

Lycée Rochambeau, French International School, Washington DC, USA

OLYMPIADES de PHYSIQUE

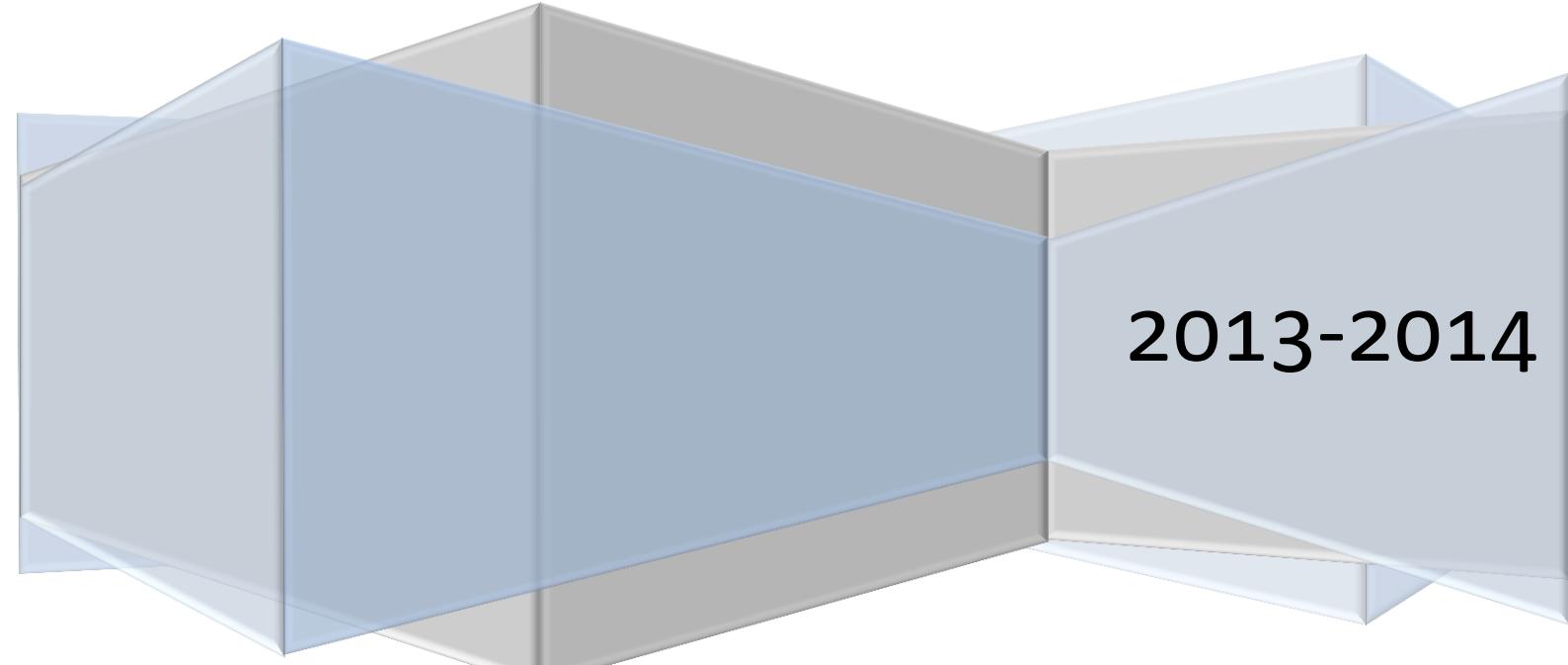
XXIème Edition

Optimisation d'un robot babyfoot

Grâce à des capteurs et des algorithmes génétiques.

Mateo Reveiz & Sacha Jungerman

Groupe F / Professeur : Victor Policarpo



2013-2014

Contenu:

Introduction au sujet	2
Méthodologie	3
1) Présentation générale du projet	3
2) La programmation : Interface Netlogo - Arduino.....	3
3) Le repérage de la balle	4
a) Les différentes méthodes.....	5
• Par l'analyse d'images provenant d'une webcam.....	5
• Grâce à des capteurs IR (infrarouges)	6
• Grâce à des capteurs de déplacement à laser.....	7
• Grâce aux capteurs d'ultrasons	9
b) Erreur systématique liée au couplage laser-capteur ultrason.	11
4) Algorithmes génétiques:.....	12
a) La matrice bidimensionnelle constitue le "génotype" d'un gardien virtuel.....	13
b) Sélection du "meilleur individu" grâce aux algorithmes génétiques.	14
5) Application pratique: construction du babyfoot.....	17
Résultats	18
1) Résultats concernant les capteurs.....	18
2) Résultats des algorithmes génétiques.....	19
Conclusions.....	21
Bibliographie.....	23
Sources des illustrations :	23

Introduction au sujet

L'idée de créer un robot babyfoot a surgi par plaisanterie autour d'un babyfoot : " Ce serait vraiment génial si en TPE on arrivait à réaliser un robot qui puisse jouer par lui-même au babyfoot ! ". La conclusion immédiate a été que ce serait un défi impossible pour nous, deux élèves de première scientifique. Puis en y pensant plus sérieusement, on s'est dit que ce serait l'occasion pour chacun d'entre nous de valoriser des compétences développées au cours de projets personnels précédents : on a commencé à y croire et on s'est mis au travail.

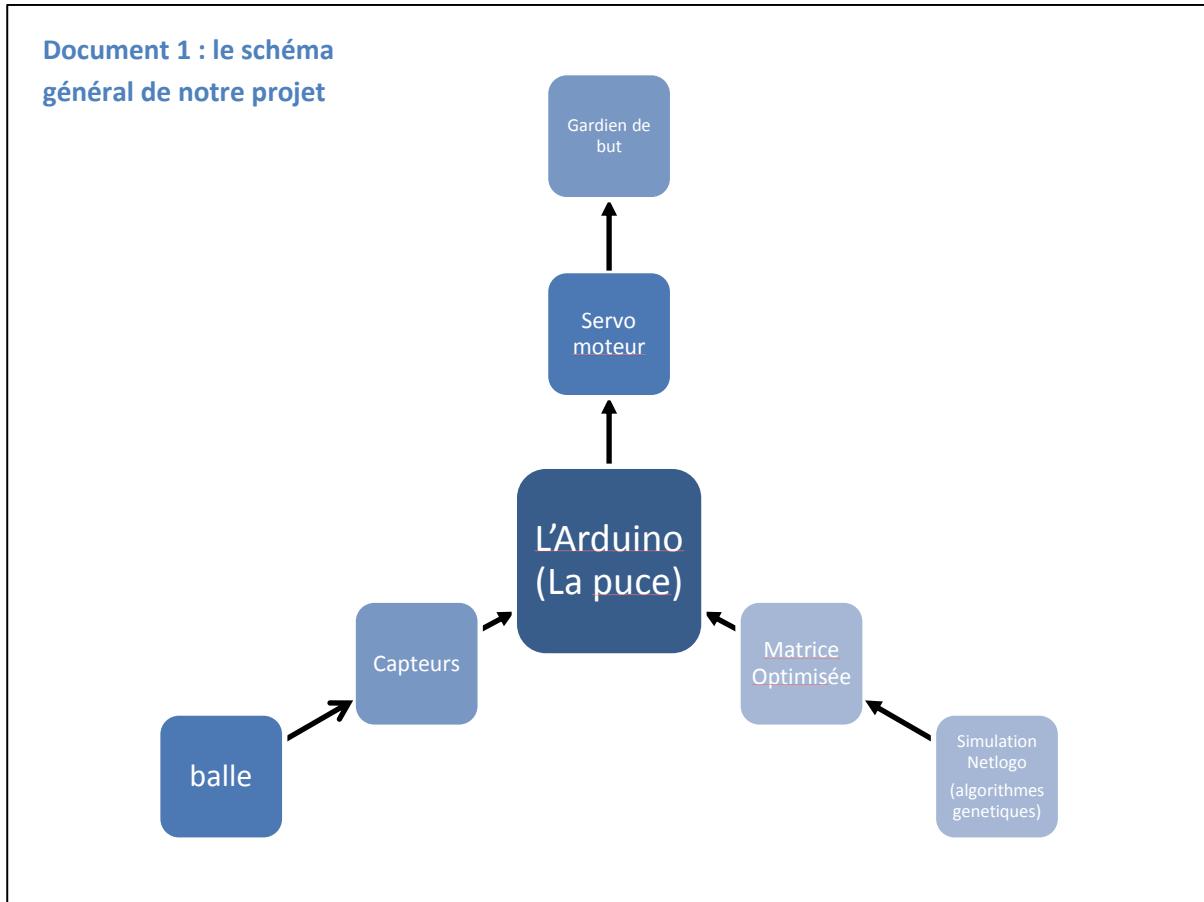
A l'origine, l'objectif du projet était de construire un robot qui joue au babyfoot et qui s'améliore à travers le temps en affrontant des joueurs humains. Cependant, cette idée devient rapidement très difficile à réaliser. On décide alors de simplifier l'idée d'origine pour créer une version plus limitée et accessible, surtout à cause du peu de temps dont on dispose. Le projet est finalement réduit à un gardien qui doit arrêter le plus grand nombre de tirs possibles. Ces tirs étant effectués par un humain, qui essaye de marquer.

Pour réaliser ce projet nous avons dû surmonter plusieurs contraintes. La première étant celle de devoir repérer la balle. Pour cela, on doit utiliser des capteurs pour trouver les coordonnées de la balle. Puis, vient le problème de créer une certaine « intelligence artificielle » qui se charge, en fonction des informations reçues par les capteurs, d'envoyer des ordres à l'actionneur qui fait bouger le gardien. Nous aurions pu utiliser une simple étude balistique pour contrôler le gardien, mais par ailleurs nous avions entendu parler chacun indépendamment des algorithmes génétiques. D'où l'envie d'essayer cette technique nouvelle et originale qui, par itération, devrait en théorie nous apporter des solutions à ce projet concret.

