



FAU Studien aus der Informatik 11

**Peter Blank**

Smart Racket –  
Instrumented Racket as Real-time  
Feedback Device for Table Tennis



Peter Blank

Smart Racket – Instrumented Racket as  
Real-time Feedback Device in Table Tennis

# FAU Studien aus der Informatik

Band 11

Herausgeber der Reihe:

Björn Eskofier, Richard Lenz, Andreas Maier,  
Michael Philippsen, Lutz Schröder,  
Wolfgang Schröder-Preikschat, Marc Stamminger, Rolf Wanka

Peter Blank

Smart Racket –  
Instrumented Racket as  
Real-time Feedback Device  
in Table Tennis

Erlangen  
FAU University Press  
2020

Bibliografische Information der Deutschen Nationalbibliothek:  
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der  
Deutschen Nationalbibliografie; detaillierte bibliografische Daten  
sind im Internet über <http://dnb.d-nb.de> abrufbar.

Das Werk, einschließlich seiner Teile, ist urheberrechtlich geschützt.  
Die Rechte an allen Inhalten liegen bei ihren jeweiligen Autoren.  
Sie sind nutzbar unter der Creative Commons Lizenz BY.

Der vollständige Inhalt des Buchs ist als PDF über den OPUS Server  
der Friedrich-Alexander-Universität Erlangen-Nürnberg abrufbar:  
<https://opus4.kobv.de/opus4-fau/home>

Bitte zitieren als

Blank, Peter. 2020. *Smart Racket – Instrumented Racket as Real-time  
Feedback Device in Table Tennis*. FAU Studien aus der Informatik  
Band 11. Erlangen: FAU University Press.  
DOI: 10.25593/978-3-96147-312-0

Verlag und Auslieferung:

FAU University Press, Universitätsstraße 4, 91054 Erlangen

Druck: docupoint GmbH

ISBN: 978-3-96147-311-3 (Druckausgabe)  
eISBN: 978-3-96147-312-0 (Online-Ausgabe)  
ISSN: 2509-9981  
DOI: 10.25593/978-3-96147-312-0

# Smart Racket – Instrumented Racket as Real-time Feedback Device for Table Tennis

Smart Racket – Intelligenter Schläger als  
Echtzeitfeedbacksystem im Tischtennis

Der Technischen Fakultät der  
Friedrich-Alexander-Universität  
Erlangen-Nürnberg

zur Erlangung des Grades

DOKTOR-INGENIEUR

vorgelegt von

Dipl.-Ing. Peter Blank

aus Bayreuth

Als Dissertation genehmigt  
von der Technischen Fakultät  
der Friedrich-Alexander-Universität Erlangen-Nürnberg

|                                    |                              |
|------------------------------------|------------------------------|
| Tag der mündlichen Prüfung:        | 24. Oktober 2019             |
| Vorsitzender des Promotionsorgans: | Prof. Dr.-Ing. Andreas Fröba |
| Gutachter:                         | Prof. Dr. Björn Eskofier     |
|                                    | Prof. Dr. Paul Lukowicz      |

# Kurzzusammenfassung

Moderne, am Körper getragene Sensorsysteme zur Leistungsdiagnostik von Amateur- von Profisportlern werden immer häufiger im Sportumfeld eingesetzt. Erhebliche Fortschritte in der Miniaturisierung elektronischer Bauteile und der Weiterentwicklung von Datenanalysetechniken eröffnen neue Möglichkeiten, kleinste Sensorik unauffällig in jede Art von Kleidung und Sportausrüstung zu integrieren. Solche Wearables sind sowohl als wichtige Datenquelle für moderne Bewegungsanalyse und Leistungsindikatoren, als auch für die Unterstützung und Beurteilung von Athleten während des Trainings und Wettkampfs unverzichtbar geworden.

Diese Arbeit befasst sich mit der Anwendung von Wearables im Tischtennis. Diese Sportart zeichnet sich zum einen durch eine große Bandbreite an unterschiedlichsten Spielweisen, Taktiken und technischen Möglichkeiten aus, was wissenschaftliche Forschung interessant macht. Zum anderen schreitet die Digitalisierung hier noch langsam voran. Um die Vorteile von Wearables auch im Tischtennis zugänglich zu machen und den Sport attraktiv zu gestalten, wird in dieser Arbeit ein Trainings- und Sportlerassistenzsystem vorgestellt.

Im Detail befasst sich die Tischtennisanalyse mit der Erkennung und Klassifizierung unterschiedlicher Schlagtypen, der Schätzung von Spin und Geschwindigkeit des Balls kurz nach dem Aufprall, sowie der Lokalisierung des Balltreffpunkts auf dem Schläger. Zusätzlich wird eine speziell entwickelte Elektronik vorgestellt, die unauffällig und nicht störend im Schlägergriff integriert werden kann. In Kombination mit einer Smartphone App können alle gesammelten Daten als Statistiken visualisiert und dem Spieler als Übersicht präsentiert werden.

Für Schlagerkennung und -klassifizierung wurden Daten eines am Schläger befestigten Inertialsensors aufgezeichnet. Spieler mit unterschiedlichen Spieleigenschaften und Leistungsniveaus führten sowohl vordefinierte Übungen als auch freie Spielzüge als normales Spiel durch. Unterschiedliche Machine Learning Algorithmen und Verfahren aus der Mustererkennung wurden zur Datenauswertung implementiert. Damit konnten sehr gute Ergebnisse und Klassifikationsraten erzielt werden.

Die Abschätzung von Ballgeschwindigkeit und Ballspin basierte ebenfalls auf Messungen eines am Schläger montierten Inertialsensors. Die gewonnenen kinematischen Daten wurden verwendet, um die Rotation und Geschwindigkeit des Schlägers von der Ausholbewegung bis zum

Treffen des Balls zu beschreiben. Diese Parameter wurden mit Hilfe von Annahmen, Vereinfachungen und einem geeigneten Reboundmodell in Bewegungseigenschaften des Balls transformiert. Der vorgestellte Ansatz lieferte wie auch gängige videobasierte Methoden eine aussagekräftige Bestimmung von Ballgeschwindigkeit und Ballspin kurz nach dem Aufprall.

Die Lokalisierung der Ballauftreffposition wurde durch Vibrationsdaten dreier piezoelektrischer Sensoren, die an der Außenkante des Schlägers angebracht waren, realisiert. Durch Differenzmessungen der Ankunftszeit von Vibrationswellen an unterschiedlichen Sensorpositionen wurde ein Modell dieser nichtlinearen Vibrationsverteilung über die gesamte Schlägeroberfläche erstellt. Unter Verwendung dieses Modells wurden verschiedene Algorithmen evaluiert, um auf den ursprünglichen Balltreffpunkt zu schließen. Im Vergleich zur Literatur werden ähnliche Ergebnisse erzielt, wobei die hier entwickelten Methoden weniger Sensorkomponenten und Rechenleistung benötigten.

Weiterhin wurde spezielle Elektronik für eine unauffällige Instrumentierung des Schlägers entwickelt. Um dabei offizielle Regeln einzuhalten, wurde eine Platine im Schlägergriff integriert. Diese beinhaltet unter anderem einen leistungsfähigen Mikrocontroller, der alle notwendigen Berechnungsschritte der implementierten Algorithmen durchführte. Die berechneten Ergebnisse wurden drahtlos an die Feedbackanwendung zur Auswertung übertragen.

Die in dieser Arbeit vorgestellten Methoden ermöglichen erstmals die Anwendung von Wearables im Tischtennisbereich. Das entwickelte System hat das Potential, sowohl neue Trainingsmöglichkeiten zu schaffen, als auch durch objektive und quantitative Bestimmung von Leistungsindikatoren die Bewertung von Sportlern zu erleichtern.

# Abstract

In modern sports, often digitalized sensor systems are applied to monitor amateur and professional athletes. Since latest advances in miniaturization of electronic components and data analysis techniques opened new possibilities in sensor design, tiny measurement devices can easily be integrated into any kind of sports equipment or body-worn accessories. These unobtrusive Wearables have become essential as important data source for state-of-the-art movement analysis and performance indication to support and assess athletes during training and competition.

This thesis presents methods and approaches to enable the analysis of wearable data in the field of table tennis. On the one hand, this sport is characterized by a wide range of various playing styles and technical opportunities, making it a rewarding area of research. On the other hand, digitalization in table tennis is still slowly progressing. In order to benefit from the advantages of Wearables in table tennis as well as keeping this sport attractive, this thesis introduces a training and support system comprising of an instrumented racket, multiple analysis algorithms and a feedback application running on a mobile device.

More precisely, the introduced table tennis analysis addressed stroke detection and stroke type classification, ball speed and spin estimation shortly after impact as well as ball impact localization on the racket's surface. In addition, suitable electronics were developed, which were unobtrusively integrated into a racket prototype. In combination with a mobile application, the obtained data were presented as visualized statistics for feedback purposes.

In stroke detection and classification, data were acquired using racket-mounted inertial sensor. Players with different playing abilities and skill levels performed both predefined exercises and free gameplays. Multiple machine learning algorithms and pattern recognition methods were evaluated resulting in overall good accuracies for stroke detection and classification rates for stroke type recognition.

The ball speed and spin estimation was based on measurements of the same racket-mounted inertial sensor. Kinematic data were used to calculate the racket speed and rotation ranging from the countermovement to the ball impact event. Applying several assumptions and simplifications, the racket motion parameters were transformed into ball characteristics by means of a suitable rebound model. Compared to video-

based methods, reasonable estimations for ball speed and spin shortly after impact were possible.

Vibration data of multiple piezo-electric sensors mounted at the outer edge of the racket were acquired to localize the initial impact position on the blade. Time difference values of the approaching wavefront from all sensor pairs were investigated to generate a valuable model of their non-linear distribution over the entire surface. Using this defined model, different approaches were evaluated to regress the initial ball impact position. Even with significantly less sensor components and computational efforts compared to literature, similar results were achieved.

Additionally, suitable electronics were developed for an unobtrusive racket instrumentation. Complying to official regulations, a small board was integrated into the racket's handle. Therewith, a powerful micro-controller was assembled, which was able to execute any necessary processing steps of the mentioned algorithms. Subsequently, computed results were wirelessly transmitted to a mobile application for statistics and feedback purposes.

To sum up, the contributions of this thesis enabled the application of Wearables in table tennis as a support system for professional and amateur athletes. The developed smart racket has the potential to facilitate monitoring and assessment of players, thus providing a considerable basis of objective and quantitative determination of performance indicators.

# Acknowledgment

Successfully completing my scientific career took longer than I expected and would not have been possible without active support. For this reason, I would like to particularly thank following persons, supporters and institutions.

Firstly, I want to thank my supervisor Prof. Dr. Björn Eskofier, who gave me the opportunity to incorporate one of my hobbies – table tennis – into the scholarly focus of my doctoral dissertation. I appreciate his open-mindedness to transfer own interests into scientific directions.

Secondly, I would like to give thanks to the Embedded Systems Initiative (ESI) for financial support of my research.

Thirdly, I want to thank the Machine Learning and Data Analytics Lab, which supplied me with tons of hardware equipment and gave me the possibility to participate in different conferences worldwide.

Moreover, I would like to thank my students Julian Hoßbach, Martin Kulesa, Steffen Hofmann, Mathias Schöll and Julia Wolf, who contributed to my thesis and supported me during the data acquisitions.

I also want to thank my colleagues, who always came up with good ideas, when I got stuck in for me almost unsolvable questions. Thereby, I want to especially thank my longstanding office colleagues and friends Thomas Kautz and Ben Groh. With you, it never got boring and I always had a great time.

Furthermore, I would like to thank my table tennis club SpVgg 1904 Erlangen e.V., which kindly provided the locations for my conducted studies. Special thanks to all voluntary participants, who performed their best table tennis for me collecting huge amounts of data.

Needless to say, I want to thank my family for unlimited support during my whole time at the university. Many thanks to my parents, who were unfailingly with me.

Last but not least, I would like to thank my wife Lara, who constantly motivated me especially during the arduous last times of writing these words.

Peter Blank



# Contents

|                                                         |           |
|---------------------------------------------------------|-----------|
| <b>List of Abbreviations</b>                            | <b>1</b>  |
| <b>List of Symbols</b>                                  | <b>5</b>  |
| <b>1 Introduction</b>                                   | <b>11</b> |
| 1.1 Overview and Motivation . . . . .                   | 11        |
| 1.1.1 Wearables in Sports . . . . .                     | 11        |
| 1.1.2 Table Tennis . . . . .                            | 13        |
| 1.2 Related Work . . . . .                              | 17        |
| 1.2.1 Stroke and Motion Analysis in Racket Sports . . . | 17        |
| 1.2.2 Ball Speed and Spin Estimation . . . . .          | 26        |
| 1.2.3 Investigations of Impacts on Rackets . . . . .    | 33        |
| 1.3 Commercial Products . . . . .                       | 36        |
| 1.4 Contributions . . . . .                             | 37        |
| 1.5 Structure of this Thesis . . . . .                  | 40        |
| <b>2 Fundamentals of Machine Learning</b>               | <b>43</b> |
| 2.1 Introduction . . . . .                              | 43        |
| 2.2 Classification Pipeline . . . . .                   | 45        |
| 2.2.1 Data Acquisition . . . . .                        | 45        |
| 2.2.2 Preprocessing . . . . .                           | 46        |
| 2.2.3 Feature Extraction . . . . .                      | 47        |
| 2.2.4 Feature Selection . . . . .                       | 49        |
| 2.2.5 Classification . . . . .                          | 49        |
| 2.3 Evaluation . . . . .                                | 54        |
| <b>3 Embedded Signal Processing</b>                     | <b>57</b> |
| 3.1 Considerations, Restrictions and Benefits . . . . . | 57        |
| 3.1.1 Memory Consumption . . . . .                      | 58        |
| 3.1.2 Execution Time . . . . .                          | 61        |

|          |                                              |            |
|----------|----------------------------------------------|------------|
| 3.1.3    | Arithmetic Precision . . . . .               | 66         |
| 3.2      | Real-Time Operating System . . . . .         | 67         |
| 3.2.1    | Machine Learning Matching . . . . .          | 68         |
| 3.2.2    | Task Scheduling . . . . .                    | 70         |
| <b>4</b> | <b>Wearable Sensor Development</b>           | <b>77</b>  |
| 4.1      | Requirements . . . . .                       | 77         |
| 4.2      | Sensor Types . . . . .                       | 79         |
| 4.2.1    | Inertial Measurement Units . . . . .         | 79         |
| 4.2.2    | Geo-magnetic Sensor . . . . .                | 83         |
| 4.2.3    | Piezo-electric Vibration Sensor . . . . .    | 86         |
| 4.3      | Sensor Calibration . . . . .                 | 87         |
| 4.4      | Sensor Synchronization . . . . .             | 88         |
| 4.5      | <i>miPod</i> Sensor Platform . . . . .       | 90         |
| <b>5</b> | <b>Stroke Detection and Classification</b>   | <b>95</b>  |
| 5.1      | Data Acquisition . . . . .                   | 96         |
| 5.1.1    | Hardware Equipment . . . . .                 | 96         |
| 5.1.2    | Study Design . . . . .                       | 97         |
| 5.2      | Data Processing . . . . .                    | 102        |
| 5.2.1    | Stroke Detection . . . . .                   | 102        |
| 5.2.2    | Stroke Classification . . . . .              | 109        |
| 5.3      | Evaluation and Results . . . . .             | 111        |
| 5.3.1    | Stroke Detection . . . . .                   | 112        |
| 5.3.2    | Stroke Classification . . . . .              | 114        |
| 5.4      | Discussion . . . . .                         | 117        |
| 5.4.1    | Stroke Detection . . . . .                   | 118        |
| 5.4.2    | Stroke Classification . . . . .              | 119        |
| 5.4.3    | Improvements . . . . .                       | 121        |
| 5.5      | Summary . . . . .                            | 122        |
| <b>6</b> | <b>Ball Spin and Speed Estimation</b>        | <b>123</b> |
| 6.1      | Video Data Labeling . . . . .                | 123        |
| 6.2      | Data Acquisition . . . . .                   | 129        |
| 6.2.1    | Hardware Equipment . . . . .                 | 129        |
| 6.2.2    | Study Design . . . . .                       | 130        |
| 6.3      | Data Processing . . . . .                    | 132        |
| 6.3.1    | Ball and Racket Motion Assumptions . . . . . | 133        |
| 6.3.2    | Racket Speed Calculation . . . . .           | 135        |
| 6.3.3    | Rebound Model . . . . .                      | 142        |

|          |                                                   |            |
|----------|---------------------------------------------------|------------|
| 6.4      | Evaluation and Results . . . . .                  | 147        |
| 6.5      | Discussion . . . . .                              | 151        |
| 6.6      | Summary . . . . .                                 | 155        |
| <b>7</b> | <b>Ball Impact Localization</b>                   | <b>157</b> |
| 7.1      | Theory . . . . .                                  | 157        |
| 7.1.1    | Time-Difference-of-Arrival . . . . .              | 158        |
| 7.1.2    | Vibration Propagation . . . . .                   | 160        |
| 7.1.3    | Technical Challenges . . . . .                    | 162        |
| 7.2      | Data Acquisition . . . . .                        | 163        |
| 7.2.1    | Study Design . . . . .                            | 163        |
| 7.2.2    | Piezo-electric Measurement Circuit . . . . .      | 166        |
| 7.3      | Data Processing . . . . .                         | 167        |
| 7.3.1    | Impact Detection . . . . .                        | 167        |
| 7.3.2    | Time Difference Distribution Model . . . . .      | 168        |
| 7.3.3    | Impact Localization . . . . .                     | 174        |
| 7.4      | Evaluation and Results . . . . .                  | 178        |
| 7.5      | Discussion . . . . .                              | 183        |
| 7.6      | Summary . . . . .                                 | 185        |
| <b>8</b> | <b>Racket Instrumentation</b>                     | <b>187</b> |
| 8.1      | Racket Regulations . . . . .                      | 187        |
| 8.2      | Hardware Design . . . . .                         | 189        |
| 8.2.1    | Sensor Placement . . . . .                        | 189        |
| 8.2.2    | Unobtrusive Integration . . . . .                 | 190        |
| 8.3      | Embedded Implementation . . . . .                 | 193        |
| 8.4      | Feedback Application . . . . .                    | 198        |
| <b>9</b> | <b>Summary, Discussion and Conclusion</b>         | <b>201</b> |
| 9.1      | Summary and Discussion of Contributions . . . . . | 201        |
| 9.1.1    | Table Tennis Analysis . . . . .                   | 201        |
| 9.1.2    | Racket Instrumentation . . . . .                  | 207        |
| 9.2      | Outlook . . . . .                                 | 208        |
| 9.3      | Conclusion . . . . .                              | 210        |
|          | <b>List of Figures</b>                            | <b>213</b> |
|          | <b>List of Tables</b>                             | <b>217</b> |
|          | <b>Bibliography</b>                               | <b>219</b> |



# List of Abbreviations

|       |                                          |     |
|-------|------------------------------------------|-----|
| ADC   | Analog-Digital Converter                 | 167 |
| AMR   | Anisotropic Magneto-Resistance           | 84  |
| BB    | Backhand Block                           | 98  |
| BBACK | Backhand Backward                        | 101 |
| BC    | Backhand Chop                            | 121 |
| BD    | Backhand Drive                           | 98  |
| BFORE | Backhand Foreward                        | 101 |
| BP    | Backhand Push                            | 98  |
| BS    | Backhand Sidespin                        | 121 |
| BSt   | Backhand Strike                          | 121 |
| BT    | Backhand Topspin                         | 98  |
| CPU   | Central Processing unit                  | 58  |
| CV    | Cross-Validation                         | 54  |
| DC    | Direct Current                           | 125 |
| DMA   | Direct Memory Access                     | 65  |
| DoF   | Degree-of-Freedom                        | 79  |
| DT    | Decision Tree                            | 59  |
| DTTB  | Deutscher Tischtennis Bund               | 98  |
| ECST  | Embedded Classification Software Toolbox | 62  |
| FB    | Forehand Block                           | 98  |
| FBACK | Forehand Backward                        | 101 |
| FC    | Forehand Chop                            | 121 |
| FD    | Forehand Drive                           | 98  |
| FFORE | Forehand Foreward                        | 101 |
| FFT   | Fast Fourier Transformation              | 209 |
| FN    | False Negative                           | 55  |
| FP    | False Positive                           | 55  |
| FP    | Forehand Push                            | 98  |
| FPU   | Floating Point Unit                      | 66  |

|      |                                        |     |
|------|----------------------------------------|-----|
| FS   | Forehand Sidespin                      | 121 |
| FSt  | Forehand Strike                        | 121 |
| FT   | Forehand Topspin                       | 98  |
| FTSP | Flooding Time Synchronization Protocol | 92  |
| GMR  | Giant Magneto-Resistance               | 84  |
| GPS  | Global Positioning System              | 89  |
| I/O  | Input/Output                           | 58  |
| IMMU | Inertial-Magnetic Measurement Unit     | 84  |
| IMU  | Inertial Measurement Unit              | 79  |
| IrDA | Infrared Data Association              | 93  |
| ITTF | International Table Tennis Federation  | 14  |
| KDE  | Kernel Density Estimation              | 176 |
| kNN  | k-Nearest-Neighbors                    | 51  |
| LARC | List of Authorized Coverings           | 187 |
| LED  | Light Emitting Diode                   | 192 |
| LIN  | Linear Kernel                          | 50  |
| LR   | Linear Regression                      | 175 |
| LSOS | Leave-One-Subject-Out                  | 54  |
| MC   | Multi Channel                          | 61  |
| MCU  | Microcontroller Unit                   | 167 |
| MEMS | Micro-Electro-Mechanical System        | 77  |
| MTJ  | Magnetic Tunneling Junction            | 84  |
| NB   | Naïve Bayes                            | 50  |
| NTP  | Network Time Protocol                  | 89  |
| PCB  | Printed Circuit Board                  | 188 |
| POLY | Polynomial Kernel                      | 50  |
| RAM  | Random Access Memory                   | 58  |
| RBF  | Radial-Basis-Function Kernel           | 50  |
| RBS  | Reference Broadcast Synchronization    | 89  |
| RF   | Random Forest                          | 50  |
| RH   | Relative Humidity                      | 93  |
| RI   | Relative Importance                    | 49  |
| RMSE | Root Mean Square Error                 | 178 |
| RT   | Response Time                          | 62  |
| RTC  | Real-Time Clock                        | 89  |
| RTOS | Real-Time Operating System             | 61  |
| SC   | Single Channel                         | 61  |
| SV   | Support Vector                         | 52  |
| SVM  | Support Vector Machine                 | 50  |

|      |                                            |     |
|------|--------------------------------------------|-----|
| TDDM | Time-Difference-Distribution-Model         | 162 |
| TDoA | Time-Difference-of-Arrival                 | 158 |
| TN   | True Negative                              | 55  |
| ToA  | Time-of-Arrival                            | 158 |
| TP   | True Positive                              | 55  |
| TPSN | Timing-Sync Protocol for Sensor Networks   | 89  |
| TTR  | Table Tennis Ranking                       | 98  |
| USB  | Universal Serial Bus                       | 91  |
| WCET | Worst Case Execution Time                  | 62  |
| WEKA | Waikato Environment for Knowledge Analysis | 110 |



# List of Symbols

|                                  |                 |                                                                             |    |
|----------------------------------|-----------------|-----------------------------------------------------------------------------|----|
| $C$ .....                        | $[-]$ .....     | Classes per classification system .....                                     | 43 |
| $\mathbf{q}_V$ .....             | $[-]$ .....     | Feature vector .....                                                        | 43 |
| $\hat{\mathbf{q}}_V$ .....       | $[-]$ .....     | Selected feature vector .....                                               | 44 |
| $\mathcal{S}(n_s, m_s, t)$ ..... | $[-]$ .....     | Digitized times-series data stream at the<br>time $t$ .....                 | 45 |
| $n_s$ .....                      | $[-]$ .....     | Number of sensor types .....                                                | 45 |
| $m_s$ .....                      | $[-]$ .....     | Number of data channels per sensor<br>type $n_s$ .....                      | 45 |
| $L$ .....                        | $[-]$ .....     | Total signal length .....                                                   | 45 |
| $x_i$ .....                      | $[-]$ .....     | Data sample at time $t$ .....                                               | 45 |
| $T_A$ .....                      | $[s]$ .....     | Sampling interval .....                                                     | 45 |
| $w_L$ .....                      | $[s]$ .....     | Window length .....                                                         | 46 |
| $w_O$ .....                      | $[s]$ .....     | Window overlap .....                                                        | 46 |
| $w_i$ .....                      | $[-]$ .....     | Data window at time $t$ .....                                               | 46 |
| $Q$ .....                        | $[-]$ .....     | Number of features in feature vector $\mathbf{q}_V$ ...                     | 47 |
| $Q_{SC}$ .....                   | $[-]$ .....     | Number of single-channel features in<br>feature vector $\mathbf{q}_V$ ..... | 47 |
| $Q_{MC}$ .....                   | $[-]$ .....     | Number of multi-channel features in<br>feature vector $\mathbf{q}_V$ .....  | 48 |
| $\Psi_{RI}$ .....                | $[-]$ .....     | Relative feature importance .....                                           | 49 |
| $\theta_{\Psi_{RI}}$ .....       | $[-]$ .....     | Threshold for the relative feature<br>importance $\Psi_{RI}$ .....          | 49 |
| $\mathbf{s}_V$ .....             | $[-]$ .....     | Support vectors of a SVM .....                                              | 51 |
| $\alpha_0$ .....                 | $[-]$ .....     | Hyperplane intercept of a SVM .....                                         | 51 |
| $\boldsymbol{\alpha}_V$ .....    | $[-]$ .....     | Weights of all support vectors $\mathbf{s}_V$ of a SVM ..                   | 51 |
| $S_V$ .....                      | $[-]$ .....     | Number of support vectors $\mathbf{s}_V$ of a SVM ...                       | 52 |
| $SVM_{size}$ .....               | $[kByte]$ ..... | Memory consumption of a SVM .....                                           | 58 |
| $A$ .....                        | $[-]$ .....     | Number of weights $\boldsymbol{\alpha}_V$ of a SVM .....                    | 59 |
| $BpV$ .....                      | $[bit]$ .....   | Number of representing bits per value .....                                 | 59 |

|                  |                  |                                            |    |
|------------------|------------------|--------------------------------------------|----|
| $\#SVM$          | $[-]$            | Number of SVMs per classification system   | 59 |
| $RF_{size}$      | $[kByte]$        | Memory consumption of a RF                 | 59 |
| $\#DT$           | $[-]$            | Number of DTs per classification system    | 59 |
| $d_{tree}$       | $[-]$            | Tree depth of a DT                         | 59 |
| $DATA_{size}$    | $[kByte]$        | Memory consumption of input data           |    |
|                  | $S(n_s, m_s, t)$ |                                            | 60 |
| $FEATURE_{size}$ | $[kByte]$        | Memory consumption of feature vector $q_V$ | 60 |
| $SVM_{effort}$   | $[-]$            | Mathematical effort of a SVM system        | 62 |
| $RF_{effort}$    | $[-]$            | Mathematical effort of a binary RF system  | 63 |
| $T_{WCET, H_i}$  | $[s]$            | Worst case execution time of thread $H_i$  | 73 |
| $T_{RT}$         | $[s]$            | Response time of a scheduling approach     | 73 |
| $h_s$            | $[-]$            | Integral factor of earliest possible event |    |
|                  |                  | recurrence                                 | 74 |
| $F$              | $[N]$            | Inertial force of a moving object          | 79 |
| $m$              | $[kg]$           | Mass of a moving object                    | 80 |
| $a$              | $[m/s^2]$        | Total acceleration of a moving object      | 80 |
| $a_0$            | $[m/s^2]$        | Inertial acceleration of a moving object   | 80 |
| $\omega_0$       | $[rad/s]$        | Inertial angular velocity of a moving      |    |
|                  |                  | object                                     | 80 |
| $r$              | $[m]$            | Position vector of a moving object         | 80 |
| $v$              | $[m/s]$          | Velocity vector of a moving object         | 80 |
| $x_F$            | $[m]$            | Displacement in X-direction based on $F_k$ |    |
|                  |                  | and $F_b$                                  | 80 |
| $F_a$            | $[N]$            | Linear acting force based on $a$           | 80 |
| $F_k$            | $[N]$            | Spring force                               | 81 |
| $F_b$            | $[N]$            | Damper force                               | 81 |
| $k$              | $[N/m]$          | Spring constant                            | 81 |
| $b$              | $[Ns/m]$         | Damper constant                            | 81 |
| $a_m$            | $[m/s^2]$        | Effective sensed acceleration of an        |    |
|                  |                  | accelerometer                              | 82 |
| $g_{earth}$      | $[m/s^2]$        | Gravitational acceleration of the earth    | 82 |
| $F_G(t)$         | $[N]$            | Gyroscope driving force                    | 82 |
| $F_0$            | $[N]$            | Maximum amplitude of $F_G(t)$              | 82 |
| $f_0$            | $[Hz]$           | Oscillating frequency of $F_G(t)$          | 82 |
| $F_c$            | $[N]$            | Coriolis force                             | 82 |
| $y_F$            | $[m]$            | Displacement in Y-direction based on $F_k$ |    |
|                  |                  | and $F_b$                                  | 82 |
| $B$              | $[T]$            | Magnetic field of the earth                | 84 |
| $I$              | $[A]$            | Flow direction of measurement current      | 85 |

|                     |              |                                                             |     |
|---------------------|--------------|-------------------------------------------------------------|-----|
| $U_H$               | [V]          | Hall voltage                                                | 85  |
| $A_H$               | [ $m^3/C$ ]  | Hall constant                                               | 85  |
| $d_H$               | [m]          | Hall conductor thickness                                    | 85  |
| $R_{AMR}$           | [ $\Omega$ ] | Total AMR resistance                                        | 85  |
| $R_{0,AMR}$         | [ $\Omega$ ] | Maximum AMR resistance                                      | 85  |
| $\Delta R_{AMR}$    | [ $\Omega$ ] | AMR resistivity factor                                      | 85  |
| $\Theta_{AMR}$      | [rad]        | Angle between $I$ and $M_{AMR}$                             | 85  |
| $M_{AMR}$           | [A/m]        | AMR magnetization field                                     | 85  |
| $U_P(t)$            | [V]          | Measured piezo voltage based on $F_P(t)$                    | 86  |
| $F_P(t)$            | [V]          | Applied piezo force                                         | 86  |
| $d_{33}$            | [C/N]        | Piezo strain constant                                       | 86  |
| $\epsilon_0$        | [C/Vm]       | Vacuum permittivity                                         | 86  |
| $\epsilon_{33}$     | [—]          | Relative piezo crystal permittivity                         | 86  |
| $h_P$               | [m]          | Piezo thickness                                             | 86  |
| $Q_P(t)$            | [C]          | Generated electric charges of the piezo crystal             | 86  |
| $E_{raw}(x_i)$      | [ $g^2$ ]    | Raw signal energy at data sample $x_i$                      | 103 |
| $E_{filt}(x_i)$     | [ $g^2$ ]    | Filtered signal energy at data sample $x_i$                 | 103 |
| $S(x_i)$            | [ $g^2$ ]    | Negative masked signal at data sample $x_i$                 | 103 |
| $\theta$            | [ $g^2$ ]    | Peak detection threshold combining $\mu$ , $h$ and $\sigma$ | 104 |
| $\kappa_d$          | [—]          | Segmentation range used for the stroke detection            | 104 |
| $h$                 | [—]          | Distance factor for the peak detection threshold $\theta$   | 104 |
| $P(x_j)$            | [—]          | Peak indication signal at data sample $x_j$                 | 104 |
| $H(x_j)$            | [ $g^2$ ]    | Local maximum signal at data sample $x_j$                   | 104 |
| $\hat{P}(x_i)$      | [—]          | Detected strokes for at sample $x_i$                        | 105 |
| $E_{raw}(\omega_i)$ | [ $g^2$ ]    | Raw signal energy at data window $\omega_i$                 | 106 |
| $P(w_j)$            | [—]          | Peak signal at data window $w_j$                            | 107 |
| $H(w_j)$            | [ $g^2$ ]    | Local maximum signal at data window $w_j$                   | 107 |
| $\kappa_v$          | [—]          | Segmentation range used for stroke validation               | 108 |
| $\kappa_c$          | [—]          | Segmentation range used for stroke classification           | 109 |
| $v_b$               | [m/s]        | Labeled speed of the ball                                   | 124 |
| $p_A$               | [m]          | Ball center vector in video frame $A$                       | 124 |
| $p_B$               | [m]          | Ball center vector in video frame $B$                       | 124 |
| $p_{w,i}$           | [1/m]        | Pixel width at frame $i$                                    | 124 |

## List of Symbols

|                                   |                 |                                                                  |     |
|-----------------------------------|-----------------|------------------------------------------------------------------|-----|
| $N_{vid}$ .....                   | $[-]$ .....     | Number of analyzed video frames $i$ .....                        | 124 |
| $M_{vid}$ .....                   | $[-]$ .....     | Number of analyzed ball envelopes $A_b$ .....                    | 124 |
| $A_b$ .....                       | $[m^2]$ .....   | Area of a ball envelope .....                                    | 124 |
| $r_b$ .....                       | $[m]$ .....     | Ball radius .....                                                | 124 |
| $\mathbf{f}_A$ .....              | $[m]$ .....     | Feature point vector in video frame $A$ .....                    | 125 |
| $\mathbf{f}_B$ .....              | $[m]$ .....     | Feature point vector in video frame $B$ .....                    | 125 |
| $\mathbf{f}$ .....                | $[m]$ .....     | Feature point translation vector .....                           | 125 |
| $\mathbf{p}$ .....                | $[m]$ .....     | Ball center translation vector .....                             | 125 |
| $\alpha_{rot}$ .....              | $[rad]$ .....   | Labeled rotation angle of the ball .....                         | 125 |
| $\omega_b$ .....                  | $[rad/s]$ ..... | Labeled angular velocity of the ball .....                       | 125 |
| $\omega_{pend}$ .....             | $[rad/s]$ ..... | Angular velocity of the pendulum .....                           | 126 |
| $G_{mot}$ .....                   | $[-]$ .....     | Gear ratio constant .....                                        | 125 |
| $D_{mot}$ .....                   | $[1/Vs]$ .....  | DC Motor constant .....                                          | 125 |
| $U_{mot}$ .....                   | $[V]$ .....     | DC motor voltage .....                                           | 126 |
| $l_{pend}$ .....                  | $[m]$ .....     | Pendulum length .....                                            | 125 |
| $v_{pend}$ .....                  | $[m/s]$ .....   | Linear pendulum speed at $\epsilon_{pend} = 0$ .....             | 126 |
| $m_{pend}$ .....                  | $[kg]$ .....    | Pendulum mass .....                                              | 126 |
| $\epsilon_{pend}$ .....           | $[rad]$ .....   | Pendulum rod deflection .....                                    | 126 |
| $J_{pend}$ .....                  | $[kgm^2]$ ..... | Pendulum inertia moment .....                                    | 126 |
| $\mathbf{v}_R(x_i)$ .....         | $[m/s]$ .....   | Linear speed of the racket at the data<br>sample $x_i$ .....     | 133 |
| $\omega_R(x_i)$ .....             | $[rad/s]$ ..... | Angular velocity of the racket at the data<br>sample $x_i$ ..... | 133 |
| $\mathbf{v}_{B,1}$ .....          | $[m/s]$ .....   | Racket speed before impact at $x_{ip}$ .....                     | 133 |
| $\mathbf{v}_{B,2}$ .....          | $[m/s]$ .....   | Racket speed after impact at $x_{ip}$ .....                      | 133 |
| $\omega_{B,1}$ .....              | $[rad/s]$ ..... | Racket rotation before impact at $x_{ip}$ .....                  | 133 |
| $\omega_{B,2}$ .....              | $[rad/s]$ ..... | Racket rotation after impact at $x_{ip}$ .....                   | 133 |
| $x_{ip}$ .....                    | $[-]$ .....     | Impact point .....                                               | 133 |
| $\mathbf{v}_\mu$ .....            | $[m/s]$ .....   | Mean ball velocity during impact .....                           | 133 |
| $\zeta_{geo}$ .....               | $[deg]$ .....   | Ball impact angle .....                                          | 133 |
| $\mathbf{c}_{geo}$ .....          | $[m]$ .....     | Ball impact position vector on the blade ..                      | 134 |
| $\mathbf{v}_{l,R}(x_i)$ .....     | $[m/s]$ .....   | Linear component of the racket speed ....                        | 134 |
| $\mathbf{v}_{r,R}(x_i)$ .....     | $[m/s]$ .....   | Rotational component of the racket speed                         | 134 |
| $\tilde{\mathbf{a}}_R(x_i)$ ..... | $[m/s^2]$ ..... | Gravitiy compensated racket acceleration .                       | 134 |
| $\mathcal{W}$ .....               | $[m/rad]$ ..... | Wrist rotation matrix .....                                      | 134 |
| $\delta_{geo}$ .....              | $[rad]$ .....   | Wrist rotation angle vector .....                                | 134 |
| $\mathbf{a}_{geo}$ .....          | $[m]$ .....     | Wrist rotation center vector .....                               | 134 |
| $x_m$ .....                       | $[-]$ .....     | Starting point for integration .....                             | 134 |
| $\mathbf{a}_R(x_i)$ .....         | $[m/s^2]$ ..... | Racket acceleration at the data sample $x_i$ ..                  | 135 |

|                                   |           |                                                                   |     |
|-----------------------------------|-----------|-------------------------------------------------------------------|-----|
| $\Lambda(x_i)$                    | $[g]$     | Root signal energy at data sample $x_i$                           | 135 |
| $\kappa_e$                        | $[-]$     | Segmentation range for ball spin and speed estimation             | 136 |
| $\mathbf{a}_{\text{filt},R}(x_i)$ | $[m/s^2]$ | Filtered acceleration of the racket at the data sample $x_i$      | 139 |
| $\Phi_R(x_i)$                     | $[rad]$   | Relative angle vector at data sample $x_i$                        | 139 |
| $\mathcal{R}_{\Delta\alpha}(x_i)$ | $[-]$     | X-axis rotation matrix                                            | 139 |
| $\mathcal{R}_{\Delta\beta}(x_i)$  | $[-]$     | Y-axis rotation matrix                                            | 139 |
| $\mathcal{R}_{\Delta\gamma}(x_i)$ | $[-]$     | Z-axis rotation matrix                                            | 139 |
| $e_R$                             | $[-]$     | Restitution coefficient                                           | 144 |
| $\mu_R$                           | $[-]$     | Friction coefficient                                              | 144 |
| $m_b$                             | $[kg]$    | Ball mass                                                         | 144 |
| $m_w$                             | $[kg]$    | Weighting object mass                                             | 144 |
| $F_N$                             | $[N]$     | Normal force based on $\mathbf{g}_{\text{earth}}$                 | 144 |
| $F_R$                             | $[N]$     | Friction force based on $\mu_R$                                   | 144 |
| $\lambda_R$                       | $[-]$     | Friction case indicator                                           | 145 |
| $m_R$                             | $[kg]$    | Displaced rubber mass during impact                               | 145 |
| $t_{ip}$                          | $[s]$     | Impact duration                                                   | 145 |
| $k_R$                             | $[N/m]$   | Rubber spring constant                                            | 145 |
| $k_{ip}$                          | $[kg]$    | Rubber elasticity parameter                                       | 145 |
| $\mathcal{A}$                     | $[-]$     | Transformation matrix of ball speed into ball speed               | 146 |
| $\mathcal{B}$                     | $[-]$     | Transformation matrix of ball spin into ball speed                | 146 |
| $\mathcal{C}$                     | $[-]$     | Transformation matrix of ball speed into ball spin                | 146 |
| $\mathcal{D}$                     | $[-]$     | Transformation matrix of ball spin into ball spin                 | 146 |
| $\Delta t_{i,j}$                  | $[s]$     | Time difference between sensor position $S_{ip}^i$ and $S_{ip}^j$ | 159 |
| $S_{ip}^i$                        | $[m]$     | Sensor position of sensor $i$                                     | 159 |
| $S_{ip}^j$                        | $[m]$     | Sensor position of sensor $j$                                     | 159 |
| $P_{ip}$                          | $[m]$     | Ball impact position                                              | 159 |
| $H_{i,j}^A$                       | $[m]$     | Hyperbola for sensor $i$ and $j$ based on the geometry            | 159 |
| $c_{vib}$                         | $[m/s]$   | Vibration propagation speed                                       | 159 |
| $H_{i,j}^B$                       | $[m]$     | Hyperbola for sensor $i$ and $j$ based on $c_{vib}$               | 159 |
| $N_m$                             | $[-]$     | Number of measuring points $P_m(X, Y)$                            | 165 |

## List of Symbols

|                                   |                                                                   |     |
|-----------------------------------|-------------------------------------------------------------------|-----|
| $P_m(X, Y) \dots [m]$             | Measuring point at position $m$ on the racket surface.....        | 165 |
| $U_{supply} \dots [V]$            | Piezo-electric supply voltage .....                               | 167 |
| $\zeta_{vib,max} \dots [V]$       | Maximum piezo-electric threshold for the impact detection.....    | 168 |
| $\zeta_{vib,min} \dots [V]$       | Minimum piezo-electric threshold for the impact detection.....    | 168 |
| $S_{ip}^j(x_i) \dots [AU]$        | Amplitude of sensor $j$ at data sample $x_i$ ..                   | 170 |
| $\hat{S}_{ip}^j(x_i) \dots [AU]$  | Normalized amplitude of sensor $j$ at the data sample $x_i$ ..... | 170 |
| $\zeta_{high,j} \dots [AU]$       | Maximum piezo-electric threshold for $\tau_j$ detection.....      | 170 |
| $\zeta_{low,j} \dots [AU]$        | Minimum piezo-electric threshold for $\tau_j$ detection.....      | 170 |
| $\tau_j \dots [-]$                | Time indicator for incoming vibration wavefront .....             | 170 |
| $q_{\Pi} \dots [-]$               | Rejection distance for outlier detection ...                      | 172 |
| $\Delta t_{j,N_m} \dots [s]$      | Averaged time difference for sensor pair $j$ and $N_m$ .....      | 172 |
| $I_{DT}(X, Y) \dots [m]$          | Impact position based on the decision tree algorithm .....        | 175 |
| $f_j(X) \dots [-]$                | Regression function for sensor pair $j$ .....                     | 175 |
| $a_j \dots [-]$                   | Slope of regression function $f_j(X)$ .....                       | 175 |
| $b_j \dots [-]$                   | Intercept of regression function $f_j(X)$ ....                    | 175 |
| $d_j(\hat{X}, \hat{Y}) \dots [-]$ | Shortest orthogonal distance to $P_{ip}(X, Y)$ .                  | 176 |
| $I_{LR}(X, Y) \dots [m]$          | Impact position based on the linear regression .....              | 176 |
| $f_{kde}(X, Y) \dots [-]$         | Kernel density estimation function .....                          | 176 |
| $K_G \dots [-]$                   | Gaussian kernel of $f_{kde}(X, Y)$ .....                          | 176 |
| $h_{kde} \dots [-]$               | Bandwidth of $f_{kde}(X, Y)$ .....                                | 177 |
| $I_{KDE}(X, Y) \dots [m]$         | Impact position based on the kernel density estimation .....      | 178 |

# Chapter 1

## Introduction

In this chapter, the purpose and general outline of the following thesis are presented. Firstly, a motivation for the use of wearable sensors applied in table tennis is given. Secondly, current state-of-the-art related to table tennis analysis and digitalization in racket sports is shown. Subsequently, the main contributions of this thesis are defined. Lastly, the structure of the following document is summarized.

### 1.1 Overview and Motivation

#### 1.1.1 Wearables in Sports

Sports is a very exciting and worthwhile field, whose impact should not be underestimated [Chi 05]. It is undeniable, that sports is essential in keeping humans physically and mentally active [Warb 06]. Thereby, sports does not only strengthen humans' bodies and minds, but also acts as a worldwide social connector. Beginning with the spirit of the first Olympic Games in 776 before Christ, sports has a long tradition fostering the contact and understanding between different cultures and people [Xiao 17]. Besides all health aspects, this importance of sports should not be left aside and is probably its greatest benefit [Chi 08].

Many things have changed since the tournaments then. There are dozens of new kinds of sports and different disciplines. Thereby, elite athletes are supported and evaluated in professional and more often commercial ways [Baca 14]. Practicing skill levels, tactical and technical abilities as well as intuition, anticipation and motivation are the most important factors to enhance the performance of athletes. However, today's training of elite athletes is comparable to research in science

[Mori 06]. Nevertheless, sports science cannot be considered as a single research domain, it is rather an interdisciplinary field of psychology, health, computer science, sports management and economy [Xiao 17].

In that context, performance assessment and evaluation of athletes combined with specialized training exercises and scenarios traditionally take place in a well-defined laboratory environment. There, various diagnostic indicators describing the biomechanics and body physiology of each athlete can be objectively investigated in detail. However, the interpretation of data obtained by such measurement laboratories is often limited to the effect that athletes behave differently in unusual training situations and performance characteristics differ from their standard values. An objective analysis is essential for a more precise understanding of athletes' constitutions and capabilities, but it should be gained without the drawbacks of a laboratory environment. Taking research about sports out of the laboratory into the natural environment of each kind of sports is the logical next step to effectively improve performances from training sessions to competitions.

For that reason, monitoring athletes within their familiar training environments is facilitated by adopting ubiquitous computing technologies. Thereby, different sensors can be integrated into sports equipment and clothes or worn as wearable devices [Chi 08]. These wearable sensors, or simply *Wearables*, could be portable accessories like smartphones and smartwatches, garments, wristbands or additional devices like heart rate and respiration sensors [Morr 08]. During the last two decades, many advances in microelectronics regarding miniaturized and low-cost sensors, low-power electronic components or energy-friendly wireless technologies opened new ways for a more comprehensive integration of those Wearables [Amft 09]. Therefore, it is also possible that almost every sensor can unobtrusively be integrated into any type of sports equipment like shoes, shirts, balls or rackets. Of course, this pervasive integration of sensors enables even deeper and more detailed insights into processes and movements of athletes as would be accessible with observational information from outside. Figuratively speaking, the athlete is not brought into the laboratory, but the laboratory to the athlete.

Among many measurable physical values and types of sensors, Wearables usually include inertial devices like accelerometers and gyroscopes, if investigations of motions are of interest. They are one of the most important data sources for state-of-the-art movement analysis [McCa 09].

Using data of these sensors, different methodologies can be applied to gather information from the athletes' motion. These methods enable the derivation and computation of further diagnostic biomechanic parameters, including motion angles, covered distances, movement speeds or trajectories.

Furthermore, sensor-based data analysis also opens an objective view especially on those kinds of sports, which involve complex and highly dynamic motions. This applies in particular, if sports equipment is necessary or if the athletes' performance can only be evaluated by a qualitative point of view with difficulty. Examples are gymnastics, dancing or swimming, which involve specific movement patterns and motion sequences. But objective assessment of athletes is also of major importance, if sports equipment is required involving balls for soccer or basketball, pucks and sticks for hockey, skis, and rackets such for squash, badminton, tennis or table tennis.

However, sensor data analysis is not only restricted to professional athletes and coaches, but can also be helpful for amateurs. Thinking about the large majority of club players, who do not have access to professional training or qualified coaches, Wearables can aid them in improving their playing abilities and supporting them during training and competition. With this background, Wearables can provide individual statistical information to each player, can be used for educational applications of youth development and enthused for a variety of entertainment purposes [Baca 06]. Acting as such personalized tracking devices, Wearables are extremely popular for application in individual health, fitness and sports monitoring, for social communication or simply in lifestyle computing [Sazo 14].

Thus, various application scenarios for Wearables in sports are conceivable, which require different hardware features and sensor configurations specifically tailored for the intended purposes [Luko 04]. This especially pertains in fitness and sports environments, because Wearables have become indispensable as training tools in general [McCa 09] and consequently, they are also used for the table tennis analysis presented within this work.

### 1.1.2 Table Tennis

This thesis focuses on the application of Wearables in the context of table tennis. It is an interesting and fascinating candidate for research. Dependent on the countless assortment of different rubbers, blades and

their combinations as well as various skills, abilities and playing styles of table tennis players worldwide, numerous variations in ball speed and spin characteristics associated with different possible types of strokes are achievable. These nearly unlimited opportunities turn table tennis into a complex and challenging sports, not only for the players themselves, but also for an objective and quantitative game analysis.

This high versatility could be one reason, why around 300 million regular players worldwide spend the majority of their sportive time playing table tennis <sup>1</sup>. This number makes table tennis one of the top ten most popular sports in the world by registered players of national and international associations. Furthermore, competitions in table tennis have a long history. World Championships with singles, doubles and mixed are continuously held since 1926. In 1988 table tennis became an Olympic discipline. Besides the organized sports within clubs, leagues and tournaments, its popularity also becomes apparent by about 850 million others who follow this sport in the media or play table tennis as a leisure-time activity themselves <sup>2</sup>. Hence, table tennis is one of the most popular sports worldwide [Gros 07].

According to the regulations of the International Table Tennis Federation (ITTF) [The 18a], a standard game is played with a small white or orange colored ball of 40 mm diameter and a mass of 2.7 g made of celluloid or polymer plastics. The wooden table is always 271 cm long, 152.2 cm wide and 76 cm above the ground. The table is split into two halves and assembled with a net, which has a height of 15.25 cm. Depending on the ball color, the table surface can either be green or blue. Besides ball and table, players need a racket comprising of the wooden core blade and two rubbers, a red one and a black one for each side. The official rules allow different racket coverings, which are permanently approved by the ITTF and included into an always up-to-date list of rubbers [The 18b]. However, there are no restrictions for the blade itself, as long as it is made out of natural wood. Since every racket is unique, no standardized size can be specified. The length and width of blades are usually between 145 cm to 165 cm.

Basic table tennis strokes are drives, pushes, blocks, topspins or the beginning serves [McAf09]. A drive is a perpendicular hit of the ball, mostly with little forward spin. During a push, the ball is diagonally hit

---

<sup>1</sup><https://www.pledgesports.org/2017> – last access 11/2018

<sup>2</sup><https://www.topendsports.com/world/lists/popular-sport> – last access 11/2018

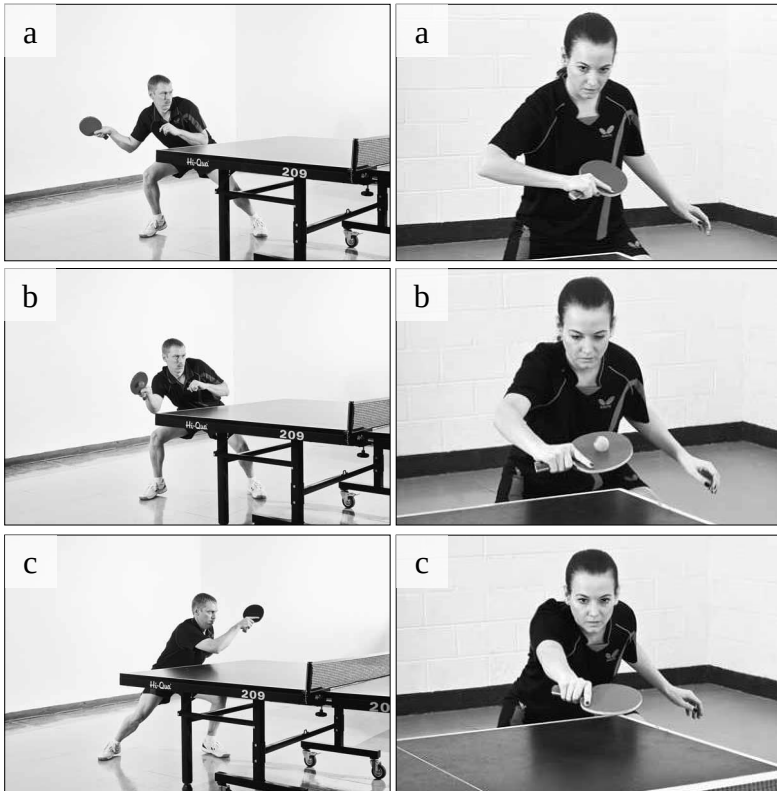


**Figure 1.1:** Example forehand topspin stroke sequence on the left side and a backhand push stroke sequence on the right side from [McAf 09]. The forehand topspin starts with a countermovement as backswing (a), then, the ball is hit with a diagonal stroke from the bottom to the top (b) and ends with an active follow-through movement (c). However, a backhand push stroke starts with centering the racket in front of the body (a), followed by a diagonal motion from the top to the bottom (b) and ends with a follow-through movement along this forearm motion (c).

from top to bottom resulting in backward spin. Block strokes are usually passive strokes with less movements, whereas topspins are active strokes, for which the ball is diagonally touched from bottom to top resulting in strong forward spin. Examples of a forehand topspin and a backhand push can be seen in Figure 1.1. Performing a serve, almost any ball speed

and spin values can be achieved. In addition to the mentioned stroke types, many more executions and modifications due to a more active or passive performance as well as speed and spin variations based on the players' skills, abilities and intentions are feasible. All these varieties can be played either with the forehand or with the backhand. Furthermore, the huge amount of different blades and rubbers enables even more technical opportunities regarding stroke types and spin generation and forwarding. The majority of players and practically all professional athletes utilize normal rubbers, but special frictionless rubbers and rubbers with short and long pimples are also allowed. Applying those materials, the speed or spin of the ball during rebound result in unusual ball rotations and flaying ball trajectories and are thus difficult to anticipate for the opponent. Moreover, different grip styles such as shakehand (usually in Western countries) or pen-holder (common in Asia) have become established [The 18a]. An example table tennis setup can be seen in Figure 1.2. The video clip<sup>3</sup> demonstrates a match from the World Championships 2018 from the official ITTF video channel<sup>4</sup> between Germany's Timo Boll and China's Ma Long.

Due to the mentioned possibilities of variation in either stroke type, ball spin and speed as well as the different materials used, table tennis is an interesting and rewarding field of research. In combination with Wearables, professional athletes as well as amateur players can benefit from sensor-based support systems and personalized match evaluation tools. Besides, Wearables in table tennis allow an unobtrusive and objective assessment of performance and progress of trainings and competitions for all athletes.

On that account, this thesis contributes with multiple sensor-based analysis approaches to provide a real-time feedback application. In particular, a detailed view on a reliable stroke detection and classification method of the mentioned basic stroke types is given. Furthermore, ball spin and speed shortly after impact are investigated indicating the player's performance during topspins and pushes. Moreover, the ball impact position on the racket is localized, which derives information about the generated ball spin and speed during a serve as well as it indicates the optimal hitting position on the blade. Since there are no ready-to-use Wearables available in table tennis, this thesis includes the

---

<sup>3</sup><https://tv.ittf.com/video/ma-long-boll-timo/1206403> – last access 11/2018

<sup>4</sup><https://www.youtube.com/user/ittfchannel> – last access 07/2018



**Figure 1.2:** Example table tennis setup of the World Championships 2018 (from video clip<sup>3</sup>). During the competitions, a blue table in combination with a white polymer plastic ball was used. In this image, Timo Boll (left) performs a topspin stroke with strong forward spin, whereas Ma Long (right) intends a block return.

development of small and light-weight sensors for the use in sports science research in general as well as specialized sensor hardware for a fully instrumentation of a table tennis racket. All computed data are finally visualized on a mobile device.

## 1.2 Related Work

In this section, related work in the field of table tennis and comparable racket sports analysis is presented. Thereby, scientific literature about studies of the players' and rackets' motion, physics of the flying ball including its rebound behavior on different kinds of rackets and bats as well as approaches for ball speed and spin estimation are considered. Furthermore, investigations of impact effects on rackets and their implications on the forwarded ball are described. Additionally, hardware developments of Wearables applied on table tennis are introduced.

### 1.2.1 Stroke and Motion Analysis in Racket Sports

Stroke and motion analysis is popular for a lot of racket sports disciplines. The following summary comprises general aspects about the human arm and body movement as well as kinematic investigations during

the swing, shot and stroke motion for general racket sports, tennis, badminton and table tennis. The presented approaches and methods were either based on the information of video cameras and image sequences or developed on the basis of time-series inertial or acoustic sensor data.

### General Stroke and Motion Analysis

Methods for a general stroke analysis using video-based data from television videos were introduced by [Duan 05]. They developed a unified framework for shot classification in tennis, soccer, basketball, volleyball and table tennis. Firstly, specific shot classes were identified. Thereby, they defined a shot class as the interaction of a player with the ball, for example, a pass in soccer or a stroke in table tennis. Secondly, a set of common representations for each shot class was generated based on motion, color or duration of the video frame segments. Thirdly, machine learning approaches including a Naïve Bayes classifier and a Support Vector Machine were applied to classify these shots into pre-defined categories. Shot classes from typically broadcasted television games were detected yielding overall accuracies of 85 % to 95 %. However, the authors emphasized that these results depended on the numbers and types of pre-defined classes as well as on the quality, perspective and noise of the input video streams.

In contrast to video-based detection of shots applied on different kinds of sports, in [Cham 15], a systematic review about the usage of wearable sensors for sport specific movement evaluation was described. Thereby, contributions for various racket and bat-and-ball sports involving player-mounted inertial sensors exemplary for baseball, tennis, cricket and many other kinds of sports were summarized. It was concluded that Wearables can be used for the detection of sport specific movements for a wide range of individual and team sports.

An approach for real-time and interactive human body control using inertial measurement units on both hands, both ankles, head and torso was presented by [Liu 11b]. Thereby, an animation of an avatar was controlled in real-time to visualize different sportive movements, including table tennis strokes. The computed motion was evaluated by a video capturing system as reference. The authors analyzed mean errors of the joint angles between the reconstructed and real movement from data of the head, the torso and both hands. Additionally, they compared their results with similar existing approaches and commercial products. Achieved errors stayed under 4 deg for all tested movements compared

to the video capturing reference and to other approaches in literature. It has yet to be shown, how these algorithms perform for intermittent movement events in contrast to separate motion sequences.

A swing motion analysis method utilizing Wearables for different kinds of sports like squash, tennis or badminton was investigated by [Anan 17]. They presented a generalized framework comprising Convolutional Neural Networks and feature-based classification approaches for the detection and recognition of various types of shots. Their model was based on data of a single wrist-worn inertial sensor. Depending on the kind of sports and the applied methodology, shots could be detected with an accuracy of 83 % to 96 % and classified into categories, for example forehand, backhand or serve with an accuracy of 76 % to 94 %. However, the better results were only possible with high computational efforts and therefore not directly realizable on Wearables, but on high performing cloud-based systems.

Swing motions were also analyzed in [Ahma 14]. A three-dimensional trajectory of a golf club was computed using the acceleration and angular velocity data of an inertial sensor mounted on the shaft of the golf club. Thereby, different filters and acceleration integrations as well as principal component analyses and regressions were used to estimate the absolute position of the golf club. An evaluation of the system was based on a motion capturing device. They achieved results with mean errors of less than 14 mm. Furthermore, they claimed that the proposed method is generalizable for swing investigation in other sports like tennis.

### **Stroke and Motion Analysis in Tennis and Badminton**

In particular, comparable to table tennis is stroke and motion analysis applied for tennis and badminton. Many contributions for motion analysis, event recognition, stroke detection and classification in tennis and badminton were based on methods from video and television broadcasting data.

One event detection example was introduced in [Tien 08]. On the basis of broadcasted television tennis videos, a mining-based approach detected faults, aces, rallies and net events within a tennis game. Temporal and audiovisual features comprising moving distance and positions of players, applause and sound effects during the match were computed and assigned to specific patterns, which were finally used to characterize the correspondence between video segments and game events. With this approach, an overall event detection precision of 82 % and a recall

of 78 % were achieved. The authors remarked that sport specific and more elaborate audiovisual features could probably enhance the event detection performance.

Another example for event detection in tennis was presented in the work of [Conn 11a]. In their paper, a complete tennis match was indexed into key events like strokes, serves, rallies, changes of ends or sets. With data from nine cameras around the tennis court, players were continuously tracked and extracted from the current scenes. Using this segmented information, significant changes within the players' histograms were interpreted as such key events. Exemplary for changes of ends and serves, these key events were detected with a precision of 100 % and 80 %, respectively. However, the proposed algorithms suffered from noisy input data and longer stationary fields of view, especially during key events like serves.

Stroke classification in tennis was also performed in [Conn 10] with a single low-cost video camera. Different types of tennis strokes including forehand and backhand strokes as well as serves were recognized using key postures of elite athletes during competitive tennis matches. Similar to the above mentioned approach in [Conn 11a], players were extracted from video images. On the basis of statistical information from image moments of these segmented players and their postures, video sequences were assigned to specific stroke types, resulting in an overall accuracy of 97 %. However, the authors mentioned that the computation overhead of image moments is too high for a real-time implementation.

Another automated stroke recognition approach in tennis based on ordinary television videos was introduced by [Petk 01]. Here, different classes comprising forehand and backhand strokes as well as serves were classified using a Hidden Markov Model. Novel features comprising pie and skeleton features for the orientation and the position determination of the players' main body parts like arms, legs and the heads were investigated. The best performing features resulted in a stroke type recognition accuracy of 93 %. They pointed out that additional data about the position of ball and player relative to each other and the tennis court would increase overall results.

Data from images and videos were also examined to classify multiple stroke types in badminton. In [Rama 14], forehand and backhand strokes, smashes and all other strokes as a rest type were classified using dense trajectories and the optical flow of pixel fields around segmented players. This meta data were subsequently provided as input

for a Support Vector Machine. Results were verified by manual annotation of videos from professional athletes. The authors achieved overall classification accuracies of 93 % on unseen data. However, the Support Vector Machine was trained with data of only one athlete and evaluated with data of only one other athlete. Therefore, it has to be expected that this approach could not generally be valid.

Further investigations in badminton incorporating image data were provided by [Naga 12] for a smash motion analysis. Within this study, they mapped the players' movements into a more abstract description space of position over time and velocity over time. This representations described the initial movements as circles, lines, dots or curves. According to the generated representations, results were qualitatively discussed regarding calculation principles and its application to performance assessment of players in a sports science context. Additionally, they emphasized their conclusions with computer simulations and data acquired from elite badminton players.

In addition to methods primarily based on computer vision, [Sali 10] applied a motion capturing system to analyze the arm movement during badminton smashes. Eight markers mounted on the upper body of the player defined the movement and absolute position of the arm during the smash performance. The authors qualitatively compared the stroke motion of male and female groups of each seven subjects with basic skills and knowledge in badminton. They concluded, that male players reached higher grip velocities and less variations in the wrist rotation than female players despite similar playing abilities.

