



HAL
open science

Réseau neuronal informé par la physique pour l'analyse cinématique 3D du mouvement d'un projectile. Application au Tennis de Table

Zaineb Chiha, Renaud Péteri, Laurent Mascarilla

► To cite this version:

Zaineb Chiha, Renaud Péteri, Laurent Mascarilla. Réseau neuronal informé par la physique pour l'analyse cinématique 3D du mouvement d'un projectile. Application au Tennis de Table. Congrès Reconnaissance des Formes, Image, Apprentissage et Perception (RFIAP 2024), Jul 2024, LILLE, France. hal-04639640

HAL Id: hal-04639640

<https://hal.science/hal-04639640>

Submitted on 9 Jul 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Réseau neuronal informé par la physique pour l'analyse cinématique 3D du mouvement d'un projectile Application au Tennis de Table

Zaineb Chiha

Renaud Péteri

Laurent Mascarilla

La Rochelle Université, MIA, La Rochelle, France
{zaineb.chiha, rpeteri, lmascari}@univ-lr.fr

Résumé

L'analyse des gestes sportifs peut bénéficier des méthodes basées sur la vision pour extraire des informations précieuses aux joueurs et entraîneurs, sans qu'il ne soit nécessaire d'utiliser des capteurs portés sur le corps. Au tennis de table, la prédiction précise des trajectoires de la balle et de leur cinématique 3D est essentielle pour évaluer l'efficacité d'un coup. Bien que plusieurs méthodes de vision par ordinateur aient été étudiées dans ce domaine, elles se concentrent souvent sur la prédiction des trajectoires uniquement, sans tenir compte des paramètres physiques. En outre, ces études sont généralement limitées à des environnements contrôlés en laboratoire, ce qui peut être invasif et nécessiter une collecte de données extensive. Dans cet article, nous présentons une nouvelle approche basée sur la vision pour l'analyse de la trajectoire des balles 3D. La méthode que nous proposons s'appuie sur l'apprentissage informé par la physique (PINNs) pour prédire la trajectoire 3D d'une balle de tennis de table. Elle permet aussi d'extraire les paramètres cinématiques de la trajectoire 3D à l'aide d'une petite quantité de données acquises à partir de séquences vidéo. À notre connaissance, il s'agit de la première tentative d'intégration d'approches basées sur des données et des contraintes physiques afin de prédire la trajectoire 3D et les paramètres cinématiques d'une balle de tennis de table.

Mots Clef

Vision par ordinateur, PINNs, cinématique 3D, mouvement de projectile, tennis de table.

Abstract

Vision-based methods for gesture analysis in sports can provide valuable insights to players and coaches, without any disturbing body-worn sensors. In table tennis, accurately predicting the trajectories of the ball and its 3D kinematics is crucial for assessing the effectiveness of a stroke. While several computer vision methods have investigated this area, they often focus on trajectory prediction solely, without considering physical parameters. Additionally, these studies are typically limited to controlled laboratory settings, which can be invasive and require extensive

data collection.

In this paper, we present a novel vision-based approach to 3D ball trajectory analysis. Our proposed method relies on Physics-Informed Learning to predict the 3D trajectory of a table tennis ball and to infer its 3D kinematics parameters using a small amount of data acquired from video sequences. To the best of our knowledge, this is the first attempt to integrate data-driven approaches with physical constraints in order to predict the 3D trajectory and kinematics parameters of a table tennis ball.

Keywords

Computer Vision, PINNs, 3D kinematics, Projectile Motion, Table Tennis.

1 Introduction

La démocratisation actuelle des technologies permettant la collecte de données individuelles relatives à l'activité physique (applications smartphone, montres et vêtements connectés, etc) est une source potentielle d'informations pour la recherche et l'innovation dans le domaine de la santé et du sport. Les méthodes basées sur la vision pour l'analyse des gestes sportifs peuvent fournir des informations précieuses aux joueurs et aux entraîneurs, sans perturbation du mouvement par des capteurs portés sur le corps [5]. Ces méthodes peuvent utiliser de la vision stéréoscopique ou monoculaire pour analyser les données 3D extraites des scènes. Afin de guider les méthodes de vision par ordinateur, certains travaux sur l'analyse du mouvement humain injectent des contraintes issues de la physique [6, 23]. Dans cet article, nous proposons une nouvelle approche utilisant la vision par ordinateur et les réseaux de neurones informés par la physique (Physics-Informed Neural Network - PINN) pour prédire la trajectoire 3D d'une balle de tennis de table mais aussi déduire ses paramètres cinématiques 3D non observables. Nous démontrons l'efficacité de notre méthode en analysant des trajectoires 3D simulées. Nous validons également l'approche sur un ensemble de données réelles, qui comprend plusieurs séquences stéréoscopiques de joueurs de tennis de table exécutant différents types de coups.

2 État de l’art

La plupart des travaux actuels sur l’analyse des actions sportives basée sur la vision utilisent des méthodes d’apprentissage profond et nécessitent souvent une annotation importante des données. L’utilisation de l’apprentissage informé par la physique [10] permet d’intégrer des connaissances préalables issues de l’observation et des équations de la physique afin d’améliorer les performances d’un algorithme d’apprentissage. Alors que l’apprentissage profond et les modèles physiques sont généralement étudiés séparément [24, 7], peu de recherches explorent leur utilisation combinée. Dans notre contexte d’estimation des trajectoires de balle et des paramètres cinématiques associés, on peut mentionner les travaux de McNally *et al.* [13]. Les auteurs proposent un modèle de trajectoire de balle de golf associant l’apprentissage profond aux équations de mouvement. Ainsi, leur méthode permet de générer des trajectoires de balle de golf simulées réalistes et de modéliser la relation entre les coefficients aérodynamiques et l’état de la balle. Proche de notre sujet de recherche, Calandre *et al.* [2] développent une méthode de vision par ordinateur monoculaire pour modéliser les trajectoires d’une balle de tennis de table. Ils utilisent un réseau neuronal convolutif (CNN) pour déduire la distance à la caméra des positions suivies de la balle dans l’image. En tenant compte de l’effet Magnus [20] dans le modèle physique, ils peuvent extraire la vitesse de rotation de la balle, supposée constante, le long de sa trajectoire.

