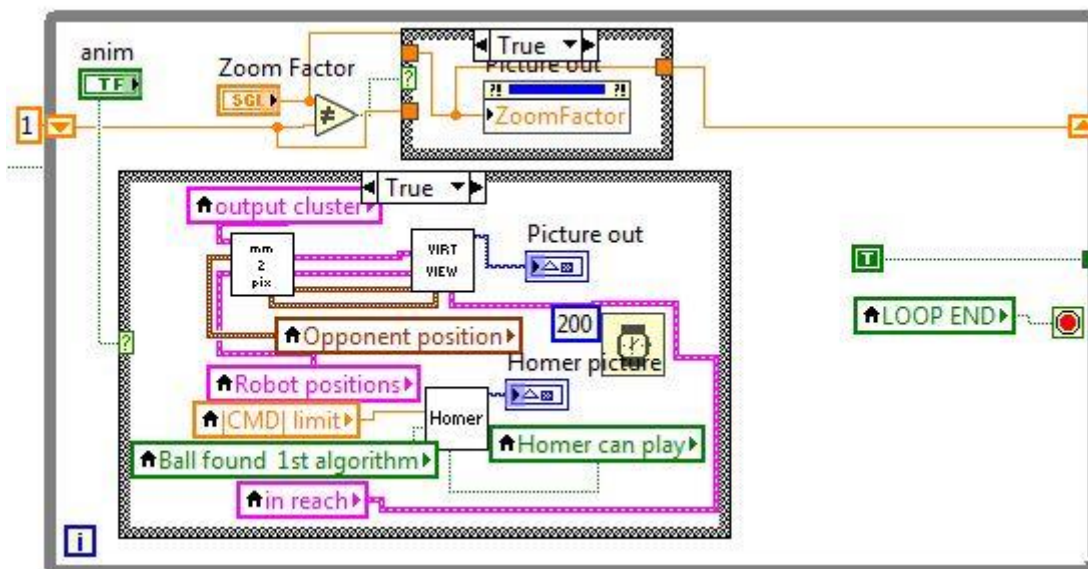


La première séquence permet d'isoler l'image de la balle ce qui servira de modèle pour toute l'exécution et permettra de la retrouver.

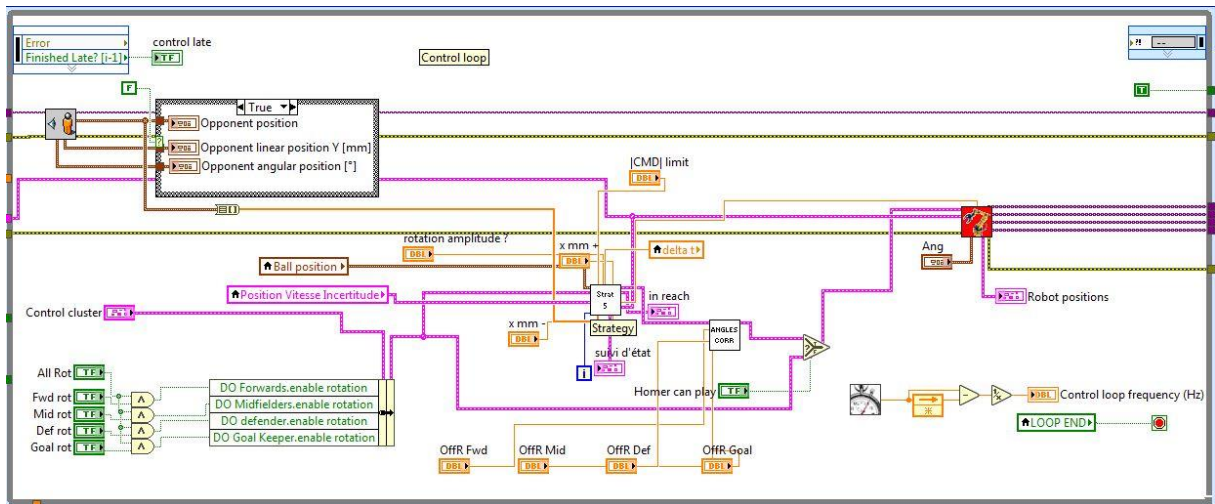
En outre, elle permet aussi de réaliser la calibration du système en corrigeant les distorsions sur l'image obtenue par la caméra. On peut aussi régler différents paramètres d'image sur le front panel : exposition, saturation et luminosité.



Cette deuxième partie du code se charge de détecter la position de la balle grâce à deux algorithmes. Le premier, plus rapide, est utilisé en premier lieu. Dans le cas où il est dans l'incapacité de trouver la balle, le deuxième algorithme, plus lent, entre en jeu. Les données sont ensuite stockées dans « Ball Position » qui est ensuite appelée dans plusieurs autres parties du VI, en particulier dans le VI stratégique.



Cette boucle gère plusieurs affichages sur le Front Panel : à travers les yeux d'Homer, une indication si, oui ou non, la balle a été détectée ; à travers la main de Homer, si les moteurs sont activés ; les positions latérales et angulaires des joueurs sur une représentation du terrain, ainsi que la position de la balle sur ce terrain.



Cette boucle implémente toute la stratégie du babyfoot. Elle se base sur les positions de la balle et des joueurs, et permet d'envoyer les commandes linéaires et de rotation au moteur.

3. Calibration de la vision

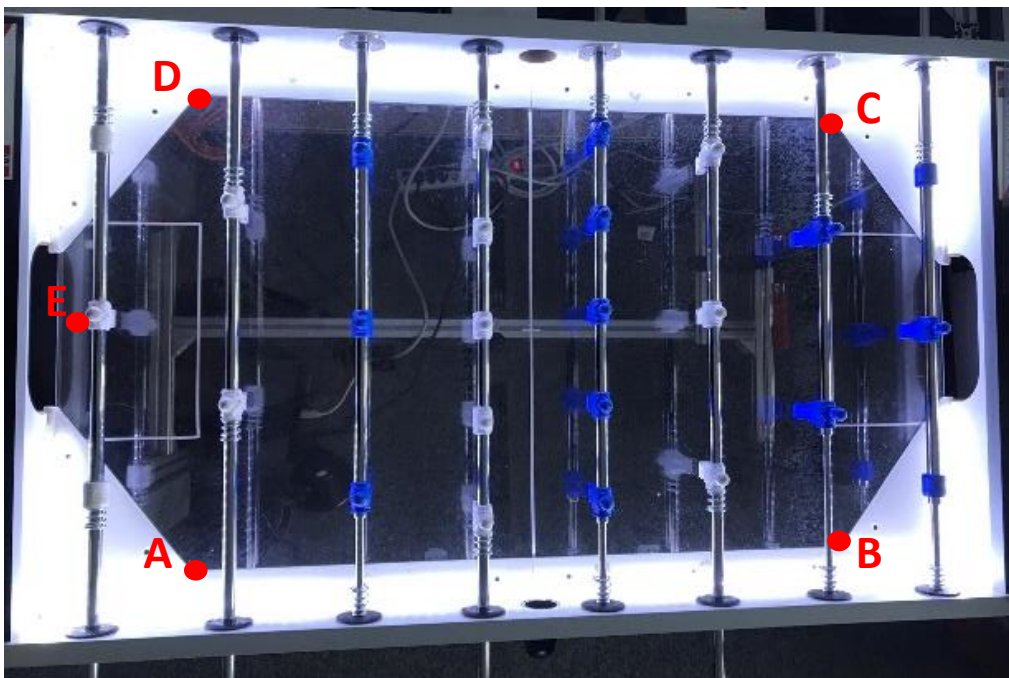
On peut voir la partie automatique du baby-foot (Vi+moteurs) comme le régulateur (*Controller*) du système « baby-foot » qui a pour entrées, les mouvements des joueurs, et pour sortie, la position de la balle. On comprend donc bien qu'une connaissance précise et sans délai de la position de la balle (l'entrée du régulateur) est indispensable pour que le régulateur fonctionne.

La mesure de la position est effectuée à l'aide d'une caméra. Cette méthode qui semble assez imprécise a priori, peut néanmoins donner des résultats tout à fait acceptables si les données sont bien traitées.

