C02018: Doctor of Philosophy

CRICOS Code: 036570B

49986 PhD Thesis: Engineering

May 2022

A Study on

Model estimation for health monitoring and rehabilitation systems

Li Wang

School of Biomedical Engineering

Faculty of Engineering and Information Technology

University of Technology Sydney

NSW - 2007, Australia

A Study on

Model estimation for health monitoring and rehabilitation systems

A thesis submitted in partial fulfilment of the requirements
for the degree of

Doctor of Philosophy
in
Engineering

by

Li Wang

to

School of Biomedical Engineering
Faculty of Engineering and Information Technology
University of Technology Sydney
NSW - 2007, Australia

May 2022

AUTHOR'S DECLARATION

I, *Li Wang* declare that this thesis, is submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the *School of Biomedical Engineering, Faculty of Engineering and Information Technology* at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

SIGNATURE: _____

[Your Name]

DATE: 03rd May, 2022

PLACE: Sydney, Australia

Abstract

Electronic trainers (e-trainers) are fitness guidance systems consisting of motion signal sensor(s), a user interface, and a control system. Owing to the widespread popularity of fitness and personal training, e-trainers have found numerous applications across many fields. However, the design of e-trainers is challenging because of their requirement for miniaturisation and problems with discrepancies, drift, lack of data, and limited resources. The primary aim of this thesis is to design an improved e-trainer with a focus on the initial measurement unit calibration algorithm and the practical implementation of pattern recognition algorithms. Several problems in the field are considered, including kernel-based heart rate regulation, practical considerations for the calibration of efficient wearable devices, and model compression using the pruning method.

The first part of this thesis investigates several practical issues associated with calibrating the proposed low-cost wearable e-trainer in clinical settings, including poor repeatability and significant volatility. In field-based environments, the parameter variation of the low-cost triaxial gyroscope requires an effective and practical calibration process to reduce the errors due to unexpected variance. To this end, an efficient in-field calibration method is developed that can readily calibrate the triaxial gyroscope without additional equipment. This experimental scheme can be easily implemented by manually rotating the triaxial gyroscope over a certain angle as the calibration reference. A linearised calibration model is developed for the proposed experimental scheme, and G-optimality is achieved. Extensive numerical simulations demonstrate that the calibration error is relatively low and the estimation of model parameters is unbiased under mild experimental conditions. After a calibration process taking less than 30 s, the absolute error of the scale factors is always less than 2.5×10^{-2} for LSM9DS1 and that of the biases is less than 1×10^{-2} for ICM20948.

In the second part of this thesis, to overcome the lack of suitable training data for modelling the human cardiovascular response, the simulation and control of the human heart rate are investigated in detail using a kernel-based nonparametric model with model predictive control. This kernel-based method introduces a kernel regularisation term that provides prior information to the model estimation phase. By adding this prior information, the experimental protocol can be significantly simplified, with a model training time of only 10 min. Based on the identified model, a controller that uses model predictive control is designed to track a predefined reference heart rate profile. One

advantage of this approach is that the speed and acceleration of the treadmill can be maintained within a safe range for vulnerable exercisers. The entire model construction process takes 10 min, including an 80-s resting period. The protocol is relatively simple and consists of only two accelerations. In the heart rate tracking task, the heart rates of 12 experiment participants follow the target heart rate to within ±3 beats per second.

The third part of this thesis leverages the state-of-the-art neural network pruning method to compress the network model. This allows the computational complexity of the inference task to be reduced by 98% without significant performance degradation. It is therefore possible to use advanced deep learning models to estimate human motion states on embedded systems with limited resources. For the user, more neural network models operating on the device means that more functions can be provided. Experimental results verify the effectiveness and efficiency of the proposed method, with up to 60% of graph links and 98% of network weights pruned across different tasks with no significant drop in accuracy.

An application of the proposed e-trainer is introduced in the fourth part of this thesis. The purpose of this application is to estimate the gait parameters (i.e. contact time (CT) and flight time (FT)) of 40 rugby players associated with the Sydney Swans Football Club. This is important because the analysis of such gait parameters can help players increase their running performance and reduce the running-related injury risk. In addition to the CT and FT, a pre-processing system that detects the running period and identifies the 95% confidence interval is introduced to analyse and enhance the detection accuracy. We also investigate the compatibility of CT and FT estimation based on the data collected from a gyroscope and an accelerometer placed in a single location. The results show that the combined accelerometer–gyroscope system obtains the desired accuracy (absolute error <20 ms) in CT and FT detection. Moreover, after introducing the confidence interval, the two systems exhibit high consistency at lower running speeds (<20 km/h).

In conclusion, this thesis describes a comprehensive solution for the design of both hardware and software for electronic virtual trainers. The first part presents an efficient calibration method for gyroscopes. This method only requires simple external devices (or may not need any external device), and can be finished within 30 s. The gyroscope reading accuracy is significantly enhanced by the use of our method. The second part aims to overcome the problem of a lack of data using kernel-based modelling. For users, fewer experiments are needed during the model building period. For the issue of limited resources, the fourth part proposes a model compression method for complex neural networks operating on resource-limited embedded systems. Thereby, novel machine learning algorithms can provide additional guidance to the user.

ACKNOWLEDGMENTS

There are many key points in life, and there are many people and many thanks to keep in mind. First and foremost, I would like to express my most sincere gratitude to my principal supervisor, A/Prof. Steven Su, for his continued support and wise guidance throughout my Ph.D. study. His knowledge, expertise, understanding, and insights greatly expanded my knowledge and skills in many areas. His patience and optimism helped me though the ups and downs of the Ph.D. life. Without his persistent help, the goal of this research would not have been achieved.

I would also like to show my great appreciation to my co-supervisor, Prof. Ren Ping Liu, for his solid support and skilful guidance. His expertise, intelligence, and encouragement enlarged my vision to a broader field and steered me to dive deeper into my research.

Many thanks to the School of Biomedical Engineering, Faculty of Engineering and Information Technology. The friendly and helpful staffs at the school create a great environment for the research study. The resources provided by the school and the faculty is the key of the success in the research.

Additionally, I would like to express gratitude to Dr. Miao Zhang for his treasured support which was really influential in shaping my experiment methods and critiquing my results. Dr. Taoping Liu is an inspiring colleague, blazing a trail I followed in writing my thesis. I also thank Dr. Kairui Guo, Dr. Wei Huang and Dr. Hairong Yu for their mentorship. I would like to thank my friends, colleagues and research team - Feng Shan, Haoding Xu, Jephil Palayil, and Dr. Feng Gao for a cherished time spent together in the lab, and in social settings.

Last but not least, I am grateful for having my wife and my family to support my research journey.

List of Publications

Published :

1. **Wang, L.**, Yang, Y., & Su, S. (2021, November 1-5). *Nonparametric Modelling Based Model Predictive Control for Human Heart Rate Regulation during Treadmill Exercise.* 43rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Guadalajara, Mexico.

2. **Wang, L.**, Zhang, T., Ye, L., Li, J. J., & Su, S. (2021). An Efficient Calibration Method for Triaxial Gyroscope. *IEEE Sensors Journal*. doi: 10.1109/JSEN.2021.3100589.

3. Yang, Y.[1], **Wang, L.**[1], Su, S., Watsford, M., Wood, L., & Duffield, R. Inertial Sensor Estimation of Initial and Terminal Contact during In-Field Running. *Sensors*.

Under Review :

3. **Wang, L.**, Fox, D., Duffield, R., Hammond, A., Zhang, A., & Su, S. An Infield Gyroscope Calibration Method In Wearable Health Monitoring. Submitted to *IEEE Transactions on Systems Man Cybernetics-Systems.*

4. **Wang, L.**[1], Huang, W.[1], Zhang, M, & Su, S. Pruning Graph Neural Network by Evaluating Edge Property. Submitted to *Knowledge-Based Systems*.

5. Zhang, M., **Wang, L.***, Campos, D., Huang, W., Guo, C., & Yang, B. Weighted Mutual Learning with Diversity-Driven Model Compression. Submitted to *NeurIPS 2022*.

6. Huang, W.[1], **Wang, L**.[1], Wang, S., & Zhang, M. Can Wide Variational Auto-encoder Generate? Submitted to *NeurIPS 2022*.

7. Liu, T., Zhang, W., **Wang, L.**, Ueland, M,. Forbes, S., Zheng, W., & Su, S. Numerical Differentiation from Noisy Signals: A Kernel Regularization Method to Improve Transient-State Features for the Electronic Nose. Submitted to *IEEE Transactions on Systems Man Cybernetics-Systems*.

UNDER PREPARATION :

9. **Wang, L.**, & Su, S. On the Influence of Rotation Speed and Noise Intensity of the Calibration Accuracy of Gyroscope. Prepare submit to *IEEE Transactions on Industrial Electronics*.

Table of Contents

List of Figures

Figure **Page**

List of Tables

CHAPTER 1

INTRODUCTION

1.1 Motivation and Scope

Exercise and general fitness practices have played a vital role in numerous cultures throughout history. From the dawn of humankind to around 10,000 BC, physical development followed a natural path that was determined by the practical demands of life in a wild landscape, as well as the vital need to avoid threats and seize opportunities for survival. Civilised populations also valued physical fitness for sports. Records of athletic competitions exist from ancient Egypt, and the ancient Greeks famously created the Olympic games. Not surprisingly, these early sports were all based on practical, natural movement skills and were fundamentally related to the preparedness needed for war.

Nowadays, however, the widespread dedication to fitness has been endowed with different functions and meanings. In fact, fitness offers benefits beyond survival. Regular exercise and physical activity promote strong muscles and bones, staying active can help maintain a healthy weight and an attractive appearance, and it has been found that people who exercise regularly are less likely to become sick than those who do not.

Thus, exercise and general fitness practices are being increasingly investigated,

appreciated, and employed in different areas. Research shows that physical activity or exercise can improve health and reduce the risk of developing illnesses such as type-2 diabetes [9–11], cancer [12, 13], and cardiovascular disease [14–16]. Numerous studies have shown that regular physical activity increases life expectancy and reduces the risk of premature mortality [17–19]. Studies also indicate that physical activity can improve mental health [20, 21] and self-confidence [22, 23]. Recently, with regard to COVID-19 infection, researchers found that general physical fitness can reduce the hospitalisation rate of the population [24] and shorten the recovery time [25].

Despite their enhanced physicality, exercisers still suffer from training injuries due to lack of training knowledge, and may be reluctant to invest time and money to learn the systematic fitness concepts required to become competent self-trainers. Owing to the urgent need for training guidance, personal trainers and physiotherapists have become popular professions. However, they have natural deficiencies: i) It takes considerable time and money to train novices to become competent in certain tasks. ii) The onset of fatigue limits the exercisers' training time. iii) Face-to-face training or rehabilitation guidance may cause the spread of COVID-19 and cannot overcome the geographical obstacles presented by lockdowns.

Aware of the power and market value of fitness and the shortages of traditional personal trainers, researchers both in academia and industry have started to devote themselves to developing electronic personal trainers. Also known as e-trainers, these devices are designed to collect and analyse human motion signals and provide training suggestions [26–29]. Nowadays, e-trainers are ubiquitous, e.g. smartwatches and fitness trackers. Figure 1.1 (left) shows a common sports band that is currently available on the market. It can collect some basic sports data, but does not provide real-time feedback based on our physical state. Figure 1.1 (right) shows a small wearable e-trainer designed in this study, which can collect, process, and classify the motion signals of the human body. In the near future, this will also have the ability to provide users with exercise suggestions in real time.

An e-trainer is composed of two major parts, namely hardware and software (see

Figure 1.1: Left: Common sports band. Right: Fit.E, designed by the candidate.

Figure 1.2). The hardware can be further classified as motion signal sensor(s), a user interface (UI), and a control system. The UI includes graphical user interfaces (GUIs), auditory interfaces, and haptic user interfaces, of which the GUIs are the most common. The software can be further classified as sensor calibration algorithms, feature extraction algorithms, and pattern recognition algorithms.

Previous studies have made great contributions to both the hardware and software of electronic personal trainer systems. Hardware design focuses on two aspects: i) Research on motion sensors focuses on updating sensing materials and/or designing transduction mechanisms. This contributes to various motion sensors, such as electrocardiograms (ECGs) [30, 31], electromyogram (EMGs) [32, 33], initial measurement units (IMUs) [34, 35], depth-of-field sensors [36, 37], and visible spectrum cameras [38, 39]. ii) Research on control systems mainly focuses on reducing the product size to make it portable and/or wearable [40, 41], increasing battery life by enhancing battery capacity and/or decreasing power consumption [42, 43], and improving communication quality in terms of data transmission speed and signal reliability [44, 45]. The software design covers sensor calibration algorithms, feature extraction algorithm design, and task-determined pattern recognition algorithm design.

This thesis mainly focuses on the design of an IMU calibration algorithm and the practical implementation of pattern recognition algorithms. The major challenges re-

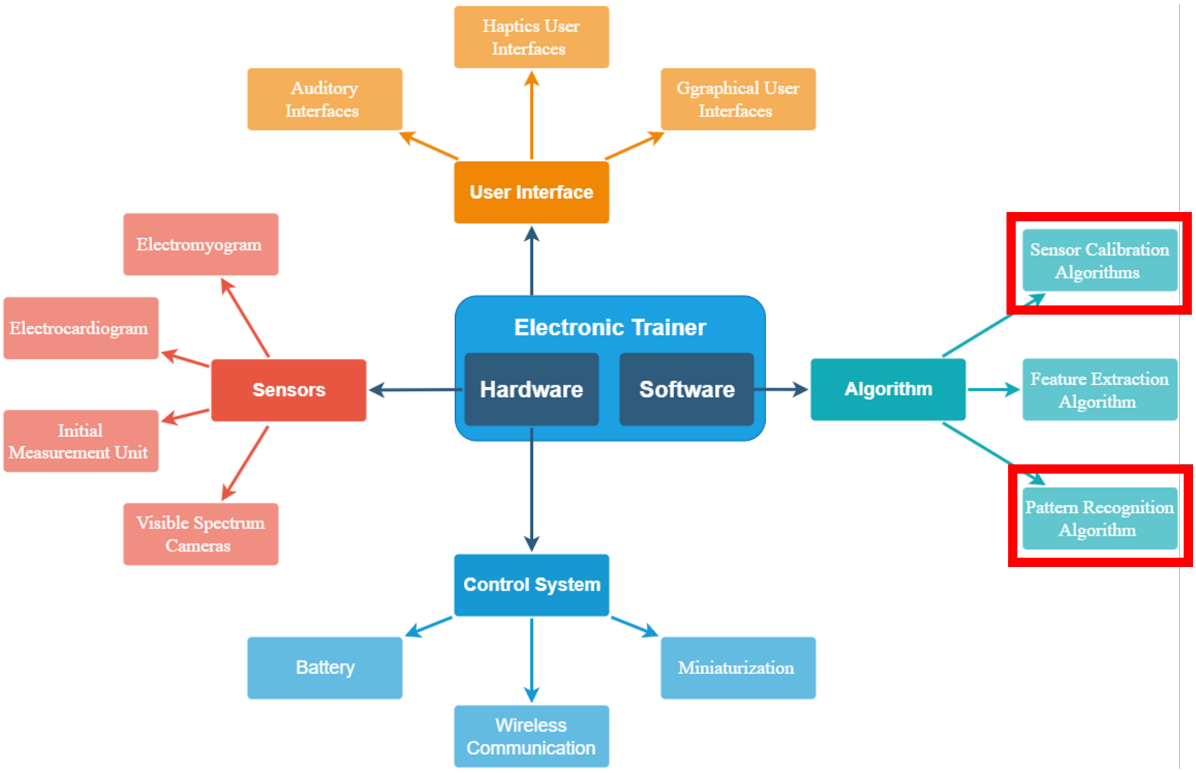

Figure 1.2: Research directions of virtual personal trainers. Red rectangles indicate the focus of this study.

garding these two parts will be discussed in Section 1.2. The instruments included in our experiments are our self-designed wearable e-trainer and self-designed portable e-physiotherapy unit, named E.Fit and Franky, respectively. E.Fit is a wearable electronic trainer with ultra-low power consumption and an IMU with nine degrees of freedom, which is capable of motion tracking and monitoring. Franky is a self-propelled telehealth rehabilitation and assistant robot for stroke patients.

1.2 Design Challenges and Solution Strategy

As people realise the convenience of miniaturised e-trainers, the design of these devices has changed significantly. On the one hand, micro-electromechanical sensors suffer from low accuracy and inconsistency, making their measurements untrustworthy. On the other hand, the implementation of advanced machine learning algorithms in such devices

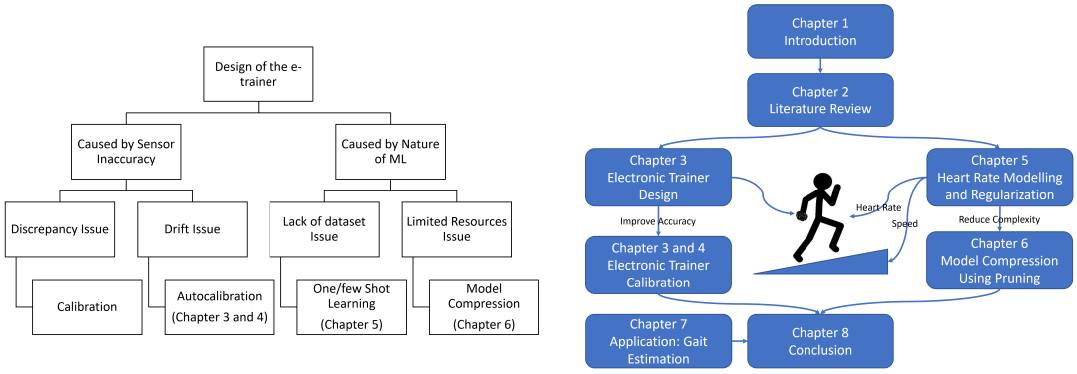

Figure 1.3: Summary of the design challenges and thesis outline.

suffers from a lack of training data and insufficient computing resources.

1.2.1 Challenges in Using Inertial Measurement Unit

1.2.1.1 Discrepancy Issue

The discrepancy issue refers to the problem of different individual sensors of the same model giving different measurements of the same signal under the same environmental conditions. Ideally, a gyroscope would only be sensitive to the rotation rate. In practice, all gyroscopes are sensitive to acceleration due to asymmetry of their mechanical designs and/or micromachining inaccuracies. There are multiple manifestations of acceleration sensitivity, with the severity varying between designs. The inevitable variability in the manufacturing process causes discrepancies between the parameters of two identical sensors. For the IMUs of micro-electromechanical systems (MEMSs), the discrepancies come from two aspects: i) Baseline discrepancy caused by the fluctuation of a sensor's capacitor value C_0 in the stable state; ii) Sensitivity discrepancy caused by the fluctuation of a sensor's rate constant K_x, y, z under actual motion. Under the influence of the discrepancy issue, the same model of IMU will obtain completely different acceleration/angular rate measurements from identical movements. For attitude estimation tasks, in particular, the error accumulates so that the estimated result is far from the ground truth.

Solution Strategy: The discrepancy issue can be solved by using calibration. The

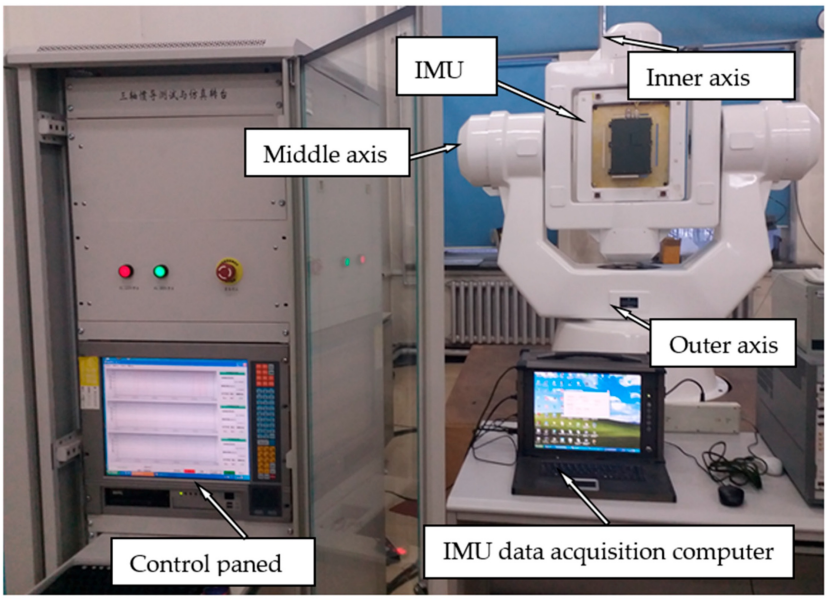

Figure 1.4: Turntable calibration system [1].

calibration process adjusts the gyroscope readings to a calibrated standard of known accuracy. The gold-standard method uses a turntable to calibrate the IMU. The IMU is precisely aligned on the turntable before rotating along x-, y-, and z-axes at a known rotation speed and angle [46].

1.2.1.2 Drift Issue

The drift issue refers to the difference caused by changes in the state of IMU sensors. We classify the drift issue into two categories: i) Irreversible drift (or long-term drift) refers to a permanent change in the state of the sensor, such as through aging (e.g. micromachining fatigue over a long period of time) and over-range damage (e.g. a shock degrades the sensor's micro-mechanical structure). ii) Reversible drift (or short-term drift) refers to a temporary change in the state of the sensor, such as through variations in environmental conditions (e.g. humidity and temperature) and memory effects (e.g. the residue of the previous movement).

Solution Strategy: Irreversible drift usually means that the sensor is damaged and needs to be replaced. The user should not exceed the measurement range and replace the

sensor as soon as possible once such damage has occurred. Reversible drift results in poor repeatability and significant volatility on every use or under different environmental conditions because the IMU parameters have changed. Therefore, it is necessary to calibrate the IMU on each initialisation or when the environmental conditions change. However, current calibration processes are typically time-consuming, and so many simple and efficient auto-calibration methods have been investigated [5, 6, 47].

1.2.2 Challenges in Implementing Pattern Recognition Algorithms

The success of deep learning can be attributed to three key factors: powerful computing resources (e.g. GPUs), complex neural networks (e.g. convolutional neural networks, long short-term memory), and large-scale datasets. However, e-trainer users often do not want to spend time training the device before it gives accurate results. Additionally, the computing resources of portable or wearable e-trainers are generally limited.

1.2.2.1 Lack of data

Typically, machine learning algorithms require large amounts of data before they can give useful results. Deep learning usually requires large data volumes as support. If insufficient training data are available, the problem of over-fitting may occur. However, the collection and labelling of data are both time- and money-consuming. In terms of e-trainer design, the model should be personalised build and trained for a specific user, while asking users to collect and label a substantial amount of data is clearly unrealistic. In addition, the data quality affects the accuracy of any machine learning algorithm. Deep learning can imitate the content of the data without understanding the data. It does not evaluate the quality of the data or reject any incorrect data, and so the final results may be erroneous.

Solution Strategy:

Few-shot learning is widely used to solve the problem of insufficient labelled data. The core idea of this method is to use prior knowledge to replace the training data. This

prior knowledge can come from either an expert or from a previously learned model. The training process starts from the previously learned model, rather than learning from scratch. Thus, the training process requires less time. The prior knowledge is contained in the regularisation term of the loss function. Consequently, the parameters are adjusted according to the prior knowledge to achieve the desired model output.

1.2.2.2 Limited Resources on Embedded Systems

Neural network models are widely used in fitness suggestion systems. Generally, a machine learning model has many parameters, necessitating a massive amount of storage and a high level of computational complexity; consequently, the time and energy costs of such models can be significant. However, due to size and weight limitations, the battery capacity of e-trainers is often small. To balance the battery life and capacity, the e-trainers' processors are generally designed to achieve ultra-low power consumption, which means that the computing power and the memory capacity on the chip are often insufficient for running neural network models. Furthermore, e-trainers should provide users with real-time feedback. Thus, the contradiction between low latency and low power consumption is increasingly prominent.

Solution Strategy:

Model compression can turn a large model into a small model, and the compressed form often achieves performance close to or even better than the large model. Commonly used model compression methods include network pruning, knowledge distillation, parameter quantisation, and model architecture design. A small model always means less computation power and less memory usage, and consequently less power consumption. Therefore, advanced neural network models can be used in small-size devices.

In conclusion, this thesis aims to solve the problems of discrepancies, drift, lack of data, and limited resources.

1.3 Influence of COVID-19

The COVID-19 pandemic has affected almost my entire candidature. Due to the impact of the epidemic, I had to modify my research plan. This section elaborates on my initial research plan and the changed research plan in response to COVID-19.

The initial research plan had three aims:

- To develop a kernel-based model of heart rate (HR) response during treadmill exercise for model predictive control and verify that the model can be used for HR tracking.
- Use the established model as a simulation environment to train a reinforcement learning agent, and apply the deep deterministic policy gradient method to a HR tracking project.
- Design a wearable exercise device that can monitor the intensity of the exerciser and recommend a suitable exercise intensity. The algorithms developed for Aims 1 and 2 would run on the designed device without the need for bulky monitoring equipment.

After completing the experiments associated with Aim 1, measures required to stop the spread of COVID-19—such as campus closures, lockdowns, and additional lab access limits—disrupted my plan. We had designed the algorithm for Aim 2, but could not carry out the necessary experiments due to stringent restrictions on human-related research during the pandemic period. Therefore, following discussions with my supervisor, I conducted some experiments that did not require human participation while waiting for the end of the pandemic. We decided to change my research plan. The experiments associated with Aim 2 were postponed, and we instead designed the wearable device for Aim 3. We also developed an efficient autocalibration method for the designed device. During the first campus reopening, we tested this device and used it in a practical project for monitoring pregnant women. In addition, to run the advanced artificial intelligence-based motion analysis method on the resource-limited wearable device, we proposed a

neural network pruning method to simplify and reduce the computational complexity of the method.

My revised research plan is as follows:

- To develop a kernel-based model of HR response during treadmill exercise for model predictive control and verify that the model can be used in HR tracking.

- To develop a wearable movement tracking device to investigate whether and how women with complex pregnancies mobilise themselves during the first stage of labour in differently designed birth rooms.

- To improve the tracking accuracy of wearable devices and make the results of attitude estimation and motion state estimation more accurate, a fast and efficient gyroscope calibration method is proposed in combination with the existing accelerometer calibration method.

- To run the designed algorithm on a resource-limited wearable device, the neural network pruning method is applied to reduce its computational complexity.

1.4 Thesis Contributions

This thesis describes a complete solution for the monitoring and control of human physical exercise intensity, including the design of an exercise-intensity monitoring device, device calibration, control algorithm design, and algorithm optimisation. The major contributions of this thesis can be summarised as follows:

- Designing the hardware for an electronic virtual trainer: *Ultra-low power consumption e-trainer – Fit.E* The wearable e-trainer, Fit.E, was designed by the candidate from scratch. Figure 1.5 shows a functional block diagram of the device. This waterproof device can be charged wirelessly and offers battery life of around 22 h. Figure 1.6 shows the interior and exterior of the device. All components are attached to a

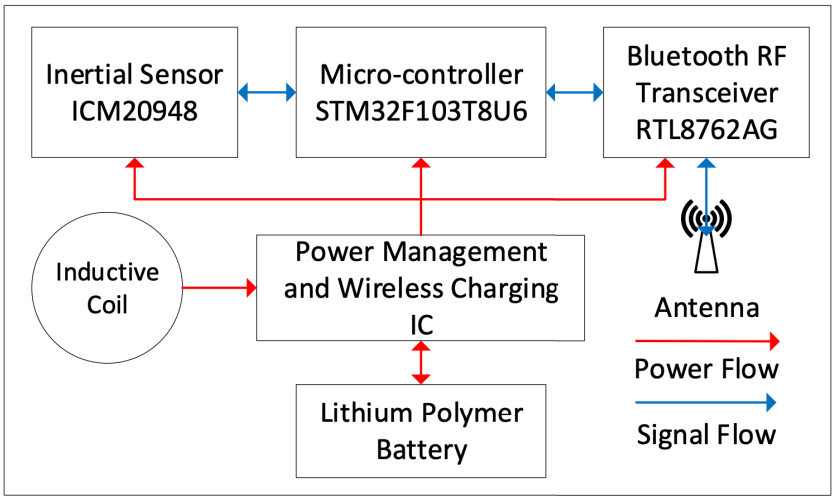

Figure 1.5: Functional block diagram of Fit.E.

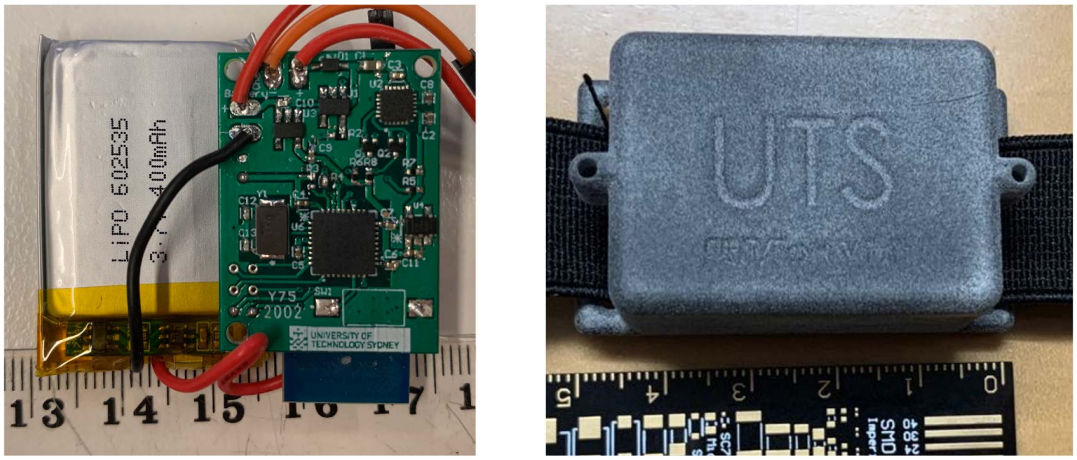

Figure 1.6: Left: Interior of Fit.E. Right: Exterior of Fit.E.

four-layer printed circuit board (PCB) using surface mounting technology. The case used in this study was 3D printed.

AI-based exoskeleton for upper limb rehabilitation – AIEXO The candidate also designed a portable electronic physiotherapist as a group work. The candidate mainly focused on designing the electrical system for this robot. Figure 1.7 (left) shows the functional diagram of the electrical system. The electrical system designed by the candidate includes a power system, motor drivers, and an embedded control system. Figure 1.7 (right) is the control PCB used by the system. Figure 1.8 shows

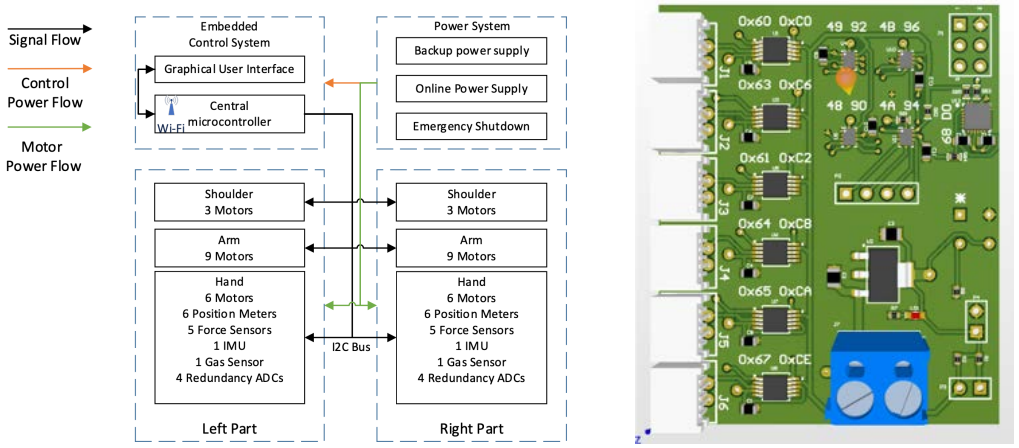

Figure 1.7: Left: AIEXO system architecture. Right: Embedded control board in hand.

a prototype of the portable electronic physiotherapist.

- Proposition for modelling and control of the human physiological system: *Human HR regulation based on nonparametric modelling.* This method applies a kernel-based nonparametric modelling method to estimate the HR response during treadmill exercise and proposes a model predictive control (MPC) method to perform HR control for an automated treadmill system. A kernel regularisation term that provides prior information to the model estimation phase was introduced. By adding this prior information, the experimental protocol was significantly simplified so that only a small number of model training experiments are needed. For the patient, this means fewer experiments during the model construction phase. Based on the identified model, an MPC controller was designed to track a predefined reference HR profile. One advantage of this scheme is that the speed and acceleration of the treadmill can be limited to within a safe range for vulnerable exercisers.

- Proposition of the model compression algorithm:

  Model compression and calculation reduction via pruning method.

  One of the main factors preventing the application of artificial intelligence-based methods to wearable devices is that the embedded systems usually have limited memory and computing power. To this end, we utilised neural network pruning

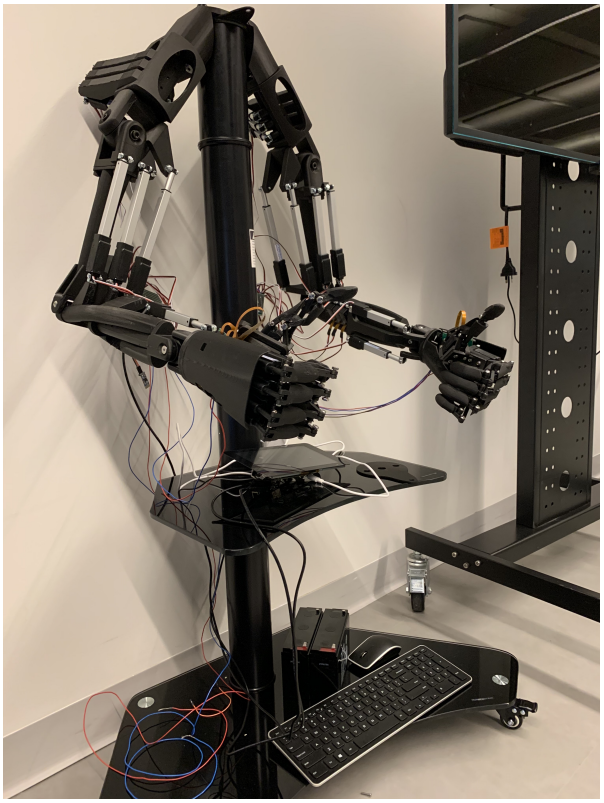

Figure 1.8: Prototype of the AIEXO.

technology to compress the model and make it feasible for use in the wearable device. By removing superfluous components of the neural network model, the number of parameters used in the model was reduced by 98% without causing a significant drop in performance. This led to remarkable reductions in memory and computation usage during the inference period. This provides a protocol for migrating the advanced artificial intelligence algorithms that previously only ran on bulky devices to wearable devices. Consumers can thus enjoy the benefits of artificial intelligence anytime and anywhere without being restricted by equipment, and the cost of purchasing the equipment will be reduced.

- Proposition of calibration method and application of wearable device:

  Efficient calibration method for triaxial gyroscope

  The parameter variation of low-cost triaxial gyroscopes requires an effective and

practical in-field calibration process to reduce errors due to unexpected variance. An efficient servomotor-aided calibration algorithm was designed for a triaxial gyroscope. The entire calibration process only takes approximately 1 min, and does not require high-precision equipment. This method is based on the idea that the measurement of the gyroscope should be equal to the rotation speed of the servomotor. A six-observation experimental design is proposed to minimise the maximum variance of the estimated scale factors and biases. In addition, a fast-converging recursive linear least-squares estimation method is presented to reduce computational complexity. The feasibility of the proposed method has been experimentally demonstrated on a robot arm, and the method has been implemented on a microcontroller. By comparing the calibrated low-cost gyroscope reading with that from a high-precision gyroscope, we can conclude that our method significantly increases the gyroscope's accuracy.

In-field gyroscope calibration method for wearable health monitoring

This method further simplifies the equipment required for calibration. Specifically, we developed an efficient in-field calibration method that readily calibrates the triaxial gyroscope without additional equipment. This experimental scheme can be easily implemented by manually rotating the triaxial gyroscope over a certain angle as the calibration reference. A linearised calibration model has been developed for the proposed experimental scheme, and we showed that G-optimality is achievable. We also designed a low-energy, cost-effective, and wearable wireless movement tracking device for health monitoring. We applied the proposed triaxial gyroscope calibration approach in the wearable device and found that it significantly improved angle estimation accuracy. Doctors can use the device to determine a patient's exercise intensity accurately with respect to time, resulting in more accurate diagnoses of the impact of exercise on the patient.

Temporal gait parameter estimation method using Fit.E

We utilised the designed device to analyse the CT, FT, and gait asymmetries of 40

rugby players from Sydney Swans Football Club while running. The results showed that the combined accelerometer-gyroscope system obtains the desired accuracy (absolute error <20 ms) in both CT and FT detection.

1.5 Thesis Outline

This thesis is arranged as follows:

CHAPTER 1

This chapter presents an introduction to the motivation and scope of the study, the challenges faced, the contributions made by study results, and an outline of the thesis.

CHAPTER 2

This chapter presents an in-depth review of each aspect of the study and provides a critical analysis of the existing literature.

CHAPTER 3

This chapter describes an efficient servomotor-aided calibration method for the triaxial gyroscope. The entire calibration process only takes approximately 1 min, and does not require high-precision equipment. This method is based on the idea that the measurement of the gyroscope should be equal to the rotation speed of the servomotor. A six-observation experimental design is proposed to minimise the maximum variance of the estimated scale factors and biases. In addition, a fast-converging recursive linear least-squares estimation method is designed to reduce computational complexity. The simulation results verify the robustness of the calibration method under normal and extreme conditions. We experimentally demonstrate the feasibility of the proposed method on a robot arm and implement the method on a microcontroller. We verify the calibration results of the proposed method through comparisons with a traditional turntable approach, and find that the results of these two methods are comparable. Thus, we conclude that our method significantly increases the gyroscope's accuracy.

The work in this chapter has been published in:

- **Wang, L.**, Zhang, T., Ye, L., Li, J. J., & Su, S. (2021). An Efficient Calibration Method for Triaxial Gyroscope. *IEEE Sensors Journal*. doi: 10.1109/JSEN.2021.3100589.

CHAPTER 4

This chapter presents an efficient in-field calibration method that readily calibrates the triaxial gyroscope without additional equipment. This experimental scheme can be easily implemented by manually rotating the triaxial gyroscopes over a certain angle as the calibration reference. A linearised calibration model is developed for the proposed experimental scheme, and we show that G-optimality is achievable. Extensive numerical simulations indicate that the calibration error is relatively low and that the estimation of model parameters is unbiased under mild experimental conditions. We also empirically validate the effectiveness of the proposed method on two commonly used low-cost gyroscopes and achieve real-time calibration on a low-energy consumption microcontroller with low computational power. In addition, we validate the effectiveness and practicality of the proposed method in comparison with three state-of-the-art methods. The absolute error of the scale factors and biases are always less than 2.5×10^{-2} for LSM9DS1 and less than 1×10^{-2} for ICM20948, and the calibration process takes less than 30 s.

The work in this chapter has been concluded in:

- **Wang, L.**, Fox, D., Duffield, R., Hammond, A., Zhang, A., & Su, S. An Infield Gyroscope Calibration Method In Wearable Health Monitoring. Submitted to *IEEE Transactions on Systems Man Cybernetics-Systems*.

- **Wang, L.**, & Su, S. On the Influence of Rotation Speed and Noise Intensity of the Calibration Accuracy of Gyroscope. Prepare submit to *IEEE Transactions on Industrial Electronics*.

CHAPTER 5

This chapter describes the application of a kernel-based nonparametric modelling method to estimate the HR response during treadmill exercise and proposes a model predictive control (MPC) method for HR control in an automated treadmill system. The kernel-based method introduces a kernel regularisation term that provides prior

information to the model estimation phase. By adding this prior information, the experimental protocol can be significantly simplified and only a small number of model training experiments are needed. The model parameters are experimentally estimated from 12 participants in a treadmill exercise with a short and practical exercise protocol. The modelling results show that the model identified using the proposed method accurately describes the HR response to the treadmill exercise. Based on the identified model, an MPC controller is designed to track a predefined reference HR profile. One advantage of this scheme is that the speed and acceleration of the treadmill can be limited to within a safe range for vulnerable exercisers. The proposed controller is experimentally validated in a self-developed automated treadmill system. The tracking results indicate that the desired automatic treadmill system can efficiently and safely regulate the participants' HRs to follow the reference profile.

The work in this chapter has been published in:

- **Wang, L.**, Yang, Y., & Su, S. (2021, November 1-5). *Nonparametric Modelling Based Model Predictive Control for Human HR Regulation during Treadmill Exercise.* 43rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Guadalajara, Mexico.

CHAPTER 6

Artificial intelligence-based human action classification and exercise intensity control algorithms provide appropriate guidance for exercisers. However, the application of these algorithms to embedded devices is limited by the computing power and memory capacity of such devices. Therefore, this chapter proposes a method that significantly reduces the computational load of graph neural networks (GNNs). Larger graphs and deeper GNNs make the training and inference procedures increasingly expensive. The recently reported graph lottery ticket (GLT) method [48] leverages the lottery ticket hypothesis (LTH), which was initially proposed for neural network pruning, to design a unified GNN sparsification framework that simultaneously prunes the graph adjacency matrix and the model weights. Besides leveraging the LTH, we further extend several existing training-free saliency metrics (SNIP, GraSP, Synflow) for network pruning to GNNs. We

find that these methods can achieve similar performance as GLT when jointly pruning the network weights and graph adjacency matrix. This reveals that these training-free saliency metrics can replace LTH with greatly reduced computational cost.

This chapter is a collaborative work and has been concluded in:

- **Wang, L*.**, Huang, W*., Zhang, M, & Su, S. Pruning Graph Neural Network by Evaluating Edge Property. Submitted to *Knowledge-Based Systems*.

