Design and Implementation of an IMU Sensor System to Estimate a Hockey Puck's Peak Velocity

Ali Gaeini

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of

Master of Applied Science (Electrical Engineering) at

Concordia University

Montréal, Québec, Canada

September 2023

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Ali Gaeini**

Entitled: **Design and Implementation of an IMU Sensor System to Estimate a Hockey Puck's Peak Velocity**

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science (Electrical Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the Final Examining Committee:

________________________________ Chair
Dr. Krzysztof Skonieczny

________________________________ External Examiner
Dr. Brandon Gordon

________________________________ Examiner
Dr. Krzysztof Skonieczny

________________________________ Supervisor
Dr. Glenn Cowan

Approved by ________________________________
Yousef R. Shayan, Chair
Department of Electrical and Computer Engineering

____________ 2023 ____________________________
Mourad Debbabi, Dean
Faculty of Engineering and Computer Science

Abstract

Design and Implementation of an IMU Sensor System to Estimate a Hockey Puck's Peak Velocity

Ali Gaeini

The rapid advancement in sensor technology can revolutionize how sports dynamics are understood and analyzed. This thesis focuses on designing and implementing an Inertial Measurement Unit (IMU) sensor system to be deployed within a hockey puck to estimate its peak velocity.

The research involved the intricate design of a sensor system comprising an accelerometer, two gyroscopes, and a magnetometer. Moreover, puck preparation was carried out to secure the sensor and battery within the puck to ensure functionality and durability. Furthermore, a data acquisition system is developed to receive, save, and plot data transmitted via Bluetooth Low Energy (BLE) protocol.

Three distinct methods for estimating the puck's peak velocity from the sensor data are compared. It is discovered that the method based on an extended Kalman filter and utilizing data from all three sensor types exhibits superior accuracy. This method is subsequently validated under various hockey shot conditions, reinforcing its practical applicability. Moreover, the relationship between velocity estimation error versus true velocity is investigated.

Primarily designed for research studies, this work offers a foundational understanding of hockey puck dynamics, despite the sensor system not being tailored for real-game scenarios. The insights gained have substantial implications for further sports analytics and player training. Furthermore, the results outline a promising pathway for future sports engineering and wearable technology investigations.

Acknowledgments

I wish to express my deepest gratitude to the following:

To Concordia University, FRQNT, NSERC, and the McGill IHRG. Their opportunities for research and financial support have been invaluable in advancing this project;

To my beloved wife, Shaghayegh, and my family and friends. Their unwavering support and encouragement have been my constant source of inspiration;

To Vinicius G. Sirtoli, Alireza Saboukhi, and Yasaman Hedayatnasab from Concordia University. Their collaborative efforts have significantly contributed to this research;

To Philippe Renaud and Andrew Phillips from McGill IHRG, who provided significant assistance during the testing phase;

Last but not least, to my supervisor, Dr. Glenn Cowan. His continuous guidance and invaluable advice have played a crucial role in shaping this research.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Ice hockey is a dynamic and fast-paced sport that involves two teams attempting to score goals by shooting a hard rubber disc, called a puck, into the opposing team's net [1]. According to the International Ice Hockey Federation, there are over 1.5 million registered players worldwide, making ice hockey a popular sport [2, 3]. The application of scientific principles and technologies has become increasingly important in the sport of ice hockey [4]. Recognizing the importance of understanding puck dynamics, this thesis focuses on designing and implementing a sensor to be placed inside the hockey puck to estimate its peak velocity.

Despite ice hockey's popularity, the game's analytics and technology still need to be fully explored. Especially in the case of the puck, understanding its dynamics is crucial, not only for strategizing and optimizing team performance but also for enhancing player training, refining stick designs for better play, and reducing the risk of injuries [5, 6]. The need for more detailed insight into puck behavior stems mainly from the lack of robust real-time tools to measure its dynamics.

Measuring the puck's velocity is challenging due to the sport's high-speed nature and the puck's rapid, unpredictable movements. Current technologies used for measuring speed, such as high-speed cameras and radar guns, have limitations. For instance, they require line-of-sight access to the puck, which is not always feasible in a real-world game scenario [7]. Moreover, these tools often provide only the average velocity over a given distance and do not offer insight into the puck's instantaneous velocities or complex motion dynamics. In research studies that require precise measurements, these tools often lack the necessary detail.

This thesis aims to develop an embedded sensor system within a standard hockey puck to measure its peak velocity in controlled settings. The sensor system, equipped with an Inertial Measurement Unit (IMU), including an accelerometer, a gyroscope, and a magnetometer, wirelessly transmits real-time data about the puck's movements. This system aims to overcome the limitations of current technologies by providing immediate, continuous, and detailed insight into the puck's motion, including its peak velocity.

The IMU is an integral part of the sensor system. By measuring the linear and rotational motion of the puck, the IMU provides the raw data that is then used to estimate the puck's velocity. Inertial sensors have been widely used in various applications since they capture high-frequency, multi-dimensional movements, making them particularly suitable for this project.

The development of this sensor system involves a multi-disciplinary approach, integrating concepts from electronics (for the sensor system), mechanical engineering (for the puck's structural modifications), and data processing (for velocity estimation). This combination of skills and fields marks a significant advancement in the sports technology domain, offering a novel method to understand and analyze the complex dynamics of a hockey puck.

Figure 1.1 presents an overview of the proposed design. The block diagram illustrates the main components and their interactions within the system.

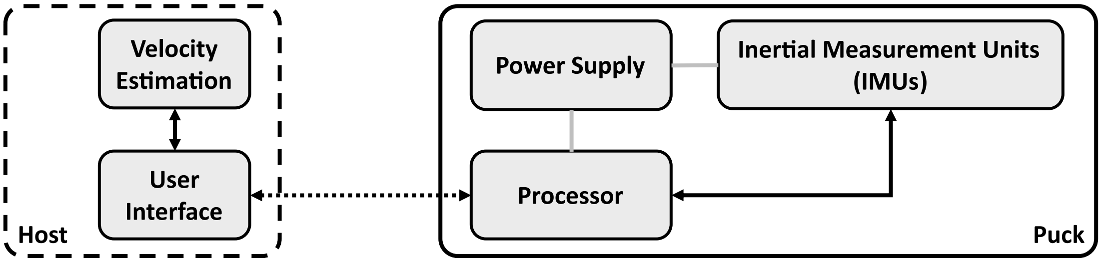


Figure 1.1: Block diagram of the sensor system embedded in the hockey puck.

The sensor system is designed for the McGill Ice Hockey Research Group (IHRG). Recognizing the McGill IHRG's objective to understand the mechanics of ice hockey, this system is intended to provide them with precise, real-time data on puck dynamics, enhancing their research capabilities and depth of analysis.

1.1 Thesis Organization

The remainder of this thesis is structured as follows, providing a comprehensive exploration of the study's objectives, methodologies, findings, and implications:

- *Chapter 2 – Literature Review:* This chapter explores the fundamentals of IMUs, various velocity estimation techniques, and relevant sensor designs from past studies. It concludes by outlining the critical specifications required for the sensor system within the scope of this work.
- *Chapter 3 – Materials and Methods:* This chapter details the design methodology used to design the sensor system, modifications to the standard hockey puck for sensor integration, and developing the data acquisition system. The reasoning behind various design decisions will also be explained.
- *Chapter 4 – Results:* In this chapter, the sensor system is subjected to practical testing using actual hockey shots. The velocity estimates produced by the sensor system are compared with measurements taken by motion camera systems, providing a comprehensive evaluation of the sensor's performance.
- *Chapter 5 – Conclusion and Discussion:* The final chapter draws conclusions based on the results and discussions presented in the previous chapters. This chapter also discusses potential improvements and future works for the sensor system, setting the stage for further research.

1.2 Contribution

A significant advancement in sports analytics technology, particularly for ice hockey, is presented in this study. A system was developed to measure puck dynamics under controlled conditions, contributing to an enhanced understanding of this field.

Current technologies in ice hockey, such as high-speed cameras and radar guns, possess inherent limitations. These tools often capture a partial view of the puck's motion, missing its rapid and unpredictable movements [7]. These limitations are addressed by embedding a sensor system directly within the puck. This innovation allows for real-time, comprehensive insights into the puck's movement, which can significantly enrich game strategies and training methods.

This research uniquely contributes by adapting a 9-axis IMU to the dynamics of hockey puck movements. Alongside the sensor integration, advancements include puck structural modifications, an intuitive GUI design, and refined data processing techniques. A distinguishing feature of the system is the dual-gyroscope strategy: the first prioritizes precision in a lower range, while the second is utilized for a broader range with a trade-off in accuracy. Notably, the second gyroscope counters an axis often saturated during forceful shots, thus ensuring consistent and reliable data acquisition in the harshest hockey scenarios.

The sensor system can transform how coaches and trainers approach player training and skill development in ice hockey. Traditional training methods often rely on subjective assessments and primary statistical data. This technology can offer a more detailed measurement of a player's shooting performance, including the velocity and orientation of the puck, bypassing the need for expensive camera systems. Such data can enable coaches and trainers to better understand the effectiveness of a player's shooting technique and make necessary adjustments to improve their performance.

Furthermore, the sensor's real-time data can give players immediate feedback during practice sessions. This immediate feedback is invaluable for skill development, as players can modify and perfect their techniques, enhance their learning, and improve their performance more effectively.

Lastly, players can fine-tune their shots for better accuracy and speed by understanding the specific dynamics of a puck's motion. This can improve scoring opportunities during games, providing a competitive advantage. Similarly, goaltenders can use the data to study the patterns of shots, helping them to predict and react to shots more effectively.

In conclusion, the sensor system's contributions are primarily rooted in a deeper, scientific understanding of puck dynamics. Providing detailed and accurate data offers avenues for more research into player training methodologies and the sport's biomechanics, potentially influencing individual and team performance metrics in ice hockey.

Chapter 2

Literature Review

This chapter focuses on the fundamentals of the Inertial Measurement Unit (IMU), pose, and three different approaches for velocity estimation. It also summarizes previously developed sensors used in similar contexts and the required specifications for the proposed sensor system.

2.1 IMU

IMUs are electronic devices commonly used to measure and track the motion of objects in three-dimensional space. These devices typically consist of accelerometers, gyroscopes, and sometimes magnetometers, which work together to capture information about an object's linear and angular acceleration, velocity, and position. IMUs have a wide range of applications in various fields, including aerospace, robotics, and sports, due to their ability to provide accurate and real-time data about the motion of an object [8].

2.1.1 Accelerometer

An accelerometer is a sensor that measures the acceleration experienced by an object, aka the proper acceleration. It detects and measures the force generated by movement or acceleration [9].

Working Principles of Accelerometers

Most accelerometers share a similar fundamental working principle. They comprise a compliant suspension attached to a proof mass. When an acceleration is applied, the proof mass undergoes an inertial force, i.e., D'Alembert's force. Equation (1) represents this force:

$$F = m \times a \tag{1}$$

where m is the proof mass and a is the applied acceleration to m. The force F results in the proof mass displacement, leading to the suspension's deformation. The suspension's stiffness determines the extent of the deformation. This deformation can be transformed into a proportional voltage change by various transduction principles, such as piezo-resistive, capacitive, piezo-electric, optical, etc. [10]. Figure 2.1 illustrates the working principle of an accelerometer.

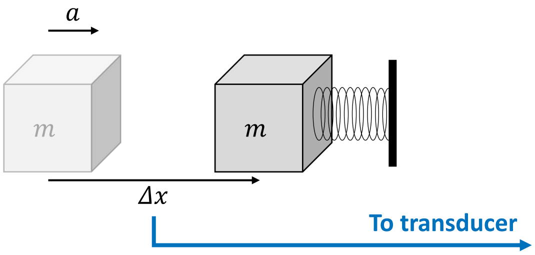


Figure 2.1: Working principle of an accelerometer [10].

In essence, the acceleration of the overall sensor is directly related to the displacement of the proof mass inside. The greater the external acceleration, the more pronounced the displacement, thus, the resultant change in voltage.

Noise Sources in Accelerometers

Accelerometer noise arises from both the mechanical and electronic components. The white noise floor is characterized by the spectral densities of the noise signal, originating from both electrical and mechanical sources, and remains constant across all frequencies. The resolution of the system is limited by its inherent noise, which is determined by its spectral density and the operating frequency range, i.e., the system's bandwidth [10].

In the case of a spring-mass-damper system, which serves as an effective model for accelerometers, energy dissipation through the damper corresponds to force fluctuations, contributing to mechanical noise. However, reducing mechanical noise significantly often limits the system's bandwidth, thus posing a trade-off between noise reduction and operating frequency range [10].

On the other hand, electronic noise is mainly caused by the amplifiers used to enhance the signal from the differential capacitance setup in accelerometers. This noise is primarily due to thermal noise from resistors in the circuitry and the flicker noise from semiconductors, which is dominant in low-frequency signals. In digital accelerometers, quantization noise forms an additional source. Therefore, electronic noise often becomes the primary limitation for the accelerometer's resolution [10].

2.1.2 Gyroscope

Gyroscopes measure an object's rotation rate or angular velocity about a specific axis. Traditional gyroscopes use a spinning rotor in a gimbal, enabling rotation in any direction. However, their need for bearings complicates miniaturization. In contrast, Micro-Electro-Mechanical System (MEMS) gyroscopes, while performing the same function, leverage the advantages of miniaturization, avoiding these limitations [11]. Their compact nature makes MEMS gyroscopes well-suited for integration into confined spaces, such as a hockey puck.

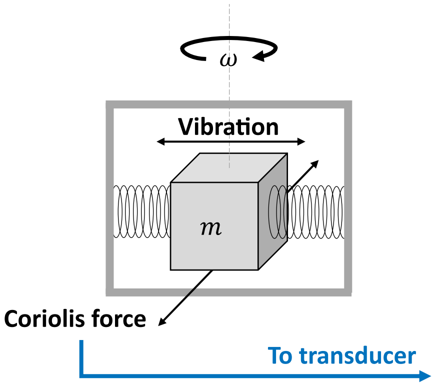


Figure 2.2: Working principle of a MEMS gyroscope [12].

Working Principle of MEMS Gyroscopes

MEMS gyroscopes are based on the principle of the Coriolis effect, which causes a vibrating mass to experience a force proportional to its angular velocity. As shown in Figure 2.2, they consist of a proof mass suspended by springs and set into motion by an actuator. As the proof mass moves, the Coriolis force acts on it, causing it to deflect in a direction perpendicular to both the input motion and the rotation axis. This deflection is detected by sensing electrodes, and the resulting signal is processed to provide an output voltage proportional to the angular velocity [11, 12].

The Coriolis Effect

The Coriolis force is a fundamental concept in physics that describes the apparent deflection of objects in motion due to the rotation of the Earth. This force acts perpendicular to the direction of movement and the axis of rotation, and it is proportional to the velocity of the moving object. The Coriolis force plays an essential role in many physical phenomena, including the motion of fluids, the formation of cyclones and hurricanes, and the movement of objects in rotating frames of reference [13].

Noise Sources in Gyroscopes

Noise in angular rate measurements is an inherent characteristic of every MEMS gyroscope, reflecting unpredictable fluctuations in the gyroscope's output in a motionless inertial state, i.e., without any rotational motion and under ideal environmental conditions, like the absence of vibration or shock. The origin of this intrinsic sensor noise lies in several sources: thermal noise due to the random motion of charge carriers, mechanical noise from the gyroscope's movable parts (related to the damping of the mechanical resonator and its movement due to thermal excitation, namely Brownian motion), electrical noise from the associated circuits like amplifiers and converters, and subtle changes in environmental factors such as temperature [14, 15].

MEMS gyroscope data sheets typically include two metrics to explain their noise characteristics: Rate Noise Density (RND) and Angle Random Walk (ARW). RND, usually measured in units of $^{\circ}/sec/\sqrt{Hz}$, estimates the overall noise in angular rate, considering the gyroscope's frequency response. Conversely, ARW, measured in units of $^\circ/\sqrt{hour}$, is useful when analyzing how noise impacts angle estimation over a specific time duration. These metrics, RND and ARW, represent the quantified effects of the combined intrinsic noise sources in the frequency and time domains, respectively [14].

2.1.3 Magnetometer

Magnetometers are sensors designed to measure the strength and direction of magnetic fields [16].

Working Principle of Magnetometers

The working principle of magnetometers is based on the physical properties of magnetic materials, such as their magnetization and susceptibility. There are several different types of magnetometers, including fluxgate, Hall effect, and magnetoresistive sensors. In general, magnetometers work by measuring the changes in a magnetic field caused by the presence of nearby magnetic materials [16].

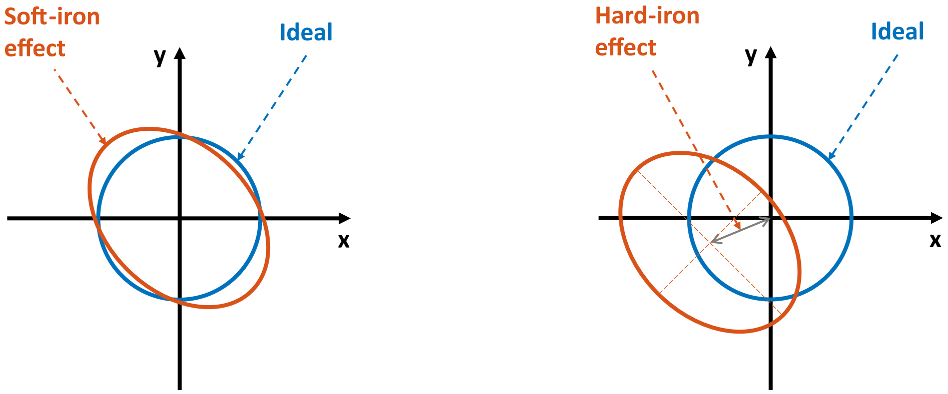
Anisotropic Magnetoresistive (AMR) magnetometers consist of a thin film of ferromagnetic material inserted between two conductive layers. When a magnetic field is applied perpendicular to the thin film, it causes the electrons in the film to experience a change in resistance. The difference in resistance is proportional to the magnitude of the magnetic field and is called the magnetoresistance effect. As the sensor rotates, the strength of the magnetic field component perpendicular to the sensor varies, causing a corresponding change in the sensor's resistance and, hence, the output voltage. The magnetic field's strength and direction can be determined by analyzing the voltage changes [16, 17].

Noise Sources in Magnetometers

Magnetometers are susceptible to noise due to various sources. One significant noise source in magnetometers is $1/f$ noise, characterized by a power spectral density inversely proportional to the frequency [18]. Another noise source is thermal noise, which is generated by the random motion of electrons and results in a Gaussian noise distribution with a power spectral density proportional to the absolute temperature [19]. Moreover, external magnetic fields can increase the magnetometer's total noise, such as geomagnetic noise, nearby electronic devices, or other magnetic objects [17].

Soft and Hard Iron Effect

Magnetometer readings are susceptible to alterations caused by several environmental factors, including the presence of soft and hard iron materials near the sensor. Soft iron materials are characterized by their narrow hysteresis loop, enabling them to be magnetized and demagnetized easily. These materials generate a distortion in the measured magnetic field, altering its magnitude and direction based on changes in the external magnetic field. In contrast, hard iron materials exhibit a broad hysteresis loop and maintain a considerable residual magnetization, resistant to demagnetization. As a result, these materials can sustain their induced magnetic field even in the presence of the Earth's or other external magnetic fields, causing a fixed bias in the magnetometer output [20]. Figure 2.3 illustrates these effects.



(a) *Soft-iron effect:* The ideal circle is distorted into an ellipse due to magnetic interference

(b) *Hard-and-soft-iron effects:* The circle is offset from the origin due to the residual magnetization and magnetic interference.

Figure 2.3: Environmental interference on magnetometer's output. The data represents the magnetometer output against 360° rotation. The ideal response should form a complete circle, but the presence of soft and hard iron effects can cause significant deviations [21].

2.2 Pose

The term “Pose” refers to the position and orientation of an object. It is commonly understood that six degrees of freedom are needed to precisely determine the pose of an object in three-dimensional space, with three of them dedicated to the position and the remaining three to the orientation [22].

2.2.1 Position and Orientation

Position refers to the location of an object in space and is often defined relative to a fixed reference point or origin. In three-dimensional space, the position is typically described using three coordinates (x, y, z) relative to a chosen origin [22].

$$p = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \in \mathbb{R}^3 \tag{2}$$

Orientation refers to the spatial arrangement of a rigid body about a reference coordinate system, often described using a set of parameters that define its position with respect to that frame [23].

Euler Angles

Euler angles refer to a set of three angles that describe the orientation of a rigid body relative to a fixed coordinate system, typically the x, y, and z axes. The three angles are usually denoted as ϕ, θ, and ψ and are known as roll, pitch, and yaw angles. The rotation about the x-axis is represented by the roll angle (ϕ), the rotation about the y-axis is represented by the pitch angle (θ), and the rotation about the z-axis is represented by the yaw angle (ψ) [23]. These angles are shown in Figure 2.4.

2.2.2 Rotation Matrices

Rotation matrices describe rotations in three-dimensional space. They are 3×3 matrices, i.e., $R \in \mathbb{R}^3$, representing the transformation of a point in space under a rotation. One of the most crucial properties of rotation matrices is that they are orthogonal, meaning that their transpose equals their

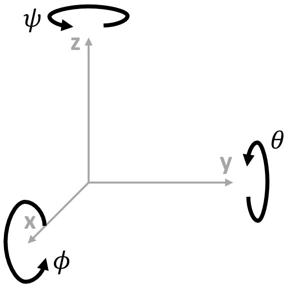


Figure 2.4: Definition of the Euler angles [23].

inverse [23] like the following.

$$RR^T = R^T R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad \det R = 1 \tag{3}$$

Another way to define rotation is using Euler angles to represent a successive rotation around three axes. In mechanics, the sequence ZYX is commonly used. This sequence involves rotating around the z-axis, then a rotation around the y-axis, and finally, a rotation around the x-axis. These rotations are operated using the fundamental rotation matrices in equations (4). Since all three rotations occur around an axis in the fixed (nonmoving) frame of reference (FOR), the rotation matrix is derived as shown in equation (5) [23, 24].

$$R_X(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \tag{4a}$$

$$R_Y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \tag{4b}$$

$$R_Z(\psi) = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4c}$$

$$R_{ZYX} = R_X(\phi)R_Y(\theta)R_Z(\psi)$$

$$= \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\psi \\ \sin\phi\sin\theta\cos\psi - \cos\phi\cos\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \sin\phi\cos\theta \\ \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi & \cos\phi\cos\theta \end{bmatrix} \tag{5}$$

Matrix representations of rotations in 3-D space present certain inefficiencies. They necessitate complex calculations to recognize the rotation axis and angle; they also prompt a redundancy with only 4 of the 9 elements being independently variable [25]. To avoid these issues, this work employs quaternions, which offer a more efficient method for handling rotations in $\mathbb{R}^3$.

2.2.3 Quaternions

Quaternions, an extension of the complex number system into a four-dimensional space, are mathematically defined as:

$$q = q_0 + q_1\boldsymbol{i} + q_2\boldsymbol{j} + q_3\boldsymbol{k} \tag{6}$$

where $q_0, q_1, q_2, q_3 \in \mathbb{R}$. The units $\boldsymbol{i}$, $\boldsymbol{j}$, and $\boldsymbol{k}$ satisfy the relationship:

$$\boldsymbol{i}^2 = \boldsymbol{j}^2 = \boldsymbol{k}^2 = \boldsymbol{ijk} = -1 \tag{7}$$

It is important to note that the elements of a quaternion, q_0, q_1, q_2, q_3, are dimensionless, having no associated units.

A quaternion q consists of a scalar part, q_0, and a vector part, $\boldsymbol{q} = (q_1, q_2, q_3)$. In $3-$D rotations, the scalar part corresponds to the rotation angle, while the vector part designates the rotation axis [25]. This is illustrated in the quaternion rotation operator, wherein quaternions effectively describe and implement 3-D rotations.

Quaternion Rotation Operator

For any given unit quaternion q, which consists of a scalar component q_0 and a vector component $\boldsymbol{q}$, it can be represented in terms of an angle θ and a unit vector $\boldsymbol{u} = \frac{\boldsymbol{q}}{\|\boldsymbol{q}\|}$ as:

$$q = \cos\frac{\theta}{2} + \boldsymbol{u}\sin\frac{\theta}{2} \tag{8}$$

The quaternion rotation operator is subsequently formulated as follows:

$$\begin{aligned} L_q(v) &= qvq^* \\ &= (q_0^2 - \|\boldsymbol{q}\|^2)v + 2(\boldsymbol{q}\cdot v)\boldsymbol{q} + 2q_0(\boldsymbol{q}\times v) \end{aligned} \tag{9}$$

where $L_q(v)$ is the result of this operation and is a vector that represents the initial vector v after the rotation by the quaternion q. For any vector $v \in \mathbb{R}^3$, the operator $L_q(v)$ translates to a rotation of v by an angle θ around the rotation axis $\boldsymbol{u}$, as illustrated in Figure 2.5 [25].

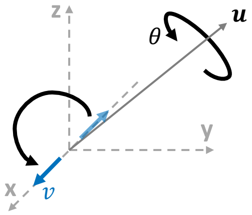


Figure 2.5: The quaternion rotation operator [26].

Using the following formula, the quaternion rotation operator $L_q(v)$ can be converted into a corresponding rotation matrix, as explained in equation (5) [27].

$$\begin{aligned} L_q(v) &= qvq^* \\ &= \begin{bmatrix} 1-2q_2^2-2q_3^2 & 2(q_1q_2+q_0q_3) & 2(q_1q_3-q_0q_2) \\ 2(q_1q_2-q_0q_3) & 1-2q_1^2-2q_3^2 & 2(q_2q_3+q_0q_1) \\ 2(q_1q_3+q_0q_2) & 2(q_2q_3-q_0q_1) & 1-2q_1^2-2q_2^2 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} \end{aligned} \tag{10}$$

The operations in equation (9) and subsequent sections are based on selected principles from quaternion algebra. For a comprehensive understanding of quaternion algebra and its broader principles, refer to [25].

Essentials of Quaternion Algebra

This section provides an overview of the principles of quaternion algebra required for this application [25]:

- *Norm determination:* The norm or magnitude of a quaternion, denoted as $\|q\|$, is:

$$\|q\| = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2} \tag{11}$$

  A unit quaternion has a norm of 1. It can be derived as:

$$q_{unit} = \frac{q_0 + q_1\boldsymbol{i} + q_2\boldsymbol{j} + q_3\boldsymbol{k}}{\|q\|} \tag{12}$$

- *Inversion:* The inverse of a quaternion q, denoted as q^{-1}, is:

$$q^{-1} = \frac{q^*}{\|q\|^2} \tag{13}$$

  where q^* is the complex conjugate of q and is represented as $q^* = q_0 - q_1\boldsymbol{i} - q_2\boldsymbol{j} - q_3\boldsymbol{k}$. This yields:

$$q^{-1} = \frac{q_0 - q_1\boldsymbol{i} - q_2\boldsymbol{j} - q_3\boldsymbol{k}}{\|q\|^2} \tag{14}$$

  Geometrically, the inversion of a quaternion corresponds to a rotation in the opposite direction about the same axis.

2.3 An Introduction to the Kalman Filters

The Kalman Filter (KF) represents a highly versatile, optimal iterative data processing technique that effectively processes every available data point, considering uncertainties in system dynamics, measurement inaccuracies, and starting conditions for the variables under consideration. By optimally merging diverse data sources and understanding system and measurement device behavior, the filter produces the best overall estimate of the target variables, reducing errors from a statistical perspective. Its iterative characteristic enables practical implementation, eliminating the need to store and reprocess all prior data whenever new information is acquired [28].

The KF is beneficial when dealing with linear systems where the process and measurement noise are Gaussian. In practice, the filter operates as software executed by a central processing unit that integrates discrete-time measurement samples [28, 29].

The KF is effective in this study since it can optimally merge diverse data sources and accommodate system behavior, yielding more accurate velocity estimates. The KF facilitates an iterative, practical, and robust process by navigating through the uncertainties in sensor dynamics and IMU measurement errors.

The Kalman Filter Algorithm

The KF operates through a two-step iterative process: prediction (*a priori*) and update (*a posteriori*). Consider a linear discrete-time system characterized by the following equations:

(1) State transition equation (prediction):

$$x_t = Ax_{t-1} + Bu_t + w_t \tag{15}$$

where x_t is the state vector at time t, A is the state transition matrix, x_{t-1} is the previous state, B is the control input matrix, u_t is the control input vector, and w_t is the process noise.

(2) Observation equation (updating):

$$z_t = Hx_t + v_t \tag{16}$$

where z_t is the observation vector, H is the observation matrix, and v_t is the observation noise.

w_t and v_t are Gaussian white noise with covariance matrices Q and R respectively. These matrices represent the noise characteristics of the system dynamics and measurements [29].

The KF algorithm can be broken down into the following stages:

(1) Initialization:

- Establish the initial state estimate $\hat{x}_0$ and initial state covariance matrix P_0.

    - The state vector x_t could be the puck's position, velocity, and orientation.
    - P Represents the uncertainty in state estimation.

(2) Forecasting (Time update):

- Predict the state at time t: $\hat{x}_t^- = A\hat{x}_{t-1} + Bu_t$
    - Matrix A describes how the puck's current state influences its future. For instance, if the puck retains its velocity, matrix A would be near-identity for velocity components.
    - Matrix B dictates how the control input forces change the puck's state.
    - u_t could be forces acting on the puck, e.g., a hockey stick strike or air resistance.
- Predict the state covariance matrix at time t: $P_t^- = AP_{t-1}A^T + Q$
    - Q describes the expected variance in the system's noise, capturing inherent unpredictabilities in the puck's movement model.

(3) Updating (Measurement update):

- Determine the Kalman gain: $K_t = P_t^- H^T (HP_t^- H^T + R)^{-1}$
    - H translates the puck state to IMU readings.
    - R represents the expected variance in IMU measurement errors.
- Update the state estimate using the measurement: $\hat{x}_t = \hat{x}_t^- + K_t(z_t - H\hat{x}_t^-)$
    - z_t as shown in equation (16) is the direct readings from the IMU, indicating its perception of the puck's state.
- Update the state covariance matrix: $P_t = (I - K_t H)P_t^-$
    - I is an identity matrix.

This process is iterated for each time step, updating the state estimate $\hat{x}_t$ and state covariance matrix P_t as needed. This allows the KF to effectively combine prior knowledge, system dynamics, and noisy observations to accurately estimate the dynamic system's state [29].

The Extended Kalman Filters

The Extended Kalman Filter (EKF) extends the primary KF designed to address nonlinear systems, e.g., a rotating puck. Unlike the linear KF, which uses linear equations for modeling system dynamics and observations, the EKF employs nonlinear functions. To manage the nonlinearity, the EKF linearizes these nonlinear functions around the current state estimate using a Taylor series approximation, enabling the EKF to carry out predictions and updates for the state estimates [29].

However, it is worth noting that while the EKF can handle mild nonlinearities, it may perform poorly or even diverge in the face of strong nonlinearities or if the initial guess is too far from the truth. Similarly, if the actual dynamics depart significantly from the linear approximation, the EKF can yield poor results [30].

2.3.1 IMU Sensor Fusion

IMU sensor fusion, with the help of algorithms like the KF and EKF, is utilized in various applications to improve the accuracy of measurements from multiple sensors like accelerometers, gyroscopes, and magnetometers. These sensors can be prone to noise and other inaccuracies while measuring linear acceleration, angular velocity, and magnetic fields to estimate an object's motion and orientation. The KF and EKF algorithms leverage a probabilistic framework to manage the uncertainty of these measurements, creating more precise estimates of the object's state. They combine data from different sensors, resulting in a better understanding of the object's motion and orientation than what could be obtained from individual sensors [31].

2.3.2 IMU Sensor Fusion Using MATLAB®

MATLAB®'s inertial sensor fusion algorithms use IMU data from multiple sensors and even combine IMU data with Global Positioning System (GPS) information when available. This fusion process involves integrating data from these various sources, comparing the results, and then using a mathematical model, such as KF or EKF, to estimate the most probable current state of the system. These algorithms are tailored to sensor setups, output needs, and motion limitations. Additionally, the filter parameters can be fine-tuned using accurate, known, or ground truth data, enhancing the overall accuracy of the estimates [32].

imufilter

The `imufilter` function employs a 6-axis KF structure, as detailed in [33]. The algorithm aims to track errors in orientation, gyroscope offset, and linear acceleration to determine the final orientation and angular velocity [34]. The filter is based on a model of errors rather than a model of the process itself. It offers the advantage of maintaining the high dynamic response necessary for attitude state variables, a key requirement in applications such as flight control or robotic navigation [31]. The state vector, x_k, is defined as:

$$x_k = \begin{bmatrix} \theta_k \\ b_k \\ a_k \end{bmatrix} \tag{17}$$

where θ_k is 3×1 orientation error vector, b_k is 3×1 gyroscope zero angular rate bias vector, and a_k is 3×1 acceleration error vector at time k. More details can be found in [34].

insfilterMARG

The `insfilterMARG` function utilizes an EKF with a 22-element state vector. This filter monitors the orientation quaternion, velocity, position, MARG (Magnetic, Angular rate, Gravity) sensor biases, and the geomagnetic vector. It performs sensor fusion using MARG and GPS data to calculate pose within the typically (North-East-Down) NED[1] reference frame. The MARG data usually is measured by magnetometers, gyroscopes, and accelerometer sensors [35].

---

[1] In this context, the North axis corresponds to the X-axis, the East Axis corresponds to the Y axis, and the Down axis corresponds to the Z axis.

The EKF's state vector, x, is defined as:

$$x = \begin{bmatrix} q \\ p_{NED} \\ v_{NED} \\ \Delta\theta bias_{XYZ} \\ \Delta v bias_{XYZ} \\ geomagneticFieldVector_{NED} \\ magbias_{XYZ} \end{bmatrix} \tag{18}$$

where q is the 4-element orientation quaternion, p_{NED} is the object's position in the NED coordinate system, v_{NED} is the object's velocity in the NED coordinate system, $\Delta\theta bias_{XYZ}$ represents the bias in the integrated gyroscope readings, $\Delta v bias_{XYZ}$ represents the bias in the integrated accelerometer readings, $geomagneticFieldVector_{NED}$ is the estimate of the geomagnetic field vector at the reference location, and $magbias_{XYZ}$ represents the bias in the magnetometer readings [35].

The function `insfilterMARG` predicts the state estimation of the velocity vector as follows:

$$\begin{aligned} v_{N_{k+1}} = v_N + (\Delta t)(g_N) + (\Delta v_X - \Delta v bias_X)(q_0^2 + q_1^2 - q_2^2 - q_3^2) \\ -2(\Delta v_Y - \Delta v bias_Y)(q_0 q_3 - q_1 q_2) \\ +2(\Delta v_Z - \Delta v bias_Z)(q_0 q_2 + q_1 q_3) \end{aligned} \tag{19a}$$

$$\begin{aligned} v_{E_{k+1}} = v_E + (\Delta t)(g_E) + (\Delta v_Y - \Delta v bias_Y)(q_0^2 - q_1^2 + q_2^2 - q_3^2) \\ +2(\Delta v_X - \Delta v bias_X)(q_0 q_3 + q_1 q_2) \\ -2(\Delta v_Z - \Delta v bias_Z)(q_0 q_1 - q_2 q_3) \end{aligned} \tag{19b}$$

$$\begin{aligned} v_{D_{k+1}} = v_D + (\Delta t)(g_D) + (\Delta v_Z - \Delta v bias_Z)(q_0^2 - q_1^2 - q_2^2 + q_3^2) \\ -2(\Delta v_X - \Delta v bias_X)(q_0 q_2 - q_1 q_3) \\ +2(\Delta v_Y - \Delta v bias_Y)(q_0 q_1 + q_2 q_3) \end{aligned} \tag{19c}$$

where Δt represents the IMU sample time, and g_N, g_E, g_D denotes the constant gravity vector in the

NED reference frame. The integrated accelerometer readings are represented by $\Delta v_X, \Delta v_Y, \Delta v_Z$. Other variables, including those not explicitly mentioned here, are defined similarly as in equation 18. Considering that the change in orientation is small, the rotation vector of the orientation can be derived as $\begin{bmatrix} \Delta\theta_X - \Delta\theta bias_X & \Delta\theta_Y - \Delta\theta bias_Y & \Delta\theta_Z - \Delta\theta bias_Z \end{bmatrix}$, where $\Delta\theta_X, \Delta\theta_Y, \Delta\theta_Z$ are the integrated gyroscope readings [35].

