



Projet de licence Travaux Encadré de Recherche



Il tire et il marque... Le rêve secret de Robotino

Second Semestre

BRAMERET Pierre-Antoine
CHEMIN Alexandre

Encadré par Jean-Marc ROUSSEL

Année 2008-2009

Département Génie Mécanique de l'ENS de Cachan

Soutenance le vendredi 12 juin

PREFACE

La société FESTO propose le système Robotino comme support didactique. Le laboratoire d'automatique du département Génie Mécanique désirent connaître le potentiel de ce robot pour des activités d'enseignement nous a demandé d'évaluer ses possibilités techniques à travers le projet suivant : Apprendre à Robotino à jouer au foot.

Pour jouer au foot, Robotino doit tout d'abord repérer la position des objets (ballon et cages), s'approcher du ballon et shooter. Pour cela, Robotino est équipé d'une caméra (résolution 1024x768) et de 3 modules de translation (disposés à 120°).



Le projet de TER qui nous a été proposé est à l'intersection des disciplines de la mécanique et de l'informatique industrielle : quels sont les programmes qui doivent être implantés pour permettre à Robotino de suivre une trajectoire donnée ?

Sommaire

I) Problématique.....	4
II) Résultats premier semestre.....	5
1) Choix de V_{max}	7
2) Choix de A_{max}	8
3) Choix de K	8
III) Reconstruction de l'environnement : théorie	9
1) Stéréoscopie.....	9
2) Calcul d'incertitude.....	10
IV) Reconstruction de l'environnement : mise en oeuvre.....	12
1) Choix d'une méthode de reconnaissance de formes	12
2) Choix d'une stratégie de déplacement	13
3) Choix d'une stratégie de tir.....	14
V) Conclusion	16

I) Problématique

Le projet de TER qui nous a été confié consiste à programmer le Robot autonome proposé par la société Festo pour qu'il soit capable de jouer au foot.



A travers ce projet de TER, le laboratoire d'Automatique du Département Génie Mécanique souhaite évaluer le potentiel de ce robot autonome pour ses activités d'enseignement, de concours ou de promotion. Ainsi « Apprendre à Robotino à jouer au foot » est un projet de TER à plusieurs facettes car il nous est demandé :

- d'étudier la cinématique du robot pour savoir comment il peut se déplacer,
- de déterminer ses capacités maximales de déplacement,
- de programmer Robotino pour qu'il soit capable de jouer au foot.

L'étude du premier semestre était portée sur ses possibilités de déplacement. Durant le second semestre, nous avons fini cette étude et nous nous sommes intéressés aux possibilités de traitement des images et à la stratégie de tir de Robotino.

L'objectif que nous nous sommes fixé est que Robotino soit capable de marquer un but en ayant sa caméra "cachée" après avoir pris seulement deux ou trois photos de la situation. Pour cela il est nécessaire de :

- savoir reconstruire l'environnement du robot avec le minimum de prises de vue
- trouver une stratégie de tir optimale
- appliquer cette stratégie pour mettre un but

II) Etude des déplacements : résultats complémentaires à ceux présentés au premier semestre

Nous avons consacré le premier semestre à l'étude des possibilités de déplacement de Robotino. Le résultat de cette période de travail a été un asservissement en position d'une précision de l'ordre du mm sur un déplacement de deux mètres. L'asservissement dépendait de cinq valeurs que nous n'avions pas optimisées : deux paramètres de la chaîne cinématique, Vmax (vitesse maximale souhaitée de Robotino), Amax (accélération maximale souhaitée de Robotino) et la constante de correction K (voir le rapport du premier semestre partie IV.1).

Nous les avons donc déterminées avant d'entamer l'élaboration de la stratégie de tir.

1) Paramètre général responsable de tout déplacement

Ce paramètre que l'on nomme CRANS_PAR_METRE est en fait le coefficient qui permet de convertir les tops du capteur de chaque moteur en distance parcourue par chaque roue. C'est le paramètre qui prend en compte tous les aspects de la cinématique. Il sert lorsque que l'on met en place la structure d'asservissement, car il permet de convertir la distance que l'on veut faire parcourir à chaque roue en unité du capteur de position, et on peut ainsi asservir la position. Nous avons utilisé une estimation rapide au premier semestre, pour que les déplacements s'effectuent de manière répétable. Nous avons obtenu une valeur moyenne de déplacement de 198,8 cm au lieu des 200,0 cm de consigne, et nous avons évoqué que la valeur du paramètre devait être mesurée