Une amélioration de l’utilisation conjointe des réseaux neuronaux avec les équations différentielles s’est produite avec les réseaux neuronaux informés par la physique (PINN) introduits par Raissi *et al.* [17]. Les auteurs définissent deux classes de PINN : les solutions guidées par les données qui tentent de résoudre des équations différentielles basées sur des connaissances physiques préalables, et la découverte guidée par les données d’équations différentielles de forme connue mais dont les paramètres sont inconnus [18]. Ces dernières ont l’avantage de s’appuyer sur une quantité limitée de données observées [16]. Diverses applications pour les problèmes directs et inverses sont examinées dans [10]. A titre d’exemple en aérodynamique, la détermination des coefficients de force pour la modélisation dynamique du vol nécessite généralement des installations expérimentales coûteuses, telles que les souffleries [8]. Dans le domaine de l’aéronautique, Michek *et al.* [14] utilisent des PINNs pour extraire des coefficients de frottement variables à partir de données de vol simulées. Dans cet article, nous proposons une approche similaire pour l’analyse sportive, spécifiquement appliquée au tennis de table. L’étude de l’aérodynamique de petits objets en mouvement rapide, tels que les balles, représente un défi important en termes de modélisation 3D. Pour les entraîneurs et les joueurs, l’analyse des coups peut être améliorée en étudiant les effets donnés sur la balle. Les types de coups sont caractérisés par leur vitesse de rotation ou de translation, et ces paramètres physiques peuvent donc être

utilisés pour quantifier leur efficacité.

À cette fin, nous introduisons une approche d’apprentissage basée sur la physique pour modéliser le vol de la balle. Nous simulons les trajectoires du projectile à partir d’un système paramétré d’Equations Différentielles Ordinaires (EDO). En utilisant ces EDO et un petit nombre de points de données observés, nous développons un modèle de réseau neuronal informé par la physique (PINN). Ce modèle est capable, à partir de séquences vidéo de tennis de table, de prédire des trajectoires 3D de balle et de déduire des paramètres non observables tels que le coefficient de frottement, les vitesses de translation et de rotation. La figure 1 présente une image d’une séquence vidéo avec la trajectoire de la balle représentée. Elle montre également les deux caméras stéréos permettant l’extraction de la trajectoire de la balle en 3D, qui sera utilisée ultérieurement pour valider notre approche (section 4.3).

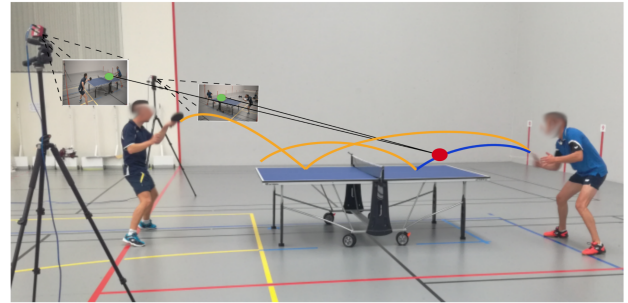


FIGURE 1 – Image d’une séquence vidéo de tennis de table. Les points verts représentent la balle suivie à partir des images de chaque caméra calibrée. La reconstruction de la position de la balle en 3D (point rouge) est réalisée par triangulation. La trajectoire de la balle est représentée en orange.

3 Extraction et analyse des trajectoires 3D

Cette section présente la modélisation physique du mouvement d’un projectile, qui implique la force gravitationnelle, la force de frottement et la force Magnus liée à la rotation du projectile. Nous présentons ensuite l’utilisation conjointe de contraintes guidées par la physique et des réseaux neuronaux profonds dans notre modèle d’apprentissage (PINN).

3.1 Modélisation du mouvement des projectiles

L’application de la deuxième loi de Newton pour une balle de tennis de table en vol conduit à l’équation suivante :

$$m\mathbf{a} = m\frac{d\mathbf{V}}{dt} = \mathbf{F}_G + \mathbf{F}_A \quad (1)$$

Ici, \mathbf{a} représente l’accélération de la balle, \mathbf{V} est son vecteur vitesse, et m est la masse de la balle. La force gravita-

tionnelle \mathbf{F}_G seule conduit à une trajectoire parabolique de la balle, influencée par sa masse m et la constante de gravité g . Les forces aérodynamiques $\mathbf{F}_A = \mathbf{F}_D + \mathbf{F}_L$ dues aux frottements de l'air et à l'effet Magnus modifient cette simple parabole, ce qui impacte la trajectoire de la balle.