Méthodologie

1) Présentation générale du projet

Ce diagramme permet de schématiser l'ensemble de notre projet qui consiste donc à piloter les déplacements d'un robot, gardien de baby-foot, chargé d'arrêter les tirs d'un humain.



A partir des grandeurs physiques mesurées par les capteurs, la puce effectue les calculs pour repérer la balle. Ensuite, à partir des informations fournies par l'algorithme génétique (dernière matrice optimisée), la puce envoie des ordres au moteur qui pilote le gardien.

2) La programmation : Interface Netlogo - Arduino

Il nous fallait un logiciel pour écrire et exécuter nos algorithmes génétiques, nous avons choisi Netlogo.

Netlogo est un langage de programmation excessivement bien documenté dont le programme installé sur l'ordinateur nous permet d'exécuter notre script. Ce langage a été créé en 1999 par Uri Wilensky pour réaliser des modèles informatiques. L'interface est relativement simple, on peut ajouter par exemple, des boutons ou graphes et les positionner où l'on veut. Netlogo est un langage de programmation très puissant, grâce à lui on peut même faire des modèles en trois dimensions. Ce langage est donc idéal pour nous aider à réaliser ce projet.

Pour traiter les informations et contrôler le gardien nous avions besoin d'un microcontrôleur : nous avons choisi l'Arduino.

L'Arduino est une plateforme de programmation puissante et compréhensible. Ce microcontrôleur appartient au domaine public, n'importe qui peut le construire et/ou le distribuer et donc cette puce est très répandue et bien documentée. Le modèle qu'on a utilisé est l'Arduino UNO R3. Celui-ci n'est pas le meilleur des Arduino mais c'est le plus récent. Il nous suffit largement et nous l'avions déjà en notre possession. Il a une vitesse d'opération de 16 MHz, il comporte 6 entrées analogiques et 14 autres qui sont soit des entrées soit des sorties (on règle cela lors de la programmation de la puce). Par la suite, on utilisera seulement 2 entrées analogiques, et 4 entrées digitales ainsi qu'une sortie.

La première étape était de créer un pont entre Netlogo (qui fera la simulation de tous les calculs), et l'Arduino (à savoir la puce qui permettra l'interface virtuel/réel). Le pont est un câble USB, qui permet également de programmer la puce, avec une vitesse maximale de communication de 1MBps (1 Mégabyte par seconde, c'est-à-dire un millions de bytes communiqués par seconde soit environ 400 pages Word). Pour un fonctionnement optimal de la puce, nous avons réduit cette vitesse de transfert à 9600 Bps (byte par seconde). Pour établir cette communication, il fallait trouver et utiliser une extension¹ de Netlogo qui nous le permette. Plusieurs extensions permettaient la communication de Netlogo à l'Arduino, mais aucune ne nous permettait d'établir une communication dans les deux sens, celle-ci était essentielle car sans elle, Netlogo n'allait pas recevoir les valeurs mesurées par les capteurs. Dans un premier temps pour résoudre ce problème, nous avons envisagé de mettre l'algorithme génétique dans sa totalité dans la puce. Cela n'a pas été possible à cause de la taille de ce dernier. On a contourné ce problème en plaçant dans la puce uniquement le résultat optimisé de l'algorithme génétique.



Document 2 : Puce Arduino UNO

3) Le repérage de la balle

Le premier problème qui s'est posé était celui lié au repérage de la balle. Naturellement, il existe un grand nombre de méthodes possibles pour la détecter, mais certaines sont trop compliquées ou hors de notre portée. Nous avons testé différentes solutions pour la détection : une caméra, un capteur de présence infrarouge et un capteur d'ultrasons.

¹ Une extension est une bibliothèque qui permet d'ajouter des fonctionnalités au programme. Celle utilisée ici est : <https://github.com/zhaga/NetLogo-Arduino-extension>

a) Les différentes méthodes

- Par l'analyse d'images provenant d'une webcam

La première idée pour repérer la balle est l'utilisation d'une caméra. Suite à une analyse des images capturées en temps réel, on pourrait retrouver les coordonnées de la balle en utilisant NetLogo. L'image est transformée pour créer une nouvelle image de moins bonne qualité et divisée en plusieurs carreaux de couleur (à peu près 10 000). Chaque carreau a une couleur déterminée par trois valeurs entre 0 et 255 correspondant au rouge, au vert et au bleu (RGB en anglais).

Pour commencer l'étude, on prend une photo de la balle et on analyse ses couleurs (les couleurs de chaque carreau composant la balle). Ainsi, on fait la médiane de toutes les valeurs R, puis de toutes les valeurs G, puis de toutes les valeurs B pour retrouver une couleur médiane de l'ensemble de la balle. Connaissant la couleur de la balle qu'on utilise, (et donc les valeurs en RGB) on détermine des encadrements « acceptables », proches de la médiane de toutes les valeurs RGB des carreaux composant la balle.

Pour suivre les coordonnées de la balle en temps réel nous analysons la couleur de chaque carreau de l'image récupérée. Si les valeurs du carreau en R en G et en B se trouvent dans l'encadrement établit précédemment, alors les coordonnées de ce carreau sont ajoutées à une liste. A la fin de l'analyse de tous les carreaux, on fait la médiane des abscisses et la médiane des ordonnées des carreaux ayant satisfait les conditions. On retrouve ainsi les coordonnées approximatives de la balle.

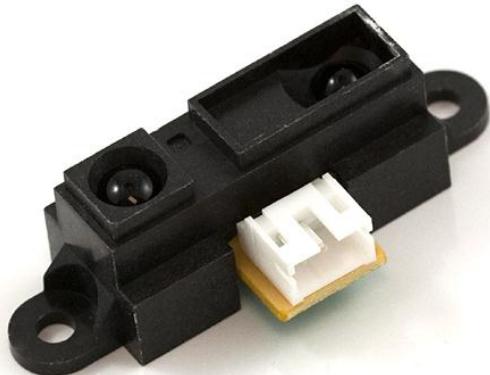
L'utilisation de la médiane dans toute cette étude est très importante. En effet, elle doit éliminer toutes les valeurs extrêmes qui pourraient influencer le résultat final. Ces extrêmes peuvent provenir des ombres, qui modifient les couleurs perçues par la caméra ou par les effets de mouvement de la balle qui peuvent déformer les images et donc fausser les véritables couleurs. D'où le rejet de la moyenne qui est sensible aux valeurs extrêmes.

Il faut remarquer que cette méthode nous permet de retrouver les coordonnées de la balle par rapport au référentiel de la fenêtre de capture de l'image. Il faut ensuite transformer cette information pour retrouver les coordonnées de la balle dans le référentiel « réel » de la table de babyfoot. Pour rendre plus facile cette tâche, on fait en sorte qu'un carreau « virtuel » de côté 1 dans le programme correspond à 1 cm ou à 0.5 cm dans la réalité.

Dans un premier temps, nous nous sommes confrontés à des difficultés de vitesse d'analyse des images, nous avons alors abandonné cette méthode. Dernièrement, suite aux suggestions faites par le jury académique des olympiades, nous avons réactivé cette méthode et nous sommes capables depuis très peu de temps de suivre la position de la balle quasiment en temps réel (23 images par seconde...) sur NetLogo.

- Grâce à des capteurs IR (infrarouges)

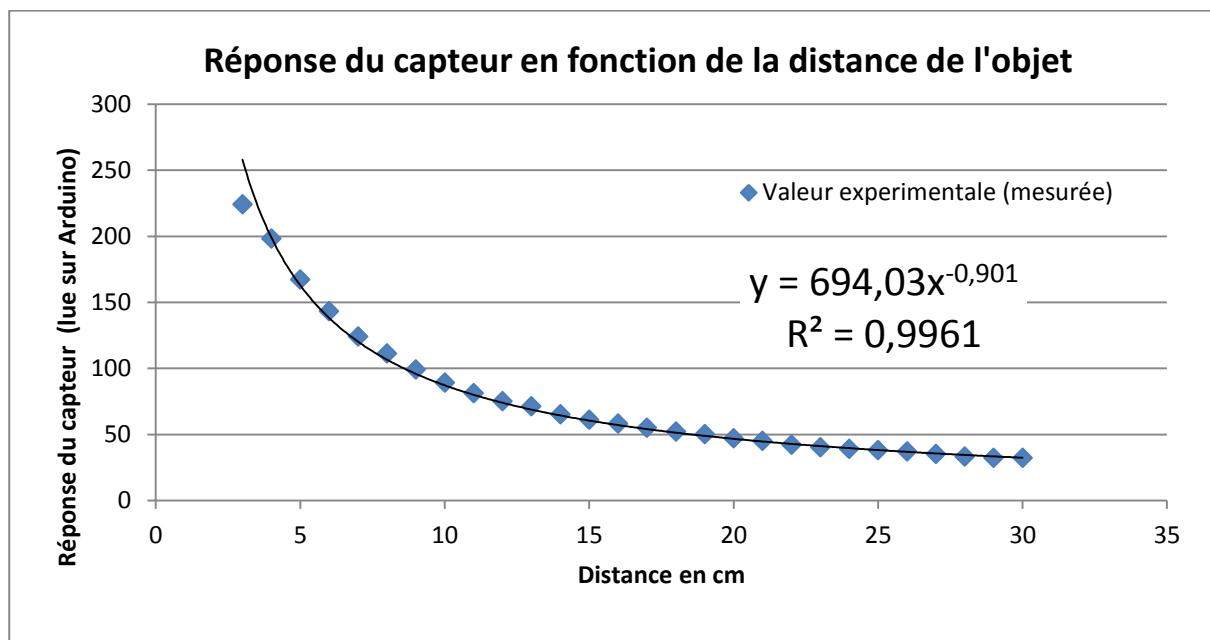
Tout d'abord un capteur IR est un capteur de présence simple, très répandu. Le capteur comporte un émetteur IR et un récepteur. L'émetteur est une diode infrarouge et le récepteur un phototransistor sensible aux radiations IR. Les infrarouges sont émis. S'il y a un objet à proximité, ils sont réfléchis et le phototransistor les détecte. Selon l'intensité des radiations reçues, la réponse analogique du capteur est plus ou moins importante. Et donc on peut connaître la distance approximative de l'objet.



