REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
الجمهورية الجزائرية الديمقراطية الشعبية

MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

HIGHER SCHOOL IN APPLIED SCIENCES
–TLEMCEN–

وزارة التعليم العالي والبحث العلمي

المدرسة العليا في العلوم التطبيقية
-تلمسان-

Mémoire de fin d'étude

Pour l'obtention du diplôme d'Ingénieur

Filière : **Électronique**
Spécialité : **Informatique industrielle**

Présenté par : Mohamed Amine LAIB & Adil Marwane MAHI

Thème

Conception et développement d'un système embarqué instrumenté connecté pour l'analyse temps réel du jeu

Sous la direction de :
M.Anisse CHIALI

Soutenu publiquement, le 22/06/2026
Devant les membres du jury, composé de :

Dr. Belloufa Fayssal	MCB	ESSA. Tlemcen	**Président**
Dr. MIDOUN Nadjia	MCB	ESSA. Tlemcen	**Examinateur**

Année universitaire : 2025 / 2026

Dédicace -01- :

À mes très chers parents, pour leur amour, leurs sacrifices et leur soutien indéfectible tout au long de mon parcours. Aucune parole ne pourra exprimer toute ma reconnaissance pour tout ce qu'ils ont fait pour moi.

À mon enseignant, M.Farid OUDJAMA, en témoignage de ma profonde gratitude pour ses conseils, son aide précieuse et son encouragement tout au long de ce travail.

À mon cher frère Sofiane et à ma chère sœur Manel , pour leur affection, leurs encouragements et leur présence constante à mes côtés.

À mes chers neveux et nièces : Lamia et Mohamed Khedim, Joury et Mohamed-Rayane, à qui je souhaite un avenir plein de réussite et de bonheur.

À mon binôme Mohammed Amine LAIB, avec qui j'ai partagé les efforts, les défis et les moments importants durant la réalisation de ce projet.

À tous mes amis, particulièrement Chakib CHATER, Aboubakar GUEBRILI et Imad HADJSLIMANE ainsi qu'à tous mes camarades de la promotion électronique, pour les bons moments partagés durant ces années d'études.

À tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail.

Merci.

- Adil Marwane MAHI

Dédicace -02- :

À mes parents, pour votre soutien moral et financier inébranlable, vos encouragements constants et les sacrifices que vous avez consentis afin que je puisse poursuivre mes études et mener à bien ce projet. Merci de croire en moi et d'être toujours à mes côtés à chaque étape de mon parcours.

À mon frère, pour avoir toujours cru en mes capacités et pour m'avoir encouragé à aller jusqu'au bout dans la réalisation de ce projet.

À mon binôme, Adil Marwane MAHI, pour ton travail continu, ta persévérance et ton enthousiasme tout au long de la réalisation de ce projet.

À mes amis, GUEBRILI Aboubakar , REZIG Ahmed Khodir, Yacine BEMRAH et Youssef BEMRAH pour votre aide précieuse, votre soutien technique et moral ainsi que pour vos conseils avisés qui ont grandement contribué à la réalisation de ce travail.

À tous ceux qui ont contribué de près ou de loin à la réalisation de ce travail.

Merci.

- Mohamed Amine LAIB

Remerciements

Avant tout, nous rendons grâce à Dieu, le Tout-Puissant, pour la force et la persévérance accordées pour mener à bien ce travail.

Nous tenons à exprimer notre profonde gratitude et nos sincères remerciements à notre encadrant, M. Anisse CHIALI, pour la confiance qu'il nous a accordée, pour son accompagnement précieux, ses orientations pertinentes et ses conseils éclairés tout au long de la réalisation de ce projet. Sa disponibilité, sa rigueur scientifique et son soutien constant ont été pour nous d'une grande valeur.

Nos remerciements les plus profonds et les plus sincères s'adressent à notre enseignante., Mme Nadjia MIDOUN, dont l'implication exceptionnelle, le suivi rigoureux, la disponibilité permanente et le dévouement sans faille ont joué un rôle central et déterminant dans la réussite de ce travail. Son soutien indéfectible, ses conseils éclairés et l'intérêt profond qu'elle a porté à notre projet nous ont été d'une valeur inestimable.

Nous avons l'honneur d'adresser nos sincères remerciements aux membres du jury, qui ont accepté d'évaluer ce modeste travail. Nous leur sommes reconnaissants pour le temps qu'ils consacreront à examiner ce mémoire ainsi que pour leurs remarques et leurs précieuses recommandations.

Nos remerciements vont également à M. Ghouti ABDELLAOUI, chef de filière, et à Mme Elyasmine KADAOUI-ELAIDI, responsable de l'incubateur, pour leur accompagnement, leurs conseils et leurs encouragements.

Nous remercions sincèrement M. Ramzou, responsable du FABLAB, pour son aide et son soutien tout au long de la réalisation de ce projet.

Nous sommes reconnaissants envers tous les enseignants pour leur contribution à notre formation, ainsi qu'à toutes les personnes ayant contribué de près ou de loin à ce travail.

Résumé

Le travail présenté s'inscrit dans le domaine de l'électronique, option informatique industrielle, et porte sur la conception et la réalisation d'un ballon de football intelligent connecté associé à une plateforme IoT pour l'analyse du jeu en temps réel. L'objectif principal est de développer un système capable de collecter des données liées au comportement du ballon pendant le jeu, puis de les transmettre vers un environnement numérique permettant leur visualisation et leur exploitation. Le système proposé repose sur l'intégration d'un dispositif électronique embarqué dans le ballon afin de mesurer certains paramètres liés aux mouvements, aux impacts et aux interactions avec le jeu. Les données recueillies sont ensuite traitées et envoyées vers une plateforme IoT, qui permet de les organiser, de les afficher et de les présenter sous forme d'indicateurs utiles pour l'analyse sportive.

Mots clés : Électronique embarquée, ballon intelligent connecté, plateforme IoT, analyse du jeu, données sportives, temps réel.

Abstract

The presented work falls within the field of electronics, specializing in industrial computing, and focuses on the design and implementation of a connected smart football associated with an IoT platform for real-time game analysis. The main objective is to develop a system capable of collecting data related to the behavior of the ball during play and transmitting it to a digital environment for visualization and exploitation. The proposed system is based on the integration of an embedded electronic device inside the ball in order to measure certain parameters related to motion, impacts, and interactions with the game. The collected data are then processed and sent to an IoT platform, which allows them to be organized, displayed, and presented as useful indicators for sports analysis.

Keywords : Embedded electronics, connected smart football, IoT platform, game analysis, sports data, real-time.

ملخص

يندرج هذا العمل ضمن مجال الإلكترونيات، تخصص الإعلام الآلي الصناعي، ويركز على تصميم وإنجاز كرة قدم ذكية متصلة مرتبطة بمنصة إنترنت الأشياء من أجل تحليل اللعب في الزمن الحقيقي. الهدف الرئيسي من هذا المشروع هو تطوير نظام قادر على جمع البيانات المتعلقة بسلوك الكرة أثناء اللعب، ثم إرسالها إلى بيئة رقمية تسمح بعرضها واستغلالها. يعتمد النظام المقترح على دمج جهاز إلكتروني مدمج داخل الكرة من أجل قياس بعض المعطيات المرتبطة بالحركة، الصدمات والتفاعلات أثناء اللعب. بعد ذلك، تتم معالجة البيانات المجمعة وإرسالها نحو منصة إنترنت الأشياء، حيث يتم تنظيمها وعرضها وتقديمها على شكل مؤشرات مفيدة للتحليل الرياضي.

الكلمات المفتاحية : الإلكترونيات المدمجة، كرة قدم ذكية متصلة، منصة إنترنت الأشياء، تحليل اللعب، البيانات الرياضية، الزمن الحقيقي.

Sommaire

Remerciements

Résumés

Liste des abréviations

Introduction générale **1**

Chapitre -01- : État de l'art sur le football connecté **4**

1.1. Introduction 5
1.2. Analyse du jeu de football basée sur les données 5
1.2.1. Importance des données dans le football moderne 5
1.2.2. Rôle du ballon dans l'analyse du jeu 6
1.2.3. Événements de jeu liés au ballon 6
1.2.4. Limites de l'observation humaine 7
1.3. Étude des solutions de ballons instrumentés du marché 7
1.3.1. Principe général de l'instrumentation d'un ballon 8
1.3.2. Adidas Al Rihla – Coupe du Monde de la FIFA 2022 8
1.3.3. Adidas Trionda – Coupe du Monde de la FIFA 2026 9
1.3.4. KINEXON xBall 10
1.3.5. Comparaison des solutions de ballons instrumentés 11
1.4. Revue des plateformes IoT et systèmes d'analyse sportive 12
1.4.1. Rôle d'une plateforme IoT dans le sport connecté 12
1.4.2. Plateforme ThingsBoard 13
1.4.3. Plateforme sportive KINEXON 14
1.4.4. Écosystème SPRO/SVIVO (WIMU PRO) 15
1.4.5. Comparaison des plateformes étudiées 15
1.5. Technologies clés des ballons instrumentés et plateformes IoT 16
1.5.1. Capteurs de mouvement 17
1.5.2. Capteurs d'impact et de contact 17
1.5.3. Technologies de localisation UWB 18
1.5.4. Microcontrôleurs et traitement embarqué 19

1.5.5. Communication sans fil et transmission des données 20
1.6. Synthèse comparative et limites des solutions existantes 20
1.6.1. Analyse de l'intégration des solutions existantes 21
1.6.2. Limites techniques 21
1.6.3. Limites économiques et d'accessibilité 21
1.6.4. Besoins identifiés pour une solution académique 21
1.7. Conclusion 22

Chapitre -02- : Modélisation théorique du ballon et simulation du circuit électronique 23

2.1. Introduction 24
2.2. Modélisation physique du ballon 25
2.2.1. Analyse des forces appliquées sur le ballon 25
2.2.2. Équations du mouvement du ballon 28
2.2.3. Résolution numérique et analyse des trajectoires 29
2.2.3.1. Définition des paramètres de simulation 29
2.2.3.2. Analyse des résultats 30
2.3. Conception et modélisation électronique du système 33
2.3.1. Architecture électronique globale du système embarqué 33
2.3.2. Modélisation et conditionnement des capteurs FSR402 35
2.3.2.1. Conversion analogique-numérique par l'ESP32 36
2.3.2.2. Localisation approximative du point d'impact 36
2.3.3. Modélisation du capteur inertiel LSM6DS3 37
2.3.4. Filtrage des données mesurées 38
2.3.5. Algorithme de traitement des données embarqué 38
2.3.6. Évaluation du bilan énergétique du système 39
2.4. Simulation et validation du circuit sous Proteus 40
2.4.1. Schématique du circuit simulé 40
2.4.2. Principe de fonctionnement de la simulation 41
2.4.3. Analyse des flux de données (terminal virtuel) 42
2.4.4. Interprétation et discussion des résultats de simulation 43
2.5. Conclusion 44

Chapitre -03- : Réalisation finale, intégration IoT et validation expérimentale du ballon instrumenté 45

3.1. Introduction 46
3.2. Réalisation finale du prototype du ballon 46
3.2.1. Réalisation des sous-systèmes du prototype 46
3.2.1.1. Sous-système mécanique 46
3.2.1.2. Fabrication de la structure externe du ballon 47

3.2.1.3. Réalisation de la capsule interne . . . 48
3.2.1.4. Sous-système d'alimentation . . . 50
3.2.1.5. Sous-système d'acquisition et de traitement . . . 51
3.2.1.6. Analyse comparative entre le circuit de simulation et réalisation . . . 54
3.2.2. Intégration finale du système complet . . . 55
3.3. Intégration IoT et réalisation du tableau de bord KOURATI . . . 58
3.3.1. Présentation générale du tableau de bord . . . 58
3.3.2. Architecture générale du système . . . 58
3.3.2.1. Les capteurs embarqués : LSM6DS3 et G-87 . . . 59
3.3.2.2. Architecture en couches détaillée . . . 59
3.3.2.3 Diagramme Fonctionnel du Système KOURATI . . . 60
3.3.2.4 Diagramme de Flux de Données . . . 61
3.3.3 Communication des Données avec MQTT . . . 62
3.3.3.1 Présentation de MQTT . . . 62
3.3.3.2 Choix du protocole MQTT . . . 62
3.3.3.3 Le rôle du broker Mosquitto . . . 63
3.3.3.4 Réception par Flask et envoi vers le dashboard . . . 63
3.3.4 Format JSON des Données . . . 63
3.3.4.1 Présentation de JSON . . . 63
3.3.4.2 Structure d'un message JSON du ballon . . . 63
3.3.5 API REST . . . 64
3.3.5.1 Principe de l'API REST . . . 64
3.3.5.2 REST vs communication temps réel . . . 64
3.3.5.3 Endpoints principaux . . . 65
3.3.6 Communication Temps Réel avec WebSocket . . . 65
3.3.6.1 Présentation du WebSocket . . . 65
3.3.6.2 Choix du WebSocket dans KOURATI . . . 66
3.3.6.3 Broadcast via Socket.IO . . . 66
3.3.7 Interface Utilisateur — Frontend . . . 66
3.3.7.1 Tableau de score et informations du match . . . 67
3.3.7.2 Indicateurs de statut et LED . . . 67
3.3.7.3 Terrain interactif et suivi du ballon . . . 67
3.3.7.4 Statistiques comparées . . . 67
3.3.7.5 Métriques en temps réel . . . 67
3.3.7.6 Orientation du ballon — Capteur G-87 . . . 67
3.3.8 Flux Complet des Données . . . 68
3.4. Tests et validation expérimentale . . . 69
3.4.1. Premier essai . . . 69
3.4.2. Deuxième essai . . . 69
3.4.3. Validation de la détection des événements de jeu . . . 71

3.4.4. Analyse des résultats expérimentaux 73
3.5. Conclusion 73

Conclusion générale **75**
Bibliographie **78**
Annexes **86**
Index des figures **99**
Index des tableaux **101**

Liste des abréviations

Abréviation	**Signification**
ADC	*Analog-to-Digital Converter*, convertisseur analogique-numérique utilisé pour convertir une tension analogique en valeur numérique.
API	*Application Programming Interface*, interface permettant l'échange de données entre plusieurs applications ou systèmes.
BLE	*Bluetooth Low Energy*, technologie Bluetooth à faible consommation énergétique.
BMS	*Battery Management System*, circuit de protection utilisé pour sécuriser la charge et la décharge d'une batterie.
CAO	Conception Assistée par Ordinateur.
CoAP	*Constrained Application Protocol*, protocole de communication adapté aux objets connectés à ressources limitées.
CSV	*Comma-Separated Values*, format de fichier utilisé pour stocker des données sous forme de tableau.
DC-DC	Convertisseur continu-continu permettant d'adapter une tension électrique continue à une autre valeur.
DW1000	Circuit intégré UWB utilisé pour la communication et la localisation à courte portée.
DWM3000	Module UWB utilisé dans les applications de localisation et de communication sans fil.
EPTS	*Electronic Performance and Tracking Systems*, systèmes électroniques de suivi et de performance utilisés dans le sport.
ESP32	Microcontrôleur intégrant notamment le Wi-Fi et le Bluetooth, utilisé dans les systèmes embarqués et les objets connectés.
ESP32-S3	Version avancée de l'ESP32 utilisée comme unité de traitement et de communication dans le prototype.

FDM	*Fused Deposition Modeling*, procédé d'impression 3D par dépôt de matière fondue.
FIFA	*Fédération Internationale de Football Association*.
Flask	Framework Python utilisé pour développer le backend du tableau de bord web KOURATI.
FSR	*Force Sensing Resistor*, capteur résistif sensible à la force.
FSR402	Modèle de capteur résistif sensible à la force utilisé pour mesurer ou détecter une pression appliquée.
FSR406	Modèle de capteur résistif sensible à la force utilisé comme équivalent de simulation sous Proteus.
GNSS	*Global Navigation Satellite System*, système global de navigation par satellite.
GPS	*Global Positioning System*, système de positionnement par satellite.
GY-87	Module inertiel utilisé pour fournir des informations complémentaires d'orientation, notamment à travers le magnétomètre.
HTTP	*HyperText Transfer Protocol*, protocole de communication utilisé pour l'échange de données sur le Web.
HTTPS	*HyperText Transfer Protocol Secure*, version sécurisée du protocole HTTP.
I^2C	*Inter-Integrated Circuit*, bus de communication série utilisé pour relier des capteurs à un microcontrôleur.
IFAB	*International Football Association Board*, organisme responsable des lois du jeu de football.
IMU	*Inertial Measurement Unit*, unité de mesure inertielle regroupant généralement un accéléromètre et un gyroscope.
IoT	*Internet of Things*, Internet des Objets, permettant de connecter des objets physiques à des systèmes numériques.
ISIS	Environnement de conception et de simulation électronique intégré dans le logiciel Proteus.
ITU-T	*International Telecommunication Union – Telecommunication Standardization Sector*, secteur de normalisation des télécommunications de l'UIT.
JSON	*JavaScript Object Notation*, format léger d'échange de données utilisé dans les applications Web et IoT.
Li-ion	*Lithium-ion*, technologie de batterie rechargeable utilisée dans les systèmes embarqués.

LiPo	*Lithium Polymer*, type de batterie rechargeable légère utilisée dans les systèmes embarqués.
LPS	*Local Positioning System*, système de positionnement local.
LSM6DS3	Capteur inertiel MEMS intégrant un accéléromètre 3 axes et un gyroscope 3 axes.
LwM2M	*Lightweight Machine to Machine*, protocole destiné à la gestion des objets connectés.
MATLAB	Environnement de calcul scientifique utilisé pour la modélisation, la simulation et la visualisation des résultats.
MEMS	*Micro-Electro-Mechanical Systems*, microsystèmes électromécaniques utilisés dans plusieurs capteurs modernes.
Mosquitto	Broker MQTT open source utilisé comme intermédiaire de communication entre l'ESP32-S3 et le serveur.
MQTT	*Message Queuing Telemetry Transport*, protocole léger de communication très utilisé dans les systèmes IoT.
MPU6050	Capteur inertiel intégrant un accéléromètre 3 axes et un gyroscope 3 axes, utilisé comme équivalent fonctionnel du LSM6DS3 dans la simulation.
NoSQL	Type de base de données non relationnelle utilisée pour stocker des données flexibles ou volumineuses.
ODE	*Ordinary Differential Equation*, équation différentielle ordinaire.
ODE45	Fonction MATLAB utilisée pour résoudre numériquement des équations différentielles ordinaires.
PCB	*Printed Circuit Board*, carte électronique imprimée.
PLA	*Polylactic Acid*, matériau rigide utilisé en impression 3D pour fabriquer la capsule interne du prototype.
REST	*Representational State Transfer*, style d'architecture utilisé pour concevoir des services Web.
RTDB	*Realtime Database*, base de données en temps réel, notamment utilisée dans Firebase.
SPI	*Serial Peripheral Interface*, bus de communication série utilisé entre microcontrôleurs et périphériques.
SPRO	Logiciel d'analyse associé à l'écosystème WIMU PRO.
SQL	*Structured Query Language*, langage utilisé pour gérer et interroger des bases de données relationnelles.
SSL/TLS	Protocoles de sécurité utilisés pour chiffrer les communications sur un réseau.

SVIVO	Solution de visualisation en temps réel associée à l'écosystème WIMU PRO.
TDK	Entreprise spécialisée dans les composants électroniques et les capteurs, notamment liée aux capteurs InvenSense.
TP4056	Module de charge utilisé pour recharger une batterie lithium-ion à une seule cellule.
TPU	*Thermoplastic Polyurethane*, matériau flexible utilisé en impression 3D pour fabriquer la structure externe du prototype.
UI	*User Interface*, interface utilisateur d'une application ou d'une plateforme.
UIT	Union Internationale des Télécommunications.
USB	*Universal Serial Bus*, interface de communication utilisée pour connecter et alimenter des dispositifs électroniques.
UWB	*Ultra-Wideband*, technologie de communication sans fil utilisée notamment pour la localisation précise à courte portée.
VAR	*Video Assistant Referee*, assistance vidéo à l'arbitrage dans le football.
VSM	*Virtual System Modelling*, technologie de simulation utilisée dans Proteus pour simuler des systèmes à microcontrôleurs.
WebSocket	Technologie de communication temps réel permettant l'échange bidirectionnel entre le serveur et le navigateur.
Wi-Fi	Technologie de communication sans fil permettant la connexion à un réseau local ou à Internet.
WIMU	Solution portable de suivi sportif utilisée pour collecter et analyser des données de performance.

Introduction générale

Le football est aujourd'hui l'un des sports les plus pratiqués et les plus suivis dans le monde. Il occupe une place importante dans la société moderne, que ce soit dans le domaine sportif, économique, technologique ou médiatique. Avec l'évolution du niveau de compétition, l'analyse du jeu est devenue un élément essentiel pour les entraîneurs, les analystes, les clubs et les structures de formation. Comprendre les actions réalisées pendant un match, suivre le comportement du ballon et interpréter certains événements importants du jeu sont désormais des besoins majeurs pour améliorer la performance et soutenir la prise de décision.

Pendant longtemps, l'analyse du football reposait principalement sur l'observation humaine, l'expérience des entraîneurs et quelques statistiques simples comme le nombre de buts, de tirs, de passes ou de fautes. Même si cette approche reste importante, elle peut parfois être limitée, car le football est un sport rapide, dynamique et difficile à analyser uniquement à l'œil nu. Une frappe, un rebond, un changement brusque de direction ou un impact peuvent se produire en une fraction de seconde. Pour cette raison, l'utilisation de systèmes intelligents capables de collecter et d'exploiter des données devient une solution intéressante pour mieux comprendre ce qui se passe réellement sur le terrain.

Les progrès réalisés dans le domaine de l'électronique embarquée, des objets connectés et des plateformes numériques ont ouvert de nouvelles possibilités pour l'analyse sportive. Il est aujourd'hui possible d'intégrer des capteurs dans des objets physiques afin de mesurer certains paramètres liés au mouvement, aux impacts ou aux interactions. Dans le cas du football, le ballon représente l'élément central du jeu. Son comportement donne des informations importantes sur les actions réalisées, comme une frappe, une passe, un rebond, un contact ou une variation brusque du mouvement. Ainsi, transformer le ballon en objet connecté peut permettre de collecter des données directement liées au déroulement du jeu.

Dans ce contexte, l'Internet des Objets, ou IoT, joue un rôle important. Il permet de relier un dispositif physique à une plateforme numérique capable de recevoir, organiser, afficher et exploiter les données mesurées. Cette approche permet de passer d'une simple mesure locale à une analyse plus structurée. Les données collectées par le ballon intelligent peuvent être envoyées vers une plateforme IoT, où elles peuvent être visualisées sous forme d'indicateurs, de courbes ou de tableaux de bord. La plateforme devient alors un outil de suivi et d'aide à l'interprétation des actions de jeu.

Cependant, la conception d'un ballon de football intelligent connecté présente plusieurs difficultés. Le système embarqué doit être léger, compact et adapté à l'intérieur du ballon afin de ne pas perturber son comportement normal. Les capteurs doivent être capables de

mesurer des événements rapides comme les impacts et les mouvements brusques. Les données collectées doivent être traitées correctement avant leur transmission vers la plateforme. De plus, la communication entre le ballon et l'environnement numérique doit être suffisamment fiable pour permettre une exploitation en temps réel ou quasi réel.

C'est dans cette perspective que s'inscrit ce travail d'ingéniorat intitulé *« Conception et réalisation d'un ballon de football intelligent connecté associé à une plateforme IoT pour l'analyse du jeu »*. Ce projet vise à proposer une solution complète allant de l'acquisition des données à l'intérieur du ballon jusqu'à leur visualisation sur une plateforme numérique. Le travail ne se limite donc pas à la réalisation d'un dispositif embarqué, mais il inclut également la transmission, l'organisation et l'exploitation des données collectées.

La problématique principale de ce mémoire peut être formulée de la manière suivante : comment concevoir et réaliser un ballon de football intelligent connecté, associé à une plateforme IoT, capable de collecter, transmettre et exploiter des données liées au comportement du ballon afin de contribuer à l'analyse du jeu en temps réel ?

Cette problématique soulève plusieurs questions importantes. Comment intégrer un système électronique dans un ballon sans modifier fortement son comportement ? Comment mesurer les mouvements, les impacts et les changements brusques de dynamique du ballon ? Comment traiter les données collectées afin d'obtenir des informations utiles ? Comment assurer la transmission des données vers une plateforme IoT ? Comment présenter ces données de manière claire et exploitable pour l'utilisateur ? Enfin, comment proposer une solution simple, accessible et adaptée à un contexte académique ou expérimental ?

L'hypothèse principale de ce travail repose sur l'idée qu'un ballon de football équipé d'un système électronique embarqué et connecté à une plateforme IoT peut fournir des informations utiles pour l'analyse du jeu. En mesurant certains paramètres liés au mouvement et aux impacts du ballon, puis en les transmettant vers une plateforme numérique, il devient possible de suivre le comportement du ballon et d'identifier certains événements physiques importants. Cette hypothèse suppose qu'une chaîne complète composée de l'acquisition, du traitement, de la transmission et de la visualisation des données peut améliorer la compréhension des actions de jeu.

L'objectif général de ce mémoire est donc de concevoir et réaliser un prototype de ballon de football intelligent connecté, capable de collecter des données liées à son comportement dynamique et de les transmettre vers une plateforme IoT. Cette plateforme doit permettre d'afficher les données reçues, de suivre leur évolution et de faciliter leur interprétation par l'utilisateur. Le système proposé vise ainsi à relier le terrain à un environnement numérique d'analyse.

De manière plus précise, ce travail poursuit plusieurs objectifs complémentaires :

- étudier les principes des ballons intelligents et des systèmes embarqués appliqués au football .
- concevoir une architecture matérielle adaptée à l'intégration dans un ballon .
- mesurer des paramètres liés aux mouvements, aux impacts et aux interactions du ballon

.

- traiter les données collectées afin d'extraire des informations exploitables.
- assurer la transmission des données vers une plateforme IoT .
- développer une interface permettant la visualisation et le suivi des données .
- tester le fonctionnement du prototype et analyser les résultats obtenus.

Le contenu du titre reflète les axes principaux de ce mémoire. La notion de *ballon de football intelligent connecté* désigne un ballon équipé d'un système électronique capable de mesurer et transmettre des données. L'expression *plateforme IoT* représente l'environnement numérique chargé de recevoir, organiser et afficher ces données. L'expression *analyse du jeu* indique que les mesures collectées ne sont pas destinées uniquement à être affichées sous forme brute, mais à être interprétées afin de mieux comprendre certains comportements du ballon pendant le jeu.

L'intérêt de cette étude réside dans le fait qu'elle propose une solution technologique accessible pour l'analyse du football. Les systèmes professionnels utilisés dans les grandes compétitions sont souvent coûteux, complexes et difficiles à reproduire dans un contexte universitaire ou local. Le développement d'un prototype basé sur un ballon intelligent connecté et une plateforme IoT peut donc représenter une alternative intéressante pour expérimenter l'analyse sportive à partir de données réelles collectées sur le terrain.

Il est important de préciser que le système proposé permet principalement de détecter et d'analyser des événements physiques liés au ballon, comme les impacts, les rebonds, les frappes et les changements brusques de mouvement. Les événements de jeu plus complexes, comme le but, le corner ou la sortie du terrain, nécessitent généralement une information supplémentaire liée à la position du ballon sur le terrain. Dans ce mémoire, ces événements peuvent être considérés comme des objectifs d'interprétation ou comme des perspectives d'amélioration, notamment par l'ajout futur d'un système de localisation ou d'une analyse complémentaire.

Ce mémoire est structuré en trois chapitres complémentaires. Le premier chapitre présente l'état de l'art des ballons instrumentés connectés ainsi que des plateformes IoT appliquées à l'analyse du jeu de football. Le deuxième chapitre est consacré à la conception et à la modélisation du système proposé, en détaillant l'architecture matérielle et logicielle du ballon instrumenté ainsi que les différentes étapes de simulation et de validation préliminaire. Le troisième chapitre présente la réalisation pratique du système, l'intégration des différents composants électroniques dans le ballon, le développement de la communication IoT, la mise en œuvre du tableau de bord de supervision, ainsi que les tests expérimentaux réalisés et l'analyse des résultats obtenus.

Chapitre I

État de l'art sur le football connecté

1.1 Introduction

Le football moderne connaît une évolution importante grâce à l'intégration des technologies numériques, des systèmes embarqués et des objets connectés. Dans ce contexte, l'analyse du jeu ne se limite plus seulement à l'observation humaine, mais s'appuie de plus en plus sur des données mesurables permettant de mieux comprendre les actions réalisées sur le terrain.

Le ballon occupe une place centrale dans le jeu, car la majorité des événements importants sont liés à son mouvement, à ses impacts, à ses rebonds et à ses interactions avec les joueurs ou le terrain. Pour cette raison, l'idée d'un ballon intelligent connecté représente une solution intéressante pour collecter des informations directement depuis l'objet principal du jeu.

Ce chapitre présente l'état de l'art des ballons intelligents connectés et des plateformes IoT utilisées pour l'analyse du football. Il aborde les principes généraux de l'analyse des données sportives, les ballons instrumentés, les capteurs embarqués, le traitement des données, la communication avec une plateforme IoT ainsi que la visualisation des informations. Cette étude permet de mieux situer notre projet et de justifier les choix techniques adoptés dans la suite du mémoire.