Alongside video-based techniques, stroking motions in tennis were also analyzed with Wearables and inertial data from gyroscopes. In a study by [Iiji 10], experiments with one single sensor located at the players' waist were presented, whereas three sensors, one on the chest, one on the upper arm and one on the hand were included in studies by [Ahma 09]. Both groups emphasized differences in the movement of expert and novice tennis players during forehand and backhand strokes as well as during serves by a qualitative comparison of angular velocity patterns. Here, no automated detection or classification of events and stroke types were conducted.

Wearables for an actual stroke classification in tennis were used in [Conn 11b]. In addition to gyroscopes, accelerometers and geo-magnetic devices were included as well. One wearable device was attached to the forearm of each player. Firstly, a stroke detection algorithm based on

the magnitude of the acceleration signal was implemented, secondly, a Naïve Bayes approach classified strokes into the categories serve, forehand stroke and backhand stroke. Data were obtained in a research study with subjects of different skill levels and playing abilities. Overall classification results of 79 % were achieved for only acceleration data, of 76 % for only gyroscope data and 76 % for only geo-magnetic data. Furthermore, the authors defined a best performing set comprising of all three sensor types for an optimal classification system, which resulted in a total accuracy of 90 %. However, an investigation of how each sensor type contributed to the final class decision was missing.

In a different study of [Ahma 06], swinging motions in tennis were investigated using only accelerometers. Here, three acceleration sensors were attached to the knee, the leg and the wrist. Based on the obtained kinematic data, key features during serves for proficiency and performance analysis were investigated. Applying qualitative analyses of multiple overlayed signal shapes during one stroke event, it was found that the linked translational and rotational waist motion as well as the pattern repeatability during backward and forward swings were key features that determined the contrast in skill level of different players.

In addition to tennis, Wearables for stroke recognition in badminton were presented in [Wang 16]. In this study, a sensor network comprising four inertial measurement units attached to both wrists, the waist and the ankle was developed. Based on the acceleration and gyroscope data, statistical features were computed and used to train a Hidden Markov Model. Then, 14 different stroke types including serves, clears, rushings, chops, lobs, pushes and hooks played with forehand and backhand were classified. The proposed algorithms were evaluated in a research study with players of different age, sex and skill level. Thereby, an overall classification accuracy of 98 % confirmed the validity of this approach. Additionally, the authors investigated the importance of different sensor locations and concluded that data from the ankle position near the racket were most decisive to the results.

A different motion analysis system in badminton was introduced by [Kian 09]. A Wearable was mounted at the base of the racket's head. It consisted of an accelerometer and an acoustic sensor. The authors investigated a possible relationship between the racket speed and the shuttle ball speed, which could quantify the analysis of badminton smashes. Both data source were merged based on a Fuzzy Inference System. It was found that the shuttle ball speed could be estimated with a root mean

square error of 3.2 m/s compared to a high-speed camera reference. Admittedly, only less training data were available and potential clipping of the acoustic signals during strong smash strokes increased the estimation error.

Acoustic and inertial data for badminton stroke classification were also used in a study of [Lin 17]. They attached the acoustic sensor at the base of the racket's head and the inertial sensor at the end of the racket's handle. Statistical features were computed and used as input for a subsequent stroke classification approach with Random Forests. Net play, lob, drive, clear and smash were differentiated as classes. The authors evaluated the trained classifier with a research study including ten badminton players. Overall, strokes could be classified with an accuracy of 95 %. However, if the model was applied to unseen data, the results decreased to 80 %, which raises the question of a general applicability to other racket sports.

One combined approach for video-based data and accelerometer data was presented in [O Co 10] for an automatic stroke classification system in tennis. Players were equipped with four acceleration sensors at both arms, the waist and the chest and were simultaneously filmed using eight video cameras. Firstly, a stroke detection algorithm was performed based on the acceleration magnitude. Secondly, a Support Vector Machine and a k-Nearest-Neighbour algorithm were implemented to classify forehand and backhand strokes as well as serves. On the one side, features were based on raw acceleration data samples, on the other side, contour-based features of the video data were computed. It was found that the k-Nearest-Neighbour algorithm in combination with data of all acceleration sensors performed best with an overall accuracy of 97 %. Similar to [Lin 17], the authors concluded that multiple data sources and sensor fusion increases the classification accuracy.

## Stroke and Motion Analysis in Table Tennis

However, in table tennis, there were only a few research papers available, which addressed stroke and motion analysis.

First studies from [Boot 90] concerned the timing of an attacking forehand drive on analog films. Kinematic parameters including the displacement, position, acceleration and velocity of the racket and the players' arm were frame-wise determined. These parameters were quantitatively analyzed in terms of timing during the impact event. Based on the comparison of five elite table tennis players, the authors found that

a higher temporal accuracy during ball-racket contacts relied not on consistent movements of the attacking forehand stroke.

Another motion analysis application in table tennis was presented in [Chen 06]. Here, trajectories of the ball and the players as well as the position of the table within standard broadcasting videos were tracked by means of a Change Decision Mask and a Bayesian Decision Framework. Dependent on the cameras' position and perspective of television videos from Olympic and Worldcup finals, the ball could be tracked with a precision of 65 % to 80 % and a recall of 91 % to 100 %. The results of the detected ball trajectory was finally used to highlight events within table tennis games.

In [Oldh 15], an event detection and classification system for table tennis was developed. Different computer vision techniques based on data of a low-cost single camera were investigated. Thereby, changes in ball motion and its trajectory were matched to events like table bounces, returns, nets, lets, serves, contacts to table edges or standrad rallies. Best results were achieved for table bounces and nets yielding classification rates of 100 %, whereas contacts of table edges were worse detected with an accuracy of 67 %. Overall, an event classification rate of 96 % was achieved. However, the small sample size of rare events like ball contacts with table edges and lets as well as ball occlusions by the players themselves decreased the results.

Motion capturing systems were also utilized to track the table tennis racket during hitting events with impact characteristics [Rusd 05]. Six optical markers were attached on the outer edge of a racket, so that racket and arm movement could be analyzed by the motion capturing system. Thereby, the authors developed a physical model, which described the racket motion and predicted the flying ball trajectory. Additionally, they analyzed the latency and prediction capability during high speed interaction scenarios in table tennis. These models and predicted motions were subsequently used as an input for a virtual reality gaming application.

In [Sore 01], a combination of video cameras and acceleration sensors was proposed to analyze discrete movements in table tennis. Thereby, two cameras around the table and one accelerometer attached at the players' forearm acquired data in an experimental setup with four subjects. Based on the acceleration data, forehand and backhand strokes were identified and categorized by velocity estimates. This meta data were then correlated with the three-dimensional positions of the tracked

ball and the players themselves. They found that the approaching ball relative to the players' position does not necessarily has influence on the decision for the performed return stroke.

Wearables in table tennis were also used in [Guo 10] for a qualitative analysis of blocks. The measurements in this study were based on a wrist-worn acceleration sensor, which captured data from forehand table tennis blocks from subjects with different skill levels. Based on the acceleration signals, a peak detection algorithm was performed to determine the beginning and the end of consecutive block stroke movements. Additionally, qualitative comparisons of different elite and amateur players were presented. The authors concluded that higher deviations during multiple strokes indicate a lower skill level than less variations performed by subjects with higher playing abilities.

An instrumented table tennis racket as Wearable was presented in [Boye 13]. An inertial sensor in combination with a microphone was integrated into a milled cavity of a racket's handle. Three-dimensional accelerations, angular velocities and impact sound signals for different stroke types comprising topspins, drives and pushes of two subjects were averaged, overlayed and qualitatively prepared for a visual feedback. The authors maintained that the presented visualization could be used to indicate the reliability of stroke motions and playing abilities of different athletes, but an automatic detection and classification of events or strokes are still missing.

A ball hit detection approach for table tennis solely based on audio analysis was developed by [Zhan 06]. In their contribution, energy peaks and coefficients based on the mel-frequency cepstrum were implemented to process the audio data of a standard table tennis match. This method yielded a precision of 91 % and a recall of 73 % within a standard 30 minutes table tennis game. Additionally, the authors proposed that the developed algorithms are also robust against influences from environmental noise, speech and cheering sounds around the actual gameplay.

## Summary

Various stroke and motion analyses for racket sports in general were presented in this section, exemplary in [Duan 05] and [Anan 17]. Among contributions for tennis and badminton, publications in table tennis were introduced. Most methods were based on information of video signals or motion capturing systems. Thereby, research focused on the

arm and racket movements during strokes impacting the gameplay and behavior of players in [Boot 90], as well as on computation of trajectories for balls and players in [Chen 06] to highlight television videos or to index table tennis matches in [Oldh 15] and [Rusd 05].

Only the authors in [Sore 01], [Guo 10], [Boye 13] and [Zhan 06] dealt with the application of wearable sensors in table tennis research. So far, no publication presented an automatic detection and classification approach of strokes. Potentially, existing approaches developed for the sports tennis in [Conn 11b], [O Co 10] and [Wang 16] as well as for badminton in [Lin 17] could be adapted to a table tennis scenario, but most methods were limited due to the lack of suitable data or applicability.

Nevertheless, accuracies of  $> 90\%$  for stroke detection and classification in tennis [Conn 11b] and badminton [Lin 17] using one single sensor indicate the basic possibility for similar approaches in table tennis. This is also confirmed by qualitative experiments in [Guo 10] and [Boye 13]. According to [Wang 16], the Wearable should thereby be placed as close as possible to the racket. Hence, a methodology that detects and classifies stroke types particularly in table tennis using Wearables is presented in this thesis.

### 1.2.2 Ball Speed and Spin Estimation

Research in the field of speed and spin estimation of balls in sports, their bouncing behaviors and the physical description of their trajectories are almost entirely based on the analysis of data acquired with high-speed cameras and mostly investigated for each particular ball type within the small cameras field of view. This constraint induced often only two-dimensional considerations for speed and spin estimation.

#### Ball Speed and Spin Physics

A first attempt of ball speed and spin measurement was made by a group in [Brig 45]. They introduced four approaches based on the physics of ballistics for projectile investigation and on rebounds from massive flat plates with variable inclination angles. As reference for algorithm validation, the spark photography methodology was used. During experiments with baseballs and golf balls, the authors observed influences of the temperature and the restitution coefficient for an accurate ball speed and spin estimation. Therefore, unspecified correction factors were applied on their physical models.

Measurements of the rebound ball spin and speed, influences of inclination angles during impacts and the determination of restitution coefficients for superballs and tennis balls were presented in [Cros 02b]. The authors developed several theoretical equations, which were validated by a high-speed video camera reference. It was shown that ball speed and spin were transformed into each other based on the restitution coefficient and the initial bouncing angle.

More sport specific bounce analysis of spinning balls were presented in [Cros 05] for tennis. Thereby, impacts of balls with a pre-defined rotation and known speed on different types of surfaces, here string beds of rackets and rigid wooden surfaces were investigated for different inclination angles. They found that the rebound angle and spin of the ball depended in a nontrivial manner on the restitution coefficient and friction parameters as well as on elastic properties of the ball. They also stated that this rebound behavior could cause control problems of spinning balls for athletes in all racket sports.

The spinning behavior of balls was also investigated in [Cros 02a] with respect to the acting forces. They conducted studies with tennis balls, golf balls, superballs, baseballs and basketballs on different types of surfaces. Normal reaction forces and friction forces were measured with movable piezo force plates to determine the prevalent friction case, which was rolling, sliding or static. It was found that the friction case is dependent on the inbound angle and that the spin of the outbound ball increases if friction was involved during the impact event.

In the study of [Garw 69], deforming effects due to elastic impacts, conservation of kinetic energy and material properties of balls were considered. Thereby, kinematics and trajectories of an ultra-elastic superball was theoretically computed after each bounce and compared to the behavior of ordinary rigid balls. During these experiments, it was found that more elastic balls conserve more energy. In turn, this implicates the change from an expected bouncing behavior to a more complex and chaotic bouncing behavior.

Besides speed and spin properties of a bouncing ball, in [Cros 99a], dynamic processes within the ball itself were analyzed. Experiments using different types of tennis balls and impacts on piezo-electric force plates were conducted to draw conclusions about the stiffness, impact duration and ball compression during the impact scenario. Results were qualitatively presented by deformation figures and force distributions.

## Ball Speed and Spin Estimation in Racket and Bat-and-Ball Sports

In literature, rebounds and impacts in tennis are often investigated with the objective to figure out influences on the spin generation caused by the string bed and material properties of the racket itself.

In [Cott 02], experiments were described, which confirmed the validity of their developed mathematical models for spin creation and forwarding in tennis. These models involved information about ball deformation as well as ball and racket properties. An evaluation based on high-speed video cameras revealed novel information about the ball-racket contact time, footprint size, deformed ball shape and the movement of the racket strings.

Considerations about the interaction of spin generation and string bed pattern in tennis were also presented in [Nico 13]. Measurements were based on topspin similar projections from a ball machine, which fired balls onto rackets with different numbers and patterns of strings. Similar to [Cott 02], high-speed cameras were used as validation reference. Thereby, the authors found a relation of the outbound velocity and the string pattern as well as a correlation between the outbound angle and spin with the number of strings.

Further research about impacts from forehand topspins in tennis was presented by a study of [Good 04]. Special features and markings applied on the tennis ball allowed more detailed conclusions about the inbound and outbound speed and spin of the ball shortly before and after impact. Experiments with a bowling machine, which varied speed, spin and incident angle of the ball revealed that the rebound spin was not a function of the racket material, tension or pattern of the strings, but almost exclusively dependent on the friction cases during impact. This results also substantiated the outcomes of experiments in [Cros 05] and [Cros 02a].

Spinning effects of a baseball were described in [Nath 08]. They deployed an equation system based on acting forces during the flight and computed the ball trajectory. Their experiments to determine the unknown lift coefficient was evaluated by means of a pitching machine, which generated a variety of initial ball speeds. Then, the resulting trajectory was observed with high-speed motion tracking system. Based on their algorithms, the lift coefficient could be estimated with an accuracy of 90 % dependent on the overall ball velocity.

Angular impacts and spin rates were also analyzed for racquetball in [Illo 14]. Balls with constant speeds were dropped onto oblique plates

and the resulting spin rates were captured by a high-speed video camera. Thereby, a proportional relationship between the impact angle and the spin rate was found, which indicated the prevalent type of friction.

In contrast to yet exclusive mentioned video-based approaches, an instrumented cricket ball with integrated inertial sensors was developed in [Fuss 12] to determine the three-dimensional spin rate during the ball flight. Thereby, acceleration and angular velocity data were measured. In the end, all data were visually prepared. In their study, only a qualitative spin rate analysis was conducted, no reference system considered and no implications drawn.

### **Ball Speed and Spin Estimation in Table Tennis**

Ball speed and spin estimation in table tennis was often addressed together with models of the ball trajectory in applications with table tennis robots.

In [Chen 10], a trajectory prediction method was proposed, which involved ball tracking with four video cameras. In addition, a dynamic non-linear model of the flying ball physics and a bouncing model between ball and table was developed. All necessary parameters and coefficients of the proposed model were iteratively calculated during the trajectory prediction. Based on the video reference, the implemented system resulted in mean absolute prediction errors of  $\sim 8$  mm,  $\sim 5$  mm and  $\sim 6$  mm for the ball position within the directions of the table length, width and height, respectively. The authors stated that their model is analytically solvable and could also be applied for bouncing problems between the ball and the racket.

Similar considerations for a ball trajectory prediction were proposed in [Huan 11]. Thereby, the flying ball was predicted using a polynomial fit based on incremental values for ball speed and spin estimates. An evaluation of the physical model was realized with a stereo video camera system. The authors achieved comparable prediction errors of  $\sim 5$  mm,  $\sim 11$  mm and  $\sim 26$  mm for the ball position within directions of the table length, width and height, respectively. In addition to [Chen 10], the authors found that an accurate prediction of the ball spin is crucial for an acceptable estimation of bouncing heights due to the influence of the restitution coefficient.

A further approach predicting the ball trajectory was presented in [Zhan 10]. Thereby, the authors focused not only on the landing position of the ball, but also on the striking position of the ball. Again, the

evaluation of the prediction model was based on a stereo video camera system. For the ball landing position and the ball striking position, overall mean prediction errors of  $\sim 7$  mm and  $\sim 13$  mm were achieved, respectively. The advantage of this system lay in a parallel processing implementation of both camera streams, which enabled a fast prediction times.

In addition to sole trajectory modeling, in [Nono 10], a ball speed and spin analysis incorporating more aerodynamic details was presented. Effects of acting aerodynamic forces resulting in ball lift and drag phenomena were investigated and adapted to existing physical models to enhance the final prediction accuracy. Therewith, averaged errors for forward spinning balls and backward spinning balls of  $\sim 9$  mm and  $\sim 4$  mm could be achieved, respectively. The authors showed that the omission of aerodynamic effects has noticeable influences on the trajectory prediction.

In [Naka 09] and [Naka 10], the physics of the flying ball were extended to spinning rebounds between balls and rackets with considerations about the prevalent friction case. Trajectory data were acquired from table tennis robots and ball catapults. Results were validated by a stereo video camera system. Both papers modeled the ball-racket rebound with an energy conserving spring-damper system between the rigid ball and the elastic rubber. Different kinds of spins involving topspin, backspin and side spin were applied in their experiments. The authors presented qualitative results for the estimation of ball spin before and after the impact and claimed that prediction errors of their algorithms stayed under 10 % for all types of spin.

Influences on the bouncing behavior of the ball caused by different table materials were investigated in [Kami 13]. In this study, the friction and restitution coefficients were measured during multiple ball-table bounces. Thereby, different inbound and outbound spin values of labeled balls were recorded with a high-speed camera. The authors found that table surfaces from various manufacturers have significant differences in the friction behavior and thus in the spin forwarding, which also has effects on the players' performance.

The physics of juggling a table tennis ball in combination with different rubber types were described in [Wide 16]. Required racket angles related to the ball speed and spin were analyzed, which allowed to a vertical movement of the table tennis ball. Thereby, a physical model based on the acting forces and momentums as well as considerations about

the friction coefficients of different rubbers was implemented. All experiments were validated with data obtained by a high-speed camera. Qualitative results about prevalent forces and friction coefficients were presented for different kinds of rubber materials.

Sole estimations for ball speed, ball spin or a combination of both were either based on experiments with single camera setups assuming an orthogonal motion projection of the ball or on multi-camera setups for a three-dimensional rotation prediction. In [Szep 11], markings on the table tennis ball's surface were tracked over consecutive frames captured by a single high-speed camera to estimate the ball spin. Thereby, markings were either painted as crosses or rectangles with conspicuous features. The authors developed an algorithm, which computed the linear and rotational displacements of the markings in each frame and derived the resulting ball spin. The approach was evaluated with synthetic simulations and real data from an experimental system containing a ball feeder and five different table tennis rackets. Results for the simulated data and for real data yielded relative errors of  $\pm 12\%$  and  $\pm 20\%$ , respectively. However, a potential dependency of the ball speed was not considered.

A three-dimensional ball spin estimation from data of images was attempted in [Tama 04]. Their proposed method estimated the shape transformation of the table tennis ball using depth and intensity information within consecutive images. With this transformation, a three-dimensional ball model was registered on the image data. The authors investigated data from an experimental bouncing setup and data of real table tennis rallies. The qualitative outcomes were used to characterize the spin of elite and amateur players. Admittedly, a validation of the results was not presented and a ball speed estimation not included.

Another ball spin estimation approach in table tennis was illustrated by [Tama 12] using an Inverse Compositional Image Alignment methodology. Based on an orthogonal ball projection, the ball motion was predicted based on parameters including the ball shape and depth as well as on warping characteristics. Similar to [Szep 11], simulated data and real game rallies were investigated. Results were presented with qualitative figures without an evaluation of the computed ball spin estimates. Furthermore, the authors stated that their algorithm could be used for a large variety of ball sports to estimate the ball spin.

Image processing and registration methods were also used in [Liu 11a]. Based on the intensity estimation of pixels and the k-Nearest Neighbor

algorithm, the three-dimensional rotation of a table tennis ball was estimated. As experiments, two high-speed cameras acquired data from a ball catapult generating different ball speed and spin values. Additionally, the authors validated their system with a special calibration platform for rotational ball movements. As many image processing techniques, the proposed work was also based on conspicuous markings on the ball surface. Ball rotations could be estimated with overall errors less than 0.5 rad/s.

A ball spin tracking approach using the factory printed logo on the ball surface was introduced by [Glov 14]. A special Quaternion Bingham Filter and an Extended Kalman Filter were developed to track the ball spin during its flight curve. Data from simulations and high-speed videos of real table tennis matches were analyzed. For different spin types like topspin, backspin and side spin, an overall mean error of 0.2 rad/s was achieved. However, these good results could only be realized, if the factory printed logo was visible during the complete image data acquisition.

Similar experiments were made by [Zhan 14] and [Zhan 15]. Thereby, the natural brand of the table tennis ball was detected as well as ball speed and spin predicted over the whole ball trajectory. All data were obtained by three high-speed cameras mounted at different spots above the table. Their algorithms were based on the Extended Kalman Filter and specific confidence criteria of the extracted brand logo image segment. Both approaches were calibrated by a special spinning platform for the evaluation of different rotational spin values as it was made by [Liu 11a]. The ball position within the table plane could be predicted with an error of  $\pm 18$  mm and for the height with an error of  $\pm 19$  mm. Spin values were estimated with an average error of 0.4 rad/s. The authors found that an accurate ball spin estimation is essential for a meaningful prediction of the ball trajectory. In addition, they discovered that an increase of ball speed leads to higher ball spin estimation errors. This fact was also confirmed by observations of [Nono 10], [Naka 09] and [Naka 10].

## Summary

Ball speed and spin estimation in racket and bat-and-ball sports is a main research topic. Thereby, all methods are based on image data of high-speed video cameras, except in [Fuss 12], where the spin of a cricket ball was measured with inertial sensors. However, most contributions considered physical descriptions about the relation between ball speed and spin in combination with their mutual transformation, exemplary

in [Cros 02b], [Cros 02a] and [Good 04] as well as the influence of restitution in [Brig 45] and friction in [Cros 05] and [Illo 14] during bouncing effects.

In table tennis, ball speed and spin estimation is often combined with ball trajectory predictions for application of table tennis robots, which is illustrated in [Chen 10], [Huan 11], [Zhan 10] and [Nono 10]. Additionally, [Naka 09] and [Naka 10] developed bouncing models for ball-racket impacts. However, no research group involved inertial sensors or any other Wearables for the estimation of ball speed and spin. One noteworthy attempt was made by [Kian 09], who estimated the ball shuttle speed in badminton using kinematic information of the racket.

Based on assumptions about characteristics of the approaching ball, the computed racket kinematics and the ball-racket rebound models in [Naka 09] and [Naka 10], a method for ball speed and spin estimation in table tennis using a single inertial sensor is introduced in this thesis.

### 1.2.3 Investigations of Impacts on Rackets

In literature, investigations of racket impacts were poorly addressed. Most contributions were employed with influences of racket characteristics on the outbound ball or analyzed material properties of the racket.

#### General Investigations of Impacts on Rackets

Impacts with sports balls and striking characters on rackets and bats were examined in [Cros 14]. A simplified physical model assuming a collision model of point masses was developed to describe the impact process with a rigid body approximation. Experiments were made with baseball bats and tennis rackets. Thereby, impact forces, energy losses and restitution coefficients were investigated for different racket types and impact angles. The authors found that rackets perform differently when they hit the ball at different positions. This behavior was described by a bouncing factor.

Similar considerations for baseball and tennis are made by [Cros 99b]. In contrast to the simplified model developed in [Cros 14], the authors extended the rigid body approximation of ball impact events with a model of flexible and deformable racket behavior. They found that this model extension covered a realistic vibration propagation which enabled the identification of sweet spot positions and dead spot positions of bats and rackets.

Sweet spots were particularly analyzed for baseball bats in [Cros 98a]. Two piezo-electric sensors were applied on the bat's surface and the appearing vibrations from different impact positions were measured. Based on a free bat model, the vibration propagation of an impact impulse was determined and implications about the handle velocity as well as forces exerted to the hand were drawn. In addition, it was found that some nodes of the bats' vibration waves are equal to sweat spot positions and that hits at this positions maximize the outbound energy of the balls.

Same knowledge was confirmed in sweet spot analysis for baseball bats by [Brod 86]. There, an acoustic sensors instead of piezo-electric sensors were applied to identify the vibration behavior. The authors developed a vibration model for various types of baseball bats, which can compute the optimal position where the ball should be hit to maximize its outbound power and velocity.

Further investigations and localizations of sweet spots and their opposites, dead spots, were presented in [Cros 98b] and [Cros 97] for tennis rackets, in [Mucc 13] for beach tennis rackets and in [Haak 96] for cricket bats. In all publications, piezo-electric vibration sensors or accelerometers were applied onto the rackets' and bats' surfaces. Common for all methods, data were collected and equations formulated to localize different regions for the optimal hit position where potential vibrations are lowest (sweet spot) or in the worst case, where the energy loss is highest (dead spot).

### **Investigations of Impacts on Rackets in Table Tennis**

In table tennis, ball impacts on rackets were processed to obtain insights into the vibration behavior of the wooden blades in [Mani 12a] and [Mani 12b]. Acoustical and vibrational analyses were performed for different blades made out of plywood and various composites of wooden layers. Both publications addressed the correlation of the acoustic frequency spectrum with performance adjectives of rackets like fastness, stiffness and controllability. Results were presented in qualitative descriptions of the vibration distributions for different wooden materials of rackets.

Physical properties comprising impacting forces, ball-racket contact times, deformations and rebound energies were quantitatively investigated by [Kawa 03]. They experimented with pendulum-like suspended rackets of different wooden and rubber materials. A unified model was

developed, which can compute the mentioned properties based on information of the inbound ball speed and impact position as well as on the used materials. Comparable to investigations in tennis and baseball by [Cros 99b], [Cros 98a] and [Brod 86], the authors found the existence of sweet spots and dead spots for table tennis rackets as well. Their results could be used to develop blades and rubbers for individual conceptions and playing characteristics.

In addition to material analyses of table tennis rackets, an approach for ball impact localization was presented in the work by [Chen 12]. A two-dimensional laser array was built-up in terms of a light barrier matrix. This matrix array contained 28 laser diode and photo diode pairs on a sensor board of 30 cm width and length. Impact positions were detected, if a ball interrupts the specific transmitter-receiver combination of the installed laser array. With this apparatus, the authors achieved a localization resolution of  $\sim 21$  mm. They claimed that their solution is easy to use and more accurate than methods with high-speed cameras, huge amounts of data and complex computer vision algorithms.

Another ball impact localization approach using an instrumented racket was introduced in [Yama 18]. Thereby, an ultrathin piezo-electric strain sensor with 25 sensors in a  $5 \times 5$  matrix was integrated between the blade and the rubber of a table tennis racket. Data were acquired from a study with two players performing chop and smash strokes. This approach was able to detect impact events and allowed the classification of both types based on the strain data distributions. Additionally, the authors stated that a rough estimation of the initial impact position is possible.

In [Wege 09], another contribution of ball impact localization on table tennis rackets using a large-scale piezo-electric sensor applied between blade and rubber was presented. Their setup collected data of the mechanical excitation during the ball-racket impact on a matrix of  $7 \times 7$  measuring points. A special connectivity of single sensors within the sensor array allowed conclusions about the impact position and the impacting forces. Admittedly, the results were only qualitatively presented indicating the piezo-electric activity of all points during an impact.

## Summary

Investigations of ball impacts on rackets and bats in sports mostly addressed material parameters and vibrational analyses. Thereby, the main emphasis lies on the identification of sweet spots and resulting

conclusions about to the rackets' performances and materials, which was exemplary made for tennis rackets in [Cros 99b], [Cros 98b] and [Cros 97] as well as for baseball bats in [Cros 98a] and [Brod 86].

Similar work was considered in table tennis by [Mani 12a], [Mani 12b] and [Kawa 03]. In addition to material analyses, some authors attempted to estimate the ball impact position with external devices in [Chen 12] and with instrumented rackets as Wearables in [Yama 18] and [Wege 09].

However, major modifications of rackets were necessary, which affect their desired properties. Nevertheless, piezo-electric sensors appear to be suitable for ball impact localization approaches, as it was proven in [Wege 09]. Therefore, a novel method involving an unobtrusively instrumented racket is presented in this thesis.

### 1.3 Commercial Products

In addition to scientific research, some commercial products for training support of athletes were developed for the disciplines tennis and badminton.

Small sensor devices for application in tennis can be attached to the handle's end of suitable tennis rackets. For example, sensors are available from *Sony*<sup>5</sup> or *Zepp*<sup>6</sup>. In addition, sensors worn on the wrist are also realized, especially products by *Babolat*<sup>7</sup>. Examples of the sensors can be seen in Figure 1.3 (a) to (c). Depending on the product design of the manufacturer, accelerometers and gyroscopes as well as vibration sensors are implemented. These devices mainly measure the power of each stroke, roughly localize the ball impact position and collect statistical information about the training and the match. Thereby, these systems can differentiate between forehand and backhand strokes, as well as between topspins and slices. Some systems can also compute the swing speed and the resulting spin of the ball. In all cases, recorded data were transmitted to a smartphone application, which visualizes data, saves statistics and keeps track of the gameplay and the training progress.

Among others *Babolat* and *Coollang*<sup>8</sup> develop smart sensors for badminton, see Figure 1.3 (d). Analogously to the attachment of sensors on tennis rackets, devices in badminton can also be mounted at the handle's

---

<sup>5</sup><https://www.sony.de/electronics/smart-geraete/sse-tn1w> – last access 11/2018

<sup>6</sup><http://www.zepp.com/en-us/tennis/court-checkin> – last access 11/2018

<sup>7</sup><https://de.babolatplay.com/pop> – last access 11/2018

<sup>8</sup><https://www.coollang-asia.com/> – last access 11/2018



**Figure 1.3:** Examples of racket-mounted Wearables in tennis used for training support of athletes from Sony are illustrated in figure (a) and from Zepp in figure (b). A wrist-worn device for tennis analysis from Babolat can be seen in figure (c). Additionally, one sensor device used in badminton from Coollang is shown in figure (d). It is also mounted at the end of the rackets handle.

end of specific rackets. Using data from acceleration and gyroscope sensors, these smart devices can analyze various stroke types, swing speeds, impact times and durations as well as classify the activities of the player's training session.

Until now, specialized Wearables for table tennis analyses are unavailable on the consumer market. This include both instrumented table tennis rackets and wearable devices worn on the player's body. Therefore, this thesis presents the development of Wearables for sports science research in general as well as specialized sensor hardware for a full instrumentation of a table tennis racket.

## 1.4 Contributions

The analysis of scientific literature as well as the commercial market revealed that a sensor-based, real-time feedback device is still missing in table tennis. Hence, this work contributes to sensor-based digitalization in this sports, which is still at its very beginning. The proposed methods aim at closing these gaps.

Despite several experiments detecting strokes and their qualitative visualization with accelerometers and acoustic sensors, a comprehensive evaluation as well as an automatic detection and classification of multiple stroke types and their swinging movements using inertial sensors in table tennis are not available. On that account, this thesis contributes with machine learning approaches, which enable a reliable detection and classification of basic stroke types in table tennis using a single wearable sensor device. Furthermore, no methodologies applying Wearables addressed either the speed or the spin of the ball and the

racket during impact events. Therefore, important quantitative information about the maximum racket acceleration and rotation, as well as correct timing throughout the stroke so far remain unknown outside the laboratory environment. Thus, this thesis introduces a ball speed and spin estimation method based on a single racket-mounted inertial sensor. Apart from that, several investigations about impacts on table tennis rackets were performed using large-scale sensors within an instrumented racket. They indeed provide more details about the ball's impact location and the rebound mechanism, but suffer from extensive processing components, which makes the racket's handling unsuitable and affects the desired playing characteristics. Therefore, a novel ball impact localization approach involving as little as possible modifications of the racket itself resulting in an unobtrusive prototype is presented. In addition to sole data analyses, this thesis includes the development of an instrumented racket prototype as a Wearable for data acquisition along with a feedback visualization application on a mobile device in table tennis, since ready-to-use hardware equipment is until now unavailable.

Summarized, all contributions led to a ready-to-use *Smart Racket* system, which is shown in Figure 1.4. This system allows professional and

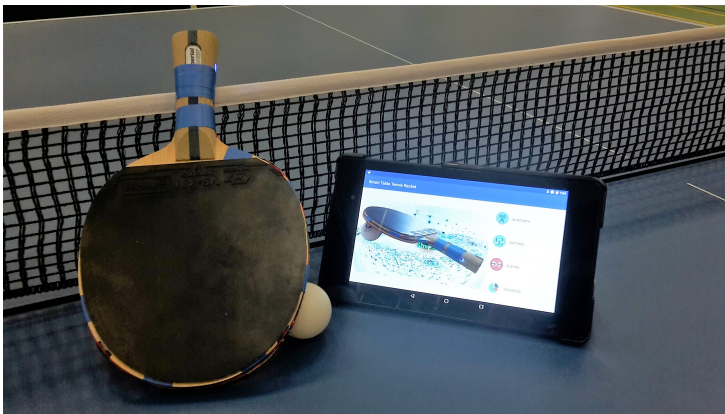


**Figure 1.4:** Smart Racket system including the instrumented table tennis racket and the feedback application on a mobile device. Data were acquired and processed by the embedded microcontroller inside the racket's handle, whereas data preparation and visualization for feedback presentation is done by the mobile application.

amateur players to analyze their playing abilities and provides feedback during training and exercises.

The following overview lists the four key contributions of this thesis, that taken together lead towards a sensor-based, real-time feedback device in table tennis:

- (I) Existing sensor-based investigations of motion analyses in table tennis are limited to qualitative evaluation and visualization of the measured data. As opposed to that this contribution presents an automatic stroke detection and classification system. Thereby, a racket prototype was instrumented with a specially developed inertial sensor, see [Blan 14a] and [Blan 14b], to obtain kinematic data from players with different skill levels and playing abilities ranging from amateur to professional athletes during exercise and gameplay scenarios. The acquired data were subsequently used as input for the evaluation of different machine learning approaches to reliably detect and classify multiple basic table tennis strokes. In Chapter 5, detailed descriptions of the algorithms are presented. Parts of these algorithms have already been published in [Blan 15].
- (II) In contrast to ball speed and ball spin evaluation in table tennis using video-based systems, so far, no methods incorporate Wearables. Therefore, an algorithm was developed, which describes ball speed and spin estimation shortly before and right after a ball impact. This approach was only based on data of a single inertial sensor inside the racket handle. Due to this restriction, several assumptions for the initial unknown inbound ball properties and simplifications of the racket motion had to be made. Thereby, acceleration and angular velocity were used to calculate the racket speed and rotation during impact. Then, a table tennis specific rebound model was applied to this input data estimating the ball speed and spin shortly after impact. The proposed technique can be found in Chapter 6. This contribution has already been published in [Blan 17].
- (III) Since existing approaches for ball impact localization on table tennis rackets are not applicable to a real-world scenario, a novel system was developed with the intention to involve as little components as necessary. The presented approach consisted of a minimum number of piezo-electric vibration sensors and tiny conditioning circuits. They were unobtrusively integrated into the

edge of the wooden blade and complied with official regulations of the ITTF. Similar to the localization of epicenters during earthquakes, the localization of the ball impact position was based on time-differences from the incoming wavefront of sensor pairs. Due to the nonlinearity of natural wooden material, a model was generated, which contained the effective time-differences of the wavefront on the racket's surface. With this distribution, the initial ball impact position was reconstructed. The illustrated approach and the investigated reconstruction methods are presented in Chapter 7. This contribution has already been published in [Blan16b].

- (IV) In literature and on the commercial market, so far, no Wearables and assessment tools for table tennis are available. Therefore, all results and outcomes of analyzed data and developed methods were combined resulting in a training and support system for table tennis players. On that account, an instrumented racket prototype was developed comprising all necessary sensors. It contained a powerful microcontroller, which was capable of processing all required algorithms in real-time and of transmitting the results to an external smartphone application. Finally, this application prepared all data for feedback purposes and statistics. The instrumented racket is described in Chapter 8. Parts of this contribution have already been published in [Blan18].

## 1.5 Structure of this Thesis

The thesis is divided into nine chapters. Following the motivation and overview (Chapter 1), the next three chapters introduce the relevant fundamentals of machine learning, give considerations about embedded signal processing and illustrate the development of wearable sensors.

- Chapter 2 briefly introduces the major pattern recognition fundamentals and machine learning approaches with respect to the methods, algorithms and classification systems used in this thesis.
- Chapter 3 summarizes important considerations for embedded signal processing and describes concepts, how machine learning can be realized in real-time on hardware-restricted microcontroller devices.

- Chapter 4 presents main aspects for the development of wearable sensors, the physical principles of the used sensor types and introduces the *miPod* hardware platform, which was used for most data acquisitions.

After these introductory chapters, the presented contributions are organized in additional four chapters.

- Chapter 5 investigates the detection and classification of table tennis strokes from exercise data and gameplay rallies using a racket-mounted inertial sensor.
- Chapter 6 presents an approach for the ball speed and spin estimation of table tennis balls based on the inertial data of a single racket-mounted sensor.
- Chapter 7 localizes the ball impact positions on table tennis rackets by analysis of the vibration data obtained from different piezoelectric sensors during ball-racket impacts.
- Chapter 8 shows the final hardware implementation of all electronic components into the racket prototype and describes the embedded implementation of the algorithms.

The last chapter (Chapter 9) summarizes the presented contributions, discusses the results and gives an outlook about future work and further investigations.



## Chapter 2

# Fundamentals of Machine Learning

Besides standard and commonly used signal processing methods applied on raw measured data in the time or frequency domain, also machine learning algorithms can be used to analyze data and to extract meaningful and relevant information. The following section briefly introduces machine learning fundamentals with respect to specific methods, algorithms and classification systems used for the stroke detection and classification chapter in this thesis (→ see Chapter 5).

### 2.1 Introduction

Machine Learning enables the automatic detection and classification of different patterns by a technical system. In general, a *Pattern* could be any type of entity [Wata 85]. For example, it could be a fingerprint image, a word within a speech signal, letters of a handwritten text or in the context of this thesis, a specific table tennis stroke. The classification task can in principle be done in a *Supervised* or an *Unsupervised* manner. The following chapters specifically focus on supervised classification, whereby patterns are categorized into different *Classes*  $C$  within the recognition context.

Mathematically, a pattern is represented by a set of a different *Features* combined into a feature vector  $q_V$  spanning a multi-dimensional data representation, called feature space. These features transfer the raw input data from the time domain into a higher abstraction level, in which trained classifiers try to establish virtual boundaries to separate the feature data into the single classes. This is done in an iterative learning procedure [Niem 83]. Such a system can mostly be well described by the

standard machine learning pipeline comprising a *Working Phase* and a *Training Phase* shown in Figure 2.1.

First of all, sensor data have to be acquired. Then, a preprocessing part can filter, normalize, segment or compact the raw input to achieve a uniform data representation for the intended patterns. Following, different features are extracted and computed. During the training phase, an optional feature selection process could be performed to identify most meaningful features, which have the strongest influence on the final class decision and are commonly representative for all patterns. Additionally, in this phase, the classifier needs to be trained to separate the high-dimensional data formed by the selected features  $\hat{q}_V$  to make a reasonable class decision. The backward feedback paths iteratively allow the optimization of preprocessing, feature extraction, feature selection and classification to obtain the best performance of a classifier according to its evaluation. However, this procedure is prone to *Overfitting*. Thereby, the model is trained with details and noise of the training data, which could result in a poor performance of the learned classification system for new data. To avoid this overfitting problem, for example, the input data can be split into a test set and a training set using a cross-validation approach or applying methods for dimensionality reduction

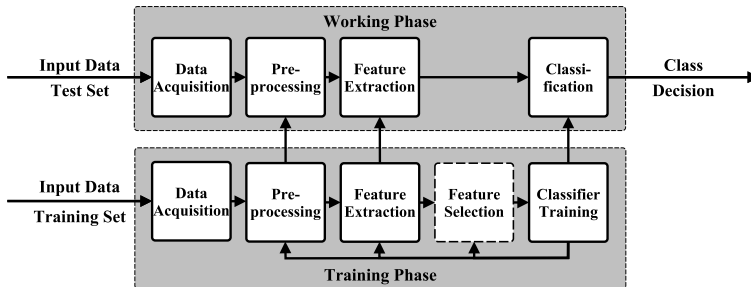


**Figure 2.1:** Machine learning pipeline adopted from [Jain 00]. It includes a working phase as well as a training phase. Raw input data are sampled from physical sensors, preprocessed and subsequently, multiple features are computed. These features are finally used as input for the classification. Additionally to the working phase, the classifier has to be trained beforehand and an optional feature selection could have identified the most meaningful features for the entire classification pipeline.

that reduce the input data to more generalized data. During the working phase, which will be finally implemented into the technical system, data are preprocessed in the same way as formerly learned, features are extracted and the trained classifier assigns the input pattern to a class with consideration to the previously computed features [Jain 00]. Overall, the presented machine learning pipeline illustrates the data processing from measured raw data input to a specific class decision output.

## 2.2 Classification Pipeline

In the following sections, the mentioned classification pipeline is described in more detail with regards to the methods applied in this thesis. These include data acquisition, the preprocessing requirements, the used feature space, the particular feature selection approach and the different types of classifiers as well as their evaluation.

### 2.2.1 Data Acquisition

Before performing the actual preprocessing step, the input data have to be obtained from physical sensors. In this context, an inertial measurement system (consisting of an accelerometer and a gyroscope) together with a geo-magnetic sensor were used for data acquisition of the stroke detection and classification part ( $\rightarrow$  see Chapter 5). Using common analog-digital-converters, these sensors internally transpose a continuous but analog physical process into a digitalized time-series data stream  $\mathcal{S}(n_s, m_s, t)$  with  $n_s$  types of different sensors,  $m_s$  data channels per sensor type and  $L$  samples for each sensor channel [Niem 83].

For example, a body-worn inertial sensor could provide this kind of a data stream, which represents the athlete's arm movement during a table tennis stroke. Thereby, all discrete time points  $t$  do not have to coincide for every sensor type  $n_s$  or channel  $m_s$ . For instance, an inertial sensor yields 2 sensors  $n_s$  (accelerometer and gyroscope) each having 3 data channels  $m_s$  resulting in three-dimensional data. In combination with the geo-magnetic sensor ( $n_s$  with 1 and  $m_s$  with 3), the amount of samples per sensor channel can be different, if the sampling interval  $T_A$  varies between these three sensors. An example configuration for all input sensors can be seen in Figure 2.2. Here, sensor  $n_1$  with 2 channels  $m_1$  and sensor  $n_2$  with 4 channels  $m_2$  are captured at different points in time  $t$  resulting in a variable amount  $L$  of samples  $x_i$  for each sensor.

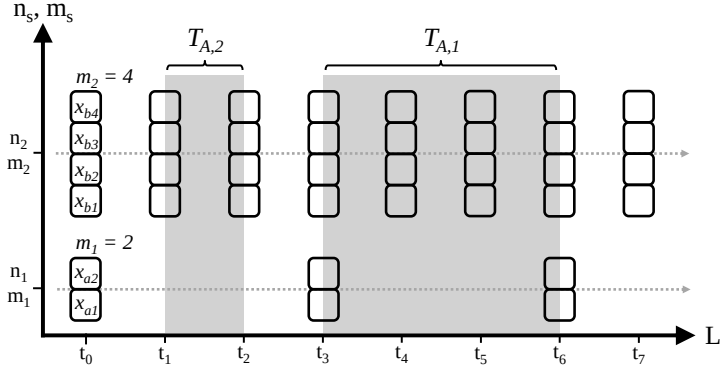


**Figure 2.2:** Example sensor input configuration. Two sensors  $n_1$  and  $n_2$  are captured with variable sampling intervals and individual sensor channels  $m_1$  with 2 and  $m_2$  with 4, respectively. The amount of samples  $L$  is different for each sensor, since sensor  $n_1$  is sampled three times more often than sensor  $n_2$ . The actual data are stored by the samples  $x_a$  and  $x_b$  for the corresponding sensor type. Both sampling intervals  $T_{A,1}$  and  $T_{A,2}$  are shown as gray boxes.

### 2.2.2 Preprocessing

Taking this time-series data  $\mathcal{S}(n_s, m_s, t)$ , the preprocessing step can add filtering to reduce noise by low-pass filters or emphasize strong changes during impact events with high-pass filters. Furthermore, outliers can also be removed from the data set by additional filtering [Bruc01]. Practically, most classification approaches cannot be applied directly on a continuous data stream. In this thesis, the time-series data were split into single windows  $w_i = [(t_i - \frac{w_L}{2 \cdot T_A}), \dots, (t_i + \frac{w_L}{2 \cdot T_A})]$  with a constant window length  $w_L$ . This window was then moved over the entire amount of samples  $L$  from  $(0 + \frac{w_L}{2 \cdot T_A}) \leq i < (L - \frac{w_L}{2 \cdot T_A})$  in steps of  $(w_L - w_O)$  with or without an also constant window overlap  $w_O$  of  $0 \leq w_O < (w_L - 1)$ . A window overlap becomes necessary, if the pattern of interest is split between two consecutive windows. For example, a window overlap could be advisable to avoid window transitions within the countermovement and stroke part of a forehand topspin in table tennis. The window length  $w_L$  and the window overlap  $w_O$  are therefore design parameters of the classification system.

Besides the moving window approach, different other techniques for data segmentation can be applied. Two relevant approaches are peak

detection and event recognition algorithms. Here, directly applied signal processing methods combined with suitable thresholds can identify a prominent peak or event at time  $t_{event}$  within the time-series data. Knowing this position, only one single window  $w_i$  applied around this point in time  $t_{event}$  is needed. For this thesis, peak detection was used for stroke detection ( $\rightarrow$  see Chapter 5.2) and ball impact localization ( $\rightarrow$  see Chapter 7.3).

### 2.2.3 Feature Extraction

The feature extraction step transforms the data of every created window  $w_i$  from the time-series stream  $\mathcal{S}(n_s, m_s, t)$  into the feature vector  $q_V$ . This is necessary to reduce the amount of raw input data into a more manageable data format without loss of important information [Alpa 14]. Additionally, this dimensionality reduction is more computationally efficient than a sample-wise classification based on every sample  $x_i$ . There are different ways of choosing the correct features out of numerous possible features. However, all computed features require incorporate discriminative and non-interdependent characteristics for the classification task.

In practice, finding appropriate features for a specific task can often be difficult and depends on the application context [Guyo 06]. The methods in this thesis use generic time-domain features, comprising statistical moments, signal characteristics and correlation features. All these features were successfully applied in cases of event detection [Avci 10] and activity recognition [Leut 13].

An overview about all used features including their individual mathematical computations can be seen in Table 2.1. Normally, all features are computed from all data samples  $x_i$  with  $0 \leq i < w_L$  within the window  $w_i$  of one sensor type  $n_s$  per sensor channel  $m_s$ . Features can be differentiated into single-channel features  $q_{V,SC}$  and multi-channel features  $q_{V,MC}$ . For instance, a multi-channel feature could be the correlation between the gyroscope X-direction and gyroscope Y-direction.

In this thesis, multi-channel features were only computed for the same sensor type  $n_s$  but not for different sensor channels  $m_s$ . In the end, all features are combined into the single feature vector  $q_V$  with  $Q$  representing the total amount of features. Taking into account all listed features from Table 2.1 obtained from all sensors and channels (accelerometer, gyroscope and geo-magnetic sensor with  $n_s$  and  $m_s$  of 3, the total number of all single-channel features  $Q_{SC}$  with 9 and the total number of all

multi-channel features  $Q_{MC}$  with 9), a maximum of 90 feature values ( $Q = n_s \cdot m_s \cdot Q_{SC} + Q_{MC}$ ) could theoretically be calculated.

**Table 2.1:** List of all used generic time-domain features. Regarding the equations,  $x_i$  is the corresponding sensor value of the sensor channels  $m_s$  within the current segmented window  $w_i$  over the total window length  $w_L$ .

| Feature                                                | Computation                                                                                                                                                                                                   |
|--------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Mean                                                   | $q_\mu = \frac{1}{w_L} \sum_{i=0}^{(w_L-1)} x_i$                                                                                                                                                              |
| Standard Deviation                                     | $q_\sigma = \sqrt{\frac{1}{(w_L-1)} \sum_{i=0}^{(w_L-1)} (x_i - q_\mu)^2}$                                                                                                                                    |
| Skewness                                               | $q_v = \frac{1}{w_L} \sum_{i=0}^{(w_L-1)} \left( \frac{x_i - q_\mu}{q_\sigma} \right)^3$                                                                                                                      |
| Kurtosis                                               | $q_w = \frac{1}{w_L} \sum_{i=0}^{(w_L-1)} \left( \frac{x_i - q_\mu}{q_\sigma} \right)^4$                                                                                                                      |
| Minimum                                                | $q_{min} = \min\{x_0, \dots, x_{(w_L-1)}\}$                                                                                                                                                                   |
| Maximum                                                | $q_{max} = \max\{x_0, \dots, x_{(w_L-1)}\}$                                                                                                                                                                   |
| Energy                                                 | $q_{egy} = \sum_{i=0}^{(w_L-1)} (x_i)^2$                                                                                                                                                                      |
| Median                                                 | $q_{median} = \begin{cases} x_{\frac{w_L}{2}}, & \text{if } w_L \in \mathbb{O} \\ \frac{1}{2} \cdot (x_{\frac{w_L-1}{2}} + x_{\frac{w_L}{2}}), & \text{if } w_L \in \mathbb{E} \end{cases}$                   |
| Interquartile Range                                    | $q_{iqr} = x_{\lfloor \frac{1}{4} \cdot w_L \rfloor} - x_{\lfloor \frac{3}{4} \cdot w_L \rfloor}$                                                                                                             |
| Correlation in<br>$ab \in \{XY, XZ, YZ\}$<br>Direction | $q_{corr, ab} = \frac{\sum_{i=0}^{(w_L-1)} (x_{a,i} - q_{\mu,a}) \cdot (x_{b,i} - q_{\mu,b})}{\sqrt{\sum_{i=0}^{(w_L-1)} (x_{a,i} - q_{\mu,a})^2} \cdot \sqrt{\sum_{i=0}^{(w_L-1)} (x_{b,i} - q_{\mu,b})^2}}$ |

### 2.2.4 Feature Selection

Since the feature vector  $q_V$  has to be generated for every single segmented window  $w_i$ , the computational efforts can grow rapidly for data acquired with short sampling intervals  $T_A$ . Apart from the high computational effort, other disadvantages for using every feature are that not all of them might be useful and therefore have only little contributions to the class decision. Furthermore, some features could even decrease the final classification accuracy [Theo 09]. To overcome these drawbacks, a feature selection step can be performed to reduce the amount of features resulting in faster training times. It also reduces the risk of the overfitting problem. Numerous techniques comprising wrapper approaches, embedded approaches and ranking approaches are described in literature [Guyo 03]. With these selection methods, the computational efforts and computing times are shifted from the working phase into the training phase resulting in longer learning periods.

In this thesis, a ranking approach using the relative importance  $\Psi_{RI}$  of a Random Forest (RF) classifier as ensemble-based decision tree classification was used [Holz 94]. Thereby, features, which appear in lower depths of the decision trees, contribute with higher weights to the final class prediction as features in higher depths of the decision trees. These weights correspond to the prediction errors of the Random Forest. By averaging all weights of a single permuted feature over a set of randomized trees, the variance of this feature weight estimation can be minimized and used as a feature selection method. Taking these weights, the relative importance  $\Psi_{RI}$  of every feature is sorted in a list. Then, all features, whose relative importances are above a certain threshold  $\Psi_{RI} > \theta_{\Psi_{RI}}$  were restructured into the feature vector  $\hat{q}_V$  and used as reduced input for the last classification step. As threshold, the statistical three-sigma rule for an outlier detection based on the mean  $\mu_{\Psi_{RI}}$  and standard deviation  $\sigma_{\Psi_{RI}}$  of the feature importance  $\Psi_{RI}$  could be used. A graphical example of this approach can be seen in Figure 2.3. Eight different features were sorted according to their relative importance  $\Psi_{RI}$ , but only those three features above the threshold  $\theta_{\Psi_{RI}}$ , which seem to have the highest impact on the classification are finally used.

### 2.2.5 Classification

The last part of the classification pipeline is the actual differentiation between multiple types of classes. In parametric classification approaches,

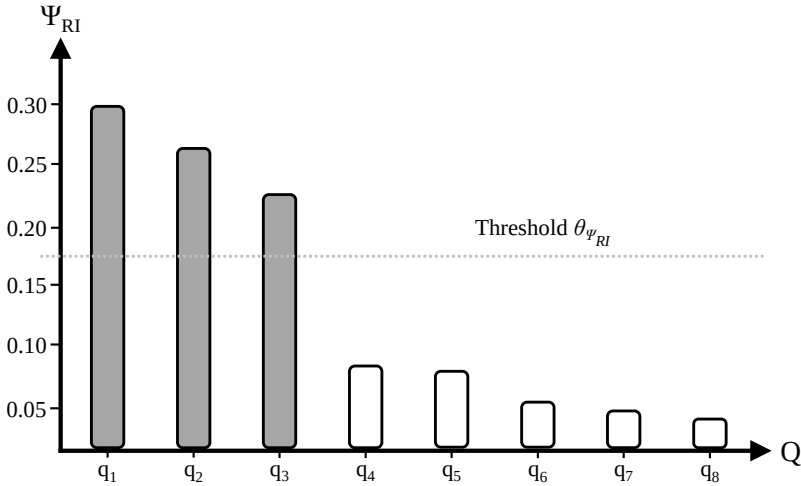


**Figure 2.3:** Feature selection approach using the relative importance  $\Psi_{RI}$  according to the final class predictions. All features are sorted based on their averaged weights. Only features (gray) above the threshold  $\theta_{\Psi_{RI}}$  are used as input for the classification.

the classifier itself can be denoted as a multi-dimensional function  $f(q_V)$  of the feature vector  $q_V$ , which outputs a final class type according to the input features of a pattern. The final classifier model, its specific parameters and constants, which represent the classification components of the working phase, have to be trained and generated as mentioned beforehand during the learning phase by means of the input training data.

Since there exists no best classifier for a specific task, different types of classifiers were tested and deployed in this thesis [Duda 12]. Due to numerous classification approaches in literature, this thesis incorporates following five methods: a

- *Naïve Bayes* (NB) [Lewi 98] classifier, a
- *Random Forest* (RF) [Brei 01] ensemble classifier,
- different types for *Support Vector Machines* (SVM) [Cort 95] and ( $\nu$ -SVM) algorithms [Scho 00], including linear (LIN), polynomial (POLY) and radial-basis-function kernels (RBF), the

- *k-Nearest-Neighbors* (kNN) [Cove 67] algorithm and
- *PART* [Eibe 98] as a rule based classification method.

An explanation of all classifiers including mathematical equations are beyond the scope of this thesis, therefore, only the Support Vector Machine is described in more detail, since it was identified as best performing classification approach during the stroke detection and classification ( $\rightarrow$  see Chapter 5) and thus implemented on the embedded microcontroller of the smart racket prototype ( $\rightarrow$  see Chapter 8.3).

### Support Vector Machine

In general, the SVM tries to separate patterns of similar feature vectors  $\mathbf{q}_V$  by finding a boundary as hyperplane with a maximum margin in the multi-dimensional feature space. The hyperplane is defined using multiple support vectors  $\mathbf{s}_V$  for each class.

An example of a linear case of a Support Vector Machine is shown in Figure 2.4 including a two-dimensional feature set with the features  $q_1$  and  $q_2$ , two separable classes  $c_1$  (cloud of circles) and  $c_2$  (cloud of black dots) and the linear hyperplane. During the working phase, the SVM function  $f_{SVM}(\mathbf{q}_V)$  (Equation 2.1) can be algebraically written as the sum of all support vectors  $\mathbf{s}_V$  multiplied with the feature vector  $\mathbf{q}_V$  using a special kernel functionality.

$$f_{SVM}(\mathbf{q}_V) = \text{sgn}\left\{\alpha_0 + \sum_{n=1}^{S_V} \left((\alpha_{V,n}) \cdot y_C \cdot K\langle(\mathbf{s}_{V,n}), \mathbf{q}_V\rangle\right)\right\} \quad (2.1)$$

$$1 \leq n < \text{Number of support vectors } S_V$$

$$f_{SVM}(\mathbf{q}_V) = \text{Support vector machine}$$

$$K\langle\mathbf{s}_V, \mathbf{q}_V\rangle = \text{Kernel functionality}$$

$$\alpha_0 = \text{Hyperplane intercept}$$

$$\alpha_V = \text{Weight of each support vector } \mathbf{s}_V$$

$$y_C = \text{Class label}$$

$$\mathbf{s}_V, \mathbf{q}_V = \text{Support vector and feature vector}$$

The class decision of this function results in either - 1 or + 1 according to the assigned class label  $y_C$  during the training phase. Thereby,  $\alpha_0$  is the intercept of the trained hyperplane,  $\alpha_V$  the weights of each associated

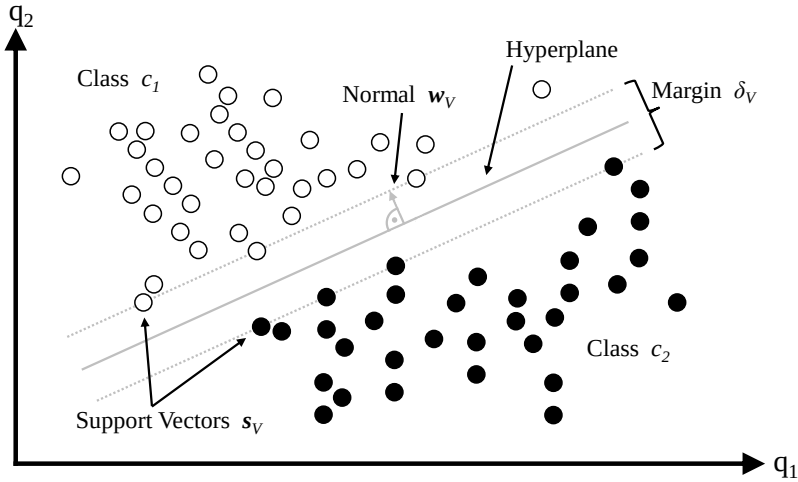


**Figure 2.4:** Simplified SVM example for a linear case. The features  $q_1$  and  $q_2$  span a two-dimensional feature space consisting of two classes  $c_1$  and  $c_2$ . The SVM is trained in such a way, that both classes should ideally be separable by a hyperplane (in this case the straight gray line) with a maximized margin between their appropriate feature vectors  $q_V$ . These feature vectors have the shortest distance to the hyperplane and are also called support vectors  $s_V$ .

support vector  $s_V$ ,  $S_V$  the number of all support vectors for each class and  $K\langle s_V, q_V \rangle$  the applied kernel functionality [Chan 11].

By itself the SVM is solely a linear classification methodology, but not all data are linearly separable within their initial data space. In order to nevertheless be able to address non-linear recognition problems, the input data again can be transformed into a higher-dimensional feature space, in which a linear separability could be possible. This approach is known as kernel trick [Bose 92]. Different kinds of kernels can be applied to recognize a wide range of various patterns comprising non-linear distributions within the feature space.

The SVM approaches investigated in this thesis use a linear kernel type  $K_{LIN}\langle s_V, q_V \rangle$  (Equation 2.2), a polynomial kernel type  $K_{POLY}\langle s_V, q_V \rangle$  (Equation 2.3) and a radial-basis-function kernel type  $K_{RBF}\langle s_V, q_V \rangle$  (Equation 2.4). During the training phase, all the parameters  $\alpha_0$ ,  $\alpha_V$ ,  $s_V$ ,  $S_V$ ,  $d_{poly}$  and  $\gamma$  of the classifier model have to be learned and evaluated to achieve the best performance of each SVM. Finding these set of

parameters means solving the hyperplane position with the maximum margin constraint.

$$K_{LIN}\langle \mathbf{s}_V, \mathbf{q}_V \rangle = (\mathbf{s}_V)^T \cdot \mathbf{q}_V \quad (2.2)$$

$$K_{POLY}\langle \mathbf{s}_V, \mathbf{q}_V \rangle = \left(1 + (\mathbf{s}_V)^T \cdot \mathbf{q}_V\right)^{d_{poly}} \quad (2.3)$$

$$K_{RBF}\langle \mathbf{s}_V, \mathbf{q}_V \rangle = \exp\left(-\gamma \cdot \|\mathbf{s}_V - \mathbf{q}_V\|^2\right) \quad (2.4)$$

$K_{\{LIN, POLY, RBF\}}\langle \mathbf{s}_V, \mathbf{q}_V \rangle$  = Kernel functions

$d_{poly}$  = Polynomial degree

$\gamma$  = Inverse of the RBF kernel standard deviation

$\mathbf{s}_V, \mathbf{q}_V$  = Support vector and feature vector

The hyperplane classifies the input data if condition (Equation 2.5) is fulfilled. For this condition,  $y_C$  describes the class label and  $\mathbf{w}_V$  the normal to the hyperplane boundary for a specific feature vector  $\mathbf{q}_V$ .

$$y_C \cdot (\langle \mathbf{w}_V, \mathbf{q}_V \rangle + \alpha_0) \geq 1 - e \quad (2.5)$$

$y_C$  = Class label

$\mathbf{w}_V$  = Normal vector

$\mathbf{q}_V$  = Appropriate feature vector

$\alpha_0$  = Intercept

$e$  = Training error

For an optimal hyperplane (data is completely separable,  $e$  is thus 0), the margin  $\delta_V$  of the hyperplane is expressed as  $\delta_V = \|\mathbf{w}_V\|^{-1}$  [Cort 95]. The required maximization of the margin can be achieved by the minimization of the term  $0.5 \cdot \|\mathbf{w}_V\|^2$ , which is mathematical equal to  $\max\{\delta_V\}$ . Both conditions lead to a quadratic constrained optimization problem, which can be solved by the Lagrangian multiplier approach [Chan 11]. If the input data are not linearly separable, which leads to training errors ( $e$  is not 0), the sum of all errors  $e$  can additionally be minimized to achieve a best possible hyperplane solution allowing minimal misclassifications. The optimization algorithms yield all model parameters including the intercept  $\alpha_0$ , all necessary feature vectors  $\mathbf{q}_V$  known as support vectors  $\mathbf{s}_V$ , the terms  $\alpha_V \cdot y_C$  including the individual

weights and class labels for the support vectors and if needed, the kernel variables  $d_{poly}$  for the polynomial kernel and  $\gamma$  for the RBF kernel.