2.4 Modelling the Rotating Puck's IMU Outputs

This section outlines a scenario to study the dynamics of a rotating hockey puck equipped with an IMU. The following analysis of sensor outputs from the accelerometer and gyroscope provides crucial data points for state estimation. The calculated changes in position, velocity, acceleration, and angular velocity form the basis of the subsequent sensor fusion process.

2.4.1 Scenario Declaration

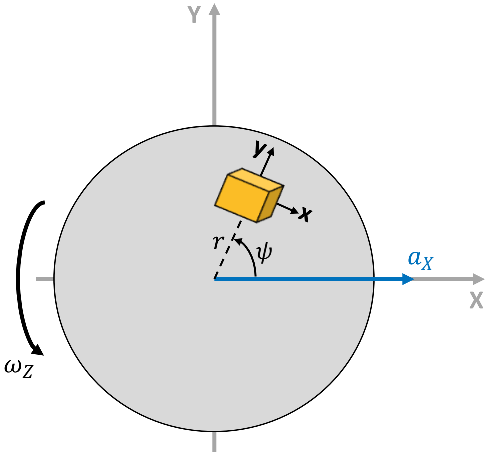


Figure 2.6: Implementing an IMU on a hockey puck.

The scenario involves a hockey puck with an IMU placed $1\ mm$ off-center, represented as r. The IMU, composed of a 3-axis accelerometer and a 3-axis gyroscope, operates at a sampling frequency of 150 Hz to record the puck's dynamics.

Upon being struck by a hockey stick, the puck undergoes an initial acceleration of $450\ m/s^2$, enabling it to attain a speed of approximately 30 m/s within $\frac{1}{15}$ s. Subsequently, the puck decelerates at $-10\ m/s^2$ for 3 seconds, finally coming to a halt 136 m from its starting point. Concurrently, it spins around the z-axis at a rate of 1000 $^{\circ}/s$ or around 17.5 rad/s.

As Figure 2.6 highlights, the puck's linear acceleration is denoted by a_X, while ω_Z indicates its angular velocity. In this context, capitalized coordinates (X, Y, Z) represent vectors in the Earth's FOR, while lowercase coordinates (x, y, z) denote vectors in the IMU's FOR.

2.4.2 IMU's Output

Considering the accelerometer in the Figure 2.6:

- The change in position along the X-axis, denoted as Δd_X, can be calculated based on the equation of motion $\Delta d_X = \frac{1}{2}a_X t^2 + v_{0_X} t + x_0$. However, since there are no initial velocities or positions, Δd_X simplifies to $\Delta d_X = \frac{1}{2}a_X t^2$. In the presence of rotation, an additional term arises due to the off-center placement of the accelerometer, resulting in $\Delta d_X = \frac{1}{2}a_X t^2 + r\cos\psi = \frac{1}{2}a_X t^2 + r\cos(\omega_Z t)$.
- For the Y-axis, the change in position, Δd_Y, is dominated by the rotational motion and can be expressed as $\Delta d_Y = r\sin\psi = r\sin(\omega_Z t)$.
- As for the Z-axis, there is no movement, indicating $\Delta d_Z = 0$.

Consolidating these results, the following equation is obtained for the change in position in a 3-D frame:

$$\Delta d = \begin{bmatrix} \frac{1}{2}a_X t^2 + r\cos(\omega_Z t) \\ r\sin(\omega_Z t) \\ 0 \end{bmatrix} \tag{20}$$

The velocity and acceleration are obtained through the differentiation of the position data. The velocity, denoted as v, can be calculated as the derivative of the position with respect to time, which gives:

$$v = \frac{dx}{dt} = \begin{bmatrix} a_X t - r\omega_Z \sin(\omega_Z t) \\ r\omega_Z \cos(\omega_Z t) \\ 0 \end{bmatrix} \tag{21}$$

Similarly, the acceleration, denoted as a, can be obtained as the derivative of the velocity with respect to time, yielding:

$$a = \frac{dv}{dt} = \begin{bmatrix} a_X - r\omega_Z^2 \cos(\omega_Z t) \\ -r\omega_Z^2 \sin(\omega_Z t) \\ 0 \end{bmatrix} \tag{22}$$

Furthermore, the effects of gravity can be incorporated into the acceleration vector, as shown in the following equation.

$$a = \begin{bmatrix} a_X - r\omega_Z^2 \cos(\omega_Z t) \\ -r\omega_Z^2 \sin(\omega_Z t) \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} = \begin{bmatrix} a_X - r\omega_Z^2 \cos(\omega_Z t) \\ -r\omega_Z^2 \sin(\omega_Z t) \\ g \end{bmatrix} \tag{23}$$

where g represents the gravitational acceleration and equals $9.81\ m/s^2$.

It is essential to mention that these equations apply when the accelerometer's FOR is fixed to the Earth's frame. However, as the puck rotates, so does the accelerometer's frame (see section 2.1.1). With the accelerometer's x-axis aligned with the rotation radius, the system dynamics can be depicted accurately by rotating the acceleration data using quaternion mathematics.

Using the quaternion rotation operator from equation (10), the acceleration data is rotated to the sensor's FOR. This transformation aligns with the motion scenario in section 2.4.1 and is given by:

$$\begin{aligned} &q = \cos\frac{\omega_Z t}{2} + 0\boldsymbol{i} + 0\boldsymbol{j} + \sin\frac{\omega_Z t}{2}\boldsymbol{k}, \\ &a_{sensor} = L_q(a) = \\ &\begin{bmatrix} \cos(\omega_Z t) & \sin(\omega_Z t) & 0 \\ -\sin(\omega_Z t) & \cos(\omega_Z t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_X - r\omega_Z^2 \cos(\omega_Z t) \\ -r\omega_Z^2 \sin(\omega_Z t) \\ g \end{bmatrix} = \begin{bmatrix} a_X \cos(\omega_Z t) - r\omega_Z^2 \\ -a_X \sin(\omega_Z t) \\ g \end{bmatrix} \end{aligned} \tag{24}$$

In the rotated accelerometer's vector, the second part of the x component, $r\omega_Z^2$, represents the centripetal acceleration, which consistently points towards the center of rotation. Moreover, Kok *et al.* in their study [23] elaborate on how other forms of acceleration, such as centrifugal acceleration and Coriolis accelerations, are incorporated into the accelerometer output. Furthermore, they discuss how the effects of both the centrifugal and Coriolis accelerations are generally negligible in real-world sensor data.

While the accelerometer provides critical data on the linear pose, the gyroscope measures the rotation rate; specifically, it detects the angular velocity. Hence, the gyroscope's output in this scenario is:

$$Angular\ Velocity = \begin{bmatrix} 0 \\ 0 \\ \omega_Z \end{bmatrix} \tag{25}$$

2.5 Velocity Estimation Using IMU

This section will discuss three distinct methods for estimating velocity using an IMU. These methods are applied to a moving hockey puck example, as outlined in section 2.4.1.

2.5.1 Accelerometer-based Method

A straightforward approach to determining the puck's velocity involves integrating the accelerometer readings. These measurements are situated within the puck's frame of reference. This method is notable for its independence from the gyroscope data, bypassing challenges associated with integrating gyroscopic readings, such as noise and synchronization with the accelerometer, among other potential complications (refer to section 2.1.2).

Researchers have demonstrated that the acceleration in Earth's FOR, represented as a_{Earth}, can be obtained by applying a 4^{th}-order Butterworth high-pass filter with a cutoff frequency, $\omega_0 = 0.2\ Hz$, to the accelerometer's readings [36]. The subsequent equation signifies the accelerometer's measurements in equation (24) once their average values have been ideally eliminated.

$$a_{Earth} = a_{sensor} - \overline{a}_{sensor} = \begin{bmatrix} a_X \cos(\omega_Z t) \\ -a_X \sin(\omega_Z t) \\ 0 \end{bmatrix} \tag{26}$$

The velocity vectors illustrated in Figure 2.7 are derived from the above acceleration vector.

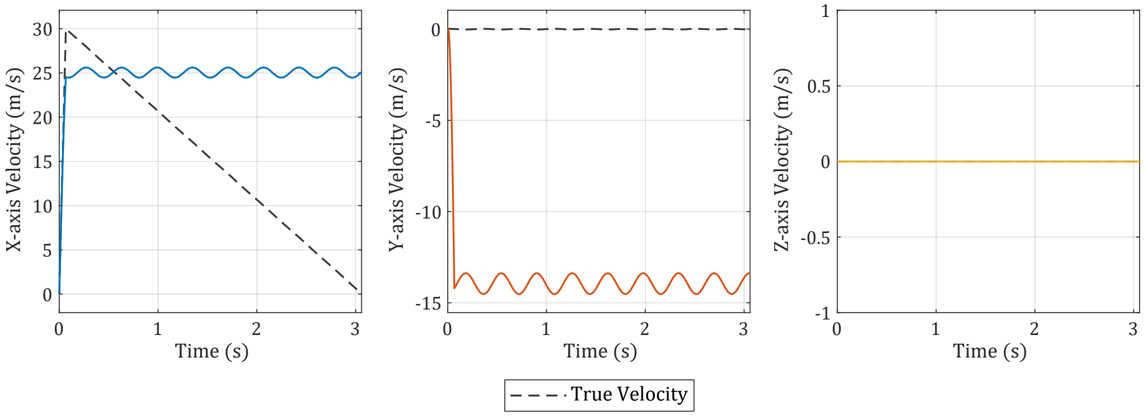


Figure 2.7: Velocity vectors in the Earth's FOR estimated using the accelerometer-based method, compared to true velocity.

As depicted in Figure 2.8, the maximum velocity magnitude estimated using the accelerometer-based method is approximately $28.34\ m/s$, which deviates from the true velocity magnitude by -5.5%. Moreover, this method exhibits limitations when the gravity component appears in the x-axis and y-axis of the accelerometer readings, indicating the presence of rotations in more than one axis. Despite these constraints, the simplicity and directness of this method make it an invaluable initial step in velocity estimation, serving as a basis for more sophisticated techniques.

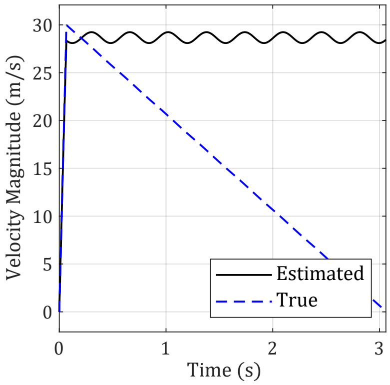


Figure 2.8: Magnitude of the estimated velocity of the puck from accelerometer readings using the accelerometer-based method, compared to the true velocity.

2.5.2 Accelerometer-and-Gyroscope-based Method

This approach primarily involves transforming the accelerometer data from the puck's FOR to the Earth's FOR using the gyroscope output and corresponding mathematical operations, such as rotation matrices. This transformation is utilized to better account for the puck's rotation's effects and estimate its velocity more accurately.

The Dead-reckoning Algorithm

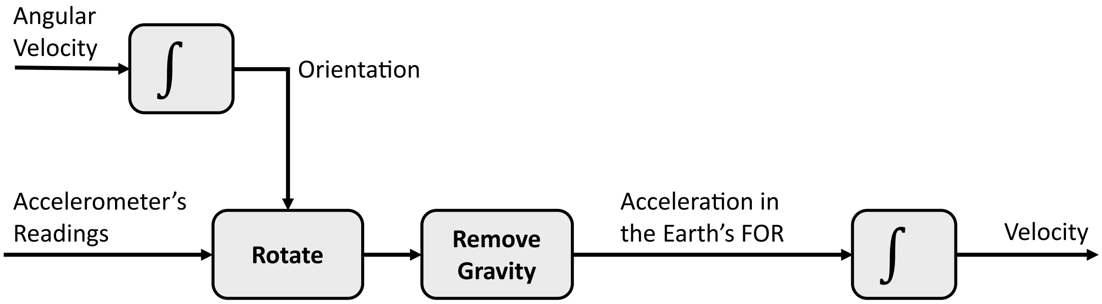


Figure 2.9: Dead-reckoning algorithm, showcasing integrating accelerometer and gyroscope measurements to yield velocity estimation [23].

As Figure 2.9 shows, the dead-reckoning process involves integrating the gyroscope's readings. This yields the puck's orientation. Following the rotation of the puck's FOR to the Earth's FOR and removing the gravitational acceleration, the sensor's velocity can be inferred by integrating accelerometer measurements. However, IMU readings are susceptible to noise and biases in practical applications. This means that "integration drift" appears when the integration steps to convert angular velocity into orientation and acceleration into velocity are performed, impacting the estimation accuracy [23].

For the previously described scenario, the rotated accelerometer reading to the Earth's FOR is derived as:

$$
\begin{aligned}
&q = \cos\frac{\omega_Z t}{2} - 0\boldsymbol{i} - 0\boldsymbol{j} - \sin\frac{\omega_Z t}{2}\boldsymbol{k},\\
&a_{Earth} = L_q(a) = \\
&\begin{bmatrix} \cos(\omega_Z t) & -\sin(\omega_Z t) & 0 \\ \sin(\omega_Z t) & \cos(\omega_Z t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_X \cos(\omega_Z t) - r\omega_Z^2 \\ -a_X \sin(\omega_Z t) \\ g \end{bmatrix} = \begin{bmatrix} a_X - r\omega_Z^2 \cos(\omega_Z t) \\ -r\omega_Z^2 \sin(\omega_Z t) \\ -g \end{bmatrix}
\end{aligned}
\tag{27}
$$

where q is the inverse of the orientation quaternion in equation (24). The next step in the dead-reckoning process is to remove the effect of gravity and then integrate the acceleration.

$$v = \int a_{Earth}\, dt = \begin{bmatrix} a_X t - r\omega_Z \sin(\omega_Z t) \\ r\omega_Z \cos(\omega_Z t) \\ 0 \end{bmatrix} \tag{28}$$

where v is the puck's velocity vector in the Earth's FOR. Figure 2.10 represents v throughout this scenario.

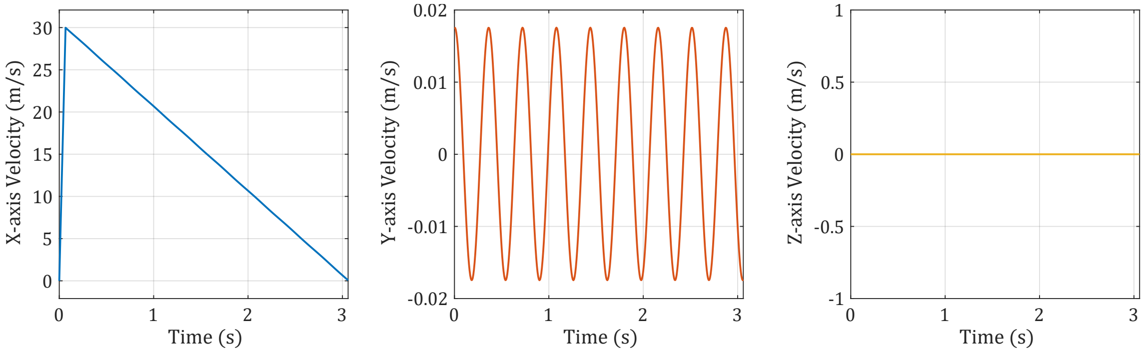


Figure 2.10: Puck's velocity vectors in the Earth's FOR, derived using the accelerometer-and-gyroscope-based method.

The magnitude of v, denoted as $|v|$, is depicted in equation (29) and illustrated in Figure 2.11.

$$\|v\| = \sqrt{\Big(a_X t - r\omega_Z \sin(\omega_Z t)\Big)^2 + \Big(r\omega_Z \cos(\omega_Z t)\Big)^2} \approx |a_X t|, \text{ when } a_X t \gg r\omega_Z \tag{29}$$

As illustrated in Figure 2.11, this method estimates a peak velocity of approximately 29.98 m/s, demonstrating a marginal error of only -0.06% from the true velocity. This highlights the method's superior performance over the accelerometer-based method, confirming the advantage of utilizing both accelerometer and gyroscope readings in velocity estimation.

Estimate Orientation Using imufilter

This approach follows the same fundamentals described, transforming accelerometer readings from the puck's FOR to the Earth's FOR. However, orientation estimation is employed by the

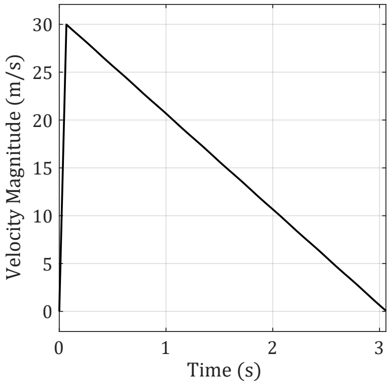


Figure 2.11: Magnitude of the puck's velocity in the Earth's FOR, derived using the accelerometer-and-gyroscope-based method.

`imufilter` function (refer to section 2.3.2). This method excels with real-world data due to its capacity to mitigate the effects of noise and bias inherent in such data. Notably, in the context of the simulated puck scenario, the use of `imufilter` may not provide significant benefits. The scenario data is free from noise or bias, rendering the noise and bias mitigation features of the `imufilter` less relevant.

2.5.3 MARG-based Method

The `insfilterMARG` method (refer to section 2.3.2) merges data from the accelerometer, gyroscope, and magnetometer. By utilizing an EKF, this method can estimate the orientation and velocity vector more accurately, especially for real-world data with sensor noise and bias. Although not applied to the scenario data within this chapter, the `insfilterMARG` approach will be utilized in the subsequent chapters for the actual puck velocity estimation.

2.6 Current Puck Sensors

2.6.1 Piezoelectric 3-axis Accelerometer

Researchers at McGill University employed a piezoelectric 3-axis accelerometer, depicted in Figure 2.12, on hockey pucks to measure the puck's acceleration, velocity, etc. Pucks were modified to embed the accelerometer at their center while retaining standard weight and size. The accelerometer specified as follows:

- linear acceleration range: ±500 g
- sensitivity: 10 mV/g
- shock tolerance: 1000 g

was utilized to acquire precise puck acceleration data during each trial [37].

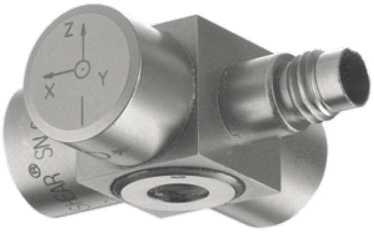


Figure 2.12: The piezoelectric 3-axis accelerometer (Model 8792A500, Kistler Instrumentation Co., Amherst, NY) used in the study [37].

The captured measurements were relayed through a charge amplifier, connected to the sensor via an extended cable, to an Analog-to-Digital Converter (ADC) board that collected data at 10 kHz. This data was then transferred to a computer for analysis. A simultaneous recording of high-speed video was conducted to synchronize the sensor data with the corresponding video footage. A blade-to-puck contact circuit was also implemented, ensuring synchronized data acquisition across the two recording systems [37].

Participants in the study were nine 9 volunteers divided into two groups: an "elite" group of four with significant competitive experience and a "recreational" group of five with only intramural playing experience. The findings from the research showed significant differences in puck velocity between elite and recreational players, with the elite group achieving a higher average puck velocity of 33.6±5 m/s compared to the recreational group's 22.3±3.2 m/s. However, when it came to the average puck acceleration, no significant differences were found between the two groups. The elite group demonstrated a slightly higher average of 63.8 ± 9.9 g compared to the recreational players' 61.8 ± 19.5 g [37]. These results are detailed in the table below:

The puck's maximum velocity was obtained using the trapezoidal integration of the accelerometer data [37]. The trapezoidal integration method is a numerical integration technique that approximates the definite integral of a function. It works by dividing the area under the function's curve

Table 2.1: The comparative performance metrics between elite and recreational hockey players, detailing puck acceleration and velocity differences [37].

Level	Elite		Recreational	
	Mean	Standard Deviation	Mean	Standard Deviation
Average puck acceleration (g)	63.8	9.9	61.8	19.5
Peak puck velocity (m/s)	33.6	5	22.3	3.2

into several trapezoids and then summing the areas of these trapezoids to approximate the total area under the curve [38].

There are a few considerations with the methodology in [37]. For instance, the extended cable connecting the sensor to the ADC could subtly alter the puck dynamics and add minor system impedances. While initial tests indicated a 10% reduction in shot velocity due to the cable, this is merely an estimate, and actual effects may vary. Additionally, the focus on accelerometer data alone may not provide a comprehensive picture of the puck's dynamics. Integrating further metrics, such as orientation and angular velocity, could enrich the dataset, potentially enhancing the precision and scope of analysis.

2.6.2 Performance Tracking Hockey Puck

As part of a capstone project at Concordia University, a performance-tracking hockey puck system was developed [39]. This system integrated a 3-axis piezoelectric accelerometer with a range of $\pm 500\ g$ and a 3-axis MEMS gyroscope with a range of $\pm 4000\ ^{\circ}/s$, aimed at capturing both acceleration and angular Velocity of the puck. A microcontroller equipped with built-in Bluetooth Low Energy (BLE) capabilities was incorporated into the system for the wireless transmission of sensor data. The entire setup was powered using coin-sized batteries, ensuring a compact design.

An interface was developed to facilitate data acquisition, storage, and visualization. A custom design was implemented for the puck, creating a hollowed-out space to accommodate the sensor assembly. A specially devised plastic cap encapsulated the entire body, preserving the standard form of a hockey puck [39]. The final sensor implementation is illustrated in Figure 2.13.

The accelerometer was successfully operated at a sampling rate of $2\ kHz$, delivering an efficient 80 samples during the typical slap shot's contact time of $0.04\ s$. Despite these advancements, the team encountered difficulties operating the gyroscope sensor, thus preventing the acquisition

Figure 2.13: The final implementation of the performance tracking hockey puck [39].

of angular movement data during their trials [39]. Although no measurements related to hockey scenarios were reported to evaluate sensor performance, the accelerometer selection was deemed appropriate. Conversely, as outlined in section 2.7, the gyroscope is prone to saturation, which could affect the angular velocity measurements.

2.6.3 Neblina™ Bluetooth® Wireless Motion Sensor by Motsai

Developed by Motsai, the Neblina™ core module is a compact, low-power IMU with wireless connectivity. Motsai designed a custom sensor to fit inside a hockey puck in collaboration with Bauer Hockey. This cylindrical sensor is robustly encased in protective resins, measuring 41 mm in diameter and 15 mm in height. It is equipped with an accelerometer, ranging $\pm 16\ g$, and a gyroscope, ranging $\pm 2000\ °/s$, thus tracking the puck's linear and angular movements.

The sensor is powered by an inbuilt battery that features wireless charging. Motsai also developed a companion app that ensures real-time connectivity with the sensor, enabling data acquisition and analysis [40]. The sensor is depicted in Figure 2.14.

Although adequate for some applications, the selected accelerometer and gyroscope ranges may not be sufficiently robust for intensive hockey scenarios, as they could face saturation during specific test conditions. The protective resin encapsulating the Neblina™ core module significantly enhances its durability and perfect fit within the puck; nonetheless, it presents a few trade-offs. After numerous charging cycles, the inbuilt Lithium Polymer (LiPo) battery may experience swelling. This swelling could potentially compromise the structural integrity of the sensor system. Furthermore, the resin casing makes the sensor unrepairable, meaning that even minor issues arising during

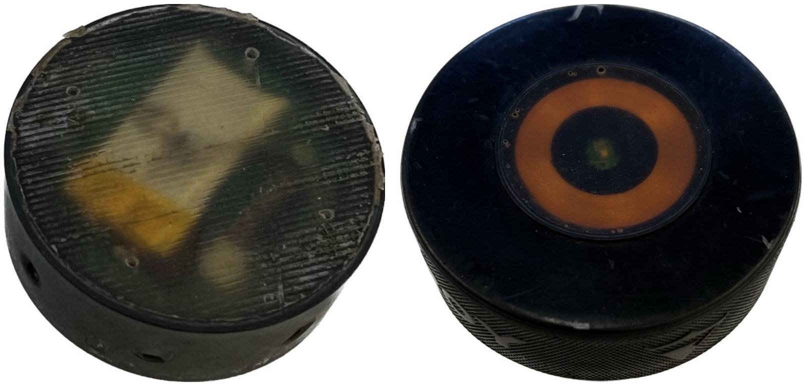


Figure 2.14: The custom Neblina core module developed for Bauer Hockey by Motsai [40].

its lifetime can make it entirely unavailable for further use.

2.6.4 Movesense Sensor HR+

The Movesense Sensor HR+, depicted in Figure 2.15, is a compact, versatile, and lightweight motion-sensing device suitable for various applications. This device integrates a 9-axis IMU system, comprising an accelerometer with a range of $\pm 16\ g$, a gyroscope with a range of $\pm 2000\ ^{\circ}/s$, and a magnetometer with a range of $\pm 50\ G$.

The sensor is powered by the Nordic Semiconductor nRF52832 microcontroller, which features an integrated BLE operating at $2.4\ GHz$ for wireless connectivity. Despite weighing only $10\ g$, the Movesense Sensor HR+ is robust enough to endure extreme sporting conditions. The device also includes smartphone applications and supports wireless firmware updates, enhancing its versatility and ease of use [41].

Figure 2.15: The Movesense Sensor HR+ [41].

The Movesense Sensor HR+ stands out with its compact and lightweight design, repairable parts, long battery life, and the added convenience of a replaceable battery. However, as outlined in section 2.7, the accelerometer and gyroscope range might be inadequate for intense hockey shots. During high-power trials, there's a risk that these sensors could approach their limits, resulting in saturation and potentially compromising the precision of the data collected.

2.7 Required Sensor Specifications

This section discusses the requirements for the IMU to capture the motion parameters of the puck effectively. Additionally, it explores the mechanical specifications necessary to accommodate the unique environmental and operational conditions inherent to the puck's usage.

Acceleration

The results from the study [37], as outlined in section 2.6.1, provide crucial insights into the dynamics of hockey puck motion. The study unveiled that the maximum acceleration achieved during the trials was around $81.3\ g$ for both player groups. Thus, the accelerometer used must be capable of measuring accelerations of this magnitude. The findings are outlined in table 2.1.

Angular Velocity

In the study [42], the puck's angular velocity, often referred to as "spin", was measured during hockey wrist shots. The experiment precisely captured the puck's trajectory using the Vicon MX digital motion camera system. The angular velocity of the puck was found to vary significantly across trials, where the maximum value was $6131 \pm 227\ ^\circ/s$. As a result, the gyroscope used must be capable of measuring angular velocities up to this value. The analysis further revealed a strong correlation between higher puck spin and greater shot accuracy, underlining the influence of the puck's angular velocity on the trajectory and accuracy of hockey wrist shots.

Physical Requirements

Integrating a sensor system into a hockey puck represents a technical challenge. It warrants careful consideration to maintain the puck's official dimensions, a cylinder with a diameter of $76.2\ mm$, a height of $25.4\ mm$, and a weight of $156\ g$ to $170\ g$. To ensure the integrity of the research and the data's validity, the sensor system must be embedded within the puck without altering its physical attributes. Accordingly, the sensor's dimensions should be designed to fit inside the puck.

Considering the nature of this research, which may involve dynamic movements and real-time data acquisition, a robust and reliable wireless data transmission system is essential. This system must function consistently and accurately during experiments. Moreover, it should support continuous measuring for a minimum of 8 hours to accommodate researchers' needs. Using a battery-powered, wireless sensor encapsulated within the puck offers a discreet and effective solution for tracking the dynamic motion of the puck. Considering the dimensions of a typical ice hockey rink, the sensor's wireless range of approximately 40 meters suffices for capturing data across the play area. The following table summarizes the requirements.

Table 2.2: Summary of Required Sensor Specifications.

Parameter	**Requirement**
Acceleration	Capable of measuring up to $81.3\ g$
Angular Velocity	Capable of measuring up to 6358 $^\circ/s$
Wireless Range	Approximately 40 meters
Operating Time	Supports continuous measurement for a minimum of 8 hours
Physical Dimensions	Must fit inside a standard hockey puck (diameter: $76.2\ mm$, height: $25.4\ mm$)
Weight	Should maintain the standard puck weight (156 g to 170 g) after sensor integration

Chapter 3

Materials and Methods

This chapter aims to delve into the details of designing a sensor system that can be integrated into a standard hockey puck while preserving its official dimensions and weight. Given the demanding dynamics involving high speed, intense impacts, and rapid directional changes, the sensor design must ensure robustness, reliability, and accurate data capture and real-time transmission. Developing such a sensor requires a subtle understanding of engineering and sport. The design process involves selecting appropriate sensors, ensuring their functional operation, devising an efficient method for wireless data transmission, and utilizing the proper velocity estimation algorithms.

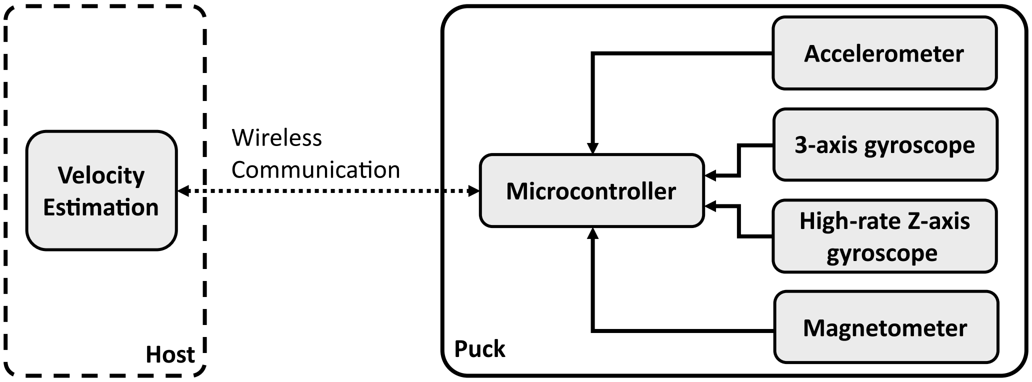


Figure 3.1: The diagram view of the proposed sensor configuration.

As shown in Figure 3.1, the proposed sensor design integrates an accelerometer, two gyroscopes, and a magnetometer, collectively enabling three methods for velocity estimation (refer to section 2.5). Additionally, a microcontroller is used to collect, process, and forward sensor data to the host, ensuring efficient real-time transmission without compromising the puck's structural integrity.

Gyroscope Integration

For the application delineated in Chapter 2, a 3-axis gyroscope capable of measuring the specified range is essential. Unfortunately, no commercial sensor meeting this specification was found. An alternative sensor was identified, which offers measurements over a more extensive range but only for one axis. However, using three such sensors is impractical since the pins do not align conventionally on the circuit board for the X and Y axes, as demonstrated in Figure 3.2. Consequently, combining the 3-axis gyroscope and a high-rate Z-axis gyroscope was adopted to ensure a comprehensive range of angular velocities.

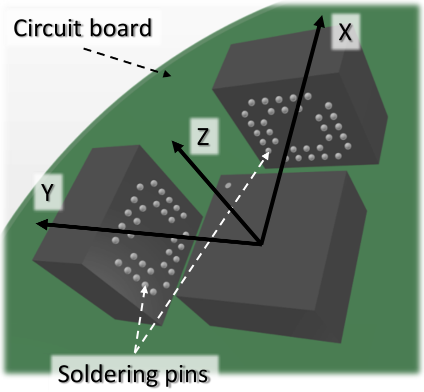


Figure 3.2: Alignment of the high-rate gyroscope's pins for 3-axis angular velocity measurement, demonstrating non-conventional positioning on the board for X and Y axes.

3.1 Accelerometer

To select the accelerometer for capturing the puck's dynamics, the following primary factors were considered:

- *Axes:* It should be a tri-axial sensor to capture the motion in three dimensions.
- *Range:* As mentioned in section 2.7, the puck can experience accelerations up to 81.3 g during trials. Thus, the sensor should be capable of measuring beyond this range to ensure

all possible movements are captured. However, selecting a range significantly higher can diminish accuracy and resolution.

- *Type:* For detecting delicate motions, piezoelectric accelerometers are often superior over MEMS, as a trade-off with a higher cost [43]. Thus, the piezoelectric type is preferred.
- *Size:* Considering the limited space inside the puck, the accelerometer selected is required to be compact and board-mountable.

Given these requirements, the accelerometer model 830M1-0100 from TE Connectivity has been selected for this application. The following list provides an overview of the critical characteristics of the chosen accelerometer [44]:

- The 830M1-0100 accelerometer provides a measurement range of $\pm 100\ g$ and exhibits a uniform frequency response exceeding 15 kHz across all three axes.
- The Piezoelectric (PE) technology employed in the 830M1 accelerometer delivers reliable and consistently stable output, making it highly suitable for condition monitoring applications. This accelerometer outperforms MEMS devices due to its superior resolution, dynamic range, and bandwidth.
- This accelerometer features three separate, stable piezoceramic crystals. This independent configuration provides an inherent advantage, as fluctuations in one crystal yield minimal interference with the others, thereby promoting overall system reliability.
- The sensor's output is analog, exhibiting a 2.5 V range. Additionally, each sensor undergoes a precise calibration process by the manufacturer to ensure precision and reliability.

3.1.1 Operation Circuit

Figure 3.3 showcases the accelerometer's operational circuit schematic.

- The power supply voltage (VDD) is set at 5 V, supplemented by a decoupling capacitor to manage potential voltage fluctuations.

- The outputs from the accelerometer are directly interfaced with an Analog-to-Digital Converter (ADC) for further processing and analysis. The ADC is elaborated upon in section 3.5.
- The accelerometer has a resonant frequency of 30 kHz [44]. Such a high frequency could introduce unwanted noise and distortions into the accelerometer readings. Therefore, as specified in section 3.5.3, an appropriate filtering mechanism is implemented to attenuate the signal frequencies above the operational range and below the resonant frequency before the signals are fed into the ADC.

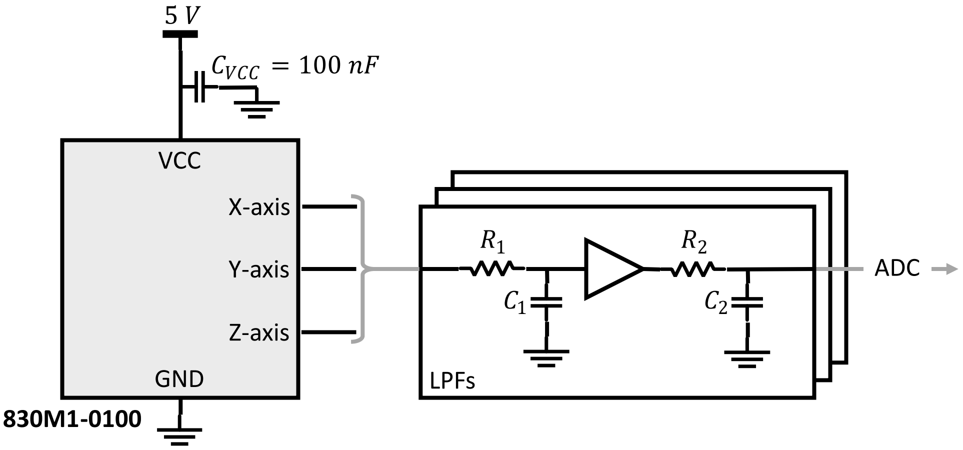
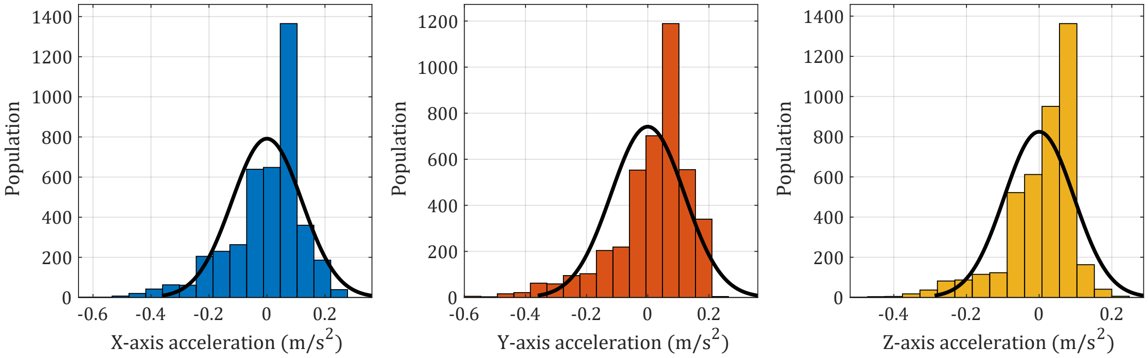


Figure 3.3: The accelerometer sensor schematic design.

