Faculteit Industriële Ingenieurswetenschappen

master in de industriële wetenschappen: elektronica-ICT

Masterthesis

Extracting features from rowing stroke accelerations to reduce the analysis effort of coaches

Dieter Verbruggen

Scriptie ingediend tot het behalen van de graad van master in de industriële wetenschappen: elektronica-ICT

PROMOTOR :

dr. Nikolaos TSIOGKAS

PROMOTOR :

ing. Luc COENEGRACHT

Gezamenlijke opleiding UHasselt en KU Leuven

Universiteit Hasselt | Campus Diepenbeek | Faculteit Industriële Ingenieurswetenschappen | Agoralaan Gebouw H - Gebouw B | BE 3590 Diepenbeek

Universiteit Hasselt | Campus Diepenbeek | Agoralaan Gebouw D | BE 3590 Diepenbeek
Universiteit Hasselt | Campus Hasselt | Martelarenlaan 42 | BE 3500 Hasselt

2022
2023

Faculteit Industriële Ingenieurswetenschappen

master in de industriële wetenschappen: elektronica-ICT

Masterthesis

Extracting features from rowing stroke accelerations to reduce the analysis effort of coaches

Dieter Verbruggen
Scriptie ingediend tot het behalen van de graad van master in de industriële wetenschappen: elektronica-ICT

PROMOTOR :
dr. Nikolaos TSIOGKAS

PROMOTOR :
ing. Luc COENEGRACHT

Preface

In this master's thesis, methods for analyzing acceleration waveforms of rowing strokes are developed. Strokes in the rowing sport is the method of propulsion using an oar connected to the rowing boat. The dynamics of applying power to this oar and the movement of the boat result in typical acceleration waveforms that can be analyzed to deduce the technique of a rower. Rowing is a difficult and technical sport. With the methods in this thesis, the efficiency of the learning process can be improved. This thesis is written in function of my master's degree in Electronics and ICT Engineering Technology of the joint study program of UHasselt and KU Leuven. This research project started in October 2022 and ended in June 2023.

The subject of this thesis was proposed by me because of my interest in rowing. I was a member of the UCLL rowing team during my bachelor's education. I did my bachelor's in applied sciences in electronics and ICT at the UCLL before taking part in the bridging program to get my master's degree. Over a period of 2 years, I learned to row and fell in love with the sport. It is both physically and technically challenging, while the physics and dynamics of the rowing boat are complex. When I learned to row, I started to research papers and read analyses of great national rowing teams. I also was a technical coach in the UCLL rowing team for the experienced boat (often referred to in rowing as the varsity boat) during my bridging year.

The inspiration for the proposal of this subject is the lack of competition between electronics manufacturers in the rowing sport. In every competitive rowing boat, the products of Nielsen-Kellerman (NK) are used. They have a monopoly on the rowing electronics market, and this results in the lack of modernization in applied technologies over the past decades. I also have first-hand experience of the tedious manual analysis process that is used today.

I would like to thank the UHasselt and KU Leuven to enable me to work on this research project. In particular prof. dr. Kris Aerts, he authorized this proposed subject and contacted colleagues to guide me through this journey. Next, I want to thank my promotors dr. Nikolaos Tsiogkas and ing. Luc Coenegracht. Nikolaos gave direct feedback with his vast amount of experience in the field of scientific research with his specialization in robotics and artificial intelligence. Luc guided me through the prototyping and data filtering with his years of experience as a lecturer on embedded Linux and microcontrollers. Without their help, this challenging research project was not possible.

Lastly, I want to thank my family who supported me throughout all my educations. Without their support I could not be here today, presenting you my master's thesis. Also, did they proofread my thesis multiple times and improving the quality of the text.

I hope you find the thesis interesting and enjoy reading it.

Dieter Verbruggen
Heist-op-den-Berg, 9th of June 2023

Table of Contents

List of tables

List of figures

Glossary

Catch

The beginning of a rowing stroke is when the blades enter the water.

Drive phase

The acceleration phase of a rowing stroke. Rowers use physiological power to propel the boat forward.

Finish

The end of the acceleration phase of a stroke. The blades are removed from the water.

Oarlock

A U-shaped plastic part used in rowing to connect the oar to the boat, allowing for smooth rotation and efficient propulsion.

Race pace

The high stroke rate at which rowers move when rowing a competitive race. This stroke rate is typically between 30 and 40 strokes a minute depending on the race plan.

Recovery Phase

The recovery phase is the movement the rowers make after the finish. They slide forward and prepare for the catch of the next phase.

Steady state

Steady state is the tempo the rowers train the most at for enhancing their technique. It is a relaxed stroke rate they can endure for a longer period to focus on control throughout the strokes. Typically, this zone ranges from 16 to 22 strokes a minute.

Stroke rate

The stroke rate is the standard in rowing for expressing the pace of the strokes. In some literature, it is referenced as SPM which means strokes per minute.

Abstract

Compared to mainstream sports, rowing lags concerning the adoption of electronic technologies and 'smart software'. By advancing the technologies used inside the rowing boat, crews can improve the efficiency of the training sessions with more real-time data and eliminate the overhead effort of coaches to analyze the captured data. In this study, a method for extracting features from acceleration waveforms is developed and validated. These features are necessary for using complex algorithms on rowing strokes to automate the analysis process. The features represent the rowing technique phases based on the use of the rower's body parts.

This thesis proposes a stroke detection algorithm to separate individual strokes. First, a Kalman filter is used on the waveform to remove noise. Then, the features are extracted from the strokes with timing relative to the stroke length. The features are validated with correlation matrices and outliers are removed from the dataset. Last, an automated analysis algorithm is developed and validated.

The algorithm can differentiate strokes with good or bad techniques with an accuracy of 97% in the acquired dataset. Three methods to identify technical errors are tested with promising results. However, to further develop these methods and algorithms, more participation in the training planning is necessary to record specific stroke rates and power efforts. Because of the lack of collaboration with the coach, there were no strokes recorded between steady state and race pace.

Abstract in Dutch

In vergelijking met andere sporttakken loopt de roeisport achter in het gebruik van elektronica en “slimme software”. Door modernisering van technologieën in een roeiboot kan de efficiëntie van trainingen verbeterd worden met snellere feedback door sterk gereduceerde analyse-tijd voor de coaches. In deze studie wordt een methode ontwikkeld en gevalideerd om ‘features’ af te leiden uit acceleratiecurves. De extractie van features is nodig voor het gebruik van complexe algoritmes om het analyseproces te automatiseren. Deze features geven de verschillende fases in een slag weer, op basis van de lichaamsdelen die een roeier gebruikt op een gegeven punt in de tijd.

In deze scriptie wordt een algoritme voorgesteld om slagen te detecteren en te scheiden. Eerst wordt een Kalman-filter toegepast op de ruwe data om de ruis te verwijderen. Daarna zijn de features geëxtraheerd in functie van de lengte van een slag. De validatie van de geëxtraheerde data gebeurt met correlatiematrices en statistische methodes. Ten slotte wordt een automatisch analyse-algoritme ontwikkeld en getest.

Het algoritme kan slagen, met goede of slechte techniek, onderscheiden met een nauwkeurigheid van 97% in de verkregen dataset. Drie methoden identificeren technische problemen met veelbelovende resultaten. Om deze methodes en algoritmes verder te ontwikkelen is er meer inspraak over de trainingen nodig voor de nodige data op te nemen. In dit onderzoek is er door de samenwerking met de trainer geen slagen tussen steady state en race pace opgenomen.

1 Introduction

Rowing as a sport is becoming more popular. Aside from competitions, is it a great way to do full-body exercise. Rowing is a combination of strength, technique, and endurance. Advancing the rowing technique is crucial for maximizing efficiency and minimizing the losses of the applied physiological power. Traditionally, the rowing technique is assessed during the training session with visual observations and after the training session by analyzing recorded data on accelerations and forces. This method is time-consuming and limits the potential gain in the technique of every training session. In recent years, there has been growing interest in using fast and accurate feedback systems to deliver more real-time data about the performance of the rowing boat.

In commercial devices, the incorporation of modern techniques has been stagnant for more than a decade. There is only one large manufacturer of rowing-specific electronics called Nielsen-Kellerman (NK). NK produces the stroke coach and speed coach; these products are used in almost every rowing boat in the world. Because of the lack of competition, there have not been many added features since the first generations launched in 1984.

Acceleration waveforms, obtained through wearables and recording devices attached to rowing equipment, provide valuable information about the rowing stroke. These waveforms contain small details that are indicators of potential problems. By analyzing these relevant details or features from the acceleration waveforms, it is possible to develop a systematic approach for identifying technical problems in the rowing technique.

The primary objective of this research project is to explore the use of feature extraction techniques on acceleration waveforms to identify and characterize technical problems in rowing strokes. The analyzed features are: peak accelerations, acceleration slopes, and duration of removing the blade from the water. The aim is to develop a comprehensive framework that can automatically detect and quantify deviations from an optimal rowing technique.

This research is motivated by the need for fast and reliable methods to assess the rowing technique with less effort, which can help coaches, athletes, and researchers in providing targeted feedback. Also, create a better understanding of the underlying biomechanical factors that contribute to technical problems. Furthermore, the outcomes of this study have the potential to enhance rowing performance, reduce the risk of injuries, and improve overall training efficiency.

To achieve these objectives, this research project will employ a mixed-methods approach, integrating data collection, signal processing, and machine learning techniques. By collecting a dataset of acceleration waveforms from experienced and novice rowers. A robust framework is defined for analyzing the raw training sessions into a dataset of all strokes. This framework contains noise filtering, stroke detection, and feature extraction. The developed framework is validated with visual representations such as scatter plots, also a machine learning classification algorithm is implemented to predict whether a stroke is good or bad. Using weights of the individual features in the classification model, new strokes are validated to identify the relatively worst-performing feature.

For creating a methodical implementation and a better understanding of the workflow the project is divided into 4 phases:

- **Data capturing:** Training sessions are recorded, and noise is filtered out.
- **Stroke detection:** Individual strokes are separated in the long training sessions.
- **Feature extraction:** Features are identified and extracted from all strokes to create a dataset.
- **Analysis models:** With machine learning and statistical methods, the worst-performing feature is identified for every stroke.

The implemented methods heavily rely on the results of the proceeding phase. Therefore, data recording is an individual section in the thesis (Chapter 4). Without the results of the recording setup and a visualization of the rowing data, the following methods cannot be explained. The other 3 phases are included under data analysis methods and results in Chapters 5 and 6.

It is important to acknowledge that this research has some limitations, the feature set used is small due to the complexity of the project and time limitations. The recorded rowers are students from the UCLL rowing team, who have limited on-the-water time. This means their training sessions are very valuable to them. They cannot change their planned training sessions for recording more desirable data.

Overall, this thesis aims to contribute to the field of rowing technique analysis by introducing an objective and data-driven approach through the feature extraction of acceleration waveforms. The findings of this research have the potential to reduce the effort of coaches to analyze recorded data and give more real-time feedback to rowers to correct their technique faster.

2 Theory of rowing

Rowing is a complex and physically demanding sport. However, the sport is gaining more traction with recreationists because of the low impact on ligaments and as a good full-body workout. In this chapter, more details and background information are given on the setup of a rowing boat and the biomechanical aspect.

2.1 Setup of a rowing boat

In Figure 1, a typical rowing boat setup is illustrated. The most important parts are the sliding seat, the oarlocks, and the foot stretcher. The rower is facing backwards compared to the movement of the boat. This is to be able to use the legs as efficiently as possible. When the rower stretches its legs, the seat will slide along with the rower. The foot stretcher is firmly connected to the hull of the boat with carbon fiber rods and a clamping system. The foot stretcher needs to be adjusted to the length of the rower's legs to correctly be positioned in comparison to the oarlocks which are mounted outboard on the riggers. This position to the oarlocks is important for the angle of the oars in relation to the max power opening of the rower's body. This position at maximum power needs to be with the oars perpendicular to the hull for maximum power transfer. When the oars and thus the blades are angled, some power will be lost due to the vectorization of the applied forces. When not perpendicular a small vector component will be directed to or away from the hull. This is one of the losses of the blade.

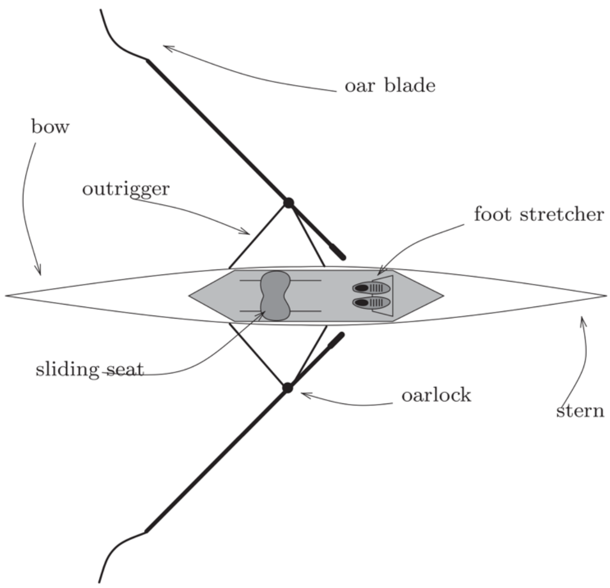

Figure 1: Top-down view of boat setup [1]

2.2 Biomechanics

A rowing stroke is divided into two parts: the recovery and the drive. In Figure 2 the transitions of these parts are visualized. During the recovery, the rower resets his position from the finish (A in Figure 2) of the stroke back to the catch (D in Figure 2). The finish is a resting moment where the blade is extracted from the water and is feathered to reduce wind resistance. The first phase of the recovery is sending away the hands with the legs and upper body fixed, this is represented in part B in Figure 2. The next step is leaning forward with the upper body followed by bending of the legs, parts C and D respectively. At this point, the rower is in the catch position and ready for the next stroke.