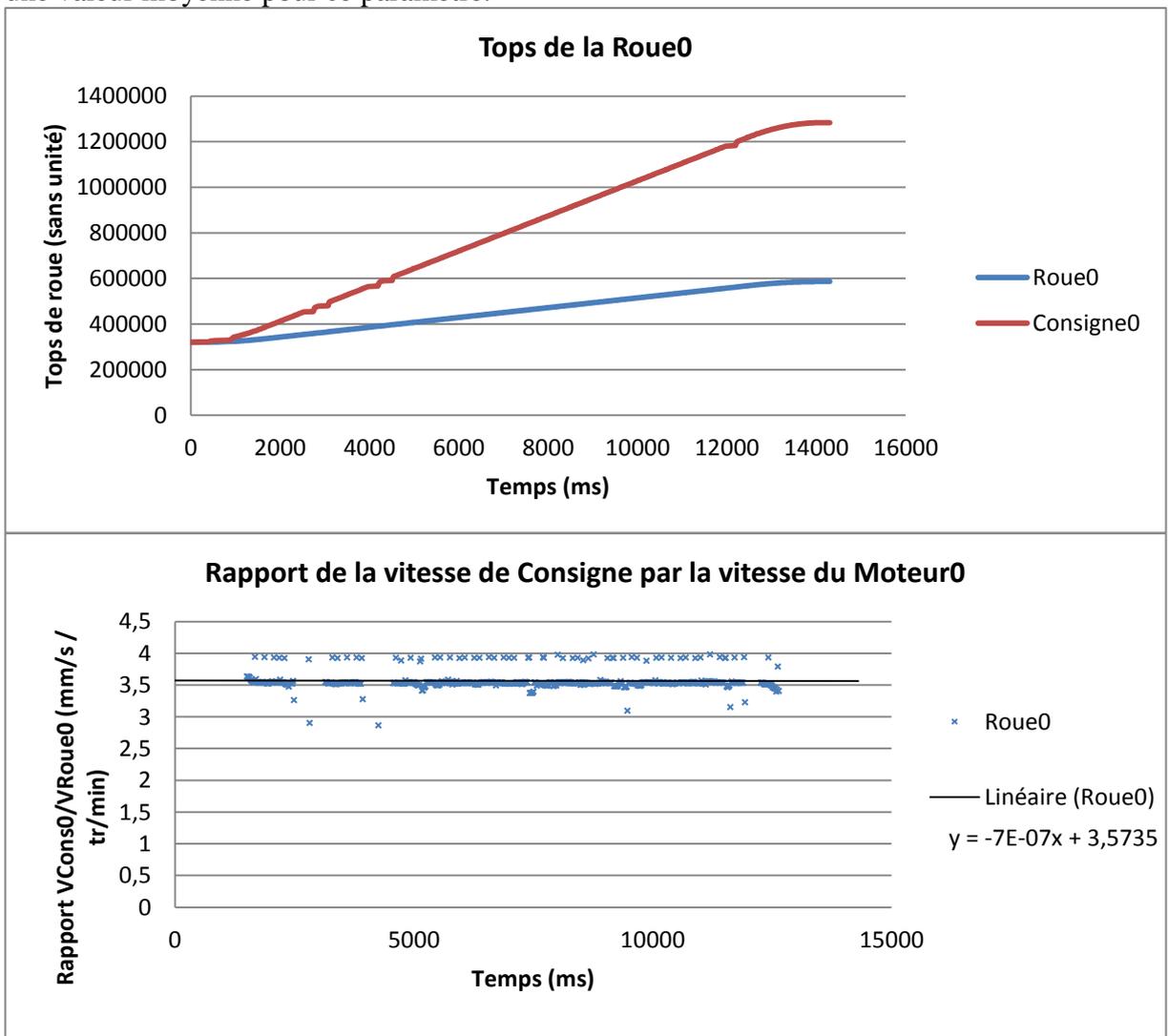
Pour faire la mesure, nous avons fait déplacer Robotino sur un nombre de tops du capteur de position, puis nous avons mesuré la distance parcourue. Nous avons pris un nombre suffisamment grand pour couvrir nos besoins en précision.

Nous avons obtenu 129374,441 tops par mètre

2) Paramètre utilisé pour la vitesse

Même si l'asservissement que nous avons mis en place permet de corriger la vitesse de rotation des roues si elle s'éloigne de la consigne, il nous faut un paramètre pour convertir la vitesse de Robotino en vitesse de rotation pour chaque moteur, qui est en tours par minute (unité imposée par l'interface de Robotino). Ce paramètre intervient également dans la chaîne cinématique, mais utilise un paramètre géométrique difficile à appréhender avec précision, le rayon effectif des roues, c'est à dire la mesure de la distance du centre de rotation d'une roue avec la ligne de contact du sol, et ce pour un tour de roue complet.

