

2012

INSA de LYON

PROJET SCOR

Chafik Bachatene, Jérôme De Potter, Henrique Gaspar Nogueira, Soraya Belhadj Aissa

Sommaire

Contexte et enjeux.....	3
Organisation.....	5
Architecture générale.....	5
Différents modules.....	6
Traitement d'images.....	6
Intelligence artificielle.....	11
Contrôleur.....	14
Problèmes rencontrés.....	15

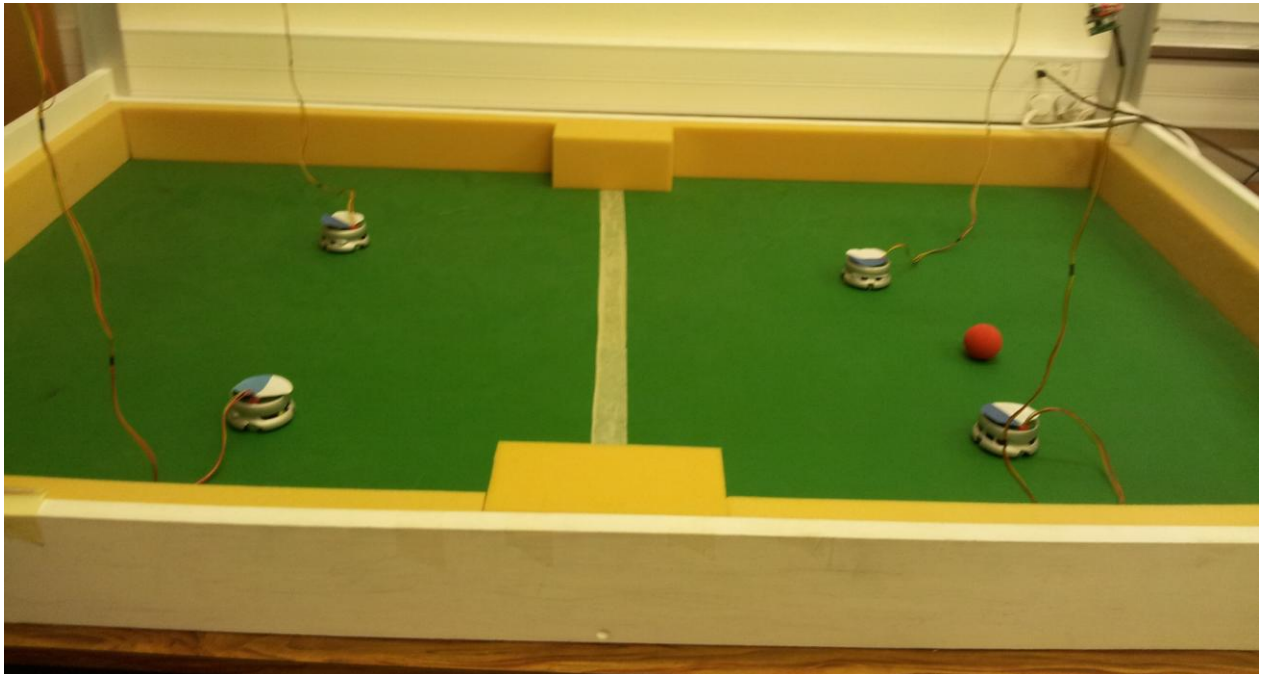
Présentation du projet

Contexte et enjeux

Le projet s'inscrit dans le cadre d'un projet spécifique proposé aux étudiants de 5ème année informatique associant à la fois une composante gestion de projet caractérisé par une certaine liberté mais aussi une composante très technique relative à une programmation très exigeante et contraignante.

Le sujet est un sujet de robotique ou il s'agit de mettre en place un tournoi de football (ou de pong) entre quatre équipes composées chacune de deux robots type Khepera.

La configuration matérielle est la suivante :



Chaque équipe dispose d'un ordinateur équipé d'un système d'exploitation Linux. Sur celui-ci est branchée une webcam permettant de visualiser le terrain, les robots et la balle.

Les robots sont reliés au pc via des ports usb, c'est à travers une liaison virtuelle série que la communication entre le pc et les robots est effectuée.

Chaque équipe est libre d'utiliser le langage de programmation qui lui convient, classiquement les équipes écrivent leur code en C++.

Des codes sources d'anciens projets (années précédentes) sont disponibles et les équipes sont vivement encouragées à y jeter un coup d'œil ...