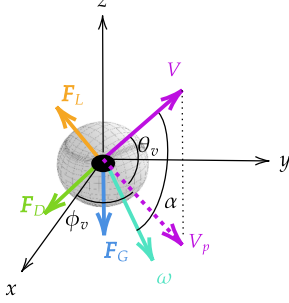


FIGURE 2 – Schéma montrant les forces appliquées sur une balle. La force de frottement, \mathbf{F}_D , est dans la direction opposée de \mathbf{V} tandis que l'effet Magnus \mathbf{F}_L est dans la direction de $\omega \wedge \mathbf{V}$. L'angle θ_v est l'angle entre le vecteur vitesse et le plan horizontal (plan x - y). L'angle ϕ_v est l'angle entre la projection du vecteur vitesse sur le plan x - y et l'axe x . ω et \mathbf{V} forment entre eux un angle noté α .

La force de frottement aérodynamique F_D agit comme une force opposée au mouvement de la balle (Fig. 2). Son expression mathématique est la suivante :

$$F_D = \|\mathbf{F}_D\| = -\frac{1}{2}C_D\rho AV^2$$

où ρ est la densité de l'air, $A = \pi R^2$ est la section transversale d'une balle de rayon R , $V = \|\mathbf{V}\|$ est la norme de \mathbf{V} , et C_D est le coefficient de frottement de l'air, dépendant de V . Une balle de tennis de table peut se modéliser comme une sphère lisse et nous adoptons l'hypothèse faite par Robinson *et al.* [19] où le coefficient de frottement C_D est supposé constant le long de la trajectoire, avec des valeurs de l'ordre de 0,45. La force \mathbf{F}_D est proportionnelle à V^2 et dans la direction opposée à \mathbf{V} .

L'effet Magnus \mathbf{F}_L est la force qui explique la déviation que subit un projectile en rotation se déplaçant dans l'air par rapport à la trajectoire parabolique. Pour certaines rotations, l'effet Magnus peut engendrer des trajectoire non planaires. Elle agit dans la direction de $\omega \wedge \mathbf{V}$, est proportionnelle à V^2 , et sa norme vaut :

$$F_L = \|\mathbf{F}_L\| = \frac{1}{2}C_L\rho AV^2$$

avec C_L le coefficient de portance (*lift*).

Stepanek [21] exprime les coefficients C_D et C_L en fonction de $\frac{\omega}{V}$. Étant donné le rayon R , la vitesse de rotation ω

et V , le coefficient de lift C_L est le suivant :

$$C_L = \frac{1}{2 + (R\frac{\omega}{V})^{-1}}$$

Dans ce travail, ω sera supposé être constant pour tous t [19], *i.e.* avec une vitesse et une direction de rotation constantes tout au long de la trajectoire.

3.2 PINNs pour la prédiction du mouvement des projectiles et l'estimation des paramètres cinématiques

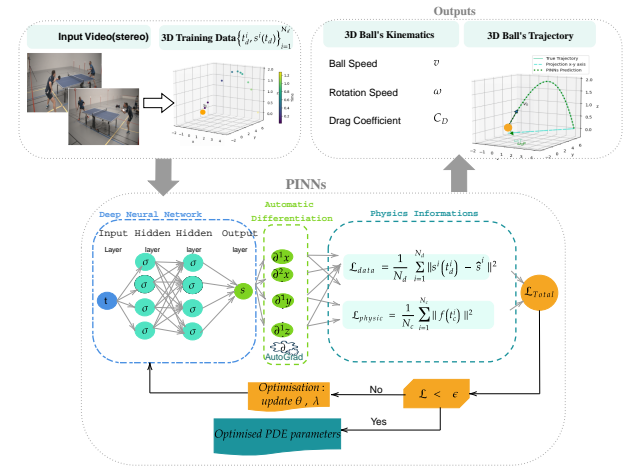


FIGURE 3 – Schéma général de la méthode proposée. Le bloc du réseau neuronal informé par la physique (PINN) consiste en un réseau neuronal profond, avec le temps comme entrée. Il prédit les coordonnées 3D de la trajectoire de la balle $\hat{s}(t) = (x(t), y(t), z(t))$. Les dérivées de $\hat{s}(t)$ par rapport aux points de collocation $\{t_c^i\}$ sont calculées par différenciation automatique (AD) afin d'obtenir les vitesses, les accélérations et les résidus des équations du mouvement. Les coefficients $\theta = \{poids, biais\}$ du réseau neuronal et les paramètres inconnus de l'EDO $\lambda = \{C_D, \omega_x, \omega_y, \omega_z\}$ peuvent être appris simultanément en minimisant la fonction d'erreur.

La physique sous-jacente du mouvement du projectile et les paramètres à estimer étant définis, nous détaillons par la suite la structure du PINN. Les PINN [3] sont composés de trois éléments principaux : un réseau neuronal profond, une fonction d'erreur basée sur des contraintes physiques et un mécanisme d'optimisation (Fig. 3).

Réseau de neurones. Le PINN comprend un perceptron multicouche (MLP) composé de plusieurs couches cachées, chacune utilisant une fonction d'activation Tanh. En tant que réseau de type "feed-forward", il ne comprend aucune boucle de rétroaction. Les données d'apprentissage sont les positions 3D d'une balle à l'instant t , et sont notées $\{t_d^i, s^i(t_d^i)\}_{i=1}^{N_d}$, avec N_d le nombre de données

d'apprentissage. Le réseau vise à prédire les positions 3D $\hat{s}(t) = (x(t), y(t), z(t))$ pour tout instant t .

Contraintes physiques. Un ensemble de points d'apprentissage $\{t_c^i\}$, appelés *points de collocation* [18], vont traduire les contraintes du modèle physique. Ces points de collocation sont obtenus par échantillonnage du domaine temporel d'entrée du réseau.