Au moment de commencer le projet, nous avons remarqué qu'il y avait parfois, des écarts importants entre la position des joueurs et celle de la balle. Notre première intuition a donc été de vérifier que le régulateur avait une bonne connaissance de la position.

a. Diagnostic et Mesure des erreurs

Grâce à une calibration de la vidéo, le Vi permet d'exprimer la position de la balle (en pixel sur l'image), en coordonnées physiques (mm) sur des axes x et y. Après inspection des différents sous-Vis (surtout le Vi stratégie) il est apparu qu'au cours des années, des termes de correction ont été ajoutés dans la stratégie pour prendre en compte le déplacement de l'origine des axes par rapport au point central du terrain. Ainsi, nous avons placé la balle en différents points du Baby-foot (voir schéma), et nous avons comparé les positions affichées à celles que nous mesurons. Voici les mesures que nous avons trouvées



Photographie du Baby-Foot vu du dessus

Point	Coordonnées réelles	Coordonnées dans le Vi
A	(-435,-305)	(-406,-260)
B	(435,-305)	(380,-245)
C	(435,305)	(379,275)
D	(-435,305)	(-413,293)
E	(526,~0)	(565,0)

Il nous est donc apparu évident que la mesure de la position était trop imprécise pour que le baby-foot fonctionne convenablement. Ainsi, il était impératif de reprendre et d'améliorer la correction de l'image.

b. Correction par Vision Assistant

Pour améliorer le positionnement de la balle, il faut d'abord comprendre comment marche cette calibration :

La caméra n'a pas de sens de la profondeur. Elle reçoit la lumière radialement ce qui a pour effet de projeter la surface, en fonction de l'angle sur la lentille. Ainsi un pixel de la caméra représente une plus grande surface au bord du baby-foot, par rapport à ce qu'un pixel représente au centre du baby-foot.

LabVIEW a une solution intégrée pour corriger ce genre de problème.

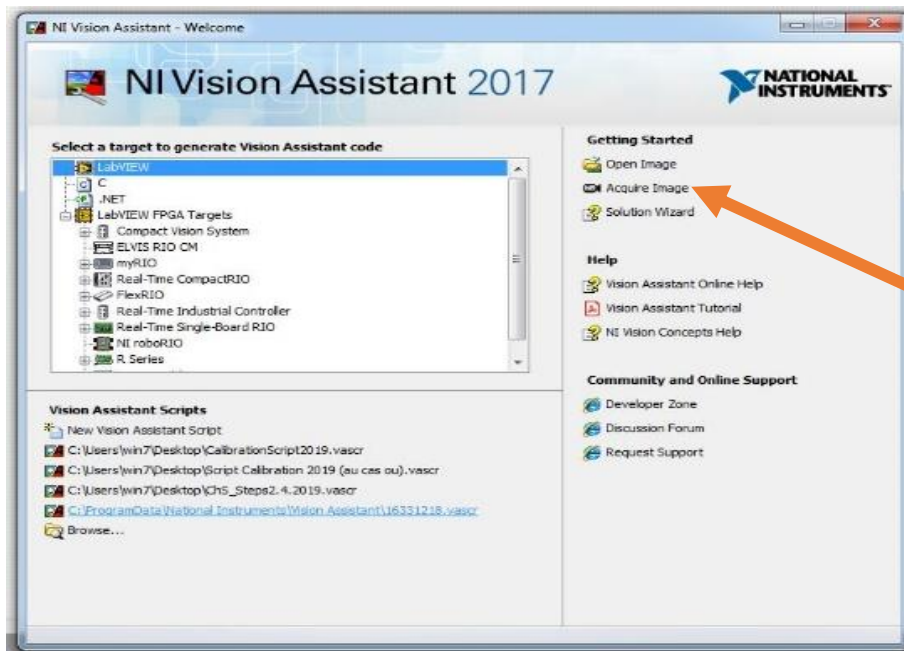
Il faut tout d'abord imprimer un pattern de calibration. Composé de cercles de tailles égales (qui seront représentés par au moins 3*3 pixels), répartis également (séparés d'une distance représentée par au moins 4 pixels) comme une grille, qui recouvre, idéalement, la surface entière du Baby-foot. La caméra ayant une résolution suffisante, en prenant les différentes dimensions à l'échelle du cm, on répondra largement aux exigences.

Nous avons dessiné la pattern à l'aide du logiciel « Adobe Illustrator » (même si PowerPoint ferait probablement l'affaire), puis nous l'avons imprimée (format A0).

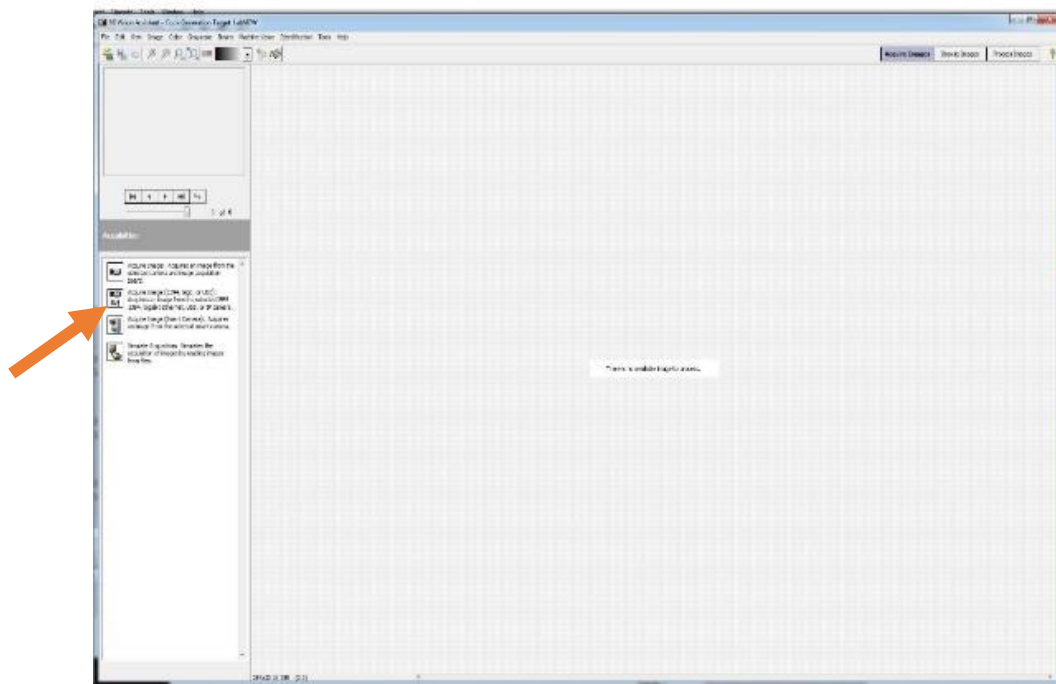
Une version du fichier est conservée dans le dossier du Baby-foot, dossier calibration.

On ouvre ensuite le Vi Vision Assistant. Attention à bien l'ouvrir à partir de l'exécutable ou d'un raccourci (et pas en l'ouvrant comme sous Vi dans le Vi principal car cela empêche l'utilisation de la caméra comme source de l'image).

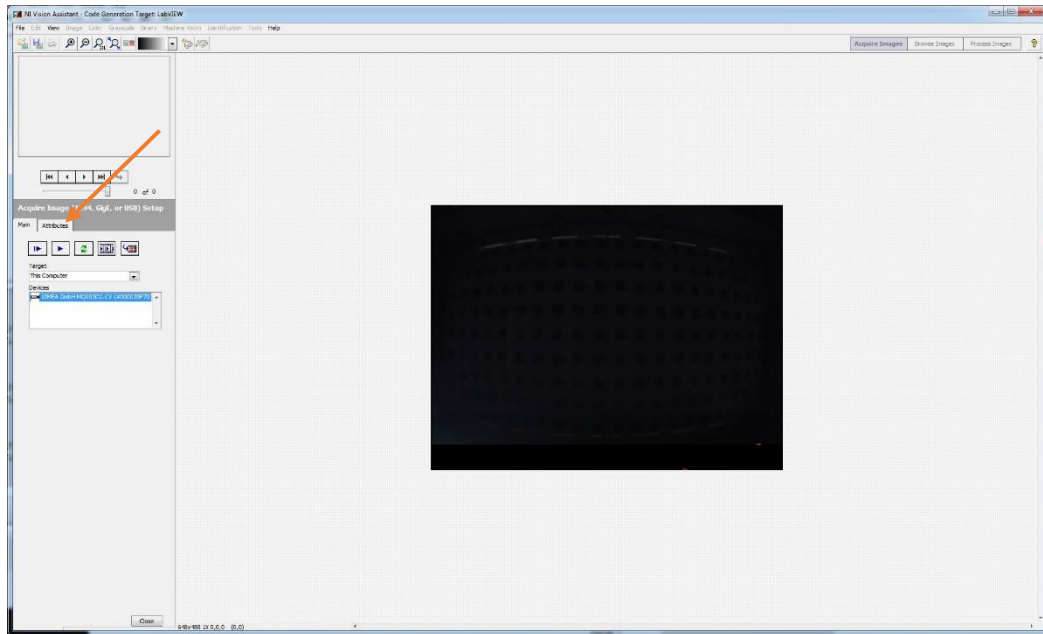
On suit ensuite ce tutoriel détaillé :