CHAPTER 7

This chapter demonstrates an in-field application of Fit.E, analysing the CT, FT, and gait asymmetries of 40 rugby players from Sydney Swans Football Club during running. We propose an intelligent running gait analysis system that can estimate the CT and FT with the desired accuracy. Furthermore, a pre-processing system that detects the running period and a 95% confidence interval are introduced to analyse and increase the detection accuracy. To the best of our knowledge, this is the first study to investigate the compatibility of CT and FT estimation based on the data collected from a gyroscope and accelerometer placed in a single location. The results show that the combined accelerometer-gyroscope system obtains the desired accuracy (absolute error <20 ms) in both CT and FT detection. Moreover, after introducing the confidence interval, the two systems show high consistency at lower running speeds (<20 km/h). The effect of the striking type on the current gait algorithms in different IMU placement locations should be investigated in future research.

This chapter is a collaborative work and has been published in:

- Yang, Y*., **Wang, L*.**, Su, S., Watsford, M., & Duffield, R. Reliable Inertial Measurement Based Temporal Gait Parameter Estimation. Submitted to *Sensors*.

CHAPTER 8

This chapter summarises the work described in this thesis and presents several prospective directions for future research.

CHAPTER 2

LITERATURE REVIEW

2.1 Review of Physiological Signal Response Modelling

2.1.1 Nonparametric Dynamical Models

For a linear time-invariant system, the output is equal to the convolution of the system input and the system impulse response. In fact, the process of finding the finite impulse response (FIR) of the system from observation data is a deconvolution problem [49]. Such problems are common in biology, physics, and engineering [49–51]. It is impossible to uniquely identify a continuous-time FIR from a finite dataset because it is always an ill-posed problem. Additionally, estimating an FIR from discrete-time signals results in an ill-conditioned situation, which means that small levels of noise in the dataset might produce large estimation errors. The regularisation method was first proposed by Tikhonov and Phillips [51, 52]. Since then, regularisation methods have become a hot topic in the field of system identification, and several techniques have been proposed, such as L_1-regularisation [53, 54] and gradient descent methods [55, 56].

The most popular system identification approach is the maximum-likelihood method [49, 57]. Based on the assumption that the model class is fixed, the maximum-likelihood method is well understood. For a large dataset, the procedure is optimal with respect to minimising the error. However, for parametric methods, the most important thing is to choose a suitable model class. In the frequentist framework, the balance between bias and variance can be problematic, and several model validation techniques have been developed to solve this issue, such as Akaike's information criterion (AIC) method [58] and the cross-validation method. In some studies [59–61], these classical approaches have been identified as unsatisfactory for the experimental data, differing from the characteristics predicted by classical statistical theory. This indicates that prediction error methods might be asymptotically efficient for Gaussian approaches.

Researchers have shown that the kernel-based method can solve the model selection problem in system identification [59]. This method leads to interesting interdisciplinary applications between the fields of control and machine learning. The traditional method always assumes that the system has a finite-dimensional space, such as ARX and ARMAX, while the new paradigm expresses the problem as one of function estimation in an infinite-dimensional space, namely a reproducing kernel Hilbert space. From a system identification point of view, the impulse response model lies in such a space. The inherent ill-posed nature of the system can be solved by the regularisation term, which can also be explained by Bayesian inference [62]. Specifically, the FIR is considered to be a zero-mean Gaussian process. Thus, prior information is added to the identification process by assigning a covariance, which is also called a kernel in the machine learning field [63].

Unless some key features of the system identification problem are considered, the direct application of these technologies to the control field is doomed to fail. First, the relationship between the unknown function and the measurement is not a direct relationship, as normally assumed in machine learning settings, but is established indirectly through the convolution of the system inputs. This has a clear analogy with studies on inverse problems [50, 64]. Additionally, for system identification, the stability

of the impulse response must be known by the estimation process. In this regard, a major recent development is the introduction of new kernels that include information on the exponential stability of the impulse response [59, 61]. These kernels depend on some hyperparameters which can be maximised using marginal likelihood. This corresponds to model order selection in the classic prediction error method (PEM) paradigm, but achieves more robust performance, which might be why these procedures are successful. Other studies have proposed different kernel designs, such as the maximum entropy kernel [60], optimal kernel [61], and wavelet kernel [65].

2.1.2 Artificial Neural Network Models

Artificial neural network models are designed on the basis of modern neuroscience, reflecting the structure and function of the human brain. Since the 1980s, research on artificial neural networks has made great progress, and the related theories and methods have developed into an interdisciplinary subject straddling physics, mathematics, computer science, and neurobiology. These networks are widely used in pattern recognition, medical image processing, rehabilitation robot control, and physiological system modelling.

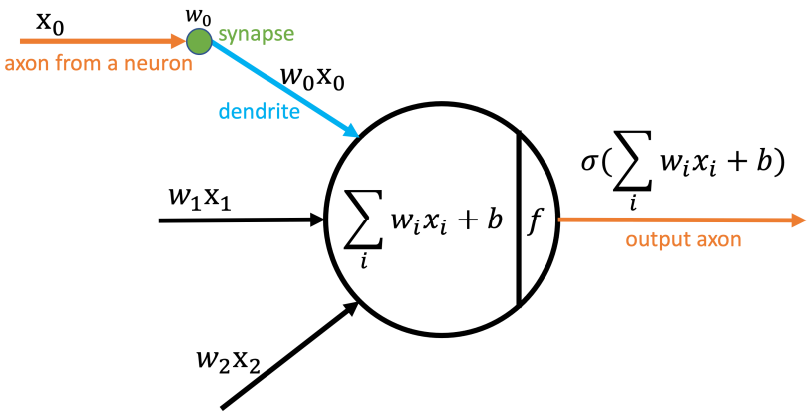

Figure 2.1: Mathematical model of an artificial neuron.

The most basic component of a neural network is the neuron model. In biological

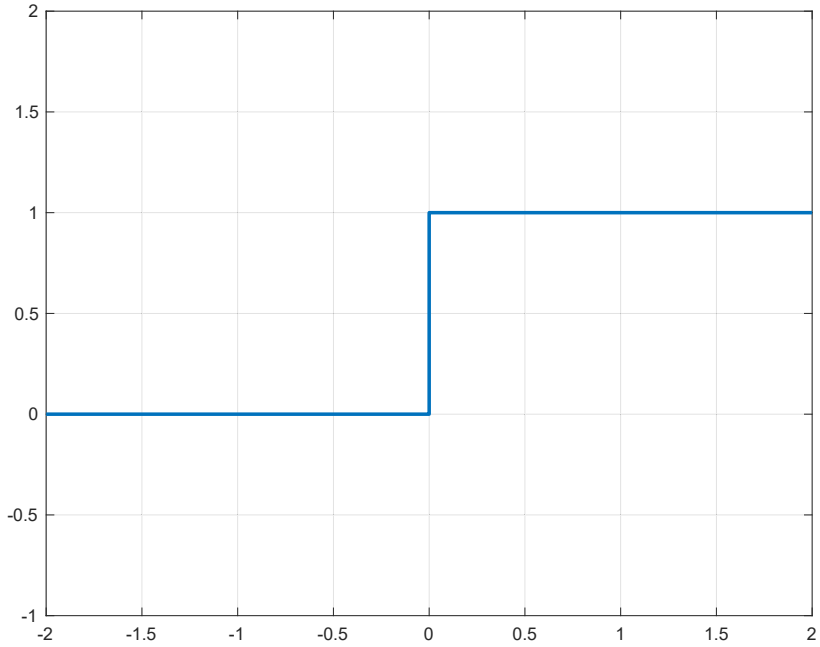

Figure 2.2: Ideal activation function: Unit step function.

neural networks, each neuron is connected to other neurons. When excited, a neuron sends chemicals to connected neurons, changing the electrical potential within those neurons. If a neuron's potential exceeds a threshold, it becomes activated, i.e. excited, and sends chemicals to other neurons.

In 1943, **(author?)** [66] abstracted the above situation into the simple model shown in Figure 2.1, which is the "McCulloch-Pitts neuron model" that has been used until now. In this model, a neuron receives input signals from i other neurons. These input signals are passed through a weighted connection, and the total input value received by the neuron is compared with the neuron's threshold. It is then processed through an activation function to produce the neuron's output.

The ideal activation function is the step function shown in Figure 2.2, which maps the input value to an output value of "0" (neuron inhibition) or "1" (neuron excitation). However, the step function has discontinuous and non-smooth properties, which will cause problems in the process of neural network training. Therefore, other activation functions such as the sigmoid function, ReLU function, tanh function, and Softplus

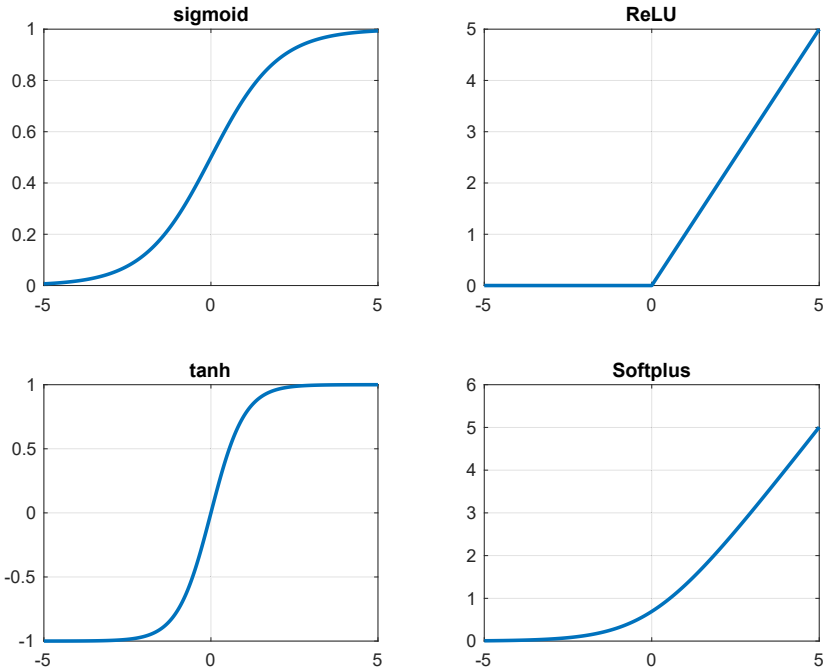

Figure 2.3: Commonly used activation functions.

function are commonly used (see Figure 2.3). The activation function squeezes the input values, which may vary over a wide range, into an output value range of (0, 1).

Artificial neural networks connect many such neurons in a certain hierarchical structure. From a computer science point of view, we can ignore whether the network really simulates a biological neural network and think of it as a mathematical model with many parameters and numerous functions.

One common neural network is the hierarchical structure shown in Figure 2.4. Each layer of neurons is fully interconnected with the lower-layer neurons. There is no same-layer connection between neurons and no cycles are formed. Such a neural network structure is usually called a “multi-layer feedforward neural network". The input-layer neuron receives an external input, the hidden-layer and output-layer neurons process the signal, and the final result is output by the output-layer neurons. In other words, the input-layer neurons only accept input and do not perform functional processing, and the hidden layer and output layer contain functional neurons. The learning process of the neural network involves adjusting the connection weights between neurons and the

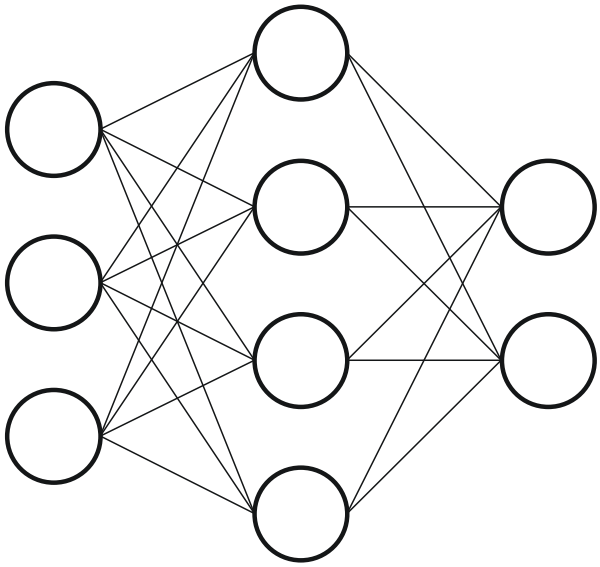

Figure 2.4: Typical neural network with one hidden layer.

threshold of each functional neuron according to the training data. In other words, the knowledge learned by the neural network is contained in the connection weights and thresholds.

2.2 Review of Inertial Measurement Units

MEMS technology is used to design, process, manufacture, measure, and control micro/-nano materials based on micro/nanotechnology. It integrates mechanical components, optical systems, drive components, and electronic control systems into an integral unit. The resulting micro-device or -system integrates some micro-mechanisms, micro-sensors, micro-actuators, signal processing and control circuits, interfaces, communication modes, and a power supply. If the silicon processing technology is similar to that for integrated circuits, large-scale and low-cost production can be achieved. Thus, the cost performance is greatly improved compared with that of traditional mechanical manufacturing technology and large-scale integrated industry can be realised.

MEMS gyroscopes are an important development direction in the MEMS field. With the development of MEMS technology, inertial micro-gyroscopes have attracted consider-

able attention due to their small size and high precision. There are broad development and market prospects in fields such as car navigation, consumer electronics, mobile applications, aerospace, and high-tech battlefields.

2.2.1 Working Principle

Traditional gyroscopes mainly use the principle of conservation of angular momentum. They are constantly rotating objects, and their axis of rotation does not change with the rotation of the bracket carrying them. However, the working principle of MEMS gyroscopes is slightly different, because it is not easy to process a rotatable structure on a silicon substrate with micromachining technology [67]. MEMS gyroscopes take advantage of the Coriolis force, which is the tangential force that a rotating object experiences when it moves radially.

Consider the dynamic coordinate system shown in Figure 2.5. Three separate accelerations can be calculated: radial acceleration, Coriolis acceleration, and centripetal acceleration [68].

$$
\begin{aligned}
&\vec{r} = r\vec{r}_0 \\
&\vec{\theta}_0 = \vec{\omega}_0 \times \vec{r}_0 \\
&\frac{d\vec{r}^2}{dt} = v_r\vec{r}_0 + r\frac{d\vec{r}_0}{dt} = v_r\vec{r}_0 - r\vec{r}_0 \times \vec{\omega} \\
&\frac{d^2\vec{r}}{dt^2} = a, \vec{r}_0 - 2v_r\vec{r}_0 \times \vec{\omega} - \omega r^2\vec{r}_0 \\
&a_{\text{Consult}} = -2v_r\vec{r}_0 \times \vec{\omega}
\end{aligned}
\tag{2.1}
$$

If there is no radial motion of the object on the disk, the Coriolis force will not arise. Therefore, in the design of MEMS gyroscopes, the object is driven to produce radial motions or back-and-forth vibrations. The lateral vibrations are small, and the phase differs by exactly 90 degrees from the driving force (Figure 2.6).

MEMS gyroscopes typically have movable capacitors in the two directions of each axis. The radial capacitor applies an oscillating voltage to force the object to move radially, and the lateral capacitor plate measures the capacitance change caused by the lateral

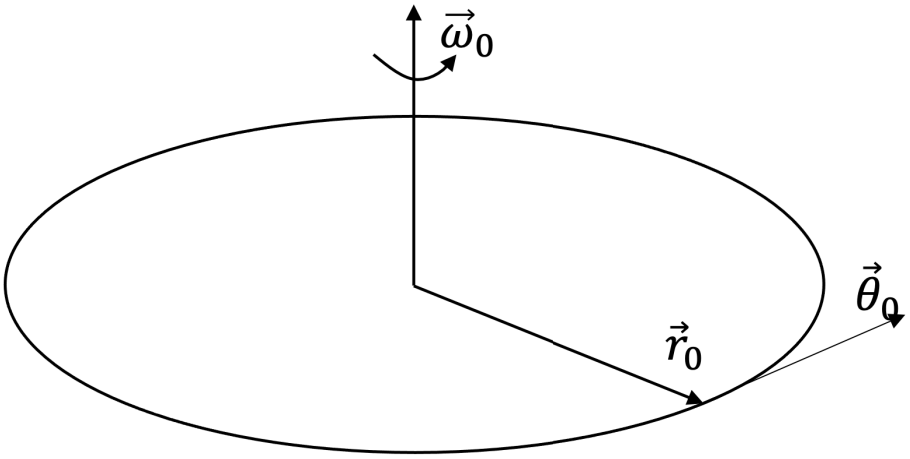

Figure 2.5: Rotation coordinate system.

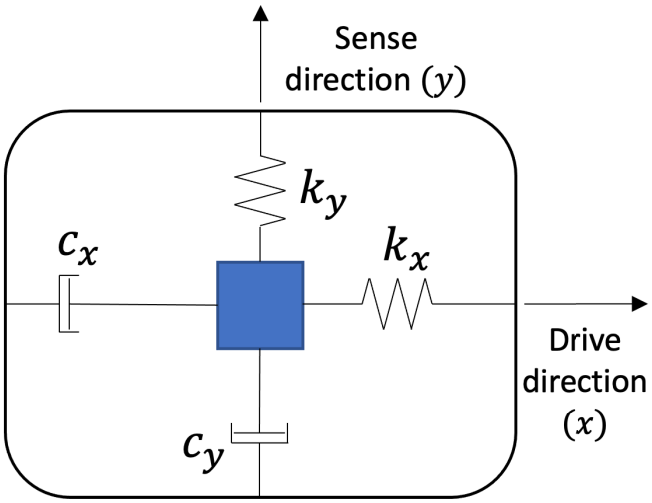

Figure 2.6: MEMS gyroscope working principle.

Coriolis motion. Because the Coriolis force is proportional to the angular velocity, the angular velocity can be calculated from the change in capacitance.

A MEMS gyroscope designed to use vibrations to induce and detect the Coriolis force has no rotating parts and no bearings [69]. Thus, it is easily mass-produced using micromachining techniques. The vast majority of MEMS gyroscopes rely on alternating Coriolis forces caused by mutually orthogonal vibrations and rotations. The vibrating object is suspended above the base by an elastic structure. The overall dynamic system is a two-dimensional elastically damped structure in which vibration- and rotation-induced

Coriolis forces convert energy proportional to the angular velocity into capacitance that can be measured by the circuit.

2.2.2 Main Performance Parameters

A gyroscope has many parameters that affect its performance. In this section, we introduce the following important parameters of a MEMS gyroscope.

Power supply (V): The DC power supply voltage range required for the gyroscope to work properly.

Power supply current (mA): The current consumed by the gyroscope in normal operation.

Sleep mode supply current (mA): The amount of current consumed by the gyroscope in sleep mode.

Power supply current in shutdown mode (μA): The amount of current consumed when the gyroscope is powered off.

Full scale (dps): The scale range of the gyroscope.

Zero rate output value: The number of zero rate output signals when the gyroscope has no angular rate applied.

Sensitivity (mV/dps or dps/LSB): The relationship between one dps and analogue gyroscope output voltage change at zero rate output value, expressed in mV/dps; or, the relationship between a digital gyroscope and the least significant bit, expressed in dps/LSB.

Sensitivity change with respect to temperature (%/°C): Percentage change in sensitivity when the temperature deviates from the standard temperature.

Zero rate output value change with respect to temperature (dps/°C): The zero rate output value change when the temperature deviates from the standard temperature.

Nonlinearity (%): Maximum error between gyroscope output and best-fit straight line as a percentage of full scale.

System bandwidth (Hz): Angular rate signal frequency range from DC to the internal bandwidth measurable by the analogue gyroscope.

Rate noise density (dps/VHz): Standard resolution of analogue gyroscopes and digital gyroscopes that can be obtained from the gyroscope output and internal bandwidth parameters.

2.2.3 Types of Gyroscope

MEMS micro-gyroscopes can be divided into four main categories based on their working principle: vibration gyroscopes, fluid-floated gyroscopes, micro-integrated optical gyroscopes, and micro-atomic gyroscopes.

2.2.3.1 Vibration Gyroscopes

Vibration gyroscopes use the Coriolis effect produced by a high-frequency vibrating mass to detect angular motion [70]. The main body of the vibrating gyroscope is a high-frequency vibration component, which has the advantages of stable performance, simple structure, high reliability, and wide measurement range. Common vibration gyroscopes include tuning fork vibration gyroscopes, piezoelectric vibration gyroscopes, and shell vibration gyroscopes.

Tuning fork vibration gyroscopes. Tuning fork vibrating gyroscopes, or tuning fork resonant gyroscopes, use the Coriolis effect from the vibrating mass at the end of the tuning fork being driven to rotate by the base to sense the angular velocity [71]. Functionally, they are single-axis rate gyroscopes. The arms of the tuning fork perform alternately reciprocating bending motions towards and away from each other under the movement of the vibration excitation device. The masses at the two ends of the tuning fork alternately face towards and away from each other. The vibration excitation device ensures that the tuning fork oscillates with equal amplitude, that is, the amplitude of the oscillation of the two arms is equal, and the phase is exactly opposite. The Coriolis effect of the tuning fork vibrating gyroscope is illustrated in Figure 2.7. Due to the mutual influence of the relative motion and the implicated motion, the two particles at the end of the tuning fork are subjected to Coriolis acceleration and are affected by the Coriolis inertial force. When the particles at the end of the tuning fork move toward each other,

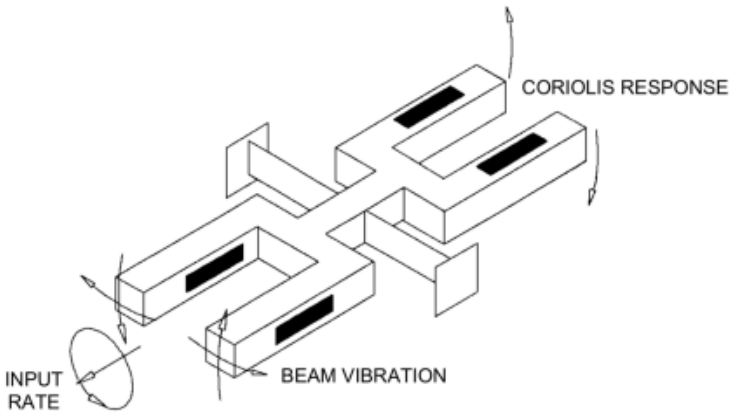

Figure 2.7: Physical model of a typical tuning fork vibration gyroscope [2].

the Coriolis inertial force vector of the two particles is on the $x$oz plane, thereby forming a torque on the central axis of the tuning fork, that is, the Coriolis moment of inertia. In addition to the above phenomenon of the two particles at the end of the tuning fork, the Coriolis effect applies to all particles in symmetrical positions along the two arms of the tuning fork. Therefore, the Coriolis moment of inertia of the entire tuning fork should be obtained by integration. The angular velocity information can be inversely deduced by integrating the output result.

Vibrating shell gyroscopes. Shell vibrating gyroscopes work by utilising the Coriolis effect of the vibrating mass of an axisymmetric shell under the action of the angular velocity [72]. In such gyroscopes, the result of the Coriolis effect is a deflection of the vibration waveform relative to the base. The core part of the shell vibration gyroscope is the resonant shell or resonator, which uses the deflection of the resonator mode shape relative to the base to measure the rotation of the base relative to the inertial space. The deflection direction of the mode shape is opposite the rotation direction of the base, and the deflection angle of the mode shape is proportional to the rotation angle of the base. The deflection of the harmonic oscillator mode shape is obtained by the Coriolis acceleration and the Coriolis inertial force, that is, the corresponding deflection angle of the harmonic oscillator mode shape is obtained through the Coriolis effect of the harmonic oscillator under the action of the angular velocity, so that the vibration of

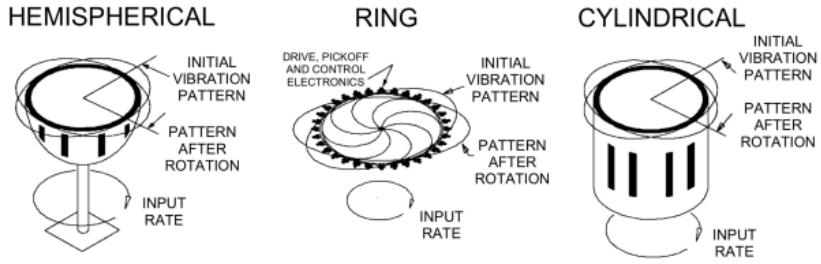

Figure 2.8: Physical model of vibrating shell gyroscope [2].

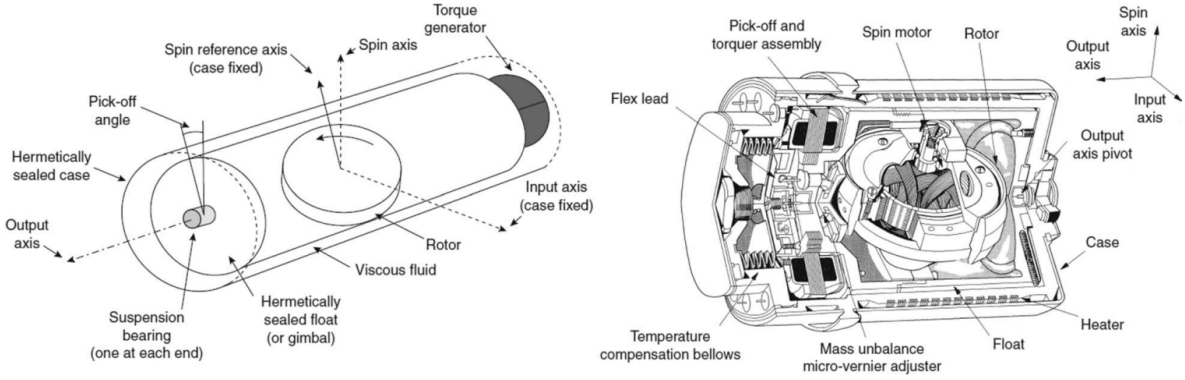

Figure 2.9: Schematic of the fluid-floated gyroscope [3].

the shell can be obtained. The angular velocity information is deduced from the output information of the gyroscope.

2.2.3.2 Fluid-floated gyroscope

The gyroscope motor is supported in a sealed ball (also called a floating ball) by high-precision ball bearings, and a T-shaped outer rotor structure is used to obtain a larger moment of momentum [73]. The gyro ball is composed of two hemispherical shells, which are filled with hydrogen or helium to reduce the wind resistance of the motor, prevent oxidation of parts and lubricants, and improve thermal conductivity. The gyro ball is equipped with a sealed insulator. A conductive hairspring is welded to it for signal transmission and power supply. A mass unbalanced micro-vernier adjuster is fixed to both ends of the gyro ball. The gyro ball is suspended in a floating liquid and is positioned with a jewel bearing. To obtain two rotational degrees of freedom, a ring device is designed

to function as a gimbal. Two pairs of orthogonal gem bearing seats are installed on the ring. The journal is equipped with an adjustment and locking mechanism to ensure axial clearance. At operating temperature, the gyroscope is mechanically suspended. The buoyancy of the floating ball is equal to gravity, so that the inner and outer ring jewel bearing supports are unloaded, and the bearing support only plays a positioning role. The two output axes of the gyro are equipped with two moving coil sensors and two direct current coil torquers. The sensor rotor assembly is installed on the gyro ball outer cover, and the stator is excited by an AC signal. The torquer rotor assembly is mounted on the gyro ball. The stator assembly is composed of magnetic steel, magnetic permeable ring, and so on. It is also installed on the outer end cover of the torquer rotor and is supplied with DC control signals. The gyro shell consists of two outer end caps and a base to form a closed container, and the suspension is filled inside. Both sides of the outer end cover are equipped with bellows components to compensate for the volume change of the floating liquid caused by temperature. Symmetrically distributed heating wires and thermal sensors in the shell form a temperature control component for gyro temperature control. The outer cover acts as magnetic shielding, protection, and incidental insulation.

For a two-degree-of-freedom liquid-floating gyroscope, a pair of pivots is mounted in jewel bearings on the inner ring. Together with the inner ring, the float is mounted in jewel bearings on the housing by a pair of pivots on the inner ring. The spin motor lies perpendicular to the plane formed by the inner and outer ring shafts, so that the gyro ball has freedom to rotate along the inner and outer ring shafts. Two orthogonally mounted sensors are used to measure the angular displacement of the float around the two orthogonal directions, and the outputs correspond to the rotations around the inner and outer ring shafts, respectively. Similarly, two orthogonally mounted torquers are used to control the motion in the two orthogonal directions, respectively, to generate moments about the inner and outer ring axes. The two-degree-of-freedom liquid-floating gyroscope is sensitive to the angular motion of two orthogonal axes at the same time, and uses two gyroscopes as sensitive components to form a three-axis stable platform

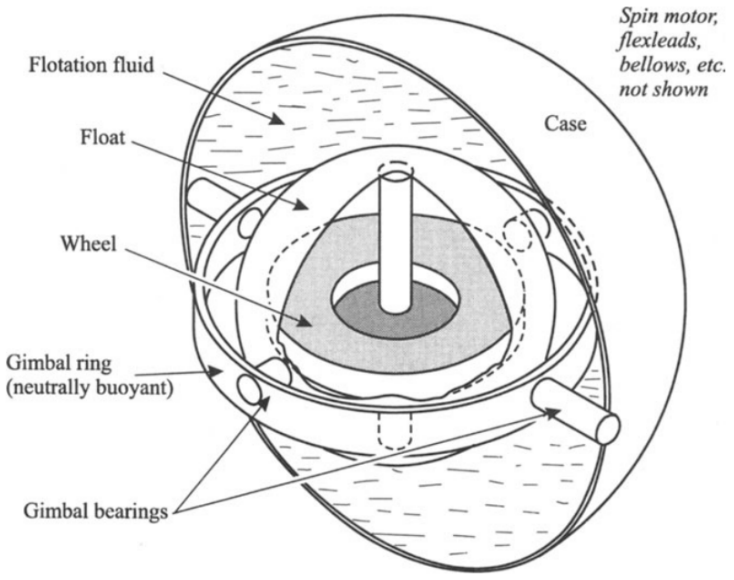

Figure 2.10: Floated two-axis gyro schematic [3].

system, which can realise one-axis redundancy. To ensure accuracy, the float has a small working angle range along the two orthogonal axes.

2.2.4 Gyroscope Calibration

There have been a number of studies on the methods and accuracy of gyroscope calibration (Figure 2.11). The conventional calibration method uses a turntable to provide a standard rotation speed for the gyroscope [46]. This method provides high calibration accuracy, but requires expensive and precise instruments and complex calibration procedures that preclude its use in consumer electronic devices or clinical settings. Several recent studies [4–6, 47] have proposed calibration methods that do not require precision equipment. Specifically, in [4], a camera-aided calibration method 2.12 was reported in which images provide orientation and position information about the sensor, resulting in a high computational complexity. In a separate method [5], a homogeneous magnetic field is employed as the calibration reference. The natural geomagnetic field is very weak and easily affected by the alternating electric field, making it difficult to implement outside the laboratory. In [6, 47], an accelerometer-aided gyroscope calibration method

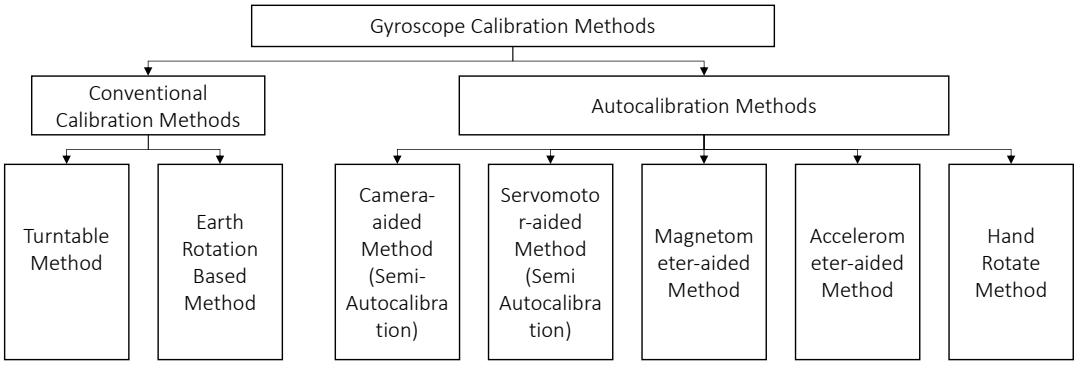

Figure 2.11: Taxonomy of existing gyroscope calibration methods.

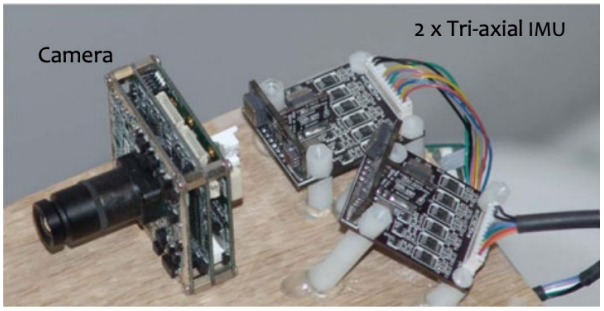

Figure 2.12: Camera-aided gyroscope calibration method.[4]

was presented. The accelerometer is first calibrated using the multi-position method, then the rotation speed of the sensor body is determined by the accelerometer. In this method, the calibration error of the accelerometer affects the calibration accuracy of the gyroscope. Additionally, the calibration process takes more than 10 min. The above discussion indicates that current gyroscope calibration methods rely on external equipment and are not suitable for scenarios where no external calibration device is available, such as in the field or in a clinical setting akin to a busy and chaotic hospital birthing room. This thesis proposes a fast in-field autocalibration method for triaxial gyroscopes that does not require any external devices to verify the IMU data collected from field-based contexts.

To facilitate frequent calibration, the calibration efficiency should be carefully examined. However, the majority of previous research [4–6, 46, 47, 74] has not explored the optimal experimental scheme. Ye et al. [75] reported a design of experiment (DoE) for

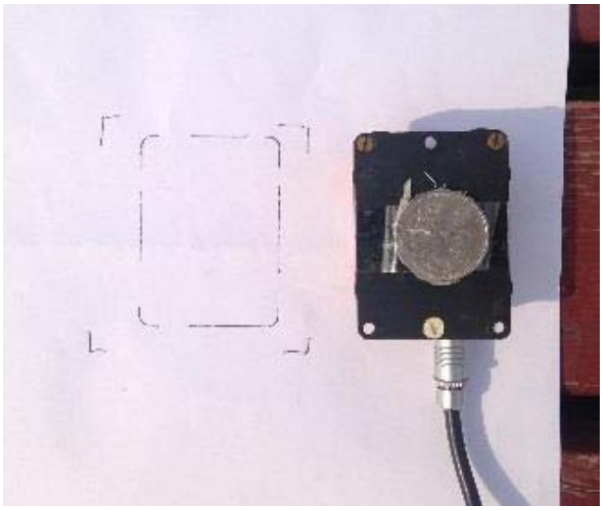

Figure 2.13: Magnetometer-aided gyroscope calibration method.[5]

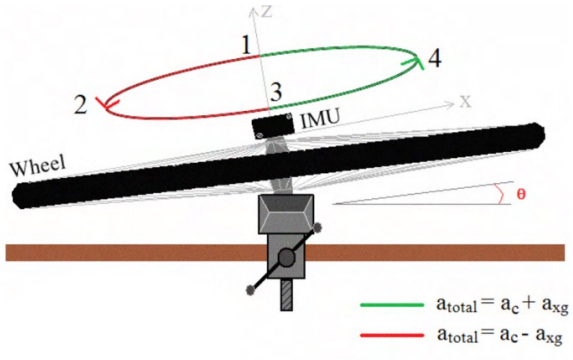

Figure 2.14: Accelerometer-aided gyroscope calibration method.[6]

accelerometer calibration. The motivation for this DoE is to minimise the number of trials required to obtain sufficient calibration information. Nevertheless, there is currently no similar DoE in the field of gyroscope autocalibration. One of the most significant challenges for such DoE is that the calibration model is highly nonlinear. Thus, in this thesis, we first describe a linearised six-parameter gyroscope calibration model prior to proposing a G-optimal DoE for the recommended model.

2.3 Review of Model Compression Methods for Pattern Recognition Algorithms

A significant disadvantage of deep neural networks is their computational load. This largely hinders the commercialisation of deep learning-based methods, especially on some edge devices. Most edge devices are not designed for computing-intensive tasks. If the neural network models are simply deployed, power consumption and latency will become problems. Even on the server side, more computation directly leads to increased costs. Researchers are trying to overcome this problem from various angles, such as specially designed neural network acceleration chips, which use dedicated hardware acceleration for a given computing task. Another idea is to consider whether all calculations in the model are necessary. If not, the model could be simplified to reduce the computation and memory footprint. This thesis focuses on a software method called model compression. Specifically, model compression can be divided into various techniques such as pruning, quantisation, low-rank factorisation, and knowledge distillation. In this thesis, we limit our attention to pruning methods.

2.3.1 Pruning Target

2.3.1.1 Parameter-level Pruning

The number of parameters and floating point operations per second (FLOPS) are widely used as metrics for evaluating the cost of a neural network. Unstructured pruning directly removes some of these parameters, thereby reducing the number of parameters that must be determined in the model. This method has been considered in several studies. Parameter pruning is the most widely used pruning method in the literature and is seen as the default framework when pruning is required.

The direct pruning of parameters has many advantages. First, it is very simple. In most deep learning frameworks, all network parameters can be easily accessed, which makes this technique very simple to implement. Additionally, parameters are the

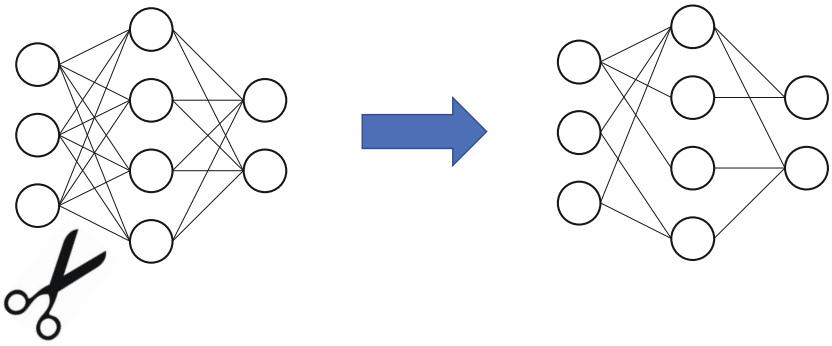

Figure 2.15: Overview of neural network pruning.

most fundamental elements of the network and most networks are over-parameterised. Thus, such networks can be heavily pruned without affecting their performance. This granularity enables pruning in very fine-grained modes, e.g. pruning of parameters within convolution kernels. As the pruning of weights is not limited by any constraints and does not modify the structure of the networks, this method is called unstructured pruning. However, one fatal disadvantage of this approach is that most deep learning frameworks and hardware cannot accelerate the computation of sparse matrices, which means that no matter how many parameters have been masked, there is no substantial impact on the actual training cost.

2.3.1.2 Filter-level Pruning

Structured pruning focuses on pruning larger structures, such as entire neurons or convolutional filters (kernels) in deep convolutional networks. Deep networks tend to contain many convolutional layers, consisting of hundreds or thousands of filters. This makes fine-grained pruning of convolutional layer filters possible. Removing this structure not only makes the layer structure of the deep neural network sparser, but also removes the feature map of the filter output.

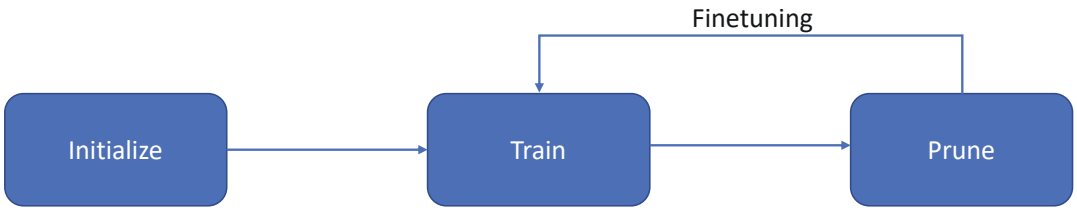

Figure 2.16: Typical pruning method flowchart.

2.3.2 Pruning Methods

Pruning is a classic method for reducing the number of model parameters and the computational load. With the rise of deep learning and the large-scale application of convolutional neural networks in the field of image classification, various pruning methods have emerged. Although there are various approaches, the core idea is to prune the structure of the neural network. Pruning includes three main tasks: training, pruning, and fine-tuning, as shown in Figure 2.16.

In the pruning process, the training process only needs to be performed once. The purpose of training is to obtain the original model trained on a specific task for the pruning algorithm. The most important part of pruning is to evaluate the importance of the network structure. This evaluation process is one of the main differences between the various pruning algorithms. The evaluated model structure mainly includes a filter, parameters, and other structures. Evaluating the importance of the network structure can be divided into two methods: network parameter-driven evaluation and data-driven evaluation. The network parameter-driven method uses the parameter information of the model itself to measure the importance of the model structure. For example, with L_1 or L_2 regularisation of parameters, the evaluation process does not depend on the input data. The data-driven method evaluates the importance of the network structure using the training data, such as by evaluating the importance of the filter by counting the number of 0 values after the output of the filter has passed through the activation layer. Fine-tuning is necessary to restore the expressiveness of models affected by pruning operations. Structural pruning adjusts the original model structure. Therefore, although the remaining model parameters are unchanged, the modified model structure may affect

the expression ability of the pruned model to some extent. The fine-tuning process can restore the expressive ability of the sub-model by adjusting the pruned sub-model on the training set. In the re-pruning process, the fine-tuned sub-model is sent to the pruning module. The model structure evaluation and pruning process are then performed again. Through the re-pruning process, each pruning operation is applied to a model with better performance, and the pruning model is continuously optimised until the model satisfies the pruning target requirements.

The above-mentioned process is the mainstream pruning algorithm [76–81]. On the basis of this pruning method, some related studies have improved the standard pruning process [82, 83]. In particular, [82] integrates the pruning process into fine-tuning process, removing the distinction between fine-tuning and pruning. A new trainable network layer is added for the pruning process. The network layer generates a binary code, and the network structure corresponding to the 0 values in the binary code is pruned. In [83], the importance of each network structure is measured by calculating the Kullback–Leibler (KL) divergence between the original model and the sub-model with the corresponding network structure removed. This method ensures that the network structure evaluation is not limited to local features or parameters, but uses global features to give more accurate evaluation results. Therefore, it achieves a good pruning effect without the need for a re-pruning process.