Finalement, la projection de l'équation physique contraignant la convergence du PINN (Eq.1) donne les fonctions résiduelles suivantes :

$$\begin{aligned} f_x &= -a_x - k \times V \left\{ C_D v_x - \left(\frac{C_L}{\omega} (\omega_y v_z - \omega_z v_y) \right) \right\} \\ f_y &= -a_y - k \times V \left\{ C_D v_y - \left(\frac{C_L}{\omega} (\omega_z v_x - \omega_x v_z) \right) \right\} \\ f_z &= -a_z - g - k \times V \left\{ C_D v_z - \left(\frac{C_L}{\omega} (\omega_x v_y - \omega_y v_x) \right) \right\} \end{aligned} \quad (2)$$

avec $k = \frac{\rho A}{2m}$.

Mécanisme d'optimisation. La phase d'apprentissage du PINN va consister à minimiser les fonctions résiduelles sur le domaine d'intérêt, tout en minimisant simultanément l'erreur par rapport aux points de données et aux conditions initiales. Le mécanisme d'optimisation vise ainsi à minimiser l'erreur totale \mathcal{L} :

$$\mathcal{L} = \beta_{phy} \mathcal{L}_f + (1 - \beta_{phy}) \mathcal{L}_s \quad (3)$$

où \mathcal{L}_s représente l'erreur relative aux données, \mathcal{L}_f représente l'erreur relative à l'EDO physique. Le paramètre $\beta_{phy} \in [0, 1]$ représente le poids relatif de la fonction d'erreur associée aux contraintes physiques. La détermination des poids optimaux pour les fonctions d'erreur dans le PINN peut améliorer les performances ainsi que les temps de convergence [4]. Ces fonctions d'erreur représentent les erreurs quadratiques moyennes suivantes [17] :

$$\begin{aligned} \mathcal{L}_s &= MSE_s = \frac{1}{N_d} \sum_{i=1}^{N_d} \|s(t_d^i) - \hat{s}^i\|^2 \\ \mathcal{L}_f &= MSE_f = \frac{1}{N_c} \sum_{i=1}^{N_c} \|f(t_c^i)\|^2 \end{aligned} \quad (4)$$

où $\{t_s^i, s^i\}_{i=1}^{N_d}$, désignent les données d'apprentissage et $\{t_c^i\}_{i=1}^{N_c}$ sont les points de collocation. L'erreur sur les données \mathcal{L}_s encode la différence entre la vérité terrain et les positions 3D prédites par le PINN. L'erreur physique \mathcal{L}_f encode l'inadéquation entre la solution prédite par le PINN et les équations différentielles qui régissent la trajectoire (2). C'est une contrainte de régularisation qui garantit que la solution apprise par le réseau est cohérente avec la physique.

L'optimisation est faite par descente de gradient stochastique Adam [11]. Les paramètres du modèle $\theta = \{\text{poids}, \text{biais}\}$, ainsi que les paramètres inconnus de l'EDO $\lambda = \{C_D, \omega_x, \omega_y, \omega_z\}$, sont appris en minimisant l'erreur totale :

$$(\theta^*, \lambda^*) = \arg \min_{\theta, \lambda} (\beta_{phy} \mathcal{L}_f + (1 - \beta_{phy}) \mathcal{L}_s) \quad (5)$$

Enfin, nous utiliserons comme métrique pour évaluer la performance du modèle l'erreur relative \mathcal{E} :

$$\mathcal{E}(\hat{\mathbf{X}}, \mathbf{X}) = \frac{\|\hat{\mathbf{X}} - \mathbf{X}\|}{\|\mathbf{X}\|} \quad (6)$$

$\hat{\mathbf{X}}$ étant la position 3D de la balle ou la valeur des paramètres cinématiques estimés, et \mathbf{X} leur vérité terrain.

4 Expérimentations

Cette section présente la méthode utilisée pour simuler les trajectoires 3D des balles puis évaluer notre modèle PINN. Des tests sont également effectués sur un jeu de données issues de vidéo réelles de joueurs de tennis de table effectuant différents types de coups.

4.1 Simulation des trajectoires des balles

Les trajectoires 3D des balles de tennis de table sont simulées grâce à l'équation 2 en faisant varier les conditions initiales. Étant donné un vecteur initial \mathbf{V}_0 , correspondant à la vitesse de la balle juste après un coup, on détermine $\omega_{0\perp}$, le vecteur de rotation orthogonal au plan vertical contenant \mathbf{V}_0 (Fig. 4.(a)).

Afin de simuler des coups avec des vecteurs de rotation non purement horizontaux, de nouveaux vecteurs de rotation ω_0 sont générés à partir de $\omega_{0\perp}$ en modifiant son orientation. Plus précisément, si l'on considère le cône porté par $\omega_{0\perp}$ d'angle ψ , les variations d'élévation et d'azimut de $\omega_{0\perp}$ sont générées aléatoirement dans l'intervalle $[-\frac{\psi}{2}, \frac{\psi}{2}]$. Ceci est illustré sur la figure 4.

Une fois les conditions initiales définies, les équations du mouvement sont résolues à l'aide de la méthode de Runge-Kutta à l'ordre 4 [15]. Trois types de coups sont simulés : *Topspin*, *Contre-Attaque*, et *Poussette*. Ces coups ont des trajectoires typiques qui diffèrent les uns des autres par leurs vitesses de translation et de rotation. Le *Topspin* est un coup offensif qui permet à la balle de se déplacer rapidement avec une grande vitesse de rotation vers l'avant. Le *Poussette* est un coup défensif, qui ralentit la vitesse de la balle après impact sur la raquette et la fait tourner vers l'arrière. Le *Contre-Attaque* est intermédiaire entre ces deux coups, et donne à la balle une vitesse de translation et de rotation moyenne. Les valeurs initiales des vitesses de rotation et de translation utilisées pour ces simulations sont déterminées par les plages décrites dans le tableau 1 (Calandre *et al.*[2]).