The stroke starts with the insertion of the blade into the water. The driving phase is the reverse order of the recovery phase. So, in part E of the figure, the legs are extended with the back muscles engaged. The second phase of the drive is the upper body, the rower uses its weight with extended arms to convert as much power to the oar, this is visualized in part F. The transition from F back to A is the last phase of the drive into the finish, pulling with the arms. The arms are not used to give additional acceleration to the boat but rather to keep the maximum velocity as long as possible before the drag of the water slows the boat down again. The balance of the boat is the most difficult and crucial during recovery. If the boat is not 'set' (or stable), the rowers will struggle to insert the blades synchronously at the catch. When the boat is not level, extra drag will be introduced due to the extra wetted area of the hull and the blades dragging across the water.

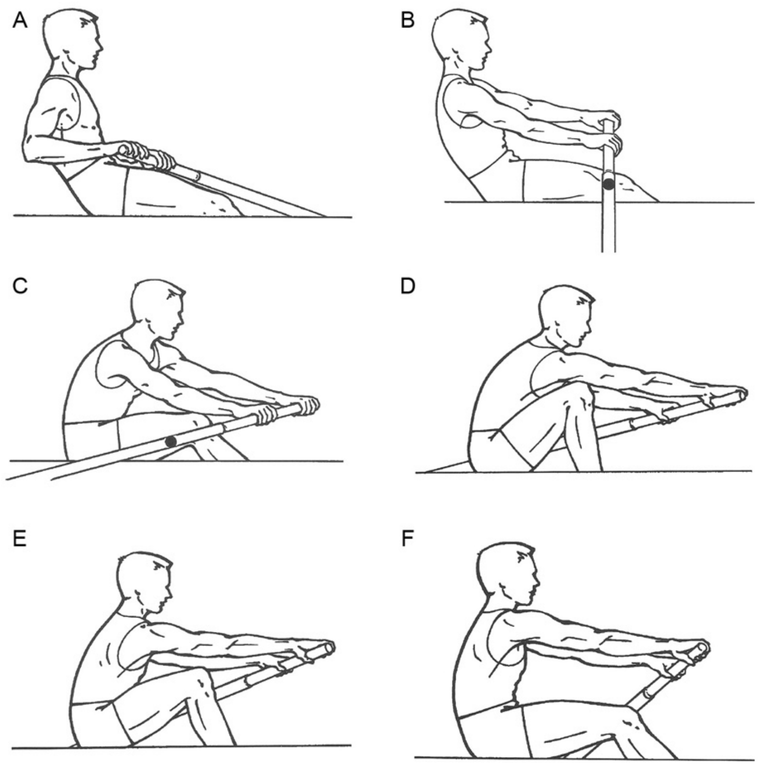

Figure 2: Body positions for the phases of a rowing stroke, finish A and end of drive F [2]

During the acceleration phase, there are microphases due to the dynamics of the rowers in relation to the boat. The rowing boat as a system is a combination of the rowing boat and the rowers. The dynamics during the drive are completely different than during recovery, due to the fixation of the blade in the water when power is applied. Not only the boat is propelled forward but also the rowers themselves. Collectively, the rowers are always the biggest mass in the boat, this is important during the recovery. When the rowers are moving in the direction of the boat, and they reset their position back for the catch. The rowers pull the boat underneath themselves giving extra acceleration to the boat and not the system. The microphases that result from the separation of the masses into rower and boat are the following: initial rowers' acceleration, initial boat acceleration, main rowers' acceleration, and main boat acceleration [3], [4]. In the paper by Kleshnev, the microphases are defined by a temporal analysis and isolating system accelerations on rowing strokes [4]. In this thesis, only boat acceleration waveforms are used.

Lastly, the transition in dynamics between steady-state and race pace. At steady state, the distribution of the drive is a third of the total stroke time. Resulting in a recovery phase that is twice as long as the drive phase. This is important to find rest in the recovery, to feel the balance in the boat. When accelerating above 22 strokes per minute, the faster stroke is a result of a more explosive drive. Hereby shortening the absolute drive time and keeping the recovery the same. During the recovery, the boat gets an extra acceleration as mentioned before. By reducing this acceleration, the drag of the water is limited. At around 24-26 strokes per minute also the recovery must be shortened to reduce the time of a stroke. At race pace, the distribution between drive and recovery is almost 50-50. This is only possible because the time between the drive phases is short enough that the higher drag of a fast recovery is outweighed by the extra acceleration.

3 Related work

In this chapter, the theories of related work are given and explained. Firstly, the comparison between national-level rowing crews and the best of the best (Olympic crews) for identifying differences in acceleration waveforms. Lastly, the change in training feedback cycles is given from novice to more advanced rowers.

3.1 Comparison rowing technique

By analyzing trends in changes of the accelerometer waveforms on rowers in the whole range of abilities, the level of technique can be identified. In Figure 3 a comparison of waveforms is given between Olympic and national-level rowing crews. The most noticeable and important differences between the two are: the minimum peak at the catch is deeper and sharper for the Olympians; the slope between negative and positive peaks is steeper for the Olympians; the initial boat acceleration is not present in the national crew [3].

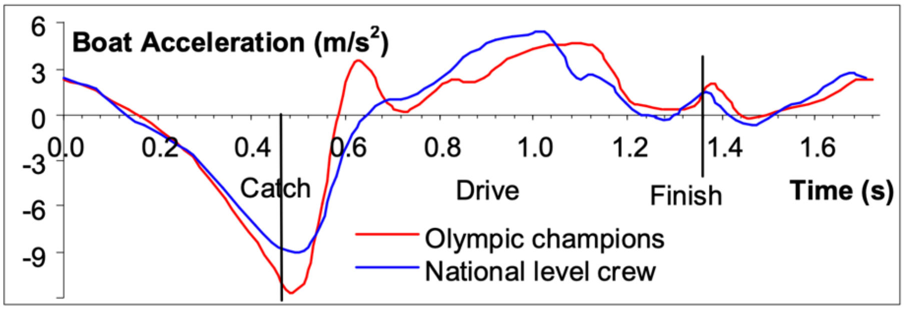

Figure 3: Comparing inertia waveforms of Olympic and national-level rowing crews [3]

3.2 Rowing training feedback

In Figure 4, two different feedback cycles are given for training athletes at high levels. Method A is the cycle that is currently being used with all national and Olympic rowers. After the training session, the coach needs time to evaluate the recorded data and it takes at least 90 minutes before the rowers get feedback on their last training session [5]. On club or college-level rowing, this delay between training sessions and the feedback is even larger. Most times is there only one feedback session in the evening from all the training sessions on that day.

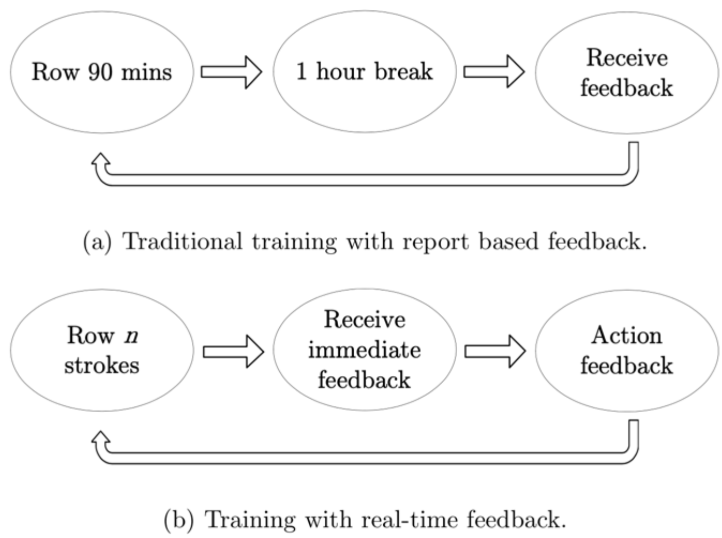

Figure 4: comparing the traditional feedback cycle to the modern approach [5]

For experienced rowers, this method works but is not as efficient as method B, which Harfield P. suggested in his Ph.D. study. With method B the rowers get more real-time feedback and specific information derived from the recorded waveforms [5]. In his conclusion, he added testimonies of elite rowers who learned so much about the technique due to the fast analysis feedback cycle [5].

For training indoors, there are a lot of analysis tools developed for giving automated feedback, ranging from vision tracking to force analysis. But there is a large problem with these rowing simulators. Besides feedback from a coach, rowers also receive auditory feedback from the sound of the rowing boat. Experienced rowers can judge the sharpness of the catch and finish by the sound of the oars. This is a large problem in the use of simulators. The feeling of handling the oar can be simulated but the auditory feedback cannot [6].

4 Data recording

In this chapter, the methods and results of the data recording are given. This phase of the project is separate as explained in the introduction. Without recorded data and a fixed framework to get this data consistently, the other phases cannot be developed. First, the methods of the recording prototype are explained. Then secondly, the results of the selected methods and recorded rowing data itself are given.

4.1 Methods and materials

For analyzing rowing strokes to detect errors in technique, first acceleration waveforms must be recorded. In this section, the hardware setup, methods of noise filtering, and development of the capturing software are explained. Also, some test scenarios are thought out for validating individual filtering techniques without using the valuable training sessions of the rowing team. All the acceleration waveforms talked about in this thesis are in the moving direction of the rowing boat. Depending on the mounting of the sensor in relation to the boat's forward direction the x or y-axis is used. Only for the blade movement detection in Chapter 5.2.2, the gravitational axis is used for detecting changes in dynamic noise.

4.1.1 Capture device setup

For the data-capturing prototype, the following components were used: Raspberry Pi 3 running Linux, an MPU9250 motion sensor, and a portable USB power bank. These components were selected because of availability and prior experiences. They are mounted firmly into a watertight box to keep the electronics safe and prevent jitter in the data. The MPU9250 accelerometer module supports both SPI[1] and I^2C[2] communication protocols. However, library support for SPI interfaces is lacking so I^2C is used in this case. The communication speed of the I^2C interface is manually set to the maximum of 400kHz in the Raspberry Pi OS which means a sample rate of 800 Hz can be achieved across the 3-axes. Python is used as the programming language on the Raspberry Pi. This decision was made because of the wide range of libraries available and the rapid prototyping aspect of this high-level programming language. The integration of tools for the Raspberry Pi itself is a nice welcome. For controlling the Raspberry Pi, the Wi-Fi functionality is used as a hotspot. This enables the connection of a smartphone or laptop without the need for an active network nearby.

[1] Serial peripheral interface: Digital communication protocols for interfacing sensors and actuators.
[2] Inter Integrated Circuit: Communication bus for digital communication between multiple slave devices and a master.

4.1.2 Noise filtering

All accelerometers are prone to static and dynamic noise, this noise will negatively impact the stroke detection and feature extraction applied in the following chapters. The MPU9250 accelerometer module has built-in digital low-pass filtering with a cutoff frequency programmable from 5 to 260Hz. This built-in filter is a good way to implement noise reduction for fast prototyping and prefiltering. In most cases, there is however a need for more intensive algorithm-based filters like Kalman or complementary filters [7]. In this application, a Kalman filter is implemented from the open-source library pykalman [8].

A Kalman filter is a two-step filter, there is a predicting step followed by the update step. In the prediction step, the filter will predict the next sample value based on the last state and the knowledge of the system. The predicted value and the measurement are then brought together with a weighted average in the update step. The weights are assigned based on the confidence of the measurement method. In the case of the accelerometer data, due to physics, the data should not erratically change. The inertia of an object cannot change so quickly. This means that there will be more confidence in the predicted measurement. The actual data point after filtering will be the relative difference between the predicted and measured values. If The difference between the two values is still significant, the filtered data will have noise still present.

Kalman is mostly used in sensor fusion. With sensor fusion, the goal is to combine multiple inaccurate sensors to use their strengths to compensate for the downsides of another sensor. For this application Kalman is only used for the noise filtering of the accelerations, there is only a linear relation. This is implemented in pykalman by setting the transition and observation matrix to 1. The initial state mean is set to the first measurement to start the filtering with the first value loaded. The observation covariance is set to 5 because the accelerometer data is very noisy. This observation covariance sets the amount of unpredictability of the measurements. Lastly, the transition covariance sets the unpredictability of the underlying system. For this case, it is set to 0.005 to give a smoother acceleration waveform.

Another filtering method that is considered for the development of the prototype is a moving average filter. This filter calculates the average over all the samples that are within a window. With every new data point, the window is advanced one step. This results in a digital low-pass filtering method where erratic changes are averaged out and significantly reduced in size. The parameter for filtering is the size of the window, a larger window will result in better rejection of sudden erratic changes in the waveform. These filtering methods are tested with different parameters and combinations in different scenarios which is explained in the next paragraph.

4.1.3 Testing scenarios

To record as much data of the limited training sessions the rowing team has, the recording setup must be validated with artificial scenarios to represent certain aspects of rowing accelerations. For this data-capturing device, three important testing scenarios are defined for finding a good balance between accuracy and noise rejection. The scenarios are:

- **No movement:** To get an idea of what the static noise of an accelerometer looks like.
- **Impulses:** At catch and finish with sudden vertical movements of the oars, the boat gets loads of vibrations which distort the waveforms. Impulses are given by tapping the accelerometer when laid down.
- **Pendulum swings:** This gives a sinewave-like acceleration curve similar to boat movements and is used to see if a filtering method deforms the waveform.

