

La Science du Coup Franc : Analyse, Modélisation et Optimisation de la Précision en Football

HALIMI Akram

2023-2024

1 Introduction

Un coup franc direct ou indirect est accordé à une équipe lorsqu'un joueur subit une faute de la part d'un adversaire. Les fautes les plus graves entraînent un coup franc direct, représentant ainsi une occasion importante de marquer si le tir est bien exécuté. Cependant, même les joueurs de classe mondiale éprouvent des difficultés à marquer régulièrement sur coup franc, comme le montre l'exemple notoire de Kylian Mbappé. Cette étude vise donc à améliorer la performance individuelle des joueurs à travers une analyse approfondie et une modélisation du coup franc.

2 Problématique

Comment la modélisation et l'analyse approfondie du coup franc en football peuvent-elles contribuer à améliorer les compétences individuelles des joueurs ?

3 Plan d'Étude

Notre démarche s'articule en plusieurs étapes :

1. **Étude et analyse des phénomènes physiques** mis en jeu lors du tir.
2. **Établissement des équations** régissant le tir et implémentation informatique.
3. **Expérimentation et validation** du modèle informatique.
4. **Conclusion et perspectives**, mettant en avant les apports et les limites de notre modèle.

4 Étude et Analyse des Phénomènes Physiques

Lors d'un coup franc, plusieurs forces influencent la trajectoire du ballon :

- **Poids :**

$$\vec{P} = m\vec{g}$$

où m est la masse du ballon et g l'accélération gravitationnelle.

- **Force de traînée :** Lorsque le ballon se déplace dans l'air à une vitesse v , l'écoulement autour devient turbulent, caractérisé par le nombre de Reynolds (Re) :

$$Re = \frac{\rho dv}{\eta}$$

avec d le diamètre du ballon, ρ la densité de l'air et η la viscosité dynamique de l'air.

Pour un ballon de football avec les paramètres suivants : $d = 0.11\text{ m}$, $\eta = 1.8 \times 10^{-5}\text{ Pa} \cdot \text{s}$, $\rho = 1.2\text{ kg} \cdot \text{m}^{-3}$ et $1 < v < 50\text{ m} \cdot \text{s}^{-1}$, on trouve :

$$7.3 \times 10^3 < Re < 3.7 \times 10^5$$

Dans ce domaine, la force de traînée est donnée par :

$$\vec{F}_d = -\frac{1}{2}\rho AC_x v \vec{v}$$

où A est la surface frontale et C_x le coefficient de traînée, supposé constant dans cette étude.

- **Force de Magnus** : Lorsqu'un ballon tourne en se déplaçant dans un fluide, une différence de pression se crée autour du ballon, causant une déviation de sa trajectoire. Cette force est décrite par :

$$\vec{F}_{mag} = \alpha \vec{\omega} \wedge \vec{v}$$

où $\vec{\omega}$ est le vecteur de rotation du ballon et α un coefficient lié aux caractéristiques physiques et aérodynamiques.

Dans notre modélisation, on utilise les expressions :

$$F_{MC} = \frac{1}{2}\rho AC_{MC}v^2 \quad \text{et} \quad F_{MB} = \frac{1}{2}\rho AC_{MB}v^2$$

avec

$$C_{MC} = \frac{1}{2 + \frac{v}{Re_z}} \quad \text{et} \quad C_{MB} = \frac{1}{2 + \frac{v}{Re_y}}$$

où Re_z et Re_y sont les nombres de Reynolds relatifs aux axes respectifs.

5 Mise en Place Détaillée des Équations

On considère le ballon comme système étudié dans un référentiel terrestre supposé galiléen. Le bilan des actions mécaniques extérieures (BAME) exercées sur le ballon est :

$$\vec{P}, \quad \vec{f}, \quad \vec{F}_{MC}, \quad \vec{F}_{MB}$$

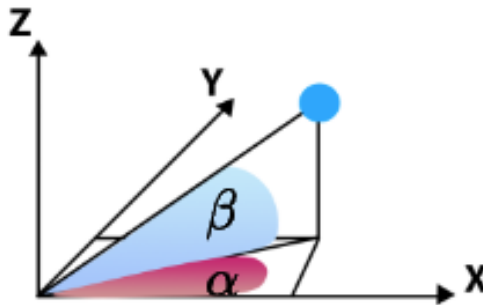
où :

- \vec{P} est le poids,
- \vec{f} la force de traînée,
- \vec{F}_{MC} et \vec{F}_{MB} les composantes horizontale et verticale de la force de Magnus.