CHAPTER 3

TRIAXIAL GYROSCOPE CALIBRATION VIA SERVOMOTOR

3.1 Introduction

To eliminate the error caused by sensor inaccuracy, we intuitively think to use the calibration methods. In particular, the conventional calibration method uses a turntable to provide a standard rotation speed for the gyroscope. This method can provide high calibration accuracy, but requires expensive and precise instruments, and complex calibration procedures that preclude its use in consumer electronic devices or clinical settings. Thus, in this chapter, we proposed a servomotor-aided autocalibration method, which only needs sample calibration equipment.

The Micro-electro-mechanical-system (MEMS) triaxial gyroscopes are commonly used devices for measuring angular velocity in a broad range of applications, such as indoor pedestrian positioning [84], health monitoring [85], and consumer electronic devices [86, 87]. Such low-cost gyroscopes usually do not show high precision, due to the accumulation of drift error from integration when calculating the attitude [88]. In addition, the low repeatability and instability of the gyroscope cause changes in the scale factor and biases on every boot or under different environmental conditions, such as temperature variation [89]. Therefore, the gyroscope needs to be calibrated before

each use or when the environmental conditions change. Simple and efficient calibration methods are required for frequent calibration to be practical.

The issue of gyroscope calibration has received considerable attention. The ordinary triaxial gyroscope calibration method involves rotating the gyroscope at known angular velocity [46]. This approach can achieve high calibration accuracy, but the complex calibration procedure and requirement for expensive equipment make it unsuitable for use outside the laboratory. Gyroscope calibration methods that do not require precise rotation velocity measurements are presented in [4–6, 47]. In [4], a camera is employed to provide position and orientation information for gyroscope calibration. This technique first requires alignment of the body frame and the image frame, and calibration of the camera to achieve high accuracy. As images are involved, the computational complexity is significantly increased. In [5], a magnetometer-aided calibration method is investigated. The gyroscope calibration reference is provided by a homogeneous magnetic field. However, a weak magnetic field (e.g., the local magnetic field) can be easily disturbed by external alternating magnetic fields such as power lines. Therefore, this method may not be suitable for certain in-field applications. In [6, 47], accelerometer-aided gyroscope calibration methods are proposed. The accelerometer is first calibrated by gravity using the multi-position method, which then provides the rotation speed of the gyroscope. The entire calibration procedure takes more than 10 minutes, which is cumbersome in practical operation. Besides, the error caused by the accelerometer may be superimposed on the gyroscope parameters. Therefore, there is a great need to find a fast and simple in-field calibration method for gyroscopes. This chapter proposes a servomotor-aided gyroscope calibration method, which does not require high precision equipment and is easy to implement outside the laboratory.

To meet the requirements of frequent calibration, improving the calibration efficiency is of primary importance. Most previous studies [4–6, 46, 47, 74] paid little attention to investigating the selection of optimal experimental design. Recently, a six-position accelerometer calibration experimental design (DoE) was proposed [75]. The purpose of DoE is to obtain sufficient information for calibrating the accelerometer using a minimum

number of experiments. To the best of our knowledge, this is the first study to apply such DoE to gyroscope calibration. In this study, we propose a G-optimal DoE for gyroscope calibration, minimizing the maximum estimation variance over the entire measurement range.

For solving the regression problem, [47, 74, 90] applied nonlinear estimation methods, such as the Nelder-Mead method[91] and Levenberg-Marquardt algorithm [92]. These methods typically require sizeable computational power, which are difficult to apply for gyroscope calibration since a gyroscope is often part of an embedded system with limited resources and battery life. Therefore, we propose a fast converging recursive linear least square estimation for the six-parameter gyroscope calibration model.

We summarise the contributions of this chapter as follows. First, we propose an efficient servomotor-aided gyroscope calibration method, which does not require the use of high precision equipment during calibration. Second, we propose a six-point G-optimal experiment for gyroscope calibration. The proposed DoE can significantly shorten the calibration time and has been empirically validated. Last, we implement a fast converging recursive linear least square estimation method to reduce the computational complexity, which makes the calibration process more adaptable for an embedded environment.

This chapter is organized as follows. In section 3.2, we discuss the proposed calibration method and the DoE. In Section 3.3, we validate the approach using simulations under different conditions. In Section 3.4, we demonstrate the implementation of the proposed method on two commonly used gyroscopes. Section 3.5 concludes this chapter.

3.2 Calibration methodology

3.2.1 Efficient Calibration Method for Triaxial Gyroscope

Various factors contribute to error in a gyroscope. In this study, scale factors and biases are considered as error sources. Thus, a six-parameter calibration model is employed to define the unknown parameters. The relation between the actual angular velocity $\mathbf{G}_i = [g_{x,i}, g_{y,i}, g_{z,i}]^T$ and the measured angular velocity $\mathbf{M}_i = [m_{x,i}, m_{y,i}, m_{z,i}]^T$ at the i

th observation is described as:

$$\begin{bmatrix} g_{x,i} \\ g_{y,i} \\ g_{z,i} \end{bmatrix} = \begin{bmatrix} k_x & 0 & 0 \\ 0 & k_y & 0 \\ 0 & 0 & k_z \end{bmatrix} \left(\begin{bmatrix} m_{x,i} \\ m_{y,i} \\ m_{z,i} \end{bmatrix} + \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} \right) \tag{3.1}$$

where k_x, k_y, k_z and b_x, b_y, b_z stand for the scale factor and the bias of each axis, respectively.

This method is based on the idea that the measurement of the gyroscope should be equal to the rotation speed, that is

$$\omega_i = \sqrt{g_{x,i}^2 + g_{y,i}^2 + g_{z,i}^2} \tag{3.2}$$

where ω_i is the rotation speed.

We can expand Eq.(3.2) and square both sides of the equation. Then, we have:

$$\begin{aligned} \omega_i^2 = {} & k_x^2 m_{x,i}^2 + k_y^2 m_{y,i}^2 + k_z^2 m_{z,i}^2 \\ & + 2k_x^2 b_x m_{x,i} + 2k_y^2 b_y m_{y,i} + 2k_z^2 b_z m_{z,i} \\ & + \sum_{j=x,y,z} k_j^2 b_j^2 + \epsilon_i . \end{aligned} \tag{3.3}$$

The error term ϵ_i is a combination of a Gaussian and a noncentral Chi-squared noise. Similar to [93], the Chi-squared noise term can be ignored when the rotating speed is high. Thus, in this study, we consider the ϵ_i as a Gaussian noise. If we let

$$\begin{cases} \beta_0 = \sum_{j=x,y,z} k_j^2 b_j^2 \\ \beta_1 = k_x^2 \\ \beta_2 = k_y^2 \\ \beta_3 = k_z^2 \\ \beta_4 = 2k_x^2 b_x \\ \beta_5 = 2k_y^2 b_y \\ \beta_6 = 2k_z^2 b_z \end{cases} \quad \begin{cases} x_{1,i} = m_{x,i}^2 \\ x_{2,i} = m_{y,i}^2 \\ x_{3,i} = m_{z,i}^2 \\ x_{4,i} = m_{x,i} \\ x_{5,i} = m_{y,i} \\ x_{6,i} = m_{z,i} \end{cases} \quad y_i = \omega_i^2,$$

then the gyroscope calibration problem becomes

$$\begin{aligned} y_i =& \beta_0 + \beta_1 x_{1,i} + \beta_2 x_{2,i} + \beta_3 x_{3,i} \\ &+ \beta_4 x_{4,i} + \beta_5 x_{5,i} + \beta_6 x_{6,i} + \epsilon_i, \end{aligned} \tag{3.4}$$

which is a linear regression problem. The cost function of this problem is defined as:

$$J = \sum_{i=1}^{n} (||y_i - y_{actual,i}||), \tag{3.5}$$

where y_{actual} is the squared rotation speed provided by the servomotor. However, there is no close form solution for this problem as the parameter β_0 can be represented by the remaining six parameters $\beta_0 = \frac{\beta_4^2}{4\beta_1} + \frac{\beta_5^2}{4\beta_2} + \frac{\beta_6^2}{4\beta_3}$. The representation introduces the nonlinearity to Eq.(3.4). Thus, the question can be solved using a nonlinear regression technique such as the Levenberg-Marquardt algorithm [92]. As mentioned above, a gyroscope is often part of an embedded system with limited resources. To reduce the computational complexity, a novel iterative least square method [75] is employed to estimate the parameters.

We can reform Eq.(3.4) in matrix form as:

$$Y = X\boldsymbol{\beta} + \boldsymbol{\beta_0} + \boldsymbol{\epsilon}. \tag{3.6}$$

The observation matrix $X \in \mathbb{R}^{6\times 6}$ consists of the measured angular velocity:

$$X = \begin{bmatrix} m_{x,1} & m_{y,1} & m_{z,1} & m_{x,1}^2 & m_{y,1}^2 & m_{z,1}^2 \\ m_{x,2} & m_{y,2} & m_{z,2} & m_{x,2}^2 & m_{y,2}^2 & m_{z,2}^2 \\ m_{x,3} & m_{y,3} & m_{z,3} & m_{x,3}^2 & m_{y,3}^2 & m_{z,3}^2 \\ m_{x,4} & m_{y,4} & m_{z,4} & m_{x,4}^2 & m_{y,4}^2 & m_{z,4}^2 \\ m_{x,5} & m_{y,5} & m_{z,5} & m_{x,5}^2 & m_{y,5}^2 & m_{z,5}^2 \\ m_{x,6} & m_{y,6} & m_{z,6} & m_{x,6}^2 & m_{y,6}^2 & m_{z,6}^2 \end{bmatrix}. \tag{3.7}$$

The response matrix $Y = [y_{actual,1}, y_{actual,2}, \ldots, y_{actual,6}]^T$, parameters $\boldsymbol{\beta} = [\beta_1, \beta_2, \ldots, \beta_6]^T$, bias term $\boldsymbol{\beta_0} \in \mathbb{R}^{1\times 6} = [\beta_0, \beta_0, \ldots, \beta_0]^T$, and noise term $\boldsymbol{\epsilon} = [\epsilon_1, \epsilon_2, \ldots, \epsilon_6]^T$.

For solving Eq.(3.6), the fast converging iterative least square method is summarized in Algorithm 1. The convergent condition is given as $0 \leq \beta_0 < 0.5$ [75]. For a low cost

Algorithm 1 Iterative least square method

Ensure: Estimated scale factors $[k_x, k_y, k_z]$ and bias $[b_x, b_y, b_z]$.

1: Set initial value $\boldsymbol{\beta_0}^{(0)} = [0,0,0,0,0,0]$;
2: Calculate initial estimation $\boldsymbol{\beta}^{(1)} = (X^T X)^{-1} X^T (Y - \boldsymbol{\beta_0}^{(0)})$;
3: **while** $\sum_{j=1}^{6} ||{\beta_j}^{(n+1)} - {\beta_j}^{(n)}|| > 10^{-6}$ **do**
4: Update $\boldsymbol{\beta_0}$ at nth iteration as follows:
5: $\gamma^{(n)} = \frac{\beta_4^{2(n)}}{4{\beta_1}^{(n)}} + \frac{\beta_5^{2(n)}}{4{\beta_2}^{(n)}} + \frac{\beta_6^{2(n)}}{4{\beta_3}^{(n)}}$;
6: $\boldsymbol{\beta_0}^{(n)} = [\gamma^{(n)}, \gamma^{(n)}, \gamma^{(n)}, \gamma^{(n)}, \gamma^{(n)}, \gamma^{(n)}]$;
7: Update $\boldsymbol{\beta}$ at nth iteration as follows:
8: $\boldsymbol{\beta}^{(n+1)} = (X^T X)^{-1} X^T (Y - \boldsymbol{\beta_0}^{(n)})$.
9: **end while**
10: **return** Scale factors and bias terms:

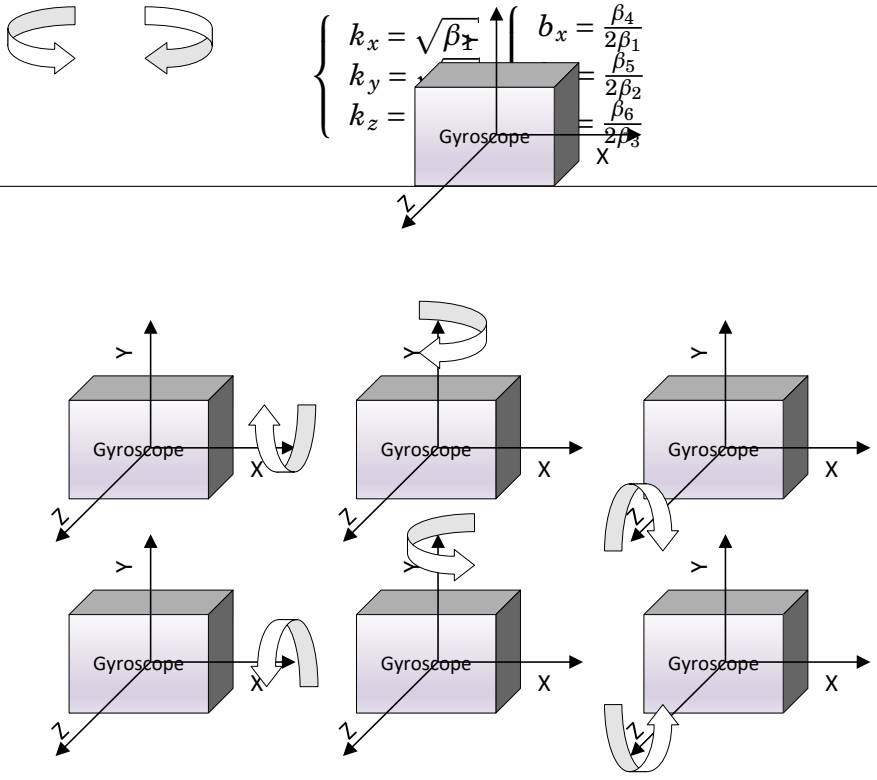

Figure 3.1: Six-observations rotation protocol for gyroscope calibration. The gyroscope is rotated at constant speed clockwise and counterclockwise along the x,y,z axis. [7]

MEMS gyroscope, the scale factor is usually within the range of $[0.8, 1.2]$. The bias term is usually between $\pm 0.1 rad/s$. Recall that $\beta_0 = \sum_{j=x,y,z} k_j^2 b_j^2$. Obviously, the convergence condition is met.

3.2.2 G-Optimal Experimental Design

Unlike an accelerometer, it is difficult for a low-cost MEMS gyroscope to perform autocalibration using the Earth's rotation. The Earth rotates at a moderate angular velocity of $7.29 \times 10^{-5} rad/s$, which is much lower than the bias term of the gyroscope. Hence, this study employs a servomotor as an external device. Considering its working principle, a servomotor may have vibrations during operation, but the time taken per revolution is still highly accurate. Owing to the linearity of Eq.(3.4), we can take the average of both sides of the equation during each revolution and consider it as one observation:

$$\begin{aligned}\frac{1}{N}\sum_{j=1}^{N} y_{i,j} = \frac{1}{N}\sum_{j=1}^{N}(&\beta_0 + \beta_1 x_{1,i,j} + \beta_2 x_{2,i,j} + \beta_3 x_{3,i,j} \\ &+ \beta_4 x_{4,i,j} + \beta_5 x_{5,i,j} + \beta_6 x_{6,i,j} + \epsilon_i).\end{aligned} \tag{3.8}$$

In this case, we can minimize the influences of vibrations and random noise to the estimated parameter.

The linear regression problem includes the estimation of six parameters. Thus, at least six observations are required [94]. To minimize the maximum variance of the estimated parameters, we introduce a G-optimal design of a second-order three variables model Eq. (3.4) for gyroscope calibration experiments. As the measurement is limited by $m_{x,i}^2 + m_{y,i}^2 + m_{z,i}^2 \approx \omega_i^2$, the design region is spherical. For a six-observations experimental scheme, the G-optimal design matrix can be expressed as:

$$D = \begin{array}{c} \\ (1) \\ (2) \\ (3) \\ (4) \\ (5) \\ (6) \end{array} \begin{array}{c} \begin{array}{ccc} m_{x,i} & m_{y,i} & m_{z,i} \end{array} \\ \begin{bmatrix} 1 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -1 \end{bmatrix} \end{array}$$

Accordingly, the rotation method of the gyroscope is shown in Fig.3.1. We rotate the gyroscope 360 degrees clockwise and counterclockwise along the x, y, z axis at the speed

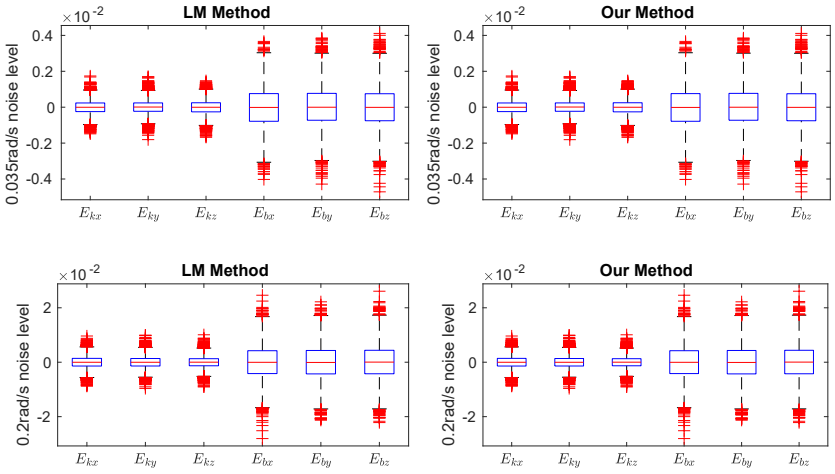

Figure 3.2: The simulation results of estimation error between estimated and actual parameters under normal conditions at different noise level using different method. Top: 0.035 rad/s noise levels. Bottom: 0.2 rad/s noise level. Left: Levenberg-Marquardt (LM) method. Right: Our proposed method. [7]

of ω, respectively. We average the data during each rotation and construct the 6-by-6 observation matrix according to Eq.(3.7). It is worth noting that no high-precision device is used to eliminate the alignment error. Once this six-observation matrix is constructed, the scale factors and bias terms can be calculated using Algorithm 1.

3.3 Simulation

With the intention of validating the feasibility of the proposed calibration method under different weights of scale factors, biases, rotation speed, orientation misalignment, and noise level, we first examined the proposed method using simulations. During each simulation, we generated a set of parameters under certain conditions, and these parameters were considered as the ground truth. Based on the actual value, the measurements of six observations were generated according to the experimental protocol in Fig.3.1. Then, the proposed method was employed to calculate the scales and biases based on the generated measurements, and the estimated parameters were stored.

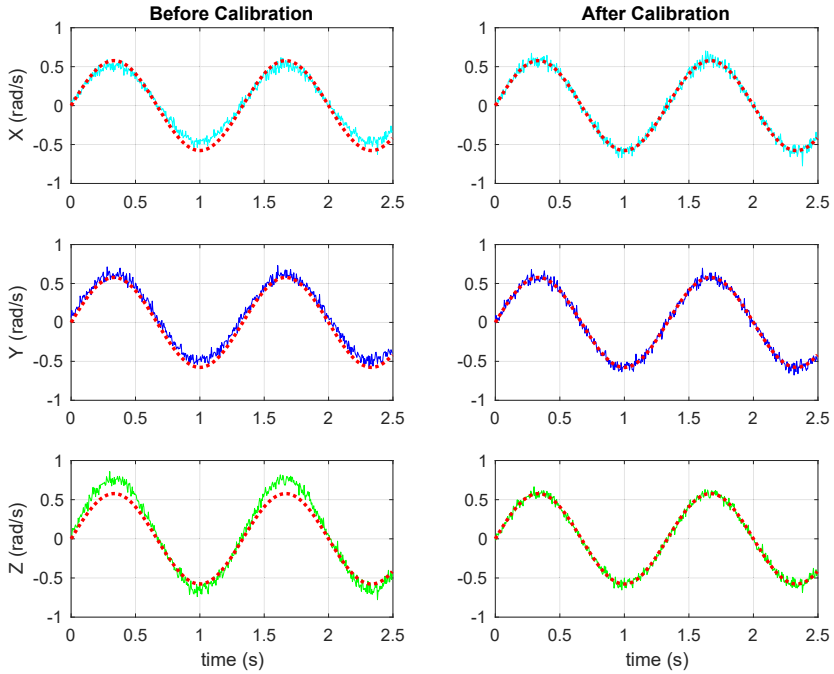

Figure 3.3: Simulation results of the desired rotation speed ω and gyroscope readings from three axes x, y, z before and after calibration. The dashed line indicates actual rotation on each axis, and the solid line represents gyroscope readings. [7]

3.3.1 Simulation Under Normal Conditions

We first conducted simulation tests under normal conditions. The simulation conditions are given based on the parameters of the commonly used gyroscopes. The following assumptions on the parameter are given, and the results are explained after them.

1. The scale factor follows a uniform distribution $U(80\%, 120\%)$ and the bias follows $U(-0.1rad/s, 0.1rad/s)$. The typical scale factors and biases of low-cost MEMS gyroscopes are usually within $\pm 20\%$ and $\pm 0.1rad/s$, respectively.

2. Misalignment on mounting follows $U(0\%, 10\%)$. In practice, without an accurate mounting platform, it is difficult for users to make measurements in the exact position specified by the experimental protocol. To demonstrate the robustness of the proposed method, we run the simulations with mounting misalignment.

3. The measurement noise is assumed to follow a Gaussian distribution with zero

mean and two different variances, $\epsilon_1 \sim \mathcal{N}(0, 0.035)$ and $\epsilon_2 \sim \mathcal{N}(0, 0.2)$. The typical noise spectral density of the MEMS gyroscope is between $1.74 \sim 6.11 \times 10^{-4} rad/s/\sqrt{Hz}$. As this study uses a 200 Hz sampling rate, the range of noise amplitude is around $0.035 \sim 0.18 rad/s$. Thus, we consider the noise vibration as 0.035 and 0.2 rad/s.

4. The variance of rotation noise is 5% of the current speed, which follows $\mathcal{N}(0, 5\%\omega)$. We use this term to simulate vibration during operation.

Based on the assumptions, we generated 30 sets of scale factors and biases to simulate different gyroscopes. For each set of parameters, we repeated the simulations 500 times. For each simulation, we took a six-observation measurement according to the experimental protocol shown in Fig.3.1, and constructed a 6-by-6 observation matrix based on Eq.(3.7). Subsequently, Algorithm 1 was implemented to estimate the scale factors and biases. Overall, 15,000 simulations were generated for testing our proposed calibration method.

To evaluate the performance of the proposed calibration method, we calculated the differences between the actual parameters and the estimated scale factors and biases. Box plots were used to analyze the differences as shown in Fig.3.2. The median values of the estimation error were 0, and the results indicated that the estimated parameters were unbiased. The majority of estimations of scale factors had an error within $\pm 9.3 \times 10^{-4}$ for the $0.035 rad/s$ noise level and $\pm 5.4 \times 10^{-3}$ for the $0.2 rad/s$ noise level. The estimation error of biases is higher, which is $\pm 3.0 \times 10^{-3}$ for the $0.035 rad/s$ noise level and $\pm 1.7 \times 10^{-2}$ for the $0.2 rad/s$ noise level. This indicated that the estimation accuracy was related to the measurement noise level. Better gyroscopes with lower noise have lower estimation error. In addition, we compared Algorithm 1 with the Levenberg-Marquardt method. The results indicated that the error of these two methods were identical. Interestingly, the scale factors had a much lower estimation error than the bias terms. This phenomenon can be explained by sensitivity analysis techniques [95]. In this particular model Eq.(3.4) and experiment design, the observability of scale factors is much higher than that of bias terms, which leads to better estimation results for the former.

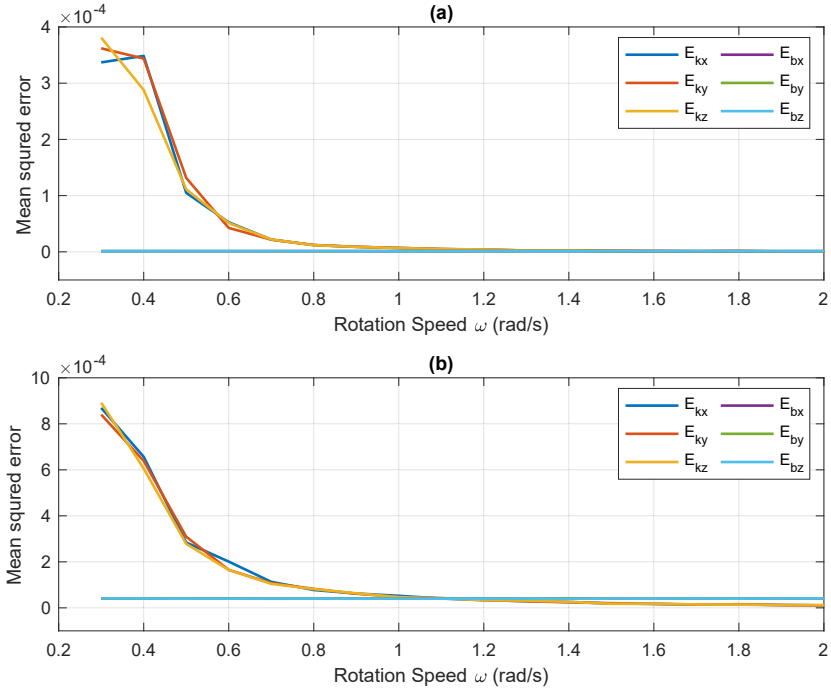

Figure 3.4: The mean squared error (MSE) between estimated and actual parameters at different rotation speeds during calibration with different measurement noise levels. (a) 0.035 rad/s noise level. (b) 0.2 rad/s noise level. [7]

To intuitively demonstrate the effectiveness of the calibration, we performed a simulation to compare the gyroscope readings before and after calibration. The desired rotation speed of the servomotor ω with respect to time was set as a sine wave with an amplitude of $1 rad/s$ and frequency of $0.75 Hz$. The three axes of the gyroscope were mounted to be equidistant from the rotation axis. Thus, the projection of the rotation speed to each axis was equal. The rotation noise variance was set to be 5% of the current speed, and the measurement noise followed $\epsilon \sim \mathcal{N}(0, 0.035)$. We randomly generated a set of parameters and use the proposed approach to estimate the scale factors and bias terms. The actual parameters $[k_x, k_y, k_z, b_x, b_y, b_z]$ and estimated parameters $[\hat{k_x}, \hat{k_y}, \hat{k_z}, \hat{b_x}, \hat{b_y}, \hat{b_z}]$ are as

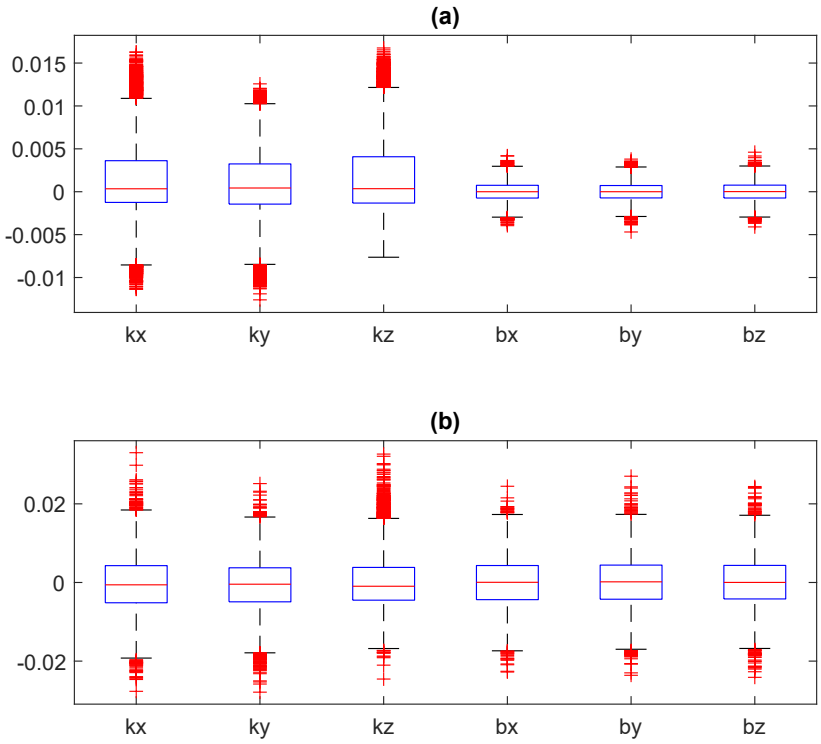

Figure 3.5: The simulation results of estimation error between estimated and actual parameters under extreme conditions at different noise level using different method. Top: 0.035 rad/s noise levels. Bottom: 0.2 rad/s noise level. [7]

follows:

$$\begin{cases} k_x = 0.9070 \\ k_y = 1.0501 \\ k_z = 0.8734 \\ b_x = 0.0528 \\ b_y = 0.0813 \\ b_z = -0.0992 \end{cases} \quad \begin{cases} \hat{k_x} = 0.9070 \\ \hat{k_y} = 1.0502 \\ \hat{k_z} = 0.8735 \\ \hat{b_x} = 0.0529 \\ \hat{b_y} = 0.0802 \\ \hat{b_z} = -0.0994 \end{cases}$$

Based on the estimated parameters, we corrected the gyroscope readings using Eq.(3.1). Fig.3.3 demonstrates that after calibration, the measured and actual values showed better fit. The fluctuations were caused by measurement noise and motor speed instability.

3.3.2 The Effect of Rotation Speed

Next, we explored the influence of rotation speed on the estimation results during calibration. We followed the 15,000 simulations procedure described above, but used different rotation speeds. The rotation speed was set within the range of 0.3 to $3rad/s$ with a step size of $0.1rad/s$. Lower rotation speeds outside this range may be covered by noise,while higher rotation speeds cannot be accurately achieved by servomotors during the real experiment. The overall mean squared error (MSE) of six estimated parameters are defined as follows:

$$e_j = \frac{1}{N}\sum_{i=1}^{N}(j_i - \hat{j}_i)^2, j_i = k_{x,i}, k_{y,i}, k_{z,i}, b_{x,i}, b_{y,i}, b_{z,i}, \tag{3.9}$$

where N is the number of simulations. Fig.3.4 shows the influence of speed on the parameter estimation accuracy during calibration. As the speed increases, the average MSE decreases exponentially. After the speed rises to 1 rad/s, the average MSE value stops decreasing. The MSE of biases remains unchanged irrespective of changes in speed, and is only affected by the measurement noise level. This is because the observability of biases still exists even in a static state, i.e. $\omega = 0$. Appealingly, when the speed is less than 1 rad/s, the MSE of scale factors drops significantly as the speed increases. At low speeds, the measurement noise occupies most of the measured value rather than the projection of the rotation component on this axis. At this time, the signal to noise ratio (SNR) of the measured value is small. The lower the rotation speed, the smaller the SNR. At the same rotation speed, when comparing (a) and (b) in Fig.3.4, the estimation with high measurement noise has a larger MSE. As for the difference between axis, during machining of MEMS gyroscopes, the x and y axes are usually machined together, while the z axis is machined separately. So the x and y axes usually have similar accuracy, and the z axis has a different accuracy comparing with the x and y axes. As a result, the error of the z-axis also varies after calibration.

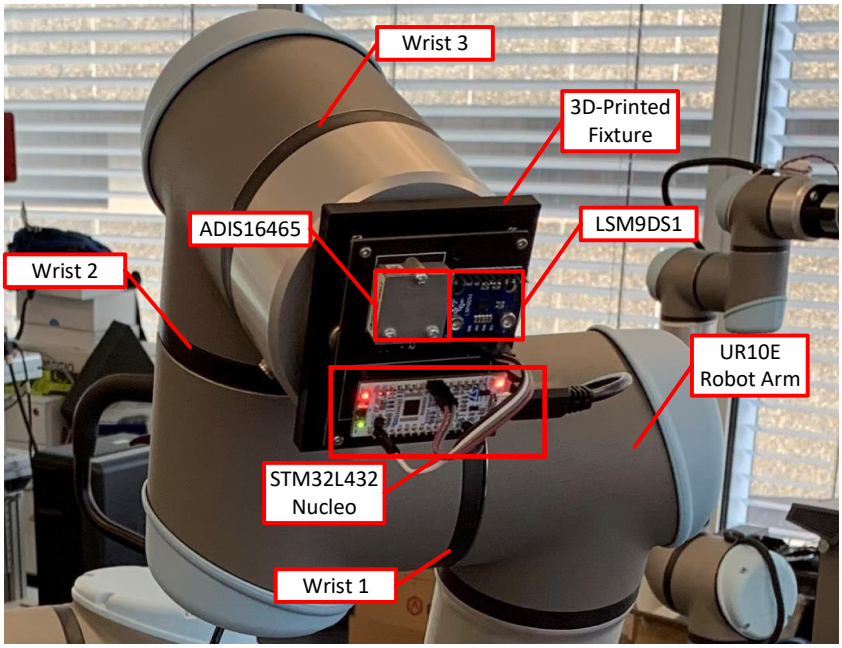

Figure 3.6: Experimental system for the gyroscope calibration on a robot arm UR10e. The part names and joint numbers are noted. [7]

Table 3.1: Convergence rate under different scale factors and biases

Number of iterations	k_x	k_y	k_z	b_x	b_y	b_z
1.High scale factor error						
Actual value	1.9074	1.9529	1.5635	0.0827	0.0265	-0.0805
1	1.9102	1.9571	1.5665	0.0822	0.0243	-0.0797
2-Converged	1.9071	1.9539	1.5640	0.0822	0.0243	-0.0797
2.High bias						
Actual value	1.0979	1.1052	0.9851	-0.1046	0.1995	0.1565
1	1.1029	1.1103	0.9900	-0.1057	0.1971	0.1545
2	1.0961	1.1035	0.9839	-0.1057	0.1971	0.1545
3-Converged	1.0962	1.1036	0.9840	-0.1057	0.1971	0.1545
3.High scale factor error and bias						
Actual value	1.5044	1.6494	1.5282	0.1483	-0.1282	0.1794
1	1.5173	1.6652	1.5423	0.1469	-0.1284	0.1802
2	1.5053	1.6521	1.5302	0.1469	-0.1285	0.1803
3-Converged	1.5055	1.6523	1.5304	0.1469	-0.1285	0.1803

3.3.3 Robustness of the Method under Extreme Conditions

To demonstrate the robustness of the proposed gyroscope calibration method, the quality of the gyroscope was assumed to be very poor. The randomly generated parameters followed $U(120\%, 200\%)$ for scale factors and $U(-0.2 \sim -0.1, 0.1 \sim 0.2)$ for biases. Other parameters followed the previous setting. The results shown in Fig.3.5 suggested that our proposed method could be applied to gyroscopes with poor manufacturing quality.

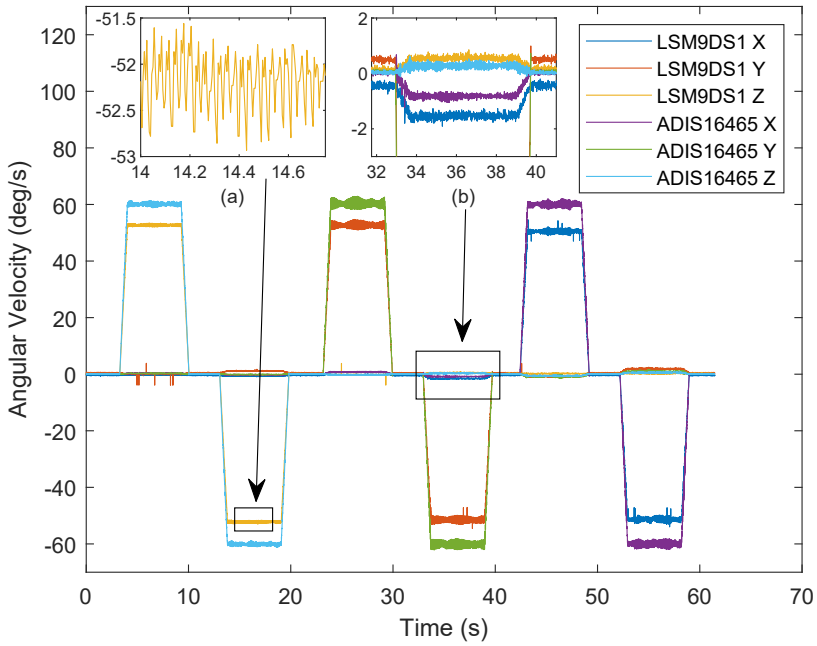

Figure 3.7: Raw gyroscope data from LSM9DS1, compared with ADIS16465 reading during calibration. (a) Periodic vibration was caused by the control strategy of the servo-motor. (b) The component on the non-rotating axis was caused by mounting misalignment. [7]

Although the errors were larger than under normal conditions, the majority of these errors were within $\pm 3 \times 10^{-3}$ for 0.035 rad/s noise level and $\pm 1.8 \times 10^{-2}$ for 0.2 rad/s noise level. Under extreme conditions, the scale factors had worse observability since larger scale factors enlarge the signal noise, thereby reducing the SNR.

To demonstrate the convergence rate of the iterative method, we performed three simulations (results shown in TABLE 3.1). The first simulation used high scale factors error and normal biases, while the second simulation used typical scale factors error and high biases. The third simulation used high scales factors error and high biases. The results indicated that less than three iterations were needed for the proposed calibration method.

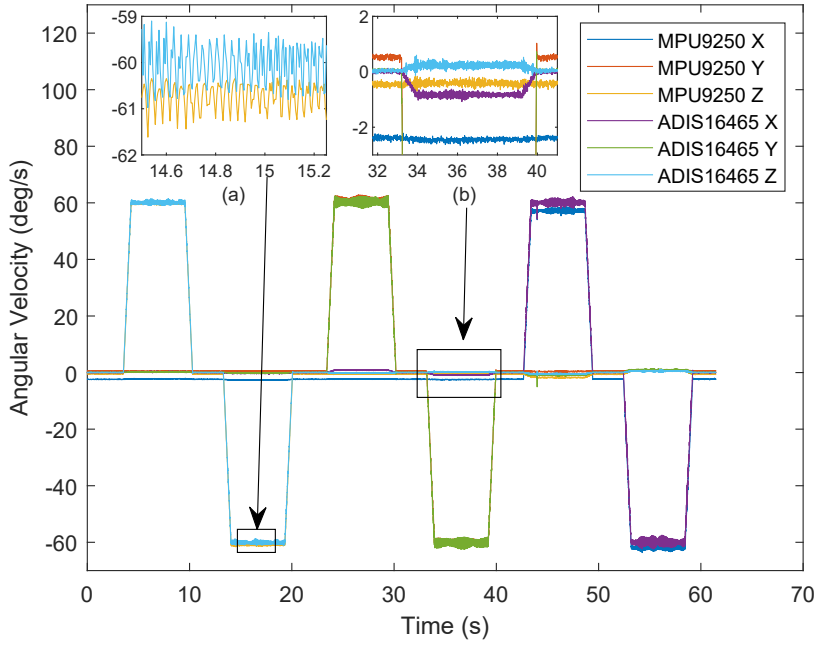

Figure 3.8: Raw gyroscope data from MPU9250, compared with ADIS16465 reading during calibration. (a) Periodic vibration was caused by the control strategy of the servomotor. (b) The component on the non-rotating axis was caused by mounting misalignment. [7]

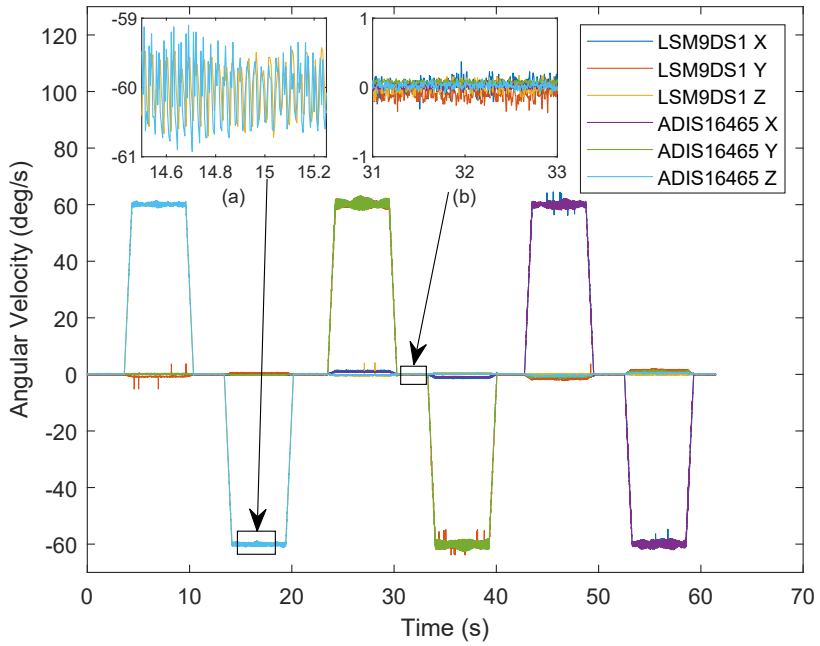

Figure 3.9: Calibrated gyroscope data from LSM9DS1, compared with ADIS16465 reading during the testing period. (a) The reading from LSM9DS1 and ADIS16465 nearly coincided with each other. (b) The biases of gyroscope reading were almost zero. [7]

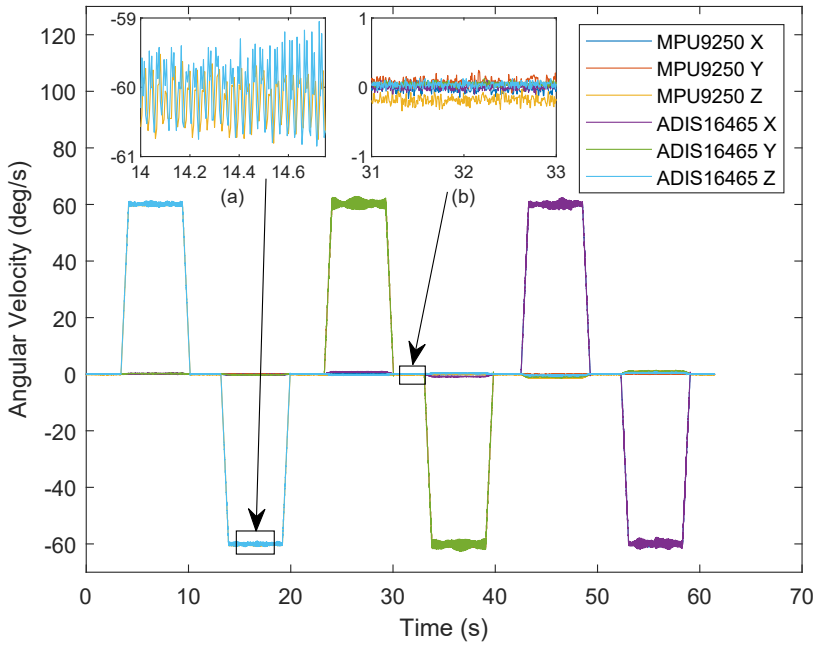

Figure 3.10: Calibrated gyroscope data from MPU9250, compared with ADIS16465 reading during the testing period. (a) The reading from MPU9250 and ADIS16465 nearly coincided with each other. (b) The biases of gyroscope reading were almost zero. [7]

3.4 Experiments

We calibrated and verified two commonly used low-cost MEMS gyroscopes: LSM9DS1 from STMicroelectronics and MPU9250 from TDK. We demonstrated the application of our proposed method on a UR10e robotic arm. Besides, our method is easy to implement on a 3-axis camera gimbal or a single-axis servomotor with proper adapter. The calibration system is shown in Fig.3.6. The gyroscope is replaceable. The digital signals were collected and calculated using an STM32L432 Nucleo board. It is worth noting that high-precision turntables or other calibration equipment were not needed in the proposed system. For the purpose of comparing the data quality of low-cost gyroscopes, an ADIS16465 sensor was mounted on the same board during calibration. The ADIS16465 was pre-calibrated using a turntable. The room temperature was set to 22 °C.