4.1.4 Capturing software

For logging purposes, there is a need for a framework that reads the 3 accelerometer axes from the module. For the offline analysis of the waveforms, samples need to be logged into a file with their corresponding timing. This way the waveform can be reconstructed after the fact. The interfacing of the MPU9250 is done with the open-source imusensor library [9]. IMU is a different term for the accelerometer and stands for inertial measurement unit. The library directly uses SMBus [10] for interfacing over the I^2C protocol. The data is stored along with epoch[3] time strings in a comma-separated value file (CSV) using the built-in file functions of Python.

[3] Time in seconds counting since January the first of 1970.

4.2 Results and discussion

In this chapter, the results of the data-capturing process are presented with also the results of the sub-components. First, the filtering techniques are validated, and a filtering technique is selected for the implementation. At last, the recorded data from a rowing training is given with the final capturing prototype.

4.2.1 Filtering results on testing scenarios

In this chapter, the results of the individual filters are given on the three different testing scenarios. Because the digital low-pass filter is applied in the sensor module itself, the unfiltered data and low-pass filtered data are not originally the same. Rather, they are recorded at a different time. However, the test is still relevant, and conclusions can still be drawn based on this comparison. In Figure 5, the results of the filters on data of the still scenario are visualized. The low-pass filtering does not perform well to reject static noise as expected. Only high-frequency (above 5 Hz) noise is reduced which introduces artifacts to the waveforms. In this case, the waveform is swinging up and down because the low-frequency oscillations are still present. The same goes for the moving average filter. Just before the 400th sample is a period where the waveform is erratic due to the large changes in the original noise. This fast-changing noise is significantly reduced but still noticeably present. In this specific scenario, the Kalman filter also struggles to reject the noise. This is due to the adaptive specification of a Kalman filter. In this case, the filter predicts the next value based on the knowledge about the noise in the system.

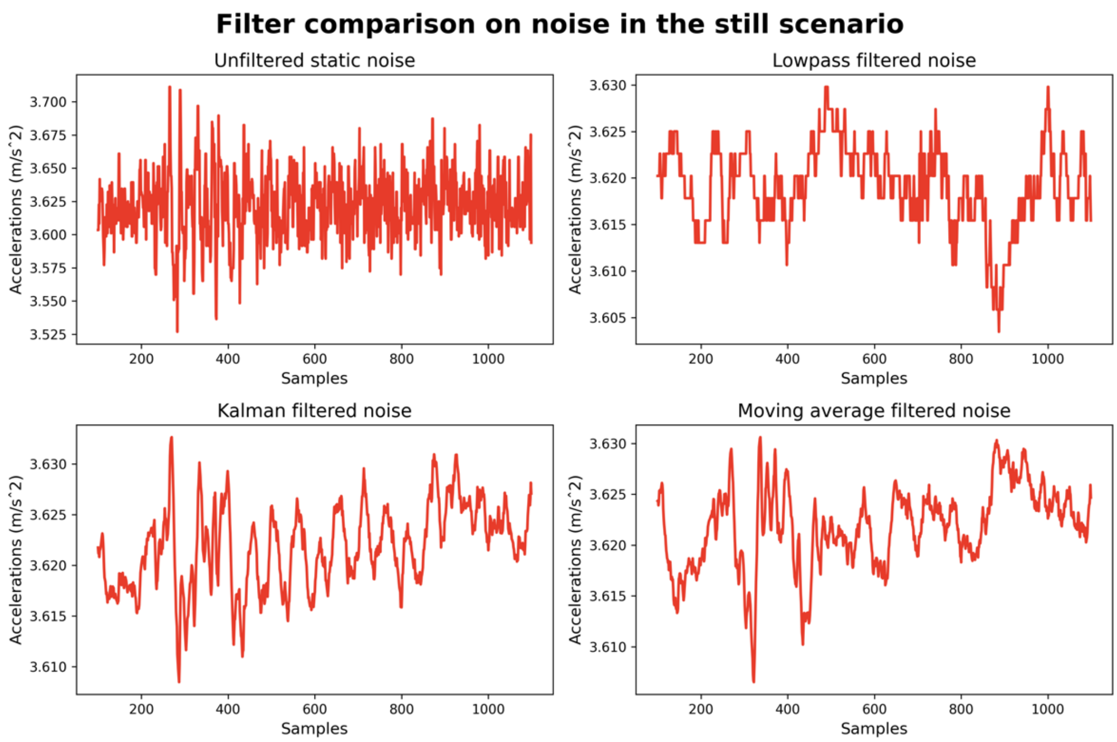

Figure 5: Filtering comparison on the still scenario

The next scenario is the vibrations applied to the sensor for simulating the shocks in the boat at the catch and finish. The results are visualized in Figure 6. Firstly, the original data is an example of the mathematical 'sinc' function with limitations in the beginning due to the cut-off frequency with the sample rate.

Secondly the low-pass filter, the first observation is the reduction in acceleration swing. The original data has a swing typically around 14 m/s^2 while the low-pass filter reduced this to only 3 m/s^2. However, the extreme dampening of the higher frequencies is not favorable for recovering faster from an impulse. Next the Kalman filter, the acceleration swing is significantly reduced to only 2,5 m/s2. On top of that, the waveform recovers a lot better after an impulse compared to low-pass filtering. However, there is a large negative peak present most likely due to the overcompensation of the adaptive filtering. When a Kalman filter is adjusted on the static noise and gets an unexpected impulse, the filter will try to adjust to the new dynamics of the system. This is a reasonable explanation when looking at the second impulse in the graph. The second impulse is marginally better filtered out compared to the first. Lastly, the moving average filter. The moving average filter reduced the acceleration swing a little less than the Kalman filter. The recovery of an impulse is similar but slightly better than the other filters.

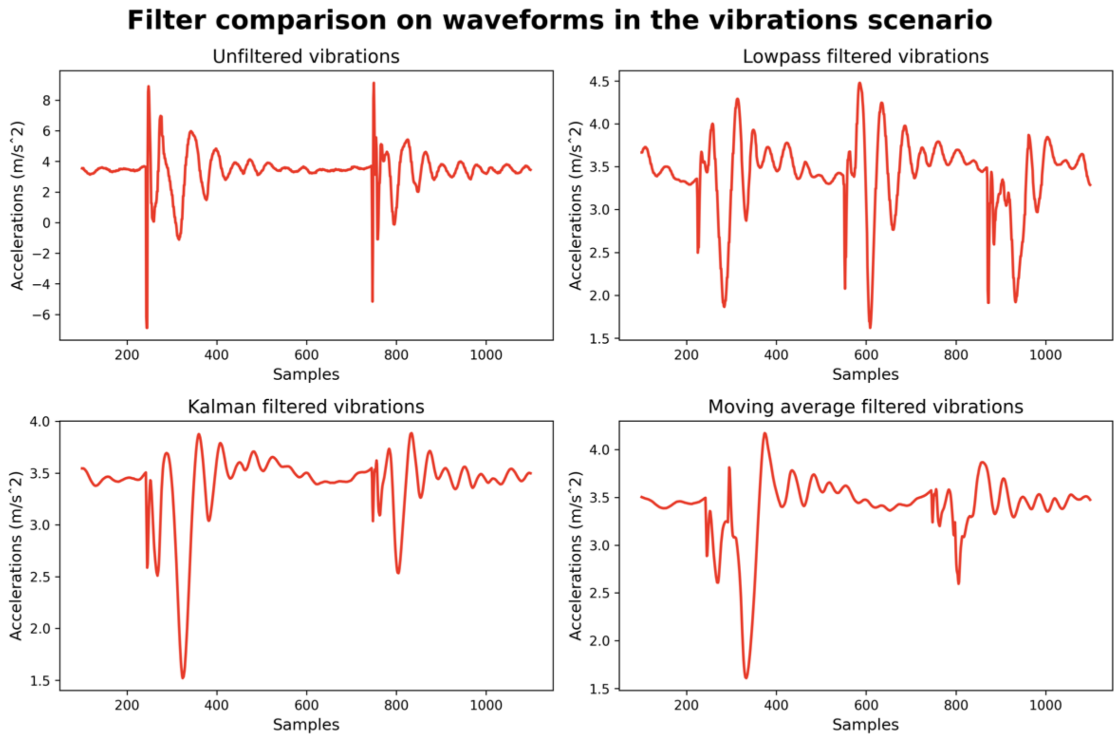

Figure 6: Filter comparison on the vibration scenario

The last testing scenario is the pendulum swings to recreate the movement of a rowing boat during the strokes. The filtering results of this scenario are given in Figure 7. The first comparison; of the low-pass filter; is difficult as the original data before filtering is not available to compare the waveforms. However, it can be concluded that in this scenario the result is also negatively impacted by the lack of higher frequencies in the spectrum. Some parts of the waveform are leveled out and there is consistently a double peak. The Kalman and moving average filters perform similarly. There is slightly more noise present in the Kalman-filtered data, but the shape of the waveform is more accurate. The peaks of the moving average waveform are blunt and wider than the original which is an important factor for data analysis on acceleration waveforms. On rowing accelerations, the details of the peaks are very important, certainly for extracting features.

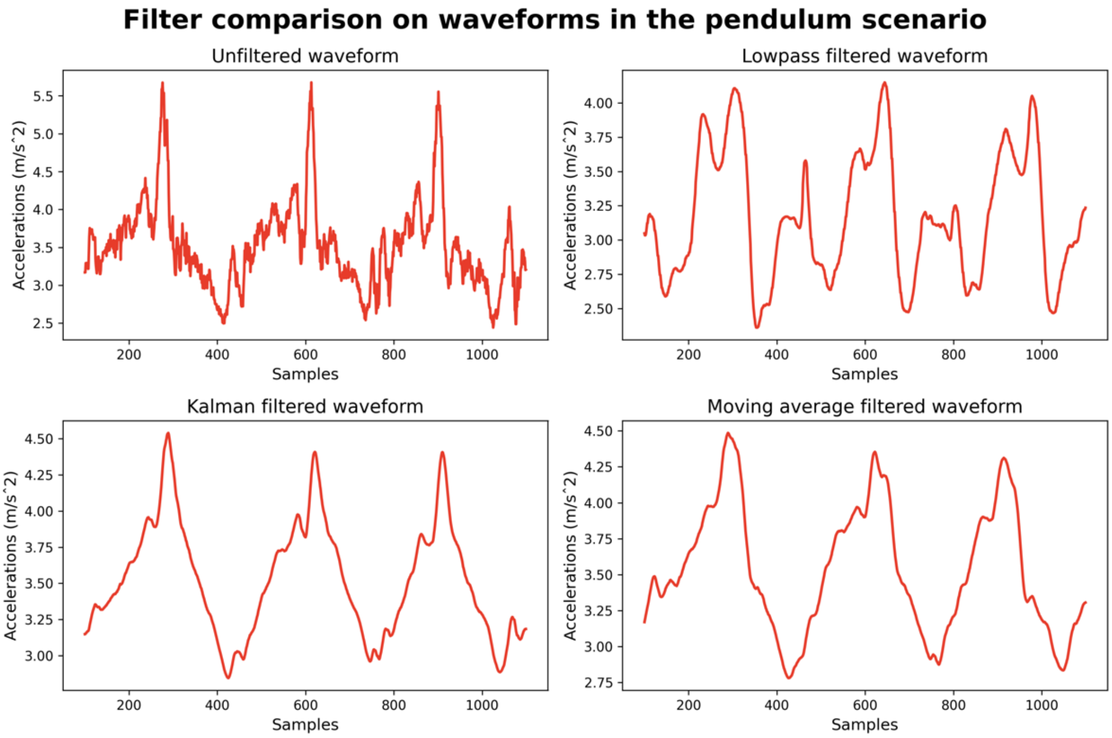

Figure 7: Filter comparison on the pendulum scenario

An important factor of the moving average filter that is not taken into consideration yet is the unreliable method of filtering. The form of the waveform is altered and smeared out due to averaging the window. This also means that the timing of the resulting zero crossings or peaks is not accurate. This is an even larger problem when the amount of time delay added to the zero crossings for example is not constant but is dependent on the slope of the waveform. When the slope is steeper, the zero crossing will be detected relatively earlier compared to a slow slope.

Following the results of the testing scenarios, a Kalman filter is the best choice for the application of removing noise from acceleration waveforms on rowing strokes. It has a good balance of noise removal, scale, and form of waveforms.

4.2.2 Rowing data results

After selecting the filtering method for the first recording prototype, the first training session is recorded. The result of 2 random rowing strokes with and without Kalman filtering is given in Figure 8. On the left is the noisy accelerometer data without any filtering displayed. The noise in this case is very erratic and a good example of how well the Kalman filter works for this application. The filtered data on the right is clean while the details are present, and the scaling is not altered. This means that the peaks still have the same maximum values, which is necessary for extracting features.