En termes de règles du jeu, il n'y a rien de particulier hormis que les deux équipes doivent s'entendre sur le type de matériel utilisé en particulier la balle. les robots de chaque équipe ne doivent également pas dépassé la moitié du terrain (ils doivent par conséquent resté dans leur propre partie du terrain)



Organisation

Chaque équipe est composée de 4 personnes. La notre s'est organisée de la sorte

- Chafik Bachatene (Chef de projet)
- Henrique Gaspar Noguiera (Responsable traitement d'images)
- Jérôme De potter (Responsable intelligence artificielle)
- Soraya Belhadj Aissa (Responsable Khepera)

Le rôle du chef de projet a été très axé animation, motivation, coordination et très peu orienté vers de la planification. Nous avons opté pour une approche très itérative utilisant des outils de gestion de projet type agile.

En termes de planning, nous avons eu un rythme assez régulier maintenant une périodicité d'une séance par semaine (le lundi soir très souvent)

Les différentes phases ont été les suivantes :

Étude : d'abord une phase exploratoire pour bien comprendre le sujet, les différentes ressources mises à notre disposition, quelques recherches sur le Khepera 2, mise en place de l'environnement de travail, répartition des tâches et macro phasage.

Tests : Nous avons par la suite lancé une batterie de tests fonctionnels, dans un premier temps en essayant simplement de communiquer avec le khepera via un terminal et en utilisant les commandes basiques fournies dans la documentation du khepera, puis nous avons tenté de lancer les anciens programmes faits par des étudiants des années précédentes. A noter que ces codes ne compilent pas souvent du premier coup...

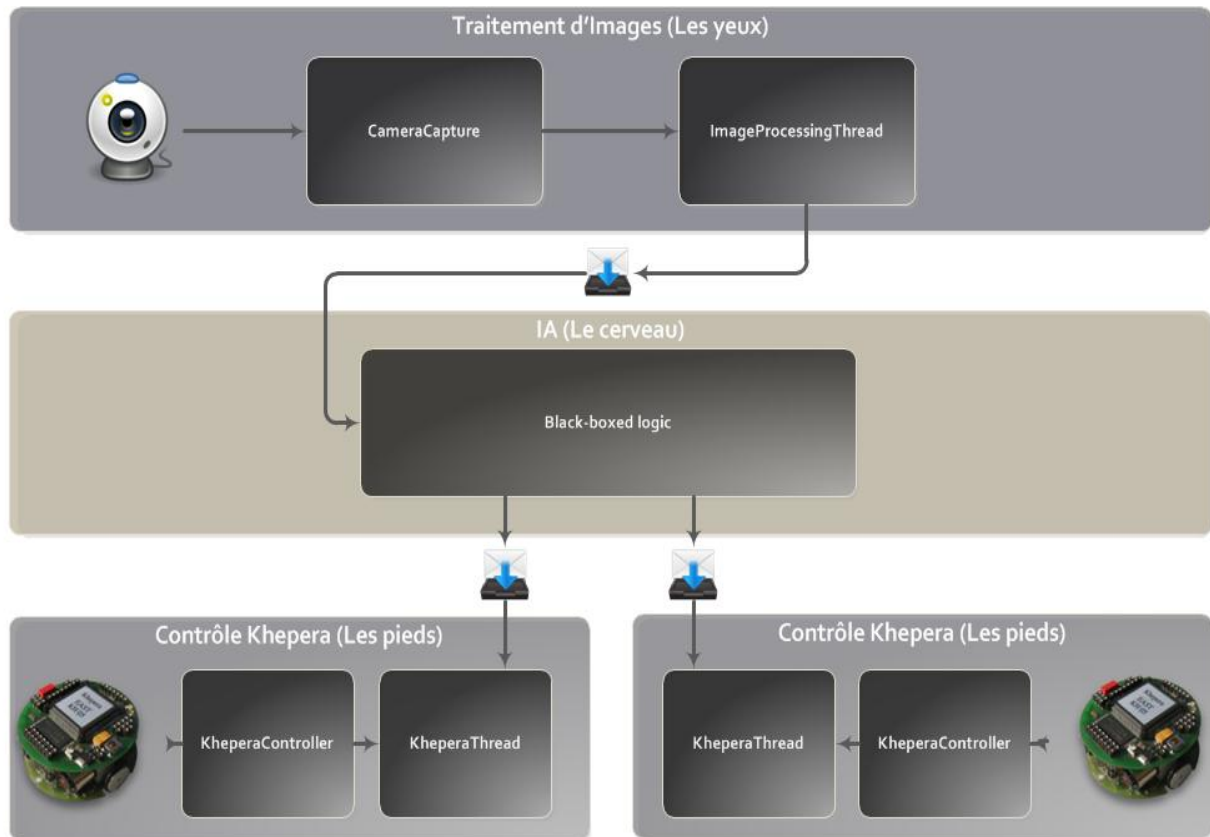
Enfin, une dernière phase, celle de projet proprement dit. il s'agissait dans un premier temps de bâtir une architecture solide s'inspirant des travaux précédents. Puis une répartition des responsabilités (un responsable par module) pour enfin commencer la réalisation.