Document 3 : Photographie d'un capteur IR

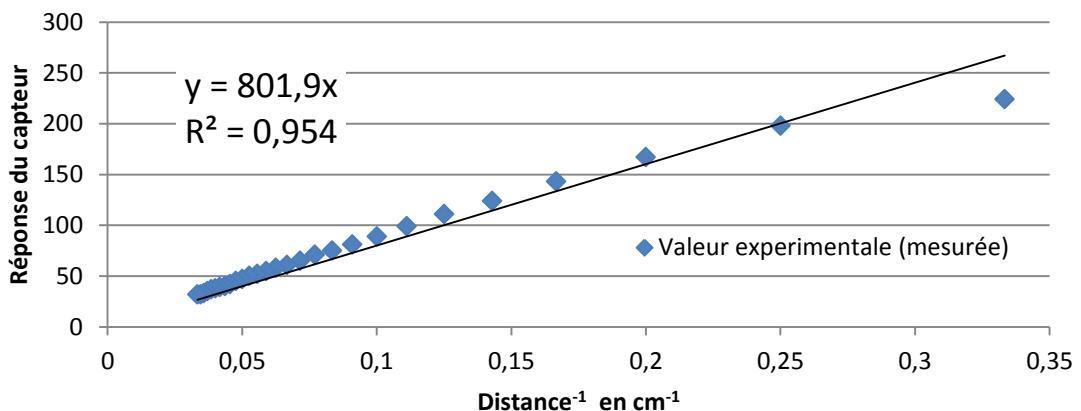
avons donc fait les premiers tests sur une boîte en carton, plus pratique à manipuler qu'une balle. Nous avons fait varier la distance d de la boîte au capteur en notant, pour chaque distance, la réponse R du capteur reçue sur l'Arduino. Chaque mesure a été effectuée plusieurs fois pour être sûr de sa répétabilité. Nous avons obtenu une très bonne corrélation mais celle-ci n'est pas linéaire. On a, à peu près, une branche d'hyperbole qui correspond à la fonction d'équation

$R(d) = 6,9 \cdot 10^2 \cdot d^{-0,90}$ donnée par le tableur. En traçant R en fonction de $(1/d)$, on obtient une droite moyenne d'équation $R = 8,0 \cdot 10^2 \cdot (1/d)$.



Document 4 : Courbe de la réponse du capteur en fonction de la distance de l'objet réflecteur.

Réponse du capteur en fonction de l'inverse de la distance de l'objet



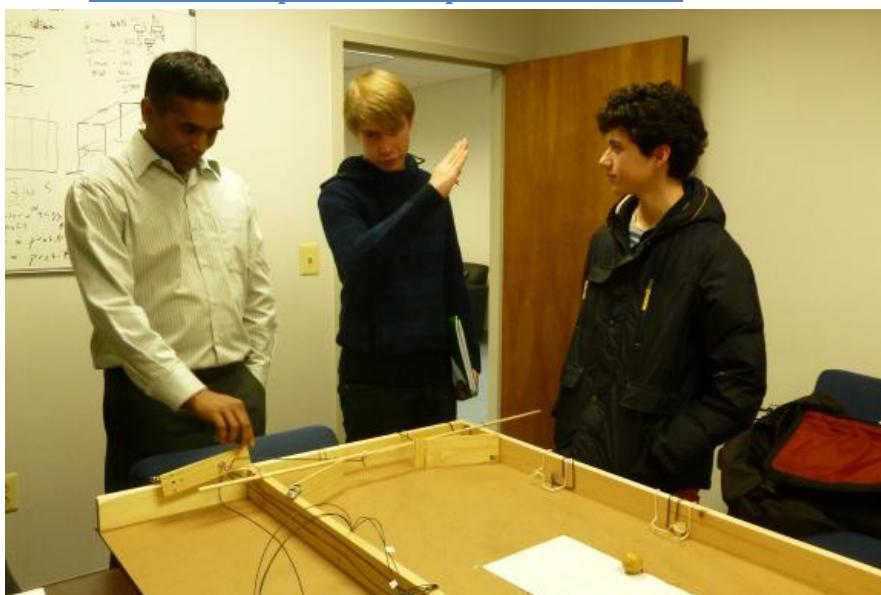
Document 5 : Courbe de la réponse du capteur en fonction de l'inverse de la distance de l'objet réflecteur.

Les raisons qui ont fait que nous avons abandonné l'idée d'utiliser un capteur IR :

- 1) Nous avons constaté que le détecteur de présence IR ne fonctionne pas lorsque l'objet est à moins de 3 cm (zone « aveugle ») et a du mal à le détecter au-delà de 30 cm. Donc, cela ne suffisait pas car notre babyfoot mesure plus 30 cm de largeur.
- 2) Nous avons également constaté que lorsque la boîte en carton était déplacée devant le capteur tout en la maintenant à la même distance (comprise entre 3 et 30 cm), l'intensité reçue fluctuait. En effet, il existe plusieurs positions possibles de la balle pour une même valeur d'intensité reçue.

On a fini alors par abandonner ce type de capteur qui ne permettait pas de déterminer de façon précise la position de la balle.

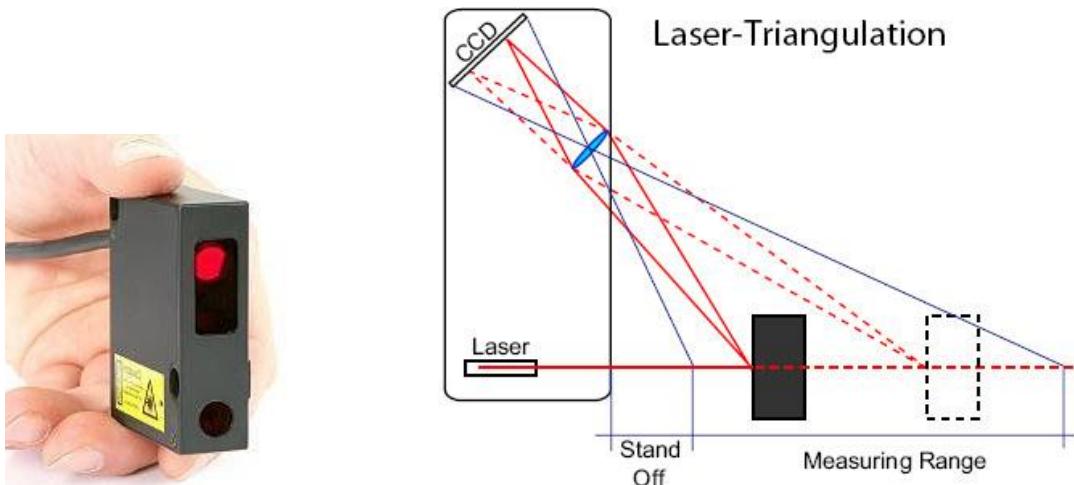
- Grâce à des capteurs de déplacement à laser



Document 6 : Photographie de notre rencontre avec Mr Mohammed Islam, vice-président de Post-IS et directeur technique au sein de l'entreprise.

Nous avons eu la chance de présenter notre projet à des professionnels. En effet, le 7 janvier 2014 nous avons passé l'après-midi à Baltimore MD, dans les locaux de l'entreprise POST-Integrated Solutions (<http://post-is.com>) spécialisée dans les automatismes appliqués au tri postal. A ce moment-là, nous avions déjà privilégié la solution des capteurs ultrasons que nous développerons plus loin.

Lors de nos discussions, Mrs Guillaume Lacalmontie et Mohammed Islam (documents 6 et 8), nous ont suggéré d'utiliser des capteurs de déplacement à laser pour repérer notre balle.



Document 7 : Photographie d'un capteur de déplacement à laser et un schéma du principe de fonctionnement

Même si par manque de temps, nous n'avons pas encore pu tester ces capteurs, cette visite a été néanmoins très enrichissante :

- Nous avons vu fonctionner des prototypes très avancés de machines de tri de colis postaux utilisant un grand nombre de types de capteurs,
- Nous avons aussi constaté que malgré l'utilisation de logiciels de simulation très performants, certaines idées devaient aussi être testées à l'aide de prototypes en bois (ce qui nous a rappelé notre maquette),
- De nombreux mails ont été échangés suite à cette visite,
- Mrs Lacalmontie et Islam nous ont suggéré de faire des enregistrements des signaux reçus et émis par le capteur ultrasons pour comprendre pourquoi il ne détecte pas la balle au-delà d'une distance limitée (environ 20 cm).

C'est l'objectif que nous nous sommes fixés après la rédaction de ce mémoire.