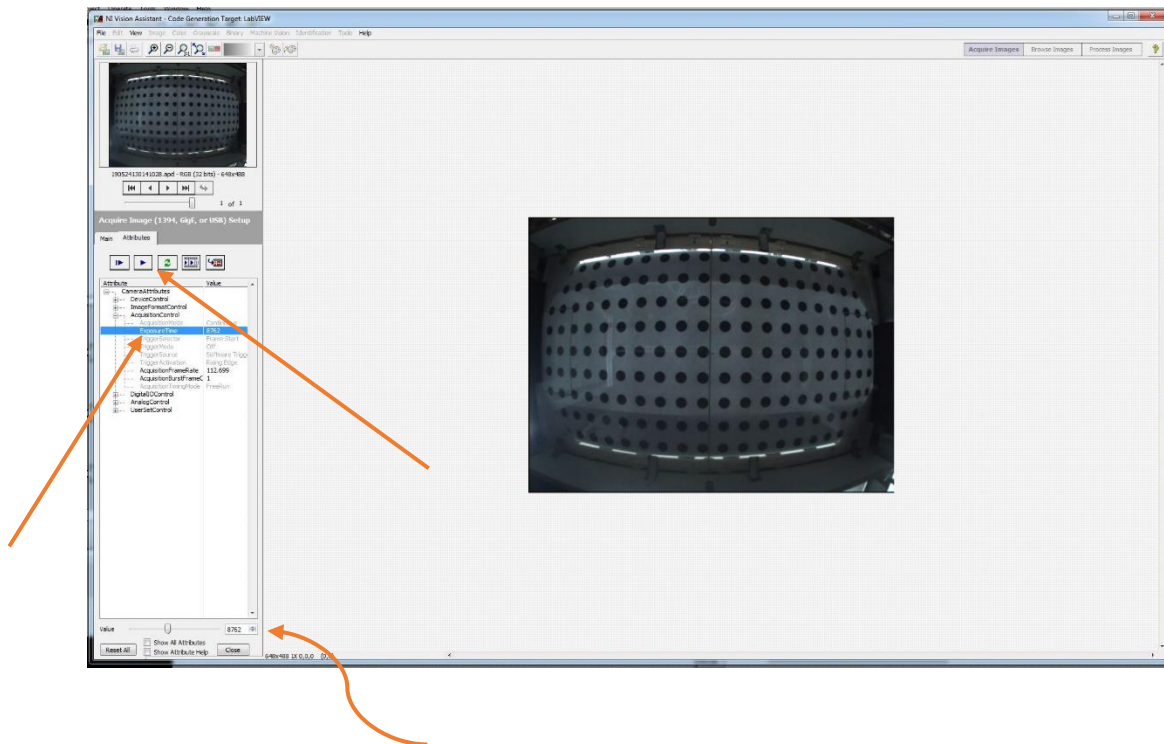
Step 1 : On commence par acquérir l'image sur laquelle on calculera les corrections



Step 2 : en vérifiant qu'on est bien dans l'onglet « Acquire images » en haut à droite. On sélectionne la 2^{ème} option d'acquisition comme indiqué.



Step 3 : on attend (une dizaine de seconde en général) que notre caméra s'affiche, on va ensuite dans l'onglet « attributes » pour modifier le temps d'exposition.



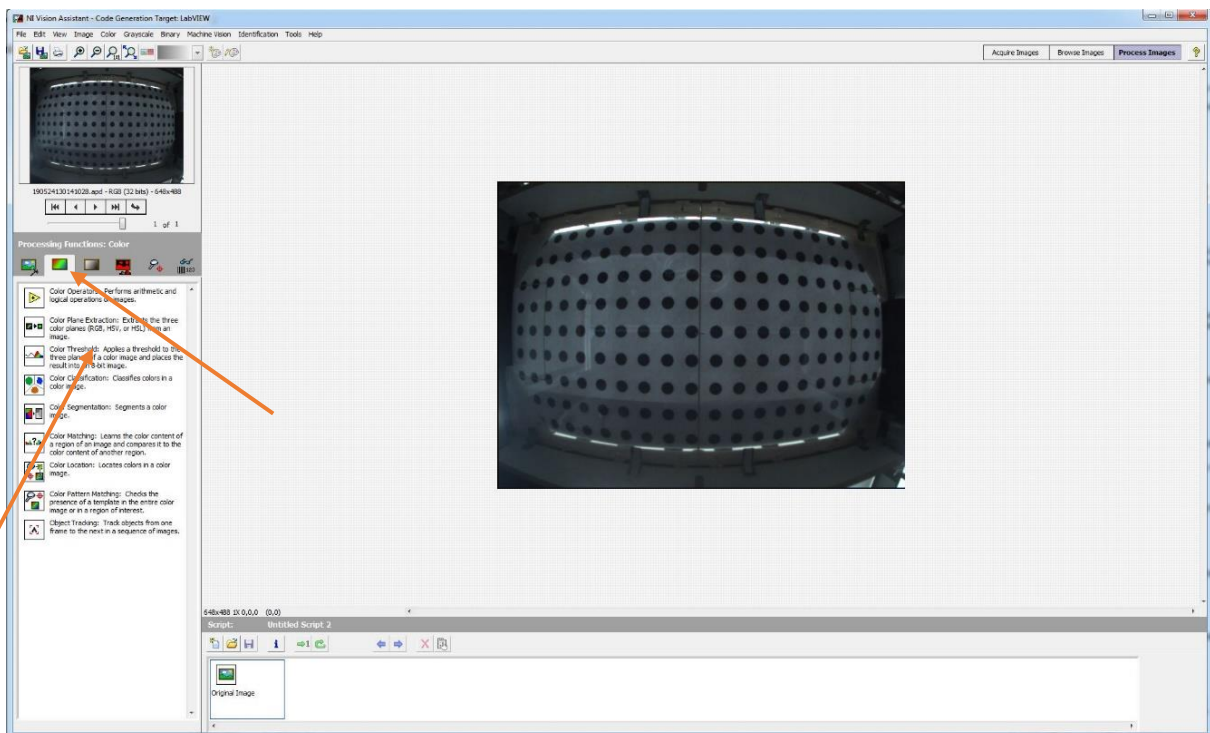
Step 4 : on clique sur la flèche pour faire une acquisition continue. Ce sera très utile pour voir, avec un très léger différé (équivalent grosso modo à la période d'acquisition), comment

l'image change en fonction des paramètres. On sélectionne ensuite dans « acquisition control » la ligne « exposure time », et on peut faire varier en bas de la fenêtre le temps d'exposition.

Attention ! c'est une étape cruciale pour la reconnaissance de la pattern. Plus l'exposition est grande, plus la caméra reçoit de lumière par image. Ceci permettra au logiciel de distinguer les cercles de la pattern du reste de l'image seulement si on est dans une fourchette d'exposition.

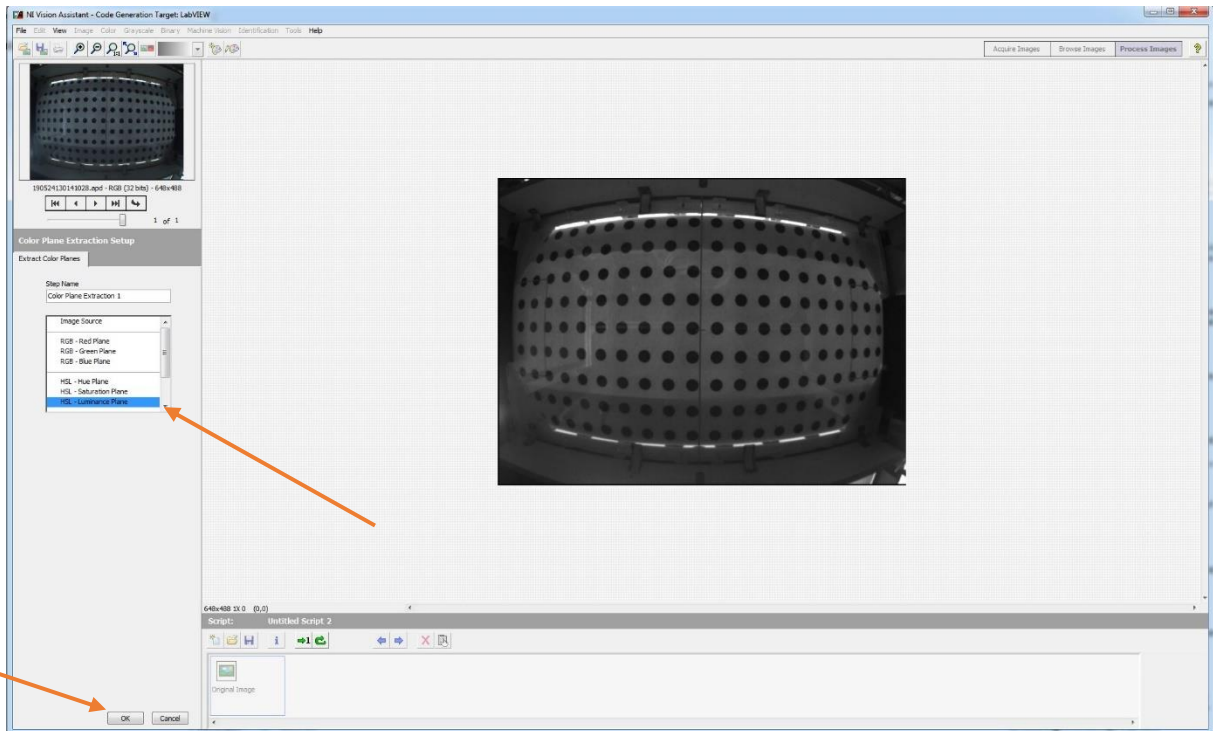
C'est l'étape la plus délicate de la calibration mais elle peut amener à de grosses erreurs si elle n'est pas réalisée correctement. On devra sans doute s'y reprendre à plusieurs fois, en essayant par exemple de bloquer tout éclairage extérieur, éteindre ou allumer les lumières de la salle, les LEDs etc.