Ces paramètres initiaux permettent de guider le choix de l'angle α entre \mathbf{V}_0 et ω_0 . Comme la force Magnus appliquée à la balle dépend de l'angle α (Section 3.1), on peut

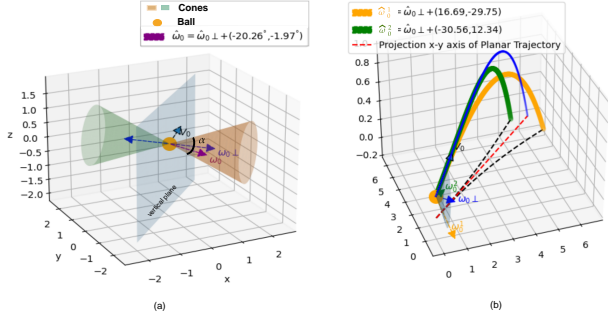


FIGURE 4 – (a.) Conditions initiales pour les vecteurs \mathbf{V}_0 et ω_0 . Le vecteur $\omega_{0\perp}$ sert de référence pour générer ω_0 par variations $(\Delta\varphi_0, \Delta\theta_0)$ des angles sphériques (φ_0, θ_0) dans des cônes d’angles fixés. (b.) Exemple de trois trajectoires simulées de type Topspin, avec $V_0=16$ m/s, et $\omega_0=37$ rps : une trajectoire planaire (en bleu) avec $\omega_0 \perp \mathbf{V}_0$ et 2 trajectoires non planaires avec ω_0 variant dans la plage définie par l’angle du cône $\in[-30^\circ, 30^\circ]$.

TABLE 1 – Plages de valeurs des conditions initiales pour chaque classe de coups.

Coup	Translation (m/s)		Rotation (rotations/s)	
	min	max	min	max
Topspin	10	16	30	70
Contre-Attaque	4	10	10	20
Poussette	1	5	-15	0

générer des trajectoires non planaires (Fig. 4.b). La possibilité de modéliser et d’extraire des trajectoires non-planaires est une nouveauté de cet article et est importante pour l’applicabilité de la méthode proposée à des scénarios réels.

4.2 Entraînement et évaluation du PINN proposé

Pour chacun des trois types de coups, nous simulons 100 trajectoires et comparons les performances du réseau proposé pour chaque classe de coups. Comme le nombre de données N_{data} utilisé est faible, nous utilisons un MLP de taille réduite avec 3 couches de 20 neurones [17]. Nous sélectionnons ces données en choisissant par tirage aléatoire uniforme des points sur la trajectoire simulée (voir figure 5(a)). Ces données d’entraînement sont utilisées pour calculer l’erreur sur les données \mathcal{L}_s . Pour calculer l’erreur physique \mathcal{L}_f , nous utilisons N_c points de collocation uniformément espacés temporellement sur la durée de la trajectoire. L’erreur totale \mathcal{L} est la somme des deux erreurs précédentes. Celles-ci ayant des intervalles de variation très différents, ce qui ralentit la convergence du PINN, nous les avons pondérées (voir équation 3) par un coefficient $\beta_{phy} \in [0, 1]$. Nos expérimentations nous ont conduits à choisir la valeur $\beta_{phy}=10^{-3}$ qui fournit un bon compromis entre la vitesse de convergence et l’apprentissage précis des paramètres de l’EDO. Après 250k itération d’entraînement, le modèle apprend les paramètres inconnus et

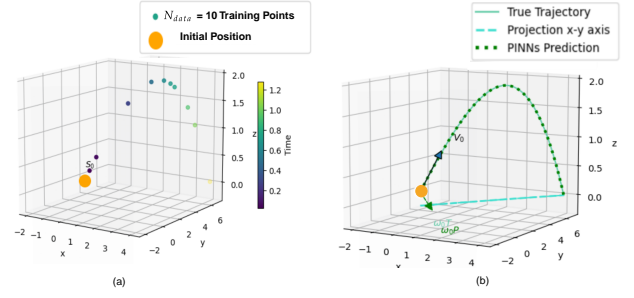


FIGURE 5 – (a.) Exemple de données d’entraînement Topspin. Les données sont choisies au hasard le long de la trajectoire. (b.) Résultat de la prédiction de la trajectoire de la balle par PINN.

prédit la position 3D de la balle avec précision. Le temps de calcul moyen est d’environ 15 minutes sur un ordinateur portable. Cependant, il convient de noter que le PINN est initialisé et réentraîné pour chaque trajectoire, mais l’architecture du PINN considéré ne dépend pas du type de coup. L’évolution des différentes erreurs durant l’apprentissage d’un exemple de trajectoire sont représentées sur la figure 6. L’erreur totale diminue au fil des itérations d’entraînement, ce qui indique que le modèle apprend et améliore ses performances, en trouvant un équilibre entre l’adaptation aux données observées et la contrainte de l’équation physique sous-jacente.

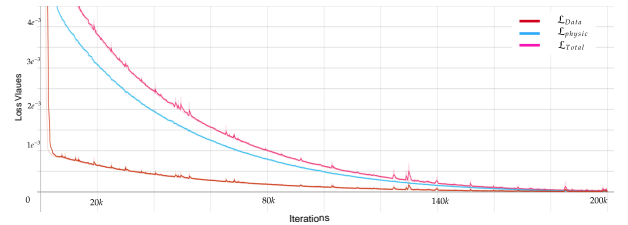


FIGURE 6 – Exemple d’évolution des erreurs durant l’entraînement sur une trajectoire Topspin avec $\beta_{phy}=10^{-3}$.