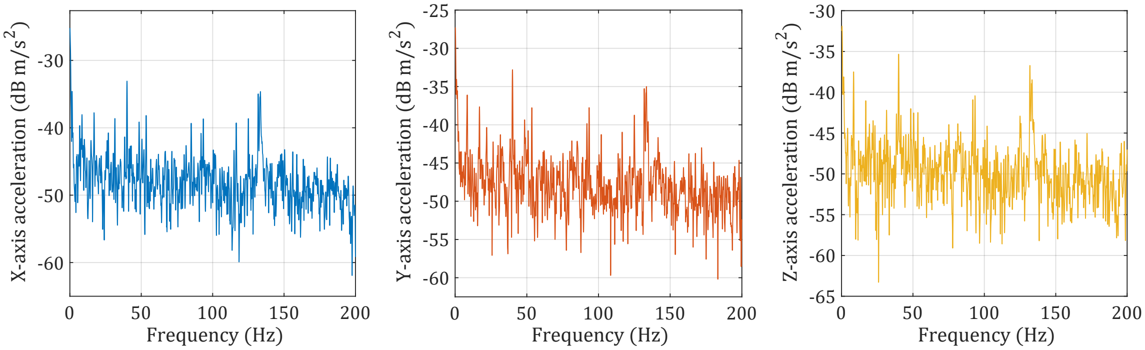
3.1.2 Output Signal

Measurement was carried out over a 10-second period under ideal environmental conditions, i.e., nonmoving, to validate the accelerometer's operation and noise level. For this test, the LPFs were bypassed to exclude any potential signal alterations caused by these filters, and the accelerometer's output was directly fed into the ADC. The results of this test are depicted in Figure 3.4.

Based on the results illustrated in Figure 3.4, Table 3.1 presents a summary of the accelerometer's noise parameters. These parameters are compared against the respective specifications mentioned in the datasheet. As the table demonstrates, the output noise of the accelerometer, as measured in this test, aligns with the noise values specified in the datasheet, thereby validating the performance and reliability of the accelerometer.



(a) Histogram and normal distribution fit



(b) Power spectral density

Figure 3.4: Accelerometer's output under ideal environmental conditions.

Table 3.1: Noise parameters of the accelerometer as compared to datasheet values.

Parameter	**Datasheet Value**	**Measured Value**		
		X-axis	Y-axis	Z-axis
Total RMS noise $(m/s^2{}_{RMS})$	0.115	0.11	0.11	0.10
Spectral noise @ 10 Hz $(10^{-3} \times m/s^2/\sqrt{Hz})$	5.7	4.23	3.48	1.42
Spectral noise @ 100 Hz $(10^{-3} \times m/s^2/\sqrt{Hz})$	2.65	2.62	2.55	2.48

3.2 3-axis Gyroscope

In line with the specifications delineated in section 2.7, the ITG-3701, a 3-axis MEMS gyroscope from TDK InvenSense, was chosen. The selection was based on the following criteria:

- *Axes:* The requirement for a sensor that can monitor angular velocities across the X, Y, and Z axes.

- *Range:* This sensor, known for possessing the world's most comprehensive 3-axis full-scale range, closely meets the specifications, mainly for the X and Y axes. Furthermore, this gyroscope is also used for tracking the Z-axis, provided that the angular velocity stays within the sensor's designated range.

The ITG-3701 gyroscope's suitability emerges from these criteria. Its salient features, as explored in detail below, further affirm its selection [45]:

- This sensor is built on a single-structure vibratory MEMS rate gyroscope, detecting rotational movements around the X, Y, and Z axes. The Coriolis Effect discussed in section 2.1.2 causes a detectable vibration upon rotation. This vibration is transformed into a proportional voltage signal via amplification, demodulation, and filtering. The analog signal is then digitized by individual on-chip 16-bit ADCs for each axis.
- The sensor supports programmable full-scale ranges of ±500, ±1000, ±2000, and ±4000 dps. User-selectable low-pass filters offer various cut-off frequencies, and the ADC sample rate can be modified between 125 Hz to 32 kHz.
- The ITG-3701 features both I^{2}C and SPI interfaces and operates in slave mode during interactions with the system processor.

3.2.1 Operation Circuit

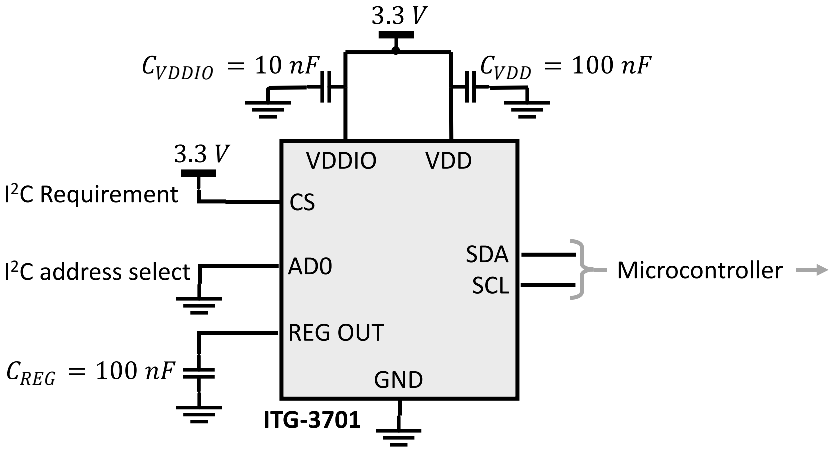


Figure 3.5: The 3-axis gyroscope sensor schematic design.

The operation circuit for the ITG-3701 has been designed following the guidelines provided by the sensor manufacturer. As part of the setup, the sensor's VDD and VDDIO are connected to a 3.3 V supply, accompanied by decoupling capacitors to stabilize the power input. Also, the Serial

Clock (SCL) and Serial Data (SDA) lines, integral to the I^{2}C protocol, have been directly linked to the microcontroller, enabling data transfer at a frequency of $400\ kHz$. In addition, other considerations evident in Figure 3.5 have been incorporated, adhering to the recommendations provided in the sensor's datasheet [45].

3.2.2 Software Driver

In the context of microcontroller software development and utilizing the `nrf_drv_twi` library from the nRF SDK, the software driver for the ITG-3701 has been developed. This includes routines for sensor initialization, data extraction, and power management.

Initialization

The following processes are performed each time the sensor is powered on to initiate and prepare it for operation:

(1) Initial communication with the sensor is verified, ensuring the correct sensor identification.

(2) The internal Phase-Locked Loop (PLL) is chosen as the clock source, in line with the required sensor performance.

(3) An internal Low-Pass Filter (LPF) with a bandwidth of $250\ Hz$ is chosen, aligning with the desired time delay for the gyroscope of $0.97\ ms$.

(4) To achieve accurate readings, a sample rate of 400 samples per second is selected (refer to section3.6.2).

(5) For the required motion tracking, the sensor's full-scale range is set to $\pm 4000\ ^\circ/s$.

(6) The 'data-ready interrupt' is enabled to signal the availability of the latest gyro sensor data, ensuring timely data retrieval.

Data Extraction

To extract the sensor data at each iteration, the procedure outlined below is executed:

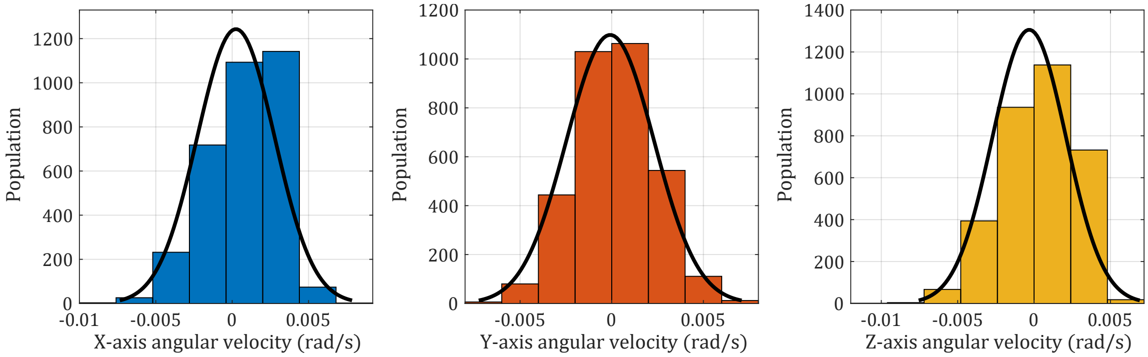
(1) The ‘data-ready interrupt’ bit in the ITG-3701’s internal register is examined, indicating the availability of new sensor data.

(2) Based on the positive indication, the X, Y, and Z output values from the gyroscope are extracted using 6 consecutive bytes.

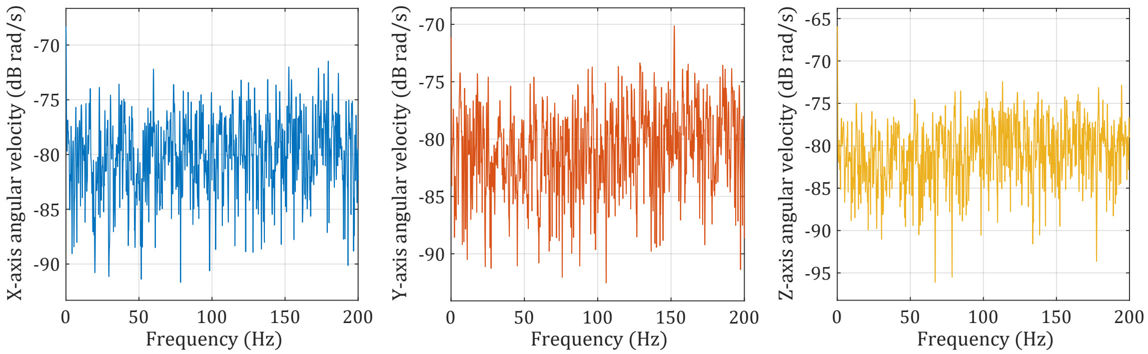
Power Management

To transition the ITG-3701 sensor into power-saving mode, the ‘sleep’ bit is activated. The sensor’s power consumption is substantially diminished in this sleep mode, drawing a minimal current of 8 μA.

3.2.3 Output Signal



(a) Histogram and normal distribution fit



(b) Power spectral density

Figure 3.6: 3-axis gyroscope’s output under ideal environmental conditions.

The gyroscope’s performance and noise level were evaluated using a similar methodology to the accelerometer (refer to section 3.1.2). Measurements were conducted over a 10-second period with the device held stationary. The results of this test are presented in Figure 3.6.

Moreover, following the data acquisition, the noise parameters of the gyroscope were extracted and compared against their datasheet values, as summarized in Table 3.2. The results confirm that the measured output noise of the gyroscope is consistent with the datasheet specifications.

Table 3.2: Noise parameters of the 3-axis gyroscope compared to datasheet values.

Parameter	**Datasheet Value**	**Measured Value**		
		X-axis	Y-axis	Z-axis
Total RMS noise $(10^{-3} \times rad/s_{RMS})$	3.49	2.53	2.39	2.41
Spectral noise @ 10 Hz $(10^{-3} \times rad/s/\sqrt{Hz})$	0.35	0.06	0.06	0.14

3.3 High-rate Z-axis Gyroscope

Building on the specifications delineated in section 2.7, the hockey puck's maximum spin rate is $\sim 6500\ °/s$. This full rotational motion is primarily observed along the Z-axis. Therefore, besides the 3-axis gyroscope –with a range of $4000\ °/s$– discussed in section 3.2, a dedicated gyroscope for the Z-axis is considered essential to capture the rotational dynamics adequately. For this purpose, the ADXRS649 gyroscope from Analog Devices, the only commercial MEMS gyroscope available for this range, has been selected. The characteristics and operational parameters of this sensor are [46]:

- The ADXRS649 features a wide measurement range of $\pm 20000\ °/s$ with a sensitivity of $0.1\ mV/°/s$.

- The ADXRS649 gyroscope utilizes a resonator gyro principle with 4 polysilicon sensing structures. These structures contain a dither frame that resonates when electrostatically activated, inducing a Coriolis force with any Z-axis angular rate (as shown in Figure 2.2). This Coriolis force interacts with an outer sense frame, resulting in a capacitive pickoff structure that transduces motion into an electrical rate signal. The unique design of the ADXRS649 allows it to effectively reject external accelerations, thereby increasing output signal accuracy and stability.

- The output signal is analog, within a range of $4.5\ V$.

3.3.1 Operation Circuit

The following steps have been taken for operation of the ADXRS649 gyroscope, as detailed in Figure 3.7:

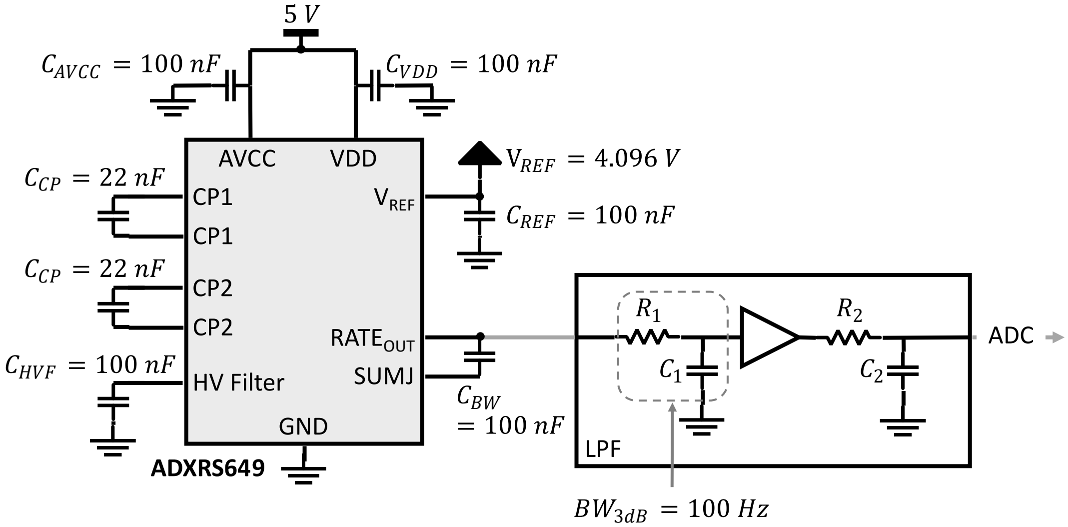


Figure 3.7: Schematic diagram of the ADXRS649 gyroscope circuit configuration.

- Considering the requirements outlined in the ADXRS649 sensor's datasheet, the electrostatic resonator of this sensor necessitates an operational voltage between 13 V and 15 V. However, the sensor has an on-chip charge pump with 5 V input [46]. Accordingly, the charge pump and the analog supply have been connected to a 5 V power source. To manage potential voltage fluctuations, decoupling capacitors are systematically deployed.

- The capacitors required for the charge pump have been chosen per the recommendations provided in the sensor's datasheet.

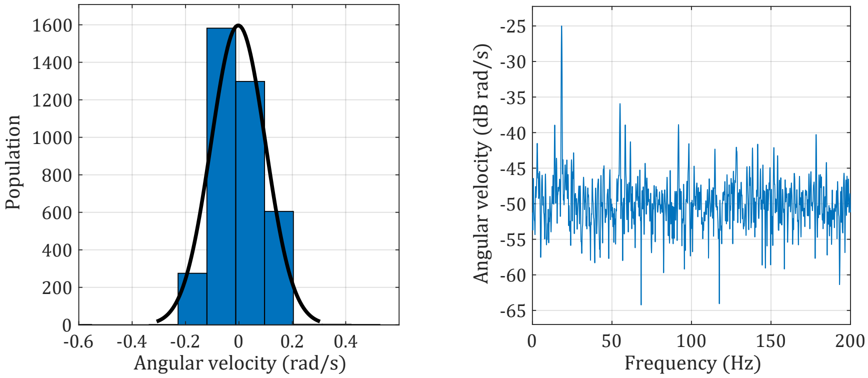
- The sensor's bandwidth has been confined to 10 Hz, a measure recommended by the manufacturer to reduce sensor noise without significantly attenuating the sensor's output signal.

- The sensor's output is directly interfaced with the ADC, discussed further in section 3.5, allowing the microcontroller to process and transmit the sensor's output for further analysis.

- According to the sensor datasheet, it is recommended to incorporate an additional LPF, as detailed in section 3.5.3. This helps to attenuate high-frequency noise that demodulation spikes of the resonant frequency might cause.

3.3.2 Output Signal

The ADXRS649 gyroscope was tested over a 10-second period under ideal environmental conditions to verify its operational performance and assess the noise level. To ensure an unaltered output signal, the LPF was bypassed, and the gyroscope's output was directly fed into the ADC. The results of this test are available in Figure 3.8.



(a) Histogram and normal distribution fit

(b) Power spectral density

Figure 3.8: High-rate Z-axis gyroscope's output under ideal environmental conditions.

Based on the results, the ADXRS649 gyroscope's noise parameters and a comparison with the datasheet values are summarized in Table 3.3.

Table 3.3: Noise parameters of the high-rate Z-axis gyroscope compared to datasheet values.

Parameter	**Datasheet Value**	**Measured Value**
Noise density ($10^{-3} \times rad/s/\sqrt{Hz}$)	4.36 ($BW = 80\ Hz$)	1.7 ($BW = 10\ Hz$)

The noise density is measured using the following formula:

$$Noise\ density = \left(\frac{\sum_{i=0}^{i=BW} Power\ spectral\ density_i}{BW} \right)^{\frac{1}{2}} \tag{30}$$

3.4 Magnetometer

To facilitate the velocity estimation method detailed in section 2.5.3, incorporating a 3-axis magnetometer into the design is necessary. Since the aim is to measure Earth's magnetic fields, which range between $0.3\ G$ and $0.6\ G$, a sensor offering precise readings in this range is required. The MMC5983MA, a 3-axis magnetometer from MEMSIC, fits this requirement due to its high precision characteristics. The applicable features of the MMC5983MA are detailed in the following list [47]:

- The MMC5983MA sensor can accurately measure magnetic fields within a full-scale range of $\pm 8\ G$, with a resolution of $0.0625\ mG$ per LSB at 18-bit operation mode. Moreover, this sensor has a digital low-pass filter, providing selectable bandwidths ranging from $100\ Hz$ to $800\ Hz$.
- The MMC5983MA sensor uses an on-chip magnetically-coupled strap to counteract changes in sensor behavior from magnetic fields over $10\ G$. The SET/RESET mechanism applies a high current pulse to create a stabilizing magnetic field, maintaining sensor performance in varying magnetic conditions.
- This sensor is equipped with I^{2}C and SPI protocols, enabling flexible and effective communication with the microcontroller.

3.4.1 Operation Circuit

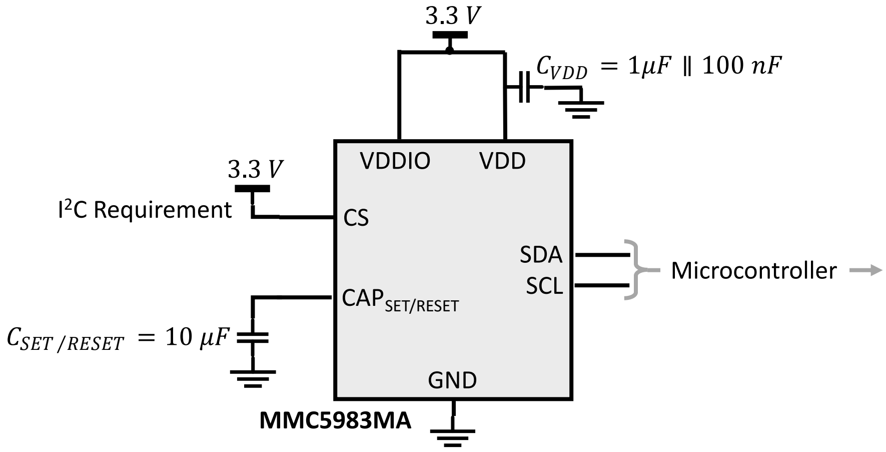


Figure 3.9: The magnetometer sensor schematic design.

For the operational circuit of the MMC5983MA sensor, both the VDD and VDDIO are connected to a $3.3\ V$ supply, with respective decoupling capacitors to stabilize the supply voltage and reduce noise. The sensor's SDA and SCL pins are interfaced directly with the corresponding pins on the microcontroller, facilitating communication between the two devices at $400\ kHz$. The external circuitry follows the specifications outlined in the sensor's datasheet to ensure optimal performance and reliability. The setup schematic is depicted in Figure 3.9.

3.4.2 Software Driver

A comprehensive software driver for the MMC5983MA magnetometer has been implemented in microcontroller software development. Leveraging the `nrf_drv_twi` library from the nRF SDK, this software driver is designed to handle initialization, data extraction, and power management.

Initialization

Each time the MMC5983MA magnetometer is powered on, the following procedures are executed to initialize and prepare it for operation:

(1) Initial communication with the sensor is verified to ensure a successful connection.

(2) The automatic SET/RESET function is activated to enhance the stability of sensor measurements.

(3) Adjustments are made to set the digital filter's bandwidth to $800\ Hz$ and shorten the measurement time to $0.5\ ms$, optimizing the sensor's response characteristics.

(4) Continuous measurement mode and the automatic SET feature are deactivated to align with the sensor's operational requirements.

Data Extraction

The acquisition of magnetic field data from the MMC5983MA magnetometer is conducted through the subsequent operation.

(1) Initially, the 'take magnetic measurement' bit is activated, signaling the sensor to initiate the measurement. This bit automatically resets upon the completion of each measurement cycle.

(2) Subsequently, 7 consecutive bytes are collected. These bytes contain magnetic field data, as outlined in Table 3.4.

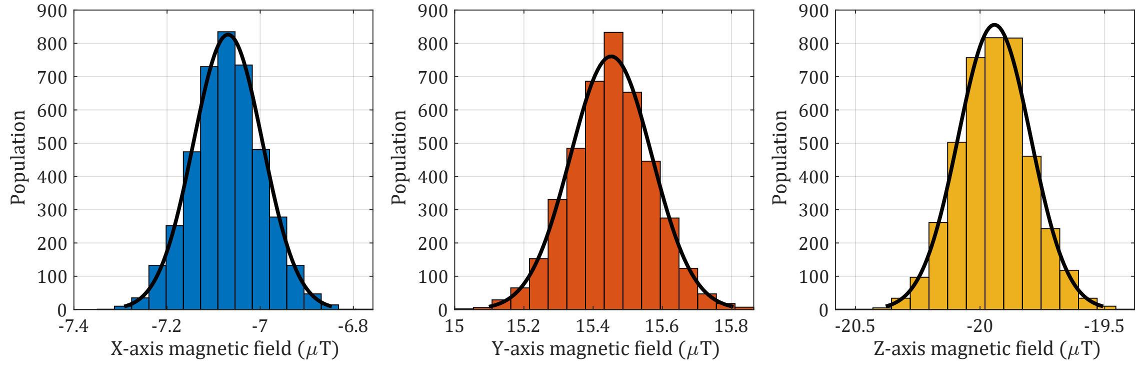
Table 3.4: MMC5983MA's output register detail.

Byte	1	2	3	4	5	6	7			
Description	$X_{OUT}0$	$X_{OUT}1$	$Y_{OUT}0$	$Y_{OUT}1$	$Z_{OUT}0$	$Z_{OUT}1$	$X_{OUT}2$	$Y_{OUT}2$	$Z_{OUT}2$	–

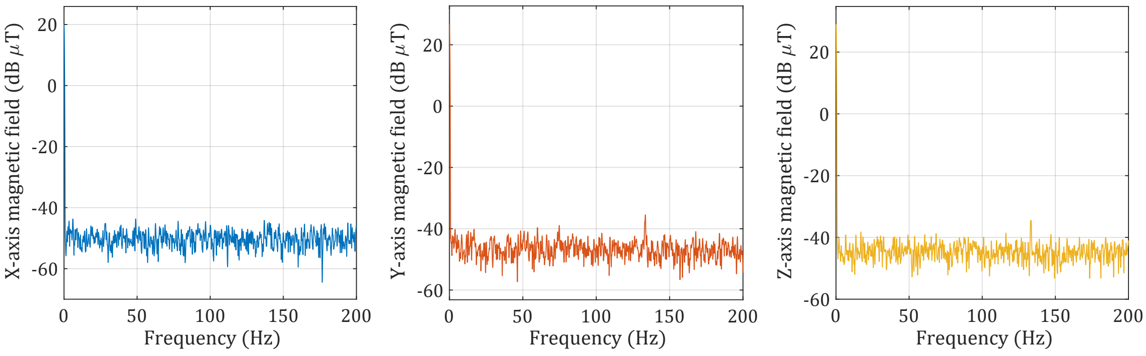
Power Management

To transition the MMC5983MA sensor into the power-down mode, the 'inhibit' bits are set. This change significantly reduces the sensor's power consumption, bringing it down to a minimal current draw of approximately 1 μA.

3.4.3 Output Signal



(a) Histogram and normal distribution fit



(b) Power spectral density

Figure 3.10: Magnetometer's output under ideal environmental conditions.

The output signal of the MMC5983MA magnetometer was evaluated over a 10-second interval under static conditions to confirm its performance and noise level. The results are presented in Figure 3.10. Moreover, the noise parameters of the magnetometer are summarized and compared with the datasheet values in Table 3.5.

Table 3.5: Noise parameters of the Magnetometer compared to datasheet values.

Parameter	**Datasheet Value**	**Measured Value**		
		X-axis	Y-axis	Z-axis
Total RMS noise (μT_{RMS})	0.12	0.09	0.11	0.12

3.5 Analog-to-Digital Converter

Incorporating an ADC into the system design is necessary to handle the data from the accelerometer sensor and the high-rate Z-axis gyroscope. The following list outlines the specifications required that this ADC should fulfill to ensure optimal performance and compatibility with the existing system.

- Given that the accelerometer outputs data across three distinct axes, and the Z-axis gyroscope provides one additional output, the ADC must feature at least 4 single-ended channels to convert the analog data from these sensors.

- To minimize the impact of noise and interference on the ADC's performance, it is crucial to maintain separate analog and digital supply lines. Moreover, the ADC should support a $5\ V$ analog input supply since the sensors output signals in this voltage range, and it should accommodate a $3.3\ V$ digital supply for efficient communication with the microcontroller.

- Based on the following calculations, to ensure the detection of each angular velocity unit from the high-rate Z-axis gyroscope, the ADC should have a minimum resolution of 16 bits.

$$\begin{gathered} Min.\ number\ of\ bits = \\ \log_2 \frac{Output\ voltage\ range}{Resolution} = \log_2 \frac{4.5\ V}{0.1\ mV} = 15.46 \longrightarrow 16\ bits\ required \end{gathered} \tag{31}$$

- Based on the sensor system's sampling rate discussed in section 3.6.2, the ADC's sampling rate should match this rate, which is $400\ Hz$.

Based on the outlined specifications, the MAX1167 from Analog Devices has been selected as the suitable ADC for the system. Here is a brief overview of the MAX1167 characteristics [48]:

- The MAX1167 is a 16-bit Successive-Approximation-Register (SAR) ADC equipped with a four-channel multiplexer. This ADC operates with a solitary $5\ V$ analog supply but also features a distinct digital supply, which permits an interface range of $2.7\ V$ to $5.5\ V$.
- This ADC can function with both external and internal reference voltages. The internal reference voltage, including a buffer, is $4.096\ V$.
- It supports external and internal clock modes, with a maximum sample rate of $200\ kSPS$ (Samples Per Second).
- The ADC includes an automatic shutdown mode, which the microcontroller can trigger. During the shutdown, the analog and digital supply's combined current is minimized to $0.6\ \mu A$.
- Communication between the ADC and the processor is facilitated via the SPI protocol. The maximum clock frequency can be configured to $4.8\ MHz$.

3.5.1 Operation Circuit

The MAX1167 operation circuit has been designed as depicted in Figure 3.11. The following list explains the steps undertaken for this purpose:

- The Analog Supply (AVDD) of the IC is connected to a $5\ V$ power source through two adjacent decoupling capacitors. The larger capacitor, often called the "bulk" or "bank" capacitor, aids in suppressing transient power draws. Conversely, the smaller bypass capacitor minimizes the potential high-frequency noise [49].
- The Digital Supply (DVDD) is linked to a $3.3\ V$ power source to facilitate effective communication with the microcontroller. As with the analog supply, two decoupling capacitors are connected to the DVDD. These capacitors fulfill a similar function, i.e., suppressing transient power draw and reducing high-frequency noise.

- The inputs to the ADC are connected to the sensor outputs via an anti-aliasing filter. The design and implementation of this filter are discussed in detail in section 3.5.3.
- For the reference voltage, internal and external sources have been evaluated during the design phase. The ADC output, consistent with the sensor's specifications, demonstrated less noise when operating with the external reference source. Consequently, an external voltage reference has been incorporated into the system design. The specifics of this reference are further elaborated in chapter 3.5.4.
- Lastly, the Chip Select (CS), Serial Clock (SCLK), Master In Slave Out (MISO), and Master Out Slave IN (MOSI) pins associated with the SPI protocol have been directly linked to the corresponding ports on the microcontroller, enabling efficient data communication between the ADC and the processing unit.

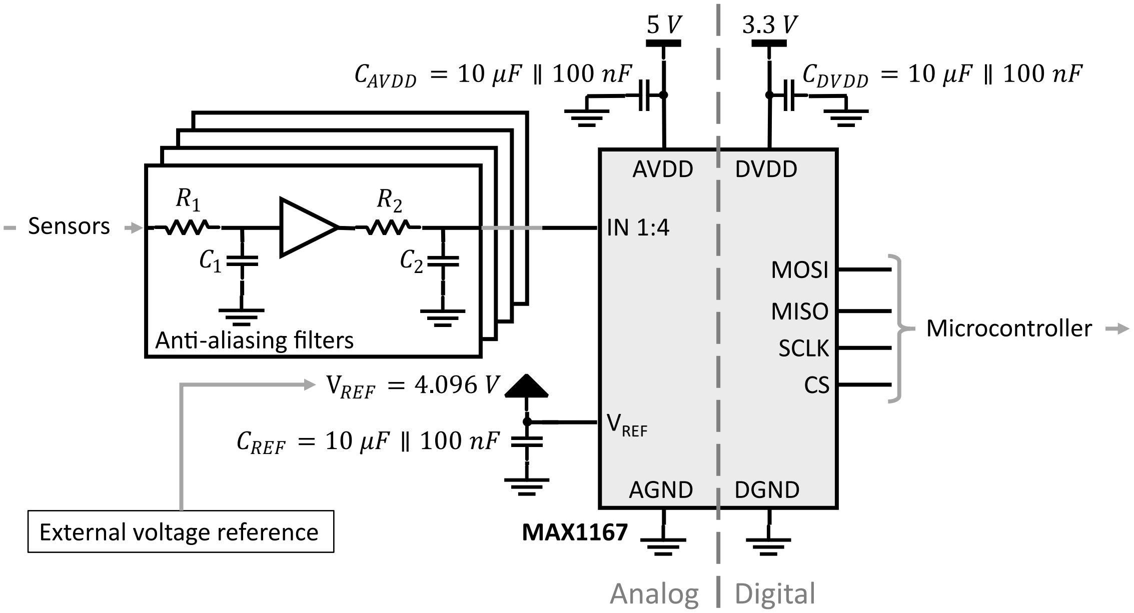


Figure 3.11: The ADC schematic design.

3.5.2 Software Driver

The software driver for the MAX1167 has been developed employing the `nrf_drv_spi` library from the nRF SDK. This library forms an abstraction layer for SPI communication, significantly streamlining the control operations for the ADC.

A command register is transmitted from the microcontroller via the MOSI pin for each sampling instance. This register embodies several key parameters controlling the ADC operation, dictating how the sensor data will be processed. The distinctive configurations housed within the command register include:

- *Clock Mode:* The clock mode has been configured to adopt an external clock, aligning with the system's requirements for precise synchronization facilitated by the SPI protocol.
- *Scan Mode:* The ADC is set to function in a single-channel scan mode, without multiple scans, a configuration compatible with the external clock mode.
- *Power-Down Mode:* The power-down mode is arranged to deactivate the internal reference and buffer perpetually. This configuration significantly curtails the current draw, enhancing the system's energy efficiency.

The 'channel select' bits within the command register are tuned to their corresponding values for each channel reading. Upon implementing this command, the ADC responds by yielding a corresponding 16-bit value onto the MISO pin during the concurrent SCLK period.

3.5.3 Anti-aliasing Filter

Anti-aliasing filters are integral to digital signal processing as they help accurately convert analog signals into digital ones. They work by diminishing the high-frequency components of an analog signal before digitization, a vital step to prevent aliasing—a type of distortion. The Nyquist-Shannon sampling theorem states that a signal should be sampled at least twice its highest frequency (the Nyquist rate) to avoid aliasing, which means that frequencies exceeding the Nyquist rate lead to aliasing, misrepresenting them as lower frequencies in the digital signal. Anti-aliasing filters address this issue by considerably reducing these high frequencies, eliminating aliasing risk, and ensuring a faithful digital conversion of the original analog signal [50].

The implemented design uses a second-order LPF as the anti-aliasing filter. The structure of this filter, constructed in each stage, involves a resistor and a capacitor, whose values determine the cut-off frequency by the relation $\frac{1}{2\pi RC}$ $[Hz]$. A buffer has been incorporated into the design to mitigate potential load effects that could compromise the filter's performance. This inclusion ensures that the output of the filter remains unaffected by the input impedance of the subsequent stage, thus assuring a stable and consistent filtering operation. Furthermore, the second stage adds a non-dominant pole to the overall filter response, enhancing the filter's efficacy in attenuating higher-frequency components. The filter schematic is shown in Figure 3.12.

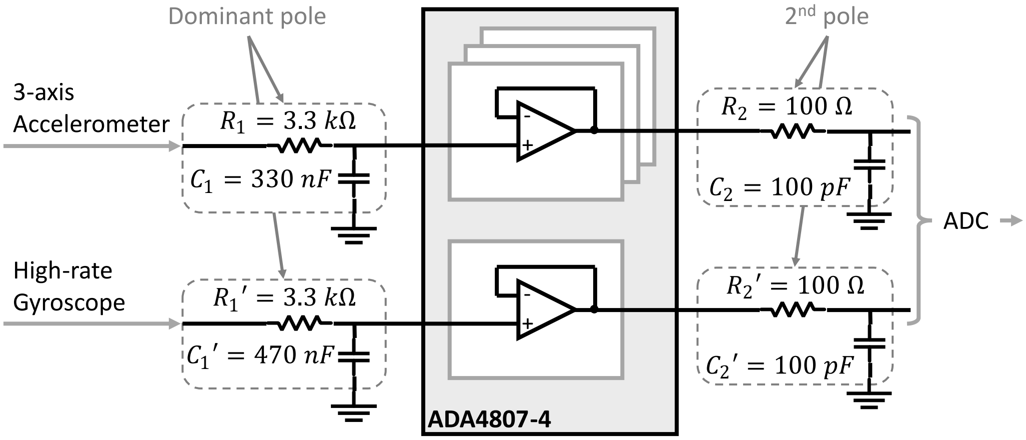


Figure 3.12: The anti-aliasing filter design.

Considering that the ADC's sampling frequency is $400\ Hz$ and the Nyquist rate is thus $200\ Hz$, the poles are found for the anti-aliasing filter design. The dominant pole for the accelerometer is set around $150\ Hz$, while for the high-rate gyroscope, it's at $100\ Hz$. Moreover, a uniform second pole for both sensors is roughly $8\ MHz$.