Pour effectuer cette mesure, nous avons demandé à Robotino de faire un déplacement quelconque (avec ce paramètre estimé) sans asservissement, mais dont nous connaissons la vitesse de consigne théorique, puis nous avons calculé à partir des échantillons discrets la vitesse envoyée au moteur, et la vitesse de consigne. Nous avons pu ensuite faire le rapport des deux et enfin en trouver une valeur moyenne pour ce paramètre.



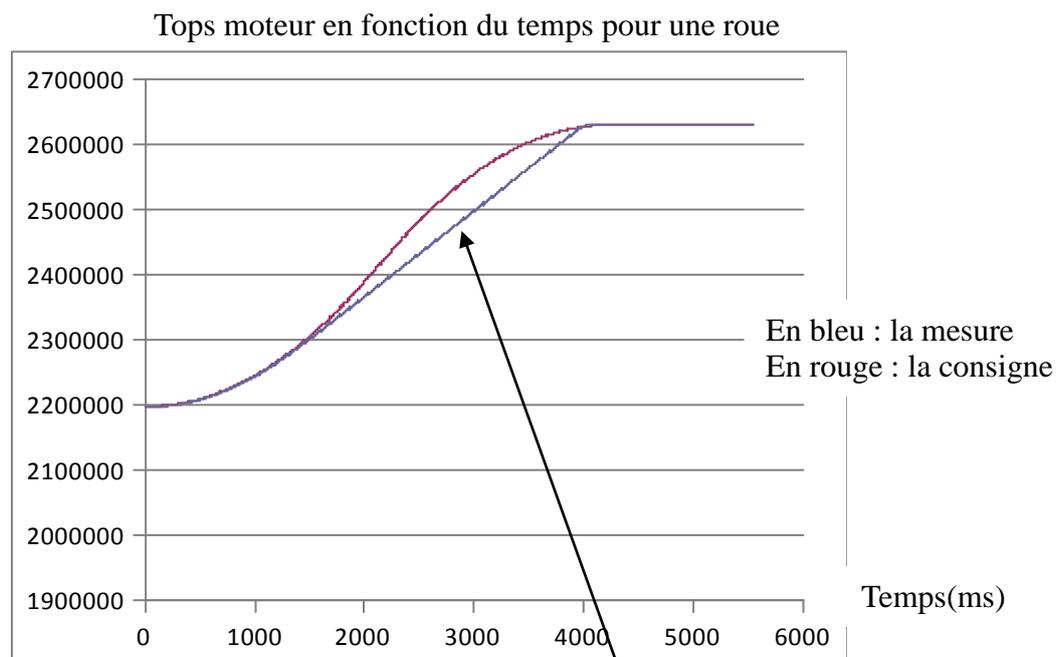
3) Choix de V_{max} .

Nous avons choisi de prendre pour V_{max} une vitesse légèrement inférieure à la vitesse maximale pouvant être atteinte par Robotino.

Pour déterminer cette vitesse, nous avons poussé Robotino hors de ses limites puis nous avons mesuré la vitesse effective. Concrètement nous avons envoyé une consigne ayant un V_{max} très élevé et observé sur le relevé des positions des roues de Robotino les vitesses de rotation maximales admissibles pour chacune d'entre elles.

La vitesse maximale obtenue pour une roue est donc de 1530 mm/s

La vitesse maximale que Robotino peut atteindre est donc de 1325 mm/s (résultat étude cinématique premier semestre).