Due to the mathematical description, it is clear, that one SVM can only differentiate between two classes. For a multi-class recognition task, multiple SVMs have to be trained to compare different classes with each other [West 99] or to compare every class with all remaining classes  $c_R$ . For instance, a recognition task with three different classes  $c_1$ ,  $c_2$  and  $c_3$  requires the trained classifiers  $\text{SVM}_{(c_1 \leftrightarrow c_2)}$ ,  $\text{SVM}_{(c_1 \leftrightarrow c_3)}$  and  $\text{SVM}_{(c_2 \leftrightarrow c_3)}$  for direct comparison. Alternatively,  $\text{SVM}_{(c_1 \leftrightarrow c_R)}$ ,  $\text{SVM}_{(c_2 \leftrightarrow c_R)}$  and  $\text{SVM}_{(c_3 \leftrightarrow c_R)}$  can be used for testing against the remaining classes. The final decision is made by a majority vote of all single SVMs.

## 2.3 Evaluation

A meaningful evaluation is essential to both quantify the performance of a machine learning algorithm as well as to optimize its learning process during the training phase. This holds true for an algorithm in general and also for a specific classifier. Using the evaluation it can be analyzed whether the developed classification pipeline fits to the initial recognition task. Additionally, different classifiers applied on one single application can be compared regarding accuracy, complexity and computational costs. Therefore, a separate test and training set of the input data is needed. The classifier is trained with data of the training set and subsequently evaluated by unseen data of the test set. This is also important to avoid the mentioned overfitting problem. Either both data sets were initially obtained during the acquisition study or the the whole input data are split into training and test data using a *Cross Validation* (CV) method afterwards [Arlo 10].

A particular case of the cross validation is the *Leave-One-Subject-Out* (LOSO) cross-validation, whereby the input data are split by subjects of the acquisition study. In the context of table tennis strokes, data of multiple players could be acquired during a study. The data set from one player could be assigned as a test data, whereas the data sets of the remaining players are used to train the classifier. This is then repeated for every player resulting in different test and training sets. All single results are averaged to achieve the overall performance of the classifier.

Finally, different numeric measures can be used for quantification of a classification system. Assuming a binary classification system, correctly

classified elements # are marked as *True Positives* ( $\#TP$ ). If these elements are wrongly classified, they are marked as *False Positives* ( $\#FP$ ). Furthermore, they can be marked as *True Negatives* ( $\#TN$ ), if the elements are correctly rejected by the classifier and as *False Negatives* ( $\#FN$ ), if the elements are wrongly rejected. These assignments can be combined to commonly used scores like *Precision* (Equation 2.6) and *Recall* (Equation 2.7), as well as *Accuracy* (Equation 2.8). The *F1-Score* (Equation 2.9) can be used as harmonic mean to combine precision and recall. It reaches its best performance by perfect precision and recall.

$$\text{Precision} = \frac{\#TP}{\#TP + \#FP} \quad (2.6)$$

$$\text{Recall} = \frac{\#TP}{\#TP + \#FN} \quad (2.7)$$

$$\text{Accuracy} = \frac{\#TP + \#TN}{\#TP + \#TN + \#FP + \#FN} \quad (2.8)$$

$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (2.9)$$

# = Amount of class elements

A detailed view on the all single class decisions can be illustrated by a *Confusion Matrix* (example shown in Table 2.2). In this matrix table, all classified elements # of a specific class  $c_{\text{classified}} = f(q_V)$  predicted by the classifier (columns) are compared with their true classes  $c_{\text{true}}$  of the labels (rows). For a perfect classification, all elements should be lo-

**Table 2.2:** Confusion matrix example with four classes  $c_1$ ,  $c_2$ ,  $c_3$  and  $c_4$ . The diagonal shows the numbers of correctly classified elements ( $\#TP$ ) for each class. Here, class  $c_4$  was 487 times correctly classified, one time classified as  $c_1$  and seven times classified as  $c_3$ , but never mixed up with class  $c_2$ .

| Class                |       | Classified $\longrightarrow$ |       |       |       |
|----------------------|-------|------------------------------|-------|-------|-------|
|                      |       | $c_1$                        | $c_2$ | $c_3$ | $c_4$ |
| True<br>$\downarrow$ | $c_1$ | 452                          | 0     | 1     | 0     |
|                      | $c_2$ | 0                            | 523   | 0     | 8     |
|                      | $c_3$ | 2                            | 3     | 456   | 4     |
|                      | $c_4$ | 1                            | 0     | 7     | 487   |

cated on the diagonal, which would imply 100 % accuracy. If other items of the matrix are not 0, the confusion matrix shows, how often classes were misclassified with other classes. The confusion matrix example in Table 2.2 shows a multi-class problem with four classes  $c_1$ ,  $c_2$ ,  $c_3$  and  $c_4$ .

Additionally to the numeric measures and confusion matrices described in this chapter, other evaluations were part of this thesis. They are presented within the appropriate results sections of each method chapter. These include the selected feature sets, class dependent accuracies and analysis of classification complexities on the basis of cost considerations regarding the required memory consumption and computational efforts.

## Chapter 3

# Embedded Signal Processing

The developed algorithms of the presented methods are intended to run within the smart racket of the feedback device. Therefore, signal processing steps and machine learning approaches executed on such real-time and resource restricted embedded microcontroller environments must be handled differently as opposed to an implementation within standard computers providing almost unlimited resources.

The following chapter summarizes important design considerations and hardware restrictions. Besides, it also deals with benefits of embedded systems based on microcontrollers. Additionally, it describes concepts how to adapt the machine learning pipeline (→ see Chapter 2.2) together with additional signal processing components for a real-time operating system running on an embedded processor. All these aspects are examined related to classification system and signal processing demands of methodologies introduced by this thesis.

### 3.1 Considerations, Restrictions and Benefits

Multiple considerations and restrictions, but also potential benefits have to be taken into account, if embedded low-end microcontoller systems in contrast to personal computers are used for signal processing tasks [Grid 07]. On the one hand, important aspects are the availability of *Hardware Resources* regarding *Memory* and *Clock Speed* and the assumed *Execution Time* of the intended processing tasks. On the other hand, the trade-off of *Costs and Complexity*, while yielding a certain *Arithmetic Precision*, has to be considered. These factors are especially important, if already developed algorithms have to be migrated for targets of real-time systems with continuous data and significantly reduced

hardware resources. Embedded hardware for sports applications like fitness trackers, smartwatches or motion sensors is mainly dimensioned to work in limited space, for low-power operation consuming as little battery power as possible and with lowest production costs for a wide consumer market [Koop 90].

On that account, a vast number of microcontrollers including all necessary components of a fully miniaturized computer comprising a central processing unit (CPU), non-volatile FLASH storage, random access memory (RAM), input and output ports (I/O) and many additional peripherals and modules are currently available from various manufacturers. Since production costs are often the principle factor for consumer products, the cheapest controller, which meets all the necessary requirements of the desired application, will be selected. Therefore, most of the available microcontrollers are tailored for specific processing tasks or application scenarios [Grid 07].

All considerations and encountered restrictions of the following sections are adapted to the characteristics and parameters of the modern and commonly used ARM Cortex-M<sup>1</sup> 32-bit microcontroller architecture. This hardware platform was used for the wearable sensor development (→ see Chapter 4) and for the implementation of the contributed methods within the smart racket instrumentation (→ see Chapter 8).

### 3.1.1 Memory Consumption

The majority of standard low-end microcontrollers feature only limited hardware resources for memory space, for example FLASH memories of < 512 kByte and integrated RAM of < 64 kByte [Grid 07]. These technical characteristics highlight the implementation challenges of expensive signal processing and machine learning algorithms. Microcontrollers have to deal with memory capacities, which are two to three orders of magnitude smaller than the resources on standard computers.

For some working phases of machine learning methods, for example the SVM and RF algorithm of this thesis, the overall memory expenditure for the storage of all model parameters can be well predicted. The total memory consumption  $SVM_{size}$  (Equation 3.1) in kByte of a SVM classification system can be determined. Every trained value and constant for each SVM, including the support vectors  $s_V$ , the weights  $\alpha_V$  and the intercept  $\alpha_0$  have to be stored within the non-volatile FLASH

<sup>1</sup><https://www.arm.com/products/processors/cortex-m> – last access 04/2018

memory. The number of all required variables multiplied with the bit length per value  $BpV$  result in an overall memory consumption estimation equation.

$$SVM_{size} = \sum_{i=n}^{\#SVM} \left( S_{V,n} \cdot Q_n + A_n + 1 \right) \cdot \frac{BpV}{8000} \quad (3.1)$$

$1 \leq n < \text{Number of support vector machines } \#SVM$

$SVM_{size}$  = Size of support vector machine in  $[kByte]$

$S_V$  = Number of support vectors  $s_V$

$Q$  = Number of features in feature vector  $q_V$

$A$  = Number of weights  $\alpha_V$

$BpV$  = Bits per value in  $[bit]$

Assuming a four-class recognition task comprising 6 SVMs  $\#SVM$  with each 200 support vectors  $S_V$  and 200 weights  $A$  including 90 features  $Q$  on a 32-bit microcontroller architecture using a floating point data representation, the memory requires at least 437 kByte merely for the storage of the classifier model.

Furthermore, an upper limit estimation for the memory consumption of a binary Random Forest classification system  $RF_{size}$  (Equation 3.2) in kByte can also be expressed.

$$RF_{size} = \sum_{i=1}^{\#DT} \left( (2^{d_{tree}})_n - 1 \right) \cdot \frac{BpV}{8000} \quad (3.2)$$

$1 \leq n < \text{Number of decision trees } \#DT$

$1 \leq d_{tree} < \text{Number of features in feature vector } q_V$

$RF_{size}$  = Size of random forest in  $[kByte]$

$BpV$  = Bits per value in  $[bit]$

All trained constants for the binary comparisons of every node for each decision tree  $\#DT$  have to be stored within the non-volatile FLASH memory. Therefore, the maximum amount of nodes per tree multiplied with the bit length per value  $BpV$  can be used as an upper limit estimation for the required memory size. Here, the depth  $d_{tree}$  of a decision tree cannot be higher than the number of available features  $Q$ , because each

feature can only be used once for comparison of one branch from the root node to a specific leaf node. In practice, the depths  $d_{tree}$  are much smaller than the total number of features using only a feature subset per branch, while the number of decision trees  $\#DT$  are increased, thus reducing the overall computational load [Brei 01]. For instance, for a Random Forest based event detection algorithm using 10 decision trees  $\#DT$  and a depth  $d_{tree}$  of 15 for each tree on the same hardware platform as mentioned before, roughly 1311 kByte of memory have to be allocated.

Both examples show, that the required FLASH memory spaces are in the range or even exceed the maximum available memory of most microcontrollers. This does not even consider additional memory space needed for the program code. That is the reason why it is essential to optimize the overall memory consumption of the intended classification system. Often, the alignment of memory space is associated with down-scaling of performance and effectiveness of the classification pipeline. Usually, a notable accuracy loss has to be accepted.

Besides the classification model stored within the non-volatile FLASH memory, additional space within the RAM has to be considered providing buffers for the input sensor data and the computed feature vector during the data collection, the preprocessing step and the subsequent feature extraction. These estimations can be expressed for the input data buffers  $DATA_{size}$  (Equation 3.3) and for the feature vector buffer  $FEATURE_{size}$  (Equation 3.4), both computed in kByte.

$$DATA_{size} = n_s \cdot m_s \cdot \frac{w_L}{T_A} \cdot \left[ \frac{w_O}{w_L - w_O} + 1 \right] \cdot \frac{BpV}{8000} \quad (3.3)$$

$$FEATURE_{size} = \left( n_s \cdot m_s \cdot Q_{SC} + Q_{MC} \right) \cdot \frac{BPV}{8000} \quad (3.4)$$

$DATA_{size}$  = Size of input data buffer in [kByte]

$FEATURE_{size}$  = Size of feature vector buffer in [kByte]

$n_s$  = Number of sensors

$m_s$  = Number of channels per sensor  $n_s$

$Q_{SC}$  = Number of single-channel features

$Q_{MC}$  = Number of multi-channel features

$w_O, w_L$  = Window overlap and window length in [s]

$T_A$  = Sampling interval in [s]

$BpV$  = Bits per value in [bit]

For example, an algorithm needs 27 kByte RAM for the input data buffers  $DATA_{size}$  and 360 Byte RAM for the feature vector  $FEATURE_{size}$ , if 3 sensors  $n_s$  with each 3 channels  $m_s$  (for instance a three-dimensional accelerometer, gyroscope and geo-magnetic sensor) are captured at a sampling interval  $\sigma T_A$  of 0.002 s. The window length  $w_L$  is chosen to 0.75 s including a window overlap  $w_O$  of 0.25 s, for example, to detect consecutive table tennis strokes. Nine single channel features  $Q_{SC}$  and 9 multi channel features  $Q_{MC}$  are computed. Again, the hardware architecture is a 32-bit processor using a floating point data representation.

Alongside these data buffers, additional values and structures during the complete run-time have to fit into the entire RAM. These could include further stacks for a potential operating system, static and non-constant variables or interfaces for passing input and output data. Admittedly, the RAM sizes of standard microcontrollers seem not to be as critical as the FLASH memory sizes for a machine learning or an embedded signal processing application within this context, but it is also not possible to freely choose any window length  $w_L$  or large overlaps  $w_O$ . High values for both parameters contribute to a quadratic increase of the data buffer  $DATA_{size}$  and are therefore exceeding the available RAM memory space rapidly.

### 3.1.2 Execution Time

Considerations about the CPU speed and the underlying hardware clock required for a specific processing task are not as simple as estimations about the memory consumption. Different factors regarding the available instruction set of the embedded microprocessor, its hardware architecture (for example reading and writing speeds for attached memory devices limited by the connecting bus system) or the efficiency of program code including the operating system and the actual algorithm implementation highly influence the execution time behavior. Basically, it is not important to process data as fast as possible, but to provide information in real-time [Koop 90]. Processing periodic table tennis strokes of a game or exercise rally can be very well described by such a real-time environment. Besides the proposed methods of this thesis, most applications comprise complex and strongly interconnected algorithms. Therefore, it is beneficial to implement a *Real-Time Operating System* (RTOS), which can assign different processing steps to specific tasks, can schedule these tasks at reasonable time points and can ensure the compliance of all real-time conditions. Time requirements, which depend

on the application, are expressed by the *Worst Case Execution Time* (WCET) and the maximum permitted *Response Time* (RT) [Kope02]. Both parameters, including a useful task scheduling approach for the machine learning pipeline and additional signal processing components within the context of this thesis, are described in the following sections (→ see Chapter 3.2) in more detail.

Nevertheless, the computation time of an algorithm or a section of program code can be roughly estimated by closer examination of the mathematical expressions used. This is also done by the *Embedded Classification Software Toolbox* (ECST) [Ring12]. This toolbox can be used to analyze the complexity and the computational effort of a classification system. The number of certain operations can give insight into the necessary computation times. Moreover, comparisons ( $<$ ,  $>$ ), additions ( $+$ ) and subtractions ( $-$ ) can normally be executed much faster than multiplications ( $\times$ ) and divisions ( $\div$ ) or even exponentiations ( $e^x$ ), logarithms ( $\log x$ ) or roots ( $\sqrt{x}$ ) [Smit13]. Although the effective execution speed of a single operation varies depending on the used hardware platform, the amount of each operation can be used to estimate the computation times or to compare the necessary efforts of different algorithms.

$$\begin{aligned} SVM_{effort} = & \left[ \#SVM + 2 \cdot Q \cdot S_V \right]_{(+,-)} + \\ & + \left[ \#SVM + C - 1 \right]_{(<,>)} + \\ & + \left[ 2 \cdot (3 + Q) \cdot S_V \right]_{(\times)} + \\ & + \left[ S_V \right]_{(e^x)} \end{aligned} \quad (3.5)$$

$SVM_{effort}$  = Mathematical effort of a SVM system

$\#SVM$  = Number of all single SVMs

$C$  = Number of classes

$S_V$  = Number of support vectors  $s_V$

$Q$  = Number of features in feature vector  $q_V$

For example, to characterize the number of operations and expected mathematical demand of a radial-basis-function (RBF) based Support Vector Machine  $SVM_{effort}$  (Equation 3.5), the number of all single Support Vector Machines  $\#SVM$  and support vectors  $S_V$  are necessary.

Additionally, the number of all features  $Q$  and the number of total classes  $C$  are needed as well [Jens 16]. Revisiting the above introduced four-class recognition task ( $C$  with 4,  $\#SVM$  with 6,  $S_V$  with 200 and  $Q$  with 90), the CPU of the microcontroller has to execute 36006 additions and subtractions, nine comparisons, 37200 multiplications and 200 exponential functions, which are in total 73415 mathematical operations.

In addition to the Support Vector Machine, the numeric demand  $RF_{effort}$  (Equation 3.6) of a binary Random Forest algorithm can also be analyzed. It is dependent on the number of decision trees  $\#DT$  and the tree depth  $d_{tree}$  of each tree [Jens 16].

$$RF_{effort} = \left[ \#DT \cdot d_{tree} + \#DT - 1 \right]_{(<, >)} \quad (3.6)$$

$$\begin{aligned} RF_{effort} &= \text{Mathematical effort of a binary RF system} \\ d_{tree} &= \text{Depth of each decision tree } \#DT \end{aligned}$$

Regarding the RF binary event detection including  $C$  with 2,  $\#DT$  with 10 and  $d_{tree}$  with 15 ( $\rightarrow$  see Chapter 3.1), the algorithm needs only 159 comparisons in total to achieve its final decision. Compared to a similar but theoretical two-class SVM case using  $C$  with 2,  $\#SVM$  with 1,  $S_V$  with 30 and  $Q$  with 15, resulting in 2013 necessary mathematical operations, the RF algorithm would have to execute overall 13 times less executions. Assuming that all mathematical operations on one specific hardware CPU need the same execution time, which is not realistic, but can be used for an rough estimation, the RF algorithm seems to be much faster than any RBF based SVM even for high numbers of decision trees  $\#DT$  and tree depths  $d_{tree}$ . However, one must live with the initial high memory costs for the storage of all comparison constants of the classifier model. For a final algorithm selection, overall costs and complexities regarding memory and computational expenditure based on the available hardware resources must be considered and weighted individually for each signal processing task of every developed application.

Nevertheless, the actual classification of a machine learning task itself is often not the most time-consuming step of the entire signal processing pipeline. In some cases conversion and preparation of measured data, preprocessing or feature extraction are equally or even more computational intensive. Considerations concerning these computational efforts of some features ( $\rightarrow$  see Chapter 2.2) illustrate this fact. The feature vector  $q_V$  contains  $Q$  features, which have to be calculated for all

values within the window length  $w_L$  of one data segment and the sampling interval  $T_A$ . According to [Jens 16], the needed efforts  $q_{\sigma, effort}$  and  $q_{w, effort}$  are exemplarily shown for the standard deviation (Equation 3.7) and for the kurtosis (Equation 3.8).

$$q_{\sigma, effort} = \left[ 3 \cdot \frac{w_L}{T_A} - 1 \right]_{(+,-)} + \left[ \frac{w_L}{T_A} \right]_{(\times)} + \left[ 2 \right]_{(\div)} + \left[ 1 \right]_{(\sqrt{x})} \quad (3.7)$$

$$q_{w, effort} = \left[ 5 \cdot \frac{w_L}{T_A} + 3 \right]_{(+,-)} + \left[ 3 \cdot \frac{w_L}{T_A} + 9 \right]_{(\times)} + \left[ 4 \right]_{(\div)} \quad (3.8)$$

$q_{\{\sigma, w\}, effort}$  = Mathematical efforts of standard deviation and kurtosis

$w_L$  = Window length in [s]

$T_A$  = Sampling interval in [s]

Taking again the last SVM example ( $w_L$  with 0.75 s and  $T_A$  with 0.002 s), the standard deviation feature  $q_{\sigma}$  requires 1502 mathematical operations whereas the kurtosis feature  $q_w$  consumes 3016 instructions. With a size of 90 features and the assumption that the feature vector  $q_V$  would only contain features with similar mathematical efforts compared to the standard deviation, the total number of needed operations would increase to 135180, which would be twice the necessary operations of the actual classification step by the RBF Support Vector Machine.

Surely, these are just rough estimations and considerations, since not all features are as costly to compute as the standard deviation or the kurtosis. Some features require only a small amount of comparisons, whereas other features need a high number of additions and multiplications. Additionally, the kinds of mathematical operations are not directly comparable, because their execution times are different. All these characteristics are summed up in Table 2.1.

It should also be noted, that many features are interdependent, which can be seen in Figure 3.1. This relationship has significant influence on the overall costs and complexity consideration. For example, omitting the standard deviation would also lead to omission of skewness and kurtosis. Another example is trying to save execution time by exclusion of the median feature  $q_{median}$ . Since most of the signal characteristics need a sorted list of the raw input data, almost all execution time is consumed by applying a sorting algorithm instead of the final feature computation, which requires often one single operation. Therefore, only the imple-

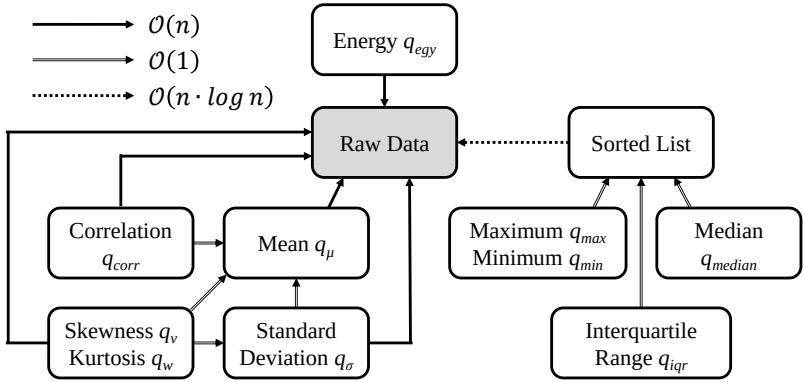


**Figure 3.1:** Visualization of computational dependencies for the presented features. On the one hand, some features need other features as basis for calculation, whereas on the other hand, different features need additional signal processing beforehand. In this case, a heap sort algorithm is imperative to be able to calculate signal characteristics. The average efforts are estimated by the Landau notation  $\mathcal{O}(\dots)$ .

mentation of the sorting algorithm itself has an influence on the overall processing times of these features. However, one has to keep in mind, that some sorting algorithms are faster than others, for instance the heap sort algorithm [Will 64], but could probably sort not in-place, as for example the standard merge sort algorithm [Knut 97], which would automatically imply doubling the required RAM space  $DATA_{size}$  for the input data buffers.

Besides the described considerations regarding the execution time, which are mainly restrictions, many microcontrollers benefit from integrated peripherals, modules and mechanisms to reduce the overall computational load of the CPU. Such peripherals are primarily designed to interact with the physical world around the embedded system and therefore intended to communicate with sensors or actors and to transport data among themselves. Elementary tasks like continuously capturing sensor data and their storage within the RAM memory space can be effectively handled concurrently using specific bus interfaces and a direct memory access (DMA) mechanism. Thereby, data can be acquired and stored self-controlled within predefined memory sections for subsequent processing. This procedure requires no extra interaction of the

CPU [Grid 07] and the gained CPU time can be used for actual signal processing tasks.

Furthermore, floating point operations can be carried out by a dedicated floating point unit (FPU) to further relieve the processor. Some microcontrollers contain coprocessors or integrated modules, which are able to rapidly execute floating point operations directly in hardware. This is highly advantageous in comparison to software-based fixed point operations, which have less precision and a lower dynamic range [Smit 13]. Moreover, a hardware FPU can execute operations faster than any instruction emulation of a slow virtual floating point library. Here again, CPU execution capacity can be obtained and used for more important time-critical tasks. Employing such auxiliary modules, the computational capacities of a microcontroller can be enhanced and in the end more processing power can be timed for the intended classification algorithms than for preprocessing and feature computation steps.

#### 3.1.3 Arithmetic Precision

Alongside memory restrictions and considerations about necessary CPU speeds of microcontrollers, the hardware platform as well as the compiler for the firmware must yield a specific arithmetic precision. Since most equations for the presented classifiers and feature computations are built upon floating point systems, it is necessary to calculate with sufficient mathematical precision. A precision loss after every conversion or mapping between floating point operations and fixed point operations is compulsory. Therefore, a sufficient precision can only be achieved if the number of these operations is kept as low as possible [Kuo 13]. Thus, a FPU should always be preferred if more complex signal processing in the context of machine learning methods has to be conducted. The resulting avoidance of dispensable operations is therefore another option to keep the computational efforts as low as necessary. Additionally, it provides the benefit of higher numeric accuracy.

An important point regarding arithmetic precision is the difference between computations on the standard computer hardware system for the learning phases of a classification pipeline and similar computations on another microcontroller target system for the working phases. Most methods and experiments within this thesis were performed on a system implementing *Python 3*<sup>2</sup> and the machine learning toolbox *Scikit-learn*

---

<sup>2</sup><https://www.python.org/> – last access April 2018

[Pedr 11]. Both are based on the numerical library *NumPy* [Olip 15]. This library performs every computation using a data representation with a double precision in comparison to the dedicated floating point units of many microcontrollers, which provide only a single precision data representation. The same is true for the ARM Cortex-M microcontroller platform within the racket instrumentation [ARM 10]. That means, all features and thus the resulting trained parameters of the classifier models are evaluated with higher precisions than the feature computation and final class decision during the working phase done on the microcontroller.

As a consequence, it has to be considered, that in certain cases erroneous classification results could occur. If possible, it is recommended to perform the parameter training of the classification system with the same single precision as provided by the target system. However, these observations and thereof possible implications have to be evaluated for every specific application, since the mathematical relations between preprocessed data, features and classification approaches always behave differently for each processing pipeline.

### 3.2 Real-Time Operating System

Approaching more complex signal processing problems often requires an underlying *Real-Time Operating System* (RTOS), which comes with some advantages compared to a direct implementation. Thereby, the operating system provides capabilities to split clustered but loosely connected processing blocks and to assign them to specific working tasks. All tasks are then scheduled based on priority with reasonable timing to ensure compliance with all real-time constraints imposed by the application [Chen 03]. The complex table tennis analysis has to be implemented on the microcontroller of the racket prototype, making a RTOS essential.

However, some requirements have to be fulfilled if using a RTOS. They include the consumption of as little memory space for FLASH and RAM as possible for the necessary RTOS kernel, almost no CPU interaction overhead during task switching, providing displacing and priority-based scheduling to maintain all time constraints as well as directives and semaphores for synchronization, event management, interrupt handling and data transfer within the scheduled tasks [Trin 12].

Similar to the availability of microcontrollers, many different real-time operating systems exist. They are customized for the intended application context or optimized for the manufacturers' hardware platform [Liu 00]. In the following sections, only the most important concepts for application of a RTOS which are directly influenced by the intended processing tasks are discussed to ensure the general validity of the proposed principles.

### 3.2.1 Machine Learning Matching

In an embedded real-time system, any upcoming computation job must be split into separate *Threads*  $H_i$  with  $i \in \mathbb{N}$ , which are initiated by a specific event or as a subsequent element of a thread queue. Thus, a thread represents the basic manageable work unit [Liu 00]. After the execution of all computations within the thread, it will be terminated or destroyed and another event can be generated. This event in turn could be used to start the next processing tasks in the form of new threads or the subsequent thread within the thread queue. Certainly, there are many ways to match classification tasks and signal processing algorithms to a meaningful construct of different independent threads. In this thesis, the matching of the working phase for the machine learning pipeline incorporates three threads, which are the

- *Data Acquisition and Preprocessing Thread*  $H_1$ , the
- *Feature Extraction Thread*  $H_2$   
(also usable as signal processing thread) and the
- *Classification Thread*  $H_3$   
(also usable as signal processing thread).

Additional processing steps like filtering, normalization and peak detection can be included into the preprocessing thread  $H_1$ , because these steps interact with preparation of raw input data for subsequent steps and are usually not computationally expensive. If neither feature extraction nor classification is needed, the threads  $H_2$  and  $H_3$  can be used for analytical time-domain computations, which are not categorized as an actual classification task. This is the case for the estimation of relative motion and orientation of the table tennis racket based on inertial sensor data ( $\rightarrow$  see Chapter 6). While doing this, the thread allocation procedure should support a high task variability while maintaining

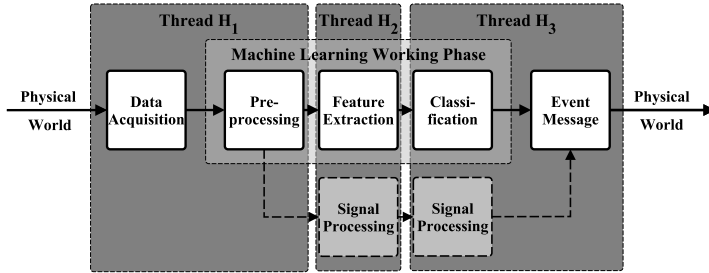


**Figure 3.2:** The working phase of a machine learning pipeline can be matched to three different threads of a real-time operating system. Thereby, thread  $H_1$  performs the data acquisition and preprocessing, whereas thread  $H_2$  and  $H_3$  contain the feature extraction and the classification. Additional signal processing tasks can be matched to the last two threads respectively, if no actual classification is intended.

minimum effort resulting from the thread management and its synchronization mechanisms [Kope 02]. This overall task and thread mapping is shown in Figure 3.2. A further breakdown of tasks into smaller threads is possible, but additional costs for scheduling and memory consumption would demand more hardware resources of the microcontroller.

### Data Acquisition and Preprocessing Thread $H_1$

The data acquisition and the preprocessing task  $H_1$  requires a continuously and recurring execution, because all data channels of each sensor have to be captured with predefined but possible sensor-dependent sampling intervals  $T_A$ . Often, variable window sizes  $w_L$  including various window overlaps  $w_O$  are necessary for different sensors types, which makes this thread highly dynamic. Since data acquisition is essential for every embedded system interfacing with the physical world,  $H_1$  must have the highest priority enabling it to interrupt any other thread.

### Feature Extraction Thread $H_2$

After accumulation of enough data by the data acquisition thread  $H_1$ , the feature extraction thread  $H_2$  computes all input features and requirements needed for the subsequent classification. This thread is relatively coherent, since only straight forward mathematical operations

have to be conducted in a defined order. Within thread  $H_2$ , all feature dependencies have to be considered and only those features computed, which are necessary for the directly following classifier of the current processing chain.

#### **Classification Thread $H_3$**

The classification thread  $H_3$  is built upon data acquisition and feature extraction. After completion of the feature extraction, thread  $H_3$  detects and classifies events based on its classification algorithm. This task can be very dynamic and complex, since different classifiers based on various algorithms and input requirements could be used during one application context in multiple processing chains. Usually, the classification itself is the last step within the pipeline with no further extensive processing components. Therefore, the classification thread  $H_3$  can also implement the final event management. This could be the storage of all detected events or computed results. In addition, these results could also be transmitted for feedback in real-time to an external device.

#### **Idle Thread $H_{idle}$**

Besides the application specific threads  $H_1$ ,  $H_2$  and  $H_3$ , an additional idle thread  $H_{idle}$  is necessary. This thread is called if the application does not need the total execution time of the available CPU capacity. Thus, the idle thread  $H_{idle}$  has the lowest priority and involves a simple endless loop. During this loop, it is possible to shutdown the CPU and currently unused modules and peripherals. Thereby, the hardware platform could enter an energy saving mode. In this case, a high amount of current consumption and battery power could be saved, which leads to a longer runtime of the entire wearable sensor system. However, this functionality is only reasonable, if the idle time between two consecutive scheduled threads is clearly longer compared to the time effort needed switching the threads itself.

### **3.2.2 Task Scheduling**

The main task of a RTOS is to provide computation capacity to an application in terms of CPU time. Admittedly, a few microcontrollers with more than one CPU core exist, but commonly, a task comprises of more threads than the number of microcontroller CPUs. Therefore, the RTOS

has to multiplex the processor capacity over time [Liu 00]. To generally enable any thread execution, a scheduler is indispensable and therefore the key part of every RTOS based system [Marw 06]. During the last decades of research and microprocessor development, numerous scheduling approaches including scheduler with soft and hard deadlines, approaches for periodic and aperiodic tasks or methods for preemptive, sequential, static or dynamic scheduling have been available [Marw 06]. Threads have to be scheduled sequentially or with interruptions according to their given priority. From the application perspective, the execution of tasks seem to be quasi-parallel, since all threads should be executed within the predefined time frames and specified deadlines.

Regarding the application of a machine learning systems according to the used methods for the intended table tennis analysis, two different scheduling approaches and the resulting thread prioritization are discussed. They involve dynamic

- *Periodic Scheduling* and
- *Aperiodic Scheduling*.

Both work with preemptive scheduling yielding hard deadlines to ideally cover many applications with signal processing requirements. In the next sections, the real-time characteristics of these scheduling approaches as well as their compliances are investigated. For this purpose, a *Worst Case Execution Time* (WCET) analysis has to be done. The WCET represents the longest possible execution time for an algorithm or a program section. However, a precise computation of the WCET is usually almost impossible, because it depends on many unpredictable parameters. Physical measurements can merely provide rough estimations for the WCET. Using those time measurements, only a lower boundary for the WCET value is ascertainable. Influencing parameters, for example, could be the environmental temperature and humidity, which affect the behavior of the semiconductor hardware, the raw input data values themselves, which determine the control flow during sorting algorithms or the error handling of an implemented function [Ferd 03].

Additionally, a deeper view on the runtime behavior gives further information about the real-time ability of the implemented system. Thereby, it is not important that processing parts are done in real-time but finished in time before a predefined deadline occurs. This is determined by the maximum *Response Time* (RT). Quantitatively, the RT is the time period from a triggered event or occurrence until the processing

results or a classification decision can be provided. Within the context of table tennis, the RT could be the time from a ball-racket impact event over the verification of a stroke until its type could be classified. Using a RTOS does not mean that an application is executable in real-time. The crucial aspect is the relation of when a result has to be provided and its effective availability.

### Periodic Scheduling

Periodic scheduling is the standard case. Every thread is executed periodically. A schematic representation is shown in Figure 3.3 for the presented machine learning pipeline and the specified threads  $H_1$ ,  $H_2$  and  $H_3$ . The total WCET for each thread is shown as multiple accumulated time frames (black boxes as execution time and dotted gray boxes as overall time).

In this example, the preprocessing and data acquisition thread  $H_1$  has a window length  $w_L$  of 6 s time frames with a window overlap  $w_O$  of 2 s time frames. For this scheduling approach, the highest priority must be assigned to  $H_1$ , because it is essential to capture data within fixed and jitter-free time intervals. Hence, the resulting preemptive scheduling allows  $H_1$  to interrupt any other thread. If enough data were collected,

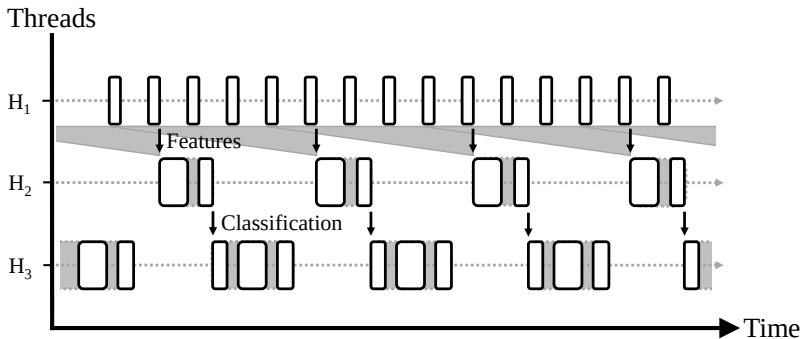


**Figure 3.3:** Periodic scheduling example of a machine learning pipeline. The preprocessing and data acquisition thread  $H_1$  has the highest priority and is able to interrupt any other thread. If enough data were collected (gray triangle-shaped accumulation), the feature extraction thread  $H_2$  is activated, which activates the classification thread  $H_3$ . Both threads are periodically interrupted by thread  $H_1$ .

which is schematically shown by the gray triangle-shaped accumulation in Figure 3.3, the feature extraction thread  $H_2$  will be activated. This thread in turn calls the classification thread  $H_3$  once finished and awaits its reactivation by thread  $H_1$ , when enough data were accumulated once more. After completion of classification and further event processing, thread  $H_3$  is also waiting to be recalled. During this procedure, it is not important, which priorities were attributed to  $H_2$  and  $H_3$ , because both threads have to be finished until  $H_2$  is reactivated.

Using this scheduling approach, an application is only feasible, if the WCET condition (Equation 3.9) can be fulfilled. Therefore, the window length  $w_L$ , the window overlap  $w_O$ , the worst case execution times  $T_{WCET, H_i}$  with  $i \in \{1, 2, 3\}$  for the preprocessing and data acquisition thread  $H_1$ , the feature extraction thread  $H_2$  and the classification thread  $H_3$ , as well as the sampling interval  $T_A$  have to be known.

Additionally, the effective real-time capability can be investigated. Thereby,  $T_{RT}$  (Equation 3.10) is the maximum tolerable response time, which is predefined by the application.

$$w_L - w_O \geq \left( \frac{w_L - w_O}{T_A} \right) \cdot T_{WCET, H_1} + T_{WCET, H_2} + T_{WCET, H_3} \quad (3.9)$$

$$T_{RT} - w_L \geq \left( \frac{w_L - w_O}{T_A} \right) \cdot T_{WCET, H_1} + T_{WCET, H_2} + T_{WCET, H_3} \quad (3.10)$$

$w_L$  = Window length in [s]

$w_O$  = Window overlap in [s]

$T_A$  = Sampling interval in [s]

$T_{WCET, \{H_1, H_2, H_3\}}$  = Worst case execution time of  
thread  $H_1, H_2$  and  $H_3$  in [s]

$T_{RT}$  = Maximum allowable response  
time in [s]

If one of both equations cannot be fulfilled, the microcontroller does not have enough CPU capacity and the developed signal processing task is not executable on the intended hardware platform. In this case the machine learning task has to be optimized regarding CPU requirements or a different CPU has to be used. Another possibility to overcome this problem would include prior knowledge about the application context, which leads to an aperiodic scheduling.

### Aperiodic Scheduling

The aperiodic scheduling approach could be used, if application specific knowledge about the expected timing of incoming events for classification tasks is available in advance or if unpredictable processor requests are possible [Marw 06].

If that is not the case, it must be assumed, that events could arise at any time and the classification thread has to be started repetitively utilizing the above mentioned periodic scheduling. Figure 3.4 shows an example for aperiodic scheduling, whereby the classification thread needs to be executed soonest every second window  $w_L$  (factor  $h_s$  of 2).

If the recognition of an event is realizable without its immediate classification, the recognition task can be moved to the feature extraction thread  $H_2$  and the classification thread  $H_3$  must only be activated if an event is actually detected. The earliest possible recurrence of an event is described by the minimum interarrival time and is shown as an question mark in Figure 3.4. The total execution time prolongs therefore by the factor  $h_s$ . The conditions for the WCET (Equation 3.11) and the RT (Equation 3.12) are accordingly modified. The WCET of  $H_3$  can now

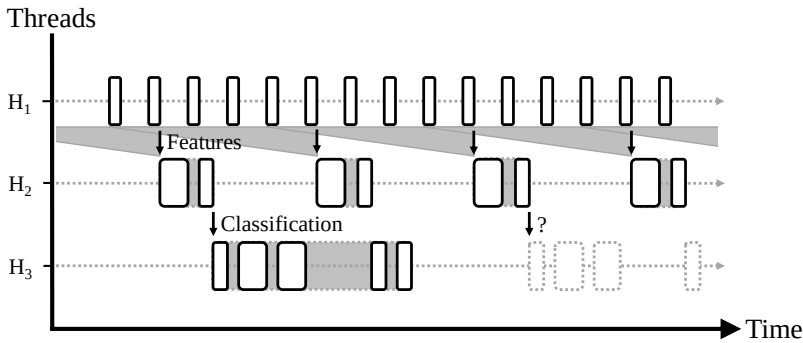


**Figure 3.4:** Aperiodic scheduling example of a recognition task with prior knowledge about the expected timing for incoming classification events. The preprocessing and data acquisition thread  $H_1$  has still the highest priority and is able to interrupt any other thread. Here, the earliest possible interarrival time of the expected event is twice (factor  $h_s$  of 2) the initial window length  $w_L$  and marked as question mark. Only at this time the classification thread  $H_3$  could to be activated, but prolongs therefore  $h_s$ -times longer. All threads can preemptively be interrupted, whereby  $H_3$  has the lowest priority.

last longer and must not be finished until a new feature extraction process is started. A generic real-time analysis for this case would lead to a strong over approximation and can only be done manually. This approach would involve the overall minimum interarrival time instead of the maximum execution time, which depending on the application is often significantly smaller.

$$h_s \cdot (w_L - w_O) \geq h_s \cdot \left( \frac{w_L - w_O}{T_A} \right) \cdot T_{WCET, H_1} + h_s \cdot T_{WCET, H_2} + T_{WCET, H_3} \quad (3.11)$$

$$T_{RT} - h_s \cdot w_L \geq h_s \cdot \left( \frac{w_L - w_O}{T_A} \right) \cdot T_{WCET, H_1} + h_s \cdot T_{WCET, H_2} + T_{WCET, H_3} \quad (3.12)$$

$w_L$  = Window length in [s]

$w_O$  = Window overlap in [s]

$h_s$  = Earliest possible event recurrence factor  
in  $\{h_s : (w_L | h_s)\}$

$T_A$  = Sampling interval in [s]

$T_{WCET, \{H_1, H_2, H_3\}}$  = Worst case execution time of thread

$H_1, H_2$  and  $H_3$  in [s]

$T_{RT}$  = Maximum allowable response time in [s]

Within the table tennis context, the presented aperiodic scheduler is the preferable approach, since strokes occur with alternating interarrival times. The window length  $w_L$  should thereby contain the total signal characteristics of a single stroke event, which has a smaller time period than the recurrence and the associated RT of consecutive table tennis strokes.



## Chapter 4

# Wearable Sensor Development

Wearable sensor development is essential, if information has to be obtained and processed within sports or mobile sensing applications. Each individual application involves different requirements in terms of sensor types, data storage, processing capabilities, communication interfaces, battery runtime, overall sensor size, as well as acceptance and unobtrusiveness. This chapter outlines some common requirements for hardware development of wearable sensors. It introduces the physical basics of the sensor types used in this thesis. Additionally, approaches for sensor calibration and synchronization are presented. Finally, the *miPod* hardware platform is described, which was used for most data acquisitions.

The presented sensor hardware development of the *miPod* platform has already been published in [Blan 14a], in [Blan 14b] and in [Blan 16a] to some extent.

### 4.1 Requirements

The wearable sensor or simply *Wearable* is basically a tiny computer with sensing, storage, processing and communication capabilities. Despite various fields of application, considerations about some main requirements can generally be done for most Wearables, since they necessitate a similar hardware architecture. These requirements comprise following key aspects [Sazo 14]:

- *Sensors Types*

The most important requirement is the basic sensing functionality. Progress in *Micro-Electro-Mechanical Systems* (MEMS) and

nanotechnologies have enabled the development of miniaturized and low-cost sensing components. They can capture electrical, mechanical, chemical or optical data. In a sports science context, sensors normally measure physiological biosignals and motion from or around the human body.

- *Data Storage*

Besides the physical sensing, the accumulation and storage of sensor data for subsequent processing is necessary. This is especially required for more complex computational tasks like real-time machine learning algorithms based on past data or if data are processed at a later stage. Furthermore, data have to be stored if no active communication link is readily available to prevent data loss.

- *Processing Capabilities*

Advances in embedded processor and microcontroller development have allowed for real-time signal processing and machine learning in battery-powered and resource-restricted embedded systems. The simultaneous increase of processing capabilities and reduction of power consumption enabled *Machine Learning* approaches. This is essential to detect and classify events of interest by the Wearable itself.

- *Communication Interfaces*

Sensor data, events or messages from the Wearable are usually delivered to external devices like smartphones, computers or server-based *Clouds* for a more pervasive processing. This can be done wired after data acquisition or instantly wireless using low-energy radio technologies. Due to this radio connectivity, most Wearables are part of a body area network or the *Internet-of-Things* (IoT). Thereby, all captured data can also be viewed as a part of and contribution to the *Big Data* challenge.

- *Runtime and Overall Size*

The total runtime of any Wearable depends on its power consumption and therefore on the used battery's capacity. This in turn determines the overall size of the entire device, since until now the battery is the major limiting factor for size. This contradiction between intended runtime and acceptable sensor size must be considered before new designs are developed.

- *Acceptance and Unobtrusiveness*

The acceptance and unobtrusiveness of a Wearable should not be underestimated. Thus, it is advisable to treat them as important points, which have to be considered during the sensor development. Both aspects could influence the general behavior of the user and therefore affect the data to be measured in an unwanted or even negative way.

In addition to the mentioned requirements, different technical challenges and trade-offs between computation and communication resources have to be tackled to ensure that the Wearable can successfully be applied to its given task [Anli 04]. Thereby, straightforward and intuitive usage, safety aspects, data security and privacy in a social media context [Sazo 14] as well as data reliability, sensor calibration and synchronization [McCa 09] play important roles. Furthermore, the design of a Wearable is also dependent on the intended application. However, commenting on all these attributes would be beyond the scope of this thesis. Therefore, the focus of the following section lies on the sensor types, which are integrated into the *miPod* hardware platform and their respective underlying physical principles.

## 4.2 Sensor Types

All methods in this thesis are mainly based on *Inertial Measurement Units* (IMU), *Geo-magnetic Sensors* and *Piezo-electric Vibration Sensors*. These sensors can be realized in MEMS technology, which combines electric or magnetic forces with mechanical motion as a miniaturized and closed coupled system [Lee 11]. They transform mechanical motion into physical quantities like electric charges, voltages and changes of capacitances or resistances [Jone 13].

### 4.2.1 Inertial Measurement Units

Inertial measurement units are the most common devices for physical motion sensing. Usually, they consist of a three-dimensional accelerometer and three-dimensional gyroscope, resulting in a *6-Degree-of-Freedom* (6-DoF) measurement system.

A coordinate system  $\mathcal{I}$  must be specified to calculate the inertial force  $\mathbf{F}$  and consequently the kinematics of a moving object in free space. It acts as observer for the occurring accelerations and angular velocities.

Thus,  $\mathbf{F} = m \cdot \mathbf{a}$  (Equation 4.1) depends on the acceleration  $\mathbf{a}_0$  and the rotation  $\boldsymbol{\omega}_0$  of the coordinate origin, as well as the mass  $m$ , the position  $\mathbf{r} = \iint \mathbf{a}_0$  and the velocity  $\mathbf{v} = \int \mathbf{a}_0$  of the moving object with respect to  $\mathcal{I}$  [Sazo 14]. It can be described as follows:

$$\begin{aligned} m \cdot \mathbf{a} &= -m \cdot \mathbf{a}_0 + 2 \cdot m \cdot \mathbf{v} \times \boldsymbol{\omega}_0 + m \cdot \boldsymbol{\omega}_0 \times (\mathbf{r} \times \boldsymbol{\omega}_0) + \\ &\quad + m \cdot \mathbf{r} \times \frac{d}{dt} \boldsymbol{\omega}_0 \end{aligned} \quad (4.1)$$

$$\Rightarrow m \cdot \mathbf{a} = -m \cdot \mathbf{a}_0 + 2 \cdot m \cdot \int \mathbf{a}_0 \times \boldsymbol{\omega}_0 + m \cdot \boldsymbol{\omega}_0 \times \left( \iint \mathbf{a}_0 \times \boldsymbol{\omega}_0 \right) +$$

$$+ m \cdot \iint \mathbf{a}_0 \times \frac{d}{dt} \boldsymbol{\omega}_0$$

$\mathbf{a}$  = Acceleration of moving object in  $[m/s^2]$

$m$  = Mass of moving object in  $[kg]$

$\mathbf{r}$  = Position of moving object in  $[m]$

$\mathbf{v}$  = Velocity of moving object in  $[m/s]$

$\mathbf{a}_0$  = Inertial acceleration in  $[m/s^2]$

$\boldsymbol{\omega}_0$  = Inertial angular velocity in  $[rad/s]$

Thereby, the first term on the right side of the equation corresponds to the linear inertial force, the second term defines the force based on the *Coriolis Effect* [Lapl 77], the third term describes the centrifugal force and the last term includes apparent forces. The variables  $\mathbf{a}_0$  and  $\boldsymbol{\omega}_0$  can be measured using an accelerometer and a gyroscope. Regarding Equation (4.1), the characterization of such a moving coordinate system  $\mathcal{I}$  is complicated and demands substantial signal processing and mathematical knowledge [Sazo 14]. For simplicity, the following acceleration and gyroscope descriptions are presented for the one-dimensional case.

## Accelerometers

Accelerometers sense the linear acceleration  $a$  of an attached object. Using MEMS technology, an accelerometer usually measures the acceleration by the displacement  $x_F$  of a seismic proof mass  $m$ , which can only move in a predefined direction inside the enclosure of the sensor. This displacement is related to the acting force  $F_a \sim x_F$ . According to *Newton's Second Law* [Newt 87], the acceleration can be calculated using  $a = \frac{F_a}{m}$ . In practice, the measuring circuit is realized as a system of

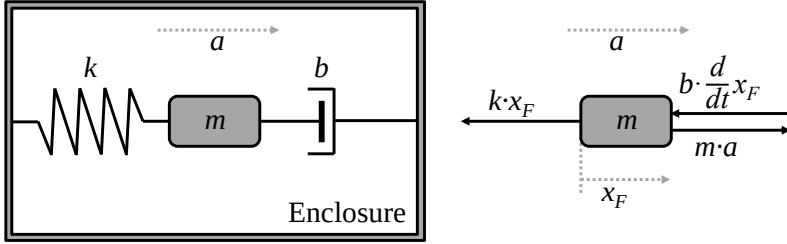


**Figure 4.1:** Mechanical model of a one-dimensional accelerometer according to [Ida 14]. On the left side, the spring-damper system is shown, which is fixed inside the sensor enclosure. On the right side, the free body diagram describes the resulting forces  $F_k = k \cdot x_F$  of the spring,  $F_b = b \cdot \frac{d}{dt} x_F$  of the damper and  $F_a = m \cdot a$  of the acceleration with regards to the proof mass  $m$  and the displacement  $x_F$ .

springs and dampers to suppress unwanted vibration or mass oscillation. A one-dimensional simplified example including its free body diagram can be seen in Figure 4.1. Thereby, a force equilibrium of the spring force  $F_k$  with the spring constant  $k$  and the damper force  $F_b$  with the damper constant  $b$  can be formulated, which is equal to the acting acceleration force  $F_a$  [Ida 14]. This force equilibrium and its related kinematics are shown in Equation (4.2) and in Equation (4.3), respectively.

$$F_a = F_k - F_b \quad (4.2)$$

$$m \cdot a = k \cdot x_F - b \cdot \frac{d}{dt} x_F \quad (4.3)$$

$$F_a = \text{Acceleration force in [N]}$$

$$F_k, F_b = \text{Spring and damper forces in [N]}$$

$$m = \text{Proof mass in [kg]}$$

$$a = \text{Acceleration in [m/s}^2\text{]}$$

$$k = \text{Spring constant in [N/m]}$$

$$b = \text{Damper constant in [Ns/m]}$$

$$x_F = \text{Displacement in [m]}$$

The only unknown within these equations is the displacement  $x_F$ , which can indirectly be derived using different measurement principles. They include piezo-electric effects, whereby small charges are generated by

the deformation of a piezo material, resistive effects due to changes of conductor dimensions or capacitive effects caused by volume changes of the dielectric material of a capacitor.

To realize a three-dimensional accelerometer, three independent mechanical circuits have to be integrated and oriented into one single enclosure perpendicularly. Moreover, the overall motion acceleration  $a$  of a 3D sensor is superimposed component-wise by the gravitational acceleration  $g_{earth}$  of the earth. Consequently, the effective sensed acceleration of an accelerometer is  $a_m = a + g_{earth}$ .

## Gyroscopes

In comparison to the accelerometer, a gyroscope quantifies the angular velocity  $\omega$  of a rotating object. Almost all miniaturized MEMS gyroscopes are based on the Coriolis effect [Kemp 11]. Thereby, the gyroscope includes an active drive axis to couple in a known oscillating force  $F_G(t) = F_0 \cdot \sin(2 \cdot \pi \cdot f_0 \cdot t)$  and a sense axis, which is affected by the Coriolis force  $F_c = 2 \cdot m \cdot \omega \cdot v$ . Here, the movement velocity  $v$  can also be noted as derivative of the displacement  $v = \frac{d}{dt}x_F$ . Both axes are oriented

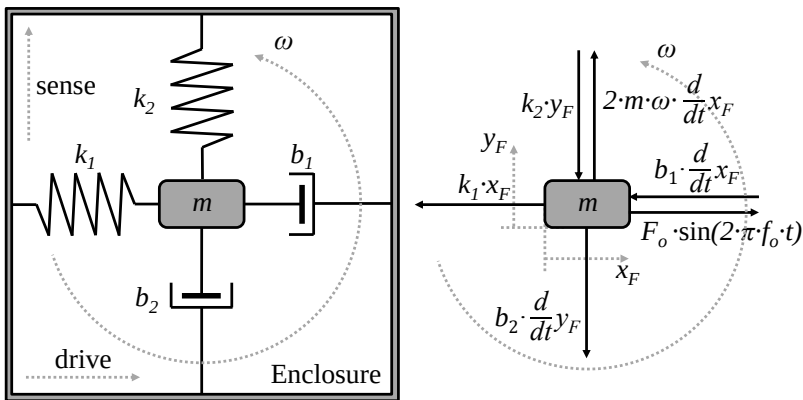


**Figure 4.2:** Mechanical model of a one-dimensional gyroscope according to [Crai 12]. The left side shows the spring-damper systems for the drive axis  $x_F$  and sense axis  $y_F$ . On the right side, the free body diagram describes the resulting forces  $F_{k1} = k_1 \cdot x_F$  and  $F_{k2} = k_2 \cdot y_F$  of the springs,  $F_{b1} = b_1 \cdot \frac{d}{dt} x_F$  and  $F_{b2} = b_2 \cdot \frac{d}{dt} y_F$  of the dampers,  $F(t)$  of the driving force and  $F_c$  of the resulting Coriolis force, caused by the rotation  $\omega$ .

perpendicularly and realized as spring-damper systems. The simplified mechanical model including its free body diagram of the acting forces is shown in Figure 4.2.

Taking all forces into account, a force equilibrium can be noted for the drive axis (Equation 4.4) and for the sense axis (Equation 4.6). They lead to the resulting kinematics described in the terms of (Equation 4.5) and (Equation 4.7), respectively.

$$F_G(t) = F_{k_1} - F_{b_1} \quad (4.4)$$

$$F_o \cdot \sin(2 \cdot \pi \cdot f_o \cdot t) = k_1 \cdot x_F - b_1 \cdot \frac{d}{dt} x_F \quad (4.5)$$

$$F_c = F_{k_2} - F_{b_2} \quad (4.6)$$

$$2 \cdot m \cdot \omega \cdot \frac{d}{dt} x_F = k_2 \cdot y_F - b_2 \cdot \frac{d}{dt} y_F \quad (4.7)$$

$$F_G(t) = \text{Driving force in } [N]$$

$$F_c = \text{Coriolis force in } [N]$$

$$F_{k_{1,2}}, F_{b_{1,2}} = \text{Spring and damper forces in } [N]$$

$$F_o = \text{Maximum driving force amplitude in } [N]$$

$$f_o = \text{Driving frequency in } [Hz]$$

$$k_{1,2} = \text{Spring constants in } [N/m]$$

$$b_{1,2} = \text{Damper constants in } [Ns/m]$$

$$\omega = \text{Angular velocity in } [rad/s]$$

$$x_F, y_F = \text{Displacements in } [m]$$

Both equations can finally be combined into a single differential equation. Similar to the accelerometer, the only unknown is the displacement of the sense axis  $y_F$ . It can be measured with the same approaches. Knowing all other constants, the angular velocity  $\omega$  can eventually be computed. Again, a three-dimensional gyroscope includes three independent mechanical circuits, which are also oriented into one sensor enclosure perpendicularly.

#### 4.2.2 Geo-magnetic Sensor

Supporting the inertial motion devices for movement and orientation determination of an object, geo-magnetic sensors are more often successfully deployed. They extend the inertial measurement systems to a

9-DoF *Inertial-Magnetic Measurement Unit* (IMMU) and enable comprehensive navigation and tracking. However, geo-magnetic sensors as such are not inertial sensors. They measure the natural magnetic field  $B$  of the earth. Due to its low field strength, those sensor devices only work properly in absence of strong artificial magnetic sources and big ferromagnetic structures. For example, power transformers or construction steel inside a building could influence the magnetic field in its strength and orientation. Nevertheless, many sensing methods are still feasible, including the

- *Hall Effect* [Hall 79],
- *Magnetic Tunneling Junction* (MTJ) quantum effect,
- approaches based on the *Giant Magneto-Resistance* (GMR),
- *Anisotropic Magneto-Resistance* (AMR) and
- *Lorentz Force* [Heav 89].

For size and power consumption reasons, most sensors with MEMS technology are based on the Hall effect or the AMR effect [Cai 12]. Both principles are shown in Figure 4.3. The Hall effect arises if a current-carrying metal conductor is exposed to an external magnetic field. Thereby, the

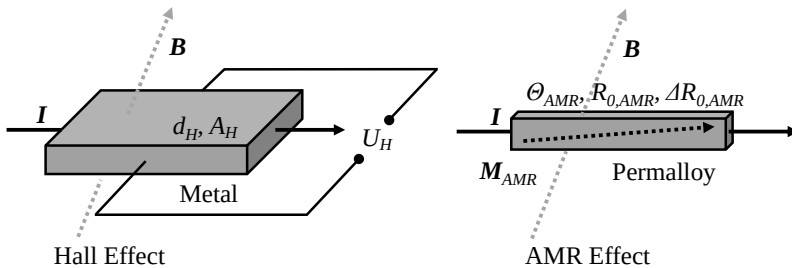


**Figure 4.3:** Example of the Hall effect on the left side and the AMR effect on the right side. If a metal plate with a thickness  $d_H$  and the Hall constant  $A_H$  is exposed to an external magnetic field  $B$ , a proportional voltage  $U_H$  can be measured perpendicular to the current direction  $I$ . In contrast, the AMR effect is based on a magnetization  $M_{AMR}$  of a thin permalloy conductor. Its resistance  $R$  changes, if the angle  $\Theta_{AMR}$  between the current flow  $I$  and the magnetization  $M_{AMR}$  is modified. The magnetization  $M_{AMR}$  is in turn affected by the external magnetic field  $B$ .

Hall voltage  $U_H$  can be measured perpendicular to the current flow  $I$ . It depends on the angle between the flowing current  $I$  and the external magnetic field  $B$ , as well as on material characteristics  $A_H$  and the conductor thickness  $d_H$ . It is mathematically described with Equation (4.8). If the Hall sensor is rotated within the magnetic field and if its initial orientation is known, the output voltage can be used as a position measure for heading or as an electronic compass to point northwards.

$$U_H = \frac{A_H}{d_H} \cdot (I \times B) \quad (4.8)$$

$U_H$  = Hall voltage in [V]

$A_H$  = Hall constant in [ $m^3/C$ ]

$d_H$  = Hall conductor thickness in [ $m$ ]

$I$  = Current flow in [ $A$ ]

$B$  = Magnetic field of the earth in [ $T$ ]

The same information can be obtained utilizing the AMR effect. The total resistance  $R_{AMR}$  of a permalloy conductor is a function of the angle  $\Theta_{AMR}$  between the magnetization  $M_{AMR}$  and the direction of the current flow  $I$  [Leit 11]. If the sensor is exposed to an external magnetic field  $B$ , the magnetization  $M_{AMR}$  changes its direction, thus, affecting the total resistance  $R_{AMR}$  of the permalloy material.

The relation including the permalloy constants  $R_{0,AMR}$  and  $\Delta R_{AMR}$  can be seen in Equation (4.9).

$$R_{AMR} = R_{0,AMR} - \Delta R_{0,AMR} \cdot \sin^2(\Theta) \quad (4.9)$$

$$\Theta_{AMR} = \angle(I, M_{AMR})$$

$R_{AMR}$  = Total AMR resistance in [ $\Omega$ ]

$R_{0,AMR}$  = Maximum AMR resistance in [ $\Omega$ ]

$\Delta R_{0,AMR}$  = AMR resistivity factor in [ $\Omega$ ]

$I$  = Current flow in [ $A$ ]

$M_{AMR} = f(B)$  = AMR magnetization in [ $A/m$ ]

$B$  = Magnetic field of the earth in [ $T$ ]

Similar to the Hall voltage  $U_H$ , the resistance value  $R_{AMR}$  of the AMR effect can be used for orientation detection as an electronic compass

within the geo-magnetic field of the earth. Here again, a 3-dimensional geo-magnetic device must integrate three independent circuits within its enclosure.

### 4.2.3 Piezo-electric Vibration Sensor

In addition to inertial and geo-magnetic sensors for position and orientation determination, piezo-electric devices are applied for impact and vibration detection. The piezo-electric effect describes spatial deformations of a quartz-based piezo material if it is exposed to an external electric field and conversely, electric charges are generated if the piezo material is mechanically deformed [Curi 82]. There are many different piezo materials and possible sensor designs. Within this context, only inverse piezo transducers were used, which are made of a single ceramic material layer implementing one sensitive axis.

As shown in Figure 4.4, an non-static oscillating force  $F_P(t)$  causes vibrations through an impact or shock event on the piezo transducer. It displaces the neutral charge concentration of the material with area  $A_P$  and the thickness  $h_P$  into an electric dipole. A proportional voltage  $U_P(t)$  can be observed [Meas 13]. Since the generated charges are extremely low, the measured voltages have to be amplified by an appropriate circuit. Knowing the direction-dependent piezo strain constant  $d_{33}$ , the relative crystal permittivity  $\epsilon_{33}$ , as well as the vacuum permittivity  $\epsilon_0$ , the generated electric charges  $Q_P(t)$  can be measured as an oscillating voltage  $U_P(t)$  from electrodes applied on the piezo surfaces. The

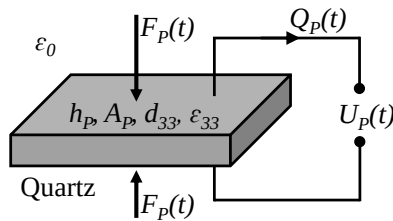


**Figure 4.4:** Example of the inverse piezo-electric effect. An oscillating force  $F_P(t)$  causes a charge generation  $Q_P(t)$  within the piezo material of the thickness  $h_P$ , the area  $A_P$  and the material constants  $d_{33}$ ,  $\epsilon_{33}$  and  $\epsilon_0$  for the piezo strain constant, the relative crystal permittivity and the vacuum permittivity, respectively. The appropriate voltage  $U_P(t)$  can be tapped on the electrodes applied on both piezo surfaces.

mathematical relationship for the generated output voltage is summed up by Equation (4.10). The presented example describes a known one-dimensional case, because the piezo strain constant  $d_{33}$  and the crystal permittivity  $\epsilon_{33}$  are dependent on the material grid structure and on the applied force direction.