Le travail de synchronisation et de communication entre les différents modules a été assuré par le chef de projet.

Aspects techniques

Architecture générale

Notre conception (assez classique) suggère donc un découpage en trois modules :



Différents modules

Traitement d'images

modélise la webcam (analogue aux yeux chez un être humain !)

Ce module est chargé de l'acquisition des images via la webcam placé au dessus du terrain. Il utilise la bibliothèque open cv pour l'acquisition des images, les diverses transformations telles que la perspective, l'identification des objets...

Fonctionnement général :

A chaque récupération d'image, le module applique une transformation perspective pour rendre le terrain à l'échelle et avoir un repère orthonormé.

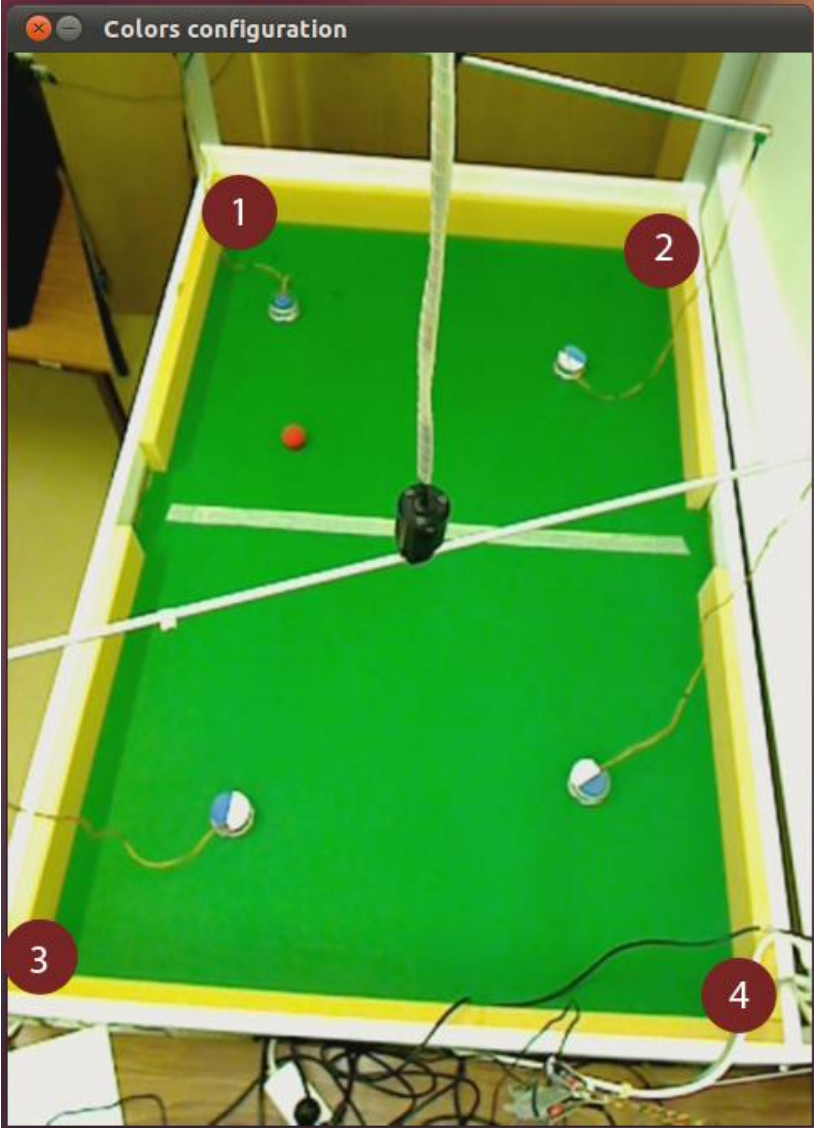
3 filtres sont ensuite appliqués à l'image obtenue de sorte à détecter les différents objets :

Filtre bleu : ce filtre n'est appliqué que sur la moitié du terrain (la notre) et ce car nous avons jugé inutile de détecter les robots adverses, sachant que ces derniers n'ont pas le droit de franchir la moitié du terrain, nous nous limiterons à la détection de nos propres robots (on minimise ainsi les calculs matriciels)

Filtre rouge : ce filtre sert à détecter la balle sur l'ensemble du terrain.