On a donc choisi V_{max} à : 500 mm/s

Cette zone correspond à la vitesse maximale atteinte par une roue de Robotino

4) Choix de Amax

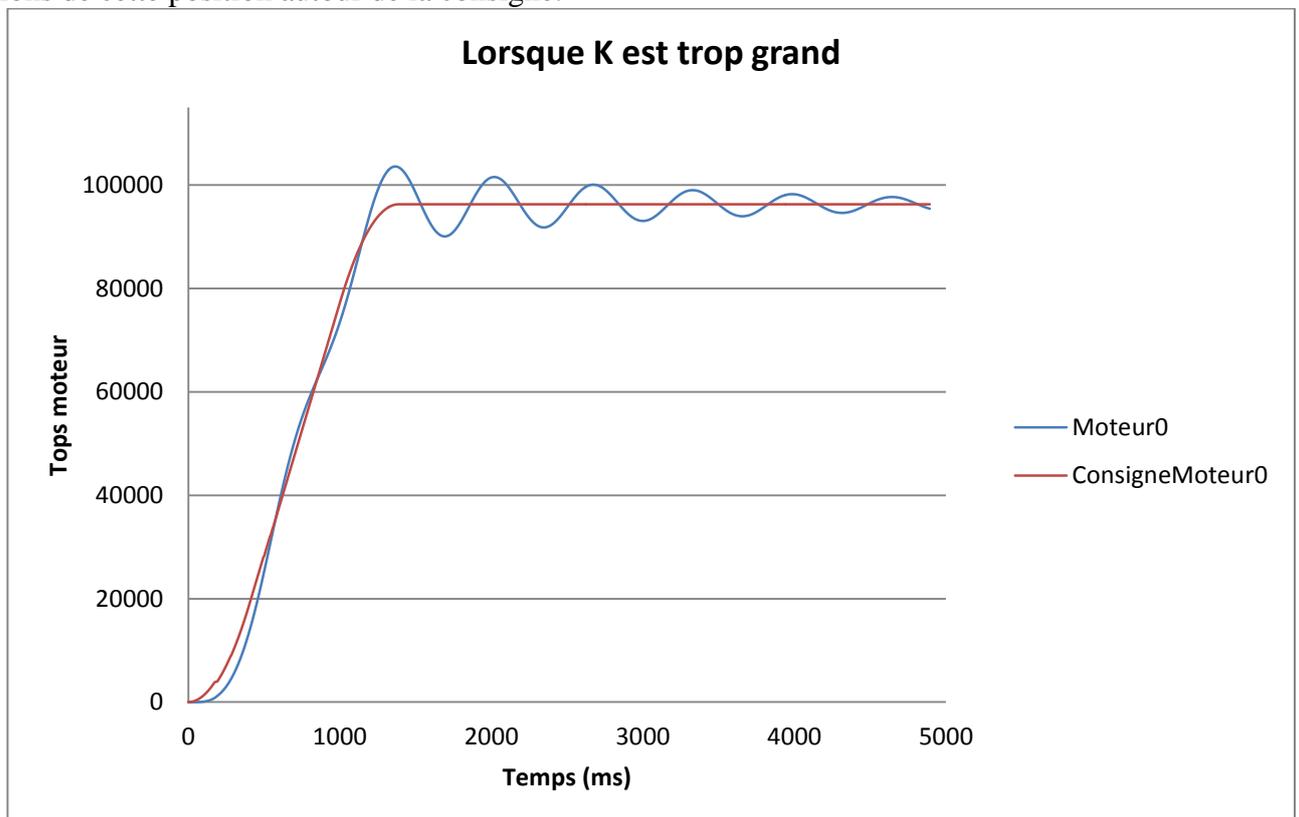
Nous voulions un paramètre Amax légèrement inférieur à l'accélération maximale qui permet un déplacement correct de Robotino. C'est à dire une accélération qui n'implique aucun dérapage des roues au démarrage et au freinage de Robotino.

Pour déterminer cette accélération, nous avons progressivement augmenté le paramètre Amax jusqu'à obtenir un dérapage des roues. La valeur obtenue (avec toute la précision qu'il est possible d'avoir, car la répétabilité du phénomène est faible) est 500 mm/s^2

Nous avons donc choisi un Amax de 300 mm/s^2

5) Choix de K

Le choix de K a été un peu plus difficile car il a fallu faire un compromis. En effet, un K élevé implique une plus grande précision sur la position finale de Robotino, mais il induit des oscillations de cette position autour de la consigne.



Nous avons mis la priorité sur l'obtention d'un déplacement sans oscillation car nous souhaitons un déplacement sans dépassement de la consigne.

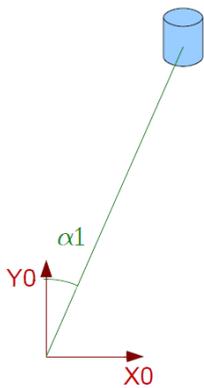
La valeur optimale de K trouvée respectant ces conditions est $K = 10,0$

III) Reconstruction de l'environnement : théorie

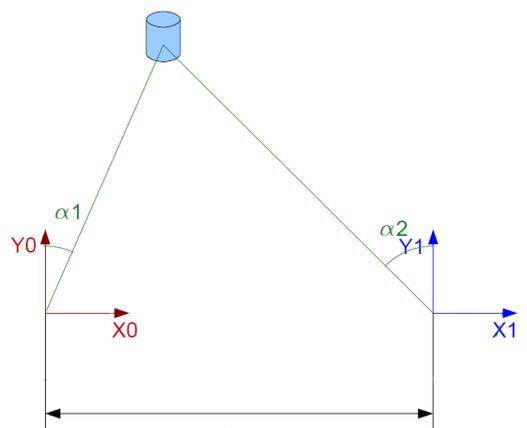
L'objectif que nous nous sommes fixé impose à Robotino de pouvoir reconstruire son environnement à partir du moins de photos possibles.

