technische universität
dortmund

Robust Auditory Perception and Fault Resilience for Humanoid Robots in Competitive Environments

A thesis approved for the academic degree of

Doktor der Ingenieurwissenschaften (Dr.-Ing.)

at the

Faculty of Electrical Engineering and Information Technology

TU Dortmund University

by

Diana Kleingarn, M.Sc.

Supervisor:
Prof. Dr. Uwe Schwiegelshohn, TU Dortmund University

Co-Advisor:
Prof. Luca Iocchi, Sapienza Universita di Roma

Day of Oral Examination:
02.10.2026

Contents

1 Introduction **1**

1.1 The NAO Robot Platform . . . 2
1.1.1 Sensors . . . 3
1.1.2 Joint Structure . . . 3
1.1.3 Camera System . . . 4
1.1.4 Summary . . . 6
1.2 Problem Definition . . . 6
1.2.1 Whistle Detection . . . 7
1.2.2 Whistle Localization . . . 7
1.2.3 Internal Fault Detection . . . 8
1.2.4 Adaptive Motion Control . . . 8
1.3 Structure . . . 9

2 Whistle Recognition **11**

2.1 Related Work . . . 14
2.2 Dataset . . . 15
2.2.1 Audio preprocessing and feature extraction . . . 15
2.2.2 Training and Validation Dataset . . . 16
2.2.3 Stress Test Dataset . . . 17
2.2.4 Simulator Integration . . . 18
2.2.5 Data Quality Considerations . . . 18
2.3 Conventional Audio Processing . . . 19
2.3.1 Whistle Characteristics . . . 19
2.3.2 Acquiring Prior Knowledge . . . 20
2.3.3 Confidence Calculation . . . 23
2.4 Neural Network . . . 24
2.4.1 Architecture . . . 24
2.4.2 Training . . . 25
2.4.3 Fine Tuning . . . 28
2.5 Ambient Noise Detection . . . 29
2.6 Hybrid Structure . . . 30

2.7 Evaluation . . . 31
2.7.1 Comparison to State-of-the-Art Solutions . . . 31
2.7.2 Influence of the detection components . . . 33

3 Whistle Localization **37**
3.1 Related Work . . . 38
3.1.1 SSL on the NAO Platform . . . 38
3.1.2 Machine-Learning Approaches . . . 38
3.1.3 SSL in Domestic Robotics . . . 39
3.1.4 Room-Acoustic Cues and DRR-Based Methods . . . 39
3.1.5 Relevance to This Work . . . 39
3.2 Dataset . . . 40
3.2.1 Recording Setup . . . 40
3.2.2 Labeling and Labeltool . . . 42
3.2.3 Evaluation Dataset . . . 46
3.2.4 Dataset Composition . . . 48
3.3 Whistle Direction . . . 49
3.3.1 Direct Sound and Reverberation . . . 50
3.3.2 Input Representations . . . 50
3.3.3 Neural Network Architecture . . . 51
3.3.4 Loss Function . . . 52
3.3.5 Training Procedure . . . 52
3.3.6 Discussion . . . 53
3.4 Whistle Distance . . . 53
3.4.1 Distance Binning and Problem Formulation . . . 53
3.4.2 Input Features . . . 54
3.4.3 Neural Network Architecture . . . 54
3.4.4 Loss Function . . . 55
3.4.5 Training Procedure . . . 56
3.4.6 Discussion . . . 56
3.5 Evaluation . . . 57
3.5.1 Evaluation Setup . . . 57
3.5.2 Standing Evaluation: Direction and Distance . . . 58
3.5.3 Walking-in-Place Evaluation: Direction and Distance . . . 59
3.5.4 Runtime Evaluation . . . 59
3.5.5 In-Game Evaluation . . . 60

4 Broken Joint Detection **69**
4.1 Related Work . . . 70

4.2 Dataset . . . 71
4.2.1 Audio Anomaly Detection Dataset . . . 72
4.2.2 FDIR Evaluation Dataset . . . 74
4.3 Audio-Based Detection and Classification . . . 76
4.3.1 Neural Network . . . 76
4.3.2 Enrichment with Prior Knowledge . . . 77
4.3.3 Runtime and Deployment Considerations . . . 77
4.4 Sensor-Based Detection . . . 79
4.5 Residual Assessment . . . 80
4.5.1 Rule-Based Evaluation Framework . . . 80
4.5.2 Diagnostic Test Motions . . . 81
4.5.3 Short-Term and Long-Term Memory . . . 81
4.5.4 Joint State Categories . . . 81
4.5.5 LED Feedback . . . 82
4.5.6 Global Joint State Determination . . . 82
4.6 Prevention of Consequential Damage . . . 82
4.6.1 Arm Joints . . . 82
4.6.2 Leg Joints . . . 83
4.7 Evaluation . . . 84
4.7.1 Experimental setups . . . 84
4.7.2 Evaluation methodology . . . 84
4.7.3 Audio anomaly detection . . . 85
4.7.4 FDIR detection performance . . . 85
4.7.5 FDIR classification performance . . . 86
4.7.6 Detection timing . . . 87

5 Motion **89**
5.1 Related Work . . . 90
5.2 Keyframe Engine . . . 90
5.2.1 Structure . . . 91
5.2.2 Conditions . . . 92
5.2.3 Stand-Up Statistics . . . 93
5.3 Stand-Up . . . 94
5.3.1 Conditional Movement Selection . . . 95
5.3.2 Interception Movement . . . 97
5.3.3 Compensation of Wear . . . 100
5.4 Robot Safety . . . 103
5.4.1 Safety Architecture Overview . . . 103
5.4.2 Interaction Between FDIR and Motion Control . . . 104

5.4.3 Safety Despite Mechanical Wear . . . 104
5.4.4 Avoiding Unsafe or Unintended Movements . . . 104
5.4.5 Fall Safety and Interception . . . 105
5.5 Evaluation . . . 106
5.5.1 Experimental Setup . . . 106
5.5.2 Baseline: 2022 Stand-Up Routine . . . 106
5.5.3 Metrics . . . 106
5.5.4 Results . . . 107
5.5.5 Interpretation . . . 108
5.5.6 Limitations . . . 109

6 Conclusion **111**

A Appendix **113**
A.1 Whistle Localization In-Game Evaluation . . . 113

List of Abbreviations **127**

Bibliography **129**

List of Own Publications **135**

Affidavit **137**

1. Introduction

Humanoid robots are increasingly expected to operate in environments that are designed for and shared with humans. In such settings, acoustic communication plays a central role. Spoken language is the most natural and expressive form of acoustic interaction for humans, and it enables nuanced and flexible communication. However, speech is also highly sensitive to environmental conditions. In noisy surroundings or during fast-paced activities, even humans struggle to understand spoken words reliably.