3.4.1 Calibration of Two Low-cost Gyroscopes

The proposed method was applied to the two gyroscopes according to the G-optimal experiment scheme shown in Fig.3.1. We summarise the calibration procedure as follows:

1. Mount the LSM9DS1 and ADIS16465 on the UR10e and turn on the system.
2. Rotate wrist 3 of the UR10e 360° clockwise. Based on the simulation results, find the balance between the rotation speed and measurement noise, and set the angular velocity to 60°/s. Then, wait for 3 seconds.
3. Rotate wrist 3 of the UR10e 360° counterclockwise. Wait for 3 seconds.
4. Repeat 2-3 for wrist 1 and wrist 2.
5. Repeat 1-4 for MPU9250.

The entire process takes about one minute. The data was recorded by the microcontroller and transferred to the computer via a serial port. After completing the calibration process on the microcontroller, the scale factors and biases were transferred to the computer. The raw data from LSM9DS1 and MPU9250 are shown in Fig.3.7 and Fig.3.8.

From the figures, we can see that the readings of the two sensors are quite different from that of ADIS16465. Since the ADIS16465 is a high-precision sensor and we have also pre-calibrated it, it can be inferred that the difference is caused by the scale factors and biases of the low-cost sensors. The servomotor control strategy leads to vibration during rotation, while the ADIS16465 reading suggests that the average angular velocity is relatively accurate. Since high-precision components were not used in the mounting process, the non-rotating axis also had a rotation component. We hence prove that the proposed method can work under conditions of vibration and misalignment.

To demonstrate the accuracy of the proposed method, we compared the calibration results with those obtained using the precision turntable method [46]. The results are shown in TABLE 3.2 and TABLE 3.3. For a fast calibration method without using any high-precision equipment, all errors less than 10^{-3} indicate a considerably accurate

Table 3.2: Comparison of LSM9DS1 calibration results

Parameter	Results of proposed calibration method	Results of conventional turntable method
k_x	1.1775	1.1771
k_y	1.1552	1.1554
k_z	1.1445	1.1440
$b_x(rad/s)$	0.0076	0.0077
$b_y(rad/s)$	-0.0103	-0.0100
$b_z(rad/s)$	-0.0042	-0.0051

Table 3.3: Comparison of MPU9250 calibration results

Parameter	Results of proposed calibration method	Results of conventional turntable method
k_x	1.0069	1.0065
k_y	0.9960	0.9955
k_z	0.9955	0.9950
$b_x(rad/s)$	0.0442	0.0411
$b_y(rad/s)$	-0.0089	-0.0099
$b_z(rad/s)$	0.0076	0.0101

Table 3.4: MSE between LSM9DS1 and ADIS16465

Axis	Error before calibration (rad/s)	Error after calibration (rad/s)
x	0.2798	0.0052
y	0.2255	0.0093
z	0.1940	0.0022

result. Besides, considering the low repeatability of the two gyroscopes used, the actual scale factors and biases when mounted on the robot arm may be different from those when mounted on the turntable.

After calibration, the parameters were stored in the microcontroller. To further test calibration effectiveness, we made the robot arm repeat the same movements as during the calibration procedure. Instead of raw data, the microcontroller sent the calibrated gyroscope readings to the computer. The results after calibration are shown in Fig.3.9

Table 3.5: MSE between MPU9250 and ADIS16465

Axis	Error before calibration (rad/s)	Error after calibration (rad/s)
x	0.1031	0.0051
y	0.0107	0.0057
z	0.0096	0.0033

and Fig.3.10. The reading from the pre-calibrated ADIS16465 was considered as a ground truth. We calculated the MSE error between the LSM9DS1/MPU9250 and ADIS16465. The results are shown in TABLE 3.4 and TABLE 3.5. Our experiments show that the gyroscope reading obtained after calibration using the proposed method is significantly more accurate than the reading before calibration.

3.5 Conclusion

This chapter proposed an efficient servomotor-aided calibration method that estimates the gain factors and biases of a triaxial gyroscope. A six-observation G-optimal experimental scheme was implemented for the calibration process, and a fast converging recursive linear least square estimation method was applied to reduce the computational complexity. We performed a series of simulations and experiments to evaluate the validity and feasiblity of the proposed method.

The simulation results indicated that the gyroscope parameters could be accurately estimated within three iterations, and demonstrated that the proposed method was robust under extreme conditions. Furthermore, the simulation results showed a balance between the rotation speed and measurement noise. The angular velocity in the experiment was set to 60°/s, accordingly.

We performed experiments on two commonly used low-cost MEMS gyroscopes. The outcomes of calibration using our proposed method and the conventional turntable method were experimentally compared. The results indicated that the errors between these two methods were less than 10^{-3}. To further test the performance of our proposed

method, we compared the calibrated reading of the two low-cost gyroscopes with a high-precision sensor. The results showed that the error was significantly decreased after calibration. More importantly, we demonstrated the possibility of implementing this method on low-precision motors such as a robot arm, as well as its applicability on a microprocessor. Using our proposed method, the entire calibration process only requires one minute, and high-precision calibration equipment is not necessary.

CHAPTER 4

TRIAXIAL GYROSCOPE CALIBRATION VIA MANUAL ROTATION

4.1 Introduction

In some in a filed or clinical setting, even simple external calibration device is not available. The autocalibration method proposed in the previous chapter was further simplified. In this chapter, we proposed an autocalibration method, which does not rely on any external devices and the calibration process can be finished within 30 seconds.

The gyroscope is essential equipment for measuring angular velocity in a wide range of technologies, such as motion tracking [96], indoor positioning [97], and wearable health monitoring [98–100]. For example, we have designed a low-cost micro-electromechanical (MEMS) inertial measurement unit (IMU) device to be worn by preganant women during childbirth to explore the impact of hospital birth room configuration upon mobility for women with complex pregnancies (See Fig. 4.1 and 4.2). However, a key concern of measurement in such ecological settings is that the accuracy of low-cost gyroscopes are usually low. For instance, when calculating the attitude, the integration will lead to

Figure 4.1: The designed wearable motion tracking device

Figure 4.2: The designed device using in the pilot study

the accumulation of drift error [88]. In addition, owing to the poor repeatability and significant volatility, on every booting or under different environmental conditions, the scale factor and biases change [89]. Therefore, it is necessary to calibrate the gyroscope on each initialization or when environmental conditions change. However, calibration processes are normally time consuming, and thus in field or clinical context, practitioners need frequent calibration of the gyroscope, which needs to be a simple and efficient process.

4.1.1 Preliminary Study

The development of the in-field wearable sensor calibration method is motivated by the transdisciplinary project 'The use of wearable technologies to explore the impact of hospital birth room configuration upon mobility in childbirth for women with complex pregnancies' under the support of the Faculty of Engineering and Information technology

Cross-Faculty Collaborative Scheme, at the University of Technology Sydney (UTS).

The aim of our study was to track, analyse and compare the mobilisation and physical positioning of women experiencing high risk pregnancy, during labour. This is important because being able to move freely and adopt a range of positions during labour is associated with reduced length of labour, decreased likelihood of caesarean section and epidural use, increased sense of choice and control and bodily autonomy, and decreased pain for women. Mobilisation during labour is expected for most women experiencing a low risk pregnancy. However, women experiencing a high risk pregnancy may be less expected to mobilise during labour, and thus may be offered less opportunity to move freely and adopt a range of positions. The experienced midwives on the research team have observed that mobilisation is rarely facilitated for women experiencing high risk pregnancies, who are usually expected to remain prone on the bed. Little is known about how women experiencing high risk pregnancies would mobilise if given opportunity to do so. It seems possible that the use of wireless fetal monitoring and a birth room that has appropriate space and equipment may facilitate mobilisation, and its associated benefits, for these women.

For the first time, we have produced data showing where and how women move around and utilise space in a (simulated) birthing environment, whilst they are undergoing continuous electronic fetal monitoring due to risk factors and medical complications. In future, the device can be used to compare efficacy and clinical impacts of different types of electronic fetal monitoring on freedom of movement, maternal and infant well-being. Future application of the device can fill a significant gap in health care design research, which has previously been unable to reliably report on user experience of birth room environments in maternity care settings.

In order to protect the privacy of potential participating birthing women in future studies, the device could not use the conventional video-camera based tracking system. It developed and pre-tested the motion tracking technology based on the integration of inertial sensors and wireless sensing. One of the subsystems is a wearable module that includes wireless IMU for gait monitoring and assisted behaviour recognition.

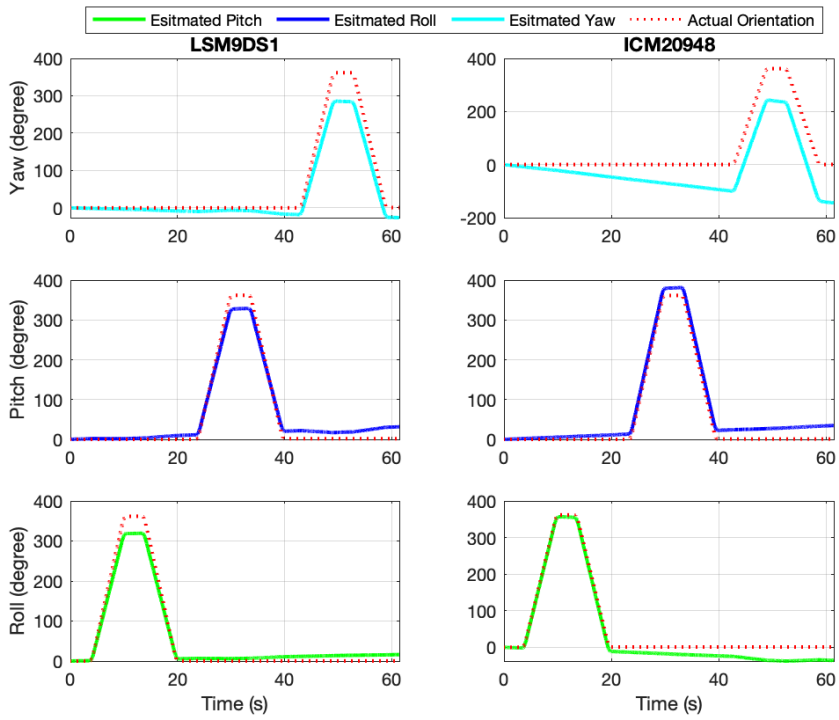

Figure 4.3: Estimated orientation using raw gyroscope readings before calibration from two models of gyroscope. Left: LSM9DS1. Right: ICM20948.

4.1.2 Pre-study

To test the accuracy of the developed portable IMU device, especially the gyroscope, we implemented the orientation angle estimation tests as shown in Fig. 4.5. The gyroscope to be tested is placed on a robotic arm, which rotates 360 degrees clockwise along the x-axis and then 360 degrees counterclockwise. We repeat this action on the y and z axes. Fig. 4.3 indicates that the angle estimation error is significant, which will potentially influence the gait estimation and gesture analyses.

To improve the estimation accuracy, we tried to develop in-field calibration methods for both accelerometers and gyroscopes. Despite the extensive amount of literature investigating in-field calibration of triaxial accelerometers [74, 75, 101, 102]; few studies discuss the calibration of the gyroscope under non-laboratory environments. Hence, in this study, we developed a new easy-to-use gyroscope calibration method, which can significantly improve the angle estimation accuracy as shown in the later discussions.

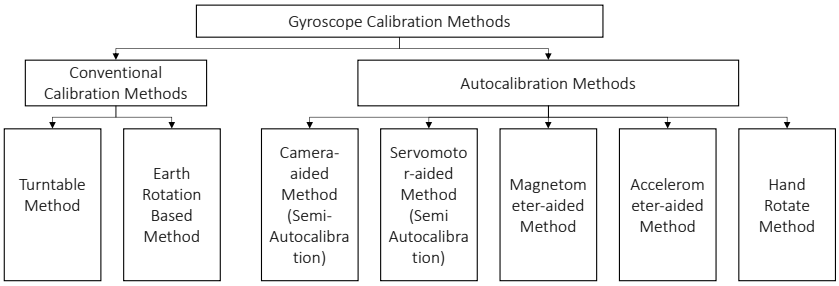

Figure 4.4: The taxonomy chart of existing gyroscope calibration methods.

4.1.3 Existing Calibration Methods

A considerable amount of literature exists on the methods and accuracy of gyroscope calibration (Fig. 4.4). In particular, the conventional calibration method uses a turntable to provide a standard rotation speed for the gyroscope [46]. This method can provide high calibration accuracy, but requires expensive and precise instruments and complex calibration procedures that preclude its use in consumer electronic devices or clinical settings. Several recent studies [4–6, 47] proposed several calibration methods that do not need precision equipment. Specifically, in [4], a camera-aided calibration method was reported. The images provide the orientation and position information of the sensor to confirm its orientation, resulting in a high computational complexity. In a separate method [5], an homogeneous magnetic field was employed as the calibration reference. The natural geomagnetic field is very weak and is easily affected by the alternating electric field, again making it difficult to implement outside the laboratory. In [6, 47], an accelerometer-aided gyroscope calibration method was presented. The accelerometer was first calibrated using the multi-position method. Then, the rotation speed of the sensor body is provided by the accelerometer. In this method, the calibration error of the accelerometer affects the calibration accuracy of the gyroscope. Besides, the calibration process takes more than ten minutes. From the above discussion, we can infer that these gyroscope calibration methods rely on external equipment are not suitable for scenarios where the external calibration device is not available, such as in a filed or clinical setting akin to a busy and chaotic hospital birthing room. This chapter proposed a fast in-field autocalibration method for triaxial gyroscopes without any external devices that would

assist provide veracity of collected IMU data from field-based contexts.

Intending to facilitate frequent calibration, the calibration efficiency deserves a careful examination. However, the majority of previous research [4–6, 46, 47, 74] did not dedicate enough time to explore the optimal experiment scheme. Ye et al. [75] reported the design of experiment (DoE) for accelerometer calibration. The motivation of DoE is that minimizing the number of trials to obtain sufficient calibration information. Nevertheless, currently there is no similar DoE in the field of gyroscope autocalibration. One of the most significant challenges in such DoE is that the calibration model is highly nonlinear. Thus, in this study, we first give a linearized six-parameter gyroscope calibration model prior to proposing a G-optimal DoE for the recommended model.

4.1.4 Summary of Our Contributions

The contributions of this chapter can be summarized as follows: (1) We propose a fast autocalibration method for triaxial gyroscopes. This calibration method is implemented in a micro-controller and only takes 30 seconds without using any external device. (2) We propose a G-optimal DoE scheme for a linearized autocalibration model. (3) We validated the effectiveness of the proposed calibration method in both numerical simulation and real-time experiments. (4) We designed a low-energy, cost-effective, and wearable wireless movement tracking device for health monitoring. We applied the proposed triaxial gyroscopes calibration approach in the wearable device and significantly improved angle estimation accuracy.

4.2 Methodology

One of the main obstacles stops gyroscope autocalibration is that the low-precision gyroscopes cannot use the earth's rotation as a reference for calibration, like accelerometer use gravity. This is because the angular velocity of the earth's rotation is often submerged in the noise of the gyroscope. In this section, we first propose an autocalibration method

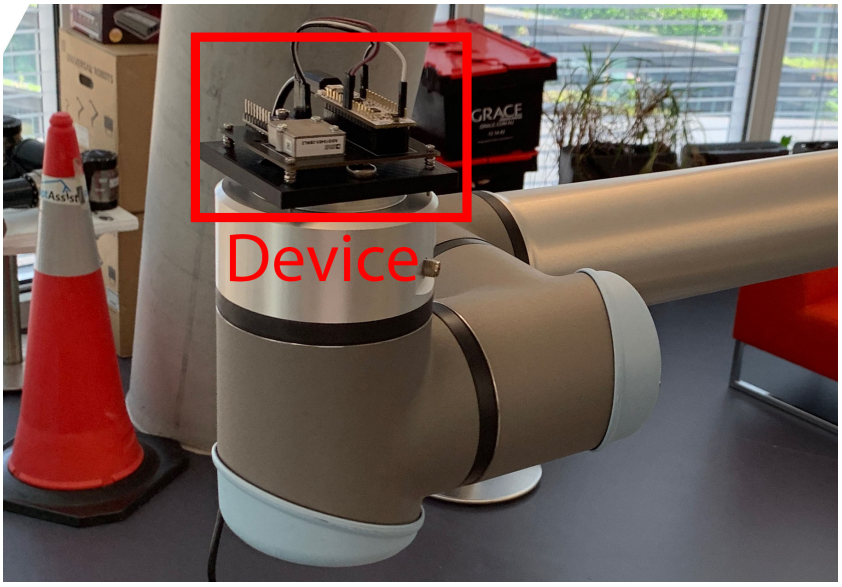

Figure 4.5: The orientation estimation tests. The IMU is mounted on a robot arm.

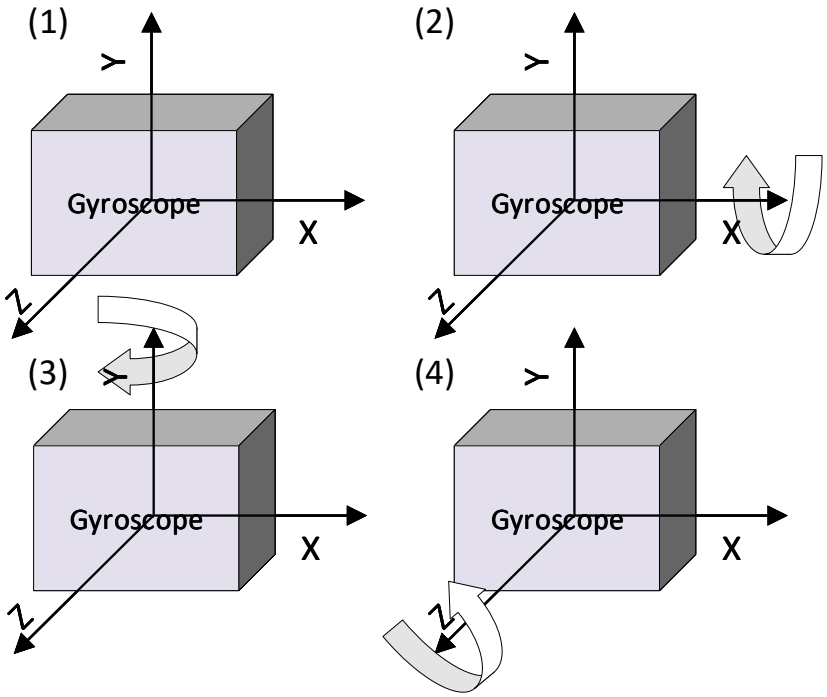

Figure 4.6: 4-observations rotation protocol for gyroscope calibration. (1) Stationary stage. (2)-(4) Rotating stage: Manually rotate the gyroscope 360 degrees clockwise along the x,y,z axis.

that uses manual rotation angle as a calibration reference. Then, a four-observation calibration method is reported.

4.2.1 Calibration principle

Numerous parameters contribute to the error in a gyroscope, among which the scale factor and biases have a significant influence. In this study, a 6-parameter calibration model is employed to estimate unknown parameters. Therefore, the relationship between the actual angular velocity components $g_{x,i}, g_{y,i}, g_{z,i}$ and the measured angular velocity $m_{x,i}, m_{y,i}, m_{z,i}$ at i th rotation are described as:

$$\begin{bmatrix} g_{x,i} \\ g_{y,i} \\ g_{z,i} \end{bmatrix} = \begin{bmatrix} k_x & 0 & 0 \\ 0 & k_y & 0 \\ 0 & 0 & k_z \end{bmatrix} \left(\begin{bmatrix} m_{x,i} \\ m_{y,i} \\ m_{z,i} \end{bmatrix} + \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} \right), \tag{4.1}$$

Where the k_x, k_y, k_z and b_x, b_y, b_z represents the scale factors and biases, respectfully.

The main idea of this method is Euler's theorem. Specifically, the total rotation vector of the gyroscope $\boldsymbol{\theta}_{total}$ should be equivalent to the sum of the rotation projections on each of these three axes $[\boldsymbol{\theta}_{x,i}, \boldsymbol{\theta}_{y,i}, \boldsymbol{\theta}_{z,i}]$.

$$\boldsymbol{\theta}_{total,i} = \boldsymbol{\theta}_{x,i} + \boldsymbol{\theta}_{y,i} + \boldsymbol{\theta}_{z,i}, \tag{4.2}$$

where i stands for the different rotations. We calculate the magnitude and square both sides of Eq. (4.2), and consider the rotation angle on each axis is equal to the integral of the angular velocity of this axis. Thus, for a discrete system, we have:

$$\boldsymbol{\theta}^2_{total,i} = \left(\sum_{j=1}^{N} g_{x,i,j} \right)^2 + \left(\sum_{j=1}^{N} g_{y,i,j} \right)^2 + \left(\sum_{j=1}^{N} g_{z,i,j} \right)^2, \tag{4.3}$$

where j is the number of the sample within one rotation and N is the total sample number in one rotation. If we substitute Eq. (4.1) into Eq. (4.3), we have:

$$\begin{aligned}\boldsymbol{\theta}_{total,i}^2 = k_x^2\left(\sum_{j=1}^{N}(m_{x,i,j}+b_x)\right)^2 + k_y^2\left(\sum_{j=1}^{N}(m_{y,i,j}+b_y)\right)^2 \\ + k_z^2\left(\sum_{j=1}^{N}(m_{z,i,j}+b_z)\right)^2 + \epsilon_i,\end{aligned} \tag{4.4}$$

where ϵ_i is the noise. Let the unknown parameters be represented by a vector $\boldsymbol{\alpha} = [k_x, k_y, k_z, b_x.b_y, b_z]$. Then, we can reformat the calibration problem into an optimization problem:

$$\min_{\alpha} J(\boldsymbol{\alpha}) = \min_{\alpha} \sum_{i=1}^{n}(||f_{\boldsymbol{\alpha}}(\boldsymbol{M}_i) - \boldsymbol{\theta}_i^2||), \quad l = x, y, z, \tag{4.5}$$

where $\boldsymbol{M}_i = [\sum_{j=1}^{N} m_{x,i,j}, \sum_{j=1}^{N} m_{y,i,j}, \sum_{j=1}^{N} m_{z,i,j}]$ is the measurement during each rotation and $f_{\boldsymbol{\alpha}}(\boldsymbol{M}_i)$ is a scalar function of $\boldsymbol{M}_i$. The minimization problem Eq. (4.5) can be solved using various nonlinear methods, such as Levenberg,ÄìMarquardt algorithm[92] and Nelder,ÄìMead method[91]. However, these nonlinear methods are relatively computationally expensive. In practice, a gyroscope is usually integrated into an embedded system with limited computational power and battery life. Therefore, a method with low computational complexity is given below to estimate the scale factors and biases.

4.2.2 Model Linearization and Experimental Design

When the gyroscope is in a static state, the left side of the Eq. (4.4) becomes to $\boldsymbol{\theta}_{total,i} = 0$. As every term on the right side of Eq. (4.4) is greater or equal to zero. Thus, every term should be equal to zeros. Consequently, biases can be obtained as follows:

$$\begin{cases} b_x = -\frac{1}{N}\sum_{j=1}^{N} m_{x,j} \\ b_y = -\frac{1}{N}\sum_{j=1}^{N} m_{y,j} \\ b_z = -\frac{1}{N}\sum_{j=1}^{N} m_{z,j} \end{cases} \tag{4.6}$$

After the biases are given, the model Eq. (4.4) can be reformed into a first-order linear

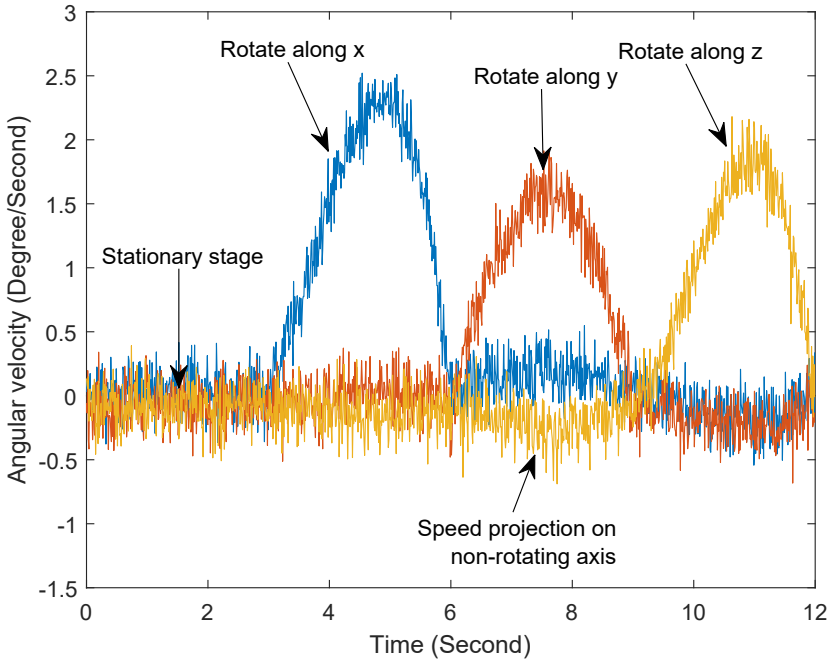

Figure 4.7: Typical simulated measurements of the proposed calibration method under $0.15°/sec$ noise level. Speed variation and speed projection on non-rotating axis were to simulate the manual rotation process.

form by redefining a new set of parameters and variables as follows:

$$\begin{cases} \beta_1 = k_x^2 \\ \beta_2 = k_y^2 \\ \beta_3 = k_z^2 \end{cases}, \begin{cases} x_{1,i} = \left(\sum_{j=1}^{N}(m_{x,i,j} + b_x)\right)^2 \\ x_{2,i} = \left(\sum_{j=1}^{N}(m_{y,i,j} + b_y)\right)^2 \\ x_{3,i} = \left(\sum_{j=1}^{N}(m_{z,i,j} + b_z)\right)^2 \end{cases}.$$

The polarity of the scale factor should be predefined. We consider the sign of the scale factors to be positive to ensure consistency with the datasheet. We also define the actual rotation angle as the response of the first-order model $y_i = \boldsymbol{\theta}_{total,i}^2$, then the model becomes:

$$y_i = \beta_1 x_{1,i} + \beta_2 x_{2,i} + \beta_3 x_{3,i} + \epsilon_i. \tag{4.7}$$

This can be easily calculated using least square method:

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{Y}, \tag{4.8}$$

where $\boldsymbol{\beta} = [\beta_1, \beta_2, \beta_3]^T$, $Y = [y_1, y_2, y_3]^T$, and $X = [x_{1,1}, x_{2,1}, x_{3,1}; x_{1,2}, x_{2,2}, x_{3,2}; x_{1,3}, x_{2,3}, x_{3,3}]$. After solving (4.8), the scale factors can be calculated by:

$$\begin{cases} k_x = \sqrt{\beta_1} \\ k_y = \sqrt{\beta_2} \\ k_z = \sqrt{\beta_3} \end{cases} . \tag{4.9}$$

In optimal experiments design, G-optimality aims to minimize the maximum variance of the predicted parameters. The scaled prediction variance (SPV) of the linearized model Eq. (4.7) can be expressed as:

$$d(\boldsymbol{x}, \boldsymbol{\xi_n}) = n \boldsymbol{f^T}(\boldsymbol{x})(\boldsymbol{X^T X})^{-1} \boldsymbol{f}(\boldsymbol{x}), \tag{4.10}$$

where $\boldsymbol{\xi_n}$ is the n observations experimental scheme and $\boldsymbol{f}(\boldsymbol{x}) = [x_1, x_2, x_3]^T$. We propose a three observations scheme and its design matrix is

$$D = \begin{array}{c} \\ (1) \\ (2) \\ (3) \end{array} \begin{array}{c} \begin{array}{ccc} x_1 & x_2 & x_3 \end{array} \\ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{array}. \tag{4.11}$$

Theorem 1 confirms that the DoE (4.11) is G-optimal.

Theorem 1. *The experimental scheme Eq.* (4.11) *is the G-optimal design of the model* (4.7) *in the spherical design region.*

Proof. To prove the G-Optimality of the experimental scheme Eq. (4.11) must make sure the maximum SPV is equivalent to the amount of unknown parameters[94]:

$$\begin{aligned} d(\boldsymbol{x}, \boldsymbol{\xi_n}) &= n \boldsymbol{f^T}(\boldsymbol{x})(\boldsymbol{X^T X})^{-1} \boldsymbol{f}(\boldsymbol{x}) \\ &= 3(x_1^2 + x_2^2 + x_3^2). \end{aligned}$$

In the spherical design region $(x_1^2 + x_2^2 + x_3^2) = 1$, $d(\boldsymbol{x}, \boldsymbol{\xi_n})$ has the maximum value of 3, which is identical to the amount of parameters. ■

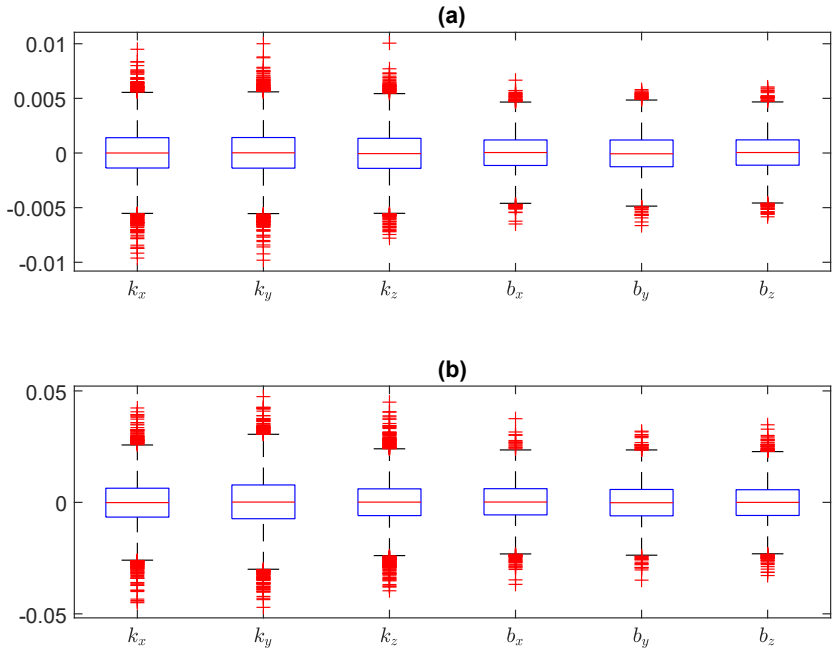

Figure 4.8: The error of estimated biases and scale factors compared with the true values at different noise levels. (a)0.03°/*sec* noise level. (b)0.15°/*sec* noise level.

4.2.3 Summary of the calibration process

The calibration process can be summarized into two stages, including a stationary and rotating stage, respectively. In the stationary stage, the gyroscope is placed on a stable surface and remains still for three seconds. In the rotating stage, two smooth planes perpendicular to each other are selected (see Fig. 4.10). The tabletop and the side of the cuboid box perpendicular to the horizontal plane are selected. The two planes are not required to be strictly vertical, but they are required to be stable. First, place the gyroscope on a horizontal table close to the side of the box, and mark its initial position on the table before gently rotating 360 degrees around the x-axis of the gyroscope. When the gyroscope coincides with the initial position of the mark, it is deemed as one rotation. This process is then repeated on the above operation in the y, z-axis to obtain a total of 3 observations. It is worth noting that high-accuracy equipment and customized case are not used in this process.

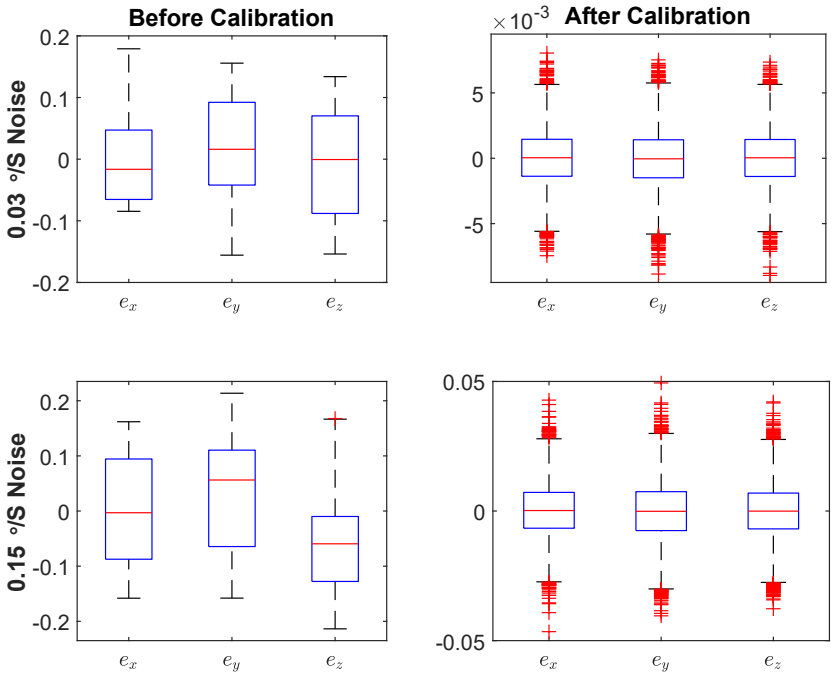

Figure 4.9: The mean error of the calibrated angular velocity compared with the actual value of each axis. Row: under different noise levels. Column: before and after calibration.

4.3 Simulation

To validate the proposed method under different conditions such as scale factors, biases, mounting misalignment, and measurement noise, we first inspect the proposed method using simulations. For the purpose of simulating under the similar hardware condition, the following assumptions on the parameter are given, and the results are explained thereafter:

1. The scale factors and biases follow uniform distributions $U(80\%, 120\%)$ and $U(-5°/s, 5°/s)$, respectively. The parameters of low-cost MEMS gyroscopes are usually in between these values.

2. The mounting misalignment on each axis follows a uniform distribution $U(-10\%, 10\%)$. In the absence of high-precision fixtures, it is almost impossible to mount the gyroscope accurately.

3. The measurement noise is assumed following a Gaussian distribution with zero

mean and two different noise $\mathcal{N}(0, 0.03^2)$ and $\mathcal{N}(0, 0.15^2)$. The typical rate noise density is between $0.003 - 0.015°/sec/\sqrt{Hz}$. At a $100Hz$ sampling frequency, we consider the standard deviation of root-mean-square noise as $0.03 - 0.15°/sec$.

Based on the above assumptions, we formulate the following simulation process. 30 sets of parameters are randomly generated based on the assumptions above. For each set of parameters, 500 simulations were repeated. During each simulation, we created a 4-observation measurement according to the experiment scheme described in Section 4.2.3. For the purpose of showing that the proposed method is not sensitive to the speed variation, the rotation speed in the rotating stage was represented by a randomly generated Bezier curve [103]. A typical run of one simulation is shown in Fig. 4.7. We also generated a testing set for each simulation. The gyroscope were assumed to rotating constantly and have equal speed projection on each axis during testing process.

After obtaining the 4-observation measurement, (4.6) and (4.8) are employed to calculate the biases and scale factors. We use the estimated biases and scale factors to compensate the test set and compare with the actual value. 15000 simulations were conducted for testing our efficient calibration method.

We first observe the error between the estimated parameters and the actual parameters. Box plot Fig. 4.8 was used to analysis the statistical characteristics of the error. The results show that the estimated scale factors and bias terms are unbiased with a zero median value of the estimation error. The majority of the estimation error are within $\pm 5.5 \times 10^{-3}$ for $0.03°/sec$ noise level and within $\pm 2.5 \times 10^{-2}$ for $0.15°/sec$ noise level. Besides, each subplot in Fig. 4.9 shows statistical characteristics of the mean error of the calibrated measurement value and the actual value in the testing set in 15,000 simulations. The error is defined as:

$$e_l = \frac{\sum_{i=1}^{N}(g_{l,i}^t - \hat{m}_{l,i}^t)}{N}, l = x, y, z, \tag{4.12}$$

where g^t and $\hat{m}^t$ stands for the true value and calibrated measurement value in the test set. Under both noise levels, the error after calibration is significantly reduced

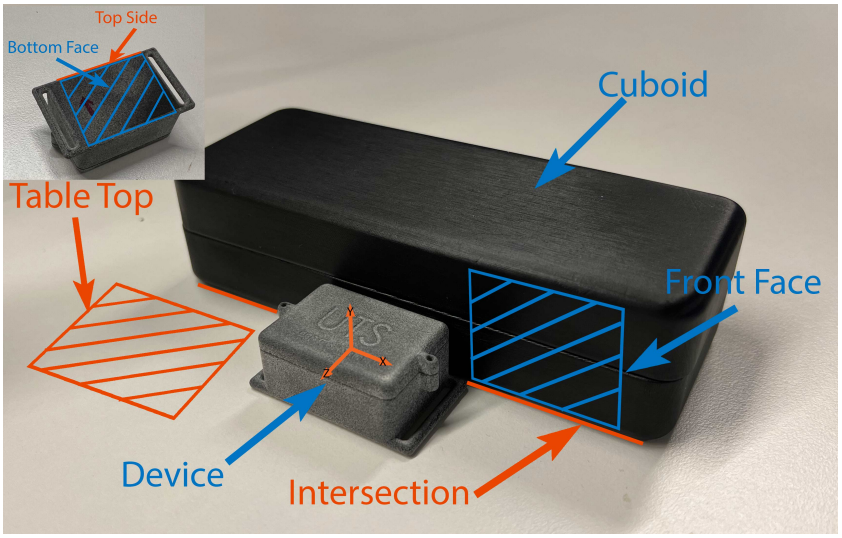

Figure 4.10: Experimental system for gyroscopes calibration. Main figure: the initial position of the device. Top left figure: bottom view of the device.

compared to the error before calibration. Among them, under the noise level of $0.03°/sec$, the measurement error is reduced to 3% of the previous.

4.4 Experiments

In this section, we empirically implemented and validated the proposed method. For the purpose of show the efficiency and accuracy of our method, we compared the proposed method with three existing method.

4.4.1 Experiments Device and Hardware Design

The developed wearable MEMS IMU device is shown in Fig. 4.1. This device is composed of a 3D printed case, a 9 degree of freedom inertial sensor (ICM20948), an RF transceiver (RTL8762AG) and a microcontroller (STM32F103T8U6). Owing to the low energy consumption features of the chosen components and adjustable inertial sensor sampling frequency ($4.4Hz - 562.5Hz$), the device can continuously work for 14 hours with a 400 mAh lithium polymer battery. Besides, to further reduce the power consumption, we use Bluetooth Low Energy technology to transmit the collected data to the host computer. In

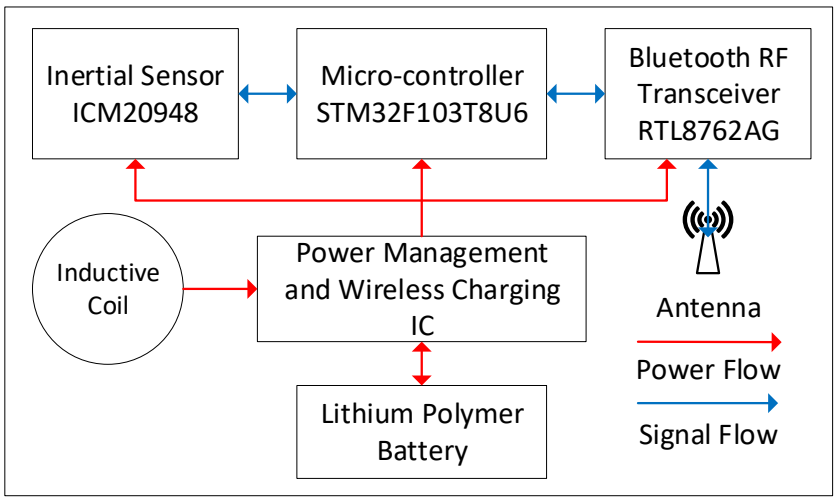

Figure 4.11: The designed wearable motion tracking device functional block diagram.

addition, the 3D printed case has good airtightness, which makes the device meet the IP67 waterproof standard. For ease of use, the device is also equipped with the wireless charging function. The system functional block diagram is shown in Fig. 4.11.