1.2 Analyse du jeu de football basée sur les données

L'analyse du jeu de football par les données consiste à collecter, organiser et interpréter des informations liées au déroulement d'un match ou d'une séance d'entraînement. Elle permet de mieux comprendre les actions réalisées, le comportement du ballon, les déplacements des joueurs et les événements qui influencent la dynamique du jeu. Cette approche ne remplace pas l'expertise de l'entraîneur, mais elle l'aide à appuyer son analyse sur des éléments mesurables [1].

Dans le football moderne, les données peuvent provenir de plusieurs sources : systèmes vidéo, capteurs embarqués, dispositifs portés par les joueurs, plateformes numériques ou systèmes de suivi. Elles peuvent concerner les passes, les tirs, les vitesses, les accélérations, les trajectoires, les contacts avec le ballon ou les événements réglementaires. Leur intérêt principal est de transformer une action observée en information exploitable pour l'analyse et la prise de décision [2].

1.2.1 Importance des données dans le football moderne

Les données occupent une place importante dans le football, car elles permettent d'analyser la performance d'une manière plus objective. Dans un match, plusieurs actions se produisent rapidement et parfois simultanément. Il devient donc difficile de tout observer avec précision uniquement à l'œil nu. Les données permettent de conserver une trace des événements et de les étudier pendant ou après l'activité [1].

L'analyse par les données peut aider à répondre à plusieurs questions :

- quelles sont les actions les plus fréquentes pendant le match ?
- quelles zones du terrain sont les plus utilisées ?
- comment le ballon circule pendant le jeu ?
- quels événements modifient le rythme du match ?
- quelles actions peuvent être améliorées à l'entraînement ?

Les systèmes électroniques de suivi de la performance, appelés EPTS, montrent l'importance accordée aux données de position, de vitesse et de mouvement dans le football moderne [6]. Cependant, une donnée brute n'est pas toujours directement utile. Elle doit être interprétée selon le contexte de l'action pour devenir une information compréhensible.

1.2.2 Rôle du ballon dans l'analyse du jeu

Le ballon est l'élément central du football. Les passes, les tirs, les centres, les rebonds, les pertes de balle, les corners, les sorties et les buts sont directement liés à son comportement. L'étude du ballon permet donc d'obtenir des informations importantes sur la dynamique du jeu [2].

L'analyse du ballon peut concerner sa trajectoire, sa vitesse, ses changements de direction, ses contacts avec les joueurs ou avec le terrain, ainsi que les zones dans lesquelles il se déplace. Ces informations permettent de mieux comprendre la construction d'une action, la progression vers le but ou l'apparition d'une situation dangereuse.

Le ballon relie aussi les actions individuelles aux actions collectives. Une passe dépend du joueur qui l'effectue, du joueur qui la reçoit, des adversaires, de la vitesse de l'action et de la trajectoire du ballon. Pour cette raison, les données liées au ballon constituent une base importante pour l'analyse sportive [6].

1.2.3 Événements de jeu liés au ballon

Les événements de jeu liés au ballon représentent les actions ou situations importantes qui apparaissent pendant un match. Ils peuvent être simples, comme une frappe ou un rebond, ou plus complexes, comme un but, un corner ou une sortie du terrain.

Les événements physiques sont directement liés au comportement du ballon. Ils comprennent notamment :

- les frappes .
- les rebonds .
- les impacts .
- les rotations .

- les changements brusques de direction.

Les événements réglementaires dépendent aussi des règles du jeu et de la position du ballon. Par exemple, le ballon est considéré hors du jeu lorsqu'il franchit entièrement la ligne de touche ou la ligne de but [3]. Un but est marqué lorsque le ballon franchit entièrement la ligne de but entre les poteaux et sous la barre transversale, à condition qu'aucune infraction ne soit commise [4]. Un corner est accordé lorsque le ballon franchit entièrement la ligne de but sans but marqué, après avoir été touché en dernier par un joueur de l'équipe défensive [5].

Cette distinction est importante, car les événements physiques peuvent être étudiés à partir du mouvement du ballon, tandis que les événements réglementaires nécessitent aussi une information de position et une interprétation du contexte.

1.2.4 Limites de l'observation humaine

L'observation humaine reste essentielle dans le football, car l'entraîneur possède une expérience et une compréhension tactique que les outils numériques ne remplacent pas. Cependant, cette observation présente certaines limites. Le football est un sport rapide, continu et complexe. Certaines actions sont très courtes, comme le moment exact d'une frappe, un léger changement de trajectoire ou une sortie proche d'une ligne.

Une autre limite est la subjectivité. Deux observateurs peuvent interpréter la même action différemment selon leur angle de vue ou leur expérience. Les données ne suppriment pas totalement cette subjectivité, mais elles apportent un support plus structuré à l'analyse. Elles permettent de comparer les actions, de suivre l'évolution des performances et de mieux comprendre certaines situations [1, 2].

1.3 Étude des solutions de ballon instrumentés du marché

Les ballons instrumentés connectés représentent une évolution importante dans le domaine du football connecté. Contrairement à un ballon classique, un ballon instrumenté contient un dispositif de mesure capable de collecter des données liées au mouvement, aux contacts, aux impacts ou à la position. Lorsqu'il est connecté, ces données peuvent être transmises vers un système externe pour l'analyse, l'arbitrage ou la visualisation [7, 10].

L'intérêt de ces solutions vient du fait que le ballon est directement au cœur du jeu. Les données collectées à partir du ballon permettent d'obtenir des informations proches de l'action réelle et peuvent compléter les systèmes vidéo ou les systèmes de suivi externes [2, 6].

1.3.1 Principe général de l'instrumentation d'un ballon

Un ballon instrumenté connecté peut être défini comme un ballon équipé d'un dispositif de mesure intégré, capable de collecter des données physiques pendant le jeu et de les transmettre vers un système numérique. Le terme *instrumenté* signifie que le ballon contient des moyens de mesure. Le terme *connecté* signifie que les données peuvent être envoyées vers une plateforme, une application, un serveur ou un système d'aide à la décision [7, 15].

Le fonctionnement général repose sur une chaîne simple :

- le ballon subit une action pendant le jeu .
- le dispositif intégré mesure les variations physiques .
- les données sont organisées et horodatées .
- les informations sont transmises vers un système externe .
- la plateforme les affiche ou les interprète.

Dans les systèmes professionnels, ces données sont souvent combinées avec d'autres informations, comme les positions des joueurs ou les images vidéo. Cette combinaison permet d'améliorer l'interprétation des événements, par exemple dans les situations de hors-jeu semi-automatique [8].

1.3.2 Adidas Al Rihla – Coupe du Monde de la FIFA 2022



Figure 1.1 – Ballon Adidas Al Rihla et architecture interne

Source : adapté de [7,10].

Le ballon *Al Rihla* est le ballon officiel de la Coupe du Monde de la FIFA Qatar 2022. Il a été développé par adidas en collaboration avec la FIFA et KINEXON. Son nom, *Al Rihla*, signifie "le voyage" en arabe. Cette appellation symbolise le parcours des équipes durant la compétition et l'esprit du tournoi. Son design s'inspire de la culture, de l'architecture, des bateaux traditionnels et du drapeau du Qatar, ce qui lui donne une identité visuelle liée au pays organisateur [10].

Ce ballon constitue l'un des exemples les plus connus de ballon instrumenté connecté dans le football professionnel. Il intègre une technologie de ballon connecté destinée à transmettre des données de mouvement en temps réel aux officiels vidéo. Cette technologie a été utilisée pour accompagner le système de hors-jeu semi-automatique lors de la compétition [7, 8].

Son principe repose sur une unité de mesure inertielle placée à l'intérieur du ballon. Selon adidas, cette unité fonctionne à une fréquence de 500 Hz, ce qui permet de détecter avec précision le moment où le ballon est touché par un joueur [10].

Les éléments annoncés publiquement sont :

- une unité de mesure inertielle à haute fréquence .
- un système de suspension interne .
- une batterie rechargeable .
- une recharge par induction .
- une transmission vers les officiels vidéo .
- une combinaison avec les données de position des joueurs.

L'usage principal de ce ballon concerne le hors-jeu semi-automatique. Les données du ballon permettent d'identifier le moment du contact, tandis que les caméras du stade fournissent la position des joueurs. Cette combinaison aide les officiels vidéo à prendre des décisions plus rapides et plus précises [8].

1.3.3 Adidas Trionda – Coupe du Monde de la FIFA 2026



Figure 1.2 – Ballon Adidas Trionda et architecture interne

Source : adapté de [9,11].

Le ballon *Trionda*, ballon officiel de la Coupe du Monde de la FIFA 2026, s'inscrit dans la continuité des ballons connectés utilisés au plus haut niveau. La FIFA indique que ce ballon intègre un capteur de mouvement de 500 Hz, capable de transmettre des données précises en temps réel vers le système d'assistance vidéo [9].

Le principe général est proche de celui d'Al Rihla : mesurer le mouvement du ballon à haute fréquence, transmettre les données en temps réel et aider à l'interprétation de certaines situations de jeu. Trionda montre aussi que la conception mécanique reste importante. La FIFA présente notamment une construction à quatre panneaux, des coutures profondes et des éléments de surface destinés à améliorer la stabilité en vol et l'adhérence [9].

Les éléments annoncés sont :

- un capteur de mouvement à 500 Hz .
- une transmission vers le système VAR .
- une construction à quatre panneaux .
- des coutures profondes .
- des reliefs de surface pour améliorer l'adhérence.

Cet exemple montre qu'un ballon connecté ne dépend pas uniquement de l'électronique. La forme du ballon, la répartition interne des éléments et la qualité de la surface influencent aussi sa trajectoire, sa stabilité et son comportement pendant le jeu.

1.3.4 KINEXON xBall



Figure 1.3 – KINEXON xBall et architecture du capteur intégré

Source : adapté de [12, 14].

KINEXON propose une solution appelée *xBall*, destinée au suivi du ballon en temps réel. Cette technologie intègre un capteur dans le ballon et permet de produire des données sur la position, la vitesse, la rotation et le moment du contact [12].

Le fonctionnement de xBall repose sur l'association de deux types d'informations. D'une part, l'IMU permet d'analyser le mouvement, l'accélération et la rotation. D'autre part, l'UWB permet d'estimer la position du ballon grâce à des antennes installées autour de la zone de jeu. KINEXON indique que les données peuvent être transmises jusqu'à 500 Hz vers l'infrastructure de réception [14].

Les éléments principaux d'une telle architecture sont :

- des capteurs intégré dans le ballon, combinant module UWB et IMU .
- une technologie UWB pour la localisation .
- une unité inertielle (IMU) pour le mouvement .
- des antennes autour du terrain .
- un serveur de traitement .
- des interfaces de visualisation.

L'avantage de xBall est de fournir une information plus complète que la simple détection du contact. La position, la vitesse et la rotation du ballon peuvent être exploitées pour l'analyse sportive, la visualisation ou l'aide à l'arbitrage [12].

1.3.5 Comparaison des solutions de ballons instrumentés

Les exemples étudiés montrent que les ballons instrumentés connectés peuvent avoir des objectifs différents. Certains sont principalement utilisés pour l'assistance vidéo à l'arbitrage, tandis que d'autres visent le suivi complet du ballon en temps réel. Le choix de la technologie dépend donc du besoin : détection du contact, analyse du mouvement ou localisation du ballon [7, 12].

Table 1.1 – Comparaison de quelques ballons instrumentés connectés

Exemple	**Technologie principale**	**Données exploitées**	**Utilisation principale**
Al Rihla 2022	IMU 500 Hz	Mouvement Contact Moment de frappe	Assistance au VAR et hors-jeu semi-automatique
Trionda 2026	Capteur de mouvement 500 Hz	Mouvement du ballon Données en temps réel	Assistance au VAR et aide à la décision
KINEXON xBall	UWB + IMU	Position Vitesse Rotation Contact	Suivi en temps réel du ballon et analyse avancée

Cette comparaison montre qu'une unité inertielle est adaptée pour identifier le mouvement et le contact, tandis qu'une technologie de localisation comme l'UWB devient nécessaire lorsque l'objectif est de suivre la position du ballon sur le terrain.

1.4 Revue des plateformes IoT et système d'analyse sportive

Les plateformes IoT et les systèmes d'analyse sportive permettent de recevoir, stocker, traiter et visualiser les données issues des objets connectés. Dans le domaine du football connecté, elles jouent un rôle essentiel, car les données produites par un ballon ou par des capteurs ne sont utiles que si elles sont organisées et présentées de manière claire [15, 16].

1.4.1 Rôle d'une plateforme IoT dans le sport connecté

Une plateforme IoT est un environnement logiciel qui relie des dispositifs physiques à un système numérique. Elle permet généralement la gestion des appareils, la collecte des données, le traitement, le stockage et la visualisation. Selon l'UIT, l'Internet des Objets repose sur l'interconnexion d'objets physiques capables de communiquer avec des systèmes d'information [15].

Dans une application sportive, une plateforme IoT peut afficher des mesures, suivre les événements, générer des alertes et conserver l'historique des données. Elle transforme donc les valeurs brutes en informations plus lisibles pour l'utilisateur.

1.4.2 Plateforme ThingsBoard

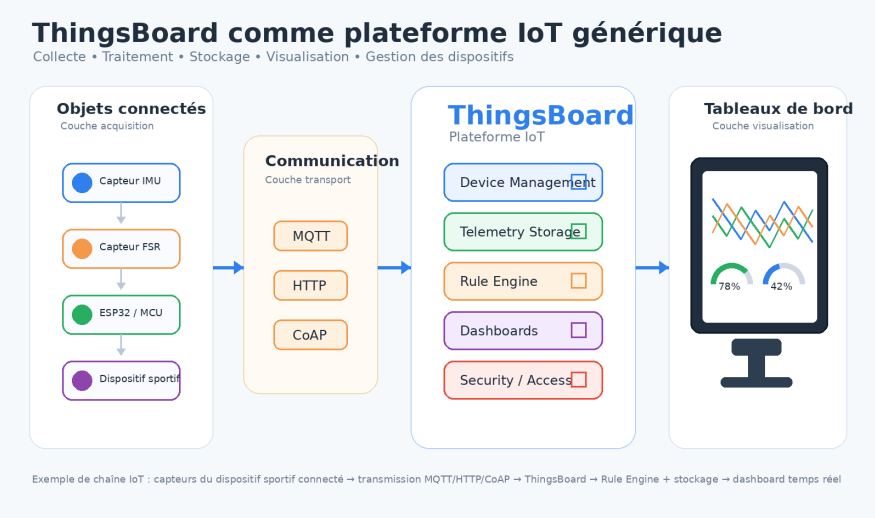


Figure 1.4 – Architecture générale de ThingsBoard

Source : réalisation personnelle adaptée des informations officielles de ThingsBoard [41–43].

ThingsBoard est une plateforme IoT open source utilisée pour la gestion des appareils, la collecte des données, leur traitement et leur visualisation. Elle prend en charge plusieurs protocoles IoT, notamment MQTT, CoAP et HTTP, ce qui la rend adaptée à différents types de systèmes connectés [16].

Dans un contexte sportif, ThingsBoard peut être utilisée pour créer des tableaux de bord affichant les données reçues depuis un dispositif embarqué. Ces tableaux de bord peuvent contenir des courbes, des valeurs numériques, des indicateurs, des cartes ou des alertes. Les tableaux de bord ThingsBoard sont configurables à l'aide de widgets destinés à visualiser les données en temps réel ou sous forme historique [17].

Les avantages principaux de ThingsBoard sont :

- la compatibilité avec plusieurs protocoles IoT .
- la création de tableaux de bord .
- la gestion des appareils connectés .
- la possibilité d'un déploiement local ou cloud .
- l'organisation des données sous forme de télémétrie.

1.4.3 Plateforme sportive KINEXON



Figure 1.5 – la plateforme KINEXON d'analyse des performances

Source : adapté de KINEXON Sports [13], [14].

Les solutions KINEXON ne se limitent pas au capteur placé dans le ballon. Elles reposent aussi sur une infrastructure logicielle capable de recevoir les données issues des capteurs, de les traiter et de les présenter aux utilisateurs. KINEXON présente ses solutions comme des systèmes de suivi en temps réel destinés aux joueurs, aux ballons et aux environnements sportifs [12, 14].

Cette plateforme est associée à une infrastructure de terrain, notamment des antennes UWB, des capteurs embarqués et des outils d'analyse. Elle peut fournir des informations sur la position, la vitesse, la distance, le contact avec le ballon et d'autres indicateurs utiles à l'analyse sportive [14].

1.4.4 Écosystème SPRO/SVIVO (WIMU PRO)



Figure 1.6 – Écosystème WIMU PRO

Source : adapté de [19, 20].

WIMU PRO est un système d'analyse sportive utilisé pour collecter des données liées à la performance des athlètes. Il est généralement associé à des plateformes logicielles permettant l'exploitation des données, comme SPRO ou SVIVO. Ce type de solution montre l'importance d'un écosystème complet associant dispositif matériel, transmission des données et plateforme d'analyse [19, 20].

Même si WIMU PRO est surtout orienté vers l'analyse des joueurs, son architecture est intéressante dans l'état de l'art, car elle montre comment un système sportif connecté peut combiner acquisition de données, communication, traitement logiciel et visualisation.

1.4.5 Comparaison des plateformes étudiées

Les plateformes étudiées n'ont pas toutes le même objectif. ThingsBoard est une plateforme IoT générale, Firebase est une base de données temps réel, KINEXON est une solution

sportive professionnelle, tandis que WIMU PRO est un écosystème orienté performance sportive. Cette diversité montre qu'une plateforme peut être choisie selon le besoin : prototypage rapide, affichage IoT, analyse sportive ou suivi temps réel.

Table 1.2 – Comparaison de quelques plateformes et systèmes d'analyse

Solution	**Fonction principale**	**Intérêt dans le sport connecté**
ThingsBoard	Plateforme IoT	Collecte Stockage Tableaux de bord Visualisation
Plateforme sportive KINEXON	Suivi sportif professionnel	Localisation Suivi du ballon Analyse en temps réel
WIMU PRO	Analyse de performance	Suivi des données sportives Exploitation logicielle

Cette comparaison montre qu'une plateforme IoT adaptée à un projet académique doit être simple, flexible, capable de recevoir les données en temps réel et suffisamment claire pour présenter les résultats sous forme de tableaux de bord.

1.5 Technologies clés des ballons instrumentés et plateformes IoT

Les systèmes de ballon instrumenté connecté reposent sur plusieurs technologies complémentaires. Les capteurs permettent de mesurer le comportement du ballon, le microcontrôleur assure le traitement embarqué, la communication sans fil transmet les données et la plateforme IoT les organise pour l'utilisateur.

1.5.1 Capteurs de mouvement

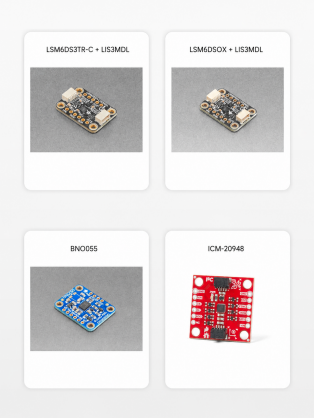


Figure 1.7 – Exemples des modules IMU

Source : adapté de [36–39].

Les capteurs de mouvement permettent de mesurer l'accélération, la rotation et les changements brusques du comportement d'un objet. Les unités de mesure inertielle, appelées IMU, regroupent généralement un accéléromètre et un gyroscope. Elles sont utilisées dans plusieurs systèmes embarqués pour analyser le mouvement d'objets mobiles [22, 77].

Dans le cas d'un ballon, une IMU peut aider à étudier les frappes, les rotations, les rebonds et les changements de direction. Elle ne donne pas directement une position absolue, mais elle fournit des informations importantes sur la dynamique du ballon.

1.5.2 Capteurs d'impact et de contact

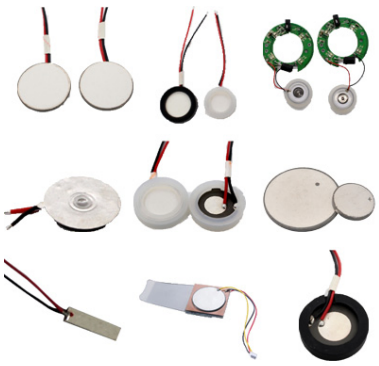


Figure 1.8 – Exemples de capteurs piézoélectriques

Source : adapté de [53].

Les capteurs d'impact et de contact sont utilisés pour détecter les chocs, les vibrations ou les pressions. Les capteurs piézoélectriques sont adaptés aux impacts rapides, car ils produisent une tension lorsqu'ils subissent une déformation mécanique. Ils peuvent donc servir à repérer un choc bref, comme une frappe ou un rebond [53].

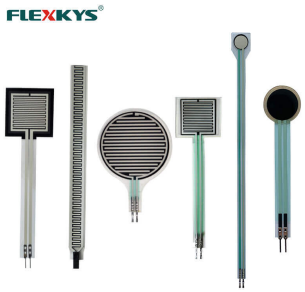


Figure 1.9 – Exemples de capteurs de force sensibles

Source : adapté de [74].

Les capteurs résistifs sensibles à la force, appelés FSR, changent de résistance lorsqu'une force est appliquée sur leur surface. Ils sont utiles pour détecter une pression ou un contact localisé, mais leur mesure reste généralement approximative et dépend de la calibration [52, 74].

1.5.3 Technologies de localisation UWB

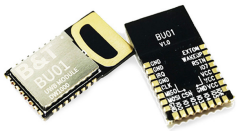


Figure 1.10 – Exemple de module de localisation UWB

Source : adapté de [26].

L'UWB, ou *Ultra-Wideband*, est une technologie radio utilisée pour la localisation à courte portée. Elle est intéressante dans les environnements intérieurs ou contrôlés, car elle ne dépend pas des satellites. Elle fonctionne généralement avec un module mobile et plusieurs balises fixes installées autour de la zone à surveiller [26, 27].

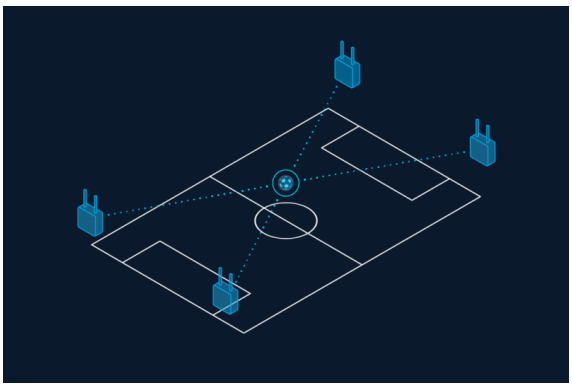


Figure 1.11 – Principe général de localisation UWB

Source : adapté de [14].

Dans un système sportif, l'UWB permet d'estimer la position d'un objet mobile à partir des distances mesurées avec plusieurs balises. Cette information est utile pour les événements liés à la position du ballon, comme une sortie, une zone de but ou une zone de corner [12,14].

1.5.4 Micro-contrôleurs et traitement embarqué

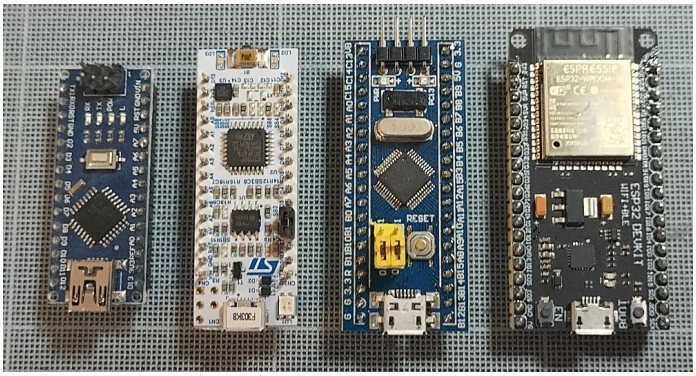


Figure 1.12 – Exemples de cartes à micro-contrôleurs

Source : adapté de [40].

Le micro-contrôleur est l'élément qui relie les capteurs, le traitement local et la communication. Il lit les données, les organise, applique un premier traitement et prépare les informations à transmettre. Dans les systèmes IoT, les micro-contrôleurs intégrant le Wi-Fi ou le Bluetooth sont très utilisés, car ils facilitent la transmission sans fil des données [28, 29].

Le traitement embarqué peut comprendre :

- la lecture des capteurs .

- l'horodatage des mesures .
- le filtrage simple des signaux .
- la détection de seuils .
- la préparation des données pour la transmission.

Cette étape permet de réduire la quantité de données envoyées et de rendre le système plus réactif [57, 61].

1.5.5 Communication sans fil et transmission des données

La communication sans fil permet d'envoyer les données du dispositif embarqué vers une plateforme externe. Le choix de la technologie dépend de la portée, du débit, de la consommation énergétique et de l'environnement d'utilisation.

Dans les applications IoT, le protocole MQTT est très utilisé. Il repose sur un modèle publication/abonnement et permet à un dispositif de publier des données vers un serveur appelé *broker*. Ce protocole est léger et adapté aux systèmes connectés ayant des ressources limitées [31, 32].

Les données envoyées peuvent contenir :

- l'identifiant du dispositif .
- l'heure ou le temps de mesure .
- les valeurs mesurées .
- l'événement détecté .
- l'état du système.

Une structure claire des données facilite leur stockage, leur affichage et leur interprétation dans la plateforme IoT.

1.6 Synthèse comparative et limites des solutions existantes

Les solutions étudiées montrent que le football connecté repose sur une combinaison de plusieurs éléments : ballon instrumenté, capteurs, communication, localisation et plateforme de visualisation. Chaque élément joue un rôle différent dans la chaîne d'analyse.

1.6.1 analyse de l'intégration des solutions existantes

Les ballons instrumentés permettent de collecter les données proches de l'action réelle, tandis que les plateformes IoT permettent de les organiser et de les présenter. Un ballon sans plateforme produit des mesures difficiles à exploiter, et une plateforme sans dispositif de mesure ne peut pas recevoir de données réelles. Les deux parties sont donc complémentaires [7, 16].

1.6.2 Limites techniques

Les principales limites techniques concernent l'intégration des composants, la précision des mesures, l'autonomie, la robustesse et la localisation. Un système placé dans un ballon doit être compact, léger et bien équilibré. Les chocs répétés peuvent aussi perturber les mesures et nécessitent un filtrage adapté [59, 60].

La détection des événements complexes représente également une difficulté. Une frappe ou un rebond peut être détecté à partir du mouvement, mais un but, un corner ou une sortie du terrain nécessitent une information de position et une interprétation des règles du jeu [3–5].

1.6.3 Limites économiques et d'accessibilité

Les solutions professionnelles comme les ballons connectés utilisés dans les compétitions internationales ou les systèmes KINEXON sont performantes, mais elles restent souvent coûteuses et propriétaires. Elles nécessitent parfois des caméras, des antennes, des logiciels spécialisés et une infrastructure complète autour du terrain [7, 12].

Cette situation limite leur utilisation dans les clubs locaux, les établissements universitaires ou les projets pédagogiques. Il existe donc un intérêt à étudier des solutions plus accessibles, même si elles sont moins précises que les systèmes professionnels.

1.6.4 Besoins identifiés pour une solution académique

Une solution académique doit chercher un équilibre entre simplicité, coût, précision et faisabilité. Elle doit permettre de comprendre le principe du ballon instrumenté connecté sans nécessiter une infrastructure trop lourde. Elle doit aussi être suffisamment claire pour que les données collectées puissent être affichées et interprétées.

Les besoins principaux sont :

- une acquisition simple des données du ballon .
- une transmission vers une plateforme numérique .
- une visualisation claire des mesures .
- une architecture évolutive .

- une possibilité d'amélioration future avec la localisation ou d'autres capteurs.

1.7 Conclusion

Ce chapitre a présenté l'état de l'art des technologies liées aux ballons instrumentés connectés et aux plateformes IoT appliquées à l'analyse du football. L'analyse sportive par les données permet de mieux comprendre les actions du jeu et de compléter l'observation humaine par des mesures objectives.

Les exemples étudiés, comme Al Rihla, Trionda et KINEXON xBall, montrent que le ballon peut devenir une source de données importante dans les systèmes modernes d'analyse et d'aide à la décision. Les plateformes comme ThingsBoard, KINEXON Sports ou WIMU PRO montrent également l'importance de l'environnement logiciel dans la collecte, l'organisation et la visualisation des données.

L'étude des technologies utilisées montre que les capteurs, les micro-contrôleurs, la communication sans fil et la localisation UWB sont des éléments essentiels dans ce type de système. Cependant, les solutions professionnelles restent souvent coûteuses et complexes. Cela justifie l'intérêt d'une solution académique plus accessible, centrée sur la mesure, la transmission et la visualisation des données du ballon.

Chapitre II

Modélisation théorique du ballon est simulation du circuit éléctronique

2.1 Introduction

Dans ce chapitre, nous présentons la modélisation globale d'un ballon de football intelligent, en intégrant à la fois l'étude physique du mouvement, l'architecture électronique du système embarqué ainsi que la simulation du circuit électronique. Cette approche permet de décrire le fonctionnement général du système depuis l'action mécanique appliquée sur le ballon jusqu'à l'acquisition, le traitement et l'exploitation des données mesurées.

L'objectif de ce chapitre est donc de comprendre comment les différentes parties du système interagissent entre elles. D'une part, la modélisation physique permet d'étudier le comportement mécanique du ballon lorsqu'il est soumis à une force extérieure, par exemple lors d'un tir, d'une passe ou d'un contact avec le joueur. D'autre part, la modélisation électronique permet de représenter le rôle des capteurs, du micro-contrôleur ESP32 et de la transmission des données. Enfin, la simulation du circuit permet de vérifier virtuellement le fonctionnement de l'architecture proposée avant une éventuelle réalisation pratique.