In these situations, people often rely on simpler and more distinctive acoustic cues. Emergency sirens, short confirmation tones in public infrastructure, and referee whistles in sports are all examples of brief, unambiguous sounds that convey information quickly and robustly. These signals remain effective even when speech becomes unintelligible, and they allow for immediate reactions without requiring complex interpretation.

For robots operating in human environments, the ability to interpret non-verbal acoustic cues is essential. Unlike consumer devices that rely on cloud-based processing and assume quiet surroundings as well as stable network connectivity, mobile robots must react immediately, operate without external computation, and remain fully functional even when wireless communication is unreliable or temporarily unavailable. They must also process acoustic information reliably despite significant background noise and limited onboard resources. Performing audio processing directly on the robot is therefore not a matter of convenience, but a requirement for robustness and real-time responsiveness. Only by avoiding external dependencies can robots guarantee timely and predictable reactions in dynamic and noisy environments.

The RoboCup Standard Platform League (SPL) provided a well-defined and challenging benchmark for studying these problems. Until its discontinuation after RoboCup 2025, all teams competed using the same humanoid robot platform, the NAO V6 by Aldebaran. This uniform hardware environment ensured that progress depended entirely on perception, decision-making, and control algorithms rather than on mechanical differences. The SPL setting combines a strictly regulated field with unpredictable external influences such as crowd noise, overlapping matches, and mechanical wear. These characteristics make the SPL a broadly used and realistic test environment for a wide range of robotic perception and control research.

The first part of this work focuses on whistle recognition, a fundamental requirement in RoboCup SPL. To address the acoustically challenging competition environment, we

combine neural networks with prior knowledge to achieve robust whistle detection under extreme noise conditions [1].

Since detection alone is insufficient, we extend our system with a neural network-based sound source localization module that estimates both direction and distance of whistle events [2]. This enables robots to ignore irrelevant signals and react only to whistles that are spatially consistent with the current game.

Building on these audio perception capabilities, we explore how auditory information can support self-diagnosis. Mechanical wear is common in humanoid robots, and audio provides an additional modality that reveals characteristic signatures of gear damage. By combining sensor-based and audio-based residuals, we develop a multimodal Fault Detection, Isolation, and Recovery (FDIR) system that identifies mechanical issues early and prevents consequential damage [3].

Finally, autonomous robots must not only detect faults but also adapt their behavior accordingly. The last part of this thesis introduces a modular, feedback-driven motion framework that selects and adapts stand-up motions based on context, success statistics, and the robot's current mechanical condition [4]. These mechanisms allow robots to remain operational longer, recover more reliably, and avoid unnecessary stress on damaged components.

Together, these four components form a coherent framework for robust audio perception and fault-aware behavior in humanoid robots, combining whistle detection, whistle localization, audio-supported self-diagnosis, and adaptive motion control into a unified approach. Each component has been published individually in the context of RoboCup symposiums [1–4]. This thesis unifies, extends, and contextualizes these contributions, demonstrating how multimodal audio perception and adaptive control can significantly enhance the autonomy and resilience of humanoid robots operating in real-world environments.

1.1. The NAO Robot Platform

The platform used in this thesis is the NAO V6 robot, developed by Aldebaran Robotics [5], and shown in Figure 1.1. The NAO is a widely adopted research and educational robot and served as the standard platform in the RoboCup SPL. It stands 57.4 cm tall, weighs 5.48 kg, and provides 25 Degrees of Freedom (DoF), enabling a wide range of human-like motions. Its compact size and standardized hardware make it an ideal platform for benchmarking algorithms in perception, control, and human-robot interaction.

The NAO V6 is powered by an Intel Atom E3845 quad-core processor running at 1.91 GHz, with 4 GB of RAM and a 2 MB L2 cache. While modest compared to modern desktop systems, these computational resources highlight the need for efficient, real-time



Figure 1.1.: NAO V6 [5].

algorithms that can operate reliably under the strict hardware constraints imposed by the SPL rules [6].

1.1.1. Sensors

A variety of sensors allow the NAO to perceive both its own state and the surrounding environment. For proprioception, each joint is equipped with position sensors offering 12-bit resolution, corresponding to a precision of approximately 0.1 °. An Inertial Measurement Unit (IMU) provides three-axis gyroscope data with a precision of 5% and angular speed up to 500 °/s, as well as a three-axis accelerometer with 10% precision and a range of approximately 2 g. Force Sensitive Resistors (FSRs) under each foot measure ground contact forces in the range of 0–25 N with a sensitivity of ±40 g, supporting balance estimation and gait stability.

For exteroception, the NAO is equipped with four omnidirectional microphones, as shown in Figure 1.2, each with a sensitivity of 250 mV/Pa ±3 dB at 1 kHz and a frequency response from 100 Hz to 10 kHz. These microphones are particularly relevant for this work, as they enable audio perception tasks such as whistle detection and sound source localization. In addition, the robot features two cameras (not used in this thesis) that provide visual input for other perception tasks.

1.1.2. Joint Structure

The NAO V6 provides 25 DoF distributed across the head, arms, legs, and hands. Each joint is actuated by a compact electric motor coupled with a multi-stage plastic gearbox that reduces motor speed and increases torque. A magnetic or optical encoder measures the joint angle, while the motor current is monitored to estimate load. The NAO does not include a physical temperature sensor in its joints; instead, temperature is estimated internally based on motor current and operating time.

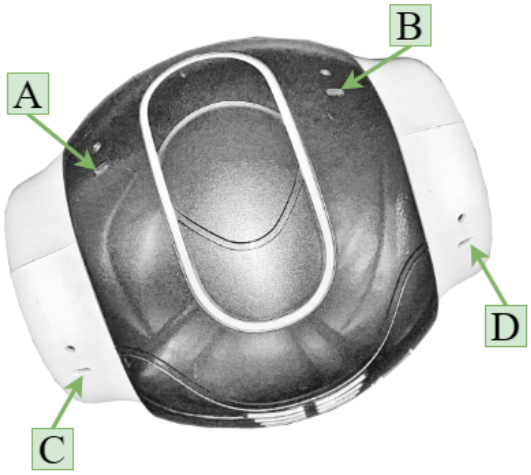


Figure 1.2.: NAO V6 microphone positioning. First published in [2].

The mechanical design of the joints follows a consistent structure across the robot, but the load and wear characteristics differ depending on the joint location. Shoulder and hip joints experience the highest mechanical stress, while elbow and wrist joints are subject to lower loads. The plastic gearboxes introduce a small amount of mechanical play that increases with wear and is a known characteristic of the NAO platform.