4.4.2 Experimental Setting

Two commercial-grade low-cost gyroscopes are tested in this study. One is LSM9DS1 from ST-Microcontroller, and the another one is ICM20948 from TDK Invensense. The LSM9DS1 was set to 104Hz sampling rate with a full-scale range of 245°/sec. On ICM20948, the sampling frequency was set to 104Hz, and the full-scale range was set to 250°/sec. The parameters are summarized in Table. 4.1. The room temperature was set to 22°C. All devices were preheated for 3 minutes before experiments.

Table 4.1: Related parameters of test gyroscope

Parameter	LSM9DS1	ICM20948
Sensitivity Scale Factor Tolerance	Not Provide	±4.5%
Zero-Rate Output Tolerance (dps)	±30	±6.25
Scale range (dps)	245	250
Sampling Frequency (Hz)	104	104

Table 4.2: Calibration results comparison of LSM9DS1.

Parameter	Results of proposed calibration method	Results of conventional turntable method [46]	Results of accelerometer -aided method [47]	Results of servomotor -aided method [104]
k_x	**1.2374 (0.0245)**	1.2619	1.2464 (0.0154)	1.2590 (0.0029)
k_y	**1.1764 (0.0202)**	1.1966	1.1806 (0.0160)	1.2030 (0.0064)
k_z	**1.1504 (0.0032)**	1.1536	1.1603 (0.0067)	1.1441 (0.0095)
$b_x(°/s)$	**-3.1065 (0.0215)**	-3.1275	-3.1156 (0.0124)	-3.1436 (0.0157)
$b_y(°/s)$	**1.6171 (0.0141)**	1.6312	1.6137 (0.0175)	1.6585 (0.0274)
$b_z(°/s)$	**-1.4467 (0.0188)**	-1.4655	-1.4794 (0.0139)	-1.4493 (0.0162)
Time (Second)	**27**	1533	570	60
Equipment	**N/A**	High-precision turntable	Accelerometer	Servomotor

Notes. The value in parentheses is the absolute error compared with the turntable method.

Table 4.3: Calibration results comparison of ICM20948

Parameter	Results of proposed calibration method	Results of conventional turntable method [46]	Results of accelerometer -aided method [47]	Results of servomotor -aided method [104]
k_x	**0.9859 (0.0037)**	0.9822	0.9745 (0.0077)	0.9780 (0.0042)
k_y	**1.0096 (0.0088)**	1.0183	1.0189 (0.0006)	1.0168 (0.0015)
k_z	**0.9710 (0.0078)**	0.9788	0.9879 (0.0091)	0.9765 (0.0023)
$b_x(°/s)$	**-0.5236 (0.0067)**	-0.5169	-0.5159 (0.0011)	-0.5250 (0.0081)
$b_y(°/s)$	**-1.4347 (0.0055)**	-1.4402	-1.4488 (0.0086)	-1.4473 (0.0071)
$b_z(°/s)$	**1.0525 (0.0023)**	1.0502	1.0537 (0.0035)	1.0566 (0.0064)
Time (Second)	**29**	1495	565	60
Equipment	**N/A**	High-precision turntable	Accelerometer	Servomotor

Notes. The value in parentheses is the absolute error compared with the turntable method.

4.4.3 Comparing with existing calibration methods

To show the efficiency and accuracy of our method, we compared our method with two state-of-the-art autocalibration method [47, 104] and the gold standard turntable method [46].

Conventional turntable method. This method uses a high-precision turntable to provide a calibration reference. The key parameters of the turntable are 0.0001°position accuracy and 0.0001°/s angular velocity accuracy. The fixture manufacture precision is $0.02mm$. We use six angular rate method in this study.

The accelerometer-aided method. This method uses the feature that most IMUs include accelerometers, and uses accelerometers to provide a calibration reference for the calibration of the gyroscope. Following the experiment process described in [47], we reproduced the experiment using our gyroscopes. In our experiment, the inclination angle between the device and the desktop is set to 0°, 30°, and 45 °. Different from the original experiment, we did not repeat the experiment multiple times and take average

to improve the calibration accuracy, because the experiment time is already as long as ten minutes.

The servomotor-aided method. This method uses a servomotor to provide a reference for calibration and provides a fast and low-cost calibration method. Due to the use of external calibration equipment, this method is not a pure autocalibration method. We have calibrated our gyroscope with reference to the calibration method provided in [104].

Our proposed method is summarized in Section 4.2.3.

To compare the efficiency of these method, we also compared the calibration time. We use a stopwatch to calculate the experiment time. Before the start of each experiment, we prepare all the test equipment. After pressing the timing button, we start the experiment, and we stop timing until all the data is collected. We did not include the data processing time, because it usually only takes a few seconds with a well written program.

4.4.4 Results and discussion

The calibration results of four different calibration methods are summarized in Table. 4.2 and 4.3. As we do not know the true scale factors and biases of the testing device, we consider the result of gold standard turntable method as ground truth in the following discussion.

The absolute error of the proposed method is less than 2.5×10^{-2} for LSM9DS1 and less than 1×10^{-2} for ICM20948. Considering the low repeatability and large measurement noise of the low-cost gyroscopes, the calibration result is considerably accurate. The estimation error of ICM20948 is significantly lower than that of LSM9DS1. We infer that this is because the noise spectral density of ICM20948 is lower. This is also in line with the simulation result, that is, the calibration result of the gyroscope with lower measurement noise has a smaller estimation error. Besides, the repeatability of ICM20948 is higher, which means that its parameters change less during the experiments.

Compared with the accelerometer-aided method, our method achieves a similar calibration accuracy within one twentieth of the time. Besides, because the accelerometer is

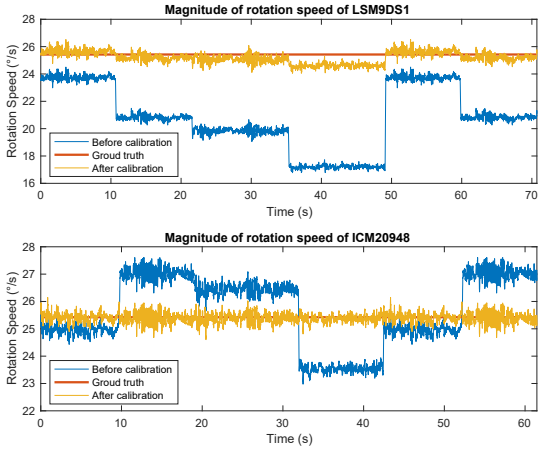

Figure 4.12: Magnitude of rotation speed before and after calibration. The acceleration and deceleration phases were omitted.

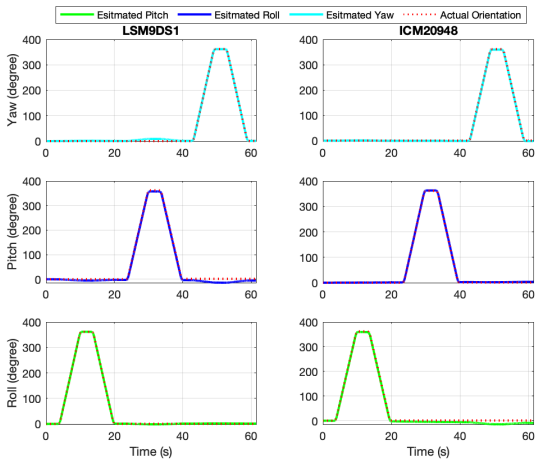

Figure 4.13: Estimated orientation using calibrated gyroscope readings after calibration from two models of gyroscope. Left: LSM9DS1. Right: ICM20948.

used to provide the gyroscope calibration reference, the calibration error of the accelerometer is also added to the gyroscope calibration process. The servomotor-aided method has a fast calibration speed and high calibration accuracy because it has a principle similar to that of a turntable. However, due to the use of external auxiliary equipment, it is not suitable for use in an in-field calibration situation.

To further verify the calibration results, we rotate the above calibrated gyroscope 360 degrees clockwise and counterclockwise along the x, y, z axis at a constant speed of 25.42°/s using a servomotor, and record the raw readings along with the calibration

parameters. When plotting, we did not plot the acceleration and deceleration phases. We calculated the magnitude of rotation speed before and after calibration. The magnitude of the rotation speed was calculated by the square root of the sum of squares of their vector components. The results are shown in Figure. 4.12. The results show that the magnitude of rotation speed after calibration is obviously closer to the ground truth, where the results of ICM20948 almost coincides with the ground truth.

As we discussed in Section 4.1.1, in the transdisciplinary project for monitoring the movements of women with complex pregnancies during labour and childbirth, without calibration, the angle estimation error was significantly high as shown in Figure.4.3. After using the proposed in-field calibration method, the estimation accuracy has been significantly enhanced. In Figure.4.13, it is clearly showed the ideal trajectories, and the estimated trajectories almost coincide when using the calibrated gyroscope readings.

Overall, the proposed method achieves a relatively high calibration accuracy. It should be emphasized that the entire calibration process only took less than 30 seconds, and any external device is not used during the process.

4.5 Conclusion

This chapter proposed an efficient in-field gyroscope calibration method for the purpose of developing a wearable monitor to track the movement of a labouring and birthing woman in a hospital birth room. We first introduced a self-designed low-cost motion tracking device. To overcome the error caused by the gyroscope during motion tracking, we presented a linearized calibration model. Based on the proposed linearized calibration model, a G-optimal experimental scheme was proposed for the calibration process. For the purpose of proving the feasibility of the proposed method, both simulations and experiments have been conducted.

The simulations were performed using the parameters which are similar to commonly used gyroscopes. The results indicate that the proposed method can achieve a relatively high calibration accuracy, and the results are unbiased. The experiments were conducted

on two commercial-level low-cost MEMS gyroscopes. To show the efficiency and accuracy of our method, the results of our method and the results of the turntable method as well as two state-of-the-art autocalibration were compared in this study. The results show that the errors between our method and turntable method were comparable. It has been shown that no high-precision equipment is required and the calibration process only takes less than 30 seconds.

CHAPTER 5

NONPARAMETRIC MODELLING BASED HUMAN HEART RATE REGULATION

5.1 Introduction

Collecting and labelling the data are both time and money consuming. In the e-trainer design aspect, the model should be personalized build and trained for a specific user, but asking users to collect and label a substantial amount of the data is unachievable. To overcome the lack of dataset issue, in this chapter, we use kernel method to provide prior information and minimize the model building phase.

During physical exercise, as the intensity of exercise changes, the cardiovascular system adjusts to the amount of blood and oxygen delivered to the working muscles, resulting in heart rate (HR) changes and respiratory rate changes. Creating a mathematical model for the cardiovascular system might give us a better understanding of exercise physiology[105]. Comprehending the aetiology of HR behaviours throughout the course of an exercise may also help predict and reduce the mortality from cardiovascular disease [106]. This is also conducive to improving athletes' performance and designing more effective weight loss procedures for obese people. It also helps to assess individual

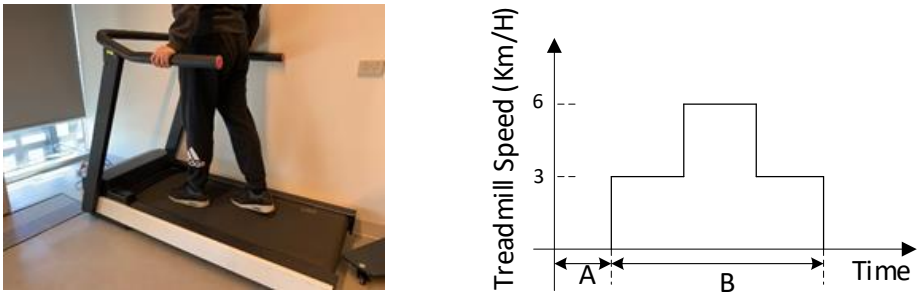

Figure 5.1: The proposed automatic treadmill system and speed profile during the identification period. (A) Resting. (B) Walking.[8]

physical health [107].

Modelling and controlling HR response during treadmill exercise has received considerable attention in the [108–110]. The variance of HR response measurement can be quite large because of the limitation of HR sensor accuracy and dislocation of the sensor [111]. Also, due to the complexity of the human cardiovascular system, it is hard to use a simple parametric model to describe the responses of the cardiovascular system to exercise. Accordingly, in this chapter we employ a nonparametric model, called a finite impulse response (FIR) model, to describe the HR response. However, due to the fact that the size of the FIR model is relatively large, the traditional system identification method usually requires a very complicated dynamic input to provide enough information for the model establishment, which will in turn lead to a long experiment time and sharp variations during model estimation. The inherent ill-posed problem caused by sensor noise and insufficient dynamic information can be solved by adding the regularisation term in the index function. This term reforms the problem into a regularised least square estimation (ReLS) problem [64]. However, ReLS only solves the ill-condition problem and is incapable of providing any prior information to the model estimation process. To this end, we reform the FIR model estimation problem as Gaussian Process modelling[112]. By adding a kernel term in reproducing kernel Hilbert space (RKHS), the prior information is embedded in the identification process by assigning a covariance which is also called a kernel in the machine learning field [63]. The participation of this prior information means fewer experiments can provide enough information for model identification.

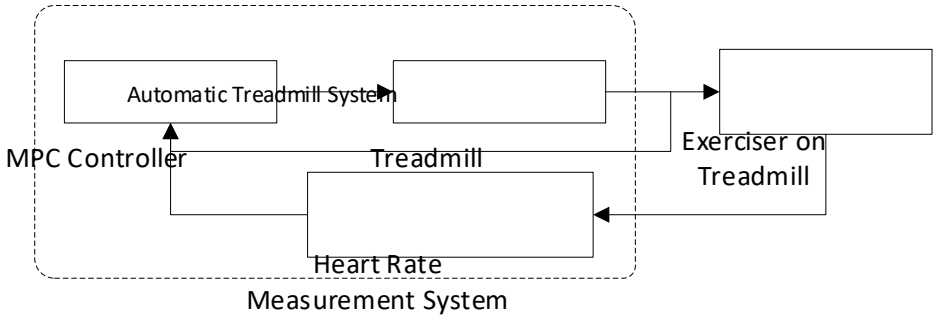

Figure 5.2: Schematic of the automatic treadmill system [8]

The contributions of this study are summarised as follows: (a) An effective kernel-based nonparametric modelling method is developed for identifying the HR response model. By applying this method, we can significantly reduce the number of experiments and the complexity of the experimental protocol to reach the desired modelling accuracy. (b) An effective HR tracking controller is developed by integrating the proposed nonparametric modelling method and the MPC. This new model predictive controller can also limit the speed and acceleration ranges to ensure the safety of the exercisers. To the best of the authors' knowledge, it is the first time that the kernel-based nonparametric modelling approach has been integrated with MPC control for HR regulation during treadmill exercises. (c) The proposed modelling and control algorithms have been experimentally validated on 12 participants and have achieved the desired HR tracking accuracy.

5.2 Kernel-based estimation method of heart rate response model

The heart rate (HR) response model can be dynamically described by its impulse response $g(k)$ as follows:

$$y(t) = \sum_{k=1}^{\infty} g(k)u(t-k) + e(t), t = 1,2,\ldots N, \tag{5.1}$$

where the $e(t)$ is the noise, and N is the total number of sampling. We can compute the system output $y(t)$, here it is the HR, by knowing the corresponding impulse response

$g(k)$ and input signal $u(t)$, here it is the treadmill speed. In general, $e(t)$ is supposed to be independent of $u(t)$.

For a stable system, its impulse response decays exponentially. Thus, the system can be approximately decrypted by its mth order finite impulse response (FIR):

$$y(t) = \sum_{k=1}^{m} g(k)u(t-k) + e(t), t = 1,2,\ldots N. \tag{5.2}$$

When we stack all the rows in $y(t)$ and $u(t)$ to the vectors form and define $[g(1), g(2), \ldots, g(m)] = \theta \in R^m$, then equation (5.2) becomes

$$Y = \phi\theta + E. \tag{5.3}$$

Apparently, the least-squares estimator of the model (5.2) is

$$\hat{\theta} = \arg\min_{\theta} ||Y - \phi\theta||^2. \tag{5.4}$$

In industrial applications, when the experiments are well designed (e.g., PRBS inputs) and comprehensively performed, the information matrix associated with equation Eq. (5.4) contains enough information to identify the parameter. As a result, even a classical least-square estimator can be applied to identify $\hat{\theta}$ which is the parameter of the nonparametric model. However, for the physiological model, in which human factors are involved, the experiments are often limited to input strength and duration. While the experimental protocol should not be too complicated, this is often the case for the modelling of the HR impulse. To ensure the safety of the treadmill exercisers, the input signal (i.e., the profile of the treadmill speed) is often confined to rectangular with moderate magnitude (treadmill speed). That is why most existing literature uses a simple parametric model, often a first-order model, to approximately descript the HR response to treadmill speed. To better accommodate the differences of various exercisers, a nonparametric model can be employed. This has the potential to develop personalised sports medicine based on accurate prediction of the cardiorespiratory response to exercise. However, using a high dimension impulse response model with limited model stimulation often leads to an ill-conditioned problem, i.e. a small error in the measurement can lead to a large estimation error. To address this issue, a commonly used technique is that of

adding a regularisation term to the estimator (5.4). In contrast to the regularised least square estimation (ReLS) method introduced in [64], we add a kernel regularisation term to the estimator [113]

$$\begin{aligned}\hat{\theta} &= \arg\min_{\theta} ||Y - \phi\theta||_2 + \gamma\theta^T\beta^{-1}\theta \\ &= \beta\phi^T\left(\phi\beta\phi^T + \gamma I_N\right)^{-1}Y,\end{aligned} \tag{5.5}$$

where the second item is a kernel regularisation term that denotes the squared Euclidean norm in reproducing kernel Hilbert space (RKHS). β is an N-by-N kernel matrix containing the prior information of FIR. Comparing to the ReLS method, the advantage of the kernel method is that it has a stronger capacity to minimise the mean square error of FIR. The ReLS method only considers solving the ill-condition problem. More importantly, the prior information brought by the kernel allows us to build an impulse response model.

5.3 MPC Controller Design

During treadmill exercise, the walking speed and acceleration must be limited to within a safe range to guarantee the safety of exercisers. Because model predictive control (MPC) has the inherent ability to deal with constraints, it is the most suitable choice. MPC depends on the dynamic model of the process to predict and optimise the future behaviour of the process. MPC uses current measurements, including the dynamic information of the current process, the model, process reference trajectory, and constraints, to calculate future changes in manipulated variables. MPC usually only implements the first optimal sequence to the plant and repeats the calculation when the next change is needed.

Let $y(t)$ and $\hat{y}(t+1|t)$ represent the current measurement and predicted measurement, respectively. The control output $u_t, u_{t+1}, \ldots u_{t+q-1}$ can be obtained by solving the following constrained optimisation problem [110]:

Table 5.1: Participant information.

Col 1	Mean	SD	Range
Age(year)	28.67	0.94	27-30
Body mass(kg)	66.58	13.07	50-86
Height(cm)	174	10.35	156-187
BMI(kg/m^2)	21.75	2.22	18.37-24.59

n=12, 5 female, 7 male.

$$\begin{aligned} \underset{\Delta u_t \ldots \Delta u_{t+q-1}}{\text{minimize}} \quad & \sum_{l=0}^{p-1} ||\hat{y}_{t+l+1} - r_{t+l+1}||^2_{Q_y} + \sum_{l=0}^{q-1} ||\Delta u_{t+l}||^2_S \\ \text{subject to} \quad & u_{min} \le u_t \le u_{max} \qquad t = 0, \ldots, N-1, \\ & \Delta u_{min} \le \Delta u_t \le \Delta u_{max} t = 0, \ldots, N-1. \end{aligned} \tag{5.6}$$

Use the kernel-based nonparametric model to predict the future output of a certain range p (called the prediction horizon) at each time t. These future outputs $\hat{y}$ are predicted based on the given information (past inputs and outputs) up to time t and the future control output u generated by the controller up to time $t+q$, where q is called the control horizon. Q_y and S are the penalty matrix for prediction errors and control moves. Here, u and Δu are constrained speed and acceleration, respectively.

Based on the model $\theta = [g(1), g(2), \ldots, g(m)]$ estimated by Eq.5.5, the predictor can be described as:

$$\begin{aligned} \hat{y}(t+l|t) = & \sum_{k=1}^{l} g(k)u(t+l-k) + \sum_{k=l+1}^{m} g(k)u(t+l-k) \\ & + y_m(t) - \sum_{k=1}^{m} g(k)u(t-k), \end{aligned} \tag{5.7}$$

where $y_m(t)$ is the measured value at time t.

5.4 Experiments and Discussion

In this section, we introduce the procedure of the experiment and discuss the result. The experiment is divided into two phases. The first one is the model estimation phase, which

builds the HR response model for each participant by using the kernel-based estimation method. The second phase is MPC control by using the proposed model. In this phase, we achieve HR tracking with constrained speed and acceleration.

The modelling and control data was obtained from 12 participants. The detail is shown in Table 5.1. The experiment was performed under the approval of the UTS Human Research Ethics Committee (ETH17-1758). The data collection was based on voluntary participation, and the informed consent from all participants was obtained before the data collection.

Before each phase, the participants were permitted to consume a light meal 2 h before the experiment. High-intensity exercise was not allowed 3 h before the experiment. Each participant wore a wireless HR sensor and stood on the treadmill for 2 minutes before the experiment started. The room temperature was set to 22 ℃.

5.4.1 Experimental Equipment

The proposed heart rate (HR) regulation treadmill system and its MPC control system are shown in Fig.5.1. A TRACKMASTER FVX 325 medical-grade treadmill, which is manufactured by Full Vision Inc, is used in the automated system. This system can send treadmill speed to a personal computer and manipulate the treadmill speed via the serial port. The HR is measured by a wireless wearable Zephyr HR sensor. The sensor collects the analogue electrocardiogram signal and calculates the HR by using the edge detection method. The sensor transmits HR data to the control system every second via Bluetooth. However, it is observed that the measured HR is often polluted by electromagnetic interference generated by other environmental equipment. To address this issue, we use the proposed kernel-based nonparametric MPC for HR tracking.

5.4.2 Model Estimation

In the model estimation phase, the participants are required to walk on the treadmill according to the desired speed protocol shown in Fig.5.1. They first stand on the treadmill for 80 seconds. Then, they are asked to walk at 3 km/h for 3 minutes, followed by a faster

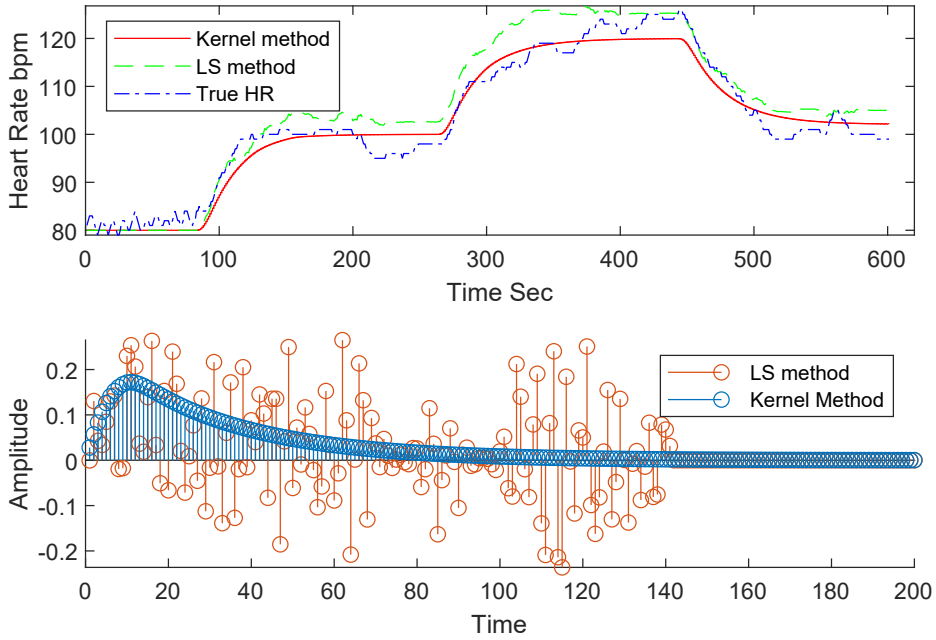

Figure 5.3: (Top) Typical estimated heart rate comparison between Kernel method and LS method. (Bottom) The estimated impulse response for one participant. [8]

walk at 6 km/h for 3 minutes. Then, another walk for 160 seconds at a speed of 3 km/h. The entire process takes 10 minutes, including 80 seconds resting period. The expertise protocol is relatively easy as it only contains two accelerations. The exerciser is informed 5 seconds before each acceleration.

For the finite impulse response model Eq.(5.2), the order m was selected as 250 and the sampling time was selected to 1 second. The proposed kernel-based estimator Eq. (5.5) was employed to identify the FIR model by using the Tuned/Correlated (TC) kernel, Diagonal kernel (DI) kernel, and Stable spline (SS) kernel. The fit error is defined as the normalised root mean square error between the estimated HR and true HR.

To select the best kernel, we estimated each participant model using three different kernels. The parameter selection method of each kernel was given in [113]. We calculated the fit error of each estimated model. As a comparison of the conventional method, we calculated the fit error of the model by using the latest square (LS) method Eq. (5.4). The fit error is recorded in Table 5.2. The kernel with the lowest fitness error was selected for the model estimation. It demonstrated that the proposed kernel method outperforms

Table 5.2: Fitness error.

Subject	TC(%)	SS(%)	DI(%)	LS(%)
Participant 1	33.86	31.68	30.79	41.91
Participant 2	21.61	30.61	21.23	25.63
Participant 3	19.05	17.59	18.81	21.85
Participant 4	26.83	27.50	29.24	30.26
Participant 5	25.56	25.53	25.46	34.61
Participant 6	50.36	66.37	37.69	53.52
Participant 7	28.75	29.07	18.41	44.62
Participant 8	48.05	48.15	47.43	82.40
Participant 9	25.85	26.60	25.84	26.83
Participant 10	54.18	56.83	53.70	84.65
Participant 11	55.61	39.07	38.00	54.51
Participant 12	22.53	23.18	22.50	30.41
Average	34.35	35.18	30.76	44.27
Standard deviation	13.11	14.07	10.85	20.24

the conventional LS method. Specifically, Fig.5.3(Top) shows a typical estimated HR comparison between the kernel-based method and LS method. The figure indicates that the proposed model response has a lower fitness error and is smoother than the LS model response in general. Fig.5.3(Bottom) is the estimated impulse response by using different methods. The LS method impulse response is extremely noisy. This indicates that the kernel-based method can solve the ill-posed problem.

5.4.3 MPC Heart Rate Regulation

In the heart rate (HR) regulation phase, using the kernel-based nonparametric model presented in Section 5.4.2, we designed a model predictive controller to track a predefined reference HR. The participants were required to stand on the treadmill for 60 seconds. The reference HR was then set to be 100 beats/minutes and to last for 6 minutes. The prediction horizon was $p = 10$ and control horizon was $q = 1$. The penalty matrix for prediction errors and control moves were set to be $Q_y = 1$ and $S = 5$, respectively. To maintain the exercise safety, the maximum speed was limited to 6 km/h and the acceleration was limited to ± 0.5 km/h/sec. Fig.5.4 demonstrates that the proposed kernel-based

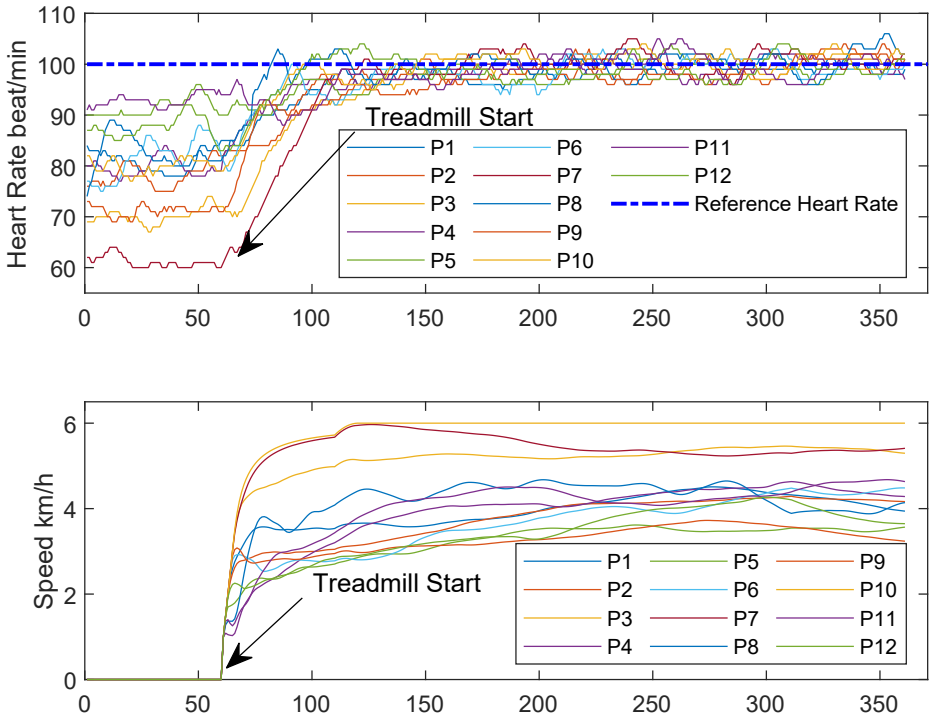

Figure 5.4: Heart rate tracking results for all 12 subjects. [8]

nonparametric model and MPC controller achieved the desired tracking performance. All 12 participants reached the target HR within 60 seconds after the treadmill started and without steady-state error. These performances are comparable with those discussed in previous literature [110]. However, it should be emphasised that the performance acquired by the proposed method requires easier experiments during the model estimation phase.

5.5 Conclusion

In this chapter, we proposed a nonparametric model and a kernel-based estimation approach describing the HR response during treadmill exercise. The proposed model and estimation method were applied to 12 participants. The experimental results indicated that the fit error of the proposed approach is lower than the least square method. In addition, the model estimation phase needs less time and does not contain com-

plex exercise protocols. By using this nonparametric model, an automatic treadmill system was built and employed for HR tracking during treadmill exercise. The MPC technique was implemented, which could achieve safe exercise by constraining both exercise speed and acceleration. Experimental results demonstrated that the proposed HR regulation system achieved low HR tracking error under the predefined acceleration and speed constraints. The proposed HR response model and MPC control approach were experimentally validated and might have important implications for cardiovascular rehabilitation, the creation of effective training plans for athletes and the development of efficient weight loss plans to combat obesity.

CHAPTER 6

MODEL COMPRESSION VIA PRUNING METHOD

6.1 Introduction

Artificial intelligence-based human motion state classification and exercise intensity control algorithms have received considerable attention in recent years. In some scenarios, exercisers wear more than one e-trainer. In this case, the state of the exerciser is a graph, and each sensor is a node in the graph. Consequently, Graph Neural Networks (GNNs) can be used to determine the sports state of the athlete. These algorithms often run on high-performance computers and cannot be applied to wearable devices. The main obstacle is that wearable devices do not have enough memory and computing power to run artificial intelligence models. Therefore, this chapter proposed a model compression method via neural network pruning, which greatly reduced the memory usage and calculation amount, so as to achieved the purpose of running the model on wearable devices.

Intricate relations and phenomena such as molecular structure, traffic networks and the Internet are naturally described as graphs. GNNs found their way on various graph-based learning tasks, such as node classification [114], link prediction [115], and graph classification [116], achieving excellent performance by leveraging graph information.

During the node feature learning, GNNs aggregate features from neighboring nodes, and then perform feedforward propagation on the aggregated embeddings hierarchically, thus cooperating structural information efficiently.

However, the inefficiency has always plagued the training and inference of GNNs, which hinders the application of GNN to deal with large-scale graphs in the real world. The recursive neighborhood aggregation scheme is the core of the GNN algorithm, in which each node aggregates the feature vectors of its multi-hop neighbors to calculate its new feature vector. In a large and dense graph, the aggregation phase consumes much computation. This makes the training and inference of GNNs easily surpass working memory. Therefore, it is critical to acquire smaller graphs and remove unnecessary weights in GNNs. To simplify the graph, one of the most popular paradigms is to leverage sampling-based methods with the local aggregations and local features updating [117–122]. In [123], the graph sparsification problem was further formulated as an optimization problem, where the alternating direction method of multipliers (ADMM) was employed as a solver. On the other hand, removing unnecessary weights, a.k.a. network pruning, is becoming a key instrument in deep neural networks (DNNs) [124–128]. However, network pruning received less attention from the GNN community since most existing GNNs only used very shallow neural networks compared with convolutional neural networks (CNNs) due to the feature over-smoothing problem. Network pruning aims to issue each weight with a saliency score and then removes the weights based on the score accordingly [124]. Frankle and Carbin [125] recently proposed a lottery ticket hypothesis (LTH) method, which empirically verified the existence of sub-networks (winning tickets). Such sub-networks were obtained by pruning the network based on the magnitude after training, which were able to reach comparable accuracy with the original network when resetting the remaining weights to their initial values. With more works focusing on alleviating the feature over-smoothing problem in GNNs [129–132], the GNNs can go as deep as CNNs in recent days [133–135], and the network pruning also becomes nontrivial in the GNN community.

Technically, GNN pruning contains the graphs sparsification and the network pruning.

Unlike heavily overparameterized DNNs, the shallow nature of GNNs may indicate that pruning the shallow GNNs will not have much benefit. However, the latest work proposed a graph lottery ticket (GLT) method that [48] simultaneously prunes the graph adjacency matrix and the model weights, and proved the existence of the lottery ticket hypothesis in GNNs. The authors directly introduced LTH, which was designed for network pruning, to GNNs, aiming to find a sub-network and a sub-graph simultaneously with comparable performance as the original GNN. It added a trainable mask to the graph and then performed training. After training, it considered the magnitude of the graph mask as the importance of graph link.

Rather than using magnitude after training for pruning as LTH, in this chapter, we propose to prune network weights without any training. Specifically, motivated by pruning-at-initialization techniques for general DNNs [126–128], wherein a per-parameter saliency metric is computed before training to inform pruning, we intuitively generalize train-free saliency metrics for general DNN pruning to the GNN pruning. Our main observations are:

First, the train-free saliency metrics can not only extremely improve the computational efficiency, but also achieve comparable performance as GLT for GNN pruning. It can still remove 98% of network weights without causing a significant drop in performance.

Second, although such pruning methods (including GLT, SNIP, Synflow, GraSP) achieve competitive performance when jointly pruning network weights and graph adjacency matrix or solely pruning network weights, they could hardly outperform random pruning when only pruning the graph.

Based on the above observations, we found that the existing pruning methods can effectively prune the network weights in GNNs, while they could hardly evaluate the importance of graph edges. Consequently, we infer that the benefits of GLT when simultaneously pruning graph adjacency matrix and network weight was brought by weight pruning, with arguing that graph links and network weight should use different scoring standards during GNNs pruning. To more precisely measure the importance of edges,

we mathematically formulate the performance of graph neural network with respect to the properties of edges, which elucidates how the loss of performance can be avoided by pruning negative edges and non-bridges, motivating a simple but effective method for graph edges pruning. In addition, different from GLT using the trained model to prune network weight, we use the training-free methods to prune network weight to further improve the efficiency of the algorithm, leading to a general two-step pruning method for GNNs that simultaneously prune network weights and graph edges. Our main contribution can be summarized as follows.

- We, for the first time, extend pruning-at-initialization methods to graph neural networks. We achieve comparable accuracy as the state-of-the-art GNN pruning method GLT [48], while extremely enhancing the computational efficiency, where we can get the sub-GNN within seconds compared with hours by GLT.

- To overcome the drawback of pruning based methods on edge pruning, we mathematically formulate the performance of GNN with respect to the properties of edges, elucidating how the loss of performance can be avoided by pruning negative edges and non-bridges, and accordingly provide an effective graph edges pruning strategy.

- We propose a General Two-Step Pruning (GTSP) method for effective graph neural network pruning, which leverages the training-free saliency metric for network pruning and sparsifies the graph by preserving performance. Experimental results verify the effectiveness of our proposed method, in which our approach outperforms GLT by large margins during the pruning process.

6.2 Preliminaries

Let $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ be an undirected graph, where $\mathcal{V}$ denotes the set of $|\mathcal{V}|$ nodes and $\mathcal{E}$ is the set of edges. An adjacency matrix $\boldsymbol{A} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$ is employed to describe the relations between nodes, where $\boldsymbol{A}[i,j] = 1$ if $(v_i, v_j) \in \mathcal{E}$ else $\boldsymbol{A}[i,j] = 0$. Moreover, let

$\boldsymbol{X} = [x_1, x_2, ..., x_{|\mathcal{V}|}]^T \in \mathbb{R}^{|\mathcal{V}| \times D}$ represents feature matrix where $x_i \in \mathbb{R}^D$ is the feature vector of node v_i, with D being the feature dimension. Thus, an alternate notation of a graph is $\mathcal{G} = \{\boldsymbol{A}, \boldsymbol{X}\}$. Consider a semi-supervised node classification setting, only a part of labels $\mathcal{Y}_t = \{y_1, y_2, \ldots, y_t\}$ are given, where $t \ll |\mathcal{V}|$. The goal of a GNN node classification task is to learn a function $f_{\boldsymbol{\Theta}}$ that maps the nodes features to the labels, where $\boldsymbol{\Theta}$ is the collection of weights in the neural network.

6.2.1 Graph Neural Networks

There are two main families of GNNs, spectral GNNs and spatial GNNs. The spectral GNN was directly derived from the spectral graph theory [136, 137]. The basis of spectral GNN is graph Fourier transform, defined by the eigendecomposition of the graph Laplacian matrix. The most popular graph network belonging to spectral class are Chebyshev polynomials [138], and Cayleynets [139]. Even though spectral-based GNNs can theoretically extract information on any frequency, the piratical power is constrained by a fixed pattern. The spatial-based GNNs that aggregate nodes neighborhood information are more flexible and have less computational complexity. It is known that GCN [114], GAT [140], and GIN [141] belong to spatial GNN. Existing researches showed that most of the existing spatial-designed convolutions were essentially low-pass filters [142].

With the growth of graph data, the computation cost and memory usage of GNNs rapidly increased. Graph sampling or sparsification aims to find a subgraph from the large original graph that best preserve desired information. A considerable literature sparsifies the graph using different metrics such as graph spectrum [143, 144], global importance [145–147] or clustering coefficient [148]. ReuralSparse [149] utilizes deep neural networks to parameterize sparsification processes. Li et al. [123] consider graph sparsification as an optimization problem and solve it by using the alternating direction method of multipliers (ADMM).

6.2.2 Network Pruning

Pruning focuses on removing superfluous parameters in a neural network. Most pruning methods operate on a pre-trained network. The main idea is to identify a saliency metric for each parameter and remove the less-important parameters. Among these methods, magnitude-based pruning algorithms [124, 125] use the weight as the saliency metric, while Hessian-based pruning algorithms [150–152] identify the importance of each weight by measuring how its removal will affect the loss. However, these methods need training before determining the saliency metric. To further reduce the computation complexity during the pruning process, the training-free pruning-at-initialization algorithms are introduced by [126] and extended by [127, 128, 153–155]. The saliency criteria are computed in a single backward propagation pass, and these methods are successfully used to heavily prune neural networks before training. SNIP [126] provides us with a pruning-at-initialization method. This method samples training data and issues weights with scores $z = |\frac{\partial \mathscr{L}}{\partial \Theta} \odot \Theta|$, where $\mathscr{L}$ is the loss function of a neural network with parameters Θ. This saliency criteria describe the loss change w.r.t. each parameter removal. In this study, we consider a recently-proposed iterative variant of SNIP [156]. Wang et al. [127] extended snip metric to *grasp* by measuring the change in gradient norm when a parameter is removed. It computes the Hessian-gradient product $\mathscr{H}$ and issues scores $z = -(\mathscr{H}\frac{\partial \mathscr{L}}{\partial \Theta}) \odot \Theta$. Besides, *SynFlow* [128] treats the input of the network as one and computes the sum R of the logits. They claim that this method can avoid layer collapse when pruning. It issues scores $z_\theta = \frac{\partial R}{\partial \theta} \odot \Theta$ and removes weights with the lowest scores.

6.3 Theoretical Framework

In this section, we study the performance of graph neural networks with respect to the properties of the input graph. In particular, we leverage the equivalence between spatial-based GNNs and spectral-based GNNs, based on which, we apply graph spectral theory. As a result, we characterize the error bound for node classification with respect to the

graph properties. Furthermore, we propose the related edge properties that are correlated with the derived error bound and investigate the influence by pruning corresponding edges.

6.3.1 Formulations

A central object in spectral graph theory is the so-called graph Laplacian [157] which is defined by $\boldsymbol{L} = \boldsymbol{I} - \boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2}$, where $\boldsymbol{A}$ is the adjacency matrix, $\boldsymbol{D} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$ is the diagonal degree matrix with $\boldsymbol{D}_{i,i} = \sum_j \boldsymbol{A}_{j,i}$. This work targets pruning the edge of the graph whose performance depends highly on the properties of the graph as shown in our theorem later. We introduce two critical properties as a preliminary,

Definition 6.1 (Positiveness and negativeness of edge)**.** For an edge $e(v, v')$, if the labels of vertexes v, v' are the same, i.e. $y(v) = y(v')$, we define it the positive edge denoted as $\mathscr{E}_{pos}$, otherwise the negative edge denoted as $\mathscr{E}_{neg}$.

Definition 6.2 (Bridge and non-birdge)**.** For an edge $e(v, v')$, a bridge is an edge of a graph whose deletion increases the graph's number of connected components, denoted as $\mathscr{E}_{bri}$. On the contrary, the edge is called non-bridge denoted as $\mathscr{E}_{non}$.