A more general consideration of piezo-electric physics modifies the coefficients  $d_{ij} = (\frac{\delta D_{P,i}}{\delta T_{P,j}})^{E_P}$  to derivatives of the electric displacement  $D_P$  and the mechanical stress  $T_P$  during an external electric field  $E_P$  for the piezo grid axes  $i$  and  $j$  [Shar 11]. The crystal permittivity  $\epsilon_{ij}$  must than be adapted in the same way as well.

$$U_P(t) = \frac{d_{33}}{\epsilon_0 \cdot \epsilon_{33}} \cdot \frac{h_P}{A_P} \cdot F_P(t) \quad (4.10)$$

$$\begin{aligned} U_P(t) &= \text{Piezo voltage in [V]} \\ d_{33} &= \text{Piezo strain constant in [C/N]} \\ \epsilon_0 &= \text{Vaccum permittivity in [C/Vm]} \\ \epsilon_{33} &= \text{Relative crystal permittivity} \\ h_P &= \text{Piezo thickness in [m]} \\ A_P &= \text{Piezo area in [m}^2\text{]} \\ F_P(t) &= \text{Applied piezo force in [N]} \end{aligned}$$

## 4.3 Sensor Calibration

If only statistical data or signal characteristics used by machine learning methods are investigated ( $\rightarrow$  see Chapter 2), a precise sensor calibration is of minor importance. As opposed to this, the calibration of sensor data is crucial, if physical quantities have to be computed out of any measured raw values. This is especially true for accelerometers and gyroscopes, since they determine the movement characteristics. Admittedly, many sensors are factory-calibrated, but according to [Hou 04], they suffer from

- *Drifts over time, Offsets and Scaling Errors,*
- *Noise* from electronic or thermal sources,
- *Mechanical Stress and Alignment Errors,* as well as
- *Environmental Influences* like temperature or humidity.

All these factors affect the quality of the sensor output and the signal reliability. Noise and environmental influences are mostly ignored or assumed to be static during the standard calibration processes, whilst drifts, offsets, scaling errors, misalignments and sensitivity errors have to be corrected.

For the following methods, a calibration of the geo-magnetic sensors and the piezo-electric vibration sensors is not necessary, since only relative changes within their signal shapes are analyzed and no absolute measurements of their physical quantities are used. Only the accelerometers and the gyroscopes are calibrated based on an extended approach of [Ferr 95] and [Camp 06]. Thereby, each entire *miPod* device carrying both inertial sensors is mounted into a specific calibration frame to ensure a correct perpendicular alignment. This frame is then set to the six static positions of each sensor axis (+ X, - X, + Y, - Y, + Z, - Z) and rotated certain times around each sensor axis ( $\odot X$ ,  $\odot Y$ ,  $\odot Z$ ), meanwhile data are captured.

Subsequently, the calibration data are used to calculate internal sensor parameters including offsets and biases, scaling and alignment factors, as well as inter-axis sensitivities. Finally, these parameters can be applied to any data set of the same *miPod* sensor platform to compensate the beforehand mentioned inaccuracies [Hofm 14].

## 4.4 Sensor Synchronization

Besides the calibration of sensor data, methods for synchronization between sensors on different platforms are important aspects for the development of Wearables. Synchronization becomes necessary, if data from multiple, but separated sources have to be merged. Two different situations are possible, comprising a

- *Data Synchronization* between sensor devices  
(for example between single *miPod* hardware platforms)
- or *External Synchronization* with any other devices  
(for example cameras for video reference).

Usually, all sensors on one hardware platform are already synchronized, because the implemented microcontroller samples each data channel almost simultaneously within the defined sampling interval.

However, the data synchronization of spatial separated and therefore not tethered hardware is more complex. Thus, imperfect crystals are the

main problem, which actually should provide the timing for data acquisition. These crystal inaccuracies lead to different time deviations for each hardware platform and therefore, the local clocks differ from each other in time and also from a common global time scale [Sivr 04]. This issue can be overcome by a variety of wired and wireless synchronization approaches. For example, global time keeping principles can be applied to maintain an accurate time scale using high-precise

- *Real-Time Clocks* (RTC), the
- *Global Positioning System* (GPS) or the
- internet-based *Network Time Protocol* (NTP).

Wireless methods are mainly build upon synchronizing local clocks within the overall sensor network of more than two devices. According to [Lasa 10], examples are the

- *Reference Broadcast Synchronization* (RBS) [Els02],
- *Timing-Sync Protocol for Sensor Networks* (TPSN) [Gane 03],
- *Tiny-Sync* and *Mini-Sync* [Sich 03], as well as
- *Tree-based Synchronization* algorithms [Van 03].

It is common for all approaches that clock drifts, sampling periods and transmission delays are broadcasted to the entire sensor network. These information are finally used to adapt each local clock providing a uniform time scale for all connected hardware platforms. The algorithms come with differences in terms of accuracy, scalability, energy efficiency or computational costs [Lasa 10].

Besides these methods, synchronization can also be achieved by the sensor data itself [Kule 17]. Fixed movement patterns within the acceleration or gyroscope data of multiple physically linked sensors can be used to align the signal shapes after data acquisition. All studies and methods of this thesis deal with only one single *miPod* sensor, which is attached to the table tennis racket. Hence, there is no need for any data synchronization between multiple sensors. In this context, only the external synchronization between the *miPod* sensor and the video reference is of importance.

## 4.5 *miPod* Sensor Platform

All inertial and geo-magnetic data processed in this thesis are captured with the *miPod* sensor platform, which was specifically developed for the purposes of this thesis. This Wearable includes all necessary components to act as an important tool for miscellaneous sports and fitness applications. It can be unobtrusively integrated into clothes or sports equipment while causing no influences on exercise or performance. There exist two hardware versions, which are intended for different application scenarios. Both hardware platforms feature different electronic components and sensors types within the same polymer case yielding an overall size of about (35 x 25 x 8) mm for length, width and thickness. They are the

- *miPod V1* meant for long-term data acquisition providing offline data processing capabilities and the
- *miPod V2* meant for real-time signal processing and embedded classification of machine learning problems.

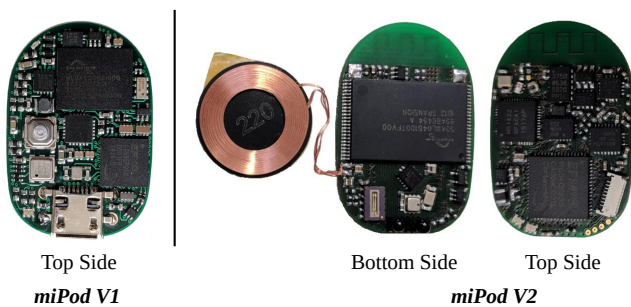


**Figure 4.5:** The left part illustrates all components of the *miPod V1* platform. Microcontroller, sensors, memory and the USB interface are mounted on the top side. The right part shows on the bottom side of the *miPod V2* the memory chip, the real-time clock, the infrared interface and the attached wireless charging coil. The top side contains the microcontroller, all sensors, the power management circuit and the wireless Bluetooth Low Energy® radio with a printed meander antenna.

## miPod V1

The *miPod V1* (published in [Blan14a] and [Blan14b]), which can be seen on the left side of Figure 4.5, contains a *MPU9150* IMU device from *Invensense*<sup>1</sup> including a three-dimensional accelerometer, a three-dimensional gyroscope and a three-dimensional geo-magnetic sensor. Additionally, temperature and barometric pressure can be recorded with the environmental unit *BMP180* from *Bosch Sensortec*<sup>2</sup>. Therefore, combining all data channels results in a 11-DoF measurement system. Precise timing is enabled by the temperature stabilized real-time clock *M41T62* from *ST Microelectronics*<sup>3</sup>. This RTC can be used to enforce an almost constant sampling rate and to synchronize data from different *miPod V1* sensors using the global time keeping approach. The processing core is an energy-friendly and powerful *EFM32LG* Cortex-M3 microcontroller from *Silicon Labs*<sup>4</sup>. All captured data are stored on a 1 GByte NAND flash memory from *Cypress Semiconductor*<sup>5</sup>.

The entire sensor is powered by a lithium-polymer battery with a capacity of 155 mAh. Charging and data transmission is realized by a wired USB connection. The maximum sampling rate can be chosen to up to 1333 Hz, meanwhile the runtime depends on the sensor configuration and the sampling rate and can range from several hours to weeks. The *miPod V1* is designed to capture data with high resolution and sampling rates for use as a scientific research sensor device.

## miPod V2

The newer *miPod V2* (published in [Blan16a]) is built with a *EFM32WG* Cortex-M4 microcontroller from *Silicon Labs*<sup>4</sup>. The onboard floating point unit and the digital signal processor enable real-time data processing and embedded classification capabilities, maintaining high sampling rates. The device integrates the *BMX055* from *Bosch Sensortec*<sup>2</sup> as the main 9-DoF inertial measurement unit and additionally includes the three-dimensional high-range accelerometer *ADXL375* from *Analog Devices*<sup>6</sup> to capture strong impacts in racket sports. In addition, the RTC

<sup>1</sup><https://www.invensense.com/> – last access 06/2018

<sup>2</sup><https://www.bosch-sensortec.com> – last access 06/2018

<sup>3</sup><https://www.st.com/> – last access 05/2018

<sup>4</sup><https://www.silabs.com/> – last access 06/2018

<sup>5</sup><https://www.cypress.com/> – last access 06/2018

<sup>6</sup><http://www.analog.com/> – last access 06/2018

*RV3049* from *Micro Crystal*<sup>7</sup> keeps track of global time and the environmental unit *BME280*, also from *Bosch Sensortec*<sup>8</sup>, captures the relative humidity, air pressure and temperature. Hence, the sensing capacity results in a 15-DoF system. Besides these input sensors, additional peripherals and interfaces support the necessary functionality for signal processing and provide communication to transmit or to store results. Data storage is handled through a 1 GByte NAND flash memory from Cypress Semiconductor<sup>5</sup>, whereas external communication is managed by the *nRF51422* Bluetooth Low Energy® wireless radio chip from *Nordic Semiconductor*<sup>9</sup>. It enables data transmission to external devices and sensor synchronization can be realized implementing the Flooding Time Synchronization Protocol (FTSP) or Tiny-Sync [Kule 17] (→ see Chapter 4.4). Moreover, a high-speed infrared link using the *TFBS4711* from *Vishay*<sup>10</sup> is included to transmit sensor data.

The *miPod V2* is powered by a lithium-polymer battery with a capacity of 110 mAh, which can be wirelessly charged based on the chip *BQ51014B* from *Texas Instruments*<sup>11</sup>, which employs the standardized charging protocol Qi [Van 10]. The *miPod V2* is intended for signal processing and embedded classification on a small and lightweight micro-controller platform.

An overview and comparison of both *miPod* hardware devices is given in Table 4.1. Available sensor types, adjustable sensing parameters, peripherals and communication interfaces are listed for quick reference. The *miPod V2* is designed for a sampling rate up to 200 Hz, which could be problematic in a high dynamical context like table tennis. Therefore, all measured data were acquired with the *miPod V1*, which provides a higher sampling frequency.

---

<sup>7</sup><http://www.microcrystal.com/> – last access 06/2018

<sup>8</sup><https://www.bosch-sensortec.com> – last access 06/2018

<sup>9</sup><http://www.nordicsemi.com/> – last access 06/2018

<sup>10</sup><https://www.vishay.com/> – last access 06/2018

<sup>11</sup><https://www.ti.com/> – last access 06/2018

**Table 4.1:** Overview and comparison of the available sensor types and features for both *miPod V1* and *miPod V2* hardware devices.

| Components               | <i>miPod V1</i>                    | <i>miPod V2</i>                                             |
|--------------------------|------------------------------------|-------------------------------------------------------------|
| Accelerometer            | X, Y, Z<br>± 16 g, 16-bit          | X, Y, Z<br>± 16 g, 12-bit                                   |
| Gyroscope                | X, Y, Z<br>± 2000 deg/s, 16-bit    | X, Y, Z<br>± 2000 deg/s, 16-bit                             |
| Geo-magnetic Sensor      | X, Y, Z<br>± 1200 $\mu$ T, 13-bit  | X, Y @ ± 1300 $\mu$ T, 13-bit<br>Z @ ± 2500 $\mu$ T, 15-bit |
| High-range Accelerometer | n/a                                | X, Y, Z<br>± 200 g, 16-bit                                  |
| Temperature              | -40 ... +85 °C<br>@ 0.1 °C         | -40 ... +85 °C<br>@ 0.01 °C                                 |
| Air Pressure             | 300 ... 1100 hPa<br>@ 0.01 hPa     | 300 ... 1100 hPa<br>@ 0.0018 hPa                            |
| Humidity                 | n/a                                | 20 ... 80 %RH<br>@ 0.008 %RH                                |
| RTC                      | ± 2 ppm (~ 7 ms/h)                 | ± 2 ppm (~ 10 ms/h)                                         |
| Controller               | 32-bit ARM Cortex-M3<br>EFM32LG395 | 32-bit ARM Cortex-M4<br>EFM32WG840                          |
| Memory                   | 1 GByte NAND flash                 | 1 GByte NAND flash                                          |
| USB                      | USB2.0 compliant<br>@ 12 Mbit/s    | n/a                                                         |
| Wireless Radio           | n/a                                | Bluetooth Low Energy®<br>ANT+®                              |
| Infrared Link            | n/a                                | IrDA compliant<br>@ 115.2 kbit/s                            |
| Charging                 | per USB                            | wirelessly per Qi                                           |
| Sampling Rate            | up to 1333 Hz                      | up to 200 Hz                                                |



## Chapter 5

# Stroke Detection and Classification

This chapter deals with the first part of the table tennis analysis described in this thesis. A fully automated stroke detection and classification system was investigated using an inertial measurement unit attached to the end of a table tennis racket handle. Stroke events were computed out of kinematic data and used as input to classify different kinds of standard table tennis strokes. Two scenarios were examined:

- *Exercise Scenario*

The first scenario covers stroke detection and classification in a well-defined environment with predefined exercises and fixed table tennis rallies.

- *Gameplay Scenario*

The second scenario investigates data collected from real table tennis games following official rules. The algorithms in this case are finally used for the feedback device and the embedded implementation (→ see Chapter 8.3).

This chapter is organized in the following sections: firstly, the used hardware equipment and the study design are presented. Secondly, the implementation of stroke detection and classification algorithms are explained. Then, all results are evaluated and discussed. Finally, key points of the chapter are summarized.

The algorithms and acquired data for stroke detection and classification of the exercise scenario have already been published in [Blan15]. New developments for this thesis were comprised of stroke detection and validation as well as stroke types classification for gameplay scenario data of table tennis rallies.

## 5.1 Data Acquisition

All motion data were acquired with the *miPod V1* sensor platform, which was described previously (→ see Chapter 4.5).

### 5.1.1 Hardware Equipment

For the exercise scenario three-dimensional acceleration data and three-dimensional gyroscope data were collected, resulting in a data set of overall 6 dimensions. The accelerometer range was chosen to be  $\pm 16$  g and the gyroscope rate was set to  $\pm 2000$  deg/s. All measurements were obtained at a sampling rate of 1000 Hz and a resolution of 16-bit. Since this was the first attempt to detect and classify strokes in table tennis with an inertial sensor, the high sampling rate was applied to cover all possible highly dynamic motions. During the exercises, all data were stored inside the memory of the *miPod V1* sensor and processed offline afterwards. To avoid misalignments throughout the attachment, the sensor itself was parallelly mounted in acceleration Y-direction at the end of every player's racket handle using a double-sided adhesive tape. The

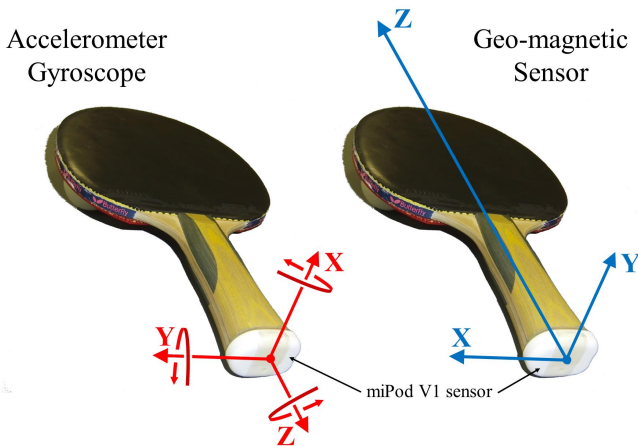


**Figure 5.1:** Axes alignment and coordinate system orientation of the attached *miPod V1* sensor. On the left side, accelerometer and gyroscope orientation are given, the right side shows the orientation of the geo-magnetic sensor (modified from [Blan 15], doi>10.1145/2802083.2802087).

mounting is shown in Figure 5.1. This area has nearly the same shape as the outline of the sensor enclosure, which allowed for correct mounting.

In contrast to the exercise scenario, the game rallies were additionally sampled with three-dimensional data from the geo-magnetic sensor with a range of  $\pm 1200 \mu\text{T}$  and a resolution of 13-bit. However, the sampling rate was reduced to 200 Hz, since the *miPod V1* sensor is not able to provide a higher sampling rate for the integrated geo-magnetic sensor. The gameplay scenario was comprised of a data set of 9 dimensions in total. Table 5.1 summarizes all data acquisition parameters. The main movement plane was spanned by the YZ-plane of the accelerometer and the XZ-plane of the geo-magnetic sensor, respectively. Changes on the spin generation and forehand-backhand differentiation became most distinct in the gyroscope and accelerometer X-axis. An example of the attached sensor, racket, alignment and corresponding axes can be seen in Figure 5.1. For both scenarios, a CASIO® Exilim HS EX-ZR200, as well as a CASIO® Exilim HS EX-ZR300 high speed camera with frame rates of 120 fps and resolutions of 640 x 480 pixels were used as video annotation references for all strokes.

**Table 5.1:** Sensor parameters including sampling rates, measurement ranges and digitalization resolutions for the accelerometer, gyroscope and geo-magnetic sensor during the exercise and gameplay scenarios.

| Sensor Type         | Exercise Scenario                                      | Gameplay Scenario                                     |
|---------------------|--------------------------------------------------------|-------------------------------------------------------|
| Accelerometer       | X, Y, Z @ 1000 Hz<br>$\pm 16 \text{ g}$ , 16-bit       | X, Y, Z @ 200 Hz<br>$\pm 16 \text{ g}$ , 16-bit       |
| Gyroscope           | X, Y, Z @ 1000 Hz<br>$\pm 2000 \text{ deg/s}$ , 16-bit | X, Y, Z @ 200 Hz<br>$\pm 2000 \text{ deg/s}$ , 16-bit |
| Geo-magnetic Sensor | n/a                                                    | X, Y, Z @ 200 Hz<br>$\pm 1200 \mu\text{T}$ , 13-bit   |

### 5.1.2 Study Design

In preparation of every study, the accelerometer and gyroscope of the *miPod V1* sensor were calibrated using a specific algorithm similar to [Ferr 95]. This calibration was repeated every day before the measurement ( $\rightarrow$  see Chapter 4.3). Calibration data and motion data were collectively processed after all studies have been completely finished. For

further processing, synchronization of inertial sensors and video streams was essential. Therefore, the sensor was knocked on a hard surface once before and three times after the measurement. This motion was filmed by the camera and the resulted acceleration peaks were used to overlay the time series signals with the camera frame stream.

### Exercise Scenario

During the exercise scenario, a predefined study was conducted to collect kinematic data of different stroke types from multiple female and male table tennis players. Data were obtained from 14 subjects within an age range of 23 to 49 years. All participants were right handed and used the standard European shakehand grip to hold their racket. All subjects were club players and therefore members of the German national table tennis association DTTB<sup>1</sup>. They represented groups of higher level and lower level players, according to a continuously updated ranking coefficient called TTR<sup>1</sup>, which theoretical ranges from 0 (beginners) to ~ 3000 (World Ranking Leader).

Players with a TTR range from 1287 to 1800 took part in this study. As a result, data with high variabilities in the stroke patterns were collected. Table 5.2 summarizes the subject database for the exercise scenario. Overall, data were collected from 8 basic stroke types: *Drives* (active strokes with forward spin, FD, BD), *Pushes* (chopped balls with backward spin, FP, BP), *Blocks* (passive strokes with forward spin, FB, BB) and *Topspins* (active strokes with strong forward spin, FT, BT). All strokes were performed using the forehand and the backhand. This resulted in a data set of 8 different stroke patterns per subject. The study exercise was organized in such a way that always two players performed subsequently 8 one-minute sub-exercises in a fixed order during one session. This session is shown in Table 5.3. First, a rally of forehand drives

**Table 5.2:** Overview of the exercise scenario study group: number of subjects (#), sex (female F or male M), age, left- and right-handed (left L or right R) and the TTR value.

| #  | Sex        | Age range | Left-/Right-handed | TTR          |
|----|------------|-----------|--------------------|--------------|
| 14 | 4 F / 10 M | 23 to 49  | 0 L / 14 R         | 1287 to 1800 |

<sup>1</sup><http://www.tischtennis.de/> – last access 11/2017

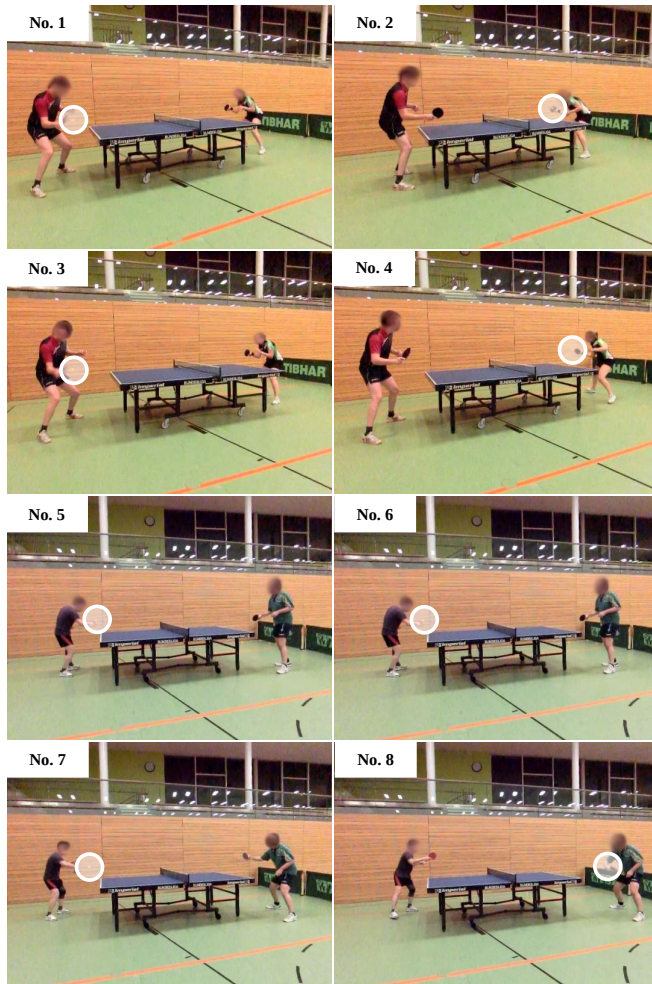


**Figure 5.2:** Sub-exercise examples: (No. 1) shows a forehand drive (FD) and (No. 2) a forehand push (FP) rally. A forehand topspin (FT) from the left player and a forehand block (FB) from the right player are shown in (No. 3). This combination is shown vise-versa in (No. 4). These sub-exercises were repeated in the same manner using the backhand (No. 5) to (No. 8). Here, the active player is marked with a white circle around the ball (modified from [Blan 15], doi>10.1145/2802083.2802087).

**Table 5.3:** Session protocol of an exercise including all sub-exercises (No. 1) to (No. 8): forehand drive, push, topspin, block and backhand drive, push, topspin, block (modified from [Blan 15], doi>10.1145/2802083.2802087).

| No. | Player A              | Player B              |
|-----|-----------------------|-----------------------|
| 1   | (FD) Forehand drive   | (FD) Forehand drive   |
| 2   | (FP) Forehand push    | (FP) Forehand push    |
| 3   | (FT) Forehand topspin | (FB) Forehand block   |
| 4   | (FB) Forehand block   | (FT) Forehand topspin |
| 5   | (BD) Backhand drive   | (BD) Backhand drive   |
| 6   | (BP) Backhand push    | (BP) Backhand push    |
| 7   | (BT) Backhand topspin | (BB) Backhand block   |
| 8   | (BB) Backhand block   | (BT) Backhand topspin |

(FD) was performed by both players (No. 1), then a forehand push (FP) rally (No. 2), followed by a combination of forehand topspins (FT) from player A and blocks (FB) of player B (No. 3) as well as forehand blocks (FB) from player A combined with forehand topspins (FT) from player B (No. 4). Similarly, these 4 sub-exercises were repeated in the same order but using the backhand (No. 5) to (No. 8). This procedure is related to standard warm-up methods in professional table tennis training [McAf09] and can be seen in Figure 5.2. All exercises were documented in a study protocol and recorded on video for later annotation.

### Gameplay Scenario

In comparison to the exercise scenario, which was intended to collect motion data from different well-defined stroke types, the game rally study aimed to capture data from strokes in a real game setting. These are unpredictable motions, since it is a challenge to anticipate spin and speed of the ball as well as the opponent's aimed impact position on the table.

Fur this study, movement data were recorded from 8 female and male subjects within an age range from 19 to 56 years. The participants were right-handers as well as left-handers. They were all club players, used the shakehand grip and their ranking coefficient was between 1363 and 1740. The subject group overview can be found in Table 5.4. In all cases,

**Table 5.4:** Overview of the gameplay scenario study group: number of subjects (#), sex (female F or male M), age, right- or left-handed (left L or right R) and the TTR value. All data of the left-handed subject were transformed by a Z-axis rotation to achieve a uniform coordination system.

| # | Sex       | Age      | Left-/Right-handed | TTR          |
|---|-----------|----------|--------------------|--------------|
| 8 | 2 F / 6 M | 19 to 56 | 1 L / 7 R          | 1363 to 1740 |

two players competed against each other in a game setting of two winning sets. This setup represented a shortened official standard game<sup>2</sup>. Every contact of ball and racket during a rally except serves and hits, which were not involved in the actual gameplay, was analyzed for stroke detection and type classification.

For evaluation, all video data were labeled by a table tennis professional and divided into the following 4 categories: *Forehand Forward* spin strokes (drives FD, blocks FB, topspins FT → FFORE), *Forehand Backward* spin strokes (pushes FP → FBACK), *Backhand Forward* spin strokes (drives BD, blocks BB, topspins BT → BFORE) and *Backhand Backward* spin strokes (pushes BP → BBACK). Two game rally examples can be seen in Figure 5.3. Here again, all games were recorded on video for later annotation.



**Figure 5.3:** Gameplay scenario: during the first exercise (Ex. 1), the left player performs in a forehand topspin stroke (FFORE), whereas the right player intends to block (BFORE). In the second exercise (Ex. 2), the right participant plays a backhand push stroke (BBACK), which the left player intends to return in the same way as forehand chopped ball (FBACK). Again, the active player is marked with a white circle around the ball.

<sup>2</sup><https://www.ittf.com/handbook/> – last access 12/2017

## 5.2 Data Processing

The processing pipelines for exercise and gameplay scenario can generally be split into a *Stroke Detection* part and a *Stroke Classification* part. Firstly, impacts were computed from the raw input inertial data including different pre-processing steps, peak detection algorithms and post-processing methods. Secondly, various features were calculated using a specific data window around each detected stroke. Thirdly, these computed features were used as input for the classification.

In this context, machine learning approaches were investigated during the exercise scenario and evaluated by classification rates and computational efforts. Lastly, the most promising approaches were applied to the gameplay scenario, since these algorithms were intended to run on the embedded system of the instrumented racket (→ see Chapter 8.3).

### 5.2.1 Stroke Detection

The stroke detection pipelines for the exercise and gameplay scenario were done in similar ways using standard signal processing methods. Generally, acceleration peaks caused by ball impacts on rackets were computed from the raw inertial data including different pre-processing steps, peak detection algorithms and post-processing methods.

#### Exercise Scenario

For each subject the entire exercise scenario data were collectively processed. Therefore, the described algorithm had to detect and classify

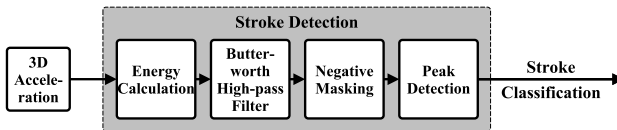


**Figure 5.4:** Stroke detection processing pipeline of the exercise scenario: firstly, the raw three-dimensional acceleration data were used as input for signal energy calculation  $E_{raw}(x_i)$ . Secondly, a Butterworth high-pass filter was applied. Then, a peak detection algorithm searched for prominent maxima  $P(x_i)$  in the negative masked signal as input for the subsequent classification step.

strokes independently from the currently active sub-exercise tasks. An overview of the pipeline can be seen in Figure 5.4.

Firstly, the signal energy  $E_{raw}(x_i)$  (Equation 5.1) was calculated, since it is a commonly used indicator for activity recognition from acceleration data [Bao 04].  $E_{raw}(x_i)$  was computed using the raw acceleration of the X-axis  $a_x(x_i)$ , Y-axis  $a_y(x_i)$  and Z-axis  $a_z(x_i)$ .  $L$  represents the signal length of all data samples  $x_i$  for each subject.

$$E_{raw}(x_i) = a_x(x_i)^2 + a_y(x_i)^2 + a_z(x_i)^2 \quad (5.1)$$

$$0 \leq i < L$$

$L$  = Total signal length

$E_{raw}(x_i)$  = Raw signal energy in  $[g^2]$

$a_{\{x,y,z\}}(x_i)$  = Raw acceleration in  $[g]$

$x_i$  = Data sample

Next, a Butterworth high-pass filter was applied on  $E_{raw}(x_i)$  to emphasize strong and rapid changes of the signal amplitude, thus, highlighting peaks during ball-racket impacts. It was also used to dampen signal artifacts caused by slower arm movements resulting in unwanted lower frequency components. The filter had an order of 1 and a cut-off frequency of 25 Hz. These values were chosen according to [Rodr 02] because of limited execution speeds of arm movements even for the fastest table tennis strokes. As mentioned beforehand, all data were processed simultaneously. This approach allowed to apply the Butterworth filter in a forward and backward manner avoiding group delays.

After filtering, all negative values of  $E_{filt}(x_i)$  were set to zero, which was an input requirement for the subsequent peak detection algorithm, resulting in a negative masked signal  $S(x_i)$  (Equation 5.2).

$$S(x_i) = \begin{cases} E_{filt}(x_i), & \text{if } E_{filt}(x_i) > 0 \\ 0, & \text{otherwise} \end{cases} \quad (5.2)$$

$S(x_i)$  = Negative masked signal in  $[g^2]$

$E_{filt}(x_i)$  = Filtered signal energy in  $[g^2]$

$x_i$  = Data sample

The peak detection method, which was derived from [Pals 09], was based two parameters: the threshold value  $\theta$  (Equation 5.3) and the segmentation range  $\kappa_d$  (Equation 5.4). The threshold value  $\theta$  for the peak detection depends on the mean value  $\mu$  and  $h$ -times the standard deviation  $\sigma$  for each subject calculated from  $S(x_i)$ . The unit-less factor  $h$  was experimentally calculated with data from all subjects and set to 1.9 for all further computations ( $\rightarrow$  see Chapter 5.3). The segmentation range  $\kappa_d$  is defined by  $\kappa_{dl}$  and  $\kappa_{dr}$ , which characterize the distance between two consecutive strokes for one player. They were set equally to 250, representing a total segmentation range of 0.5 seconds.

$$\theta = \mu + h \cdot \sigma \quad (5.3)$$

$$\kappa_d = [x_{i-\kappa_{dl}}, \dots, x_{i+\kappa_{dr}}] \quad (5.4)$$

$$\theta = \text{Threshold in } [g^2]$$

$$\mu, \sigma = \text{Mean and standard deviation in } [g^2]$$

$$h = \text{Empirical factor}$$

$$\kappa_{dl} + \kappa_{dr} = \text{Size of segmentation range } \kappa_d$$

$$x_i = \text{Data sample}$$

Now, peaks  $P(x_j)$  (Equation 5.5) could be detected, if the local maximum  $H(x_j)$  was equal to  $S(x_j)$  and if additionally  $S(x_j)$  exceeded the threshold value  $\theta$  [Pals 09].

$$P(x_j) = \begin{cases} 1, & \text{if } H(x_j) = S(x_j) \wedge S(x_j) > \theta \\ 0, & \text{otherwise} \end{cases} \quad (5.5)$$

$$H(x_j) = \max\{S(x_{j-\kappa_{dl}}), \dots, S(x_{j+\kappa_{dr}})\}$$

$$i + \kappa_{dl} \leq j < L - \kappa_{dr}$$

$$P(x_j) = \text{Peak signal}$$

$$L = \text{Total signal length}$$

$$\kappa_{dl} + \kappa_{dr} = \text{Size of segmentation range } \kappa_d$$

$$H(x_j) = \text{Local maximum signal in } [g^2]$$

$$S(x_j) = \text{Neagtive masked signal in } [g^2]$$

$$\theta = \text{Peak detection threshold in } [g^2]$$

$$x_j = \text{Data sample}$$

Afterwards, the post-processing retained only those peaks  $\hat{P}(x_i)$  within the signal length  $L$  of all peaks in  $P(x_j)$ , at which  $S(x_j)$  exceeded the threshold  $\theta$  since the last valid peak occurred outside the current range  $\kappa_d$ . This was done to avoid multiple occurrences  $P(x_j)$  for one effective stroke. Example data for each processing step are shown in Figure 5.5.

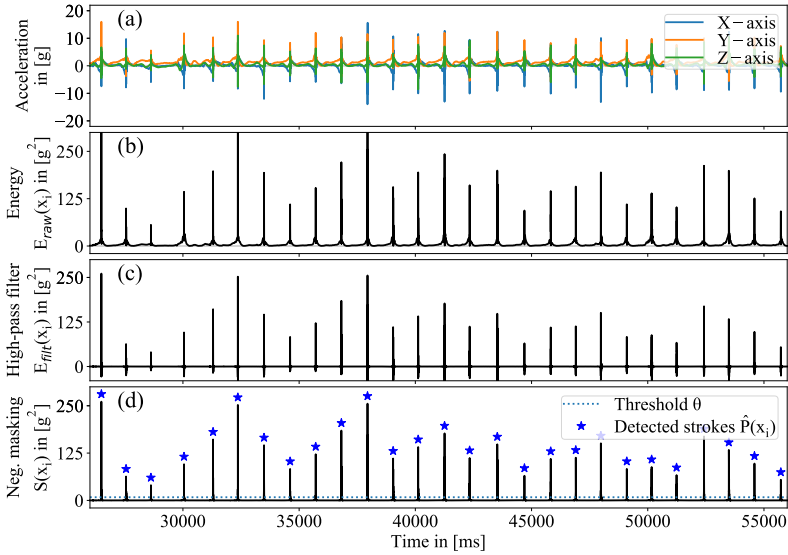


**Figure 5.5:** Example stroke detection data of the exercise scenario: (a) raw 3D-acceleration data of one player, (b) signal energy, (c) high-pass filter with an order of 1 and a frequency of 25 Hz, (d) negative masking and detected peaks  $\hat{P}(x_i)$  (blue stars). The peak detection threshold  $\theta$  is shown with a blue dotted line (modified from [Blan 15], doi>10.1145/2802083.2802087).

## Gameplay Scenario

In comparison to the previously mentioned exercise scenario pipeline, the gameplay data were differently processed, which can be seen in Figure 5.6. Any computations were only initiated by a positive indication of an external trigger. Different signal sources, for example, impact detection using sound or vibration sensors are possible indicators. Here, the piezo-electric shock sensors ( $\rightarrow$  see Chapter 7.2) were used as initiation for the final stroke detection pipeline. Before detected strokes could

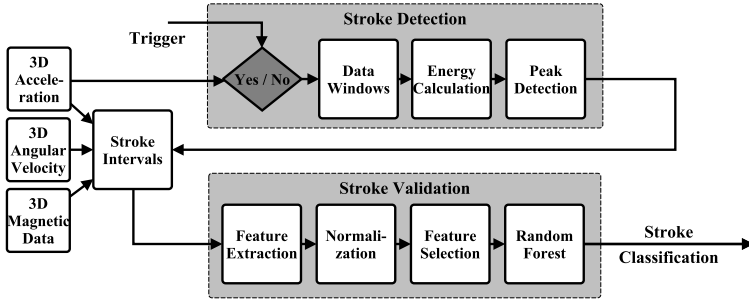


**Figure 5.6:** Stroke detection processing pipeline for the gameplay scenario: computations were only initiated by a positive indication of an impact trigger from outside. Acceleration data were used for the signal energy calculation  $E_{raw}(w_i)$ . Afterwards, a simple peak detection method extracted single strokes  $P(x_i)$ , which were passed to a stroke validation process. Thereby, features were extracted, normalized and selected as input for a Random Forest classifier to validate the current ball-racket impact.

be passed to the stroke classification, a *Stroke Validation* as an intermediate step had to be performed. The stroke validation is needed to filter out data from unwanted ball-racket impacts like services or hits that are not part of the game rally.

The raw acceleration for the energy signal  $E_{raw}(w_i)$  (Equation 5.6) were windowed. According to [Boot 90], an forehand drive takes in total less than 0.25 seconds. Thus, every window  $w_i$  had a length of 0.75 s, to cover the duration of all strokes.

$$E_{raw}(w_i) = a_x(w_i)^2 + a_y(w_i)^2 + a_z(w_i)^2 \quad (5.6)$$

$$0 + \frac{w_T}{2 \cdot T_A} \leq i < L - \frac{w_T}{2 \cdot T_A}$$

$$E_{raw}(w_i) = \text{Raw signal energy in } [g^2]$$

$$L = \text{Total signal length}$$

$$a_{\{x,y,z\}}(w_i) = \text{Raw acceleration in } [g]$$

$$w_T = \text{Window length in } [s]$$

$$T_A = \text{Sampling interval in } [s]$$

$$w_i = \text{Data window}$$

All windows  $w_i = [i - \frac{w_T}{2 \cdot T_A}, \dots, i + \frac{w_T}{2 \cdot T_A}]$  were formed by continuously going through the data from  $0 \leq i < L$  in steps of  $i = \frac{w_T - w_O}{T_A}$  with a constant window overlap of  $w_O$  of 0.25 s. Based on the signal energy  $E_{\text{raw}}(w_i)$ , a rough peak detector searched for the signal maximum  $H(w_j)$  (Equation 5.7) within the related window  $w_j$ , which is equivalent for the likeliest impact peak  $P(w_j)$  and stroke event candidate.

$$P(w_j) := H(w_j) = \max\{E_{\text{raw}}(w_j)\} \quad (5.7)$$

$$i - \frac{w_T}{2 \cdot T_A} \leq j < i + \frac{w_T}{2 \cdot T_A}$$

$$P(w_j) = \text{Peak signal in } [g^2]$$

$$H(w_j) = \text{Local maximum signal in } [g^2]$$

$$E_{\text{raw}}(w_j) = \text{Raw signal energy in } [g^2]$$

$$w_T = \text{Window length in } [s]$$

$$T_A = \text{Sampling interval in } [s]$$

$$w_j = \text{Data window}$$

Again, post-processing steps left only one peak  $\hat{P}(x_i)$  representing the stroke event of all peaks in  $P(w_j)$  within all overlapping frames  $w_j$ . Figure 5.7 illustrates example data of the gameplay scenario. Before the

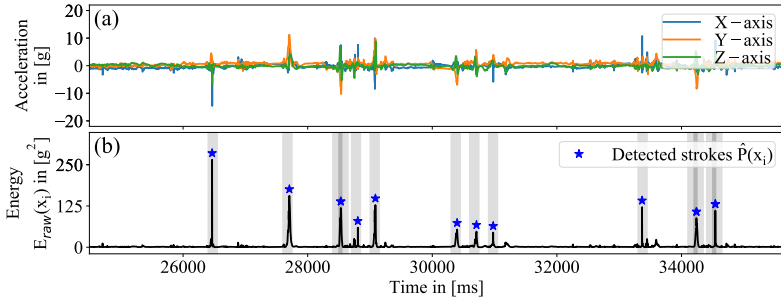


**Figure 5.7:** Example stroke detection data of the gameplay scenario: (a) raw acceleration data from 11 random strokes of one player (but not necessarily from one rally) and (b) the computed signal energy including detected peaks  $\hat{P}(x_i)$  (blue stars). Additionally, light gray boxes show the appropriate data windows  $w_j$ , whereas dark gray boxes indicate a stroke within 2 overlapping windows.

detected strokes  $\hat{P}(x_i)$  from the gameplay scenario data could be passed to the stroke classification, the stroke validation step had to be performed. Similar to the window generation  $w_j$ , intervals  $\kappa_v$  were formed based on  $\hat{P}(x_i)$  including data of the accelerometer, gyroscope and additionally of the geo-magnetic sensor. These intervals were defined in the range  $\kappa_v = [i - \kappa_{vl}, \dots, i + \kappa_{vr}]$  with  $\kappa_{vl}$  of 600 and  $\kappa_{vr}$  of 400, similar to the segmentation range  $\kappa_d$  of the peak detection algorithm. The interval represented a total time of 1 second. This data interval was moved more to past data, since pre-impact countermovements reveal more meaningful information about the performed stroke type than post-impact data [Rodr02]. Subsequently, the generated data intervals were used for the feature extraction. Multiple features were calculated from all sensor axes for every data interval.

According to Table 2.1 ( $\rightarrow$  see Chapter 2.2), multiple features were computed. Since many classification methods need a determined range distribution of the their input data, the feature vector  $q_V$  was normalized between -1 and 1 (Equation 5.8). The minimum and maximum values of the corresponding single features were extracted from the appropriate sensor axis over the whole data length  $L$ .

$$\hat{q}_V = 2 \cdot \left( \frac{q_V - \min\{x_i\}}{\max\{x_i\} - \min\{x_i\}} \right) - 1 \quad (5.8)$$

$$0 \leq i < \text{Total data length } L$$

$q_V$ ,  $\hat{q}_V$  = Raw and normalized feature vector

$x_i$  = Data sample

After the normalization, a feature selection was performed to reduce the computational effort of the later embedded implementation. An overall amount of 90 values, formed by nine features times nine sensor axes, plus nine correlation features  $((9 \times 9) + 9 = 90)$  had to be computed. This was done by access to the relative importance  $\Psi_{RI}$  ( $\rightarrow$  see Chapter 2.2) of each feature with respect to the target class, here *Stroke* or *No Stroke* of the following Random Forest (RF) classifier. For the gameplay scenario data, three different feature selection thresholds  $\theta_{\Psi_{RI}}$  were used: a naive solution using all features  $\Psi_{RI} > \theta_{\Psi_{RI}} = 0$ , a light selection criteria using all features above the mean  $\Psi_{RI} > \theta_{\Psi_{RI}} = \mu_{\Psi_{RI}}$  and a heavy selection criteria using those features above the mean added with three times the standard deviation  $\Psi_{RI} > \theta_{\Psi_{RI}} = (\mu_{\Psi_{RI}} + 3 \cdot \sigma_{\Psi_{RI}})$ . In the

end, the actual validation step was done by an odd number of decision trees  $n_{trees} \in \{3, 7, 11, 15\}$  to achieve a unique decision and a fixed depth  $d_{tree}$  of 50. Finally, the majority vote of all decision trees decided for or against performing a subsequent classification of the stroke type.

### 5.2.2 Stroke Classification

Following the stroke detection and validation steps, the final processing part used these computed features as input to machine learning methods for the stroke type classification.

In this context, multiple approaches were investigated during the exercise scenario and evaluated by their classification rates and related computational efforts. Lastly, the most promising approaches were applied to the gameplay scenario, since these algorithms and methods were intended to run on the embedded microcontroller system of the instrumented racket ( $\rightarrow$  see Chapter 8.3).

#### Exercise Scenario

The exercise scenario classification pipeline can be seen in Figure 5.8. Based on  $\hat{P}(x_i)$ , specific intervals  $\kappa_c$  were created and defined in a range of  $\kappa_c = [x_{i-\kappa_{cl}}, \dots, x_{i+\kappa_{cr}}]$ , similar to the stroke validation during the gameplay scenario. The parameters  $\kappa_{cl}$  and  $\kappa_{cr}$  were set to 600 and 400, respectively. Compared to stroke detection, acceleration and angular velocity data were used. Multiple features taken from Table 2.1

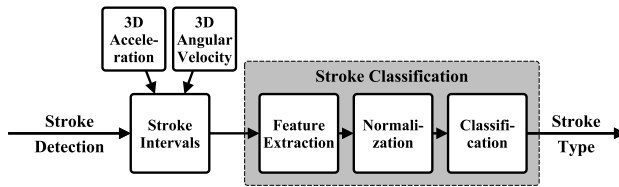


**Figure 5.8:** Stroke classification pipeline for the exercise scenario: first, data intervals were created using three-dimensional acceleration, three-dimensional angular velocity and the peak occurrences from the former stroke detection step. Then, multiple features were extracted, normalized and fed to the final stroke type classification.

(→ see Chapter 2.2) were calculated from all six sensor axes for every data interval  $\kappa_c$ .

In contrast to the gameplay scenario features, the normalization in this case was done between 0 and +1 (Equation 5.9) to satisfy the input requirements of all following investigated classifiers. The minimum and maximum values of the corresponding features were similarly extracted from the appropriate sensor axis over the whole data length  $L$ . Overall, 60 values consisting of nine features times six axes plus six correlation features  $((9 \times 6) + 6 = 60)$  were extracted.

$$\hat{q}_V = \frac{q_V - \min\{x_i\}}{\max\{x_i\} - \min\{x_i\}} \quad (5.9)$$

$$0 \leq i < \text{Total data length } L$$

$q_V, \hat{q}_V$  = Raw and normalized feature vector

$x_i$  = Data sample

In the end, different classifiers were trained using the *WEKA* data mining software [Hall 09] and the *Embedded Classification Toolbox* (ECST) [Ring 12]. These tools provide the training phases for multiple classification systems and can evaluate their accuracies. The ECST can also analyze the necessary computational effort of a classification system. Six different classification approaches were tested and compared using the exercise scenario data: a statistical *Naïve Bayes* (NB) classifier, an ensemble-based *Random Forest* (RF) classification approach with  $n_{trees} \in \{1, 2, 5, 10, 15, 25, 50, 75, 100\}$ , a *Support Vector Machine* (SVM) as supervised learning model with linear (LIN) and radial basis function kernels (RBF), the *k-Nearest Neighbors* (kNN) algorithm with  $k_{neighbors} \in \{1, 2, 3, 4, 5, 6, 7, 8\}$  and *PART* [Eibe 98] as a rule-based classifier. For both SVM approaches, cost parameter  $c \in \{1, 10, 100, 1000\}$  and  $\gamma \in \{0.001, 0.01, 0.05, 0.1\}$  were optimized using a grid search in an inner cross validation loop.

## Gameplay Scenario

The stroke type classification of the gameplay exercise data can be seen in Figure 5.9. If the validation step indicated a true stroke, a special SVM called  $\nu$ -SVM [Scho 00] was applied. The  $\nu$ -SVM appeared to perform best in the application environment of classifying table tennis strokes (→ see Chapter 5.3). An additional advantage of the  $\nu$ -SVM is, that the

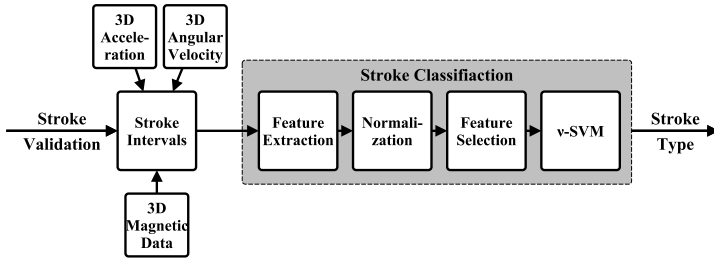


**Figure 5.9:** Stroke classification pipeline of the gameplay scenario data: firstly, data intervals were created using three-dimensional acceleration, three-dimensional angular velocity, three-dimensional geo-magnetic data and the peak occurrences  $\dot{P}(x_i)$  from the stroke detection step. Features calculated for stroke validation were selected and used as input for a  $\nu$ -SVM classifier.

parameter  $\nu$ , which is used instead of the  $c$ -parameter, does control the final number of support vectors. Thus, it can simplify the embedded implementation by reducing memory consumption and computational time. Additionally, a feature selection using the relative importance was investigated. In this case, the needed threshold  $\theta_{\Psi_{RI}}$  was evaluated in more detail with an additional parameter of  $\lambda \in \{0, 0.125, 0.25, 0.5, 1, 3\}$  in  $\theta_{\Psi_{RI}} = (\mu_{\Psi_{RI}} + \lambda \cdot \sigma_{\Psi_{RI}})$ . Here, only radial basis function (RBF) kernels with  $\gamma \in \{0.001, 0.01, 0.05, 0.1\}$  were assessed.

### 5.3 Evaluation and Results

The presented results for stroke detection and classification have already been published in [Blan 15]. New within this thesis are results for stroke detection, validation and classification of all gameplay scenario data. The video annotation and labeling for stroke detection and classification was done manually by a table tennis expert with over 20 years of experience for the exercise and gameplay scenarios. Labels included serves, hits, which did not fit to the related sub-exercises, failed hits or other random ball-racket impacts

All detected strokes were compared with all stroke labels within an empirical defined validation range of  $\pm 0.05$  seconds. This was sufficient for the chosen synchronization method between video and IMU data.

The stroke detection algorithms wwere described with *Precision*, *Recall* and the combined harmonic mean *F1-Score* ( $\rightarrow$  see Chapter 2.3). All classifiers were evaluated with a performance assessment including mean and class-dependent *Accuracies*, overall classification rates and partly with estimations for complexity and computational effort. All evaluations were computed with a LOSO-CV.

5.3.1 Stroke Detection

Exercise Scenario

Table 5.5 shows the total numbers of all labeled and detected strokes from all subjects for the exercise scenario. In total, 4154 ball contacts were labeled, whereof 2821 strokes were counted during every single sub-exercise (FD, FP, FT, FB, BD, BP, BT and BB) as valid strokes. In contrast to the labels, 3988 ball contacts were counted by the peak detection algorithm, whereof 2802 were identified as true positives. All these true positive values were subsequently used as input for the evaluation of the classification.

Furthermore, different peak detection thresholds  $\theta = \mu + h \cdot \sigma$  were evaluated using multiple  $h$ -values between 0 and 10 in steps of 0.1. The results for  $\theta$  are shown in Figure 5.10. Best outcomes were achieved at a

**Table 5.5:** List of all labeled and detected strokes for the exercise scenario. They were split into each sub-exercise category. Some strokes were easier to play than others during the sub-exercises. Overall, 2821 ball contacts were labeled, whereof 2802 were detected (modified from [Blan15], doi>10.1145/2802083.2802087).

| Stroke Type | Labeled Strokes | Detected Strokes |
|-------------|-----------------|------------------|
| FD          | 528             | 528              |
| FP          | 301             | 298              |
| FT          | 410             | 409              |
| FB          | 330             | 329              |
| BD          | 431             | 428              |
| BP          | 288             | 282              |
| BT          | 311             | 307              |
| BB          | 222             | 221              |
| Total       | 2821            | 2802             |

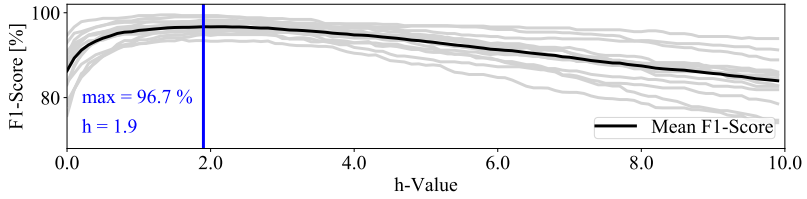


**Figure 5.10:** Threshold dependent F1-Scores (gray lines) for all subjects with the mean value (black line). Varying thresholds  $\theta = \mu + h \cdot \sigma$  within the range  $[0 ; 10]$  of  $h$  in steps of 0.1 resulted in a maximum F1-Score of 96.7 % at a  $h$ -value of 1.9 (modified from [Blan 15], doi>10.1145/2802083.2802087).

$h$ -value of 1.9 resulting in a mean F1-Score of 96.7 %, which led to mean subject-independent Precision and Recall values of 94.9 % and 98.6 %, respectively.

### Gameplay Scenario

The stroke detection of the gameplay data was evaluated with 7001 windows containing 607 labeled strokes, whereof 536 were detected. This can be seen in Table 5.6.

For stroke validation, different estimations of Random Forests (RF) were analyzed using multiple numbers of trees  $n_{trees} \in \{3, 7, 11, 15\}$ , one fixed tree depth  $d_{tree}$  of 50 and three different feature selection methods: a naive solution using all features  $\Psi_{RI} > \theta_{\Psi_{RI}} = 0$ , a light selection criteria using those features above the mean  $\Psi_{RI} > \theta_{\Psi_{RI}} = \mu_{\Psi_{RI}}$  and a heavy selection criteria using those features above the mean added with three

**Table 5.6:** List of all labeled and detected strokes of every player for the gameplay scenario. Overall, 607 valid ball contacts were labeled, whereof 536 were detected.

| Stroke Type | Labeled Strokes | Detected Strokes |
|-------------|-----------------|------------------|
| FFORE       | 206             | 187              |
| FBACK       | 23              | 13               |
| BFORE       | 231             | 207              |
| BBACK       | 147             | 129              |
| Total       | 607             | 536              |

**Table 5.7:** Stroke validation evaluation of the gameplay scenario data. Precision, Recall and F1-Score are given in [%] for different numbers of trees  $n_{trees}$  and feature selection thresholds  $\theta_{\Psi_{RI}}$ . Gray boxes indicate the best performing algorithm with the chosen number of features  $Q$ .

| Number of<br>Trees $n_{trees}$ | $\Psi_{RI} > 0$ (naive)                                              |               |            |              |
|--------------------------------|----------------------------------------------------------------------|---------------|------------|--------------|
|                                | $Q$                                                                  | Precision [%] | Recall [%] | F1-Score [%] |
| 3                              | 90                                                                   | 79.9          | 95.3       | 86.9         |
| 7                              | 90                                                                   | 87.3          | 96.4       | 91.6         |
| 11                             | 90                                                                   | 88.8          | 96.8       | 92.6         |
| 15                             | 90                                                                   | 89.1          | 97.8       | 93.2         |
| Number of<br>Trees $n_{trees}$ | $\Psi_{RI} > \mu_{\Psi_{RI}}$ (light)                                |               |            |              |
|                                | $Q$                                                                  | Precision [%] | Recall [%] | F1-Score [%] |
| 3                              | 20                                                                   | 76.4          | 93.4       | 84.0         |
| 7                              | 22                                                                   | 81.5          | 97.4       | 88.7         |
| 11                             | 20                                                                   | 86.5          | 99.2       | 92.4         |
| 15                             | 19                                                                   | 88.3          | 99.8       | 93.7         |
| Number of<br>Trees $n_{trees}$ | $\Psi_{RI} > (\mu_{\Psi_{RI}} + 3 \cdot \sigma_{\Psi_{RI}})$ (heavy) |               |            |              |
|                                | $Q$                                                                  | Precision [%] | Recall [%] | F1-Score [%] |
| 3                              | 4                                                                    | 78.1          | 89.6       | 83.5         |
| 7                              | 2                                                                    | 84.2          | 92.9       | 88.3         |
| 11                             | 5                                                                    | 86.5          | 98.1       | 91.9         |
| 15                             | 4                                                                    | 89.6          | 97.3       | 93.3         |

times the standard deviation  $\Psi_{RI} > \theta_{\Psi_{RI}} = (\mu_{\Psi_{RI}} + 3 \cdot \sigma_{\Psi_{RI}})$ . The results are shown in Table 5.7. Precision, Recall and F1-Score in percent are listed depending on the number of used trees  $n_{trees}$  and the three different feature selections thresholds  $\theta_{\Psi_{RI}}$ . The best classifiers are individually highlighted in gray. The selected features can be seen in Table 5.8.

### 5.3.2 Stroke Classification

#### Exercise Scenario

The results for stroke classification can be seen in Table 5.9. Overall accuracies and total computational costs are listed individually for each

**Table 5.8:** Feature selection results for the gameplay data of the stroke validation for light (19 features) and heavy (4 features) feature selection. Mainly the standard deviation as statistical moment and different signal characteristics were chosen as most significant features.

| Axis      | Feature Selection Thresholds $\theta_{\Psi_{RI}}$            |                                                                         |
|-----------|--------------------------------------------------------------|-------------------------------------------------------------------------|
|           | $\Psi_{RI} > \mu_{\Psi_{RI}}$<br>(light)                     | $\Psi_{RI} > (\mu_{\Psi_{RI}} + 3 \cdot \sigma_{\Psi_{RI}})$<br>(heavy) |
| $X_{Acc}$ | $q_{max}, q_{\sigma}, q_{iqr}, q_{egy}$                      | $q_{\sigma}$                                                            |
| $Y_{Acc}$ | $q_{max}, q_{\mu}, q_{\sigma}, q_{iqr}, q_{egy}, q_{median}$ | $q_{max}, q_{\sigma}$                                                   |
| $Z_{Acc}$ | $q_{max}, q_{\sigma}, q_{iqr}, q_{egy}$                      | $q_{\sigma}$                                                            |
| $X_{Gyr}$ | —                                                            | —                                                                       |
| $Y_{Gyr}$ | $q_{min}$                                                    | —                                                                       |
| $Z_{Gyr}$ | $q_{min}, q_{\sigma}, q_{iqr}, q_{egy}$                      | —                                                                       |

**Table 5.9:** Overall classification accuracies and computational efforts (low, middle and high) of the exercise scenario regarding the number of executed operations. The RBF SVM classifier with cost parameter  $c$  of 10 and  $\gamma$  of 0.001 was identified as best performing classifier with an overall accuracy of 95.9 %. The linear SVM performed best with a  $c$ -value of 10, the RF with 50 trees  $n_{trees}$  and a tree depth  $d_{tree}$  of 20 the kNN with  $k_{neighbors}$  of 1. The least computational effort was identified with the PART algorithm with only 236 comparisons in total (modified from [Blan 15], doi>10.1145/2802083.2802087).

| Method          | NB     | RF   | SVM    |        | kNN    | PART |
|-----------------|--------|------|--------|--------|--------|------|
|                 |        |      | LIN    | RBF    |        |      |
| Accuracy [%]    | 87.7   | 95.5 | 95.3   | 95.9   | 94.4   | 89.0 |
| Effort          | middle | low  | middle | high   | high   | low  |
| $+, -$          | 472    | 0    | 1680   | 270720 | 162517 | 0    |
| $*, \div$       | 3776   | 0    | 1652   | 142228 | 165318 | 0    |
| $e^x, \sqrt{x}$ | 944    | 0    | 0      | 2294   | 2802   | 0    |
| $\leq$          | 7      | 1050 | 35     | 35     | 2808   | 236  |

classification method. The best performance was achieved with the RBF kernel of a SVM using  $c$  as 10 and  $\gamma$  as 0.001 with an accuracy of 95.9 %. One of the lowest computational efforts regarding mathematical oper-

**Table 5.10:** Confusion matrix of the RBF SVM classifier of the exercise scenario. Rows represent the labeled strokes, classified strokes are indicated by columns. On the left side, the class-dependent classification is broke down for each stroke type (modified from [Blan 15], doi>10.1145/2802083.2802087).

| Accuracy [%] | Stroke Type | FD  | FP  | FT  | FB  | BD  | BP  | BT  | BB  |
|--------------|-------------|-----|-----|-----|-----|-----|-----|-----|-----|
| 98.9         | FD          | 522 | 0   | 5   | 1   | 0   | 0   | 0   | 0   |
| 100.0        | FP          | 0   | 298 | 0   | 0   | 0   | 0   | 0   | 0   |
| 97.8         | FT          | 8   | 0   | 400 | 0   | 0   | 1   | 0   | 0   |
| 91.8         | FB          | 25  | 0   | 0   | 302 | 0   | 2   | 0   | 0   |
| 94.9         | BD          | 0   | 0   | 0   | 0   | 406 | 0   | 17  | 5   |
| 98.9         | BP          | 0   | 3   | 0   | 0   | 0   | 278 | 0   | 0   |
| 97.4         | BT          | 0   | 0   | 0   | 0   | 8   | 0   | 298 | 0   |
| 83.2         | BB          | 0   | 3   | 0   | 0   | 34  | 0   | 0   | 183 |

ations were provided by a RF with 50 trees  $n_{trees}$ , a tree depth  $d_{tree}$  of 20 using 1050 comparisons and gaining an accuracy of 95.5 %. In contrast, the PART algorithm needed only a quarter of comparisons (here 236), but had clearly worse accuracy. Additionally, the class-dependent classification rates and the confusion matrix of the best performed classifier (here, RBF SVM) can be seen in Table 5.10. Labeled strokes and detected strokes were compared by corresponding columns and rows for each class-dependent decision. The forehead push (FP) stroke type classification achieved the best accuracy with 100 %.

### Gameplay Scenario

The evaluation of the gameplay scenario data combined all stroke types into four classes: FFORE, FBACK, BFORE and BBACK. The overall classification rates of different  $\nu$ -SVM kernels depending on the applied feature selection criteria can be seen in Table 5.11. The best results were achieved with a RBF  $\nu$ -SVM using no feature selection with an accuracy of 95.9 %. The parameters were thereby set to 0.1 for both parameters  $\nu$  and  $\gamma$ . Furthermore, the class-dependent classification rates for the RBF  $\nu$ -SVM can be seen in Table 5.12. Again, labeled and detected stroke

**Table 5.11:** Stroke classification evaluation of the RBF  $\nu$ -SVM kernel and feature selection criteria of the gameplay scenario data. The parameters for the RBF kernel were set to 0.1 for both parameters  $\nu$  and  $\gamma$ . Additionally, the maximum number of selected features  $Q$  are listed. Best performed the RBF  $\nu$ -SVM with 95.9 % using all 90 features.

| $\nu$ -SVM Kernel | Accuracy [%] for $\theta_{\Psi_{RI}} = (\mu_{\Psi_{RI}} + \lambda \cdot \sigma_{\Psi_{RI}})$<br>$\lambda =$ |      |       |      |      |      |      |
|-------------------|-------------------------------------------------------------------------------------------------------------|------|-------|------|------|------|------|
|                   | $\Psi_{RI} > 0$                                                                                             | 0    | 0.125 | 0.25 | 0.5  | 1    | 3    |
| RBF               | 95.9                                                                                                        | 77.6 | 85.8  | 72.8 | 56.0 | 52.9 | 41.5 |
| $Q$               | 90                                                                                                          | 34   | 31    | 28   | 22   | 13   | 3    |

**Table 5.12:** Confusion matrix of the RBF  $\nu$ -SVM classifier of the gameplay scenario. Rows represent the labeled strokes classes, classified stroke categories are indicated by columns. On the left side, the class-dependent classification is broke down for each stroke category.

| Accuracy [%] | Stroke Category | FFORE | FBACK | BFORE | BBACK |
|--------------|-----------------|-------|-------|-------|-------|
|              |                 | FFORE | FBACK | BFORE | BBACK |
| 96.6         | FFORE           | 199   | 0     | 7     | 0     |
| 100.0        | FBACK           | 0     | 23    | 0     | 0     |
| 95.2         | BFORE           | 3     | 0     | 220   | 8     |
| 95.2         | BBACK           | 1     | 0     | 6     | 140   |

categories were compared by corresponding columns and rows for each class-dependent decision. In this case, forehand backward spin strokes (FBACK) were best classified with 100 %, even if only few hits were performed during the gameplay in comparison to other types.