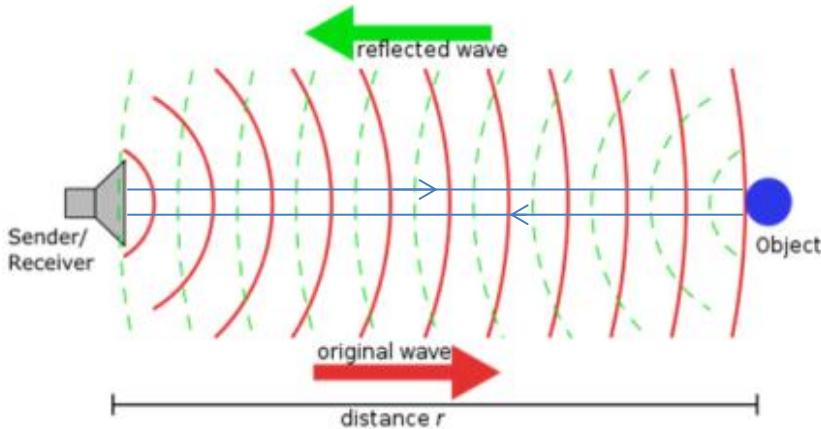
Document 8 : Photographie avec toute l'équipe de post-IS. Mr Lacalmontie, à gauche, est ingénieur système spécialiste des automatismes appliqués au tri postal.

- Grâce aux capteurs d'ultrasons

Nous étions déjà familiarisés avec des capteurs de ce type avant de commencer ce projet. Ces capteurs sont très similaires au SONAR (SOund Navigation And Ranging) utilisé par les sous-marins et leur principe de fonctionnement est le même. Ils ont été développés dès la première guerre mondiale par les anglais. Le capteur est équipé d'un émetteur d'ultrasons (ondes mécaniques non audibles pour l'homme, c'est-à-dire plus de 20 000 Hertz) et d'un récepteur. Contrairement aux capteurs IR, on ne mesure pas l'intensité reçue, mais le temps qui s'écoule entre le moment d'émission et l'instant où les ondes réfléchies reviennent au récepteur. A partir de là on peut calculer la distance de l'objet si l'on connaît la vitesse du son dans le milieu où l'on se trouve.



**Document 9 :
Photographie de capteurs ultrasons**

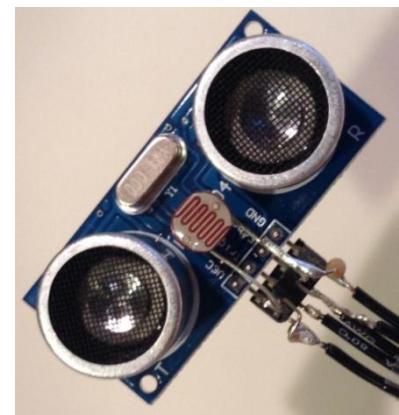
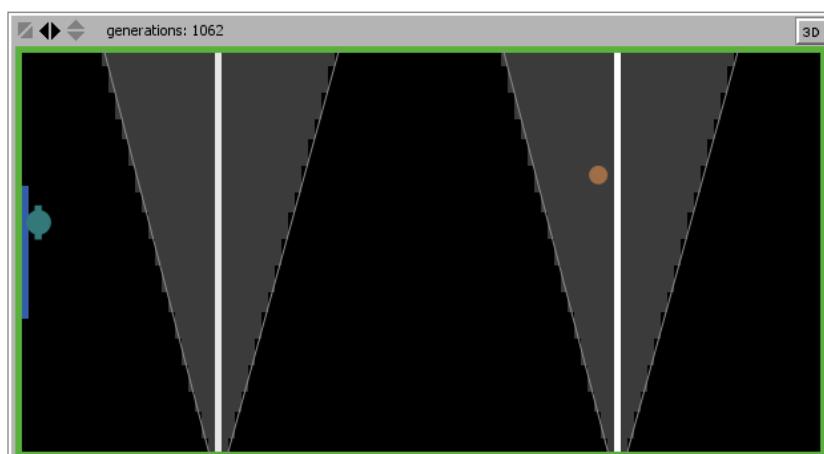


Document 10 : Principe de fonctionnement d'un capteur à ultrasons
(source : Wikipédia)

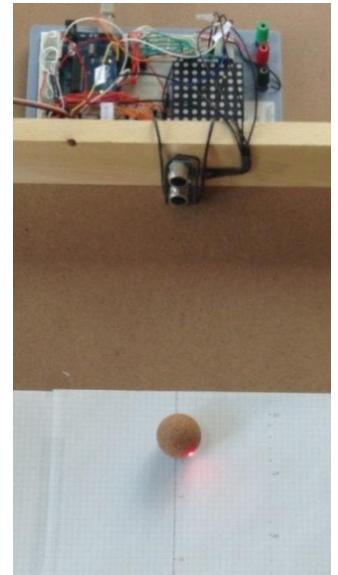
La formule pour calculer la distance de l'objet : $d = \frac{V_{son} \cdot \Delta t}{2}$

Bien entendu, nous avons réalisé une série de tests ici aussi, et nous avons été agréablement surpris par la précision et la distance jusqu'à laquelle ce capteur était sensible (jusqu'à environ 5 mètres). Ces capteurs sont assez précis, ceux qu'on utilise ont une précision de plus ou moins 3 mm. On s'est alors demandé comment nous allions utiliser ces capteurs pour repérer le plus précisément possible les coordonnées d'une balle. Le plus simple était de mesurer la distance balle-capteur quand la balle était à une abscisse bien spécifique et connue, la distance mesurée serait alors l'ordonnée de la balle. Ainsi on pouvait déterminer une position de la balle. Il nous fallait donc un autre outil qui puisse déclencher la mesure (par l'envoi d'une salve d'ultrasons) au moment où la balle traverse une abscisse choisie. C'est là que l'idée d'utiliser des lasers a surgi : matérialiser une abscisse par un faisceau laser pointant vers une photorésistance placée à proximité du capteur à ultrasons (documents 11 et 12). Ainsi, dès que la balle coupe le faisceau laser, on pourra commander l'activation du capteur à ultrasons.

Voici une image issue d'une de nos innombrables simulations NetLogo qui illustre bien les deux capteurs ultrasons avec leur "champ de vision", en gris, et les deux lasers, en blanc (il y a aussi le gardien et son but en bleu et la balle en marron):



Document 11 :
Notre module Sonar / photorésistance



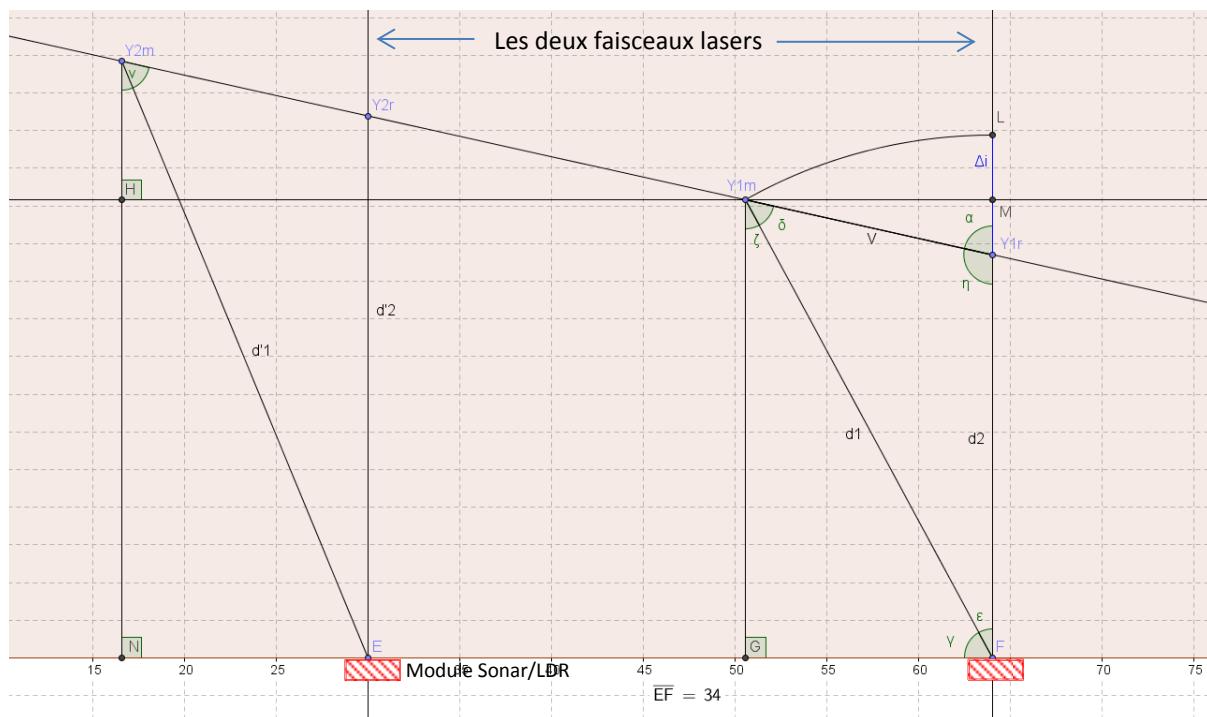
Document 12 :
Test avec un prototype

Document 13 : Copie d'écran du modèle Netlogo pour étudier les capteurs

b) Erreur systématique liée au couplage laser-capteur ultrason.

En simulant sur NetLogo notre idée d'utiliser deux lasers couplés avec deux capteurs ultrasons, nous avons détecté un décalage entre la simulation et la réalité : les ultrasons, étant relativement lents, pendant leur aller-retour entre la balle et le capteur, la balle pourrait dépasser l'abscisse du laser. La distance mesurée par le capteur ultrasons pourrait donc ne plus correspondre exactement à l'ordonnée de la balle pour l'abscisse matérialisée par le faisceau laser. La distance d_1 mesurée par le capteur à ultrasons pour une position de la balle Y_1m (voir document 14) n'est donc pas d_2 ($= Y_1rF$), l'ordonnée « réelle » pour l'abscisse du laser. La véritable ordonnée cherchée est donc la distance d_2 .