Figure 1.3–Figure 1.6 provide an overview of the joint layout.

1.1.3. Camera System

In addition to the audio sensors, the NAO V6 is equipped with two front-facing cameras located in the robot's head. Both cameras use an OV5640 Complementary Metal-Oxide-Semiconductor (CMOS) image sensor with a resolution of 5 Mpx and an active pixel array of 2592×1944. The sensor features a pixel size of $1.4 \times 1.4\,\mu m$, a dynamic range of 68 dB at $8\times$ gain, and a maximum signal-to-noise ratio of 36 dB. For typical perception tasks, the camera outputs 640×480 px images at 30 Hz in either YUY or RGB format using a rolling-shutter exposure mode.

A central characteristic of the NAO's visual system is its field of view. The lens provides a diagonal field of view of 67.4 °, corresponding to a horizontal field of view of 56.3 ° and a vertical field of view of 43.7 °. While this field of view is sufficiently wide for many robotic perception tasks, it is considerably narrower than human vision. Humans achieve a binocular horizontal field of view of approximately 120 °, with peripheral perception extending up to 200 °, and a vertical field of view of roughly 130 °. Consequently, the NAO perceives only a limited visual cone at any given moment and must rely on active head movements to compensate for its restricted coverage.

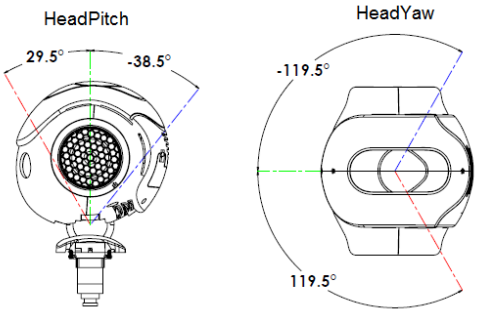


Figure 1.3.: Head joints (Yaw, Pitch) [7].

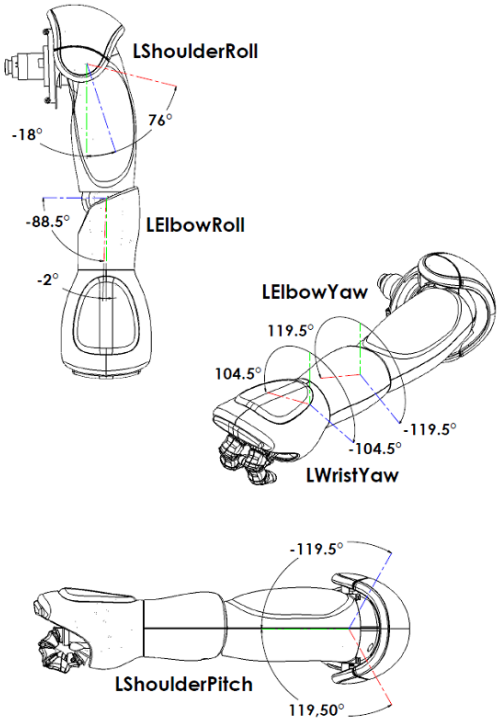


Figure 1.4.: Arm joint structure left arm [7].

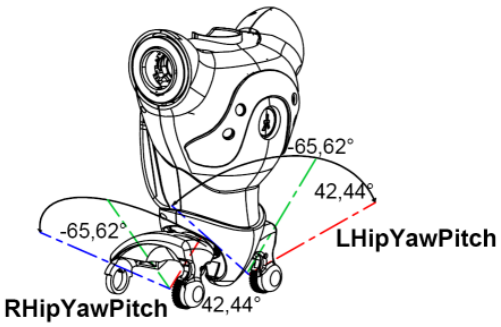


Figure 1.5.: Hip joint structure [7].

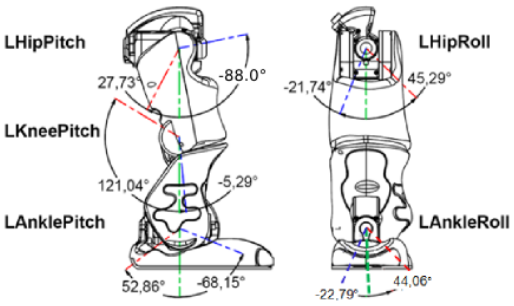


Figure 1.6.: Left leg joint structure [7].

The camera supports an autofocus mechanism with a focus range from approximately 10 cm to infinity. Figure 1.8 and Figure 1.7 illustrate the horizontal and vertical fields of view, respectively, highlighting the effective visual coverage of the NAO V6.

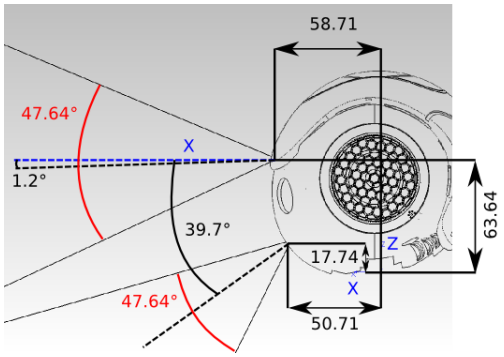


Figure 1.7.: Side view of the NAO V6 camera field of view [7].

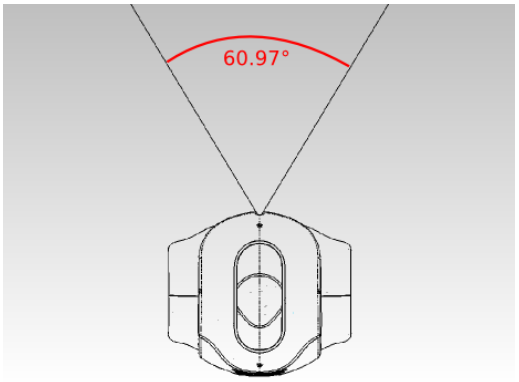


Figure 1.8.: Top-down view of the NAO V6 camera field of view [7].

1.1.4. Summary

In summary, the NAO V6 combines a compact humanoid form factor with a diverse sensor suite and constrained computational resources. These characteristics make it a representative platform for developing and evaluating methods in audio perception, fault detection, and motion control under real-world conditions.

