



Babyfoot : Vision et Stratégie

Laboratoire d'Automatique

Supervisé par :

PROF. CHRISTOPHE SALZMANN



Mahdi Ihdeme

Génie Mécanique - Master semestre 3
École Polytechnique Fédérale de Lausanne

20 / 12 / 2021

Table des matières

1	Introduction	2
1.1	Objectifs	2
1.2	Structure mécanique	2
1.3	Structure générale du code	3
2	Stratégies d'attaque et de défense	4
2.1	DefenseAllPlayers	4
2.1.1	Entrées	4
2.1.2	Sortie	5
2.1.3	Processus	5
2.1.4	Limites	5
2.2	ShootDynamic	6
2.2.1	Entrées	6
2.2.2	Sortie	6
2.2.3	Processus	6
2.2.4	Limites	7
3	Position de la balle : Prédiction	7
3.1	Solution existante	7
3.2	Nouvelle prédiction : Régression Linéaire Conditionnée	9
3.2.1	PredY.VI	10
3.3	Résultats et discussion	12
4	Stratégie de défense : Suivi de la prédiction	13
4.1	Non-interception à vitesse élevée	14

5 Conclusions et pistes d'amélioration	18
Références	19

Résumé et contenu

Le but principal de ce projet de semestre a été découvrir les stratégies de contrôle réalisées au cours des projets précédents et d'affiner la prédiction de la position de la balle afin d'améliorer la trajectoire des joueurs dans les différentes configurations. L'interception, par exemple, est une étape nécessaire dans un bon nombre de stratégies, nécessitant une prédiction à haute précision de la trajectoire la balle.

Points essentiels

- **Compréhension de la structure générale du code** : Première étape du processus, permettant de situer la partie du code responsable de chacun des processus.
- **DefenseAllPlayers & ShootDynamic** : Découverte et évaluation des deux stratégies les plus récentes de défense et d'attaque.
- **Prédiction de la position de la balle** : Étude de la solution existante et peaufinage de la prédiction.
- **Défense** : Nouvelle stratégie de défense basée sur les nouvelles prédictions.

Remerciements

Je tiens à exprimer ma profonde gratitude à l'égard du Professeur Salzmann, qui a toujours été d'une grande assistance durant tout le projet. Ses précieux conseils et sa disponibilité ont permis de mener ce travail de semestre vers la bonne direction et de résoudre un grand nombre des problèmes rencontrés, à travers sa connaissance parfaite de l'état du système, ses points forts et ses défauts.

1 Introduction

Le système en question est une table de Babyfoot automatique, disposant de moteurs linéaires et de servomoteurs pour le mouvement en translation et en rotation des barres des joueurs, respectivement.

Un grand nombre de projets autour du système ont permis de mettre en place une base solide dans tous aspects nécessaire à son fonctionnement, que se soit au niveau du **hardware** (Moteurs, cartes d'acquisition...) ou des **algorithmes** (Vision, contrôle et stratégie). L'ensemble du code pour l'ensemble de ses parties est écrit en LabVIEW, avec quelques fois des scripts Matlab qui interviennent lors de calculs conséquents.

1.1 Objectifs

Dans ce projet, on s'est tout d'abord intéressé à la compréhension du code derrière le mécanisme. Ensuite, il a été question d'étudier les différents algorithmes de jeu conçus auparavant pour enfin partir sur une piste d'amélioration et d'établir de nouvelles perspectives pour des développements éventuels.

1.2 Structure mécanique

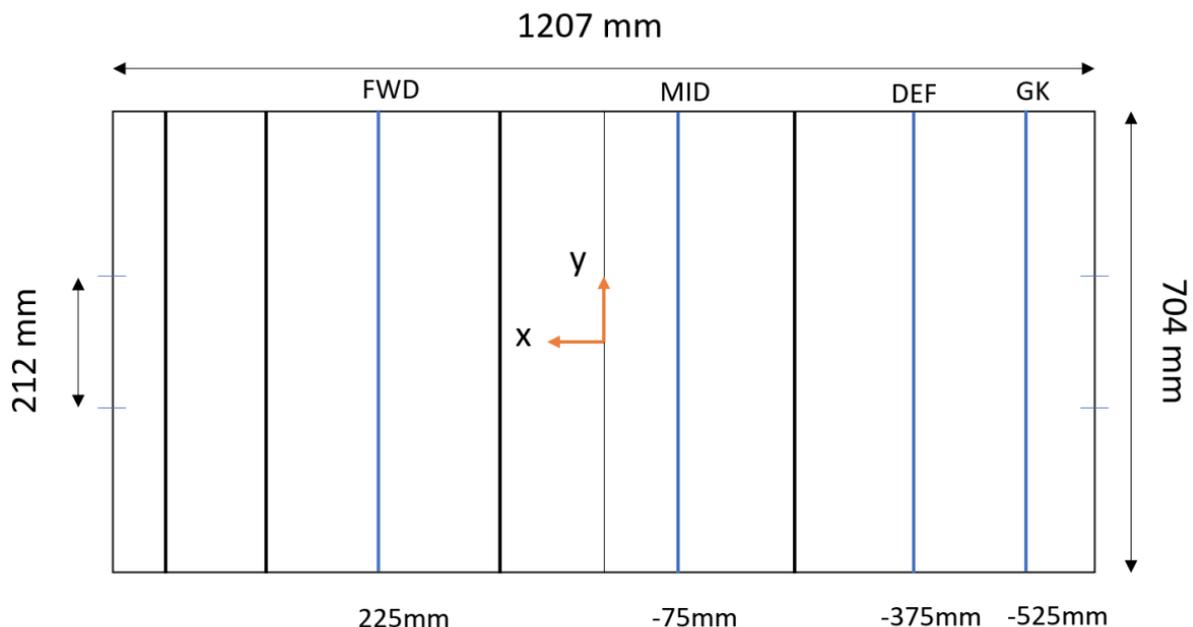


FIGURE 1 – Dimensions de la table de Babyfoot

La structure de babyfoot dont dispose le laboratoire d'automatique est similaire à une table ordinaire, avec 4 rangées de joueurs de chaque côté.

Les rangées robotisées sont indiquées en bleu sur la figure 1 : Attaquant (FWD), milieux de terrain (MID), défenseurs (DEF) et gardien de but (GK). Les barres peuvent être dirigées en translation et en rotation. Le déplacement admissible est différent pour chacune des barres, vu qu'elles ne disposent pas du même nombre de joueurs.