Buffer

The ADA4807-4 from Analog Devices has been selected for the anti-aliasing filter buffer component. This Operational Amplifier (Op-Amp) is crucial in achieving a 16-bit accuracy for the ADC. The key attributes of this Op-Amp include [51]:

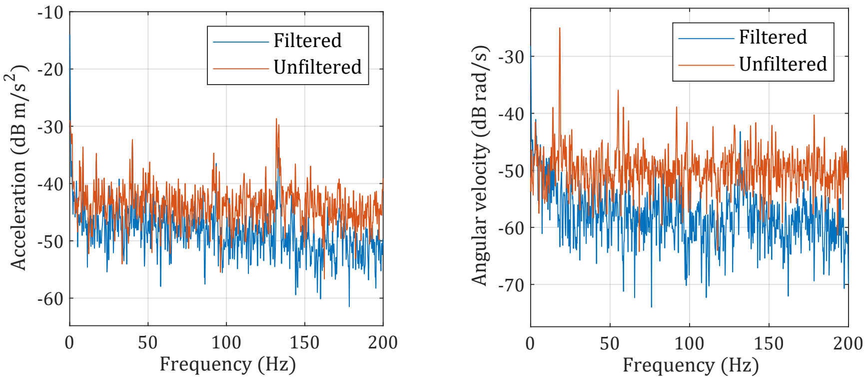
- The ADA4807-4, as a quad operational amplifier, conveniently accommodates the total number of required buffers within a single component.
- The ADC specifies a minimum gain-bandwidth product of $10\ MHz$. The ADA4807-4 significantly surpasses this requirement by having a gain-bandwidth product of $170\ MHz$. This

extensive bandwidth ensures high-precision analog-to-digital conversion, adequately accommodating the application's demands.

- In line with the gain-bandwidth requirement, the buffer is expected to exhibit a minimum slew rate of 2 $V/\mu s$ to ensure the completion of the required output-voltage change before the end of the acquisition time. The ADA4807-4 exceeds this demand, with a slew rate of 160 $V/\mu s$, thus demonstrating its suitability for this high-precision application.
- The ADA4807-4 is a rail-to-rail Op-Amp, meaning its saturated output voltage swing is no more than $0.05\ V$ away from the power supply rails, i.e., VDD and GND. This property allows for a broader range of output voltages, thereby enhancing the reliability of sensor data acquisition.

Filter Effect on Sensors' Output

The impact of the anti-aliasing filter on the output signals from the accelerometer and the high-rate gyroscope is visually depicted in Figure 3.13, where it can be seen that the high-frequency noise is reduced post-filtering.



(a) 3-axis accelerometer (b) High-rate Z-axis gyroscope

Figure 3.13: Anti-aliasing impact on the analog-output sensors.

3.5.4 Voltage Reference

The ADR4540 from Analog Devices has been employed as the voltage reference. This device is characterized by a fixed output voltage of $4.096\ V$ and an efficient output voltage error of 0.02%. Its output voltage noise is $2.7\ \mu V_{pp}$, which is sufficiently low to satisfy the required noise performance of the ADC, given that the latter's input noise is $40\ \mu V_{RMS}$. Thus, the ADR4540 ensures that the ADC's performance is not compromised by the voltage reference's noise contribution [52].

The output impedance of the ADR4540 can rise to $5\ \Omega$ near DC frequencies, which could potentially affect the performance of the ADC. To mitigate this issue, an additional buffer is incorporated into the design. The buffer utilizes the ADA4807-1, a single operational amplifier variant of the ADA4807-4. By deploying this buffer, the output impedance is further reduced to less than $0.1\ \Omega$, significantly improving the stability and precision of the system.

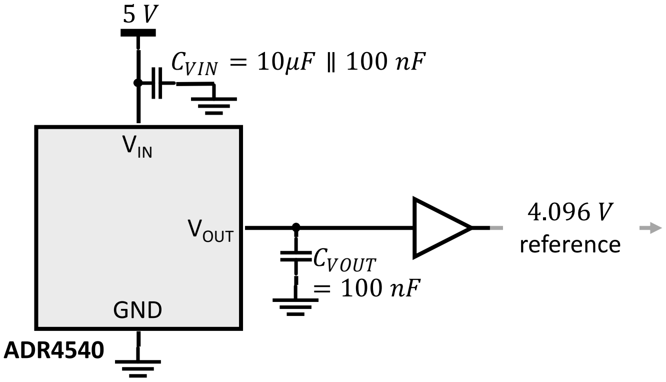


Figure 3.14: The voltage reference schematic design.

Figure 3.14 presents the schematic diagram of the voltage reference, devised in compliance with the specifications detailed in its datasheet.

3.6 Microcontroller

For the sensor-integrated hockey puck design, the selected microcontroller must satisfy the following criteria:

- *Integrated Wireless Transceiver:* To ensure design simplicity and avoid compatibility challenges, a microcontroller with an integrated wireless transceiver is desired.
- *BLE Support:* Given its low energy consumption and robust data transfer capabilities, the microcontroller should support the BLE protocol. Specifically, the ability to achieve a data rate of 2 $Mbps$ is pivotal for maximum data throughput.

- *Peripheral Support:* The microcontroller should support vital peripherals, like General Purpose Input-Outputs (GPIOs), Two-wire Interface (TWI), and Serial Peripheral Interface (SPI), to guarantee seamless interfacing with sensors and the ADC. Other key peripherals, including timers and the Universal Asynchronous Receiver-Transmitter (UART), facilitate operational and debugging processes. Notably, an integrated ADC in the microcontroller is unnecessary.
- *In-built Antenna:* Preferring a microcontroller with an integrated antenna simplifies the overall design.

PAN1781

The PAN1781 from Panasonic Industry was selected for this project due to its alignment with the sensor-integrated hockey puck criteria. As shown in Figure 3.15, this module integrates the nRF52820 from Nordic Semiconductor as its primary processor. Accompanied by a dedicated power management system, crystal oscillators, and an integrated antenna, this choice significantly eases the design, eliminating intricate antenna design considerations [53].

The PAN1781 module provides full access to the power and I/O pins of the nRF52820. Key features of the nRF5280 comprise:

- An Arm® Cortex®-M4, operational at 64 MHz. It also endorses the Serial Wire Debug (SWD) protocol, optimizing the debugging and programming processes.
- A suite of analog and digital peripherals – including GPIOs, UART, TWI, SPI, timers, and an analog comparator, all required for the sensor system.
- Comprehensive support for the complete suite of Bluetooth® Low Energy (BLE) features. The high-throughput 2-Mbps feature aligns with the project's requirements.
- Power management features that ensure energy efficiency, highlighting features like low current consumption and peripheral power management.

Such attributes underscore the module's aptness in collecting and wirelessly transmitting sensor data [54].

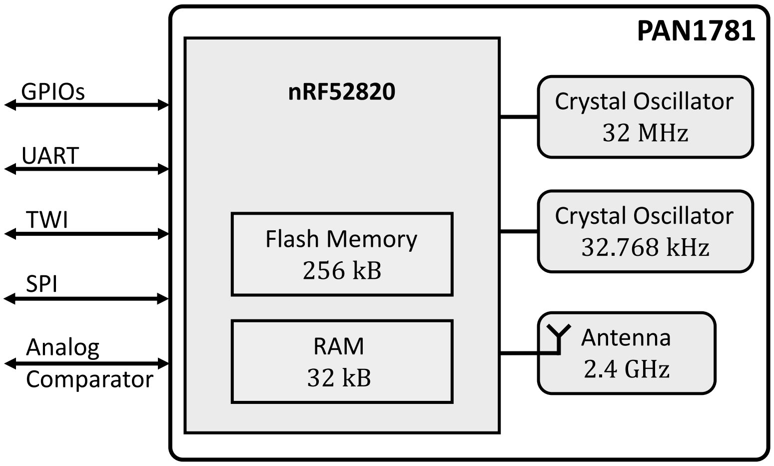


Figure 3.15: PAN1781 by Panasonic Industry [53].

3.6.1 Operation Circuit

As depicted in Figure 3.16, the implementation of the PAN1781 microcontroller in the sensor system design demanded the following considerations:

- The power supply is regulated at 3.3 V for standard operation. High-performance RF capacitors decouple the DC supply voltage at the VDD pins, providing the necessary stabilization.

- The I/O pins on the nRF52820 are classified into standard drive and high drive. The standard drive pins are located near the internal radio routes of the chip, potentially causing interference with the BLE service if subjected to high current or high-frequency operations. Therefore, high-frequency functions such as UART, TWI, and SPI are allocated to high drive pins, as outlined in the datasheet [54], leaving standard drive pins for other tasks.

- The sensor system enters shutdown mode to halt all processes and conserve battery life. A magnetic reed switch is employed to transition the sensor from shutdown to active mode. Given the sensor's encapsulated position within the puck, the magnetic switch was selected for its practicality.

- An LED indicator is included in the design to provide information on the sensor's operational status.

- A physical switch is incorporated to trigger the RESET pin, allowing for forced reset when necessary.

- The sensor is programmed and debugged using the SWD and UART communication protocols.

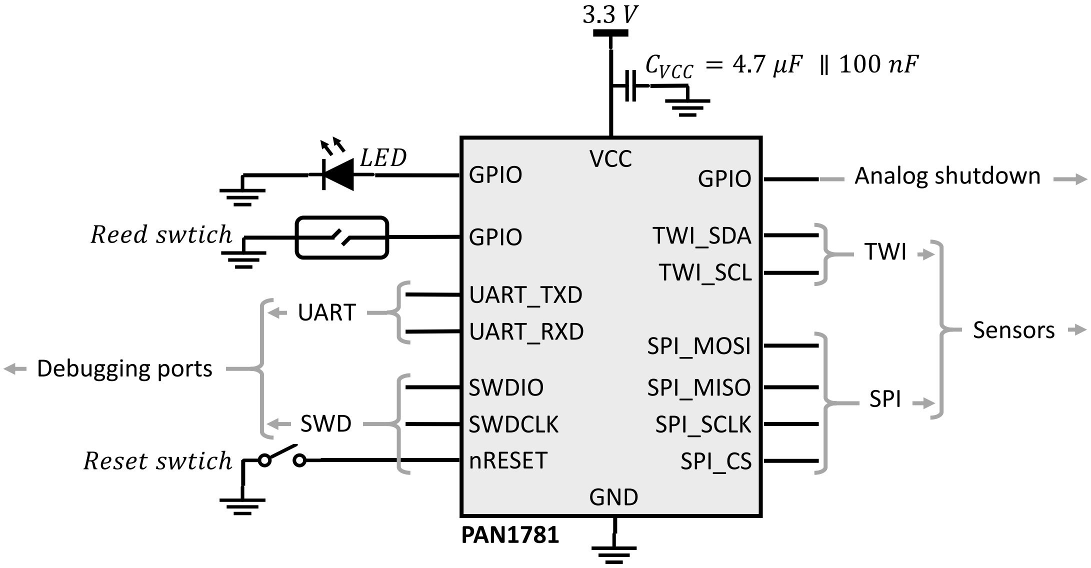


Figure 3.16: The Microcontroller schematic design.

3.6.2 Software Development

The software development for the nRF52820 is a critical aspect of the sensor system design. The software's primary function is to arrange the sensor data collection, carry out the required processing, and manage the wireless transmission. This development process is facilitated by the nRF5 Software Development Kit (SDK) version 17.1.0 provided by Nordic Semiconductor. The SDK includes a comprehensive collection of software libraries and Hardware Abstraction Layers (HALs), simplifying interfacing with the device's onboard peripherals and writing the application logic. The software can be developed in C/C++ language. Furthermore, Segger Embedded Studio, a powerful cross-platform Integrated Development Environment (IDE), can write, compile, and debug the software.

Bluetooth® Low Energy

The primary requirements for wireless communication in the sensor-integrated hockey puck system involve power efficiency, robust protocols, and short-range communication. The technology that best fits these prerequisites is Bluetooth® Low Energy (BLE), a wireless communication standard designed specifically for power efficiency and short-range device communication.

BLE operates within the 2.4 GHz Industrial, Scientific, and Medical (ISM) band and boasts an established set of protocols for managing connections, data formatting, secure data exchange, and more [55]. These features make it an excellent choice for the wireless, real-time transmission of sensor data in the proposed sensor system design.

The development of the BLE functionality within the sensor system was facilitated by leveraging the resources provided by Nordic Semiconductor. Specifically, the example code provided in the nRF5 SDK, `ble_app_uart` for the `pca10100e` board, was used as a foundation for managing the BLE service.

To further modify the BLE service to the requirements of the hockey puck system, a comprehensive approach was taken to develop a custom BLE service [56]. This approach provides systematic guidance on defining and establishing a custom BLE service, utilizing SoftDevices, and performing read, write, and notification operations. Appendix C documents the specific implementation details.

SoftDevices

SoftDevices are pre-compiled, pre-linked binary files provided by Nordic Semiconductor. These files encompass the entire Bluetooth Low Energy (BLE) protocol stack and manage the radio hardware, freeing the application developer from the intricacies of low-level radio operations. The SoftDevice is programmed into a specific flash memory area and provides Application Programming Interfaces (APIs) for interaction with the application layer.

The S140 SoftDevice, a feature-rich protocol stack designed for the nRF52 series, has been implemented in this project. Supporting multiple concurrent Central and Peripheral roles with encrypted communication, this SoftDevice is perfectly aligned with the application's requirements. All necessary functions for establishing device roles, managing connections, data transmission, and administering security protocols are encapsulated within the SoftDevice API. The utilization of SoftDevices streamlines the software development process by addressing low-level BLE operations, thereby allowing the focus to be placed on higher-level tasks at the application layer [57].

Sampling Frequency

The selected BLE service allows a maximum payload of 244 bytes in a single packet. For efficiency and maximizing data throughput, transmitting sensor data using this total payload capacity is desirable.

In the initial testing phase, it was found that for maintaining stable communication with the host system within the required range, the packet transmission frequency should not exceed 50 Hz. A packet transmission frequency of 40 Hz was chosen to incorporate a safety margin.

The packet transmission frequency choice determined the sensor sampling rate. The accelerometer, gyroscopes, and magnetometer data account for 21 bytes. To fill the payload of the BLE data packet efficiently, 10 samples of the sensor data (totaling 210 bytes) are collected before transmission. This data collection strategy corresponds to a sensor sampling rate of 400 Hz, providing a balance between data throughput and wireless communication stability.

3.7 Power Supply

The power supply unit of the sensor system consists of a battery and two voltage regulators. These regulators are designed to provide two separate outputs, specifically 3.3 V and 5 V, thus accommodating the voltage requirements of the sensor system components.

Battery

Given the need for a compact, lightweight, and rechargeable power supply, a single-cell Lithium Polymer (LiPo) battery with a nominal voltage of 3.7 V was selected. Considering the sensor's current draw of 37 mA during the measurement mode and the need to sustain an 8-hour measurement period with a safety margin, a battery with a capacity of 400 mAh was found suitable. LiPo batteries, characterized by their high energy density, offer the benefits of minimal size and weight while also being rechargeable, thereby fitting the system requirements effectively.

3.7.1 Voltage Regulators

The power management of the sensor system necessitates the deployment of two voltage regulators to supply distinct voltage levels to various components. Firstly, a low-dropout (LDO) regulator is utilized to provide a stable 3.3 V supply, primarily dedicated to the "digital" parts of the system. These components encompass the microcontroller, the 3-axis gyroscope, the magnetometer, and the digital subcircuit of the ADC. In contrast, the "analog" components of the system require a 5 V supply. This category includes the accelerometer, the high-rate gyroscope, the anti-aliasing filter, the external voltage reference, and the analog subcircuit of the ADC. A step-up switching regulator is incorporated into the design to cater to this requirement.

3.3 V Supply

The digital components of the system require a stable 3.3 V supply, ideally provided by a low dropout voltage regulator capable of handling the necessary power requirements. Among the available options, the LTC1844ES5-3.3 from Analog Devices has been selected due to its suitable maximum output current of 150 mA and a notably low output voltage noise of 30 μV_{RMS} within the 100 Hz to 100 kHz frequency range [58].

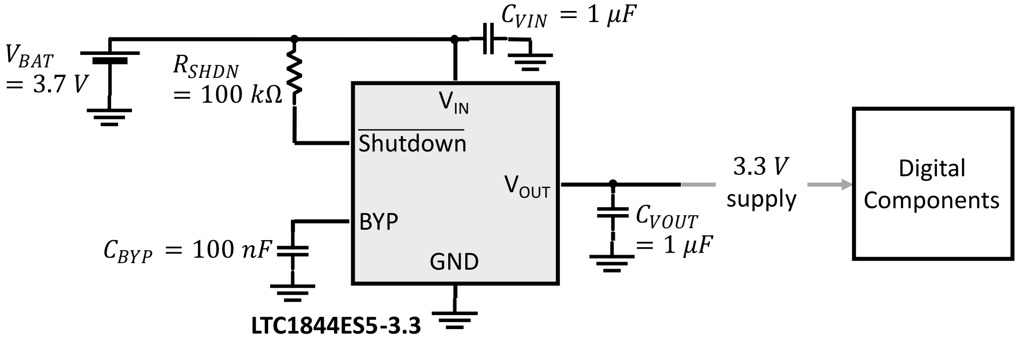


Figure 3.17: The 3.3 V voltage regulator schematic design.

For the LTC1844ES5-3.3 regulator, a circuitry configuration is implemented to optimize stability and noise performance. A bypass capacitor is coupled from the BYP pin to the ground to create a low-frequency noise pole, while an output capacitor is added to improve the transient response to load current changes. Additionally, considering the output impedance of a battery increases with frequency, an input bypass capacitor is introduced in battery-powered circuits. As illustrated in Figure 3.17, this setup mitigates impedance fluctuations, ensuring a reliable 3.3 V supply for the system's digital components [58].

5 V Supply

The power supply for the analog components of the sensor system necessitates a fixed 5 V voltage. The LTC3200ES6-5 from Analog Devices provides this. This integrated circuit is especially notable for using a switched capacitor charge pump rather than inductors for boost operation. This characteristic is significant since inductors could potentially interfere with the operation of the magnetometer. The output of the LTC3200ES6-5 is consistently fixed at 5 V (with a margin of $\pm4\%$) and a steady output current of 100 mA. It operates with a switching frequency of 2 MHz, which results in an output ripple of approximately 1 mA within this circuit. In addition, the LTC3200ES6-5 features a shutdown logic pin, which enables the microcontroller to switch off all the analog circuitry during the deep sleep mode of the hockey puck sensor, thereby conserving power [59].

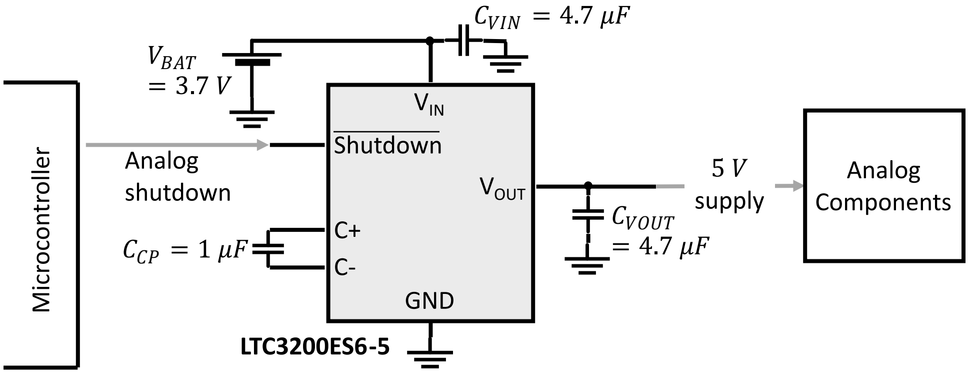


Figure 3.18: The 5 V voltage regulator schematic design.

Specific capacitor components are required within the circuit to ensure optimal performance of the LTC3200ES6-5. Specifically, a flying capacitor should be connected between the C+ and C- pins. This capacitor modulates the strength of the charge pump, hence playing a pivotal role in the operation of the voltage regulator. Additionally, the output capacitor influences the amount of output ripple corresponding to a specific load current, whereas the input capacitor controls the ripple present at the input pin. To maximize the effectiveness of these capacitors, it is recommended to utilize those with low Equivalent Series Resistance (ESR) [59]. The schematic diagram of this setup is presented in Figure 3.18.

3.7.2 Power Management

Efficient power management is crucial in battery-operated devices to maximize battery life and optimize system performance. A magnetic reed switch is incorporated into the sensor system to manage power. This switch is activated when a magnet approaches it, waking the sensor system.

Under standard operation, if the system remains inactive for 60 seconds, it automatically transitions into shutdown mode to preserve battery life. This state turns off all the analog circuitry, sensors, core processor activities, and BLE functionality. When a magnet is brought close to the reed switch, the system resets and powers up, restarting operations. This power management approach significantly extends battery life and enhances the overall operational efficiency of the sensor system.

Table 3.6 below summarizes the current drawn during each operational mode:

Table 3.6: Current drawn during different operational modes

Mode	Current from battery
Measuring	37 mA
Advertising	29 mA
Shutdown	64 μA

Drawing upon the measurements documented in Table 3.6, the energy management efficiency of the sensor system becomes evident. When the system is in shutdown mode, the battery life extends significantly, allowing the sensor system to remain operational in this mode for more than 200 days on a single charge. When the system is in measuring mode, the battery life supports around 8 hours of continuous operation. Importantly, recharging the battery only takes approximately 45 minutes.

3.8 Prototype Implementation

A preliminary prototype was developed to validate the sensor system's viability and functionality before integrating it within a hockey puck. Assembled using evaluation boards for the microcontroller, sensors, ADC, and voltage regulators, all arranged on breadboards, this prototype allowed for a comprehensive functionality test. This early-stage setup proved instrumental in identifying and rectifying potential issues before finalizing the design. The components and their connections within the prototype are represented in Figure 3.19.

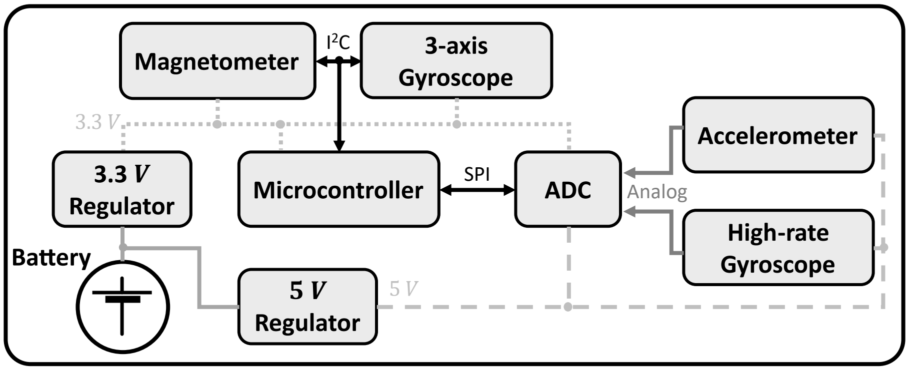


Figure 3.19: Block diagram of the preliminary prototype for the sensor system.

The breadboard was attached to a circular piece of cardboard to simulate the physical conditions associated with actual puck rotations. Figure 3.20 showcases the prototype, illustrating the components' arrangement on the breadboard and their attachment to the cardboard circle.

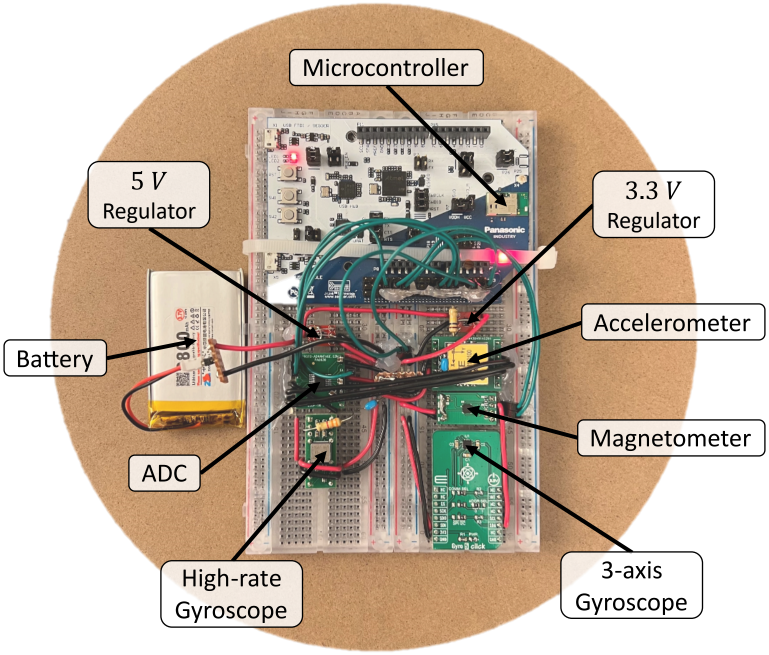


Figure 3.20: Prototype of the sensor system.

The prototype deviates from the final design in some aspects. It has a lower sampling frequency of 100 Hz and utilizes a larger capacity battery of 2800 mAh. Additionally, it lacks critical components such as the anti-aliasing filters, the external voltage reference, and the magnetic reed switch. Despite these differences, the prototype was a crucial step in testing and refining the design before its finalization.

3.8.1 Prototype Tests and Results

The prototype was trialed at the Ice Hockey Research Group (IHRG) Lab of McGill University, leveraging its synthetic ice environment for realistic simulations. The sensor's velocity estimation accuracy was compared against the Vicon motion capture system. This comparison evaluated the implemented velocity estimation methods. The following sections provide detailed discussions of these tests and their results.

Push to Stop

This test involves manually pushing and spinning the puck until it decelerates and stops, challenging the sensor system's ability to record combined translation and rotation dynamics. Figure 3.21 shows the IMU results.

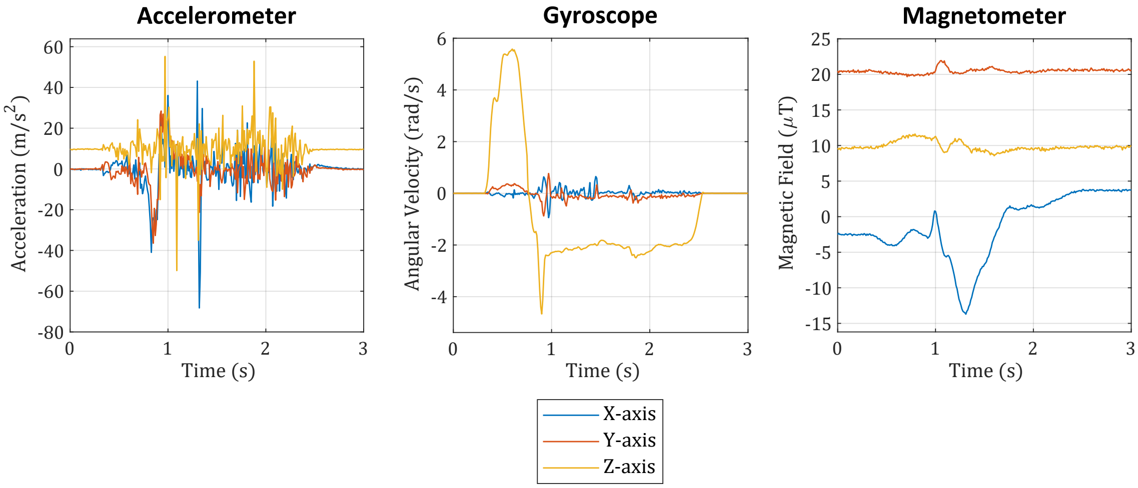


Figure 3.21: Prototype IMU readings during the "Push to Stop" test.

Figure 3.22 compares the peak velocity estimations from various methods. The accelerometer-based method has a significant estimation error of approximately -55.6%. The accelerometer-and-gyroscope-based method performs better, with a velocity estimation error margin of about -13.53%. The MARG-based method is the most accurate, with an error margin of -5.29%.

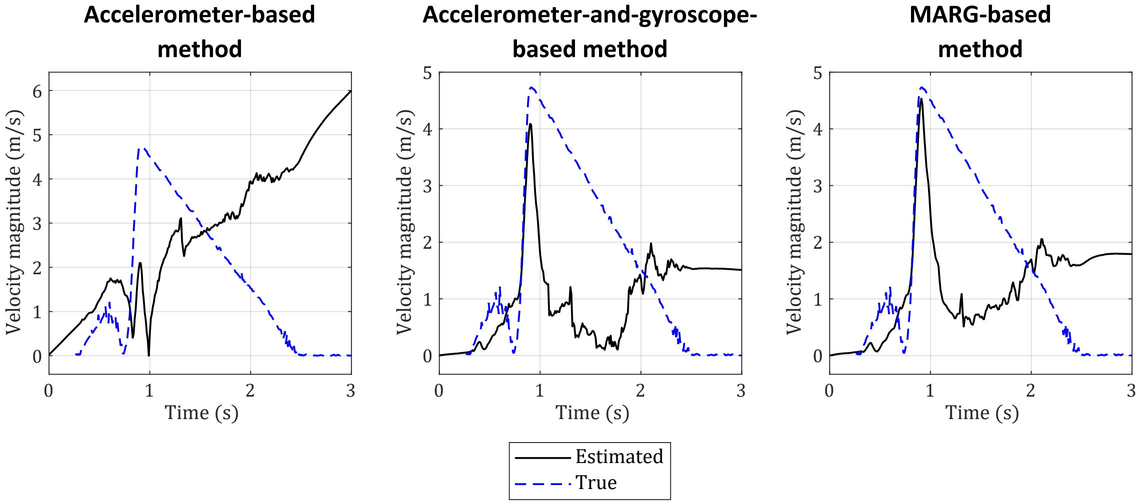


Figure 3.22: Velocity estimations of the puck prototype during the “Push to Stop” test using different methods.

Table 3.7 summarises the performance of the three methods over 10 test iterations, highlighting the MARG-based method’s superior accuracy in estimating the puck’s movements.

Table 3.7: Error estimates of the peak velocity from the three methods over 10 iterations of the “Push to Stop” test on the puck prototype.

Trial	**True Velocity** (m/s)	**Accelerometer-based Method** %	**Accelerometer-and-gyroscope-based Method** %	**MARG-based Method** %
1	4.58	-46.51	-8.08	2.84
2	4.70	-22.55	7.89	5.32
3	4.07	–	8.60	6.88
4	3.89	–	-80.98	-79.75
5	4.17	-29.98	-14.31	-12.06
6	3.67	-32.15	9.26	-4.63
7	3.73	-71.58	-1.34	0.27
8	4.73	-55.60	-13.53	-5.29
9	3.63	-71.90	22.95	17.36
10	4.05	-48.39	-7.94	-3.73
Average Error		-47.33	-7.75	-7.28

Wrist Impact

This test scenario involves spinning a puck by hand towards a stationary object, simulating situations where it hits items like hockey sticks, goal bars, or rink boards. Upon impact, the puck's direction changes before eventually stopping. The sensor system's ability to accurately track and interpret these movements is evaluated. Figure 3.23 exhibits the IMU readings obtained during this test.

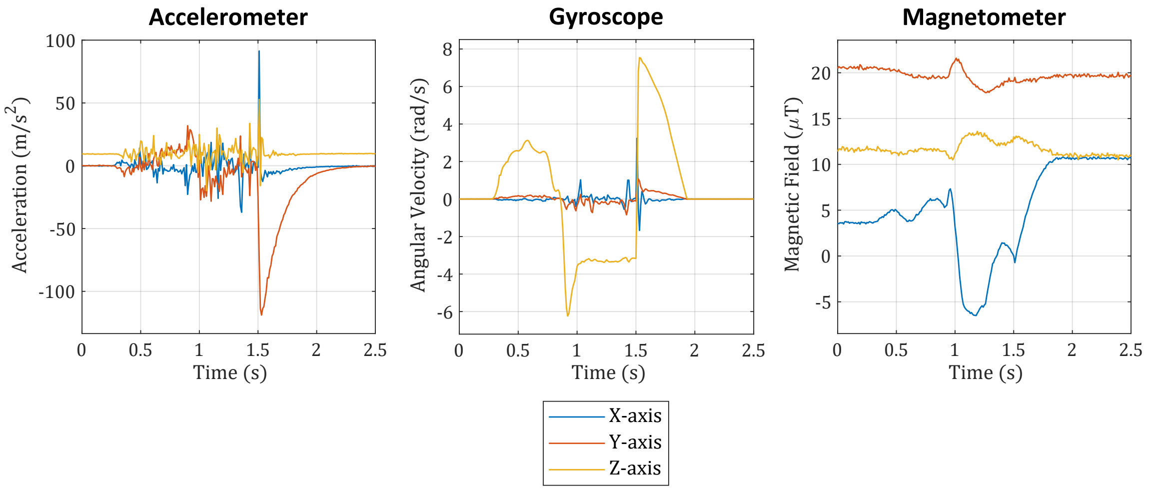


Figure 3.23: Prototype IMU readings during the "Wrist Impact" test.

As shown in Figure 3.24, results are similar to the "Push to Stop" test. The accelerometer-and-gyroscope method estimates velocity with a -14.74% error margin, while the MARG-based method performs better with a -9.47% error margin. The accelerometer-based method, however, fails to provide reasonable peak velocity estimations.

A sudden surge is observed in the estimated velocity upon puck impact, which continues even after the impact. This surge suggests the presence of unaccounted forces or parameters in the current models, prompting the need for further investigation to refine the estimation algorithms. Table 3.8 compares the three methods' performance over 10 iterations of the "Wrist Impact" test. The MARG-based method consistently outperforms the others in estimating the puck's movements.

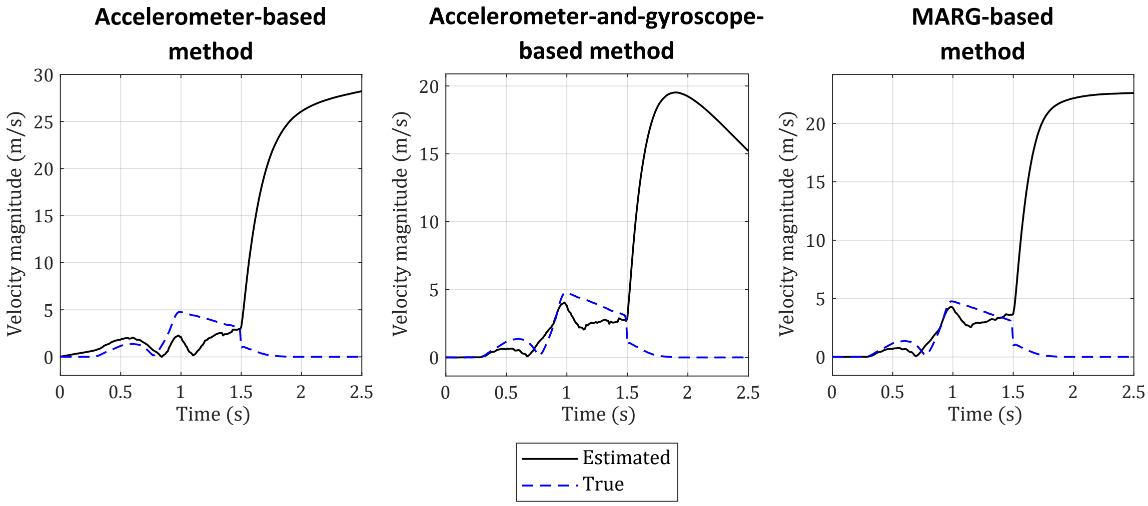


Figure 3.24: Velocity estimations of the puck prototype during the "Wrist Impact" test using different methods.

Table 3.8: Error estimates of the peak velocity from the three methods over 10 iterations of the "Wrist Impact" test on the puck prototype.

Trial	True Velocity (m/s)	Accelerometer-based Method %	Accelerometer-and-gyroscope-based Method %	MARG-based Method %
1	4.75	-52.21	-14.74	-9.47
2	4.43	-41.76	-20.99	-16.03
3	4.09	-72.13	-26.16	-17.85
4	4.05	-45.93	-37.53	-31.85
5	4.46	-74.22	-47.31	-38.57
6	4.22	-49.53	-40.76	-32.70
7	3.47	-63.40	-57.06	-48.13
8	3.65	-62.19	-59.45	-57.26
9	3.91	-64.45	-49.10	-42.20
10	4.25	-55.29	-43.06	-35.29
Average Error		-58.11	-39.61	-32.93

3.9 Final Sensor System Implementation

In this phase of the sensor system development, the transition from the prototype to the final sensor system necessitated the design of a Printed Circuit Board (PCB) that could seamlessly integrate into the hockey puck. To accomplish this, the PCB was designed to accommodate all necessary components within the confined dimensions of the puck (as outlined in section 2.7). The entire design process was carried out using Altium Designer, a comprehensive tool for electronic design automation. The process comprised two significant steps: schematic design, where the electronic components and their interconnections were defined, and PCB design, where the physical layout of the elements and traces on the board was determined.

3.9.1 Schematic Design

The schematic design phase is a critical step that lays the foundation for the subsequent PCB layout design. This phase acts as a blueprint for the physical arrangement of parts on the PCB, thereby representing the overall performance and functionality of the system. Figure 3.25 presents the block diagram of the complete sensor system, showing the components and their connections.