The spectral analysis for graph Laplacian resort to the eigendecompostion, namely, $\boldsymbol{L} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{U}^T$, where $\boldsymbol{U} = [\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_{|\mathcal{V}|}]$ is the eigenvector and $\boldsymbol{\Sigma} = \text{diag}[\lambda_1, \lambda_2, \ldots, \lambda_{|\mathcal{V}|}]$. The spectral network can be written by a sum of filtered signals [158],

$$\boldsymbol{H}_j^{(l+1)} = \sigma\Big(\sum_{i=1}^{f_l} \boldsymbol{U}\mathrm{d}iag(\boldsymbol{K}_{i,j,l})\boldsymbol{U}^T\boldsymbol{H}_i^{(l)}\Big)$$

where $j \in \{1, \ldots, f_{l+1}\}$, σ is the activation function, $\boldsymbol{H}_i^{(l)}$ is the i-th feature vector in l-th layer, $\boldsymbol{K}_{i,j,l} \in \mathbb{R}^{|\mathcal{V}|}$ is the corresponding trainable weight vector. On the other hand, the spatial GNN can be generalized as propagation of node features to the neighborhood node followed by activation function, of the form,

$$\boldsymbol{H}^{(l+1)} = \sigma\big(\boldsymbol{C}\boldsymbol{H}^{(l)}\boldsymbol{\Theta}^{(l)}\big)$$

where $\boldsymbol{\Theta}^{(l)}$ is the trainable weight, $\boldsymbol{H}^{(l)}$ is the feature matrix in l-th layer, and $\boldsymbol{C}$ is the convolution kernel. GCN [114] adopts a single convolution kernel $\boldsymbol{C} = \tilde{\boldsymbol{D}}^{-1/2}\tilde{\boldsymbol{A}}\tilde{\boldsymbol{D}}^{-1/2}$, where $\tilde{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{I}$ is the adjacency matrix of the undirected graph with added self-connections, and $\tilde{\boldsymbol{D}}$ is the corresponding degree matrix. On the other hand, GIN [141] is equipped with $\boldsymbol{C} = \boldsymbol{A} + \boldsymbol{I}$. While GCN and GIN use fixed-design matrices $\boldsymbol{C}$, GAT [140] uses trainable convolution kernels.

6.3.2 An Error Bound for GNNs through Graph Spectral theory

The equivalence between spatial and spectral GNNs has been revealed by [142, 159] recently and stated as follows,

Lemma 6.1 (Equivalence between spatial and spectral GNNs)**.** *With the convolution kernel of spatial GNNs set to $\boldsymbol{C} = \boldsymbol{U}diag(\boldsymbol{K}(\lambda))\boldsymbol{U}^T$, where $\boldsymbol{U}$ are the eigenvectors of the Laplacian for studied graph, $\boldsymbol{K}(\lambda)$ is the corresponding trainable weight depending on eigenvalue λ, the spatial GNN is equivalent to spectral GNN.*

The detailed proof can be found in [142]. With this results, we can further bound the prediction error of GNNs.

Theorem 6.1 (Spectral learning of GNNs)**.** *Consider a L-layer spatial GNN with $\boldsymbol{C} = \boldsymbol{U}diag(\boldsymbol{K}(\lambda))\boldsymbol{U}^T$ and last layer being linear, i.e., $\boldsymbol{H}^{(L)} = \boldsymbol{U}diag(\boldsymbol{K}(\lambda))\boldsymbol{U}^T\boldsymbol{H}^{(L-1)}\boldsymbol{\Theta}^{(L-1)}$. Then there exists a linear weight $\boldsymbol{\Theta}^{(L)}$, such that the linear probe error of embedding $\boldsymbol{H}^{(L)}$ is,*

$$E(\boldsymbol{H}^{(L)}) = O\left(\frac{\alpha}{\kappa}\right)$$

where $E(\boldsymbol{H}^{(L)})$ is the mean squared error as the best possible linear classifier on the embedding, defined as $E(\boldsymbol{H}^{(L)}) = \min_{\boldsymbol{\Theta}^{(L)}} \frac{1}{|\mathcal{V}|}\|[\boldsymbol{H}^{(L)}\boldsymbol{\Theta}^{(L)} - \mathcal{Y}\|_2^2$, α is the density of edges that connects different labels, i.e., $\alpha = \frac{|\mathcal{E}_{neg}|}{|\mathcal{V}|}$, and κ is the $(d+1)$-th smallest eigenvalue of graph Laplacian, with d being the dimension of embedding.

Proof. Let $\vec{g} = \{0,1\}^{|\mathcal{V}|} \in \mathbb{R}^{|\mathcal{V}|\times 1}$ be the vector the labels of all vertexes (nodes) under the an optimal classifier g. Then we show there is a constraint on the quadratic form with

respect to the optimal classifier g. According to graph spectral theory [136], the quadratic form $\vec{g}^T \boldsymbol{L} \vec{g} = \frac{1}{2} \sum_{v,v'} \frac{\boldsymbol{A}(v,v')}{\sqrt{\boldsymbol{D}(v)\boldsymbol{D}(v')}} (\vec{g}_v - \vec{g}_{v'})^2$ captures the amount of edges connecting different labels. Since we have defined that α as the density of edges that connects different labels, we can directly show that $\vec{g}^T \boldsymbol{L} \vec{g} \leq \alpha |\mathcal{V}|$. This implies that the quality of classifier depends on the density of negative edges in the graph.

Next we show that there exist a linear weight $\boldsymbol{\Theta}^{(L)}$ such that the linear probe of embedding can approximate the optimal classifier, namely, $\boldsymbol{H}^{(L)} \boldsymbol{\Theta}^{(L)} \approx \vec{g}$. Denote the eigenvalue of Laplacian in an ascending manner, i.e., $0 \leq \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_d \leq \cdots \leq \lambda_{|\mathcal{V}|} \leq 2$, and the corresponding eigenvectors are denoted as $\vec{u}_1, \cdots, \vec{u}_{|\mathcal{V}|}$. Let $\boldsymbol{\Pi} = \sum_{i=1}^{d} \vec{u}_i \vec{u}_i^T \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$, and $\boldsymbol{\Pi}^{\perp} = \sum_{i=d+1}^{|\mathcal{V}|} \vec{u}_i \vec{u}_i^T \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$. Because $\boldsymbol{H}^{(L)} = \boldsymbol{U} \mathrm{d}iag(\boldsymbol{K}(\lambda)) \boldsymbol{R}$, where $\boldsymbol{R} = \boldsymbol{U}^T \boldsymbol{H}^{(L-1)} \boldsymbol{\Theta}^{(L-1)}$ contains a space spanned by eigenvectors of $\vec{u}_1, \ldots, \vec{u}_d$, then there exists a linear matrix $\boldsymbol{\Theta}^{(L)}$ such that $\boldsymbol{H}^{(L)} \boldsymbol{\Theta}^{(L)} = \boldsymbol{\Pi} \vec{g}$. As a result, the difference between optimal classifier and our linear predictor is $\|\vec{g} - \boldsymbol{H}^{(L)} \boldsymbol{\Theta}^{(L)}\|_2^2 = \|\vec{g} - \boldsymbol{\Pi} \vec{g}\|_2^2 = \|\boldsymbol{\Pi}^{\perp} \vec{g}\|_2^2$. On the other hand, we can expand the quadratic form by $\vec{g}^T \boldsymbol{L} \vec{g} = (\boldsymbol{\Pi} \vec{g} + \boldsymbol{\Pi}^{\perp} \vec{g})^T \boldsymbol{L} (\boldsymbol{\Pi} \vec{g} + \boldsymbol{\Pi}^{\perp} \vec{g}) \geq (\boldsymbol{\Pi}^{\perp} \vec{g})^T \boldsymbol{L} (\boldsymbol{\Pi}^{\perp} \vec{g}) \geq \lambda_{d+1} \|\boldsymbol{\Pi}^{\perp} \vec{g}\|_2^2$. Combine the two equations, we have, $\frac{1}{|\mathcal{V}|} \|\mathcal{Y} - \boldsymbol{H}^{(L)} \boldsymbol{\Theta}^{(L)}\|_2^2 \leq \frac{\alpha}{\kappa}$, where $\kappa = \lambda_{d+1}$.

■

6.3.3 Edge Property Matters

Theorem 6.1 implies that the performance of GNNs (including GCN, GIN and GCN) depends on two quantities of the studied graph, namely the number of edges that connects different labels and the largest eigenvalue of graph Laplacian λ. In this subsection, we show that how this result can further be utilized to guide graph adjacency pruning, through pruning negative edges and non-bridges respectively.

Definition 6.1 divides the edges of a graph into two classes. Based on Theorem 6.1, it is easy to see that pruning negative edge will decrease the error bound when κ is fixed, thus leading to the following corollary,

Corollary 6.1. *Denote $\mathscr{B}(\mathcal{G}) = \frac{\alpha(\mathcal{G})}{\kappa(\mathcal{G})}$ as the linear probe error bound of a GCN satisfying conditions stated in Theorem 6.1 trained on a Graph $\mathcal{G}$. If we only prune negative edges*

in $\mathcal{G}$ and resulting graph is denoted as $\mathcal{G}'$. Suppose that κ changes rarely during negative edge pruning, then $\mathcal{B}(\mathcal{G}') < \mathcal{B}(\mathcal{G})$.

Proof. Because the number of negative edges in $\mathcal{G}'$ is less than negative edges in $\mathcal{G}$ after pruning, we have $|\mathcal{E}_{neg}(\mathcal{G}')| < |\mathcal{E}_{neg}(\mathcal{G})|$. As a result, we have $\alpha(\mathcal{G}') < \alpha(\mathcal{G})$. In addition, since κ changes rarely during negative edge pruning, which implies $\kappa(\mathcal{G}') = \kappa(\mathcal{G})$. Finally, we can arrive at $\mathcal{B}(\mathcal{G}') < \mathcal{B}(\mathcal{G})$. ■

The next corollary accounts for eigenvalue of graph Laplacian, and show that pruning bridges can lead to error bound increase.

Corollary 6.2. *Denote the $\mathcal{G}'$ as the graph with bridges pruned from its origin, then $\mathcal{B}(\mathcal{G}') > \mathcal{B}(\mathcal{G})$.*

Proof. According to the definition 6.2 for bridge, pruning bridge will lead to the increase of independent component. Suppose the resulting graph $\mathcal{G}'$ has M independent components, then eigenvalue of the corresponding Laplacian is $\lambda_1 = \lambda_2 = \cdots = \lambda_M = 0$. As we keep pruning bridges until we have $M \geq d + 1$, which means $\kappa(\mathcal{G}') = 0$. Thus we have $\mathcal{B}(\mathcal{G}') = \frac{\alpha(\mathcal{G}')}{\kappa(\mathcal{G}')} > \mathcal{B}(\mathcal{G})$. ■

6.4 Build Effective Pruning Method for GNNs

In this section, we first revisit the graph lottery ticket [48] and leverage the pruning-at-initialization saliency metrics to GNNs. Then, we propose our novel pruning method, which removes the negative edges and non-bridges to preserve the performance. After that, we propose a general two-step pruning method for GNNs which simultaneously prunes the graph links and network weights.

6.4.1 Revisit Graph Lottery Ticket

Graph Lottery Ticket. Similarly to LTH, Graph Lottery Ticket (GLT) [48] hypothesizes that given a GNN $f(\cdot, \Theta)$ and a graph $\mathcal{G} = \{\boldsymbol{A}, \boldsymbol{X}\}$, where $\boldsymbol{\Theta}$ is the collection of all weights,

there exists sub-networks $f(\cdot, \boldsymbol{m}_{\boldsymbol{\Theta}} \odot \boldsymbol{\Theta})$ trained on sparse graphs $\mathcal{G}_s = \{\boldsymbol{m}_g \odot \boldsymbol{A}, \boldsymbol{X}\}$ reach test accuracy comparable to the original GNN trained on the full graph $\mathcal{G}$. We define $f(\{\boldsymbol{m}_g \odot \boldsymbol{A}, \boldsymbol{X}\}, \boldsymbol{m}_{\boldsymbol{\Theta}} \odot \boldsymbol{\Theta})$ as a unified graph lottery tickets (GLTs), where $\boldsymbol{m}_g$ and $\boldsymbol{m}_{\boldsymbol{\Theta}}$ are two trainable masks for indicating the importance of graph links and network weights in the graph and GNNs, respectively.

Picking winning tickets. Classical LTH leverages iterative magnitude-based pruning (IMP) to identify lottery tickets. In a similar way, GLT identify the winning tickets after the network is trained, as described as follows:

1. Randomly initialize a neural network $f(\cdot, \boldsymbol{\Theta}_0)$, initial mask $\boldsymbol{m}_g^0 = \boldsymbol{A}$, $\boldsymbol{m}_{\boldsymbol{\Theta}}^0 = \mathbf{1} \in \mathbb{R}^{\|\boldsymbol{\Theta}_0\|_0}$, and $j = 0$.
2. Train the network for k iterations, arriving at convergence with parameters $\boldsymbol{\Theta}_j$ and $\boldsymbol{m}_g^j$.
3. Set $s_g\%$ of the lowest magnitude values in $\boldsymbol{m}_g^j$ to 0 and others to 1, then obtain $\boldsymbol{m}_g^{j+1}$.
4. Set $s_{\boldsymbol{\Theta}}$ % of the lowest magnitude values in $\boldsymbol{\Theta}_j$ to 0 and others to 1, then obtain $\boldsymbol{m}_{\boldsymbol{\Theta}}^{j+1}$.
5. Reset the unpruned weights in $\boldsymbol{\Theta}_j$ to their initial values in $\boldsymbol{\Theta}_0$), $j = j + 1$.
6. Loop step 2-5 until termination condition is satisfied, and get the winning ticket $f(\{\boldsymbol{m}_g^j \odot \boldsymbol{A}, \boldsymbol{X}\}, \boldsymbol{m}_{\boldsymbol{\Theta}}^j \odot \boldsymbol{\Theta})$.

Extend Pruning-at-initialization Methods.

As described above, GLT directly leverages the trained magnitudes in the lottery ticket hypothesis (LTH) as scores for GNN pruning. LTH verified the existence of sub-networks, and it obtained the winning tickets by pruning the network based on the magnitude after training. Recently-proposed pruning-at-initialization methods further abandoned the training part through devising different training-free saliency metrics (SNIP[126, 156], GraSP [127], Synflow [128]) to measure the importance of weights, and achieved comparable results as LTH. The success of GLT and pruning-at-initialization methods intuitively motivate us to extend these training-free saliency metrics to GNN

Algorithm 2 Pruning Graph via Edge Property (PGEP)

Require: Graph $\mathcal{G} = \{\boldsymbol{X}, \boldsymbol{A}\}$; Training labels $\mathcal{Y}_t = \{y_1, y_2, \ldots, y_t\}$, desired pruning ratios p_g, network mask $\boldsymbol{m}_{\boldsymbol{\Theta}}$.

Ensure: Sparsified Graph mask $\boldsymbol{m}_g$.

1: Train $f_{\boldsymbol{m}_{\boldsymbol{\Theta}} \odot \boldsymbol{\Theta}}(\boldsymbol{X}, \boldsymbol{m}_g \odot \boldsymbol{A})$ with $\mathcal{Y}_t$ until convergence.
2: $\hat{\mathcal{Y}} = f_{\boldsymbol{m}_{\boldsymbol{\Theta}} \odot \boldsymbol{\Theta}}(\boldsymbol{X}, \boldsymbol{m}_g \odot \boldsymbol{A})$.
3: $\boldsymbol{A}_{neg}(i,j) = \boldsymbol{m}_g \odot \mathbb{1}[\hat{\mathcal{Y}}(i) \neq \hat{\mathcal{Y}}(j)]$, $\forall i, j \in \{1 \cdots |\mathcal{V}|\}$.
4: Random remove $p_g \times |\boldsymbol{A}|$ of $\boldsymbol{A}_{neg}$ by setting corresponding $m_g = 0$. ▷ Find and remove negative edge
5: Find all non-bridges $\boldsymbol{A}_{non}$.
6: Random remove $p_g \times |\boldsymbol{A}|$ of $\boldsymbol{A}_{non}$ by setting corresponding $m_g = 0$.

pruning, by replacing the scoring mechanism in GLT with training-free saliency metrics. Experimental results verified the effectiveness of our extension, where these pruning-at-initialization methods can achieve similar performance as GLT for the GNN pruning.

6.4.2 Pruning Graph via Edge Property

Although existing pruning methods (including LTH, SNIP, GraSP, and Synflow) achieve competitive performance when jointly pruning the network weights and graph adjacency matrix, such saliency metrics (including GLT) could hardly outperform random pruning in most cases when solely sparsifying the graph. Experiments in Sec.6.5.2 demonstrate the failure of existing pruning methods on graph edges pruning, so here we follow the theoretical framework proposed in Sec.6.3 to introduce a new way to prune graph adjacency matrix. Specifically, we first train the network with the training labels $\mathcal{Y}_t = \{y_1, y_2, \ldots, y_t\}$, and predict all labels using the trained network with getting the predicted labels of all the nodes $\hat{\mathcal{Y}} = \{\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_{|\mathcal{V}|}\}$. Then, we prune negative edges iteratively until all such edges are pruned. The non-bridges are pruned afterwards until it reaches the target graph sparsity. We summarized our method in Algorithm 2.

6.4.3 Pruning Graph via Edge Property

Although existing pruning methods (including LTH, SNIP, GraSP, and Synflow) achieve competitive performance when jointly pruning the network weights and graph adjacency

matrix, such saliency metrics (including GLT) could hardly outperform random pruning in most cases when solely sparsifying the graph. Experiments in Sec.6.5.2 demonstrate the failure of existing pruning methods on graph edges pruning, so here we follow the theoretical framework proposed in Sec.6.3 to introduce a new way to prune graph adjacency matrix. Specifically, we first train the network with the training labels $\mathscr{Y}_t = \{y_1, y_2, \ldots, y_t\}$, and predict all labels using the trained network with getting the predicted labels of all the nodes $\hat{\mathscr{Y}} = \{\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_{|\mathcal{V}|}\}$. Then, we prune negative edges iteratively until all such edges are pruned. The non-bridges are pruned afterwards until it reaches the target graph sparsity. We summarized our method in Algorithm 2.

6.4.4 A General Two-Step Pruning Method for GNNs

When solely sparsifying the graph, the unified saliency metrics could hardly outperform random pruning in most cases, implying that the graph and network should use a different pruning strategy. Therefore, we propose an effective framework, a general two-step pruning method for GNNs (GTSP), to simultaneously reduce edges in the graph and parameters in the network. Specifically, for the graph pruning part, we use the proposed method in Algorithm 2 as this method significantly outperforms the saliency metrics methods for pruning the graph. For network pruning, we use iterative-SNIP [156] since experimental observations showed that this method can achieve similar accuracy to the GLH and found the winning tickets within seconds. In this case, the effectiveness of PGEP and the efficiency of iterative SNIP have successfully made up for the two major problems of GLT. We summarized our method in Algorithm 3.

6.5 Experiments

In this section, we answer the following questions sequentially by conducting extensive experiments with different pruning methods. **Q1:** Can the training free method achieve a similar pruning performance to GLT? **Q2:** Where the pruning benefit come from? Is that from the graph sparsification part and network pruning part together, or only from

Algorithm 3 A General Two-Step Pruning Method for GNNs (GTSP)

Require: Graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$; Loss function $\mathcal{L}$; Training labels $\mathcal{Y}_t = \{y_1, y_2, \ldots, y_t\}$; Desired network pruning ratios $p_{\boldsymbol{\Theta}}$. Iteration steps m.

Ensure: Graph mask $\boldsymbol{m}_g$ and network mask $\boldsymbol{m}_{\boldsymbol{\Theta}}$

1: **for** k in $[1, \cdots, m]$ **do**
2: Define $\hat{\boldsymbol{\Theta}} = \boldsymbol{m}_{\boldsymbol{\Theta}} \odot \boldsymbol{\Theta}$
3: $\boldsymbol{S}(\hat{\boldsymbol{\Theta}}) \leftarrow |\frac{\partial \mathcal{L}}{\partial \hat{\boldsymbol{\Theta}}} \odot \hat{\boldsymbol{\Theta}}|$ ▷ Score weight by SNIP
4: Compute $p_{\boldsymbol{\Theta}}$ th percentile of $\mathcal{S}(\hat{\boldsymbol{\Theta}})$ as t
5: $m_{\boldsymbol{\Theta}} = \mathbb{1}[\mathcal{S}(\hat{\boldsymbol{\Theta}}) < t]$
6: Get sparsified graph $\boldsymbol{m}_g$ with network mask $\boldsymbol{m}_{\boldsymbol{\Theta}}$ using Algorithm 2.
7: **end for**

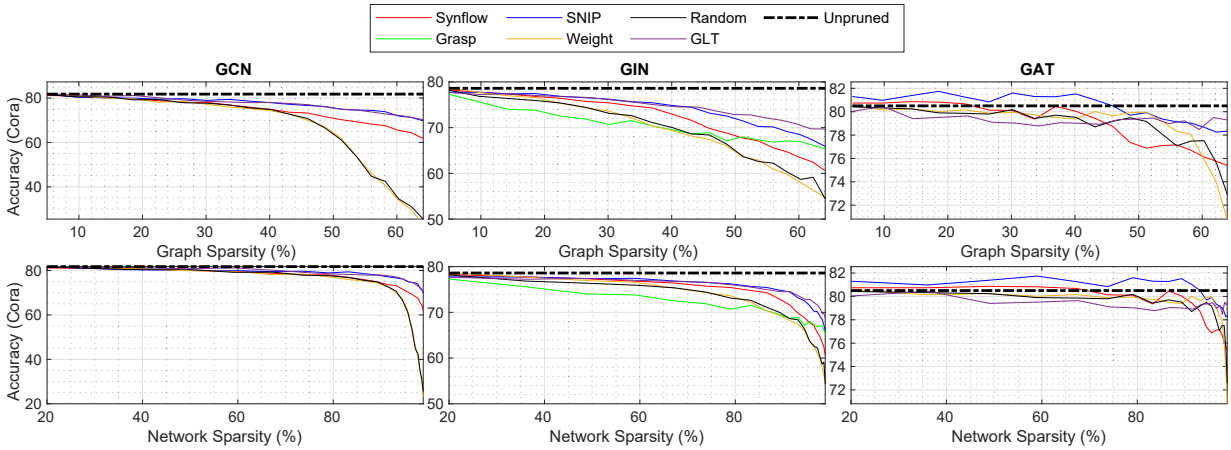

Figure 6.1: Pruning-at-initialization Methods performance when jointly pruning graph links and network weights over achieved graph sparsity levels and network sparsity of GCN, GIN, and GAT on Cora. Note: GraSP is no draw when layer collapse occurs.

a single part? **Q3:** Can the performance of our proposed GTSP in Sec.6.4.4 exceed the existing methods?

6.5.1 Dataset

6.5.2 The Training-free Pruning Methods on GNNs

We first apply the training-free pruning method (Synflow, Grasp, SNIP and magnitude before training) to GNN pruning, simultaneously pruning the graph connections and network weights.

We progressively prune GNNs and each time prunes 5% of graph links and 20%

Table 6.1: Test accuracies on Cora, Citeseer, PubMed for different pruning-at-initialization methods when jointly pruned 98% weights and 60% graph links.

Network	GCN			GIN			GAT		
Dataset	Cora	Citeseer	PubMed	Cora	Citeseer	PubMed	Cora	Citeseer	PubMed
Synflow	65.48	63.36	**78.20**	63.44	62.16	**76.88**	76.14	68.74	77.76
SNIP	**71.92**	**67.00**	75.90	68.46	**65.24**	75.48	**78.68**	**70.12**	**78.16**
Random	34.60	33.72	54.55	58.68	57.52	73.90	77.52	66.74	77.48
GraSP	13.00	7.70	18.00	66.98	61.06	70.90	13.00	7.70	18.00
Magnitude	34.20	32.66	58.90	57.98	57.76	74.70	75.98	67.48	77.78
GLT	**72.66**	**67.66**	**78.65**	**70.80**	**65.12**	76.10	**78.48**	**69.68**	**78.96**

Dataset	Nodes	Edges	Features	Classes
Cora	2708	5429	1433	7
Citeseer	3327	4732	3703	6
PubMed	19717	44338	500	3
Ogbn-ArXiv	169343	1166243	128	40

Table 6.2: Dataset Statistics

of network weights. Fig. 6.1 shows the test accuracies on Cora data set over achieved graph sparsity (first row) and network sparsity (second row). On GAT of cora dataset, SNIP initially outperformed the unpruned network. As the pruning progresses, the accuracy gradually decreases to the same performance as GLT. Since GNNs have few layers and GraSP did not design a mechanism to prevent layer collapse, it is more likely to occur layer collapse. In Table. 6.1, we summarize the accuracy of training-free pruning methods when 60% of the graph edges and 98% of the network weight are pruned for the Cora, Citeseer, and PubMed data sets. We found that Synflow and SNIP have similar performance to GLT on all networks and data sets, especially SNIP performs best. Among them, SNIP has a performance that exceeds GLT on GAT.

Overall, SNIP and Synflwo have a similar performance to GLT when pruning the graph and network of GNNs at the same time. SNIP persistently has good accuracy, while Synflow has a surprising performance on the PubMed data set. GraSP is unsatisfactory due to layer collapse [128].

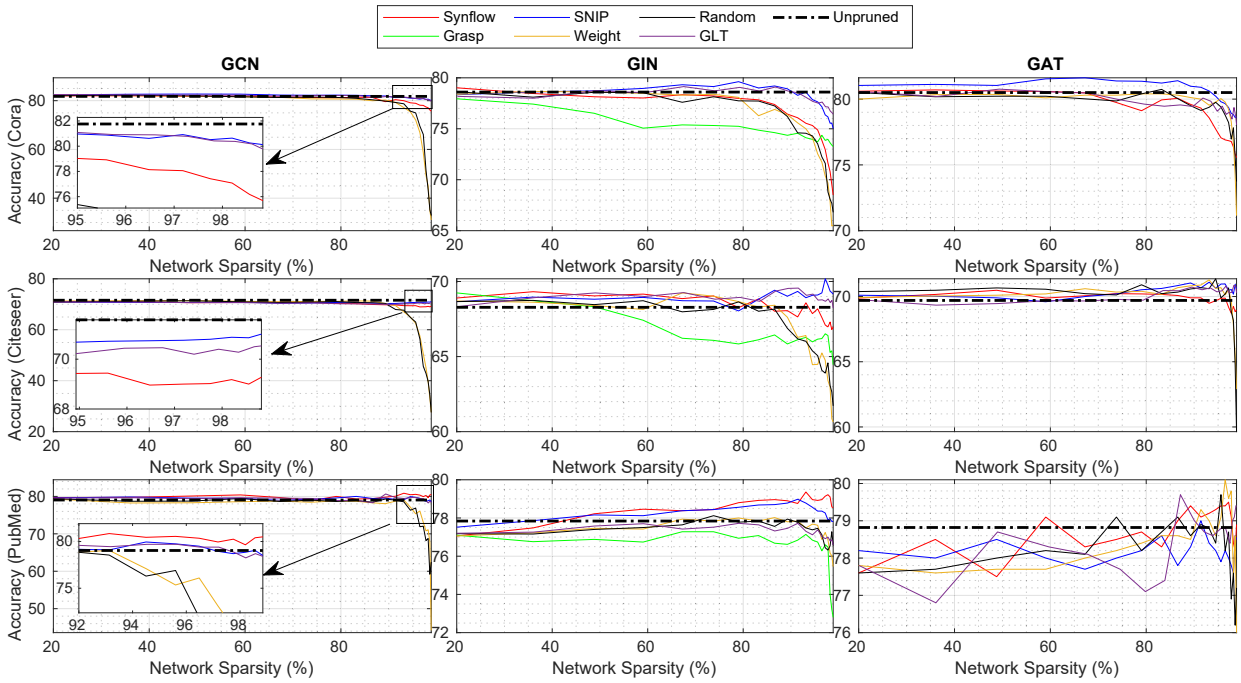

Figure 6.2: Pruning-at-initialization Methods performance when solely pruning network weights over achieved graph sparsity levels and network sparsity of GCN, GIN, and GAT on Cora, Citeseer, and PubMed datasets, respectively.

6.5.3 Ablations for graph pruning methods

In this subsection, we aim to understand where the pruning benefit come from? By individually pruning each part, we explore which part contributes to the performance gap between random pruning and the aforementioned methods.

Solely prune network weights. Fig. 6.2 demonstrated the test performance against network sparsity of these pruning-at-initialization methods when solely pruning the network weights. The results of SNIP and Synflow in all experiments far exceed the results of random pruning, and have similar results to GLT. In GCN, random pruning began to show a significant performance decline when the network parameters were pruned more than 90%. In contrast, the accuracy of SNIP, Synflow and GLT have not dropped until the sparsity reaches 98%. On PubMed, after using SNIP and Synflow pruning, with the decrease of network parameters, the prediction accuracy rate increased, among which Synflow performed the best. Overall, we believe that these pruning methods

are capable for prune network weights in GNNs.

Solely prune graph links. We then discuss the situation when only pruning graph links. We conducted the experiments that only prune graph links while the network weights remain unchanged. The results can be found in Fig. 6.5. Surprisingly, all these pruning methods, including SNIP, Syflow, GraSP, GTL, magnitude before training, hardly surpass random pruning. Especially in GAT, random pruning is even better than some specially designed scoring standards. GraSP did not experience performance degradation in the early stages, because only pruning the graph links did not trigger layer collapses. Although in a few cases, such as GCN's Citeseer, GLT is slightly outperformed random pruning, the overall accuracy downward trend remains unchanged. Overall, graph link pruning has a scarce contribution to GNN pruning and existing pruning methods are unsatisfied for graph link pruning.

In summary, based on the above experimental phenomenon in Sec.6.5.2 and 6.5.3, we argue that the benefit of GLT and pruning-at-initialization methods when jointly pruning graph adjacency matrix and network weight was brought by weight pruning. Thus, graph links and network weight should use different scoring standards during GNNs pruning.

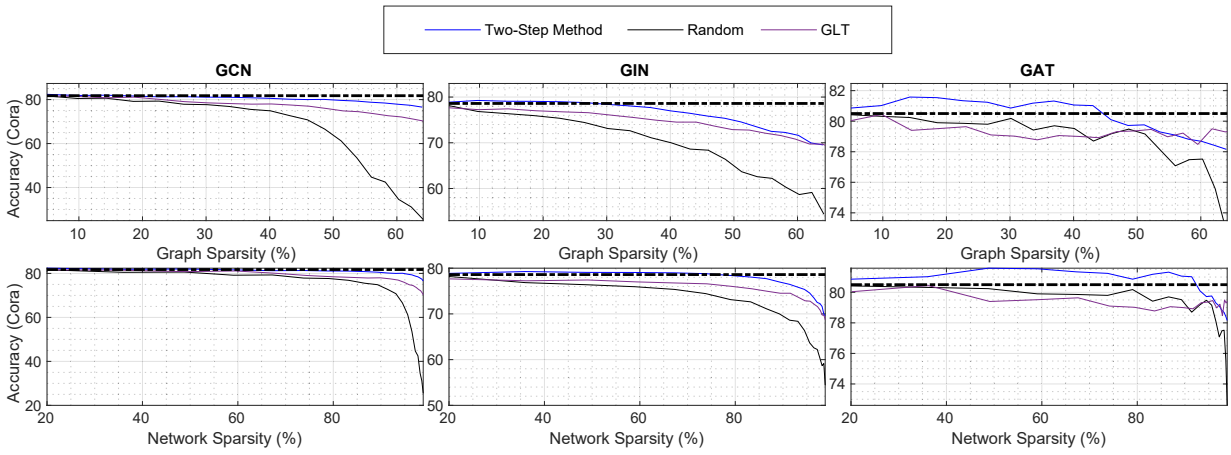

Figure 6.3: The General Two-Step Pruning method performance when jointly pruning graph links and network weights over achieved graph sparsity levels and network sparsity of GCN, GIN, and GAT on Cora.

Table 6.3: Test accuracies on Cora, Citeseer, PubMed for the General Two-Step Pruning methods at different sparsities.

Sparsity	**Network**	GCN			GIN			GAT		
Graph/ Weight	**Dataset**	Cora	Citeseer	PubMed	Cora	Citeseer	PubMed	Cora	Citeseer	PubMed
60%/ 98%	GLT	72.66	67.66	78.65	70.8	65.12	76.1	78.48	69.68	**78.96**
	Random	34.6	33.72	54.55	58.68	57.52	73.9	77.52	66.74	77.48
	GTSP (Ours)	**77.88**	**68.48**	**79.1**	**71.62**	**65.64**	**77.6**	**78.68**	**70.04**	78.1
49%/ 95%	GLT	75.94	68.74	**79.20**	72.88	**66.40**	75.70	79.36	69.96	**79.10**
	Random	66.56	57.94	71.30	66.38	61.28	74.60	79.48	**71.04**	79.00
	GTSP (Ours)	**80.08**	**69.58**	**79.20**	**75.36**	**66.18**	**78.15**	**79.72**	70.16	78.50
40%/ 90%	GLT	78.00	**70.32**	78.10	74.52	**67.84**	77.60	79.00	69.46	77.60
	Random	74.88	67.10	75.90	70.00	64.28	76.20	79.52	**70.98**	78.80
	GTSP (Ours)	**80.56**	**69.70**	**79.50**	**76.98**	66.52	**78.20**	**81.06**	**70.12**	**79.10**

6.5.4 A General Two-Step Pruning Method for GNNs

In Sec.6.5.3, we showed that saliency metrics are not suitable for scoring the graph edges. In this section, we evaluate the performance of the proposed method when simultaneously pruning graph link and net weight.

Fig. 6.3 shows the test accuracies comparison between the proposed method, GLT and random on Cora data set over achieved graph sparsity (first row) and network sparsity (second row). On GAT of Cora dataset, our method outperforms unpruned networks. Table. 6.3 summarized the test accuracy when the GNNs is pruned at a three different sparsity. On all data sets and networks, our method has greatly improved compared to the baseline methods. On the Cora dataset, the maximum performance improvement in the case of matching sparsity is 5.22%.

In general, our algorithm exceeds baseline performance in all aspects due to the excellent graph pruning method. In addition, as the training free pruning method is used when pruning the network weights, the weight pruning process can also be completed within a few seconds, which greatly reduce the computational complexity of pruning.

6.5.5 Ablations on pre-trained GNNs

Though the recent study [160] shows the pruning at initialization methods are underperformed on CNNs, the performance did not drop on GNN. We provide the comparison

between leveraging the scores in an initialized GNN and a pretrained GNN in Table 6.4. We tested two saliency scores, SNIP and Synflow. As shown, there is no significant performance drop between pruning before training and after training when solely pruning network weights, and before training even achieves better results in many cases. From the result shown in Tabel 6.4, we concluded that pruning-at-initialization methods do not under-perform the traditional pruning after training for shallow GNNs.

Table 6.4: Performance compare by applying SNIP, and SynFlow before training (BF) and after training (AT). Pruned ratio: 95%.

		GCN		GIN		GAT	
		AF	BF	AF	BF	AF	BT
Cora	SNIP	80.95	**81.04**	**77.90**	77.86	80.15	**80.22**
	Syn	79.10	**79.12**	75.30	**75.32**	78.98	**79.08**
Citeseer	SNIP	**70.44**	70.22	68.00	**69.16**	70.36	**70.96**
	Syn	**69.51**	69.42	**68.28**	67.86	**70.30**	69.86
PubMed	SNIP	80.17	**80.45**	**78.63**	78.56	**77.80**	78.20
	Syn	79.55	**79.70**	**79.22**	78.96	79.15	**79.30**

As to why this observation in GNN is different from deep CNNs, [160] and [161] pointed out that the useful information that pruning methods extract is not which individual weights to remove, but rather the layerwise proportions to prune in the network. [161] experimentally proved that how to obtain the proportion of each layer is more important. Training-free methods can get very similar proportions as the training method [160], however, deep CNNs (VGG-16, ResNet-50) are more sensitive to the proportions, and the slight difference brings marginal improvement over training-free methods. In contrast, for GNNs used in this work, which are much smaller than deep CNNs, the proportions gained by the training-free method are sufficiently informative and the insignificant difference in proportions between training and training-free methods won't affect the accuracy.

6.5.6 Complementary experiments

In this section, we aims to provide extra experiments on different data sets. Fig. 6.4 is a supplement to Fig. 6.1 and Fig. 6.7 is a supplement of Fig. 6.3. Fig. 6.5 shows these pruning methods (Synflow, Grasp, SNIP, magnitude before training and GLT) when solely pruning the graph links.

Specifically, Fig. 6.4 shows the test accuracies on Cora, Citeseer, PubMed data set over achieved graph sparsity (first row) and network sparsity (second row). We found that Synflow and SNIP have similar performance to GLT on all networks and data sets. Specificlly, on the PubMed data set, Synflow dominates from pruning starts until graph sparsity and network sparsity reach 65% and 99%, respectively. Surprisingly, the accuracy of Synflow increased with the pruning continuum. On GAT cora dataset, SNIP initially outperformed the unpruned network. As the pruning progresses, the accuracy gradually decreases to the same performance as GLT. Since GNNs have fewer layers and GrsSP did not design a mechanism to prevent layer collapse, it is more likely to occur layer collapse.

Fig. 6.5 demonstrated the test performance against graph sparsity of these pruning-at-initialization methods when solely pruning the graph links. Surprisingly, all these pruning methods, including SNIP, Syflow, GraSP, GTL, magnitude before training, hardly surpass random pruning. Especially in GAT, random pruning is even better than some specially designed scoring standards. GraSP did not experience performance degradation in the early stages, because only pruning the graph links did not trigger layer collapses. Although on a few cases, such as GCN's Citeseer, GLT is slightly outperform random pruning, the overall accuracy downward trend remains unchanged. In summary, graph link pruning has scarce contribution to GNN pruning.

We evaluate the performance of the proposed method when simultaneously pruning graph link and net weight. Fig. 6.7 shows the test accuracies comparison between the proposed method, GLT and random on Cora data set over achieved graph sparsity (first row) and network sparsity (second row). On all data sets and networks, our method has greatly improved compared to the baseline methods. On the Cora dataset, the maximum

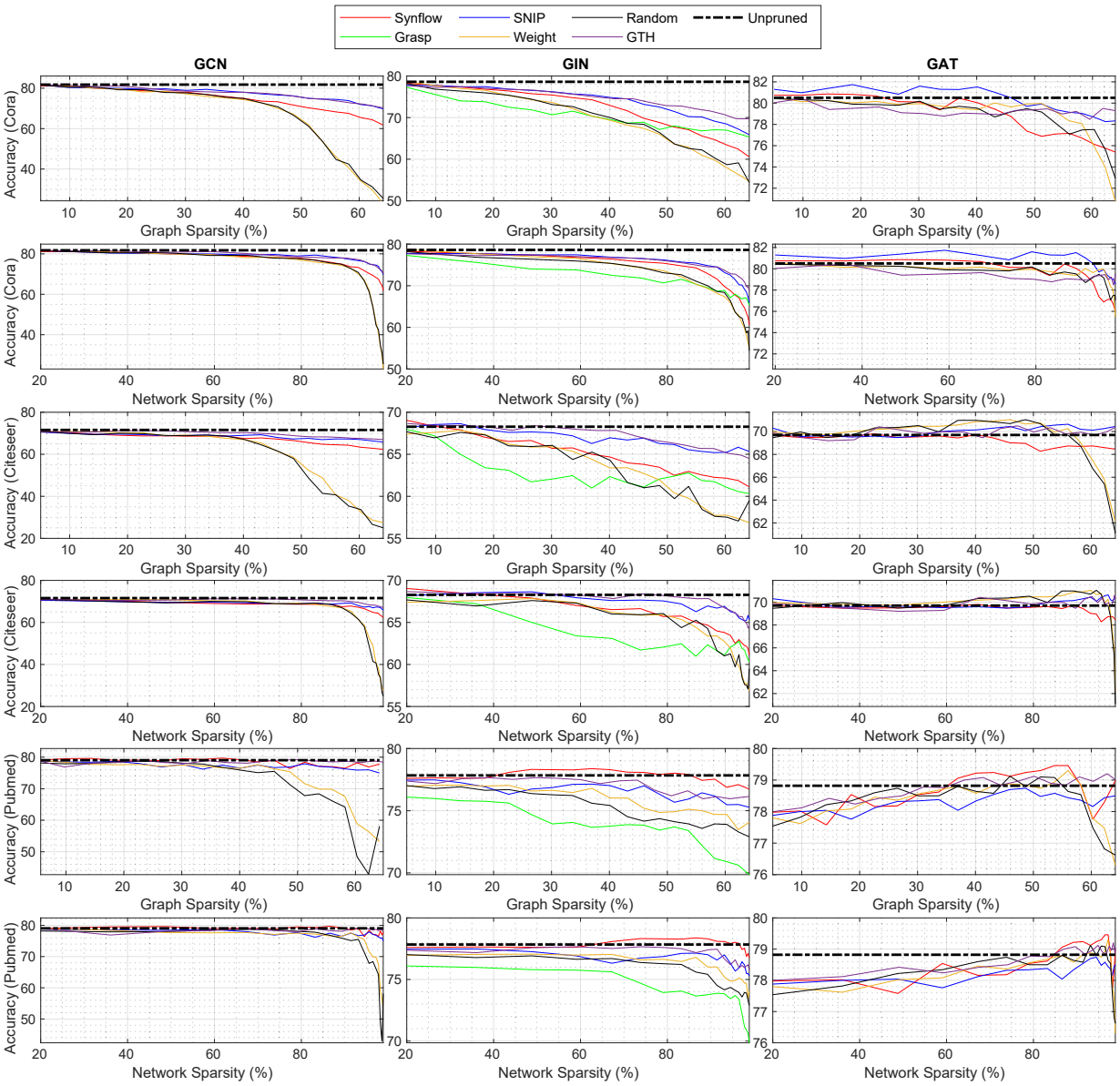

Figure 6.4: Pruning-at-initialization Methods performance when jointly pruning graph links and network weights over achieved graph sparsity levels and network sparsity of GCN, GIN, and GAT on Cora, Citeseer, and PubMed. Note: GraSP is no draw when layer collapse occurs.

performance improvement in the case of matching sparsity is 6.36%. On and GIN network with PubMed dataset, our method outperform unpruned network until 60% graph link and 97.75% network parameters were pruned.