Différents types de capteurs sont utilisés pour contrôler la machine. Deux caméras situées sous le plateau en plexiglas sont utilisées pour traquer le mouvement de la balle. La différence entre les deux réside en la résolution et le nombre d'images fournies par seconde. Les angles des tiges contrôlées sont mesurés avec des encodeurs angulaires, montés sur les tiges. Quant aux déplacements latéraux, ils sont directement mesurés à partir des capteurs du moteur.

Quant aux barres contrôlées par le joueur humain, elles sont dotées de laser mesurant la translation de ces dernières. Cette mesure peut servir d'entrée de commande pour les tiges robotiques, permettant d'avoir connaissance de la position des joueurs adverses en chaque instant.

1.3 Structure générale du code

La compréhension du code responsable du fonctionnement du mécanisme présente une tâche assez conséquente. On pourrait dénombrer deux grandes parties, assurant l'acquisition des données et leur utilisation dans des stratégies de contrôle diverses :

- **Boucle de vision :** C'est la partie du code responsable du traitement des données provenant de la caméra. On y retrouve par exemple les algorithmes pour la détection de la balle mais aussi pour la transformation des informations des images en des données plus pertinentes, à savoir le passage de pixels représentant la balle à sa position en millimètres ou encore l'estimation de la vitesse de la balle à travers les mesures.
- **Boucle de contrôle :** Dans cette boucle, il s'agit des différentes stratégies de contrôle et de jeu développées au cours du développement du système. On y retrouve un nombre assez conséquents d'algorithmes, utilisant les données mises en place par la boucle de vision.

Après avoir pris connaissance de la structure générale, son fonctionnement et de la communication entre ses deux parties principales, on s'est attaqué à l'étude des stratégies les plus récentes en termes de tactiques d'attaque et de défense dont dispose le système.

2 Stratégies d'attaque et de défense

2.1 DefenseAllPlayers

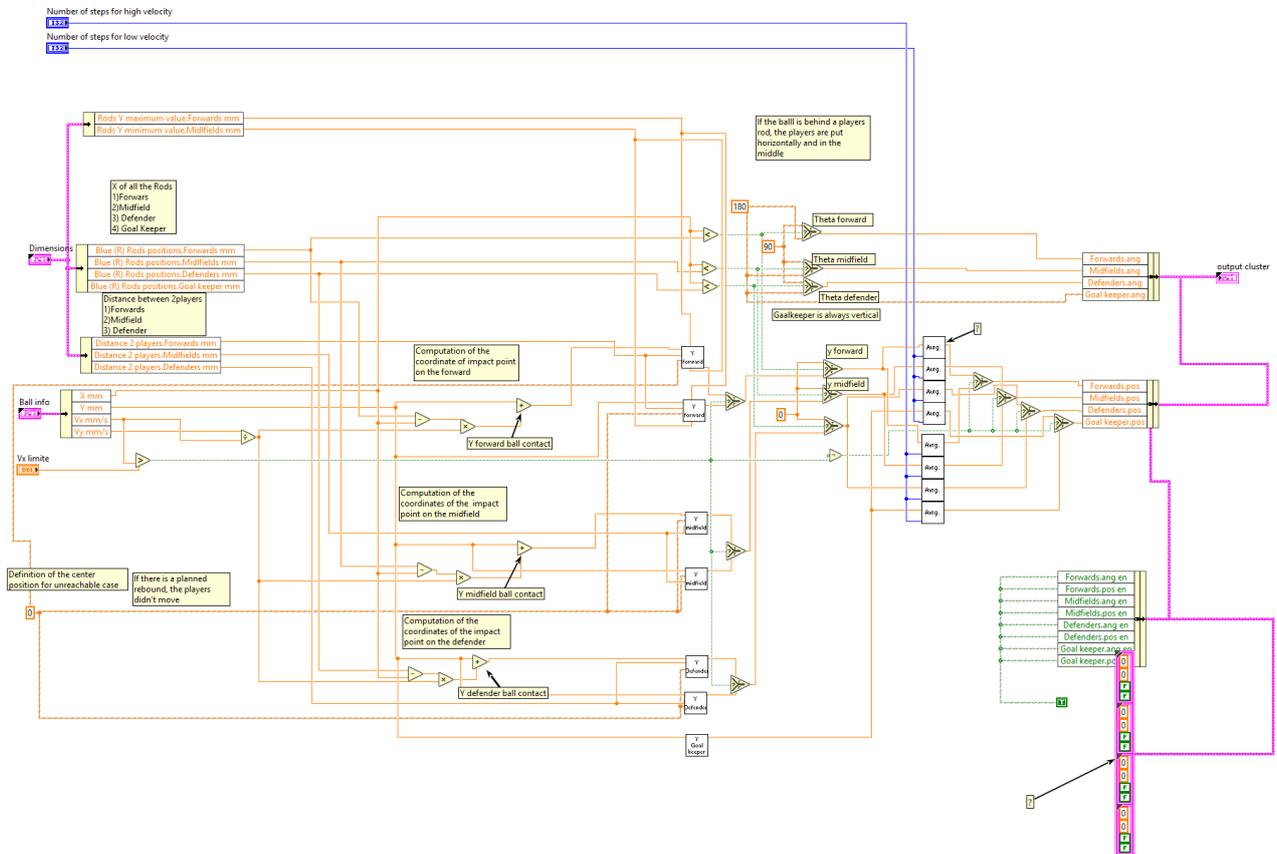


FIGURE 2 – DefenseAllPlayers - Stratégie de défense

Ce VI est la principale stratégie de défense dont on dispose pour le moment. Il consiste à positionner les joueurs en fonction de la position de la balle pour bloquer les tirs de l'adversaire, en vérifiant une série de conditions [1].

2.1.1 Entrées

- **Dimensions** : VI regroupant toutes les grandeurs significatives du mécanisme, que ce soit en terme des dimensions de la table, les distances entre les joueurs ou leur positionnement relatif à un repère absolu.
- **Ball Info** : Il s'agit du cluster contenant les informations élaborées à travers la boucle de vision. Il met à disposition les positions en x et en y de la balle dans le repère et l'estimation des vitesses v_x et v_y .
- **Vx Limit** : C'est l'estimation de la vitesse limite de la balle en x .