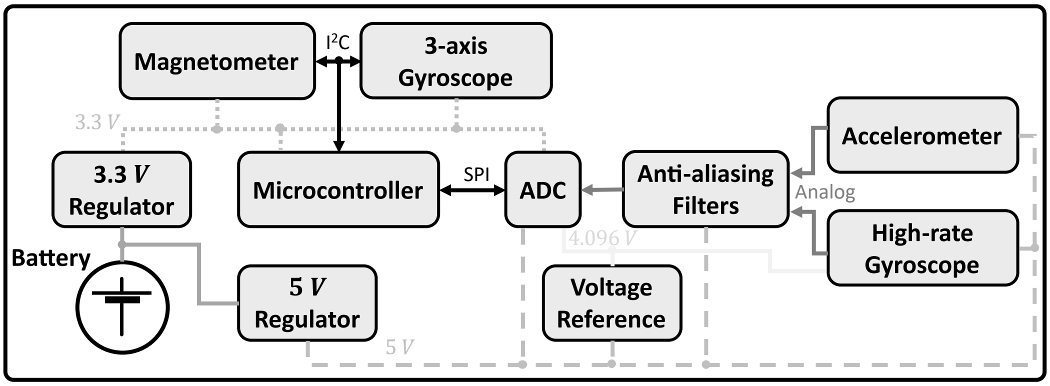


Figure 3.25: Block diagram of the complete sensor system.

In addition to defining the connections between components, it is vital to establish an effective grounding scheme. As discussed in section 3.7.1, the sensor system comprises both “analog” and digital” components. These require separate ground planes to prevent digital noise from interfering with sensitive analog signals. These different ground planes are connected through a ferrite bead. This device acts as a high-pass filter and minimizes potential signal interference between the analog and digital sections of the board, thus enhancing the overall system performance. The schematic design is documented and found in Appendix A.

3.9.2 PCB Design

The design of the PCB incorporates crucial considerations, most notably the board’s shape and the sensors’ placement. The board has been designed to be circular to align with the internal geometry of the hockey puck. This design decision has two primary benefits: it ensures the board fits snugly within the puck and enhances its resilience against the forceful impacts commonly encountered during hockey shots. The absence of edges in a circular design contributes to increased durability.

The placement of the sensors on the PCB is vitally important because the dynamics of the hockey puck can significantly influence the IMU readings at various positions. Appropriate sensor placement guarantees accurate capture of puck motion and provides the necessary information for successful velocity estimation. The following list outlines the strategic considerations made for the placement of critical sensors and other crucial components:

- To minimize the undesired effects of centripetal, centrifugal, and Coriolis accelerations, the accelerometer is strategically positioned at the center of the board.

- The 3-axis gyroscope is also positioned at the center of the board, co-localized with the puck's center of mass.[1] This placement is crucial for capturing accurate angular velocities. The sensitivity of gyroscopes, especially more affordable ones, to both gravitational force and vibration due to minor asymmetries can introduce reading errors. By mounting the gyroscope in the center, it experiences a constant $1\ g$ force, reducing such errors. While these errors might seem minor, even small inaccuracies can be consequential in high-precision applications. Thus, the 3-axis gyroscope is placed at the puck's center to maximize accuracy [60].

- The placement of the magnetometer requires additional care due to its sensitivity to varying magnetic fields. Specific components, such as electrical traces carrying current or electronic components like capacitors, can generate magnetic fields that distort the magnetometer's readings. Following the manufacturer's guidelines, the magnetometer is positioned safely from such elements to avoid potential interference.

- The microcontroller module, specifically the Bluetooth Low Energy (BLE) antenna, is strategically placed to avoid any copper beneath it. Copper can absorb antenna power and alter its characteristics, potentially interfering with BLE communications. The module's placement minimizes these effects, ensuring efficient wireless communication.

- The reed switch, a magnetically operated device, is positioned safely from the magnetometer. This design decision mitigates the potential soft and hard iron effects (refer to section 2.1.3)

---

[1] The 3-axis gyroscope is placed on the top side of the board, while the accelerometer is situated on the bottom side to maintain the central position for both.

that might distort the magnetometer readings, ensuring accurate capture of magnetic field data.

The precision of analog sensors is crucial in the PCB design. Therefore, careful grounding is implemented to prevent digital noise from interfering with the analog signal. Accordingly, the following measures have been undertaken:

- The analog components have been positioned on the bottom layer of the PCB, while the digital parts are on the top layer. This spatial separation reduces Electro-Magnetic Interference (EMI) between the digital and analog circuits, thereby preserving the integrity of the analog signals.
- The sensitive components are intentionally placed apart to avoid potential interference. For example, the microcontroller is positioned away from analog components, and no digital sensors or devices are situated over the anti-aliasing filter or voltage reference. This layout helps preserve the analog signal integrity by reducing capacitive or inductive coupling from digital components, enhancing the accuracy of sensor readings.
- Dual ground planes have been implemented: one for analog and one for digital components. As the ADC's datasheet recommended, they are linked using a ferrite bead near the ADC pins. This design ensures that digital return currents do not interfere with the analog ground, reducing error voltages and providing more accurate signal conversion.
- Via stitching is implemented to lower return path impedance for analog and digital currents. This method reduces EMI, improving signal integrity and overall system performance.
- Following the recommendations in the ADC's datasheet, the board layout was carefully designed to keep digital and analog signal lines separate. This arrangement helps prevent cross-interference, preserving signal integrity and improving system accuracy.

These design choices reflect the best practices for grounding in mixed-signal systems, as discussed in a comprehensive guide on grounding techniques for mixed-signal devices [61].

The complete design of the PCB can be found in Appendix B. This detailed design showcases how practical constraints and technical principles have been balanced to achieve optimal performance for the proposed hockey puck IMU system.

Assembly

The assembly of the PCB was performed using a solder paste and a pick-and-place machine. The soldering process adhered to the temperature specifications outlined in the datasheets of all sensitive components. It is worth noting that leadless parts, including the microcontroller, high-rate gyroscope, and magnetometer, required an additional verification step due to their more complex soldering requirements. An X-ray scan was conducted after the assembly to verify the proper soldering of these components. Figure 3.26 presents the X-ray scan results.

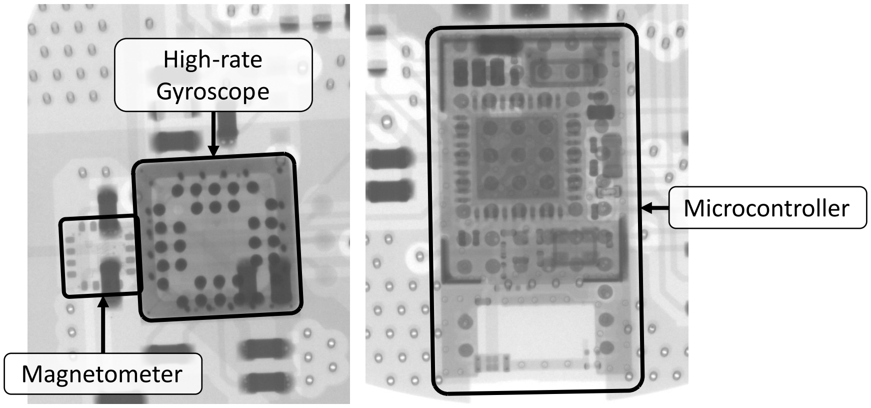


Figure 3.26: X-Ray results showing the soldering of the leadless components.

In the X-ray scans, the quality of soldering can be evaluated based on several factors, including [62]:

- *Solder Fill:* One of the critical inspection parameters in an X-ray scan is solder fill. Proper soldering should show a complete filling in the pad area underneath the component, ensuring a solid electrical and mechanical connection.

- *Voiding:* This refers to the presence of voids or empty spaces within the solder, which can impair the quality of the solder joint. A low level of voiding is typically acceptable, but excessive voiding can lead to issues such as poor thermal transfer, which can affect component performance.

- *Alignment:* The alignment of the component over the pads is inspected. Misaligned components can indicate a problem in the pick-and-place process and may lead to functional issues.
- *Bridging:* This involves looking for unintentional solder connections between adjacent pads or pins, known as bridging. Solder bridges can cause short circuits, leading to malfunctioning of the circuit.
- *Solder Balling:* This involves looking for tiny balls of solder that may have formed during the soldering process. Solder balls can cause short circuits if they break loose and move to other parts of the PCB.

3.9.3 Puck Preparation

To preserve the puck's original dynamics, a method for modifying the puck to integrate the sensor system was designed using SolidWorks, a 3-D CAD software. The following steps ensured that the board would be centrally positioned within the puck:

- *Puck Hollowing:* Precision machinery was used to carefully hollow out the puck, creating a cavity perfectly sized to house the assembled PCB.
- *Spacer Installation:* Spacers were inserted to stabilize the board and secure its position within the puck. This strategic alignment minimizes potential shifts in the puck's dynamics during operation.
- *Battery Holder Integration:* A custom battery holder was engineered to sit above the secured PCB. The holder was designed to contain the battery while allowing for easy replacement.
- *Cap Deployment:* A specially designed cap was affixed to the puck to enclose the sensor system securely. Apart from providing physical protection for the system, this cap offers some level of waterproofing to shield the electronic components from ice and moisture.

The design of the sensor system integration within the hockey puck is illustrated in Figure 3.27. This design successfully embedded the sensor system into the puck without altering its inherent dynamics and ensured that the electronics were protected from environmental conditions.

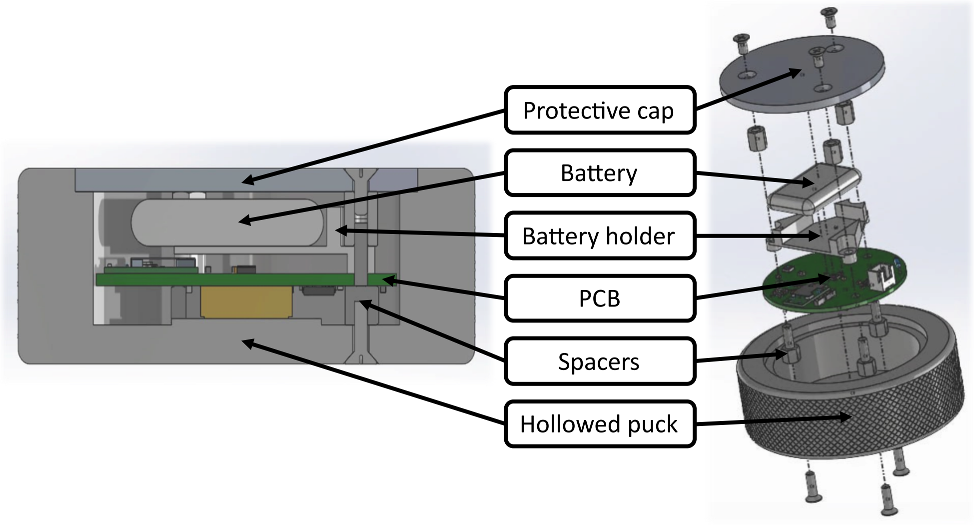


Figure 3.27: The design of the sensor system integration into the hockey puck in the SolidWorks environment.

Both the custom battery holder and the protective cap were created using 3-D printing technology, which allowed for the accurate realization of the design specifications derived from the SolidWorks environment.

(a) Battery holder

(b) Protective cap

Figure 3.28: The 3-D printed parts for the sensor system integration into the hockey puck.

The final step involved fitting the sensor system and the battery into the puck. Figure 3.29 displays the final arrangement with the sensor system and battery securely installed within the puck.

In conclusion, integrating the sensor system into the hockey puck was accomplished without

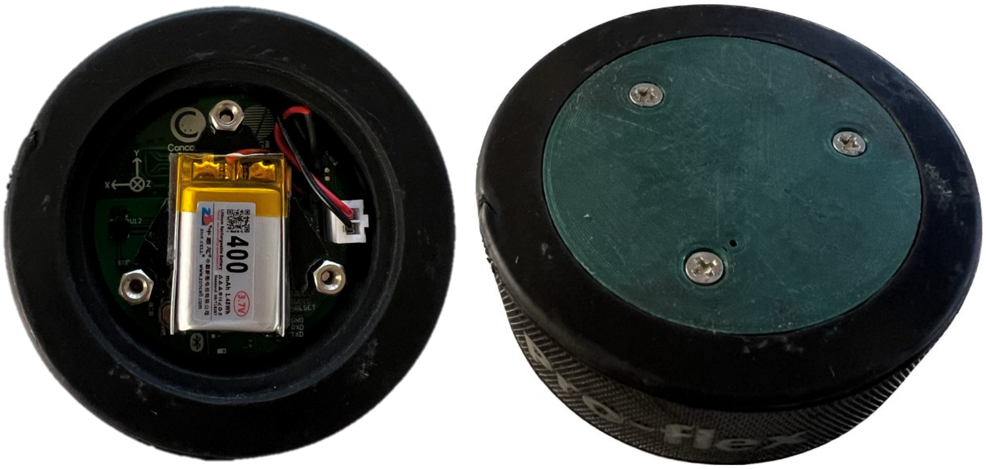


Figure 3.29: Sensor system and battery integrated into the hockey puck.

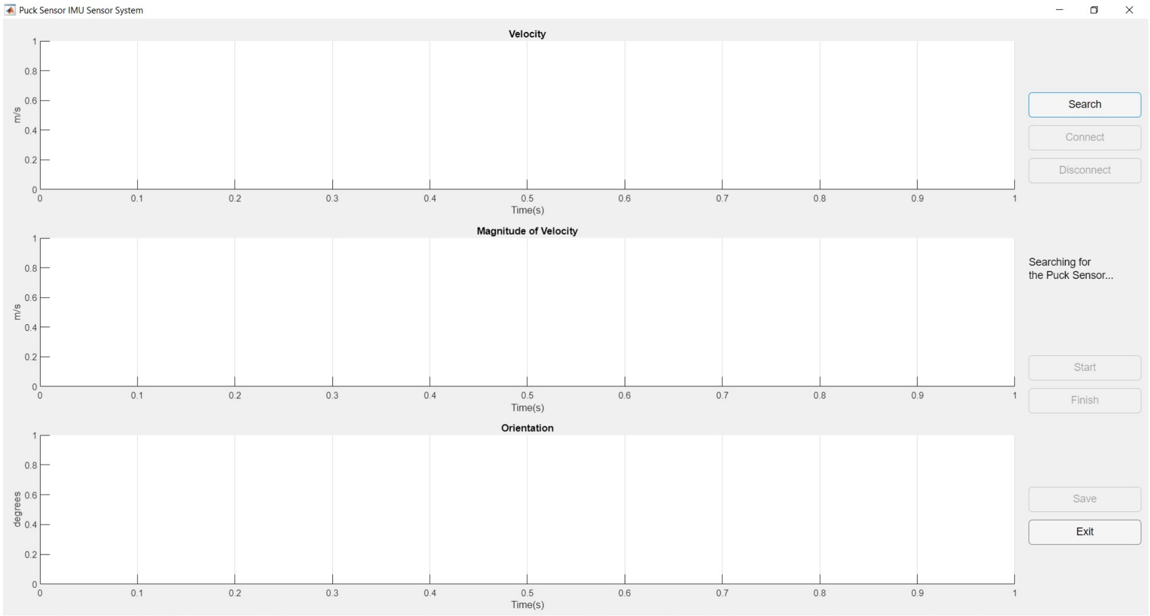
altering the puck's standard dimensions. The dimensions of a regulation ice hockey puck are a diameter of $76.2\ mm$ and a height of $25.4\ mm$. Furthermore, the final weight of the puck after integration was measured to be $137\ g$. This is approximately 12% to 19% lighter than the standard puck weight of $156\ g$ to $170\ g$. This integration might have also relocated the center of mass of the puck.

To compensate for the weight difference and possibly realign the center of mass, non-metallic and non-magnetic materials, such as rubber pieces, can be added. These additions should ensure that the weight and center of mass align closely with standard hockey pucks while not interfering with the BLE signal or magnetometer readings. Alternatively, the puck's cap design could be modified to fill the empty spaces to achieve the same goal.

3.10 Data Acquisition

"Data acquisition" was developed using MATLAB® R2022b in the App Designer environment. This design was utilized in a research study to gather and analyze sensor data efficiently. Each component on the interface had a specific callback function associated with it. Figure 3.30 provides a glance into the design view of the data acquisition system in the App Designer.

The functionality of each component in the data acquisition system is outlined as follows:



Searching for
the Puck Sensor...

Figure 3.30: The design view of the data acquisition system in MATLAB App Designer.

Search Button

To scan for available puck sensor systems, the ‘Search’ button utilizes the `blelist` function in MATLAB®. It initiates a search and waits until it identifies a device with the “Puck Sensor” string in its name. Once this device is found, the puck sensor’s complete name is returned, indicating that the puck sensor is within range and ready for connection.

Connect Button

The ‘Connect’ button establishes a connection to the specific BLE device identified by the ‘Search’ button using the `ble` function. Once connected, the code proceeds to access a particular characteristic responsible for sensor data using the `characteristic` function. This characteristic is uniquely identified using a specific Service UUID[1] and Characteristic UUID.

This characteristic object is assigned to a callback function, `readCharacteristicData`, to manage the receipt of new data packets. This function is notified each time a new data packet becomes available. This callback function enhances the efficiency as it only activates when new data is available, thus conserving MATLAB®’s computational resources. Following notification,

---

[1] Universally Unique Identifier

the function performs the subsequent processing steps:

(1) The function reads the “oldest” data packet available.

(2) The function then processes the final two bytes of the received packet as the “counter bytes”. These bytes increment by 1 in each iteration of the microcontroller. Therefore, if one or more packets are missing during data transmission, they are identifiable during later data processing stages.

(3) The remaining 210 bytes of the received packet are then reshaped into a 10×21 matrix to facilitate more straightforward and more understandable subsequent processing steps.

(4) The initial 6 bytes, representing the accelerometer readings, are converted to their respective voltage values. Subsequently, these voltage values are transformed into acceleration values using the accelerometer’s calibration data.

(5) The following 6 bytes are allocated for the 3-axis gyroscope readings. These bytes are converted into angular velocity values using the information in the gyroscope’s datasheet.

(6) The subsequent 2 bytes are dedicated to the high-rate Z-axis gyroscope. The ADC value extracted from these bytes is converted into a voltage reading, which is then transformed into the corresponding Z-axis angular velocity utilizing the information from the sensor’s datasheet.

(7) The final 7 bytes correspond to the magnetometer’s output. These bytes are transformed into the appropriate magnetic field readings based on the information provided in the sensor’s datasheet.

(8) During steps (4) to (7), the processed value is concatenated to the specific variable corresponding to the sensor reading at each iteration. Additionally, leveraging the counter bytes extracted in step (2), any missing rows in the data (due to missed packets) are populated with `NaN`[1] values, preserving the continuity of the data for further processing steps.

---

[1] Not a Number

Disconnect Button

The ‘Disconnect’ button terminates the connection with the specific BLE device. Initially, it unsubscribes from the designated characteristic using the `unsubscribe` function. Following this, the process disconnects from the BLE device using the `disconnect` function to remove the device object, thereby entirely severing the connection.

Start Button

Upon pressing the ‘Start’ button, the data acquisition process commences. All sensor-specific variables are initially cleared to ensure a fresh start. Subsequently, the program is set to receive and save each sensor data packet, preparing it for further analyses and operations.

Finish Button

The ‘Finish’ button signifies the completion of the desired measurements. When this button is pressed, the following post-acquisition operations are performed:

(1) The sensor outputs, excluding the magnetometer, are normalized by subtracting the average value measured during the initial 250 ms phase. This operation removes any baseline drift present in the data. Subsequently, the sensor outputs are aligned with the board’s defined coordinate system, corresponding to the accelerometer’s alignment.

(2) The instances in the sensor data where `NaN` values are present, indicating lost packets, are handled by applying linear interpolation using MATLAB®’s `fillmissing` function. This ensures that the sensor data is continuous and ready for further analysis.

(3) If the Z-axis of the 3-axis gyroscope reaches or exceeds its nominal range of 4000 dps for five or more data samples, its value is substituted with readings from the high-rate Z-axis gyroscope. This strategy ensures accurate angular velocity measurements during low-spin scenarios and prevents saturation during high-spin instances.

(4) After the data preparation, the IMU fusion algorithm, implemented using `insfilterMARG` (refer to section 2.3.2), is deployed for velocity and orientation estimation. The `predict`

function, which fuses accelerometer and gyroscope readings, is called for each data packet. In contrast, the magnetometer's output is incorporated into the fusion algorithm every $0.5\ s$, a rate empirically found to yield optimal results. Eventually, both the orientation and velocity estimates are obtained using the `pose` function.

(5) As the final step of this process, the estimated three-dimensional velocity vector, the magnitude of the velocity, and the orientation vector are visualized in three separate plots for comprehensive review and analysis. This visual representation aids in interpreting the sensor readings and validating the performance of the sensor system.

Save Button

When pressed, the 'Save' button prompts a dialog box, enabling the user to store the gathered IMU data and the estimated velocity vectors and orientation quaternion. The collected data is saved in a MAT-file format for potential future use.

Exit Button

The 'Exit' button terminates the ongoing BLE connection and clears all variables associated with the data acquisition session from MATLAB®'s workspace. Subsequently, the application window is closed, bringing the data acquisition process to a conclusive end. This button ensures a clean exit, preventing residual connection or memory usage issues.

The entire code used for the data acquisition in the MATLAB® App Designer environment, as described in the steps above, can be found in Appendix D. The comprehensive code provides a detailed understanding of the procedures and techniques for acquiring and processing sensor data.

Chapter 4

Results

This chapter focuses on validating the data collected from the puck sensor system developed in this work. For this purpose, the sensor system was put into real-world testing scenarios involving various hockey shots. The readings from the developed sensors were compared to those captured by a Vicon motion capture camera system. Table 4.1 summarizes the test scenarios conducted for this evaluation.

Table 4.1: Summary of test scenarios.

Test Name	Description
Push to Stop	This test involves pushing and spinning a stationary puck with a stick until it decelerates and stops.
Wrist Impact	This test involves gently pushing the puck toward an object, allowing it to halt upon impact.
Wrist Shot	This test involves shooting a hockey puck toward a goal using a standard wrist shot technique.
Snap Shot	This test involves shooting a puck using a quicker and more deceptive method than a wrist shot.
Slap Shot	This test involves striking a puck using hockey's most powerful shot technique.

The Vicon system provides an objective benchmark to measure the performance of the puck sensor system. For these trials, affixing a minimum of 3 specific markers to the puck, as demonstrated in Figure 4.1, was essential to enable accurate tracking by the Vicon camera system.

Figure 4.1: The Vicon-specific markers attached to the hockey puck.

Figure 4.2: Sequential images capturing the execution of the “Push to Stop” test.

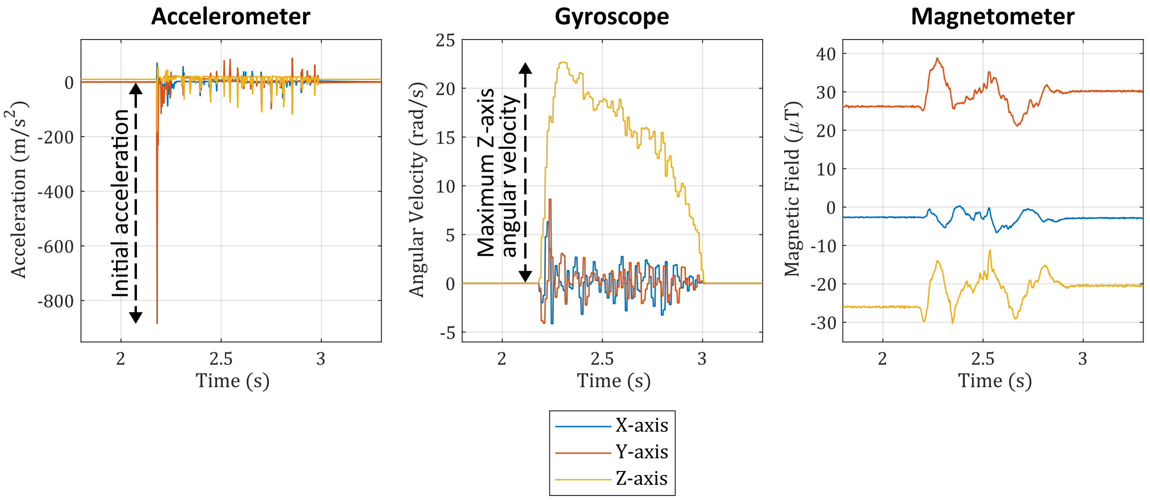
The testing procedure was as follows: One operator initiated data acquisition through the IMU’s GUI, while another activated the Vicon system. Once these systems were operational, a hockey player executed various shots. Data collection ceased once the puck came to a stop. Although there was an intent for both systems to begin data collection simultaneously, occasional timing discrepancies were noted. These discrepancies were subsequently calibrated. The plots presented in this chapter primarily focus on sensor data and velocity estimations captured during the puck’s active movement.

4.1 Push to Stop

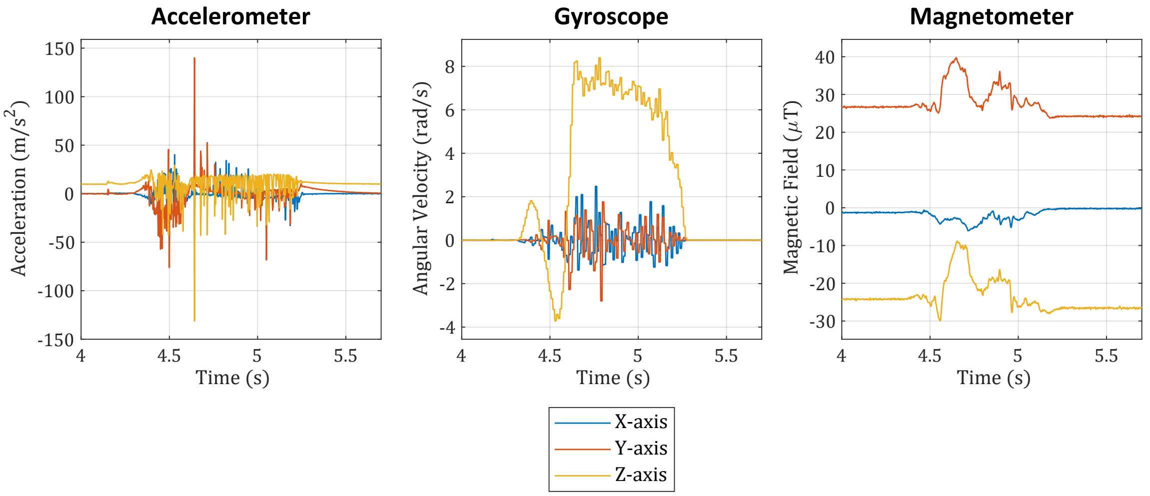
A “Push to Stop” test was performed using a hockey stick, similar to the one executed with the prototype (refer to section 3.8.1). The primary objective of this test was to evaluate the performance of the sensor system during low-power shots without any object impacts. In this experiment, a stationary hockey puck was lightly pushed using the hockey stick and allowed to glide until it naturally halted, as depicted in the captured images in Figure 4.2. The collected data from this test provided valuable insights into the sensor system’s performance and reliability in low-impact, real-world scenarios.

4.1.1 IMU Results

Figure 4.3 provides two examples of outputs recorded during the “Push to Stop” test, one demonstrating a peak velocity estimation with less error and another leading to an estimate with



(a) IMU readings leading to an estimation with less error.



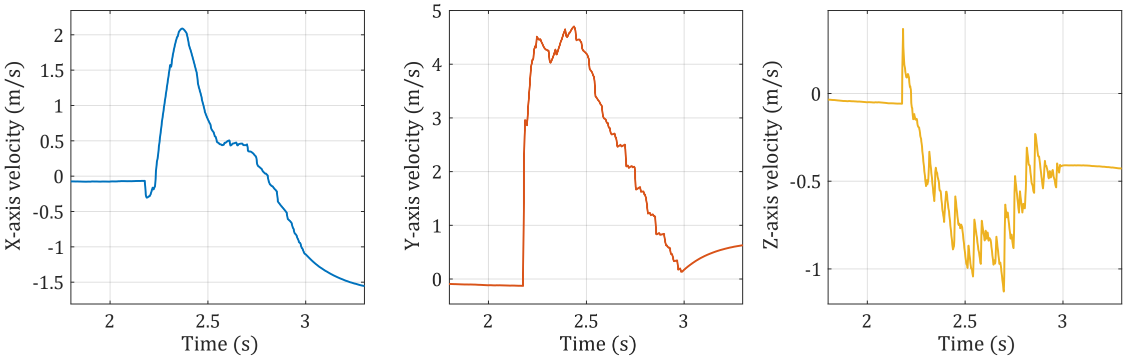
(b) IMU readings leading to an estimation with more error.

Figure 4.3: The sample IMU outputs during the “Push to Stop” test.

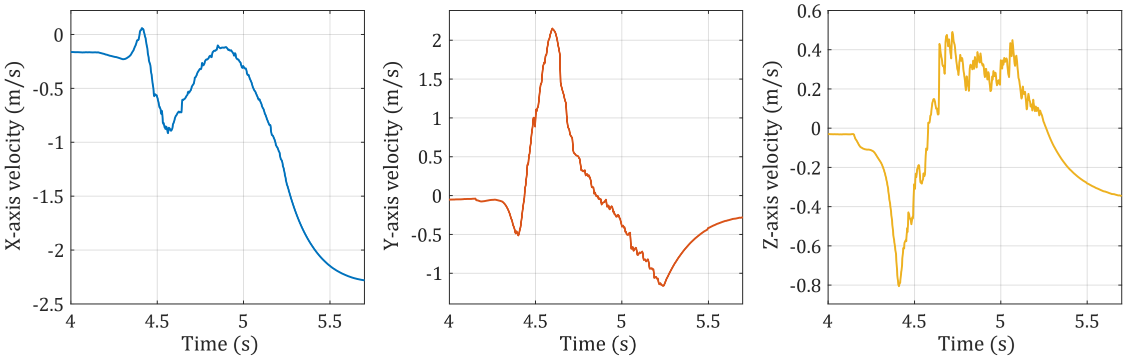
more error. These sample outputs allow for a comparative visual interpretation of the sensor data and the system’s varying performance in this test scenario.

4.1.2 Estimated Velocity

Figure 4.4 represents the estimated velocity vector during the “Push to Stop” test for both instances of IMU readings. This estimation was achieved using the MARG-based method. The initial orientation was defined as $\phi = 0°$, $\theta = 0°$, and $\psi = 0°$ in Euler angles or expressed as the quaternion $q_0 = 1 + 0\boldsymbol{i} + 0\boldsymbol{j} + 0\boldsymbol{k}$. The figure displays the puck’s speed in each axis of motion with

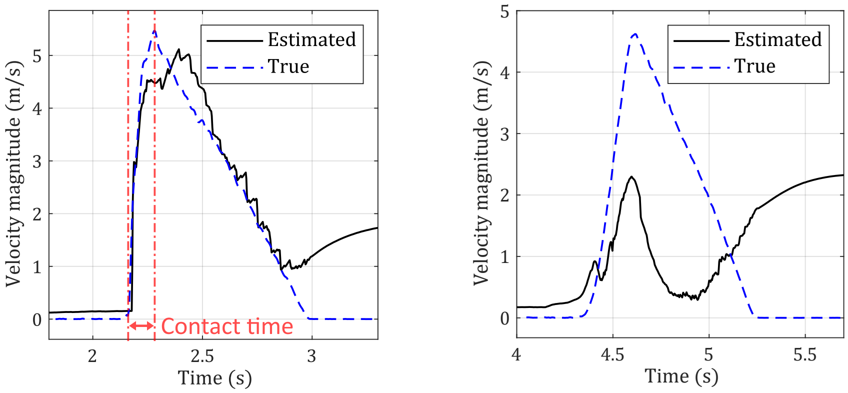


(a) Estimation with less error.



(b) Estimation with more error.

Figure 4.4: Estimated velocity vector during the "Push to Stop" test.



(a) Estimation with -4.37% error.

(b) Estimation with -49.75% error.

Figure 4.5: Comparison of estimated and true velocity magnitudes during the "Push to Stop" test.

respect to the Earth's FOR. While the Z-axis displacement and velocity were neglectable in this test, some variation is observed in the estimated velocity vector along this axis. This may originate from the vibrations and impacts influencing the Z-axis piezoelectric sensing component within the accelerometer during the experiment. It should be noted that the initial velocity observed in this figure is attributed to the integration error of acceleration data from Time $= 0\ s$ to the initial Time displayed in the plots.

Figure 4.5 compares the estimated and true velocity magnitudes acquired by the Vicon system for both estimation instances. This comparison helps in quantifying the accuracy of the sensor system.

The collective data from 10 trials of the "Push to Stop" test, including the estimated and true velocity, initial acceleration, maximum Z-axis angular velocity (as depicted in Figure 4.3a), and hockey stick contact time (as shown in Figure 4.5a), are summarized in Table 4.2. As a result, the estimated velocity deviates from the true velocity by $-1.225 \pm 0.939\ m/s$, resulting in an average -28.36% error across all trials.

4.2 Wrist Impact

The "Wrist Impact" test, similar to the prototype test (refer to section 3.8.1), was designed to evaluate the sensor system's response to low-power shots involving object impacts. In this setup,

Table 4.2: Summary of test replication results for the “Push to Stop” scenario.

Trial	**Initial Acceleration** (m/s^2)	**Max. Z-axis Angular Velocity** (rad/s)	**Contact Time** (s)	**True Velocity** (m/s)	**Estimated Velocity** (m/s)	**Error** (m/s)	**Error** $\%$
1	878	7.96	0.08	3.32	3.80	0.49	14.62
2	279	5.34	0.13	4.80	3.28	−1.52	−31.67
3	203	46.45	0.08	3.67	2.39	−1.28	−34.91
4	947	13.67	0.10	4.55	3.94	−0.60	−13.26
5	258	11.47	0.16	4.74	2.60	−2.14	−45.10
6	745	21.88	0.07	4.74	2.89	−1.84	−38.87
7	196	8.39	0.29	4.62	2.32	−2.30	−49.75
8	905	22.68	0.12	5.47	5.23	−0.24	−4.37
9	446	18.43	0.13	4.81	2.75	−2.06	−42.79
10	460	17.59	0.12	5.21	3.26	−1.96	−37.52

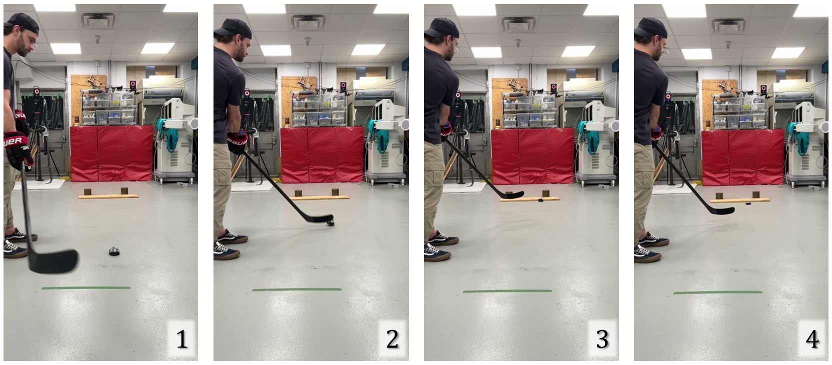
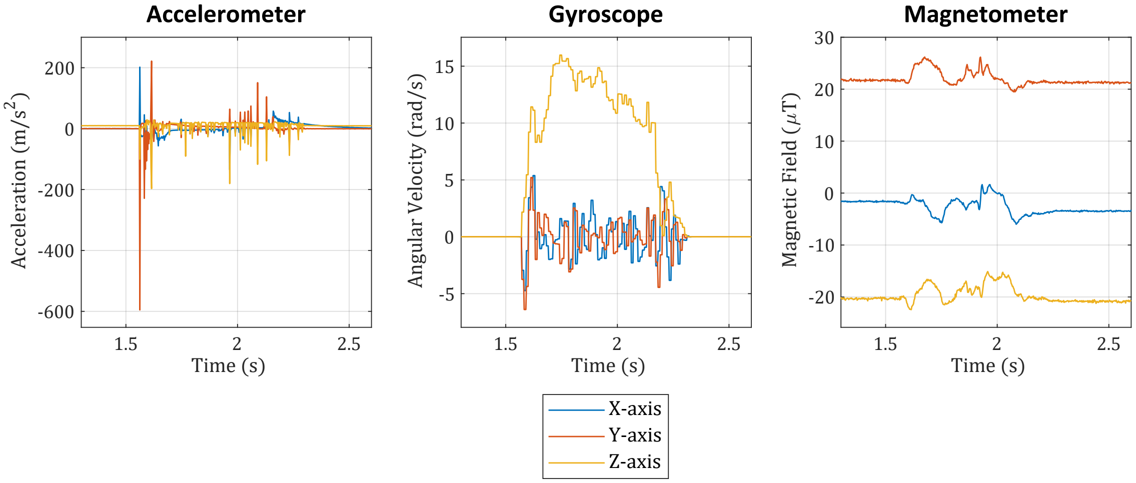


Figure 4.6: Sequential images capturing the execution of the “Wrist Impact” test.