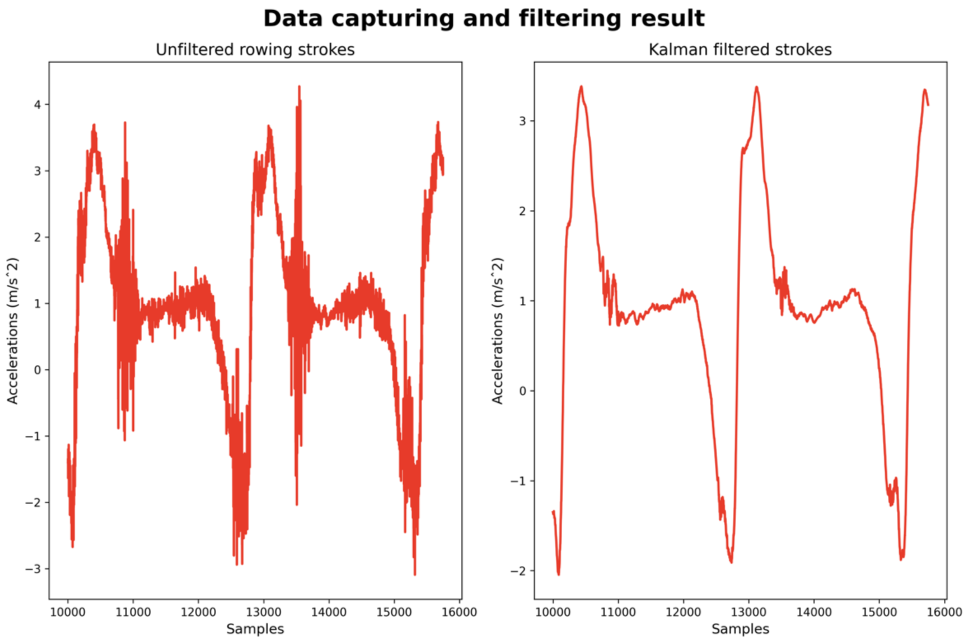

Figure 8: Recorded data comparison, with and without Kalman applied

Following the results of recording acceleration waveforms inside a rowing boat, Kalman is the best filtering method for this use case. The recording setup and software work perfectly, now the data can be recorded and filtered with a consistent method for the further analysis of the next chapters. The next step of data recording is gathering as much data as possible on the rowing training sessions of novice and more experienced rowers to create a dataset.

5 Data analysis methods

In this chapter, the methods of analyzing the captured data are given, this includes the methods of phase 2 to phase 4 of the project. Firstly, strokes are detected and separated from the raw waveform. Secondly, the features to extract are identified and then extracted. With the extraction, a dataset of the features of all strokes is created. Lastly, the dataset is preprocessed for the machine learning implementation and the worst-performing feature methods are implemented.

5.1 Stroke detection

Stroke detection is a very important phase of the project (phase 2). The results of the feature extraction and analysis models are directly dependent on the accuracy and efficiency of the stroke detection algorithm. Therefore, a good and efficient method of detecting strokes and separating them from training sessions is needed. Firstly, the requirements of the algorithm are analyzed. Then secondly, a method is given for calculating zero crossings and filtering the correct ones with peak detection algorithms. The stroke detection and the steps that follow are done completely offline in Jupyter Notebooks[4].

5.1.1 Problem statement

The stroke detection needs to be performed consistently because this detection will impact the timing of extracted features. If the detection is based only on the maximum of positive peaks the timing can vary every stroke. In theory, the negative peak could be suited for peak detection. Because in an experienced boat the most negative peak is always right after the catch, before the power is applied in the drive. However, the impulse of the catch gives noise even after Kalman filtering. The problem is even worse in boats with less experienced rowers. The smallest deviation in synchrony of the rowers will result in erratic noise and peaks with inconsistent timing. With this method, the timing could variate 13% of the full stroke on steady state strokes. This is derived from the two negative peaks in Figure 9, these peaks have both the potential to be the detected minimum. It is statistically also preferred that the stroke detection is as accurate as possible so that when features are extracted it can be concluded that outliers in the dataset are irregularities of the boat and not from the detection algorithm.

[4] Jupyter Notebooks is an open-source web application for creating interactive documents containing code, visualizations, and text, commonly used for data engineering tasks.

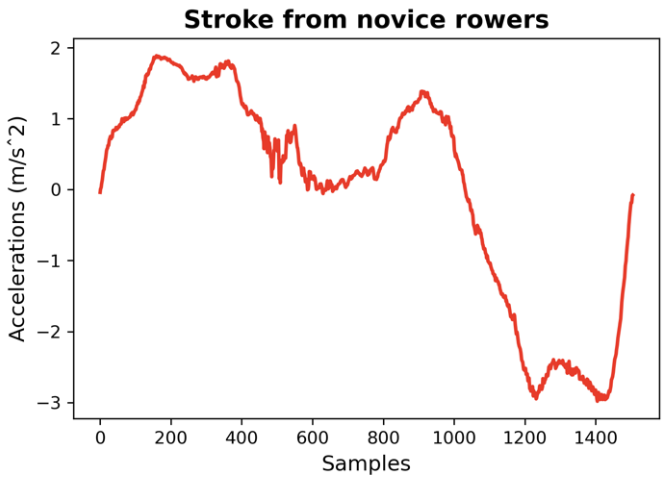

Figure 9: Stroke of novice rowers with 2 potential min peaks

The implementation should also easily be implemented to run real-time with the data recording so that in future projects all these methods can be built into the rowing device. This means that computationally expensive machine learning or deep learning cannot be applied to adapt to the setup of the boat. The proposed methods should work for every training session recorded with the prototype in the boat. When implementing these methods in an embedded device the algorithms and the whole analysis process will be written in a low-level programming language like C or C++ to minimalize the overhead and increase the performance.

5.1.2 Implementation

When analyzing the recorded data after filtering, it is apparent that the only time of the stroke there is clean data is in the transition between the negative and positive peak. Even for the most inexperienced rowers, the drive phase has the least chance of fatal errors which could prevent the algorithm to work. The zero crossings between the two peaks have to be selected for the most accurate stroke detection.

The method in this case is calculating all the zero crossings of the waveform data and then filtering out the zero crossings in the drive phase from minimum to maximum peaks. With NumPy [11] the zero crossings can easily be calculated with the difference in the sign bit function. For filtering out the correct crossings two peak detection algorithms are needed, one for the negative and one for positive peaks. The requirement for peak detection is that it needs to work on all rowing boats with different levels of technique.

The solution is to dynamically calculate the threshold for deciding if a peak is the largest of the stroke, a window function is used for this. The window is 2000 samples in size and keeps track of the (largest) peaks and the relative samples they were recorded at. If the peak is outside the window the data is thrown away, and the second-largest peak of the last frame became the largest. If the boat is decelerating by putting down less power in the drive phase this window function will make sure that the threshold is dynamically lowered to still detect only the strokes. For accelerating the algorithms just updates the new maximum peak with every stroke. In Figure 10 there is a secondary relative maximum between 1000th and 1250th sample in the stroke due to the recovery, the boat is being pulled forward by the rowers. This peak is rejected because it is lower than the dynamic threshold, if in some cases this peak is still registered the algorithm will still work perfectly for filtering the zero crossings. This windowed peak detection is implemented for the negative peaks in the same way.

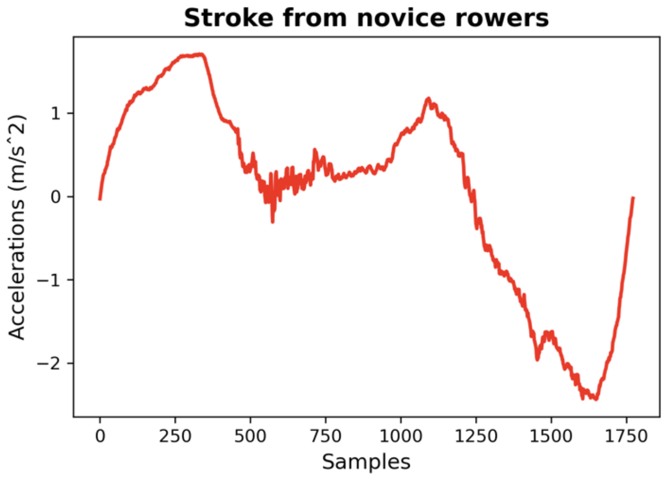

Figure 10: Recorded stroke from novice rowers

In the last step, the correct zero crossings are filtered out by accepting the crossings between a negative peak and a positive peak. When the algorithm goes through all zeros it remembers the zeros in a list with the sample number. After the maximum peak is detected, the temporary zero is accepted and the start of a stroke is detected. The zero crossings are also counted, if there is more than one zero detected in the filter window everything is rejected because this means something went wrong with the peak detection algorithm.

The stroke detection during the entire training session is validated by plotting the stroke rate of the training session. This is done by finding the time delta between two detected zero crossings and calculating the strokes per minute. If the change in stroke rate is too large from one stroke to the next, something went wrong in the detection algorithm. There is however not a threshold defined to automatically dismiss fast changes in stroke rates. Experienced rowers can accelerate very fast from a dead stop to race pace in 2 or 3 strokes. With stroke rate plots, it takes a few seconds to validate if the stroke rates are consistent with the training that was recorded.

5.2 Feature extraction

After the strokes are successfully detected, they are separated from each other for creating a dataset of all recorded strokes. The individual strokes of all training sessions are serialized into a single dataset of waveforms. The feature extraction methods are applied to this entire dataset. Feature extraction (phase 3) itself is necessary for providing details of a waveform in a compact and easily comprehensive way. The dataset of the extracted features of all strokes is used in the last phase of the project to detect inconsistencies in a rowing stroke.

First, the features to extract need to be identified with the main question, "Which key moments and occurrences are important for validating the technique? ". In the second step, the methods of extracting these features are developed. A new dataset is created with only the extracted features of every stroke and some extra information for the analysis models. And lastly, the feature extraction methods are validated by reviewing the generated data in comparison to the theory of trends in increasing rowing technique. The manual validations are done with a correlation analysis and scatter plots.

5.2.1 Identifying relevant features

In chapter 2.2 about the rowing technique theory, the microphases of a stroke and how they evolve with better technique are explained. These microphases are the basis for selecting the features. In the newsletter by V. Kleshnev, a clear overview of these important key moments is given as to how they progress [3]. During the selection process, the difficulty of the implementation is also considered. The following features were selected, and their explanation is given.

- **Maximum peak:** Indication of the power and effectiveness of the drive phase.
- **Minimum peak:** This represents the "sharpness" of the catch.
- **Blade removal:** Gives information about the finish of the stroke.
- **Drive slope:** The steepness of the transition between catch and maximum peak.
- **Stroke rate:** The amount of time a stroke takes represented in strokes per minute.

For the first three features, the timing is also recorded in relation to the stroke duration. There are still a lot of other features which are interesting for evaluating strokes but due to time limitations, the feature set is kept limited.

5.2.2 Extraction implementation

The maximum and minimum peaks are detected with a simplified version of the detection algorithm in Chapter 5.1.2. For every value it checks if the current acceleration is higher/lower than the maximum/minimum from that stroke. If this check is successful the peak values and time sample are updated, otherwise nothing happens. After all the stroke values were checked, the results are then the maximum and minimum values of that stroke. The timing is converted to relative timing with the total amount of samples in that stroke.

The blade removal is detected on a different accelerometer axis, the gravitational axis. In Figure 11 the acceleration waveform in the gravitational axis is plotted alongside the Kalman-filtered acceleration waveform in the moving direction. At the catch and finish of every stroke, the blades are inserted/removed from the water and a significant impulse is observed on the raw data.

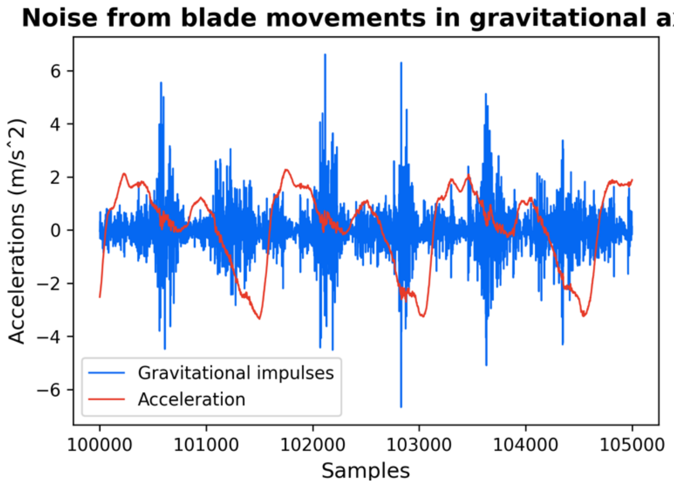

Figure 11: High-pass filtered gravitational impulses

The gravitational axis is first filtered with a Butterworth high pass filter with a cutoff frequency of 50 Hz to remove the gravitation and slow fluctuations in the signal. A Butterworth filter is a multi-order filter designed to have an almost perfectly flat frequency response in the pass band. In this case, a digital variant is used with the SciPy python package [12]. This filter solely aims to remove the gravitational bias and small inconsistencies from the data. The actual detection of the finish is done with a windowed RMS filter. The RMS filter calculates the root mean square (RMS) for the given window of data samples. This method is commonly used for vibration analysis and power measurements. In this case, it will calculate the RMS of the noise of the impulses, and if the calculated value is larger than a threshold the finish is detected. For this application, the window is 100 samples in length. The duration and timing of the detected blade removal are recorded in relation to the full stroke length. The duration represents how synchronous the finish is between rowers. The timing is used for comparing strokes with the same stroke rate against each other.

Then the last actual feature is the drive slope. This slope is the transition between the negative and positive peaks and a positive correlation exists between the steepness of this slope and the level of technique of the crew [3]. Because the strokes are detected on the zero crossing after the negative peak of the slope, the stroke starts with the positive part of the acceleration slope. The negative part is included in the last stroke. This means that the slope calculation is done only on the positive part of the acceleration slope, given in Figure 12. In theory the negative part and positive part of the slope should be equal. This presumption could be used to compare the slopes of two different drive phases in one single stroke due to the stroke detection on the zero crossings. In this thesis this last option was not implemented because the labeling of the strokes becomes harder. The calculation of one slope is done by using the elevation divided over the time delta of the samples.