Le schéma qui suit (document 14) n'est pas à l'échelle : les distances Y_1rY_1m et Y_2rY_2m parcourues par la balle entre le moment où elle franchit le faisceau laser et le moment du retour de la salve d'ultrasons sur le capteur ont été exagérées pour faciliter la lecture du schéma.



Document 14 : Modélisation du problème avec GeoGebra

Remarques :

- 1) On néglige ici le temps de réaction de la puce électronique Arduino, qui est extrêmement peu élevé en raison de la vitesse d'opération du processeur: 16 MHz.
- 2) On suppose qu'il est possible dans le cadre de cette étude de faire l'hypothèse que les frottements de la balle avec le support et l'air sont négligeables, ce qui nous permet de considérer que la balle a un mouvement rectiligne uniforme.

On cherche donc, en s'appuyant sur le document 14, à déterminer la distance d_2 , abscisse réelle de la balle au moment de la traversée du faisceau laser n°1.

On suppose que les paramètres connus sont :

- la distance EF (distance entre les deux lasers),
- la valeur de d1 (mesurée par le capteur ultrasonique),
- la valeur de d'1 (mesurée par le capteur ultrasonique),
- la durée Δt_1 , nécessaire aux ultrasons pour parcourir la distance d1
- la durée $\Delta t'_1$, nécessaire aux ultrasons pour parcourir la distance d'1
- la durée $t_2 - t_1$ entre la traversée du faisceau laser n°1 et la traversée du faisceau laser n°2

Si le mouvement est rectiligne uniforme, la composante horizontale V_x du vecteur vitesse du centre de gravité de la balle est constante. Cette valeur peut être calculée : $V_x = \frac{EF}{t_2 - t_1}$

Ce qui nous permet de calculer les distances $NE = V_x \cdot \Delta t'_1$ et $GF = V_x \cdot \Delta t_1$

Par application du théorème de Pythagore :

$$Y2_mN = \sqrt{d'^1_1^2 - NE^2}$$

$$Y1_mG = \sqrt{d1^2 - GF^2}$$

Ce qui nous permet de calculer HY2m : $HY2m = Y2mN - Y1mG$

Il nous reste à utiliser le théorème de Thalès ((HM) et (Y2mY1m) sont deux droites sécantes en Y1m et (NH) et (FM) sont deux droites parallèles) :

$$\frac{HY2m}{Y1_rM} = \frac{Y1_mH}{Y1_mM} \quad \text{Par ailleurs :} \quad \frac{Y1_mH}{Y1_mM} = \frac{GN}{GF} \quad \text{D'où :} \quad \frac{HY2m}{Y1_rM} = \frac{GN}{GF}$$

Ainsi $Y1_rM = \frac{GN}{GF} \cdot HY2m$

Et enfin : $d2 = Y1_mG - Y1_rM$

Un raisonnement similaire nous permettrait de calculer d'2.

Pour l'instant dans l'avancement actuel de notre projet nous n'avons pas pris en compte ces calculs d'erreurs systématiques car pour des vitesses Vx inférieure à 2 m/s, la différence entre la distance d1 mesurée et la distance d2 n'excède pas 5 mm.

4) Algorithmes génétiques:

Les algorithmes génétiques appartiennent à la famille des algorithmes évolutionnistes.

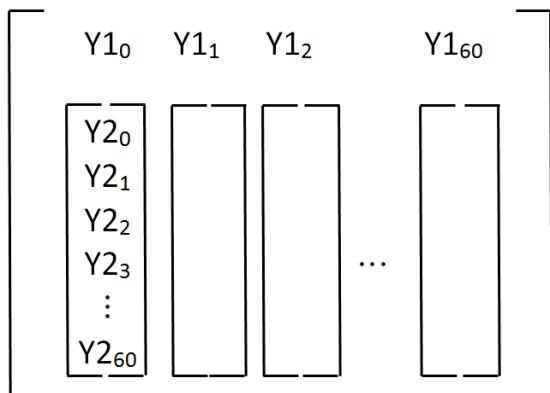
L'algorithme génétique se base sur des phénomènes biologiques. Il fait donc appel à des notions d'évolution telles que la sélection naturelle et un vocabulaire « génétique » peut également être utilisé.

L'objectif des algorithmes génétiques est de trouver des solutions approchées à des problèmes posés qui comportent souvent plusieurs variables. Il s'agit donc d'une méthode qui, par itération, sert à résoudre des problèmes par optimisations successives.

a) La matrice bidimensionnelle constitue le "génotype" d'un gardien virtuel

Dans notre cas, l'algorithme génétique est chargé de trouver la position optimale du gardien en fonction de deux coordonnées successives de la balle. C'est-à-dire que pour chaque couple d'ordonnées de la balle lors d'un tir, l'algorithme doit ressortir la position du gardien qui lui permettrait d'arrêter la balle.

La largeur de la table de baby-foot étant de 60 cm, il existe donc 60 valeurs possibles pour la première ordonnée Y_{1r} (si on l'arrondit au cm près). De même, il existe aussi 60 valeurs possibles pour la deuxième ordonnée Y_{2r} .



Document 15 : Structure générale de la matrice d'un individu

Une matrice 60x60 à 2 dimensions serait donc l'outil mathématique idéal pour stocker l'information de la position que doit prendre le gardien pour chaque couple d'ordonnées mesurées.

Exemple : la valeur contenue dans la colonne $Y_{1_{20}}$ et sur la ligne $Y_{2_{10}}$ correspond à la position que doit prendre le gardien si la balle est détectée à 20 cm par le premier capteur et à 10 cm par le deuxième.

On comprend ici pourquoi l'erreur systématique (finalement inférieure à 5 mm) n'a pas été prise en compte puisque notre matrice bidimensionnelle est construite à partir des ordonnées arrondies au cm près.

A ce stade, il est essentiel de bien comprendre la notion d'individu (terme plus général que "gardien virtuel" utilisé plus haut). Si nous avions pu réaliser notre projet avec un gardien et un défenseur, le terme "individu" aurait englobé le gardien et le défenseur. Dans la logique des algorithmes génétiques, un individu est entièrement déterminé par son « patrimoine génétique » qui correspond à l'ensemble des valeurs contenues dans une matrice. Un individu réagira à des situations diverses (dans le cas de notre projet, à différents tirs) en fonction de sa matrice. Pour une même situation ou tir, l'individu aura toujours la même réaction (dans notre cas, le gardien se positionnera en fonction de la valeur qu'il trouvera dans sa matrice).

Une matrice 60x60 quelconque s'apparente donc au génotype d'un individu (qu'il est possible aussi d'appeler "gardien virtuel") : en fonction des valeurs contenues dans cette matrice le gardien sera plus ou moins bon.

Le rôle de l'algorithme génétique sera de sélectionner "le meilleur gardien", c'est-à-dire d'optimiser les valeurs de la matrice pour que le gardien arrête le plus de buts possibles, cela revient donc à sélectionner l'individu qui possède le meilleur génotype.

b) Sélection du "meilleur individu" grâce aux algorithmes génétiques.

Notre algorithme génétique suit un modèle tout à fait conventionnel. L'algorithme commence avec la création d'une population d'individus (robots virtuels).

Chaque individu a donc son patrimoine génétique contenu dans une matrice de 60×60 (puisque la table de babyfoot a une largeur de 60 cm). La longueur de ces « chromosomes » est donc établie et invariable. Les « allèles » ou valeurs contenues dans la matrice sont cependant totalement aléatoires dans la première génération (les valeurs correspondent à la position que prend l'individu "gardien virtuel" dans le but de largeur 20 cm).

Une fois que l'on crée une population initiale, on commence des « sessions de test » pour pouvoir mesurer les performances de chaque individu de la génération. Cette étape se fait par simulation sur NetLogo et elle repose sur une fonction qui mesure les performances grâce à une série de tirs (1200 tirs par session par individu, en général). Chaque individu a donc un « score ». Ces séries de tirs s'effectuent pour tous les individus de la génération.

La première méthode pour mesurer les performances fonctionne avec ces règles : si l'individu arrête un tir, il reçoit un point. Sinon, il ne reçoit pas de points. Le score final est déterminé par la moyenne de tous les scores obtenus pour chaque tir de la session. Cette méthode revient à des scores « binaires » et à un score final entre 0 et 1. Après quelques essais, on s'aperçoit que cette méthode ne réussit pas à bien discerner les différentes performances des individus ; en effet, cette idée de performance binaire est trop radicale et à la fin des sessions de tests les scores des individus de la génération s'avèrent quasiment identiques.

On doit alors trouver une méthode qui permette de récompenser les individus qui sont « proches » de la solution (arrête « presque » un tir) pour chaque tir.

Pour cela, nous avons créé une nouvelle fonction qui permet de mesurer les performances des individus de façon plus fine :

$$\text{Score final obtenu après une série de } n \text{ tirs simulés sur NetLogo} = \frac{1}{n} \sum_{i=0}^n \frac{1}{1 + |y_{\text{gardien}_i} - y_{\text{balle}_i}|}$$

y_{gardien} et y_{balle} correspondent aux positions respectives du gardien et de la balle lorsque la balle se trouve dans l'abscisse des buts (c'est-à-dire lorsque la balle passe la ligne de but).

n correspond au nombre de tirs simulés par session pour chaque individu (généralement aux alentours de 1200 tirs par individu).