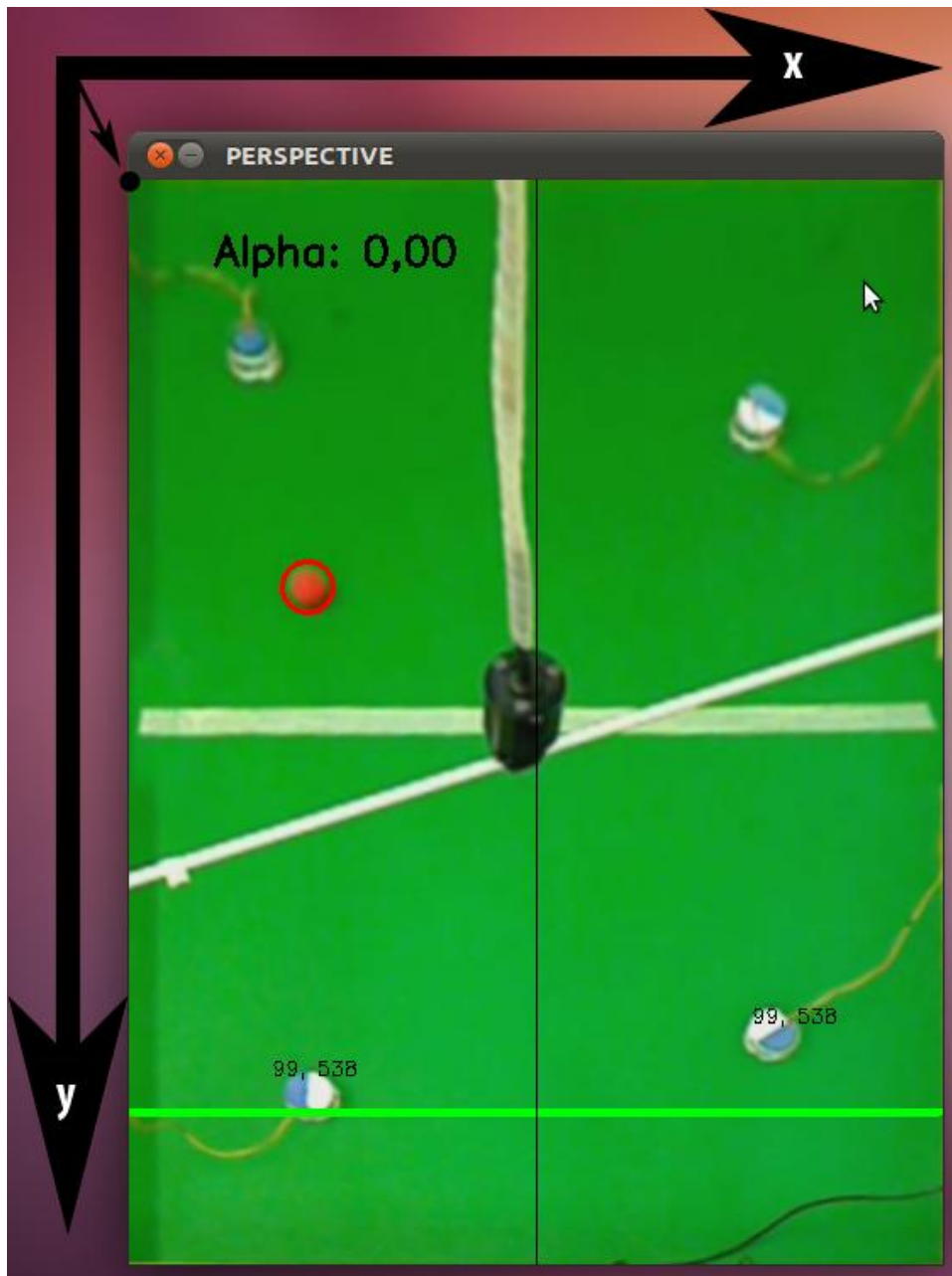
Nous ne faisons donc que de la détection de couleurs contrairement à ce qui a été fait durant les années précédentes ou une détection basée sur les forme a été mise en place.