1.2. Problem Definition

In this thesis, we aim to strengthen the autonomy of humanoid robots by focusing on audio perception and adaptive control as key modalities. For a robot to act independently, it must perceive its surroundings, diagnose its internal state, and adapt its actions without relying on external infrastructure such as servers or additional cameras, which are not always available or comparable to humans and for that reason prohibited in RoboCup SPL settings [6]. We take human perception and behavior as inspiration and investigate how robots can use audio and modular motion strategies to improve their awareness, robustness, and adaptability on the field. Since all processing must be performed on-board under strict real-time constraints, we rely on lightweight, task-specific models rather than large end-to-end architectures, which would exceed the available computational resources and reduce the predictability required for autonomous behavior. Our work addresses four interconnected challenges:

1. whistle detection under noisy conditions,
2. sound source localization to assess the relevance of detected signals,

3. fault detection to identify and classify mechanical issues, and

4. motion control that adapts dynamically to faults and changing conditions while minimizing unnecessary damage.

1.2.1. Whistle Detection

We first consider whistle recognition, which represents the initial audio challenge in RoboCup SPL. Referees use whistles to signal game events, and the SPL rules enforce timely detection by delaying the corresponding game controller signal. By default, these game controller signals are delayed by 15 s, ensuring that robots that fail to detect the whistle immediately are penalized by starting later. Conversely, if a robot begins play before the referee whistle, it receives a 15 s time penalty. While humans can easily distinguish whistles from background noise, robots must rely on signal processing to define and detect whistle-like sounds. Although audio processing is not new in robotics, it is far less common than vision-based perception, making whistle detection a representative example of how non-verbal acoustic cues can support reliable decision-making under challenging conditions. The noisy competition environment, including high-pitched voices, mechanical disturbances, and overlapping matches, makes reliable whistle detection particularly demanding. A common error occurs when loud or high-frequency sounds are misclassified as whistles. Since both false positives and false negatives are penalized, we must balance sensitivity and specificity. In critical areas such as the goal zone, falsely reacting to noise is particularly unfortunate, yet waiting for the delayed game controller signal reduces the time available for the robot to reach the kick-off position. This trade-off illustrates the importance of robust whistle detection under noisy conditions.

1.2.2. Whistle Localization

Recognizing a whistle alone is not sufficient. We also need to determine whether the signal is relevant to the current match. Humans achieve this by localizing sound sources, an ability that is essential in RoboCup due to the cocktail-party scenario created by spectators and simultaneous games. Improved whistle detection increases the risk of reacting to irrelevant whistles from other fields, which makes sound source localization indispensable. Direction estimation relies on the Time Difference of Arrival (TDOA) principle, which requires at least two sensors. The NAO robot's four-microphone array enables not only direction estimation but also a distinction between front and back, since the spatial arrangement of the microphones provides asymmetry. Humans, in contrast, learn to differentiate front and back primarily through the unique shape of the auricle and the head, which is described by the head-related transfer function (Head-Related Transfer Function (HRTF)). Distance estimation is even more complex: while humans can roughly

judge whether a sound is near or far, precise measurement is difficult. For robotic decision-making in a RoboCup scenario, a coarse classification into near, medium, and far distances is sufficient to evaluate whistle relevance. This step illustrates how external acoustic signals can be transformed into actionable information that directly influences the robot's behavior, such as whether to initiate a kick-off routine or ignore irrelevant whistles from neighboring fields.

1.2.3. Internal Fault Detection

In addition to external signals such as the referee whistle, we also use audio perception to monitor the robot's internal state and detect unusual changes such as broken joints or gear damage. Like any other machine, robots are subject to wear and tear, which over time leads to increased joint play and potential failures in mechanical components. Many defects can be detected with existing joint sensors, for example by calculating the angle error from the joint angle sensor or by monitoring the current and stiffness values. However, our objective is not only to identify that a defect exists, but also to decide how the robot should react. To select an appropriate recovery strategy, the robot requires a more detailed understanding of the underlying cause. We therefore employ audio as an additional source of information for self-diagnosis. This allows us to differentiate between an arm that is blocked by external factors, such as contact with other robots, obstacles, or the robot lying on its own arm, and an arm that is stuck by a sheared tooth of a damaged gear. In contrast to whistle perception, these acoustic cues originate from within the robot itself, forming a complementary counterpart to external signals and enabling a unified perspective on how audio supports both situational awareness and self-assessment. The latter case produces characteristic mechanical noise, and the robot should avoid further actuation of the affected joint to prevent additional damage. In contrast, when the blockage is external, it is preferable to wait until the obstacle is removed, request assistance, or attempt to free the arm autonomously. Since a robot cannot successfully execute every motion with an unusable joint, distinguishing between these cases is essential. Our main goal is to prevent unnecessary damage while keeping the robots operational and in the game for as long as possible. Detecting faults, however, is only useful if the robot can adapt its behavior accordingly.

1.2.4. Adaptive Motion Control

To respond effectively to these diverse situations, we design the motion engine to be modular and feedback-driven. Inspired by human movement, we avoid repeating failed motions and adapt strategies based on context. For instance, a person with an injured shoulder would not use that arm to stand up, nor would this person persist with a movement that consistently leads to falling. Accordingly, we divide robot motions into smaller segments

and link them in a decision-tree-like structure. Success statistics guide the selection among alternatives, enabling the robot to choose the most promising recovery strategy. This approach reduces fall risk, accelerates recovery, and adapts automatically to changing field conditions or mechanical wear, without requiring manual calibration.

1.3. Structure

This thesis is organised into four main chapters that build upon one another and together form a coherent framework for robust auditory perception and fault-aware motion control on the NAO robot.

Chapter 2 introduces the whistle recognition system. It describes the dataset, the conventional audio processing pipeline, the neural network architecture, and the hybrid fusion mechanism that combines prior knowledge with learned representations. The chapter concludes with an evaluation under laboratory and real-world conditions.

Chapter 3 extends whistle perception by estimating the direction and distance of whistle events. It presents the dataset, the input representations, the neural network architectures for direction and distance estimation, and a detailed evaluation including in-game results from multiple RoboCup tournaments.

Chapter 4 focuses on fault detection, isolation, and recovery. It introduces two datasets for audio-based and sensor-based detection, describes the anomaly detection network, the rule-based residual assessment, and the integration with the robot's behavior system. The chapter concludes with an evaluation of detection performance and timing behavior.

Chapter 5 addresses the motion control perspective. It introduces the keyframe engine, the conditional stand-up framework, interception mechanisms, and the compensation of mechanical wear. The chapter evaluates the robustness of the proposed stand-up routine across different surfaces and hardware conditions.

Finally, Chapter 6 summarises the contributions of this work and outlines directions for future research.