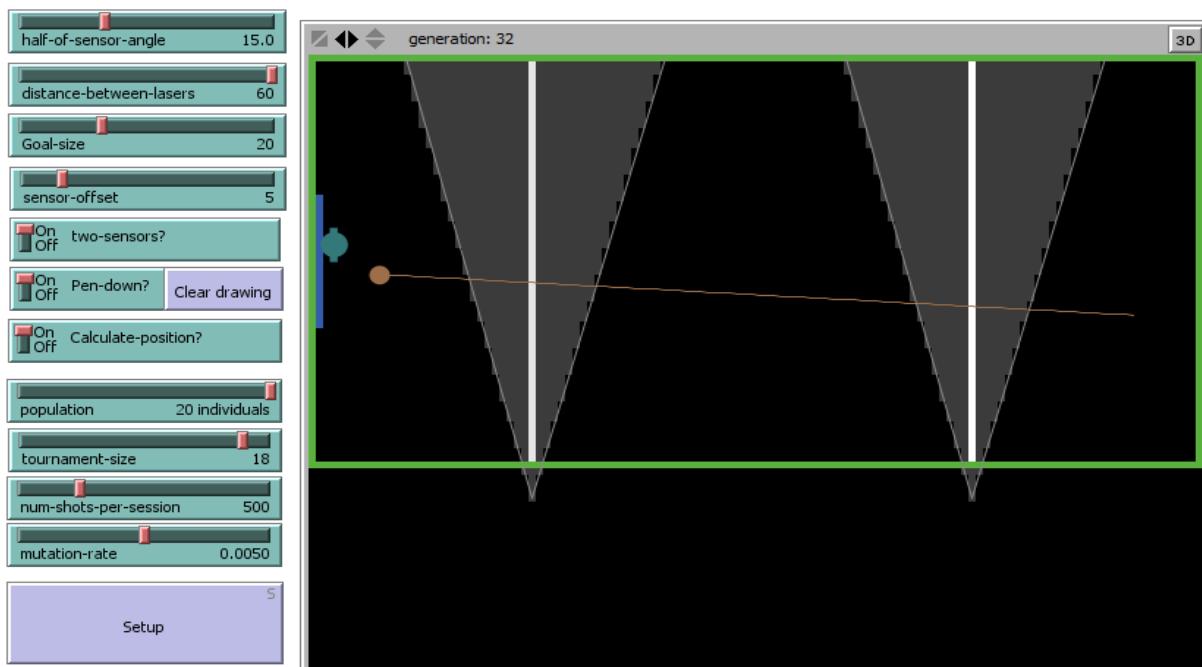
Cette nouvelle méthode ne mesure donc pas si le gardien virtuel arrête ou pas les tirs mais elle permet de mesurer les performances de chaque individu de manière beaucoup plus fine. Ainsi, un bon score dépend de la taille de la balle et du gardien.

Cette fonction permet d'affecter un score proche de 0 à un très mauvais placement du gardien virtuel, un score égal à 1 à un placement exact mais grâce à cette fonction, tous les scores intermédiaires entre 0 et 1 sont aussi possibles, d'où une meilleure finesse dans l'évaluation des individus.

Exemple pour une série de 3 tirs ($n = 3$):

- tir 1 : $y_{gardien} - y_{balle} = 0 \Rightarrow \text{score (tir 1)} = 1$
- tir 2 : $y_{gardien} - y_{balle} = 1 \text{ cm} \Rightarrow \text{score (tir 2)} = 0,50$
- tir 3 : $y_{gardien} - y_{balle} = 2 \text{ cm} \Rightarrow \text{score (tir 3)} = 0,33$

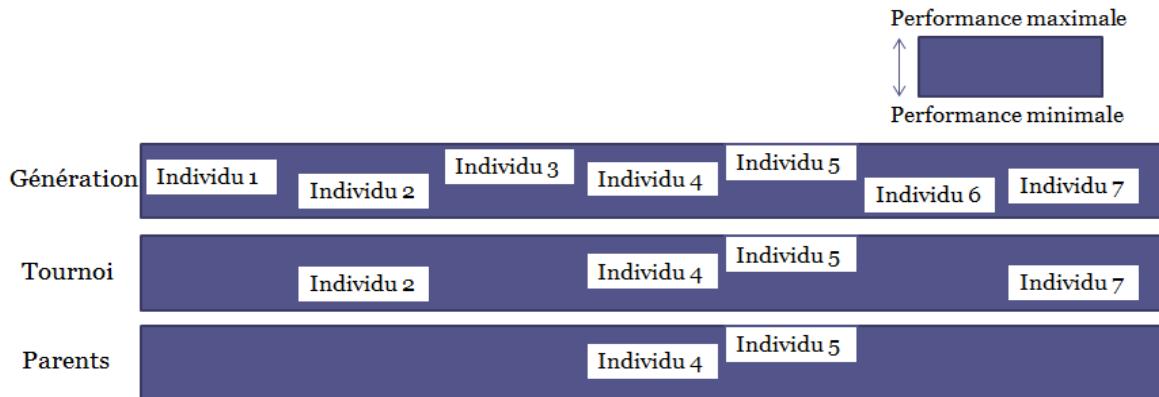
Score final = Moyenne des scores = $(1 + 0,50 + 0,33)/3 = 0,61$



Document 16 : Copie d'écran d'un test sur Netlogo

Une fois que ces sessions de test finissent, on procède à l'étape de « sélection ». Dans ce programme, on utilise la technique de « Sélection par tournoi ». Cette méthode consiste à prendre au hasard un nombre d'individus (ce nombre d'individus est une variable définie au début de l'algorithme) indépendamment de leurs résultats pour créer un groupe, le tournoi. Puis on sélectionne les deux meilleurs individus de ce tournoi.

Cette technique présente quelques avantages. Elle permet de rendre la sélection moins brutale et plus réaliste. Ainsi, les deux meilleurs individus de la génération ne deviennent pas toujours les parents même s'ils ont plus de chances d'être sélectionnés. Un facteur aléatoire est donc rajouté au programme, ce qui le rend plus vraisemblable biologiquement. Mais avant tout, il s'agit d'une méthode pour garder une certaine diversité génétique et éviter une convergence trop importante vers un même patrimoine génétique. Ainsi, plus le tournoi est petit, plus les individus avec les résultats les plus bas ont des chances d'être sélectionnés. Au contraire, si le tournoi se compose par exemple, du même nombre d'individus que la génération (population totale), les deux meilleurs individus seront toujours sélectionnés.



Document 17 : Schéma explicatif d'un tournoi

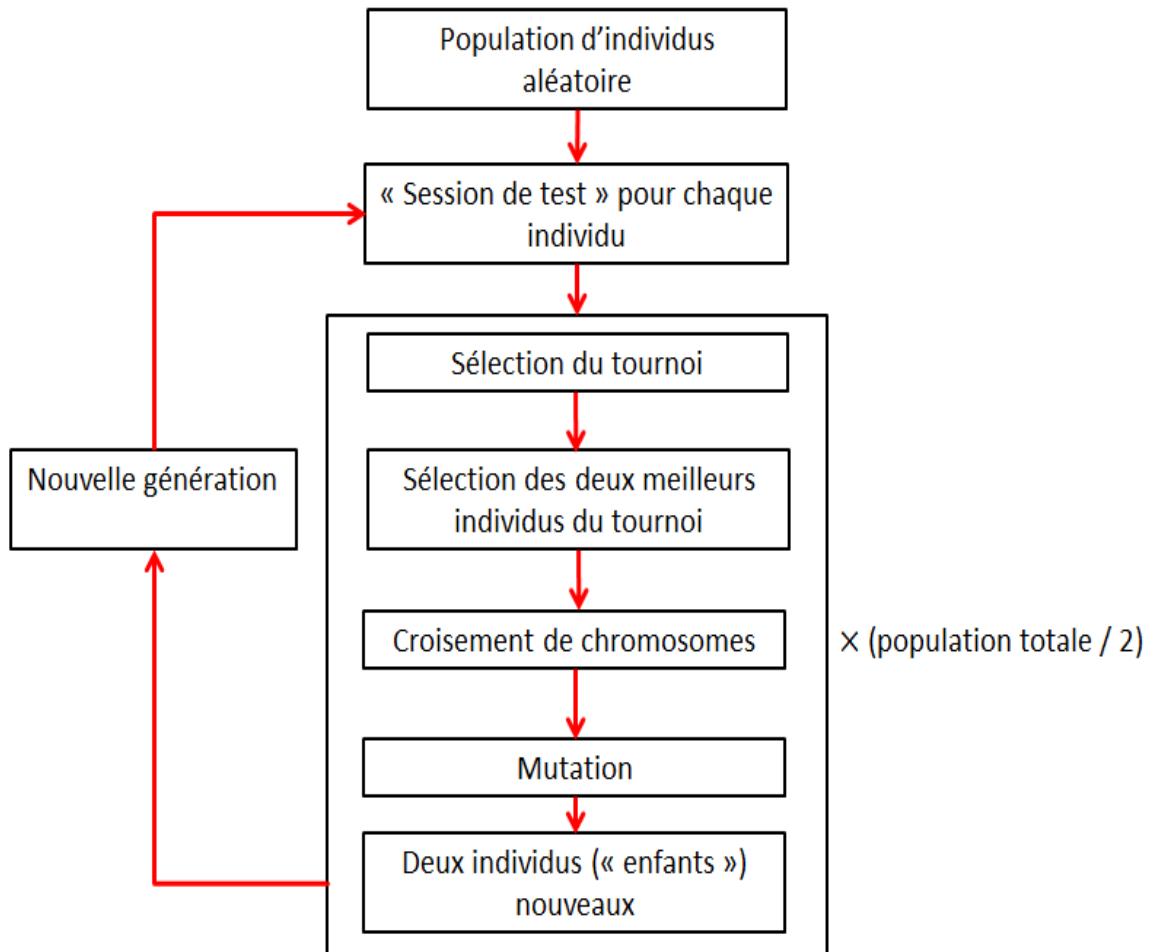
Les deux meilleurs individus du tournoi deviennent les « parents ». On procède donc au croisement des chromosomes. Pour cela, on prend les « colonnes » des matrices des deux parents, dans l'ordre. Puis on détermine un point aléatoire où l'on coupe la liste. Les différents « bouts de chromosomes » sont ainsi échangés, pour former deux nouvelles colonnes. Cette procédure se répète pour chaque colonne du code génétique (ou de la matrice) pour aboutir dans la création de deux nouveaux génotypes (matrices), qui représentent deux nouveaux individus.