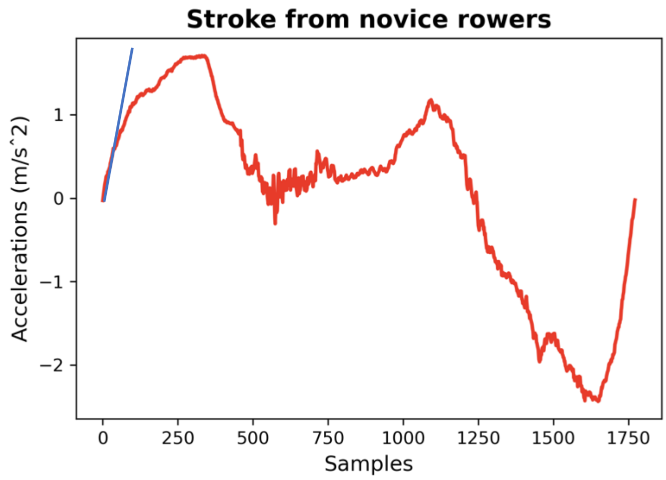

Figure 12: Stroke from the novice boat for visualizing the drive slope

With only one calculation there is a large chance that the measurement is slightly off for that sample due to small inconsistencies in measurements, this is why the slope is calculated at five points in that initial acceleration slope. These five measurements are then averaged out for calculating the feature. This minimizes small errors; the downside is that if the slope is rounded, the slope will level off slightly and is not representing the drive around the zero crossing.

Then the stroke rate is also calculated to add to the feature dataset. The stroke rate is not a traditional feature like the before mentioned features. In the sense that it is not a key moment from the stroke. However, it is an important characteristic of the stroke for giving more context to the algorithms. The dynamics of a rowing boat change with increasing stroke rates. Most on-the-water training sessions are steady state while the races are on race pace, so it is very important that this analysis works for both or can differentiate the two. These features are then validated using two methods in the next section.

5.2.3 Validation of features

The validation of extracted features is a difficult topic. Some features are more important for a classification machine learning (ML) model than others, this however does not mean that the features of less importance are not necessary. In chapter 5.3.3, the method for classifying these strokes is given with some explanation about the importance of the model. In this chapter, the validation is done with visual and statistical analysis methods. The first is creating a correlation matrix of the features. The second method is analyzing scatter plots, if the groups of data can be visually divided then a machine learning model or similar methods certainly can. The data is divided into groups of the novice and the more experienced boat, but also into race pace and steady state.

A correlation matrix is used in data science for verifying the importance of a feature based on the correlation between two features. A feature with a low absolute correlation brings all new information to the dataset. A feature with a high correlation with another feature means these are somewhat proportionate. Having a low correlation does not mean that the feature brings useful data to the machine-learning model. The importance of a feature will later be validated more with machine learning weights after training in Chapter 5.3.3.

5.3 Analysis model for detecting the worst features

The analysis of the features extracted from individual strokes is the last phase of the project. By identifying the worst-performing feature for a given stroke, the errors in technique can be identified and a rowing coach can give real-time feedback.

In this chapter, the feature dataset is used for evaluating the concept of this project. Firstly, the raw features after extraction are preprocessed to remove outliers to help the classification algorithm. Secondly, the features are normalized to convert the values to the same scale. Thirdly, with classification machine learning models, a stroke prediction is done. When a newly recorded stroke is presented to the model, can it predict if it originated from the novice of the experienced boat? Lastly, methods for analyzing the individual features are given to compare the mathematical distance to a perfect stroke. With this distance, feedback can be generated for the rowing technique based on the largest distance. This is called the worst-performing feature of a specific stroke. By improving the technique which causes this worst feature, the technique of the rowers will improve the most. The high-level functions used in this part of the research project are implemented with the Sci-kit Learn (SKLearn) library [13].

5.3.1 Preprocessing dataset

For the best performance of the model to predict if a stroke originated from a novice or experienced boat, the model needs good consistent data. This means that the outliers of the individual features need to be removed, these outliers have no statistical significance to the model. The cause of these outliers is due to major mistakes in rowing techniques like “catching a crab” or a problem with the detection of strokes. Knowing the cause of the problems is difficult because there is no extra data available like video footage to support these presumptions.

To identify these outliers, the features are plotted in box plots. Both for uncategorized and categorized features. The categories used are the boat type (novice or experienced) and stroke rate (steady state and race pace). The data points outside of the box plots are outliers and are removed from the dataset. The division into categories is important for the removal of outliers if the experienced boat had one stroke like the novice rowers this should not affect the reference stroke model. The presumption of the model is that all strokes of the experienced boat are good, this is a necessary presumption for the classification task and the identification of technical problems. By using boxplots to visualize the features, the timing and value of a feature can be analyzed separately. If the timing of the maximum peak is good but the peak itself is not in the range for a good stroke, the stroke will negatively impact the model. This cannot be properly identified in the scatterplots given in Chapter 6.2 of the extraction features.

The boxplots themselves are only for identifying the limits to remove the outliers and do not give more information about the effectiveness. In Chapter 5.3.1 of the results, one example of this method to remove outliers is given as an example using the 'bladeRemove' feature. The outlier removal is also visualized using the scatter plots from Chapter 6.2.3 but with the 4 classes separate to compare the raw dataset to the resulting dataset with the outliers removed.

5.3.2 Feature normalization

The raw features after preprocessing have the scale of the actual parts of the waveform they represent. For example, the maximum peak has the values of the peaks from the original stroke. When using a wide variety of value ranges the learning algorithm can have difficulties calculating the weights for all the features. This results in models that don't use all the features optimally or not at all. When the features are normalized the resulting weights are a direct representation of the importance of the feature in the model. The scaling is done with the "standardScaler" function of SKLearn.

5.3.3 Stroke classification model

For evaluating the dataset, three classification algorithms are implemented to predict if a stroke originated from a novice or experienced boat. The first model is logistic regression, with logistic regression the weights are calculated for the specific features. This means that these weights can be used in the next step for incorporating the importance of the feature into the equation. The second model is a neural network, here every node of the network has a trained weight. For all layers in the network except the input layer, this weight is not representing a single feature but the combinations of all features. The third model is a support vector machine (SVM). With SVM the weights are trained for maximizing the margin on the support vectors. Support vectors are representing a subset of the training samples close to the decision boundary. This means that the weights are not directly applied to the features. For the next chapter, the weights of logistic regression are used in the identification of the worst feature.

The logistic regression model in this case is a supervised classification model for predicting if a stroke originated from a novice or experienced boat. Before the training, the dataset is split up into a train-test split of 80-20%. The test set is later used for validating if the model generalizes well to the dataset. When the results on the training set are significantly better than on the test set, the model is overfitting on the training set. This means that the model only works well on the training set, and with the introduction of new data, the model fails to classify the strokes.

For validating the model, the f1-score is used. F1-score is the combination of precision and recall, two important terms in machine learning. Precision is the ratio of true positive predictions to the total positive predictions of the dataset. A high precision score is obtained when the model predicts a small number of false positives. On the other hand, recall is the ratio of positive predictions to the total true positive records in the dataset. A high recall score is obtained when the model predicts a low number of false negatives. The f1-score is the harmonic mean of the recall and precision. Even with very unbalanced datasets, this score is a good representation of the model's performance. The f1-score is calculated on the test set with the three models and the results are given in Chapter 6.3.2.

5.3.4 Rowing technique problem identification

To identify the faults in the rowing technique, it is necessary to select a feature that is performing the least compared to a good stroke. This is done using the weights of the trained model with normalized features. The weights in this case are a direct representation of how important every feature is to the prediction. A statistical calculation is done with the weights multiplied by the feature distance to the reference stroke to analyze the individual features. The statistical methods which are individually validated are the mean, the standard deviation, and the z-score of the features of the strokes originating from the experienced boat. The z-score is a combination of the mean and the standard deviation. Without applying the weights to the features, the algorithm will not consider the amount of contribution of the feature in the model.

For comparing the methods, these three models are implemented with or without feature scaling, and with or without weights of the model. The results are compared by counting how many strokes have the same feature selected across the methods. With a function compareLists, the elements of two lists with the same length are evaluated and a score is returned of how many elements (strokes in this case) had the same value. This is used in the function compareToAll. Here the compareLists function is called for the primary list and a list of other lists. In this case, the list to compare is evaluated to the lists of the other methods. The number of element-wise correlations is returned. When calculating this for all the lists as the primary, the scores can be compared to know which lists have the same in common. This gives an idea of the performance identifying the features. However, this method gives not the full picture of how the algorithms perform.

The results are also validated by creating frequency distributions of the features in the identified worst-performing features. If the frequency of a given feature is not as expected based on the representation in the dataset of good and bad strokes, conclusions can be drawn about the confidence in the model.

6 Data analysis results

After the implementation of the methods for analyzing Kalman-filtered waveforms to correctly classify and identify the worst-performing features, the results are given by these methods. Firstly, the results of the stroke detection algorithm are given, this is phase 2 of the project. Also, the performance of identifying all strokes is given in the recorded training sessions. Secondly, the feature extraction is validated using a correlation matrix and scatter plots for visualizing the groupings in function of the categories, this is phase 3. Lastly phase 4, the results of the analysis model are given along with the performance of the machine learning classification model. The classification of the strokes to predict the origin of the recorded strokes is also a good validation if the extracted features give a meaningful representation of the strokes.

6.1 Stroke detection results

In this chapter, the results of the stroke detection algorithm are given. First, the working principle of the detection is validated using the detections in stroke waveforms. Secondly, stroke rate plots are given from training sessions. The stroke rate plots give a clear view of the accuracy of the algorithm, when an error does occur it will be significant enough to detect with drops or spikes.

6.1.1 Results of stroke detection

The first step in the detection algorithm is the identification of all zero crossings. The novice rowers have the highest probability of having zero crossings at the finish, this is visualized in Figure 13. On the rising edge between the catch and the drive peak, only one zero crossing is detected.

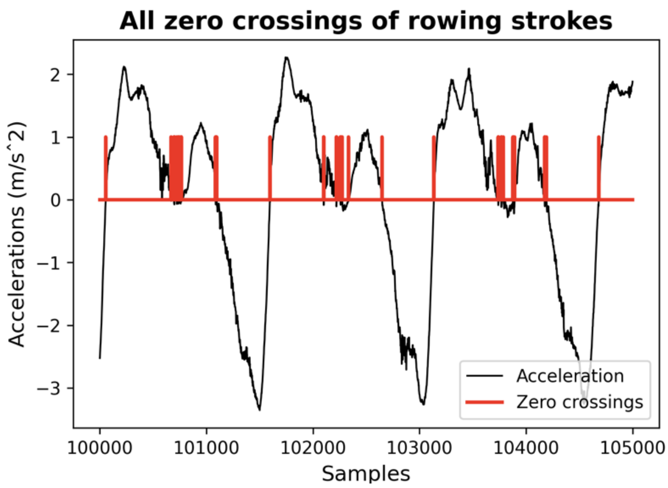

Figure 13: All zero crossings on rowing strokes

The strokes cannot be detected without filtering the zero crossings on the rising edge. However, an important conclusion is that the high accuracy of the zero crossing between the catch and maximum drive peak. Due to the physics of a rowing stroke and the working principle of the accelerometer sensor, these correct zero crossings are always in a timeframe of 2 samples off the actual zero crossing. With a typical sample size of 1400 for a steady state stroke, this results in a maximum deviation of 0.14% in time. This is significantly better than the 13% deviation previously stated in Chapter 5.1.1 using only peak detection.

The problem of detecting peaks as explained in Chapter 5.1.1 is applicable in the case of the windowed peak detection algorithm. If there are two possible maximum peaks, either of the two has a chance to be detected and this results in a deviation of the detection. In the use case of filtering the zero crossings, this is not a problem. In Figure 14 the detected peaks are given on rowing strokes, both for the catch (negative peak) and the maximum drive peak. The algorithm works perfectly for rejecting local maxima based on the dynamic threshold of the window filter. Specifically, the acceleration because of the recovery, in Chapter 2.2 this phenomenon is already explained.

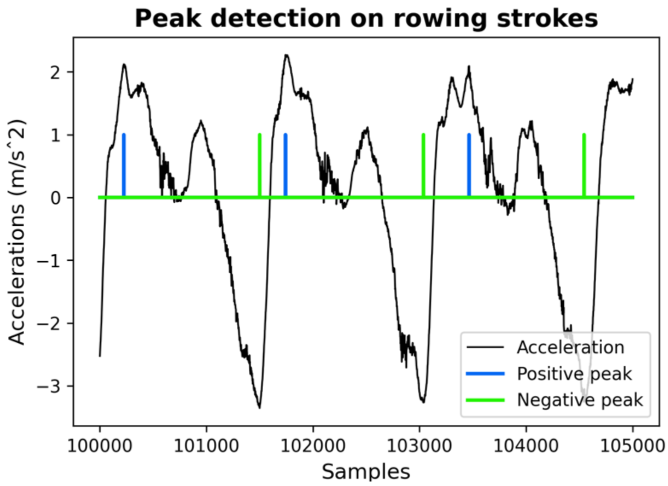

Figure 14: Detected positive and negative peaks on rowing strokes

Next, the results of the zero-crossing filter in total and subsequently the performance of the stroke detection are given. After filtering, the zero crossings left are the detected strokes. In Figure 15, the correct zero crossings for detecting the strokes are indicated in red.