2. Whistle Recognition

Whistle recognition was introduced in the RoboCup SPL in 2015 as the official signal to indicate the start of a game. Initially, the disadvantage in the form of a time delay of 15 s of the game controller signal compared to the whistle was only applied from the quarterfinals onwards. Since then, this delay has been enforced for all whistle signals during matches. With the in-game Visual Referee Challenge [8] in 2024, reliable whistle recognition has become increasingly crucial for competitive performance. This delay directly affects match performance, as missed whistles reduce the available time for critical actions, while false detections can disrupt team coordination. After the initial kickoff, an additional challenge arises: when the ball is located near the goal, a whistle cannot be interpreted as a goal signal without visual confirmation that a goal has actually been scored. This requirement for cross-modal verification further increases the importance of reliable and low-latency whistle detection, as the robot must integrate auditory cues with visual evidence before executing game-critical state transitions.

Since whistle recognition is relevant for all RoboCup teams, a variety of solutions have already been presented or published in teams' code releases. The existing approaches, summarized in Section 2.1, each have advantages but fail to cover all critical situations. Some methods are effective at detecting soft whistles but tend to misclassify loud noise as a whistle. Other approaches improve robustness against noise but suffer from a high rate of false negatives. Maintaining the balance between false positives and false negatives is particularly challenging in noisy environments. False recognitions are undesirable in any soccer game, especially given the disadvantages of missing a whistle. For example, robots must return to their own half after scoring a goal within 30 s instead of the 45 s that would be available if the whistle were detected immediately, or wait 15 s to receive the referee command for kick-off or a penalty kick. Conversely, misclassifying crowd noise as a whistle is even more problematic, since a whistle during a match indicates that a goal has been scored and triggers robots to return to their starting positions. This related work provides the foundation for our own approach and motivates the hybrid architecture described in this chapter.

The competition environment poses significant challenges for whistle recognition. Robots must react reliably to whistles despite reverberation in indoor halls, overlapping matches on adjacent fields, and crowd noise. High-pitched voices, particularly from children, overlap with the fundamental frequency range of whistles (2–3 kHz) and their harmonics, which

increases the risk of misclassification. Mechanical sounds generated by robots, such as servomotor movements, impacts from falls, or gear noise, contribute additional broadband disturbances. Furthermore, the reverberation of indoor sports halls alters the spectral characteristics of both whistles and background sounds, creating complex room impulse responses that vary depending on the position of the robot on the field. This reverberation can smear the temporal onset of a whistle, making it harder to distinguish from impulsive noises.

Another challenge arises from the cocktail-party scenario typical of RoboCup events. Multiple matches can take place simultaneously on adjacent fields, and whistles from other referees propagate across the venue. While humans can usually identify whether a whistle is relevant to their own game by combining auditory cues with contextual knowledge, robots must rely primarily on acoustic features. This reliance on audio is further motivated by the physical constraints of the NAO robot. Due to its height of approximately 58 cm, the robot cannot simultaneously perceive the ball and the referee, as both are outside its limited field of view, see Section 1.1. The onboard camera provides only a narrow perspective of the field, which makes visual detection of referee signals impractical. Consequently, whistle recognition through audio becomes the most reliable modality for detecting game events. At the same time, recognizing the presence of a whistle alone is insufficient: robots must also determine whether the whistle originates from their own field or from an adjacent match. This requirement introduces the challenge of direction estimation, which we address in detail in Chapter 3.

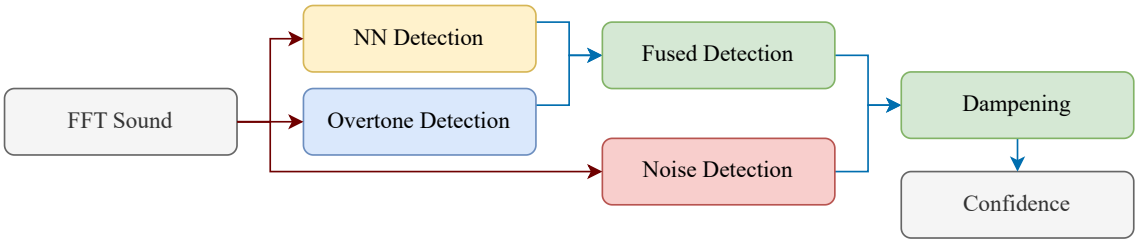


Figure 2.1.: Overview of our whistle recognition pipeline where red arrows indicate the flow of Fast Fourier Transform (FFT) information of the current audio and blue arrows indicate the confidence flow. First published in [1].

As illustrated in Figure 2.1, our whistle recognition pipeline consists of several interconnected modules. The audio signal is first transformed into the frequency domain using an FFT. Overtone detection (Section 2.3) and noise detection (Section 2.5) provide prior knowledge about characteristic whistle patterns and ambient disturbances. In parallel, a neural network (*NN Detection*) analyzes the spectral features. The outputs of the overtone detection and the neural network are then fused into a combined detection result. Subsequently, a dampening module adjusts this fused confidence according to the overall noise level, reducing the likelihood of false positives in loud environments. Finally, the

confidence value is passed on as the decision output of the pipeline. This architecture ensures that the advantages of traditional signal processing and machine learning complement each other, while the dampening factor provides robustness under challenging acoustic conditions. The following sections describe each module in detail and evaluate the complete system under realistic RoboCup conditions. To illustrate how whistle recognition influences the robot's behavior during a match, Table 2.1 summarises the typical actions triggered by different whistle types in RoboCup SPL.

Table 2.1.: Rule-compliant robot responses to whistle-triggered game events in RoboCup SPL. Only behaviors directly implied by the rules are listed; team-specific strategies are not included.

Game event	**Rule-compliant robot response**
Kick-off	Robots may begin playing immediately after detecting the whistle. Detecting the whistle provides up to a 15 s advantage over waiting for the delayed GameController signal.
Goal scored	Robots must stop ongoing actions and return to their own half. Detecting the whistle gives the full 45 s return window; otherwise only 30 s remain once the GameController state changes.
Free kick / Penalty	Robots must avoid interfering with the restart, remain stationary if required, and maintain the mandated distance from the ball.

For the evaluation, presented in Section 2.7, we investigate situations that have led to or could potentially lead to misrecognitions. The evaluation consists of two complementary parts. First, we compare our proposed hybrid solution simultaneously with the code releases of two other SPL teams and with our previous whistle recognizer based solely on a neural network. This comparison provides a benchmark for robustness against noise and highlights improvements in balancing sensitivity and specificity under realistic match conditions.

Second, we analyze the contribution of the individual components within our pipeline. In particular, we examine how overtone detection, neural network detection, and the dampening module influence the overall confidence and recognition accuracy. By isolating and testing these modules, we can quantify their respective strengths and identify which combinations are most effective in reducing false positives and false negatives. This constitutes an ablation study, in which individual components are selectively enabled or disabled to assess their specific contribution to the overall system performance. Such component-level

analysis ensures that the improvements of the hybrid system are not only demonstrated in aggregate but also understood in terms of their internal mechanisms.