1) Stéréoscopie

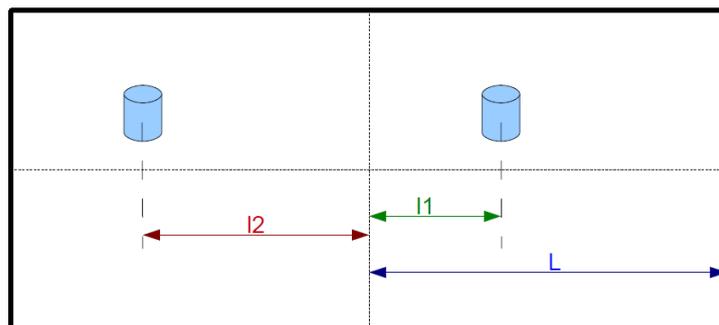
Le procédé que nous allons utiliser pour cela est la stéréoscopie. Il s'agit du même procédé qui nous permet une vision en trois dimensions avec nos deux yeux : deux images d'une même scène provenant de deux points de vue suffisent à situer dans l'environnement les différents objets.



Prise de vue n°1 (repère 0)



Prise de vue n°2 après un déplacement X selon X_0 (repère 1)



Paramètres utiles sur une prise de vue
 L'indice 1 correspond à la prise de vue n°1
 L'indice 2 correspond à la prise de vue n°2

Robotino est maintenant capable de reconstruire son environnement à partir de quelques photos. Il nous reste cependant à savoir avec quelle précision il situe les objets, car il faut être suffisamment précis pour marquer le but de ses rêves.

2) Calcul d'incertitude

Lors de ses mouvements et du traitement de l'information, quelques imprécisions sont commises. Le but est de réussir à quantifier l'erreur effectuée sur le positionnement des objets dans l'environnement de Robotino en fonction de ces imprécisions, pour ensuite trouver le jeu de paramètres permettant de minimiser celle-ci.

Nous avons pour cela introduit les imprécisions dans les formules et obtenu l'expression de l'erreur sur la position.

$$\begin{aligned}
 &> \mathbf{dx2} := \mathbf{x2} - \mathbf{x2th}; \\
 &\mathbf{dy2} := \mathbf{y2} - \mathbf{y2th}; \\
 dX2 &:= ((l2 + dl2) L \tan(\alpha) (X + dX)) / (L^2 \sin(\phi) + \tan(\alpha) l1 \cos(\phi) L + \tan(\alpha) dl1 \cos(\phi) L - \tan(\alpha) L \cos(\phi) l2 - \tan(\alpha) L \cos(\phi) dl2 \\
 &\quad + \tan(\alpha)^2 l1 \sin(\phi) l2 + \tan(\alpha)^2 l1 \sin(\phi) dl2 + \tan(\alpha)^2 dl1 \sin(\phi) l2 + \tan(\alpha)^2 dl1 \sin(\phi) dl2) - \frac{l2 X}{l1 - l2} \\
 dY2 &:= (L^2 (X + dX)) / (L^2 \sin(\phi) + \tan(\alpha) l1 \cos(\phi) L + \tan(\alpha) dl1 \cos(\phi) L - \tan(\alpha) L \cos(\phi) l2 - \tan(\alpha) L \cos(\phi) dl2 + \tan(\alpha)^2 l1 \sin(\phi) l2 \\
 &\quad + \tan(\alpha)^2 l1 \sin(\phi) dl2 + \tan(\alpha)^2 dl1 \sin(\phi) l2 + \tan(\alpha)^2 dl1 \sin(\phi) dl2) - \frac{L X}{\tan(\alpha) (l1 - l2)}
 \end{aligned}$$

- dX : erreur sur le déplacement de Robotino*
- φ : erreur sur la position angulaire de Robotino*
- dl1 : erreur commise sur la mesure de l1*
- dl2 : erreur commise sur la mesure de l2*
- dX2 : erreur sur la position de l'objet en abscisse*
- dY2 : erreur sur la position de l'objet en ordonnée*

Afin d'être en mesure d'évaluer l'impact de chacun des paramètres sur l'erreur que l'on obtient pour la position d'un objet, nous avons construit le vecteur incertitude dont les six composantes sont les dérivées partielles de dX2 et dY2 par rapport à l1, l2 et X.

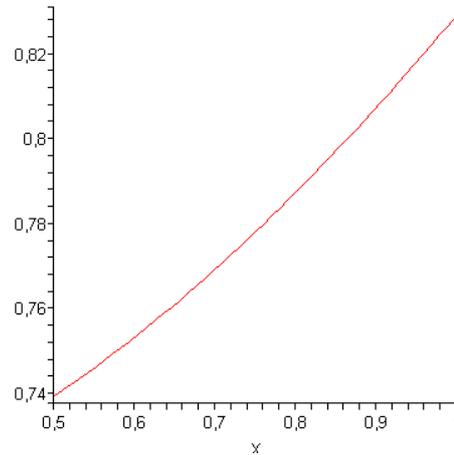
La norme de ce vecteur nous permet donc de quantifier l'impact global d'un jeu de paramètres (l1, l2, X). Nous cherchons donc le jeu de paramètres minimisant la norme du vecteur incertitude.