2.1.2 Sortie

- **Output cluster** : Positions désirées des joueurs en angle et en position.

2.1.3 Processus

Le VI prend les entrées décrites ci-dessus et, pour chaque rangée de joueurs (défenseurs, milieux de terrain et attaquants), prends la position de la balle pour en déduire le joueur qui est censé l'intercepter, puis calcule la commande du moteur.

Cette procédure est basée sur deux conditions principales :

- $v_x > v_{xlim}$: La commande du moteur (y) pour chaque rangée de joueurs est calculée à partir du vecteur vitesse de la balle (utilisée également pour calculer la coordonnée du point d'impact), la position du centre de l'axe des joueurs, la distance entre deux joueurs et les valeurs minimales et maximales de la tige.
- $v_x \leq v_{xlim}$: La commande du moteur (y) pour chaque rangée de joueurs est calculée en fonction du point d'impact, qui est défini comme étant la coordonnée y de la balle dans ce cas, de la position du centre de l'axe des joueurs, de la distance entre deux joueurs d'une rangée et des valeurs minimale et maximale de la translation de la tige.

L'étape suivante est formulée sous une autre condition qui stipule que si la balle est derrière une tige de joueur, les joueurs sont placés horizontalement et au milieu de la table.

Ensuite, les positions souhaitées calculées sont moyennées en utilisant le nombre de pas pour la vitesse élevée/faible afin de calculer le cluster final de sortie.

2.1.4 Limites

L'avant dernière étape du processus, à savoir le positionnement des joueurs horizontalement si la balle se retrouve derrière une rangée induit un problème majeur. En effet, il a été remarqué qu'elle induit dans une grande partie des cas une propulsion de la balle à cause du mouvement brusque des joueurs vers l'arrière, aspect qui non seulement mène vers une grande variabilité de la position de la balle, mais qui peut aussi nuire à la stabilité du système avec la succession de mouvement brusques engendré par cette condition. D'autre part, on peut aussi constater que le gardien a une réactivité très limitée.

Un autre problème réside en le fait que l'algorithme suit simplement la position de la balle en chaque instant, en ne prenant aucun compte de la trajectoire de la balle. Il n'y a en effet aucune utilisation d'une quelconque prédiction de la position de cette dernière. Il a été remarqué que cette stratégie peine à intercepter la balle quand il s'agit de trajectoires à grand angle ou quand il y a un rebond contre les coins de la table, ou lorsque la vitesse de la balle est assez élevée.

2.2 ShootDynamic

Cette démarche est le processus le plus à jour dont on dispose pour la stratégie d'attaque du système. Il s'agit ici d'un modèle capable de tirer au but avec un degré élevé de précision, prenant en compte le positionnement du gardien de but adverse et pouvant tirer en translation et/ou en rotation [1].

2.2.1 Entrées

Le VI reçoit les mêmes informations que la stratégie de défense décrite précédemment, en plus de la période d'échantillonnage du système ($2ms$). Il est en effet doté du VI `DefenseAllPlayers` pour la défense.

2.2.2 Sortie

- **Output cluster** : Positions désirées des joueurs en angle et en position.

2.2.3 Processus

Le but de cette stratégie est de pouvoir prendre en considération la position des joueurs adverses avant de tirer, permettant des tirs précis à angle variable. Pour ce faire, il a été nécessaire de calculer pour chaque pas de temps le déplacement nécessaire pour un angle donné.

Le VI se sert de 4 sous-VI principaux :

- **DefenseAllPlayers** : Stratégie de défense.
- **ShootDynamic** : Stratégie d'attaque basée sur le calcul du déplacement nécessaire.
- **est Tir** : Décision de tir, sélectionne le joueur qui doit intercepter et tirer la balle pour chaque rangée de joueurs.
- **Add Command** : Calcule la commande de sortie en se basant sur les positions initiales issues de la stratégie de défense et celles désirées.

L'innovation principale qu'apporte cette stratégie réside en le fait de combiner des calculs trigonométriques simples à une régression polynomiale afin de calculer le déplacement nécessaire en y pour un angle désiré. La méthode construit trois modèles de tir : Tir en translation, en rotation et finalement la combinaison des deux.

Après de nombreux essais avec les différents modèles construits pour chaque méthode, il a été déterminé que la méthode de rotation est la plus précise pour les angles compris entre 30° et 150° . Quant aux angles compris entre 0° et 30° et entre 150° et 180° , le modèle qui arrive aux meilleurs résultats est celui la méthode de tir en translation, qui repose uniquement sur les calculs trigonométriques.

Enfin, cette méthode est très efficace et parvient à des précisions de tir satisfaisantes pour des vitesses modérées.

2.2.4 Limites

Deux problèmes principaux ont été identifiés concernant cette stratégie :

- La stratégie d'attaque est déployée pour l'ensemble des joueurs. Les défenseurs et le gardien de but n'ont en effet pas tellement d'opportunités de tir. Après une multitude de tests, il s'est avéré que les conditions sur la fenêtre de tir ne sont presque jamais satisfaites pour ces derniers.
- La complexité de la stratégie entraîne des mouvements très brusques de la table, ce qui pourrait éventuellement mener vers une décalibration de la caméra.

3 Position de la balle : Prédiction

La stratégie de défense décrit dans les parties précédentes, comme souligné précédemment, présente un problème majeur : La méthodologie est entièrement basée sur la position actuelle de la balle, sans aucune prédiction ou mesure de sa trajectoire. Cette complication entraîne une réactivité limitée du système.

Dans ce qui suit, on présente une méthodologie de prédiction de trajectoire basée sur un certain nombre de conditions qui ont été élaborées en se basant sur le comportement du système après une série de tests.

3.1 Solution existante

Un des étudiants durant un projet précédant a élaboré une méthode pour la prédiction de la position de la balle basée sur une **régression linéaire** des dix positions passées, l'hypothèse principale étant la linéarité de la trajectoire de la balle [2].