## 5.4 Discussion

In general, all results show that it is possible to detect and classify different stroke types in table tennis with overall high accuracies using the above mentioned algorithms and methods. This is also confirmed

by the results of previous investigations for stroke detection and classification within other racket sports by [Rama 14] and [Sali 10] for badminton or [Tien 08] and [Conn 11b] for tennis. Additionally, the results are especially competitive to previous table tennis analysis, regardless of whether camera-based ([Sore 01] and [Oldh 15] with 95 %) or sensor-based ([Guo 10], [Boye 13] or [Zhan 06] with ~ 81 % methodologies were applied (→ see Chapter 9.1).

### 5.4.1 **Stroke Detection**

During the exercise scenario, 4154 ball-racket contacts of 14 both female and male club players within eight different sub-exercises were collected. However, only 2821 strokes could be labeled as intended strokes during the sub-exercises (FD, FP, FT, FN, BD, BP, BT and BB). One reason was, that some strokes were wrongly performed during the sub-exercises (for example forehand stroke instead of a backhand stroke). Furthermore, serves, netballs, edgeballs and ball-racket contacts with no rally character were also excluded. The actual stroke detection algorithm could detect 3988 of 4154 strokes with a mean Precision of 94.9 % and a mean Recall of 98.6 %, respectively (see Table 5.10). In total, 2821 strokes were part of related sub-exercises, whereof 2802 strokes were detected.

However, these good results were achieved by a study design proposed to detect and classify strokes in pre-defined sessions with a fixed order and duration of sub-exercises. Within these sub-exercises, consecutive strokes with similar and periodical motion sequences had to be performed. In addition, all data were processed at once per subject using all true positives, true negatives and false positives and therefore an evaluation about potential false negatives was not possible. Considering the type-dependent numbers of every stroke type (see Table 5.5), some strokes were performed with higher reliability than others. Standard and most common strokes (e.g. 528 x FD) occurred more often than passive strokes (e.g. 301 x FP) or strokes, which are more complex in a correct movement execution (e.g. 311 x BB). Players normally try to score and to prevent own errors during a match, so that difficult strokes were less correct executed and expedient strokes more often played.

The evaluation of the gameplay scenario data was based on 607 ball-contacts of eight female and male players within 7001 windows. Every window was analyzed by the Random Forest algorithm, checking if a valid stroke of one of the four specified categories (FFORE, FBACK, BFORE and BBACK) occurred or not. The good results of > 93 % for the

F1-Score were achieved with an increasing number of trees  $n_{trees}$  and the chosen feature selection thresholds  $\theta_{\Psi_{RI}}$  for light and heavy selection criteria (see Table 5.7). Even for only 4 single features, the F1-Score of 93.3 % was as high as the F1-Score of 93.2 % with the total feature set 90 features.

A detailed view on the selected features could explain this behavior (see Table 5.7). Mainly the standard deviation feature and some signal characteristics of the accelerometer were selected as most important indicators. They describe the signal shape in the time-domain and can therefore be used to estimate rapid and strong changes during a ball-racket contact. These impacts contain sharp edges and peaks, which are not present during miscellaneous arm movements.

However, these overall good results were only feasibly using an external trigger system (see Figure 5.6), which indicated a potential stroke by detecting its ball-racket impact. Without this trigger system, every window  $w_j$  would contain a signal energy maximum  $H(w_j)$ , no matter if a stroke happened or not. These maximums would wrongly be assigned to an impact. This would lead to false positives of the stroke detection resulting in a low precision and, Also the classification would fail, since many wrongly detected strokes would be classified into one of the defined categories.

### 5.4.2 Stroke Classification

The stroke type classification of the exercise scenario performed best with a the RBF kernel of a SVM reaching an overall mean accuracy of 95.9 % (see Table 5.7). Nevertheless, this type of classification method comes along with high computational efforts in terms of mathematical operations and memory consumption. Regarding an embedded implementation, which is finally done using the approaches present for the gameplay scenario, other classifiers with lower accuracies but less resource requirements become more advantageous. Promising candidates are rule or decision tree based systems like PART and the Random Forest. The RF showed a negligible worse classification rate with 99.5 % but needed only fast executable comparisons.

A deeper view into the proposed classification system can be seen in (see Table 5.8). Stroke type dependent classification rates and misclassifications are listed in a confusion matrix. Most incorrect classified strokes are observed for blocks, which have similar stroke patterns to corresponding drives (FD  $\longleftrightarrow$  FB and BD  $\longleftrightarrow$  BB). Some players perform

blocks in a more active way than other players. It depends on the one hand on the skill level and ability of the player himself and on the other hand on the spin and speed of the approaching ball. Thus, active blocks and passive drives lead to nearly identical signals regarding the raw inertial input data. In contrast, the differentiation between forehand and backhand strokes (FD, FP, FT, FB  $\longleftrightarrow$  BD, BP, BT, BB), as well as the spin distinction (topspin  $\longleftrightarrow$  backspin) show no such issues. This might be caused due to the fundamental movement changes of the total racket motion during hitting. According to Figure 5.1, rotation around the gyroscope X-axis and impact peaks within the accelerometer X-axis mostly determine the differentiation between forehand and backhand strokes. Conversely, the moving YZ-plane of the accelerometer specifies more or less the spin generation. However, the spin direction of the approaching ball must therefore be identical, but should be given within one sub-exercise. Consequently, top down movements in positive Y-direction create backspin, while bottom up movements indicate topspin.

In contrast to the exercise scenario, stroke types of the gameplay scenario were classified using multiple  $\nu$ -SVM evaluated with RBF kernels only. For this scenario, a RF algorithm was not investigated, since it would have needed a high memory consumption for the decision trees ( $\rightarrow$  see Chapter 3.1), which could not have been implemented on an embedded device. Best accuracy with 95.9 % was reached by the RBF  $\nu$ -SVM without any feature selection (see Table 5.11). With this configuration, results are similar to the exercise scenario data. However, varying the total numbers of features, the classification accuracy rapidly dropped under 50 %. This behavior could be explained by the summarized stroke categories FFORE, FBACK, BFORE and BBACK of the game rallies. Players were no longer able to perform regular movement patterns for each stroke type, because not expectable and unpredictable motion components induced different kinds of motion artifacts. This could also be confirmed by the selected features. No prominent features were identified by the relative importances feature selection  $\Psi_{RI}$ . They were all randomly distributed, whereas the feature selection for the stroke detection still picked out the mainly standard deviation feature and statistical moments.

Furthermore, there are no clearly misclassified stroke types similar to the exercise scenario regarding the class-dependent classification (see Table 5.12). Interestingly, mixed-up stroke types were mostly observed between otherwise clearly separable forehand and backhand forward

spin strokes (FFORE  $\leftrightarrow$  BFORE) as well as forward and backward spin types of the backhand (BFORE  $\leftrightarrow$  BBACK). This issue is still hard to interpret, since the same classification method of the exercise scenario showed for this situation a better evaluation. One explanation could be the player's intention of performance and execution of a gampelay stroke, which can influence the quality of the playing ability [Raab 05]. During critical situations (e.g. a fast approaching ball, a wrongly anticipated spin rotation or an unexpected impact position on the table), the player had to change his original intended stroke or to perform some kind of emergency stroke. This could lead to irregular and untrained signal shapes within the inertial sensor data and finally to the observed misclassification. In this case, a closer investigation into the counter-movement data of the stroke intervals  $\kappa_c$  could bring more details or information forward to estimate such movement components. Additionally, higher level information about the speed and the spin of the ball could reveal more about the underlying stroke type ( $\rightarrow$  see Chapter 6).

### 5.4.3 Improvements

Further improvements could be done by a closer combination of stroke detection and classification. Currently, only valid table tennis strokes were considered for stroke detection and only true positives were used as input for the subsequent classification step. Introducing a *Null* stroke type category could detect ball-racket events, which must not be necessarily assigned to a specific exercise or a labeled stroke. This could be done by adding machine learning methods to the stroke detection pipeline as it was partly done for the gameplay scenario.

Apart from that, strokes could be directly classified without any impact detection beforehand. The consequence would probably a lower overall accuracy, since a null class could consist of various input signals, even similar to already defined stroke types. For instance, any spin direction can be applied both with forehand and backhand during a serve, which makes it hard to separate other stroke types from a serve. In this case, increasing the data interval  $\kappa_c$  to even more past data allowing an analysis of longer countermovement periods would be certainly beneficial.

Another improvement would be to extend the stroke type categories by *Chops* (FC, BC) and *Strikes* (FSt, BSt) as well as the spin categories with a third *Side Spin* (FSIDE, BSIDE) category. However, different challenges would come up, since pure side spin can not physically occur as

the only spin type of a single stroke [Rodr02]. There must always be a combination of topspin and side spin or backspin and side spin. This makes a categorization of different stroke types by their spin components difficult.

## 5.5 Summary

In this chapter, an automated stroke detection and classification pipeline for different kinds of table tennis strokes was presented. Motion data of racket-mounted inertial sensors were acquired in research studies of two different scenarios. During the exercise scenario, inertial data were collected by table tennis players performing different stroke types within short sub-exercise. Additional, data were obtained in a gameplay scenario study, where subjects performed rallies in matches with official rules.

The stroke detection method for the exercise scenario consisting of signal energy calculation, filtering and a threshold based peak detection achieved an overall Precision of 94.9 % and a Recall of 98.6 %. Based on the detected strokes, multiple features (signal characteristics, statistical moments and heuristic features) were extracted and different kinds of classifiers tested. Best results were given by a RBF-based SVM and a mean classification rate of 95.9 %.

The stroke detection of the gameplay scenario was implemented using a Random Forest approach. Best classification rates > 98 % were achieved by using a relative importance based feature selection method. The subsequent classification with a  $\nu$ -SVM and a RBF kernel yielded equally mean accuracy of 95.9 %. All results showed that is possible to detect and classify different kinds of strokes in the context of a table tennis application using machine learning algorithms with overall high accuracies.

## Chapter 6

# Ball Spin and Speed Estimation

The following chapter deals with an approach for ball speed and spin estimation used in the table tennis feedback device. It is based on a single racket-mounted IMU sensor. With the help of different assumptions and simplifications of the initial speed and spin of the approaching ball, kinematic data are used to estimate the speed and spin of the ball shortly after impact.

The next sections are organized as follows: firstly, the video labeling tools for the speed and spin estimations of balls on high-speed camera frames are described and evaluated for the usage in this scenario. Secondly, all motion assumptions, simplifications and data acquisitions are explained. Thirdly, data are processed using the racket speed calculation and by specific rebound models. Then, all results are evaluated and discussed. Finally, key points of the chapter are summarized.

The algorithms for ball spin and speed estimation as well as the video labeling and its evaluation have already been published in [Blan 17].

### 6.1 Video Data Labeling

A suitable reference for evaluation of ball speed and spin values had to be found by extracting ball properties from video camera frames during ball-racket impact scenes to achieve reliable results. The maximum ball speed and spin of a standard 40 mm table tennis ball can reach up to 30 m/s and  $\pm 800$  rad/s [McAf09]. Therefore, cameras with frame rates of  $> 600$  fps for ball speed measurement and  $> 254$  fps for ball spin measurement are necessary to achieve resolutions of  $\pm 0.1$  m/s and  $\pm 1$  turns/s, respectively. These resolution values are assumed to be suitable as reference in the context of ball speed and spin estimation using

inertial sensors. Unfortunately, this comes along with low pixel resolutions and thus with narrow fields of view. A high-end professional camera equipment is costly and requires complex data processing.

However, a ball spin and speed estimation algorithm presented in [Szep11] was modified and extended to extract ball properties out of camera frame streams. Firstly, the estimated ball speed  $v_b$  was computed by a pixel displacement of the ball center vectors  $\mathbf{p}_A$  and  $\mathbf{p}_B$  during two consecutive frames  $A$  and  $B$ . Since the ball speed  $v_b$  was assumed to be constant within the period of two frames, the pixel displacement was also assumed to be constant and was therefore averaged 10 times over eleven consecutive frames  $N_{vid}$ . Additionally, the pixel width  $p_{w,i}$  (Equation 6.1) was needed during each video frame and was calculated by 4 manually selected envelopes  $M_{vid}$  around the ball image area  $A_b$  for every single frame  $i$ .

$$p_{w,i} = \frac{1}{M_{vid}} \cdot \sum_{j=1}^{M_{vid}} \sqrt{\frac{\pi}{A_{b,i,j}}} \quad (6.1)$$

$$1 \leq i < \text{Number of frames } N_{vid}$$

$$1 \leq j < \text{Number of ball envelopes } M_{vid}$$

$$p_{w,i} = \text{Pixel width in } [1/m]$$

$$A_{b,i,j} = \text{Ball envelope area in } [m^2]$$

Knowing the ball radius  $r_b$  with 0.02 m and the sampling interval  $T_A$  with 0.001 s between two frames due to the 1000 Hz sampling rate of the high-speed camera, the ball speed  $v_b$  (Equation 6.2) can be described as follows:

$$v_b = \frac{r_b}{N_{vid} \cdot T_A} \cdot \sum_{i=1}^{N_{vid}} \left\| \mathbf{p}_{B,i} - \mathbf{p}_{A,i} \right\| \cdot p_{w,i} \quad (6.2)$$

$$1 \leq i < \text{Number of frames } N_{vid}$$

$$v_b = \text{Ball speed in } [m/s]$$

$$r_b = \text{Ball radius in } [m]$$

$$T_A = \text{Sampling interval in } [s]$$

$$\mathbf{p}_{A,i}, \mathbf{p}_{B,i} = \text{Ball center vectors in } [m]$$

$$p_{w,i} = \text{Pixel width in } [1/m]$$

Secondly, the ball spin was similarly calculated using the difference of consecutive feature point vectors  $\mathbf{f} = \mathbf{f}_B - \mathbf{f}_A$  and the ball translation vectors  $\mathbf{p} = \mathbf{p}_B - \mathbf{p}_A$  within two video frames. A feature point could be any prominent marking on the ball surface, which is clearly trackable during all consecutive frames  $N_{vid}$ . In this experiment, the intersection of a painted cross was used as prominent feature point. The frame-wise rotation of the ball was defined by a rotation angle  $\alpha_{rot}$  (Equation 6.3) between the resulting vector and the coordinate system axes.

$$\alpha_{rot,i} = \arccos \left( \frac{(\mathbf{f}_{A,i} - \mathbf{p}_{A,i}) \cdot (\mathbf{f}_{B,i} - \mathbf{p}_{B,i})}{\|\mathbf{f}_{A,i} - \mathbf{p}_{A,i}\| \cdot \|\mathbf{f}_{B,i} - \mathbf{p}_{B,i}\|} \right) \quad (6.3)$$

$$\omega_b = \frac{1}{N_{vid} \cdot T_A} \cdot \sum_{i=1}^{N_{vid}} \text{sgn} \left( (\mathbf{f}_i - \mathbf{p}_i) \times (\mathbf{f}_{A,i} - \mathbf{p}_{A,i}) \right) \cdot \alpha_{rot,i} \quad (6.4)$$

$$1 \leq i < \text{Number of frames } N_{vid}$$

$$\alpha_{rot,i} = \text{Rotation angle in } [rad]$$

$$\omega_b = \text{Ball rotation in } [rad/s]$$

$$T_A = \text{Sampling interval in } [s]$$

$$\mathbf{f}_{A,i}, \mathbf{f}_{B,i} = \text{Feature point vectors in } [m]$$

$$\mathbf{p}_{A,i}, \mathbf{p}_{B,i} = \text{Ball center vectors in } [m]$$

$$\mathbf{f}_i = \text{Feature point translation vector in } [m]$$

$$\mathbf{p}_i = \text{Ball center translation vector in } [m]$$

An example video frame is shown in Figure 6.1. The ball rotation  $\omega_b$  was given by equation (6.4) with respect to the related spin direction.

Due to the fact, that other software and camera equipment with different sampling rate parameters than suggested in [Szep 11] was used, an evaluation method of this proposed reference system had to be made. Here, a pendulum approach was applied, which can be seen in Figure 6.2. The pendulum consisted of a ball bearing mounted rod  $l_{pend}$  with a length of 0.45 m, which was rotated at its end by a DC gear motor from Maxon Motor<sup>1</sup> with the motor constant  $D_{mot}$  of 35.5 turns/Vs and the gear ratio  $G_{mot}$  of 4.4. A table tennis ball with multiple feature markings was precisely mounted at the gear axis to generate an accurate rotation

<sup>1</sup><https://www.maxonmotor.ch> – last access 12/2017

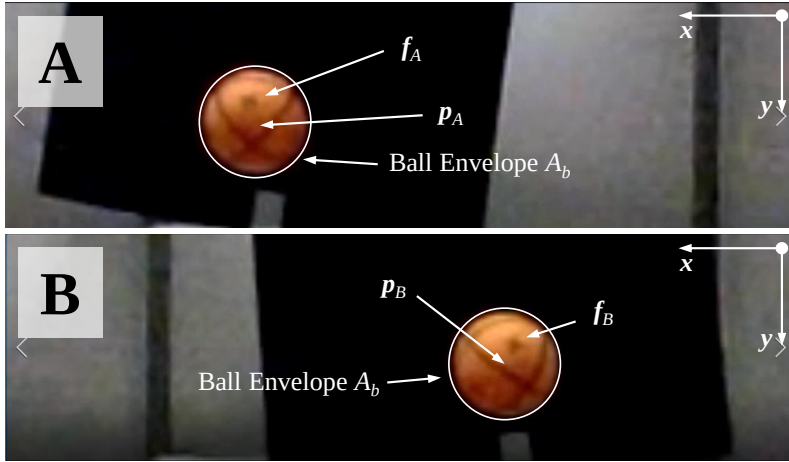


**Figure 6.1:** Example high-speed video data of the labeled ball of two frames A and B, with resolutions of 224 x 64 pixels (→ see Chapter 6.2). The surface of the ball was marked with cross and line features.  $p_A$  and  $p_B$  are the center of the mapped ball envelope  $A_b$  (white circle).  $f_A$  and  $f_B$  can be any prominent trackable feature point (here: cross on the upper surface side). All envelopes and vectors were labeled manually by a single expert to ensure a uniform data analysis (modified from [Blan 17], doi>10.1145/3123021.3123040).

and angular velocity  $\omega_{pend}$  (Equation 6.5) of the ball by variation of the DC motor supply voltage  $U_{mot}$ .

$$\omega_{pend} = 2 \cdot \pi \cdot \frac{1}{G_{mot}} \cdot \frac{D_{mot}}{60} \cdot U_{mot} \quad (6.5)$$

$\omega_{pend}$  = Angular velocity of pendulum in [rad/s]

$G_{mot}$  = Gear ratio constant

$D_{mot}$  = Motor constant in [1/Vs]

$U_{mot}$  = Motor voltage in [V]

Additionally, the linear speed  $v_{pend}$  (Equation 6.6) of the ball could be calculated at the turning point of the pendulum knowing the rod deflection angle  $\epsilon_{pend}$ , the total mass of the pendulum  $m_{pend}$  with 0.614 kg, the total inertia moment of the pendulum  $J_{pend}$  with  $50.988 \cdot 10^{-3} \text{ kgm}^2$  and the gravitational constant  $g_{earth}$  of  $9.80665 \text{ m/s}^2$ .

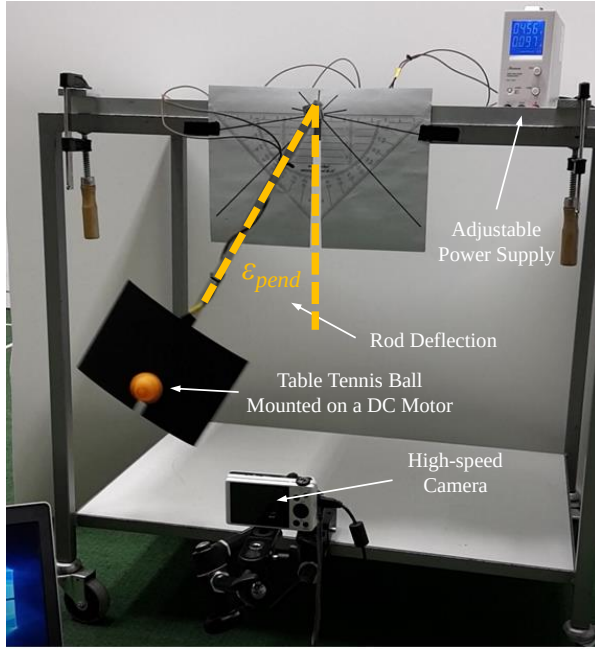


**Figure 6.2:** Experimental pendulum system for the video labeling reference. A high-speed camera was mounted in front of the pendulum rod at the lowest point. The ball rotation was generated by a DC gear motor controlled by an adjustable power supply. The rod deflection angle  $\epsilon_{pend}$  could be measured manually (modified from [Blan 17], doi>10.1145/3123021.3123040).

$$v_{pend} = \sqrt{\frac{2 \cdot l_{pend}^2 \cdot m_{pend} \cdot g_{earth} \cdot [l_{pend} - \cos(\epsilon_{pend}) \cdot l_{pend}]}{l_{pend}^2 \cdot m_{pend} + J_{pend}}} \quad (6.6)$$

$v_{pend}$  = Linear pendulum speed in  $[m/s]$

$l_{pend}$  = Length of pendulum rod in  $[m]$

$m_{pend}$  = Total mass of pendulum in  $[kg]$

$g_{earth}$  = Gravitational constant  $[m/s^2]$

$\epsilon_{pend}$  = Rod deflection in  $[rad]$

$J_{pend}$  = Total pendulum inertia moment in  $[kgm^2]$

In this case, the total inertia moment of all objects with mass was calculated using the *Steiner's Theorem* [Haas 29] comprising the pendulum rod, the motor, the ball, the black contrast sheet, the cables and the mounting equipment. Applying different linear speeds and angular velocities, a meaningful evaluation for both low and high values as well as for rotations  $\omega_{pend}$  with 0 rad/s could be achieved. The proposed video labeling method obtained overall relative errors  $f_v$  of - 2.3 % and  $f_\omega$  of - 8.4 % for ball speed and ball spin, respectively. A detailed view on the results can be seen in Figure 6.3.

With a rod deflection  $\epsilon$  of 60 deg resulting in a linear ball speed  $v_{pend}$  of 1.74 m/s, ball spin and speed were estimated by the video labeling method. Both underestimated errors could be explained by inadequate modeling of the inertia moment components  $J_{pend}$  or by possible neglected friction parts at the point of fixation. The standard deviation increases along with rising pendulum ball spin. The high values can be explained by the large pixel width  $p_w$  caused by the low resolution of the high-speed camera. In contrast, the applied ball speed and ball spin was always well reflected in the mean values. Reliable results were ensured by labeling every scene 20 times and averaging these measurements. The results of the video labeling methodology were in the range of the underlying original contribution [Szep 11] and even in the ranges of other mentioned video-based approaches in [Tama 04] and [Zhan 15].

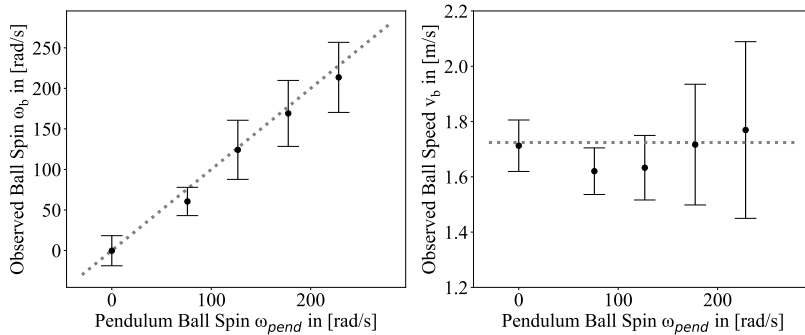


**Figure 6.3:** Evaluation data of the pendulum validation approach for a ball speed  $v_{pend}$  of 1.72 m/s. Observed ball spin and ball speed values are compared for different ball spin values of the pendulum. The dotted gray lines indicate ideal results. For all measurements, the overall mean values of ball spin and speed resulted in relative errors of  $f_v$  with - 2.3 % and  $f_\omega$  with - 8.4 %, respectively.

Compared to these methods, the presented approach is applicable as a reference system for ball speed and spin estimation on video frames in the context of table tennis.

## 6.2 Data Acquisition

### 6.2.1 Hardware Equipment

All data of the following methods were acquired with the *miPod V1* inertial sensor platform ( $\rightarrow$  see Chapter 4.5). For all measurements, a triaxial accelerometer and a triaxial gyroscope were used. In total, this resulted in a six-dimensional data set for each player and stroke movement. Here, the accelerometer range was set to  $\pm 16$  g and the gyroscope rate to  $\pm 2000$  deg/s, respectively. All data were recorded at a sampling rate of 1000 Hz and digitized with a resolution of 16-bit. This ensured that all motion patterns could be captured properly. An overview of the sampling parameters can be seen in Table 6.1.

One single IMU sensor was attached to a racket prototype, which can be seen in Figure 6.4. Therefore, a cavity with a minimal smaller dimension than the sensor's outline was milled into the handle's end ensuring a tight fit. During all measurements, the main racket movement was represented by data of the XY-plane, whereas spin distinction and rotation could mainly be detected by data of the Z-axis. Furthermore, the racket consisted of a *XIOM<sup>2</sup> Classic Allround S* blade and two identical *JOOLA<sup>3</sup> energy X-tra* rubbers. Both rubbers had a thickness of 2.0 mm.

**Table 6.1:** Sensor parameters including sampling rates, measurement ranges and resolutions for the accelerometer and the gyroscope of the ball spin and speed estimation.

| Sensor Type   | Sensor Parameter                                       |
|---------------|--------------------------------------------------------|
| Accelerometer | X, Y and Z axis<br>@ 1000 Hz, $\pm 16$ g, 16-bit       |
| Gyroscope     | X, Y and Z axis<br>@ 1000 Hz, $\pm 2000$ deg/s, 16-bit |

<sup>2</sup><https://www.xiom.tt/> – last access 01/2018

<sup>3</sup><http://www.joola.de/> – last access 01/2018

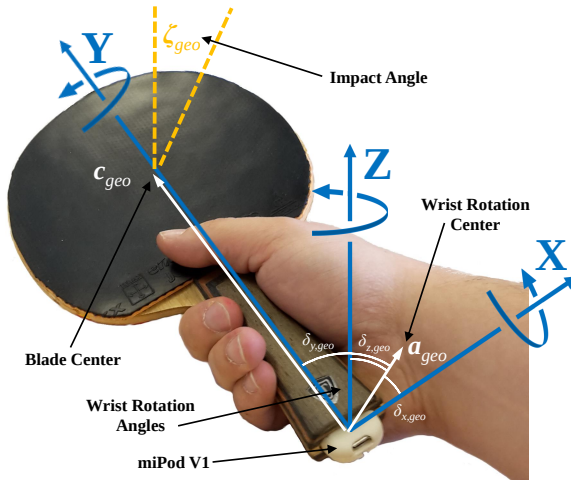


**Figure 6.4:** Racket prototype with sensor placement and different geometric parameters. The *miPod V1* IMU was mounted inside a milled cavity at the end of the racket's handle. The coordinate system (blue arrows) originated from the handle's endpoint and were aligned to the main axes of the racket itself. Here, the X-axis and the Y-axis spanned the main movement plane, whereas the  $a_{geo}$  describe the racket rotation around the wrist. Additionally, the wrist rotation center, which was defined by the appropriate angles  $\delta_{x,geo}$ ,  $\delta_{y,geo}$  and  $\delta_{z,geo}$ , as well as the blade's center  $c_{geo}$  are shown. Moreover, the ball impact angle  $\zeta_{geo}$  can be seen at the blade's center. These geometric parameters become necessary in subsequent processing ( $\rightarrow$  see Chapter 6.3) (modified from [Blan 17], doi>10.1145/3123021.3123040).

The video reference for ball spin and speed was provided by a CASIO®Exilim HS-ZR300 high-speed camera with a frame rate of 1000 Hz and a resolution of 224 x 64 pixels. Additionally, a second camera (here, a CASIO®Exilim HS-ZR200) with 120 Hz and 640 x 480 pixels was used to capture a scene overview.

### 6.2.2 Study Design

Accelerometer and gyroscope were calibrated using a similar method to [Ferr 95] ( $\rightarrow$  see Chapter 4.3). The research study was conducted to collect movement data of different arm and racket motions. Two types of strokes including *Topspins* or *Drives* (T, D) and *Pushes* (P) resulting in a forward ball spin and a backward ball spin were performed by eight

players. All subjects were male and within an age range of 13 to 50 years. The participants were right-handed as well as left-handed. They were all club players, used the shakehand grip and had a TTR coefficient from 1050 to 1975. Including subjects with different playing abilities resulted in receiving a high variability of racket motions and therefore of different ball spin and speed values. The subject overview can be seen in Table 6.2.

A schematic overview of the two proposed exercises can be seen in Figure 6.5. In the first exercise, all subjects were performing pushes on a ball approaching parallel to the table. The pass itself could have either no spin or a backward spin. The push resulted in a backward spin

**Table 6.2:** Overview of the ball spin and speed study group: number of subjects (#), sex (female F or male M), age, left- or right-handed (left L or right R) and ranking coefficient TTR. All data of the left-handed subject were transformed by a Z-axis rotation to achieve a uniform coordination system.

| # | Sex       | Age Range | Left-/Right-handed | TTR          |
|---|-----------|-----------|--------------------|--------------|
| 8 | 0 F / 8 M | 13 to 50  | 1 L / 7 R          | 1050 to 1975 |

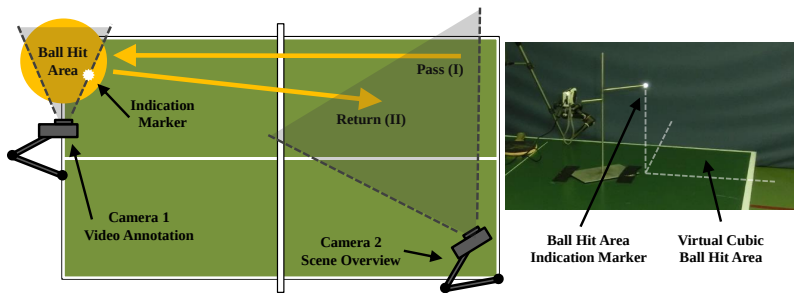


**Figure 6.5:** Schematic overview of the ball spin and speed exercise. Firstly, a parallel pass was initiated from the upper right quadrant of the table towards the ball hit area (I). Secondly, every player performed forward spin strokes (topspin or drive) and backward spin strokes (pushes) to return the ball into the same corner (II). Spin and speed data of the ball were captured by the high-speed video annotation camera (Camera 1), while the scene overview camera (Camera 2) observed the overall scene and the correct return position. Since the ball hit area was very narrow, a visual support for the subjects was given by a bright marker, which indicated the endpoint of a virtual cubic area, where the video annotation camera could definitely record the performed stroke (modified from [Blan 17], doi>10.1145/3123021.3123040).



**Figure 6.6:** Different exercise examples from the video reference: a topspin (T) stroke was performed on the left side, whereas a push (P) is shown on the right side. Additionally, both ball envelopes  $A_b$  and one of the feature markings  $f$  are highlighted.

of the returning ball. During the second exercise, the players executed topspins or drives resulting in low or strong forward spin. Here, the approaching ball involved also the same rotational direction as forward spin. Examples from the video reference of both strokes are shown in Figure 6.6. Every ball had to be returned into the right upper quadrant of the table, which was verified by the scene overview camera (Camera 2). The study design has ensured, that both directions of the flying ball, that means pass and return, mainly generated speed components parallel to the image plane of the video annotation camera (Camera 1). Additionally, this has guaranteed, that most of the generated ball spin was perpendicular to the flight direction and was therefore also perpendicular to the image plane of video annotation camera (Camera 1). Because of the restrictive parameters of the high-speed camera 1, the camera's field of view was very narrow. In order to support the subjects finding the correct position to hit the ball, a bright marker was mounted at the outer edge of the visible video. From that point to the outermost corner of the table, a virtual cube could be spanned to visualize the ball hit area.

### 6.3 Data Processing

Before processing, multiple assumptions and simplifications had to be made assessing unknown initial ball properties and racket movements. Firstly, all kinematic motion data collected by the accelerometer and the gyroscope were used to determine the racket motion during the stroke. Secondly, this racket motion was transferred to ball spin and speed shortly before impact with the help of predefined assumptions.

Finally, the ball spin and speed shortly after impact were estimated using a rebound model for table tennis rackets between rubber and ball.

During the following sections of this chapter,  $v_R(x_i)$  and  $\omega_R(x_i)$  denote the racket speed and racket spin at  $x_i$ , whereas  $v_B$  and  $\omega_B$  denote the ball speed and ball spin during impact. The indices 1 and 2 differentiate shortly before impact and shortly after impact, therefore,  $v_{B,1}$  describes the ball speed before impact and  $v_{B,2}$  the ball speed after impact. In the same manner describes  $\omega_{B,1}$  the ball spin before impact and  $\omega_{B,2}$  the ball spin after impact. However,  $v_b$  and  $\omega_b$  indicate the labeled ball speed and spin during impact from the video reference.

### 6.3.1 Ball and Racket Motion Assumptions

Since characteristics of the approaching ball were completely unknown, considerations about ball speed  $v_{B,1}$  and ball spin  $\omega_{B,1}$  before impact had to be made. Both parameters were necessary for the subsequent rebound modeling, therefore, the first assumption (A1) equalizes the incoming ball speed  $v_{B,1}$  with the sum of the corresponding racket speed  $v_R(x_{ip})$  during the impact at  $x_{ip}$  in opposite direction and a mean ball velocity  $v_\mu$ . This mean velocity depended on the performed stroke type of each player. In general, a topspin (T) or drive (D) stroke coincides with a higher  $v_\mu$  than pushes (P). The initial ball spin  $\omega_{B,1}$  was set to the parallel rotation of the racket  $\omega_R$  during the impact at  $x_{ip}$ .

$$(A1) \quad v_{B,1} = -\left(v_R(x_{ip}) + v_\mu\right), \quad \omega_{B,1} = \omega_R(x_{ip})$$

As ball rotations in the orthogonal blade direction (around the Z-axis, see Figure 6.4) are not possible by natural playing techniques, the component in Z-direction  $\omega_{R(z)}(x_i)$  of  $\omega_R(x_i)$  was set to 0. For the mean velocities of the approaching ball, absolute speeds of  $\sim 7$  m/s for topspin or drive strokes and  $\sim 2$  m/s for pushes were assumed empirically by video annotation for each stroke type over all players (A2).

$$(A2) \quad \|v_{\mu,(T,D)}\| \sim 7 \frac{m}{s}, \quad \|v_{\mu,(P)}\| \sim 2 \frac{m}{s}$$

Both ball velocities had to be mapped to the individual components of the racket motion in X, Y and Z-direction. Therefore, the incident angle  $\zeta_{geo}$  (see Figure 6.4) was also empirically found by video annotation to 45 deg for topspins or drives and to 30 deg for pushes (A3).

$$(A3) \quad \zeta_{geo,(T,D)} \sim 45 \text{ deg}, \quad \zeta_{geo,(P)} \sim 30 \text{ deg}$$

With assumption (A4), the ball impact position  $c_{geo}$  (see Figure 6.4) was in the center of the forehand side of the blade. Regarding the geometry of the racket, the impact coordinate was given as follows.

$$(A4) \quad c_{geo} \sim [0 \ 0.19 \ 0.007]^T \text{ in } [m]$$

Table tennis strokes are complex motions composed of arm, body and leg movements. Since only one sensor was mounted at the handle's end, it was not possible to describe the correct motion of the racket in a world frame, as it would be by a complete biomechanical model. Therefore, the complex differential equations of such a rigid body kinematic were strongly simplified in such a way, that the racket speed  $v_R(x_i)$  consisted of the linear component ( $v_{l,R}(x_i)$ ) and the rotational component ( $v_{r,R}(x_i)$ ) (A5). These parts were sample-wise calculated by integration of the gravity compensated acceleration  $\tilde{a}_R(x_i)$  and the derivative of the angular velocity  $\omega_R$ , transformed by a special wrist rotation matrix  $\mathcal{W}$ . Therewith, wrist rotations were assumed to be the most prominent rotational parts during the stroke movement. Other motions like arm or body rotations were assumed to have only minor influence on the overall racket speed estimation.

$$(A5) \quad v_R(x_i) = v_{l,R}(x_i) + v_{r,R}(x_i) = \\ = \int \tilde{a}_R(x_i) + \frac{d}{dt}(\mathcal{W} \cdot \omega_R(x_i))$$

The matrix  $\mathcal{W}$  was centered at the wrist of every subject. Hence, every subject's wrist center was measured before the study begin with a tape resulting in the wrist rotation angle vector  $\delta_{geo} = [\delta_{x,geo} \ \delta_{y,geo} \ \delta_{z,geo}]^T$  and the wrist rotation center vector  $a_{geo}$ , shown Figure 6.4. These parameters were finally averaged over all subjects to ensure a uniform definition of  $\mathcal{W}$ . They were defined as follows (A6).

$$(A6) \quad \delta_{geo} \sim [25 \ 60 \ -20]^T \text{ in } [deg] \\ a_{geo} \sim [-0.055 \ 0.035 \ 0.045]^T \text{ in } [m]$$

Moreover, a known point in time  $x_m$  was needed to determine a robust starting point for the acceleration integration, with as little motion artifacts as possible. Due to the highly dynamic nature of table tennis, there is almost no motionless point in time. For this reason, the turning point of the appropriate stroke countermovement was found as the best

starting point (A7). At this point, the overall acceleration  $\mathbf{a}_R(x_i)$  was assumed to equal the gravitational acceleration  $\mathbf{g}_{earth}$  and that no further racket motion occurred.

$$(A7) \quad \|\mathbf{a}_R(x_m)\| \stackrel{!}{\sim} \|\mathbf{g}_{earth}\|, \quad \|\mathbf{v}_R(x_m)\| \stackrel{!}{=} \|\mathbf{0}\|$$

Additionally, other indicators for suitable motionless points in time were investigated. One promising indicator was the global minimum in the overall racket velocity before an impact, expecting that there would be only little racket motion before the subsequent stroke acceleration had started. Furthermore, the zero-crossing of the rotation axis orthogonal to the racket (X-axis) could have been decisive for the countermovement's turning point and thus a potential option for  $x_m$ . However, all other approaches were not considered for further processing or algorithm development.

To sum up, using all these assumptions led to independence from the subject physiology or skill level and reduced the ball spin and speed calculation exclusively to kinematic motion equations of the racket as well as racket and rubber properties and characteristics.

### 6.3.2 Racket Speed Calculation

The processing pipeline for calculation of the racket speed  $\mathbf{v}_R$  can be seen in Figure 6.7. It was based on the six-dimensional racket motion data of the accelerometer  $\mathbf{a}_R$  and the gyroscope  $\boldsymbol{\omega}_R$ . According to (A5), the overall racket speed consisted of a linear and a rotational part.

#### Impact and Countermovement Detection

First of all, the actual impact had to be detected. Therefore, the root signal energy  $\Lambda(x_i)$  (Equation 6.7) based on the accelerometer X-axis  $a_x(x_i)$ , Y-axis  $a_y(x_i)$  and Z-axis  $a_z(x_i)$  was computed.

$$\Lambda(x_i) = \sqrt{a_x(x_i)^2 + a_y(x_i)^2 + a_z(x_i)^2} \quad (6.7)$$

$$0 \leq i < \text{Total signal length } L$$

$$\Lambda(x_i) = \text{Root signal energy in } [g]$$

$$a_{\{x,y,z\}}(x_i) = \text{Raw acceleration in } [g]$$

$$x_i = \text{Data sample}$$

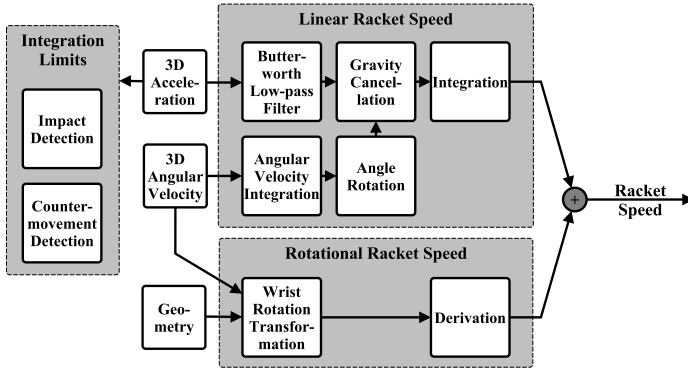


**Figure 6.7:** Calculation of racket speed  $v_R(x_i)$ . Firstly, an impact and counter-movement detection was performed to compute the integration limits  $x_{ip}$  and  $x_m$ . Secondly, the angular velocity  $\omega_R(x_i)$  was integrated resulting in relative rotation angles  $\Phi_R(x_i)$ . Using these relative angles, the initial gravity vector  $g_{earth}$  was rotated sample-wise and subtracted from the filtered racket acceleration  $a_{filt,R}(x_i)$ . Then, the pure movement acceleration  $\tilde{a}_R(x_i)$  was integrated to obtain the linear component of the racket speed  $v_{l,R}(x_i)$ . In parallel, the angular velocity  $\omega_R(x_i)$  was transformed by the wrist rotation matrix  $\mathcal{W}$  and derived to determine the rotational component of the racket speed  $v_{r,R}(x_i)$ . Finally, both components were combined to  $v_R(x_i)$ .

Subsequently, the exact impact position  $x_{ip}$  was calculated using the peak detection algorithm presented in ( $\rightarrow$  see Chapter 5.2). Here, the threshold value  $\theta$  (Equation 6.8) as well as the segmentation range  $\kappa_e$  (Equation 6.9) were chosen as the half of the total energy maximum of  $\Lambda(x_i)$  and as the minimal temporal distance between two consecutive strokes, respectively.

$$\theta = \frac{1}{2} \cdot \max\{\Lambda(x_i)\} \quad (6.8)$$

$$\kappa_e = [x_{i-\kappa_{el}}, \dots, x_{i+\kappa_{el}}] \quad (6.9)$$

$$\theta = \text{Threshold in } [g]$$

$$\Lambda(x_i) = \text{Root signal energy in } [g]$$

$$\kappa_{el} + \kappa_{er} = \text{Size of segmentation range } \kappa_e$$

$$x_i = \text{Data sample}$$

Compared to the threshold value for stroke detection ( $\theta = \mu + h \cdot \sigma$ ), here,  $\theta$  was set to a higher value avoiding the recognition of countermovement parts or components of past strokes as the actual impact event. The parameters  $\kappa_{el}$  and  $\kappa_{er}$  were set to 1200 and 300 resulting in a overall segmentation range of 1.5 seconds. Consequently, the window was intentionally moved to include more past data as it was used for the stroke detection and classification.

On the one hand, the prior countermovements contain the most important information about the current stroke itself [Rodr 02] and were here of special importance, on the other hand, enough past data had to be assigned to the impact event so that all data about the turning point of the countermovement  $x_m$  could be included. Therefore, a wider date range containing more information was used.

Peaks  $P(x_j)$  (Equation 6.10) within  $\kappa_e$  could be detected, if the window maximum  $H(x_j)$  was equal to  $V(x_j)$  and if  $V(x_j)$  exceeded the specified threshold  $\theta$ .

$$P(x_j) = \begin{cases} 1, & \text{if } H(x_j) = \Lambda(x_i) \wedge \Lambda(x_i) > \theta \\ 0, & \text{otherwise} \end{cases} \quad (6.10)$$

$$H(x_j) = \max\{\Lambda(x_{i-\kappa_{el}}), \dots, \Lambda(x_{i+\kappa_{er}})\}$$

$$i + \kappa_{el} \leq j < L - \kappa_{er}$$

$$P(x_j) = \text{Peak signal}$$

$$L = \text{Total signal length}$$

$$\kappa_{el} + \kappa_{er} = \text{Size of segmentation range } \kappa_e$$

$$H(x_j) = \text{Local maximum signal in } [g]$$

$$\Lambda(x_i) = \text{Root signal energy in } [g]$$

$$\theta = \text{Threshold in } [g]$$

$$x_i, x_j = \text{Data samples}$$

A post-processing step retained only one maximum, which had the highest energy in  $\Lambda(x_j)$ . This indicated the most likely impact point  $x_{ip}$  (Equation 6.11) and was used as endpoint for the integration of the racket acceleration  $a_R(x_j)$ .

In contrast, the starting point for integration  $x_m$  (Equation 6.12) was found according to assumption (A7) by minimization of the absolute

difference between the signal root energy  $\Lambda(x_j)$  and the gravitational acceleration of the earth  $g_{earth}$  with  $9.80665 \text{ m/s}^2$ .

$$x_{ip} = \text{argmax}\{P(x_j)\} \quad (6.11)$$

$$x_m = \text{argmin}\{\text{abs}\{\Lambda(x_i) - g_{earth}\}\} \quad (6.12)$$

$x_{ip}$  = Impact point

$x_m$  = Starting point for integration

$P(x_j)$  = Detected peaks

$\Lambda(x_i)$  = Root signal energy in  $[g]$

$g_{earth}$  = Gravitational acceleration in  $[g]$

$x_i, x_j$  = Data samples

## Linear Racket Speed Calculation

In the next steps, the linear component of the racket speed  $v_{l,R}(x_i)$  had to be calculated (see gray box in the processing overview of Figure 6.7) using the raw acceleration data  $a_R(x_i)$  and the raw angular velocity  $\omega_R(x_i)$ . Both were captured by the racket-mounted inertial sensor.

First of all, the acceleration  $a_R(x_i)$  was integrated from  $x_m$  to  $x_{ip}$ . However, all gravitational components and artifacts had to be eliminated beforehand. Therefore, a Butterworth low-pass filter with an order of 1 and a cut-off frequency of 25 Hz was applied to  $a_R(x_i)$  within the window  $\kappa_e$  in a forward and backward manner to avoid any group delay.

$$\phi_R(x_i) = \begin{bmatrix} \Delta\alpha(x_i) \\ \Delta\beta(x_i) \\ \Delta\gamma(x_i) \end{bmatrix} = \int \omega_R(x_i) \cdot T_A \quad (6.13)$$

$$x_m \leq i < x_{ip}$$

$\phi_R(x_i)$  = Relative angle vector in  $[rad]$

$\Delta\alpha(x_i), \Delta\beta(x_i), \Delta\gamma(x_i)$  = Relative rotation angles in  $[rad]$

$\omega_R(x_i)$  = Angular velocity in  $[rad/s]$

$T_A$  = Sampling interval in  $[s]$

$x_m, x_{ip}$  = Integration limits

$x_i$  = Data sample

The filter was intended to suppress all high-frequency motion interferences and to emphasize the raw arm and racket movement [Rodr 02]. Regarding assumption (A7), the initial acceleration  $\mathbf{a}_{filt,R}(x_i)$  was equal to the gravity value  $\mathbf{g}_{earth}(x_m)$  at  $x_m$ . In addition, a sample-wise multiplication of the angular velocity  $\boldsymbol{\omega}_R(x_i)$  concluded the relative angles  $\boldsymbol{\Phi}_R(x_i) = [\Delta\alpha(x_i) \ \Delta\beta(x_i) \ \Delta\gamma(x_i)]^T$  (Equation 6.13) of the racket rotation during the interval  $\kappa_e$  in steps of 0.001 seconds from point  $x_m$  to point  $x_{ip}$ . As a result, the initial gravity vector  $\mathbf{g}_{earth}(x_m)$  could be rotated by the relative angles  $\phi_R(x_i)$  for every  $x_j$  by standard rotational matrices  $\mathcal{R}_{\Delta\alpha}(x_i)$  for X-direction (Equation 6.14),  $\mathcal{R}_{\Delta\beta}(x_i)$  for Y-direction (Equation 6.15) and  $\mathcal{R}_{\Delta\gamma}(x_i)$  for Z-direction (Equation 6.16).

$$\mathcal{R}_{\Delta\alpha}(x_i) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\Delta\alpha(x_i)) & -\sin(\Delta\alpha(x_i)) \\ 0 & \sin(\Delta\alpha(x_i)) & \cos(\Delta\alpha(x_i)) \end{bmatrix} \quad (6.14)$$

$$\mathcal{R}_{\Delta\beta}(x_i) = \begin{bmatrix} \cos(\Delta\beta(x_i)) & 0 & \sin(\Delta\beta(x_i)) \\ 0 & 1 & 0 \\ -\sin(\Delta\beta(x_i)) & 0 & \cos(\Delta\beta(x_i)) \end{bmatrix} \quad (6.15)$$

$$\mathcal{R}_{\Delta\gamma}(x_i) = \begin{bmatrix} \cos(\Delta\gamma(x_i)) & -\sin(\Delta\gamma(x_i)) & 0 \\ \sin(\Delta\gamma(x_i)) & \cos(\Delta\gamma(x_i)) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (6.16)$$

$\mathcal{R}_{\Delta\alpha}(x_i), \mathcal{R}_{\Delta\beta}(x_i), \mathcal{R}_{\Delta\gamma}(x_i)$  = Rotational matrices

$\Delta\alpha(x_i), \Delta\beta(x_i), \Delta\gamma(x_i)$  = Relative rotation angles in [rad]

$x_i$  = Data sample

$$\tilde{\mathbf{a}}_R(x_i) = \mathbf{a}_{filt,R}(x_i) - \mathcal{R}_{\Delta\gamma}(x_i) \cdot \left[ \mathcal{R}_{\Delta\beta}(x_i) \cdot \left[ \mathcal{R}_{\Delta\alpha}(x_i) \cdot \mathbf{g}_{earth}(x_m) \right] \right] \quad (6.17)$$

$$x_m \leq i < x_{ip}$$

$\tilde{\mathbf{a}}_R(x_i)$  = Gravity compensated acceleration in [g]

$\mathbf{a}_{filt,R}(x_i)$  = Filtered acceleration in [g]

$\mathcal{R}_{\Delta\alpha}(x_i), \mathcal{R}_{\Delta\beta}(x_i), \mathcal{R}_{\Delta\gamma}(x_i)$  = Rotational matrices

$\mathbf{g}_{earth}(x_m)$  = Gravitational acceleration at  $x_m$  in [g]

$x_j$  = Data sample

A subtraction of the rotated gravity from the filtered acceleration data  $\mathbf{a}_{filt,R}(x_i)$  completed the calculation of the intended gravity compensated straight racket acceleration  $\tilde{\mathbf{a}}_R(x_i)$  (Equation 6.17).

Considering assumption (A5) and (A7), the linear component of the racket speed  $\mathbf{v}_{l,R}(x_i)$  (Equation 6.18) could be calculated by integration of the gravity compensated acceleration  $\tilde{\mathbf{a}}_R(x_i)$  from  $x_m$  to  $x_{ip}$ , incipiently with  $\mathbf{v}_{l,R}(x_m)$  equal to the null-vector in steps of 0.001 seconds from the point  $x_m$  to the point  $x_{ip}$ .

$$\mathbf{v}_{l,R}(x_i) = \int \tilde{\mathbf{a}}_R(x_i) \cdot T_A \quad (6.18)$$

$$x_m \leq i < x_{ip}$$

$\mathbf{v}_{l,R}(x_i)$  = Linear racket speed in [m/s]  
 $\tilde{\mathbf{a}}_R(x_i)$  = Gravity compensated acceleration in [g]  
 $T_A$  = Sampling interval in [s]  
 $x_m, x_{ip}$  = Integration limits  
 $x_i$  = Data sample

## Rotational Racket Speed Calculation

The next step was the computation of the rotational racket speed component  $\mathbf{v}_{r,R}(x_i)$  (Equation 6.19) according to the processing overview of Figure 6.7 using the angular velocity  $\boldsymbol{\omega}_R(x_i)$  of the racket motion.

With assumption (A5), the rotational racket speed component was mainly based on the wrist rotation. Hence, all angular velocity parts in X-direction, Y-direction and the Z-direction of  $\boldsymbol{\omega}_R(x_i)$  were transformed into tangential linear velocities components using the transformation matrix  $\mathcal{W}$  (Equation 6.20) and the *Law of Lever* principle.

$$\mathbf{v}_{r,R}(x_i) = \mathcal{W} \cdot \boldsymbol{\omega}_R(x_i) \quad (6.19)$$

$$\mathbf{v}_{r,R}(x_i) = \text{Rotational racket speed in [m/s]}$$

$$\mathcal{W} = \text{Wrist rotation matrix in [m/rad]}$$

$$\boldsymbol{\omega}_R(x_i) = \text{Angular velocity in [rad/s]}$$

$$x_i = \text{Data sample}$$

Additionally, the ball impact position  $\mathbf{c}_{geo}$  was geometrically defined by assumption (A4). Furthermore, using (A6), the wrist rotation angles

$\delta_{geo} = [\delta_{x,geo} \ \delta_{y,geo} \ \delta_{z,geo}]^T$  were measured from the associated coordinate system axes to the wrist center using the wrist rotation center vector  $a_{geo} = [a_{x,geo} \ a_{y,geo} \ a_{z,geo}]^T$ . All parameters are exemplary illustrated in combination with the racket prototype in Figure 6.4.

$$\mathcal{W} = \begin{bmatrix} 0 & a_{z,geo} \cdot \cos(b_{z,geo}) & a_{x,geo} \cdot \cos(b_{x,geo}) \\ a_{y,geo} \cdot \sin(b_{y,geo}) & 0 & a_{x,geo} \cdot \sin(b_{x,geo}) \\ -a_{y,geo} \cdot \cos(b_{y,geo}) & a_{z,geo} \cdot \sin(b_{z,geo}) & 0 \end{bmatrix} \quad (6.20)$$

$$b_{k,geo} = \frac{a_{k,geo} \cdot \arcsin(\frac{\pi}{2} - \delta_{k,geo})}{\sqrt{c_{k,geo}^2 + a_{k,geo}^2 - 2 \cdot c_{k,geo} \cdot a_{k,geo} \cdot \cos(\frac{\pi}{2} - \delta_{k,geo})}}$$

$k \in \{X, Y, Z\}$  representing each axis

$\mathcal{W}$  = Wrist rotation matrix in  $[m/rad]$

$c_{geo} = [c_{x,geo} \ c_{y,geo} \ c_{z,geo}]^T$  = Ball impact position in  $[m]$

$\delta_{geo} = [\delta_{x,geo} \ \delta_{y,geo} \ \delta_{z,geo}]^T$  = Wrist rotation angles in  $[rad]$

$a_{geo} = [a_{x,geo} \ a_{y,geo} \ a_{z,geo}]^T$  = Wrist rotation center in  $[m]$

## Racket Speed Calculation

Lastly, the wanted racket speed during impact  $v_R$  (Equation 6.21) was given as the sum of both racket speed components (see Figure 6.7) for the linear racket speed  $v_{l,R}(x_i)$  and the rotational racket speed  $v_{r,R}(x_i)$  at  $x_{ip}$ , according to assumption (A5).

$$v_R = [v_{l,R}(x_{ip}) + v_{r,R}(x_{ip})] \quad (6.21)$$

$$\omega_R = [\omega_R(x_{ip})] \quad (6.22)$$

$v_R$  = Racket speed during impact in  $[m/s]$

$\omega_R$  = Racket rotation during impact in  $[rad/s]$

$v_{l,R}(x_{ip})$  = Linear racket speed at  $x_{ip}$  in  $[m/s]$

$v_{r,R}(x_{ip})$  = Rotational racket speed at  $x_{ip}$  in  $[m/s]$

$\omega_R(x_{ip})$  = Angular velocity at  $x_{ip}$  in  $[rad/s]$

Similar, the racket rotation during impact  $\omega_R$  (Equation 6.22) is given as follows regarding  $x_{ip}$ . The overall racket speed  $v_R(x_i)$  including the countermovement for a topspin stroke is exemplary shown in Figure 6.8.

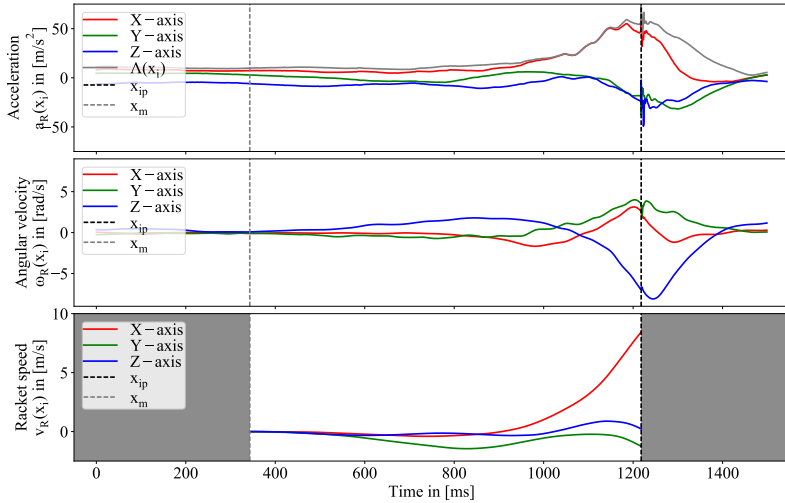


**Figure 6.8:** Example acceleration  $a_R(x_i)$  and angular velocity  $\omega_R(x_i)$  data of a topspin (T) stroke, shown in the first two graphs. Furthermore, the root signal energy  $\Lambda(x_i)$  is shown in the topmost graph. Additionally, the integration borders  $x_m$  (motionless point in time) and  $x_{ip}$  (impact time) are marked as dotted gray and black lines. In the topmost figure, the motionless point in time  $x_m$  was almost located where  $\Lambda(x_j)$  was closest to the gravitational acceleration  $g_{earth}$  before the impact  $x_{ip}$ . This was also confirmed by nearly zero angular speed at  $x_m$  in the middle figure. Finally, the combined racket speed  $v_R(x_i)$  is shown in the lower figure. In all graphs, data are split into X-axis (red), Y-axis (green) and Z-axis (blue) (modified from [Blan 17], doi>10.1145/3123021.3123040).

### 6.3.3 Rebound Model

Knowing the racket motion  $v_R$  and  $\omega_R$  during impact, the prior ball movement  $v_{B,1}$  and  $\omega_{B,1}$  could be estimated according to assumptions (A1), (A2) and (A3). Then, the final ball speed  $v_{B,2}$  and ball spin  $\omega_{B,2}$  were computed using models for the rebound phenomenon of elastic rubbers and rigid balls with friction and spinning effects introduced in [Naka 09] and [Naka 10]. An overview of the rebound model and the

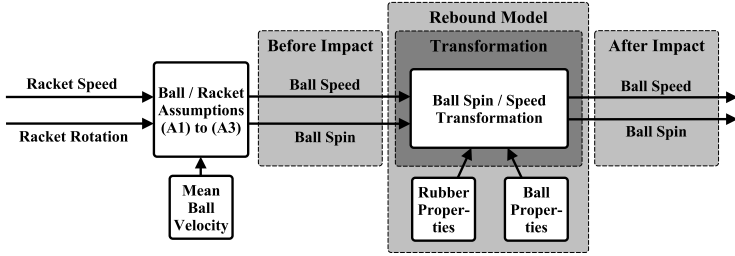


**Figure 6.9:** Schematic overview of the rebound model for the transformation of ball spin into speed and vice versa. Firstly, the racket speed  $v_R$  and the mean ball velocity  $v_\mu$  are linked to yield the assumed initial ball speed  $v_{B,1}$ , whereas the initial ball spin  $\omega_{B,1}$  was built on the racket rotation  $\omega_R$  regarding assumptions (A1), (A2) and (A3). Secondly, both values were transformed to  $v_{B,2}$  and  $\omega_{B,2}$  using racket, rubber and ball properties and the transformation matrices  $\mathcal{A}$ ,  $\mathcal{B}$ ,  $\mathcal{C}$  and  $\mathcal{D}$ , as well as the restitution coefficient  $e_R$  and the rubber friction  $\mu_R$ .

spin and speed transformation can be seen in Figure 6.9. The initial ball speed  $v_{B,1}$  (Equation 6.23) and ball spin  $\omega_{B,1}$  (Equation 6.24) had to be estimated using assumptions (A1), (A2) and (A3).

$$v_{B,1} = -(v_R + v_\mu) = -v_R - \|v_\mu\| \cdot \begin{bmatrix} \sin(\zeta_{x,geo}) \\ 0 \\ \cos(\zeta_{z,geo}) \end{bmatrix} \quad (6.23)$$

$$\omega_{B,1} = \omega_R \quad (6.24)$$

$v_{B,1}$  = Initial ball speed in  $[m/s]$

$\omega_{B,1}$  = Initial ball rotation in  $[rad/s]$

$v_R$  = Racket speed in  $[m/s]$

$\omega_R$  = Racket rotation in  $[rad/s]$

$v_\mu$  = Mean ball speed in  $[m/s]$

$\zeta_{x,geo}$ ,  $\zeta_{z,geo}$  = Impact angle in X and Z-direction in  $[rad]$

However, in this study setup the Y-component of the mean ball speed was assumed to be zero, because it is almost impossible to hit the ball along the racket handle axis during topspin (T), drive (D) and push (P) strokes resulting in pure side spin. The proposed rebound model not only depended on the approaching ball motion parameters, but also on

characteristics of the table tennis ball itself as well as on frictional and elastic specifications of the used racket and rubber combination. These are defined by the restitution coefficient  $e_R$  [Brig 45] and the friction  $\mu_R$ .