Pour simplifier les notations, on définit la constante :

$$k = \frac{1}{2m}\rho A$$

La vitesse initiale du ballon est projetée selon les axes du repère :



$$\begin{cases} v_x = v \cos(\beta) \cos(\alpha) \\ v_y = v \cos(\beta) \sin(\alpha) \\ v_z = v \sin(\beta) \end{cases}$$

où α et β sont les angles définissant la direction du tir.

5.1 Principe Fondamental de la Dynamique (PFD)

En appliquant le PFD, on obtient :

$$m\vec{a} = \vec{P} + \vec{f} + \vec{F}_{MC} + \vec{F}_{MB}$$

5.2 Projections sur les Axes

En projetant les forces sur les axes, on obtient :

Sur l'axe x :

$$a_x = -kv(C_x v_x + C_{MC} v \sin(\alpha) - C_{MB} v_z \cos(\alpha))$$

Sur l'axe y :

$$a_y = -kv(C_x v_y - C_{MC} v \cos(\alpha) - C_{MB} v_z \sin(\alpha))$$

Sur l'axe z :

$$a_z = -kv(C_x v_z + C_{MB} v \cos(\beta)) - g$$

5.3 Système d'Équations Différentielles

Le mouvement du ballon est ainsi régi par un système d'équations différentielles du second ordre :

$$\begin{cases} \ddot{x} = -kv(C_x \dot{x} + C_{MC} v \sin(\alpha) - C_{MB} \dot{z} \cos(\alpha)) \\ \ddot{y} = -kv(C_x \dot{y} - C_{MC} v \cos(\alpha) - C_{MB} \dot{z} \sin(\alpha)) \\ \ddot{z} = -kv(C_x \dot{z} + C_{MB} v \cos(\beta)) - g \end{cases}$$

Ces équations seront résolues numériquement par la méthode de Runge-Kutta.