Pour éviter d'avoir des prédictions erronées ou non nécessaires, une série de conditions a été mise en place :

- $250 \text{ mm/s} < \|v\| < 8000 \text{ mm/s}$: La norme de la vitesse de la balle devrait être raisonnable.
- $v_x < 0$: On impose que la vitesse selon x doit être négative, afin de s'assurer que la balle va vers les joueurs dans la direction désirée ; celle contraire à l'axe x 1.
- $|y_{\text{pred}}| < 350 \text{ mm}$: La prédiction devrait être délimitée par la moitié de la largeur de la table.
- $d_{\text{ball} - \text{player}} > 25 \text{ mm}$: C'est la condition posée sur la distance entre le centre de la balle et l'axe d'un joueur donné, la valeur de 25 mm étant le cas où la balle est collée au joueur.

Évaluation

Cette méthode a été testée dans différentes configurations afin de pouvoir mettre au clair sa performance.

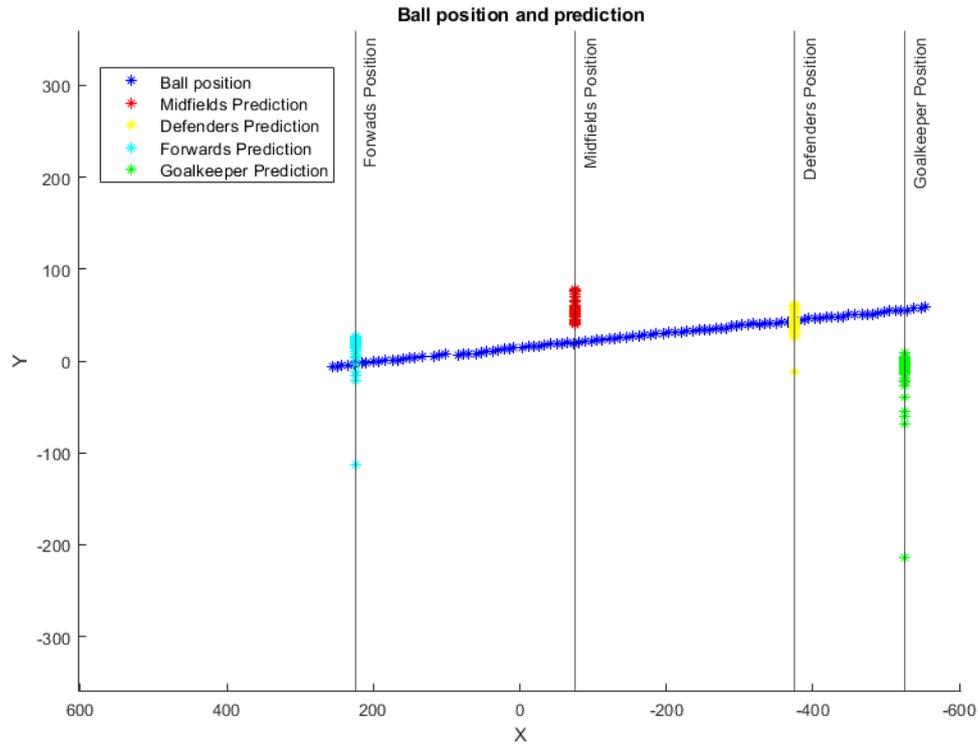


FIGURE 3 – Prédictions de la position en y de la balle - Petit angle

La figure ci-dessus montre les résultats de la prédiction pour un tir à petit angle. Il est remarquable que la prédiction n'est pas au point, surtout lorsqu'il s'agit des milieux de terrain et du gardien de but. On retrouve aussi des valeurs aberrantes qui sont loin de la trajectoire réelle de la balle.

TABLE 1 – Statistiques - Petit angle [mm]

	Prédiction Attaquants	Prédiction Milieux	Prédiction Défenseurs	Prédiction Gardien
Moyenne	4.5355	42.4185	29.7928	-20.7199
Ecart type	6.1780	4.1928	4.2752	9.0949
Maximum	27.4182	79.9093	61.7456	8.7217
Minimum	-112.3699	39.4398	-11.1635	-213.5764
Valeur cible	-3.0547	19.3279	44.2322	54.9718

Le tableau ci-dessus, décrivant les statistiques des prédictions décrites par la figure précédente, donne un aperçu de l'erreur que l'on peut constater. La prédiction pour le gardien par exemple est complètement décalée de la valeur cible, avec un écart type d'à peu près 9mm.

On a aussi pu constater que le modèle a des difficultés avec une situation en particulier. Il s'agit du cas où il y a un ricochet avec les coins de la table. La figure ci-dessous démontre ce propos. Il est en effet remarquable que la méthode nous rends en sortie des prédictions extrêmement bruitées.

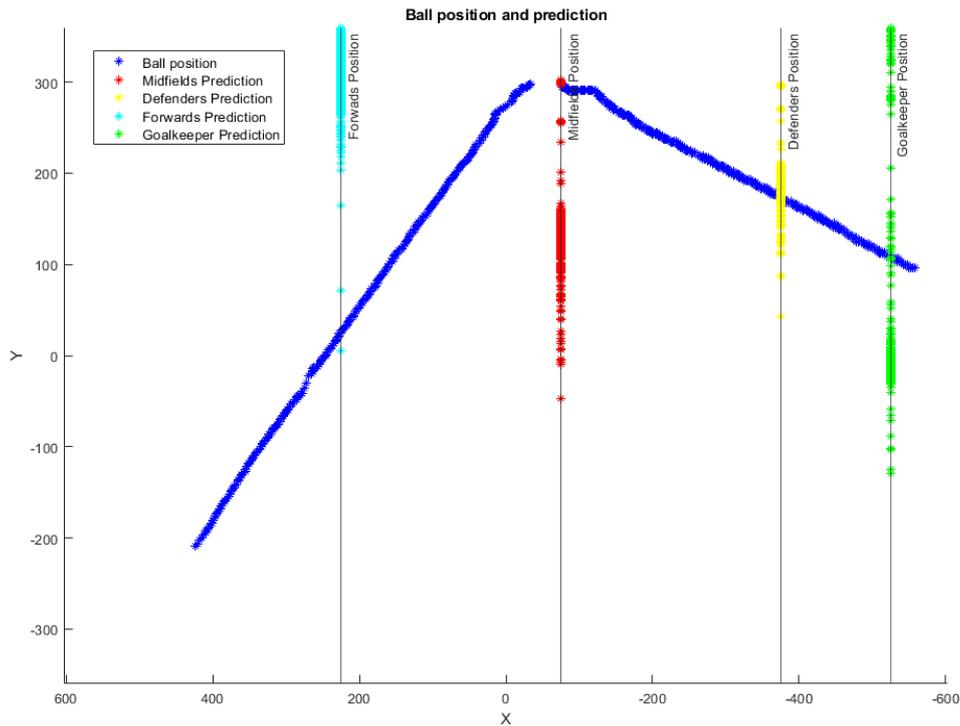


FIGURE 4 – Prédictions de la position en y de la balle - Ricochet

Dans les parties suivantes, on va proposer une nouvelle approche au problème, basée sur la même méthodologie mais reposant sur des conditions et des mises à jour plus poussées en ce qui concerne la prédiction.