The restitution coefficient  $e_R$  (Equation 6.25) can be described as the energy loss during the elastic impact between ball and racket. Assuming a gravity acceleration  $g_{earth}$  of  $9.80665 \text{ m/s}^2$  and ball velocities before and after impact of  $v_1^b = -\sqrt{2 \cdot g \cdot h_1}$  and  $v_2^b = \sqrt{2 \cdot g \cdot h_2}$  as well as a stationary racket with  $v_1^r$  of  $0 \text{ m/s}$  and  $v_2^r$  of  $0 \text{ m/s}$ , the restitution coefficient  $e_R$  could be computed during a free fall of the ball. For this study, ten free fall trials were performed with the tested racket rubber combination ( $\rightarrow$  see Chapter 6.2). The measures for the heights  $h_1$  and  $h_2$  were finally averaged to achieve meaningful results for restitution the coefficient  $e_R$ .

$$e_R = \frac{v_2^r - v_2^b}{v_1^b - v_1^r} = \frac{-v_2^b}{v_1^b} = \frac{-\sqrt{2 \cdot g_{earth} \cdot h_2}}{-\sqrt{2 \cdot g_{earth} \cdot h_1}} = \sqrt{\frac{h_2}{h_1}} \quad (6.25)$$

$e_R$  = Restitution coefficient

$v_1^b, v_1^r$  = Ball and racket speed before impact in  $[m/s]$

$v_2^b, v_2^r$  = Ball and racket speed after impact in  $[m/s]$

$h_1, h_2$  = Height before and after impact in  $[m]$

$g_{earth}$  = Gravitational acceleration in  $[m/s^2]$

The friction coefficient  $\mu_R$  (Equation 6.26) is defined as quotient of the dry friction force  $F_R$  and the normal force  $F_N$  of a mass, here three time the ball mass  $m_b$  of  $0.0027 \text{ kg}$  and a weighing object  $m_w$  with  $0.01 \text{ kg}$ .

$$\mu_R = \frac{F_N}{F_R} = \frac{(3 \cdot m_b + m_w) \cdot g_{earth}}{F_R} \quad (6.26)$$

$\mu_R$  = friction coefficient

$F_N, F_R$  = Normal force and friction force in  $[N]$

$m_b, m_w$  = Ball and test object mass in  $[kg]$

$g_{earth}$  = Gravitational acceleration in  $[m/s^2]$

This setup was used to obtain a movable test object, which always had optimal contact between the ball and the rubber surface. Therefore, three table tennis balls were arranged in the form of a triangle, bound together and weighted with the additional mass to ensure proper contact

of all three balls with the rubber surface. Attached to this object, a force meter measured the force needed to slide the object along the elastic surface. Similar to  $e_R$ , the measures for  $F_R$  of ten trials were averaged to achieve meaningful results for the friction coefficient  $\mu_R$ .

During the impact process, it is important whether the ball slides (I) or rolls (II) over the rubber surface. It influences the final transformation of ball spin and speed. This is identified by  $\lambda_R$  (Equation 6.27) [Naka 09]. If the friction case indicator  $\lambda_R$  is  $> 0$ , the requirements for the sliding friction (I) were fulfilled, while if  $\lambda_R$  is  $< 0$ , the ball rolls over the rubber surface (II).

$$\lambda_R = 1 - \frac{5 \cdot \mu_R \cdot (1 + e_R) \cdot \text{abs}\{v_{B,1(z)}\}}{2 \cdot \sqrt{v_{B,1(x)}^2 + v_{B,1(y)}^2}} \quad (6.27)$$

$\lambda_R$  = Friction case indicator

$\mu_R$  = Friction coefficient

$e_R$  = Restitution coefficient

$v_{B,1(x,y,z)}$  = Initial ball speed in X, Y and Z-direction in [m/s]

Considering case (I), the friction caused a rolling ball contact. This could be modeled as a virtual spring-absorber system, which saved and released deformation energy during the whole contact time. Thereby it was assumed, that the energy conservation was done without any dissipation, that the total displaced mass of the rubber  $m_R$  was constant over the impact duration  $t_{ip}$  and that the spring constant  $k_R$  of the rubber was uniform in any direction. Knowing these parameters, the rubber elasticity  $k_{ip}$  (Equation 6.28) could be calculated as follows [Naka 10].

$$k_{ip} = \frac{t_{ip}}{2} \cdot \sqrt{m_R \cdot k_R} \quad (6.28)$$

$k_{ip}$  = Rubber elasticity parameter in [kg]

$t_{ip}$  = Impact duration in [s]

$m_R$  = Displaced rubber mass in [kg]

$k_R$  = Rubber spring constant in [N/m]

The elasticity  $k_{ip}$  was approximated by a fifth degree polynomial function. This function was modeled for topspin (T) and drive (D) strokes as well as for pushes (P), since differences within the impact duration and

therefore the rubber deformation were expected during various strokes. The resulting parameters  $k_{ip,(T,D)}$  and  $k_{ip,(P)}$  were found using a least squares minimization with averaged data from all subjects.

Knowing the ball mass  $m_b$  with 0.0027 kg and the ball radius  $r_b$  with 0.02 m, the transformation matrices  $\mathcal{A}$  and  $\mathcal{B}$  for ball speed (Equation 6.29) as well as  $\mathcal{C}$  and  $\mathcal{D}$  for ball spin (Equation 6.30) can be computed.

$$\mathcal{A}_{(I)} = \begin{bmatrix} 1 - \frac{k_{ip}}{m_b} & 0 & 0 \\ 0 & 1 - \frac{k_{ip}}{m_b} & 0 \\ 0 & 0 & -e_R \end{bmatrix} \quad \mathcal{B}_{(I)} = \begin{bmatrix} 0 & \frac{k_{ip} \cdot r_b}{m_b} & 0 \\ -\frac{k_{ip} \cdot r_b}{m_b} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (6.29)$$

$$\mathcal{C}_{(I)} = \begin{bmatrix} 0 & -\frac{3 \cdot k_{ip}}{2 \cdot m_b \cdot r_b} & 0 \\ \frac{3 \cdot k_{ip}}{2 \cdot m_b \cdot r_b} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \mathcal{D}_{(I)} = \begin{bmatrix} 1 - \frac{3 \cdot k_{ip}}{2 \cdot m_b} & 0 & 0 \\ 0 & 1 - \frac{3 \cdot k_{ip}}{2 \cdot m_b} & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (6.30)$$

$\mathcal{A}_{(I)}, \mathcal{B}_{(I)}, \mathcal{C}_{(I)}, \mathcal{D}_{(I)}$  = Transformation matrices for case (I)

$k_{ip}$  = Rubber elasticity parameter in [kg]

$m_b$  = Ball mass in [kg]

$r_b$  = Ball radius in [m]

$e_R$  = Restitution coefficient

Considering case (II), the contact ball speed  $v_{B,1(z)}$  was not high enough to pass noteworthy energy to the spring-absorber system of the elastic rubber during the impact process.

$$\mathcal{A}_{(II)} = \begin{bmatrix} \frac{3}{5} & 0 & 0 \\ 0 & \frac{3}{5} & 0 \\ 0 & 0 & -e_R \end{bmatrix} \quad \mathcal{B}_{(II)} = \begin{bmatrix} 0 & \frac{2 \cdot r_b}{5} & 0 \\ -\frac{2 \cdot r_b}{5} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (6.31)$$

$$\mathcal{C}_{(II)} = \begin{bmatrix} 0 & -\frac{3}{5 \cdot r_b} & 0 \\ \frac{3}{5 \cdot r_b} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad \mathcal{D}_{(II)} = \begin{bmatrix} \frac{2}{5} & 0 & 0 \\ 0 & \frac{2}{5} & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (6.32)$$

$\mathcal{A}_{(II)}, \mathcal{B}_{(II)}, \mathcal{C}_{(II)}, \mathcal{D}_{(II)}$  = Transformation matrices for case (II)

$r_b$  = Ball radius in [m]

$e_R$  = Restitution coefficient

Hence, this specific influence could be neglected and the transformation matrices between ball speed and ball spin were simplified to the equations (6.31) and (6.32).

Besides the restitution coefficient  $e_R$ , the rebound process is only dependent on the ball radius  $r_b$  and additional geometric data. Finally, the ball speed  $v_{B,2}$  (Equation 6.33) and the ball spin  $\omega_{B,2}$  (Equation 6.34) after impact are estimated using either the transformation matrices for the sliding case (I) or the rolling case (II).

$$v_{B,2} = \mathcal{A} \cdot v_{B,1} + \mathcal{B} \cdot \omega_{B,1} \quad (6.33)$$

$$\omega_{B,2} = \mathcal{C} \cdot v_{B,1} + \mathcal{D} \cdot \omega_{B,1} \quad (6.34)$$

$$v_{B,2} = \text{Rebound ball speed in } [m/s]$$

$$\omega_{B,2} = \text{Rebound ball spin in } [rad/s]$$

$$v_{B,1} = \text{Initial ball speed in } [m/s]$$

$$\omega_{B,1} = \text{Initial ball spin in } [rad/s]$$

$$\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D} = \text{Transformation matrices either for case (I) or (II)}$$

## 6.4 Evaluation and Results

Using the equations (6.33) and (6.34) for the final ball speed and ball spin, the equations (6.21) and (6.22) for the racket motion and the assumptions (A1) to (A7), it was possible to estimate the ball speed  $v_{B,2}$  and spin  $\omega_{B,2}$  right after impact. In this study it was assumed that the ball moved parallel towards the image plane of the high-speed camera. Accordingly, the labeled ball speed  $v_b$  was approximately the estimated ball speed  $v_{B,2(xz)}$  (Equation 6.35) within the main movement plane in XZ-direction of the table tennis racket.

$$v_b \sim v_{B,2(xz)} = \sqrt{v_{B,2(x)}^2 + v_{B,2(z)}^2} \quad (6.35)$$

$$v_{B,2(xz)} = \text{Estimated ball speed in XZ-plane in } [m/s]$$

$$v_{B,2(x)}, v_{B,2(z)} = \text{Estimated ball speed in X, and Z-direction in } [m/s]$$

The labeled ball spin  $\omega_b$  was nearly the opposite estimated ball spin  $\omega_{B,2(y)}$  (Equation 6.36) alongside the racket Y-axis. Thus, the ball rotated orthogonal to the image plane of the reference camera. The mean values for  $\mu_{v_{B,2(xz)}}$  and  $\mu_{\omega_{B,2(y)}}$  as well as the standard deviations for  $\sigma_{v_{B,2(xz)}}$  and  $\sigma_{\omega_{B,2(y)}}$  were calculated to compare labeled with computed

data separately for each stroke type class of the topspin or drive strokes (T, D) and of the push strokes (P).

$$\omega_b \sim -\omega_{B,2(y)} \quad (6.36)$$

$$\omega_{B,2(y)} = \text{Estimated ball spin in Y-direction in } [rad/s]$$

Additionally, overall accuracies and qualitative diagrams substantiated the quantitative validation. Compared to the accuracy definition for a binary classification system ( $\rightarrow$  see Chapter 2.3), the here mentioned accuracy describes the mean percentage difference between labeled and estimated values.

Moreover, the rubber coefficients for restitution  $e_R$  and friction  $\mu_R$  are unitless and calculated as  $0.746 \pm 0.013$  and  $1.533 \pm 0.124$ , respectively. The mean ball velocities stated in (A2) were identified from the video and averaged for all subjects. Thus, the mean ball velocity for topspin and drive strokes  $\|v_{\mu,(T,D)}\|$  was found to be  $\sim 7$  m/s and  $\sim 2$  m/s for pushes  $\|v_{\mu,(P)}\|$ . The elasticity parameters  $k_{ip,(T,D)}$  and  $k_{ip,(P)}$  were approximated as  $1.625 \cdot 10^{-3}$  kg and  $1.760 \cdot 10^{-3}$  kg, respectively. During the exercises, 109 valid topspin, drive and push strokes were collected. Two conditions had to be fulfilled, so that a stroke was accepted as valid. Firstly, the entire impact scene of every stroke had to be completely visible on the high-speed camera 1. Secondly, the return position was inside the correct table quadrant, which could be verified by the scene camera 2. Qualitative results are shown in Figure 6.10 and Figure 6.11, whereas Figure 6.12 and Figure 6.13 provide *Bland-Altman* plots [Blan 86] to compare the agreement of measurement approaches. Furthermore, all results for ball speed  $v_{B,2(xz)}$  and ball spin  $\omega_{B,2(y)}$  estimation can be quantitatively seen in Table 6.3 and Table 6.4, respectively.

**Table 6.3:** Overall results for the ball speed estimation  $v_{B,2(xz)}$ , given in [m/s]. Additionally, the outcomes are separated into topspin (T), drive (D) and push (P) strokes. Moreover, the accuracy is given in [%] (modified from [Blan 17], doi>10.1145/3123021.3123040).

| Stroke Type | $\mu \pm \sigma$ of $v_{B,2(xz)}$ in $[\frac{m}{s}]$ | Accuracy [%] |
|-------------|------------------------------------------------------|--------------|
| T, D        | $-0.09 \pm 0.59$                                     | 79.4         |
| P           | $-0.01 \pm 0.57$                                     | 87.4         |

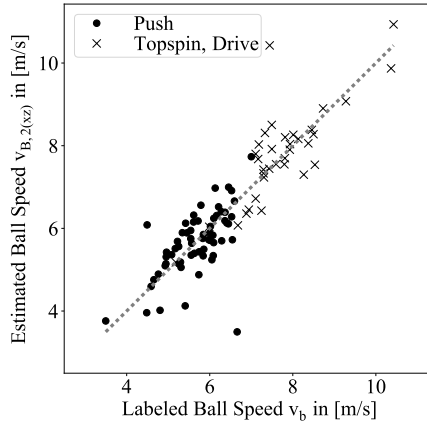


**Figure 6.10:** The estimated ball speed  $v_{B,2(xz)}$  in [m/s] is plotted against the labeled ball speed  $v_b$ . Topspin and drive strokes are marked as crosses ( $\times$ ), whereas pushes are marked as dots ( $\bullet$ ). The dotted gray line would indicate an ideal estimation (modified from [Blan 17], doi>10.1145/3123021.3123040).

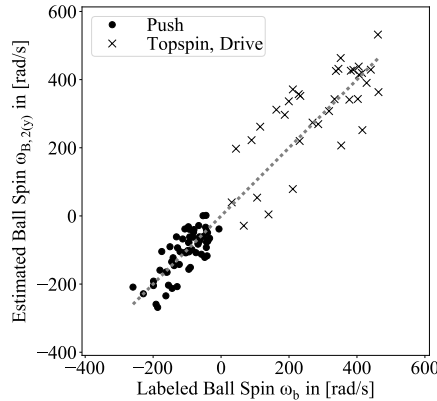


**Figure 6.11:** The estimated ball spin  $\omega_{B,2(y)}$  in [rad/s] is plotted against the labeled ball spin  $\omega_b$ . Topspin and drive strokes are marked as crosses ( $\times$ ), whereas pushes are marked as dots ( $\bullet$ ). The dotted gray line would indicate an ideal estimation. Negative rotations describe backward spin, positive values show forward spin (modified from [Blan 17], doi>10.1145/3123021.3123040).

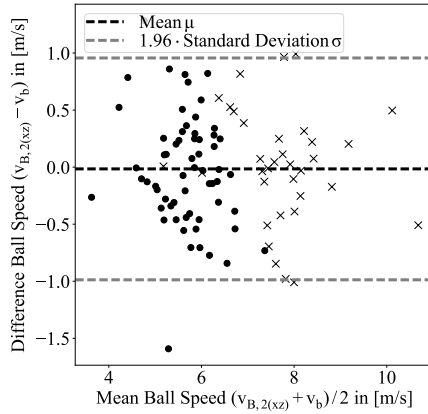


**Figure 6.12:** The ball speed difference  $(v_{B,2(xz)} - v_b)$  of estimated and labeled data are plotted against  $0.5 \cdot (v_{B,2(xz)} + v_b)$  of both. The mean value  $\mu$  of the differences is shown as dotted black line, whereas the 1.96-fold standard deviation limits  $\sigma$  are shown in dotted gray lines. Crosses (x) are topspins or drives, dots (•) represent pushes (modified from [Blan 17], doi>10.1145/3123021.3123040).

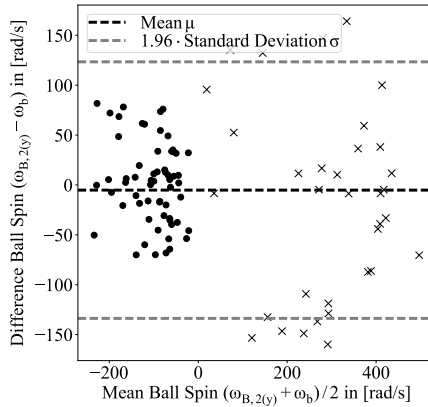


**Figure 6.13:** The ball spin difference  $(\omega_{B,2(xz)} - \omega_b)$  of estimated and labeled data are plotted against  $0.5 \cdot (\omega_{B,2(xz)} + \omega_b)$  of both. The mean value  $\mu$  of the differences is shown as dotted black line, whereas the 1.96-fold standard deviation limits  $\sigma$  are shown in dotted gray lines. Crosses (x) are topspins and drives, dots (•) represent pushes (modified from [Blan 17], doi>10.1145/3123021.3123040).

**Table 6.4:** Overall results for the ball spin estimation  $\omega_{B,2(y)}$ , given in [rad/s]. Similarly, the outcomes are separated into topspin (T), drive (D) and push (P) strokes. Again, the total accuracy is given in [%] (modified from [Blan17], doi>10.1145/3123021.3123040).

| Stroke Type | $\mu \pm \sigma$ of $\omega_{B,2(y)}$ in $\left[\frac{rad}{s}\right]$ | Accuracy [%] |
|-------------|-----------------------------------------------------------------------|--------------|
| T, D        | - 22.40 $\pm$ 95.78                                                   | 73.5         |
| P           | - 6.06 $\pm$ 41.99                                                    | 75.0         |

## 6.5 Discussion

The achieved outcomes show that it is feasible to estimate the ball speed and spin shortly after impact using only one single racket-mounted IMU sensor. Although several ball and racket assumptions as well as motion simplifications were made, the ball speed of topspin and drive strokes could be estimated with an overall accuracy of 79.4 % resulting in a mean error - 0.09  $\pm$  0.59 m/s. Pushes were computed with an accuracy of 87.4 % resulting in a mean error of - 0.01  $\pm$  0.57 m/s. The ball spin could be calculated less accurate with 73.5 % and a mean rotation error of - 22.40  $\pm$  95.78 rad/s for topspin and drive strokes and with an accuracy of 75.0 % and a mean rotation error of - 6.03  $\pm$  41.99 rad/s for push strokes.

Compared to other approaches for ball spin and speed estimation in literature, which used almost exclusively video analysis methods and thereby reached total accuracies from 75 % to 95 % in [Szep11], [Glov14] and [Zhan15] depending on their complexity, the presented single sensor concept has slightly worse results ( $\rightarrow$  see Chapter 9.1).

However, all results were clearly related to high standard deviations regarding their mean values, which indicated a high error distribution. It is particularly true for the higher ball rotations caused by fast topspins or drives. This fact is distinctly illustrated in Figure 6.11 by large distances of the data measured from the ideal estimation line and additionally in Figure 6.13 by a wide data spread around the mean as well as exceeding the limits of agreement within the *Bland-Altman* plot. One explanation could be the mutual transformation of speed and spin during the impact. Regarding assumptions (A1) and (A2), an average speed  $v_\mu$  was suggested for the unknown approaching ball, depending of the performed stroke type. But actually, this is only a rough estimation for some return cases, which could probably cause large outliers within the

presumed ball speed before impact. For example, a topspin can be performed in steeper or flatter ways resulting in similar speed values but with a different amount of spin.

Furthermore, strokes do not have to be returned with the same stroke type in the context of similar ball speeds for fast approaching balls like topspins or drives ( $T, D \rightarrow T, D$ ) as well as for slow approaching balls like pushes ( $P \rightarrow P$ ). Thereby, the ball spin direction is reversed. But it is also possible to return the ball in the context of same ball rotation directions. This technique could be used to return a push stroke by a strong topspin ( $P \rightarrow T$ ), whereby the ball rotation direction is preserved and reinforced. Admittedly, a push stroke cannot be performed to return an approaching topspin ball ( $T \nrightarrow P$ ). Due to the elastic characteristics of the rubber, the ball would bounce in unpredictable directions, depending on the racket angle during impact. Nevertheless, different materials like pimples or antispin rubbers could be used to perform those strokes, but this was not considered within this thesis.

In addition to  $v_\mu$ , the fixed incident impact angles for  $\zeta_{geo, (T, D)}$  and  $\zeta_{geo, (P)}$  as a rough estimation were considered in (A3), but  $\zeta_{geo}$  had a strong influence on the component-wise summation of the mean ball velocities  $v_{\mu, (T, D)}$  and  $v_{\mu, (P)}$  with the racket speed  $v_R(x_i)$ . Considering real batting situations, the ball can be hit with similar angles, but it is not achievable to hit the ball with the exact same angles for similar stroke types. Consequently, both factors  $v_\mu$  and  $\zeta_{geo}$  contributed in an uncertain manner to the error. This could be one reason for the observed underestimated ball speed and spin results over all stroke types, shown in Table 6.3 and Table 6.4. Contrarily, low speed values during high ball rotations (true for pushes) were estimated more accurately. On the one hand, this is caused by a less complex stroke movement, which leads to decreased motion variabilities. On the other hand, pushes are performed with more reliability, independent of the players' ability. Hence, the initial guesses for  $v_{\mu, (P)}$  and  $\zeta_{geo, (P)}$  were more applicable leading to better ball speed and spin results.

Considering the estimated ball speed distribution from low to high values, the calculation performed best for the middle third of all measured values (see Figure 6.10). This is also shown in the *Bland-Altman* (see Figure 6.12), where the mean ball speed values for low a high ball speeds are more spread and come close to the confidence intervals. Figure 6.11 shows that negative ball rotations (pushes), nearly no and minimal positive ball rotations (drives) were well estimated. This is

again illustrated in the *Bland-Altman* in Figure 6.13. Here, the ball spin differences for push strokes are grouped near the mean values, whereas topspin and drive spins are widely spread and additionally often located outside the limits of agreement. An explanation for this discrepancy could on the one hand be caused by the used labeling method ( $\rightarrow$  see Chapter 6.1) and on the other hand by the proposed study design ( $\rightarrow$  see Chapter 6.2). For low ball velocities, the absolute difference  $\|\mathbf{p}_B - \mathbf{p}_A\|$  of both ball center displacement vectors  $\mathbf{p}_A$  and  $\mathbf{p}_B$  of two consecutive video frames could be too small to be well recognizable, since the pixel width  $p_w$  was relatively high due to the low resolution of the high-speed camera. As a result, this term directly influences the labeled ball speed  $v_b$  and finally the evaluation of  $v_{B,2(xz)}$ .

Regarding high spin rotations, the low resolution affected both feature vectors  $\mathbf{f}_A$  and  $\mathbf{f}_B$  in the same uncertain way as their appropriate ball center displacement vectors  $\mathbf{p}_A$  and  $\mathbf{p}_B$ . The intention of the data collection was to force the return in the same table corner as the initiated pass to ensure parallel ball speed vector towards the image plane. However, slightly diagonal returns could not be avoided. Therefore, a slow but steady increasing ball envelope  $A_b$  appeared for higher frame numbers. This in turn led to more inaccurate computations of the ball center and feature marking displacement vectors  $\|\mathbf{x}_A - \mathbf{p}_A\|$  and  $\|\mathbf{x}_B - \mathbf{p}_B\|$  affecting the rotation angle  $\alpha_{rot}$  and finally impacting the spin values for  $\omega_b$  and the evaluation of the ball rotation  $\omega_{B,2(y)}$ .

A deeper look into the reference video reveals, that the impact position  $\mathbf{c}_{geo}$  is usually not the center of the racket blade, independent of player and stroke type. Nevertheless, the impact position was defined as the blade's center ( $A_4$ ). With assumption (A6), the wrist position  $\delta_{geo}$  and  $\mathbf{a}_{geo}$  concerning blade center and coordinate origin was furthermore fixed for every subject. Thus, the wrist transformation matrix  $\mathbf{W}$  was also estimated and could lead to inaccuracies the calculation of the rotational racket speed component  $\mathbf{v}_{r,R}$ . Modifying these racket and physiological parameters can quantify the total influence of  $\mathbf{W}$  and the resulting accuracies. Experimentally, wrist geometry deviations for  $\delta_{geo}$  with a maximum error of  $\delta_{geo,\Delta} = \pm[10 \ 10 \ 10]^T [deg]$ , for  $\mathbf{a}_{geo}$  and  $\mathbf{c}_{geo}$  with  $\mathbf{a}_{geo,\Delta} = \pm[10 \ 10 \ 10]^T [mm]$  and  $\mathbf{c}_{geo,\Delta} = \pm[10 \ 10 \ 10]^T [mm]$  respectively for the Z-axis, resulting in an error of less than 1 % caused therefore no significant error within the computation of the angular velocity speed component  $\mathbf{v}_{r,R}$ . However, the same deviations in X-direction and Y-direction lead to much higher errors of up to 15 % for the  $\mathbf{v}_{r,R}$

component. Looking at Figure 6.4, the X-axis and the Y-axis spanned the main racket movement plane and defined the geometric relation between impact position  $c_{geo}$  and wrist center  $a_{geo}$  within the local racket coordinate frame. Thus, this geometric dependency has obviously high implications for the transformation of the rotational speed component and the contrived assumptions were not optimal regarding the racket motion simplification.

Certainly, primary accountable for inaccuracies were the strongly simplified motion equations in (A5) and (A7), by breaking down the complex motions of multiple degree-of-freedom systems to only one measurable segment (movement of table tennis racket). With the neglect of leg, arm and body interactions, too much information was presumably lost to characterize the entire stroke sequence.

Moreover, both integrations of acceleration  $a_R(x_i)$  and angular velocity  $\omega_R(x_i)$  can lead to higher drifts within longer integration periods. Besides that, it was assumed by (A7) that the starting point for integration  $x_m$  was set by the minimization of  $(\Lambda(x_i) - g_{earth})$  before impact and there was no further racket motion. But this did not include any linear or non-accelerated movement components, because these are not recognizable by an accelerometer or a gyroscope. To address this issue, a known resting point in a global world frame must be known, but only inertial data from the local racket frame were available as it was the intention of this approach. Furthermore, this resting point was only based on data of the accelerometer but also used as starting point for integration of the angular velocity. This required the implicit assumption that accelerometer and gyroscope showed no motion at the same time, but in practice, both events must not coincide. However, all other mentioned indicators for  $x_m$  described in (A7) showed worse results.

In addition to the aspects mentioned above, the rebound model based on [Naka 09] and [Naka 10] contains different assumptions and simplifications as well. According to their contribution, the authors always assumed point contacts between the ball and the racket during the impact. This implied, that no momentum effects apart from the perpendicular blade direction in Y-axis were considered during the transformation of ball speed and spin. However, this assumption is not reasonable, due to the pimple structure of the rubbers' bottom side. During an impact, multiple pimples of the contact areas are bended in random and unpredictable directions. Thus, slight momentums parallel to the XY-plane of the blade are generated, which drain energy from the as ideal elastic

defined impact. These effects depend on the rubber structure and impact position. Furthermore, ball and rubber deformations, any energy loss during the elastic impact process as well as air resistance were neglected. This resulted in wrongly estimated elasticity constants  $k_{ip}$  and consequently in erroneous transformation matrices for sliding friction case (II).

## 6.6 Summary

This chapter introduced an approach for the estimation of ball speed and spin using only one single racket-mounted IMU sensor. Motion data of different stroke types resulting in various ball speed and spin values were collected in a research study. Topspin strokes, drives and pushes were performed by table tennis club players to acquire ball rotations in forward and backward directions as well as slow and fast approaching ball velocities. With assumptions for unknown initial ball and racket properties, subject-dependent physiological characteristics and simplifications of the overall motion equations, the ball speed and spin shortly after impact could be estimated.

First of all, a suitable methodology for evaluation was developed. Using high-speed camera frames, different ball features, the ball center and the ball envelope were extracted to calculate the absolute ball speed and spin by pixel displacements over time. This method was finally verified by a pendulum approach, where certain ball speeds and rotations could be adjusted. Evaluation results with relative errors of - 2.3 % for ball speed and - 8.4 % for ball spin confirmed this approach as reference method, since it has similar outcomes as the approaches in literature.

Then, the initial ball speed and spin were estimated by computation of the linear and rotational racket motion components. On the one side, the gravity compensated racket acceleration was integrated based on several conditions and restrictions to conclude the racket velocity during impact. On the other side, all rotational racket parts were transformed by a special wrist transformation matrix into linear components and added to the final racket velocity during impact. Then, with the help of special rebound modeling between rigid balls and elastic surfaces, the wanted ball properties shortly after impact could be estimated by conversion of speed into spin and vice versa.

The ball speed results showed, that the forward spin strokes were estimated with an accuracy of 79.4 % and a mean error of - 0.09  $\pm$  0.59 m/s,

whereas the backward spin strokes were computed with an accuracy of 87.4 % and a mean error of  $-0.01 \pm 0.57$  m/s. The spin was estimated with an accuracy of 73.5 % and of 75.0 %, respectively. Here, mean errors of  $-22.40 \pm 95.78$  rad/s and  $-6.03 \pm 41.99$  rad/s were achieved.

To sum it up, all results showed that it is possible to estimate ball speed and spin with one single racket-mounted IMU sensor using different assumptions and simplifications.

## Chapter 7

# Ball Impact Localization

Within this chapter, the ball impact position on the racket is analyzed. Vibration data from three independent piezo-electric shock sensors mounted at the outer edge of the blade are investigated. The ball impact position is finally localized using an approach similar to detecting epicenters of earthquakes and different regressing techniques.

The following sections are organized as follows: at the beginning, an overview of the theory about standard triangulation methods, vibration propagation and technical challenges especially for this scope is given. Then, data acquisition and the piezo-electric measurement setup are described. Data processing is explained focusing on impact detection, generation of the developed time difference model and the actual impact localization. Subsequently, all results are evaluated and discussed. A final summary recapitulates the content of this chapter.

The data acquisition and algorithms for the ball impact localization have already been published in [Blan 16b].

### 7.1 Theory

The intention of the ball impact localization methods was to compensate the mentioned disadvantages of available and presented approaches (→ see Chapter 1.2). On the one hand, the focus thereby was to minimize the number of sensors, cables and electronics, which would have significant influence on the overall racket characteristics. On the other hand, the proposed system should fulfill the general regulations of the ITTF, which states among others that “[a]t least 85 % of the blade by thickness shall be of natural wood [...]” [The 18a]. Therefore, this approach uses only three small-sized piezo-electric shock sensors mounted

on the outer edge of the blade resulting in minimal hardware efforts and almost no disturbances of the originally desired racket characteristics. The presented system is similar to the localization methodology of earthquakes, where time differences of significant amplitude changes from separate incoming P-waves and S-waves with unique geological origins are measured by several monitoring stations [Inco 14]. The epicenter is then located with different triangulation methods like *Time-of-Arrival* (ToA) or *Time-Difference-of-Arrival* (TDoA) principles. If this context is reduced from a three-dimensional spatial consideration on earth to a two-dimensional surface view and translated to a table tennis application, the epicenter correlates with the ball impact position, the monitoring stations with the piezo-electric sensors and the seismic waves with the induced ball impact vibrations through the racket, whereas the processing principles themselves stay unchanged. If using the ToA principle, absolute time stamps for accurate synchronization between all receivers are necessary, since further processing requires the measurement of all propagation times [Otex 17]. For instance, this absolute time could be provided by a certain transmitter with a known initial position. As the points in time of the earthquake and the ball impact are unknown, time references were not applicable for this case. Instead, only time differences between the receivers could be measured by the TDoA principle.

### 7.1.1 Time-Difference-of-Arrival

The two-dimensional TDoA approach in X-direction and Y-direction is based on time measurements of at least  $> 3$  different sensors  $n_s$ .

$$H_{i,j}^A = \sqrt{(S_{ip,X}^i - P_{ip,X})^2 + (S_{ip,Y}^i - P_{ip,Y})^2} - \sqrt{(S_{ip,X}^j - P_{ip,X})^2 + (S_{ip,Y}^j - P_{ip,Y})^2} \quad (7.1)$$

$$i, j \in \{1, \dots, n_s\}$$

$$n_s = \text{Number of sensors}$$

$$H_{i,j}^A = \text{Hyperbolic curve in } [m]$$

$$S_{ip,X}^i, S_{ip,X}^j, S_{ip,Y}^i, S_{ip,Y}^j = \text{Sensor positions in } [m]$$

$$P_{ip,X}, P_{ip,Y} = \text{Impact positions in } [m]$$

These time difference measurements  $\Delta t_{i,j}$  between significant changes of signal amplitudes for two single sensor positions  $S_{ip}^i$  and  $S_{ip}^j$  with  $i, j \in \{1, \dots, n_s\}$ . Mathematically, all identical propagation time differences  $\Delta t_{i,j}$  of a single vibration wavefront between the impact position  $P_{ip}$  and the sensor locations  $S_{ip}^i$  and  $S_{ip}^j$  are located on a hyperbolic curve  $H_{i,j}^A$  (Equation 7.1) with an eccentricity  $e_H$  of 0. Assuming a known and constant vibration propagation speed  $c_{vib}$ , the hyperbolic curve  $H_{i,j}^B$  (Equation 7.2) can also be expressed as follows:

$$H_{i,j}^B = \Delta t_{i,j} \cdot c_{vib} \quad (7.2)$$

$$i, j \in \{1, \dots, n_s\}$$

$$n_s = \text{Number of sensors}$$

$$H_{i,j}^B = \text{Hyperbolic curve in } [m]$$

$$\Delta t_{i,j} = \text{Time difference in } [s]$$

$$c_{vib} = \text{Propagation speed in } [m/s]$$

The resulting equation system can only be solved numerically by an optimization algorithm with respect to  $P_{ip,X}$  and  $P_{ip,Y}$ . The impact position  $P_{ip}$  (Equation 7.3) can finally be computed as the intersection of all hyperbolic curves  $H_{i,j}^\#$ , with  $\# = \frac{(n_s)!}{2 \cdot (n_s-2)!}$  possible combinations. For example, this procedure could be done by a least squares minimization algorithm [Jin18].

$$P_{ip} = \operatorname{argmin} \left\{ \sum_{n_s} (H_{i,j}^A - H_{i,j}^B) \right\} \quad (7.3)$$

$$i, j \in \{1, \dots, n_s\}$$

$$n_s = \text{Number of sensors}$$

$$P_{ip} = \text{Impact position in } [m]$$

$$H_{i,j}^A, H_{i,j}^B = \text{Hyperbolic curves in } [m]$$

However, this theoretical approach supposes a constant propagation speed  $c_{vib}$  over all racket distances equally in X-direction and Y-direction and no measurement offset  $t_{off,i,j}$  between any data samples. An example with three sensor positions  $S_{ip}^1$ ,  $S_{ip}^2$ , and  $S_{ip}^3$  on a table tennis racket

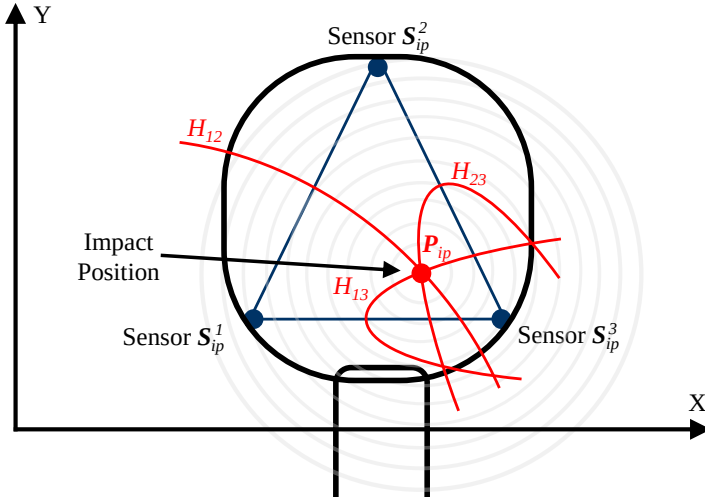


**Figure 7.1:** Theoretical TDoA measurement approach on the surface of a table tennis racket. All sensors are fixed at the positions  $S_{ip}^1$ ,  $S_{ip}^2$ , and  $S_{ip}^3$  on the outer edge of the racket with a maximum distance between each other to achieve longest possible time differences  $\Delta t_{i,j}$ . The hyperbolic curves  $H_{1,2}$ ,  $H_{1,3}$  and  $H_{2,3}$  have therefore a definite intersection at impact point  $P_{ip}$ , from which all vibrations propagate circularly.

can be seen in Figure 7.1. All hyperbolic curves  $H_{1,2}$ ,  $H_{1,3}$  and  $H_{2,3}$  have a definite intersection at impact point  $P_{ip}$ .

### 7.1.2 Vibration Propagation

As mentioned beforehand, the traditional TDoA approach requires a constant propagation speed through the whole surface. A detailed look into the profile of a standard assembled racket including classic pimples inside the rubbers (see Figure 7.2) meanwhile shows a very inhomogeneous structure.

However, a uniform medium is requirement for a time and place invariant omnidirectional sound speed propagation. In case of an impact on the surface, the impact force generates vibrations, which propagate with the appropriate sound speed of the layer in any direction through the racket. Firstly, the vibrations have to pass elastic layers comprising the flat rubber finish, the pimples and the sponge. Here, first dis-

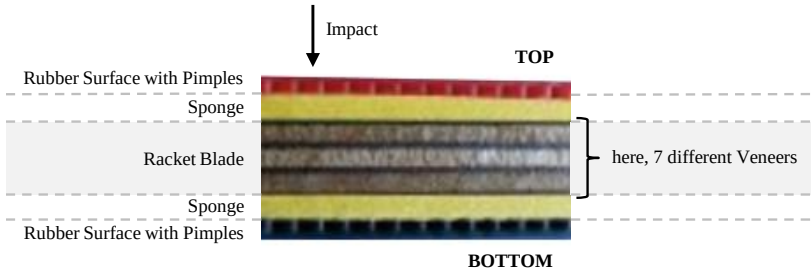


**Figure 7.2:** Profile view of a racket assembled with classic pimples inside rubbers. Top and bottom side of the racket have a flat rubber surface, followed by a layer of pimples and a sponge. The blade itself consists of different kinds and numbers of wooden or carbon veneers. All parts are glued together. Therefore, the impact vibrations have to cross every layer to propagate through the racket.

turbances result from many gaps within the rubber material and sharp edges of the cylindric or trapezoidal pimples. Secondly, the vibration transition from the sponge into the wooden core is determined by a manually applied layer of viscous glue with inconsistencies during application and different thicknesses. Thirdly, all vibrations have to travel through multiple kinds and numbers of wooden or carbon veneers with various densities and qualities.

Wooden veneers are natural products. They are irregular and show a preferred grain orientation. This causes an anisotropical sound speed distribution, since vibrations propagate faster along the fiber structure than orthogonal to the fiber direction [Gree 99]. Typical veneers for table tennis blades are *Hinoki*, *Limba*, *Beech*, *Spruce*, *Balsa* and others [Mani 12a]. Moreover, the grain orientation is rotated for each additional layer for stability reasons. Besides these material inhomogeneities, vibrations could be reflected at the outer edge of the blade resulting in overlapped and erroneous signal amplitudes. Furthermore as in every racket sports, table tennis rackets are manufactured in such a way that the natural frequency of the blade is located at the main hitting area. This *Sweet Spot* can conserve a maximum of the forward momentum from the impact vibration energy, thus, amplifying the return of the ball.

In contrast, a *Dead Spot* absorbs the highest amount of impact energy and therefore suppresses any vibration propagation. Data driven evaluations of sweet spots and dead spots of the example racket used in this

study are also investigated and presented in the end of the results section of this chapter (→ see Chapter 7.4).

The vibration wave type is still unknown. Presuming mostly vertical ball impacts, the expected displacements by the vibration wave are mainly in vertical direction, which concludes a *Transverse Wave* form. Depending on the inclination angle of the impacting ball, the elasticity of the rubber and the sponge as well as the buckling of the inside pimples, the vibration wave could also partly feature *Longitudinal Wave* components.

### 7.1.3 Technical Challenges

All vibration propagation characteristics in combination with the classic TDoA principle lead to some technical challenges.

Firstly, the TDoA must be adapted in a methodological way to handle time differences generated by discontinuous vibration speeds. Instead of numerical localization of the impact position, the approach presented in this thesis tries to match all equal time difference occurrences obtained by multiple trials of a study to predefined measurement points. Subsequently, the generated *Time-Difference-Distribution-Model* (TDDM) attempts to regress the actual impact position using different kinds of optimization algorithms.

Secondly, the vibrations must be sampled fast enough to reconstruct the incoming signal shape and to detect significant changes in the amplitude indicating an incoming wavefront. Sound speeds in structural wooden materials vary strongly with grain direction, temperature, moisture and other environmental influences.

For example, the sound speed  $c_{vib}$  in spruce can reach values of up to  $\sim 6000$  m/s along the fiber orientation [Gree 99]. Assuming a maximum racket diameter of  $\sim 15$  cm, the vibration would pass through the racket within  $\sim 25$   $\mu$ s. If a localization resolution of  $\sim 1$  mm is pursued, according to the *Sampling Theorem* [Whit 28], all data would have to be acquired with a frequency of  $> 120$  MHz.

In practice, the elastic rubbers and glue components as well as a propagation across the grain direction operate as a strong vibration damper, which empirically decreases the expected sound speed  $c_{vib}$  to  $< 150$  m/s. The resulting sampling frequency of  $> 3$  MHz is indeed more realistic to achieve with inexpensive standard consumer microcontrollers, but yet difficult to reach.

## 7.2 Data Acquisition

### 7.2.1 Study Design

For data acquisition, three *PKGSooLDPiR* piezo-electric shock and high acceleration sensors from *muRata*<sup>1</sup> were used. Hereby, the focus lay on the small form factor of the sensors with 6.4 mm length, 2.8 mm width and 1.2 mm thickness to be fully integrated into the wooden racket core in the future. Since these sensors have only one active sensing direction, the measurement axes had to be mounted perpendicular to the racket blade towards the expected amplitudes of the incoming transverse waves. Sensor position and orientation can be seen in Figure 7.3.

Every sensor comprised of the piezo-electric measurement element itself and conditioning electronics on a printed circuit board (PCB) with 12 mm length and 5 mm width resulting in an overall board thickness of 1.7 mm. The total weight of all boards and cables was < 1 % of the wooden

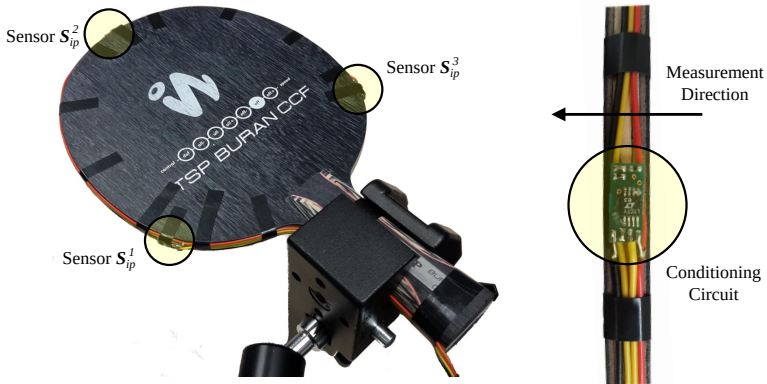


**Figure 7.3:** The left part shows the sensor position and orientation on an example racket with removed rubbers. Three sensors were mounted on the outer edge of the racket. Sensor  $S_{ip}^1$  and sensor  $S_{ip}^3$  were located at same heights measured from the handle on both sides. Sensor  $S_{ip}^2$  was mounted on top of the blade. All sensors were attached with maximum distances to each other. A detailed view including the conditioning circuit board, cables and the sensitive measurement direction can be seen on the right part of the figure (modified from [Blan 16b], doi>10.1145/2971763.2971778).

<sup>1</sup><https://www.murata.com/> – last access on 02/2018

racket weight and therefore in line with the official regulations. A reasonable coupling from the wooden core into the sensor was achieved by fixing all shock elements with an elastic glued tape, which allowed proper vibration forwarding. For further data processing, power supply and data signals were provided by thin cables.

Impact data were collected with multiple racket type and rubber type combinations. Two different blades were used, the *TSP Buran CCF*<sup>2</sup> with a harder wooden material and a weight of 91 g as well as the *Imperial All-round ST*<sup>3</sup> with a softer wooden core and 85 g weight. Both blades were combined with various kinds of rubbers including different degrees of hardness. The hardness of table tennis rubbers is not officially regulated and therefore different between manufacturers. Usually, the hardness is classified by the *Shore Hardness Test* [Shor 10] as it is common in production technology for polymers. It is indicated in degrees from lower values (softer material) to higher values (harder material).

For a wide test set variety, every racket was combined with an *Adidas R6*<sup>4</sup> rubber showing a low hardness, a *JOOLA Energy X-tra*<sup>5</sup> rubber with a medium hardness and a *Tibhar Genius*<sup>6</sup> rubber with a high hardness. All investigated combinations can be seen in Table 7.1. The surface of

**Table 7.1:** All tested racket-rubber combinations. A low Shore hardness indicates softer rubber materials, while high Shore hardness shows harder rubber materials. The right column shows the total weight in [g] of every racket-rubber combination (modified from [Blan 16b], doi>10.1145/2971763.2971778).

| No. | Combination       |               | Hardness<br>in [°] | Weight<br>in [g] |
|-----|-------------------|---------------|--------------------|------------------|
|     | Racket            | Rubber        |                    |                  |
| 1   | TSP Buran         | Tibhar Genius | 46.0               | 140              |
| 2   | TSP Buran         | JOOLA Energy  | 42.5               | 130              |
| 3   | TSP Buran         | Adidas R6     | 35.0               | 130              |
| 4   | Imperial Allround | Tibhar Genius | 46.0               | 124              |
| 5   | Imperial Allround | JOOLA Energy  | 42.5               | 114              |
| 6   | Imperial Allround | Adidas R6     | 35.0               | 114              |

<sup>2</sup><https://www.victas.com/> – last access on 02/2018

<sup>3</sup><https://shopware.imperial-sport.de/> – last access on 02/2018

<sup>4</sup>not available anymore (production stop)

<sup>5</sup><http://www.joola.de/> – last access on 02/2018

<sup>6</sup><http://www.tibhar.com/> – last access on 02/2018

every racket-rubber combination was subdivided into single measuring points  $P_m(X, Y)$  within a grid of 20 mm. The total number of these measuring point  $N_m$  were fixed to 48. The coordinate origin thereby lay at the blade's throat. The distribution of all measuring points  $P_m(X, Y)$  is shown in Figure 7.4.

Although the sensitive area was within  $[-75 ; 75]$  mm in X-direction and  $[0 ; 150]$  mm in Y-direction, points on all columns of the X-axis ranged between  $[-60 ; 60]$  mm, while points on all rows of the Y-axis

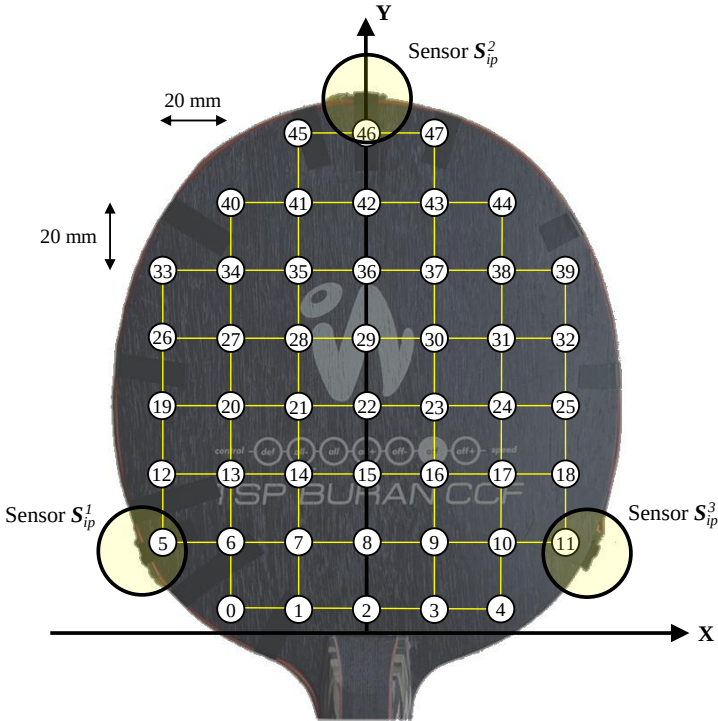


**Figure 7.4:** Distribution of all 48 measuring points  $P_m(X, Y)$  within a grid of 20 mm. The coordinate origin is fixed at the blade's throat equally for every combination. Sensors  $S_{ip}^1$ ,  $S_{ip}^2$  and  $S_{ip}^3$  are mounted on the outer edge and are highlighted with circles. The X-axis represents the transverse direction, that mean orthogonal to grain orientation, whereas the Y-axis indicates the longitudinal direction along fibers (modified from [Blan 16b], doi>10.1145/2971763.2971778).

lay between [5 ; 145] mm. The reason for this was, that it was difficult to reliably hit the ball close to the outer edges. According to this coordinate system, the sensors  $S_{ip}^1$ ,  $S_{ip}^2$  and  $S_{ip}^3$  were located on the positions (- 67 , 25) mm, (0 , 145) mm and (67 , 25) mm, respectively.

For data acquisition, a steady suspension for every racket-rubber combination was needed. Therefore, the rackets were mounted horizontally with a *Manfrotto Magic Arm*<sup>7</sup>, which clamped the handle of the racket, as it would be hold by the player. Table tennis balls were dropped onto the surface ten using with a guided rail to ensure identical impact positions per measuring point  $P_m(X, Y)$ . This procedure was repeated twice per racket-rubber combination to receive a data set for training and a data set for testing. Overall, 960 impacts were collected for each racket-rubber combination.

### 7.2.2 Piezo-electric Measurement Circuit

According to the manual of the used piezo-electric sensors, generated charges were in the range of 85 fC/g [Kita 13]. Due to the extreme small amplitudes, the signal amplification had to be very close to the sensitive sensor element to avoid any resistive and capacitive influences. Therefore, every single sensor had its own individual conditioning board. The circuit components can be schematically seen in Figure 7.5.

Firstly, the vibration signal generated specific electrical charges. The electric charge had to be transformed into an appropriate voltage by a

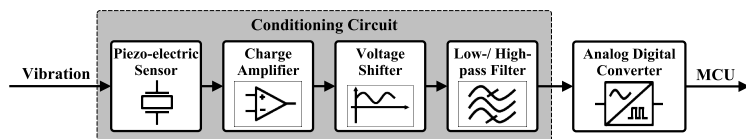


**Figure 7.5:** Component overview of the conditioning circuit. The piezo-electric sensor generated electric charges according to the vibration signal, which had to be transformed into an appropriate voltage. This voltage was then amplified, shifted and filtered by a low-pass and a high-pass filter to process the signal for the final analog-digital conversion (modified from [Blan 16b], doi>10.1145/2971763.2971778).

<sup>7</sup><https://www.manfrotto.com/> – last access on 02/2018

rail-to-rail charge mode amplifier with a gain  $G_{amp}$  of 50 to achieve amplitudes of several 100 mV. Secondly, the converted voltage was biased to the half of the supply voltage  $U_{supply}$  with 3.3 V input to raise the resulting voltage swings from oscillating around 0 V to oscillating around 1.65 V. This was necessary since the *Analog-Digital Converter* (ADC) of most microcontrollers is unable to measure negative voltages. Thirdly, the signal was filtered by a low-pass filter with a cut-off frequency of 100 kHz to suppress high frequency noise, followed by a high-pass filter with a cut-off frequency of 100 Hz to avoid influences by the mains or other low frequency noise impinged by mechanical artifacts.

Finally, the signal was sampled by the integrated ADC of a single-chip *EFM32LG990 Microcontroller Unit* (MCU) from *SiliconLabs*<sup>8</sup>. As the used MCU was able to capture one analog channel every 1.09  $\mu$ s [Sili 14], all three sensors were sampled within a time period  $T_A$  of 3.27  $\mu$ s resulting in a maximum sampling rate of 305.81 kHz. Although the sample-and-hold unit of the ADC guaranteed, that every single analog channel could be captured at the same time, the vibration could propagate forward in time with an unknown offset within  $T_A$ , which had to be considered for further processing steps.

## 7.3 Data Processing

The data processing was divided into three consecutive steps. Firstly, a rough and general impact detection, secondly, the underlying time difference calculation followed by the TDDM modeling and thirdly, the actual localization of the initial impact position.

The impact detection and the time difference calculation during this study were completely implemented on the MCU, whereas TDDM generation and final localization were computed as post processing steps on a more powerful computer.

### 7.3.1 Impact Detection

The data for impact detection was collected continuously using a double-buffered ping-pong method [Sili 14]. Thereby, new incoming data could be concurrently stored into the buffer A by the DMA system of the MCU, while the CPU searched for impacts in past data located in the buffer B. Then, both buffers were interchanged in such a way, that new data could

<sup>8</sup><https://www.silabs.com/> – last access on 02/2018

be saved into buffer B and past data of buffer A could be checked for impacts. This technique was repeated until an impact was detected.

Due to the low sampling interval  $T_A$  and therefore the high resulting number of samples per analog channel as well as the limitation of the integrated 32 kByte RAM [Sili 14] of the MCU, vibration data could only be hold within buffers with a length  $L$  of 600 samples for further processing. This was equal to a time period  $T_A$  of  $\sim 3.3$  ms.

During the impact detection process, each data sample  $x_i$  was compared with two thresholds  $\zeta_{vib,min}$  and  $\zeta_{vib,max}$ . If any sample  $x_i$  exceeded the maximum threshold  $\zeta_{vib,max}$  or had fallen below the minimum threshold  $\zeta_{vib,min}$ , a possible impact was detected. Both thresholds did not need to be very exact, since they should only indicate a present vibration. According to the supply voltage  $U_{supply}$ , they were chosen to be 7/16 of  $U_{supply}$  resulting in 1.44 V for  $\zeta_{vib,min}$  and 9/16 of  $U_{supply}$  resulting in 1.86 V for  $\zeta_{vib,max}$ , respectively. With these adjustments, both limits were far enough away from the quiescent vibration signal to avoid false detections. Referring to the measurement grid in Figure 7.4, example vibration data for the impact point  $P_m(20)$  and the appropriate thresholds can be seen in Figure 7.6.

### 7.3.2 Time Difference Distribution Model

Compared to impact detection, the computation of the TDDM needed a more accurate temporal determination of the incoming vibration wavefront and therefore more precise thresholds.

$$\hat{S}_{ip}^j(x_i) = \frac{S_{ip}^j(x_i) - \mu(S_{ip}^j(x_i))}{\sigma(S_{ip}^j(x_i))} \quad (7.4)$$

$$0 \leq i < \text{Buffer length } L$$

$$j \in \{1, 2, 3\}$$

$$\hat{S}_{ip}^j(x_i) = \text{Normalized sensor signal in } [AU]$$

$$S_{ip}^j(x_i) = \text{Raw sensor signal in } [V]$$

$$\mu(S_{ip}^j(x_i)), \sigma(S_{ip}^j(x_i)) = \text{Mean and standard deviation of}$$

$$S_{ip}^j(x_i) \text{ in } [V]$$

$$x_i = \text{Data sample}$$

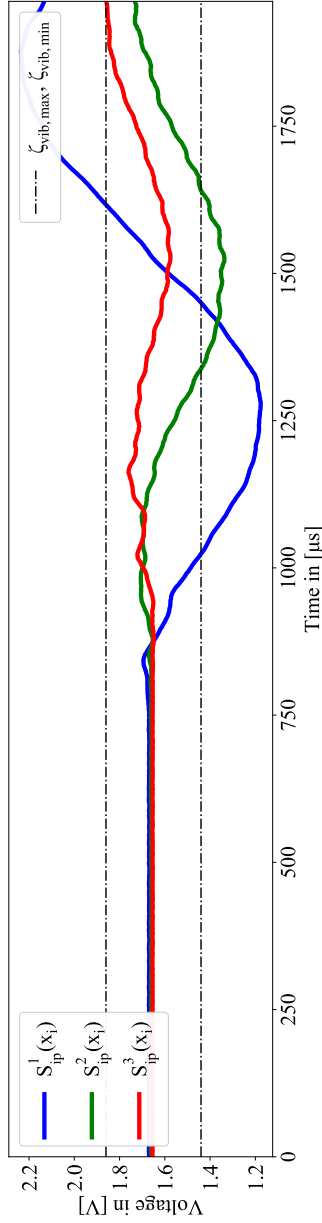


**Figure 7.6:** Example vibration data in [V] for impact point  $P_m(20)$  of the measurement grid. Solid lines in blue, green and red represent the vibration data  $S_{ip}^1(x_i)$ ,  $S_{ip}^2(x_i)$  and  $S_{ip}^3(x_i)$  captured by each sensor, respectively. The dashed-dotted line indicates the threshold limits  $\zeta_{vib,min}$  and  $\zeta_{vib,max}$  for the actual impact detection. Here, signal  $S_{ip}^1(x_i)$  had firstly fallen below  $\zeta_{vib,min}$ , which indicated a possible impact (modified from [Blan 16b], doi>10.1145/2971763.2971778).

Firstly, all three piezo-electric signals were buffer-wise normalized to  $\hat{S}_{ip}^j(x_i)$  (Equation 7.4) by the mean value  $\mu(S_{ip}^j(x_i))$  and the standard deviation  $\sigma(S_{ip}^j(x_i))$  of each sensor. Therefore, the mean value  $\mu(S_{ip}^j(x_i))$  was subtracted from the raw signal  $S_{ip}^j(x_i)$  and the resulting value was divided by the standard deviation  $\sigma(S_{ip}^j(x_i))$  for every data sample  $x_i$ . Secondly, the first 50 samples of each data buffer were used to precise estimate the noise level of the quiescent signal behavior separately for all single sensors.

Hence, the mean value  $\mu_{noise,j}$ , the data maximum  $max_{noise,j}$  and the data minimum  $min_{noise,j}$  of the noise level were calculated. With these parameters, two more accurate thresholds  $\zeta_{high,j}$  (Equation 7.5) and  $\zeta_{low,j}$  (Equation 7.6) for the upper amplitude boundary and lower amplitude boundary could be determined using an empirical knowledge-based factor  $k_{vib}$  of 10.

$$\zeta_{high,j} = \mu_{noise,j} + k_{vib} \cdot max_{noise,j} \quad (7.5)$$

$$\zeta_{low,j} = \mu_{noise,j} - k_{vib} \cdot min_{noise,j} \quad (7.6)$$

$$j \in \{1, 2, 3\}$$

$$k_{vib} = 10$$

$$\zeta_{high,j}, \zeta_{low,j} = \text{Higher and lower threshold in } [AU]$$

$$\mu_{noise,j} = \text{Noise mean value in } [AU]$$

$$min_{noise,j}, max_{noise,j} = \text{Noise minimum and maximum in } [AU]$$

Thirdly, the specific samples  $x_i$ , at which the normalized signals  $\hat{S}_{ip}^j(x_i)$  crossed one of both thresholds  $\zeta_{high,j}$  or  $\zeta_{low,j}$  were marked as relative local time indicators  $\tau_j$  for an occurred vibration within each data buffer. Figure 7.7 shows a more detailed view of the mentioned impact on measuring point  $P_m(20)$  including the normalized signals  $\hat{S}_{ip}^j(x_i)$ , the individual thresholds  $\zeta_{high,j}$  and  $\zeta_{low,j}$  as well as the corresponding time computations  $\tau_j$  for each sensor signal.

Finally, all time differences  $\Delta t_{12}$  (Equation 7.7),  $\Delta t_{13}$  (Equation 7.8) and  $\Delta t_{23}$  (Equation 7.9) for each sensor pair could be computed. Therefore, the appropriate relative times  $\tau_j$  are subtracted and the sampling interval  $T_A$  is multiplied. Taking these time differences, the TDDM could be computed for every specific racket-rubber combination.

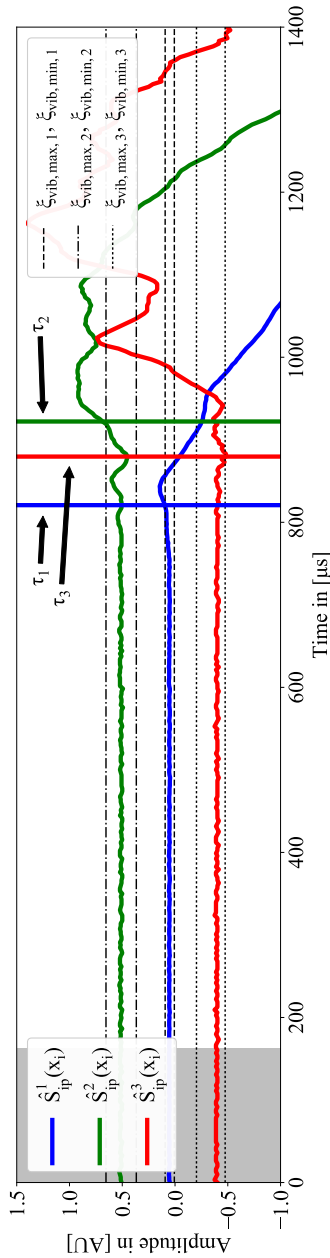


**Figure 7-7:** Detailed view of the vibration for impact position  $P_m$  (20). The normalized signals  $\hat{S}_{ip}(x_i)$ ,  $\hat{S}_{ip}^2(x_i)$  and  $\hat{S}_{ip}^3(x_i)$  [AU] are represented with a solid blue, green and red line, respectively. The grayed block on the left side indicates the data subset used for noise level determination. The horizontal lines display the threshold limits  $\zeta_{high,i}$  and  $\zeta_{low,j}$  for each sensor signal, whereas the arrows reference to the vertical lines, which show the corresponding relative times  $\tau_j$  (modified from [Blan16b], doi>10.1145/2971763.2971778).