La figure 5(b) illustre, sur une trajectoire Topspin, le résultat obtenu après convergence du PINN. Les positions 3D prédites et le vecteur ω_0 sont superposés à la trajectoire simulée. Il est clair que le modèle prédit avec précision la trajectoire 3D de la balle et sa cinématique. Pour quantifier la qualité des résultats, nous calculons l’erreur relative entre les valeurs prédites et les trajectoires simulées (équation 6) : la figure 7 montre la distribution de cette erreur à l’aide de diagrammes en violon pour les différentes catégories de coups. Nous observons que pour les trois types de coups, la plupart des erreurs relatives se situent entre $[0, 0.001]$, avec une valeur maximale de 0.0017. Ces erreurs sont négligeables, comme le montre la très bonne correspondance entre les trajectoires prédites et estimées dans la figure 5. Outre l’estimation des trajectoires 3D, l’objectif est éga-

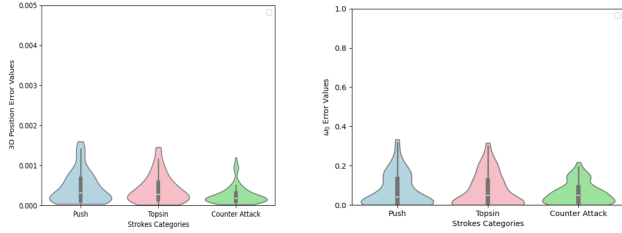


FIGURE 7 – (a) Diagrammes en violon des erreurs sur les estimations des positions 3D par le PINN. (b) Diagrammes en violon des erreurs sur les estimations de ω_0 par le PINN.

lement d’extraire des paramètres cinématiques non observables tels que la vitesse de rotation ω_0 et le coefficient de traînée C_d . Sans recours à un modèle physique, tel que celui utilisé, l’inférence de ces paramètres cinématiques à partir des trajectoires prédites est une tâche très difficile. Notre PINN fournit une estimation de la vitesse de rotation, avec une distribution d’erreur allant de 0 à 0.21 pour la classe Contre-Attaque et jusqu’à 0.36 pour Poussette et Topspin. Cette variation dans la distribution des erreurs peut être attribuée aux différences dans les caractéristiques de rotation et de translation de chaque type de coups. Plus précisément, les trajectoires Topspin présentent les variations les plus importantes, tandis que les coups Poussette présentent une variation moindre. En ce qui concerne le coefficient de traînée C_d , fixé ici à 0,5 (section 3.1), le modèle l’estime avec précision dès les premières itérations de l’entraînement. Notre méthode, qui re-

TABLE 2 – Erreurs moyennes des vitesses de rotation et de translation des différentes méthodes pour toutes les classes de coups. Elles sont arrondies au centième. \mathcal{E}_{ω_0} en rps, \mathcal{E}_{V_0} en m/s.

Méthode	Calandre <i>et al.</i> [2]		PINNS	
	\mathcal{E}_{ω_0}	\mathcal{E}_{V_0}	\mathcal{E}_{ω_0}	\mathcal{E}_{V_0}
TopSpin	4.48	0.75	3.13	0.00
Contre Attaque	2.70	0.16	0.86	0.00
Poussette	0.99	0.09	0.85	0.00
Moyenne	3.04	0.41	1.61	0.00

pose sur un PINN, améliore les résultats de prédiction des paramètres cinématiques, le tableau 2 montre les erreurs moyennes de la vitesse de rotation et de translation pour les différentes méthodes. En outre, il est important de noter que notre modèle est capable de traiter des trajectoires non planaires, produites par l’effet Magnus, ce qui n’a pas été pris en compte dans l’article cité en référence.

4.3 Validation de la méthode proposée sur des séquences vidéo réelles

Dans cette section, notre méthode est testée sur des séquences réelles provenant de [2]. Ces séquences ont été filmées dans une salle de sport par deux caméras synchro-

nisées à grande vitesse (240 images par secondes). Après calibrage des caméras, les positions 3D des balles sont reconstruites par triangulation [9] à partir de leurs positions 2D sur chacune des caméras, obtenues par un tracker 2D CRST [12]. La figure 1 montre la configuration du dispositif d’acquisition. Le jeu de données est annoté en fonction des trois catégories de coups (voir figure 8). La méthode

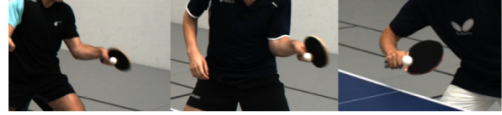


FIGURE 8 – Images extraites d’un Topspin (gauche), d’une Contre-Attaque (au centre) et d’une Poussette (à droite) (d’après [2])