Le principe général du projet repose sur la chaîne suivante :

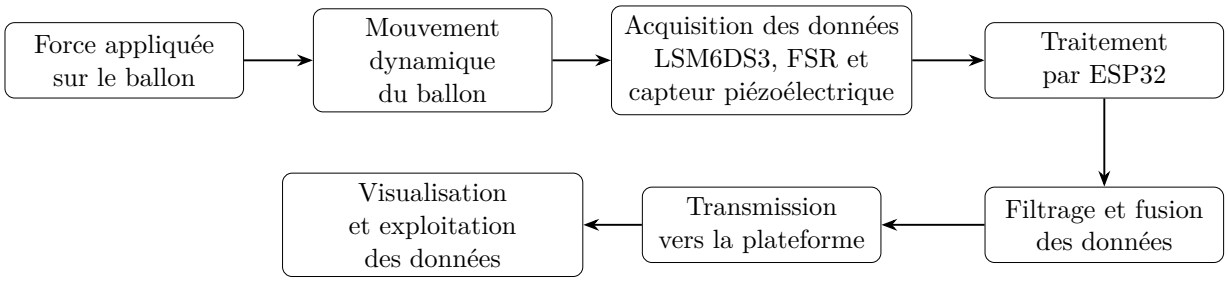


Figure 2.1 – Chaîne fonctionnelle du système intelligent de suivi et d'analyse du ballon

Lorsqu'un joueur applique une force sur le ballon, cette action provoque un mouvement qui peut être analysé à l'aide de capteurs intégrés. Les capteurs FSR permettent d'estimer les forces appliquées lors des impacts, tandis que le capteur inertiel LSM6DS3 permet de mesurer les paramètres liés au mouvement, notamment l'accélération et la rotation du ballon. Les données mesurées sont ensuite traitées par le micro-contrôleur ESP32, qui joue le rôle d'unité de traitement embarquée.

La simulation du circuit électronique constitue une étape importante dans cette étude. Elle permet de représenter les connexions entre les capteurs, le micro-contrôleur et les différents éléments du système. Elle permet également de vérifier la cohérence du câblage, le principe d'acquisition des signaux et le fonctionnement général du système avant de passer à une réalisation matérielle réelle. Cette étape réduit les risques d'erreurs techniques et facilite la compréhension de l'architecture proposée.

Pour organiser cette étude, le chapitre est divisé en trois parties principales. La première partie concerne la modélisation physique du ballon, afin d'étudier son comportement mécanique sous l'effet d'une force appliquée. La deuxième partie présente la modélisation électronique du système embarqué, en décrivant le rôle des capteurs, du micro-contrôleur

ESP32, du traitement des données et de leur transfert. La troisième partie est consacrée à la simulation du circuit électronique, qui permet de valider le fonctionnement global du système dans un environnement virtuel.

2.2 Modélisation physique du ballon

2.2.1 Analyse des forces appliquées sur le ballon

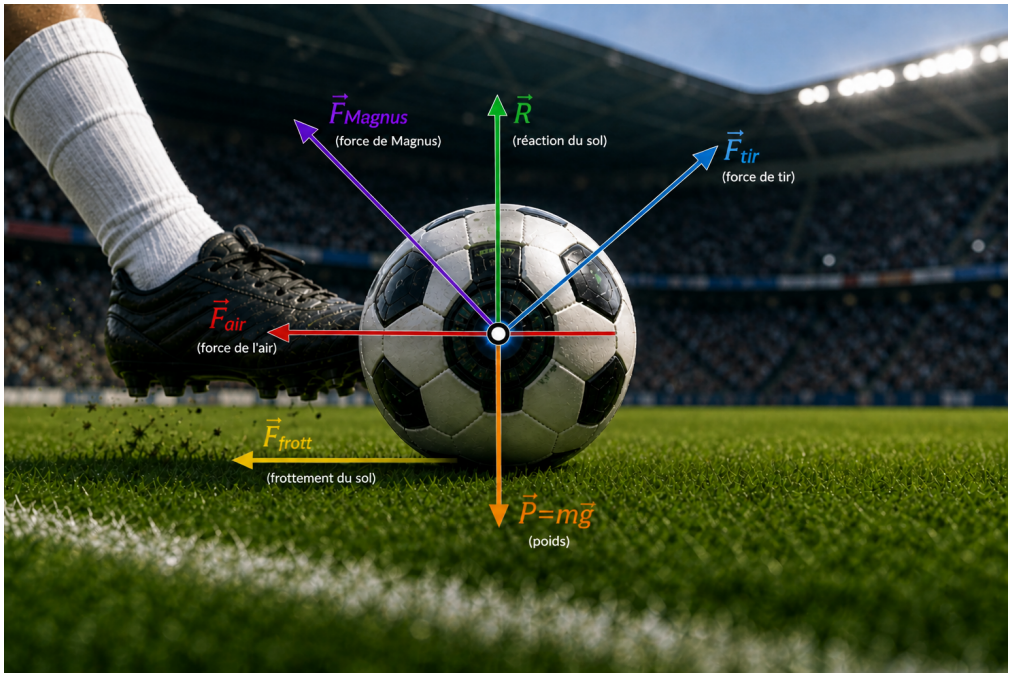


Figure 2.2 – Représentation des forces appliquées sur le ballon

La figure précédente représente les différentes forces appliquées sur le ballon de football lors d'un tir. Ce schéma permet de visualiser les actions mécaniques qui influencent le mouvement du ballon et sa trajectoire. Dans cette étude, le ballon est considéré comme un solide soumis à plusieurs forces extérieures, appliquées soit au centre de masse, soit au point de contact avec le sol ou avec le pied du joueur [44].

a) Le poids $\vec{P}$

Le poids est la force gravitationnelle exercée par la Terre sur le ballon. Il est appliqué au centre de masse du ballon et il est toujours dirigé verticalement vers le bas [44].
Son expression est :

$$\vec{P} = m\vec{g} \tag{2.1}$$

où :
m : représente la masse du ballon, exprimée en kilogrammes (kg).

$\vec{g}$: représente le vecteur de l'accélération de la pesanteur, dont la valeur est environ 9,81 m/s^2.

b) La réaction du sol $\vec{R}$

La réaction du sol est une force de contact exercée par le terrain sur le ballon. Elle apparaît uniquement lorsque le ballon touche le sol. Elle est appliquée au point de contact entre le ballon et la pelouse, et elle est dirigée verticalement vers le haut si le terrain est horizontal [44].

Dans une simulation plus réaliste, la réaction du sol peut être modélisée par un modèle ressort-amortisseur :

$$R_z = k_s\delta - c_s v_z \tag{2.2}$$

avec :

$$\delta = \max(0, R_{\text{ballon}} - z) \tag{2.3}$$

où k_s est la raideur du contact, c_s le coefficient d'amortissement, R_{ballon} le rayon du ballon et z la hauteur du centre du ballon. Ce type de modèle est souvent utilisé pour représenter le contact entre deux corps dans les simulations dynamiques [45, 46].

c) La force de tir $\vec{F}_{tir}$

La force de tir est la force exercée par le pied du joueur sur le ballon. Elle est appliquée au point de contact entre le pied et le ballon. C'est cette force qui met le ballon en mouvement et qui détermine principalement sa vitesse initiale, sa direction et son angle de lancement [44,47].

d) La force de l'air $\vec{F}_{air}$

La force de l'air, appelée aussi force de traînée, est une force aérodynamique qui s'oppose au mouvement du ballon. Elle agit dans le sens opposé à la vitesse. Elle devient importante lorsque la vitesse du ballon est élevée. Les travaux sur l'aérodynamique du ballon de football montrent que la traînée dépend fortement de la vitesse, de la surface du ballon et du régime d'écoulement de l'air autour du ballon [48, 49].

Elle est généralement modélisée par :

$$\vec{F}_{air} = -\frac{1}{2}\rho C_d A \|\vec{v}\|\vec{v} \tag{2.4}$$

Avec :

$$A = \pi R^2 \tag{2.5}$$

où :

- ρ est la masse volumique de l'air (kg/m^3), dont la valeur est environ 1,225 kg/m^3 ;

- C_d est le coefficient de traînée. Il représente la résistance aérodynamique du ballon lorsqu'il se déplace dans l'air. Pour un ballon de football, on peut prendre généralement $C_d = 0{,}47$;
- A est la surface frontale du ballon ;
- R est le rayon du ballon ;
- $\|\vec{v}\|$ est la norme de la vitesse ;
- $\vec{v}$ est le vecteur vitesse du ballon.

Cette expression est couramment utilisée pour modéliser la force de traînée appliquée à un ballon en mouvement dans l'air [47, 49].

e) La force de frottement $\vec{F}_{frott}$

La force de frottement apparaît lorsque le ballon est en contact avec le terrain. Elle est tangentielle au sol et s'oppose au mouvement relatif entre le ballon et la pelouse. Cette force influence la diminution de la vitesse lorsque le ballon roule ou glisse sur le terrain [44].

Son expression est :

$$\vec{F}_{frottement} = -\mu R_N \vec{u}_v \tag{2.6}$$

où :
R_N est la force normale égale à la force de réaction R.
μ est le coefficient de frottement.
$\vec{u}_v$ est le vecteur unitaire dans la direction du mouvement horizontal.

Avec :

$$\vec{u}_v = \frac{1}{\sqrt{\dot{x}^2 + \dot{y}^2}} \begin{bmatrix} \dot{x} \\ \dot{y} \\ 0 \end{bmatrix} \tag{2.7}$$

f) La force de Magnus $\vec{F}_{Magnus}$

La force de Magnus apparaît lorsque le ballon se déplace dans l'air tout en tournant autour de son axe. Elle est due à l'interaction entre la rotation du ballon et l'écoulement de l'air. Cette force est responsable des trajectoires courbées observées dans les tirs avec effet, par exemple les coups francs. Les modèles de trajectoire de ballon de football intègrent souvent la traînée et la portance liée à l'effet Magnus pour simuler correctement le vol du ballon [47, 48].

Son expression est :

$$\vec{F}_{Magnus} = \alpha\,(\vec{\omega} \times \vec{v}) \tag{2.8}$$

où :

- α est un coefficient lié à l'air et aux caractéristiques du ballon, il est approximé par :

$$\alpha = \frac{1}{2}\rho R^3$$

- $\vec{\omega}$ est la vitesse angulaire du ballon ;
- $\vec{v}$ est la vitesse linéaire du ballon.

Cette force dépend de la vitesse linéaire du ballon, de sa vitesse angulaire et des propriétés de l'écoulement de l'air autour du ballon [47, 50].

2.2.2 Équations du mouvement du ballon

La dynamique du ballon est basée sur la deuxième loi de Newton :

$$\sum \vec{F} = m\vec{a} \tag{2.9}$$

Avec :

$$\vec{a} = \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} \tag{2.10}$$

la somme des forces appliquées au ballon peut être exprimée par :

$$\vec{F}_{tot} = \vec{P} + \vec{R} + \vec{F}_{frott} + \vec{F}_{air} + \vec{F}_{Magnus} + \vec{F}_{tir} \tag{2.11}$$

Selon les trois axes, elles deviennent :

$$\begin{cases} m\ddot{x} = F_{frottement,x} + F_{air,x} + F_{Magnus,x} + F_{tir,x} \\ m\ddot{y} = F_{frottement,y} + F_{air,y} + F_{Magnus,y} + F_{tir,y} \\ m\ddot{z} = P + R + F_{air,z} + F_{Magnus,z} + F_{tir,z} \end{cases} \tag{2.12}$$

En remplaçant chaque force par son expression, les équations du mouvement du ballon s'écrivent :

$$\begin{cases} m\ddot{x} = -\mu R_N \dfrac{\dot{x}}{\sqrt{\dot{x}^2 + \dot{y}^2}} - \dfrac{1}{2}\rho C_d A \|\vec{v}\| \dot{x} + \alpha_M(\omega_y \dot{z} - \omega_z \dot{y}) + F_{tir}\cos(\theta)\cos(\beta) \\ m\ddot{y} = -\mu R_N \dfrac{\dot{y}}{\sqrt{\dot{x}^2 + \dot{y}^2}} - \dfrac{1}{2}\rho C_d A \|\vec{v}\| \dot{y} + \alpha_M(\omega_z \dot{x} - \omega_x \dot{z}) + F_{tir}\cos(\theta)\sin(\beta) \\ m\ddot{z} = -mg + R_N - \dfrac{1}{2}\rho C_d A \|\vec{v}\| \dot{z} + \alpha_M(\omega_x \dot{y} - \omega_y \dot{x}) + F_{tir}\sin(\theta) \end{cases} \tag{2.13}$$

2.2.3 Résolution numérique et analyse des trajectoire

La fonction ode45 est utilisée pour résoudre le système d'équations différentielles obtenu. Les résultats de modélisation sont ensuite représentés sous forme de courbes, ce qui permet d'analyser l'influence des forces telles que le poids, la force de tir, la résistance de l'air, le frottement, la force de contact et la force de Magnus sur le déplacement du ballon.

2.2.3.1 Définition des paramètres de simulation

Table 2.1 – Paramètres physiques utilisés dans la modélisation du mouvement du ballon

Symbole / Variable	**Description**	**Valeur**	**Unité**
m	Masse du ballon	0.45	kg
g	Accélération de la pesanteur	9.81	m/s^2
R	Rayon du ballon	0.1115	m
μ	Coefficient de frottement sol-ballon	0.5	sans unité
ρ	Masse volumique de l'air	1.225	kg/m^3
C_d	Coefficient de traînée	0.47	sans unité
$A = \pi R^2$	Surface frontale du ballon	0.039	m^2
$\alpha_M = \frac{\rho R^3}{2}$	Coefficient de Magnus	0.000761	kg/m
ω_x	Vitesse angulaire selon x	0	rad/s
ω_y	Vitesse angulaire selon y	0	rad/s
ω_z	Vitesse angulaire selon z	200	rad/s
F_{tir0}	Force maximale du tir	80	N
θ	Angle vertical du tir	20	degré
β	Angle horizontal du tir	0	degré
$k_{contact}$	Raideur du contact avec le sol	10^4	N/m
$c_{contact}$	Amortissement du contact avec le sol	50	N.s/m
t_0	Temps initial de simulation	0	s
t_f	Temps final de simulation	5	s

Table 2.2 – Conditions initiales utilisées dans la simulation du mouvement du ballon

Variable	Description	Valeur initiale	Unité
x_0	Position initiale selon x	0	m
y_0	Position initiale selon y	0	m
z_0	Position initiale selon z	0.11	m
v_{x0}	Vitesse initiale selon x	0	m/s
v_{y0}	Vitesse initiale selon y	0	m/s
v_{z0}	Vitesse initiale selon z	0	m/s

La trajectoire tridimensionnelle obtenue du ballon est présentée dans la figure suivante :

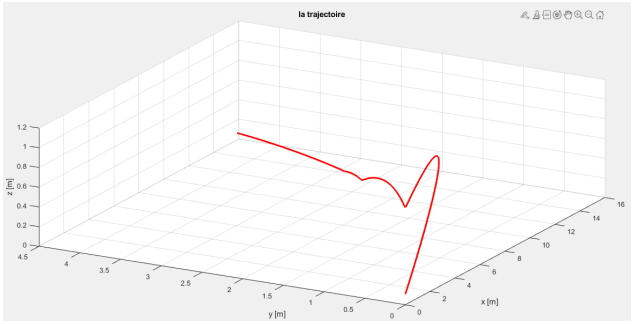
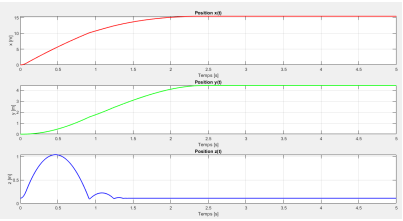


Figure 2.3 – La trajectoire 3D du ballon

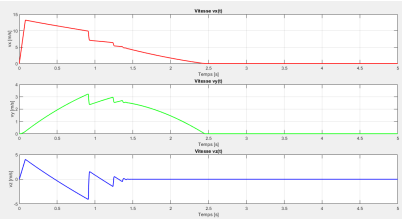
Cette trajectoire montre que le ballon avance dans l'espace après l'application de la force de tir. Son altitude varie sous l'effet de la gravité et de la composante verticale du tir. On remarque aussi une légère déviation latérale, ce qui indique que le mouvement n'est pas parfaitement rectiligne. Cette déviation peut être expliquée par l'effet combiné de la résistance de l'air, du contact avec le sol et de l'effet Magnus. Ainsi, la trajectoire obtenue représente le mouvement global du ballon en tenant compte des différentes forces physiques appliquées pendant la modélisation.

2.2.3.2 Analyse des résultats

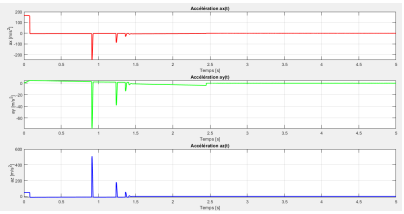
a) Évolution de la position, de la vitesse et de l'accélération en fonction du temps :



(a) Position du ballon



(b) Vitesse du ballon



(c) Accélération du ballon

Figure 2.4 – Évolution de la position, de la vitesse et de l'accélération du ballon en fonction du temps

Source : réalisé sous MATLAB.

Les courbes obtenues montrent l'évolution du mouvement du ballon selon les trois axes x, y et z. Elles confirment la relation fondamentale entre les grandeurs cinématiques : la vitesse représente la variation de la position au cours du temps, tandis que l'accélération représente la variation de la vitesse. Ainsi, lorsque la vitesse est élevée, la position varie rapidement ; et lorsque l'accélération présente des pics, cela traduit un changement brusque de la vitesse(Bray et Kerwin, 2003 ; Goff et Carré, 2009).

La courbe de position montre que le ballon se déplace principalement selon l'axe x, qui représente la direction dominante du tir. La position $x(t)$ augmente rapidement au début, puis sa pente diminue progressivement jusqu'à devenir presque constante. Cela signifie que le ballon avance d'abord avec une vitesse importante, puis ralentit sous l'effet des forces résistantes, notamment la résistance de l'air et le frottement avec le terrain.

Selon l'axe y, on observe une variation latérale de la position. Cette déviation peut être expliquée par la composante latérale du tir, mais aussi par l'effet Magnus. En effet, lorsque le ballon possède une vitesse de rotation, la force de Magnus agit perpendiculairement à la vitesse du ballon et peut modifier sa trajectoire. Elle ne s'oppose pas directement au mouvement comme la force de l'air ; elle provoque plutôt une courbure de la trajectoire. Ce phénomène est particulièrement visible dans les tirs avec effet, comme les coups francs courbés.

La position verticale $z(t)$ montre la phase de montée et de descente du ballon. Au début, le ballon prend de la hauteur grâce à la composante verticale de la force de tir. Ensuite, sous l'effet du poids, il redescend vers le sol. Les petites oscillations observées après le premier contact correspondent aux rebonds successifs. L'effet Magnus peut également influencer cette direction verticale si l'axe de rotation du ballon produit une force orientée vers le haut ou vers le bas.

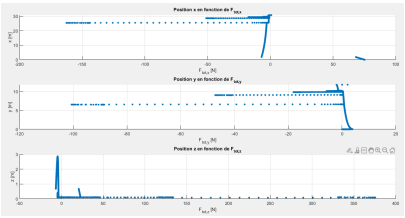
Les courbes de vitesse confirment cette évolution. Une vitesse positive indique un déplacement dans le sens positif de l'axe étudié, tandis qu'une vitesse négative indique un déplacement dans le sens opposé. Ainsi, lorsque v_z est positive, le ballon monte ; lorsqu'elle devient négative, le ballon descend. La diminution progressive des vitesses horizontales montre que le ballon perd de l'énergie à cause des forces résistantes.

Les courbes d'accélération montrent les variations rapides de la vitesse. Au début du

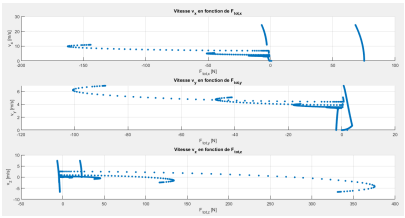
mouvement, l'accélération est élevée à cause de la force de tir. Ensuite, les pics d'accélération, surtout selon l'axe z, correspondent aux contacts entre le ballon et le sol. Ces pics traduisent l'action importante de la réaction normale du terrain pendant les impacts. Le signe de l'accélération indique son orientation : si elle est de même signe que la vitesse, le mouvement s'intensifie ; si elle est de signe opposé, le ballon ralentit selon l'axe considéré.

Ainsi, les trois courbes montrent un comportement cohérent du ballon : la force de tir initialise le mouvement, la vitesse fait évoluer la position, l'accélération traduit les variations de vitesse, et les forces aérodynamiques comme la traînée et la force de Magnus influencent la forme de la trajectoire. En particulier, l'effet Magnus explique la déviation du ballon lorsque celui-ci est en rotation.

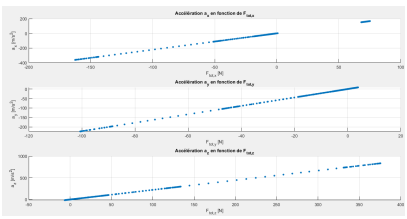
b) Évolution de la position, de la vitesse et de l'accélération en fonction des forces appliquées :



(a) Position en fonction de la force totale



(b) Vitesse en fonction de la force totale



(c) Accélération en fonction de la force totale

Figure 2.5 – Évolution de la position, de la vitesse et de l'accélération du ballon en fonction des forces appliquées

Source : réalisé sous MATLAB.

Les figures représentent l'évolution de l'accélération, de la vitesse et de la position du ballon en fonction des composantes de la force totale appliquée selon les trois axes. Les courbes obtenues sont tracées sous forme de points, car chaque point correspond à un instant précis de la simulation.

l'accélération varie de manière régulière avec la force totale. On remarque que les points sont presque alignés, ce qui montre que l'accélération est directement influencée par la force appliquée. Les valeurs les plus importantes apparaissent surtout selon l'axe vertical, ce qui peut être expliqué par les phases de contact avec le sol, notamment lors de l'impact ou du rebond du ballon.

la vitesse présente une forme plus dispersée. Les points ne sont pas parfaitement alignés, mais forment plusieurs branches. Cela signifie que la vitesse du ballon change selon les différentes phases du mouvement : le tir initial, le déplacement dans l'air, la diminution de la vitesse sous l'effet des résistances, puis le contact avec le sol. Cette dispersion est normale, car la vitesse ne dépend pas uniquement de la force à un instant donné, mais aussi de l'état précédent du ballon.

la position apparaît également sous forme de nuage de points. Cela montre que la position du ballon n'est pas directement liée à une seule valeur de force. Le ballon peut se trouver à différentes positions pour des valeurs de force proches, selon le moment de la simulation. La position verticale permet particulièrement de distinguer les phases où le ballon est en l'air et celles où il est proche du sol. Les fortes valeurs de force verticale correspondent principalement aux moments où le ballon entre en contact avec le terrain.

2.3 Conception et modélisation électronique du système

2.3.1 Architecture électronique globale du système embarqué

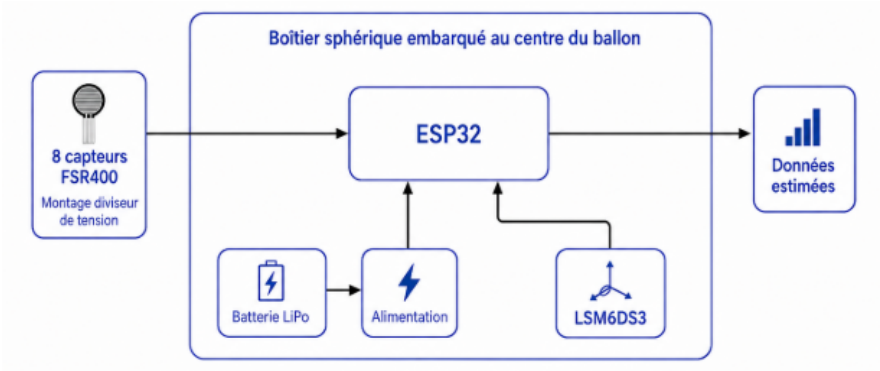


Figure 2.6 – Schéma bloc simplifié du système électronique intégré dans le ballon

a) Capteurs de force FSR402

Pour mesurer la force de tir lors de l'impact, nous avons utilisé 8 capteurs FSR402 disposés autour du ballon [52, 74].
Le montage est en diviseur de tension, transformant la résistance variable en tension mesurable par l'ESP32 [52].

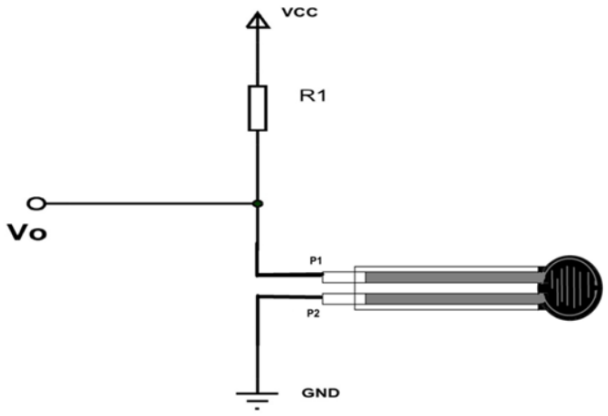


Figure 2.7 – Schéma de fonctionnement du capteur FSR402

Pourquoi FSR402 plutôt qu'un autre capteur de force ?

- Souplesse et légèreté, adaptée à la membrane du ballon [52, 74].
- Montage simple et robuste, facile à intégrer dans le prototype [52].
- Meilleur rapport coût et performance comparé aux capteurs piézoélectriques ou capacitifs [52, 53].

b) Capteur inertiel LSM6DS3

Le LSM6DS3 est un module MEMS combinant un accéléromètre et un gyroscope 3 axes, pour un total de 6 axes. Il fournit les mesures nécessaires pour calculer l'orientation, la vitesse et l'accélération du ballon [77].

Pourquoi LSM6DS3 ?

- Compacité et faible encombrement, idéal pour un ballon sphérique [77].
- Consommation énergétique réduite, importante pour maximiser l'autonomie de la batterie [77].
- Bonne précision et stabilité pour les mouvements rapides, contrairement aux IMU basiques ou aux capteurs seuls (seul le gyroscope ne suffirait pas à mesurer la translation) [77].
- Communication I2C simple, permettant une intégration facile avec l'ESP32 [69, 77].

c) Microcontrôleur ESP32

L'ESP32 est un microcontrôleur intégré puissant qui collecte, traite et transmet les données [69].

Pourquoi ESP32 ?

- Comparé à un Arduino classique, l'ESP32 offre plus de puissance de calcul et une fréquence d'échantillonnage plus élevée, essentielle pour suivre des impacts rapides et les accélérations [69].
- Intègre Wi-Fi et Bluetooth, ce qui permet une transmission directe vers un ordinateur ou une application mobile, sans modules additionnels [56, 69].
- Consommation énergétique faible et coût abordable, ce qui en fait un choix optimal pour un dispositif embarqué [69].

d) Batterie LiPo 3,7 V 1000 mAh

Une batterie LiPo fournit une alimentation compacte et légère [58].

Pourquoi LiPo 3,7 V 1000 mAh ?

- La tension de 3,7 V est compatible avec les capteurs et l'ESP32 [58, 69].
- Capacité 1000 mAh pour garantir une autonomie suffisante pour plusieurs heures d'utilisation [58].
- Léger et flexible, adapté à un boîtier sphérique sans modifier la dynamique du ballon [58].

2.3.2 Modélisation et conditionnement des capteurs FSR402

Un capteur FSR402 possède une résistance variable. Lorsque la force appliquée augmente, sa résistance diminue. Pour rendre cette variation exploitable par l'ESP32, le capteur est monté en diviseur de tension [52, 74].

Le montage peut être modélisé par l'équation suivante :

$$V_{\mathrm{o}} = V_{\mathrm{cc}} \times \frac{R_{\mathrm{FSR}}}{R_1 + R_{\mathrm{FSR}}} \tag{2.14}$$

Avec :

- V_{o} est la tension mesurée par l'ESP32.
- V_{cc} est la tension d'alimentation.
- R_{FSR} est la résistance variable du capteur FSR.
- R_1 est la résistance fixe du diviseur de tension.

La force appliquée sur un capteur peut être estimée par une relation expérimentale obtenue après calibration [52] :

$$F_i = f(V_i) \tag{2.15}$$

Avec :

- F_i représente la force estimée par le capteur numéro i.
- V_i représente la tension mesurée aux bornes du capteur numéro i.

la force totale appliquée peut être estimée par la somme des forces mesurées par chaque capteur :

$$F_{\text{totale}} = \sum_{i=1}^{8} F_i \tag{2.16}$$

2.3.2.1 Conversion analogique-numérique par l'ESP32

Les tensions issues des capteurs FSR sont des signaux analogiques. L'ESP32 ne traite pas directement une force, mais une valeur numérique issue de son convertisseur analogique-numérique ADC [57, 69].