2.1. Related Work

Whistle recognition has been studied in several robotic and non-robotic contexts, although the specific requirements of the SPL impose additional constraints on robustness, real-time performance, and on-board processing. Early work by Bonarini et al. [9] introduced a composite whistle recognizer combining spectral feature extraction with a neural classifier. Their system was designed for the Middle-Size League and focused on identifying whistle patterns rather than single events. While pattern recognition is not required in the SPL, their use of a frequency mask and a lightweight neural classifier illustrates the effectiveness of combining prior knowledge with learned components. They also provided a tuning tool to adapt the system to different whistles and noise conditions.

Poore et al. [10] proposed two whistle recognition approaches for the NAO robot. The first relies on a frequency or band-pass filter to isolate the fundamental frequency of the whistle, conceptually similar to the frequency mask used by Bonarini et al. The second approach applies logistic regression with ℓ_2-regularization to log-energy spectra. Although logistic regression is a linear statistical classifier rather than a learning-based model in the modern sense, their evaluation across multiple environments showed that this data-driven method consistently outperformed the purely filter-based approach, particularly under noisy conditions.

Several SPL teams have developed whistle recognizers tailored to the NAO platform. The HULKs [11] extend the frequency-mask idea by explicitly validating the presence of the first overtone. Since referee whistles exhibit a stable harmonic structure, the detection of at least one overtone provides a strong indicator of a true whistle event. The spectrogram in Figure 2.2 illustrates this characteristic pattern for different whistle types.

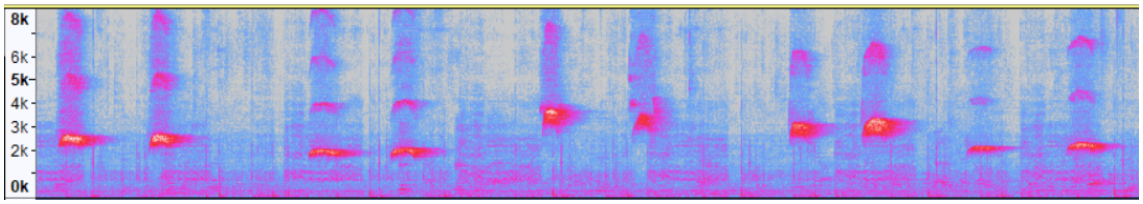


Figure 2.2.: Spectrogram of different whistles. First published in [1].

In their 2022 code release, B-Human [12] use a correlation-based method to capture the harmonic structure of whistles. Their system maintains reference samples of multiple whistle types, enabling not only detection but also identification of the specific whistle model. This approach accounts for variations in fundamental frequency across different whistles.

Neural network-based approaches have gained popularity in recent years. The HTWK Robots [13] apply a neural classifier directly to Fourier-transformed audio frames, replacing the frequency mask with a learned representation. A similar direction was explored in our 2022 code release [14], where a neural network was trained to recognize the characteristic spectral pattern of whistles. As shown in Section 2.7, this approach performed well in many cases but exhibited sensitivity to sound amplitude, motivating the integration of prior knowledge into the recognition process.

The idea of combining physically motivated models with neural networks has been explored in other domains. Forssell and Lindskog [15] demonstrated how semi-physical modeling can be used to constrain neural networks and improve robustness, using a water tank system as an example. Their work inspired the hybrid approach presented in this thesis, where domain knowledge about harmonic structure and noise characteristics is integrated with a neural classifier to improve reliability under challenging acoustic conditions.

Foundational concepts in digital speech and audio signal processing, such as spectral analysis, windowing, and noise robustness, are well established in the literature [16]. These principles underpin many of the methods used in whistle recognition and provide the theoretical basis for the feature representations employed in this work.

2.2. Dataset

Preparing the dataset is one of the most critical steps in developing a reliable whistle recognition system. Although machine learning models can in principle operate directly on time-domain signals, the characteristic structure of whistle sounds is more clearly expressed in the frequency domain, as illustrated in Figure 2.2. Empirical tests confirmed that spectral representations expose the relevant patterns more distinctly and lead to improved recognition performance. For this reason, the experiments in this thesis rely on a consistent audio preprocessing pipeline, which provides both time- and frequency-domain representations as needed. Subsection 2.2.1 describes this preprocessing pipeline. Subsection 2.2.2, Subsection 2.2.3, Subsection 2.2.4, and Subsection 2.2.5 present the datasets used for training, validation, robustness analysis, and evaluation.

2.2.1. Audio preprocessing and feature extraction

This subsection details the preprocessing steps applied to the raw audio stream recorded by the NAO robot.

Framing and windowing. Audio is captured at a sampling rate of 22 050 Hz. The continuous stream is segmented into windows of 1024 samples, corresponding to a frame length of approximately 46.4 ms. Adjacent windows overlap by 50 %, which increases temporal resolution while maintaining sufficient frequency resolution. Before computing the FFT,

each frame is multiplied by a Hamming window to reduce spectral leakage caused by the sharp boundaries of the rectangular window.

Spectral transformation. For real-valued input signals, the discrete Fourier transform exhibits Hermitian symmetry, meaning that the second half of the spectrum is fully determined by the first. Consequently, only the first 513 frequency bins, including the Nyquist frequency, are retained. The resulting complex coefficients $X(f)$ provide both magnitude and phase.

Magnitude representation. The magnitude is converted to a logarithmic scale in decibel using

$$20 \cdot \log_{10}(|X(f)|) \tag{2.1}$$

This log-magnitude representation emphasises tonal structures and makes the characteristic whistle pattern more visible, which was confirmed in preliminary experiments. The phase component encodes temporal alignment and is essential for direction estimation, but it did not provide additional discriminative power for whistle recognition and is therefore not used for that task.

Labeling support. The preprocessing pipeline is integrated into the *Auto-Label-Tool*, which supports both manual and model-assisted labeling. The tool visualises the log-magnitude spectrogram, allows annotators to listen to short audio segments, and displays all four microphone channels. To accelerate annotation, the most recent whistle recognition network is used to pre-label candidates: all detections with a confidence above 0.65 are marked as whistle events, while *negative_label* and the derived *non_whistle_label* explicitly identify regions without whistles. This ensures that the dataset contains both positive and informative negative examples.