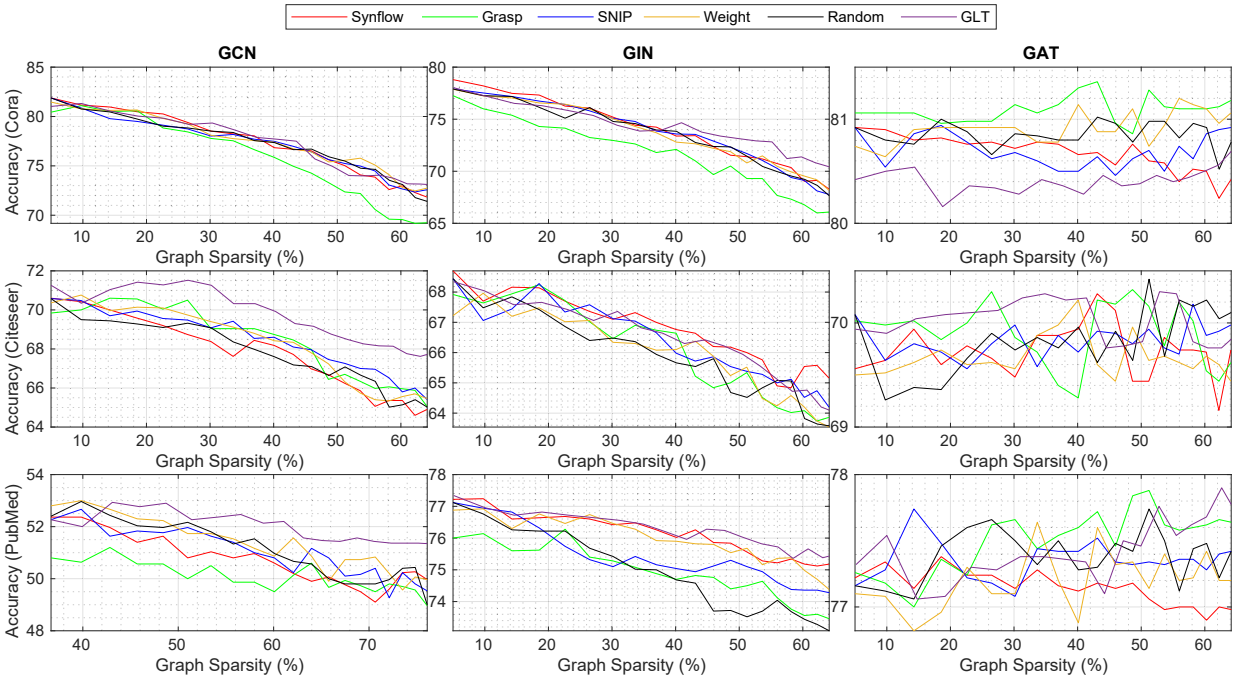

Figure 6.5: Pruning-at-initialization Methods performance when solely pruning graph edges over achieved graph sparsity levels and network sparsity of GCN, GIN, and GAT on Cora, Citeseer, and PubMed datasets, respectively.

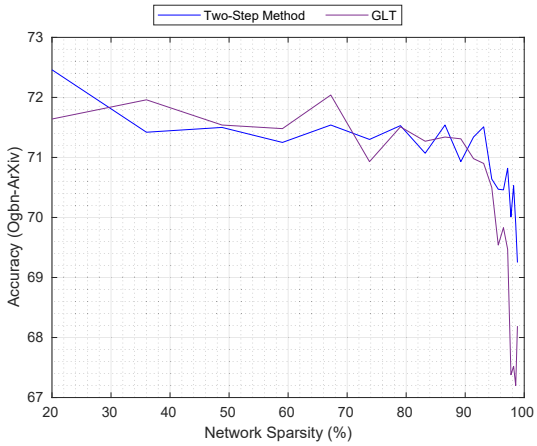

Figure 6.6: The performance of 28-layer deep ResGCN on large-scale graph data set.

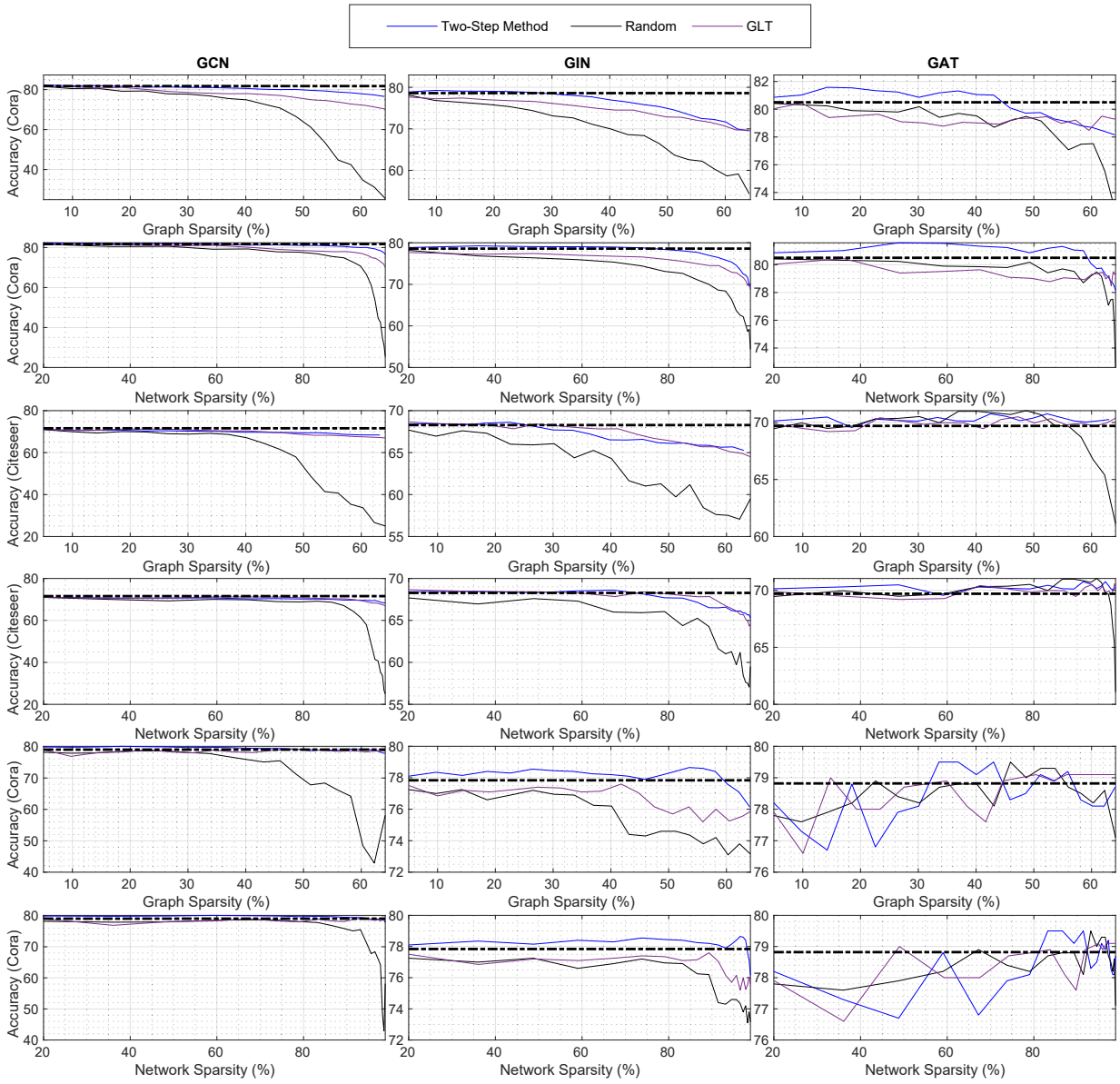

Figure 6.7: The General Two-Step Pruning method performance when jointly pruning graph links and network weights over achieved graph sparsity levels and network sparsity of GCN, GIN, and GAT on Cora, Citeseer, and PubMed

6.5.7 Large-scale Graphs with 28-layers ResGCN

Following experiment settings in [48], we also extend the proposed method on large scale data set and deeper GNNs. Fig. 6.6 demonstrates that the proposed method can be scaled up to deep GNN on large-scale graphs. The performance is comparable at the beginning of pruning process and is better than the baseline method at extreme sparsity.

6.6 Findings

GNNs are Heavily Over-parameterized. Since GNN is usually very shallow, it is generally believed that GNN does not have great pruning potential. Through experiments, we found that GNN is also heavily over-parameterized, where after removing 98% of the network weights, the accuracy of GNN still holds. This means removing the unnecessary weights in GNNs can significantly reduce the inference computation complexity. With the emergence of deeper GNNs, the reduction in computational complexity will bring more substantial benefits.

Easy Saliency Metric Can Get Good Performance. The experiment results in Sec.6.5.2 indicated that using a simple saliency metric such as SNIP can achieve a good pruning effect. Notably, these training-free approaches are considerably simpler than GLT, with no requirements of pretraining.In general, the training-free method, such as SNIP and Synflow, can achieve similar performance to the after training pruning method on GNNs.

Unified GNN Pruning Method Remain Challenging. The experimental results in Sec.6.5.3 verified that existing saliency metrics cannot evaluate the importance of graph links, so we need different scoring criteria to evaluate graph edges and neural network parameters separately. This makes it difficult to find a unified pruning method for GNNs. The graph pruning method we designed is not a train-free method, which remains an open problem for future work.

6.7 Conclusion

This chapter extended several existing training-free saliency metrics (SNIP, GraSP, Synflow) to GNNs pruning, and found that these methods can achieve similar performance as GLT when jointly pruning the network weights and graph adjacency matrix, showing that these training-free saliency metrics can substitute LTH with extremely reducing the computational cost. However, when solely sparsifying the graph, such saliency metrics (including GLT) could hardly outperform random pruning in most cases. Thus, we argue that graph links and network weight should use different scoring standards during GNNs pruning. Rather than measuring the importance of graph edges based on saliency metrics, we mathematically formulate the performance of graph neural network with respect to the properties of edges, elucidating how the loss of performance can be avoided by pruning negative edge and non-bridge. This leads to our simple but effective two-step method for graph neural network pruning, leveraging the saliency metrics for the network pruning while sparsifying the graph by preserving the loss. Experimental results verified the effectiveness and efficiency of the proposed method, where our method can prune up to 60% of graph links and 98% of network weights on different tasks with no significant accuracy drop.

CHAPTER 7

RELIABLE TEMPORAL GAIT PARAMETER ESTIMATION

7.1 Introduction

In this chapter, we demonstrated an in field application of the Fit.E, analysing the contact time, flight time, and asymmetric of 40 rugby players from Sydney Swan Football Club during running.

During running, the detection of two essential incidents is required to perform gait analysis on each step of the running process, they are heel stride or initial contact (IC) and toe-off or terminal contact (TC). By finding these two events, the main temporal parameters of every stride like pace, contact time, flight stage length, and swing stage length can be determined [162]. IC is identified at the instance when one of the foot is initially touching or landing on the surface of running. On the other hand, TC represents the termination of the pushing phase, meaning when one of the foot finishes its contact with the surface. There is a wide range of research finding and proving the intrinsic relationships between these temporal parameters and other critical factors related to running, such as the running economy [163], performance of running [164] and RRI risks [165]. Thus, a precise recognition of IC and TC is paramount.

In the early stage of gait research, many laboratory tools were used to detect the

parameters that are mentioned above. For example, [166] conducted a study to capture and to analyse gait patterns using a single camera system. Similarly, another study carried out by [167] recorded the gait movement signal using wearable sensors. However, these studies using laboratory equipment have several limitations. [168] conducted a symmetric review of the current gait analysis method and suggested that earlier approaches of gait analysis required appropriately equipped laboratories with high-speed cameras or force pads. However, these requirements could bring many limitations on the gait study as the in-lab environment could hardly simulate the outdoor situation. Additionally, such equipment is difficult to set up or to be accessed by most of the runners. Hence, there is a need to develop a gait analysis method with affordable and portable tools, allowing gait studies to be carried out in different locations and for a more extended period to gather data that can be applied to real-life exercise [169].

Herewith, the application of inertia measurement units (IMU) such as gyroscopes and accelerometers can be an excellent solution to the problem. This is because they can capture the resistance in linear or angular motions[170] and they have low manufacturing cost, reduced size and weight, and they can monitor kinematic throughout an extended period [163]. These advantages have made IMU a widely used system to investigate human movement.

There are many ways to collect IC and LC using IMU. Currently, the approaches differ in the IMU placement and the way of how to utilise the data that are collected. There are several sites where IMU is located among different studies. [171] conducted a review of the current gait research, and they found that the major locations of placing IMU are pelvis, shank, and foot. They concluded that particular sites of IMU placement requires different data analysis methods to detect IC and LC, which could result in diverse accuracy. As for the detection method, they are mainly IMU or Gyroscope based. [172] investigated tibial acceleration around the knee joint, in which they are most interested in the mediolateral plane, thus they studied varus/valgus knee movement throughout running. [173] studied gait movement using accelerometers and established that accelerometer positioning was vital in delivering accurate analysis. In terms of

using gyroscopes, [174] analysed the difference between unprofessional and elite runners using an IMU containing a tri-axial gyroscope located at the lower back (L1) area.

From this literature review, it is found that most of the studies use either accelerometer or gyroscope or focus on the different placement of the IMU. However, the accuracy of using gyroscope and accelerometer data at the same location is not compared. Furthermore, no research method proposed a system to detect IC and LC using both gyroscope and accelerometer.

Therefore, this study aims to determine the performance of the adaptive algorithms built based on data from gyroscope and accelerometer relatively for the detection of IC and LC during running, and hence to propose algorithms that are based on each method. Furthermore, to increase and analyse the prediction accuracy, we included a pre-processing algorithm that detects the period of running to remove unwanted data such as standing still or putting on the IMU. Likewise, we used confidence intervals to analyse the systems,Äô consistency and to further enhance the system and its accuracy.

7.2 Method

7.2.1 Data pre-processing

The collected raw data from the IMU contains noise and data from untargeted periods such as putting on the IMU, walking or standing still. These sections of the data are needed to be removed for the running stance phase detection. Hence, we introduce a pre-processing algorithm that can detect the running period. This algorithm first calculates the square value of the acceleration in the three directions as shown in Eq.(7.1), the square value amplifies the difference between non-running period and running period. This algorithm contains a sliding window of 0.5s and it compares the average square value of the sum of accelerations in three directions to a threshold value within the window. If the value output exceeds the threshold for more than 3s, it will be marked as the beginning of running, and when the value drops below the threshold, it is marked as the end. The calculation of the sliding average value is shown in Eq.(7.2).

$$A_{res} = (Ax^2 + Ay^2 + Az^2) \tag{7.1}$$

$$A_{threshold} = \frac{\sum_{n}^{n+s*f}(A_{res})^2}{s*f} \tag{7.2}$$

where $A_{threshold}$ is the output value that is compared to the threshold. f is the sample frequency of the IMU. A_x, A_y, A_z are the linear acceleration value in three directions. S is the time frame of the sliding window.

7.2.2 Algorithm design

Based on the data collected from gyroscope and accelerometer, two different algorithms were built for IC and TC detection.

7.2.2.1 Accelerometer-based algorithm

In the accelerometer-based IC and TC detection, the algorithm was built upon the acceleration data on Z-axis. Before plotting the graph, the square value of the acceleration was found to amplify the data and the graph trend. The instant of IC was detected at the peak of foot-resultant acceleration as shown in figure 7.1. The detection of the TC was conducted in the region of interest when the fluctuation of the z-axis acceleration started to show a gentle trend after the peak point (IC). Figure7.1 represents the TC point and the area of interest for the TC detection.

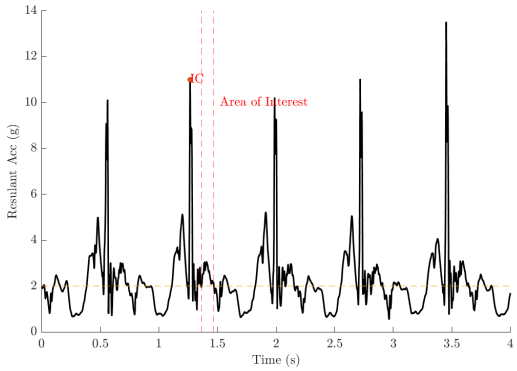

Figure 7.1: Random samples of right ankle acceleration value in g A_{ankleR} of a participant during 10km/h running. Note: the peak resultant acceleration is marked as IC, and the 2g-threshold is the area of interest for TC detection.

During running, as the speed increases, the stride length of the runner tends to decrease, hence the ground contact time is often less than 50 percent of the total stride time during high speed running [175]. Therefore, to find the exact point of TC, we designed a window to highlight the area of TC detection. This window has a length of N samples, which is half of the total number of samples in a specific stride, and the window is located between 25 percent to 75 percent of the stride, Eq.(7.3) shows the calculation of N. After that, all the data within this region was normalized using Min-max normalization and scanned until the first point that exceeded the empirical threshold had been reached. That point was marked as the TC.

$$N_{window} = \frac{1}{2} * (Peak_{n+1} - Peak_n) \tag{7.3}$$

The pseudo-code of the acceleration algorithm on terminal contact points detection is shown in Algorithm 4:

Algorithm 4 Finding TC using Acc data

Output: TC

$StartTime \leftarrow fromPre-processing$
$EndTime \leftarrow fromPre-processing$

$AccZ \leftarrow AccelerationZ(From\ StartTime\ to\ EndTime)$
$AccY \leftarrow AccelerationY(From\ StartTime\ to\ EndTime)$
$AccX \leftarrow AccelerationX(From\ StartTime\ to\ EndTime)$

$resAcc \leftarrow \sqrt{AccX^2 + AccY^2 + AccZ^2}$
$SqreDiffAcc \leftarrow ((resAcc(2) - resAcc(1)),(resAcc(3) - resAcc(2)),,Ä¶,Ä¶(resAcc(n) - resAcc(n-1)))^2$
$[PeakValue, InitialContactx] \leftarrow localmaximum(SqreDifffAcc)$

$Flag \leftarrow 0$
$Count1 \leftarrow 1$
$Count2 \leftarrow 1$
$Peakx \leftarrow 0$

for $i \leftarrow 0\ to\ Length(SqreDiffAcc)$ **do**
  if $Flag = 0\ \&\ SqreDifffAcc(i) = PeakValue(Count1)$ **then**
    $Flag \leftarrow 1$
    $Peakx \leftarrow i$
    $INCREMENT\ Count1$
  else
    if $Flag = 1\ \&\ i = (Peakx + (InitialContactx(Count1) - Peakx)/4)$ **then**
      for $h \leftarrow i\ to\ (Peakx + (InitialContactx(Count1) - Peakx)/2)$ **do**
        $NrmlsACC \leftarrow (SqreDiffAcc(h) - Minimum(SqreDiffAcc(h\ to(Peakx + (InitialContactx(Count1) - Peakx)/2))))/(Maximum(SqreDiffAcc(h\ to(Peakx + (InitialContactx(Count1) - Peakx)/))))) - Minimum(SqreDiffAcc(h\ to(Peakx + (InitialContactx(Count1) - Peakx)/2))))$
        if $NrmlsAcc \geq EmpiricalThreasholdAcc$ **then**
          $TC = h$
        end if
      end for
      $TC(Count2) \leftarrow TC$
      $INCREMENT\ Count2$
      $Flag \leftarrow 0$
    end if
  end if
end for

7.2.2.2 Gyroscope-based algorithm

The precise detection of leg moments such as initial contact (IC), terminal contact (TC), and mid-swing(MS) is essential for the computation of temporal parameters including stride time, swing and stance duration in gait analysis. Such instance can create unique signal characteristics that both acceleration and angular velocity data show distinctively positive or negative peaks consisting of medium to rather high frequencies.The instance of IC and TC is defined at the first and the second local maximum, where MS is the local minimum that is located after TC, the locations of IC, TC and MS are shown in figure 7.2.While the scale of such peaks are affected by several factors such as the participant,Äôs running intensity or speed, they tend to show similar tends among different runs and hence can be detected in the specific area of interest or frequency domains. Following previous work of gait analysis by [176], who proposed a detection algorithm by first detecting the location of mid-swing (MS), followed by searching backwards for TC and forwards for IC.

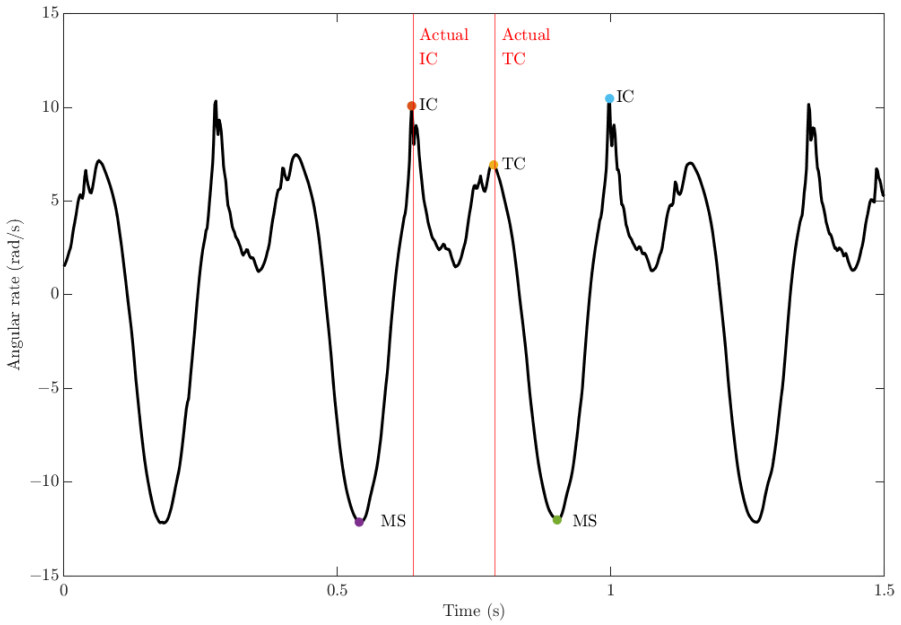

Figure 7.2: Random samples of right ankle angular rate $\omega ankleR$ of a participant during 10km/h running. Note: MS = Mid-Swing; IC = Initial-Contact; TC = Terminal-Contact.

The proposed algorithm first detects TC to minimise the computational requirements for backward searching. To enhance and detect the mid-swing peak, a 2nd-order Butterworth low-pass filter was designed with a cut-off frequency of 10 Hz. In the detection of the possible location of TC, the proposed system first detects the first local maximum point. After that, the algorithm search for the presence of a local minimum point after TC which is the MS. Next, the following local maximum is then recognised as the location of IC. These possible points (PP) of gait event location are then evaluated under the set of rules shown in Table 7.1.

Table 7.1: The detection logic and conditions for IC, MS and TC detection.

Gait event	Conditions
TC	TC must fulfil below conditions: a) It is the local minimum b) A local maximum (MS) is after TC_{pp} c) A local minimum IC is after MS d) It is the local minimum between MS n and MS n-1
MS	MS must fulfil below conditions: a) It is the local maximum b) MS (n, 1) ‚Äì MS (n-1, 1) > 300ms
IC	IC must fulfil below conditions: a) A MS is identified before the location of IC b) IC (n,1) ‚Äì TC (n,1) >100ms

7.2.2.3 Confidence Interval

To test the accuracy and the consistency of the output of gyroscope and accelerometer algorithms, confidence interval (CI) is used to perform comparison and to verify the range of the difference between the two algorithms. In the case of a CI in a particular measure X. The sample size is defined as N, where m is defined as the hypothetical average value or mean, and s is the standard deviation. The pre-defined confidence level is represented as $1-\alpha$, in this study, we used 95 % confidence level. Hence, the CI of the given sample or the range of values can be defined by: m,àí$c_\alpha s < X < m + c_\alpha s$, and as $\alpha = 0.05$, then $c\alpha = 1.96$. In this project, to test the consistency between the two algorithms, a right-side test was conducted to the difference between the calculated contact time of the two algorithms ($H_0 : \mu \leq \mu_0 \mid H_1 : \mu > \mu_0, \mu_0 = 0.03 - 0.05$). The average of the absolute difference between the two algorithms $\bar{x}_{diff}$ is calculated by Eq.(7.4). The z-score value is then calculated using Eq.7.5 and the output Z value is then compared to the Critical Value $Z\alpha/2$ which is 1.654.

$$\bar{x}_{diff} = \frac{\sum_{i=1}^{n} |CT_{Gyro} - CT_{Acc}|}{n} \tag{7.4}$$

where $\bar{x}_{diff}$ is the average value of the absolute difference between the outputs from the two algorithms. CT_{Gyro} and CT_{Acc} are the calculated contact time output from the two algorithms.

$$Z = \frac{\bar{x}_{diff} - \mu_0}{s/\sqrt{n}} \tag{7.5}$$

where z is the z-score value of the current $\bar{x}_{diff}$, Œž$_0$ is the maximum tolerance of the $\bar{x}_{diff}$ value, and s is the standard deviation of the $\bar{x}_{diff}$ dataset.

The output z-score value is then accessed and compared to test if the system‚Äôs consistency is within the confidence level of 95% where the maximum inconsistency of the two algorithms is 0.03s.

If $Z \leq Z_{\alpha/2}(1.645), \mu \leq \mu_0, H_0$: the system‚Äôs consistency is within the confidence level.

If $Z > Z_{\alpha/2}(1.645), \mu > \mu_0, H_1$: the system,Äôs consistency is not within the confidence level.

7.3 Results and Discussion

7.3.1 Data collection Protocol

In total, 40 rugby players from Sydney Swan Football Club were asked to run at three different speeds. All the participants were healthy and do not suffer from any symptomatic musculoskeletal injuries. The 40 rugby players were volunteered to take part in this data collection protocol. These trails were conducted in a laboratory provided by the Faculty of Health of the University of Technology Sydney. This study was approved by the ethics committee of the University of Technology Sydney. Each player was requested to run 10 meters with three different trails at different speed of 10kmh, 15kmh and 20kmh. The data collection was carried out on an instrumented treadmill that was connected to a camera system that has 60 FPS. During running, all participants were wearing their own comfortable shoes. An IMU device (ImeasureU) was attached to the ankle area of each foot of the participant. The IMU device was consisted of an accelerometer ($500HZ$, $\pm$ $200g$ recording range), a gyroscope ($500HZ$, $\pm$ $35rad/s$ recording range) and a barometric sensor. In this study, only the data collected from gyroscope and the accelerometer was utilized. A 6 min familiarization period was conducted on the same treadmill with identical setup as a warm-up for each player. A rest period was given between each trail for all participants, and they were free to determine the duration of resting. Figure7.3shows the attachment of the IMU device on the ankle area.

Figure 7.3: The placement of the IMU device on the ankle area of a participant.

Additional trails were carried out with a camera-based system and setup, the participants were asked to repeat the previous trials with the same setup and protocols with the camera capturing the feet transitional motion of the participant. The collected data was used as a validation data set to test the validity and accuracy of the two algorithms. In total, a sample which consisted of 900 steps were recorded. Within this data set, 180 of the steps were recorded with the camera-based system and were used as the ground truth to test the algorithm accuracy, whereas the 720 steps were used for algorithm construction and consistency testing.

7.3.2 Data Pre-processing

The purpose of the pre-processing algorithm was to filter out the non-running data ranges and to output the running period start time and end time for IC and TC detection in the later part of the algorithm. Figure 7.4 demonstrates a random sample of a participant's whole running data, including all three speeds (10km/h, 15km/h and 20km/h). The algorithm clearly distinguished the three different running period form the whole data set which was 370s. As the sliding window has a length of 0.5s and the algorithm used average value within the window, the output value could have slight error of which was

up to 0.5s which could affect the detection of IC and TC on a specific step. However, one step can be considered as a minor influence to the algorithm's accuracy to a data set that was consisted with a large amount of steps.

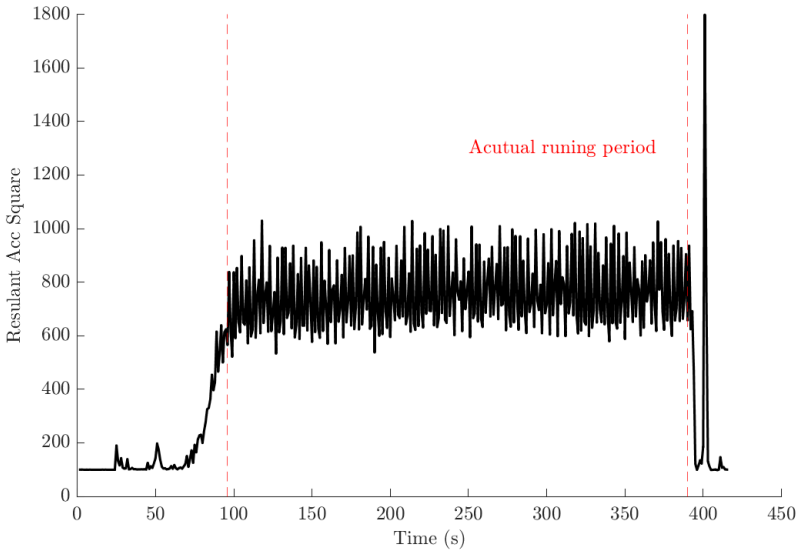

Figure 7.4: Random sample of a participant's pre-processed data with three different running speeds (10km/h, 15km/h and 20km/h).

Table 7.2: The consistency of the two systems under different speeds and maximum tolerance values of the confidence interval.

		Percentage of datasets within the maximum tolerance value with 95% confidence interval(s)			
		0.05	0.04	0.03	Average
Speed (KM/H)	10	100%	95	86.5%	93.75%
	15	100%	96.25%	88.75%	95%
	20	100%	92.25%	93.75%	91.67%
	Average	100%	94.17%	86.26%	

7.3.3 Algorithm consistency

To test the consistency of the gyroscope and accelerometer algorithms, 95% right-sided confidence intervals with different maximum tolerance value (0.05s, 0.04s and 0.03s) were constructed on the difference between the output value of the two algorithms. Table 2 summarizes the consistency of the two algorithms under different speed. The consistency was calculated by finding the percentage of datasets that has a lower difference than the given tolerance under 95% confidence level between the output of the two algorithms. As the results are shown in Table 7.2, a 100% consistency between the two systems was found for all speeds with a maximum difference value of 0.05s. An average consistency of 94.17% was found for all datasets under maximum difference of 0.04s and for 0.03s tolerance, an average consistency of 86.26% was found. The two algorithms showed the highest consistency at the speed of 15 km/h, and the average consistency is 95%. From the results, we found that the consistency of the two algorithms reduced sharply as the running speed increased to 20 km/h. On top of this, the highest consistency for each tolerance group was found at the running speed of 15 km/h, meaning the two algorithms were more consistent at this speed level.

7.3.4 Accuracy of Detection

The output values of the two algorithms were then compared to the video reference to test the validity of the systems. The results were recorded in Table 7.3. The mean errors of the accelerometer algorithm are 0.0159s, 0.008s, 0.015s for speeds of 10km/h, 15km/h and 20km/h relatively. The mean errors of the gyroscope algorithm are 0.0239s, 0.0214s, 0.023s for speeds of 10km/h, 15km/h and 20km/h respectively. Like the result of the consistency between the algorithms, the speed of 15km/h shows the lowest mean error for both accelerometer and gyroscope.

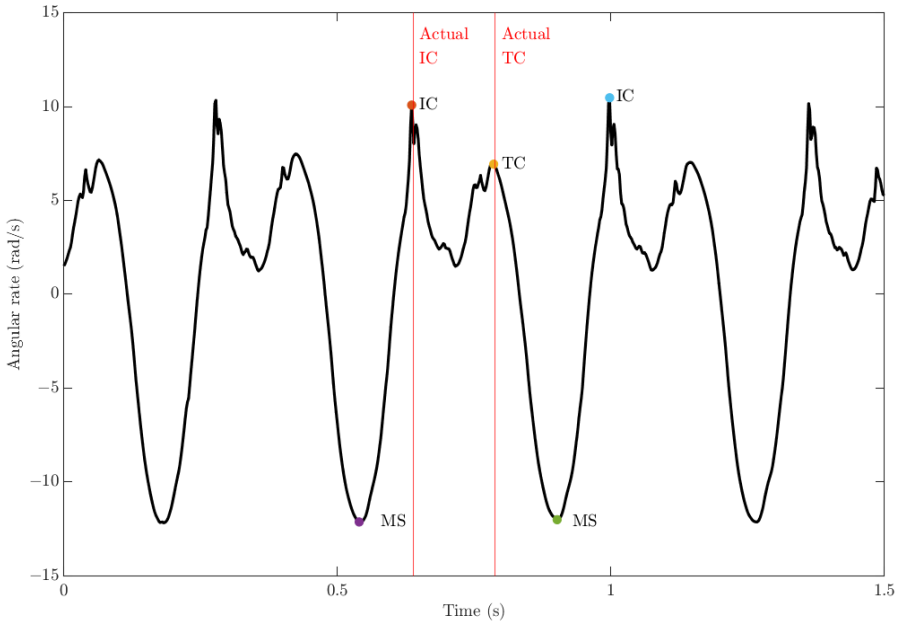

Figure 7.5: Comparesion between actual result (red line) and estimated results (dots). Note: MS = Mid-Swing; IC = Initial-Contact; TC = Terminal-Contact.

Table 7.3: The mean error of the two algorithms under different speeds.

Speed(km/h)	Mean error(s)	
	Accelerator algorithm	Gyroscope algorithm
10	0.0159 ± 0.03	0.0239 ± 0.03
15	0.008 ± 0.03	0.0214 ± 0.03
20	0.025 ± 0.03	0.029 ± 0.03

Table 7.4: The consistency of the two systems under different speed and maximum tolerance value of the confidence interval.

Speed(km/h)	Strike type			
	Forefoot		Rear-foot	
	Appearance	Consistency at 0.03 maximum tolerance	Appearance	Consistency at 0.03 maximum tolerance
10	0%	N.A	100%	100%
15	8.75%	42.15%	91.25%	96.25%
20	86.25%	35.75%	13.75%	91.25%

The result of this study shows that both accelerometer and gyroscope have good potential in developing IMU based gait analysis algorithms, especially in contact time (CT) and flight time (FT) detection. By comparing the accuracy of the two algorithms at three different running speeds, we found that at medium running speed (around 15 km/h), both algorithms showed higher consistency and accuracy and the mean errors were at their lowest, 0.008s and 0.0214s. Another interesting trend is that both the accuracy and consistency are at their lowest during high speed running.

As both the consistency and accuracy of the two systems showed decreasing trend at higher speed, we studied the ground truth video and found that the performance of the algorithms can be related to the striking style, which are forefoot strike and rear-foot strike. Rear-foot strike means the heel of the runner makes the first contact with the ground where forefoot strike is when the toe of the runner contacts the ground first [177]. Table 7.4 summarizes the appearance of the two strike styles and the consistency of the two algorithms under different speeds and strike types. We also found that when the running speed increased, the runners tend to shift from rear-foot strike to forefoot strike as 86.25% of the strikes were shifted from rear to fore foot when the speed increased to 20km/h. The result also supported our hypothesis that the forefoot strike could affect the consistency and accuracy of the algorithms. The reason which causes this reduction could be due to the placement of the IMU. As the IMU was placed in the ankle area, which was closed to the rear-foot, and as the runner struck the ground with forefoot,

there was some movement and turbulence on the foot which could generate noise and affected the accuracy of the algorithms. The future studies should investigate the effect of different striking styles have on the current gait algorithms. At the same time, the investigation should also include the impact of different striking patterns on the accuracy of the algorithms in different IMU placement locations. Furthermore, the future design and development of the IMU-based gait algorithms should consider and build upon both striking styles to increase the accuracy.

7.4 Discussion

This study aimed to develop and validate adaptive algorithms based on gyroscope and accelerometer data for the detection of IC and TC during running. Such a system is desirable as it provides the potential of instant analysis and feedback of ground contact time for athletes and coaches in trainings or other circumstances in field settings. Specifically, such system based on IMU can overcome the limitations related to other gait analysis methods like motion tracking system, force plate, footswitch systems or camera-based systems which are highly lab-based and are difficult to use in field-based settings or in most team sports where training is undertaken concurrently in large groups. The application of gyroscope-based algorithms in IC and TC detection has been investigated in several running gait studies for healthy athletes [178], disabled athlete with transtibial amputees [179], athletics recovery [180] and many other conditions. On top of this, the gyroscope-based algorithm can maintain the same waveform shape under different or increasing running speeds as only the magnitude of the foot's angular rate and frequency are affected. Referring to the results from this study, the absolute mean error presented a less variation when running speed increased as compared to the acceleration-based algorithm. Therefore, the algorithm we proposed can reliably identify the key gait events of IC and TC in many running-related applications for both in-field settings and a large group of individuals. Despite these advantages, gyroscope-based algorithms usually require wavelet transformation and different filter design according

to different speeds, running intensiveness, and IMU placement. For example, Gouwanda et al. [178] designed a 2nd order digital low pass filter with cut-off frequency of 15 Hz to make the waveform peaks stands out for IC and TC detection; McGrath et al. [181] utilized a low pass filtered with zero-phase fifth order Butterworth filter with a 50.2 Hz corner frequency, where in this study, the cut-off frequency was set to 10 Hz to obtain the best waveform information. The performance of these algorithms can be affected with expanded sample size or running styles. Whereas for the accelerometer-based algorithm, the detection of TC is more crucial and usually generates more errors as compared to the detection of IC. In support of this, Jasiewicz et al. [182] studied the current accelerometer-based gait study algorithms with IMU placement on foot and suggested that the mean absolute error (MAE) of TC (27.6 ms) is much higher than the MAE of IC detection (4.2 ms). In this study, the result showed similar trend to the findings from this reviewed study by yielding higher errors in TC than IC, as the MAE of TC had a minimum of 23.8 ms and a maximum of 31.4 ms, whereases the IC had the minimum of 4.5 ms and the maximum of 6.7 ms. Moreover, the accuracy of the TC detection in acceleration-based analysis has not been widely investigated. Hence future should investigate the accuracy of TC detection and the factors which lead to the drop in accuracy. Thereby, the combination of using both accelerometer and gyroscope-based algorithms can be a good approach to mitigate the limitations of these two gait analysis methods. In this study, we investigated the consistency of the two algorithms with confidence interval and discovered that the proposed systems had good overall consistency despite the lower consistency (83.75 %) at the higher speed which could be resulted from the striking style and required further investigation. Hence, this study discovered the feasibility of using the combination of both algorithms where the advantage of each approach could be utilised to further improve the detection accuracy of IC and TC. An example to this could be using a weightage algorithm in which the accelerometer-based system has more weightage in IC calculation and gyroscope-based system being more impactful in TC detection. Accordingly, future studies should investigate the effect of different striking styles have on the current gait algorithms. Moreover, the accuracy of TC detection should

also be studied, and the cause of accuracy reduction should be investigated. Finally, the future study should examine the feasibility of the combination of accelerometer and gyroscope-based systems to further increase the IC and TC detection accuracy. To achieve these, the future experimental sample size should be expanded to include more runners with a larger variety of foot-striking characteristics, running speeds or intensities and with different IMU placement locations. Furthermore, the future design and development of the IMU-based gait algorithms should consider and build upon both striking styles to increase the accuracy.

7.5 Conclusions

In this chapter, we proposed an intelligent running gait analysis system that can estimate contact time (CT) and flight time (FT) by detecting key running instance such as IC and TC. The system had Two algorithms which were designed based on the data from gyroscope and accelerometer relatively. We also introduced a pre-processing algorithm which can detect the running period to increase the detection accuracy. Furthermore, the consistency of the two algorithms were studied using a 95% confidence interval and the accuracy of the system was investigated using the validation data set.

The result showed that the accelerometer and gyroscope combined system can obtain the desired accuracy (absolute error: <20ms) in CT and FT detection. Moreover, after introducing the confidence interval, the two systems showed high consistency in lower speed running (< 20km/h). It was found that the reduced consistency was due to the change in the foot striking style (rearfoot strike to forefoot strike) as the running speed increased.

Our ultimate objective is to deign a highly-accurate IMU-based gait analysis system which utilize both gyroscope and accelerometer. The system should be compatible to both striking styles (rear-foot and fore-foot) that were found to be affecting the systems' consistency and accuracy in the proposed algorithms. Therefore, the future research is to study the effect of the striking type on the current gait algorithms in different IMU

placement locations and to further improve the compatibility and detection accuracy of the system.

CHAPTER 8

CONCLUSION AND FUTURE WORK

8.1 Conclusion

This thesis has focused on improving the performance of IMU sensor-based electronic devices using autocalibration and model compression. To achieve this goal, we conducted research in two directions: i) The design of efficient autocalibration methods to provide accurate sensor readings; and ii) The design of novel model pruning and few-shot learning methods to ensure that neural network models can be executed on resource-limited devices.

For the first task, Chapter 3 described a novel gyroscope calibration method using simple equipment. A servomotor was employed to provide the calibration reference. The entire calibration process only takes approximately 1 min, and does not require high-precision equipment. Moreover, a six-observation experimental design was proposed to minimise the maximum variance of the estimated scale factors and biases. In addition, a fast-converging recursive linear least-squares estimation method was presented to reduce the computational complexity. The simulation results reflect the robustness of the calibration method under normal and extreme conditions. We experimentally demonstrated the feasibility of the proposed method on a robot arm, and implemented

the method on a microcontroller. We verified the calibration results of the proposed method through comparisons with a traditional turntable approach, and this experiment indicated that the results of these two methods are comparable.

With the aim of completely discarding the shackles of external calibration equipment, an improved method was proposed in Chapter 4. We developed an efficient in-field calibration method that readily calibrates triaxial gyroscopes without additional equipment. This experimental scheme can be easily implemented by manually rotating the triaxial gyroscope over a certain angle as the calibration reference. A linearised calibration model was developed for the proposed experimental scheme, and we showed that G-optimality was achievable. The absolute errors of the scale factors and biases were always less than 2.5×10^{-2} for LSM9DS1 and less than 1×10^{-2} for ICM20948, with the calibration process taking less than 30 s.