Afin d'effectuer les transformations en perspective et la sensibilité des couleurs, une phase de paramétrage est requise. Celle-ci est effectuée grâce à IHM suivante :





L'utilisateur doit cliquer sur les quatre coins du terrain dans l'ordre indiqué.
Ensuite, la configuration des couleurs s'effectue en cliquant respectivement sur le blanc du robot, le bleu du robot puis le rouge de la balle (le clic sur le blanc du robot n'a aucune influence sur le fonctionnement de détection mais reste obligatoire dans cette version)
Enfin, nous avons mis en place un tableau de bord visuel nous permettant de vérifier la bonne détection des objets via des indicateurs visuels tels que la position des objets, les différentes stratégies en cours, l'orientation des robots.

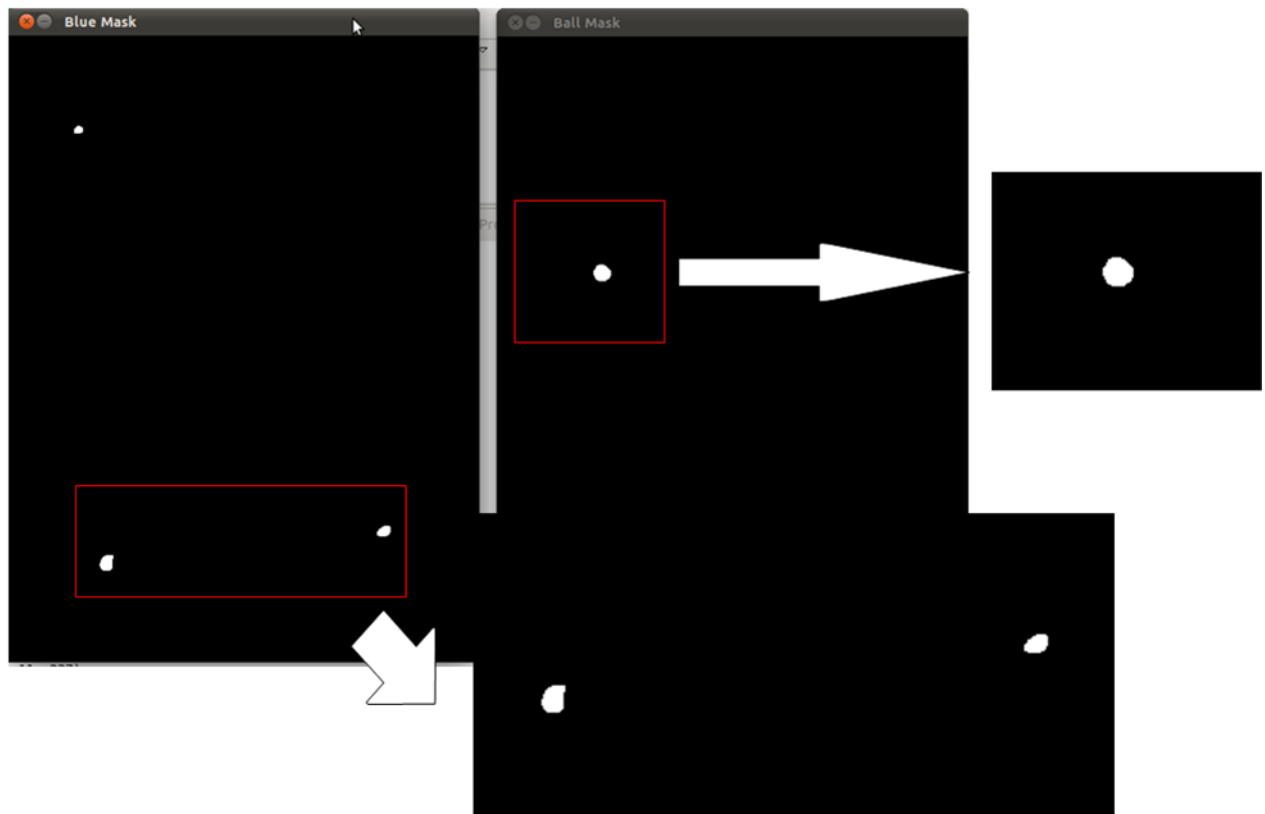


Détection d'objets :

Le module de traitement d'image détecte les objets et fourni les informations suivantes aux autres modules :