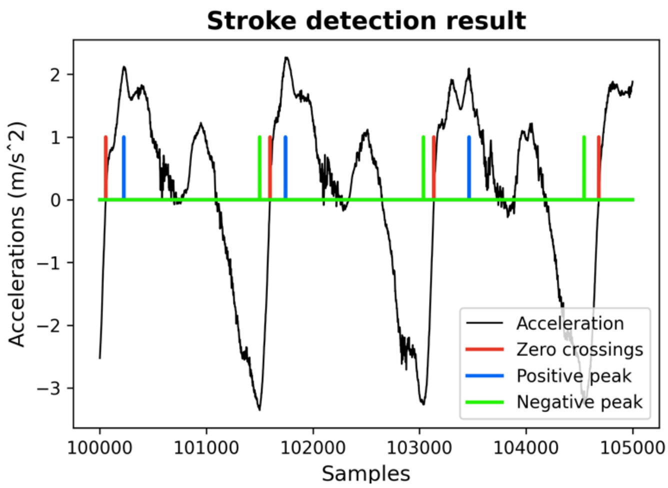

Figure 15: Resulting zero crossings after filtering

6.1.2 Stroke rate results

In this chapter, the stroke detection algorithm is validated by applying the algorithm to the training sessions and looking for inconsistencies in the stroke rate. A good example of the stroke rates of a training session is given in Figure 16.

Figure 16: Stroke rates of a consistent training session

When there is a problem with the detection, or the rowers keep starting and stopping for technical drills it manifests as given in Figure 17. When the stroke is incorrectly identified and one zero crossing is incorrectly filtered out, the resulting stroke is 2 or 3 strokes long.

Figure 17: Stroke rates of an inconsistent training session

For the total of 900 strokes recorded in this research project only 8 were incorrectly identified. Which is a success rate of 99,11% for the stroke detection algorithm. This is a very good result when considering that the probability is large for those 900 strokes that one rower makes a large mistake, which could lead to severe changes to the waveform. The conclusion from these statistics is that the detection algorithm is robust enough to generalize well to inexperienced rowers' waveforms.

6.2 Feature extraction results

In this section, the results of the feature extraction are given. Feature extraction is phase 3 of the project. Firstly, the blade detection waveforms for every step are explained. Secondly, the validation of the features is given in the form of a correlation matrix and scatter plots. By reviewing what the expected outcome is of a feature with the increase in stroke rate and experience the extraction is validated.

6.2.1 Blade detection results

With the RMS filter applied to the gravitational data, the amount of noise is calculated in Figure 18. With an increase in noise from the blade movements the RMS waveform also follows with a small delay, this is due to the sample window.

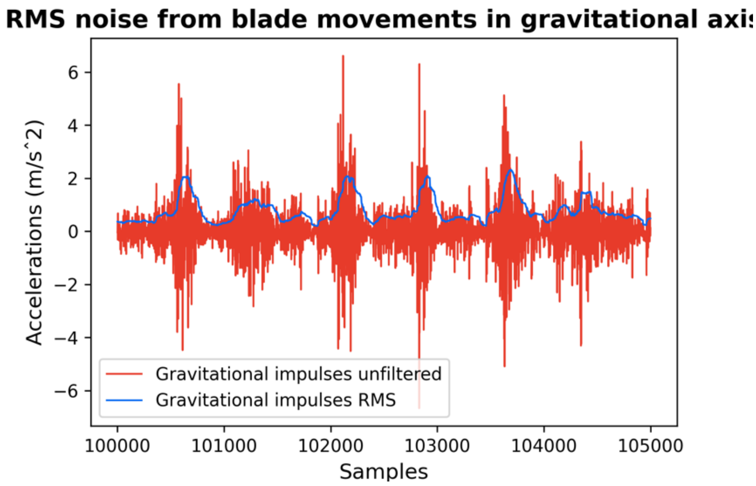

Figure 18: RMS-filtered gravitational impulses

In Figure 19 the RMS waveform is compared with a threshold to decide if the noise level is an actual detected blade movement. From the fourth pulse on, there are two detected blade movements because of a discontinued RMS signal. This is because the threshold is fixed, and no conditioning is done on the selection of movements. This error did not occur often enough to increase the complexity of the algorithm. In Chapter 5.3.1, these incorrectly extracted features will be filtered out.

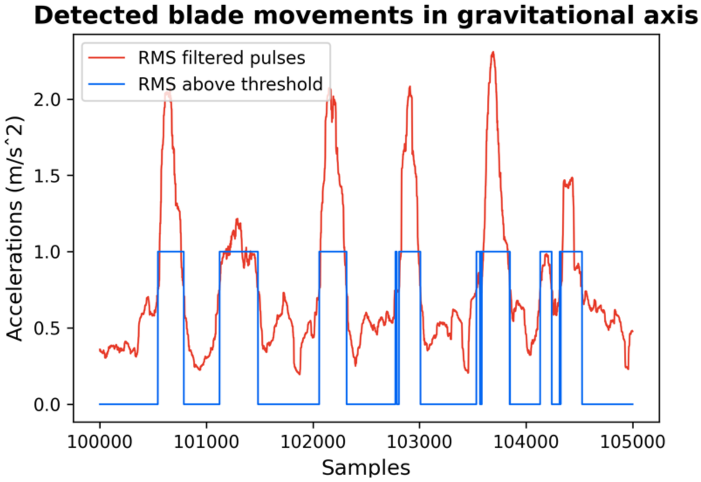

Figure 19: Detected blade movements on the RMS waveform

Lastly, the detected blade movements are given in combination with the stroke accelerations in Figure 20. When focusing on only the extraction of the blade at the finish the problem of multiple detections is only presented once, specifically in the third stroke. In total, only 2% of all recorded strokes had this issue. These strokes were recorded in the novice rowing boat, no strokes in the experienced boat suffered from too much noise fluctuations. In the future, this could be used as an additional metric. If an algorithm detects anomalies this does not have to be a downside. If a feature is extracted with this uncertainty of this detection algorithm, the prediction of problems on novice rowers could be improved.

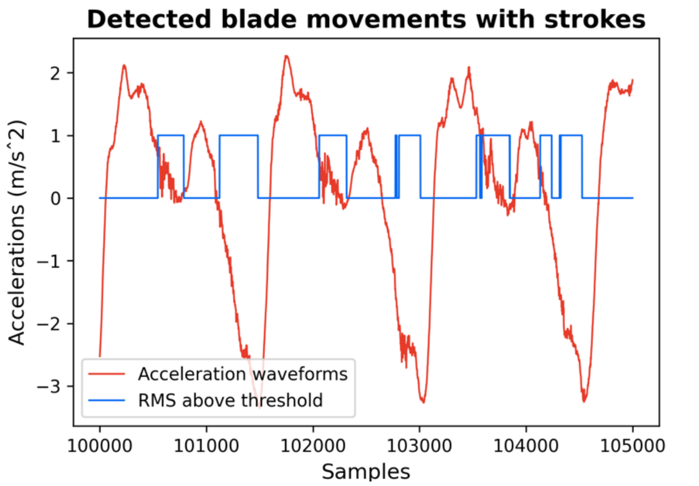

Figure 20: Detected blade movements on stroke waveform

6.2.2 Correlation matrix

The correlation matrix of the full dataset of extracted features is given in Figure 21. All the features are in both axes represented, which causes the diagonal of 100% correlation with themselves.

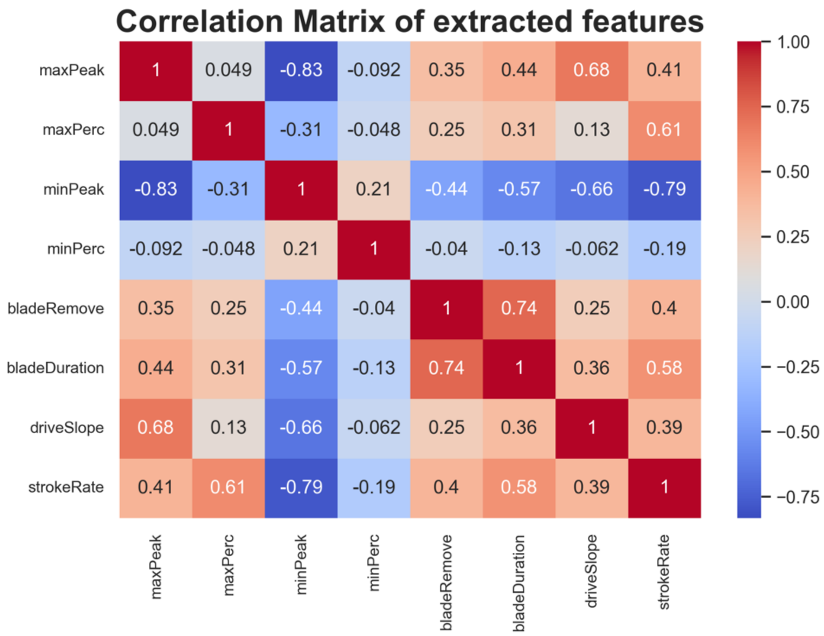

Figure 21: Correlation matrix of the extracted features

In the first column, the maximum drive peak, the lowest correlating feature is the timing of this peak with a 4.9% correlation. This means that the timing of the maximum drive peak is not related to the maximum peak value. However, the timing has a correlation factor of 61% with the stroke rate. The drive slope has a correlation of 68% with the drive peak. These results are expected based on the dynamics of a rowing boat. The high negative correlation between the maximum drive peak and the minimum peak at the catch is also expected. In the article of Kleshnev V., these observations are also made with the increase of experience in a rowing boat. The maximum and minimum peaks become larger and sharper with better rowing techniques [3].

The negative correlation between the minimum peak at the catch and the stroke rate indicates that the rowers are more synchronized at higher stroke rates. This is because rowers feel the movement of the boat better with larger acceleration swings because of the faster drive phases. The high correlation factor between the timing and the duration of the blade removal points is caused by problems at the finish of the drive.

From the correlation matrix, the importance of the minimum peak timing can be derived. It has no correlation with other features, not even stroke rate. This means that this feature most likely is not giving a lot of new data. This is also consistent with the theory of a rowing stroke; the stroke starts at the catch which is the most negative peak. The zero crossing after this peak is used for stroke detection. The timing between the catch and this detected stroke will be within a small range always the same due to the drive acceleration. The small difference will be in the slope of this initial acceleration. The slope of the initial acceleration is more defining the outcome in the positive peak.

6.2.3 Scatter plots of unprocessed features

The observations from the correlation matrix are better visualized in the scatter plots of the extracted features in Figure 22. The data points in these scatter plots are the raw data after extraction including outliers of small inconsistencies in detections. For the drive peak, the blade, and the drive slope the data are grouped together and a pattern is observed. The data points of the negative peaks are also grouped but they all have the same timing except for some outliers. This validates the assumptions from the correlation matrix, the minimum peak timing is not relevant for identifying the quality of a stroke. For the rest of the features, there is a need for more context by dividing the data points into categories.

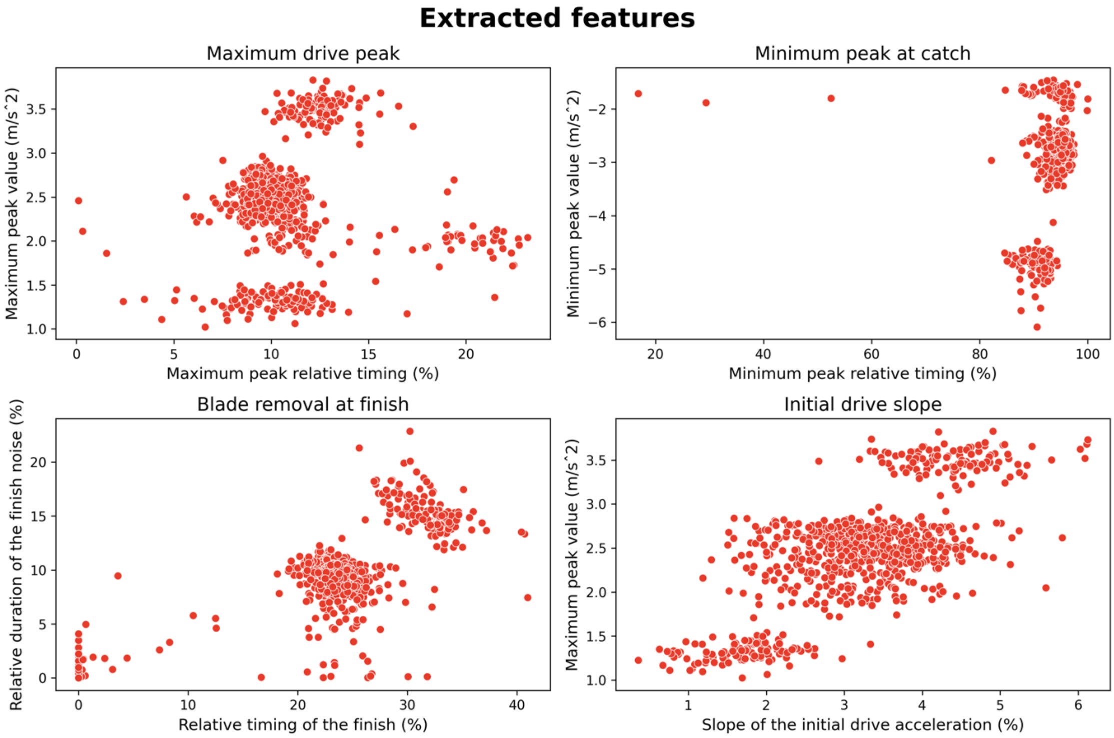

Figure 22: Scatter plots of the raw extracted features with no categories

With the data points divided into the stroke rate classes, the following results are observed in Figure 23. At race pace the main drive (maximum positive) peaks are higher or slightly lower with more varying timing. This is an expected result based on the changes in boat dynamics in the transition from steady state to race pace. With faster stroke rates, the boat reaches a higher average boat speed which in term increases the drag. This means that the boat decelerates faster and must accelerate more during the drive for continuing this high velocity [14]. This increase in drag is also observed in the minimum negative peak, the higher stroke rates have lower negative peaks. Although, this observation can also be due to the faster recovery and thus larger boat acceleration because of the preparation of the next stroke.