Pour un ADC de résolution N bits, la valeur numérique obtenue est donnée par :

$$ADC_i = \frac{V_i}{V_{\text{ref}}} \times \left(2^N - 1\right) \tag{2.17}$$

Dans le cas d'un ADC 12 bits, on obtient :

$$ADC_i = \frac{V_i}{V_{\text{ref}}} \times 4095 \tag{2.18}$$

Avec :

- ADC_i est la valeur numérique lue par l'ESP32.
- V_i est la tension d'entrée (la tension mesurée aux bornes du capteur i).
- V_{ref} est la tension de référence de l'ADC généralement $3.3\,V$.

La tension peut donc être retrouvée à partir de la valeur ADC :

$$V_i = \frac{ADC_i}{4095} \times V_{\text{ref}} \tag{2.19}$$

Cette étape permet de transformer le signal électrique analogique en donnée numérique exploitable par le programme embarqué [57].

2.3.2.2 Localisation approximative du point d'impact

L'utilisation de plusieurs capteurs FSR402 permet d'estimer approximativement la zone où l'impact a eu lieu sur le ballon. Dans notre prototype, chaque capteur est placé dans une position connue autour du ballon [52, 74].

On note r_i la position du capteur numéro i, et F_i la force estimée par ce même capteur.

Après l'estimation des forces appliquées sur les différents capteurs, le système effectue une comparaison entre les valeurs obtenues :

$$F_1, F_2, F_3, \ldots, F_8$$

Le capteur qui mesure la force la plus élevée est considéré comme le capteur le plus proche de la zone d'impact. On détermine alors l'indice du capteur correspondant par la relation suivante :

$$i_{\max} = \arg \max_{i \in \{1,\ldots,8\}} F_i \tag{2.20}$$

La position approximative du point d'impact est alors donnée par :

$$r_{\text{impact}} \approx r_{i_{\max}} \tag{2.21}$$

où :

- $i_{\max}$ représente le numéro du capteur qui mesure la force maximale .
- $r_{i_{\max}}$ représente la position du capteur le plus sollicité .
- r_{impact} représente la position approximative de l'impact.

2.3.3 Modélisation du capteur inertiel LSM6DS3 :

La sortie du capteur peut représenter par le vecteur suivant [77] :

$$X_{\text{IMU}} = \begin{bmatrix} a_x \\ a_y \\ a_z \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \tag{2.22}$$

Avec :

- a_x, a_y, a_z représentent les accélérations linéaires selon les axes x, y et z.
- $\omega_x, \omega_y, \omega_z$ représentent les vitesses angulaires selon les mêmes axes.

La norme de l'accélération est calculée par :

$$a_{\text{norme}} = \sqrt{a_x^2 + a_y^2 + a_z^2} \tag{2.23}$$

Cette grandeur est utilisée pour détecter les changements brusques de mouvement, comme un tir, un rebond ou une collision.

De la même manière, la norme de la vitesse angulaire est donnée par :

$$\omega_{\text{norme}} = \sqrt{\omega_x^2 + \omega_y^2 + \omega_z^2} \tag{2.24}$$

Elle permet d'évaluer la rotation globale du ballon.

2.3.4 Filtrage des données mesurées :

Les signaux mesurés par les capteurs contiennent souvent du bruit. Pour améliorer la qualité des données, il est nécessaire d'appliquer un filtrage numérique [59, 60].

Un filtre simple peut être utilisé sous la forme suivante :

$$y_k = \alpha x_k + (1 - \alpha) y_{k-1} \tag{2.25}$$

Avec :

- y_k est la mesure filtrée.
- y_{k-1} est la mesure filtrée précédente.
- x_k est la nouvelle mesure brute.
- α est un coefficient compris entre 0 et 1.

Si α est grand, le filtre réagit rapidement mais laisse passer plus de bruit. Si α est petit, le signal devient plus stable mais plus lent.

Ce filtrage est appliqué aux mesures des FSR et du LSM6DS3 afin d'obtenir des données plus fiables.

2.3.5 Algorithme de traitement des données embarqué :

L'ESP32 exécute un programme qui permet de lire, filtrer, traiter et transmettre les données [57, 61].

Le fonctionnement peut être résumé par l'algorithme suivant :

Algorithme 1 : Algorithme de traitement des données embarqué sur l'ESP32

Initialiser les capteurs FSR402;
Initialiser le capteur LSM6DS3;
Initialiser la communication Wi-Fi ou Bluetooth;
while *le système est actif* **do**
  Lire les valeurs analogiques des 8 capteurs FSR402;
  Convertir les valeurs ADC en tensions;
  Estimer les forces appliquées sur chaque capteur;
  Calculer la force totale F_{totale};
  Lire les données du capteur LSM6DS3;
  Lire les accélérations a_x, a_y, a_z;
  Lire les vitesses angulaires ω_x, ω_y, ω_z;
  Filtrer les données mesurées;
  if $F_{totale} > F_{seuil}$ **then**
    Détecter un impact;
    Estimer la zone d'impact;
  Préparer le paquet de données P sous la forme suivante;

$$P = \begin{cases} \text{Forces : } F_1, F_2, F_3, F_4, F_5, F_6, F_7, F_8 \\ \text{Accélérations : } a_x, a_y, a_z \\ \text{Vitesses angulaires : } \omega_x, \omega_y, \omega_z \\ \text{Temps de mesure : } t \end{cases}$$

  Envoyer les données vers la plateforme externe;

2.3.6 Évaluation du bilan énergétique du système

Le système étant embarqué à l'intérieur du ballon, la consommation énergétique doit être prise en compte. L'autonomie dépend principalement de la capacité de la batterie et du courant consommé par les différents composants [58, 69, 77].

Le courant total consommé est donné par :

$$I_{\text{total}} = I_{\text{ESP32}} + I_{\text{LSM6DS3}} + \sum_{i=1}^{8} I_{\text{FSR}_i} \tag{2.26}$$

L'autonomie théorique peut être estimée par :

$$T = \frac{C_{\text{batterie}}}{I_{\text{total}}} \tag{2.27}$$

avec :

- T : l'autonomie théorique en heures .

- C_{batterie} : la capacité de la batterie en mAh pour une batterie LiPo 3,7 V / 1000 mAh $C_{\text{batterie}} = 1000\,\text{mAh}$
- I_{total} : le courant total consommé par le système en mA .
- I_{ESP32} : le courant consommé par l'ESP32 .
- I_{LSM6DS3} : le courant consommé par le capteur inertiel LSM6DS3 .
- I_{FSR} : le courant consommé par le capteur FSR402 .

2.4 Simulation et validation du circuit sous Proteus

La simulation du circuit électronique constitue une étape importante avant la réalisation pratique du système. Elle permet de vérifier le fonctionnement général du montage, de tester la lecture des capteurs et d'observer les données affichées sur le terminal série. Dans ce travail, la simulation a été réalisée sous l'environnement Proteus ISIS, qui permet de concevoir et de simuler des circuits électroniques à base de microcontrôleurs avant le passage à la réalisation matérielle [62, 63].

Il faut noter que certains composants utilisés dans la réalisation finale ne sont pas disponibles directement dans la bibliothèque ISIS Proteus. Pour cette raison, des composants équivalents ont été utilisés dans la simulation. Le capteur FSR402, prévu pour la réalisation pratique, a été remplacé par un capteur de force FSR406 disponible dans Proteus. Ce remplacement reste acceptable dans le cadre de la simulation, car les deux capteurs appartiennent à la même famille de capteurs résistifs sensibles à la force et reposent sur le même principe de variation de résistance sous l'effet d'une force appliquée [74, 75]. De même, le capteur inertiel LSM6DS3 a été remplacé par le module MPU6050, car ce dernier intègre également un accéléromètre trois axes et un gyroscope trois axes, ce qui permet de valider le principe général d'acquisition des données inertielles [77, 78]. Enfin, l'ESP32 prévu pour la réalisation finale a été remplacé dans la simulation par une carte Arduino Uno, car cette carte est disponible dans Proteus et permet de valider le principe général de lecture des entrées analogiques, de communication avec le capteur inertiel et d'affichage des résultats dans le terminal virtuel [62, 64, 69].

Ainsi, cette simulation ne vise pas à reproduire exactement tous les composants utilisés dans la réalisation finale. Elle a pour objectif principal de valider le principe général du circuit, c'est-à-dire la lecture des capteurs de force, l'acquisition des données inertielles, le traitement des mesures et l'affichage des résultats dans le terminal virtuel.

2.4.1 Schématique du circuit simulé

La figure 2.8 présente le schéma du circuit simulé sous Proteus ISIS. Le montage comprend une carte Arduino Uno, quatre capteurs de force FSR406 montés en diviseur de tension, un module inertiel MPU6050 et un terminal virtuel. Les capteurs FSR406 sont utilisés comme

équivalents de simulation des capteurs FSR402 afin de représenter le comportement d'un capteur résistif sensible à la force. Le MPU6050 est utilisé comme équivalent fonctionnel du LSM6DS3 pour représenter l'acquisition des accélérations et des vitesses angulaires [64,75,78].

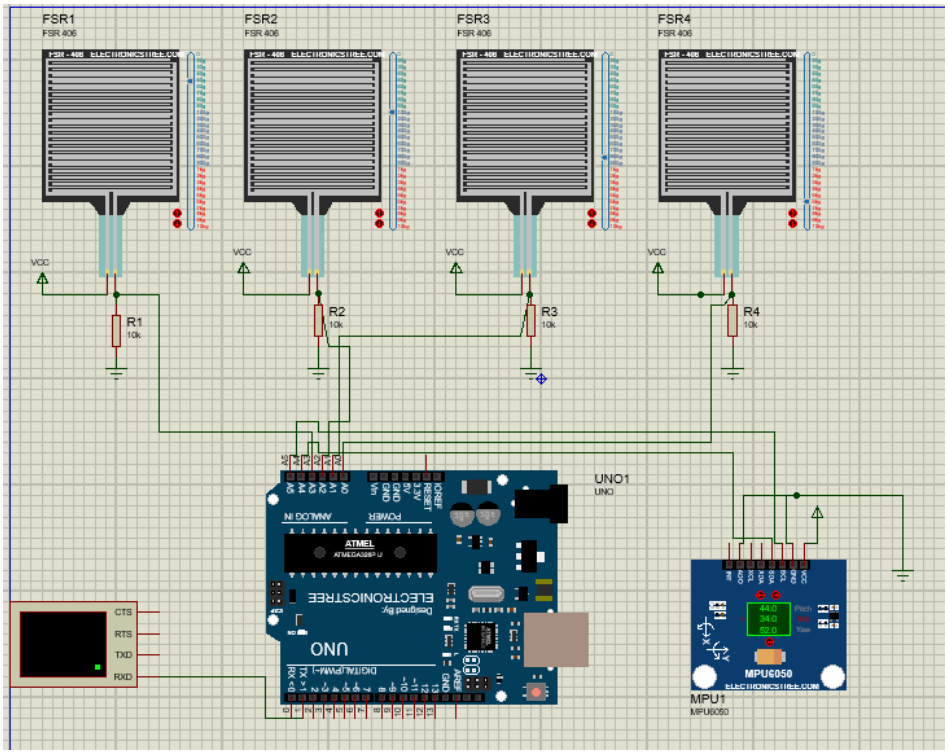


Figure 2.8 – Schéma de simulation du circuit électronique

Source : réalisé sous Proteus ISIS.

Les capteurs FSR406 sont reliés à des résistances fixes afin de former des diviseurs de tension. Lorsque la force appliquée sur un capteur varie, sa résistance change, ce qui entraîne une variation de la tension de sortie. Cette tension est ensuite lue par une entrée analogique de l'Arduino Uno. Ce principe permet de transformer une variation mécanique en une donnée électrique exploitable par le microcontrôleur [64, 75].

Le module MPU6050 est relié à l'Arduino Uno à travers le bus I2C. Il fournit les valeurs d'accélération selon les axes x, y et z, ainsi que les vitesses angulaires autour de ces mêmes axes. Ces données permettent de représenter le comportement inertiel du ballon dans la simulation, même si le composant utilisé dans la réalisation finale est le LSM6DS3 [77, 78].

2.4.2 Principe de fonctionnement de la simulation

Le fonctionnement de la simulation repose sur l'acquisition des données issues des capteurs. Les capteurs FSR406 produisent une variation de tension selon la force appliquée. Cette tension est lue par les entrées analogiques de l'Arduino Uno, puis convertie en valeurs

numériques par le convertisseur analogique-numérique. Les valeurs obtenues permettent ensuite d'estimer la force appliquée sur chaque capteur [64, 75].

Le MPU6050 fournit les données d'accélération et de vitesse angulaire. Ces informations permettent de vérifier la lecture des grandeurs inertielles et leur affichage dans le terminal virtuel. Le programme embarqué assure donc la lecture des capteurs, la conversion des mesures, l'estimation des forces et l'affichage des résultats.

Le principe général de la simulation peut être résumé comme suit :

- lecture des tensions issues des capteurs FSR406 ;
- conversion des tensions analogiques en valeurs numériques ;
- estimation de la force appliquée sur chaque capteur ;
- acquisition des accélérations et des vitesses angulaires du MPU6050 ;
- affichage des résultats dans le terminal virtuel.

Cette simulation permet donc de vérifier la chaîne de traitement depuis les capteurs jusqu'à l'affichage des données. Elle valide le comportement logique du système avant l'étape de réalisation pratique.

2.4.3 Analyse des flux de données (Terminal virtuel)

La figure 2.9 présente les résultats affichés dans le terminal virtuel de Proteus. Les valeurs affichées montrent les mesures ADC des capteurs FSR406, les forces estimées en newtons, ainsi que les accélérations et les vitesses angulaires fournies par le MPU6050.

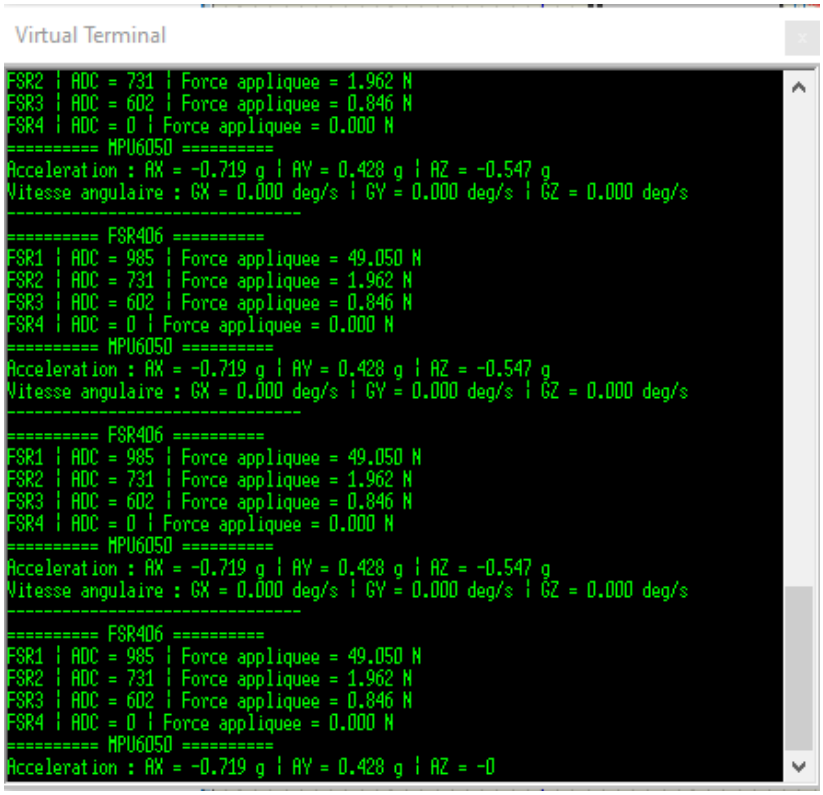


Figure 2.9 – Résultats de simulation affichés dans le terminal virtuel

Source : résultats obtenus sous Proteus ISIS.

Dans les résultats affichés, chaque capteur FSR406 est représenté par une valeur ADC et une force estimée. Lorsque la valeur ADC augmente, la force calculée augmente également. Cela confirme que le montage en diviseur de tension permet de transformer la variation de résistance du capteur en une tension mesurable, puis en une donnée numérique exploitable par le programme [64, 75].

Les données du MPU6050 affichent les accélérations selon les axes x, y et z, ainsi que les vitesses angulaires selon les mêmes axes. Ces informations permettent de vérifier que la communication avec le capteur inertiel fonctionne correctement et que les mesures peuvent être affichées dans le terminal série [78].

2.4.4 Interprétation et discussion des résultats de simulation

Les résultats obtenus montrent que le circuit simulé est capable de lire plusieurs capteurs de force, d'estimer les forces appliquées et d'afficher les données inertielles. Par exemple, lorsqu'un capteur reçoit une valeur ADC élevée, la force affichée devient plus importante. Cela montre que le système réagit correctement aux variations appliquées sur les capteurs FSR406.

La simulation permet aussi de vérifier la communication entre les différents blocs du

système. Les capteurs de force sont lus à travers les entrées analogiques, le capteur inertiel communique par I2C, et les résultats sont transmis vers le terminal virtuel. Cette organisation valide le principe général d'acquisition et d'affichage des données.

Cependant, les résultats de simulation ne représentent pas parfaitement les conditions réelles d'utilisation. Dans un vrai ballon, les capteurs seront soumis à des vibrations, des chocs rapides, des erreurs de placement et des contraintes mécaniques. De plus, les composants utilisés dans la simulation sont des équivalents et non les composants exacts de la réalisation finale. La simulation permet donc de valider le principe électronique et logiciel du système, mais les performances finales doivent être confirmées par des tests expérimentaux.

2.5 Conclusion

Dans ce chapitre, nous avons présenté la modélisation et la simulation du ballon instrumenté. La première partie a été consacrée à la modélisation physique du ballon, en étudiant les principales forces qui influencent son mouvement, comme le poids, la réaction du sol, la force de tir, la force de l'air, le frottement et la force de Magnus. Cette étude permet de mieux comprendre le comportement dynamique du ballon lors d'un tir, d'un rebond ou d'un déplacement sur le terrain.

La deuxième partie a porté sur la modélisation électronique du système intégré dans le ballon. Les différents blocs du système ont été présentés, notamment les capteurs de force, le capteur inertiel, le microcontrôleur et l'alimentation. Cette modélisation a permis d'expliquer le principe d'acquisition des données, la conversion analogique-numérique, l'estimation des forces appliquées, la lecture des grandeurs inertielles, le filtrage des mesures et le traitement embarqué des informations.

Enfin, une simulation du circuit électronique a été réalisée sous Proteus afin de valider le principe général de fonctionnement avant la réalisation pratique. Comme certains composants réels ne sont pas disponibles directement dans ISIS Proteus, des modèles équivalents ont été utilisés. Les capteurs FSR406 ont été employés comme équivalents des FSR402, et le MPU6050 a été utilisé comme équivalent fonctionnel du LSM6DS3. Les résultats affichés dans le terminal virtuel ont montré que le système est capable de lire les capteurs, d'estimer les forces appliquées et d'afficher les données d'accélération et de vitesse angulaire.

Ainsi, ce chapitre a permis de poser les bases théoriques et fonctionnelles du système proposé. La modélisation a facilité la compréhension du comportement physique et électronique du ballon, tandis que la simulation a confirmé la cohérence du principe d'acquisition et de traitement des données. Ces résultats préparent le passage vers le chapitre suivant, consacré à la réalisation pratique du ballon instrumenté connecté et de la plateforme IoT.

Chapitre III

Réalisation finale, intégration IoT et validation expérimentale du ballon instrumenté

3.1 Introduction

Ce chapitre présente la phase finale de réalisation du ballon instrumenté. Après les étapes de conception, de modélisation et de simulation, cette partie décrit le passage vers un prototype physique fonctionnel intégrant la structure mécanique, le système d'alimentation, les capteurs, la carte de traitement et la plateforme IoT. L'objectif principal est de montrer comment les différents éléments étudiés précédemment ont été assemblés pour obtenir un système embarqué capable de mesurer le comportement du ballon et de transmettre les données vers un tableau de bord.

La démarche adoptée repose sur une réalisation progressive du système. Dans un premier temps, les sous-systèmes du prototype ont été développés séparément afin de faciliter le montage, les essais et les corrections. Le sous-système mécanique regroupe la structure externe du ballon et la capsule interne destinée à recevoir les composants électroniques. Le sous-système d'alimentation assure l'autonomie du prototype, tandis que le sous-système d'acquisition et de traitement permet la lecture des capteurs et la préparation des données. Dans un second temps, ces différents blocs sont intégrés dans une structure complète afin de valider la faisabilité du ballon instrumenté.

Le chapitre présente ensuite l'intégration IoT du système, à travers la communication entre le prototype et le tableau de bord. Cette partie permet de visualiser les informations mesurées par les capteurs, notamment les données liées aux impacts, au mouvement et à l'orientation du ballon. Enfin, les essais expérimentaux sont présentés afin de vérifier le fonctionnement global du système, d'évaluer la cohérence des données obtenues et de discuter les limites observées ainsi que les améliorations possibles.

3.2 Réalisation finale du prototype du ballon

3.2.1 Réalisation des sous-systèmes du prototype

Après la phase de conception et de choix des composants, la réalisation du prototype a été effectuée progressivement. Afin de faciliter le montage, les essais et les corrections, le système n'a pas été assemblé directement sous sa forme finale. Il a d'abord été divisé en plusieurs sous-systèmes : le sous-système mécanique, le sous-système d'alimentation et le sous-système d'acquisition et de traitement des données. Cette organisation permet de vérifier chaque partie séparément avant de procéder à l'intégration finale du ballon instrumenté. Elle permet également d'identifier plus facilement les contraintes liées à l'encombrement, à la fixation des composants, au passage des câbles et à l'équilibrage général du prototype.

3.2.1.1 Sous-système mécanique

Le sous-système mécanique constitue la partie physique du prototype. Il a pour rôle de supporter les composants électroniques, de protéger les éléments internes et de conserver une

forme proche de celle d'un ballon. Dans cette réalisation, deux éléments principaux ont été utilisés : une structure externe imprimée en TPU flexible et une capsule interne imprimée en PLA. L'impression 3D a été choisie pour fabriquer ces différentes pièces, car elle permet de produire rapidement des formes personnalisées et adaptées aux dimensions du système. Le procédé FDM repose sur le dépôt successif de couches de matière thermoplastique afin de construire une pièce en trois dimensions [70, 71].

3.2.1.2 Fabrication de la structure externe du ballon

La structure externe du ballon a été conçue sous forme d'un modèle 3D ajouré afin de permettre l'intégration du système électronique à l'intérieur du prototype. Cette conception permet de garder une forme sphérique proche de celle d'un ballon, tout en laissant un espace interne suffisant pour la capsule et les composants embarqués. Le modèle 3D a d'abord été visualisé dans une visionneuse 3D afin de vérifier la forme générale, les ouvertures, l'épaisseur des parois et la continuité de la structure avant le lancement de l'impression.

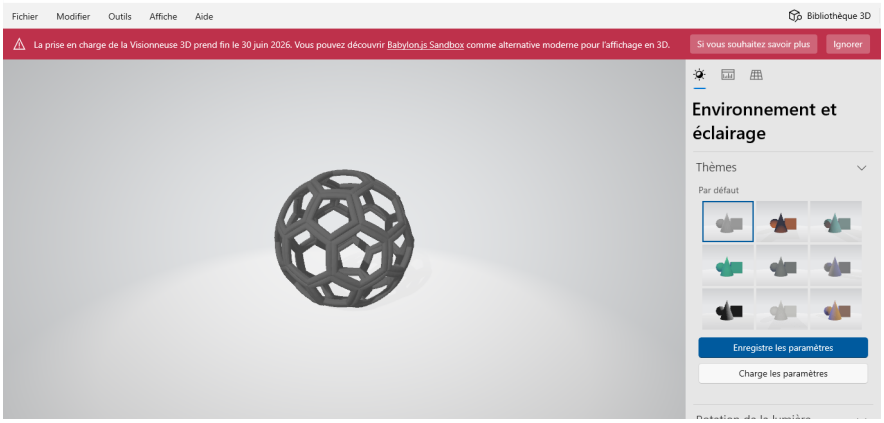


Figure 3.1 – Modèle 3D de la structure du ballon instrumenté avant impression.

Source : réalisé par l'auteur.

La fabrication a ensuite été réalisée par impression 3D en utilisant un matériau TPU flexible. Ce matériau a été choisi pour sa souplesse et sa capacité à supporter certaines déformations, ce qui le rend plus adapté à une structure de ballon qu'un matériau totalement rigide. L'objectif n'est pas de reproduire exactement les propriétés d'un ballon professionnel, mais d'obtenir un prototype expérimental permettant d'intégrer les capteurs et de tester le principe du système embarqué.

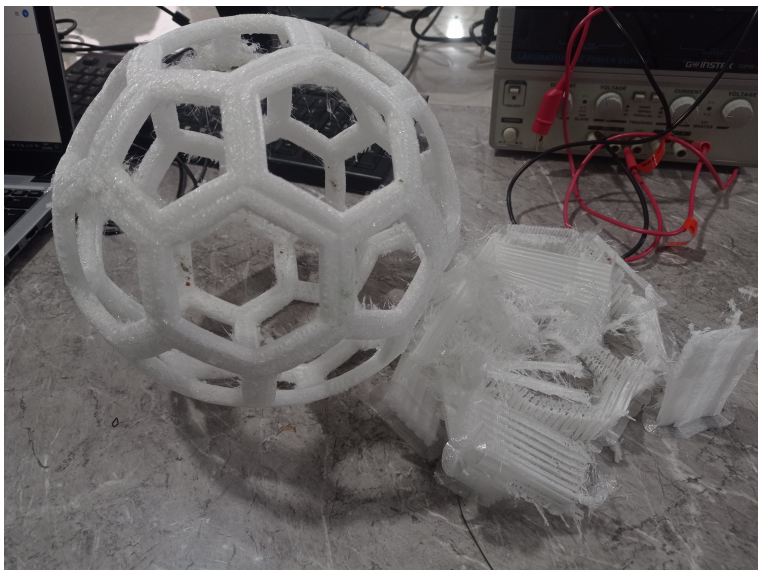


Figure 3.2 – Prototype de la structure externe du ballon imprimée en TPU flexible.

Source : réalisé par l'auteur.

La figure 3.2 montre la structure imprimée du ballon. On remarque une géométrie ouverte composée de plusieurs formes polygonales. Cette forme permet de réduire la masse du prototype, de faciliter l'accès à l'intérieur pendant l'intégration du système et de vérifier plus facilement le placement de la capsule, des fils et des composants avant l'assemblage final. La structure imprimée présente un léger défaut localisé sur un côté du ballon. Ce défaut est lié au procédé d'impression 3D et se manifeste par une petite déformation avec une zone partiellement coupée. Toutefois, il ne s'agit pas d'un défaut majeur, car la forme générale du ballon reste conservée et la structure demeure exploitable pour les essais. Une correction mécanique simple a été réalisée à l'aide d'une colle adaptée afin de renforcer la zone concernée et d'assurer la continuité de la structure.

3.2.1.3 Réalisation de la capsule interne

La capsule interne a été conçue pour recevoir le système électronique embarqué à l'intérieur du ballon. Sa forme sphérique permet de mieux s'adapter à la géométrie du prototype et de répartir les composants de manière plus équilibrée. Elle est composée de deux demi-coques assemblables, ce qui facilite l'ouverture, l'intégration des composants et les opérations de maintenance. La conception de la capsule comporte plusieurs ouvertures réparties sur sa surface. Ces ouvertures permettent de réduire la masse de la pièce, de faciliter le passage des fils et d'améliorer l'accessibilité aux composants internes. Elles peuvent également servir au positionnement ou à la fixation de certains éléments du système embarqué.

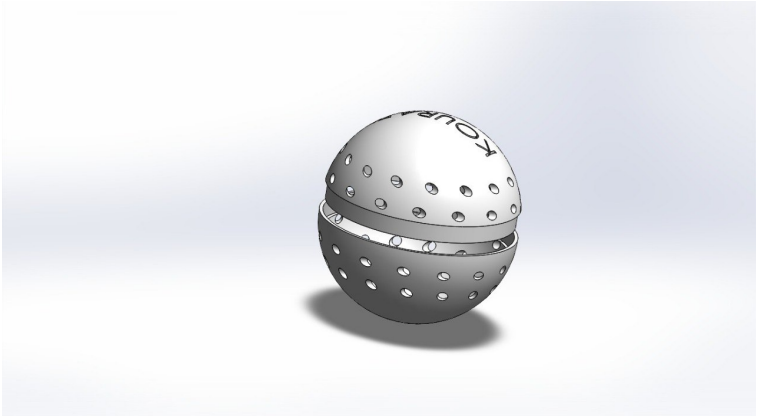


Figure 3.3 – Modèle 3D de la capsule interne du ballon instrumenté.

Source : réalisé par l'auteur.

La capsule a été réalisée en PLA, car ce matériau est rigide, facile à imprimer et adapté à la fabrication de supports personnalisés. Cette rigidité est importante pour maintenir le microcontrôleur, les capteurs, la batterie et les connexions dans une position stable pendant les essais. Le PLA est un matériau couramment utilisé en impression 3D en raison de sa simplicité d'utilisation et de sa stabilité dimensionnelle [72]. Après l'impression, les deux parties de la capsule ont été préparées pour recevoir le système électronique. Les tiges filetées visibles sur le prototype ne sont pas utilisées pour fermer la capsule, mais pour assurer la fixation mécanique entre la capsule interne en PLA et la structure externe du ballon imprimée en TPU. Cette solution permet de maintenir la capsule au centre du ballon, de limiter son déplacement pendant les mouvements et de renforcer la stabilité de l'ensemble lors des essais.