Nous avons utilisé le logiciel de calcul formel Maple pour résoudre ce problème de minimisation.

$$\begin{aligned}
 &> \mathbf{impact} := \mathbf{sqrt}(\mathbf{sum}(\mathbf{listeIncertainite}[\mathbf{i}]**2, \mathbf{i}=1..6)); \\
 \mathbf{impact} &= \sqrt{\frac{l2^2}{(l1 - l2)^2} + \frac{l2^2 X^2}{(l1 - l2)^4} + \left(\frac{X}{l1 - l2} + \frac{l2 X}{(l1 - l2)^2}\right)^2 + \frac{l2^2 X^2 (1 + l1 l2)^2}{(l1 - l2)^4} + \frac{1}{(l1 - l2)^2} + \frac{X^2}{(l1 - l2)^4}} \\
 &> \mathbf{Minimize}(\mathbf{impact}, \mathbf{x}=0.5..3, \mathbf{l2}=-1..1, \mathbf{l1}=-1..1, \mathbf{initialpoint} = \{\mathbf{X}=10, \mathbf{l1} = -0.1, \mathbf{l2} = 0.1\}); \\
 &\quad [0.739509972887452016, [\mathbf{X} = 0.500000000000000000, \mathbf{l1} = -1., \mathbf{l2} = 1.]]
 \end{aligned}$$

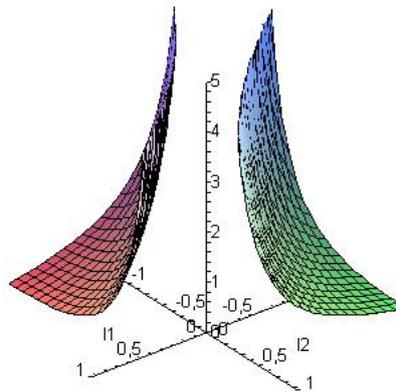
Le jeu de paramètres idéal obtenu est le suivant :

- X le plus petit possible
- $\text{abs}(l1-l2)=2*L$. C'est à dire que lors des prises de vue, l'objet doit se situer le plus près possible de chaque bord.



Erreur en fonction de X pour $l1 = 1$ et $l2 = -1$

On constate bien que X doit être le plus faible possible pour $l1$ et $l2$ fixés



Erreur en fonction de $l1$ et $l2$ pour X fixé.

Le minimum est atteint en deux points : $l1=1$ et $l2=-1$, ou $l1=-1$ et $l2=1$

Nous utiliserons ces informations pour définir une stratégie lors de la prise d'images.

IV) Reconstruction de l'environnement : mise en oeuvre

1) Choix d'une méthode de reconnaissance de formes

Nous utiliserons l'"œil" de Robotino pour acquérir des images, puis nous les traiterons sur l'ordinateur le commandant, pour permettre une analyse suffisamment rapide.

Nous nous sommes renseignés sur Internet pour mieux connaître différentes méthodes de reconnaissance de formes, puis nous avons discuté avec un contact au sein du LMT de l'ENS. Il nous a conseillé d'utiliser la librairie OpenCV, qui propose une mise en œuvre optimisée des méthodes sus-mentionnées.

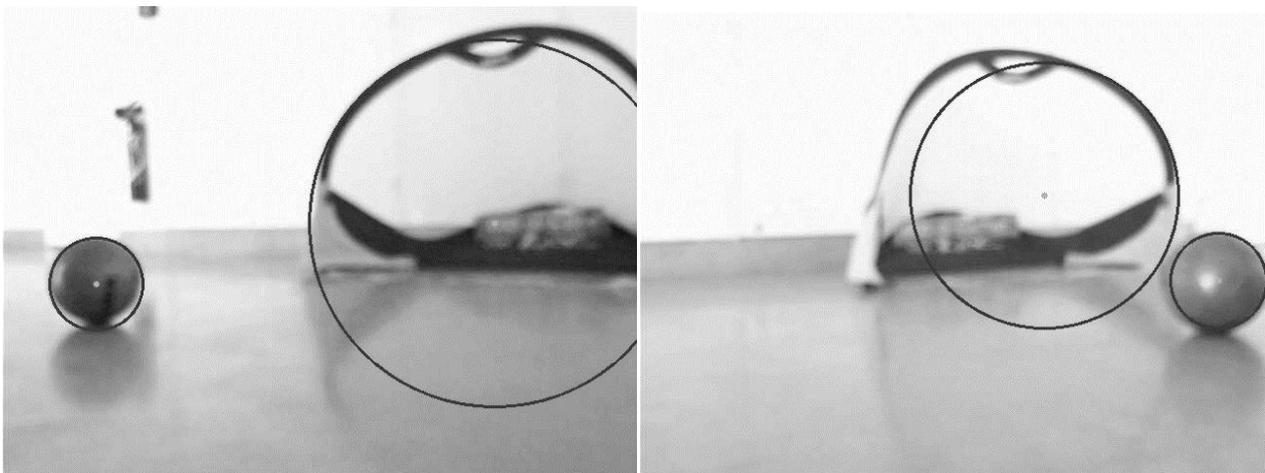
Pour les formes que nous pouvons reconnaître, nous avons principalement les cercles (par transformation de Hough appliquée pour les cercles) et les droites (par transformation de Hough). Nous obtenons pour les droites : une distance depuis un point de l'image et l'inclinaison (système de coordonnées polaires). Pour les cercles, nous avons la position du centre et le rayon, ce qui est rapidement utilisable.