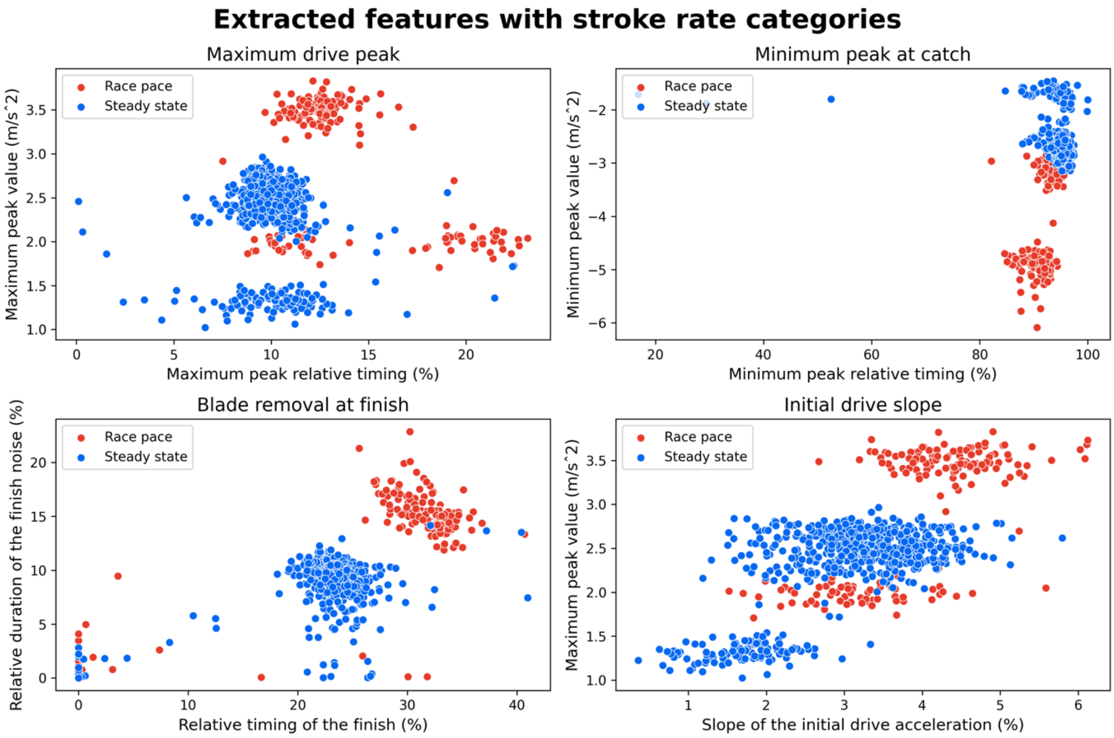

Figure 23: Scatter plots of the raw extracted features with stroke rate categories

When comparing the scatter plots with stroke rate classes against the boat classes in Figure 24 more conclusions can be drawn about the trends of these features. The drive slope for example of the experienced boat at steady state is a lot higher than the race pace of the novice rowers. This indicates that the technique of the experienced rowers is significantly better. The same observation can be made for the maximum and minimum peaks, the experienced rowers consistently have higher acceleration peaks and swings. This is a sign that their catch and finish are sharper and more synchronized. The timing of the positive peak of the novice crew at race pace is not consistent, which is a result of synchronization issues.

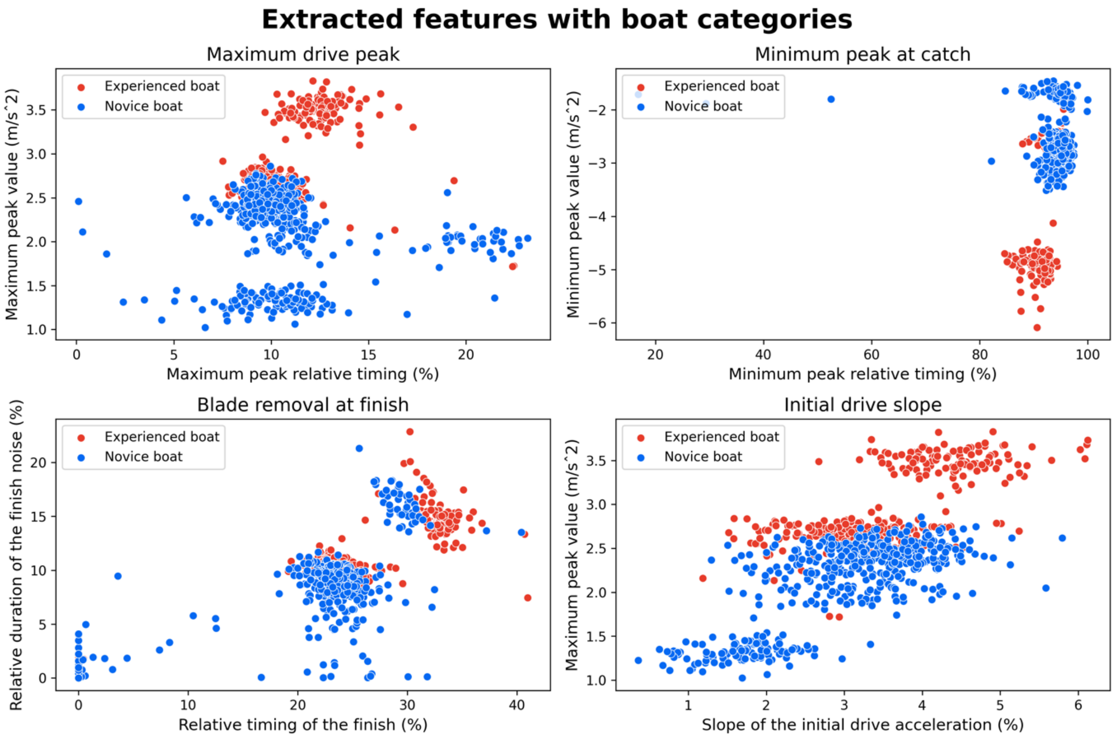

Figure 24: Scatter plots of the raw extracted features with boat categories

From these conclusions of the rowing technique by looking at the scatter plots the features are very useful for identifying technical problems and progress in the rowing technique. The extraction is accurate enough to manually analyze the patterns in the scatter plots. In the next chapter, the analysis is automated with algorithms to identify the least contributing feature value to compare to the good stroke model.

6.3 Analysis model results

In this chapter, the results are discussed in the last step of the proposed analysis technique, phase 4. Here, the individual stroke analysis for identifying the worst-performing features is performed. Firstly, the representation of boat type and stroke rates in the dataset are given after the outliers are removed. Secondly, the validation of the machine learning classification model is done with f1-scores. Lastly, the feature identification models are analyzed based on the frequency of different feature predictions.

6.3.1 Dataset after preprocessing

In Figure 25, the outlier removal process on a single feature is given. In this case for the bladeRemove feature which is the relative time of the finish. Due to extra noise in the waveform at the beginning of the stroke, some data points were detected around zero. Also, one late finish was detected at 40%. With the outliers removed based on the left boxplot, the data points range from 20.5 to 26%.

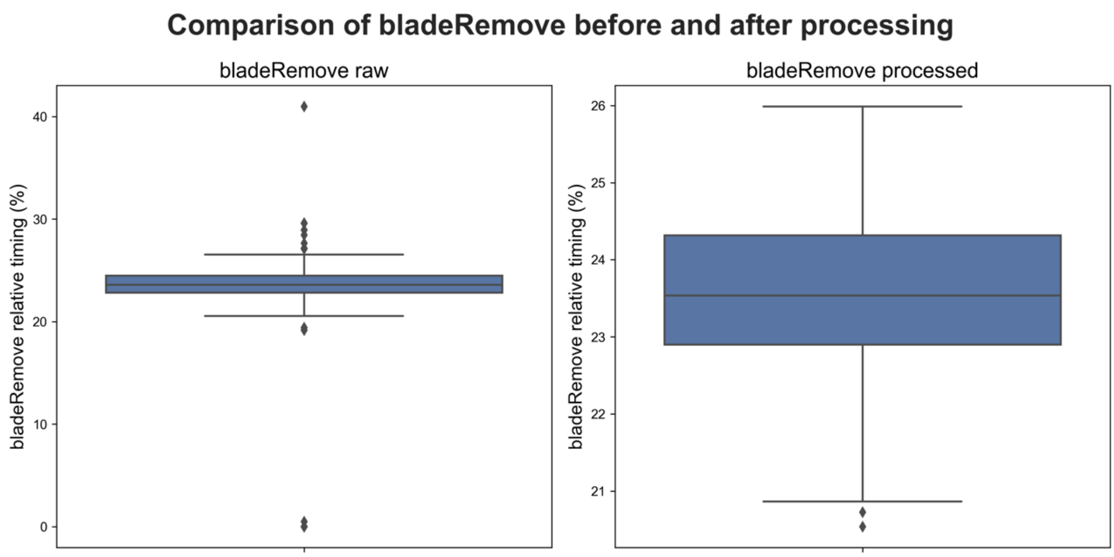

Figure 25: Boxplot comparison bladeRemove before and after preprocessing

When comparing the raw dataset to the preprocessed features in Figure 26, the following conclusions can be drawn. Firstly, the effectiveness of the outlier removal is depended on the sample size of the individual classes. The novice race-pace subset is too small to remove outliers. Secondly, this subset varies in all the individual features because of the inconsistency in technique. The variation is noticeable in the blade removal features. Too many of the data points were located far from the group's epicenter. When removing these outliers, the already small subset will halve in size.

A good example of the positive effect of removing the outliers is observed in the minPeak feature. A few strokes were incorrectly detected which resulted in an early minimum acceleration peak. When these few data points are removed the clusters are much cleaner (plot on the right).

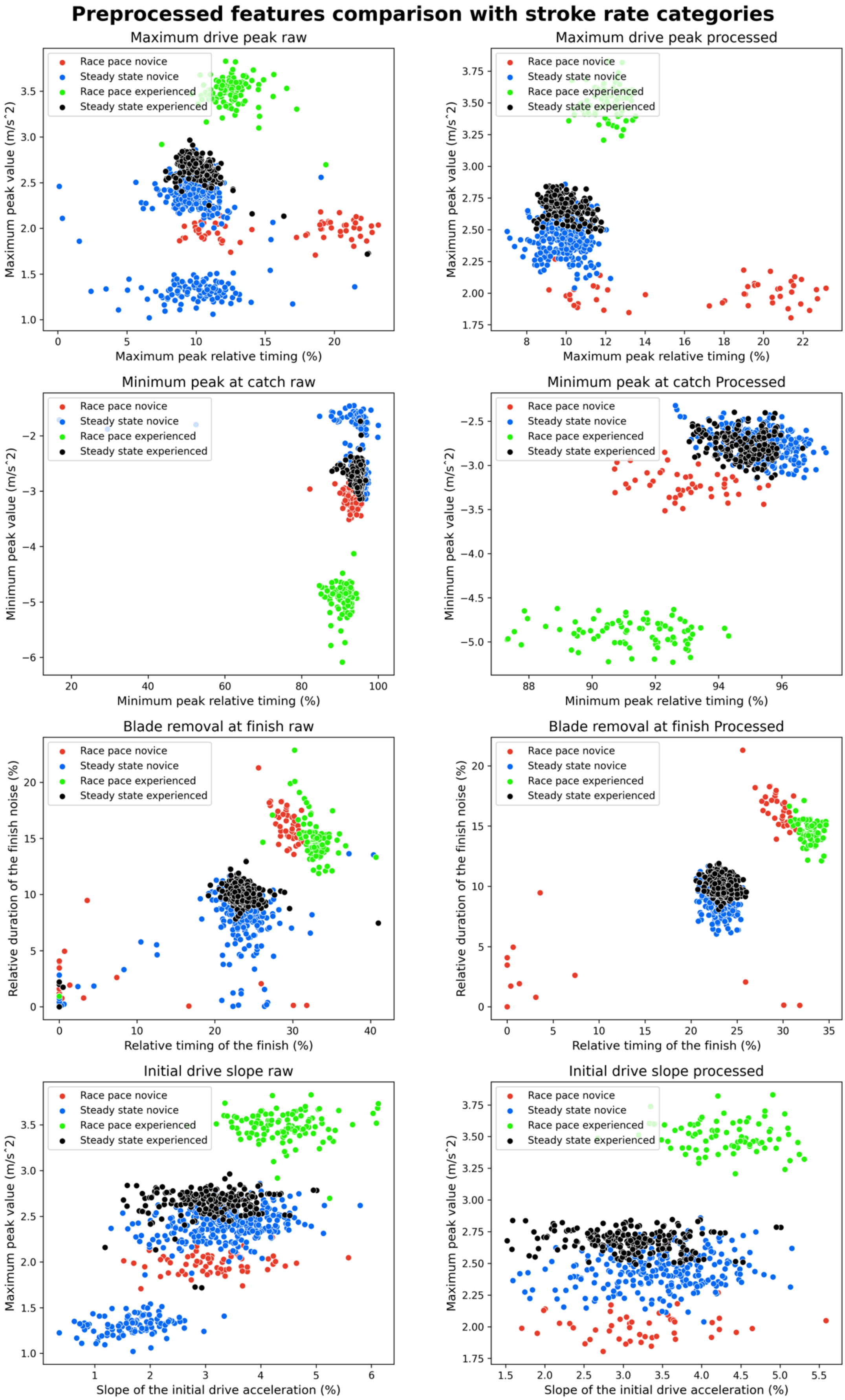

Figure 26: Preprocessed features comparison, raw features left and processed right

The representation of the 4 categories in data points after preprocessing is needed to evaluate the analysis model after preprocessing of the dataset. The representation in the dataset is given for these classes in Table 1. The representation of recorded strokes of the experienced and novice boat is equal, with both 309 recorded strokes. However, an important difference lies in the steady state and race pace categories. Only 15% of the recorded novice strokes are race pace while for the more experienced boat, 27% are recorded at race pace. When dividing up the dataset into subsets to create more specific models, this can only be applied to the steady-state subset. There are not enough race-pace strokes recorded to get the full idea of the effectiveness of the implementation. This concept is derived from overfitting in machine learning.