Figure 3.4 – Capsule interne imprimée en PLA.

Source : réalisé par l'auteur.

3.2.1.4 Sous-système d'alimentation

Le sous-système d'alimentation a été réalisé séparément afin de rendre le prototype autonome. Il assure l'énergie nécessaire au fonctionnement du système embarqué placé dans le ballon. Il est constitué d'une batterie lithium-ion 18650 de 3,7 V, d'une carte BMS 1S 3,7 V, d'un module de charge TP4056 USB-C, d'un interrupteur et d'un convertisseur DC-DC ajustable. La batterie 18650 représente la source principale d'énergie du système. Elle fournit une tension nominale de 3,7 V et une capacité indiquée de 2500 mAh. Cette batterie a été choisie pour son format compact et sa capacité suffisante pour alimenter un système embarqué de petite taille.

La carte BMS 1S 3,7 V est ajoutée afin d'assurer la protection de la cellule lithium-ion, notamment contre la surcharge, la décharge excessive et les courts-circuits. Le module TP4056 USB-C est utilisé pour la recharge de la batterie. Ce module est basé sur un circuit de charge adapté aux batteries lithium-ion à une seule cellule et fonctionne selon une méthode de charge à courant constant puis à tension constante [73]. L'interrupteur permet de mettre le système sous tension ou de l'arrêter manuellement. Il est placé entre la partie alimentation et le reste du circuit afin de contrôler facilement l'alimentation générale du prototype. Le convertisseur DC-DC ajustable permet ensuite d'adapter la tension fournie par la batterie aux besoins des différents composants électroniques. Ce bloc est important, car les capteurs et le microcontrôleur nécessitent une alimentation stable pour fonctionner correctement.

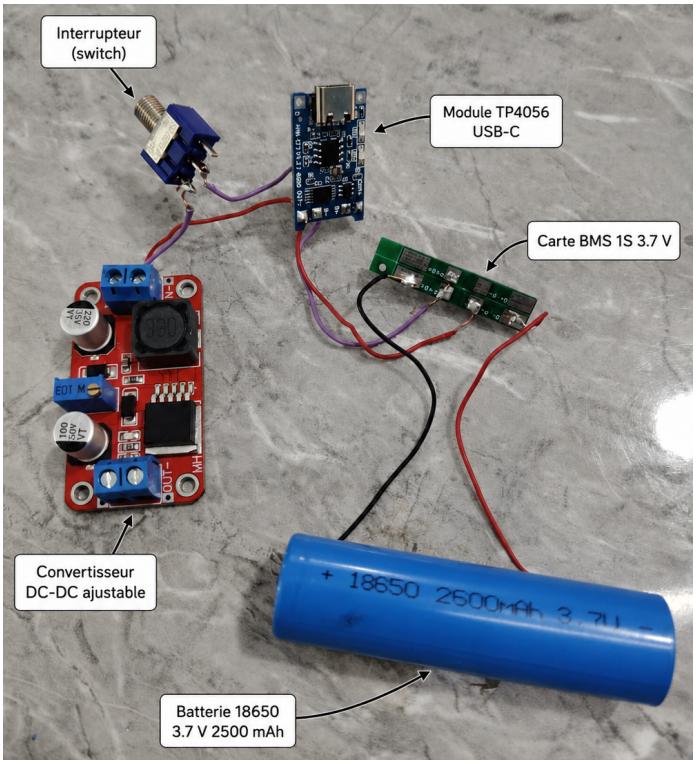


Figure 3.5 – Circuit d'alimentation du système embarqué.

Source : réalisé par l'auteur.

La figure 3.5 présente le circuit d'alimentation embarqué. Cette partie permet de fournir une énergie stable au système tout en assurant la recharge et la protection de la batterie. Elle constitue donc une étape importante pour garantir l'autonomie et la sécurité de fonctionnement du prototype.



Figure 3.6 – Bloc d'alimentation intégré dans la capsule interne.

Source : réalisé par l'auteur.

La figure 3.6 montre l'intégration pratique du bloc d'alimentation dans la capsule interne. Cette intégration permet de vérifier l'encombrement de la batterie, du module de protection et du convertisseur DC-DC, ainsi que le passage des fils vers le reste du système électronique.

3.2.1.5 Sous-système d'acquisition et de traitement

Le sous-système d'acquisition et de traitement regroupe les capteurs, les circuits de conditionnement et le microcontrôleur utilisés pour mesurer le comportement du ballon. Cette partie permet de collecter les informations liées aux impacts, aux mouvements et à l'orientation, puis de les préparer pour la transmission vers la plateforme IoT. Les capteurs FSR402 sont utilisés pour détecter les forces appliquées sur le ballon. Comme ce type de capteur fonctionne avec une résistance variable, il est nécessaire de l'associer à une résistance fixe afin de former un diviseur de tension. Ce montage permet de transformer la variation de résistance en une tension analogique mesurable par l'ESP32-S3 [74, 75].

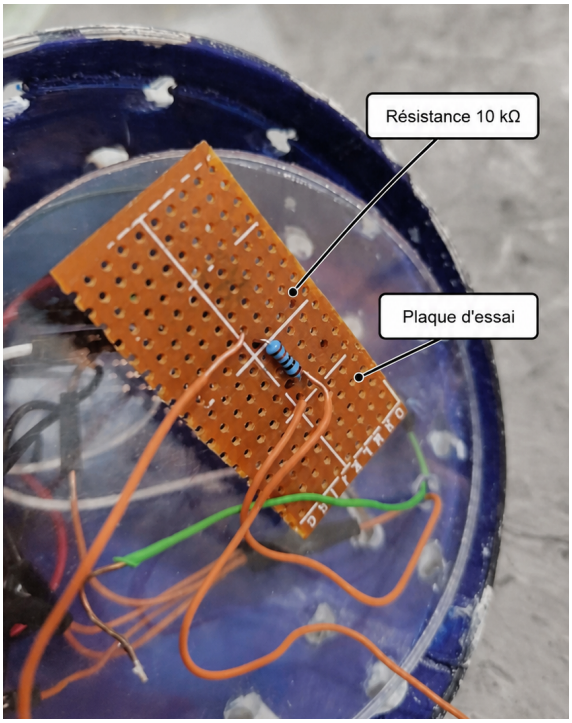


Figure 3.7 – Montage du diviseur de tension utilisé avec le capteur FSR402.

Source : réalisé par l'auteur.

Afin d'améliorer la transmission de l'effort vers la surface active du capteur, le FSR402 a été placé dans un tube en silicone au niveau de la zone sensible du ballon. Cette disposition permet de maintenir le capteur en position et d'obtenir un meilleur contact mécanique lors des impacts. Le tube en silicone joue aussi un rôle de support flexible, ce qui limite les contraintes directes sur les connexions du capteur.

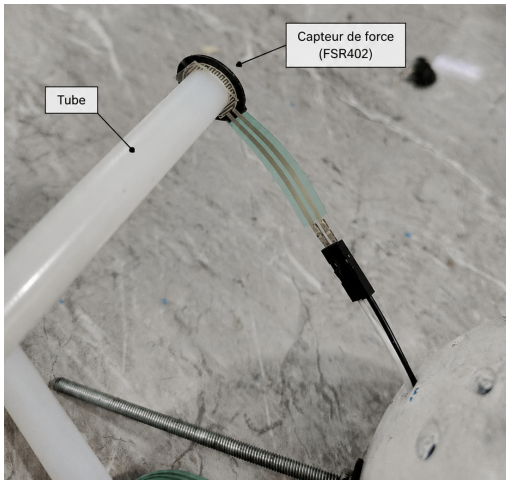


Figure 3.8 – Placement du capteur FSR402

Source : réalisé par l'auteur.

Remarque

Le prototype utilise 4 capteurs FSR402 au lieu de 8 dans l'étude théorique, en raison de contraintes d'intégration, de coût et de complexité matérielle.

La carte ESP32-S3 constitue l'unité principale de traitement. Elle reçoit les signaux issus des capteurs, effectue la lecture des données et assure leur transmission vers la plateforme IoT. Le module LSM6DS3 est utilisé pour mesurer les accélérations linéaires et les vitesses angulaires du ballon. Le module GY-87 est ajouté pour compléter l'estimation de l'orientation, notamment à travers les données du magnétomètre. L'ensemble de ces composants est monté sur une plaque de prototypage afin de faciliter le câblage, les essais et les modifications avant l'intégration finale dans la capsule interne [76–78].

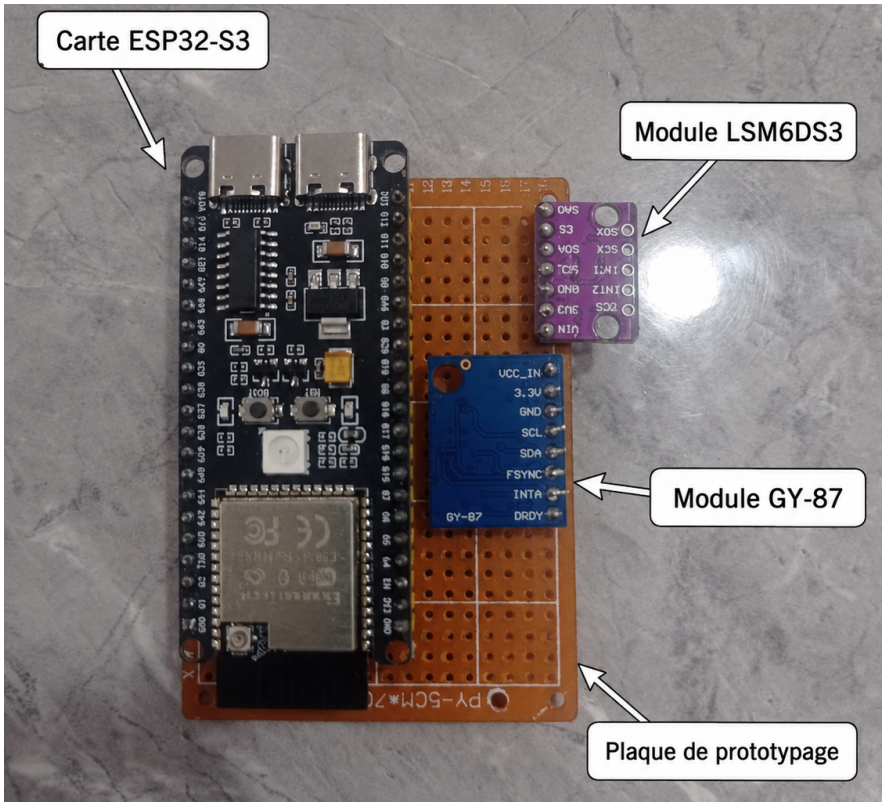


Figure 3.9 – Plaque de prototypage embarquée regroupant l'ESP32-S3 et les modules de mesure.

Source : réalisé par l'auteur.

Cette architecture permet d'obtenir une chaîne d'acquisition compacte et adaptée au prototype. Les données issues des capteurs de force et des capteurs inertiels sont centralisées par l'ESP32-S3, puis exploitées pour suivre l'état du ballon. Une attention particulière est portée à l'organisation des fils et à la fixation des composants afin de réduire les risques de mauvais contact, de déconnexion ou de déplacement pendant les essais.

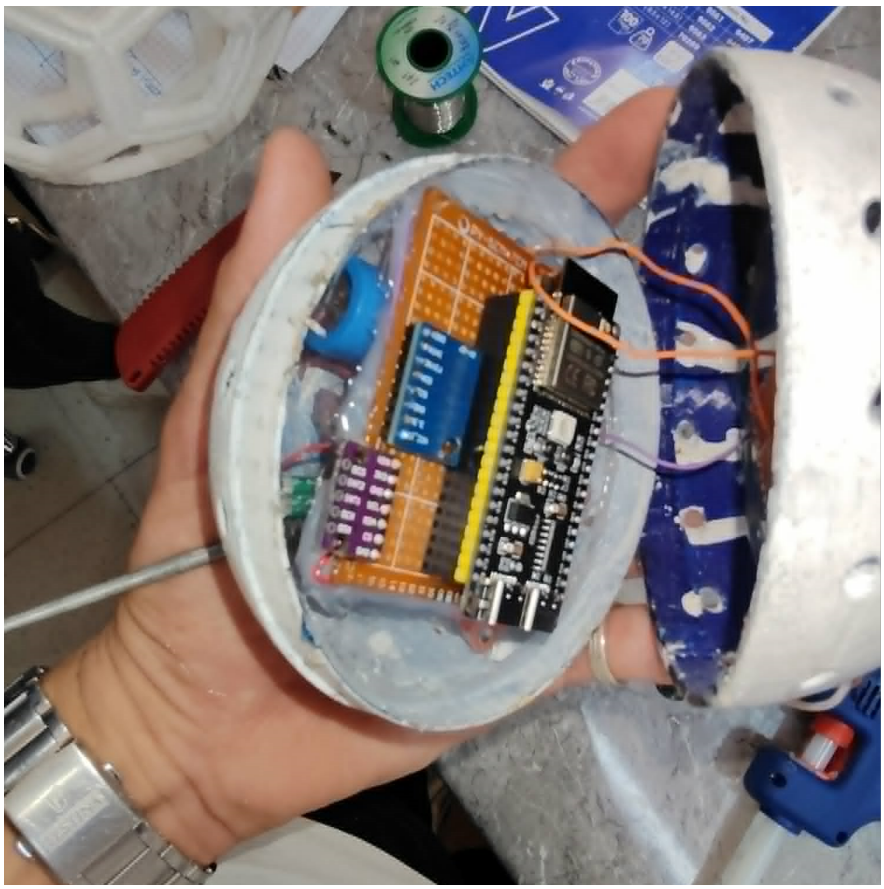


Figure 3.10 – Intégration du circuit d'acquisition dans la capsule interne.

Source : réalisé par l'auteur.

La figure 3.10 montre l'intégration du circuit d'acquisition dans la capsule interne. Cette étape permet de vérifier la compatibilité entre la partie électronique et la partie mécanique, notamment en ce qui concerne l'espace disponible, le positionnement des modules et le passage des câbles.

3.2.1.6 Analyse comparative entre le circuit de simulation et réalisation

Table 3.1 – Comparaison entre simulation et réalisation

Composant	Simulation	Réalisation	Différences / Similarités	Justification
µcontrôleur	Arduino	ESP32	8-bit vs 32-bit, Wi-Fi absent vs intégré. UART, I2C, SPI communs.	ESP32 choisi pour performance et connectivité sans fil.
Capteur pression	FSR406	FSR402	Différence de taille et intégration. Même principe FSR.	FSR402 adapté à l'intégration dans un ballon.
IMU	MPU6050	LSM6DS3 + GY87	6 axes vs 9 axes + magnétomètre + baromètre. I2C commun.	Amélioration de la précision et fusion sensorielle.
Interface	Terminal série	Dashboard Wi-Fi	Texte vs interface graphique. Local vs sans fil.	Supervision temps réel et visualisation.
Alimentation	Idéale	Réelle	Source parfaite vs bruit + régulation.	Stabilité nécessaire en système embarqué.
Protocole	I2C virtuel	I2C physique	Pas de contraintes vs pull-up nécessaires.	Réduction du câblage multi-capteurs.

La comparaison entre la simulation et la réalisation réelle montre que le système conserve le même principe de fonctionnement malgré l'utilisation de composants différents.

3.2.2 Intégration finale du système complet

Après la réalisation séparée des différents sous-systèmes, l'étape finale consiste à assembler l'ensemble dans une seule structure fonctionnelle. Le sous-système d'acquisition, le sous-système d'alimentation et les connexions sont placés à l'intérieur de la capsule interne. Cette capsule est ensuite fixée dans la structure externe du ballon imprimée en TPU. Cette intégration permet de passer d'un ensemble de modules séparés à un prototype complet de ballon instrumenté. Elle permet également de vérifier l'encombrement interne, la stabilité des composants, le passage des câbles et la cohérence de l'assemblage mécanique.

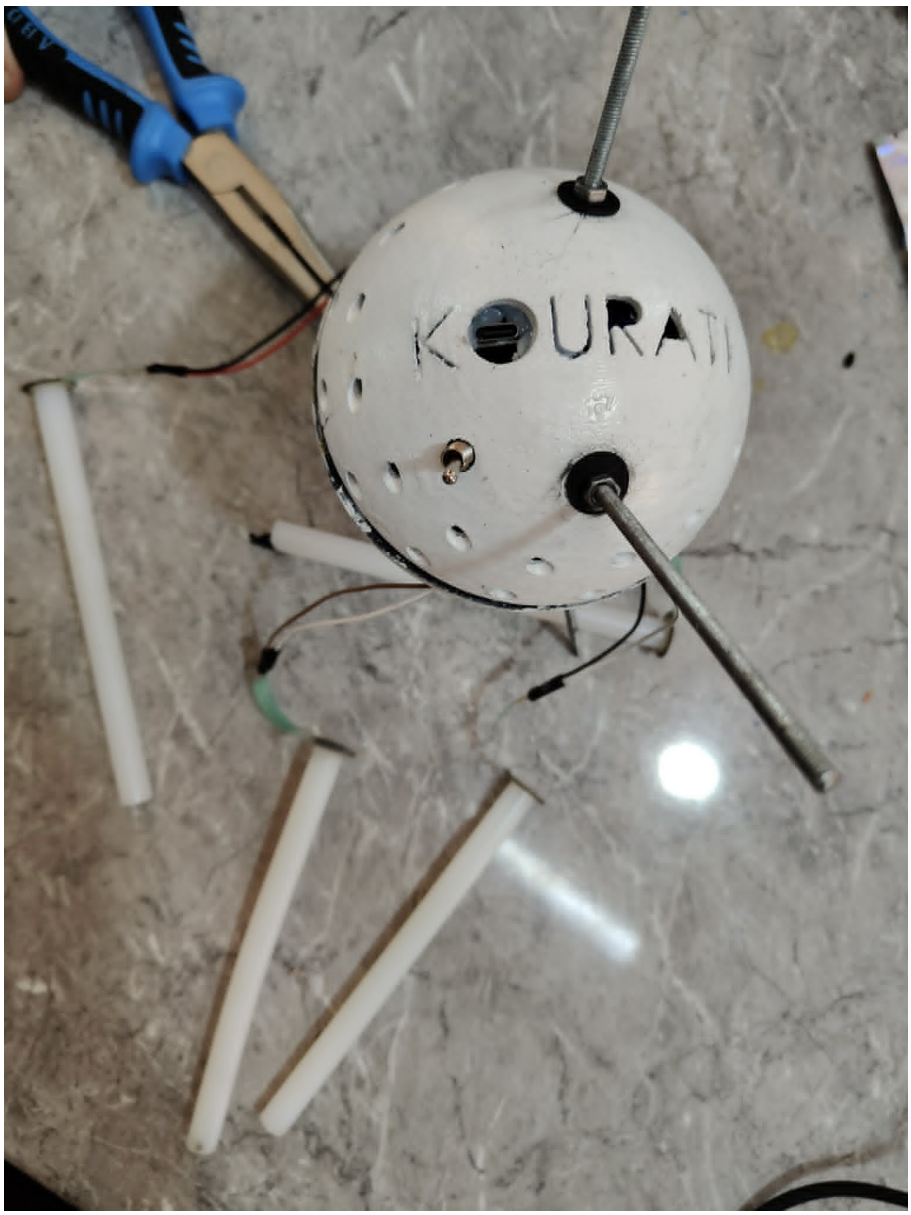


Figure 3.11 – Système électronique intégré dans la capsule interne avant assemblage complet.

Source : réalisé par l'auteur.

La figure 3.11 montre le regroupement des éléments électroniques à l'intérieur de la capsule. Cette configuration permet de centraliser les composants dans un volume réduit, ce qui facilite l'intégration dans la structure externe du ballon.

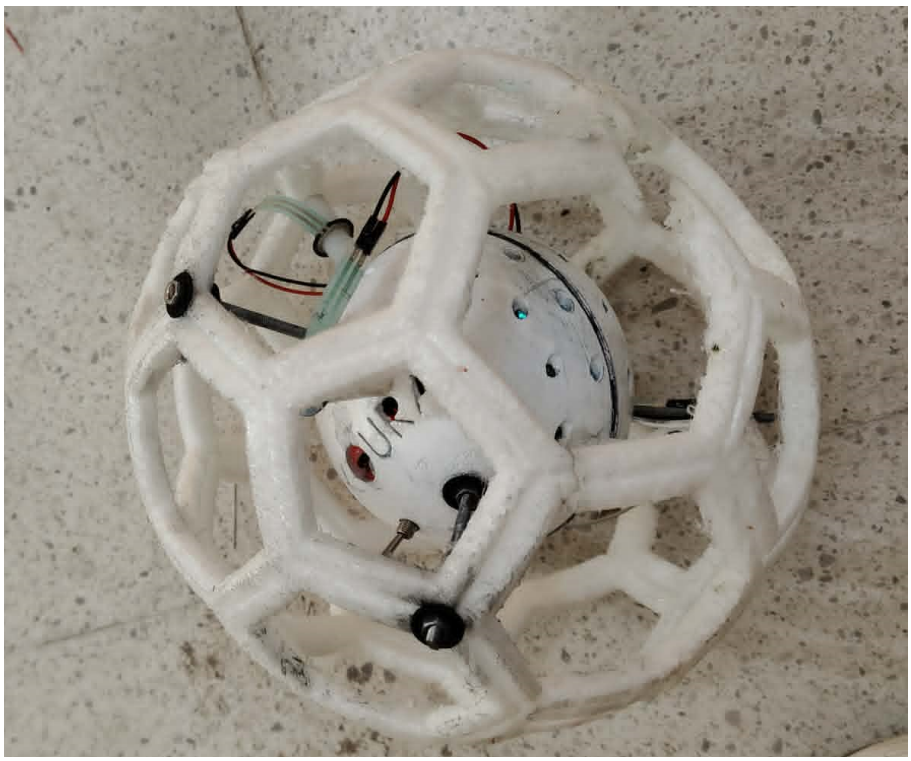


Figure 3.12 – Système complet du ballon instrumenté après assemblage.

Source : réalisé par l'auteur.

La figure 3.12 présente le prototype final après assemblage. Le système complet regroupe la structure mécanique, la capsule interne, le circuit d'alimentation, la carte de traitement et les capteurs. Cette réalisation valide la possibilité d'intégrer un système embarqué autonome dans une structure sphérique destinée à l'étude du comportement d'un ballon.

Il est important de préciser que ce prototype constitue une version expérimentale destinée à la validation du principe de fonctionnement. Il permet de tester l'acquisition des données, l'alimentation autonome, la transmission des informations et l'intégration mécanique dans une structure de ballon. Il ne s'agit donc pas encore d'un ballon final destiné à une utilisation sportive réelle, mais d'une plateforme expérimentale permettant de valider les choix techniques retenus. Des améliorations restent nécessaires pour obtenir une version plus robuste, notamment l'optimisation de la masse, l'équilibrage du ballon, la réduction du volume du circuit, la fixation définitive des fils et le renforcement de la protection mécanique contre les chocs. Malgré ces limites, cette intégration finale démontre la faisabilité de la solution proposée et prépare la phase des essais expérimentaux ainsi que la visualisation des données sur le tableau de bord IoT.

3.3 Intégration IoT et réalisation du tableau de bord KOURATI

Après la réalisation matérielle du prototype, l'étape suivante consiste à assurer la communication entre le ballon instrumenté et l'interface de supervision. Cette partie présente l'intégration IoT du système, depuis l'acquisition des données par l'ESP32-S3 jusqu'à leur affichage sur le tableau de bord web KOURATI. L'objectif est de mettre en place une chaîne complète permettant de transmettre les mesures issues des capteurs, de les traiter côté serveur et de les présenter sous une forme visuelle compréhensible pour l'utilisateur.

Le tableau de bord KOURATI constitue la partie visible du système. Il permet d'afficher les données transmises par le ballon instrumenté, notamment les mesures d'accélération, de rotation, d'orientation et les informations liées aux événements de jeu. Son objectif n'est pas seulement de présenter des valeurs numériques, mais aussi de transformer les données issues des capteurs en informations exploitables, telles que l'état du ballon, sa position estimée sur le terrain, les impacts détectés et certaines statistiques du match.

3.3.1 Présentation générale du tableau de bord

Le dashboard web KOURATI est l'interface de supervision centrale du projet de ballon intelligent. Son rôle est de collecter, traiter et afficher en temps réel l'ensemble des données produites par le ballon, sans qu'aucune intervention manuelle ne soit nécessaire. Il transforme des mesures physiques brutes — accélérations, rotations et orientation magnétique — en informations footballistiques directement exploitables : position du ballon sur le terrain, vitesse, orientation 3D [88], score, possession, statistiques des équipes, et détection des événements tels que les corners, les touches ou les buts. L'objectif du dashboard est triple. Il supervise l'état physique du ballon en continu, présente les statistiques du match de manière visuelle et comparative pour les deux équipes, et signale immédiatement les événements footballistiques importants grâce à des indicateurs visuels dédiés. La contrainte de faible latence a directement guidé les choix architecturaux : chaque composant du système — du protocole MQTT [80] au WebSocket [82] — a été sélectionné pour minimiser le délai entre l'action sur le terrain et son affichage à l'écran.

3.3.2 Architecture Générale du Système

Le système KOURATI repose sur quatre composants principaux qui communiquent entre eux de manière séquentielle, chacun ayant un rôle bien défini.

- Le ballon intelligent (ESP32 [87] + LSM6DS3 [86] + G-87 [88]) est le producteur de données. Il embarque deux capteurs inertiels : le LSM6DS3 (6 axes) pour l'accélération et la rotation, et le G-87 (9 axes) qui ajoute le magnétomètre pour l'orientation absolue du ballon. L'ensemble est échantillonné à 50 Hz et transmis via WiFi.

- Le broker Mosquitto [81] joue le rôle d'intermédiaire de messagerie MQTT [80]. Il reçoit les données du ballon et les redirige automatiquement vers le serveur.
- Le backend Flask [90] est le cerveau du système. Il reçoit les données, les traite par filtrage numérique et filtre de Kalman [85], détecte les événements, les enregistre en base de données [91], et les envoie vers le dashboard.
- Le dashboard web (frontend) reçoit les données en temps réel via WebSocket [82] / Socket.IO [92] et les présente à l'utilisateur sous forme graphique.

3.3.2.1 Les capteurs embarqués : LSM6DS3 et G-87

Le ballon intelligent intègre deux capteurs inertiels complémentaires. Le LSM6DS3 [86] est un capteur 6 axes qui associe un accéléromètre 3 axes et un gyroscope 3 axes. Il mesure les forces d'accélération subies par le ballon et sa vitesse de rotation sur chacun des trois axes. Le G-87 [88] est un capteur 9 axes qui ajoute un magnétomètre 3 axes aux données d'accélération et de gyroscopie. Ce magnétomètre fonctionne à la manière d'une boussole : en mesurant le champ magnétique terrestre, il permet de calculer l'orientation absolue du ballon (son cap magnétique). Ensemble, ces deux capteurs fournissent une description complète du mouvement du ballon dans l'espace à trois dimensions [88, 93].

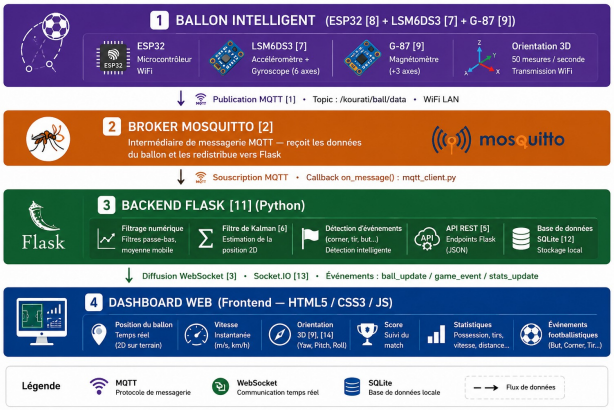


Figure 3.13 – Architecture générale du système KOURATI

Comme l'illustre la Figure 3.13, les données parcourent un chemin défini : le ballon publie ses mesures vers Mosquitto [81], Flask s'y abonne pour les recevoir, les traite, puis diffuse les résultats vers le dashboard via WebSocket [82]. Chaque étape de ce pipeline est automatique et s'exécute en quelques dizaines de millisecondes.

3.3.2.2 Architecture en couches détaillée

Afin de mieux appréhender la structure interne du système, il est utile de le décrire selon le modèle d'architecture en couches. Chaque couche encapsule un niveau de responsabilité

précis et communique uniquement avec les couches adjacentes. Ce découplage garantit la maintenabilité du projet : modifier un composant d'une couche n'impacte pas les autres [89].

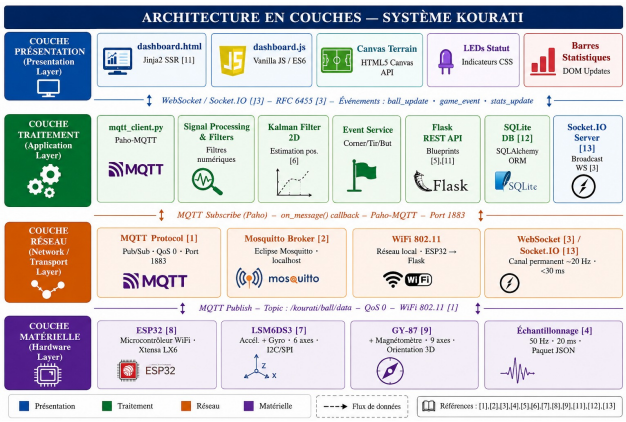


Figure 3.14 – Architecture en couches du système KOURATI

La Figure 3.14 présente le système selon quatre couches logiques. La couche matérielle regroupe les composants physiques (ESP32 [87], LSM6DS3 [86], G-87 [88]). La couche réseau assure le transport des données via MQTT [80] et le broker Mosquitto [81]. La couche traitement concentre toute la logique applicative : filtrage du signal, filtre de Kalman [85], détection d'événements, API REST [84] et persistance SQLite [91]. Enfin, la couche présentation affiche les résultats en temps réel via WebSocket [82] / Socket.IO [92].