Par ailleurs, c'est seulement à partir du Step 10 qu'on pourra savoir si l'image permet une bonne acquisition.

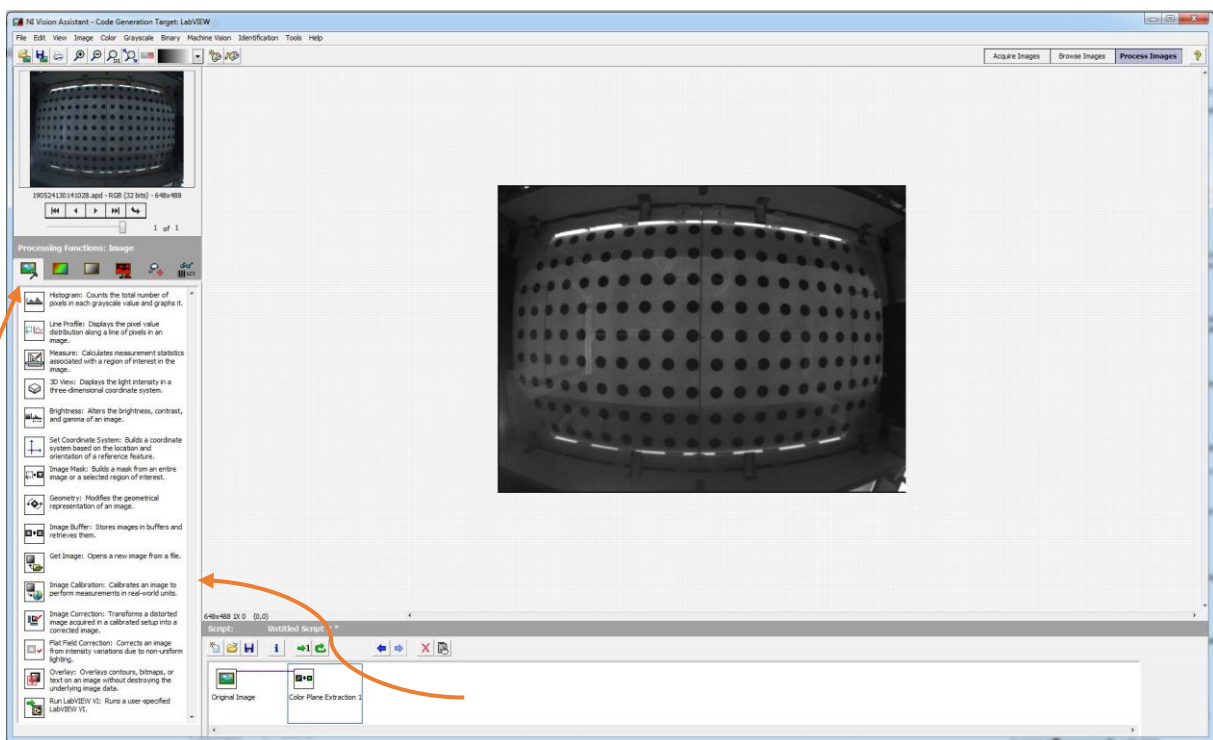


Step 5 : on applique ensuite sur l'image une sorte de filtre qui permet de traiter l'image en nuance de gris. Cette étape facilite grandement la reconnaissance des cercles, qui devient automatique.

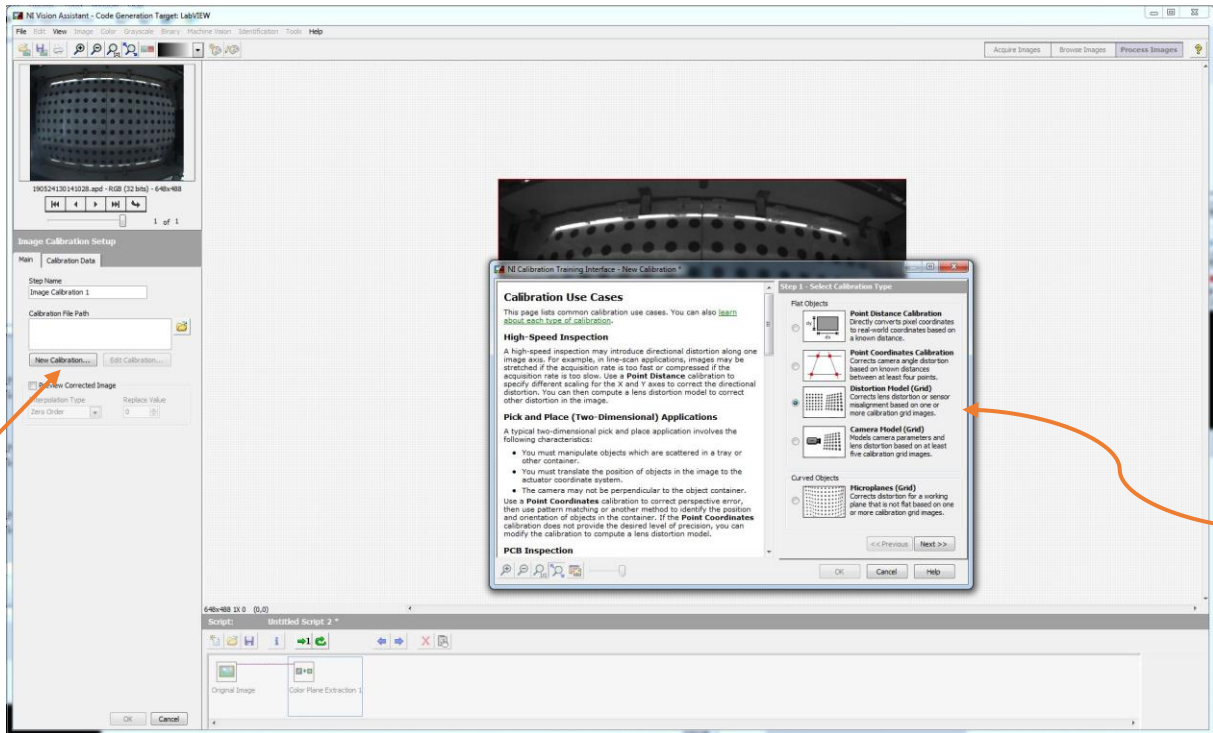
On choisit donc l'onglet « Processing Function : Color » et la 2^e ligne « Color Plane Extraction ».



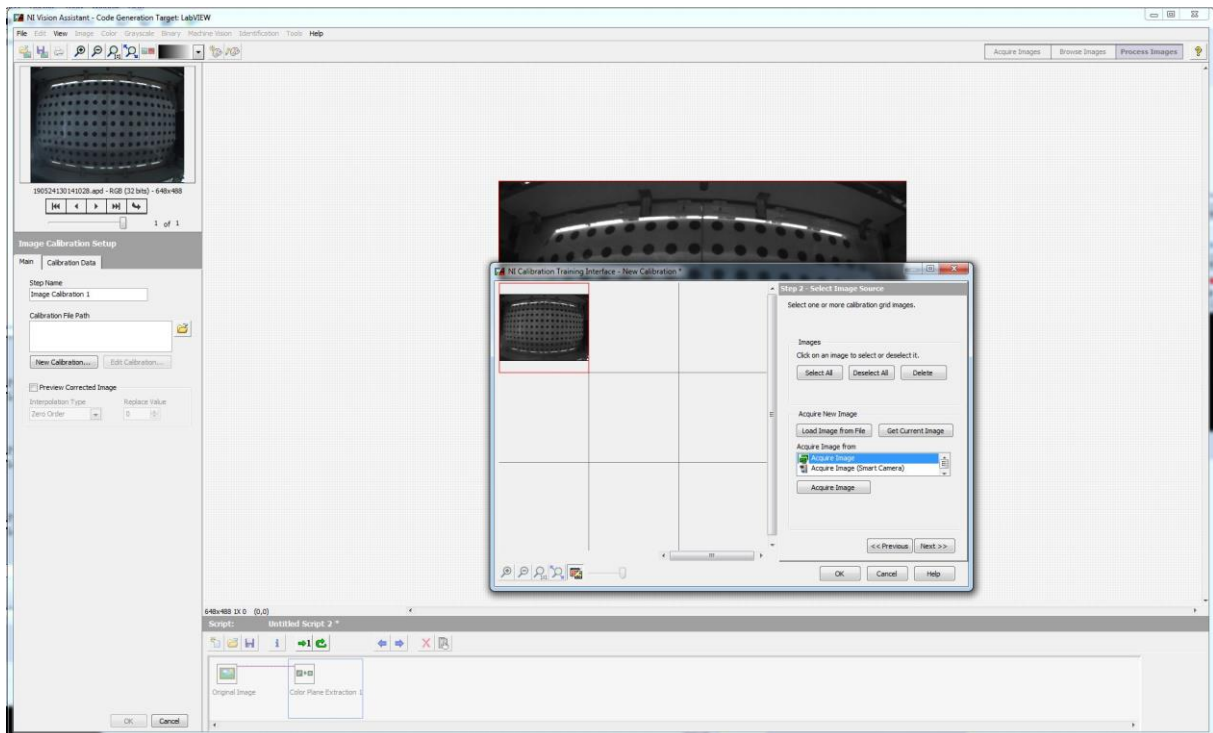
Step 6 : On sélectionne « HSL- Luminance Plane » et on clique sur OK