3.2 Nouvelle prédiction : Régression Linéaire Conditionnée

Avant de conclure avec cette approche, plusieurs autres méthodologies ont été testées, à savoir la régression polynomiale et le filtre Kalman par exemple. Après une longue analyse des opportunités qu'offre chacune des éventuelles prédictions, il s'est avéré que l'hypothèse sur la linéarité de la trajectoire de la balle est consistante, sauf dans le cas spécial du ricochet.

En effet, au vu de la rampe des coins de la table, le cas du ricochet entraîne une déviation de la trajectoire suivant un **choc non-élastique**, menant à une trajectoire avec un angle d'après-choc différent de celui d'avant. Cette non-linéarité entraîne, comme on a pu le constater à travers la figure 4, à des prédictions erronées.

3.2.1 PredY.VI

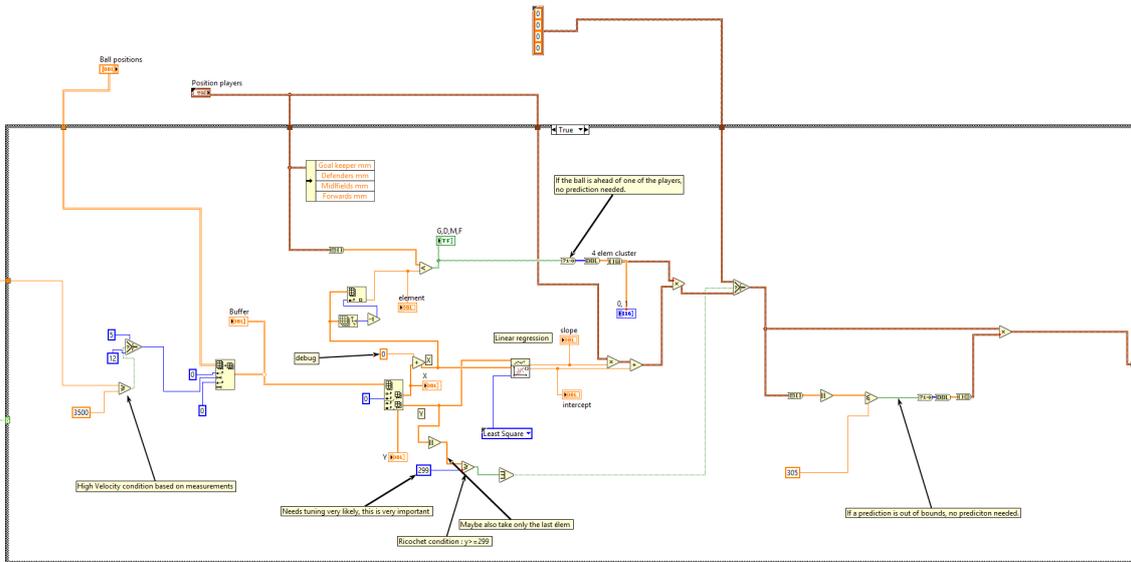


FIGURE 5 – Code LabVIEW - Nouvelle prédiction

La figure ci-dessus présente le code LabVIEW pour la prédiction conditionnée. Le processus est basé sur une régression linéaire sur les positions antérieures de la balle. Le VI pré-construit nous retourne la pente (a) et l'ordonnée à l'origine (b) de la droite de régression, calculées au sens des moindres carrés. Cette droite minimise la somme des carrés des différences entre les points donnés (y_i) et l'estimation :

$$J(a, b) = \sum_{i=1}^n (y_i - ax_i - b)^2$$

Afin d'affiner la prédiction et de mettre au point les erreurs constatées auparavant, il a été nécessaire de poser un certain nombre de conditions.

Conditions sur la vitesse de la balle

On dispose d'une estimation de la vitesse de la balle selon les deux coordonnées en chaque instant. A travers cette estimation, la norme de la vitesse est calculée et deux conditions sur cette dernière sont formulées comme suit :

- On ne calcule les prédictions que si la vitesse est supérieure à **150mm/s**. Cette mesure permet de filtrer les erreurs de mesure de la vitesse en ignorant le cas où la balle est en arrêt.
- On conditionne le nombre de points antécédents à prendre pour le calcul de la droite de régression selon la vitesse : **5** pour les grandes vitesses ($\geq 3500mm/s$), **12** sinon. Ce procédé

permet de réduire le nombre de points à prendre en compte pour les vitesses élevées vu que le système dispose de moins de temps de réaction.

Condition sur la position de la balle

Pour éviter d'avoir des valeurs aberrantes qui peuvent par la suite nuire à la stabilité du système, on calcule la prédiction pour une rangée de joueurs donnée **seulement si la balle se trouve en amont de la position de la tige en question.**

Cas particulier : Ricochet

L'éventualité du ricochet a été traitée en **remettant à zéro le buffer des points de régression lorsque la balle est proche des bords de la table :**

$$|y_{pred}| \geq 299$$

En effet, si la balle est dans cette région et qu'elle est en mouvement, le ricochet avec les bords de la table va certainement avoir lieu. La ré-initialisation des mesures permet donc d'ignorer la partie non-linéaire du comportement de la balle. Il faudrait aussi noter que cette partie de la table n'est pas desservie par les images de la caméra.