Nous avons donc choisi de prendre un embut circulaire, car on peut utiliser plus facilement les informations issues de l'analyse que dans le cas d'une cage standard.

Dans ces conditions, on utilisera la méthode de Hough pour les cercles, et on obtiendra la position du centre de la cage sur l'image, et la position du centre du ballon.

Nous avons considéré que dans la plupart des cas, Robotino est suffisamment loin du ballon pour que celui-ci apparaisse plus petit que la cage sur l'image, et c'est ainsi que nous avons différencié les deux dans nos analyses.

Les photos sont passées en noir et blanc avant l'analyse. Ici on peut voir le résultat avec les cercles mis en évidence.



Notre première approche est de considérer que le plus grand cercle est la cage alors que le plus petit est le ballon.

2) Choix d'une stratégie de déplacement

Les contraintes imposées par notre objectif nous limitent à prendre seulement deux ou trois photos pour reconstruire l'environnement.

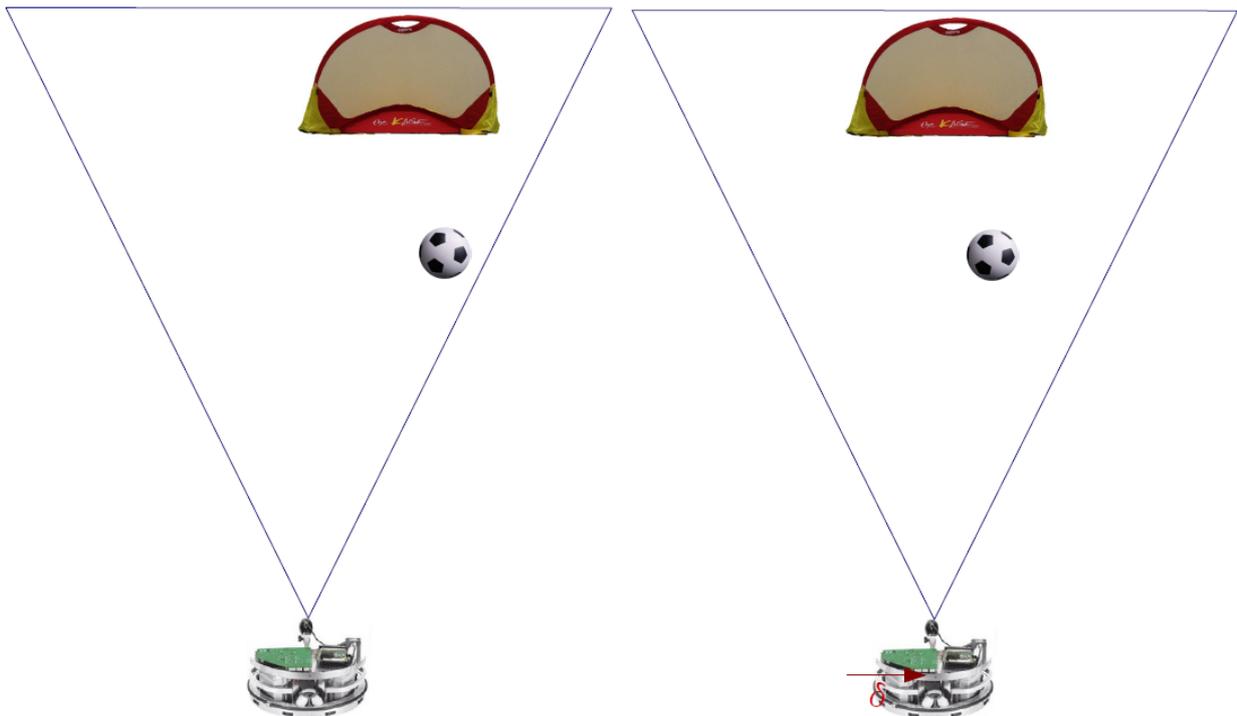
Tout d'abord, nous avons décidé que la précision sur la position du ballon était plus importante que la précision sur la précision de la cage (une erreur de 30cm sur le centre de la cage nous permet quand même de marquer, alors qu'une erreur de 30cm sur la position du ballon entraîne un échec certain).