Table 1: Dataset representation of the 4 categories based on stroke rate and boat type

Class	Total strokes	Dataset representation (%)
Steady state experienced	226	36.6
Steady state novice	262	42.4
Race pace experienced	83	13.4
Race pace novice	47	7.6
Total	**618**	**100**

6.3.2 Stroke classification results

In Table 2 the f1-scores calculated on the test set with the three classification models are given. As expected, scaling the features enables the models to generalize better. The improvement for the logistic regression is 5%. With a neural network, the overall performance, and the increase in performance with scaling the features are the best. The difference in performance however is not large enough to steer clear from using logistic regression. Not all strokes of the experienced boat are perfect, creating an uncertain area in the dataset, even after removing the outliers. For the classification model, this means the possibility of a slight increase in prediction performance is not increasing the model's effectiveness for analyzing the features.

Table 2: F1-score comparison on the test set

ML model	F1 without scaling (%)	F1 with scaling (%)
Logistic regression	86	91
Neural network	89	97
SVM	86	94

The recall and precision results of the logistic regression model are given in Table 3. The results are given for the boat categories separately and then the weighted average is calculated. There is only one conclusion, but it can be looked at from different angles. In short, the conclusion of the model is it has a preference to identify a stroke as recorded from a novice instead of an experienced boat. This is preferable for the same reason as in the last paragraph. The chance that a stroke recorded from the experienced boat is less performing is larger than if a stroke from the novice boat is perfect. Often there are still minor faults in the experienced-level rowers' technique.

Table 3: Recall and precision of logistic regression with normalized features

Boat type	Recall (%)	Precision (%)
Novice	92	90
Experienced	90	92
Weighted avg	91	91

Overall, these results for classifying the strokes validate the success of the feature extraction. With machine learning algorithms there is a clear distinction between technical good and bad strokes. This method can be applied to even more data, with more general information to detect problems in technique. If the dataset includes a wider range of rowing strokes in different levels of technique, the prediction could be made of how experienced the rowers are with a quality factor. Also, when data on technical errors or general feedback from a training session is included, these machine-learning classification models can be used for predicting faults without further statistical analysis. Next, the results of the identification of the least-performing feature of a particular stroke are discussed.

6.3.3 Feature identification results

In this section, the results are given of the statistical analysis methods to identify the worst-performing feature in a particular stroke. The identification of the features is given for the methods applied to the normalized features. First, the logistic regression weights are given for the classification of the strokes as novice or experienced. These weights are the representation of the importance of the features in the classification model and are given in Table 4. These weights are used in the statistical methods to calculate a reference of a good stroke.

Table 4: Learned weights of the logistic regression classification model

Features:	Learned weights:
maxPeak	5.75
maxPerc	-0.55
minPeak	0.48
minPerc	-1.01
driveSlope	-0.66
bladeRemove	1.22
bladeDuration	1.84

Secondly, the results using both the steady state and race pace strokes together are displayed to create the reference model of a good stroke. The predictions of worst-performing features are given in Table 5. The methods without the weights look promising. These predicted features are varying but the identification is different for every method, so no conclusions can be drawn because these identifications cannot be verified in the current dataset. For the weighted methods, the models are not working. The maximum drive peak is predicted between 70 and 90% for all the strokes. This is not logical because the representation of good strokes is 50% in the dataset. With the reference stroke calculated based on these theoretical good strokes, at least 25% of the strokes should have other features identified as the worst-performing.

Table 5: Worst-feature identification methods on the dataset

Stroke	Means	Standard deviation	Z-score	Weighted means	Weighted std	Weighted z-score
1	minPerc	driveSlope	minPerc	maxPeak	minPerc	maxPeak
2	minPeak	maxPeak	bladeDuration	maxPeak	maxPeak	maxPeak
3	minPerc	minPerc	maxPeak	maxPeak	maxPeak	maxPeak
4	minPeak	driveSlope	maxPeak	maxPeak	maxPeak	maxPeak
5	minPerc	bladeDuration	bladeDuration	maxPeak	bladeDuration	maxPeak
6	driveSlope	driveSlope	driveSlope	maxPeak	driveSlope	maxPeak
7	bladeDuration	bladeDuration	bladeDuration	maxPeak	maxPeak	maxPeak
8	minPerc	minPerc	maxPeak	maxPeak	maxPeak	maxPeak
…	…	…	…	…	…	…
618	driveSlope	driveSlope	bladeDuration	maxPeak	maxPeak	maxPeak

The frequency distribution of the predicted worst-performing features is given in Table 6. Here it is clear that the weighted models are indeed not performing well. Based on the theory of the incorporation of weights to represent the importance of every feature, the models without weights implemented cannot be accurate. This is also noticeable with the 'maxPeak', this is the most important feature for the machine learning classification. With the standard deviation model, only 9.8% of the strokes are identified with the maxPeak as the worst-performing.

Table 6: Frequency table of identified features on the dataset

Feature	Means (%)	Std (%)	Z-score (%)	Weighted means (%)	Weighted std (%)	Weighted Z-score (%)
maxPeak	19.1	9.8	20.7	98.7	76.4	98.2
maxPerc	4.0	33.0	17.5	-	2.1	-
minPeak	20.9	2.1	5.0	-	-	-
minPerc	20.7	12.0	13.4	-	3.7	-
driveSlope	19.1	23.9	14.6	0.3	3.2	0.2
bladeRemove	5.5	3.6	7.1	0.3	1.4	0.3
bladeDuration	10.7	15.5	21.7	0.6	13.1	1.3

The reason for these results is that the reference stroke model is generated with all stroke rates of the dataset. The following results are given on the implementation with only the steady state subset of the dataset. The steady-state category is the largest category and still has enough records to test the proposed methods. In Table 7, the identified worst-performing features of strokes are given on only the steady state subset. This time only implemented on the weighted identification models.

Table 7: Worst-feature identification methods on the steady-state subset

Stroke	Weighted means	Weighted std	Weighted z-score
1	maxPeak	MinPerc	maxPeak
2	bladeDuration	bladeDuration	bladeDuration
3	maxPeak	bladeRemove	maxPeak
4	bladeDuration	maxPeak	bladeDuration
5	bladeDuration	maxPeak	bladeDuration
6	bladeDuration	maxPeak	bladeDuration
7	bladeDuration	maxPeak	bladeDuration
8	bladeDuration	bladeDuration	bladeDuration
…	…	…	…
488	bladeDuration	bladeDuration	bladeDuration

The frequency in the steady-state subset of the identified worst-performing feature is given in Table 8. These results are more realistic compared to the models applied to all records in the full dataset.

Table 8: Frequency table of identified features on the steady-state subset

Feature	Weighted means (%)	Weighted std (%)	Weighted Z-score (%)
maxPeak	46.3	63.7	58.6
maxPerc	-	-	-
minPeak	-	-	-
minPerc	5.9	3.1	3.3
driveSlope	2.7	0.4	1.0
bladeRemove	11.1	2.3	2.4
bladeDuration	34.0	30.5	34.6

With the count function applied to these 3 weighted methods using the steady-state subset, the z-score method scores the best. This means that the z-score identifies the worst-performing features with the highest correlation to the other two methods. As previously stated, this is not a definitive conclusion of the analysis methods. For accurate validation, extra information is needed on the individual strokes to check if the identification is true. In the next chapter, the conclusion, this shortcoming will be elaborated more.

An important limitation resulting of the recorded training sessions has become clear with the separation of the dataset into the subsets of race pace and steady state. The models are too different for the dynamics of the 2 different zones in stroke rates. Without more data between these 2 stroke rate zones, a model that generalizes well to all stroke rates cannot be implemented.

7 Conclusion

In this thesis, a method for analyzing technical problems in rowing strokes is developed. These methods should reduce the analysis effort of coaches to give feedback on rowing techniques. With the automation of the analysis and identification of potential errors, the delay in feedback on the training session can be reduced. This will result in a more efficient learning process for good rowing techniques. The project is divided into 4 phases. Firstly, acceleration waveforms are recorded of training sessions, with novice and experienced rowers. Secondly, individual strokes are detected and separated. Thirdly, features are extracted from the recorded rowing strokes, and a dataset is generated. Lastly, strokes are analyzed with a machine learning classification model and statistical methods to identify technical problems using the worst-performing feature.

The first phase, data capturing, is done with a prototype. The recording of data with a consistent method is necessary for analyzing the acceleration waveforms in a framework. With Kalman filtering the static and dynamic noise is removed on the forward axis of the rowing boat. The recording prototype is tested on 3 artificial scenarios to verify the principle without using the precious training sessions of the rowers. The data of training sessions are successfully recorded with noise filtering.

In phase 2, stroke detection is implemented. Stroke detection analyzes strokes separately and keeps a consistent workflow for all recorded data. Without accurate stroke detection, the timing of the features is not valuable. The resulting stroke detection is more accurate and robust than traditional detection algorithms. Without this proposed method, the implemented feature extraction in phase 3 is not possible.

After the stroke detection, the next phase is feature extraction. With feature extraction, the dataset used for further analysis of the technique is generated. A total of 4 features are extracted from the waveform. The timing of these features is also recorded. The resulting dataset has 9 characteristics for all strokes recorded from training sessions. The effectiveness of the features is analyzed using correlation and visual validation. The comparison between the features of novice and experienced rowers at race pace or steady state are in line with the findings of other researchers and personal experience as a coach. The generated dataset clearly represents the level of technique with a clear distinction between novice and experienced rowers.

The last phase is the analysis model for identifying potential problems in a single stroke. The analysis uses the dataset to detect which problems are present in a rowing stroke. The stroke is compared to a reference stroke created with the ‘good’ strokes. With a machine learning classification model, the importance of every feature is learned by predicting if a stroke is recorded in the experienced boat. The 3 methods with the incorporation of these machine learning weights give promising results. However, no definitive comparison can be made due to the shortcomings of the dataset. For validating the worst-performing feature method, extra information is needed on the errors of specific strokes.

Overall, it can be concluded that feature extraction on rowing strokes is necessary to drive the next evolution of technical analysis in the rowing sport. However, without the validation of the 3 worst-performing feature algorithms, a final conclusion about the effectiveness cannot be drawn. The results of the first 3 phases are a good basis to drive the next development of integrating modern data analysis technologies into the rowing sport.

With the cooperation of expert rowing coaches and a more diverse focus group to record strokes, the analysis models can be properly validated in the future. Even training models for predicting the score per feature are possible if the features are labeled with the level of performance. Then, classification algorithms can be used directly to predict the errors in technique.

By diversifying the team of rowers and maximizing the influence the researchers have on the training session, a better dataset can be created. Currently, no strokes are recorded between steady state and race pace. This is because of the planned training sessions of the rowing team. For a better understanding of the dynamics of a rowing boat, the whole spectrum of stroke rates needs to be recorded with a significant number of strokes in every zone. However, this limitation does not disprove the effectiveness of the results of this thesis.

References

[1] L. Formaggia, E. Miglio, A. Mola, and A. Montano, “A model for the dynamics of rowing boats,” *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS Int. J. Numer. Meth. Fluids*, vol. 61, pp. 119–143, 2009, doi: 10.1002/fld.1940.

[2] T. M. Hosea and J. A. Hannafin, “Rowing Injuries,” *https://doi.org/10.1177/1941738112442484*, vol. 4, no. 3, pp. 236–245, Apr. 2012, doi: 10.1177/1941738112442484.

[3] Kleshnev Valery, “Analysis of ‘boat acceleration,’” *Rowing Biomechanics Newsletter*, vol. 12, no. 140, Nov. 2012, Accessed: May 20, 2023. [Online]. Available: www.biorow.com

[4] V. Kleshnev, “Boat acceleration, temporal structure of the stroke cycle, and effectiveness in rowing”, doi: 10.1243/17543371JSET40.

[5] P. D. Harfield, “Enhancing the Mechanical Efficiency of Skilled Rowing through Shortened Feedback Cycles,” 2015.

[6] J. Von Zitzewitz *et al.*, “Real-time rowing simulator with multimodal feedback,” *Sports Technology*, vol. 1, no. 6, pp. 257–266, Jan. 2008, doi: 10.1002/JST.65.

[7] W. T. Higgins, “A Comparison of Complementary and Kalman Filtering,” *IEEE Trans Aerosp Electron Syst*, vol. AES-11, no. 3, pp. 321–325, 1975, doi: 10.1109/TAES.1975.308081.

[8] “pykalman — pykalman 0.9.2 documentation.” https://pykalman.github.io/ (accessed Jun. 10, 2023).

[9] “niru-5/imusensor: Python library for communication between raspberry pi and MPU9250 imu.” https://github.com/niru-5/imusensor (accessed Jun. 10, 2023).

[10] “smbus2 · PyPI.” https://pypi.org/project/smbus2/ (accessed Jun. 10, 2023).

[11] “NumPy.” https://numpy.org/ (accessed Jun. 10, 2023).

[12] “SciPy.” https://scipy.org/ (accessed Jun. 10, 2023).

[13] “scikit-learn: machine learning in Python — scikit-learn 1.2.2 documentation.” https://scikit-learn.org/stable/ (accessed Jun. 10, 2023).

[14] M. J. Hofmijster, E. H. J. Landman, R. M. Smith, and & A. J. Knoek Van Soest, “Effect of stroke rate on the distribution of net mechanical power in rowing”, doi: 10.1080/02640410600718046.