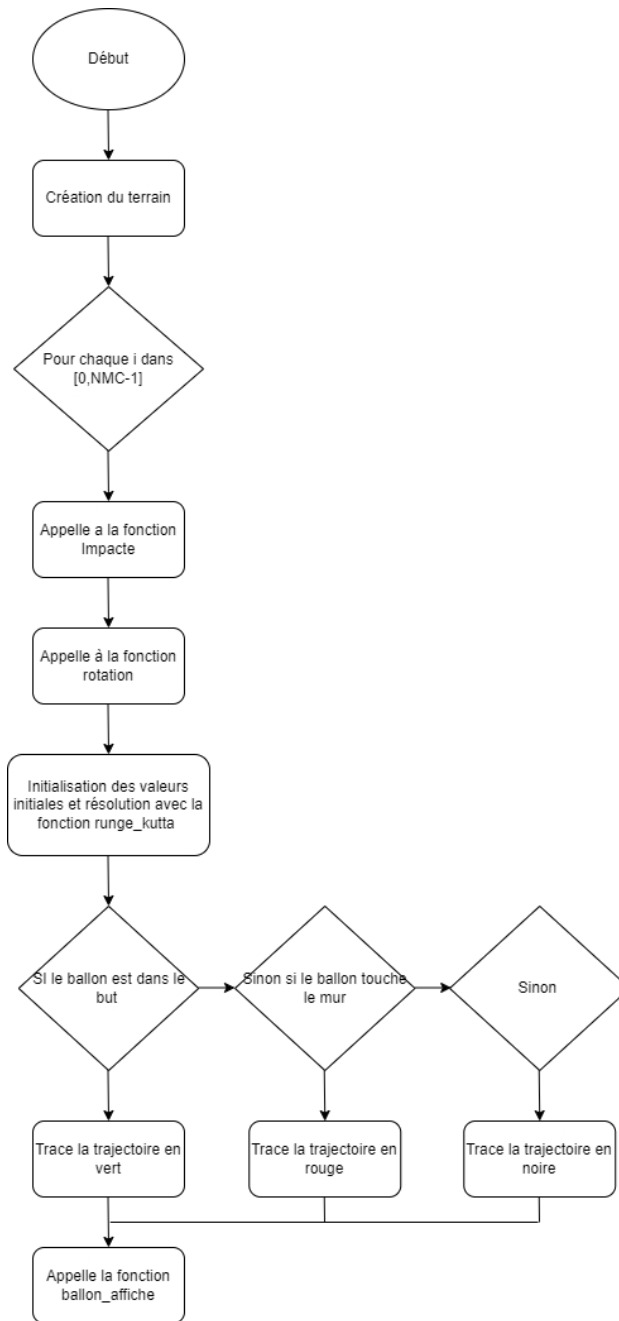
6 Implémentation Informatique Détaillée

6.1 Hypothèses et Détails

On suppose que le ballon est une sphère lisse avec un coefficient de traînée $C_x = 0.47$. Le contact entre le pied et le ballon est supposé ponctuel, et les trajectoires sont calculées dans des conditions météorologiques standards avec un vent négligeable.

6.2 Description du Programme

Le programme informatique prend en compte la force exercée par le pied sur le ballon afin de calculer les coordonnées du point d'impact et les angles optimaux pour un tir précis, comme l'illustre le diagramme général fourni.



6.3 Fonction impact : Simulation Monte-Carlo

La fonction **impact** constitue le cœur du modèle statistique. Elle génère des nombres aléatoires selon une distribution normale pour simuler l'incertitude du point d'impact sur le ballon. Elle retourne un pourcentage pour les coordonnées Y et Z , ainsi que les angles α et β nécessaires pour optimiser la trajectoire.

```

1 def Impact():
2     ...
3     Randomly generates impact parameters:
4     Yc = y-coordinate of impact point as a percentage of radius
5     Zc = z-coordinate of impact point as a percentage of radius
6     alphaHit = shot angle in degrees (x-y plane)
7     betaHit = shot angle in degrees (x-y-r plane)
8     ...
9     randomY = 0.1 * np.random.randn(1, 1)
10    randomZ = 0.1 * np.random.randn(1, 1)
11    randomalpha = 0.1 * np.random.randn(1, 1)
12    randombeta = 0.1 * np.random.randn(1, 1)
13
14    Yc = randomY[0][0]
15    Zc = randomZ[0][0]
16    alphaHit = randomalpha[0][0]
17    betaHit = randombeta[0][0]
18
19    return Yc, Zc, alphaHit, betaHit

```

6.4 Fonction rotation : Calcul des Composantes de Rotation

La fonction **rotation** détermine les composantes de rotation du ballon en utilisant le moment cinétique :

$$\vec{J}\omega = \overrightarrow{OM} \wedge m\vec{v} = \begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ a & b & c \\ mv_x & mv_y & mv_z \end{vmatrix} = \begin{pmatrix} bv_z - cv_y \\ cv_x - av_z \\ av_y - bv_x \end{pmatrix}$$

Avec la formule de l'impulsion :

$$\left\{ v_x = \frac{F_x}{m}, \quad v_y = \frac{F_y}{m}, \quad v_z = \frac{F_z}{m} \right.$$

Et le moment d'inertie pour une sphère :

$$J = \frac{2}{5}mr^2$$

Les composantes de rotation deviennent :

$$\begin{cases} \omega_x = \frac{5}{2mr^2}(bF_z - cF_y) \\ \omega_y = \frac{5}{2mr^2}(cF_x - aF_z) \\ \omega_z = \frac{5}{2mr^2}(aF_y - bF_x) \end{cases}$$