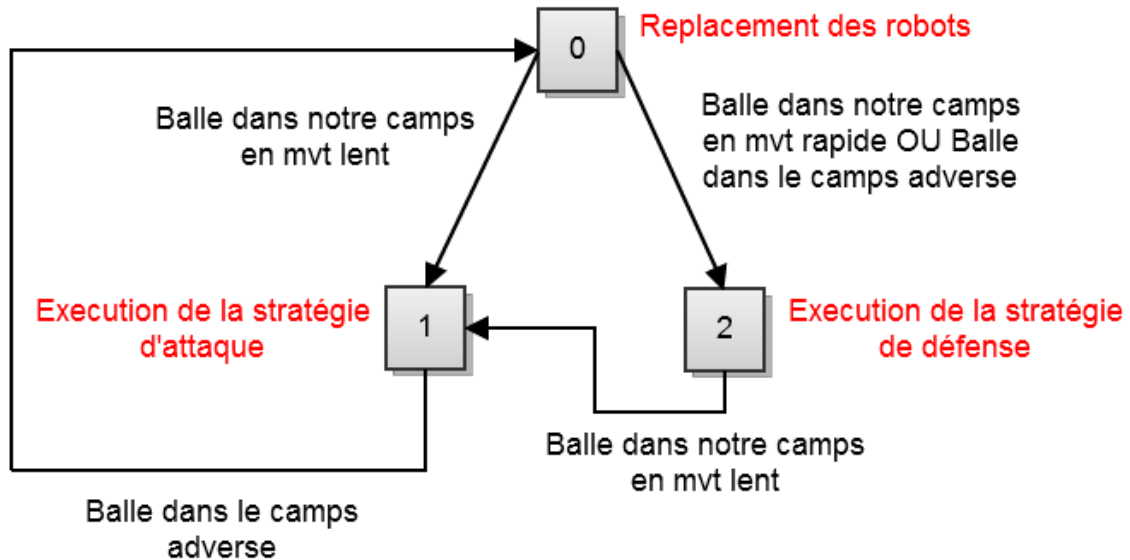
Coordonnées des deux robots nous appartenant sous forme d'abscisse et ordonné dans un repère que l'on a défini (cf voir figure)

Coordonnées de la balle sous forme d'abscisse et ordonné également.



Intelligence artificielle

Diagramme d'état



Le fonctionnement général de ce module suit les étapes suivantes :

Lecture périodique de la position des robots

- Détermination de la position par détection du bleu

- Calcul de l'orientation par incrémentation de l'angle de rotation

L'angle de rotation des robots est initialisé à zéro, a chaque fois que l'on demande au robot de tourner, on incrémente l'angle d'une valeur égale à l'angle effectif duquel il aura tourné.

Lecture périodique de la position de la balle.

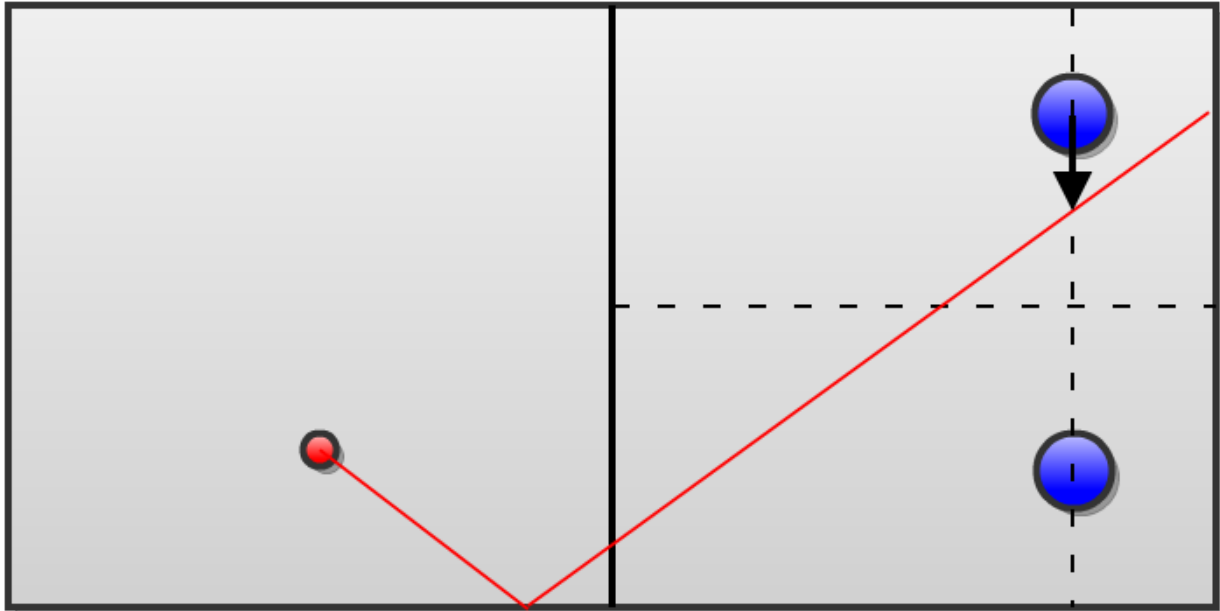
- Détermination de la position par détection du rouge

- Calcul de la trajectoire à partir de deux positions successives :

Calcul prévisionnel de l'impact de la balle par un calcul d'équation de droite en connaissant deux points appartenant à cette droite.