Furthermore, an outlier rejection based on the mean and  $q_{\Pi}$ -times the standard deviation was performed to exclude erroneous data during the distribution generation. The rejection distance  $q_{\Pi}$  was thereby 3.

$$\Delta t_{12} = (\tau_1 - \tau_2) \cdot T_A \quad (7.7)$$

$$\Delta t_{13} = (\tau_1 - \tau_3) \cdot T_A \quad (7.8)$$

$$\Delta t_{23} = (\tau_2 - \tau_3) \cdot T_A \quad (7.9)$$

$\Delta t_{12}, \Delta t_{13}, \Delta t_{23}$  = Time differences in [s]

$\tau_{\{1,2,3\}}$  = Relative time indicators

$T_A$  = Sampling interval in [s]

Then, all recorded and non-rejected time differences of one measuring point  $P_m(X, Y)$  were averaged to  $\overline{\Delta t}_{j, N_m}$  (Equation 7.10) and arranged over the XY-coordinate plane of the racket. Since only 48 discrete measuring points  $P_m(X, Y)$  on the 20 mm grid of the racket plane were investigated, the resulting time difference distribution was discrete. The remaining gaps in between the grid had to be filled to handle time differences that were not yet represented by the model, thus, overcoming this disadvantage. Therefore, an interpolation algorithm using the convex hulls approach developed by [Barb 96] was used to upgrade the grid to a 1 mm pattern over the whole sensitive racket area.

$$\overline{\Delta t}_{j, N_m} = \frac{1}{N_{\Pi}} \cdot \sum_{i=1}^{N_{\Pi}} \Delta t_{j, N_m} \Big|_{\text{abs}\{\Delta t_{j, N_m}\} < \Pi_{j, N_m}} \quad (7.10)$$

$$\Pi_{j, N_m} = \mu_{\Delta t_{j, N_m}} + q_{\Pi} \cdot \sigma_{\Delta t_{j, N_m}}$$

$N_{\Pi}$  = Number of not rejected values

$0 \leq N_m < 48$  measuring points

$j = \{12, 13, 23\}$

$q_{\Pi} = 3$

$\overline{\Delta t}_{j, N_m}$  = Averaged time differences in [s]

$\Delta t_{j, N_m}$  = Time differences in [s]

$\mu_{\Delta t_{j, N_m}}, \sigma_{\Delta t_{j, N_m}}$  = Mean and standard deviation of  $\Delta t_{j, N_m}$  in [s]

The resulting TDDM can be seen in Figure 7.8 exemplary for racket-rubber combination (No. 1) and impact point  $P_m(20)$ .

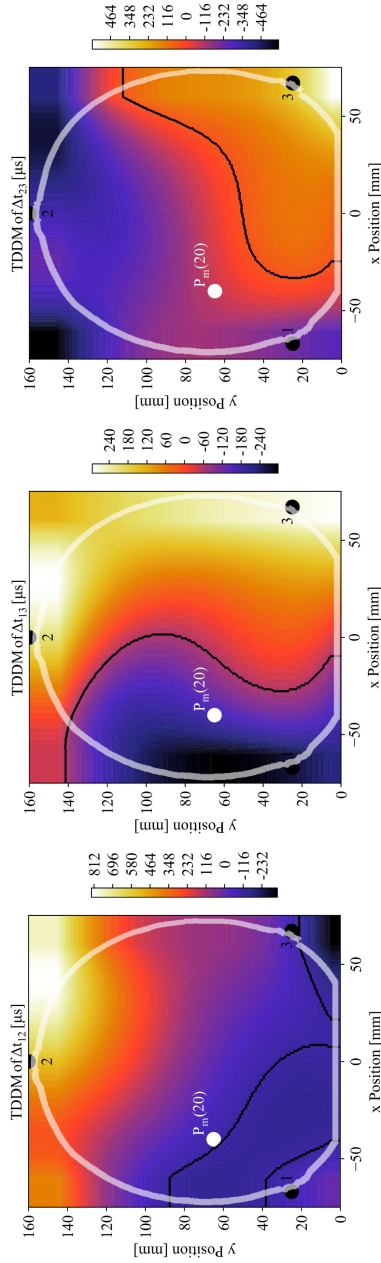


**Figure 7.8:** Example interpolated TDDM as colored maps for the time differences  $\Delta t_{12}$  of -31  $\mu s$ ,  $\Delta t_{13}$  - 18  $\mu s$  and  $\Delta t_{23}$  13  $\mu s$  for the racket-rubber combination (No. 1) and impact on  $P_m(20)$  (white circle). Equal time differences are indicated with one color allocated over the XY-coordinate system of the racket. All sensor positions 1, 2 and 3 are marked with black circles, whereas the racket boundary is indicated by the white contour. Here, black lines show all equal time occurrences for the specific measured impact time differences between one sensor pair (modified from [Blan16b], doi>10.1145/2971763-2971778).

Each interpolated time difference occurrences for  $\Delta t_{12}$ ,  $\Delta t_{13}$  and  $\Delta t_{23}$  of the training set data were mapped onto the racket surface with an individual color on a separate map for better visual understanding. All equal time differences were indicated as single forming lines within these colored maps. Figure 7.8 clearly shows, that all time differences are highly non-linear distributed and that the traditional TDoA approach using hyperbolic curves as assumption for the time difference distribution would fail. However, for each time difference map model, an increasing and continuous propagation time tendency between two sensors could be conducted. This behavior served also as basis for initial assumptions to derive the impact position using only three sensors by the following triangulation approaches.

### 7.3.3 Impact Localization

The actual localization of the initial impact position on the blade was done by investigation and evaluation of three different methods. These approaches differed in modalities like complexity, computational effort and runtime. For simplicity, the following calculations are considered as unitless.

The simplest algorithm was a *Decision Tree* (DT) based approach. This is the only method, for which no TDDM has to be generated and no interpolation techniques are needed. Here, all three time differences  $\Delta t_j$  captured from an impact event were compared to those time differences  $\overline{\Delta t}_{j,N_m}$ , which were assigned to each measuring point  $P_m(X, Y)$  of the training data set. Thereby, the comparison was not mathematically performed by the number of the value but only regarding the algebraic sign of the value. This was done for all 48 measuring points and for each sensor pair.

$$I_{DT}(X, Y) = \frac{1}{N_{cand}} \sum_{i=1}^{N_{cand}} P_m(X, Y) \Big|_{\text{sgn}(\Delta t_j) = \text{sgn}(\overline{\Delta t}_{j,N_m})} \quad (7.11)$$

$$N_{cand} = \text{Number of marked candidates } P_{ip}(\hat{X}, \hat{Y})$$

$$0 \leq N_m < 48 \text{ Measuring points}$$

$$j = \{12, 13, 23\}$$

$$I_{DT}(X, Y) = \text{Most likely impact position}$$

$$\overline{\Delta t}_{j,N_m} = \text{Averaged time differences}$$

If the sign coincided with the sign of a measuring point  $P_m(X, Y)$ , it was marked as possible candidate  $P_{ip}(\hat{X}, \hat{Y})$ . Subsequently, the average based on all identified candidates regarding X-direction and Y-direction indicated the impact position  $I_{DT}(X, Y)$  (Equation 7.11) on the blade.

If no candidate could be found during all comparisons or if the finally calculated position laid outside the racket boundary, the input data were discarded and the impact position could not be localized.

The second investigated method was an approach based on *Linear Regression* (LR). For a specific time difference set  $\Delta t_j$  from one impact, all equal time difference occurrences  $\Delta t_{j,tdm}$  from the generated TDDM of the training data set (indicated as black lines and dots in Figure 7.8) were accumulated and used as input for three independent linear regression functions  $f_j(X)$ . Despite the non-linear distribution of the time differences  $\Delta t_{j,tdm}$  within the model, a significant gradient between two sensors was recognizable. Thus, the correlation for  $\Delta t_{12}$  could be expressed as a linear function with a negative slope (Equation 7.12) and the correlation of  $\Delta t_{23}$  was represented by a linear function with a positive slope (Equation 7.14), respectively.

However, best results could be achieved by an inverse linear function (Equation 7.13) for  $\Delta t_{13}$ , depending on the sign of the measured time difference. In the following equations, all slopes were defined as  $a_j$  and all intercepts were indicated by  $b_j$ . Both parameters were computed by a least-squares optimization algorithm.

$$f_{12}(X) = -a_{12} \cdot X + b_{12} \quad (7.12)$$

$$f_{13}(X) = \begin{cases} -\frac{1}{a_{13}} \cdot (X - b_{13}) & \Delta t_{13} > 0 \\ \frac{1}{a_{13}} \cdot (X - b_{13}) & \Delta t_{13} \leq 0 \end{cases} \quad (7.13)$$

$$f_{23}(X) = a_{23} \cdot X + b_{23} \quad (7.14)$$

$$a_j, b_j = \text{Slopes and intercepts} \\ j = \{12, 13, 23\}$$

$$f_{12}(X), f_{13}(X), f_{23}(X) = \text{Regression functions}$$

$$\Delta t_{13} = \text{Time difference from sensor } S_{ip}^1 \text{ and } S_{ip}^3$$

Due to measurement errors, unknown analytical system characteristics and simplified assumptions, it was almost impossible, that all three functions  $f_j(X)$  intersected at one single point within the coordinate system. Thus, the point which had the shortest orthogonal distance

$d_j(\hat{X}, \hat{Y})$  to each function graph was assumed to be the most likely impact position  $I_{LR}(X, Y)$ . This line distances  $d_j(\hat{X}, \hat{Y})$  (Equation 7.15) could be computed using the dot product relations and an initial impact position guess for the X-position  $\hat{X}$  and the Y-position  $\hat{Y}$ .

$$d_j(\hat{X}, \hat{Y}) = \frac{\text{abs}\{-a_j \cdot \hat{X} + \hat{Y} - b_j\}}{\sqrt{a_j^2 + 1}} \stackrel{!}{=} \text{const.} \quad (7.15)$$

$d_j(\hat{X}, \hat{Y}) =$  Line distances

$a_j, b_j =$  Slopes and intercepts

$j = \{12, 13, 23\}$

$\hat{X}, \hat{Y} =$  Initial guess

As multiple points  $P(\hat{X}, \hat{Y})$  could fulfill above condition, a solution where all distances were minimized was implemented. Hence, the minimum sum (Equation 7.16) for all points  $P(\hat{X}, \hat{Y})$  indicated the origin impact position  $I_{LR}(X, Y)$ .

$$I_{LR}(X, Y) = \text{argmin} \left\{ \sum_{i=1}^{N_{\text{cand}}} d_j(\hat{X}, \hat{Y}) \right\} \quad (7.16)$$

$N_{\text{cand}} =$  Number of possible impact points  $P(\hat{X}, \hat{Y})$

$j \in \{12, 13, 23\}$

$I_{LR}(X, Y) =$  Most likely impact position

$d_j(\hat{X}, \hat{Y}) =$  Line distances

Here again, if the calculated impact position was localized outside the racket dimensions, the input was discarded and no valid impact position could be estimated.

As a third method, a *Kernel Density Estimation* (KDE) was evaluated [Parz 62]. In this context, the kernel density function  $f_{kde}(X, Y)$  was estimated in a multivariate way using the positions in X-direction, Y-direction and a Gaussian kernel  $K_G$ . All points  $P(X, Y)$  from the generated TDDM, which have the same time differences for  $\Delta t_{12}$ ,  $\Delta t_{13}$  and  $\Delta t_{23}$  as the currently measured time differences, were combined into one single distribution model. Then, the kernel density estimation  $f_{kde}(X, Y)$  (Equation 7.17) was calculated with *Scott's Rule* [Scot 15]

as selected bandwidth  $h_{kde}$  using the new distribution within the same 1 mm grid.

$$f_{kde}(X, Y) = \frac{1}{N_{grid} \cdot h_{kde}^2} \cdot \sum_{i=1}^{N_{grid}} K_G \left( \frac{P(X, Y) - P_{grid}(X, Y)}{h_{kde}} \right) \quad (7.17)$$

$$K_G(z) = \frac{1}{\sqrt{2\pi}} \cdot \exp \left( -\frac{1}{2} z^2 \right) \quad (7.18)$$

$N_{grid}$  = Number of all points  $P_{grid}(X, Y)$  within the grid

$h_{kde} = N_{grid}^{-\frac{1}{6}}$  = Bandwidth using Scott's rule

$f_{kde}(X, Y)$  = Kernel density estimation

$P(X, Y)$  = Related points of the generated TDDM

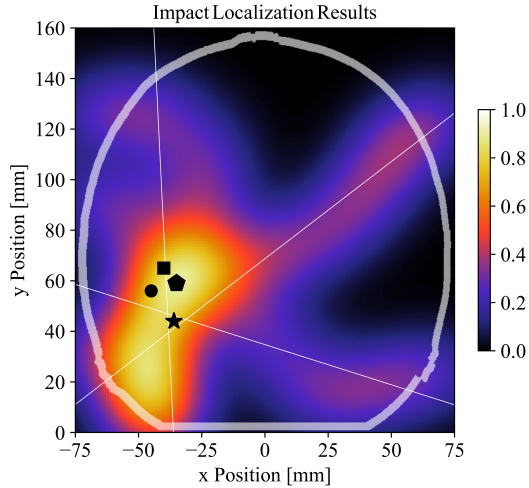


**Figure 7.9:** Visual localization results for an impact on blade position  $P_m(20)$  as black square (■). The estimations for the decision tree (●), the linear regression (★) and the kernel density estimation (◆) are marked. The linear regression lines are indicated as white lines for each time difference. The probability of the kernel density estimation is shown as a colored map ranging from 0.0 as low probability to 1.0 as high probability. The racket boundary is defined by a white shape (modified from [Blan 16b], doi>10.1145/2971763.2971778).

Afterwards, the most likely impact position  $I_{KDE}(X, Y)$  (Equation 7.19) was defined as the point where  $f_{kde}(X, Y)$  had its global maximum.

$$I_{KDE}(X, Y) = \operatorname{argmax}\{f_{kde}(X, Y)\} \quad (7.19)$$

$I_{KDE}(X, Y)$  = Most likely impact position

$f_{kde}(X, Y)$  = Kernel density estimation

In Figure 7.9 exemplary results of an ball impact on the racket position  $P_m(20)$  can be seen for the decision tree as black circle (●), the linear regression as black star (★) including the regression lines and the kernel density estimation as black pentagon (◆) with the probability distribution between 0.0 and 1.0 as colored background map, as well as the initial impact position as black square (■).

## 7.4 Evaluation and Results

The evaluation was based on the TDDMs generated in the training data sets of every racket-rubber combination. These models were accordingly validated by the data test sets of equal and different racket-rubber combinations using the data of all 48 measuring points  $P_m(X, Y)$ . The intention behind this approach was to investigate, how applicable a model to other unknown rackets, rubbers or combinations of these, was.

Additionally, the *Sweet Spot* area was separately evaluated omitting the measuring points near the outer edge and the handle. Furthermore, the bidirectional error distribution according to the coordinate system was determined to identify influences on the vibration propagation caused by irregularities within the wooden racket and elastic rubber material. Quantitatively, the impact positions  $I(X, Y)$  for each localization method were calculated and compared to their appropriate labeled equivalent  $P_m(X, Y)$ . The resulting errors were described by the root mean square error (RMSE) (Equation 7.20) for X-direction and Y-direction. Overall, data of 3840 impacts were recorded.

Regarding Table 7.1, test data sets were collected for each of the six racket-rubber combinations, whereas training data sets for the TDDM generation were only acquired from combination (No. 1) and (No. 4). Every data set contained 10 impacts per measuring point  $P_m(X, Y)$ , which resulted in 480 impacts per combination. The TDDMs generated by data of combinations comprising the same racket type were finally

evaluated against each other. The results for the TDDM generated by combination (No. 1) are shown in Table 7.3, whereas the results for the TDDM combination (No. 4) are presented in Table 7.4.

$$\text{RMSE}(p) = \sqrt{\frac{1}{N_{ip}} \sum_{i=1}^{N_{ip}} \left( P_m(q) - I(q) \right)^2} \quad (7.20)$$

$N_{ip}$  = Number of investigated impacts

$p \in \{X, Y\}$

$\text{RMSE}(p)$  = Root mean square error in  $[m]$

$P_m(p)$  = Measuring points in  $[m]$

$I(p)$  = Impact points in  $[m]$

Results for the sweet spot area for TDDM and test set combination (No. 1) can be seen in Table 7.2. The sweet spot area contained only those measuring points  $P_m(X, Y)$ , which did not lay at the outer edge of the racket handle. Therefore, measuring points 7, 8, 9, 13, 14, 15, 16, 17, 20, 21, 22, 23, 24, 27, 28, 29, 30, 31, 35, 36 and 37 were used as input for the evaluation. An example error distribution based on the RMSE can be seen in Figure 7.10 for the racket-rubber combination (No. 4) and the LR method.

**Table 7.2:** Evaluation results for the sweet spot area are shown for the TDDM based on racket-rubber combination (No. 1). RMSE in [mm] achieved by the decision tree (DT), the linear regression (LR) and kernel density estimation (KDE) is given for test data of combination (No. 1). Additionally, the total number of localized impacts out of all usable impact data and the corresponding allocation rate is presented (modified from [Blan 16b], doi>10.1145/2971763.2971778).

| Combination<br>Test Set → TDDM | (No. 1) → (No. 1) |      |      |
|--------------------------------|-------------------|------|------|
| Method                         | DT                | LR   | KDE  |
| RMSE(X) in [mm]                | 20.3              | 18.3 | 18.9 |
| RMSE(Y) in [mm]                | 22.7              | 19.2 | 20.3 |
| Localized Impacts              | 338/342           |      |      |
| Allocation Rate                | 98 %              |      |      |

**Table 7-3:** Evaluation results for the TDDM based on racket-rubber combination (No. 1). RMSE in [mm] achieved by the decision tree (DT), the linear regression (LR) and kernel density estimation (KDE) is given for test data of combination (No. 1), (No. 2) and (No. 3). Additionally, the total number of correctly localized impacts out of all allocated impacts is presented in numbers and [%] (modified from [Blan 16b], doi>10.1145/2971763.2971778).

| Combination<br>Testset → TDDM |  | (No. 1) → (No. 1) |      |      | (No. 2) → (No. 1) |      |      | (No. 3) → (No. 1) |      |      |
|-------------------------------|--|-------------------|------|------|-------------------|------|------|-------------------|------|------|
| Method                        |  | DT                | LR   | KDE  | DT                | LR   | KDE  | DT                | LR   | KDE  |
| RMSE(X) in [mm]               |  | 20.1              | 20.0 | 25.2 | 20.4              | 23.1 | 28.1 | 19.3              | 31.5 | 41.8 |
| RMSE(Y) in [mm]               |  | 25.1              | 18.9 | 20.0 | 21.6              | 21.6 | 30.0 | 27.2              | 32.8 | 34.6 |
| Localized Impacts             |  | 760/819           |      |      | 677/753           |      |      | 734/819           |      |      |
| Allocation Rate               |  | 92 %              |      |      | 89 %              |      |      | 88 %              |      |      |

**Table 7.4:** Evaluation results are shown for the TDDM based on racket-rubber combination (No. 4). RMSE in [mm] achieved by the decision tree (DT), the linear regression (LR) and kernel density estimation (KDE) is given for test data of combination (No. 4), (No. 5) and (No. 6). Additionally, the total number of correctly localized impacts out of all allocated impacts is presented in numbers and [%] (modified from [Blan 16b], doi>10.1145/2971763.2971778).

| Combination<br>Testset → TDDM |  | (No. 4) → (No.4) |      |      | (No. 5) → (No. 4) |      |      | (No. 6) → (No. 4) |      |      |
|-------------------------------|--|------------------|------|------|-------------------|------|------|-------------------|------|------|
| Method                        |  | DT               | LR   | KDE  | DT                | LR   | KDE  | DT                | LR   | KDE  |
| RMSE(X) in [mm]               |  | 22.7             | 22.1 | 25.4 | 22.5              | 23.3 | 26.0 | 18.9              | 36.7 | 42.8 |
| RMSE(Y) in [mm]               |  | 26.2             | 19.8 | 19.8 | 24.4              | 22.1 | 26.0 | 27.4              | 32.3 | 35.4 |
| Localized Impacts             |  | 753/816          |      |      | 663/753           |      |      | 714/783           |      |      |
| Allocation Rate               |  | 92 %             |      |      | 88 %              |      |      | 91 %              |      |      |

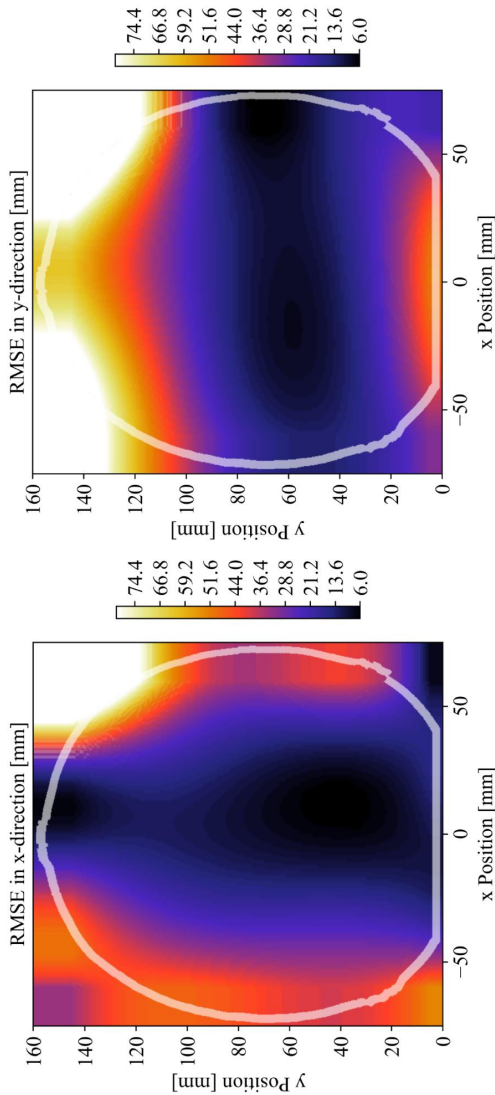


**Figure 7.10:** The left figure shows an example RMSE distribution in X-direction, whereas the right figure shows the RMSE distribution in Y-direction. Both are based on the racket-rubber combination (No. 4) and the linear regression approach. Here again, the racket dimensions are indicated as white shapes (modified from [Blan 16b], doi>10.1145/2971763.2971778).

## 7.5 Discussion

The results verified, that it was possible to estimate the ball impact position on a table tennis racket using three single piezo-electric shock sensors. They were also comparable with outcomes of contributions for ball impact localization approaches implemented for other sports [Cros 97] and [Cros 98a] and even for methods applied in table tennis analysis [Chen 12] and [Wege 09]. However, within these contributions only qualitative evaluations about impact strength analysis and impact position detection were made (→ see Chapter 9.1).

The localization errors thereby laid notably below the actual ball diameter of 40 mm and within the range of the measurement grid of 20 mm. Nevertheless, not all data of the recorded 3840 impacts could be used for the final evaluation, since position candidates determined by the decision tree approach could not be found, estimated impact positions were calculated outside the racket boundary during the linear regression or the test and training data set contained too many outliers. One explanation for this could be anomalous vibration behavior caused by hidden defects of the blade or uneven adhesive films between racket and rubber. Additionally, impacts near the racket's outer edge had the strongest leverage effect on the handle suspension which led to damping and low-pass filter effects on the vibration signal and therefore negative influences on the vibration characteristics.

Taking the TDDM and test data set based on the racket-rubber combination (No. 1), impact positions could be located best with an overall RMSE of 20.0 mm in X-direction and a RMSE of 18.9 mm in Y-direction using the linear regression method (Table 7.3). Similar results were observed with the linear regression and an overall RMSE of 22.1 mm in X-direction and a RMSE of 19.8 mm in Y-direction for the TDDM and the test data of combination (No. 4) (Table 7.4). However, it had to be considered, that the achieved results were in the range of the predefined measurement grid. A finer grid for the training data set could have produced better results, but accurate impact data were certainly more difficult to generate. Consequently, the 1 mm grid interpolation for the TDDM showed no influence on the final results.

The TDDM for a specific racket-rubber combination could surprisingly be used for the ball impact localization on other combinations, with the premise for rackets with same stiffnesses defined by the layer structure and the veneers as well as similar rubber hardnesses. This became clear by evaluation of test data obtained from combination (No. 3)

using TDDM of combination (No. 1), shown in Table 7.3 and test data from combination (No. 6) using TDDM of combination (No. 4), which can be seen in Table 7.4. Here, noticeably worse results confirmed the major difference in rubber hardness, whereas all other evaluations of combinations with softer rubbers showed better outcomes. This is illustrated by the evaluation of test data from combination (No. 2) using TDDM of combination (No. 1) and by test data from combination (No. 5) using TDDM of combination (No. 4). Investigating the different three localization approaches, no best method could be recognized. All results were in the same range for the decision tree method with lower complexity and computational effort, for the linear regression with a standard minimization algorithm and for kernel density estimation with most complex and expensive efforts. In addition, this behavior confirmed the initial experimental accuracy and that the results would not substantially benefit from a finer measurement grid.

Considering Table 7.2, the results got slightly better, if only those impacts located on sweet spots were analyzed. With test data of combination (No. 1) using TDDM of combination (No. 1) impacts were localized with an overall RMSE of 18.3 mm in X-direction and a RMSE of 19.2 mm in Y-direction, again using the linear regression method. This behavior could be explained with possible dead spots ( $\rightarrow$  see Chapter 7.1) near the outer edge, which are already mentioned in [Cros 98a]. Due to multiple mirrored reflections at the flat sides of the blade or time-based wave compressions by the outer edge of the racket, harmonic vibration components of the natural resonance frequency may occur. All these accumulated artifacts could have influenced the natural vibration propagation and deform its shape and amplitudes, thus, making it hard to detect the incoming wavefront by simple thresholding resulting in erroneous time difference computation.

This can also be verified by Figure 7.10. Lowest errors  $< 10$  mm in X-direction as well as in Y-direction hint at the location of the sweet spot in the middle of the racket, whereas very high errors  $> 50$  mm near the edges showed potential dead spots. A different indication could also be based on the fibers and the grain orientation. Errors were continuously low for the X-direction along the longitudinal orientation of the grain, as well as errors stayed low for the transverse Y-direction against the grain orientation. Vibration propagation in any diagonal direction to the fiber structures accordingly caused the highest errors for the impact localization.

On the one hand, improvements should focus on the generation of the TDDM for every racket-rubber combination. Although acceptable results could be achieved with different TDDMs and combinations comprising rackets with same stiffnesses and rubbers with similar hardnesses, it is still necessary to have an appropriate modeling of the time difference distribution. One possibility could be providing a suitable model by the manufacturers for common racket and rubber combinations. Another option would be the development of a more generalized model, which could learn and adapt the model parameters over time or which could be calibrated in a fast and simple way by the player itself. On the other hand, further advances could be accomplished by a more reliable time difference computation. Speeding up the sampling frequency results in a higher temporal resolution of the vibration data. Thus, rising or falling amplitudes could be detected more precisely leading to a better time difference estimation. Moreover, the TDDM could be modified to better reflect racket areas near dead spots. Incorporating a fourth sensor would expand the number of time differences  $\Delta t$  from three to six resulting in a distribution holding more information, which could be used to derive a more accurate impact position. All results were best using the linear regression method. Looking at Figure 7.8, a linear approach does not be the most accurate mathematical description of equal time differences. Different approaches like parabolic or polynomial functions could represent the line and dot distribution more reasonably.

However, more sensors and thus more analyzable data together with more elaborate regression functions would lead to higher computational efforts. In the end, the contrast between complexity and accuracy has to be assessed in the context of a final embedded implementation. For wearable devices, the focus lies mainly on the real-time ability of the processing pipeline, since all computations have to be performed between two consecutive strokes. Consequently, a deeper investigation of a minimum sampling rate, potential signal filtering or a suitable impact localization algorithm with least memory consumption and least computational power is required.

## 7.6 Summary

In this chapter, a system for ball impact localization on table tennis rackets was described. Vibration data of three different piezo-electric shock sensors, which were mounted on the outer edge of the racket having

maximum distances from each other were acquired to calculate three pairs of time differences  $\Delta t$ . Taking these time differences, a TDoA similar approach was applied to triangulate the initial ball impact position.

Firstly, training and test data of six different racket-rubber combinations were collected in a pre-defined study comprising 48 measuring points  $P_m(X, Y)$  spread over the whole sensitive racket surface. Secondly, this data was used to generate a time difference distribution model. According to this model, all equal time differences of an impact were used as input for subsequent localization algorithms. A simple decision tree based method, a linear regression approach including a least squares optimization and a more complex kernel density estimation with Gaussian kernels were investigated. Then, all test data sets were applied to the TDDMs based on the training data of equal and different racket-rubber combinations to evaluate the applicability of the models to different combinations.

The quantitative evaluation was based on the RMSE separately for the longitudinal racket orientation in X-direction and the transverse Y-direction. The best performance for each racket-rubber combination was achieved by the linear regression yielding an average RMSE in X-direction of 22.1 mm and a RMSE in Y-direction of 19.8 mm. These results showed, that it is possible to localize the impact position on table tennis rackets with data of three piezo-electric sensors with minimal electronic and computational efforts.

## Chapter 8

# Racket Instrumentation

This chapter presents an unobtrusive prototype for integration of sensors and electronics into a table tennis racket. Firstly, applicable regulations of the International Table Tennis Federation about approved racket materials, coverings and their combinations have to be considered. Secondly, hardware design and embedded integration as well as algorithm migration and implementation are described. Finally, a first draft of the mobile feedback application is introduced.

The embedded integration of hardware and software into a smart racket prototype has already been published in [Blan 18].

### 8.1 Racket Regulations

Whenever thinking about changing up to now officially approved sports equipment, it has to be ensured that all modifications are compliant with the latest regulations. This is especially necessary in table tennis, since each racket can individually be combined by numerous types of blades as well as by countless rubbers and their corresponding sponges.

The ITTF releases every year an official handbook [The 18a], which describes the general laws of table tennis and its regulations for international competitions. These laws are mostly adopted one-to-one by the national associations. In addition to the general laws, common regulations for the sports equipment including racket coverings [The 18c] and racket control [The 15] are stated by official technical leaflets. Moreover, a list of authorized racket coverings (LARC) is published, which contains all currently permitted rubbers [The 18b].

With these documents, the here relevant restrictions for the racket and the blade are summarized:

- The entire racket can be of any size or weight but has to be flat and rigid.
- At least 85 % of the blade by thickness has to be made out of natural wood. Also allowed is plywood, but no other composites.
- The blade can be reinforced by carbon, glass or compressed paper fibres, but these elements must not be thicker than 7.5 % or 0.35 mm of the total thickness.
- Plastic layers above 0.1 mm are not permitted, whereas thin lacquer layers for anchoring the wooden fibers and adhesive foils for rubbers are possible.
- The sides of the blade have to be covered by rubbers specified in LARC.
- All layers within the blade and the covering have to be continuous and even.
- The racket coverings must not be treated physically or chemically, but cleaning with water or alcohol dilutions are permitted.

In contrast to these detailed requirements of the racket's blade, its veneer structure and coverings, specific restrictions about the handle itself and the outer edges are not provided. Moreover, no information about attachments to the handle, for example electronic components, can be found within the rules. In combination with the wooden blade, several racket manufacturers utilize additional handle materials different from wood, like cork or plastics, which are expected to be more comfortable to wear. Furthermore, many professional club players wrap adhesive tapes around the handle to increase the grip. An edge binding tape can also be applied to the outer edges to cover open splices, thus avoiding fraying of the wooden fibres and the pimples of the rubbers.

To stay in accordance with the rules, apparently, sensors and electronic components could only be mounted within the racket's handle or at the outer edges, since any modifications of the blade are prohibited. Admittedly, a thin printed circuit board (PCB) could be integrated into the blade as a veneer, but this PCB should not entail any assembled components and ought to fill the entire blade dimension to not violate the requirements for even, flat and continuous blades.

## 8.2 Hardware Design

All sensor types used for the presented methods and therewith necessary electronic components were integrated into the *Smart Racket* prototype to implement most of the proposed algorithms. Therefore, a reasonable sensor placement for the piezo-electric vibration sensors, the IMU and, in particular, an unobtrusive integration of all electronic components is of major importance.

### 8.2.1 Sensor Placement

The placement of the piezo-electric vibration sensors were similarly distributed over the blade regarding the developed ball impact localization approach ( $\rightarrow$  see Chapter 7.2). This was necessary to maintain the validity of the developed algorithms. In addition to the proposed set of three sensors, one more vibration sensor was mounted at the outer edge of the racket. This could be done as the evaluated regression techniques can be easily migrated to operate with four sensors without any negative influences on the results and thus could achieve higher impact localization accuracies. Therefore, only the time differences  $\Delta t_{i,j}$  with  $i, j \in \{1, 2, 3, 4\}$  for the TDDM generation of all 48 measuring points  $P_m(X, Y)$  have to be reacquired. These time differences had to be measured anyway, since the racket prototype differed in blade and rubbers from the combinations used for the original data acquisition.

The IMU sensors for stroke detection and stroke type classification algorithm ( $\rightarrow$  see Chapter 5.1) as well as for the ball speed and spin estimation method ( $\rightarrow$  see Chapter 6.2) originally had different sensor positions and orientations. In the former case, a *miPod V1* sensor was mounted at the end of the racket's handle. Thereby, the X-axis was perpendicular to the blade, the Y-axis and the Z-axis pointed alongside and orthogonal to the handle, respectively. As illustrated in Figure 5.1, the coordinate origin was in the middle of the handle's end, but slightly above the grip. In the latter case, a *miPod V1* sensor was plugged into a milled cavity of a test racket. Here, the Z-axis was orthogonal and the Y-axis alongside the handle, whereas the X-axis pointed perpendicular to the blade. This sensor placement can be seen in Figure 6.4. The coordinate origin was still in the middle at the handle's end, but slightly inside the grip.

The final IMU position within the instrumented racket was also in the middle of the handle, but roughly 25 mm alongside the grip axis inside

the milled cavity on a PCB together with the remaining electronic components. Nevertheless, this difference to the original sensor positions had only little influences on the algorithms. For ball spin and speed estimation, multiple assumptions were formulated to specify the impact position  $c_{geo}$  (A4), the wrist rotation angles  $\delta_{geo}$  and the wrist rotation center  $a_{geo}$  (A6) in relation to the coordinate origin of the attached sensor unit ( $\rightarrow$  see Chapter 6.3). Only these geometric parameters had to be adapted to ensure the functionality of the evaluated algorithms. Due to the rigid and inflexible nature of the racket, the angular velocity measured by the IMU appears everywhere identical, regardless of its mounted position. Similar behavior was exhibited by the accelerometer, as long as the position and the distance to the pivot point remained constant. Admittedly, the sensor placement for stroke detection and classification differed from the sensor placement of the instrumented racket, but as only signal shapes with no absolute physical quantities were calculated for feature extraction and classification, it was more important to maintain the perpendicularity of the coordinate system and that no oblique axis transformation was necessary. Most determinative hereby was the same alignment of the main movement directions (axes spanning the blade's plain) and the main spin distinctions (axes perpendicular to the blade) of both sensor placements.

### 8.2.2 Unobtrusive Integration

An unobtrusive integration of sensors and necessary electronic components inside the racket is essential to avoid any effects on the players' performance and behavior during their exercises. Additionally, an invisible instrumentation could enhance the overall acceptance of digitalized equipment.

Therefore, four small notches were milled into the outer edge of the racket hosting all piezo-electric vibration sensors in combination with their proposed conditioning circuits. These notches had the same size of (12 x 5 x 1.7) mm as the PCBs used initially for sensor data acquisition ( $\rightarrow$  see Chapter 7.2) and their positions spanned a square with maximum side lengths around the blade. In addition, a small channel containing six thin cables, required for power supply, ground and four times data signals, was milled into the blade's outer edge. The cable routing started from the farthestmost sensor on the right side of the forehand, back to the first sensor on the left side and finally through a tiny hole at the handle's throat into the hollowed interior of the grip. Sensors and cables were

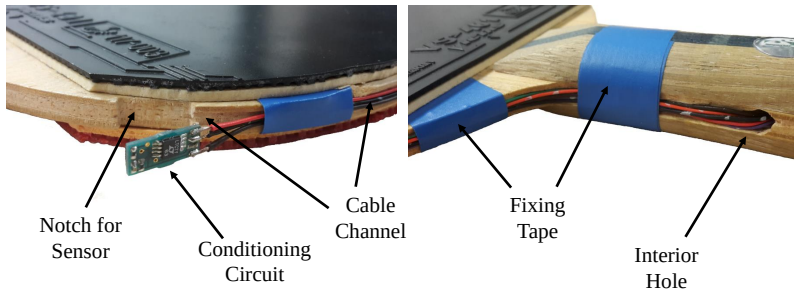


**Figure 8.1:** Integration of the piezo-electric vibration sensors. The housing notches for each sensor are filed into the outer edge of the blade. They contain pieces of adhesive tapes to attach the sensor chips and the conditioning circuits. All necessary cables are routed inside a small channel around the blade and guided through a tiny hole into the interior of the handle. For better visibility, the edge binding tape covering the cables and the electronics was removed.

fixed with normal and double-sided adhesive tapes. Furthermore, the overall outer edge was overlaid by an edge binding type to cover all electronics and to protect the blade, as it is commonly made by most players. Integration and placement of the piezo-electric vibration sensors and the cable routing of the prototype racket can be seen in Figure 8.1.

The IMU sensor and the remaining electronics were integrated into the grip of the handle. A PCB with the dimensions of (29 x 24 x 6.5) mm was plugged into a milled cavity, which had a slightly larger outline. Besides the *BMX055* inertial measurement unit from *Bosch Sensortec*<sup>1</sup>, which was also used for the *miPod V2* sensor, the board contained a low-power 32-bit Cortex-M4 *EFM32GG11* microcontroller from *Silicon Labs*<sup>2</sup> with 2048 kByte FLASH, 512 kByte RAM, 70 MHz CPU speed and a FPU. Using this features, the MCU was capable to execute the developed algorithms (→ see Chapter 8.3). As communication device, the *SPBT3.0DP2* Bluetooth® Classic 3.0 radio module from *ST Microelectronics*<sup>3</sup> was applied. In contrast to Bluetooth Low Energy® modules, the Classic module could also be used to transmit high amounts of raw data for debugging purposes and it provides a better interconnectivity. Additionally, the mobile device might not always be close enough to the player,

<sup>1</sup><https://www.bosch-sensortec.com/> – last access 07/2018

<sup>2</sup><https://www.silabs.com/> – last access 07/2018

<sup>3</sup><https://www.st.com/> – last access 07/2018

since its wearing could impede the athlete during playing. Moreover, the power management controlling supply voltage and battery charging was realized by the *LTC4081* chip from *Analog Devices*<sup>4</sup>. Apart from these components, some LEDs for status indication, soldering pads for the connection of the piezo-electric data and supply signals, as well as an USB for external power and battery charging, a debugging and programming connector and an on-off switch were integrated. Energy was provided by small lithium-polymer battery with a capacity of 155 mAh. The runtime should last for a training session or a complete match.

To ensure a tight fit of the PCB and the battery inside the interior of the handle, two supplementary foam sponges, one on the top and one on the bottom of the cavity were inserted. They prohibited any displacements and wobbling of the board and therefore of the assembled IMU sensor. Furthermore, a small screwed bar at the handle's end prevented the board from accidentally dropping out the cavity. Figure 8.2 illustrates the assembled PCB, the milled cavity and the entire instrumented racket prototype. The actual *Smart Racket* is differently composed as

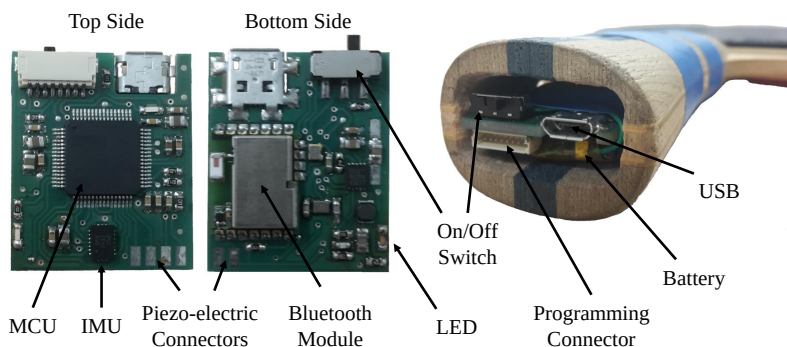


**Figure 8.2:** Main electronics and the handle integration. The top side contains the microcontroller, the IMU sensor, the programming connector as well as the data connection for the piezo-electric sensors. On the bottom side, the USB connector, the on-off switch, all components of the power management and supply connectors for the conditioning circuits of the sensors as well as the indication LED are assembled. The entire board including its battery is integrated into the racket's handle. For this figure, the screwed bar and the foam sponges were omitted.

<sup>4</sup><http://www.analog.com/> – last access 07/2018

presented in the last chapters and consisted of an *Imperial Allround ST*<sup>5</sup> blade combined with a *Victas VS401*<sup>6</sup> rubber on the forehand side and a *Sriver L7* rubber on the backhand side with a thickness of 1.9 mm and 2.1 mm, respectively.

## 8.3 Embedded Implementation

All developed algorithms were completely processed by the microcontroller. The implementation contained the combination of impact detection and validation, stroke classification, impact localization and ball speed and spin estimation. Before any calculations could be performed, data acquisition and preparation steps had to be conducted. All computations were embedded and implemented into tasks and threads similar to the discussed aperiodic scheduling approach within the implemented RTOS (→ see Chapter 3.2). The employed *FreeRTOS* [Barr 12] derivate maintained all real-time constraints for the necessary processing tasks imposed by the intended table tennis application. Thereby, a maximum allowable response time of ~ 0.8 seconds for two consecutive strokes of one player was assumed, for which the interarrival time for stroking frequencies of elite table tennis athletes was well represented [Xie 02]. An overview about the processing pipeline and its working threads are shown in Figure 8.3.

The data acquisition and preparation step utilized a double-buffered architecture to ensure simultaneous data storage within the RAM using the first buffer and data processing by the CPU using the second buffer. Both buffers were repeatedly interchanged until an impact was detected. All data were directly acquired by the DMA system without any interaction of the CPU itself, thus saving capabilities for actual computations. Since all algorithms required different sampling settings and sensor configurations, each two buffers were allocated to provide input data and each one buffer was created to store the feature vectors if needed. According to Equation 3.3 (→ see Chapter 3.1), the necessary RAM requirements can be determined. The impact localization part needed a data buffer size  $DATA_{size}$  of 24.2 kByte stored in the RAM. In addition, the stroke detection and validation step allocated 10.8 kByte for the input sensor data  $DATA_{size}$  and 0.7 kByte to store the feature vec-

<sup>5</sup><https://shopware.imperial-sport.de/> – last access on 02/2018

<sup>6</sup><https://www.victas.com/> – last access on 02/2018

<sup>7</sup><https://www.butterfly.tt/> – last access on 08/2018

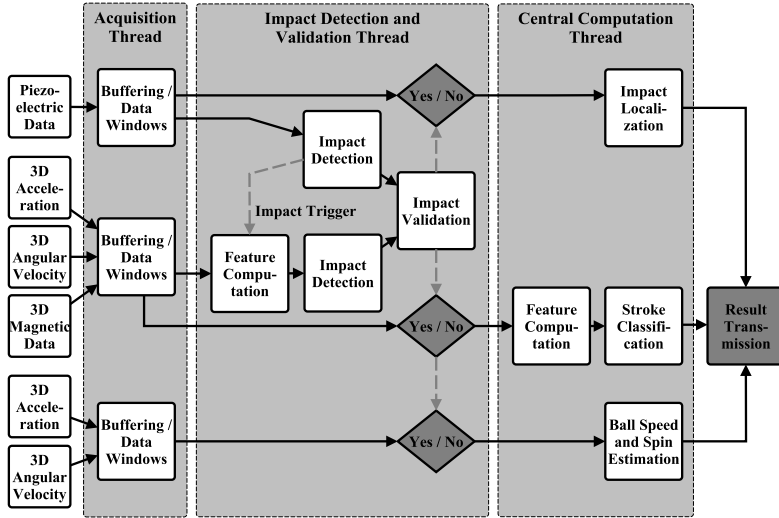


**Figure 8.3:** Processing pipeline of the algorithms implemented on the micro-controller. Firstly, a data acquisition and preparation thread samples data from the piezo-electric vibration sensors and the IMU sensor. Using these data, specific window frames for subsequent processing are generated. Secondly, an impact detection and validation thread detects impacts from vibration data as well as from inertial sensor data. Hereby, impact detection using inertial and geomagnetic data is triggered by an impact event recognized by the vibration sensors. If both events pass the impact validation, the central computation thread is started. It localizes the initial ball impact position, classifies the stroke type and estimates ball spin and speed. Finally, all results are transmitted to a mobile device.

for  $FEATURE_{size}$  within the RAM memory. Thereby, the applied RF algorithm was trained with 15 decision trees  $\#DT$  and the light feature selection resulting in 19 single channels features  $Q_{SC}$ . Moreover, the stroke classification was based on a quad RBF SVM using the entire feature set  $Q$  of 90 features and thus had to implement a storage buffer for the feature vector  $FEATURE_{size}$  of 3.0 kByte in RAM. Furthermore, the algorithms for ball speed and spin estimation necessitated 72.0 kByte RAM size to hold the obtained inertial data  $DATA_{size}$ . In addition to the remaining RAM demands for program execution, the MCU had to provide at least 110.7 kByte RAM for input data buffers and feature vectors.

**Table 8.1:** Minimum RAM requirements in [kByte] for impact localization (IL), stroke detection (SD), stroke classification (SC) and ball speed and spin estimation (SSE). They can be determined depending on the number of sensors  $n_s$ , the number of sensor channels  $m_s$ , the window lengths and window overlaps  $w_L$  and  $w_O$ , the sampling intervals  $T_A$ , the used data representation  $BpV$ , as well as on the number of single channels features  $Q_{SC}$  and on the number of multi channel features  $Q_{MC}$ .

| Algorithm                       | $DATA_{size}$ in [kByte] | $FEATURE_{size}$ in [kByte] | Sensors $n_s$ | Sensor Channels $m_s$ | Window Length $w_L$ in [ms] | Window Overlap $w_O$ in [ms] | Sampling Interval $T_A$ in [kByte] | Features $Q_{SC}$ | Features $Q_{MC}$ | Bits per Value $BpV$ |
|---------------------------------|--------------------------|-----------------------------|---------------|-----------------------|-----------------------------|------------------------------|------------------------------------|-------------------|-------------------|----------------------|
| IL                              | 24.2                     | —                           | 4             | 1                     | 3.3                         | 1.65                         | 0.00436                            | —                 | —                 | 16                   |
| SD                              | 10.8<br>—                | —<br>0.7                    | 3             | 3                     | 750                         | 250                          | 5                                  | —<br>19           | —                 | 16<br>32             |
| SC                              | —                        | 3.0                         | 3             | 3                     | 750                         | 250                          | 5                                  | 81                | 9                 | 32                   |
| SSE                             | 72.0                     | —                           | 2             | 3                     | 1500                        | 500                          | 1                                  | —                 | —                 | 16                   |
| Total Size in RAM = 110.7 kByte |                          |                             |               |                       |                             |                              |                                    |                   |                   |                      |

A detailed overview of the RAM expenses and the algorithm requirements can be seen in Table 8.1. Apart from requirements concerning the RAM memory, the microcontroller needed also sufficient FLASH space to store the appropriate TDDM, the decision trees of the developed RF, as well as the support vectors and weights of all single RBF SVMs. Based on the blade's dimension of the instrumented racket of (150 x 160) mm and on the interpolated grid of 1 mm, the TDDM had a

size  $TDDM_{size}$  of 48.0 kByte for each time difference mapping, if a 16-bit data representation was used. As 4 vibration sensors were implemented,  $\frac{n_s!}{2 \cdot (n_s - 2)!}$ , which corresponds to 6, time difference mappings had to be stored within the FLASH memory. Moreover, forehand and backhand of the racket had different assembled rubbers, which required a separate TDDM for each side resulting in an overall 576.0 kByte FLASH memory space. Furthermore, additional free space had to be available containing the parameters of the RF and RBF SVM classifiers. According to Equation 3.1 and Equation 3.2 ( $\rightarrow$  see Chapter 3.1), the Random Forest needed 245.7 kByte of memory space  $RF_{size}$  incorporating a reduced tree depth  $d_{tree}$  of 12 and 15 decision trees  $\#DT$ .

In contrast, the RBF SVM required 294.5 kByte memory  $SVM_{size}$  in FLASH, assuming 4 single SVMs, with each 256, 89, 264 and 200 support vectors and weights. For both classification systems, 32-bit floating point values were presumed. In addition to the remaining program code, the microcontroller had to reserve a minimum of 1116.2 kByte of constants and parameters within the FLASH memory for all integrated algorithms. The overview of FLASH requirements can be seen in Table 8.2.

Following data acquisition and preparation, a robust impact detection procedure embed into a separate thread was necessary. This step ensured that only valid table tennis strokes with ball impact characteristics were detected. Firstly, possible impact candidates were recognized by thresholding the input vibration signals as it is proposed for ball impact localization ( $\rightarrow$  see Chapter 7.3). Secondly, this event in turn triggered the impact detection based on acceleration data and its maximum root signal energy, which is presented in stroke detection and classification for the analysis of gameplay rallies ( $\rightarrow$  see Chapter 5.2). Thirdly, if both events passed the impact validation incorporating the above mentioned RF classification system, a valid impact was assumed and further processing steps could be initiated.

After impact detection, the main computation thread was started. This thread localized the initial ball impact position using either the DT or the LR approach ( $\rightarrow$  see Chapter 7.3), classified the stroke type applying the beforehand described RBF SVM ( $\rightarrow$  see Chapter 5.2) and estimated ball speed and spin shortly after the rebound ( $\rightarrow$  see Chapter 6.3). Thereby, the  $\nu$ -SVM version with the parameters  $\nu$  and  $\gamma$  (both 0.01) classified the performed stroke into the 4 predefined categories FFORE, FBACK, BFORE and BBACK. According to the introduced assumptions and simplifications, ball speed and spin estimation needed additional

**Table 8.2:** Minimum FLASH requirements in [kByte] for impact localization (IL), stroke detection (SD), stroke classification (SC) and ball speed and spin estimation (SSE). The computation of the expenses depends on different parameters for each method. However, for the ball speed and spin estimation no notable FLASH requirements have to be considered.

| Algorithm                          | $TDDM_{size}$ in [kByte]<br>Blade (160 x 150) mm Grid 1 mm<br>both Racket Sides | $RF_{size}$ in [kByte]<br>Number of # $DT$ with 15<br>Tree Depth $d_{tree}$ with 12<br>$Q_{SC}$ with 19, $Q_{MC}$ with 0 | $SVM_{size}$ in [kByte]<br>Number of # $SVM$ of 4<br>$S_V$ with 256, 89, 264 and 200<br>$Q_{SC}$ with 81, $Q_{MC}$ with 9 | Bits per Value $BpV$ |
|------------------------------------|---------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------|----------------------|
| IL                                 | 576.0                                                                           | —                                                                                                                        | —                                                                                                                         | 16                   |
| SD                                 | —                                                                               | 245.7                                                                                                                    | —                                                                                                                         | 32                   |
| SC                                 | —                                                                               | —                                                                                                                        | 294.5                                                                                                                     | 32                   |
| SSE                                | —                                                                               | —                                                                                                                        | —                                                                                                                         | 32                   |
| Total Size in FLASH = 1116.2 kByte |                                                                                 |                                                                                                                          |                                                                                                                           |                      |

geometric information about the IMU origin  $c_{geo}$ , the wrist rotation center  $a_{geo}$  and  $\delta_{geo}$  of each player, as well as the rubber properties  $e_R$ ,  $\mu_R$  and  $k_{ip}$  in advance. Combined with the TDDM description of the used racket-rubber combination and the effective outline of the racket, all necessary parameters could be stored within the feedback application and therefore shared with the microcontroller. Finally, all computed results were instantly transmitted via Bluetooth® to the mobile device application for a recording overview, storage and feedback purposes.

Summarized, the chosen EFM32GG11 microcontroller with 2048 kByte FLASH and 512 kByte RAM was therefore sufficient to store all buffers, features and model parameters. Moreover, it had enough memory reserve for additional program code and future implementations. Admittedly, no estimations for the minimum required CPU speed were possible, as all single WCETs for processing tasks and threads were not directly computable. However, experience in firmware development as well as empirically assessed and measured timing values indicated that a CPU speed of 70 MHz in combination with a FPU were sufficient to execute all calculations within the postulated RT time period of 0.8 seconds. This was finally confirmed by the implementation of the RTOS, including the aperiodic scheduling approach and the thread management presented in Figure 8.3.

## 8.4 Feedback Application

The computed results were conditioned for the feedback application running on a Android® mobile device. The following draft of the *Smart Table Tennis Racket* application visualized any input data and presented them to the player. Thereby, initial parameters necessary for processing had to be entered and the Bluetooth® connection to the smart racket had to be established. Currently, two modes were implemented: a free training mode and an exercise mode. Within the free training mode, any data were collected and instantly visualized for immediate feedback.

Thereby, the impact position was graphically displayed by a ball image on the racket surface, the matching stroke category of the four types FFORE, FBACK, BFORE and BBACK was highlighted and ball speed in [m/s] (blue) and spin in [rad/s] (red) were jointly shown in a bar chart. Positive spin values indicated forward spin, whereas negative values represented backward spin. A free training example can be seen in Figure 8.4.

In contrast, the exercise mode allowed the training of specific events. An event could be the practice of predefined stroke types, hitting the ball within a user-defined impact area or reaching preset speed and spin values. As feedback, visual and audio sources were implemented, if the defined settings were satisfied or not. After data recording, it was possible to generate a statistical overview and evaluation about the own performance, which can be seen in Figure 8.5 and in Figure 8.6. The evaluation view presented the rally overview and displayed the number of

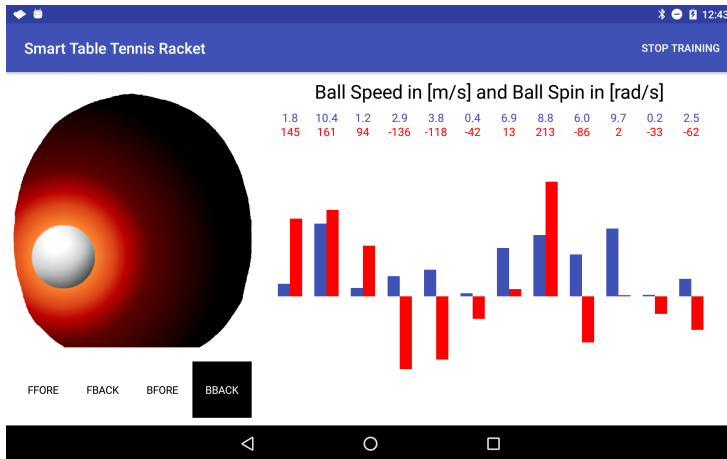


**Figure 8.4:** Recording view of the free training mode. On the left side, the actual impact position and the performed stroke type are highlighted. On the right side, ball speed and spin of each stroke are jointly presented considering the spin direction.

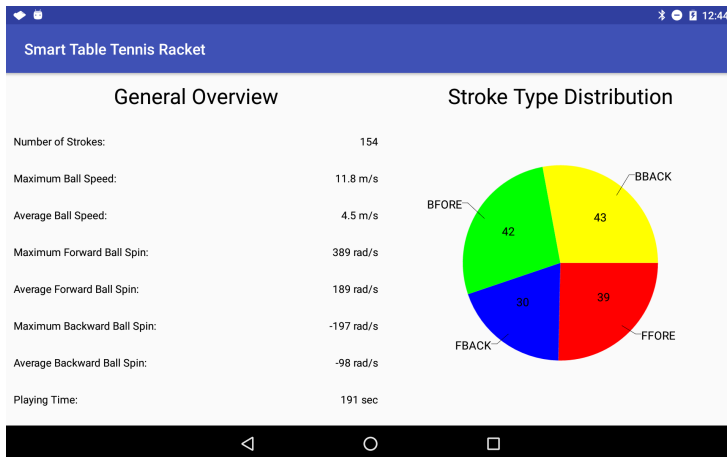


**Figure 8.5:** Example evaluation of the rally overview. The figure gives a general overview about the rally and breaks down all strokes into their categories and displays the distribution within pie chart.

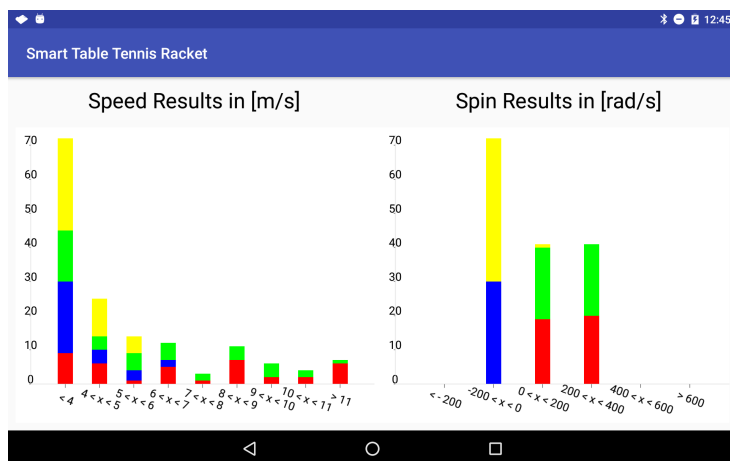


**Figure 8.6:** Example evaluation of the ball spin and speed distribution. Ball speed and spin are presented in stacked bar charts. They indicate the number of strokes considering the appropriate stroke type and the distribution of achieved speed and spin values, respectively. Here, same colors as for the rally overview were used.

stroke types in colored pie chart. Additionally, the distribution of the ball impact locations were commonly shown on a racket surface similar to the recording view. Furthermore, ball speed and spin were provided using stacked bar charts. Thereby, same stroke types were indicated by same colors defined for the pie chart. Speed and spin values were split into blocks ranging from low to high values for better interpretation and evaluation of the performed strokes.

## **Chapter 9**

# **Summary, Discussion and Conclusion**

This chapter summarizes the results and draws implications from the contributions. Firstly, the methods for table tennis analysis and the racket instrumentation approach are discussed. Secondly, future work and an outlook about further investigations are given. Finally, a conclusion completes this thesis.

### **9.1 Summary and Discussion of Contributions**

The following chapter comments on the implementation of the listed contributions (→ see Chapter 1.4). On the one hand, results and outcomes of the presented methods are summarized, discussed and compared with existing approaches in literature. On the other hand, the described techniques are assessed considering to what extent they contribute to a reasonable feedback system usable by professional and amateur table tennis players.

#### **9.1.1 Table Tennis Analysis**

##### **Contribution (I) – Stroke Detection and Classification**

The first part of the table tennis analysis provided a detection and classification method of basic table tennis strokes from players with different skill levels and playing abilities. Motion data of a racket-mounted inertial sensor were acquired during real gameplays and exercise scenarios.

The obtained kinematic information were used to detect hitting events and to classify stroke types incorporating different machine learning algorithms. The results for stroke detection of  $> 93\%$  and stroke type classification of  $> 95\%$  reached over the entire training set for the gameplay and exercise scenario data proved that an analysis of stroke movements in table tennis using only a single inertial sensor is possible.

This is also confirmed by existing literature listed in the state-of-the-art, which was based on data from Wearables. Similar accuracies were achieved using wearable sensors in other racket sports like tennis or badminton. In [Iji10] and [Ahma09], inertial data were investigated to differentiate between professional and amateur tennis players. In [Conn11b], tennis strokes were detected with an accuracy of  $85\%$  and classified into categories with a classification rate of  $90\%$  using a sensor attached to the player's forearm. An approach based on Support Vector Machines containing multiple body-worn IMUs was presented in [OCo10] to classify several types of tennis strokes resulting in classification rates of  $97\%$ . Moreover, [Rama14] investigated Support Vector Machines to identify badminton strokes yielding an overall accuracy of  $93\%$ . Further machine learning methods were contributed by [Wang16]. Thereby, overall results of  $98\%$  were achieved for the detection of different badminton strokes with four body-worn inertial sensors. Similar outcomes of  $95\%$  accuracy for stroke classification in badminton were presented by [Lin17] using a single racket-mounted sensor system. Compared to investigations of other racket sports, in table tennis itself, most approaches were limited to a qualitative evaluation of the stroke movements and presented only comparisons of different skill levels of athletes or simply visualized varied motion patterns, regardless whether wearable sensors or video data sources were used. Examples therefor were described in [Boot90], [Guo10] and [Boye13]. However, in [Zhan06], table tennis strokes were automatically detected using acoustic sensors resulting in an overall accuracy with a precision of  $91\%$  and a recall of  $73\%$ . The video-based stroke classification approach in table tennis purposed by [Oldh15] yield an overall accuracy of  $96\%$ . Thereby, multiple stroke types and events were detected and classified.

Regarding the overview, multiple machine learning approaches, especially Support Vector Machines, reflect swing motions for racket sports with reasonable performances [Cham15]. This extends both for predefined exercises scenarios as well as for the free gameplay rallies. On that account, the herewith developed machine learning algorithms allow the

detection and classification of stroke events for training and competition. So far, processing of inertial data of Wearables has yet been neglected in the digitalization of table tennis.

Nevertheless, there are several issues for an unrestrained analysis, which could negatively influence the practical usage within the Smart Racket application. On the one hand, the handling of extraordinary impacts or impacts, which are not part of the actual gameplay, are still difficult to detect. On the other hand, in a real gameplay situation yet uncategorized stroke types exist. Admittedly, the stroke validation step should filter out the largest amount of unwanted ball contacts, but ball-racket impacts similar to valid motion patterns could occur and therefore are wrongly matched within the subsequent classification step. For example, contacts between the racket's outer edge with the ball or table surface could lead to correctly detected ball impacts, but also to unknown motion patterns. During a match, the number of these occurrences should not be underestimated, as they happen relatively often.

Additionally, strokes at which the ball was touched by the fingers or the wrist of the hand holding the racket also count as permitted strokes. Currently, these strokes would completely be ignored by the algorithms, since no impact vibrations were recognizable and therefore discarded by the stroke validation step. Further considerations for data processing and exclusion procedures would be necessary to overcome these drawbacks within the analysis of gameplay situations. It is reasonable to add a presorting approach, which differentiates ball impacts into the following classes: one class for regular ball impacts, which can subsequently be classified into stroke type categories, one class for irregular ball impacts, which are still valid for scoring points and finally one class with ball impacts, which are certainly not part of the actual gameplay.

As indicated, some stroke type categories are yet unconsidered in this contribution. Currently, strikes, chops and even all serves cannot be classified with the algorithms developed in this thesis and neither within existing approaches in literature. If such stroke types are performed within a match, they will be sorted into one of the predefined classes for which the classification method assumed them to fit best. Furthermore, only forward and backward spin strokes are investigated so far, but side spin strokes are also often used to vary the gameplay.

Hence, the extension of classifiable stroke categories to yet unconsidered types as well as the challenge of differentiation between valid and non-valid strokes should move into the focus of future research.