Document 18 : Croisement des matrices

La dernière étape est la mutation. Chaque « allèle » ou valeur dans la matrice a une probabilité de changer par une autre valeur aléatoire. Le taux de mutation varie entre 0 et 0,01. Cette variable est essentielle dans le programme et ses effets sur le comportement du programme et de l'évolution des individus ne sont pas négligeables. Il est ainsi très important de trouver un taux de mutation optimal. Plusieurs comparaisons de taux de mutations seront présentées dans les résultats. La mutation permet de maintenir une certaine diversité génétique et d'éviter une convergence trop rapide. Elle doit cependant rester assez basse pour ne pas créer un algorithme basé que sur des changements aléatoires (et perdre tout le travail de « sélection naturelle ».)

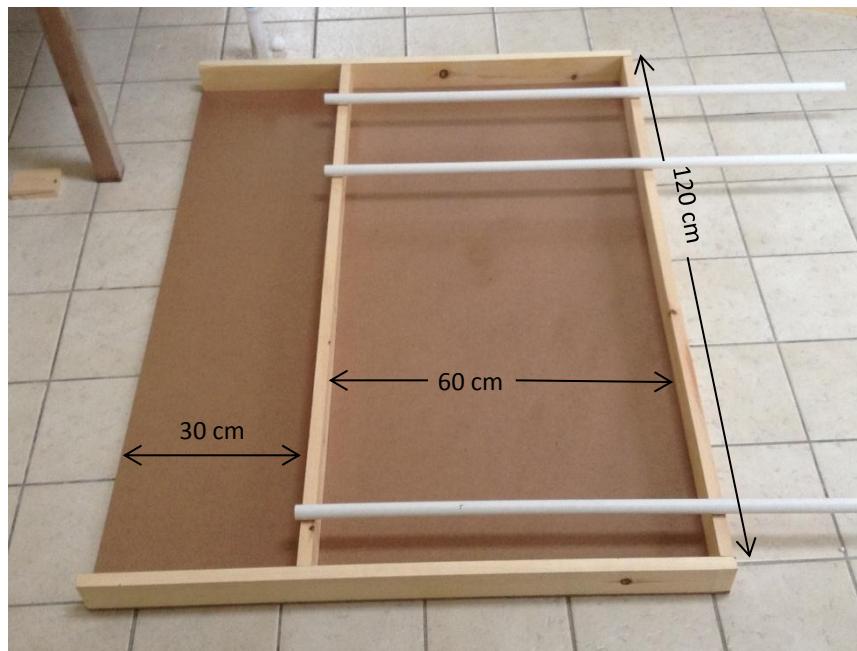
Ce processus consiste alors à créer un tournoi, choisir les deux meilleurs individus du tournoi pour qu'ils deviennent les parents, croiser leurs chromosomes, puis les muter. À chaque nouvelle génération ce processus est répété. Une fois que la nouvelle génération se crée, on recommence le processus avec les « sessions de test », et ainsi de suite...



Document 19 : Schéma récapitulatif du processus de sélection

5) Application pratique: construction du babyfoot

Pour construire le babyfoot, on a choisi un matériel qui répondait à nos besoins et facile à se procurer, le bois. À partir de là, on a élaboré un plan, détaillé du babyfoot. L'aire de jeux, c'est-à-dire l'espace où la balle pouvait rouler, allait être proche des dimensions réelles d'un babyfoot, mais un peu plus petit, pour des raisons pratiques (transport, poids), mais aussi à cause de certaines contraintes imposées par les capteurs. De plus, on désirait avoir un espace sur le côté où l'on puisse mettre tout le système électronique. Après une excursion chez le menuisier, ils ne nous restaient qu'à entreprendre la construction...



Document 20 : La caisse en bois du baby-foot

La construction a fait l'objet de nombreuses rectifications, les installations mécaniques et électriques changent de prototype en prototype dans le but de trouver la meilleure organisation de notre babyfoot. Dans un premier temps, tout était mobile et détachable, on pouvait facilement changer les capteurs de place, ajuster l'intensité des lasers...

Résultats

1) Résultats concernant les capteurs

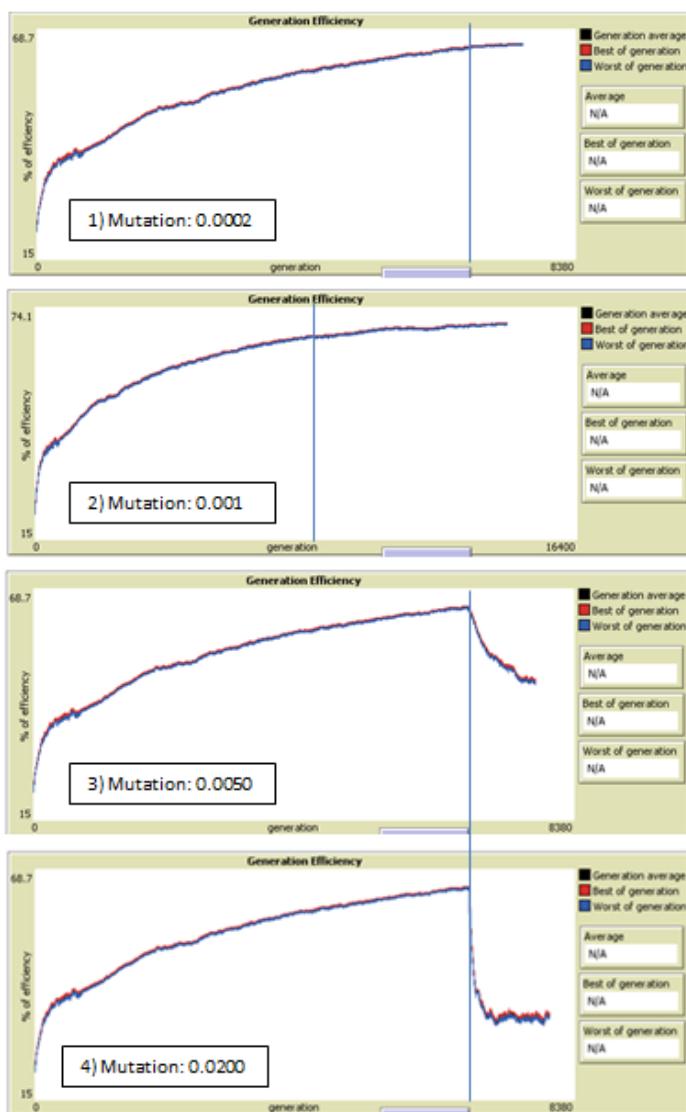
Lors de nos premiers tests, nous nous sommes rendu compte que les capteurs ne pouvaient pas détecter systématiquement notre balle de babyfoot officielle (celle en liège compressé)! Le problème était d'importance, nous avons donc réalisé une série de tests statiques, où, petit à petit on a rapproché la balle du capteur. La balle était légèrement trop petite mais surtout peu dense. Elle absorbait une grosse partie des ultrasons et n'en réfléchissait donc pas assez. Elle était détectée qu'à environ 18 cm du capteur. Logiquement, il nous fallait une autre balle, plus dense plus grosse. On a donc essayé plusieurs balles (une balle de ping-pong, de baseball, de billard, d'acier, de golf et de tennis). La balle de billard et de baseball, en raison de leur taille et de leur densité, donnèrent de bons résultats, mais elles étaient trop grosses, et leurs proportions comparées à celle d'une table de babyfoot normale n'étaient donc plus compatibles. Toutes les autres balles n'ont pas donné de résultats satisfaisants, à l'exception de celle en acier, mais elle était trop petite. Il nous fallait donc une balle en métal plus grosse, à l'idéal en l'aluminium en raison du poids.

Par exemple, pour la présentation de notre projet lors de la phase académique des olympiades, nous avons utilisé une boule de billard.

2) Résultats des algorithmes génétiques

D'emblée, on passe d'une performance de l'ordre du 20% à la génération 0 (et donc complètement aléatoire) à une performance du 70% dans le meilleur des cas que l'on a pu simuler, après 14288 générations et 43 heures de simulation.

La première comparaison se centre sur l'influence des taux de mutation dans un état avancé du programme. Toutes les autres variables restent constantes.



Document 21 : Etude de l'importance du taux de mutation

plus prononcée, après une multiplication du taux de mutation par 20. Cette expérience montre bien l'impact du taux de mutation dans l'évolution à travers les générations, même si ce taux est très faible et statistiquement, ne modifie pas plus que 2% (quatrième cas) des valeurs ou « allèles » du code génétique d'un individu.

Le taux de mutation ne doit pas être très élevé, mais il est cependant essentiel pour l'algorithme. Ainsi, on a commencé une simulation avec un taux de mutation de 0. Les performances n'augmentent pas beaucoup. Puis, lorsqu'on augmente le taux de mutation (au 0.01), les performances commencent à évoluer beaucoup plus rapidement.