Le module IA gère deux stratégies : une stratégie d'attaque et une stratégie de défense

Stratégie de défense



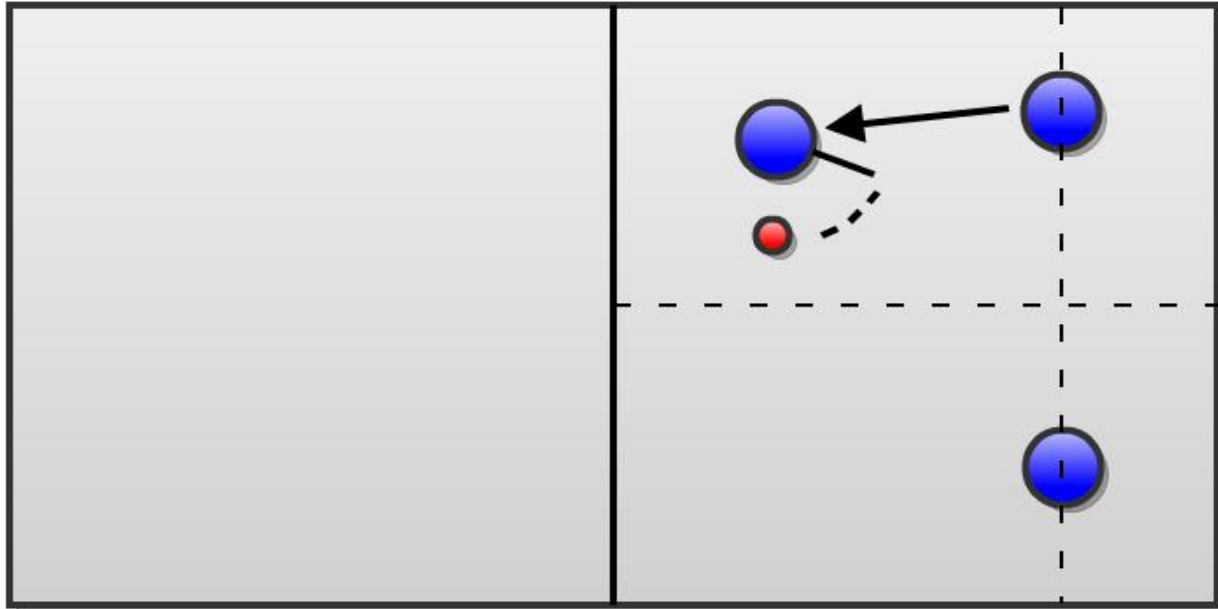
Notre stratégie consiste à consacrer chaque robot à une moitié du terrain (verticalement), ainsi nous avons un robot gauche et un robot droit (voir figure)

Une ligne de défense à également été mise en place, c'est-à-dire que le robot doit défendre et intercepter la balle au niveau de cette ligne horizontal

Les différentes étapes en stratégie de défense sont donc :

- Placement sur la ligne de defense
- Calcul du point d'intersection de la trajectoire de la balle avec la ligne de defense
- Envoi du robot approprié à ce point
- Interception de la balle

Stratégie d'attaque :



La stratégie d'attaque met en place un mécanisme de tir utilisant cette année l'extension du bras pour tirer en pivotant.

L'idée est de se placer systématiquement à droite de la balle et toujours tirer dans le même sens.

Le critère de changement de stratégie est à la fois la position de la balle mais aussi sa vitesse.

Les différentes étapes de cette stratégies sont donc :