## **Contribution (II) – Ball Speed and Spin Estimation**

The second part of the contributed table tennis analysis introduced the ball speed and spin estimation shortly after impact. Kinematic motion data of a racket-mounted inertial sensor were collected to calculate the racket speed and the racket rotation during impact events. Using several assumptions and simplifications as well as a specialized rebound model, the racket motion parameters were transformed into ball characteristic shortly after impact. Thereby, ball speed and spin were estimated with overall accuracies of 79 % and 74 % as well as mean errors of  $-0.09$  m/s and  $-22.4$  rad/s, respectively. Additionally, an exiting video labeling approach was extended to extract the ball speed and spin from a single high-speed camera with relative errors of 9 % for ball spin and 3 % for ball speed. The implemented approach indicated, that ball speed and spin in table tennis can be estimated with a single racket-mounted IMU considering necessary assumptions and simplifications.

Referring to existing approaches for ball speed and spin estimation in table tennis, exclusively video-based analyses were performed. In [Tama 04], [Tama 12] and [Liu 11b], methods were conducted to describe the properties of the flying ball including speed and spin estimates using qualitative evaluations. Spin estimations of synthetic and real game data were conducted in [Szep 11]. Thereby, the authors achieved relative errors of  $\pm 12$  % for spin computation in synthetic images and  $\pm 20$  % in real video sequences. These results are comparable with the extended video labeling approach of the presented contribution. In general, all video-based methods show better outcomes than the developed methodology on the basis of inertial data. In [Liu 11a], an overall mean error for ball spin estimation of  $0.5$  rad/s was achieved, but absolute error depended on the performed stroke type and the initial ball rotation. Those dependencies were also apparent within the results of this contribution and were recognizable in investigations by [Zhan 14] and [Zhan 15]. Within their work, best accuracies yielding and average error for ball spin values of  $0.4$  deg/s were only reached for low ball rotations and low linear ball speeds. If both parameters were increased, the estimation errors increased as well. However, both publications analyzed only ball rotations  $< 100$  rad/s, which were three to four times lower than the ball rotations of strong forward topspins conducted within the studies of the here proposed methodology.

It is obvious that video-based techniques can reach better results for ball speed and spin estimation, since they use data of images and video

sequences containing directly correlated information of the ball rotation. The racket-mounted inertial sensor only captured kinematic data of the attached racket. Thus, the provided information about the table tennis ball were not sufficient for an overall description of its properties. As a consequence, the proposed assumptions and simplifications were needed to connect ball and racket properties, which indeed induced weighty errors. However, most often it is not important to know the exact value for the speed and spin of the outbound ball, it is more useful for the players and coaches to assess the basic spin and speed tendencies during training and exercise. On that account, this contribution aimed not at most accurate computations of ball speed and spin values, but at reasonable estimations of either strong or weak ball rotations for the performed topspin or push stroke resulting in fast and slow flying balls as well as in forward and backward spinning balls. The overall purpose was to provide a feedback system with as less sensors as possible allowing only sensory components inside the racket.

Furthermore, the implemented algorithms can provide additional but not yet mentioned performance indicators other than ball speed and spin estimations. Implicitly, the racket acceleration and wrist rotation during impact have to be calculated, which also contain important information for elite athletes. Thereby, the optimal point in time at which the ball should be hit is when the racket acceleration reaches its maximum during the stroke motion. Often players suffer from hitting the ball too early or too late resulting in suboptimal energy and impulse transfer. Knowing these offsets between the detected ball impact and the effective acceleration maximum, this timing information can be very useful to improve one's movement process.

### **Contribution (III) – Ball Impact Localization**

The third part of the table tennis analysis contribution dealt with the ball impact localization on the racket surface. Vibration data of multiple piezo-electric sensors mounted on the blade's outer edge were acquired to calculate pairs of time differences of the incoming wavefront. Taking these time differences, various approaches were investigated to triangulate the initial ball impact position by means of racket-rubber specific generated TDDMs. Best performance was achieved yielding an average RMSE of 22.1 mm alongside and an average RMSE of 19.8 mm orthogonal to the handle direction. These results confirmed that a ball impact

localization can be realized using only a small number of piezo-electric sensors with minimal electronic and computational efforts.

Looking at existing literature, the presented localization approach performed with similar or better results compared to other methods applied in table tennis. In [Chen 12], ball impact positions were detected using an external laser array. The authors achieved accuracies of around 21 mm independently from the impact location. Moreover, a qualitative impact localization approach was implemented in [Yama 18]. They integrated a strain sensor array underneath the rubber. With this system, different impact positions could be located and multiple stroke types were recognized. Similarly, a large-scale piezo-electric sensor array between blade and rubber was applied in [Wege 09]. Therewith, impact localization accuracies of around 20 mm were achieved, which represented the implemented measurement grid of the array elements. All localization methods including the thesis' contribution have in common that the resulting accuracies reflect the initial measurement grid.

Compared to both approaches in [Yama 18] and [Wege 09] applying wearable sensors, the presented technique has the advantage of not interfering with the racket characteristics due to embedded materials. The impact position was localized using small sensors attached at the blade's outer edge without any influence on the desired racket characteristics. This approach is solely conform with official regulations of the ITTF and therefore the only reasonable technique which can practically be used providing feedback for exercise and competition.

However, the developed approach depended on other requirements comprising the wooden material and the layer structure of the racket as well as the assembled types of rubbers. As multiple time differences for each sensor pair based on the vibration propagation within the racket were measured, every individual assembled racket has to be calibrated once to receive an appropriate TDDM. Usually, players utilize different rubbers on each side of the racket to expand their gameplay, which required a generation of two TDDMs for both forehand and backhand sides. Furthermore, these calibration procedures have to be redone, if rubbers are replaced or new blades are tested. In practice, these calibration steps performed by the players themselves are often cumbersome and lead to erroneous model parameters. Therefore, alternatives have to be found for an unobstructed way of TDDM generation and thus a better acceptance by the athletes. On the one hand, a database for most common racket-rubber combinations provided by the manufacturers could

be established to simply select the current combination of the own racket assembly within the mobile application. On the other hand, a solution comprising self-learning algorithms could be implemented to automatically train the necessary model with preferably only a low number of fixed measuring points.

### 9.1.2 Racket Instrumentation

#### Contribution (IV) – Racket Prototype

The fourth part of this thesis combined all developed algorithms into a single embedded system. In comparison to [Boye 13], which used measurement electronics originally intended for music applications, an individual hardware platform for table tennis was realized. The data acquisition and all processing steps needed for the algorithms were executed on the developed hardware platform including an energy-saving but powerful microcontroller. Important requirements of the electronic components were selected using the described fundamentals and with experience resulting from the presented *miPod* development. Besides pure hardware aspects, considerations about official regulations for permitted racket materials and modifications were made prior to the actual racket instrumentation. To sum it up, an unobtrusively instrumented Smart Racket prototype containing the ability to execute all introduced algorithms and computations in combination with a feedback application running on a mobile device was developed.

However, a couple of modifications were necessary to integrate all components into the racket. Since cables, tapes and electronics inside the handle are still visible, it is recognizable that something is different compared to a normal table tennis racket. This knowledge could lead to involuntary behavior within motion patterns of players or could influence the users' gameplay. A deeper instrumentation would be preferable regarding the acceptance especially of elite athletes. Often, players claim to realize any additional gram or different weight distributions caused by the milled cavity, thus they are no longer able to play as usual.

One compromise could be to omit the vibration sensors at the outer edge of the racket. Therewith, estimations of the impact position would no longer be possible, but also all filed notches, cables and the hole within the handle's throat could be omitted. In addition, a significant smaller PCB could be realized and the embedded integration of all electronics into the handle could be done with minimum efforts or even

more simple if only a thin sensor can be mounted outside at the handle's end. Furthermore, the calibration procedure of the assembled racket-rubber combination would become dispensable and such tiny sensors could be easily transferred to another racket model if desired.

In general, future racket instrumentation should balance the analysis aspects that can be measured and evaluated with the acceptance of the instrumentation needed. Thereby, any modifications of the racket must stay in accordance to the regulations.

## 9.2 Outlook

As all presented contributions provide a novel table tennis feedback system, considerations in many direction are possible. In the future, the previously mentioned more reliable stroke detection and validation step could be implemented. Linked considerations of inertial data, vibration data and sound data acquired from small MEMS microphones similarly to [Zhan 06] could be helpful to validate only those impacts containing hitting characteristics. Furthermore, players would benefit from the introduction of yet uncategorized stroke types using an extension of the evaluated machine learning algorithms. Besides including a separate serve analysis, the feedback application could additionally support the recognition of side spin strokes.

In addition to ball speed and spin estimation, all necessary computations of stroke properties could be provided to the user as additional feedback options. The maximum racket acceleration and the racket rotation during impact are also important performance indicators and can give insights into the timing during the entire stroke motion ranging from the countermovement to the ball impact event. Obviously, improvements of the applied assumptions and simplifications would lead to better reflections of the estimated ball properties. As these parameters are unknown to the racket-mounted sensor, additional information regarding the current stroke type and motion patterns as well as one's stroking history and therefore predictions of ball speed and spin in combination with their tendencies over time could lead to overall better estimations. Better results could also be achieved by more detailed investigations of the resting point and starting event for acceleration integration as well as the racket motion during the countermovement.

Regarding the impact localization, the detection of an incoming wavefront is crucial. To determine this point in time more precisely, more

complex methods instead of simple thresholding would be recommendable. Here, a frequency analysis using a *Fast Fourier Transformation* (FFT) or template matching approaches for vibration detection would enable a more accurate determination of the necessary time differences and finally of the generated TDDM.

Certainly, improvements within the measurement equipment and the sensor components would lead to advantages for the developed algorithms, which are based on their provided data. Due to recent advances in sensor technologies, special high-impact IMUs with acceleration ranges of  $\pm 32$  g and gyroscope rates of  $\pm 4000$  deg/s are available<sup>1</sup>, yielding twice the resolution as currently integrated inertial sensors. Using these devices, a more detailed insight into impact processes would be possible and potential clipping of signals would no longer cause problems for processing. On that account, a completely reworked electronics is currently under development. Thereby, just one single PCB will realized, which contains all necessary components. It will be as small as the maximum permitted thickness of a veneer inside the layer structure of the wooden blade. Hence, it will be integrated into the racket as unobtrusively as possible. This PCB will contain an IMU in the middle of the blade, which will make the wrist rotation matrix  $\mathcal{W}$  obsolete. Therewith, the IMU origin will be assumed to be located nearly at the same position where the impact occurs, thus no rotation components will be induced and wrist movements will no longer have to be considered. Additionally, the conditioning circuits for the piezo-electric shock sensors will be extended by an own microcontroller to enhance the individual sampling frequency of each sensor element. Only the calculated time differences will have to be transmitted to the main MCU, which will be still located inside the handle. Furthermore, the power management, the Bluetooth® transceiver and a wireless charging system will be embedded into the handle, so that cables and switches will no longer be needed. Since all components will be integrated within the layer structure of the blade and the handle, this instrumented racket will not be distinguishable from any standard table tennis racket.

An important aspect is the preparation and visualization of results within the feedback application on the mobile device, which should be upgraded in future. Here, all data have to be intuitively conditioned in

---

<sup>1</sup><http://www.tdk.com/> — last access 08/2018

such a way that coaches and players can easily draw conclusions from performance and training data.

Coming from a more general point of view, data of Smart Rackets from other players, information of an instrumented table or data from already digitalized scoring displays could span a whole sensory network. If both players during a match would utilize an instrumented racket, more statistics within the game could be collected leading to an entire database of one's own gameplay and useful information concerning different playing styles of other players. This could reveal reasons for winning or losing a game using statistics including performed strokes, spins and other characteristics. Benefits would also be generated from the information about the ball impact location on the table and ball-net contacts, which is similarly investigated with piezo-electric sensors in [Hahn 17]. Combined with the Smart Racket, a variety of additional training scenarios and exercises would be possible.

### 9.3 Conclusion

Overall, the presented contributions cover a wide field in table tennis analysis using Wearables, which was by now unaddressed. Despite single investigations using body-worn sensors by other authors attempting to digitize this kind of sports, until now none of their concepts have been developed to such an extent that a complete feedback application would be possible. Certainly, numerous playing materials and racket-rubber combinations as well as lots of different technical aspects and motion patterns during the gameplay exist in table tennis, which makes a comprehensive and quantitative evaluation highly complex compared to other sports with simple periodic motion sequences. The integration of stroke detection and classification, ball speed and spin estimation as well as ball impact localization algorithms into a single Smart Racket system represents a novelty in the context of table tennis analysis.

Indeed, this is the first step into digitalization of table tennis, but there is still room for improvements. More precisely, the presented analysis steps are realized from a technical point of view, which have to be transformed into evaluations concerning the most important aspects for coaches and athletes. In addition, modifications and adjustments of outcomes from the implemented algorithms would provide a more detailed assessment. Therefore, an in-field survey should be performed

to obtain worthwhile input for an optimal retrieval of meaningful performance and statistical indicators.

The advantage of applying a sensor system into a table tennis racket resulting in a wearable training tool is to extend its usage to every playing session. Therefore, no predefined laboratory environment nor expensive camera equipment is needed. Additionally, amateur club players or beginners benefit from such an analysis tool if no qualified coach is currently available and professional support cannot be guaranteed. Utilizing the Smart Racket as assessment system, objective and quantitative determination of performance indicators could help to support every player, coach and even manufacturers of rubber and blade materials to achieve more opportunities for training progress and material selection.



# List of Figures

|     |                                                                                                          |    |
|-----|----------------------------------------------------------------------------------------------------------|----|
| 1.1 | Example forehand topspin and backhand push stroke from [McAf09]. . . . .                                 | 15 |
| 1.2 | Example table tennis setup of the World Championships 2018. . . . .                                      | 17 |
| 1.3 | Examples of Wearables used for training support in tennis and badminton. . . . .                         | 37 |
| 1.4 | Smart Racket system including the instrumented table tennis racket and the feedback application. . . . . | 38 |
| 2.1 | Machine learning pipeline adopted from [Jain 00]. . . . .                                                | 44 |
| 2.2 | Example input configuration of a data stream $\mathcal{S}(n_s, m_s, t)$ for two sensors. . . . .         | 46 |
| 2.3 | Feature selection approach using the relative importance $\Psi_{RI}$ . . . . .                           | 50 |
| 2.4 | Simplified SVM example with a linear kernel functionality. . . . .                                       | 52 |
| 3.1 | Visualization of the computational dependencies for the presented features. . . . .                      | 65 |
| 3.2 | Adaption of the machine learning pipeline to a real-time operating system. . . . .                       | 69 |
| 3.3 | Periodic scheduling example of a machine learning pipeline. . . . .                                      | 72 |
| 3.4 | Aperiodic scheduling example of a recognition task. . . . .                                              | 74 |
| 4.1 | Mechanical model of a one-dimensional accelerometer example according to [Ida 14]. . . . .               | 81 |
| 4.2 | Mechanical model of a one-dimensional gyroscope example according to [Crai 12]. . . . .                  | 82 |
| 4.3 | Example of the Hall effect and the AMR effect. . . . .                                                   | 84 |
| 4.4 | Example of the inverse piezo-electric effect. . . . .                                                    | 86 |
| 4.5 | Components of the <i>miPod</i> sensor platforms. . . . .                                                 | 90 |

|      |                                                                                                |     |
|------|------------------------------------------------------------------------------------------------|-----|
| 5.1  | Axes alignment and coordinate system orientation of the <i>miPod V1</i> sensor. . . . .        | 96  |
| 5.2  | Sub-exercise examples for the exercise scenario. . . . .                                       | 99  |
| 5.3  | Playing examples for the gameplay scenario. . . . .                                            | 101 |
| 5.4  | Stroke detection processing pipeline of the exercise scenario. . . . .                         | 102 |
| 5.5  | Example stroke detection data of the exercise scenario. . . . .                                | 105 |
| 5.6  | Stroke detection processing pipeline for the gameplay scenario. . . . .                        | 106 |
| 5.7  | Example stroke detection data of the gameplay scenario. . . . .                                | 107 |
| 5.8  | Stroke classification pipeline for the exercise scenario data. . . . .                         | 109 |
| 5.9  | Stroke classification pipeline of the gameplay scenario data. . . . .                          | 111 |
| 5.10 | Peak detection dependent F1-Scores for different $h$ -values. . . . .                          | 113 |
| 6.1  | Example high-speed video data of the labeled ball. . . . .                                     | 126 |
| 6.2  | Experimental pendulum system for the video labeling reference. . . . .                         | 127 |
| 6.3  | Evaluation data of the pendulum validation approach. . . . .                                   | 128 |
| 6.4  | Racket prototype with sensor placement and geometric definitions. . . . .                      | 130 |
| 6.5  | Schematic overview of the ball spin and speed exercise. . . . .                                | 131 |
| 6.6  | Different exercise examples from the video reference. . . . .                                  | 132 |
| 6.7  | Schematic overview of the racket speed $v_R(x_i)$ calculation. . . . .                         | 136 |
| 6.8  | Example acceleration and angular velocity data of a top-spin stroke. . . . .                   | 142 |
| 6.9  | Schematic overview of the rebound model for the transformation of ball spin and speed. . . . . | 143 |
| 6.10 | Estimated ball speed $v_{B,2(xz)}$ . . . . .                                                   | 149 |
| 6.11 | Estimated ball spin $\omega_{B,2(y)}$ . . . . .                                                | 149 |
| 6.12 | Ball speed difference $(v_{B,2(xz)} - v_b)$ of estimated and labeled data. . . . .             | 150 |
| 6.13 | Ball spin difference $(\omega_{B,2(xz)} - \omega_b)$ of estimated and labeled data. . . . .    | 150 |
| 7.1  | Theoretical TDoA measurement approach on the surface of a table tennis racket. . . . .         | 160 |
| 7.2  | Profile view of a racket assembled with classic pimples inside rubbers. . . . .                | 161 |
| 7.3  | Piezo-electric sensor positions and their orientation on an example racket prototype. . . . .  | 163 |

|      |                                                                                                   |     |
|------|---------------------------------------------------------------------------------------------------|-----|
| 7.4  | Distribution of all 48 measuring points $P_m(X, Y)$ within a grid of 20 mm on the racket. . . . . | 165 |
| 7.5  | Component overview of the piezo-electric conditioning circuit. . . . .                            | 166 |
| 7.6  | Example vibration data for the impact position $P_m(20)$ . . . . .                                | 169 |
| 7.7  | Detailed view of the vibration behavior for impact position $P_m(20)$ . . . . .                   | 171 |
| 7.8  | Example interpolated TDDM as colored maps. . . . .                                                | 173 |
| 7.9  | Visual localization results for an impact on blade position $P_m(20)$ . . . . .                   | 177 |
| 7.10 | Example RMSE distribution for impact localization in X-direction and Y-direction. . . . .         | 182 |
| 8.1  | Racket instrumentation: integration of the piezo-electric vibration sensors. . . . .              | 191 |
| 8.2  | Racket instrumentation: main electronics and handle integration. . . . .                          | 192 |
| 8.3  | Processing pipeline of the algorithms implemented on the microcontroller. . . . .                 | 194 |
| 8.4  | Feedback application: recording view of the free training mode. . . . .                           | 199 |
| 8.5  | Feedback application: example evaluation of the rally overview. . . . .                           | 199 |
| 8.6  | Feedback application: example evaluation of the ball spin and speed distribution. . . . .         | 200 |



# List of Tables

|      |                                                                                                            |     |
|------|------------------------------------------------------------------------------------------------------------|-----|
| 2.1  | List of all used generic time-domain features. . . . .                                                     | 48  |
| 2.2  | Confusion matrix example with four classes. . . . .                                                        | 55  |
| 4.1  | Overview and comparison of the available sensor types and features for the <i>miPod</i> platforms. . . . . | 93  |
| 5.1  | Sensor parameters during the exercise and gameplay scenarios. . . . .                                      | 97  |
| 5.2  | Overview of the exercise scenario study group. . . . .                                                     | 98  |
| 5.3  | Session protocol of an exercise scenario including all sub-exercises. . . . .                              | 100 |
| 5.4  | Overview of the gameplay scenario study group. . . . .                                                     | 101 |
| 5.5  | List of all labeled and detected strokes for the exercise scenario. . . . .                                | 112 |
| 5.6  | List of all labeled and detected strokes of the gameplay scenario. . . . .                                 | 113 |
| 5.7  | Stroke validation evaluation of the gameplay scenario data. . . . .                                        | 114 |
| 5.8  | Feature selection results for gameplay data of the stroke validation. . . . .                              | 115 |
| 5.9  | Overall classification accuracies and computational efforts of the exercise scenario. . . . .              | 115 |
| 5.10 | Confusion matrix of the RBF SVM classifier of the exercise scenario data. . . . .                          | 116 |
| 5.11 | Stroke classification evaluation of the gameplay scenario data. . . . .                                    | 117 |
| 5.12 | Confusion matrix of the RBF $\nu$ -SVM classifier of the gameplay scenario data. . . . .                   | 117 |
| 6.1  | Sensor parameters for ball speed and spin estimation. . . . .                                              | 129 |
| 6.2  | Overview of the ball spin and speed study group. . . . .                                                   | 131 |

|     |                                                                                                                |     |
|-----|----------------------------------------------------------------------------------------------------------------|-----|
| 6.3 | Overall results for the ball speed estimation $v_{B,2(xz)}$ . . . .                                            | 148 |
| 6.4 | Overall results for the ball spin estimation $\omega_{B,2(y)}$ . . . .                                         | 151 |
| 7.1 | Shore hardness and weight of all tested racket-rubber combinations. . . . .                                    | 164 |
| 7.2 | Evaluation results for the sweet spot area of the TDDM based on the racket-rubber combination (No. 1). . . . . | 179 |
| 7.3 | Evaluation results are shown for the TDDM based on racket-rubber combination (No. 1). . . . .                  | 180 |
| 7.4 | Evaluation results for the TDDM based on racket-rubber combination (No. 4). . . . .                            | 181 |
| 8.1 | Minimum RAM requirements for the implemented algorithms. . . . .                                               | 195 |
| 8.2 | Minimum FLASH requirements for the implemented algorithms. . . . .                                             | 197 |

# Bibliography

- [Ahma 06] A. Ahmadi, D. Rowlands, and D. A. James. “Investigating the translational and rotational motion of the swing using accelerometers for athlete skill assessment”. In: *Proceedings of the IEEE Conference on Sensors*, pp. 980–983, Daegu, South Korea, 2006.
- [Ahma 09] A. Ahmadi, D. Rowlands, and D. A. James. “Towards a wearable device for skill assessment and skill acquisition of a tennis player during the first serve”. *Sports Technology*, Vol. 2, No. 3-4, pp. 129–136, 2009.
- [Ahma 14] A. Ahmadi, F. Destelle, D. Monaghan, N. E. O’Connor, C. Richter, and K. Moran. “A framework for comprehensive analysis of a swing in sports using low-cost inertial sensors”. In: *Proceedings of the IEEE Conference on Sensors*, pp. 2211–2214, Valencia, Spain, 2014.
- [Alpa 14] E. Alpaydin. *Introduction to Machine Learning*. MIT Press, Cambridge, 2014.
- [Amft 09] O. Amft and P. Lukowicz. “From backpacks to smartphones: Past, present, and future of wearable computers”. *IEEE Pervasive Computing*, Vol. 8, No. 3, pp. 8–13, 2009.
- [Anan 17] A. Anand, M. Sharma, R. Srivastava, L. Kaligounder, and D. Prakash. “Wearable Motion Sensor Based Analysis of Swing Sports”. In: *Proceedings of the 16<sup>th</sup> IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 261–267, Cancun, Mexico, 2017.
- [Anli 04] U. Anliker, J. Beutel, M. Dyer, R. Enzler, P. Lukowicz, L. Thiele, and G. Tröster. “A systematic approach to the de-

- sign of distributed wearable systems". *IEEE Transactions on Computers*, Vol. 53, No. 8, pp. 1017–1033, 2004.
- [Arlo 10] S. Arlot and A. Celisse. "A survey of cross-validation procedures for model selection". *Statistics surveys*, Vol. 4, pp. 40–79, 2010.
- [ARM 10] ARM Limited. "Cortex-M4 Technical Reference Manual". Technical Report, ARM Limited, 2010.
- [Avci 10] A. Avci, S. Bosch, M. Marin-Perianu, R. Marin-Perianu, and P. Havinga. "Activity Recognition Using Inertial Sensing for Healthcare, Wellbeing and Sports Applications: A Survey". In: *Proceedings of the 23<sup>th</sup> International Conference on Architecture of Computing Systems (ARCS)*, pp. 1–10, Hannover, Germany, 2010.
- [Baca 06] A. Baca and P. Kornfeind. "Rapid feedback systems for elite sports training". *IEEE Pervasive Computing*, Vol. 5, No. 4, pp. 70–76, 2006.
- [Baca 14] A. Baca. *Computer science in sport: research and practice*. Routledge, London New York, 2014.
- [Bao 04] L. Bao and S. S. Intille. "Activity recognition from user-annotated acceleration data". In: *Proceedings of the 2<sup>nd</sup> International Conference on Pervasive Computing*, pp. 1–17, Vienna, Austria, 2004.
- [Barb 96] C. B. Barber, D. P. Dobkin, and H. T. Huhdanpaa. "The quick-hull algorithm for convex hulls". *ACM Transactions on Mathematical Software (TOMS)*, Vol. 22, No. 4, pp. 469–483, 1996.
- [Barr 12] R. Barry. "FreeRTOS reference manual". Technical Report, Real Time Engineers Ltd., 2012.
- [Blan 14a] P. Blank, P. Kugler, and B. M. Eskofier. "miPod - A Wearable Sports and Fitness Sensor". In: *10. Symposium der dvs-Sektion Sportinformatik*, pp. 78–79, Vienna, Austria, 2014.
- [Blan 14b] P. Blank, P. Kugler, H. Schlarb, and B. M. Eskofier. "A Wearable Sensor System for Sports and Fitness Applications". In:

- Book of Abstracts of the 20<sup>th</sup> Annual Congress of the European College of Sport Science (ECSS)*, p. 703, Amsterdam, The Netherlands, 2014.
- [Blan 15] P. Blank, J. Hoßbach, D. Schuldhaus, and B. M. Eskofier. “Sensor-based stroke detection and stroke type classification in table tennis”. In: *Proceedings of the 2015 ACM International Symposium on Wearable Computers (ISWC)*, pp. 93–100, Osaka, Japan, 2015.
- [Blan 16a] P. Blank, S. Hofmann, M. Kulesa, and B. M. Eskofier. “mi-Pod 2: a new hardware platform for embedded real-time processing in sports and fitness applications”. In: *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pp. 881–884, Heidelberg, Germany, 2016.
- [Blan 16b] P. Blank, T. Kautz, and B. M. Eskofier. “Ball impact localization on table tennis rackets using piezo-electric sensors”. In: *Proceedings of the 2016 ACM International Symposium on Wearable Computers (ISWC)*, pp. 72–79, Heidelberg, Germany, 2016.
- [Blan 17] P. Blank, B. H. Groh, and B. M. Eskofier. “Ball speed and spin estimation in table tennis using a racket-mounted inertial sensor”. In: *Proceedings of the 2017 ACM International Symposium on Wearable Computers (ISWC)*, pp. 2–9, Maui, Hawaii, USA, 2017.
- [Blan 18] P. Blank and B. M. Eskofier. “Smart Racket - Echtzeitfeedbacksystem im Tischtennis”. *Technologien im Leistungssport 3: Tagungsband zur 19. Frühjahrsschule am 14./15. Mai 2018 in Leipzig*, Vol. 3, No. 13, pp. 126–133, 2018.
- [Blan 86] J. M. Bland and D. G. Altman. “Statistical methods for assessing agreement between two methods of clinical measurement”. *The Lancet*, Vol. 327, No. 8476, pp. 307–310, 1986.
- [Boot 90] R. J. Bootsma and P. C. W. van Wieringen. “Timing an attacking forehand drive in table tennis”. *Journal of Experimental Psychology: Human Perception and Performance*, Vol. 16, No. 1, p. 21, 1990.

- [Bose 92] B. E. Boser, I. M. Guyon, and V. N. Vapnik. "A training algorithm for optimal margin classifiers". In: *Proceedings of the 5<sup>th</sup> Annual Workshop on Computational Learning Theory*, pp. 144–152, Pittsburgh, USA, 1992.
- [Boye 13] E. Boyer, F. Bevilacqua, F. Phal, and S. Hanneton. "Low-cost motion sensing of table tennis players for real time feedback". *International Journal of Table Tennis Sciences*, Vol. 8, pp. 1–4, 2013.
- [Brei 01] L. Breiman. "Random forests". *Machine learning*, Vol. 45, No. 1, pp. 5–32, 2001.
- [Brig 45] L. J. Briggs. *Methods for measuring the coefficient of restitution and the spin of a ball*. Vol. 34, United States Department of Commerce, National Bureau of Standards, 1945.
- [Brod 86] H. Brody. "The sweet spot of a baseball bat". *American Journal of Physics*, Vol. 54, No. 7, pp. 640–643, 1986.
- [Bruc 01] E. N. Bruce. *Biomedical signal processing and signal modeling*. Vol. 49, John Wiley & Sons, New York, 2001.
- [Cai 12] Y. Cai, Y. Zhao, X. Ding, and J. Fennelly. "Magnetometer basics for mobile phone applications". *Electronic Products*, Vol. 54, No. 2, pp. 1–3, 2012.
- [Camp 06] D. Campolo, M. Fabris, G. Cavallo, D. Accoto, F. Keller, and E. Guglielmelli. "A novel procedure for in-field calibration of sourceless inertial/magnetic orientation tracking wearable devices". In: *Proceedings of the 1<sup>st</sup> IEEE International Conference on Biomedical Robotics and Biomechatronics (RAS-EMBS)*, pp. 471–476, Pisa, Italy, 2006.
- [Cham 15] R. Chambers, T. J. Gabbett, M. H. Cole, and A. Beard. "The use of wearable microsensors to quantify sport-specific movements". *Sports Medicine*, Vol. 45, No. 7, pp. 1065–1081, 2015.
- [Chan 11] C. Chang and C. Lin. "LIBSVM: a library for support vector machines". *ACM Transactions on Intelligent Systems and Technology (TIST)*, Vol. 2, No. 3, pp. 1–27, 2011.

- [Chen 03] A. M. K. Cheng. *Real-time systems: scheduling, analysis, and verification*. Wiley-Interscience, Hoboken, 2003.
- [Chen 06] W. Chen and Y.-J. Zhang. “Tracking ball and players with applications to highlight ranking of broadcasting table tennis video”. In: *Proceedings of the Multiconference on Computational Engineering in Systems Applications (CESA)*, pp. 1896–1903, Beijing, China, 2006.
- [Chen 10] X. Chen, Y. Tian, Q. Huang, W. Zhang, and Z. Yu. “Dynamic model based ball trajectory prediction for a robot ping-pong player”. In: *Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 603–608, Tianjin, China, 2010.
- [Chen 12] Y. F. Chen. “Development of a Two-Dimensional Laser Array Sensor to Detect the Ball Location for Table Tennis Training”. *Advanced Science Letters*, Vol. 8, No. 5, pp. 257–261, 2012.
- [Chi 05] E. H. Chi, G. Borriello, G. Hunt, and N. Davies. “Guest editors’ introduction: Pervasive computing in sports technologies”. *IEEE Pervasive Computing*, Vol. 4, No. 3, pp. 22–25, 2005.
- [Chi 08] E. H. Chi. “Sensors and ubiquitous computing technologies in sports”. *WIT Transactions on State-of-the-art in Science and Engineering*, Vol. 32, No. 1, pp. 249–267, 2008.
- [Conn 10] D. Connaghan, C. O. Conaire, P. Kelly, and N. E. O’Connor. “Recognition of tennis strokes using key postures”. In: *Proceedings of the IET Irish Signals and Systems Conference (ISSC)*, pp. 1–4, Cork, Ireland, 2010.
- [Conn 11a] D. Connaghan, P. Kelly, and N. E. O’Connor. “Game, shot and match: Event-based indexing of tennis”. In: *Proceedings of the 9<sup>th</sup> International Workshop on Content-Based Multimedia Indexing (CBMI)*, pp. 97–102, Madrid, Spain, 2011.
- [Conn 11b] D. Connaghan, P. Kelly, N. E. O’Connor, M. Gaffney, M. Walsh, and C. O’Mathuna. “Multi-sensor classification of tennis strokes”. In: *Proceedings of the IEEE Conference on Sensors*, pp. 1437–1440, Limerick, Ireland, 2011.

- [Cort 95] C. Cortes and V. Vapnik. "Support-Vector Networks". *Machine Learning*, Vol. 20, No. 3, pp. 273–297, 1995.
- [Cott 02] R. A. Cottey. *The modelling of spin generation with particular emphasis on racket ball games*. PhD thesis, Loughborough University, 2002.
- [Cove 67] T. Cover and P. Hart. "Nearest Neighbor Pattern Classification". *IEEE Transactions on Information Theory*, Vol. 13, No. 1, pp. 21–27, 1967.
- [Crai 12] K. Craig. "Modeling the MEMS Gyroscope". Technical Report, EDN Network, 2012.
- [Cros 02a] R. Cross. "Grip-slip behavior of a bouncing ball". *American Journal of Physics*, Vol. 70, No. 11, pp. 1093–1102, 2002.
- [Cros 02b] R. Cross. "Measurements of the horizontal coefficient of restitution for a superball and a tennis ball". *American Journal of Physics*, Vol. 70, No. 5, pp. 482–489, 2002.
- [Cros 05] R. Cross. "Bounce of a spinning ball near normal incidence". *American Journal of Physics*, Vol. 73, No. 10, pp. 914–920, 2005.
- [Cros 14] R. Cross. "Impact of sports balls with striking implements". *Sports Engineering*, Vol. 17, No. 1, pp. 3–22, 2014.
- [Cros 97] R. Cross. "The dead spot of a tennis racket". *American Journal of Physics*, Vol. 65, No. 8, pp. 754–764, 1997.
- [Cros 98a] R. Cross. "The sweet spot of a baseball bat". *American Journal of Physics*, Vol. 66, No. 9, pp. 772–779, 1998.
- [Cros 98b] R. Cross. "The sweet spots of a tennis racquet". *Sports Engineering*, Vol. 1, No. 2, pp. 63–78, 1998.
- [Cros 99a] R. Cross. "Dynamic properties of tennis balls". *Sports Engineering*, Vol. 2, No. 1, pp. 23–34, 1999.
- [Cros 99b] R. Cross. "Impact of a ball with a bat or racket". *American Journal of Physics*, Vol. 67, No. 8, pp. 692–702, 1999.

- [Curi 82] J. Curie and P. Curie. “Phénomènes électriques des cristaux hémiedres à faces inclinées”. *Journal de Physique Théorique et Appliquée*, Vol. 1, No. 1, pp. 245–251, 1882.
- [Duan 05] L. Duan, M. Xu, Q. Tian, C. Xu, and J. S. Jin. “A unified framework for semantic shot classification in sports video”. *IEEE Transactions on Multimedia*, Vol. 7, No. 6, pp. 1066–1083, 2005.
- [Duda 12] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Vol. 2, John Wiley & Sons, New York, 2012.
- [Eibe 98] F. Eibe and I. H. Witten. “Generating accurate rule sets without global optimization”. In: *Proceedings of the 15<sup>th</sup> International Conference on Machine Learning (ICML)*, pp. 144–151, Madison, Wisconsin, USA, 1998.
- [Elso 02] J. Elson, L. Girod, and D. Estrin. “Fine-grained network time synchronization using reference broadcasts”. *ACM SIGOPS Operating Systems Review*, Vol. 36, No. SI, pp. 147–163, 2002.
- [Ferd 03] C. Ferdinand, D. Kästner, F. Martin, M. Langenbach, M. Sicks, S. Wilhelm, R. Heckmann, N. Fritz, S. Thesing, F. Fontaine, H. Theiling, M. Schmidt, A. Evstiougov-Babaec, and R. Wilhelm. “Validierung des Zeitverhaltens von kritischer Echtzeit-Software”. In: *Proceedings of the 33. Jahrestagung der Gesellschaft für Informatik e.V. (GI)*, pp. 335–339, Frankfurt am Main, Germany, 2003.
- [Ferr 95] F. Ferraris, U. Grimaldi, and M. Parvis. “Procedure for Effortless Infield Calibration of Three-Axial Rate Gyro and Accelerometers”. *Sensors and Materials*, Vol. 7, No. 5, pp. 311–330, 1995.
- [Fuss 12] F. K. Fuss, R. M. Smith, and A. Subic. “Determination of spin rate and axes with an instrumented cricket ball”. *Procedia Engineering*, Vol. 34, No. 1, pp. 128–133, 2012.
- [Gane 03] S. Ganeriwal, R. Kumar, and M. B. Srivastava. “Timing-sync protocol for sensor networks”. In: *Proceedings of the 1<sup>st</sup> International Conference on Embedded Networked Sensor Systems*, pp. 138–149, Los Angeles, California, USA, 2003.

- [Garw 69] R. L. Garwin. “Kinematics of an ultraelastic rough ball”. *American Journal of Physics*, Vol. 37, No. 1, pp. 88–92, 1969.
- [Glov 14] J. Glover and L. P. Kaelbling. “Tracking the spin on a ping pong ball with the quaternion Bingham filter”. In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4133–4140, Hong Kong, China, 2014.
- [Good 04] S. R. Goodwill and S. J. Haake. “Ball spin generation for oblique impacts with a tennis racket”. *Experimental Mechanics*, Vol. 44, No. 2, pp. 195–206, 2004.
- [Gree 99] D. W. Green, J. E. Winandy, and D. E. Kretschmann. “Mechanical properties of wood”. Technical Report, USDA Forest Service, Forest Products Laboratory, 1999.
- [Grid 07] G. Gridling and B. Weiss. “Introduction to microcontrollers”. Technical Report, Vienna University of Technology Institute of Computer Engineering Embedded Computing Systems Group, 2007.
- [Gros 07] B. U. Groß. *Tipps für Tischtennis*. Vol. 4, Meyer & Meyer Verlag, Aachen, 2007.
- [Guo 10] Y.-W. Guo, G.-Z. Liu, B.-Y. Huang, G.-R. Zhao, Z.-Y. Mei, and L. Wang. “A pilot study on quantitative analysis for table tennis block using a 3d accelerometer”. In: *Proceedings of the 10<sup>th</sup> IEEE International Conference on Information Technology and Applications in Biomedicine (ITAB)*, pp. 1–4, Corfu, Greece, 2010.
- [Guy0 03] I. Guyon and A. Elisseeff. “An introduction to variable and feature selection”. *Journal of Machine Learning Research*, Vol. 3, No. 3, pp. 1157–1182, 2003.
- [Guy0 06] I. Guyon and A. Elisseeff. *An introduction to feature extraction*. Vol. 207, Springer, Berlin Heidelberg, 2006.
- [Haak 96] S. Haake. *The Engineering of Sport*. Taylor & Francis, London New York, 1996.
- [Haas 29] A. E. Haas. *Introduction to Theoretical Physics*. Vol. 2, Van Nostrand, New York, 1929.

- [Hahn 17] M. Hahnenkamm. *Ball Impact Localization on Tables and Ball-Net Contact Detection in Table Tennis using Mobile Sensors*. Master's thesis, Friedrich-Alexander University of Erlangen-Nürnberg, supervised by the author, 2017.
- [Hall 09] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. "The WEKA data mining software: an update". *ACM SIGKDD Explorations Newsletter*, Vol. 11, No. 1, pp. 10–18, 2009.
- [Hall 79] E. H. Hall. "On a new action of the magnet on electric currents". *American Journal of Mathematics*, Vol. 2, No. 3, pp. 287–292, 1879.
- [Heav 89] O. Heaviside. "On the electromagnetic effects due to the motion of electrification through a dielectric". *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, Vol. 27, No. 167, pp. 324–339, 1889.
- [Hofm 14] F. Hofmann. *9D-IMU Calibration and Alignment Correction*. Bachelor's thesis, Friedrich-Alexander University of Erlangen-Nürnberg, 2014.
- [Holz 94] H. J. Holz and M. H. Loew. "Relative feature importance: A classifier-independent approach to feature selection". *Machine Intelligence and Pattern Recognition*, Vol. 16, No. 1, pp. 473–487, 1994.
- [Hou 04] H. Hou. *Modeling inertial sensors errors using Allan variance*. Master's thesis, University of Calgary, 2004.
- [Huan 11] Y. Huang, D. Xu, M. Tan, and H. Su. "Trajectory prediction of spinning ball for ping-pong player robot". In: *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS)*, pp. 3434–3439, San Francisco, California, USA, 2011.
- [Ida 14] N. Ida. *Sensors, Actuators, and Their Interfaces: A Multidisciplinary Introduction*. Vol. 2, SciTech Publishing, Edison, 2014.
- [Iiji 10] Y. Iijima, K. Watanabe, K. Kobayashi, and Y. Kurihara. "Measurement and analysis of tennis swing motion using 3D gyro

- sensor". In: *Proceedings of the SICE Annual Conference*, pp. 274–277, Taipei, Taiwan, 2010.
- [Illo 14] D. Illouz. "Spin Rate of a Racquetball Due To Angular Impact". *ISB Journal of Science*, Vol. 8, No. 5, pp. 1–4, 2014.
- [Inco 14] Incorporated Research Institutions for Seismology. "How are Earthquakes Located?". Technical Report, The IRIS consortium, 2014.
- [Jain 00] A. K. Jain, R. P. W. Duin, and J. Mao. "Statistical Pattern Recognition: A Review". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, No. 1, pp. 4–37, 2000.
- [Jens 16] U. Jensen. *Design Considerations and Application Examples for Embedded Classification Systems*. PhD thesis, Friedrich-Alexander University Erlangen-Nürnberg, 2016.
- [Jin 18] B. Jin, X. Xu, and T. Zhang. "Robust Time-Difference-of-Arrival (TDOA) Localization Using Weighted Least Squares with Cone Tangent Plane Constraint". *MDPI Sensors*, Vol. 18, No. 3, p. 778, 2018.
- [Jones 13] T. B. Jones and N. G. Nenadic. *Electromechanics and MEMS*. Cambridge University Press, Cambridge, 2013.
- [Kami 13] K. Kamijima, Y. Ushiyama, T. Yasaka, and M. Ooba. "Effect of different playing surfaces of the table on ball bounces in table tennis". *International Journal of Table Tennis Science*, Vol. 1, No. 1, pp. 11–14, 2013.
- [Kawa 03] Y. Kawazoe and D. Suzuki. "Prediction of table tennis racket restitution performance based on the impact analysis". *Theoretical and Applied Mechanics*, Vol. 52, No. 1, pp. 163–174, 2003.
- [Kemp 11] V. Kempe. *Inertial MEMS: Principles and Practice*. Cambridge University Press, Cambridge, 2011.
- [Kian 09] C. T. Kiang, C. K. Yoong, and A. Spowage. "Local sensor system for badminton smash analysis". In: *Proceedings of the IEEE Instrumentation and Measurement Technology Conference (I2MTC)*, pp. 883–888, Singapore, Singapore, 2009.

- [Kita 13] H. Kita and T. Kawamoto. “Specification for Transducer PKGS-00LDP1-R”. Technical Report, Murata Manufacturing, 2013.
- [Knut 97] D. E. Knuth. *The Art of Computer Programming; Volume 2: Seminumerical Algorithms*. Addison-Wesley, Upper Saddle River, 1997.
- [Koop 90] P. Koopman. “Design constraints on embedded real time control systems”. Technical Report, Harris Semiconductor, 1990.
- [Kope 02] H. Kopetz. *Real-time systems: design principles for distributed embedded applications*. Kluwer Academic Publishers, Bosten, 2002.
- [Kule 17] M. Kulesa. *Synchronization Methods of Wearable Sensors*. Master’s thesis, Friedrich-Alexander University of Erlangen-Nürnberg, supervised by the author, 2017.
- [Kuo 13] S. M. Kuo, B. H. Lee, and W. Tian. *Real-time digital signal processing: fundamentals, implementations and applications*. John Wiley & Sons, New York, 2013.
- [Lapl 77] P.-S. de Laplace. *Recherches sur plusieurs points de système du monde*. Mémoires de l’Académie royale des sciences de Paris, Paris, 1777.
- [Lasa 10] S. M. Lasassmeh and J. M. Conrad. “Time synchronization in wireless sensor networks: A survey”. In: *Proceedings of the IEEE SoutheastCon Conference (SoutheastCon)*, pp. 242–245, Concord, North Carolina, USA, 2010.
- [Lee 11] K. B. Lee. *Principles of microelectromechanical systems*. John Wiley & Sons, New York, 2011.
- [Leit 11] K. Leitis and T. Halim. “Modeling of a 3D magnetic field (AMR) sensor”. In: *Proceeding of the IEEE International Conference n Microwaves, Communications, Antennas and Electronics Systems (COMCAS)*, pp. 1–4, Tel Aviv, Israel, 2011.
- [Leut 13] H. Leutheuser, D. Schuldhaus, and B. M. Eskofier. “Hierarchical, multi-sensor based classification of daily life activities: comparison with state-of-the-art algorithms using a

- benchmark dataset". *PLOS ONE*, Vol. 8, No. 10, p. e75196, 2013.
- [Lewi 98] D. D. Lewis. "Naive (Bayes) at forty: The independence assumption in information retrieval". In: *Proceedings of the European Conference on Machine Learning (EMCL)*, pp. 4–15, Chemnitz, Germany, 1998.
- [Lin 17] J. Lin, C.-W. Chang, C.-H. Wang, H.-C. Chi, C.-W. Yi, Y.-C. Tseng, and C.-C. Wang. "Design and implement a mobile badminton stroke classification system". In: *Proceedings of the 19<sup>th</sup> Asia-Pacific Network Operations and Management Symposium (APNOMS)*, pp. 235–238, Seoul, South Korea, 2017.
- [Liu 00] F. Liu, A. Narayanan, and Q. Bai. *Real-time systems*. Prentice-Hall, Upper Saddle River, 2000.
- [Liu 11a] C. Liu, Y. Hayakawa, and A. Nakashima. "A registration algorithm for on-line measuring the rotational velocity of a table tennis ball". In: *Proceedings of the IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 2270–2275, Karon Beach, Phuket, Thailand, 2011.
- [Liu 11b] H. Liu, X. Wei, J. Chai, I. Ha, and T. Rhee. "Realtime human motion control with a small number of inertial sensors". In: *Proceedings of the 11<sup>th</sup> Symposium on Interactive 3D Graphics and Games (i3D)*, pp. 133–140, San Francisco, California, USA, 2011.
- [Luko 04] P. Lukowicz, T. Kirstein, and G. Tröster. "Wearable systems for health care applications". *Methods of information in medicine*, Vol. 43, No. 3, pp. 232–238, 2004.
- [Mani 12a] L. Manin, F. Gabert, M. Poggi, and N. Havard. "Vibro-acoustic of table tennis rackets at ball impact: Influence of the blade plywood composition". *Procedia Engineering*, Vol. 34, No. 1, pp. 604–609, 2012.
- [Mani 12b] L. Manin, M. Poggi, and N. Havard. "Vibrations of table tennis racket composite wood blades: modeling and experiments". *Procedia Engineering*, Vol. 34, No. 1, pp. 694–699, 2012.

- [Marw 06] P. Marwedel. *Embedded system design*. Springer, Berlin Heidelberg, 2006.
- [McAf 09] R. McAfee. *Table tennis: Steps to success*. Human Kinetics, Champaign, 2009.
- [McCa 09] J. McCann and D. Bryson. *Smart clothes and wearable technology*. Woodhead Publishing, Cambridge, 2009.
- [Meas 13] Measurement Specialties Sensor Products Division. “Piezo Film Sensors Technical Manual”. Technical Report, Measurement Specialties, Inc., 2013.
- [Mori 06] E. F. Moritz and S. Haake. *The Engineering of Sport 6*. Vol. 3, Springer, Berlin Heidelberg, 2006.
- [Morr 08] D. Morris, B. Schazmann, Y. Wu, S. Coyle, S. Brady, J. Hayes, C. Slater, C. Fay, K. T. Lau, G. Wallace, *et al.* “Wearable sensors for monitoring sports performance and training”. In: *Proceedings of the 5<sup>th</sup> IEEE International Summer School and Symposium on Medical Devices and Biosensors (MDBS)*, pp. 121–124, Hong Kong, China, 2008.
- [Mucc 13] E. Mucchi. “On the sweet spot estimation in beach tennis rackets”. *Measurement*, Vol. 46, No. 4, pp. 1399–1410, 2013.
- [Naga 12] M. Nagasawa, Y. Hatori, M. Kakuta, T. Hayashi, and Y. Sekine. “Smash motion analysis for badminton from image”. In: *Proceeding of the IEEE 3<sup>rd</sup> Image Electronics and Visual Computing Workshop (IEVC)*, Kuching, Malaysia, 2012.
- [Naka 09] A. Nakashima, Y. Kobayashi, Y. Ogawa, and Y. Hayakawa. “Modeling of rebound phenomenon between ball and racket rubber with spinning effect”. In: *Proceedings of the ICCAS-SICE International Joint Conference*, pp. 2295–2300, Fukuoka, Japan, 2009.
- [Naka 10] A. Nakashima, Y. Ogawa, Y. Kobayashi, and Y. Hayakawa. “Modeling of rebound phenomenon of a rigid ball with friction and elastic effects”. In: *Proceedings of the 2010 American Control Conference (ACC)*, pp. 1410–1415, Baltimore, Maryland, USA, 2010.

- [Nath 08] A. M. Nathan. “The effect of spin on the flight of a baseball”. *American Journal of Physics*, Vol. 76, No. 2, pp. 119–124, 2008.
- [Newt 87] I. Newton. *Philosophiæ naturalis principia mathematica*. Jussu Societatis Regiæ ac typis Josephi Streater, Londini, 1687.
- [Nico 13] A. Nicolaides, N. Elliott, J. Kelley, M. Pinaffo, and T. Allen. “Effect of string bed pattern on ball spin generation from a tennis racket”. *Sports Engineering*, Vol. 16, No. 3, pp. 181–188, 2013.
- [Niem 83] H. Niemann. *Klassifikation von Mustern*. Springer, Berlin Heidelberg, 1983.
- [Nono 10] J. Nonomura, A. Nakashima, and Y. Hayakawa. “Analysis of effects of rebounds and aerodynamics for trajectory of table tennis ball”. In: *Proceedings of the SICE Annual Conference*, pp. 1567–1572, Taipei Taiwan, 2010.
- [O Co 10] C. O Conaire, D. Connaghan, P. Kelly, N. E. O'Connor, M. Gaffney, and J. Buckley. “Combining inertial and visual sensing for human action recognition in tennis”. In: *Proceedings of the 1<sup>st</sup> ACM International Workshop on Analysis and Retrieval of Tracked Events and Motion in Imagery Streams (ARTEMIS)*, pp. 51–56, Firenze, Italy, 2010.
- [Oldh 15] K. M. Oldham. *Table tennis event detection and classification*. PhD thesis, Loughborough University, 2015.
- [Olip 15] T. E. Oliphant. *Guide to NumPy*. Vol. 2, CreateSpace Independent Publishing Platform, 2015.
- [Otex 17] B. O’Keefe. “Finding Location with Time of Arrival and Time Difference of Arrival Techniques”. Technical Report, Department of Electrical & Computer Engineering, Tufts University, 2017.
- [Pals 09] G. Palshikar. “Simple Algorithms for Peak Detection in Time-series”. In: *In Proceedings of the IIMA 1<sup>st</sup> International Conference of Advanced Data Analysis, Business Analytics and Intelligence*, pp. 1–14, Ahmedabad, Gujarat, India, 2009.

- [Parz 62] E. Parzen. "On estimation of a probability density function and mode". *The Annals of Mathematical Statistics*, Vol. 33, No. 3, pp. 1065–1076, 1962.
- [Pedr 11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. "Scikit-learn: Machine Learning in Python". *Journal of Machine Learning Research*, Vol. 12, No. 1, pp. 2825–2830, 2011.
- [Petk 01] M. Petkovic, W. Jonker, and Z. Zivkovic. "Recognizing Strokes in Tennis Videos Using Hidden Markov Models". In: *Proceedings of the IASTED International Conference on Visualization, Imaging and Image Processing (VIIP)*, pp. 512–516, Marbella, Spain, 2001.
- [Raab 05] M. Raab, R. S. W. Masters, and J. P. Maxwell. "Improving the 'how' and 'what' decisions of elite table tennis players". *Human Movement Science*, Vol. 24, No. 3, pp. 326–344, 2005.
- [Rama 14] S. Ramasinghe, K. M. Chathuramali, and R. Rodrigo. "Recognition of badminton strokes using dense trajectories". In: *Proceedings of the 7<sup>th</sup> International Conference on Information and Automation for Sustainability (ICIAfS)*, pp. 1–6, Colombo, Sri Lanka, 2014.
- [Ring 12] M. Ring, U. Jensen, P. Kugler, and B. M. Eskofier. "Software-based performance and complexity analysis for the design of embedded classification systems". In: *Proceedings of the 21<sup>th</sup> International Conference on Pattern Recognition (ICPR)*, pp. 2266–2269, Tsukuba, Japan, 2012.
- [Rodr 02] S. T. Rodrigues, J. N. Vickers, and M. A. Williams. "Head, eye and arm coordination in table tennis". *Journal of Sports Sciences*, Vol. 20, No. 3, pp. 187–200, 2002.
- [Rusd 05] S. Rusdorf and G. Brunnett. "Real time tracking of high speed movements in the context of a table tennis application". In: *Proceedings of the ACM Symposium on Virtual Reality Software and Technology (VRST)*, pp. 192–200, Monterey, California, USA, 2005.

- [Sali 10] M. S. Salim, H. N. Lim, M. S. M. Salim, and M. Y. Baharuddin. "Motion analysis of arm movement during badminton smash". In: *Proceedings of the IEEE Biomedical Engineering and Sciences (IECBES)*, pp. 111–114, Kuala Lumpur, Malaysia, 2010.
- [Sazo 14] E. Sazonov and M. R. Neuman. *Wearable Sensors: Fundamentals, implementation and applications*. Elsevier, Amsterdam, 2014.
- [Scho 00] B. Schölkopf, A. J. Smola, R. C. Williamson, and P. L. Bartlett. "New Support Vector Algorithms". *Neural Computation*, Vol. 12, No. 5, pp. 1207–1245, 2000.
- [Scot 15] D. W. Scott. *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons, New York, 2015.
- [Shar 11] V. Sharapov. *Piezoceramic sensors*. Springer, Berlin Heidelberg, 2011.
- [Shor 10] A. F. Shore. "Testing the hardness of metals". *Transactions (Society of Automobile Engineers)*, Vol. 5, No. 1, pp. 203–213, 1910.
- [Sich 03] M. L. Sichitiu and C. Veerarittiphan. "Simple, accurate time synchronization for wireless sensor networks". In: *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1266–1273, New Orleans, Louisiana, USA, 2003.
- [Sili 14] Silicon Laboratories Inc. "EFM32LG Reference Manual". Technical Report, Silicon Laboratories Inc., 2014.
- [Sivr 04] F. Sivrikaya and B. Yener. "Time synchronization in sensor networks: a survey". *IEEE Network*, Vol. 18, No. 4, pp. 45–50, 2004.
- [Smit 13] S. Smith. *Digital signal processing: a practical guide for engineers and scientists*. Elsevier, Amsterdam, 2013.
- [Sore 01] V. Sørensen, R. P. Ingvaldsen, and H. T. A. Whiting. "The application of co-ordination dynamics to the analysis of discrete movements using table-tennis as a paradigm skill". *Biological Cybernetics*, Vol. 85, No. 1, pp. 27–38, 2001.

- [Szep 11] A. Szép. “Measuring ball spin in monocular video”. In: *Proceedings of the 16<sup>th</sup> Computer Vision Winter Workshop (CVWW)*, pp. 83–89, Mitterberg, Austria, 2011.
- [Tama 04] T. Tamaki, T. Sugino, and M. Yamamoto. “Measuring ball spin by image registration”. In: *Proceedings of the 10<sup>th</sup> Korea-Japan Joint Workshop on Frontiers of Computer Vision (FCV)*, pp. 269–274, Hiroshima, Japan, 2004.
- [Tama 12] T. Tamaki, H. Wang, B. Raytchev, K. Kaneda, and Y. Ushiyama. “Estimating the spin of a table tennis ball using Inverse Compositional Image Alignment”. In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1457–1460, Kyoto, Japan, 2012.
- [The 15] The International Table Tennis Federation. “Technical Leaflet T9: Racket Control”. Technical Report, The International Table Tennis Federation, 2015.
- [The 18a] The International Table Tennis Federation. *Handbook 2018*. Vol. 46, The International Table Tennis Federation, Lausanne, 2018.
- [The 18b] The International Table Tennis Federation. “LARC (List Authorized Racket Coverings) No. 2018A”. Technical Report, The International Table Tennis Federation, 2018.
- [The 18c] The International Table Tennis Federation. “Technical Leaflet T4: Racket Coverings”. Technical Report, The International Table Tennis Federation, 2018.
- [Theo 09] S. Theodoridis and K. Koutroumbas. *Pattern recognition*. Elsevier, Amsterdam, 2009.
- [Tien 08] M.-C. Tien, Y.-T. Wang, C.-W. Chou, K.-Y. Hsieh, W.-T. Chu, J.-L. Wu, *et al.* “Event detection in tennis matches based on video data mining”. In: *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1477–1480, Hannover, Germany, 2008.
- [Trin 12] A. Tringham. “ARM Extends CMSIS with RTOS API and System View Description”. Technical Report, ARM Limited, 2012.

- [Van 03] J. Van Greunen and J. Rabaey. “Lightweight time synchronization for sensor networks”. In: *Proceedings of the 2<sup>nd</sup> ACM International Conference on Wireless Sensor Networks and Applications (WSNA)*, pp. 11–19, San Diego, California, USA, 2003.
- [Van 10] D. Van Wageningen and T. Staring. “The Qi wireless power standard”. In: *Proceedings of 14<sup>th</sup> International Power Electronics and Motion Control Conference (PEMC)*, pp. S15–25, Ohrid, Macedonia, 2010.
- [Wang 16] Z. Wang, M. Guo, and C. Zhao. “Badminton stroke recognition based on body sensor networks”. *IEEE Transactions on Human-Machine Systems*, Vol. 46, No. 5, pp. 769–775, 2016.
- [Warb 06] D. E. R. Warburton, C. W. Nicol, and S. S. D. Bredin. “Health benefits of physical activity: the evidence”. *Canadian Medical Association Journal*, Vol. 174, No. 6, pp. 801–809, 2006.
- [Wata 85] S. Watanabe. *Pattern Recognition: Human and Mechanical*. John Wiley & Sons, New York, 1985.
- [Wege 09] M. Wegener, T. Greco, K. Arlt, R. Schuberth, A. Morschhauser, and T. Otto. “Ferroelectric P(VDF-TrFE) as a large-scale piezoelectric sensor in a table-tennis racquet”. In: *Proceedings of the 18<sup>th</sup> IEEE International Symposium on the Applications of Ferroelectrics (ISAF)*, pp. 1–5, Xian, China, 2009.
- [West 99] J. Weston and C. Watkins. “Support Vector Machines for Multi-Class Pattern Recognition”. In: *Proceedings of the 7<sup>th</sup> European Symposium on Artificial Neural Networks (ESANN)*, pp. 219–224, Bruges, Belgium, 1999.
- [Whit 28] J. M. Whittaker. “The “Fourier” theory of the cardinal function”. *Proceedings of the Edinburgh Mathematical Society*, Vol. 1, No. 3, pp. 169–176, 1928.
- [Wide 16] R. Widenhorn. “The physics of juggling a spinning ping-pong ball”. *American Journal of Physics*, Vol. 84, No. 12, pp. 936–942, 2016.

- [Will 64] J. W. J. Williams. “Algorithm 232: heapsort”. *Communications of the ACM*, Vol. 7, No. 6, pp. 347–348, 1964.
- [Xiao 17] X. Xiao, J. Hedman, F. T. C. Tan, C.-W. Tan, E. T. K. Lim, T. Clemenson, S. Henningsson, R. Vatrupu, R. R. Mukkamala, and J. Van Hilleberg. “Sports digitalization: an overview and a research agenda”. In: *Proceedings of the 38<sup>th</sup> International Conference on Information Systems (ICIS)*, pp. 1–21, Seoul, South Korea, 2017.
- [Xie 02] W. Xie, K. C. Teh, and Z. F. Qin. “Speed and spin of 40mm table tennis ball and the effects on elite players”. In: *Proceedings of the 20<sup>th</sup> International Symposium on Biomechanics in Sports (ISBS)*, pp. 623–626, Caceres, Spain, 2002.
- [Yama 18] T. Yamashita and T. Kobayashi. “Smart ping pong racket by ultrathin piezoelectric strain sensor array”. In: *Proceedings of the Symposium on Design, Test, Integration & Packaging of MEMS and MOEMS (DTIP)*, pp. 1–3, Roma, Italy, 2018.
- [Zhan 06] B. Zhang, W. Dou, and L. Chen. “Ball hit detection in table tennis games based on audio analysis”. In: *Proceedings of the 18<sup>th</sup> International Conference on Pattern Recognition (ICPR)*, pp. 220–223, Hong Kong, China, 2006.
- [Zhan 10] Z. Zhang, D. Xu, and M. Tan. “Visual measurement and prediction of ball trajectory for table tennis robot”. *IEEE Transactions on Instrumentation and Measurement*, Vol. 59, No. 12, pp. 3195–3205, 2010.
- [Zhan 14] Y. Zhang, Y. Zhao, R. Xiong, Y. Wang, J. Wang, and J. Chu. “Spin observation and trajectory prediction of a ping-pong ball”. In: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4108–4114, Hong Kong, China, 2014.
- [Zhan 15] Y. Zhang, R. Xiong, Y. Zhao, and J. Wang. “Real-time spin estimation of ping-pong ball using its natural brand”. *IEEE Transactions on Instrumentation and Measurement*, Vol. 64, No. 8, pp. 2280–2290, 2015.

In modern sports, often digitalized sensor systems are applied to monitor amateur and professional athletes. Since latest advances in miniaturization of electronic components and data analysis techniques opened new possibilities in sensor design, tiny measurement devices can easily be integrated into any kind of sports equipment or body-worn accessories. These Wearables have become essential as important data source for state-of-the-art movement analysis and performance indication to support and assess athletes during training and competition.

This thesis presents methods and approaches to enable the analysis of wearable data in the field of table tennis. On the one hand, this sport is characterized by a wide range of various playing styles and technical opportunities, making it a rewarding area of research. On the other hand, digitalization in table tennis is still slowly progressing.

In order to benefit from the advantages of Wearables in table tennis, this thesis introduces a training and support system comprising of an instrumented racket, multiple analysis algorithms and a feedback application running on a mobile device.