3.3.2.3 Diagramme Fonctionnel du Système KOURATI

Le diagramme fonctionnel ci-dessous 3.14 représente le système KOURATI selon ses huit fonctions principales, organisées en séquence logique depuis l'acquisition physique des données jusqu'à leur visualisation sur le dashboard. Chaque fonction est indépendante et produit un résultat précis, transmis à la fonction suivante. Cette vue fonctionnelle permet de comprendre le rôle de chaque module du système sans entrer dans les détails de l'implémentation.

La première fonction correspond à l'acquisition des données inertielles par les capteurs LSM6DS3 [86] (accélération + gyroscope) et G-87 [88] (magnétomètre), à une fréquence de 50 mesures par seconde. Ces données brutes sont ensuite soumises à un prétraitement par filtres numériques (filtre passe-bas, moyenne mobile) afin d'éliminer le bruit de mesure. Le filtre de Kalman 2D [85] prend alors le relais pour estimer la position du ballon sur le terrain. L'orientation 3D complète (roulis, tangage, lacet) et la vitesse linéaire sont calculées à partir des données filtrées et magnétométriques [88, 93].

Le module de détection identifie automatiquement les événements footballistiques (tir, corner, but, faute, sortie), alimentant le module de génération des statistiques (possession, tirs, distance parcourue). Les données traitées sont enfin transmises via MQTT [80] et WebSocket [82] vers le dashboard web qui les affiche en temps réel [89].



Figure 3.15 – Diagramme fonctionnel du système KOURATI

3.3.2.4 Diagramme de Flux de Données

Le diagramme de flux de données 3.16 illustre le chemin physique emprunté par les données à travers les composants du système KOURATI. Contrairement au diagramme fonctionnel qui décrit les transformations logiques, ce diagramme montre les connexions concrètes entre les composants matériels et logiciels, ainsi que les protocoles de communication utilisés à chaque étape.

Les deux capteurs — LSM6DS3 [86] (accélération et gyroscope, 6 axes) et G-87 [88] (magnétomètre et boussole, 9 axes) — sont connectés à l'ESP32 [87] via le bus I2C. L'ESP32 agrège les données des deux capteurs dans un payload JSON [83] unique et le transmet par WiFi au broker MQTT Mosquitto [81]. Le message est publié sur le topic `kourati/imu`.

Le serveur backend Flask [90] souscrit à ce topic et reçoit chaque message via son client MQTT. À la réception, Flask effectue deux opérations simultanées : il insère les données traitées dans la base de données SQLite [91] (fichier `kourati.db`) pour la persistance, et diffuse les résultats en temps réel vers le dashboard web via WebSocket (Socket.IO [92]).

Le dashboard, fonctionnant comme une Single Page Application (SPA), reçoit les

événements WebSocket [82] et met à jour l'affichage instantanément.

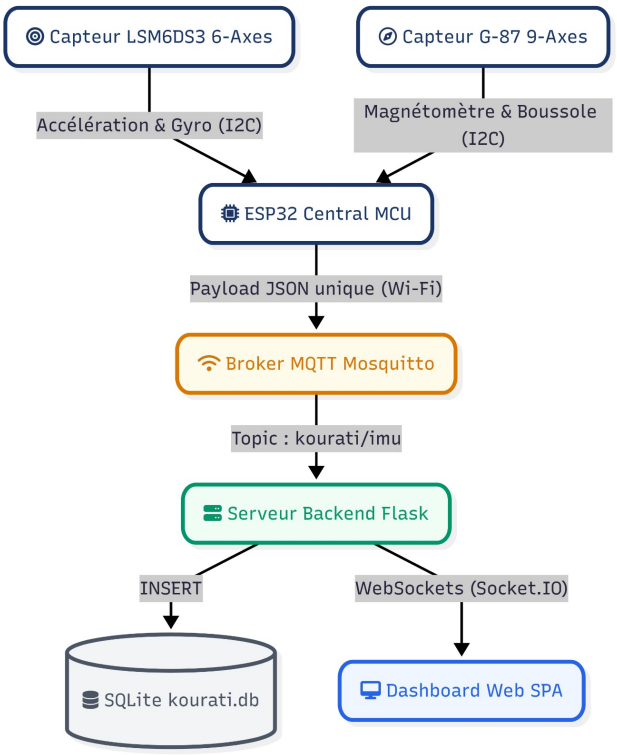


Figure 3.16 – Diagramme de flux de données KOURATI

3.3.3 Communication des Données avec MQTT

3.3.3.1 Présentation de MQTT

MQTT (Message Queuing Telemetry Transport) est un protocole de communication léger conçu pour les objets connectés [80]. Il fonctionne sur le modèle Publish/Subscribe : un émetteur publie un message sur un sujet (topic), et tous les abonnés à ce sujet le reçoivent automatiquement. Dans le domaine de l'Internet des Objets, MQTT s'est imposé comme le protocole de référence [89] grâce à sa faible empreinte en bande passante et sa compatibilité avec les microcontrôleurs à ressources limitées comme l'ESP32 [87].

3.3.3.2 Choix du protocole MQTT

MQTT a été retenu pour deux raisons principales. Premièrement, il est extrêmement léger et adapté aux microcontrôleurs comme l'ESP32 [87], qui dispose de ressources limitées. Deuxièmement, sa structure découplée permet à l'ESP32 de publier ses données sans se préoccuper de savoir si Flask est disponible ou non. Le niveau QoS 0 (au plus une fois) est utilisé pour privilégier la faible latence au détriment de la garantie de livraison, ce qui est adapté au streaming de données physiques [80].

3.3.3.3 Le rôle du broker Mosquitto

Le broker Eclipse Mosquitto [81] est l'intermédiaire central de communication. Installé localement, il écoute sur le port 1883 et assure la distribution des messages entre l'ESP32 — qui publie sur le topic `/kourati/ball/data` — et le serveur Flask qui s'est abonné à ce même topic. Mosquitto garantit que chaque message publié par le ballon est immédiatement transmis à Flask, sans perte et sans délai perceptible [81].

3.3.3.4 Réception par Flask et envoi vers le dashboard

Du côté serveur, le fichier `MQTT/mqtt_client.py` implémente un client MQTT utilisant la bibliothèque Paho-MQTT. Dès qu'un message arrive, la fonction `on_message()` est automatiquement déclenchée. Elle désérialise le message JSON [83], le soumet au pipeline de traitement (filtrage, filtre de Kalman [85], détection d'événements), puis diffuse les résultats vers le dashboard via WebSocket [82]. Tout ce processus s'effectue en moins de 30 ms.

3.3.4 Format JSON des Données

3.3.4.1 Présentation de JSON

JSON (JavaScript Object Notation) est un format texte standard pour représenter des données structurées, défini par l'IETF dans la RFC 8259 [83]. Il est lisible par les humains et facilement traitable par les machines. Dans le projet KOURATI, JSON est le format universel utilisé pour transporter les données entre l'ESP32 [87], le broker Mosquitto [81], le serveur Flask [90] et le dashboard.

3.3.4.2 Structure d'un message JSON du ballon

À chaque mesure, l'ESP32 [87] encapsule les données des capteurs — LSM6DS3 [86] (6 axes) et G-87 [88] (magnétomètre) — dans un objet JSON avant de le publier sur le broker. Flask enrichit ensuite ce message avec les valeurs calculées (vitesse via le filtre de Kalman [85], position estimée, orientation 3D [88,93], événements détectés). Voici un exemple représentatif :

```
{
        // --- LSM6DS3 (6 axes) ---
        "ax": 0.43,
        "ay": -0.12,
        "az": 9.81,
        "gx": 0.02,
        "gy": -0.01,
        "gz": 0.03,

        // --- G-87 -Magnétomètre (9 axes) ---
```

```
            "mx": 23.4,
            "my": -15.2,
            "mz": 41.7,
            "roll": 12.3,
            "pitch": -5.1,
            "yaw": 87.6,

            // --- Données calculées par Flask ---
            "speed": 12.4,
            "x": 67.3,
            "y": 48.9,
            "status": "in",
            "is_impact": true
    }
```

Listing 3.1 – Exemple d'un message JSON KOURATI incluant les données LSM6DS3, G-87 et les valeurs calculées

L'utilisation de JSON [83] présente plusieurs avantages : compatibilité universelle entre tous les langages de programmation, lecture native côté JavaScript dans le navigateur, et flexibilité pour ajouter de nouveaux champs (comme les données magnétométriques du G-87 [88]) sans modifier l'architecture existante.

3.3.5 API REST

3.3.5.1 Principe de l'API REST

Une API REST (Representational State Transfer), concept formalisé par Roy T. Fielding dans sa thèse de doctorat [84], est un ensemble d'adresses web (appelées endpoints) que le frontend peut interroger pour obtenir des informations spécifiques. Chaque endpoint correspond à une ressource précise et répond avec des données au format JSON [83]. Dans KOURATI, l'API REST est implémentée avec Flask [90] en utilisant le patron Blueprint pour organiser les routes par domaine fonctionnel.

3.3.5.2 REST vs communication temps réel

L'API REST fonctionne sur le modèle classique requête/réponse [84] : le navigateur demande, le serveur répond, puis la connexion se ferme. C'est utile pour charger des données initiales, mais inadapté à un flux de 20 mises à jour par seconde. C'est pourquoi KOURATI utilise WebSocket [82] pour les données en temps réel, et l'API REST pour les requêtes ponctuelles (chargement du dashboard, démarrage de match).

3.3.5.3 Endpoints principaux

Table 3.2 – Principaux endpoints REST du dashboard KOURATI

Endpoint	Format	Description
GET /	HTML	Page principale du dashboard (rendu Jinja2 avec données initiales)
GET /ball/live	JSON	État actuel du ballon : vitesse, accélération, position
GET /ball/position	JSON	Position x, y du ballon sur le terrain (%)
GET /ball/speed	JSON	Vitesse en m/s et accélération en m/s²
GET /stats/all	JSON	Toutes les statistiques du match en une seule requête
GET /stats/possession	JSON	Pourcentage de possession des deux équipes
GET /stats/shots	JSON	Nombre de tirs cadrés et non cadrés
GET /stats/score	JSON	Score actuel du match
POST /match/start	JSON	Démarre un nouveau match
POST /match/end	JSON	Termine le match en cours
GET /match/status	JSON	Statut et durée du match en cours
POST /ball/data	JSON	Réception des données IMU directement depuis l'ESP32 (mode HTTP)

Les endpoints `GET /ball/*` et `GET /stats/*` sont utilisés au premier chargement. Pendant le match, toutes les mises à jour passent par WebSocket [82] / Socket.IO [92].

3.3.6 Communication Temps Réel avec WebSocket

3.3.6.1 Présentation du WebSocket

Un WebSocket est une technologie qui établit une connexion permanente et bidirectionnelle entre un navigateur et un serveur, standardisée par l'IETF dans la RFC 6455 [82].

Contrairement à HTTP, où chaque échange nécessite une nouvelle connexion, le WebSocket maintient un canal ouvert pendant toute la durée de la session. Cela permet au serveur d'envoyer des données au navigateur à tout moment, sans attendre de requête.

3.3.6.2 Choix du WebSocket dans KOURATI

Le dashboard reçoit des données de position et de vitesse environ 20 fois par seconde. Utiliser des requêtes HTTP classiques signifierait 20 connexions TCP ouvertes et refermées chaque seconde. Avec WebSocket [82], une seule connexion est maintenue en permanence et le serveur envoie les données dès qu'elles sont disponibles, ce qui réduit la latence à moins de 30 ms en réseau local.

3.3.6.3 Broadcast via Socket.IO

Dans KOURATI, le serveur Flask [90] utilise la bibliothèque Flask-SocketIO [92] pour gérer les connexions WebSocket. Quand un événement se produit (nouvelle position du ballon, corner détecté, but marqué), le serveur émet un message vers tous les navigateurs connectés simultanément grâce au mécanisme de broadcast.

Trois types d'événements WebSocket sont utilisés :

- `ball_update` : position, vitesse, accélération et orientation du ballon ;
- `game_event` : corner, touche, tir ou but détecté ;
- `stats_update` : mise à jour périodique des statistiques du match.

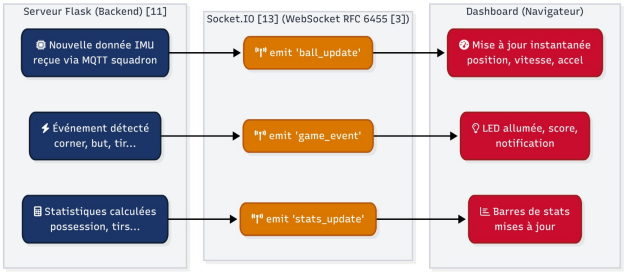


Figure 3.17 – Principe du broadcast WebSocket (RFC 6455 [82]) via Socket.IO [92] dans KOURATI.

3.3.7 Interface Utilisateur — Frontend

L'interface du dashboard a été conçue pour être lisible et réactive. Elle est organisée autour de plusieurs zones d'affichage qui se mettent à jour automatiquement dès que de nouvelles données arrivent via WebSocket [82].

3.3.7.1 Tableau de score et informations du match

En haut de l'interface se trouve le tableau de score avec les deux équipes, leurs logos, et le score mis à jour en temps réel à chaque but détecté. La liste des joueurs de chaque équipe est affichée latéralement avec leurs statistiques individuelles : passes réussies, précision et nombre de tirs. Ces données sont injectées directement dans la page HTML au premier chargement par le moteur de template Jinja2 de Flask, évitant tout appel API supplémentaire [90].

3.3.7.2 Indicateurs de statut et LED

Une barre d'indicateurs lumineux signale en permanence l'état du ballon. Un voyant vert indique que le ballon est en jeu. Des voyants spécifiques s'allument pour signaler un corner (ambre), une touche (bleu), une entrée en zone des 6 mètres (violet), ou un hors-terrain générique (rouge). Un grand LED jaune clignote lors d'un but, accompagné d'un flash visuel sur le tableau de score.

3.3.7.3 Terrain interactif et suivi du ballon

Au centre du dashboard, un terrain est rendu en temps réel via l'API Canvas HTML5. La position du ballon se déplace à chaque mise à jour WebSocket [82], avec un point orange en jeu et rouge hors terrain. Les 22 joueurs des deux équipes sont représentés avec leurs numéros, et les zones des 6 mètres sont mises en évidence par une coloration violette.

3.3.7.4 Statistiques comparées

La zone centrale présente les statistiques des deux équipes sous forme de barres de progression : possession de balle (%), passes réussies, tirs cadrés et non cadrés, corners et touches. Ces barres s'actualisent dynamiquement à chaque réception d'un événement `stats_update` via WebSocket [82].

3.3.7.5 Métriques en temps réel

Deux indicateurs numériques affichent en permanence la vitesse du ballon (m/s en régime normal, km/h lors d'une frappe pendant 2 secondes) et la magnitude de l'accélération (m/s^2). Ces valeurs proviennent du traitement des données IMU par le filtre de Kalman [85] côté serveur.

3.3.7.6 Orientation du ballon — Capteur G-87

Grâce au capteur G-87 à 9 axes [88], le dashboard peut afficher l'orientation complète du ballon dans l'espace. Les trois angles d'Euler — roulis (roll), tangage (pitch) et lacet (yaw) —

sont calculés en combinant les données du gyroscope, de l'accéléromètre et du magnétomètre selon un algorithme de fusion inertielle [88, 93].

Le magnétomètre fournit un cap absolu par rapport au nord magnétique, ce qui permet de corriger la dérive du gyroscope sur l'axe yaw. Cette information permet une analyse plus précise du mouvement du ballon, notamment l'effet appliqué lors des tirs.



Figure 3.18 – Interface du dashboard KOURATI.

3.3.8 Flux Complet des Données

Cette section décrit le parcours complet des données depuis leur acquisition jusqu'à leur affichage.

1. Acquisition des données : le capteur LSM6DS3 [86] mesure l'accélération et la rotation à 50 Hz. Le G-87 [88] ajoute les données magnétométriques permettant le calcul de l'orientation 3D.
2. Envoi MQTT : l'ESP32 [87] encapsule les données dans un message JSON et les publie vers Mosquitto [81].
3. Réception Flask : le client MQTT reçoit les données via `on_message()`.
4. Traitement : filtrage numérique et filtre de Kalman [85], puis détection d'événements.
5. Encapsulation JSON : création d'un objet structuré contenant position, vitesse et état du ballon.

6. Diffusion WebSocket : émission via Socket.IO [92].
7. Affichage : mise à jour en temps réel du dashboard.

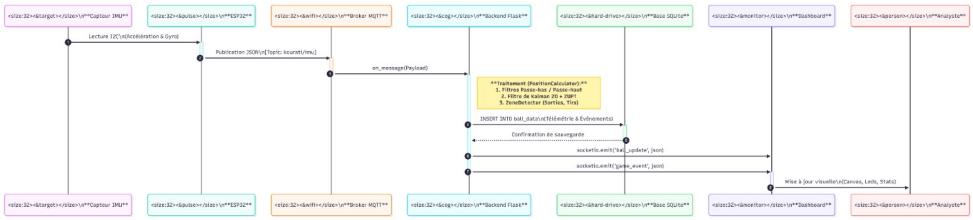


Figure 3.19 – Diagramme de Séquence UML (PlantUML)

3.4 Tests et validation expérimentale

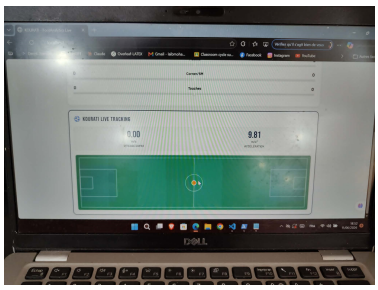
Après la réalisation du prototype du ballon instrumenté et la mise en place du tableau de bord KOURATI, une phase de validation expérimentale a été menée afin d'évaluer le fonctionnement global du système. L'objectif de cette partie n'est pas de tester séparément chaque composant électronique, mais d'observer directement la cohérence entre le déplacement réel du ballon dans l'espace expérimental et sa représentation sur l'interface de supervision.

Dans ce cadre, deux essais principaux ont été réalisés. Le premier essai a été effectué avec un repère physique matérialisé par une croix noire au sol, servant de référence visuelle pour les positions du ballon. Le second essai a été réalisé sans ce repère central, afin d'évaluer la reproductibilité du système dans des conditions légèrement différentes. Dans les deux cas, le ballon a été placé successivement dans trois positions notées A, B et C. Les positions réelles ont ensuite été comparées avec les positions affichées sur le tableau de bord KOURATI.

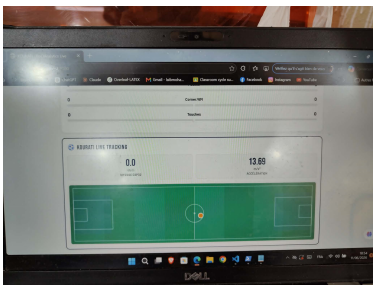
Afin de faciliter la comparaison, les résultats sont présentés sous forme de figures comparatives. Pour chaque essai, la ligne supérieure contient les captures du tableau de bord correspondant aux positions A, B et C, tandis que la ligne inférieure contient les photographies réelles du ballon dans les mêmes positions. Cette organisation permet de comparer directement la position physique du ballon avec sa représentation numérique sur l'interface.

3.4.1 Premier essai

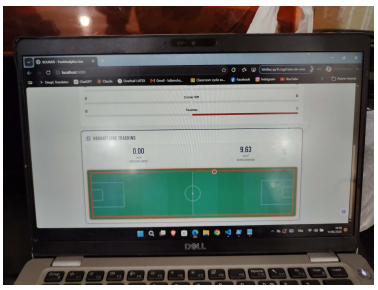
Le premier essai a été réalisé avec un repère visuel placé au sol sous forme d'une croix noire. Ce repère permet de stabiliser l'observation de la zone de test et de faciliter la comparaison entre la position réelle du ballon et la position affichée sur le tableau de bord. Le ballon a été déplacé successivement de la position A vers la position B, puis de la position B vers la position C.



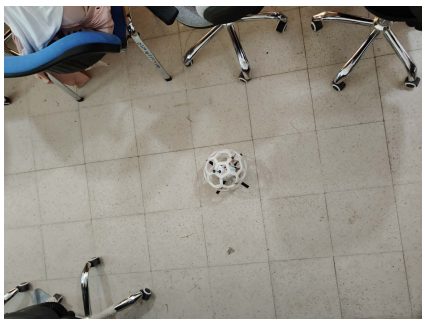
(a) Dashboard : position A



(b) Dashboard : position B



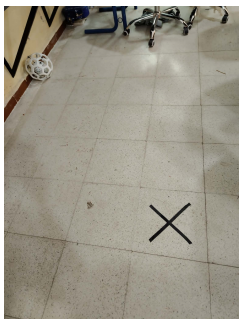
(c) Dashboard : position C



(d) Position réelle A



(e) Position réelle B



(f) Position réelle C

Figure 3.20 – Comparaison des positions lors du premier essai

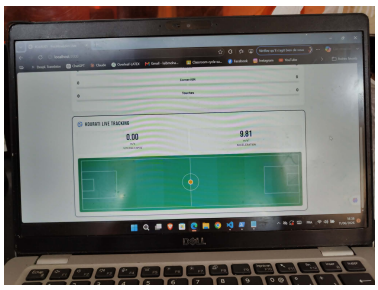
Source : réalisé par l'auteur.

Les résultats du premier essai montrent une bonne cohérence entre la position réelle du ballon et la position affichée sur le tableau de bord. Lorsque le ballon est placé successivement dans les positions A, B et C, le point représentant le ballon sur l'interface évolue de manière cohérente. La trajectoire observée respecte l'ordre du déplacement réel et forme un parcours triangulaire logique entre les trois positions.

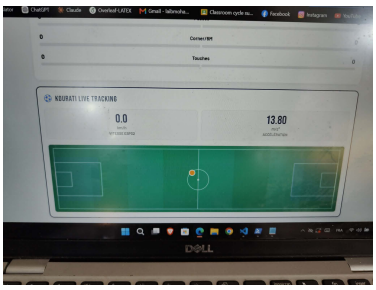
Ce premier essai permet donc de valider le principe général du système complet. Le prototype est capable de transmettre les données vers le tableau de bord et de représenter le déplacement du ballon d'une manière visuellement cohérente. Dans ces conditions, la correspondance entre l'espace réel et l'espace virtuel peut être considérée comme satisfaisante.

3.4.2 Deuxième essai

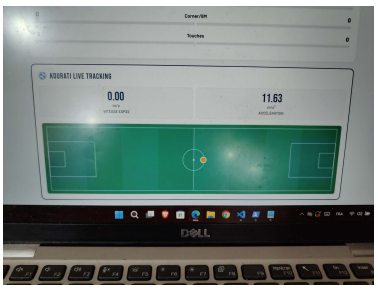
Le deuxième essai a été réalisé sans le repère noir central utilisé lors du premier test. L'objectif de cette répétition est d'évaluer la stabilité du système lorsque la référence visuelle au sol est absente. Le ballon a de nouveau été déplacé entre trois positions A, B et C, puis les résultats ont été comparés avec l'affichage du tableau de bord.



(a) Dashboard : position A



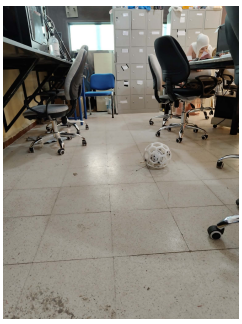
(b) Dashboard : position B



(c) Dashboard : position C



(d) Position réelle A



(e) Position réelle B



(f) Position réelle C

Figure 3.21 – Comparaison des positions lors du deuxième essai

Source : réalisé par l'auteur.

Les observations du deuxième essai montrent que la forme globale de la trajectoire reste correcte. Le déplacement entre les positions A, B et C conserve une logique triangulaire identifiable sur le tableau de bord. Cela signifie que le système reste capable de suivre les variations relatives du déplacement du ballon.

Cependant, une différence apparaît au niveau de la position absolue affichée. La correspondance entre la position réelle du ballon et sa représentation sur le terrain virtuel n'est plus parfaitement alignée. Les axes semblent avoir changé ou été inversés par rapport au repère réel. Ainsi, même si la trajectoire générale reste correcte, la projection du déplacement sur le tableau de bord devient moins stable d'un essai à l'autre.

3.4.3 Validation de la détection des événements de jeu

En plus du suivi de la trajectoire du ballon, les essais expérimentaux ont permis d'observer la capacité du tableau de bord KOURATI à interpréter certaines situations de jeu. L'objectif de cette fonctionnalité est de ne pas se limiter à l'affichage des grandeurs physiques, comme l'accélération ou la vitesse, mais de transformer les données reçues en informations footballistiques compréhensibles par l'utilisateur.

La détection des événements repose sur une logique de traitement basée principalement sur la position estimée du ballon dans le terrain virtuel. Lorsque le ballon dépasse certaines limites définies dans l'interface, le système peut interpréter cette situation comme un

événement de jeu. Par exemple, une sortie latérale du ballon peut être associée à une touche, tandis qu'une sortie dans une zone proche du but peut être associée à un corner ou à un dégagement des six mètres. De la même manière, le franchissement d'une zone correspondant à la ligne de but peut être utilisé pour signaler un but et mettre à jour le score.

Lors des essais réalisés, l'événement de touche a été observé sur le tableau de bord. Lorsque la position estimée du ballon atteint la limite latérale du terrain virtuel, le compteur correspondant aux touches est mis à jour. Cette observation confirme que le système est capable de convertir une condition de position en événement de jeu affiché sur l'interface.

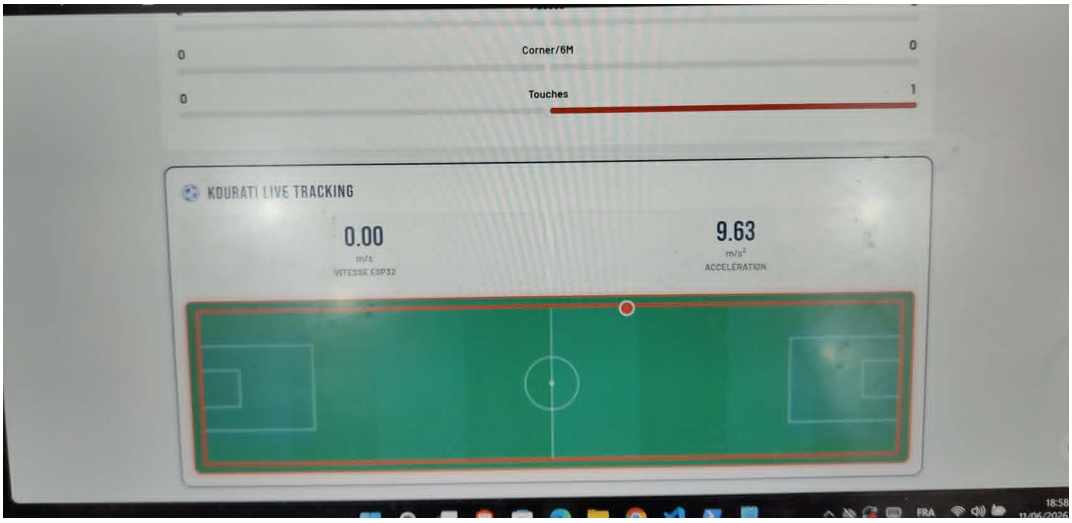


Figure 3.22 – Détection de l'événement de touche sur le tableau de bord KOURATI.

Source : capture réalisée par l'auteur.

La figure 3.22 montre un exemple de détection d'une touche. Cette capture permet de valider expérimentalement la réaction du tableau de bord lorsqu'un événement lié à la sortie latérale du ballon est détecté. Elle montre également que l'interface ne se limite pas à afficher la position du ballon, mais permet aussi de suivre certains événements liés au déroulement du jeu.

Il convient cependant de préciser que la validation expérimentale complète des autres événements, tels que le corner, le dégagement des six mètres et le but, nécessite des essais supplémentaires dans des conditions mieux définies. Ces événements sont prévus dans la logique de traitement du tableau de bord, mais leur fiabilité dépend directement de la précision du positionnement, de la stabilité du repère spatial et des seuils utilisés dans le programme.

Ainsi, les résultats obtenus montrent que le principe de détection des événements est fonctionnel, notamment à travers l'exemple de la touche. Toutefois, pour obtenir une détection plus robuste de l'ensemble des événements footballistiques, il serait nécessaire d'améliorer la précision de la localisation du ballon et de réaliser une série de tests plus

complète couvrant les différentes zones du terrain.

3.4.4 Analyse des résultats expérimentaux

L'analyse des deux essais met en évidence deux points essentiels. Le premier est que le système complet permet de suivre le déplacement relatif du ballon et de restituer une trajectoire cohérente sur le tableau de bord. Cette capacité constitue une validation fonctionnelle importante du prototype, car elle montre que la chaîne complète, composée du ballon instrumenté, de la transmission IoT, du traitement logiciel et de l'interface KOURATI, fonctionne correctement.