Cette simulation commence avec une population de 10 individus, un tournoi de 10 individus également (donc, les deux meilleurs individus de la génération seront toujours les parents, ce qui permet d'enlever une variable aléatoire et pouvoir faire une meilleure comparaison), et 1200 tirs différents sont effectués pour déterminer les performances d'un individu (les tirs ne sont pas aléatoires, chaque individu sera donc évalué sous les mêmes conditions ou les mêmes tirs).

La simulation commence avec un taux de mutation de 0.001. Le graphique 2 est donc le graphique de référence. (Remarquez que le graphique 2 porte sur une plus longue durée, 16400 générations, alors que les autres finissent à 8300 générations). La ligne bleue marque le changement des taux de mutation aux différentes valeurs affichées. Ce changement se fait dans la génération 6730, alors que la performance moyenne de la génération est du 64.4%. Le changement n'est pas très important si l'on compare le premier et le deuxième graphique. Cependant, lorsqu'on multiplie le taux de mutation par 5, (graphique 3), les performances chutent brutalement et semblent se stabiliser aux alentours des 35%.

Dans le quatrième cas, la chute est encore

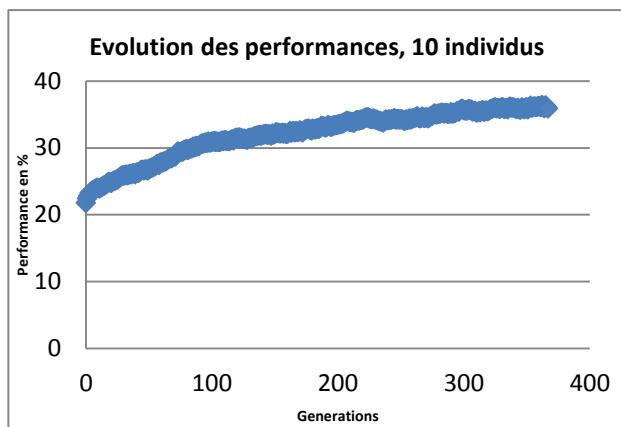
plus prononcée, après une multiplication du taux de mutation par 20. Cette expérience montre bien

l'impact du taux de mutation dans l'évolution à travers les générations, même si ce taux est très faible et

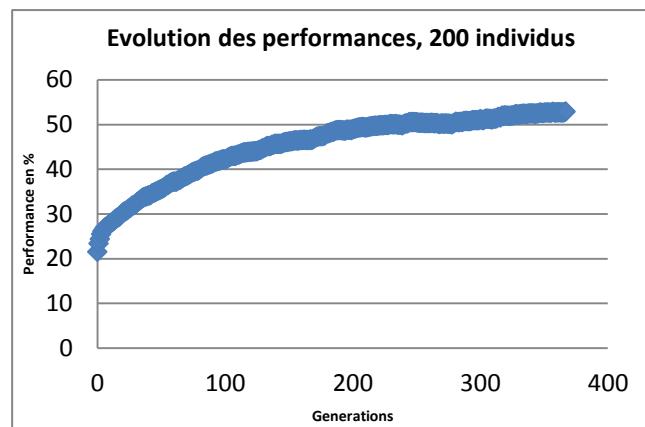
statistiquement, ne modifie pas plus que 2% (quatrième cas) des valeurs ou « allèles » du code génétique

d'un individu.

Un autre paramètre essentiel important est celui du nombre d'individus dans la génération. Pour comparer ce paramètre, on a comparé l'évolution de deux environnements. L'un avec 10 individus et l'autre avec 200 individus. Les autres variables restent les mêmes.



Document 22 :
Etude avec 200 individus par génération



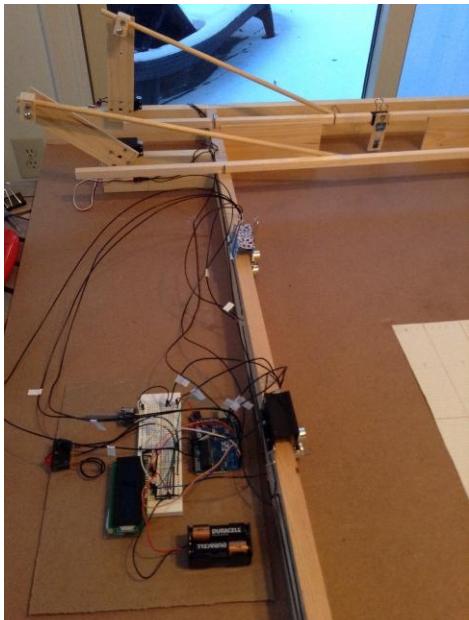
Document 23 :
Etude avec 10 individus par génération

Conclusions

Les résultats liés aux capteurs ultrasons sont pour l'instant satisfaisants : grâce aux conseils de Mrs Lacalmontie et Islam, nous avons réussi à augmenter la distance maximale de détection de la balle. Nous avons toujours un problème lorsque la balle va trop vite : dans ce cas, la distance maximale de détection est plus limité (18 cm). Des expériences sont toujours en cours pour obtenir un bon repérage de la balle dans des conditions de plus en plus difficiles.

Les résultats avec la caméra sont satisfaisants, et ont permis de suivre les coordonnées de la balle. Cependant, des essais de connexion avec la puce n'ont pas encore été effectués. Des études plus approfondies sont donc nécessaires pour tirer des conclusions sur l'efficacité de cette méthode.

Le document suivant montre l'état de la maquette au jour où nous finalisons ce mémoire :



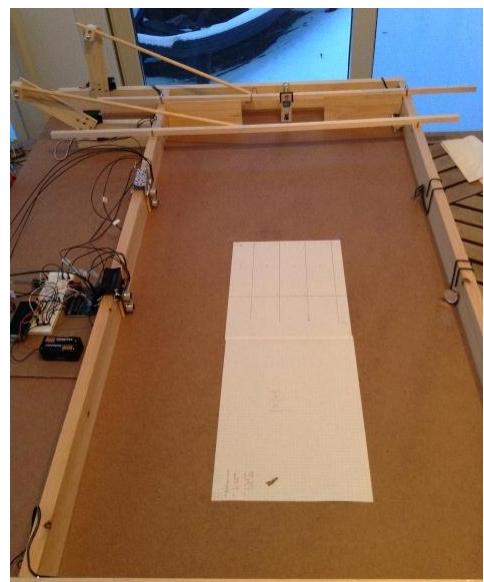
Document 24a : Montage électronique au premier plan. Le gardien en fond. Nous sommes en train de rajouter un défenseur.



Document 24b : La liaison mécanique entre le servomoteur et le gardien.



Document 24c : Le gardien vu de face. La barre au premier plan est celle du futur défenseur.



Document 24d : vue d'ensemble de la maquette.

Les algorithmes génétiques nous donnent des résultats assez intéressants. En effet, ils remplissent leur fonction de trouver des solutions au problème posé et donc d'optimiser les arrêts du gardien.

On pense que de meilleurs résultats (supérieurs au 70 % actuel) pourraient être obtenus avec d'autres paramètres. L'étude comparative de différents paramètres nous a, en effet, permis de mieux comprendre le comportement de la simulation. Les variables sont essentielles pour le déroulement de la simulation. L'évolution est donc très sensible à ces paramètres.

On peut en conclure qu'il n'existe pas de taux de mutation optimal pour rendre plus efficace et plus rapide l'algorithme génétique. Le taux de mutation devrait constamment changer en fonction de l'état du programme. En effet, après quelques études, on s'est aperçu que les taux de mutations élevés étaient bien plus efficaces aux débuts des simulations. Et au contraire, comme démontré par les graphiques déjà présentés, le taux de mutation doit diminuer lorsque les performances deviennent plus importantes. Quant aux autres paramètres, on peut en déduire qu'une grande population est bien plus avantageuse qu'une population moins nombreuse. En effet, une population importante donne une grande stabilité génétique à la génération, ce qui la rend moins vulnérable aux accidents aléatoires causés par les mutations, tout en gardant une grande diversité, nécessaire pour trouver de nouveaux résultats plus innovants.

Il serait très intéressant de pouvoir créer une sorte « d'algorithme génétique dans l'algorithme génétique », une mise en abîme pour pouvoir retrouver les paramètres (taux de mutation, tournoi...) les plus adaptés par rapport à la situation du programme.

Ces résultats prouvent que l'utilisation des algorithmes génétiques pourrait être très utile pour des projets encore plus ambitieux. Ils pourraient en effet jouer un rôle primordial dans le cadre d'un robot babyfoot complet (avec 11 joueurs). On pourrait également adapter les algorithmes pour que le robot puisse évoluer en temps réel, en affrontant des adversaires humains, ce qui rendrait les résultats encore plus proches de la réalité pratique.

Bibliographie

Wilensky, U. (1999). NetLogo. <http://ccl.northwestern.edu/netlogo/>. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.

Sources:

<http://www.complexityexplorer.org/> et cours développés par le [Santa Fe Institute](http://www.santafe.edu/) (<http://www.santafe.edu/>)

<http://arduino.cc/en/Main/arduinoBoardUno>

http://en.wikipedia.org/wiki/Genetic_algorithm

<http://www.geogebra.org/cms/en/>

Sources des illustrations :

Document 2 : <http://arduino.cc/en/Main/arduinoBoardUno>

Document 3 : <https://www.sparkfun.com/products/242>

Document 7 :

<http://www.variohmholdings.com/assets/press/1268671158.jpg>

<http://www.tjsolution.com/products/Laser%20displacement%20sensor/Main%20laser%20displacement%20sensor.html>

Document 10 : http://en.wikipedia.org/wiki/File:Sonar_Principle_EN.svg