Condition sur les prédictions

La dernière condition concerne la sortie de l'algorithme. Pour éviter des valeurs qui sont en dehors des limites de la table, il a été nécessaire de **rejeter toute prédiction en dehors :**

$$|y_{pred}| \leq 305$$

3.3 Résultats et discussion

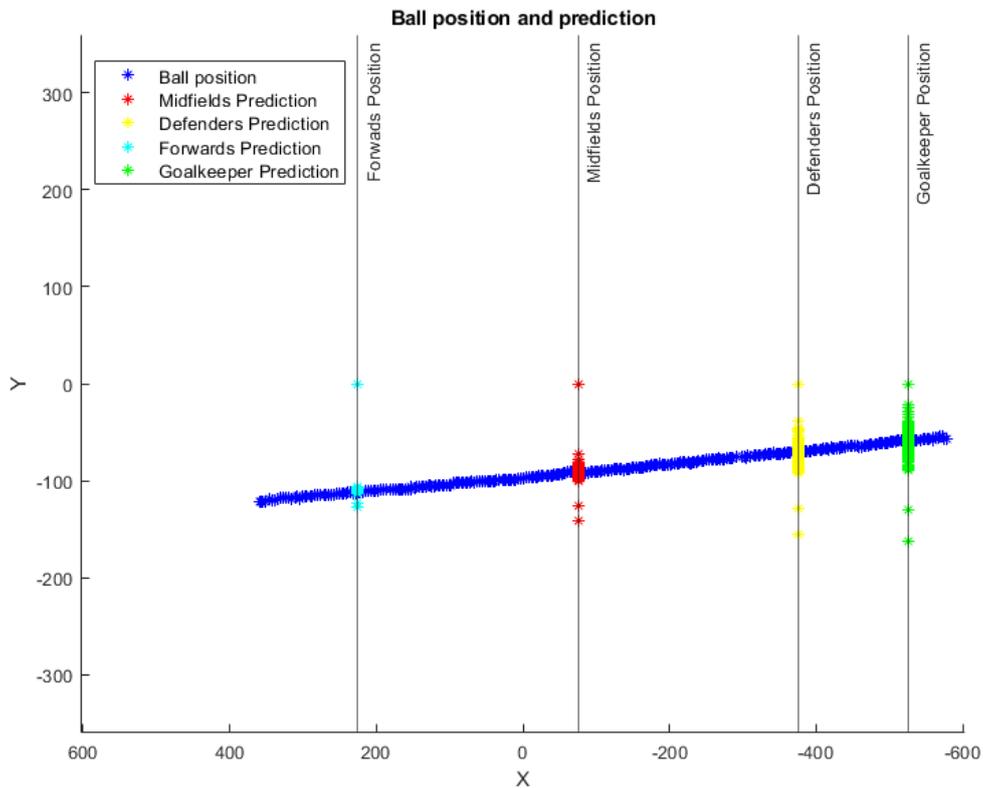


FIGURE 6 – Nouvelle prédiction - Petit angle

La figure ci-dessus présente les résultats obtenus avec la nouvelle méthodologie de prédiction de la position de la balle pour un angle de tir modéré. On peut clairement voir que les prédictions sont plus précises, centrées sur la valeur cible. On peut cependant remarquer quelques valeurs qui sortent du cadre acceptable, malgré leur nombre réduit par rapport à la version précédente. Ce constat peut être expliqué par les erreurs de mesure de la caméra, qui quelques fois retourne des positions fluctuantes de la balle malgré qu'elle est au repos, donnant lieu à une estimation incorrecte de la vitesse.

Quant la prédiction n'est pas nécessaire, on impose que cette dernière soit égale à zéro, d'où les points distincts que l'on peut apercevoir en haut du graphe.

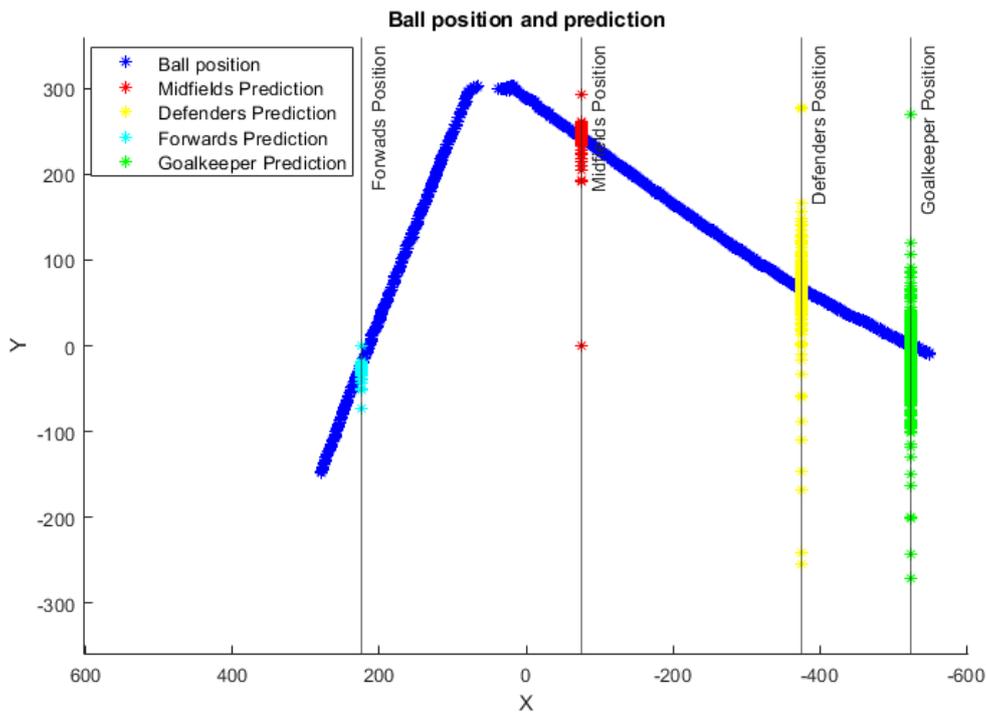


FIGURE 7 – Nouvelle prédiction - Ricochet

Concernant le ricochet sur la figure 7, on peut bel et bien voir une prédiction centrée sur la valeur cible. Il y a néanmoins beaucoup de valeurs qui sortent du cadre acceptable. Cette expérience a été menée avec une vitesse faible afin de démontrer le résultat, aspect qui peut ajouter du bruit au vu du grand nombre de points et de prédictions à faire ou encore créer un conflit avec la condition sur la vitesse de la balle.

4 Stratégie de défense : Suivi de la prédiction

Le modèle de prédiction a servi à élaborer une nouvelle stratégie de défense, qui consiste en le suivi en chaque instant de la prédiction calculée, au lieu de la méthode précédente où les joueurs suivaient tout simplement la position instantanée de la balle.