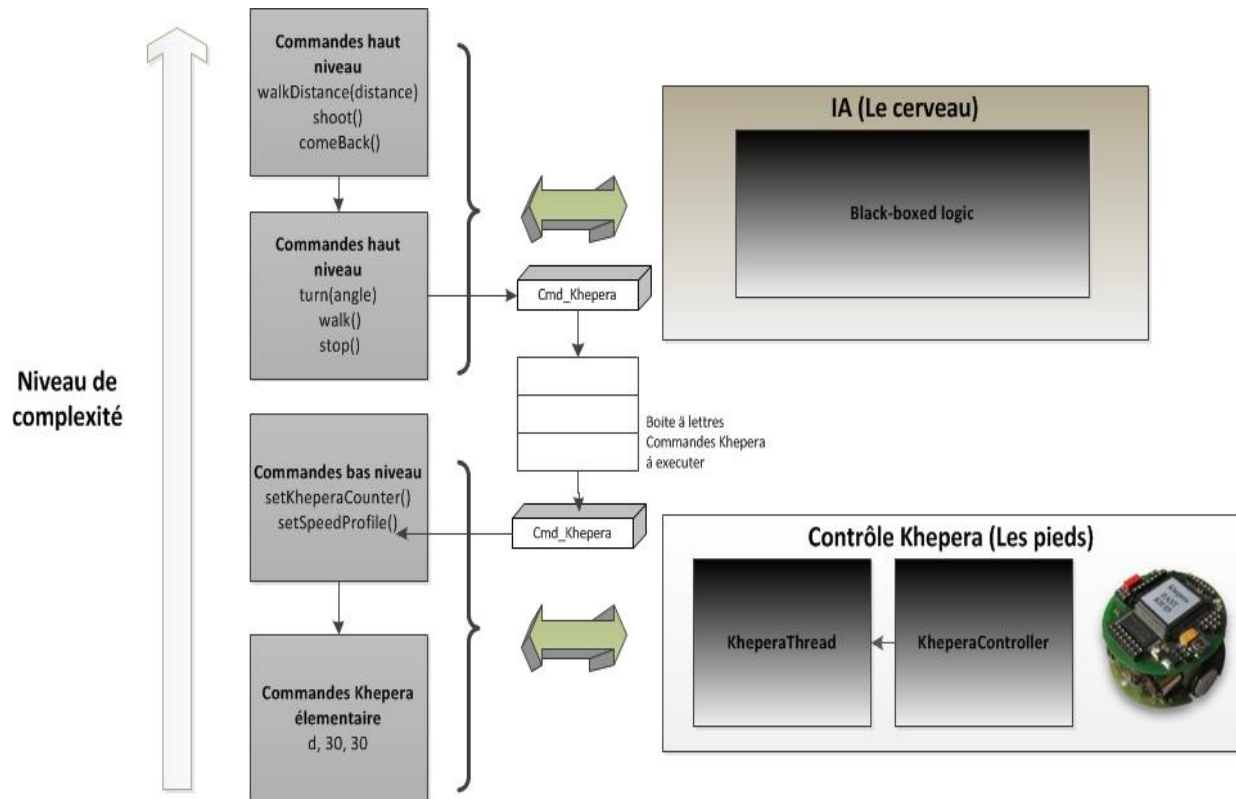
Principe :

- Attendre l'arrêt de la balle ou une faible vitesse de déplacement
- Envoie du robot à droite de la balle
- Calcul de l'angle de rotation
- Rotation du robot
- Frappe de la balle avec le bras

Contrôleur

Nous avons mis en place différentes couches pour communiquer avec le robot, allant du plus bas niveau (c'est-à-dire les commandes basique spécifiées dans la documentation du khepera jusqu'à une couche très haut niveau implémentant des commandes telles que le tir (voir code source)

Le schéma suivant illustre bien cette hiérarchie de couches.



La communication entre ces modules et la synchronisation est basée sur des boîtes aux lettres, nous n'avons pas utilisé de mutex ni de signaux.

Fonctionnement général

L'application est multitâches, ou chaque module représente une tâche à part. la communication entre ces modules et la synchronisation de ces derniers est basée sur des boîtes aux lettres, nous n'avons pas utilisé de mutex ni de signaux.

Problèmes rencontrés

Ci-dessous une liste de bugs techniques liés aux limites du khepera :

1. Limites matérielles du Khepera II

- Une vitesse bridée

2. Limites logicielles du Khepera II

- Pas de feedback ou très aléatoire
- Difficultés de transmission
- Faible vitesse de réaction

3. Les aleas de la vie

- Le frottement des roues contre le terrain irrégulier d'une roue à l'autre
- Le cas du "aller tout droit" est difficile lorsque la vitesse est importante

Difficultés communes aux 4 équipes ayant participé au projet :

Difficultés Organisationnelles

- Phase de démarrage
- Capitalisation de l'existant
- Travail collaboratif
- Partage des ressources (salle)
- Techniques
 - Compréhension du code existant
 - Configuration (bibliothèques...)
 - Modélisation/Réalisation du module de contrôle robot
- Matérielles
 - câbles/boîtiers défectueux
 - problème de communication robot (cf. longueur du câble)
 - qualité de réponse aux commandes
 - Delta vitesses moteur gauche/droit
 - déviation du robot liée à la tension du câble (en partie)
- Motivations
 - Inversement proportionnelle aux difficultés matérielles...