To merge fragmented detections into coherent whistle events, the tool computes an *attention_length*. It identifies the longest sequence of consecutive detected whistles and multiplies its length by an *attention_multiplier* of 2.0, capped at a maximum of three frames. These parameters were chosen as conservative defaults for the datasets used in this work but may require adjustment for recordings with different acoustic characteristics. Final labels are exported together with the corresponding spectral features, ensuring consistent preprocessing across all datasets.

2.2.2. Training and Validation Dataset

The primary dataset for training, validating, and testing the whistle recogniser consists of audio extracted from log files recorded with NAO V6 robots during soccer games, test matches, competitions, and explicit whistle tests. To reduce redundancy, audio segments

containing only silence or walking noise were removed, as they do not contribute additional information for training. In contrast, segments that had previously triggered false positives or were likely to do so were deliberately retained to expose the network to challenging negative examples and improve robustness. After preprocessing and labeling, the dataset contains 369 680 non-whistle samples and 18 052 whistle samples, resulting in a class imbalance of approximately 20:1. This imbalance reflects the natural frequency of whistle events in real games; strategies for handling it, such as oversampling or undersampling, are discussed in Section 2.4.

The dataset is publicly available as the *Whistle Detection Dataset* [17]. Labels were created using an iterative annotation process. Initially, all samples were labeled manually. After the first model iteration, the *Auto-Label-Tool* [18] was integrated into the workflow. The tool uses the most recent network to pre-label whistle candidates, and human annotators verify and correct these suggestions, combining efficiency with human oversight.

2.2.3. Stress Test Dataset

The second dataset is used to tune parameters such as thresholds and to benchmark the whistle recognizer under particularly difficult conditions. It consists of three log files, each reflecting specific challenges for whistle recognition.

Show Game at Maker Faire Rome. The first log was recorded during a show game at the Maker Faire in Rome. Due to the cheering crowd directly around the field, this dataset produced more false positives than any other recording. In particular, at least one child's voice overlapped with the fundamental frequency of the whistle. To further increase the difficulty, we used the log file of the goalkeeper, who was constantly near the spectators. This log contains only three whistles (one at the start and two at the end of the game), reflecting the scarcity of whistle events in real matches. Its challenging acoustic conditions make it well suited for stress testing and ablation studies.

Laboratory Whistle and Noise Tests. The second log was recorded in our lab in three steps. First, we tested whistles where the airflow through the whistle was controlled, highlighting the variability of intensity and length depending on how the whistle was blown. Second, we tested different types of noise to identify triggers for false positives. Finally, we combined whistles and noise to create realistic benchmarks where the desired signal was masked. The noises were systematically categorized into four groups:

- **Robot sounds:** falling and standing up, servomotor movements.
- **Human sounds:** finger snapping, clapping, human whistling, referee whistle.

- **Spectator/environment sounds:** plastic bag for candy, cheering crowd, birds' twittering.
- **Mechanical tools:** circular saw, angle grinder, drilling machine, vacuum cleaner robot, hitting a pot.

This log was kept in its original format to allow reproducible replay in the simulator, ensuring identical conditions for every test run. It was used both for threshold tuning and for ablation studies to analyze the contribution of individual components.

RoboCup 2024 Test Game. The third log was recorded during a test game at RoboCup 2024. In this case, the referee's voice repeatedly triggered the whistle recognition system, highlighting the sensitivity of the recognizer to high pitch human sounds. Due to the clear recognizability of the voice, this log cannot be published. Nevertheless, it was used internally to study the robustness of the system and to adjust parameters.

Together, these three logs form the stress test dataset. They were used not only for parameter tuning but the first two also for ablation studies, allowing us to evaluate the impact of individual modules and to identify scenarios that lead to misrecognitions.

2.2.4. Simulator Integration

Unlike the primary training and validation dataset, which is stored as preprocessed audio, the stress-test dataset remains in log-file format. While this choice has certain limitations, it offers a decisive advantage for our evaluation: the logs can be replayed directly in our simulator using the same code that runs on the robot. This ensures that the behavior observed during testing closely matches the robot's real-world behavior, which would not be guaranteed if the audio were exported and processed externally, for example in Python.

Replaying log files also provides strong reproducibility, as identical conditions can be recreated for every test run, enabling fair comparisons across different recognition methods. The main drawback is that our simulator relies on framework-specific functions and is therefore not compatible with other software frameworks. Nevertheless, the ability to evaluate the actual on-robot implementation under controlled and repeatable conditions outweighs these limitations and makes log-based simulation the most appropriate choice for this dataset.

2.2.5. Data Quality Considerations

The quality of the recorded audio is influenced by several factors. First, the microphones of the NAO robots are not optimized for acoustic measurements and exhibit a limited frequency response, which can attenuate very high or very low signals. Second, the recording environments, typically indoor sports halls, introduce reverberation and uncontrolled background noise, resulting in a reduced signal-to-noise ratio. Although the

labeling of the *Whistle Detection Dataset* [17] is documented, ambiguous cases such as short or overlapping whistles remain challenging. Furthermore, the dataset reflects the natural imbalance between whistle and non-whistle events (approximately 20:1), which is realistic but complicates training. Variability in whistle types and intensities adds diversity and improves robustness, but also increases intra-class variance. Finally, while the use of log files in the simulator ensures reproducibility under identical conditions, the reliance on framework-specific functions limits compatibility with external systems. Overall, these factors highlight both the strengths and limitations of the dataset and motivate the hybrid recognition approach presented in this work.

2.3. Conventional Audio Processing

Conventional signal and audio processing methods are used to calculate a confidence value based on characteristic spectral patterns of whistles. This confidence is later fused with the confidence provided by the neural network, ensuring that domain knowledge and data-driven learning complement each other. In contrast to purely neural approaches, these methods incorporate prior knowledge about whistle characteristics and noise patterns, which improves robustness and reduces false positives in noisy environments. In the following subsections, we describe the spectral analysis, peak detection, overtone relations, and energy distribution that form the basis of whistle recognition.

2.3.1. Whistle Characteristics

Whistles exhibit a fundamental frequency typically in the range of 2–4 kHz, depending on the whistle model and recording conditions. The first overtone at approximately $2f_0$ is the most prominent and the only one used for validation in our system. Higher-order harmonics may be present but are typically weaker and therefore not considered for detection. Different whistle models produce slightly varying spectral patterns, but the harmonic relation remains consistent across all examined samples. Figure 2.2 illustrates spectrograms of several whistles, highlighting the recurring overtone structure: clear peaks appear at the fundamental frequency and at multiples thereof, while the surrounding spectrum remains comparatively quiet.