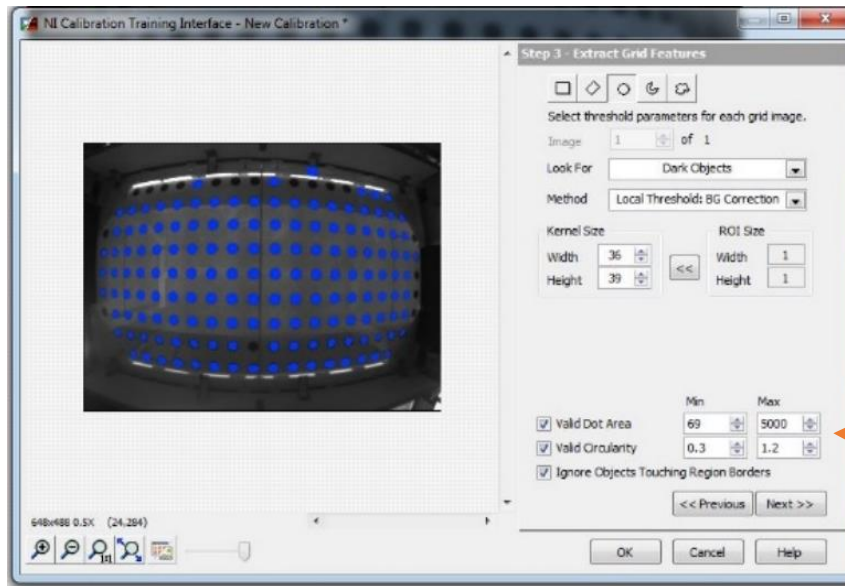
Step 7 : on s'attaque enfin à la calibration. On passe dans l'onglet « Processing Function : Image » et la ligne « Image Calibration »



Step 8 : on sélectionne « New Calibration » puis dans la fenêtre de calibration, « Distortion Model (Grid) », puis « next »

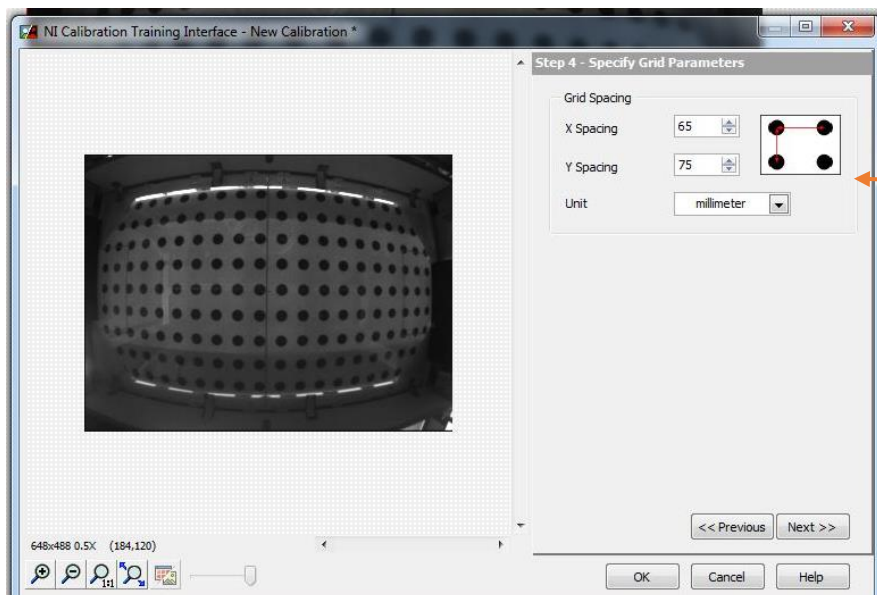


Step 9 : on vérifie que l'image est bien sélectionnée et on appuie ensuite sur « next ».

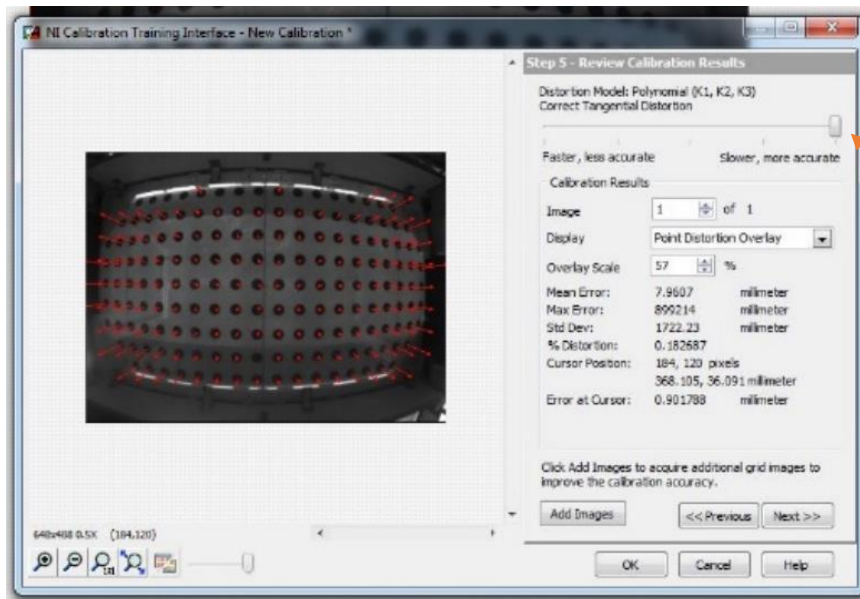


Step 10 : Dans cette étape, il s'agit de vérifier que tous (ou presque tous) les cercles de la pattern soient bien en bleu, et surtout qu'aucun autre point de l'image ne soit en bleu par erreur (cela pourrait avoir de grosses conséquences sur la formule de correction). Pour corriger ces problèmes, on peut jouer avec les paramètres, surtout ceux en bas. Notamment le min. « Valid Dot Area » qui peut être augmenté de façon assez conséquente pour exclure tous les points parasites, sans pour autant exclure nos cercles d'intérêt. Néanmoins, certaines images (comme celle prise dans cet exemple) ne pourront jamais produire une calibration satisfaisante, il faudra donc retourner au Step 4.

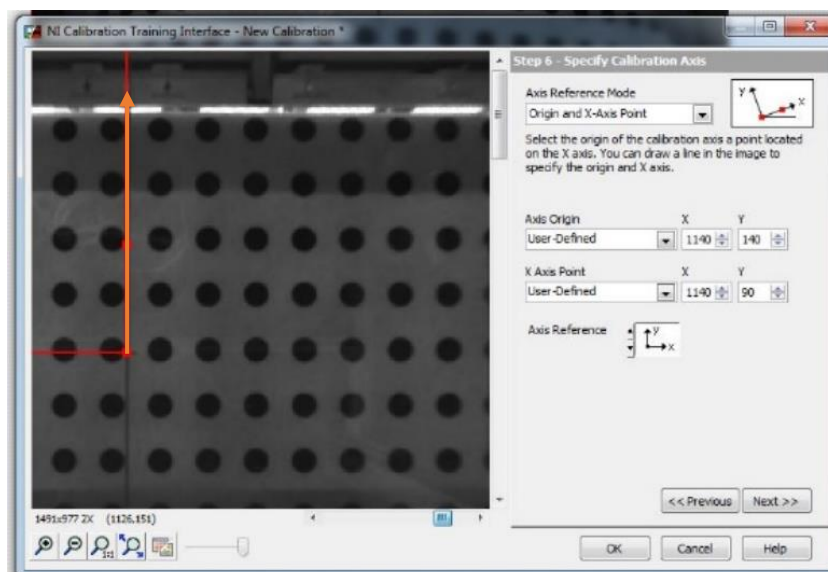
Enfin, quand l'image nous paraît satisfaisante, on appuie sur « next ».