Le deuxième point concerne la stabilité du repère spatial. Le premier essai, réalisé avec un repère noir au sol, présente une meilleure correspondance entre les positions réelles et les positions affichées. Le deuxième essai, réalisé sans repère central, conserve la forme triangulaire de la trajectoire, mais présente une erreur dans l'orientation ou l'interprétation des axes. Cette différence montre que le système est capable de suivre les variations de déplacement, mais que la position absolue reste sensible aux conditions d'initialisation.

Cette limite peut être expliquée par la nature des capteurs utilisés et par l'absence d'un système de localisation absolue. Les capteurs inertiels permettent de mesurer les accélérations et les rotations, mais leur utilisation pour estimer une position peut entraîner des erreurs cumulées, notamment à cause du bruit, des dérives et de la calibration. Dans ce prototype, la position affichée sur le tableau de bord doit donc être interprétée comme une position estimée ou représentative, et non comme une localisation absolue précise.

Conclusion

Ce chapitre a permis de valider expérimentalement le principe général du ballon instrumenté connecté à un tableau de bord IoT. Le prototype réalisé a montré qu'il est possible d'acquérir des données à partir des capteurs embarqués, de les transmettre vers la plateforme KOURATI et de les afficher sous forme visuelle sur le tableau de bord.

Les résultats obtenus montrent que le système fonctionne correctement dans certaines conditions. Le premier essai, réalisé avec un repère physique au sol, a donné une bonne correspondance entre les positions réelles A, B et C du ballon et les positions affichées sur l'interface. La trajectoire triangulaire a été correctement représentée, ce qui confirme la cohérence de la chaîne complète allant du ballon instrumenté jusqu'au tableau de bord.

Cependant, les essais suivants ont montré une limite importante du système. Lors du deuxième test, réalisé sans repère central, ainsi que lors de certaines répétitions, une erreur de positionnement est apparue. Même si la forme générale de la trajectoire restait correcte, la position affichée sur le tableau de bord ne correspondait plus exactement à la position réelle du ballon. Dans certains cas, les axes semblaient changés ou inversés, ce qui montre que le système reste sensible aux conditions d'initialisation, au repère utilisé et à l'orientation initiale du ballon.

Cette limite est principalement liée aux capteurs et aux composants utilisés dans ce prototype. Les capteurs inertiels, comme le LSM6DS3 et le GY-87, permettent de mesurer l'accélération, la rotation et l'orientation, mais ils ne permettent pas d'obtenir une localisation absolue précise du ballon dans l'espace. Leur utilisation pour estimer une position peut provoquer des erreurs cumulées, dues au bruit, à la dérive, à la calibration et aux perturbations du magnétomètre.

Ainsi, les erreurs observées ne remettent pas en cause le principe du système, mais elles montrent que les composants utilisés ne sont pas suffisants pour garantir un positionnement stable et reproductible dans toutes les conditions. Pour améliorer la précision, il serait nécessaire d'intégrer un système de localisation absolue, comme l'UWB, un système de vision par caméra, des balises de positionnement ou une centrale inertielle plus performante associée à un algorithme de fusion de données plus avancé.

Dans le cadre de ce travail, le choix des composants a également été limité par des contraintes de coût et de disponibilité locale. Certains modules spécialisés nécessaires pour améliorer la localisation du ballon ne sont pas facilement accessibles sur le marché local ou nécessitent des coûts plus élevés. Le prototype réalisé représente donc un compromis entre faisabilité technique, disponibilité des composants, coût de réalisation et objectifs expérimentaux du projet.

Les essais ont aussi permis d'observer la détection d'un événement de jeu, notamment la touche, à travers la mise à jour du compteur correspondant sur le tableau de bord. Cette fonctionnalité montre que le système peut transformer certaines conditions de position en informations footballistiques exploitables. Toutefois, la validation complète d'autres événements, comme le corner, le dégagement des six mètres ou le but, nécessite des essais supplémentaires avec une meilleure précision de positionnement.

En conclusion, le prototype développé constitue une validation fonctionnelle expérimentale du concept proposé. Il permet de démontrer la faisabilité d'un ballon instrumenté capable de transmettre ses données vers une plateforme IoT, d'afficher une trajectoire sur un tableau de bord et de signaler certains événements du jeu. Néanmoins, la précision du positionnement reste la principale limite du système actuel. Une version future devra donc intégrer un meilleur recalage des axes, une calibration plus robuste et surtout un système de localisation absolue plus adapté afin d'obtenir des résultats plus fiables et plus proches d'une utilisation sportive réelle.

Conclusion générale

Ce mémoire a porté sur la conception et la réalisation d'un ballon de football intelligent connecté associé à une plateforme IoT pour l'analyse du jeu. L'objectif principal était de proposer une solution expérimentale capable de collecter des données liées au comportement du ballon, de les transmettre vers un environnement numérique et de les présenter sous une forme exploitable par l'utilisateur.

Le travail réalisé a permis de répondre, de manière fonctionnelle, à la problématique posée au début du mémoire. En effet, il a été possible de concevoir une chaîne complète allant de l'acquisition des données par des capteurs embarqués jusqu'à leur visualisation sur un tableau de bord web. Cette chaîne comprend la partie mécanique du prototype, l'intégration des composants électroniques, le traitement embarqué par l'ESP32-S3, la transmission des données à travers une architecture IoT et l'affichage des informations sur le tableau de bord KOURATI.

L'hypothèse principale du travail est donc validée sur le plan fonctionnel. Les essais réalisés montrent qu'un ballon équipé d'un système électronique embarqué peut fournir des informations utiles sur son comportement, notamment le mouvement, l'orientation, les impacts et certaines situations de jeu. Le système développé a également permis de visualiser une trajectoire estimée du ballon et d'observer la détection d'un événement de jeu, comme la touche, à travers la mise à jour du tableau de bord. Ces résultats confirment l'intérêt d'associer un dispositif embarqué à une plateforme IoT pour transformer des mesures physiques en informations plus compréhensibles dans un contexte sportif.

Le premier chapitre a permis de situer le projet dans le contexte du football connecté et des ballons instrumentés. L'étude des solutions existantes a montré que les systèmes professionnels reposent généralement sur des capteurs embarqués, des technologies de localisation, des plateformes de traitement et des interfaces de visualisation. Cette étude a également montré que ces solutions restent souvent coûteuses, complexes et difficiles à reproduire dans un contexte académique. Cela a justifié l'intérêt de proposer une solution plus accessible, basée sur des composants disponibles et une architecture ouverte.

Le deuxième chapitre a permis de poser les bases théoriques du projet. La modélisation physique du ballon a montré l'influence des principales forces appliquées sur son mouvement, notamment la force de tir, le poids, la réaction du sol, les frottements, la traînée aérodynamique et l'effet Magnus. La modélisation électronique a ensuite permis d'expliquer le rôle des capteurs, du micro-contrôleur et du traitement des données. La simulation sous Proteus a validé le principe général d'acquisition et de traitement, même si certains

composants réels ont été remplacés par des équivalents disponibles dans l'environnement de simulation.

Le troisième chapitre a présenté la réalisation pratique du système. Le prototype a été réalisé progressivement à travers plusieurs sous-systèmes : la structure mécanique, la capsule interne, l'alimentation, le circuit d'acquisition et l'intégration finale. Cette démarche a permis de passer d'une architecture théorique à un prototype physique fonctionnel. Le tableau de bord KOURATI a ensuite été développé pour assurer la visualisation des données et la supervision du ballon. L'utilisation de MQTT, Mosquitto, Flask, WebSocket et JSON a permis de construire une architecture IoT modulaire, capable de transmettre les données et de mettre à jour l'interface en temps réel.

Les tests expérimentaux ont permis de vérifier le fonctionnement global du système. Le premier essai, réalisé avec un repère physique au sol, a donné des résultats satisfaisants. Les positions réelles A, B et C du ballon étaient globalement cohérentes avec les positions affichées sur le tableau de bord, et la trajectoire triangulaire a été correctement représentée. Ce résultat confirme que le système peut fonctionner correctement dans des conditions bien définies et que la chaîne complète d'acquisition, de transmission, de traitement et d'affichage est opérationnelle.

Cependant, les essais suivants ont montré une limite importante. Lors du deuxième test, réalisé sans repère central, ainsi que lors de certaines répétitions, une erreur de positionnement est apparue. Même si la forme générale de la trajectoire restait cohérente, la position affichée sur le tableau de bord ne correspondait plus exactement à la position réelle du ballon. Dans certains cas, les axes semblaient changés ou inversés. Ce résultat montre que le système est capable de suivre une trajectoire relative, mais qu'il ne garantit pas encore une localisation absolue stable et reproductible.

Cette limite est principalement liée à la nature des capteurs et composants utilisés. Les capteurs inertiels, comme le LSM6DS3 et le GY-87, permettent de mesurer l'accélération, la rotation et l'orientation, mais ils ne fournissent pas directement une position absolue précise du ballon sur le terrain. Leur utilisation pour estimer une position peut provoquer des erreurs cumulées dues au bruit, à la dérive, à la calibration et aux perturbations du magnétomètre. De même, le capteur FSR402 permet de détecter un effort ou un impact, mais il ne permet pas de localiser précisément le ballon dans l'espace de jeu.

Ainsi, les résultats obtenus doivent être interprétés comme une validation fonctionnelle expérimentale, et non comme une validation définitive de la précision de localisation. Le prototype confirme la faisabilité du concept, mais il montre également que les composants utilisés ne sont pas suffisants pour garantir un positionnement fiable dans toutes les conditions. Pour atteindre un niveau de précision plus élevé, il serait nécessaire d'intégrer un système de localisation absolue, comme l'UWB, un système de vision par caméra, des balises de positionnement ou une centrale inertielle plus performante associée à un algorithme de fusion de données plus avancé.

Dans le cadre de ce projet, le choix des composants a aussi été influencé par des contraintes pratiques liées au coût et à la disponibilité locale du matériel. Certains modules spécialisés nécessaires pour améliorer la localisation du ballon ne sont pas facilement accessibles sur le

marché local ou nécessitent des coûts plus élevés. Le prototype réalisé représente donc un compromis entre faisabilité technique, accessibilité des composants, coût de réalisation et objectifs expérimentaux du projet.

En conclusion, ce travail a permis de démontrer qu'il est possible de concevoir un ballon instrumenté connecté capable de collecter des données, de les transmettre vers une plateforme IoT et de les afficher sur un tableau de bord. L'objectif général du mémoire est donc atteint sur le plan de la faisabilité et de la validation fonctionnelle. Néanmoins, la précision du positionnement reste la principale limite du système actuel. Les perspectives d'amélioration concernent principalement le recalage des axes, la calibration initiale, la robustesse mécanique, l'intégration d'un système de localisation plus précis et la validation expérimentale d'un plus grand nombre d'événements de jeu. Ces améliorations permettraient de faire évoluer le prototype vers une solution plus fiable, plus robuste et plus proche d'une utilisation sportive réelle.

Bibliography

[1] H. Sarmento, R. Marcelino, M. T. Anguera, J. Campaniço, N. Matos, and J. C. Leitão, "Match analysis in football: A systematic review," *Journal of Sports Sciences*, vol. 32, no. 20, pp. 1831–1843, 2014, doi: 10.1080/02640414.2014.898852.

[2] R. Rein and D. Memmert, "Big data and tactical analysis in elite soccer: Future challenges and opportunities for sports science," *SpringerPlus*, vol. 5, no. 1, Art. no. 1410, 2016, doi: 10.1186/s40064-016-3108-2.

[3] The IFAB, "Law 9: The Ball in and out of Play," *Laws of the Game*. [En ligne]. Disponible : https://www.theifab.com/fr/laws/latest/the-ball-in-and-out-of-play/. [Consulté le : 08 mai 2026].

[4] The IFAB, "Law 10: Determining the Outcome of a Match," *Laws of the Game*. [En ligne]. Disponible : https://www.theifab.com/fr/laws/latest/determining-the-outcome-of-a-match/. [Consulté le : 08 mai 2026].

[5] The IFAB, "Law 17: The Corner Kick," *Laws of the Game*. [En ligne]. Disponible : https://www.theifab.com/fr/laws/latest/the-corner-kick/. [Consulté le : 08 mai 2026].

[6] FIFA, "Electronic Performance and Tracking Systems," FIFA Quality Programme. [En ligne]. Disponible : https://inside.fifa.com/innovation/standards/epts. [Consulté le : 08 mai 2026].

[7] FIFA, "Connected ball technology," FIFA Innovation. [En ligne]. Disponible : https://inside.fifa.com/innovation/innovating-the-game/connected-ball-technology. [Consulté le : 08 mai 2026].

[8] FIFA, "Semi-automated offside technology," FIFA Innovation. [En ligne]. Disponible : https://inside.fifa.com/innovation/world-cup-2022/semi-automated-offside-technology. [Consulté le : 08 mai 2026].

[9] FIFA, "TRIONDA – FIFA World Cup 2026 Official Match Ball," FIFA. [En ligne]. Disponible : https://www.fifa.com/en/tournaments/mens/worldcup/canadamexicousa2026/official-match-ball. [Consulté le : 08 mai 2026].

[10] adidas, "adidas reveals the first FIFA World Cup official match ball featuring connected ball technology," adidas News, Jul. 2022. [En ligne]. Disponible : https://news.adidas.

com/football/adidas-reveals-the-first-fifa-world-cup--official-match-ball-featuring-connected-ball-technology/s/cccb7187-a67c-4166-b57d-2b28f1d36fa0. [Consulté le : 08 mai 2026].

[11] FOX Sports, "2026 World Cup Ball Revealed! Inside the Details of TRIONDA," FOX Sports. [En ligne]. Disponible : https://www.foxnews.com/sports/trionda-adidas-world-cup-2026-ball-usa-mexico-canada. [Consulté le : 08 mai 2026].

[12] KINEXON Sports, "Premier Handball and Soccer Ball Tracking," KINEXON Sports. [En ligne]. Disponible : https://kinexon-sports.com/products/xball. [Consulté le : 08 mai 2026].

[13] KINEXON Sports, "Pioneering Sports Tracking and Analytics," KINEXON Sports. [En ligne]. Disponible : https://kinexon-sports.com. [Consulté le : 08 mars 2026].

[14] KINEXON Sports, "Experience Live Handball and Football Tracking," KINEXON Sports. [En ligne]. Disponible : https://kinexon-sports.com/technology/ball-tracking. [Consulté le : 08 mai 2026].

[15] ITU-T, *Recommendation ITU-T Y.2060: Overview of the Internet of Things*, International Telecommunication Union, Jun. 2012. [En ligne]. Disponible : https://www.itu.int/rec/T-REC-Y.2060-201206-I. [Consulté le : 08 mai 2026].

[16] ThingsBoard, "ThingsBoard Open-source IoT Platform." [En ligne]. Disponible : https://thingsboard.io/. [Consulté le : 08 mai 2026].

[17] ThingsBoard, "Dashboards," ThingsBoard Documentation. [En ligne]. Disponible : https://thingsboard.io/docs/pe/user-guide/dashboards/. [Consulté le : 08 mai 2026].

[18] Google Firebase, "Firebase Realtime Database," Firebase Documentation. [En ligne]. Disponible : https://firebase.google.com/docs/database. [Consulté le : 08 mai 2026].

[19] Hudl, "WIMU Pro," Hudl. [En ligne]. Disponible : https://www.hudl.com/products/wimu. [Consulté le : 08 mai 2026].

[20] Hudl Support, "WIMU FAQ," Hudl Support. [En ligne]. Disponible : https://support.hudl.com/. [Consulté le : 08 mai 2026].

[21] STMicroelectronics, "LSM6DS3TR-C: iNEMO inertial module: always-on 3D accelerometer and 3D gyroscope," Datasheet, 2017. [En ligne]. Disponible : https://www.st.com/resource/en/datasheet/lsm6ds3tr-c.pdf. [Consulté le : 08 mars 2026].

[22] Z. Wang, X. Yang, Y. Zhang, and J. Li, "Motion analysis of football kick based on an IMU sensor," *Sensors*, vol. 22, no. 16, Art. no. 6244, 2022, doi: 10.3390/s22166244.

[23] TE Connectivity, "Piezoelectric Sensors: Film, Cable and Elements." [En ligne]. Disponible : https://www.te.com/en/products/sensors/piezo-film-sensors.html. [Consulté le : 08 mars 2026].

[24] Interlink Electronics, “FSR 402,” Interlink Electronics. [En ligne]. Disponible : https://www.interlinkelectronics.com/fsr-402. [Consulté le : 08 mars 2026].

[25] Interlink Electronics, “FSR 400 Series Data Sheet,” Datasheet. [En ligne]. Disponible : https://cdn.sparkfun.com/assets/8/a/1/2/0/2010-10-26-DataSheet-FSR402-Layout2.pdf. [Consulté le : 08 mars 2026].

[26] Qorvo, “DW1000 Ultra-Wideband Transceiver IC.” [En ligne]. Disponible : https://www.qorvo.com/products/d/da007946. [Consulté le : 08 mars 2026].

[27] Qorvo, “DWM3000 Ultra-Wideband Transceiver Module.” [En ligne]. Disponible : https://www.qorvo.com/products/d/da008334. [Consulté le : 08 mars 2026].

[28] Espressif Systems, “ESP32: Wi-Fi & Bluetooth SoC,” Espressif. [En ligne]. Disponible : https://www.espressif.com/en/products/socs/esp32. [Consulté le : 08 mai 2026].

[29] Espressif Systems, “ESP32-WROOM-32 Datasheet,” Datasheet. [En ligne]. Disponible : https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf. [Consulté le : 08 mai 2026].

[30] Espressif Systems, *ESP32 Technical Reference Manual*. [En ligne]. Disponible : https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf. [Consulté le : 08 mai 2026].

[31] OASIS, “MQTT Version 5.0,” OASIS Standard, 2019. [En ligne]. Disponible : https://www.oasis-open.org/standard/mqtt-v5-0-os/. [Consulté le : 08 mai 2026].

[32] MQTT.org, “MQTT: The Standard for IoT Messaging.” [En ligne]. Disponible : https://mqtt.org/. [Consulté le : 08 mai 2026].

[33] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, 3rd ed. Upper Saddle River, NJ, USA: Pearson, 2010.

[34] R. G. Lyons, *Understanding Digital Signal Processing*, 3rd ed. Upper Saddle River, NJ, USA: Prentice Hall, 2011.

[35] M. Barr and A. Massa, *Programming Embedded Systems: With C and GNU Development Tools*, 2nd ed. Sebastopol, CA, USA: O’Reilly Media, 2006.

[36] Adafruit, “Adafruit LSM6DS3TR-C + LIS3MDL - Precision 9 DoF IMU,” Adafruit Industries. [En ligne]. Disponible : https://www.adafruit.com/product/5543. [Consulté le : 08 mai 2026].

[37] Adafruit, “Adafruit LSM6DSOX + LIS3MDL - Precision 9 DoF IMU,” Adafruit Industries. [En ligne]. Disponible : https://www.adafruit.com/product/4517. [Consulté le : 08 mai 2026].

[38] Adafruit, “Adafruit 9-DOF Absolute Orientation IMU Fusion Breakout - BNO055,” Adafruit Industries. [En ligne]. Disponible : https://www.adafruit.com/product/2472. [Consulté le : 08 mai 2026].

[39] SparkFun Electronics, “SparkFun 9DoF IMU Breakout - ICM-20948 Qwiic,” SparkFun. [En ligne]. Disponible : https://www.sparkfun.com/sparkfun-9dof-imu-breakout-icm-20948-qwiic.html. [Consulté le : 08 mai 2026].

[40] T. Locoduino, “Quel microcontrôleur choisir ?,” *LOCODUINO*, 26 nov. 2024. [En ligne]. Disponible : https://www.locoduino.org/spip.php?article303. [Consulté le : 08 mai 2026].

[41] ThingsBoard, “Open-source IoT Platform,” ThingsBoard. [En ligne]. Disponible : https://thingsboard.io [Consulté le : 09 mai 2026].

[42] ThingsBoard, “ThingsBoard Architecture Reference,” ThingsBoard Documentation. [En ligne]. Disponible : https://thingsboard.io/docs/reference/ [Consulté le : 09 mai 2026].

[43] ThingsBoard, “Rule Engine Documentation,” ThingsBoard Documentation. [En ligne]. Disponible : https://thingsboard.io/docs/user-guide/rule-engine/ [Consulté le : 09 mai 2026].

[44] R. C. Hibbeler, *Engineering Mechanics: Dynamics*, 14th ed. Hoboken, NJ, USA: Pearson, 2016.

[45] P. Flores, M. Machado, M. T. Silva, and J. M. Martins, “On the continuous contact force models for soft materials in multibody dynamics,” *Multibody System Dynamics*, vol. 25, no. 3, pp. 357–375, 2011, doi: 10.1007/s11044-010-9237-4.

[46] J. Zhang, W. Li, L. Zhao, and G. He, “A continuous contact force model for impact analysis in multibody dynamics,” *Mechanism and Machine Theory*, vol. 153, Art. no. 103946, 2020, doi: 10.1016/j.mechmachtheory.2020.103946.

[47] K. Bray and D. G. Kerwin, “Modelling the flight of a soccer ball in a direct free kick,” *Journal of Sports Sciences*, vol. 21, no. 2, pp. 75–85, 2003, doi: 10.1080/0264041031000071009.

[48] T. Asai, K. Seo, O. Kobayashi, and R. Sakashita, “Fundamental aerodynamics of the soccer ball,” *Sports Engineering*, vol. 10, no. 2, pp. 101–110, 2007, doi: 10.1007/BF02844207.

[49] J. E. Goff and M. J. Carré, “Trajectory analysis of a soccer ball,” *American Journal of Physics*, vol. 77, no. 11, pp. 1020–1027, 2009, doi: 10.1119/1.3197187.

[50] T. Kray, M. Franke, and T. W. Nitsche, “Magnus effect on a rotating soccer ball at high Reynolds numbers,” *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 124, pp. 46–53, 2014, doi: 10.1016/j.jweia.2013.10.010.

[51] Interlink Electronics, “FSR 402,” Interlink Electronics. [En ligne]. Disponible : https://www.interlinkelectronics.com/fsr-402. [Consulté le : 10 mars 2026].

[52] Interlink Electronics, “FSR 400 Series Data Sheet,” Datasheet. [En ligne]. Disponible : https://cdn.sparkfun.com/assets/8/a/1/2/0/2010-10-26-DataSheet-FSR402-Layout2.pdf. [Consulté le : 10 mars 2026].

[53] TE Connectivity, “Piezoelectric Sensors: Film, Cable and Elements,” TE Connectivity. [En ligne]. Disponible : https://www.te.com/en/products/sensors/piezo-film-sensors.html. [Consulté le : 10 mars 2026].

[54] STMicroelectronics, “LSM6DS3TR-C: iNEMO inertial module: always-on 3D accelerometer and 3D gyroscope,” Datasheet, 2017. [En ligne]. Disponible : https://www.st.com/resource/en/datasheet/lsm6ds3tr-c.pdf. [Consulté le : 10 mars 2026].

[55] Espressif Systems, “ESP32 Series Datasheet,” Datasheet. [En ligne]. Disponible : https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf. [Consulté le : 10 mars 2026].

[56] Espressif Systems, “ESP32-WROOM-32 Datasheet,” Datasheet. [En ligne]. Disponible : https://www.espressif.com/sites/default/files/documentation/esp32-wroom-32_datasheet_en.pdf. [Consulté le : 10 mars 2026].

[57] Espressif Systems, *ESP32 Technical Reference Manual*. [En ligne]. Disponible : https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf. [Consulté le : 11 mars 2026].

[58] Battery University, “BU-808: How to Prolong Lithium-based Batteries.” [En ligne]. Disponible : https://batteryuniversity.com/article/bu-808-how-to-prolong-lithium-based-batteries. [Consulté le : 11 mars 2026].

[59] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, 3rd ed. Upper Saddle River, NJ, USA: Pearson, 2010.

[60] R. G. Lyons, *Understanding Digital Signal Processing*, 3rd ed. Upper Saddle River, NJ, USA: Prentice Hall, 2011.

[61] M. Barr and A. Massa, *Programming Embedded Systems: With C and GNU Development Tools*, 2nd ed. Sebastopol, CA, USA: O’Reilly Media, 2006.

[62] Labcenter Electronics, “Proteus VSM for Arduino,” Labcenter Electronics. [En ligne]. Disponible : https://www.labcenter.com/. [Consulté le : 08 mars 2026].

[63] Labcenter Electronics, “Circuit Simulation Software with SPICE,” Labcenter Electronics. [En ligne]. Disponible : https://www.labcenter.com/simulation/. [Consulté le : 08 mars 2026].

[64] Arduino, “UNO R3,” Arduino Documentation. [En ligne]. Disponible : https://docs.arduino.cc/hardware/uno-rev3. [Consulté le : 08 mars 2026].

[65] Interlink Electronics, “FSR 402,” Interlink Electronics. [En ligne]. Disponible : https://www.interlinkelectronics.com/fsr-402. [Consulté le : 08 mars 2026].

[66] Interlink Electronics, “FSR 400 Series Data Sheet,” Datasheet. [En ligne]. Disponible : https://cdn.sparkfun.com/assets/8/a/1/2/0/2010-10-26-DataSheet-FSR402-Layout2.pdf. [Consulté le : 08 mars 2026].

[67] STMicroelectronics, "LSM6DS3TR-C: iNEMO inertial module: always-on 3D accelerometer and 3D gyroscope," Datasheet, 2017. [En ligne]. Disponible : https://www.st.com/resource/en/datasheet/lsm6ds3tr-c.pdf. [Consulté le : 08 mars 2026].

[68] TDK InvenSense, "MPU-6050: Six-Axis Gyroscope and Accelerometer," TDK InvenSense. [En ligne]. Disponible : https://invensense.tdk.com/products/motion-tracking/6-axis/mpu-6050/. [Consulté le : 08 mars 2026].

[69] Espressif Systems, "ESP32 Series Datasheet," Datasheet. [En ligne]. Disponible : https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf. [Consulté le : 08 mars 2026].

[70] I. Gibson, D. Rosen, B. Stucker, and M. Khorasani, *Additive Manufacturing Technologies*, 3rd ed. Cham, Switzerland: Springer, 2021.

[71] UltiMaker, "What is FDM 3D printing?," UltiMaker. [En ligne]. Disponible : https://ultimaker.com/learn/what-is-fdm-3d-printing/. [Consulté le : 15 mars 2026].

[72] UltiMaker, "PLA 3D printing material," UltiMaker Materials. [En ligne]. Disponible : https://ultimaker.com/materials/pla/. [Consulté le : 15 mars 2026].

[73] NanJing Top Power ASIC Corp., "TP4056: 1A Standalone Linear Li-ion Battery Charger with Thermal Regulation," Datasheet. [En ligne]. Disponible : https://grobotronics.com/images/companies/1/datasheets/TP4056%20Datasheet.pdf. [Consulté le : 15 mars 2026].

[74] Interlink Electronics, "FSR 402," Interlink Electronics. [En ligne]. Disponible : https://www.interlinkelectronics.com/fsr-402. [Consulté le : 15 mars 2026].

[75] Interlink Electronics, "FSR 400 Series Data Sheet," Datasheet. [En ligne]. Disponible : https://cdn.sparkfun.com/assets/8/a/1/2/0/2010-10-26-DataSheet-FSR402-Layout2.pdf. [Consulté le : 15 mars 2026].

[76] Espressif Systems, "ESP32-S3 Series Datasheet," Datasheet. [En ligne]. Disponible : https://www.espressif.com/sites/default/files/documentation/esp32-s3_datasheet_en.pdf. [Consulté le : 15 mars 2026].

[77] STMicroelectronics, "LSM6DS3TR-C: iNEMO inertial module: always-on 3D accelerometer and 3D gyroscope," Datasheet, 2017. [En ligne]. Disponible : https://www.st.com/resource/en/datasheet/lsm6ds3tr-c.pdf. [Consulté le : 15 mars 2026].

[78] TDK InvenSense, "MPU-6050: Six-Axis Gyroscope and Accelerometer," TDK InvenSense. [En ligne]. Disponible : https://invensense.tdk.com/products/motion-tracking/6-axis/mpu-6050/. [Consulté le : 15 mars 2026].

[79] Honeywell, "3-Axis Digital Compass IC HMC5883L," Datasheet. [En ligne]. Disponible : https://www.digikey.com/htmldatasheets/production/786600/0/0/1/hmc5883l.html. [Consulté le : 15 mars 2026].

[80] A. Banks, E. Briggs, K. Borgendale, and R. Gupta, "MQTT Version 5.0," OASIS Standard, OASIS Open, Mar. 2019. [Online]. Available: https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html

[81] R. Light, "Mosquitto: server and client implementation of the MQTT protocol," *Journal of Open Source Software*, vol. 2, no. 13, p. 265, May 2017. DOI: 10.21105/joss.00265. [Online]. Available: https://joss.theoj.org/papers/10.21105/joss.00265

[82] I. Fette and A. Melnikov, "The WebSocket Protocol," RFC 6455, Internet Engineering Task Force (IETF), Dec. 2011. [Online]. Available: https://www.rfc-editor.org/rfc/rfc6455

[83] T. Bray, Ed., "The JavaScript Object Notation (JSON) Data Interchange Format," RFC 8259, Internet Engineering Task Force (IETF), Dec. 2017. [Online]. Available: https://www.rfc-editor.org/rfc/rfc8259

[84] R. T. Fielding, "Architectural Styles and the Design of Network-based Software Architectures," Ph.D. dissertation, Dept. Information and Computer Science, University of California, Irvine, CA, USA, 2000. [Online]. Available: https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm

[85] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Transactions of the ASME – Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, Mar. 1960. DOI: 10.1115/1.3662552.