La méthode visera donc à minimiser l'erreur sur la position du ballon.

L'étude de l'erreur commise sur la position des objets par rapport à Robotino nous montre qu'il faut que la position d'un même objet entre deux photos différentes soit la plus grande possible, et que le déplacement de Robotino entre les deux photos soit le plus faible possible.

Or il est impossible à partir de la première photo de savoir de quelle distance il faut se déplacer pour réaliser ces deux conditions car la position des objets n'est pas connue.

La stratégie que nous avons donc choisi de mettre en place est la suivante : une première photo est prise, ensuite Robotino se déplace légèrement afin d'être sûr que les deux objets apparaissent sur la deuxième photo.



Une première analyse est faite à partir de ces deux photos. On obtient ainsi la position approximative des objets par rapport à Robotino. On peut maintenant calculer le déplacement de Robotino qui permet de réaliser au mieux les conditions pour minimiser l'erreur sur la position des objets.



Une seconde analyse est ensuite effectuée à partir de la deuxième et la troisième photo.

Robotino connaît donc avec la plus grande précision possible la position du ballon, et celle de la cage. Il ne lui reste plus qu'à savoir comment envoyer le ballon au fond des filets.

3) Choix d'une stratégie de tir

Nous n'avons pas pu résoudre à temps les problèmes d'analyse d'image, en plus du temps pris au premier semestre, nécessaire pour optimiser les déplacements du robot, ce qui nous empêche de réaliser cette partie complètement.

Lors de nos essais de pilotage "à la WiiMote" de Robotino, nous nous sommes aperçus que foncer dans le ballon est une bonne approche de stratégie de tir dans un premier temps.

Une stratégie de tir possible serait donc de fixer des calles en "V" à l'avant de Robotino pour taper le ballon au bon endroit, de s'aligner dans la direction ballon-cage, et de foncer dans le ballon.

4) Les limites de la solution mise en place

Nous rencontrons des limitations dans l'utilisation de notre logiciel d'exploitation de Robotino.

a) Limitations d'ordre matériel

- Le support de l'analyse est un ordinateur, et il faut trouver un compromis entre qualité de cette analyse et vitesse d'exécution.

- Les images envoyées par Robotino sont souvent de faible qualité car l'outil d'acquisition est une WebCam. Ceci pose des problèmes de luminosité et de contraste.

- L'environnement dans lequel évolue Robotino, qui n'est pas forcément une aide à l'amélioration des contrastes.

b) Limitations d'ordre logiciel

- Les réglages du filtre pour la détection sont à effectuer photo par photo pour ne sélectionner que les cercles correspondant à la cage et au ballon.

- Les réglages sont souvent difficiles car la cage n'est pas strictement ronde, et le ballon bleu se fond souvent dans le décor, qu'il soit blanc, gris ou noir.

Ces réglages sont surtout sensibles aux conditions de luminosité et de contraste. Des pré-filtres sélectionnant les couleurs des cages et du ballon pourront être mis en place, mais ceux-ci nécessiteront un étalonnage unique pour chaque situation de luminosité. Leur réglage pourrait être cependant plus rapide à effectuer. On pourrait également travailler la mise en scène du stand d'évolution de Robotino en construisant un fond (vert pour augmenter les contrastes), et en soignant l'éclairage. Cette dernière méthode nécessite de mettre en place la présélection par couleur, puisque l'analyse s'effectue actuellement à partir des niveaux de gris puis sur des critères de contraste. Les couleurs contrastées peuvent alors se confondre.

V) Conclusion

Le cahier des charges que nous avons fixé imposait à Robotino d'être capable de reconstruire son environnement en seulement quelques prises de vue, puis d'élaborer une stratégie de tir afin de marquer.

Le travail que nous avons réalisé durant le second semestre permet de répondre à ce cahier des charges.

Il ne reste plus qu'à tester le programme avec Robotino ce qui n'a pas été possible jusqu'ici, le robot nous ayant été retiré trop tôt. Il resterait cependant quelques réglages à réaliser, notamment concernant la répétabilité du processus de reconnaissance de forme (qui nécessite un réglage différent à chaque expérience, ce réglage étant fonction de la luminosité, des couleurs de l'environnement, etc.) et du processus permettant la distinction du ballon et des cages.