For the second task, Chapter 5 described a novel kernel-based nonparametric modelling method to estimate the heart rate response during treadmill exercise and proposed an MPC method for heart rate control. This kernel-based method introduced a kernel regularisation term that provides prior information to the model estimation phase. By adding this prior information, the experimental protocol can be significantly simplified and only a small number of model training experiments are needed. For the exerciser, this means fewer experiments during the model building phase. Based on the identified model, an MPC controller was designed to track a predefined reference heart rate profile. Another advantage of this approach is that the speed and acceleration of the treadmill can be limited to within a safe range for vulnerable exercisers.

To overcome the issue of limited resources, Chapter 6 described a pruning method that significantly reduces the computational load of GNNs. Larger graphs and deeper GNNs make the training and inference processes increasingly expensive. The GLT method, which leverages the LTH, was used to design a unified GNN sparsification framework that simultaneously prunes the graph adjacency matrix and the model weights. Apart from leveraging the LTH, we further extended several existing training-free saliency metrics (SNIP, GraSP, Synflow) for network pruning to GNNs. We found that these

methods can achieve similar performance as GLT when jointly pruning the network weights and graph adjacency matrix. This revealed that these training-free saliency metrics can replace LTH while achieving a significant decrease in computational cost.

Overall, this study has considered the design and implementation of e-trainers with a focus on IMU calibration algorithm design and practical implementation of pattern recognition algorithms. The entire design process was discussed, ranging from hardware design, some practical considerations regarding efficient wearable device calibration, and model compression using the pruning method.

8.2 Future Work

Several interesting directions for future work are as follows.

In Chapters 3 and 4, we proposed gyroscope calibration algorithms using a mathematical model. We assumed that the noise involved in these methods was additive white Gaussian noise. In the field experiments, however, we found that the noise tended to follow a chi-square distribution, and the accuracy of these methods was observed to be highly dependent on the rotation speed and calibration time. We believe that this is caused by the model and estimate used being incorrect. Therefore, future work will focus on i) analysing the noise source and ii) designing a proper estimator that can handle non-Gaussian noise.

In Chapter 5, we proposed heart rate regulating algorithms using kernel-based non-parametric modelling. The performance of this method was found to be highly dependent on the optimisation of the kernel hyperparameters. However, we used a grid search for hyperparameter tuning, which is time-consuming and computationally expensive. Therefore, a future improvement to the proposed methods will involve the automatic identification of the desired hyperparameters.

In Chapter 6, we proposed a pruning method that reduces the model complexity and allows the neural network model to be executed on an ultra-low-power micro-controller. In the experiments, we found that most floating-point operations were used for element-wise

operations, which can easily be calculated in parallel. Thus, future work will investigate whether we can design an application-specific integrated circuit to solve the problems of limited energy and resources.

In Chapter 7, we demonstrated an in-field application of the Fit.E, analysing the CT, FT, and asymmetric gait of 40 rugby players from Sydney Swans Football Club while running. This shown the usefulness of the proposed E.Fit and the potential further cooperation with other sports organizations.

BIBLIOGRAPHY

[1] C.-m. Dong, S.-q. Ren, X.-j. Chen, and Z.-h. Wang, "A separated calibration method for inertial measurement units mounted on three-axis turntables," *Sensors*, vol. 18, no. 9, p. 2846, 2018.

[2] Y. Wang, R. Cao, C. Li, and R. N. Dean, "Concepts, roadmaps and challenges of ovenized mems gyroscopes: a review," *IEEE Sensors Journal*, vol. 21, no. 1, pp. 92–119, 2020.

[3] N. El-Sheimy and A. Youssef, "Inertial sensors technologies for navigation applications: State of the art and future trends," *Satellite Navigation*, vol. 1, no. 1, pp. 1–21, 2020.

[4] M. Hwangbo, J.-S. Kim, and T. Kanade, "Imu self-calibration using factorization," *IEEE Transactions on Robotics*, vol. 29, no. 2, pp. 493–507, 2013.

[5] Y. Wu and L. Pei, "Gyroscope calibration via magnetometer," *IEEE Sensors Journal*, vol. 17, no. 16, pp. 5269–5275, 2017.

[6] A. Olivares, G. Olivares, J. Gorriz, and J. Ramirez, "High-efficiency low-cost accelerometer-aided gyroscope calibration," in *2009 International Conference on Test and Measurement*, vol. 1. IEEE, 2009, pp. 354–360.

[7] L. Wang and S. Su, "An efficient calibration method for triaxial gyroscope," *arXiv preprint arXiv:2103.11096*, 2021.

[8] L. Wang, Y. Yang, and S. Su, "Nonparametric modelling based model predictive control for human heart rate regulation during treadmill exercise," in *2021*

43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC). IEEE, 2021, pp. 1015–1018.

[9] N. G. Boulé, G. P. Kenny, E. Haddad, G. A. Wells, and R. J. Sigal, "Meta-analysis of the effect of structured exercise training on cardiorespiratory fitness in type 2 diabetes mellitus," *Diabetologia*, vol. 46, no. 8, pp. 1071–1081, 2003.

[10] S. Carbone, M. G. Del Buono, C. Ozemek, and C. J. Lavie, "Obesity, risk of diabetes and role of physical activity, exercise training and cardiorespiratory fitness," *Progress in cardiovascular diseases*, vol. 62, no. 4, pp. 327–333, 2019.

[11] A. Pandey, K. V. Patel, J. L. Bahnson, S. A. Gaussoin, C. K. Martin, A. Balasubramanyam, K. C. Johnson, D. K. McGuire, A. G. Bertoni, D. Kitzman *et al.*, "Association of intensive lifestyle intervention, fitness, and body mass index with risk of heart failure in overweight or obese adults with type 2 diabetes mellitus: an analysis from the look ahead trial," *Circulation*, vol. 141, no. 16, pp. 1295–1306, 2020.

[12] S. G. Lakoski, B. L. Willis, C. E. Barlow, D. Leonard, A. Gao, N. B. Radford, S. W. Farrell, P. S. Douglas, J. D. Berry, L. F. DeFina *et al.*, "Midlife cardiorespiratory fitness, incident cancer, and survival after cancer in men: the cooper center longitudinal study," *JAMA oncology*, vol. 1, no. 2, pp. 231–237, 2015.

[13] S. S. Sawada, T. Muto, H. Tanaka, I.-M. Lee, R. S. Paffenbarger Jr, M. Shindo, and S. N. Blair, "Cardiorespiratory fitness and cancer mortality in japanese men: a prospective study." *Medicine and science in sports and exercise*, vol. 35, no. 9, pp. 1546–1550, 2003.

[14] L. A. Kaminsky, R. Arena, Ø. Ellingsen, M. P. Harber, J. Myers, C. Ozemek, and R. Ross, "Cardiorespiratory fitness and cardiovascular disease-the past, present, and future," *Progress in cardiovascular diseases*, vol. 62, no. 2, pp. 86–93, 2019.

[15] E. Ekblom-Bak, M.-L. Hellénius, Ö. Ekblom, L.-M. Engström, and B. Ekblom, "Independent associations of physical activity and cardiovascular fitness with

cardiovascular risk in adults," *European Journal of Preventive Cardiology*, vol. 17, no. 2, pp. 175–180, 2010.

[16] J. M. Gill and D. Malkova, "Physical activity, fitness and cardiovascular disease risk in adults: interactions with insulin resistance and obesity," *Clinical science*, vol. 110, no. 4, pp. 409–425, 2006.

[17] F. Zaccardi, M. J. Davies, K. Khunti, and T. Yates, "Comparative relevance of physical fitness and adiposity on life expectancy: a uk biobank observational study," in *Mayo Clinic Proceedings*, vol. 94, no. 6. Elsevier, 2019, pp. 985–994.

[18] J. A. Laukkanen and S. K. Kunutsor, "Fitness equals longer life expectancy regardless of adiposity levels," in *Mayo Clinic Proceedings*, vol. 94, no. 6. Elsevier Limited, 2019, pp. 942–945.

[19] J. A. Barthold Jones, A. Lenart, and A. Baudisch, "Complexity of the relationship between life expectancy and overlap of lifespans," *PloS one*, vol. 13, no. 7, p. e0197985, 2018.

[20] C. H. Folkins and W. E. Sime, "Physical fitness training and mental health." *American psychologist*, vol. 36, no. 4, p. 373, 1981.

[21] T. Wassenaar, C. Wheatley, N. Beale, P. Salvan, A. Meaney, J. Possee, K. Atherton, J. Duda, H. Dawes, and H. Johansen-Berg, "Effects of a programme of vigorous physical activity during secondary school physical education on academic performance, fitness, cognition, mental health and the brain of adolescents (fit to study): study protocol for a cluster-randomised trial," *Trials*, vol. 20, no. 1, pp. 1–14, 2019.

[22] K. Khair, M. Holland, S. Dodgson, P. McLaughlin, S. Fletcher, and D. Christie, "Fitness enhances psychosocial well-being and self-confidence in young men with hemophilia: Results from project gym," *Research and Practice in Thrombosis and Haemostasis*, vol. 5, no. 8, p. e12622, 2021.

[23] J. H. Kerr and M. C. Vos, "Employee fitness programmes, absenteeism and general well-being," *Work & Stress*, vol. 7, no. 2, pp. 179–190, 1993.

[24] A. Af Geijerstam, K. Mehlig, M. Börjesson, J. Robertson, J. Nyberg, M. Adiels, A. Rosengren, M. Åberg, and L. Lissner, "Fitness, strength and severity of covid-19: a prospective register study of 1 559 187 swedish conscripts," *BMJ open*, vol. 11, no. 7, p. e051316, 2021.

[25] J. H. Hull, M. Wootten, M. Moghal, N. Heron, R. Martin, E. S. Walsted, A. Biswas, M. Loosemore, N. Elliott, and C. Ranson, "Clinical patterns, recovery time and prolonged impact of covid-19 illness in international athletes: the uk experience," *British Journal of Sports Medicine*, vol. 56, no. 1, pp. 4–11, 2022.

[26] N. A. M. Mokmin and N. Jamiat, "The effectiveness of a virtual fitness trainer app in motivating and engaging students for fitness activity by applying motor learning theory," *Education and Information Technologies*, vol. 26, no. 2, pp. 1847–1864, 2021.

[27] J. M. Eickhoff-Shemek and T. Topalian, "Virtual fitness programs: Safety and legal liability issues to consider: Part 1," *ACSM's Health & Fitness Journal*, vol. 26, no. 1, pp. 48–51, 2022.

[28] G. M. Lucas, N. Krämer, C. Peters, L.-S. Taesch, J. Mell, and J. Gratch, "Effects of perceived agency and message tone in responding to a virtual personal trainer," in *Proceedings of the 18th International Conference on Intelligent Virtual Agents*, 2018, pp. 247–254.

[29] V. Jadhav, G. Jain, O. Khandekar, and B. Smita, "Virtual fitness trainer with spontaneous feedback using a line of motion sensing input device kinect xbox 360," *International Research Journal of Engineering and Technology*, vol. 5, no. 10, pp. 1627–1630, 2018.

[30] M. Hooshmand, D. Zordan, T. Melodia, and M. Rossi, "Surf: Subject-adaptive unsupervised ecg signal compression for wearable fitness monitors," *IEEE Access*, vol. 5, pp. 19 517–19 535, 2017.

[31] H. Bjørnstad, L. Storstein, H. D. Meen, and O. Hals, "Electrocardiographs findings according to level of fitness and sport activity," *Cardiology*, vol. 83, no. 4, pp. 268–279, 1993.

[32] J. E. Muñoz, E. R. Gouveia, M. S. Cameirão, and S. B. i Badia, "Physiolab-a multivariate physiological computing toolbox for ecg, emg and eda signals: a case of study of cardiorespiratory fitness assessment in the elderly population," *Multimedia Tools and Applications*, vol. 77, no. 9, pp. 11 521–11 546, 2018.

[33] J. M. Martuscello, J. L. Nuzzo, C. D. Ashley, B. I. Campbell, J. J. Orriola, and J. M. Mayer, "Systematic review of core muscle activity during physical fitness exercises," *The Journal of Strength & Conditioning Research*, vol. 27, no. 6, pp. 1684–1698, 2013.

[34] C. Crema, A. Depari, A. Flammini, E. Sisinni, T. Haslwanter, and S. Salzmann, "Imu-based solution for automatic detection and classification of exercises in the fitness scenario," in *2017 IEEE Sensors Applications Symposium (SAS)*. IEEE, 2017, pp. 1–6.

[35] P. Hausberger, A. Fernbach, and W. Kastner, "Imu-based smart fitness devices for weight training," in *IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society*. IEEE, 2016, pp. 5182–5189.

[36] S. Noor, M. Hasham, Z. Habib, and A. B. Saragano, "Real-time rehabilitation and fitness system using depth sensor," in *2019 International Conference on Frontiers of Information Technology (FIT)*. IEEE, 2019, pp. 215–2155.

[37] C. Chen, K. Liu, R. Jafari, and N. Kehtarnavaz, "Home-based senior fitness test measurement system using collaborative inertial and depth sensors," in *2014*

36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE, 2014, pp. 4135–4138.

[38] W. Wang, B. Balmaekers, and G. De Haan, "Quality metric for camera-based pulse rate monitoring in fitness exercise," in *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2016, pp. 2430–2434.

[39] W. Wang, A. C. den Brinker, S. Stuijk, and G. de Haan, "Robust heart rate from fitness videos," *Physiological measurement*, vol. 38, no. 6, p. 1023, 2017.

[40] J. Zhang, R. Song, X. Zhao, R. Fang, B. Zhang, W. Qian, J. Zhang, C. Liu, and D. He, "Flexible graphene-assembled film-based antenna for wireless wearable sensor with miniaturized size and high sensitivity," *ACS omega*, vol. 5, no. 22, pp. 12 937–12 943, 2020.

[41] M. S. Mahmud, H. Fang, and H. Wang, "An integrated wearable sensor for unobtrusive continuous measurement of autonomic nervous system," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 1104–1113, 2018.

[42] M. Hooshmand, D. Zordan, D. Del Testa, E. Grisan, and M. Rossi, "Boosting the battery life of wearables for health monitoring through the compression of biosignals," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1647–1662, 2017.

[43] U. Maurer, A. Rowe, A. Smailagic, and D. P. Siewiorek, "ewatch: a wearable sensor and notification platform," in *International Workshop on Wearable and Implantable Body Sensor Networks (BSN'06)*. IEEE, 2006, pp. 4–pp.

[44] W. Zhang, R. Fan, Y. Wen, and F. Liu, "Energy optimal wireless data transmission for wearable devices: A compression approach," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 10, pp. 9605–9618, 2018.

[45] S. K. Lynn, C. M. Watkins, M. A. Wong, K. Balfany, and D. F. Feeney, "Validity and reliability of surface electromyography measurements from a wearable athlete

performance system," *Journal of sports science & medicine*, vol. 17, no. 2, p. 205, 2018.

[46] A. B. Chatfield, *Fundamentals of high accuracy inertial navigation*. American Institute of Aeronautics and Astronautics, 1997.

[47] C. Ren, Q. Liu, and T. Fu, "A novel self-calibration method for mimu," *IEEE Sensors Journal*, vol. 15, no. 10, pp. 5416–5422, 2015.

[48] T. Chen, Y. Sui, X. Chen, A. Zhang, and Z. Wang, "A unified lottery ticket hypothesis for graph neural networks," in *International Conference on Machine Learning*. PMLR, 2021, pp. 1695–1706.

[49] L. Lennart, "System identification: theory for the user," *PTR Prentice Hall, Upper Saddle River, NJ*, pp. 1–14, 1999.

[50] M. Bertero, *Linear inverse and III-posed problems*. Elsevier, 1989, vol. 75, pp. 1–120.

[51] A. Tarantola, *Inverse problem theory and methods for model parameter estimation*. SIAM, 2005.

[52] D. L. Phillips, "A technique for the numerical solution of certain integral equations of the first kind," *Journal of the ACM (JACM)*, vol. 9, no. 1, pp. 84–97, 1962.

[53] M. Y. Park and T. Hastie, "L1‚Äêregularization path algorithm for generalized linear models," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 69, no. 4, pp. 659–677, 2007.

[54] M. Schmidt, A. Niculescu-Mizil, and K. Murphy, "Learning graphical model structure using l1-regularization paths," in *AAAI*, vol. 7, 2007, Conference Proceedings, pp. 1278–1283.

[55] M. Hanke, "The minimal error conjugate gradient method is a regularization method," *Proceedings of the American mathematical Society*, vol. 123, no. 11, pp. 3487–3497, 1995.

[56] Y. Yao, L. Rosasco, and A. Caponnetto, "On early stopping in gradient descent learning," *Constructive Approximation*, vol. 26, no. 2, pp. 289–315, 2007.

[57] T. Soderstrom and P. Stoica, *System identification*. Prentice-Hall International, 1989.

[58] H. Akaike, "A new look at the statistical model identification," *IEEE transactions on automatic control*, vol. 19, no. 6, pp. 716–723, 1974.

[59] G. Pillonetto and G. De Nicolao, "A new kernel-based approach for linear system identification," *Automatica*, vol. 46, no. 1, pp. 81–93, 2010.

[60] G. Pillonetto, A. Chiuso, and G. De Nicolao, "Prediction error identification of linear systems: a nonparametric gaussian regression approach," *Automatica*, vol. 47, no. 2, pp. 291–305, 2011.

[61] T. Chen, H. Ohlsson, and L. Ljung, "On the estimation of transfer functions, regularizations and gaussian processes,Äîrevisited," *Automatica*, vol. 48, no. 8, pp. 1525–1535, 2012.

[62] J. G. Rasmussen, "Bayesian inference for hawkes processes," *Methodology and Computing in Applied Probability*, vol. 15, no. 3, pp. 623–642, 2013.

[63] B. Scholkopf, A. J. Smola, and F. Bach, *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

[64] A. N. Tikhonov and V. Y. Arsenin, "Solutions of ill-posed problems," *New York*, pp. 1–30, 1977.

[65] T. Liu, W. Zhang, L. Ye, M. Ueland, S. L. Forbes, and S. W. Su, "A novel multi-odour identification by electronic nose using non-parametric modelling-based feature extraction and time-series classification," *Sensors and Actuators B: Chemical*, vol. 298, p. 126690, 2019.

[66] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, no. 4, pp. 115–133, 1943.

[67] G. Zhanshe, C. Fucheng, L. Boyu, C. Le, L. Chao, and S. Ke, "Research development of silicon mems gyroscopes: A review," *Microsystem Technologies*, vol. 21, no. 10, pp. 2053–2066, 2015.

[68] D. Piyabongkarn, R. Rajamani, and M. Greminger, "The development of a mems gyroscope for absolute angle measurement," *IEEE transactions on control systems technology*, vol. 13, no. 2, pp. 185–195, 2005.

[69] Q. Shi, S. Wang, A. Qiu, Y. Xu, and X. Ji, "Design principle of suspension of mems gyroscope," in *2006 1st IEEE International Conference on Nano/Micro Engineered and Molecular Systems*. IEEE, 2006, pp. 242–245.

[70] R. Antonello, R. Oboe, L. Prandi, and F. Biganzoli, "Automatic mode matching in mems vibrating gyroscopes using extremum-seeking control," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 10, pp. 3880–3891, 2009.

[71] J. Bernstein, S. Cho, A. King, A. Kourepenis, P. Maciel, and M. Weinberg, "A micromachined comb-drive tuning fork rate gyroscope," in *[1993] Proceedings IEEE Micro Electro Mechanical Systems*. IEEE, 1993, pp. 143–148.

[72] S. A. Zotov, A. A. Trusov, and A. M. Shkel, "Three-dimensional spherical shell resonator gyroscope fabricated using wafer-scale glassblowing," *Journal of microelectromechanical systems*, vol. 21, no. 3, pp. 509–510, 2012.

[73] S. Antsiferov, D. Kondratov, and L. Mogilevich, "Perturbing moments in a floating gyroscope with elastic device housing on a vibrating base in the case of a nonsymmetric end outflow," *Mechanics of Solids*, vol. 44, no. 3, pp. 352–360, 2009.

[74] S.-h. P. Won and F. Golnaraghi, "A triaxial accelerometer calibration method using a mathematical model," *IEEE transactions on instrumentation and measurement*, vol. 59, no. 8, pp. 2144–2153, 2009.

[75] L. Ye, Y. Guo, and S. W. Su, "An efficient autocalibration method for triaxial accelerometer," *IEEE Transactions on Instrumentation and Measurement*, vol. 66, no. 9, pp. 2380–2390, 2017.

[76] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf, "Pruning filters for efficient convnets," *arXiv preprint arXiv:1608.08710*, 2016.

[77] H. Hu, R. Peng, Y.-W. Tai, and C.-K. Tang, "Network trimming: A data-driven neuron pruning approach towards efficient deep architectures," *arXiv preprint arXiv:1607.03250*, 2016.

[78] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," *arXiv preprint arXiv:1510.00149*, 2015.

[79] J.-H. Luo and J. Wu, "An entropy-based pruning method for cnn compression," *arXiv preprint arXiv:1706.05791*, 2017.

[80] J.-H. Luo, J. Wu, and W. Lin, "Thinet: A filter level pruning method for deep neural network compression," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5058–5066.

[81] Y. He, P. Liu, Z. Wang, Z. Hu, and Y. Yang, "Filter pruning via geometric median for deep convolutional neural networks acceleration," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4340–4349.

[82] J.-H. Luo and J. Wu, "Autopruner: An end-to-end trainable filter pruning method for efficient deep model inference," *Pattern Recognition*, vol. 107, p. 107461, 2020.

[83] ——, "Neural network pruning with residual-connections and limited-data," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1458–1467.

[84] M. Ma, Q. Song, Y. Gu, and Z. Zhou, "Use of magnetic field for mitigating gyroscope errors for indoor pedestrian positioning," *Sensors*, vol. 18, no. 8, p. 2592, 2018.

[85] M. Khedr and N. El-Sheimy, "A smartphone step counter using imu and magnetometer for navigation and health monitoring applications," *Sensors*, vol. 17, no. 11, p. 2573, 2017.

[86] K. Han, H. Han, Z. Wang, and F. Xu, "Extended kalman filter-based gyroscope-aided magnetometer calibration for consumer electronic devices," *IEEE Sensors Journal*, vol. 17, no. 1, pp. 63–71, 2016.

[87] M. B. Rhudy and J. M. Mahoney, "A comprehensive comparison of simple step counting techniques using wrist-and ankle-mounted accelerometer and gyroscope signals," *Journal of Medical Engineering & Technology*, vol. 42, no. 3, pp. 236–243, 2018.

[88] P. Zhang, X. Zhan, X. Zhang, and L. Zheng, "Error characteristics analysis and calibration testing for mems imu gyroscope," *Aerospace Systems*, vol. 2, no. 2, pp. 97–104, 2019.

[89] H. Yang, B. Zhou, L. Wang, H. Xing, and R. Zhang, "A novel tri-axial mems gyroscope calibration method over a full temperature range," *Sensors*, vol. 18, no. 9, p. 3004, 2018.

[90] U. Qureshi and F. Golnaraghi, "An algorithm for the in-field calibration of a mems imu," *IEEE Sensors Journal*, vol. 17, no. 22, pp. 7479–7486, 2017.

[91] J. A. Nelder and R. Mead, "A simplex method for function minimization," *The computer journal*, vol. 7, no. 4, pp. 308–313, 1965.

[92] G. A. Seber and C. J. Wild, "Nonlinear regression. hoboken," *New Jersey: John Wiley & Sons*, vol. 62, p. 63, 2003.

[93] I. Frosio, F. Pedersini, and N. A. Borghese, "Autocalibration of triaxial mems accelerometers with automatic sensor model selection," *IEEE Sensors Journal*, vol. 12, no. 6, pp. 2100–2108, 2012.

[94] J. Kiefer and J. Wolfowitz, "The equivalence of two extremum problems," *Canadian Journal of Mathematics*, vol. 12, pp. 363–366, 1960.

[95] J. C. Helton, "Uncertainty and sensitivity analysis techniques for use in performance assessment for radioactive waste disposal," *Reliability Engineering & System Safety*, vol. 42, no. 2-3, pp. 327–367, 1993.

[96] H. Chen, M. C. Schall, and N. B. Fethke, "Measuring upper arm elevation using an inertial measurement unit: An exploration of sensor fusion algorithms and gyroscope models," *Applied Ergonomics*, vol. 89, p. 103187, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0003687020301411

[97] N. Bai, Y. Tian, Y. Liu, Z. Yuan, Z. Xiao, and J. Zhou, "A high-precision and low-cost imu-based indoor pedestrian positioning technique," *IEEE Sensors Journal*, vol. 20, no. 12, pp. 6716–6726, 2020.

[98] Y. Cao and N. Sepúlveda, "Design of flexible piezoelectric gyroscope for structural health monitoring," *Applied Physics Letters*, vol. 115, no. 24, p. 241901, 2019.

[99] A. Petropoulos, D. Sikeridis, and T. Antonakopoulos, "Wearable smart health advisors: An imu-enabled posture monitor," *IEEE Consumer Electronics Magazine*, vol. 9, no. 5, pp. 20–27, 2020.

[100] S. Zihajehzadeh and E. J. Park, "A gaussian process regression model for walking speed estimation using a head-worn imu," in *2017 39th Annual International*

Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). IEEE, 2017, pp. 2345–2348.

[101] M. Sipos, P. Paces, J. Rohac, and P. Novacek, "Analyses of triaxial accelerometer calibration algorithms," *IEEE Sensors Journal*, vol. 12, no. 5, pp. 1157–1165, 2011.

[102] M. Sun, Y. Jiang, Q. Liu, and X. Liu, "An auto-calibration approach to robust and secure usage of accelerometers for human motion analysis in fes therapies," *Computers, Materials & Continua*, vol. 60, no. 1, pp. 67–83, 2019.

[103] X.-A. Han, Y. Ma, and X. Huang, "A novel generalization of bézier curve and surface," *Journal of Computational and Applied Mathematics*, vol. 217, no. 1, pp. 180–193, 2008.

[104] L. Wang, T. Zhang, L. Ye, J. J. Li, and S. Su, "An efficient calibration method for triaxial gyroscope," *IEEE Sensors Journal*, pp. 1–1, 2021.

[105] Y. J. van de Vegte, B. S. Tegegne, N. Verweij, H. Snieder, and P. van der Harst, "Genetics and the heart rate response to exercise," *Cellular and Molecular Life Sciences*, vol. 76, no. 12, pp. 2391–2409, 2019.

[106] J. C. Jentzer, S. van Diepen, G. W. Barsness, T. D. Henry, V. Menon, C. S. Rihal, S. S. Naidu, and D. A. Baran, "Cardiogenic shock classification to predict mortality in the cardiac intensive care unit," *Journal of the American College of Cardiology*, vol. 74, no. 17, pp. 2117–2128, 2019.

[107] K. M. Godfrey, A. Juarascio, S. Manasse, A. Minassian, V. Risbrough, and N. Afari, "Heart rate variability and emotion regulation among individuals with obesity and loss of control eating," *Physiology & behavior*, vol. 199, pp. 73–78, 2019.

[108] S. W. Su, L. Wang, B. G. Celler, A. V. Savkin, and Y. Guo, "Identification and control for heart rate regulation during treadmill exercise," *IEEE Transactions on biomedical engineering*, vol. 54, no. 7, pp. 1238–1246, 2007.

[109] K. J. Hunt, A. Zahnd, and R. Grunder, "A unified heart rate control approach for cycle ergometer and treadmill exercise," *Biomedical Signal Processing and Control*, vol. 54, p. 101601, 2019.

[110] S. W. Su, S. Huang, L. Wang, B. G. Celler, A. V. Savkin, Y. Guo, and T. M. Cheng, "Optimizing heart rate regulation for safe exercise," *Annals of biomedical engineering*, vol. 38, no. 3, pp. 758–768, 2010.

[111] J. Kranjec, S. Begus, G. Gersak, and J. Drnovsek, "Non-contact heart rate and heart rate variability measurements: A review," *Biomedical signal processing and control*, vol. 13, pp. 102–112, 2014.

[112] C. E. Rasmussen, "Gaussian processes in machine learning," in *Summer School on Machine Learning*. Springer, 2003, pp. 63–71.

[113] T. Chen and G. Pillonetto, "On the stability of reproducing kernel hilbert spaces of discrete-time impulse responses," *Automatica*, vol. 95, pp. 529–533, 2018.

[114] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[115] M. Zhang and Y. Chen, "Link prediction based on graph neural networks," *Advances in Neural Information Processing Systems*, vol. 31, pp. 5165–5175, 2018.

[116] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph convolutional neural networks for web-scale recommender systems," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 974–983.

[117] C. Hübler, H.-P. Kriegel, K. Borgwardt, and Z. Ghahramani, "Metropolis algorithms for representative subgraph sampling," in *2008 Eighth IEEE International Conference on Data Mining*. IEEE, 2008, pp. 283–292.

[118] D. Calandriello, A. Lazaric, I. Koutis, and M. Valko, "Improved large-scale graph learning through ridge spectral sparsification," in *International Conference on Machine Learning*. PMLR, 2018, pp. 688–697.

[119] B. Adhikari, Y. Zhang, S. E. Amiri, A. Bharadwaj, and B. A. Prakash, "Propagation-based temporal network summarization," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 4, pp. 729–742, 2017.

[120] T. Eden, S. Jain, A. Pinar, D. Ron, and C. Seshadhri, "Provable and practical approximations for the degree distribution using sublinear graph samples," in *Proceedings of the 2018 World Wide Web Conference*, 2018, pp. 449–458.

[121] J. Chen, J. Zhu, and L. Song, "Stochastic training of graph convolutional networks with variance reduction," *arXiv preprint arXiv:1710.10568*, 2017.

[122] F. Monti, D. Boscaini, J. Masci, E. Rodola, J. Svoboda, and M. M. Bronstein, "Geometric deep learning on graphs and manifolds using mixture model cnns," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5115–5124.

[123] J. Li, T. Zhang, H. Tian, S. Jin, M. Fardad, and R. Zafarani, "Sgcn: A graph sparsifier based on graph convolutional networks," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2020, pp. 275–287.

[124] S. Han, J. Pool, J. Tran, and W. J. Dally, "Learning both weights and connections for efficient neural networks," *arXiv preprint arXiv:1506.02626*, 2015.

[125] J. Frankle and M. Carbin, "The lottery ticket hypothesis: Finding sparse, trainable neural networks," *arXiv preprint arXiv:1803.03635*, 2018.

[126] N. Lee, T. Ajanthan, and P. H. Torr, "Snip: Single-shot network pruning based on connection sensitivity," *arXiv preprint arXiv:1810.02340*, 2018.

[127] C. Wang, G. Zhang, and R. Grosse, "Picking winning tickets before training by preserving gradient flow," *arXiv preprint arXiv:2002.07376*, 2020.

[128] H. Tanaka, D. Kunin, D. L. Yamins, and S. Ganguli, "Pruning neural networks without any data by iteratively conserving synaptic flow," *arXiv preprint arXiv:2006.05467*, 2020.

[129] L. Zhao and L. Akoglu, "Pairnorm: Tackling oversmoothing in gnns," *arXiv preprint arXiv:1909.12223*, 2019.

[130] Q. Li, Z. Han, and X.-M. Wu, "Deeper insights into graph convolutional networks for semi-supervised learning," in *Thirty-Second AAAI conference on artificial intelligence*, 2018.

[131] Y. Yan, M. Hashemi, K. Swersky, Y. Yang, and D. Koutra, "Two sides of the same coin: Heterophily and oversmoothing in graph convolutional neural networks," *arXiv preprint arXiv:2102.06462*, 2021.

[132] C. Cai and Y. Wang, "A note on over-smoothing for graph neural networks," *arXiv preprint arXiv:2006.13318*, 2020.

[133] W. Huang, Y. Li, W. Du, R. Y. Da Xu, J. Yin, L. Chen, and M. Zhang, "Towards deepening graph neural networks: A gntk-based optimization perspective," *arXiv preprint arXiv:2103.03113*, 2021.

[134] G. Li, M. Muller, A. Thabet, and B. Ghanem, "Deepgcns: Can gcns go as deep as cnns?" in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 9267–9276.

[135] Y. Rong, W. Huang, T. Xu, and J. Huang, "Dropedge: Towards deep graph convolutional networks on node classification," *arXiv preprint arXiv:1907.10903*, 2019.

[136] F. R. Chung and F. C. Graham, *Spectral graph theory*. American Mathematical Soc., 1997, no. 92.

[137] F. McSherry, "Spectral partitioning of random graphs," in *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*. IEEE, 2001, pp. 529–537.

[138] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," *Advances in neural information processing systems*, vol. 29, pp. 3844–3852, 2016.

[139] R. Levie, F. Monti, X. Bresson, and M. M. Bronstein, "Cayleynets: Graph convolutional neural networks with complex rational spectral filters," *IEEE Transactions on Signal Processing*, vol. 67, no. 1, pp. 97–109, 2018.

[140] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[141] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" *arXiv preprint arXiv:1810.00826*, 2018.

[142] M. Balcilar, G. Renton, P. Héroux, B. Gauzere, S. Adam, and P. Honeine, "Bridging the gap between spectral and spatial domains in graph neural networks," *arXiv preprint arXiv:2003.11702*, 2020.

[143] B. Adhikari, Y. Zhang, S. E. Amiri, A. Bharadwaj, and B. A. Prakash, "Propagation-based temporal network summarization," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 4, pp. 729–742, 2017.

[144] A. Chakeri, H. Farhidzadeh, and L. O. Hall, "Spectral sparsification in spectral clustering," in *2016 23rd international conference on pattern recognition (icpr)*. IEEE, 2016, pp. 2301–2306.

[145] J. Chen, T. Ma, and C. Xiao, "Fastgcn: fast learning with graph convolutional networks via importance sampling," *arXiv preprint arXiv:1801.10247*, 2018.

[146] X. Gao, W. Hu, and Z. Guo, "Exploring structure-adaptive graph learning for robust semi-supervised classification," in *2020 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2020, pp. 1–6.

[147] W. Jin, Y. Ma, X. Liu, X. Tang, S. Wang, and J. Tang, "Graph structure learning for robust graph neural networks," in *Proceedings of the 26th ACM SIGKDD*

International Conference on Knowledge Discovery & Data Mining, 2020, pp. 66–74.

[148] W.-L. Chiang, X. Liu, S. Si, Y. Li, S. Bengio, and C.-J. Hsieh, "Cluster-gcn: An efficient algorithm for training deep and large graph convolutional networks," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 257–266.

[149] C. Zheng, B. Zong, W. Cheng, D. Song, J. Ni, W. Yu, H. Chen, and W. Wang, "Robust graph representation learning via neural sparsification," in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, H. D. III and A. Singh, Eds., vol. 119. PMLR, 13–18 Jul 2020, pp. 11 458–11 468. [Online]. Available: https://proceedings.mlr.press/v119/zheng20d.html

[150] Y. LeCun, J. S. Denker, and S. A. Solla, "Optimal brain damage," in *Advances in neural information processing systems*, 1990, pp. 598–605.

[151] B. Hassibi and D. G. Stork, *Second order derivatives for network pruning: Optimal brain surgeon*. Morgan Kaufmann, 1993.

[152] P. Molchanov, S. Tyree, T. Karras, T. Aila, and J. Kautz, "Pruning convolutional neural networks for resource efficient inference," *arXiv preprint arXiv:1611.06440*, 2016.

[153] S. Verdenius, M. Stol, and P. Forré, "Pruning via iterative ranking of sensitivity statistics," *arXiv preprint arXiv:2006.00896*, 2020.

[154] D. C. Mocanu, E. Mocanu, P. Stone, P. H. Nguyen, M. Gibescu, and A. Liotta, "Scalable training of artificial neural networks with adaptive sparse connectivity inspired by network science," *Nature communications*, vol. 9, no. 1, pp. 1–12, 2018.

[155] T. Dettmers and L. Zettlemoyer, "Sparse networks from scratch: Faster training without losing performance," *arXiv preprint arXiv:1907.04840*, 2019.

[156] P. de Jorge, A. Sanyal, H. S. Behl, P. H. Torr, G. Rogez, and P. K. Dokania, "Progressive skeletonization: Trimming more fat from a network at initialization," *arXiv preprint arXiv:2006.09081*, 2020.

[157] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE signal processing magazine*, vol. 30, no. 3, pp. 83–98, 2013.

[158] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," *arXiv preprint arXiv:1312.6203*, 2013.

[159] Z. Chen, F. Chen, L. Zhang, T. Ji, K. Fu, L. Zhao, F. Chen, L. Wu, C. Aggarwal, and C.-T. Lu, "Bridging the gap between spatial and spectral domains: A unified framework for graph neural networks," *arXiv preprint arXiv:2107.10234*, 2021.

[160] J. Frankle, G. K. Dziugaite, D. M. Roy, and M. Carbin, "Pruning neural networks at initialization: Why are we missing the mark?" *arXiv preprint arXiv:2009.08576*, 2020.

[161] J. Su, Y. Chen, T. Cai, T. Wu, R. Gao, L. Wang, and J. D. Lee, "Sanity-checking pruning methods: Random tickets can win the jackpot," *Advances in Neural Information Processing Systems*, vol. 33, pp. 20 390–20 401, 2020.

[162] M. Falbriard, F. Meyer, B. Mariani, G. P. Millet, and K. Aminian, "Accurate estimation of running temporal parameters using foot-worn inertial sensors," *Frontiers in physiology*, vol. 9, p. 610, 2018.

[163] B. Mariani, H. Rouhani, X. Crevoisier, and K. Aminian, "Quantitative estimation of foot-flat and stance phase of gait using foot-worn inertial sensors," *Gait & posture*, vol. 37, no. 2, pp. 229–234, 2013.

[164] J. T. Fuller, C. R. Bellenger, D. Thewlis, M. D. Tsiros, and J. D. Buckley, "The effect of footwear on running performance and running economy in distance runners," *Sports Medicine*, vol. 45, no. 3, pp. 411–422, 2015.

[165] A. Hulme, R. O. Nielsen, T. Timpka, E. Verhagen, and C. Finch, "Risk and protective factors for middle-and long-distance running-related injury," *Sports Medicine*, vol. 47, no. 5, pp. 869–886, 2017.

[166] C. Yang, U. Ugbolue, B. Carse, V. Stankovic, L. Stankovic, and P. Rowe, "Multiple marker tracking in a single-camera system for gait analysis," in *2013 IEEE International Conference on Image Processing*. IEEE, 2013, pp. 3128–3131.

[167] J. Wang, M. She, S. Nahavandi, and A. Kouzani, "A review of vision-based gait recognition methods for human identification," in *2010 international conference on digital image computing: techniques and applications*. IEEE, 2010, pp. 320–327.

[168] M. Norris, R. Anderson, and I. C. Kenny, "Method analysis of accelerometers and gyroscopes in running gait: A systematic review," *Proceedings of the Institution of Mechanical Engineers, Part P: Journal of Sports Engineering and Technology*, vol. 228, no. 1, pp. 3–15, 2014.

[169] C. J. de Ruiter and J. H. van Dieën, "Stride and step length obtained with inertial measurement units during maximal sprint acceleration," *Sports*, vol. 7, no. 9, p. 202, 2019.

[170] J. Lobo and J. Dias, "Relative pose calibration between visual and inertial sensors," *The International Journal of Robotics Research*, vol. 26, no. 6, pp. 561–575, 2007.

[171] S. Mo and D. H. Chow, "Accuracy of three methods in gait event detection during overground running," *Gait & posture*, vol. 59, pp. 93–98, 2018.

[172] R. A. Clark, S. Bartold, and A. L. Bryant, "Tibial acceleration variability during consecutive gait cycles is influenced by the menstrual cycle," *Clinical Biomechanics*, vol. 25, no. 6, pp. 557–562, 2010.

[173] A. Muro-De-La-Herran, B. Garcia-Zapirain, and A. Mendez-Zorrilla, "Gait analysis methods: An overview of wearable and non-wearable systems, highlighting clinical applications," *Sensors*, vol. 14, no. 2, pp. 3362–3394, 2014.

[174] E. Bergamini, P. Picerno, H. Pillet, F. Natta, P. Thoreux, and V. Camomilla, "Estimation of temporal parameters during sprint running using a trunk-mounted inertial measurement unit," *Journal of biomechanics*, vol. 45, no. 6, pp. 1123–1126, 2012.

[175] A. G. Schubert, J. Kempf, and B. C. Heiderscheit, "Influence of stride frequency and length on running mechanics: a systematic review," *Sports health*, vol. 6, no. 3, pp. 210–217, 2014.

[176] J. K. Lee and E. J. Park, "Quasi real-time gait event detection using shank-attached gyroscopes," *Medical & biological engineering & computing*, vol. 49, no. 6, pp. 707–712, 2011.

[177] A. I. Daoud, G. J. Geissler, F. Wang, J. Saretsky, Y. A. Daoud, and D. E. Lieberman, "Foot strike and injury rates in endurance runners: a retrospective study," *Med Sci Sports Exerc*, vol. 44, no. 7, pp. 1325–1334, 2012.

[178] D. Gouwanda, A. A. Gopalai, and B. H. Khoo, "A low cost alternative to monitor human gait temporal parameters–wearable wireless gyroscope," *IEEE Sensors Journal*, vol. 16, no. 24, pp. 9029–9035, 2016.

[179] R. W. Selles, M. A. Formanoy, J. B. Bussmann, P. J. Janssens, and H. J. Stam, "Automated estimation of initial and terminal contact timing using accelerometers; development and validation in transtibial amputees and controls," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 13, no. 1, pp. 81–88, 2005.

[180] H. G. Espinosa, J. Lee, and D. A. James, "The inertial sensor: A base platform for wider adoption in sports science applications," *Journal of Fitness Research*, vol. 4, no. 1, pp. 13–20, 2015.

[181] D. McGrath, B. R. Greene, K. J. O,ÄôDonovan, and B. Caulfield, "Gyroscope-based assessment of temporal gait parameters during treadmill walking and running," *Sports Engineering*, vol. 15, no. 4, pp. 207–213, 2012.

[182] J. M. Jasiewicz, J. H. Allum, J. W. Middleton, A. Barriskill, P. Condie, B. Purcell, and R. C. T. Li, "Gait event detection using linear accelerometers or angular velocity transducers in able-bodied and spinal-cord injured individuals," *Gait & posture*, vol. 24, no. 4, pp. 502–509, 2006.