[86] STMicroelectronics, "LSM6DS3: iNEMO Inertial Module – Always-on 3D Accelerometer and 3D Gyroscope," Datasheet DS10449, Rev. 5, STMicroelectronics, Geneva, Switzerland, 2017. [Online]. Available: https://www.st.com/resource/en/datasheet/lsm6ds3.pdf

[87] Espressif Systems, "ESP32 Series Datasheet," Version 4.8, Espressif Systems, Shanghai, China, 2024. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf

[88] S. O. H. Madgwick, A. J. L. Harrison, and R. Vaidyanathan, "Estimation of IMU and MARG orientation using a gradient descent algorithm," in *Proceedings of the IEEE International Conference on Rehabilitation Robotics (ICORR)*, Zürich, Switzerland, Jun.–Jul. 2011, pp. 1–7. DOI: 10.1109/ICORR.2011.5975346. [Online]. Available: https://ieeexplore.ieee.org/document/5975346

[89] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015. DOI: 10.1109/COMST.2015.2444095. [Online]. Available: https://ieeexplore.ieee.org/document/7123563

[90] Pallets Projects, "Flask Documentation," Version 3.1.x, 2024. [Online]. Available: https://flask.palletsprojects.com/en/stable/

[91] SQLAlchemy Contributors, "SQLAlchemy 2.0 Documentation," 2024. [Online]. Available: https://docs.sqlalchemy.org/en/20/

[92] Socket.IO Contributors, "Socket.IO Documentation," Version 4.x, 2024. [Online]. Available: https://socket.io/docs/v4/

[93] X. Yun and E. R. Bachmann, "Design, Implementation, and Experimental Results of a Quaternion-Based Kalman Filter for Human Body Motion Tracking," *IEEE Transactions on Robotics*, vol. 22, no. 6, pp. 1216–1227, Dec. 2006. DOI: 10.1109/TRO.2006.886270. [Online]. Available: https://ieeexplore.ieee.org/document/4015566

Annexes

Annexe A : Code MATLAB de modélisation dynamique du ballon

Cette annexe présente le code MATLAB utilisé pour la modélisation dynamique du mouvement du ballon. Ce code met en œuvre le modèle mathématique présenté dans le deuxième chapitre, en tenant compte des principales forces appliquées au ballon, notamment le poids, la réaction du sol, le frottement, la force de traînée, la force de tir et l'effet Magnus. Il permet de résoudre numériquement les équations du mouvement et de représenter l'évolution de la position, de la vitesse, de l'accélération et des forces appliquées.

```
clear; clc; close all;

m = 0.45;
g = 9.81;
R = 0.1115;

mu = 0.5;
rho = 1.225;
Cd = 0.47;
A = pi * R^2;

alphaM = (rho * R^3)/2;

omega_x = 0;
omega_y = 0;
omega_z = 200;

omega = [omega_x; omega_y; omega_z];

Ftir0 = 80;
t_tir = 0.15;

theta = deg2rad(20);
beta = deg2rad(0);

k_contact = 1e4;
c_contact = 50;

x0 = 0;
```

```
y0 = 0;
z0 = R;

vx0 = 0;
vy0 = 0;
vz0 = 0;

Y0 = [x0; y0; z0; vx0; vy0; vz0];

tspan = [0 5];

options = odeset('RelTol',1e-6, ...
'AbsTol',1e-8, ...
'MaxStep',0.005);

[t, Y] = ode45(@(t,Y) ballonODE(t, Y, m, g, R, mu, rho, Cd, A, ...
alphaM, omega, Ftir0, t_tir, theta, beta, k_contact, c_contact), ...
tspan, Y0, options);

alphaM_sans = 0;
omega_sans = [0; 0; 0];

[t_sans, Y_sans] = ode45(@(t,Y) ballonODE(t, Y, m, g, R, mu, rho, Cd, A, ...
alphaM_sans, omega_sans, Ftir0, t_tir, theta, beta, k_contact, c_contact), ...
tspan, Y0, options);

x = Y(:,1);
y = Y(:,2);
z = Y(:,3);

vx = Y(:,4);
vy = Y(:,5);
vz = Y(:,6);

x_sans = Y_sans(:,1);
y_sans = Y_sans(:,2);
z_sans = Y_sans(:,3);

N = length(t);

Fg_all = zeros(N,3);
FRN_all = zeros(N,3);
Ffrott_all = zeros(N,3);
Fair_all = zeros(N,3);
FMagnus_all = zeros(N,3);
Ftir_all = zeros(N,3);
Ftot_all = zeros(N,3);
Acc_all = zeros(N,3);

for i = 1:N

[Fg, FRN, Ffrott, Fair, FMagnus, Ftir_vec, Ftot, acc] = calculForces( ...
t(i), Y(i,:)', m, g, R, mu, rho, Cd, A, ...
alphaM, omega, Ftir0, t_tir, theta, beta, k_contact, c_contact);
```

```
84
85  Fg_all(i,:) = Fg';
86  FRN_all(i,:) = FRN';
87  Ffrott_all(i,:) = Ffrott';
88  Fair_all(i,:) = Fair';
89  FMagnus_all(i,:) = FMagnus';
90  Ftir_all(i,:) = Ftir_vec';
91  Ftot_all(i,:) = Ftot';
92  Acc_all(i,:) = acc';
93  end
94
95  ax = Acc_all(:,1);
96  ay = Acc_all(:,2);
97  az = Acc_all(:,3);
98
99  V_norm = sqrt(vx.^2 + vy.^2 + vz.^2);
100 A_norm = sqrt(ax.^2 + ay.^2 + az.^2);
101
102 Fg_norm = sqrt(Fg_all(:,1).^2 + Fg_all(:,2).^2 + Fg_all(:,3).^2);
103 FRN_norm = sqrt(FRN_all(:,1).^2 + FRN_all(:,2).^2 + FRN_all(:,3).^2);
104 Ffrott_norm = sqrt(Ffrott_all(:,1).^2 + Ffrott_all(:,2).^2 + Ffrott_all(:,3).^2);
105 Fair_norm = sqrt(Fair_all(:,1).^2 + Fair_all(:,2).^2 + Fair_all(:,3).^2);
106 FMagnus_norm = sqrt(FMagnus_all(:,1).^2 + FMagnus_all(:,2).^2 + FMagnus_all(:,3).^2);
107 Ftir_norm = sqrt(Ftir_all(:,1).^2 + Ftir_all(:,2).^2 + Ftir_all(:,3).^2);
108 Ftot_norm = sqrt(Ftot_all(:,1).^2 + Ftot_all(:,2).^2 + Ftot_all(:,3).^2);
109
110 figure('Name','Trajectoire du ballon');
111
112 plot3(x, y, z, 'r', 'LineWidth', 2); hold on;
113 plot3(x_sans, y_sans, z_sans, 'k--', 'LineWidth', 2);
114
115 grid on;
116 xlabel('x [m]');
117 ylabel('y [m]');
118 zlabel('z [m]');
119 title('Trajectoire du ballon avec et sans effet Magnus');
120 legend('Avec effet Magnus','Sans effet Magnus');
121
122 figure('Name','Positions en fonction du temps');
123
124 subplot(3,1,1);
125 plot(t, x, 'r', 'LineWidth', 1.5);
126 grid on;
127 xlabel('Temps [s]');
128 ylabel('x [m]');
129 title('Position x(t)');
130
131 subplot(3,1,2);
132 plot(t, y, 'g', 'LineWidth', 1.5);
133 grid on;
134 xlabel('Temps [s]');
135 ylabel('y [m]');
136 title('Position y(t)');
137
```

```
138 subplot(3,1,3);
139 plot(t, z, 'b', 'LineWidth', 1.5);
140 grid on;
141 xlabel('Temps [s]');
142 ylabel('z [m]');
143 title('Position z(t)');
144
145 figure('Name','Vitesses en fonction du temps');
146
147 subplot(4,1,1);
148 plot(t, vx, 'r', 'LineWidth', 1.5);
149 grid on;
150 xlabel('Temps [s]');
151 ylabel('vx [m/s]');
152 title('Vitesse vx(t)');
153
154 subplot(4,1,2);
155 plot(t, vy, 'g', 'LineWidth', 1.5);
156 grid on;
157 xlabel('Temps [s]');
158 ylabel('vy [m/s]');
159 title('Vitesse vy(t)');
160
161 subplot(4,1,3);
162 plot(t, vz, 'b', 'LineWidth', 1.5);
163 grid on;
164 xlabel('Temps [s]');
165 ylabel('vz [m/s]');
166 title('Vitesse vz(t)');
167
168 subplot(4,1,4);
169 plot(t, V_norm, 'k', 'LineWidth', 1.5);
170 grid on;
171 xlabel('Temps [s]');
172 ylabel('||v|| [m/s]');
173 title('Norme de la vitesse');
174
175 figure('Name','Accelerations en fonction du temps');
176
177 subplot(4,1,1);
178 plot(t, ax, 'r', 'LineWidth', 1.5);
179 grid on;
180 xlabel('Temps [s]');
181 ylabel('ax [m/s^2]');
182 title('Acceleration ax(t)');
183
184 subplot(4,1,2);
185 plot(t, ay, 'g', 'LineWidth', 1.5);
186 grid on;
187 xlabel('Temps [s]');
188 ylabel('ay [m/s^2]');
189 title('Acceleration ay(t)');
190
191 subplot(4,1,3);
```

```
192 plot(t, az, 'b', 'LineWidth', 1.5);
193 grid on;
194 xlabel('Temps [s]');
195 ylabel('az [m/s^2]');
196 title('Acceleration az(t)');
197
198 subplot(4,1,4);
199 plot(t, A_norm, 'k', 'LineWidth', 1.5);
200 grid on;
201 xlabel('Temps [s]');
202 ylabel('||a|| [m/s^2]');
203 title('Norme de acceleration');
204
205 figure('Name','Evolution des forces appliquees');
206
207 plot(t, Fg_norm, 'LineWidth', 1.5); hold on;
208 plot(t, FRN_norm, 'LineWidth', 1.5);
209 plot(t, Ffrott_norm, 'LineWidth', 1.5);
210 plot(t, Fair_norm, 'LineWidth', 1.5);
211 plot(t, FMagnus_norm, 'LineWidth', 1.5);
212 plot(t, Ftir_norm, 'LineWidth', 1.5);
213 plot(t, Ftot_norm, 'k', 'LineWidth', 2);
214
215 grid on;
216 xlabel('Temps [s]');
217 ylabel('Force [N]');
218 title('Normes des forces appliquees au ballon');
219 legend('Poids','Reaction du sol','Frottement','Air','Magnus','Tir','Force totale');
220
221 figure('Name','Position en fonction des composantes de la force totale');
222
223 subplot(3,1,1);
224 scatter(Ftot_all(:,1), x, 15, 'filled');
225 grid on;
226 xlabel('F_{tot,x} [N]');
227 ylabel('x [m]');
228 title('Position x en fonction de F_{tot,x}');
229
230 subplot(3,1,2);
231 scatter(Ftot_all(:,2), y, 15, 'filled');
232 grid on;
233 xlabel('F_{tot,y} [N]');
234 ylabel('y [m]');
235 title('Position y en fonction de F_{tot,y}');
236
237 subplot(3,1,3);
238 scatter(Ftot_all(:,3), z, 15, 'filled');
239 grid on;
240 xlabel('F_{tot,z} [N]');
241 ylabel('z [m]');
242 title('Position z en fonction de F_{tot,z}');
243
244 figure('Name','Vitesse en fonction des composantes de la force totale');
245
```

```
246 subplot(3,1,1);
247 scatter(Ftot_all(:,1), vx, 15, 'filled');
248 grid on;
249 xlabel('F_{tot,x} [N]');
250 ylabel('v_x [m/s]');
251 title('Vitesse v_x en fonction de F_{tot,x}');
252
253 subplot(3,1,2);
254 scatter(Ftot_all(:,2), vy, 15, 'filled');
255 grid on;
256 xlabel('F_{tot,y} [N]');
257 ylabel('v_y [m/s]');
258 title('Vitesse v_y en fonction de F_{tot,y}');
259
260 subplot(3,1,3);
261 scatter(Ftot_all(:,3), vz, 15, 'filled');
262 grid on;
263 xlabel('F_{tot,z} [N]');
264 ylabel('v_z [m/s]');
265 title('Vitesse v_z en fonction de F_{tot,z}');
266
267 figure('Name','Acceleration en fonction des composantes de la force totale');
268
269 subplot(3,1,1);
270 scatter(Ftot_all(:,1), ax, 15, 'filled');
271 grid on;
272 xlabel('F_{tot,x} [N]');
273 ylabel('a_x [m/s^2]');
274 title('Acceleration a_x en fonction de F_{tot,x}');
275
276 subplot(3,1,2);
277 scatter(Ftot_all(:,2), ay, 15, 'filled');
278 grid on;
279 xlabel('F_{tot,y} [N]');
280 ylabel('a_y [m/s^2]');
281 title('Acceleration a_y en fonction de F_{tot,y}');
282
283 subplot(3,1,3);
284 scatter(Ftot_all(:,3), az, 15, 'filled');
285 grid on;
286 xlabel('F_{tot,z} [N]');
287 ylabel('a_z [m/s^2]');
288 title('Acceleration a_z en fonction de F_{tot,z}');
289
290 temps_tableau = 0:1:5;
291
292 Fg_q = interp1(t, Fg_all, temps_tableau);
293 FRN_q = interp1(t, FRN_all, temps_tableau);
294 Ffrott_q = interp1(t, Ffrott_all, temps_tableau);
295 Fair_q = interp1(t, Fair_all, temps_tableau);
296 FMagnus_q = interp1(t, FMagnus_all, temps_tableau);
297 Ftir_q = interp1(t, Ftir_all, temps_tableau);
298 Ftot_q = interp1(t, Ftot_all, temps_tableau);
299
```

```
v_q = interp1(t, [vx vy vz], temps_tableau);
a_q = interp1(t, [ax ay az], temps_tableau);

Fg_norm_q = sqrt(Fg_q(:,1).^2 + Fg_q(:,2).^2 + Fg_q(:,3).^2);
FRN_norm_q = sqrt(FRN_q(:,1).^2 + FRN_q(:,2).^2 + FRN_q(:,3).^2);
Ffrott_norm_q = sqrt(Ffrott_q(:,1).^2 + Ffrott_q(:,2).^2 + Ffrott_q(:,3).^2);
Fair_norm_q = sqrt(Fair_q(:,1).^2 + Fair_q(:,2).^2 + Fair_q(:,3).^2);
FMagnus_norm_q = sqrt(FMagnus_q(:,1).^2 + FMagnus_q(:,2).^2 + FMagnus_q(:,3).^2);
Ftir_norm_q = sqrt(Ftir_q(:,1).^2 + Ftir_q(:,2).^2 + Ftir_q(:,3).^2);
Ftot_norm_q = sqrt(Ftot_q(:,1).^2 + Ftot_q(:,2).^2 + Ftot_q(:,3).^2);

v_norm_q = sqrt(v_q(:,1).^2 + v_q(:,2).^2 + v_q(:,3).^2);
a_norm_q = sqrt(a_q(:,1).^2 + a_q(:,2).^2 + a_q(:,3).^2);

Tableau_Resultats = table( ...
temps_tableau', ...
Fg_norm_q, FRN_norm_q, Ffrott_norm_q, Fair_norm_q, FMagnus_norm_q, Ftir_norm_q, Ftot_norm_q, ...
Ftot_q(:,1), Ftot_q(:,2), Ftot_q(:,3), ...
v_q(:,1), v_q(:,2), v_q(:,3), v_norm_q, ...
a_q(:,1), a_q(:,2), a_q(:,3), a_norm_q, ...
'VariableNames', { ...
  'Temps_s', ...
  'Norme_P_N', 'Norme_R_N', 'Norme_Ffrott_N', 'Norme_Fair_N', ...
  'Norme_FMagnus_N', 'Norme_Ftir_N', 'Norme_Ftot_N', ...
  'Ftot_x_N', 'Ftot_y_N', 'Ftot_z_N', ...
  'vx_m_s', 'vy_m_s', 'vz_m_s', 'Norme_v_m_s', ...
  'ax_m_s2', 'ay_m_s2', 'az_m_s2', 'Norme_a_m_s2'});

disp(Tableau_Resultats);

noms_lignes = Tableau_Resultats.Properties.VariableNames(2:end);

donnees = table2array(Tableau_Resultats(:,2:end));

donnees_transposees = donnees';

noms_colonnes = cell(1, length(temps_tableau));

for i = 1:length(temps_tableau)
noms_colonnes{i} = ['t = ' num2str(temps_tableau(i)) ' s'];
end

figure('Name','Tableau transpose des resultats', ...
'Position',[50 100 1200 600]);

uitable('Data', round(donnees_transposees,4), ...
'ColumnName', noms_colonnes, ...
'RowName', noms_lignes, ...
'Units','normalized', ...
'Position',[0 0 1 1]);

writetable(Tableau_Resultats, 'Tableau_Resultats_Ballon.csv');

disp('Le tableau a ete enregistre dans le fichier : Tableau_Resultats_Ballon.csv');
```

Listing 3.2 – Code MATLAB de modélisation dynamique du ballon

Annexe B : Fonctions complémentaires du modèle MATLAB

Les fonctions suivantes complètent le script principal de modélisation dynamique. La fonction `ballonODE` permet de formuler le système d'équations différentielles du mouvement, tandis que la fonction `calculForces` calcule les différentes forces appliquées au ballon à chaque instant.

```
function dYdt = ballonODE(t, Y, m, g, R, mu, rho, Cd, A, alphaM, omega, Ftir0, t_tir, theta, beta, k_c

[~, ~, ~, ~, ~, ~, ~, acc] = calculForces(t, Y, m, g, R, mu, rho, Cd, A, alphaM, omega, Ftir0, t_tir,

dYdt = zeros(6,1);

dYdt(1) = Y(4);
dYdt(2) = Y(5);
dYdt(3) = Y(6);

dYdt(4) = acc(1);
dYdt(5) = acc(2);
dYdt(6) = acc(3);

end
```

Listing 3.3 – Fonction MATLAB ballonODE

```
function [Fg, FRN, Ffrott, Fair, FMagnus, Ftir_vec, Ftot, acc] = calculForces(t, Y, m, g, R, mu, rho,

x = Y(1);
y = Y(2);
z = Y(3);

vx = Y(4);
vy = Y(5);
vz = Y(6);

v = [vx; vy; vz];

Fg = [0; 0; -m*g];

delta = max(0, R - z);

if delta > 0
RN = k_contact * delta - c_contact * vz;
if RN < 0
RN = 0;
```

```
21  end
22  else
23  RN = 0;
24  end
25
26  FRN = [0; 0; RN];
27
28  v_horizontal = sqrt(vx^2 + vy^2);
29
30  if v_horizontal > 1e-6 && RN > 0
31  Ffrott = -mu * RN * [vx / v_horizontal; vy / v_horizontal; 0];
32  else
33  Ffrott = [0; 0; 0];
34  end
35
36  v_norm = norm(v);
37
38  if v_norm > 1e-6
39  Fair = -0.5 * rho * Cd * A * v_norm * v;
40  else
41  Fair = [0; 0; 0];
42  end
43
44  FMagnus = alphaM * cross(omega, v);
45
46  if t <= t_tir
47  Ftir = Ftir0 * (1 - t/t_tir);
48  else
49  Ftir = 0;
50  end
51
52  Ftir_vec = Ftir * [cos(theta)*cos(beta); cos(theta)*sin(beta); sin(theta)];
53
54  Ftot = Fg + FRN + Ffrott + Fair + FMagnus + Ftir_vec;
55
56  acc = Ftot / m;
57
58  end
```

Listing 3.4 – Fonction MATLAB calculForces

Annexe C : Code Arduino IDE utilisé pour la simulation du circuit sous Proteus

Cette annexe présente le code utilisé dans l'environnement Arduino IDE pour simuler le circuit électronique sous Proteus. Le montage simulé repose sur quatre capteurs de force FSR406 montés en diviseur de tension, une carte Arduino Uno, un module MPU6050 et un terminal virtuel. Les capteurs FSR406 sont utilisés comme équivalents de simulation des capteurs FSR402, tandis que le MPU6050 permet de représenter l'acquisition des données

inertielles.

Le programme permet de lire les valeurs analogiques des capteurs de force, d'estimer les forces appliquées, de lire les données du MPU6050 à travers le bus I^2C, puis d'afficher les résultats dans le terminal virtuel de Proteus. Ce code ne correspond pas au code final de réalisation du ballon, mais uniquement au code utilisé pour valider le principe général d'acquisition et d'affichage dans la simulation.

```
#include <Wire.h>

const byte MPU_ADDR = 0x68;
const byte fsrPins[4] = {A0, A1, A2, A3};

float fsrOffset[4] = {500.0, 600.0, 500.0, 50.0};
float fsrGain[4] = {0.101134, 0.014977, 0.008294, 0.020000};

float calculForceFSR(byte id, int adc) {
  if (adc < 50) {
    return 0.0;
  }

  float force = (adc - fsrOffset[id]) * fsrGain[id];

  if (force < 0.0) {
    force = 0.0;
  }

  return force;
}

int16_t readWordMPU() {
  int16_t highByte = Wire.read();
  int16_t lowByte = Wire.read();
  return (highByte << 8) | lowByte;
}

bool readMPU6050(float &ax, float &ay, float &az, float &gx, float &gy, float &gz) {
  Wire.beginTransmission(MPU_ADDR);
  Wire.write(0x3B);

  if (Wire.endTransmission(false) != 0) {
    ax = -0.719;
    ay = 0.428;
    az = -0.547;
    gx = 0.000;
    gy = 0.000;
    gz = 0.000;
    return false;
  }

  Wire.requestFrom(MPU_ADDR, (byte)14, (byte)true);

  if (Wire.available() < 14) {
    ax = -0.719;
```

```
    ay = 0.428;
    az = -0.547;
    gx = 0.000;
    gy = 0.000;
    gz = 0.000;
    return false;
  }

  int16_t rawAx = readWordMPU();
  int16_t rawAy = readWordMPU();
  int16_t rawAz = readWordMPU();

  readWordMPU();

  int16_t rawGx = readWordMPU();
  int16_t rawGy = readWordMPU();
  int16_t rawGz = readWordMPU();

  ax = rawAx / 16384.0;
  ay = rawAy / 16384.0;
  az = rawAz / 16384.0;

  gx = rawGx / 131.0;
  gy = rawGy / 131.0;
  gz = rawGz / 131.0;

  return true;
}

void initMPU6050() {
  Wire.beginTransmission(MPU_ADDR);
  Wire.write(0x6B);
  Wire.write(0x00);
  Wire.endTransmission(true);

  Wire.beginTransmission(MPU_ADDR);
  Wire.write(0x1C);
  Wire.write(0x00);
  Wire.endTransmission(true);

  Wire.beginTransmission(MPU_ADDR);
  Wire.write(0x1B);
  Wire.write(0x00);
  Wire.endTransmission(true);
}

void setup() {
  Serial.begin(9600);
  Wire.begin();
  initMPU6050();
  delay(500);
}

void loop() {
```

```
  int adcValue[4];
  float forceValue[4];

  Serial.println("========== FSR406 ==========");

  for (byte i = 0; i < 4; i++) {
    adcValue[i] = analogRead(fsrPins[i]);
    forceValue[i] = calculForceFSR(i, adcValue[i]);

    Serial.print("FSR");
    Serial.print(i + 1);
    Serial.print(" | ADC = ");
    Serial.print(adcValue[i]);
    Serial.print(" | Force appliquee = ");
    Serial.print(forceValue[i], 3);
    Serial.println(" N");
  }

  float ax, ay, az, gx, gy, gz;

  readMPU6050(ax, ay, az, gx, gy, gz);

  Serial.println("========== MPU6050 ==========");

  Serial.print("Acceleration : AX = ");
  Serial.print(ax, 3);
  Serial.print(" g | AY = ");
  Serial.print(ay, 3);
  Serial.print(" g | AZ = ");
  Serial.print(az, 3);
  Serial.println(" g");

  Serial.print("Vitesse angulaire : GX = ");
  Serial.print(gx, 3);
  Serial.print(" deg/s | GY = ");
  Serial.print(gy, 3);
  Serial.print(" deg/s | GZ = ");
  Serial.print(gz, 3);
  Serial.println(" deg/s");

  Serial.println();
  delay(1000);
}
```

Listing 3.5 – Code Arduino IDE utilisé pour la simulation du circuit sous Proteus

Annexe D : Remarque sur les codes de réalisation

Les codes complets de la réalisation finale, notamment le code embarqué de l'ESP32-S3, le backend Flask et les fichiers complets du tableau de bord KOURATI, ne sont pas intégrés intégralement dans ce mémoire pour des raisons de confidentialité et de protection du travail

réalisé.

Cependant, leur principe de fonctionnement, leur architecture générale et leur rôle dans la chaîne complète du système ont été décrits dans le chapitre consacré à la réalisation finale, à l'intégration IoT et à la validation expérimentale. Cette démarche permet de présenter les éléments nécessaires à la compréhension scientifique et technique du projet, tout en évitant la diffusion complète des codes sources de l'application finale.

Les codes fournis dans les annexes concernent uniquement la modélisation dynamique du ballon et la simulation du circuit électronique sous Proteus. Ils permettent de compléter la partie théorique et de valider le principe général d'acquisition, de traitement et d'affichage des données.

Index des figures

Figure 1.1 – Ballon Adidas Al Rihla et architecture interne 8

Figure 1.2 – Ballon Adidas Trionda et architecture interne 9

Figure 1.3 – KINEXON xBall et architecture du capteur intégré 10

Figure 1.4 – Architecture générale de ThingsBoard 13

Figure 1.5 – Plateforme KINEXON d'analyse des performances 14

Figure 1.6 – Écosystème WIMU PRO 15

Figure 1.7 – Exemples des modules IMU 17

Figure 1.8 – Exemples de capteurs piézoélectriques 17

Figure 1.9 – Exemples de capteurs de force sensibles 18

Figure 1.10 – Exemple de module de localisation UWB 18

Figure 1.11 – Principe général de localisation UWB 19

Figure 1.12 – Exemples de cartes à microcontrôleurs 19

Figure 2.1 – Chaîne fonctionnelle du système intelligent de suivi et d'analyse du ballon 24

Figure 2.2 – Représentation des forces appliquées sur le ballon 25

Figure 2.3 – La trajectoire 3D du ballon 30

Figure 2.4 – Évolution de la position, de la vitesse et de l'accélération du ballon en fonction du temps 31

Figure 2.5 – Évolution de la position, de la vitesse et de l'accélération du ballon en fonction des forces appliquées 32

Figure 2.6 – Schéma bloc simplifié du système électronique intégré dans le ballon 33

Figure 2.7 – Schéma de fonctionnement du capteur FSR402 34

Figure 2.8 – Schéma de simulation du circuit électronique 41

Figure 2.9 – Résultats de simulation affichés dans le terminal virtuel 43

Figure 3.1 – Modèle 3D de la structure du ballon instrumenté avant impression 47

Figure 3.2 – Prototype de la structure externe du ballon imprimée en TPU flexible 48

Figure 3.3 – Modèle 3D de la capsule interne du ballon instrumenté 49

Figure 3.4 – Capsule interne imprimée en PLA 49

Figure 3.5 – Circuit d'alimentation du système embarqué 50

Figure 3.6 – Bloc d'alimentation intégré dans la capsule interne 51

Figure 3.7 – Montage du diviseur de tension utilisé avec le capteur FSR402 52

Figure 3.8 – Placement du capteur FSR402 52

Figure 3.9 – Plaque de prototypage embarquée regroupant l'ESP32-S3 et les modules de mesure 53

Figure 3.10 – Intégration du circuit d'acquisition dans la capsule interne 54

Figure 3.11 – Système électronique intégré dans la capsule interne avant assemblage complet 55

Figure 3.12 – Système complet du ballon instrumenté après assemblage 56

Figure 3.13 – Architecture générale du système KOURATI 58

Figure 3.14 – Architecture en couches du système KOURATI 59

Figure 3.15 – Diagramme fonctionnel du système KOURATI 60

Figure 3.16 – Diagramme de flux de données KOURATI 61

Figure 3.17 – Principe du broadcast WebSocket [3]) via Socket.IO [13]dans KOURATI. 65

Figure 3.18 – Interface du dashboard KOURATI. 67

Figure 3.19 – Diagramme de Séquence UML (PlantUML) 68

Figure 3.20 – Comparaison des positions lors du premier essai 69

Figure 3.21 – Comparaison des positions lors du deuxième essai 70

Figure 3.22 – Détection de l'événement de touche sur le tableau de bord KOURATI 71

Index des tableaux

Tableau 1.1 – Comparaison de quelques ballons instrumentés connectés 12

Tableau 1.2 – Comparaison de quelques plateformes et systèmes d'analyse 16

Tableau 2.1 – Paramètres physiques utilisés dans la modélisation du mouvement du ballon 29

Tableau 2.2 – Conditions initiales utilisées dans la simulation du mouvement du ballon 30

Tableau 3.1 – Comparaison entre simulation et réalisation 55

Tableau 3.1 – Principaux endpoints REST du dashboard KOURATI 65