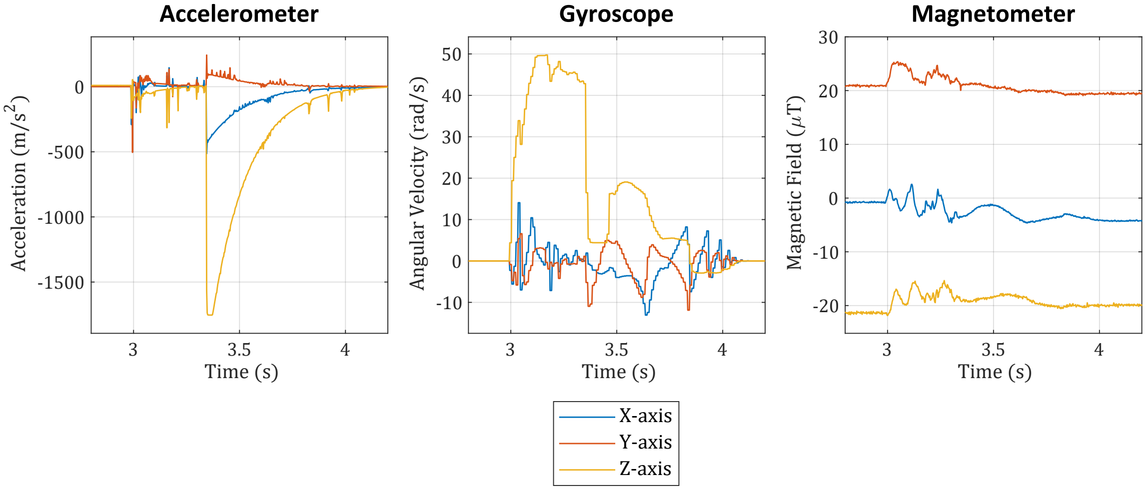
a stationary hockey puck was pushed with a hockey stick, allowing it to glide and eventually halt upon hitting an object, as shown in Figure 4.6. The collected data provided significant insights into the sensor system’s performance and reliability when dealing with impacts in real-world scenarios.

4.2.1 IMU Results

Figure 4.7 demonstrates two instances of IMU output from the “Wrist Impact” test, graphically representing the system’s response to an impact scenario.



(a) IMU readings leading to an estimation with less error.



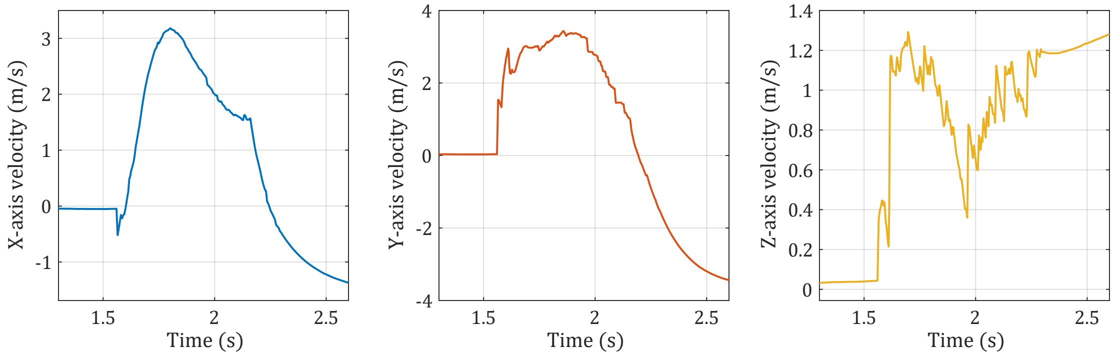
(b) IMU readings leading to an estimation with more error.

Figure 4.7: The sample IMU outputs during the “Wrist Impact” test.

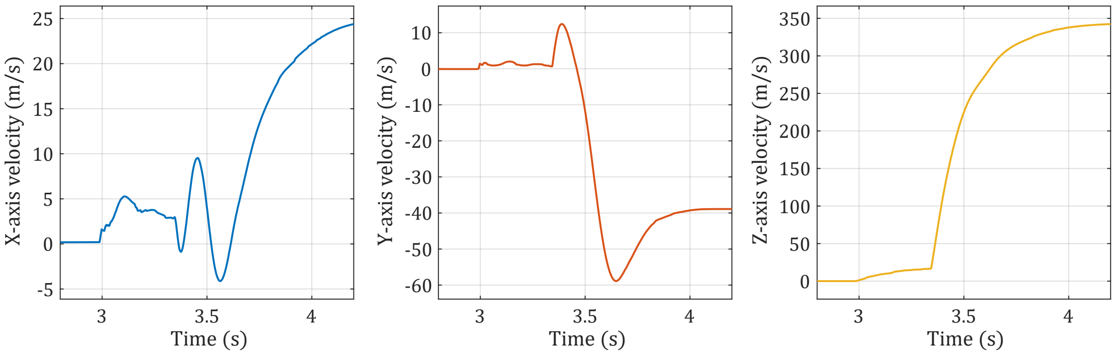
4.2.2 Estimated Velocity

Figure 4.8 depicts the estimated velocity vector during the “Wrist Impact” test. Figure 4.9 compares estimated and true velocity magnitudes during the “Wrist Impact” test.

The “Wrist Impact” test was conducted 10 times under the same conditions, with the sensor estimating the velocity in 8 trials. As outlined in Table 4.3, the compiled data from these repetitions demonstrate that the estimated velocity differs from the true velocity by an average of $-1.052 \pm 0.905\ m/s$, with -20.63% error across estimations.

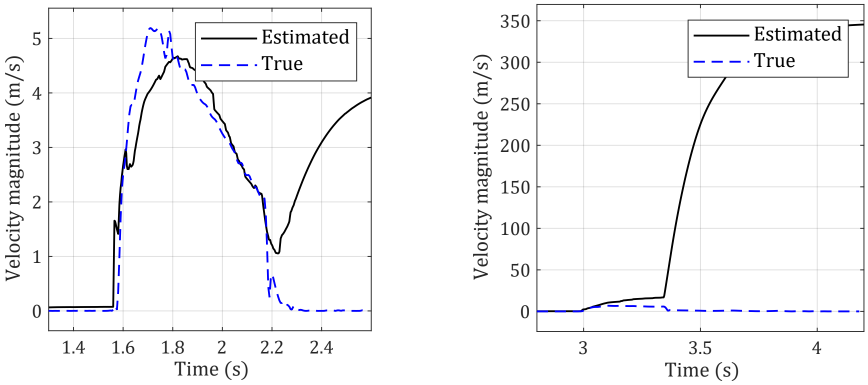


(a) Estimation with less error.



(b) Estimation with more error.

Figure 4.8: Estimated velocity vector during the "Wrist Impact" test.



(a) Estimation with -10.2% error.

(b) Estimation with significant error.

Figure 4.9: Comparison of estimated and true velocity magnitudes during the "Wrist Impact" test.

Table 4.3: Summary of test replication results for the “Wrist Impact” scenario.

Trial	**Initial Acceleration** (m/s^2)	**Max. Z-axis Angular Velocity** (rad/s)	**Contact Time** (s)	**True Velocity** (m/s)	**Estimated Velocity** (m/s)	**Error** (m/s)	**Error** %
1	699	10.73	0.15	5.14	3.65	-1.49	-28.96
2	518	16.85	0.18	5.13	6.11	0.98	19.04
3	586	21.65	0.14	5.46	3.77	-1.69	-30.93
4	636	15.98	0.14	5.19	4.66	-0.53	-10.19
5	541	13.92	0.18	5.30	4.14	-1.17	-22.00
6	805	15.56	0.23	4.55	2.81	-1.74	-38.30
7	692	13.57	0.15	5.15	3.89	-1.26	-24.49
8	344	14.16	0.16	5.18	3.67	-1.52	-29.25
9	578	49.74	0.11	6.82	–	–	–
10	594	21.25	0.17	5.69	–	–	–

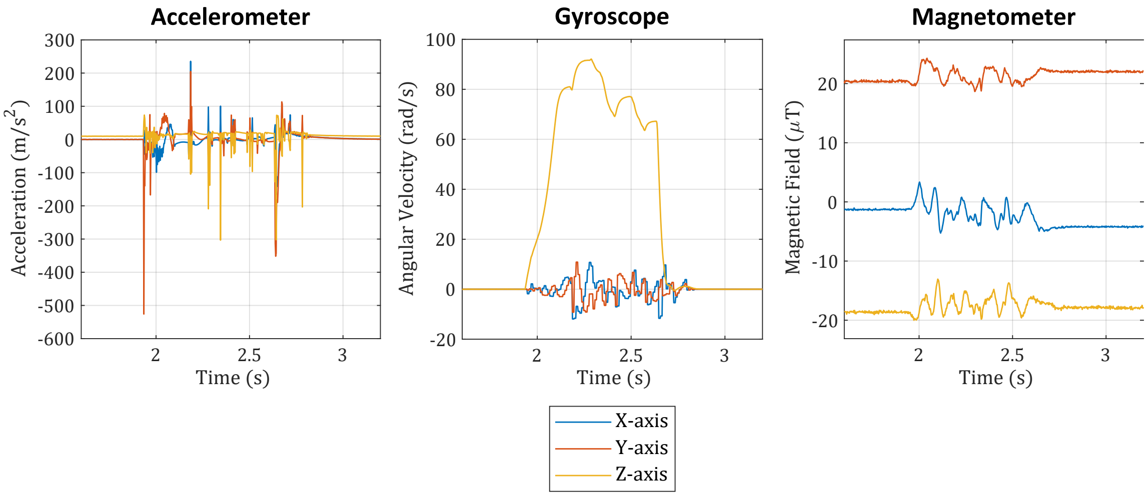
Figure 4.10: Sequential images capturing the execution of the “Wrist Shot” test.

4.3 Wrist Shot

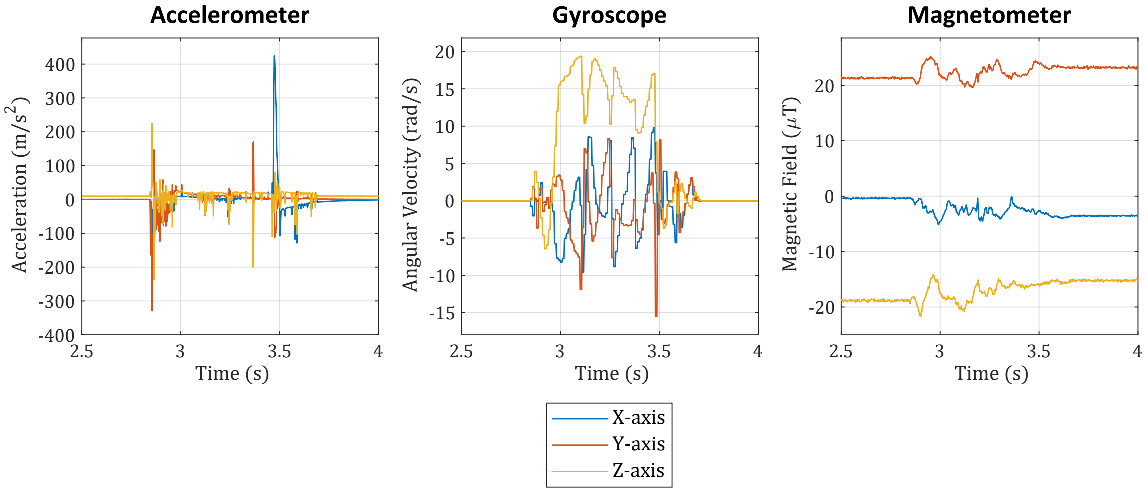
The “Wrist Shot” test evaluated the sensor system’s performance during scenarios common in hockey games. This test setup involved a hockey puck shot toward a goal using a standard wrist shot technique. As depicted in Figure 4.10, the puck was launched, made contact with the goal pads, and eventually stopped. The data collected from this experiment is crucial for assessing the sensor system’s accuracy and reliability in handling low-speed, high-spin dynamic motions.

4.3.1 IMU Results

Figure 4.11 provides examples of IMU output recorded during the “Wrist Shot” test, offering a graphical depiction of the system’s response under a dynamic scenario that involves low-speed, high-spin motion.

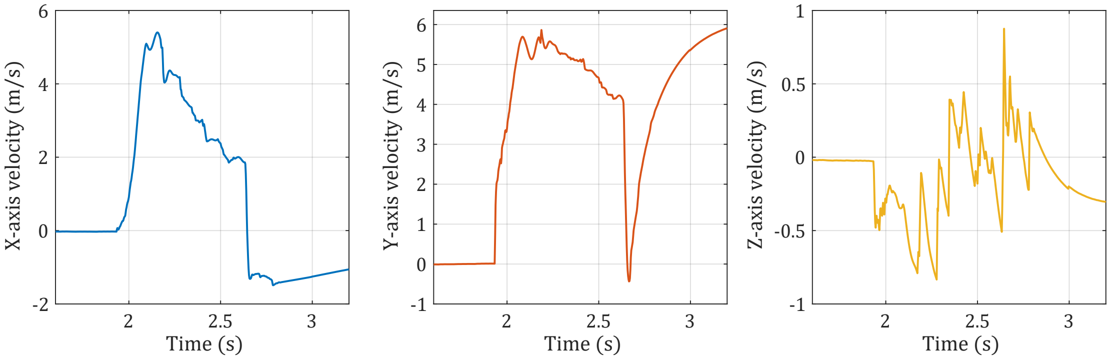


(a) IMU readings leading to an estimation with less error.

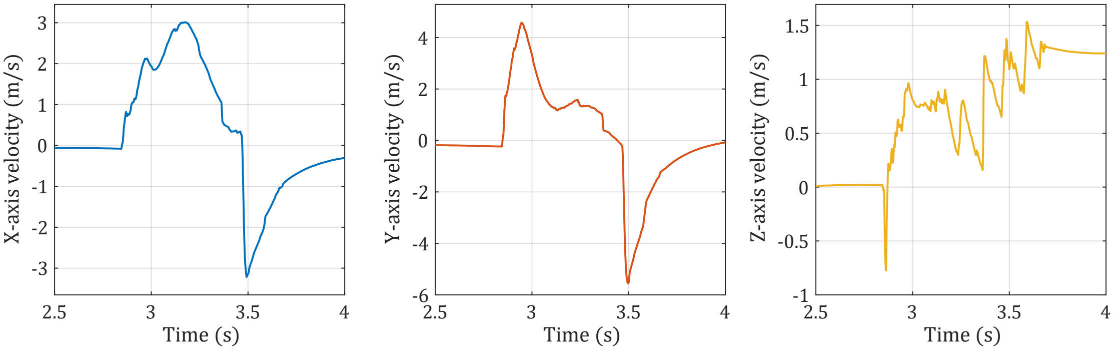


(b) IMU readings leading to an estimation with more error.

Figure 4.11: The sample IMU outputs during the “Wrist Shot” test.



(a) Estimation with less error.



(b) Estimation with more error.

Figure 4.12: Estimated velocity vector during the “Wrist Shot” test.

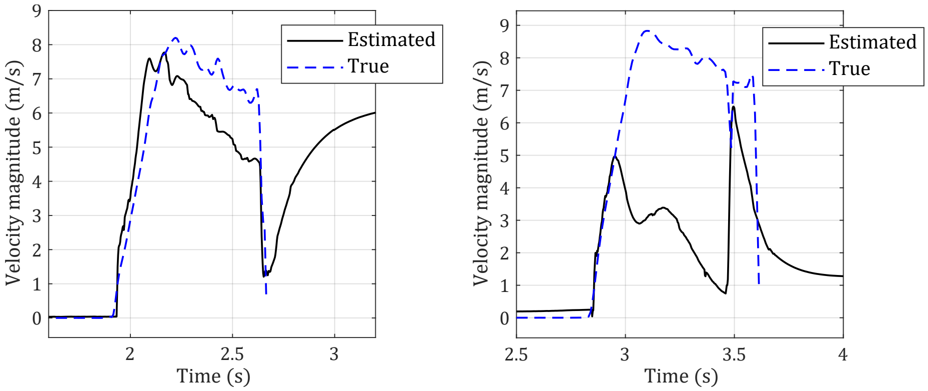
4.3.2 Estimated Velocity

Figure 4.12 presents the estimated velocity vector during the “Wrist Shot” test, demonstrating the puck’s trajectory during the execution of a moderate-power shot. Figure 4.13 compares the estimated and actual velocity magnitudes during the “Wrist Shot” test.

The “Wrist Shot” test was repeated 20 times with different shot power and spins. As detailed in Table 4.4, the aggregated data indicates that the estimated velocity deviates from the actual speed by an average of $-2.05 \pm 1.18\ m/s$ and an average -23.08% error.

4.4 Snap Shot

The “Snap Shot” test was carried out to evaluate the sensor system’s performance during fast, sudden movements. This test involved a hockey puck being shot toward a goal using the snap shot technique, which is quicker and often more deceptive than a standard wrist shot. As illustrated in



(a) Estimation with -4.72% error.

(b) Estimation with -41.34% error.

Figure 4.13: Comparison of estimated and true velocity magnitudes during the “Wrist Shot” test.

Table 4.4: Summary of test replication results for the “Wrist Shot” scenario.

Trial	**Initial Acceleration** (m/s^2)	**Max. Z-axis Angular Velocity** (rad/s)	**Contact Time** (s)	**True Velocity** (m/s)	**Estimated Velocity** (m/s)	**Error** (m/s)	**Error** %
1	555	18.52	0.16	7.96	6.37	-1.59	-19.96
2	455	29.04	0.23	7.67	4.79	-2.88	-37.56
3	912	18.00	0.14	8.89	7.24	-1.64	-18.51
4	642	17.80	0.12	8.81	6.48	-2.33	-26.42
5	896	19.03	0.17	8.21	5.59	-2.63	-31.97
6	281	12.79	0.14	8.10	5.51	-2.58	-31.90
7	415	19.37	0.15	8.83	5.18	-3.65	-41.34
8	614	21.22	0.14	7.81	5.11	-2.70	-34.59
9	736	12.95	0.13	7.99	6.22	-1.77	-22.19
10	179	29.68	0.14	8.90	5.41	-3.50	-39.27
11	276	90.07	0.15	11.90	8.70	-3.20	-26.90
12	213	62.29	0.18	7.88	7.27	-0.61	-7.73
13	137	61.48	0.23	9.43	6.92	-2.50	-26.55
14	644	62.86	0.14	10.75	7.47	-3.29	-30.56
15	601	68.90	0.13	7.96	7.31	-0.65	-8.11
16	482	65.29	0.14	8.81	8.09	-0.72	-8.18
17	365	63.48	0.15	8.42	6.45	-1.97	-23.43
18	861	76.57	0.20	10.07	9.28	-0.79	-7.81
19	423	58.76	0.17	9.60	8.26	-1.34	-13.98
20	530	92.11	0.21	8.20	7.81	-0.39	-4.72

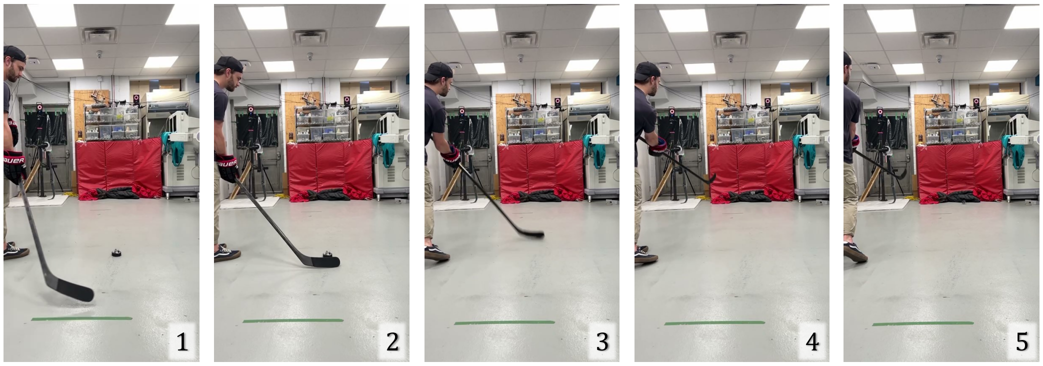


Figure 4.14: Sequential images capturing the execution of the “Snap Shot” test.

Figure 4.14, the puck was propelled, impacted the goal pads, and ultimately came to rest. The data gathered from this test is vital for evaluating the sensor system’s precision and dependability when managing high-speed, sudden motions.

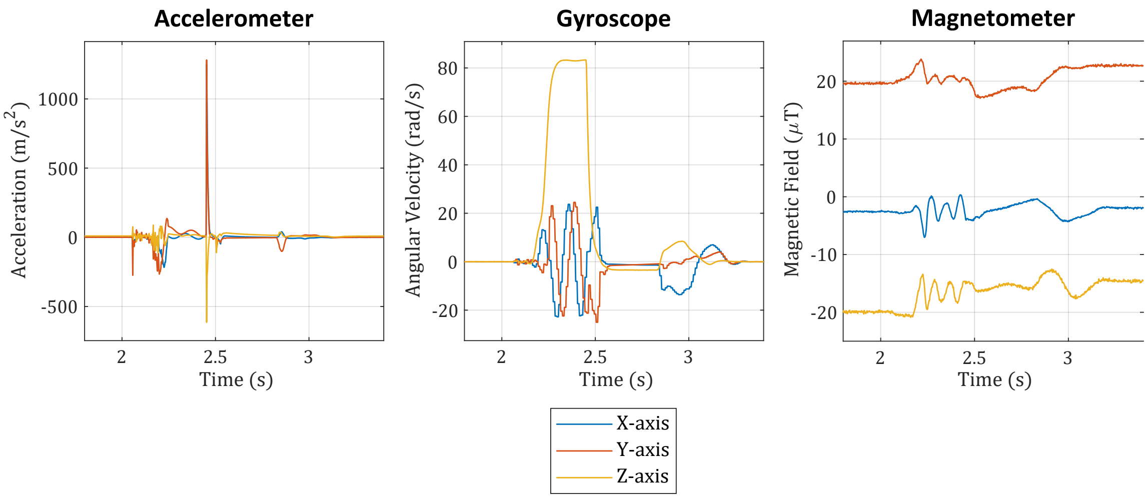
4.4.1 IMU Results

Figure 4.15 shows samples of IMU outputs during the “Snap Shot” test, offering a graphical depiction of the system’s response to the high-speed, abrupt motions typical of snap shots. Notably, upon the puck’s impact with the goal pad, the accelerometer’s output exceeded its measuring range, i.e., $100\ g$ or $981\ m/s^2$, potentially leading to inaccurate estimations.

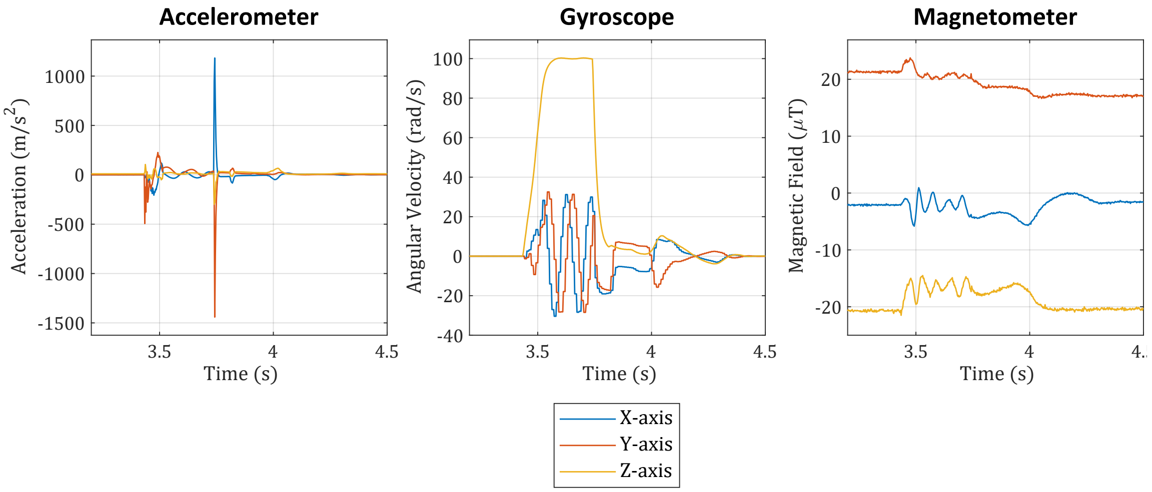
4.4.2 Estimated Velocity

Figure 4.16 illustrates the estimated velocity vector during the “Snap Shot” test, capturing the puck’s trajectory during the execution of high-speed, immediate shots. Figure 4.17 contrasts the estimated and true velocity magnitudes during the “Snap Shot” test.

The “Snap Shot” test was conducted 10 times, incorporating different power levels and spins. As illustrated in Table 4.5, the collated data indicates that the inconsistency between estimated and actual velocities averages $-1.30 \pm 1.21\ m/s$, and -7.05% average error.



(a) IMU readings leading to an estimation with less error.

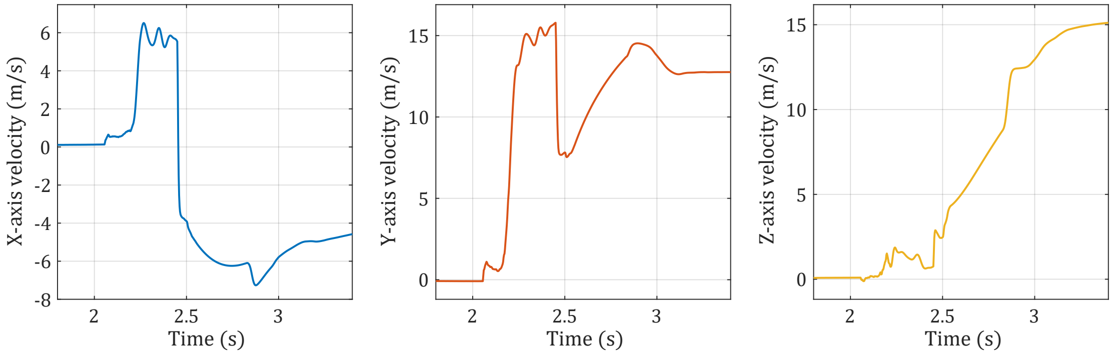


(b) IMU readings leading to an estimation with more error.

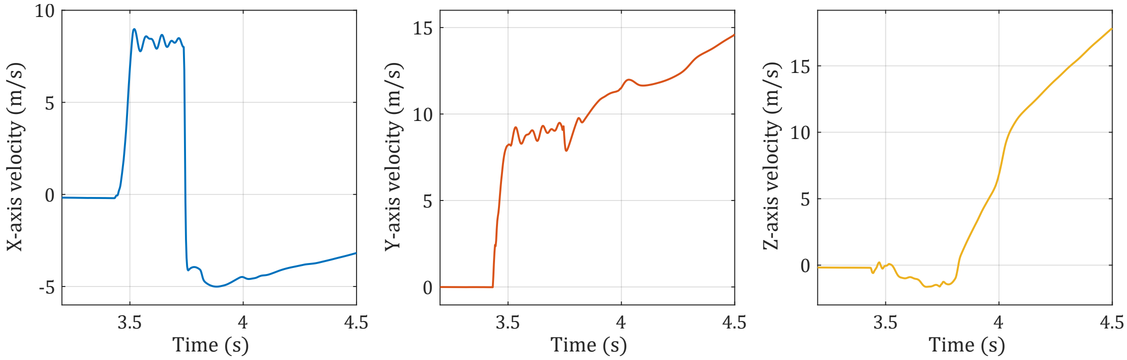
Figure 4.15: The sample IMU outputs during the "Snap Shot" test.

Table 4.5: Summary of test replication results for the "Snap Shot" scenario.

Trial	**Initial Acceleration** (m/s^2)	**Max. Z-axis Angular Velocity** (rad/s)	**Contact Time** (s)	**True Velocity** (m/s)	**Estimated Velocity** (m/s)	**Error** (m/s)	**Error** %
1	857	64.49	0.08	16.19	14.18	-2.01	-12.40
2	324	64.69	0.13	15.65	15.32	-0.33	-2.13
3	495	100.35	0.09	15.43	12.28	-3.16	-20.45
4	297	83.30	0.10	16.79	16.60	-0.19	-1.12
5	391	103.88	0.10	22.00	19.93	-2.07	-9.42
6	496	71.26	0.08	20.33	18.74	-1.58	-7.78
7	975	81.79	0.07	20.53	20.07	-0.46	-2.24
8	362	84.11	0.09	20.81	18.07	-2.74	-13.16
9	421	67.93	0.10	19.87	18.82	-1.06	-5.32
10	166	70.67	0.09	16.11	16.68	0.57	3.51

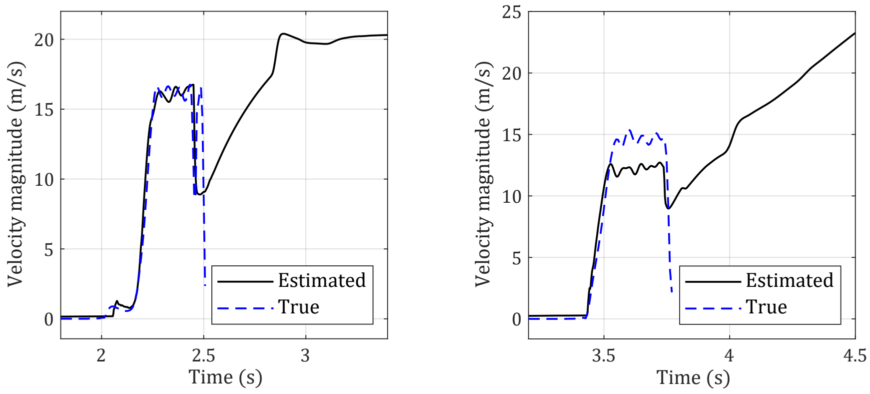


(a) Estimation with less error.



(b) Estimation with more error.

Figure 4.16: Estimated velocity vector during the “Snap Shot” test.



(a) Estimation with -1.12% error.

(b) Estimation with -20.45% error.

Figure 4.17: Comparison of estimated and true velocity magnitudes during the “Snap Shot” test.

Figure 4.18: Sequential images capturing the execution of the “Slap Shot” test.

4.5 Slap Shot

The “Slap Shot” test assessed the sensor system’s capability to handle high-power shots, characteristic of hockey games. This test setup required a hockey puck to be struck toward a goal using the forceful slap shot technique, often considered the most potent shot in hockey. As displayed in Figure 4.18, the puck was launched with significant force, collided with the goal pads, and finally stopped. The data obtained from this experiment are crucial for analyzing the sensor system’s accuracy and reliability under high-speed, high-impact conditions.

4.5.1 IMU Results

Figure 4.19 illustrates a sample IMU output during the “Slap Shot” test, visualizing the sensor system’s response to slap shots’ extreme speed and force. During the trial, the accelerometer’s readings across all axes and the gyroscope’s readings on the X and Y axes exceeded their measurement range. This excess could lead to inaccurate estimations, underscoring the challenges of high-speed, high-spin puck movements.

In the accelerometer’s outputs, saturation was observed beyond the $100\ g$ threshold, suggesting that during impactful moments, the accelerations might considerably exceed their predefined limits. An earlier study provided details of the average puck acceleration during slap shots, represented solely by its mean and standard deviation, with a peak of approximately $81.3\ g$ [37]. The specific accelerations experienced during direct puck impacts were not delineated in this study, leaving room for further exploration and clarification in subsequent research.

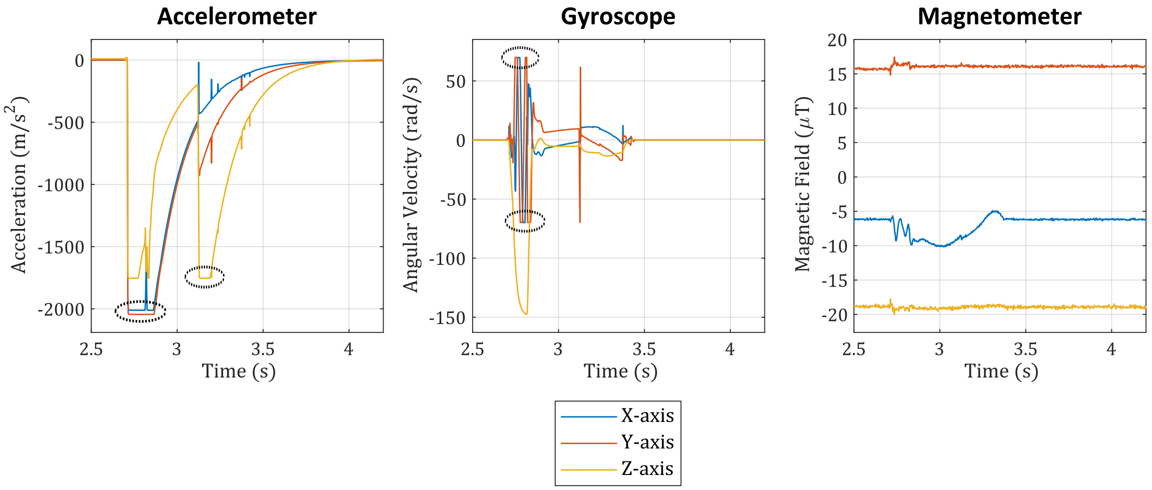


Figure 4.19: The sample IMU output during the "Slap Shot" test. The saturated measurements are marked.

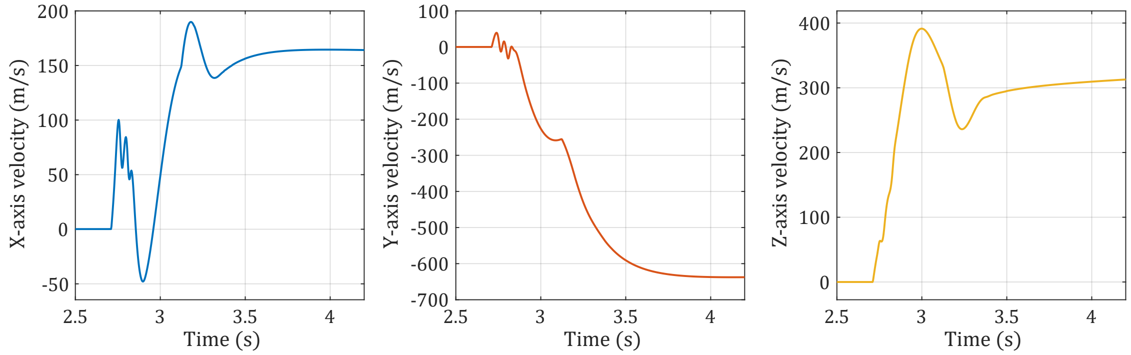


Figure 4.20: Velocity vector estimation during the "Slap Shot" test.

4.5.2 Estimated Velocity

Figure 4.20 presents the velocity vector estimation during the "Slap Shot" test, capturing the puck's trajectory in response to a rapid, fast-rotating, forceful shot with significant estimation error.

Figure 4.21 represents the estimated velocity magnitude during the "Slap Shot" test, with the actual velocity being $25.56\ m/s$, measured using an IR speed meter. While the IMU system could identify the occurrence of the peak velocity on the plot, it failed to estimate the velocity magnitude accurately. The inability to generate a credible analysis could be attributed to exceeding the accelerometer's and gyroscope's measuring range.

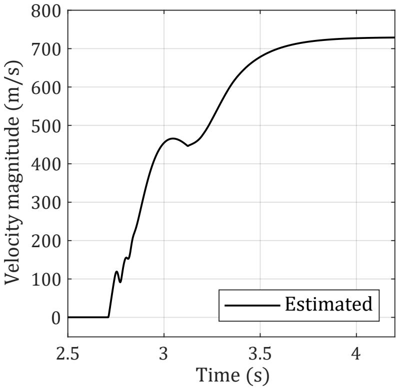


Figure 4.21: Estimated velocity magnitude during the “Slap Shot” test, exhibiting significant error.