proposée a été testée sur 33 trajectoires différentes (12 Topspin, 8 Contre-Attaque et 13 Poussette). Classiquement, pour un objet de faible diamètre apparent comme une balle, les changements soudains de direction et le flou de mouvement compliquent son suivi. Toutefois, il convient de noter que les points manquants pendant le suivi ne constituent pas nécessairement un problème pour notre méthode, car celle-ci peut fonctionner avec peu de données comme cela est illustré sur la figure 5(a). Actuellement, notre modèle ne prend pas en compte le rebond de la balle sur la table et nous traitons chaque trajectoire séparément. En utilisant le même processus d’apprentissage appliqué sur les données simulées, nous considérons en entrée quelques points de la trajectoire réelle 3D pour calculer l’erreur sur les données \mathcal{L}_s et quelques points de collocation \mathcal{N}_c pour l’erreur physique. La figure 9 montre quelques trajectoires obtenues par le PINN pour différentes classes de coups, ainsi que les vitesses de translation et de rotation estimées. Nous observons que le coup Poussette, correspond, comme prévu, à un ω_0 négatif, alors qu’il est positif pour les deux autres classes de coups. Pour évaluer notre modèle, nous mesurons l’erreur relative \mathcal{E}_p sur les positions 3D et sur la vitesse de translation \mathcal{E}_{V_0} . \mathcal{E}_{V_0} est calculée en utilisant l’estimation V_0 et les valeurs obtenues à partir des gradients des positions 3D reconstruites par stéréoscopie. Les résultats des erreurs relatives pour chaque type de coups sont présentés dans le tableau 3. L’erreur \mathcal{E}_p ,

TABLE 3 – Erreur relative moyenne sur les positions 3D et la vitesse de translation pour chaque type de coups.

Coup	Position \mathcal{E}_p	Translation \mathcal{E}_{V_0}
Topspin	0.003	0.064
Poussette	0.013	0.067
Congtre Attaque	0.005	0.093

commise sur l’estimation des positions, est très faible pour tous les types de coups. Ces résultats sont similaires à la validation sur des données simulées, avec une légère différence au bruit dans l’estimation des positions réelles par

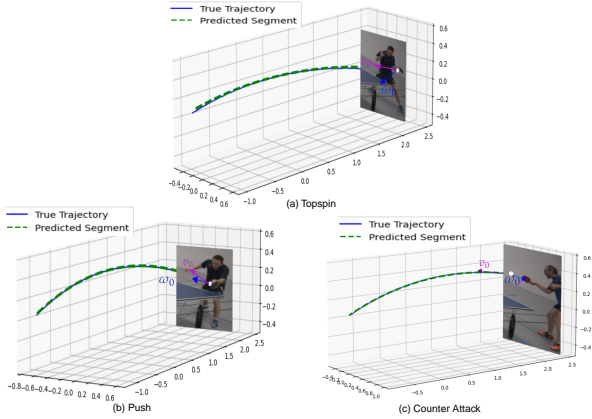


FIGURE 9 – Trajectoires prédites par le PINN à partir de données réelles pour différentes classes de coups.

triangulation. De même, l'erreur \mathcal{E}_{V_0} est faible sur les séquences réelles, ce qui signifie que la vitesse de translation est bien estimée pour toutes les classes de coups.

Contrairement aux séquences simulées, la vérité terrain pour ω_0 et C_d est non observable avec notre dispositif, et donc inconnues dans le jeu de données réelles. Cependant, ces valeurs peuvent être estimées pour vérifier et analyser leur cohérence. Ainsi, la moyenne du coefficient de traînée C_d , estimée à 0.44, est compatible avec les contraintes physiques sous-jacentes [19]. Le tableau 4 présente les valeurs moyennes des paramètres estimés par classes de coups.

TABLE 4 – Valeurs moyennes estimées de ω_0 , V_0 et C_d pour chaque type de coups ; ω_0 en rps, V_0 en m/s et C_d est sans dimension.

coup	Rotation ω_0	Translation V_0	Coefficient de traînée C_d
Topspin	52.38	14.18	0.38
Poussette	-28.07	6.73	0.53
Contre-Attaque	24.78	19.96	0.40

On peut noter que pour la classe `Topspin`, les valeurs de ω_0 et V_0 se situent dans l'intervalle des valeurs utilisées pour les données simulées (Section 4.1). Comme prévu, la rotation négative pour la coup `Poussette` a été estimée par la méthode. Afin d'évaluer si les valeurs de ω_0 et V_0 sont des attributs pertinents et peuvent être utilisées pour déterminer le type d'un coup donné, une classification à l'aide d'un SVM linéaire a été effectuée sur les 33 séquences réelles. Comme on peut l'observer sur la figure 10, il existe des frontières de décision linéaires claires, ce qui traduit une bonne séparation des classes par ces caractéristiques cinématiques.

5 Conclusion et perspectives

Dans ce travail, nous analysons les trajectoires de balles de tennis de table à l'aide d'une méthode basée sur une combinaison de réseaux neuronaux et d'équations physiques, appelée réseau neuronal physiquement informé

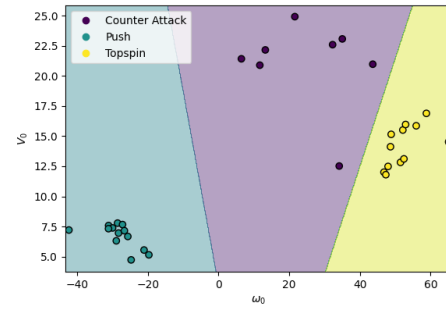


FIGURE 10 – Classification des séquences réelles en fonction des ω_0 et V_0 estimés.