Step 11 : il s'agit maintenant de définir les dimensions caractéristiques de la pattern. Comme on le voit sur l'image, les dimensions nécessaires sont les distances, centre à centre entre les rangées de ronds. Pour notre Pattern, nous avons mesuré 65mm en x et 75mm en y (attention à bien changer les unités de « pixels » à « millimeters »). « next »



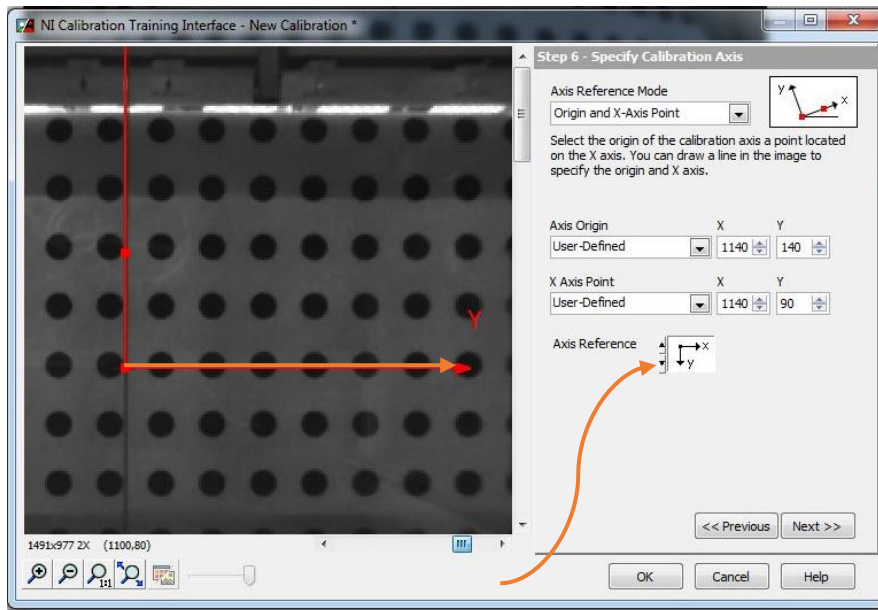
Step 12 : ici on choisit le modèle de distorsion qui sera utilisé. La transformation étant très rapide (que ce soit au moment du calcul des coefficients, ou la transformation des coordonnées de la balle en temps réel) on choisit le modèle polynomial qui est le plus exact malgré le fait que des précédents projets n'aient pas trouvé de différences significatives entre les différents modèles. On peut vérifier à cette étape si les différentes données de correction semblent correctes ou pas. « next »



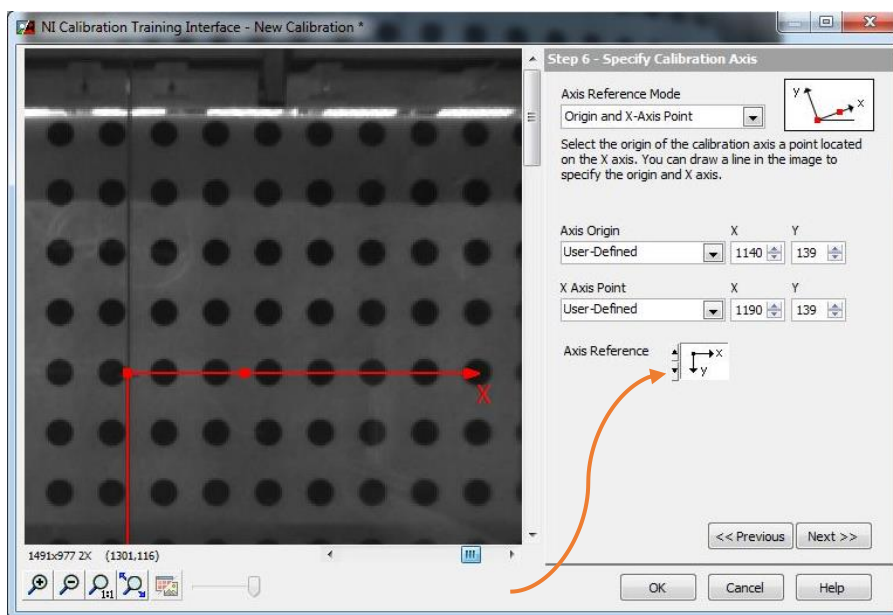
Step 13 : il faut maintenant placer le repère (son origine et son orientation). Le Vi principal du Baby-foot utilise un repère assez intuitif avec l'origine à la croix centrale (l'engagement), l'abscisse x dans la longueur (vers la droite sur l'image) et l'ordonnée y dans la largeur vers le haut.

Pour réussir à suivre précisément la longueur et la largeur, nous avons élaboré une procédure qui utilise le trait central, exactement vertical par rapport au Baby-foot :

Ainsi, on commence par un trait qui part de la croix et qui suit la ligne du milieu.

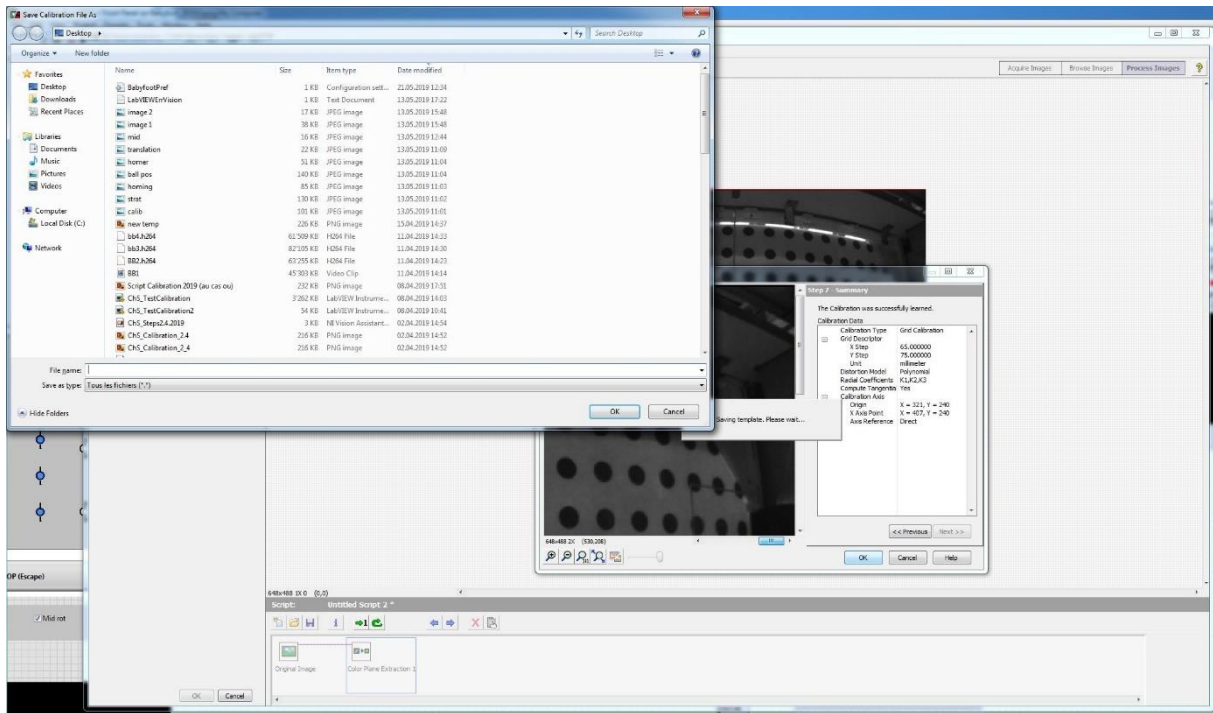


On change ensuite « axis reference » pour avoir l'axe y vers la droite. On l'utilise donc comme support pour tracer l'axe x.



Enfin, on change à nouveau « axis reference ». On a donc les x vers la droite et les y vers le haut.

« next »



On clique enfin sur OK et on enregistre pour pouvoir utiliser cette image dans le Vi principal.