6.5 Résolution Numérique par Runge-Kutta

Pour appliquer Runge-Kutta, on transforme le système en équations du premier ordre :

$$\begin{cases} x_1 = \dot{x}, & y_1 = \dot{y}, & z_1 = \dot{z} \\ \dot{x}_1 = \ddot{x}, & \dot{y}_1 = \ddot{y}, & \dot{z}_1 = \ddot{z} \end{cases}$$

```

def runge_kutta_4th_order(f, y0, t_0, t_end, h, v, C_MC, C_MB, sinA, cosA, cosB, distance):
    ...
    Solves the system using 4th-order Runge-Kutta method.
    ...
    num_steps = int((t_end - t_0) / h) + 1
    t_values = np.linspace(t_0, t_end, num_steps)
    y_values = np.zeros((num_steps, len(y0)))
    y_values[0] = y0
    for i in range(1, num_steps):
        k1 = h * f(t_values[i - 1], y_values[i - 1], v, C_MC, C_MB, sinA, cosA, cosB)
        k2 = h * f(t_values[i - 1] + 0.5 * h, y_values[i - 1] + 0.5 * k1, v, C_MC, C_MB, sinA, cosA, cosB)
        k3 = h * f(t_values[i - 1] + 0.5 * h, y_values[i - 1] + 0.5 * k2, v, C_MC, C_MB, sinA, cosA, cosB)
        k4 = h * f(t_values[i - 1] + h, y_values[i - 1] + k3, v, C_MC, C_MB, sinA, cosA, cosB)
        y_values[i] = y_values[i - 1] + (k1 + 2 * k2 + 2 * k3 + k4) / 6
    # Check if the ball is out of bounds
    if (y_values[i, 0] < 0) or (y_values[i, 2] < -10) or (y_values[i, 2] > 10) or y_values[i, 0] > distance:
        return y_values[i+1]
    # Check if the ball hits the ground
    if y_values[i, 4] < 0:
        return y_values[i+1]
    # Check if the ball scores a goal
    if y_values[i, 0] == distance and (y_values[i, 4] > 2.44) and (-3.66 < y_values[i, 2] < 3.66):
        return y_values[i+1]
    # Check if it hits the wall
    if y_values[i, 0] == 9.1 and ((y_values[i, 4] < 2.00) and (-1.5 < y_values[i, 2] < 0)):
        return y_values[i+1]
    return y_values

```

7 Expériences et Validation du Modèle

7.1 Expérience 1 : Reproduction du Tir de Roberto Carlos

Nous reproduisons le célèbre tir de Roberto Carlos depuis 35 mètres avec une impulsion estimée à 16.72 N.s, en générant 1000 tirs aléatoires via une méthode de Monte Carlo.

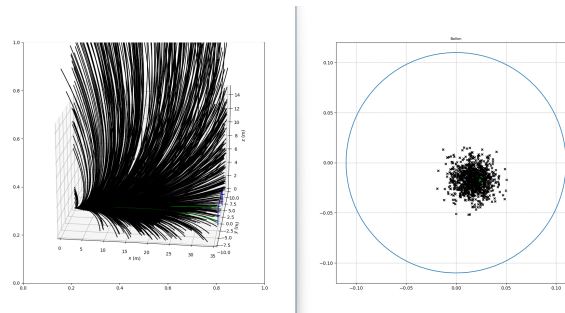
7.1.1 Première Série

$$Y = 0.1 \times np.random.randn(1, 1) + 0.16$$

$$Z = 0.1 \times np.random.randn(1, 1) - 0.150$$

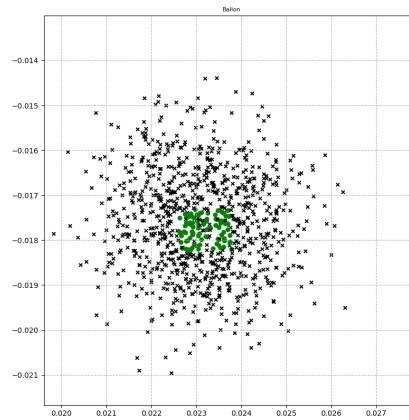
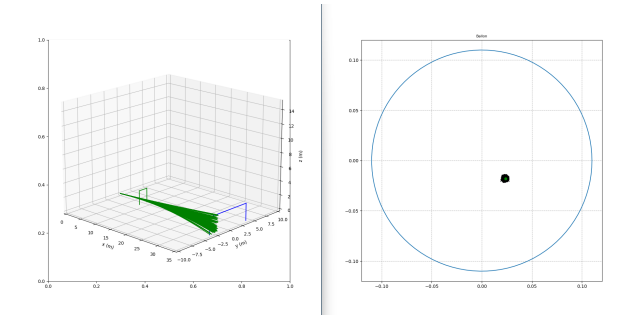
$$\alpha = 0.1 \times np.random.randn(1, 1) - 14$$