(Physics-Informed Neural Network - PINN). Partant d'une loi de mouvement, nous simulons plusieurs classes de trajectoires de balles de tennis de table. Un petit nombre de données simulées associées aux contraintes physiques, nous permettent de construire un réseau neuronal capable de prédire les positions 3D et la cinématique 3D de la balle tout au long de sa trajectoire, qu'elle soit plane ou non, en prenant en compte les déviations dues à l'effet Magnus. Le modèle proposé est comparé à une autre méthode d'extraction de paramètres 3D à partir de trajectoires synthétiques de tennis de table, et nos résultats montrent que les PINNs améliorent nettement la précision de l'estimation des paramètres cinématiques (vitesse de rotation, vitesse de translation, coefficient de traînée). La validation est également faite sur un jeu de données réelles, qui montre que le PINN reste efficace malgré la présence de bruit d'acquisition et de reconstruction par stéréovision. La prochaine étape de ce travail sera d'utiliser des données 3D obtenues par vision monoculaire à l'aide d'un réseau profond qui estime la distance entre la caméra et la balle [2], l'objectif étant de permettre une utilisation non intrusive de la méthode dans des conditions de jeu réelles.

La méthode proposée s'est montrée efficace mais plusieurs pistes d'amélioration seront explorées. En particulier, la vitesse d'apprentissage du PINN peut être considérablement améliorée dans notre cas. Deux pistes sont à envisager : l'utilisation de l'intégration des caractéristiques de Fourier et de la factorisation des poids aléatoires est fortement recommandée pour accélérer la convergence [22], et l'amélioration du modèle en réécrivant les équations cinématiques dans un schéma numérique plus efficace et la modification des fonction d'erreur liée au physique [1].

Références

- [1] Hubert Baty and Léo Baty. Solving differential equations using physics informed deep learning : a hand-on tutorial with benchmark tests. *arXiv preprint arXiv :2302.12260*, 2023.

- [2] Jordan Calandre, Renaud Péteri, Laurent Mascarrilla, and Benoit Tremblais. Extraction and analysis of 3D kinematic parameters of table tennis ball from a single camera. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 9468–9475. IEEE, 2021.
- [3] Salvatore Cuomo, Vincenzo Schiano Di Cola, Fabio Giampaolo, Gianluigi Rozza, Maziar Raissi, and Francesco Piccialli. Scientific machine learning through physics-informed neural networks : where we are and what’s next. *Journal of Scientific Computing*, 92(3) :88, 2022.
- [4] Taco de Wolff, Hugo Carrillo, Luis Martí, and Nayat Sanchez-Pi. Towards optimally weighted physics-informed neural networks in ocean modelling. *arXiv preprint arXiv :2106.08747*, 2021.
- [5] Luca Fortini, Mattia Leonori, Juan M Gandarias, Elena de Momi, and Arash Ajoudani. Markerless 3D human pose tracking through multiple cameras and ai : Enabling high accuracy, robustness, and real-time performance. *arXiv preprint arXiv :2303.18119*, 2023.
- [6] Erik Gärtner, Mykhaylo Andriluka, Erwin Coumans, and Cristian Sminchisescu. Differentiable dynamics for articulated 3D human motion reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13190–13200, 2022.
- [7] Knut Erik Teigen Giljarhus, Mika Tobias Gooding, and Joar Njærheim. Disc golf trajectory modelling combining computational fluid dynamics and rigid body dynamics. *Sports Engineering*, 25(1) :26, 2022.
- [8] John Eric Goff and Matt J Carré. Soccer ball lift coefficients via trajectory analysis. *European Journal of Physics*, 31(4) :775, 2010.
- [9] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York, NY, USA, 2 edition, 2003.
- [10] George Em Karniadakis, Ioannis G Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6) :422–440, 2021.
- [11] Diederik P Kingma and Jimmy Ba. Adam : A method for stochastic optimization. *arXiv preprint arXiv :1412.6980*, 2014.
- [12] Alan Lukezic, Tomas Vojir, Luka ˇCehovin Zajc, Jiri Matas, and Matej Kristan. Discriminative correlation filter with channel and spatial reliability. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6309–6318, 2017.
- [13] William McNally, Jacob Lambeth, and Dustin Brekke. Combining physics and deep learning models to simulate the flight of a golf ball. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5118–5127, 2023.
- [14] Nathaniel Michek, Piyush Mehta, and Wade Huebsch. Methodology development of a free-flight parameter estimation technique using physics-informed neural networks. In *2023 IEEE Aerospace Conference*, pages 1–18. IEEE, 2023.
- [15] Douglas V Nance. Two numerical algorithms for the ballistic motion equations. Technical report, Technical Report, DOI : 10.13140/RG.2.2.20482.32960, 2019.
- [16] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks : A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378 :686–707, 2019.
- [17] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics informed deep learning (part i) : Data-driven solutions of nonlinear partial differential equations. *arXiv preprint arXiv :1711.10561*, 2017.
- [18] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics informed deep learning (part ii) : Data-driven discovery of nonlinear partial differential equations. *ArXiv*, abs/1711.10566, 2017.
- [19] Garry Robinson and Ian Robinson. The motion of an arbitrarily rotating spherical projectile and its application to ball games. *Physica Scripta*, 88(1) :018101, 2013.
- [20] Jost Seifert. A review of the magnus effect in aeronautics. *Progress in aerospace sciences*, 55 :17–45, 2012.
- [21] Antonín Štěpánek. The aerodynamics of tennis balls—the topspin lob. *American Journal of Physics*, 56(2) :138–142, 1988.
- [22] Sifan Wang, Shyam Sankaran, Hanwen Wang, and Paris Perdikaris. An expert’s guide to training physics-informed neural networks. *arXiv preprint 2308.08468*, 2023.
- [23] Petrisa Zell, Bastian Wandt, and Bodo Rosenhahn. Joint 3D human motion capture and physical analysis from monocular videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 17–26, 2017.
- [24] Aleš Zita and Filip Šroubek. Tracking fast moving objects by segmentation network. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 10312–10319. IEEE, 2021.