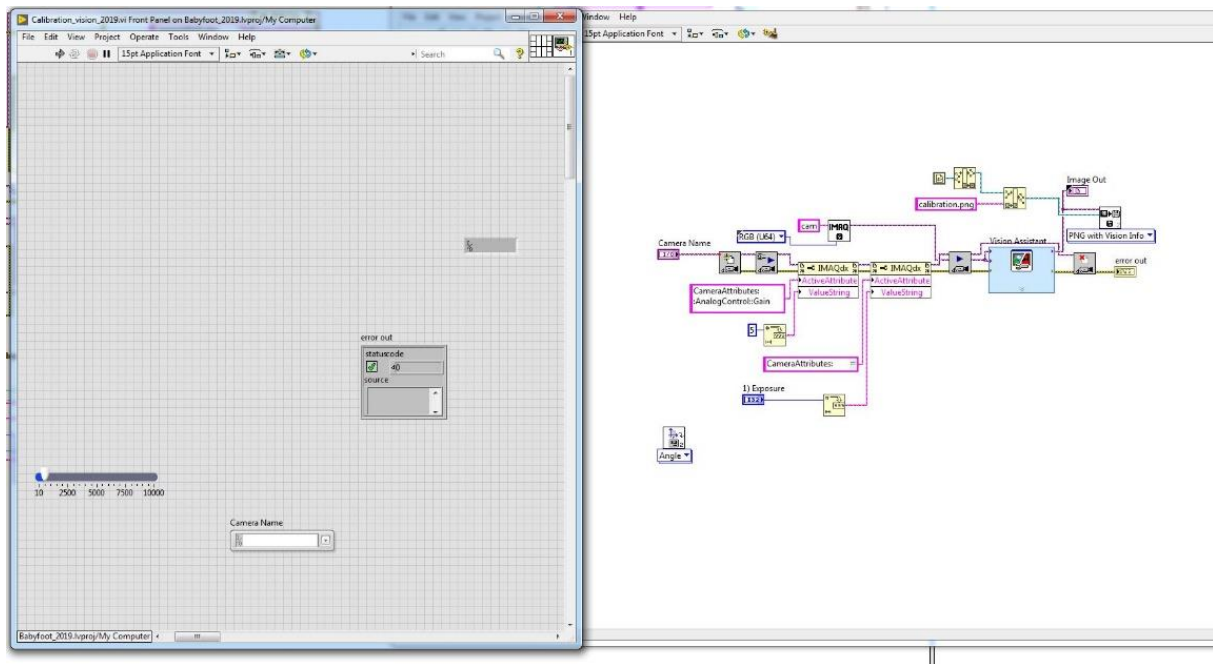
c. Implémentation dans le programme

Ainsi, la calibration sert à déplier la « peau d'orange » captée par la caméra. Les données de calibration (selon notre compréhension, il s'agit essentiellement du polynôme de calibration) sont stockées comme des sortes de métadonnées dans le PNG en sorti. L'image du PNG en sorti n'est pas elle-même corrigée, mais en appliquant dans LabVIEW la correction en mémoire dans le PNG sur l'image du PNG lui-même, on peut retrouver l'image corrigée.

d. Etablissement d'une procédure automatique et améliorations futures

Voyant l'importance du travail qu'a représenté la compréhension de la procédure de calibration, nous avons essayé de rendre automatique la procédure entière. Ainsi, il ne serait plus nécessaire de comprendre le fonctionnement et de prendre 20 minutes par calibration, il faudrait simplement installer la pattern sur le baby-foot, éventuellement régler l'exposition de la caméra, lancer le Vi, et tout serait fait (avec l'image calibrée déjà placée au bon endroit pour que le Vi principal puisse l'utiliser).

On peut imaginer qu'ainsi, on puisse faire une calibration très rapidement, à chaque fois qu'on déplace le baby-foot, ou même qu'on le démarre en début de journée. Voici une esquisse de ce à quoi devrait ressembler le diagramme du Vi qui permettrait une telle intégration (elle est sauvegardée dans le dossier calibration : « Calibration_vision_2019 ») :



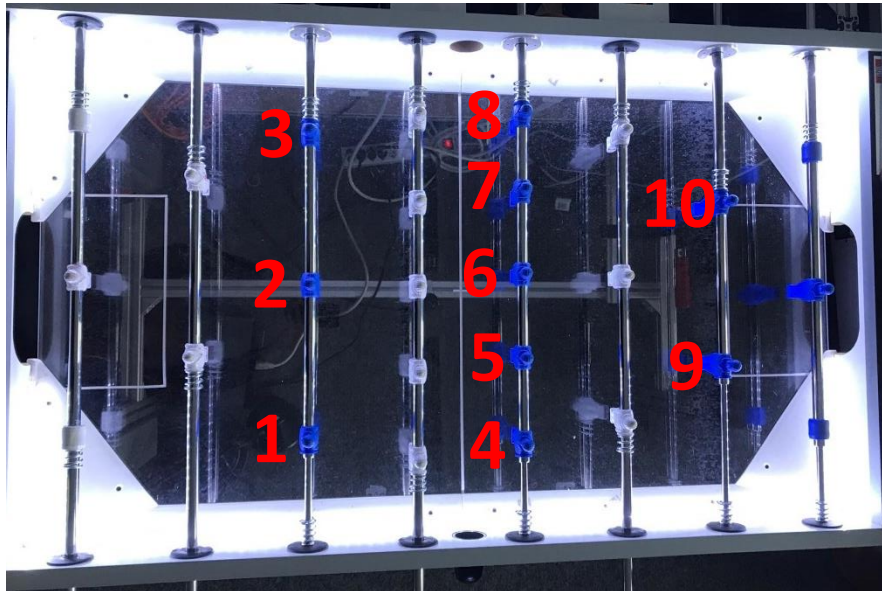
L'élément clé vient en fait de l'ouverture du sous-Vi « NI Vision Assistant », qui programme à l'avance les opérations que l'on souhaite appliquer à l'image entrante.

Il reste néanmoins un problème que nous n'avons pas pu résoudre : la définition de l'orientation du repère ainsi que de son centre nécessite encore d'être faite à la main. C'est une étape essentielle pour la calibration qui met un frein important à notre espoir d'automatisation.

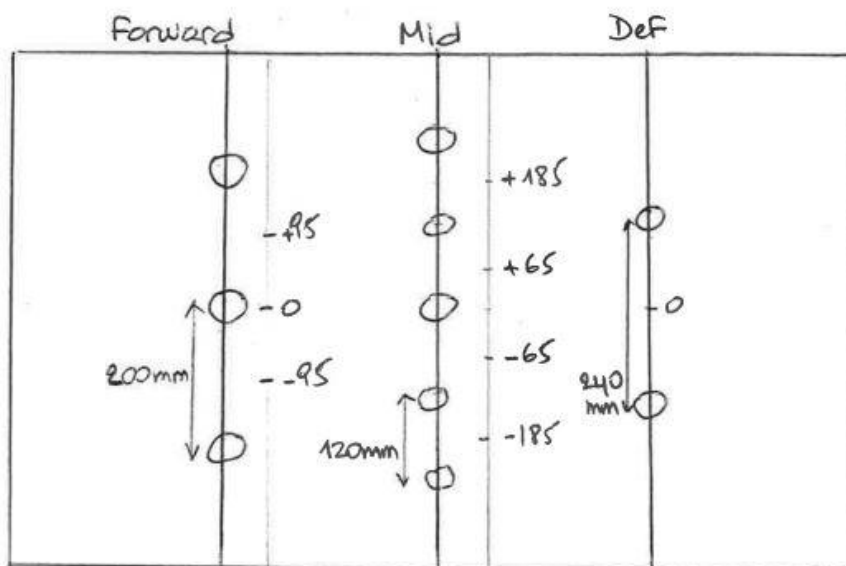
Néanmoins, un futur développement du baby-foot pourrait essayer de tirer profit de la reconnaissance de forme, intégrée à LabVIEW, qui servirait à repérer le centre et l'orientation exacte du baby-foot grâce à la croix.

4. Correction des translations

Afin d'opérer en translation, il faut d'abord assigner à chaque joueur une partie de la largeur (zone d'action), puis selon la position de la balle, un changement de coordonnées est fait par rapport au joueur concerné. Cela permet de toujours avoir un joueur devant la balle.

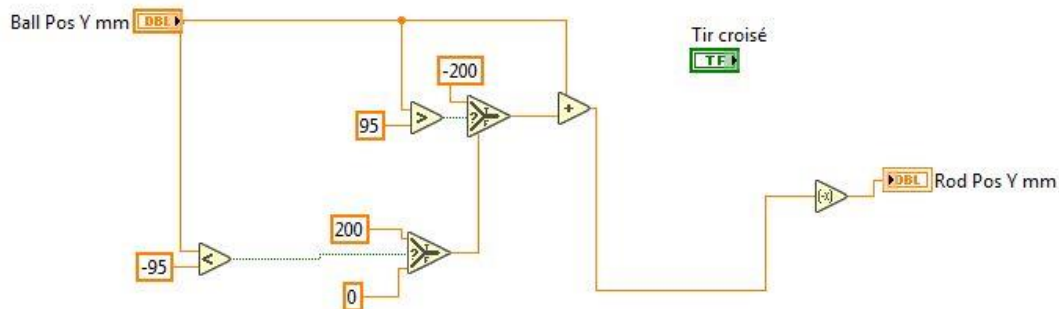


Numérotation des joueurs



Les différentes mesures pour la correction

a. Attaquants : *joueurs 1, 2 et 3.*



Ce VI est basé sur la position en Y de la balle. Différents cas se présentent.