$$\beta = 0.1 \times np.random.randn(1, 1) + 7.5$$



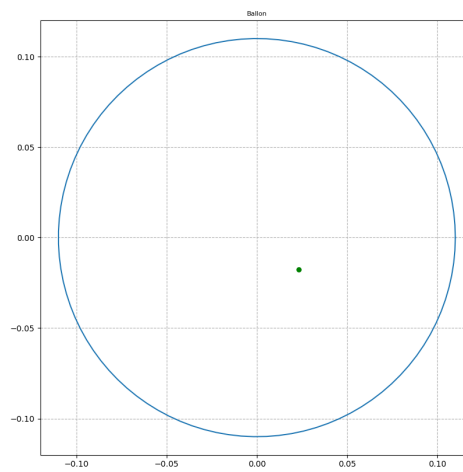
7.1.2 Affinement

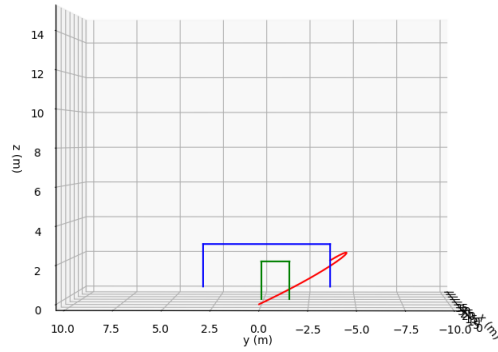
En recentrant la moyenne des tirages aléatoires sur les meilleures valeurs trouvées :



7.1.3 Dernière Série

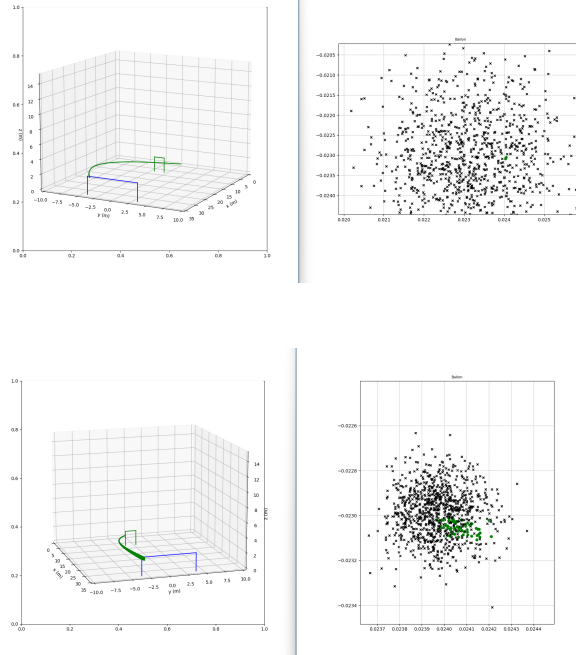
Paramètres finaux : $Y = 0.21$, $Z = -0.16$, $\alpha = -14$, $\beta = 7.66$, vitesse initiale 38.7 m/s proche des valeurs réelles (38.1 m/s , angle 12°).





7.2 Expérience 2 : Optimisation Personnelle d'un Coup Franc

En visant la lucarne droite avec une vitesse mesurée de 21.5 m/s , trois séries de simulations Monte Carlo aboutissent à une trajectoire optimale :



8 Conclusion

Notre modélisation permet de reproduire des coups francs avec une certaine variabilité. Cependant, bien qu'elle constitue une aide précieuse pour la compréhension et l'amélioration des tirs, ses résultats restent limités : sur 20 tirs, seulement 3 étaient pleinement satisfaisants.

Elle ne remplace pas l'entraînement, mais sert d'outil complémentaire. De nombreux facteurs restent ignorés : conditions météorologiques, texture du ballon, état mental du joueur. Ces éléments peuvent influencer la trajectoire comme ce fut le cas avec le ballon Jabulani lors de la Coupe du Monde 2010.

Notre travail constitue donc une base théorique solide, mais montre l'importance d'intégrer progressivement ces variables pour rendre le modèle plus réaliste et performant.