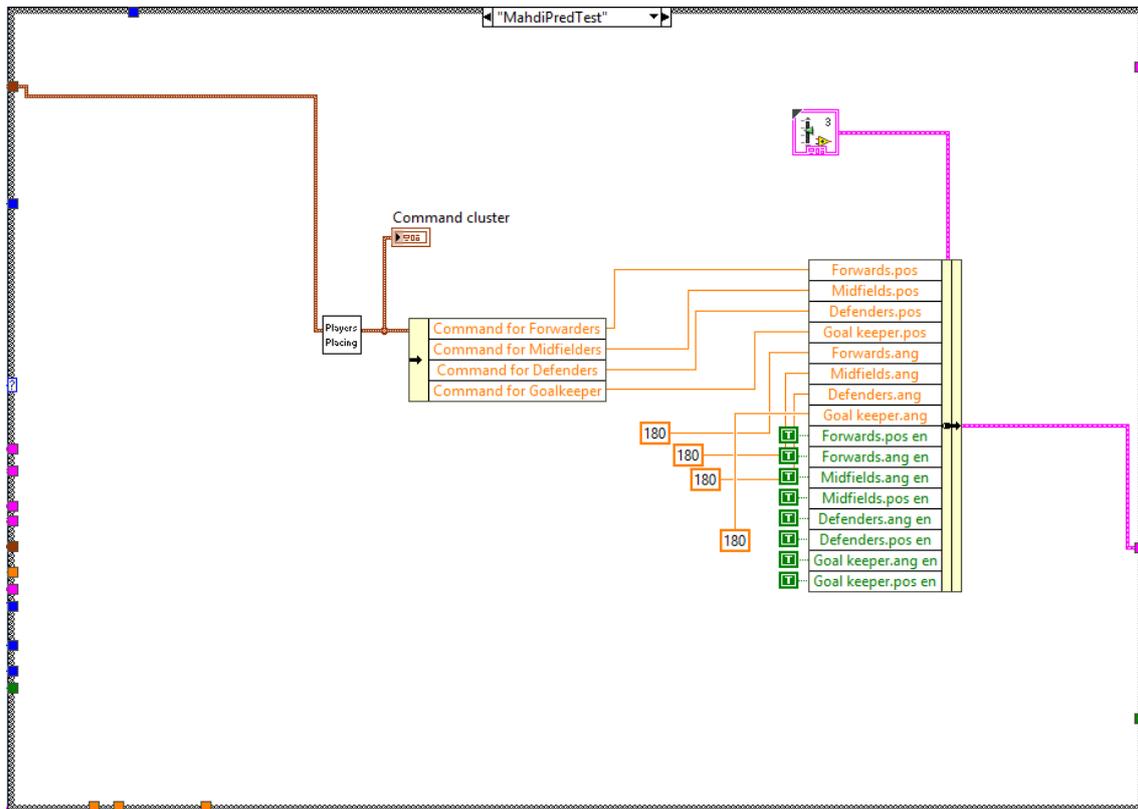


FIGURE 8 – VI de la nouvelle stratégie de défense

On positionne les joueurs horizontalement (pas de rotation, 180). En ce qui concerne la translation, elle est dictée par la prédiction qui a été calculée auparavant. Le VI **Players Placing** est seulement là pour choix du joueur qui devrait intercepter la balle pour chacune des barres.

En pratique, on a pu constater à travers plusieurs tests selon différentes configurations que la prédiction de la trajectoire de la balle est dotée d'une très haute précision, aspect qui apporte une solution efficace à la stratégie de défense du système.

Le système a néanmoins du mal à intercepter la balle lorsque la vitesse est très élevée.

4.1 Non-interception à vitesse élevée

Le test qui a été réalisé consiste à tirer la balle à très grande vitesse pour tester la réactivité du mécanisme. On a remarqué dans un grand nombre de cas que ce n'était pas possible d'intercepter la balle avec des vitesses aussi grandes. Il a été donc nécessaire d'investiguer la raison derrière ce constat.

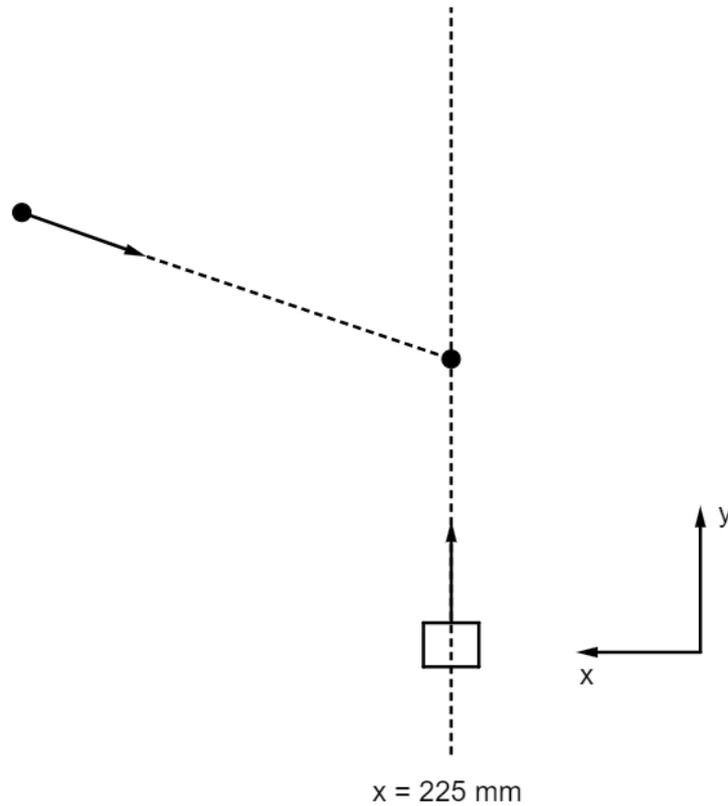


FIGURE 9 – Modèle physique

La figure ci-dessus représente le modèle du tir réalisé : Le point représente la balle et le rectangle représente l'attaquant qui est censé intercepter cette dernière.

La position initiale de la balle dans le repère est donnée par : $x_{0b} = 476.46, y_{0b} = 57.28$. Sa position finale est la suivante : $x_{fb} = 476.46, y_{fb} = 57.28$.