Si la condition « $Y < -95$ » est vraie, la fonction retourne la valeur 200 correspondant à la distance entre deux joueurs.

- Dans ce cas-là, la condition « $Y > 95$ » est forcément fausse et la deuxième fonction retourne 200. La barre prend alors la position $Y + 200 \text{ mm}$ et le joueur 1 se positionne alors devant la balle

Si la condition « $Y < -95$ » est fausse, la fonction retourne la valeur 0 et deux cas se présentent :

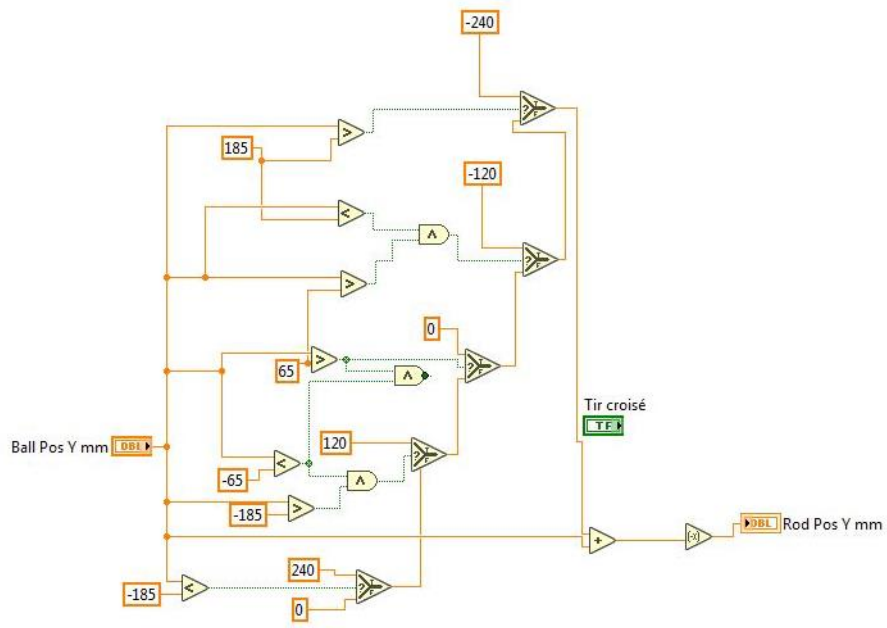
- « $Y > 95$ » est vraie. La fonction retourne -200. La barre prend alors la position $Y - 200 \text{ mm}$ et le joueur 3 frappe la balle.
- « $Y > 95$ » est faux. Cela implique que $-95 < Y < 95$. La fonction retourne 0. La barre prend alors la position Y . C'est le joueur 2 qui se positionne devant la balle.

b. Milieu : *joueurs 4, 5, 6, 7, 8.*

Pour les milieux, c'est plus compliqué. On utilise des fonctions ET : La fonction ET ne renvoie la valeur Vrai que si les deux entrées sont définies à Vrai.

La distance entre les joueurs est de 120 mm . Ainsi, plusieurs cas se présentent :

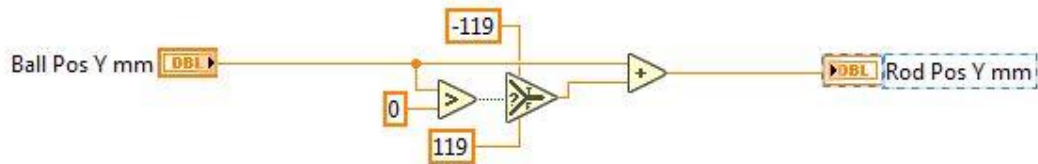
- Si $Y < -185$: la barre prend la position $Y + 240 \text{ mm}$. Le joueur 4 se positionne devant la balle.
- Si $-185 < Y < -65$: la barre prend la position $Y + 120 \text{ mm}$. Le joueur 5 se positionne devant la balle.
- Si $-65 < Y < 65$: la barre prend la position Y . Le joueur 6 se positionne devant la balle.
- Si $65 < Y < 185$: la barre prend la position $Y - 120 \text{ mm}$. Le joueur 7 se positionne devant la balle.
- Si $Y > 185$: la barre prend la position $Y - 240 \text{ mm}$. Le joueur 8 se positionne devant la balle.



c. Défenseurs : *joueurs 9, 10*.

La distance entre les deux joueurs est de 240 mm. Deux cas se présentent :

- $Y > 0$: la barre prend la position $Y - 119$ mm. Le joueur 9 se positionne devant la balle.
- $Y < 0$: la barre prend la position $Y + 119$ mm. Le joueur 10 se positionne devant la balle.



5. Rotation

Le VI chargé du contrôle de la rotation contenant trop de constantes (et trop peu de documentation), il nous a été très difficile de le comprendre et encore plus de l'améliorer. Néanmoins, nous avons saisi l'idée générale du VI :

Les moteurs sont contrôlés en position, ainsi afin d'effectuer la rotation, à chaque itération de la boucle, une commande est envoyée afin d'incrémenter la position angulaire des joueurs jusqu'à atteindre 360 degrés, ce qui correspond à un tour complet. Ce tour est divisé en plusieurs zones avec différents angles maximum. Une fois que le joueur atteint l'angle maximum d'une zone, il passe à la zone suivante. Les divisions par 360 permettent d'obtenir des angles modulo 360 degrés

Comme le couplage entre les moteurs et les joueurs est fait par frottement, il crée des décalages. La commande de rotation qui repose sur la position initiale avant le début de la rotation, cause une accumulation d'erreur, les joueurs ne sont plus verticaux à la position angulaire 0.

En conclusion, nous devons admettre que malgré de grands efforts déployés (notamment la capture par un espion artificiel de la consigne de rotation), nous ne sommes pas capables d'expliquer le fonctionnement exact de la rotation. Néanmoins, les prochains groupes qui s'y attèleront pourront utiliser les commentaires que nous avons ajoutés, surtout pour essayer de comprendre les 3 boucles de rétroaction booléennes.

6. Contrôle

Jusque-là les moteurs sont contrôlés en position, c'est-à-dire que nous donnons simplement la position désirée aux drivers des moteurs, et les moteurs s'occupent du reste. Cela implique que nous ne contrôlons pas plusieurs paramètres. Entre autres, un homing (établissement des translations tolérées et de la position du centre) est nécessaire à chaque démarrage des moteurs. De la même manière, nous ne pouvons pas vraiment décider des paramètres du contrôleur au niveau du Vi pour avoir la réponse que nous attendons en sortie des moteurs (bien que ces paramètres puissent peut-être être modifiés au niveau des drivers des moteurs). Ainsi, nous avons des problèmes d'hypersensibilité qui causaient des grandes vibrations à tout l'appareil. Ces vibrations accélèrent probablement les problèmes d'usure.

Nous avons donc commencé avec Monsieur Salzman à changer la commande pour donner comme consigne aux moteurs une vitesse plutôt qu'une position. Nous avons cependant rencontré un problème : ne sachant pas quelle était la réponse exacte (rapidité, sensibilité, stabilité...) que nous souhaitions avoir en sortie des moteurs, il nous a été compliqué de concevoir un contrôleur approprié.

7. Conclusion

En conclusion, nous pouvons dire que nous sommes satisfaits des modifications que l'on a apportées au Babyfoot surtout au niveau de la translation. Avant de nous soucier d'ajouter des nouvelles fonctionnalités au Babyfoot, il nous a fallu reculer d'un pas, revenir à la base du jeu et corriger ce qui n'allait pas.

En effet, après quelques tests, nous nous sommes rendu compte que la distorsion des distances causée par la caméra était mal corrigée par le VI. Il a donc fallu refaire la calibration depuis le début.

Un autre problème majeur résidait au niveau de la translation de la barre. La base du jeu est qu'à tout instant, un joueur doit se trouver en face de la balle, ce qui n'était pas le cas. Pour chacune des trois barres principales (attaquants, milieux, défenseurs), nous avons donc défini une zone d'action pour chaque joueur à l'aide des outils de comparaison de LabVIEW. Désormais, un joueur assure la balle à tout moment du jeu.

Pour la suite du projet Babyfoot, il reste évidemment de nombreux domaines d'amélioration : Tout d'abord, régler le problème des différentes calibrations (homing moteur et repérage de la balle) récurrentes. Il faudrait essayer de comprendre le mécanisme de rotation des joueurs et peut-être prendre mieux en compte le décalage dû au couplage. Comme mentionné, on pourrait également automatiser la procédure de calibration avec la pattern. Une autre amélioration à apporter au jeu serait d'implémenter la stratégie du tir croisé. Ainsi, le Babyfoot jouera plus intelligemment et se rapprochera de plus en plus du modèle idéal du Babyfoot toujours gagnant.