Chapter 5

Conclusion and Discussion

This final chapter aims to synthesize the significant findings of this study, providing a comprehensive overview of the achieved outcomes and possible future improvements in the context of the research objectives.

5.1 Conclusion

The research presented in this thesis has outlined the comprehensive process involved in the design, development, and testing of a sensor system incorporated into a standard hockey puck. The system was equipped with IMUs, including an accelerometer, gyroscope, and magnetometer, allowing for measuring acceleration, angular velocity, and the magnetic field.

The sensor data was sent via Bluetooth Low Energy (BLE) for in-depth analysis. Enclosed within a hockey puck, the sensor system operated on an onboard battery. A highly efficient power management system extended the battery life to 200 days in shutdown mode. Furthermore, this system ensured consistent, reliable operation for 8 hours in active measuring mode.

During testing, it was observed that forceful shots could potentially lead to the saturation of the accelerometer and gyroscope components. This highlighted the need for further research into sensors or solutions to address these saturation events, ensuring the sensor system's accuracy and reliability under all scenarios.

Significant mechanical development was required to accommodate the sensor system and the battery inside a standard hockey puck. This process involved hollowing out the puck, designing and 3-D printing a custom-designed battery holder and encapsulating cap, and incorporating spacers to ensure a secure and optimal fit for the circuit board. The design provided the sensor system's integrity and functionality under various testing scenarios by effectively shielding sensitive internal components.

Table 5.1 summarizes all the sensor system's characteristics.

Table 5.1: Puck IMU sensor system characteristics

Parameter	Condition	Value	Unit
Accelerometer			
Range	Nominal	±981	m/s^2
Resolution	Nominal	0.05	m/s^2
RMS noise	Measured by ADC	0.11	$m/s^2{}_{RMS}$
3-axis gyroscope			
Range	Nominal	±69.81	rad/s
Resolution	Nominal	2.1	$10^{-3} \times rad/s$
RMS noise	Measured	2.44	$10^{-3} \times rad/s_{RMS}$
High-rate Z-axis gyroscope			
Range	Nominal	±349	rad/s
Resolution	Nominal	10.91	$10^{-3} \times rad/s$
Noise density	Measured by ADC	1.7	$10^{-3} \times rad/s/\sqrt{Hz}$
Magnetometer			
Range	Nominal	±800	μT
Resolution	Nominal	3.05	$10^{-3} \times \mu T$
RMS noise	Measured	10.7	$10^{-3} \times \mu T_{RMS}$
Wireless Range	Maximum	30	m
Battery Life	Measuring mode	8	$hours$
	Deep sleep mode	200	$days$
Mechanical characteristics			
Diameter		76.2	mm
Height		25.4	mm
Weight		137	g

The study proposed three distinct methods for estimating the puck's velocity from the collected IMU data. Among these, a technique utilizing a MATLAB® function based on Extended Kalman Filters (EKF) yielded superior results. Besides, a prototype system was developed and tested to validate these methods and verify the sensors' performance before the final design was implemented.

Following the development phase, the sensor system's performance was evaluated using Vicon motion cameras. Tests varied from low-impact shots to high-power, fast-spinning shots, replicating realistic scenarios. The measured error rates in determining the puck's peak velocity are presented in Chapter 4.

The absolute error in velocity estimation exhibited a trend across 48 different shots tabulated in Table 4.2 through Table 4.5, including even the saturated measurements. As illustrated in Figure 5.1, there is an inverse relationship between the true peak velocity and the estimation errors: as the true velocity increases, the absolute value of the error remains relatively constant, suggesting that higher true velocities tend to be associated with proportionally smaller errors. These findings can guide future research and optimization endeavors.

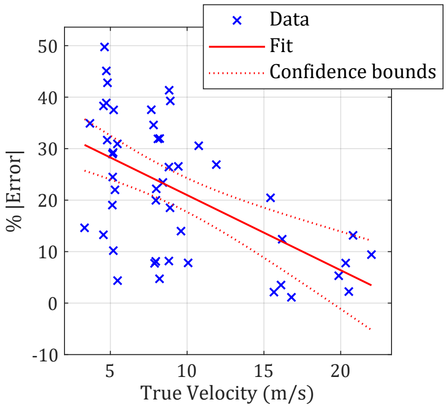


Figure 5.1: Linear regression model fit between estimated error and true velocity.

The sensor system designed in this study signifies a marked advancement in sports technology, particularly for hockey analytics. Compact enough to fit within a standard hockey puck, the system integrates an accelerometer, two gyroscopes, and a magnetometer with high measurement ranges and satisfactory accuracy. It transmits data wirelessly, making it a practical solution. It also houses a robust power management system, facilitating up to 8 hours of continuous operation or 200 days in standby mode. This surpasses other commercial and research solutions regarding versatility and longevity, offering valuable insights into puck dynamics, potentially developing game strategies, player performance, and scientific research.

5.2 Future Work

Given the broad scope of this study, future work suggestions are categorized according to the specific research areas involved. The following sections provide these recommendations.

Sensor System

While the component selection in this study was partly influenced by the chip shortage following the COVID-19 pandemic, a range of factors were considered in system optimization. As supply chains recover and further advancements emerge, the following refinements are proposed:

- *Accelerometer Range:* Some tests, as noted in sections 4.4 and 4.5, exceeded the accelerometer's range. The 830M1 series by TE Connectivity offers wider ranges within the same package size. Adopting this would not necessitate board design changes.

- *Gyroscope Range:* Outputs in section 4.5 surpassed the gyroscope's range for the X and Y axes. Considering three-axis high-range gyroscopes in subsequent designs is advised.

- *Magnetometer Range:* The current magnetometer met requirements, yet future iterations might leverage devices with a narrower range but superior resolution.

  Table 5.2 offers a brief overview of potential IMU range upgrades.

Table 5.2: Suggested IMU enhancement ranges.

Component	**Suggested Range**
Accelerometer	3-axis, $\pm 500\ g$
Gyroscope	3-axis, $\pm 10000\ ^{\circ}/s$
Magnetometer	3-axis, $\pm 1\ G$

- *MEMS Accelerometer:* The utilized piezoelectric accelerometers, while precise, do not offer a DC response. Integrating MEMS accelerometers could provide a broader response range, capturing static and dynamic accelerations.

- *Filter Design:* Regular underestimations in peak velocity were noted, potentially due to the anti-aliasing filter on the accelerometer. Filter adjustments or recalibration post-filtering could rectify this.

- *Battery Voltage Indicator:* The incorporated battery voltage indicator circuit was unused. Future improvements could include the development of a dedicated library for this indicator.

- *Battery Type:* Using a Li-Po battery currently, the switch to coin cell batteries might increase system durability, given their resistance to swelling.

- *Accelerometer Repositioning:* To better understand the numerical effect of centripetal acceleration, controlled rotations could be applied to the sensor system. The acquired data might slightly guide a repositioning of the accelerometer, mitigating this acceleration type's impact.

Puck Preparation

- *3-D Printed Parts:* The sensor system's 3-D printed parts, used in a dummy puck, showed signs of failure after high-speed shots, notably breaking at approximately 22 m/s, as illustrated in Figure 5.2. Utilizing stronger materials or enhanced designs could offer better resistance to such impacts.

Figure 5.2: Components damaged after high-speed impacts.

- *Resin Encapsulation:* Encapsulating the PCB with resin improves durability. However, this approach restricts future modifications due to the permanent seal.
- *Vibration Isolation:* Implementing vibration isolation materials, especially around the accelerometer, may mitigate surge effects from puck impacts. However, careful application is essential to avoid unwanted attenuation of accelerations.

Velocity Estimation

An EKF from MATLAB® validated the sensor system's functionality. The study focused not on the filter's details but its application as a proof-of-concept. Future research can refine this velocity estimation approach and delve deeper into the filter's design.

- *Delay Compensation:* Varying sensor and filter bandwidths may introduce signal delay. Adapting the EKF to account for these differences can enhance estimated velocity reliability.

- *Enhancing EKF Using External Systems:* To address acceleration integration errors in the EKF, external positioning tools can help improve velocity and position estimation:
  - Incorporate GPS data into the EKF for drift correction, refining velocity, and aiding precise position determination. The chosen GPS should be highly accurate.
  - Periodically integrate motion tracking cameras, like the Vicon system. While they provide accurate position data, considerations include their cost and potential coverage limitations.
- *Velocity Endpoint Detection:* In the conducted trials, the puck consistently comes to a stop, but the EKF does not represent this behavior. Future work should emphasize refining the detection of terminal conditions, particularly as the derivative of the IMU data approaches zero. Such improvements can lead to a more accurate representation of the puck's velocity profile throughout its trajectory.
- *Reverse Data Feeding:* In hockey, players often dribble before shooting, posing challenges for peak velocity estimation. A proposed solution is to reverse-feed the IMU data into the EKF, starting from the static endpoint and tracing back to the initial motion. This adjustment could better replicate typical in-game scenarios and enhance velocity estimation accuracy.
- *Integration of Sensor Fusion Techniques:* Enhancing velocity estimation reliability may involve exploring sensor fusion techniques like Complementary and Mahony filters, which can reduce sensor bias and noise for more accurate results.
- *Adaptive Kalman Filtering:* The existing model has fixed noise covariance values, varying with different conditions. An adaptive Kalman filter that adjusts noise covariance based on measurements can enhance velocity estimation in varied scenarios.
- *Deep Learning Approaches:* Using a significant dataset of labeled IMU readings, deep learning models offer the potential to predict velocity, capturing the complex nonlinear relationships that traditional filter-based methods might miss.

Appendix A

Schematic Design

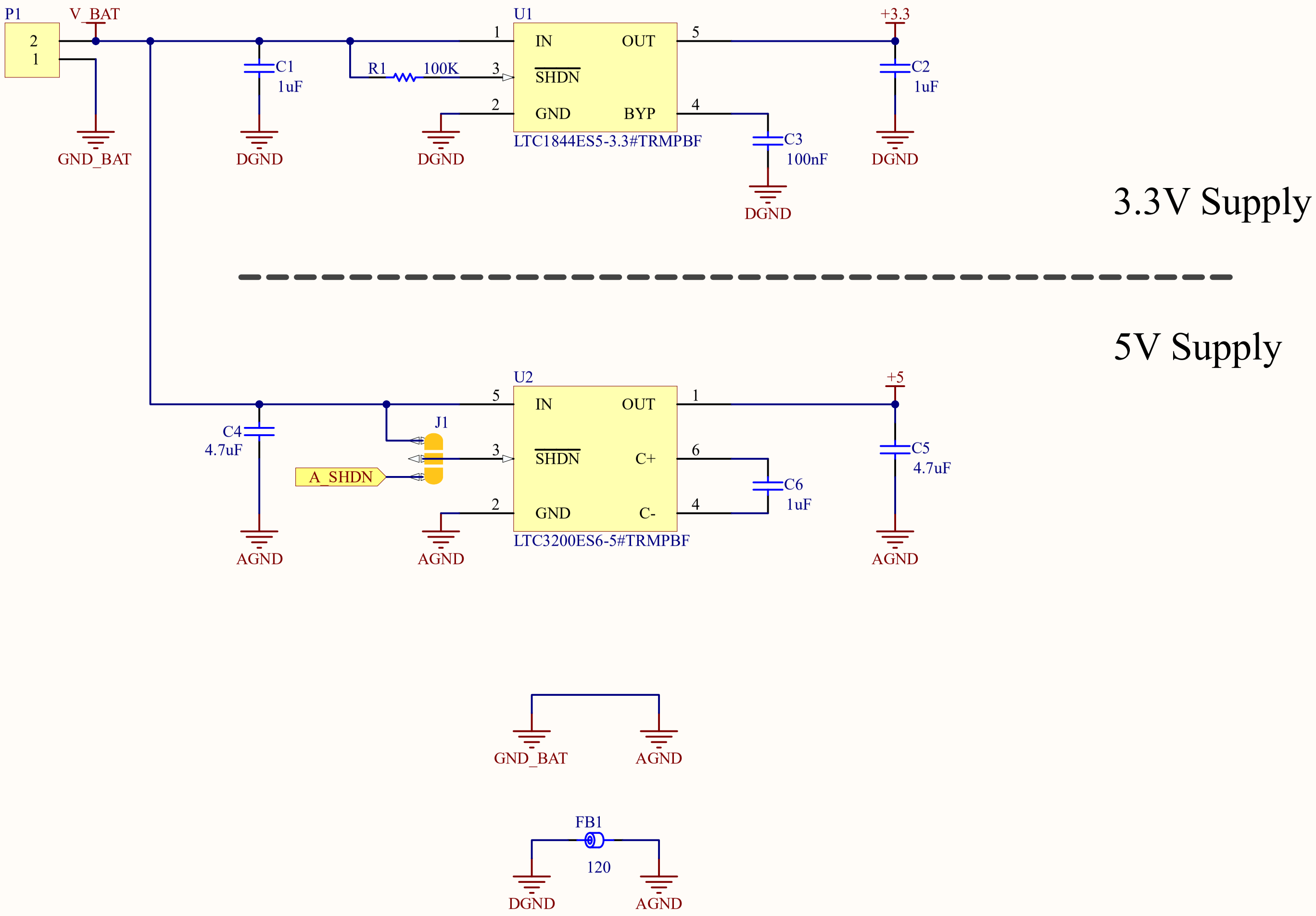
The schematic design of the hockey puck sensor system was developed in the Altium Designer environment. The design encompasses sheets for:

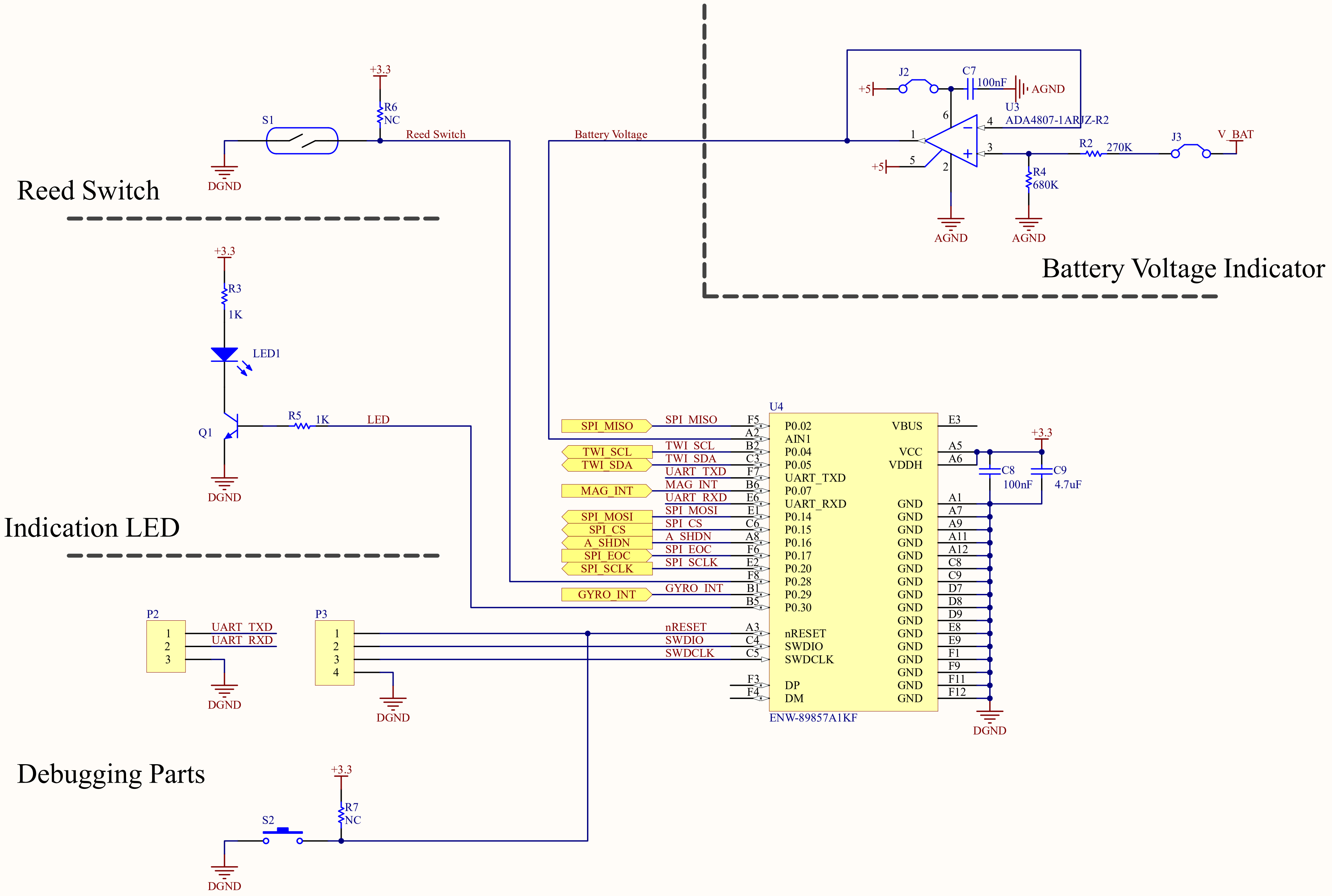
(1) Power Supply Unit

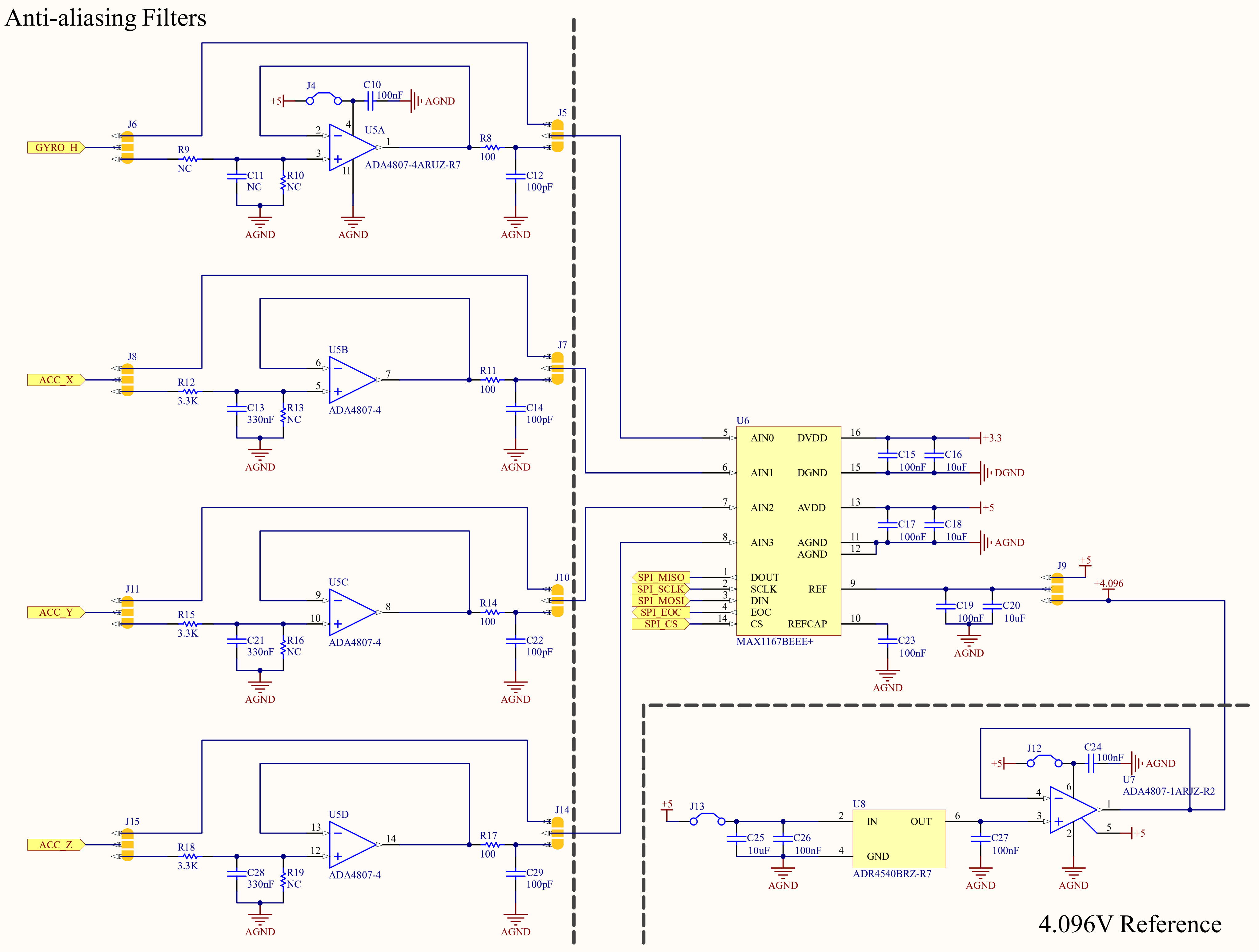
(2) Microcontroller

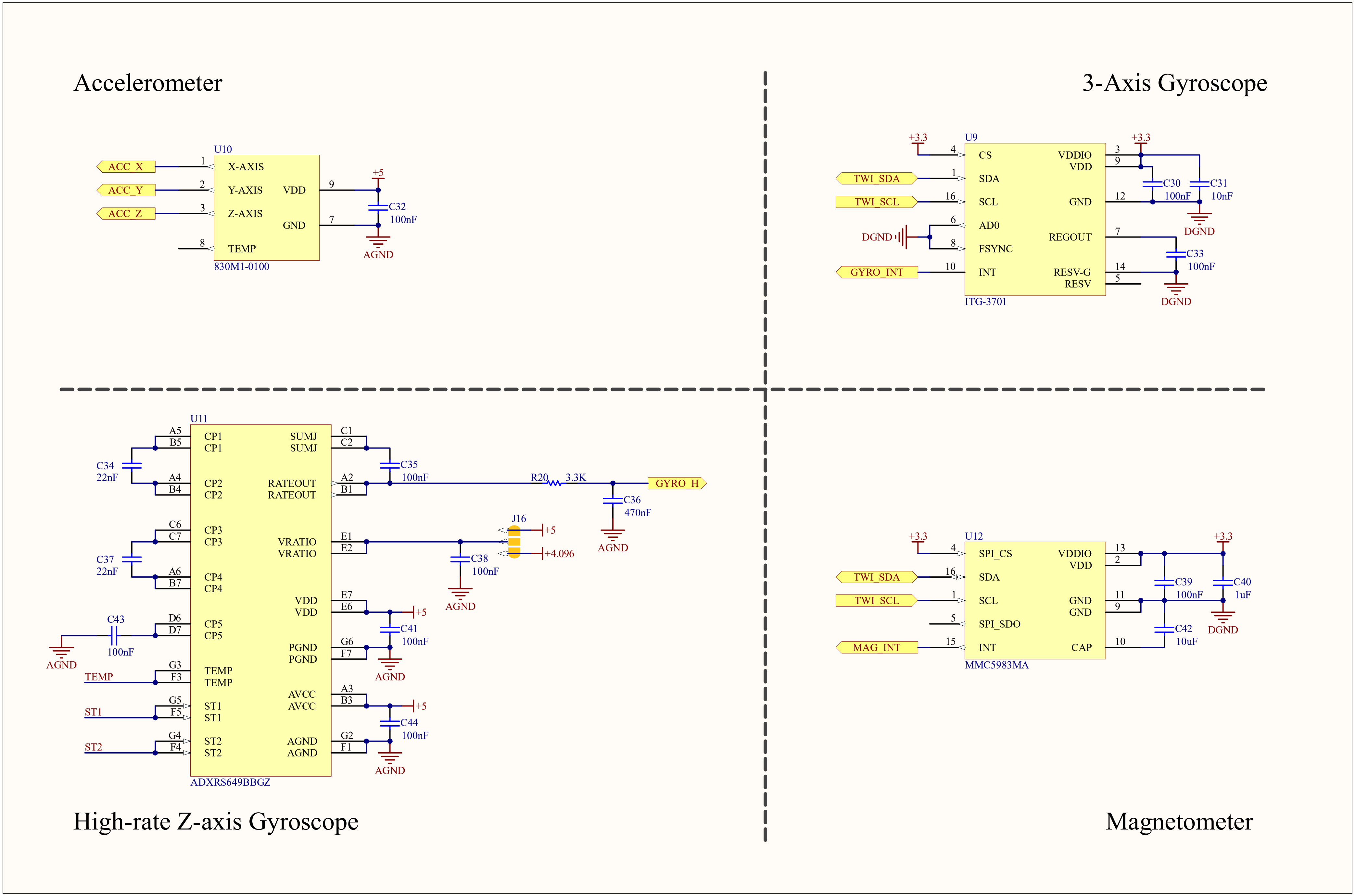
(3) ADC

(4) Sensors


P1
2
1
V_BAT
GND_BAT
C1
1uF
DGND
R1
100K
U1
IN
SHDN
GND
OUT
BYP
1
3
2
5
4
LTC1844ES5-3.3#TRMPBF
DGND
C3
100nF
DGND
+3.3
C2
1uF
DGND
3.3V Supply
5V Supply
U2
IN
SHDN
GND
OUT
C+
C-
5
3
2
1
6
4
LTC3200ES6-5#TRMPBF
C4
4.7uF
AGND
J1
A_SHDN
AGND
C6
1uF
+5
C5
4.7uF
AGND
GND_BAT
AGND
FB1
120
DGND
AGND


Reed Switch
S1
R6
NC
+3.3
Reed Switch
DGND
Battery Voltage
Battery Voltage Indicator
J2
+5
C7
100nF
AGND
U3
ADA4807-1ARJZ-R2
1
2
3
4
5
6
+5
R2
270K
J3
V_BAT
R4
680K
AGND
AGND
Indication LED
+3.3
R3
1K
LED1
Q1
R5
1K
LED
DGND
U4
ENW-89857A1KF
SPI_MISO
TWI_SCL
TWI_SDA
MAG_INT
SPI_MOSI
SPI_CS
A_SHDN
SPI_EOC
SPI_SCLK
GYRO_INT
SPI_MISO F5 P0.02
A2 AIN1
TWI_SCL B2 P0.04
TWI_SDA C3 P0.05
UART_TXD F7 UART_TXD
MAG_INT B6 P0.07
UART_RXD E6 UART_RXD
SPI_MOSI E1 P0.14
SPI_CS C6 P0.15
A_SHDN A8 P0.16
SPI_EOC F6 P0.17
SPI_SCLK E2 P0.20
F8 P0.28
GYRO_INT B1 P0.29
B5 P0.30
nRESET A3 nRESET
SWDIO C4 SWDIO
SWDCLK C5 SWDCLK
F3 DP
F4 DM
VBUS E3
VCC A5
VDDH A6
GND A1
GND A7
GND A9
GND A11
GND A12
GND C8
GND C9
GND D7
GND D8
GND D9
GND E8
GND E9
GND F1
GND F9
GND F11
GND F12
+3.3
C8
100nF
C9
4.7uF
DGND
Debugging Parts
P2
1
2
3
UART_TXD
UART_RXD
DGND
P3
1
2
3
4
DGND
S2
R7
NC
+3.3
DGND


Anti-aliasing Filters
GYRO_H
J6
R9
NC
C11
NC
R10
NC
AGND
J4
+5
C10
100nF
AGND
U5A
ADA4807-4ARUZ-R7
2
3
4
11
1
AGND
R8
100
C12
100pF
AGND
J5
ACC_X
J8
R12
3.3K
C13
330nF
R13
NC
AGND
U5B
6
5
7
ADA4807-4
R11
100
C14
100pF
AGND
J7
ACC_Y
J11
R15
3.3K
C21
330nF
R16
NC
AGND
U5C
9
10
8
ADA4807-4
R14
100
C22
100pF
AGND
J10
ACC_Z
J15
R18
3.3K
C28
330nF
R19
NC
AGND
U5D
13
12
14
ADA4807-4
R17
100
C29
100pF
AGND
J14
U6
AIN0
AIN1
AIN2
AIN3
DOUT
SCLK
DIN
EOC
CS
DVDD
DGND
AVDD
AGND
AGND
REF
REFCAP
MAX1167BEEE+
5
6
7
8
1
2
3
4
14
16
15
13
11
12
9
10
SPI_MISO
SPI_SCLK
SPI_MOSI
SPI_EOC
SPI_CS
C15
100nF
C16
10uF
+3.3
DGND
C17
100nF
C18
10uF
+5
AGND
C19
100nF
C20
10uF
AGND
C23
100nF
AGND
J9
+5
+4.096
+5
J13
C25
10uF
C26
100nF
AGND
U8
IN
OUT
GND
2
6
4
ADR4540BRZ-R7
C27
100nF
AGND
J12
+5
C24
100nF
AGND
U7
ADA4807-1ARJZ-R2
6
4
3
2
1
5
+5
AGND
4.096V Reference


Accelerometer
3-Axis Gyroscope
High-rate Z-axis Gyroscope
Magnetometer
U10
ACC_X
ACC_Y
ACC_Z
X-AXIS
Y-AXIS
Z-AXIS
TEMP
VDD
GND
830M1-0100
+5
C32
100nF
AGND
U9
+3.3
CS
SDA
SCL
AD0
FSYNC
INT
VDDIO
VDD
GND
REGOUT
RESV-G
RESV
ITG-3701
TWI_SDA
TWI_SCL
DGND
GYRO_INT
C30
100nF
C31
10nF
C33
100nF
U11
CP1
CP2
CP3
CP4
CP5
TEMP
ST1
ST2
SUMJ
RATEOUT
VRATIO
VDD
PGND
AVCC
AGND
ADXRS649BBGZ
A5
B5
A4
B4
C6
C7
A6
B7
D6
D7
G3
F3
G5
F5
G4
F4
C1
C2
A2
B1
E1
E2
E7
E6
G6
F7
A3
B3
G2
F1
C34
22nF
C37
22nF
C43
100nF
C35
100nF
R20
3.3K
C36
470nF
GYRO_H
J16
+4.096
C38
100nF
C41
100nF
C44
100nF
U12
SPI_CS
SDA
SCL
SPI_SDO
INT
VDDIO
VDD
GND
CAP
MMC5983MA
MAG_INT
C39
100nF
C40
1uF
C42
10uF

Appendix B

PCB Design

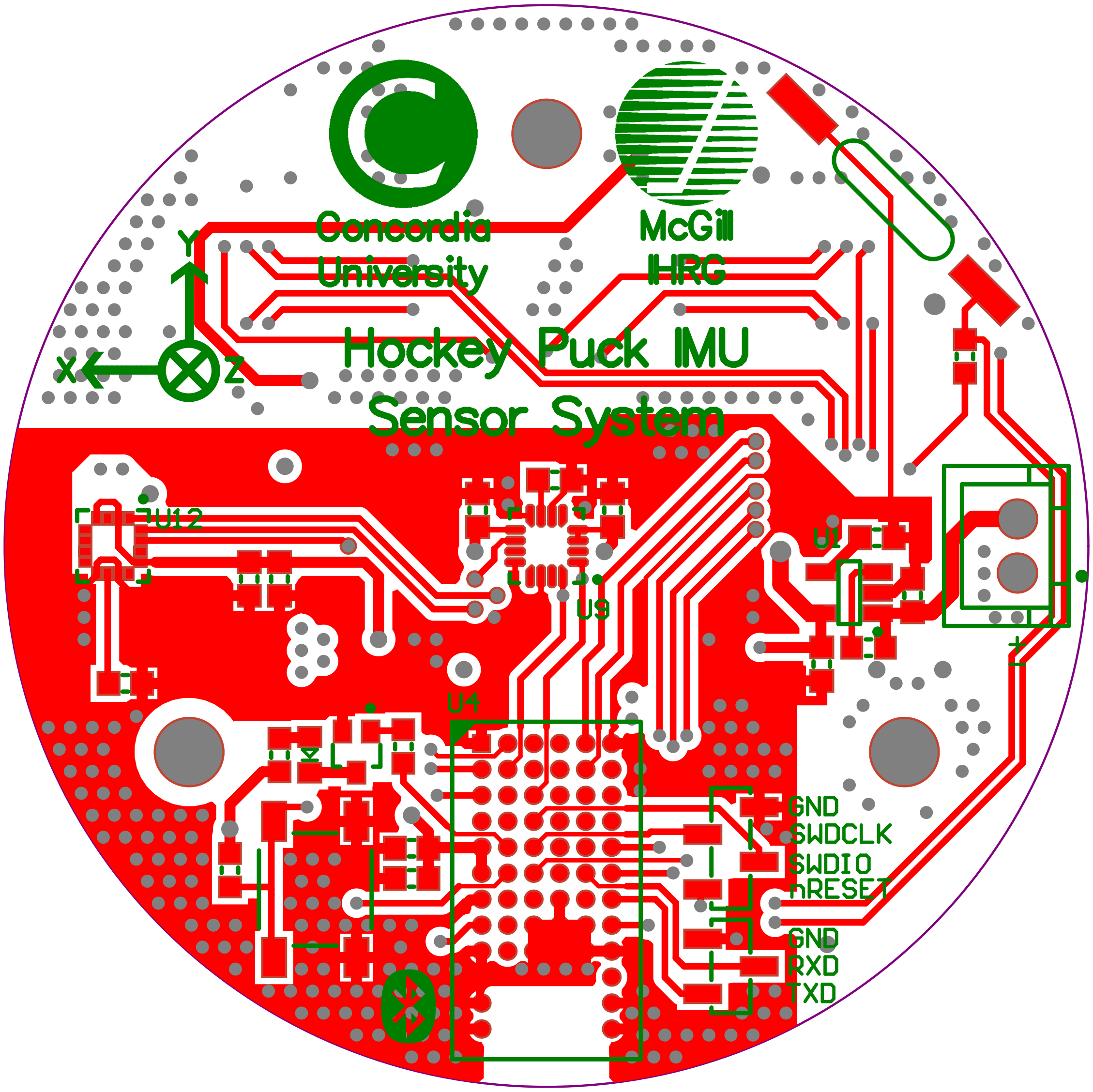
The PCB layout for the hockey puck sensor system was designed in the Altium Designer environment. The design comprises a multi-layer architecture distributed as follows:

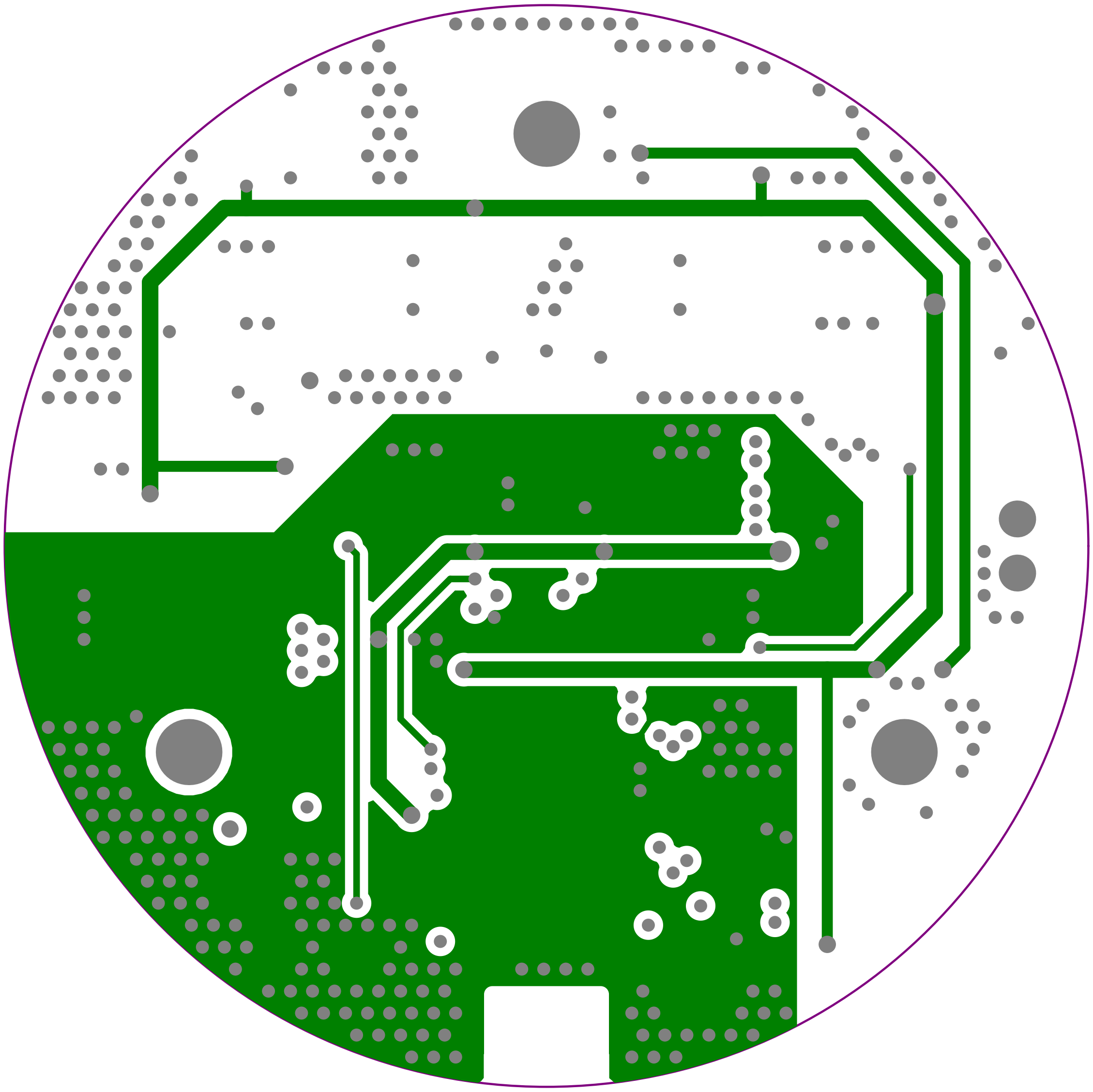
(1) Top Layer

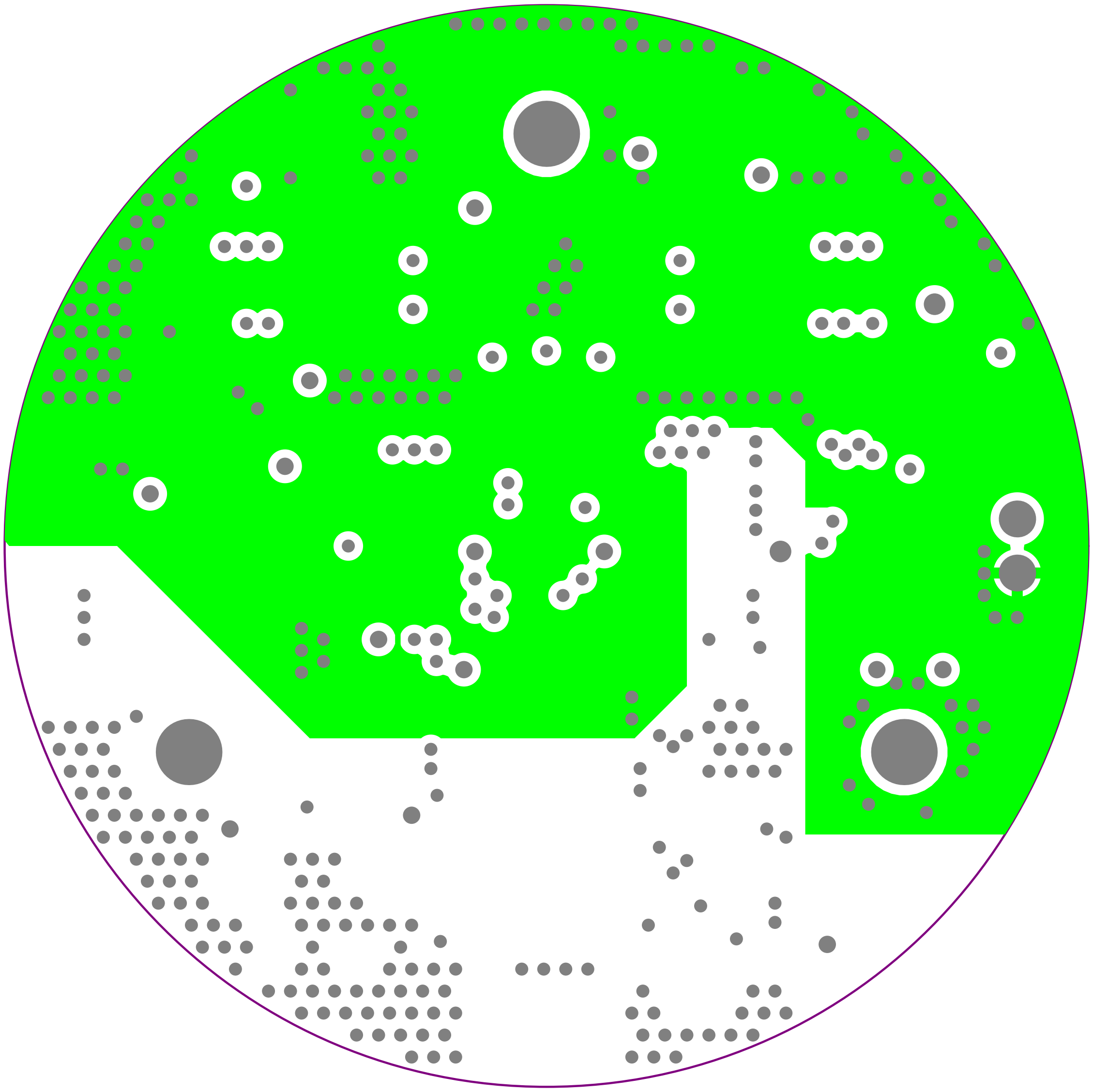
(2) Power Layer

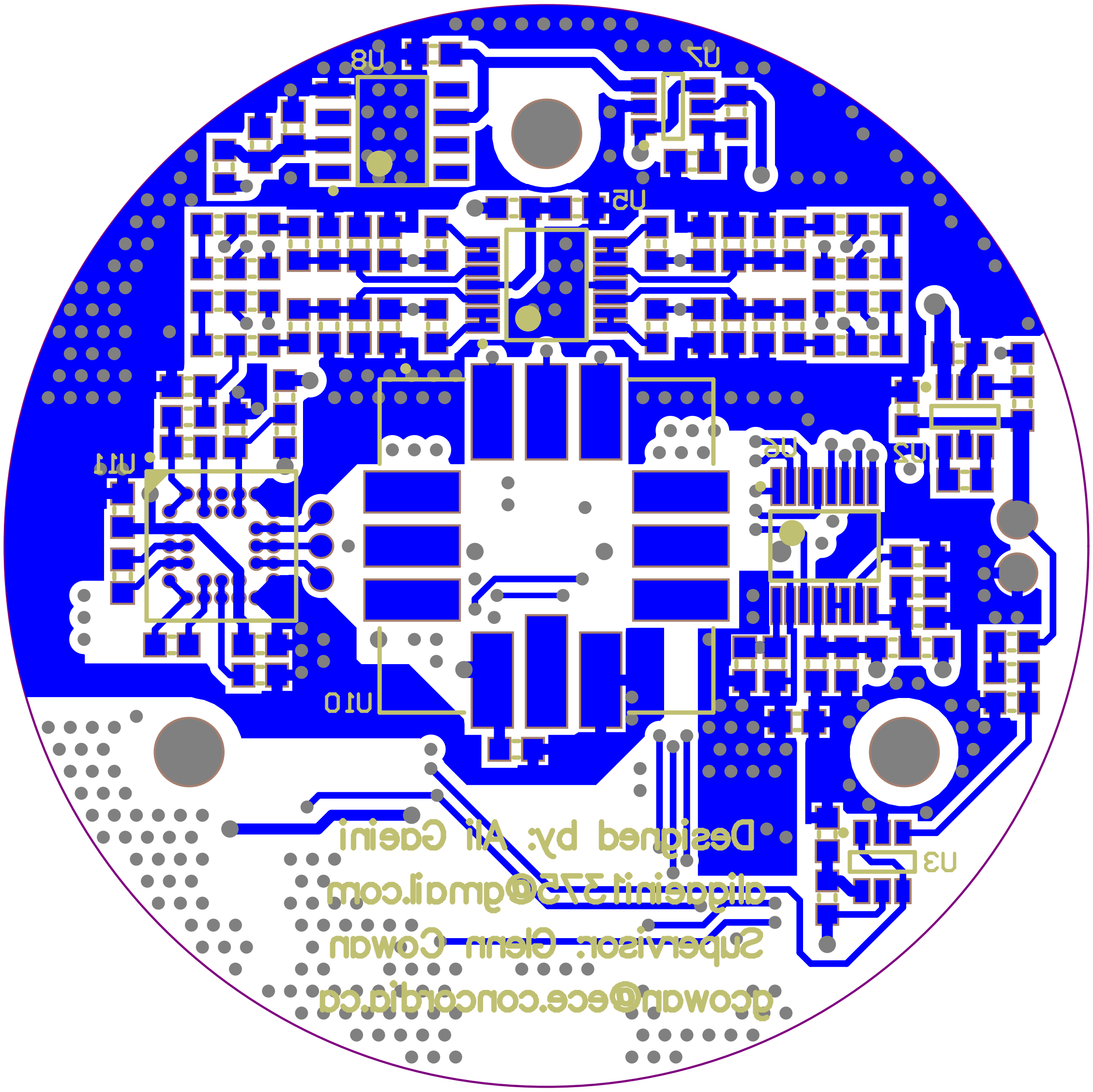
(3) GND Layer

(4) Bottom Layer


Concordia
University
McGill
IHRG
Y
X
Z
Hockey Puck IMU
Sensor System
U12
U9
U1
U4
+
GND
SWDCLK
SWDIO
nRESET
GND
RXD
TXD






U8
U7
U5
U11
U10
U6
U2
U3
Designed by: Ali Gaeini
aligaeini1375@gmail.com
Supervisor: Glenn Cowan
gcowan@ece.concordia.ca

Appendix C

Microcontroller Source Code

The resources provided by Nordic Semiconductor profoundly influenced the development of the BLE functionality within the sensor system. The `ble_app_uart` example code from the nRF5 SDK tailored for the `pca10100e` board served as the template for development. The microcontroller code development is structured into three distinct sections:

(1) main

(2) BLE custom service

(3) sensors

For a complete source code overview, refer to the corresponding GitHub repository as indicated in [63]. Below is a snippet of the loop within the “main” section.

```
// Enter the main loop.
for (;;) {
    if (m_data_flag == true
    && m_conn_handle != BLE_CONN_HANDLE_INVALID) {

        (void) ReadAcc(m_acc_value);
        (void) ReadHighGyro(m_high_gyro_value);
        (void) (ITG3701_ReadGyro(m_gyro_value));
        (void) (MMC5983MA_ReadMag(m_mag_value));

```

```
        for (uint8_t i = 0; i < (sizeof(m_acc_value) / 2); i++)
        {
          m_data[index++] = m_acc_value[i] >> 8;
          m_data[index++] = m_acc_value[i] & 0xFF;
        }
        for (uint8_t i = 0; i < sizeof(m_gyro_value); i++)
        {
          m_data[index++] = m_gyro_value[i];
        }
        for (uint8_t i = 0; i < (sizeof(m_high_gyro_value) / 2); i++)
        {
          m_data[index++] = m_high_gyro_value[i] >> 8;
          m_data[index++] = m_high_gyro_value[i] & 0xFF;
        }
        for (uint8_t i = 0; i < sizeof(m_mag_value); i++)
        {
          m_data[index++] = m_mag_value[i];
        }

    m_data_flag = false;
    m_standby_counter = 0;
    }

    if (m_ble_flag == true && m_conn_handle != BLE_CONN_HANDLE_INVALID) {
        m_data[index++] = m_counter >> 8;
        m_data[index] = m_counter & 0xFF;
        m_counter++;
        index = 0;
        value_update();
        m_ble_flag = false;
    }

    if (m_standby_counter == 60) {
        m_standby_counter = 0;

```

```
        (void) itg3701_goto_sleep();
        (void) mmc5983ma_goto_sleep();
        nrf_gpio_pin_clear(16);
        nrf_delay_ms(500);

        nrf_pwr_mgmt_shutdown(NRF_PWR_MGMT_SHUTDOWN_GOTO_SYSOFF);
    }

    idle_state_handle();
}
```

Appendix D

Data Acquisition Source Code

The subsequent data acquisition code for the sensor system has been developed within the MATLAB® App Designer environment. This representation only includes the core functions and callbacks. For a comprehensive view of the source code, refer to the designated GitHub repository, as cited in [63].

```
classdef PuckSensorApp < matlab.apps.AppBase

    % Properties that correspond to app components
    properties (Access = public) ...

    properties (Access = private) ...

    methods (Access = private)

        function readCharacteristicData(app, src, ~)
            d = read(src, 'oldest');
            app.Data = [app.Data; d];
            processCounter(app, d);
            d = reshape(d(1:210), [21, 10])';
            processAccel(app, d);
            processGyro(app, d);
            processHighGyro(app, d);
            processMag(app, d);
            app.Time = [app.Time; datetime('now', ...
```

```
                'Format', 'dd-MMM-uuuu HH:mm:ss.SSS')];
        end

        function processCounter(app, data)
            c = double(swapbytes(typecast(uint8(data(211:212)), 'uint16')));
            app.Counter = [app.Counter; c];
        end

        function processAccel(app, data)
            a = swapbytes(typecast(reshape( ...
                uint8(data(:, 1:6)'), 1, []), 'uint16''));
            a = reshape(a, [3, 10])';
            a = double(a) * 4096 / 65535;
            switch app.DeviceName
                case 'Puck Sensor 1'
                    a = a ./ [12.279 12.037 13.999];
                case 'Puck Sensor 2'
                    a = a ./ [12.130 12.555 14.329];
            end
            a = a * 9.81;
            app.Accel = [app.Accel; ...
                NaN([(app.Counter(end)-app.Counter(end-1)-1)*10 3]); a];
        end

        function processGyro(app, data)
            g = swapbytes(typecast(reshape( ...
                uint8(data(:, 7:12)'), 1, []), 'int16'));
            g = reshape(g, [3, 10])';
            g = double(g) * 4000 / 32768;
            g = deg2rad(g);
            app.Gyro = [app.Gyro; ...
                NaN([(app.Counter(end)-app.Counter(end-1)-1)*10 3]); g];
        end

        function processHighGyro(app, data)
```

```
            hg = swapbytes(typecast(reshape( ...
            uint8(data(:, 13:14)’), 1, []), ’uint16’));
            hg = reshape(hg, [1, 10])’;
            hg = double(hg) * 4096 / 65535;
            hg = hg / 0.0796;
            hg = deg2rad(hg);
            app.HighGyro = [app.HighGyro; ...
                NaN([(app.Counter(end)-app.Counter(end-1)-1)*10 1]); hg];
        end

        function processMag(app, data)
            m = swapbytes(typecast(reshape( ...
                uint8(data(:, 15:20)’), 1, []), ’uint16’));
            m = reshape(m, [3, 10])’;
            m = uint32(m);
            m(:, 1) = bitor(bitsll(m(:,1),2), ...
                uint32(bitsrl(bitand(uint8(data(:, 21)), 0xC0), 6)));
            m(:, 2) = bitor(bitsll(m(:,2),2), ...
                uint32(bitsrl(bitand(uint8(data(:, 21)), 0x30), 4)));
            m(:, 3) = bitor(bitsll(m(:,3),2), ...
                uint32(bitsrl(bitand(uint8(data(:, 21)), 0x0C), 2)));
            m = double(m) - 131072;
            m = m / 16384;
            m = m * 100;
            app.Mag = [app.Mag; ...
                NaN([(app.Counter(end)-app.Counter(end-1)-1)*10 3]); m];
        end

        function estimateVelocity(app)
            app.Vel = [];
            app.Ornt = quaternion();
            fuse = insfilterMARG;
            fuse.IMUSampleRate = 400;
            fuse.StateCovariance = 1e-12 * eye(22);

```

```
        idx = find(abs(app.FinalGyro(:, 3)) > 69);
        if (length(idx) >= 5)
            temp = app.FinalGyro(:, 3);
            app.FinalGyro(:, 3) = app.FinalHighGyro;
        end

        for ii = 1:size(app.FinalAccel, 1)
            fuse.predict(app.FinalAccel(ii, :), app.FinalGyro(ii, :));
            if ˜mod(ii, 200)
                fuse.fusemag(app.FinalMag(ii, :), ...
                    cov(app.FinalMag(1:100, :)));
            end
            [˜, app.Ornt(ii), app.Vel(ii, :)] = pose(fuse);
        end
        app.FinalGyro(:, 3) = temp;

        for ii = 1:size(app.Vel, 1)
            app.Vel(ii, 4) = norm(app.Vel(ii, 1:3));
        end
    end

    function plotData(app)
        t = 0:2.5e-3:(size(app.Vel, 1)-1)/400;
        plot(app.UIAxes, t, app.Vel(:, 1:3))
        legend(app.UIAxes, {’X Axis’, ’Y Axis’, ’Z Axis’})
        plot(app.UIAxes2, t, app.Vel(:, 4))
        text(app.UIAxes2, ’String’, sprintf(...
            ’Maximum Estimated Velocity: %.3f m/s’, max(app.Vel(:,4))), ...
            ’Units’, ’normalized’, ’Position’, [0.025 0.9], ...
            ’VerticalAlignment’,’top’, ’EdgeColor’,’k’, ...
            ’BackgroundColor’,’w’);
        plot(app.UIAxes3, t, rad2deg(quat2eul(app.Ornt)))
        legend(app.UIAxes3, {’X Axis’, ’Y Axis’, ’Z Axis’})
    end
end
```

```
    % Callbacks that handle component events
    methods (Access = private)

        % Code that executes after component creation
        function startupFcn(app)
            app.Label.Text = ’Waiting for Search...’;
        end

        % Button pushed function: SearchButton
        function SearchButtonPushed(app, event)
            device = [];
            app.Label.Text = ’Searching for’ + newline + ’the Puck Sensor...’;
            while (isempty(device))
                device = blelist(’Name’, ’Puck Sensor’);
            end
            app.DeviceName = device.Name;
            app.Label.Text = app.DeviceName + newline + ’detected’;
            app.ConnectButton.Enable = ’on’;
        end

        % Button pushed function: ConnectButton
        function ConnectButtonPushed(app, event)
            app.BLE = ble(app.DeviceName);
            app.BLEChar = characteristic(app.BLE, ...
                ’0A361000-115B-4E3F-80E6-05D7F5F2730E’, ...
                ’0A361001-115B-4E3F-80E6-05D7F5F2730E’);
            app.BLEChar.DataAvailableFcn = @(src, event) ...
                app.readCharacteristicData(src, event);
            app.Label.Text = ’Connected to’ + newline + app.DeviceName;
            app.SearchButton.Enable = ’off’;
            app.ConnectButton.Enable = ’off’;
            app.DisconnectButton.Enable = ’on’;
            app.StartButton.Enable = ’on’;
        end
```

```

        % Button pushed function: DisconnectButton
        function DisconnectButtonPushed(app, event)
            unsubscribe(app.BLEChar);
            disconnect(app.BLE);
            app.BLEChar = [];
            app.BLE = [];
            app.Time = [];
            app.Label.Text = 'Disconnected from' + newline + app.DeviceName;
            app.SearchButton.Enable = 'on';
            app.ConnectButton.Enable = 'on';
            app.DisconnectButton.Enable = 'off';
            app.StartButton.Enable = 'off';
        end

        % Button pushed function: StartButton
        function StartButtonPushed(app, event)
            app.Time = datetime('now', 'Format', 'dd-MMM-uuuu HH:mm:ss.SSS');
            app.Data = [];
            app.Counter = app.Counter(end);
            app.Accel = [];
            app.Gyro = [];
            app.HighGyro = [];
            app.Mag = [];
            app.Label.Text = 'Getting data from' + newline + app.DeviceName;
            app.StartButton.Enable = 'off';
            app.FinishButton.Enable = 'on';
        end

        % Button pushed function: FinishButton
        function FinishButtonPushed(app, event)
            app.FinalTime = app.Time;
            app.FinalData = app.Data;
            app.FinalCounter = app.Counter;
            app.FinalAccel = app.Accel - ...
```

```
                mean(app.Accel(1:100, :), 'omitnan') + [0 0 9.81];
            app.FinalGyro = (app.Gyro - ...
                mean(app.Gyro(1:100, :), 'omitnan')) .* [1 -1 -1];
            app.FinalHighGyro = (app.HighGyro - ...
                mean(app.HighGyro(1:100, :), 'omitnan')) * -1;
            app.FinalMag = app.Mag .* [1 -1 1];

            app.FinalAccel = fillmissing(app.FinalAccel, 'linear');
            app.FinalGyro = fillmissing(app.FinalGyro, 'linear');
            app.FinalHighGyro = fillmissing(app.FinalHighGyro, 'linear');
            app.FinalMag = fillmissing(app.FinalMag, 'linear');

            estimateVelocity(app);
            plotData(app);

            app.Label.Text = 'Elapsed time:' + newline + ...
                string(app.FinalTime(end) - app.FinalTime(1), 'mm:ss.SSS');
            app.StartButton.Enable = 'on';
            app.FinishButton.Enable = 'off';
            app.SaveButton.Enable = 'on';
        end

        % Button pushed function: SaveButton
        function SaveButtonPushed(app, event)
            TimeVar = duration(app.FinalTime - app.FinalTime(1), ...
            'Format', 'mm:ss.SSS'); %#ok
            CounterVar = app.FinalCounter - app.Counter(1); %#ok
            RawData = app.FinalData; %#ok
            Accelerometer = app.FinalAccel; %#ok
            Gyroscope = [app.FinalGyro app.FinalHighGyro]; %#ok
            Magnetometer = app.FinalMag; %#ok
            Velocity = app.Vel; %#ok
            Orientation = app.Ornt'; %#ok
            uisave({'TimeVar', 'CounterVar', 'RawData', 'Accelerometer', ...
                'Gyroscope', 'Magnetometer', 'Velocity', 'Orientation'})
```

```
        end

        % Button pushed function: ExitButton
        function ExitButtonPushed(app, event)
            delete(app)
        end
    end

    % Component initialization
    methods (Access = private) ...

    % App creation and deletion
    methods (Access = public) ...

end
```

Bibliography

[1] Hockey Canada, *Hockey Canada Playing Rules*, 2022. [Online]. Available: https://cdn.hockeycanada.ca/hockey-canada/Hockey-Programs/Officiating/Downloads/rulebook_casebook_e.pdf

[2] S. M. Robbins, P. J. Renaud, N. MacInnis, and D. J. Pearsall, "The relationship between trunk rotation and shot speed when performing ice hockey wrist shots," *Journal of Sports Sciences*, vol. 39, no. 9, pp. 1001–1009, Dec. 2020. [Online]. Available: https://doi.org/10.1080/02640414.2020.1853336

[3] International Ice Hockey Federation, "International Ice Hockey Federation 2022 Season Summary," 2022. [Online]. Available: https://blob.iihf.com/iihf-media/iihfmvc/media/downloads/annual%20report/seasonsummary2022.pdf

[4] D. Pearsall, R. Turcotte, and S. Murphy, "Biomechanics of ice hockey," *Exercise and Sport Science*, pp. 675–692, 01 2000.

[5] C. Popkin, B. Schulz, C. Park, T. Bottiglieri, and T. S. Lynch, "Evaluation, management and prevention of lower extremity youth ice hockey injuries," *Open Access Journal of Sports Medicine*, vol. Volume 7, pp. 167–176, Nov. 2016. [Online]. Available: https://doi.org/10.2147/oajsm.s118595

[6] C. Lee and J. Yoo, "The Effects of a Rehabilitation Program for Ice Hockey Players with Shoulder Injury," *Journal of Physical Therapy Science*, vol. 25, no. 2, pp. 177–181, 2013. [Online]. Available: https://doi.org/10.1589/jpts.25.177

[7] K. Vats, M. Fani, D. A. Clausi, and J. Zelek, "Puck localization and multi-task event recognition in broadcast hockey videos," 2021. [Online]. Available: https://arxiv.org/abs/2105.10563

[8] L. Zhou, E. Fischer, C. Tunca, C. M. Brahms, C. Ersoy, U. Granacher, and B. Arnrich, "How We Found Our IMU: Guidelines to IMU Selection and a Comparison of Seven IMUs for Pervasive Healthcare Applications," *Sensors*, vol. 20, no. 15, p. 4090, Jul. 2020. [Online]. Available: https://doi.org/10.3390/s20154090

[9] R. F. Tinder, *Relativistic Flight Mechanics and Space Travel.* Springer International Publishing, 2007. [Online]. Available: https://doi.org/10.1007/978-3-031-79297-7

[10] G. Krishnan, C. Kshirsagar, G. Ananthasuresh, and N. Bhat, "MicromachinedHigh-Resolution Accelerometers," *Journal of the Indian Institute of Science*, vol. 87, 07 2007.

[11] V. M. N. Passaro, A. Cuccovillo, L. Vaiani, M. D. Carlo, and C. E. Campanella, "Gyroscope Technology and Applications: A Review in the Industrial Perspective," *Sensors*, vol. 17, no. 10, p. 2284, Oct. 2017. [Online]. Available: https://doi.org/10.3390/s17102284

[12] W. A. Gill, I. Howard, I. Mazhar, and K. McKee, "A Review of MEMS Vibrating Gyroscopes and Their Reliability Issues in Harsh Environments," *Sensors*, vol. 22, no. 19, p. 7405, Sep. 2022. [Online]. Available: https://doi.org/10.3390/s22197405

[13] A. Persson, "How Do We Understand the Coriolis Force?" *Bulletin of the American Meteorological Society*, vol. 79, no. 7, pp. 1373–1385, Jul. 1998. [Online]. Available: https://doi.org/10.1175/1520-0477(1998)079⟨1373:hdwutc⟩2.0.co;2

[14] M. Looney, "Designing for Low Noise Feedback Control with MEMS Gyroscopes," 2016. [Online]. Available: https://www.analog.com/en/analog-dialogue/articles/low-noise-feedback-control.html#:~:text=All%20MEMS%20gyroscopes%20have%20noise,(no%20vibration%20or%20shock).

[15] X. Ding, J. Jia, Y. Gao, and H. Li, "Mechanical and Electrical Noise in Sense Channel of MEMS Vibratory Gyroscopes," *Sensors*, vol. 17, no. 10, p. 2306, Oct. 2017. [Online]. Available: https://doi.org/10.3390/s17102306

[16] M. A. Khan, J. Sun, B. Li, A. Przybysz, and J. Kosel, "Magnetic sensors-A review and recent technologies," *Engineering Research Express*, vol. 3, no. 2, p. 022005, Jun. 2021. [Online]. Available: https://doi.org/10.1088/2631-8695/ac0838

[17] J. Lenz and S. Edelstein, "Magnetic sensors and their applications," *IEEE Sensors Journal*, vol. 6, no. 3, pp. 631–649, Jun. 2006. [Online]. Available: https://doi.org/10.1109/jsen.2006.874493

[18] P. Ripka, "Improving the Accuracy of Magnetic Sensors," in *Lecture Notes Electrical Engineering*. Springer Berlin Heidelberg, pp. 45–60. [Online]. Available: https://doi.org/10.1007/978-3-540-69033-7_3

[19] A. Grosz, V. Mor, S. Amrusi, I. Faivinov, E. Paperno, and L. Klein, "A High-Resolution Planar Hall Effect Magnetometer for Ultra-Low Frequencies," *IEEE Sensors Journal*, vol. 16, no. 9, pp. 3224–3230, May 2016. [Online]. Available: https://doi.org/10.1109/jsen.2016.2523463

[20] T. Lapucci, L. Troiano, C. Carobbi, and L. Capineri, "Soft and Hard Iron Compensation for the Compasses of an Operational Towed Hydrophone Array without Sensor Motion by a Helmholtz Coil," *Sensors*, vol. 21, no. 23, p. 8104, Dec. 2021. [Online]. Available: https://doi.org/10.3390/s21238104

[21] MATLAB, "magcal: Magnetometer calibration coefficients." [Online]. Available: https://www.mathworks.com/help/nav/ref/magcal.html.

[22] R. Campa and H. de la Torre, "Pose control of robot manipulators using different orientation representations: A comparative review," in *2009 American Control Conference*. IEEE, 2009. [Online]. Available: https://doi.org/10.1109/acc.2009.5160254

[23] M. Kok, J. D. Hol, and T. B. Schön, "Using Inertial Sensors for Position and Orientation Estimation," *Foundations and Trends® in Signal Processing*, vol. 11, no. 1-2, pp. 1–153, 2017. [Online]. Available: https://doi.org/10.1561/2000000094

[24] J. J. Craig, *Introduction to robotics*, 3rd ed. Upper Saddle River, NJ: Pearson, Sep. 2003.

[25] Y.-B. Jia, "Quaternions and Rotations," vol. 477, 2008. [Online]. Available: http://www.fuchs-braun.com/media/56347a973e30256ffff802effffffff1.pdf

[26] MATLAB, "quaternion: Create a quaternion array." [Online]. Available: https://www.mathworks.com/help/robotics/ref/quaternion.html.

[27] ——, "quatrotate: Rotate vector by quaternion." [Online]. Available: https://www.mathworks.com/help/aerotbx/ug/quatrotate.html.

[28] P. S. Maybeck, *Stochastic models: V. 1: Estimation and control*. San Diego, CA: Academic Press, 1979.

[29] G. Welch and G. Bishop, "An introduction to the Kalman filter," 2006. [Online]. Available: https://www.cs.unc.edu/~welch/media/pdf/kalman_intro.pdf

[30] E. Wan and R. V. D. Merwe, "The unscented Kalman filter for nonlinear estimation," in *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373)*. IEEE. [Online]. Available: https://doi.org/10.1109/asspcc.2000.882463

[31] D. Roetenberg, H. Luinge, C. Baten, and P. Veltink, "Compensation of magnetic disturbances improves inertial and magnetic sensing of human body segment orientation," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 13, no. 3, pp. 395–405, Sep. 2005. [Online]. Available: https://doi.org/10.1109/tnsre.2005.847353

[32] MATLAB, "Inertial Sensor Fusion: IMU and GPS sensor fusion to determine orientation and position." [Online]. Available: https://www.mathworks.com/help/fusion/inertial-sensor-fusion.html.

[33] MEMS Industry Group, “Open-Source-Sensor-Fusion,” 2015. [Online]. Available: https://github.com/memsindustrygroup/Open-Source-Sensor-Fusion.

[34] MATLAB, “imufilter: Orientation from accelerometer and gyroscope readings.” [Online]. Available: https://www.mathworks.com/help/nav/ref/imufilter-system-object.html.

[35] ——, “insfilterMARG: Estimate pose from MARG and GPS data.” [Online]. Available: https://www.mathworks.com/help/nav/ref/insfiltermarg.html.

[36] V. T. van Hees, L. Gorzelniak, E. C. D. León, M. Eder, M. Pias, S. Taherian, U. Ekelund, F. Renström, P. W. Franks, A. Horsch, and S. Brage, “Separating Movement and Gravity Components in an Acceleration Signal and Implications for the Assessment of Human Daily Physical Activity,” *PLoS ONE*, vol. 8, no. 4, p. e61691, Apr. 2013. [Online]. Available: https://doi.org/10.1371/journal.pone.0061691

[37] A. Villaseñor, R. Turcotte, and D. Pearsall, “Recoil Effect of the Ice Hockey Stick during a Slap Shot,” *Journal of Applied Biomechanics*, vol. 22, no. 3, pp. 202–211, Aug. 2006. [Online]. Available: https://doi.org/10.1123/jab.22.3.202

[38] MATLAB, “trapz: Trapezoidal numerical integration.” [Online]. Available: https://www.mathworks.com/help/matlab/ref/trapz.html.

[39] E. Chin, D. Hine, L. Kyropoulos, and M. West, “Performance Tracking Hockey Puck,” 2019, Capstone project.

[40] Motsai, “Neblina™ Bluetooth® Wireless Motion Sensor.” [Online]. Available: https://motsai.com/solutions/neblina.

[41] Movesense, “Movesense Sensor HR+.” [Online]. Available: https://www.movesense.com/wp-content/uploads/2021/08/Movesense-Sensor-HRplus-Spec-Sheet-08-2021.pdf.

[42] Y. Michaud-Paquette, D. J. Pearsall, and R. A. Turcotte, “Predictors of scoring accuracy: ice hockey wrist shot mechanics,” *Sports Engineering*, vol. 11, no. 2, pp. 75–84, Oct. 2008. [Online]. Available: https://doi.org/10.1007/s12283-008-0009-9

[43] H.-P. Huang, F. Hsu, and A. Ahn, “MEMS vs. Piezoelectric Accelerometer.” [Online]. Available: https://www.labfront.com/article/mems-vs-piezoelectric-accelerometer#:~:text=In%20general%2C%20PA%20is%20more,not%20along%20the%20gravity%20axis..

[44] TE Connectivity, “830M1 Triaxial Condition Monitoring Accelerometer.” [Online]. Available: https://www.te.com/commerce/DocumentDelivery/DDEController?Action=showdoc&DocId=Data+Sheet%7F830M1_Triaxial_Accelerometer%7FA4%7Fpdf%7FEnglish%7FENG_DS_830M1_Triaxial_Accelerometer_A4.pdf%7F20011424-01.

[45] TDK InvenSense, “ITG-3701.” [Online]. Available: https://product.tdk.com/system/files/dam/doc/product/sensor/mortion-inertial/gyro/data_sheet/itg-3701-datasheet.pdf.

[46] Analog Devices, “ADXRS649.” [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/ADXRS649.pdf.

[47] MEMSIC, “MMC5983MA.” [Online]. Available: https://www.memsic.com/Public/Uploads/uploadfile/files/20220119/MMC5983MADatasheetRevA.pdf.

[48] Analog Devices, “MAX1167.” [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/MAX1167-MAX1168.pdf.

[49] Texas Instruments, “The Bypass Capacitor in High-Speed Environments.” [Online]. Available: https://www.ti.com/lit/an/scba007a/scba007a.pdf.

[50] A. V. Oppenheim and A. S. Willsky, *Signals and Systems*, 2nd ed. Upper Saddle River, NJ: Pearson, Aug. 1996.

[51] Analog Devices, “ADA4807.” [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/ADA4807-1_4807-2_4807-4.pdf.

[52] ——, “ADR4540.” [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/adr4520_4525_4530_4533_4540_4550.pdf.

[53] Panasonic Industries, “PAN1781 (nRF52820).” [Online]. Available: https://mediap.industry.panasonic.eu/assets/custom-upload/Devices/Wireless%20Connectivity/Bluetooth%20Low%20Energy%20Modules/PAN1781/WM%20PAN1781%20Product%20Specification.pdf.

[54] Nordic Semiconductor, “nRF52820 Product Specification.” [Online]. Available: https://infocenter.nordicsemi.com/pdf/nRF52820_PS_v1.3.pdf.

[55] C. Gomez, J. Oller, and J. Paradells, “Overview and Evaluation of Bluetooth Low Energy: An Emerging Low-Power Wireless Technology,” *Sensors*, vol. 12, no. 9, pp. 11 734–11 753, Aug. 2012. [Online]. Available: https://doi.org/10.3390/s120911734

[56] B. A. Spockeli, “nRF5x-custom-ble-service-tutorial.” [Online]. Available: https://github.com/NordicPlayground/nRF5x-custom-ble-service-tutorial.

[57] Nordic Semiconductor, “S140 SoftDevice.” [Online]. Available: https://infocenter.nordicsemi.com/pdf/S140_SDS_v2.1.pdf.

[58] Analog Devices, “LTC1844.” [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/1844fa.pdf.

[59] ——, “LTC3200ES6-5.” [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/32005f.pdf.

[60] M. D. (https://electronics.stackexchange.com/users/150007/mark davis), “Does a gyro need to be placed at the center of rotation?” Electrical Engineering Stack Exchange, uRL:https://electronics.stackexchange.com/q/307004 (version: 2017-05-23). [Online]. Available: https://electronics.stackexchange.com/q/307004

[61] H. Zumbahlen, “Staying Well Grounded.” [Online]. Available: https://www.analog.com/en/analog-dialogue/articles/staying-well-grounded.html.

[62] B. Mathiszik, “X-Ray Inspection – Lead-Free Solder Drives X-Ray Inspection.” [Online]. Available: https://www.electronicdesign.com/technologies/test-measurement/article/21200935/xray-inspection-leadfree-solder-drives-xray-inspection.

[63] A. Gaeini, “IMU-Sensor-System-for-Hockey-Puck,” 2023. [Online]. Available: https://github.com/ali-gaeini/IMU-Sensor-System-for-Hockey-Puck.