La position initiale du joueur est donnée par : $x_{0j} = 225, y_{0j} = 0$

Une mesure du temps de trajet de la balle nous indique qu'elle mets $t = 0.0397s$ pour arriver au point de rencontre présumé, incluant tous les retards liés au système.

Équations de mouvement

Une estimation de l'accélération moyenne de la balle selon y a été calculée en se basant sur les vitesses instantanées dont on dispose :

$$a_{by} = -1.5211 \times 10^4 [mm/s^2]$$

Quant au joueur, on a pris comme accélération la valeur extrême, à savoir l'accélération maximale

des moteurs linéaires :

$$a_{jy} = 222[m/s^2]$$

On peut donc écrire les équations suivantes pour les deux objets :

$$y_b = \frac{1}{2}a_{by}t^2 + v_{0b,y}t + y_{0b}$$

$$y_j = \frac{1}{2}a_{jy}t^2$$

Avec : $v_{0b,y} = -517[mm/s]$.

Critère de rencontre

Le critère de rencontre $y_j = y_p$ nous donne le temps de rencontre estimé suivant :

$$t_r = 0.072s,$$

qui est supérieur à la durée du trajet mesuré de la balle.

Mesure du temps de réponse du système

On dispose d'un VI de mesure qui permet d'estimer le temps de réponse du système suite à une sollicitation.

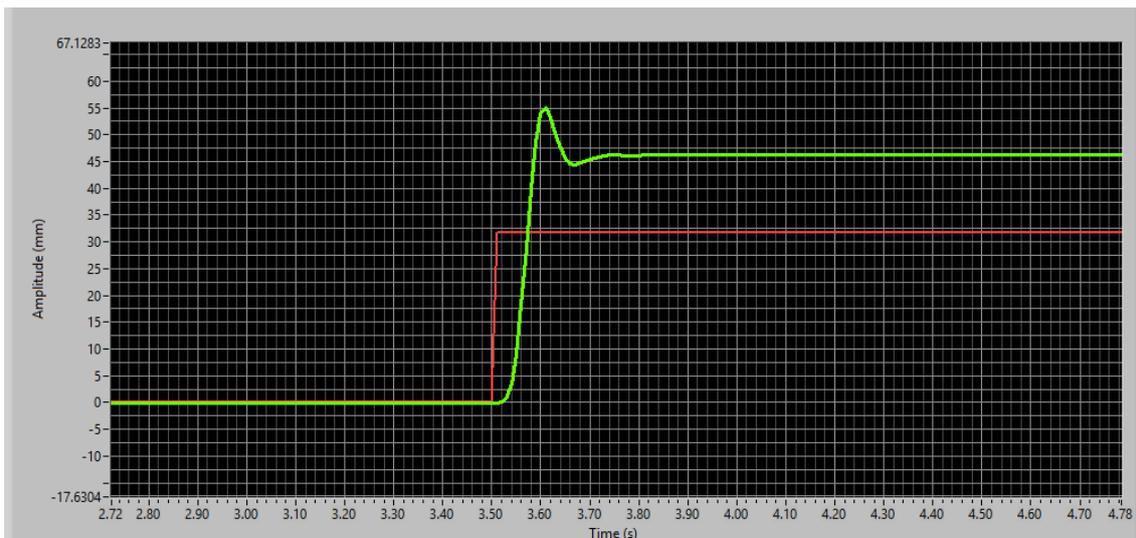


FIGURE 10 – Réponse du système suite à une sollicitation de 5 Volts

La figure ci-dessus montre la réponse du système suite à une sollicitation de 5 Volts, entraînant un déplacement en translation de la barre d'attaquants de $y = 46.36mm$.

Cette figure permet de constater que le temps de transition vers l'état stationnaire, ou le temps de réponse du système est approximativement :

$$t_s = 3.8 - 3.5 = 0.3s$$

Résultat

En se basant sur les résultats décrits plus tôt, on voit clairement que c'est **la dynamique des moteurs** qui empêche un déplacement aussi rapide. Le résultat précédent confirme cette déduction : Le temps de réponse n'est pas suffisant pour intercepter la balle à cette vitesse.

5 Conclusions et pistes d'amélioration

Ce projet a été une opportunité pour se familiariser avec l'un des systèmes les plus complexes dont dispose le Laboratoire d'Automatique. Cette étude nous a permis d'examiner l'état d'art du système de plus près, tout en proposant des solutions efficaces qui ne pourront qu'améliorer sa performance dans le futur.

Une méthode de prédiction de la position de la balle a été élaborée, qui pourra permettre par la suite une finition des stratégies de jeu du système. La stratégie de capture a par exemple été testée avec les nouvelles prédictions et ça a l'air de marcher beaucoup mieux qu'avant, même si cette piste n'a pas été étudiée suffisamment par manque de temps.

D'après les résultats de cette étude, on peut partir vers ces horizons pour les projets suivants :

- **Peaufiner la stratégie de jeu** : La stratégie de capture, par exemple, est assez robuste pour les attaquants. Il faudrait cependant la généraliser aux autres joueurs afin d'avoir un système complet. On pourrait aussi penser à élaborer un mécanisme de passes entre les joueurs, tout en combinant les deux approches à une stratégie d'attaque dynamique, à savoir ShootDynamic qui reste la plus efficace à ce jour.
- **Améliorer la précision des mesures de la caméra** : Les positions de la balle et l'estimation de la vitesse qui en découle restent assez bruitées. Il faudrait penser à une approche pour estimer la variance des mesures et d'établir un modèle qui se base sur ceci pour les mesures. On pourrait par exemple enregistrer des mesures de la balle en étant statique dans plusieurs position différentes pour mesurer la variance de la mesure.
- **Identifier le système** : Une identification du système permettrait de mettre en place des modèles de contrôle plus complexes. Cette approche est cependant très difficile à réaliser.
- **Reinforcement learning** : Lorsque le mécanisme disposera d'un grand nombre de stratégies, on pourrait penser à créer une simulation où on entraîne le système à choisir la réponse la plus judicieuse à une offensive du joueur humain.

Pour conclure, j'ai apprécié travailler sur ce projet et tiens à remercier encore une fois le professeur Salzman pour son soutien tout au long du semestre.

Références

- [1] Yann Morize. Vision and strategy : Ball capture. 2020.
- [2] Sameer Roge. Babyfoot fine control. 2022.