The harmonic relation can be expressed formally as shown in Equation 2.2, where each overtone f_n is an integer multiple of the fundamental frequency f_0:

$$f_n = n \cdot f_0, \quad n \in \mathbb{N},\ n \geq 2 \tag{2.2}$$

This relation explains why whistles are particularly well suited for detection in the frequency domain. Once the fundamental frequency f_0 is identified, the presence of its

harmonics at predictable positions serves as a strong indicator of a whistle event. For example, if f_0 is detected at 2.5 kHz, then the first overtone f_1 should appear near 5 kHz, and the second overtone f_2 near 7.5 kHz. Even if the fundamental frequency is partially masked by noise, the overtone pattern often remains visible, providing redundancy that increases robustness.

In practice, the overtone relation is used to validate candidate peaks in the spectrum. A detection is only considered reliable if at least one overtone can be confirmed within the expected range. This prevents false positives from isolated tonal peaks that may arise from crowd noise or mechanical sounds. As shown in Section 2.7, the physical model component reduces misclassifications in noisy environments. Since overtone validation is a central part of this component, it contributes substantially to this improvement.

Table 2.2.: Comparison of whistle and noise frequency ranges relevant for recognition.

Source	Typical Frequency Range	Notes
Whistle (fundamental f_0)	2–4 kHz	Clear tonal peak, stable pattern
Whistle (first overtone f_1)	4–8 kHz	Harmonic relation to f_0
Whistle (second overtone f_2)	6–12 kHz	Weaker but still detectable
Crowd cheering / children's voices	2–4 kHz	Overlaps with f_0, irregular pattern
Robot servomotors / impacts	Broadband	Non-harmonic, impulsive noise
Mechanical tools (drill, saw)	1–8 kHz	Strong harmonics, but not whistle-like
Environmental sounds (birds, clapping)	Broadband	Irregular, short bursts
Fan (NAO)	0–1 kHz	Low-frequency hum, typically steady

2.3.2. Acquiring Prior Knowledge

The recognition of whistles relies on characteristic patterns in the frequency domain. After buffering the audio stream into windows of 1024 samples at a sampling rate of 22050 Hz, a Hamming window is applied to reduce spectral leakage a rectangular window would have. The FFT is then computed, and only half of the spectrum plus the Nyquist frequency is retained due to symmetry. The magnitude spectrum is expressed in decibel (dB) as shown in Equation 2.1.

This logarithmic representation emphasizes tonal peaks and makes the whistle pattern more visible compared to linear amplitude. Although the phase component contains information about temporal alignment, it is not used here, as whistle recognition primarily depends on harmonic magnitude relations.

Peak Detection and Dynamic Boundaries Whistles exhibit a fundamental frequency f_0 in the range of 2–4 kHz, but the exact value varies between whistle models and depends on environmental conditions. To ensure robust detection, the search boundaries are adapted in two stages: first based on the observed frequency distribution, and second based on the spatial alignment of the sound source and robot.

Frequency-based calibration To adapt the whistle detection to varying acoustic conditions, the system maintains a buffer of recently detected whistle frequencies. From this buffer, the average frequency $\bar{f}$ as well as the minimum and maximum observed values are computed. These statistics are used to update the dynamic lower and upper frequency boundaries.

For each new detection, the system first determines how far the current boundaries deviate from the buffer average. If the minimum value in the buffer falls below the current lower boundary, the parameter *minFreqOffset* is increased accordingly. Otherwise, it is reduced but never below the predefined *permanentMinFreqOffset*. An analogous update is applied to *maxFreqOffset* using the maximum value in the buffer and the parameter *permanentMaxFreqOffset*. These offsets define how far the dynamic boundaries may shift relative to the average frequency.

If the newly detected frequency lies within the static whistle range $[f_{\min}, f_{\max}]$, the dynamic boundaries are updated as follows:

$$\begin{aligned} f_{\min}^{dyn} &= \max\Big(\min\left(\bar{f} - \tfrac{\delta_{\min}}{2},\ \bar{f} - \text{minFreqOffset}\right),\ f_{\min} - \text{permanentMinFreqOffset}\Big), \\ f_{\max}^{dyn} &= \min\Big(\max\left(\bar{f} + \tfrac{\delta_{\max}}{2},\ \bar{f} + \text{maxFreqOffset}\right),\ f_{\max} + \text{permanentMaxFreqOffset}\Big). \end{aligned} \tag{2.3}$$

Here, $\delta_{\min}$ and $\delta_{\max}$ denote the deviations between the average frequency and the current dynamic boundaries. The offsets *minFreqOffset* and *maxFreqOffset* allow the search range to expand when new whistle frequencies deviate from the expected pattern, while the permanent offsets ensure that the boundaries cannot collapse to unrealistically narrow intervals even under stable conditions.

Spatial adaptation In addition to frequency variation, the perceived whistle frequency is influenced by the spatial relation between the robot and the sound source. The parameter d_x (*distanceRatioX*) models this geometric effect and reflects how far the robot is

positioned along the field relative to the whistle source. Building on the dynamically calibrated boundaries from the previous subsection, these values are further adapted based on the robot's distance.

The static limits $f_{\min}$ and $f_{\max}$ represent the expected whistle frequency range derived from prior knowledge. The dynamic boundaries $f_{\min}^{dyn}$ and $f_{\max}^{dyn}$—computed during frequency calibration—are compared against these static limits. Depending on the robot's distance, the boundaries are widened proportionally to the deviation between the dynamic and static values:

$$\begin{aligned} f_{\min}^{adapt} &= f_{\min}^{dyn} - d_x \cdot |f_{\min}^{dyn} - f_{\min}|, \\ f_{\max}^{adapt} &= f_{\max}^{dyn} + d_x \cdot |f_{\max}^{dyn} - f_{\max}|. \end{aligned} \tag{2.4}$$

This corresponds directly to the implementation: if the current dynamic lower boundary lies above the static minimum, it is shifted downward by a distance-dependent amount. Similarly, if the dynamic upper boundary lies below the static maximum, it is shifted upward. The parameter d_x therefore controls how strongly the search range expands with increasing distance from the sound source.

This spatial adaptation widens the search range when the robot is further away, compensating for distance-related shifts in the measured whistle frequency. The final adapted boundaries $f_{\min}^{adapt}$ and $f_{\max}^{adapt}$ are then used for peak detection.

Overtone Relation A characteristic property of referee whistles is the presence of harmonic overtones. While the exact spectral shape depends on the whistle model and the acoustic environment, the first overtone at approximately twice the fundamental frequency is typically the most prominent and reliably detectable. Higher-order harmonics are often weak, especially in indoor environments or when multiple sound sources overlap. For this reason, the system focuses exclusively on the first overtone for validating whistle candidates.

After identifying the strongest peak within the adapted fundamental frequency range (as defined in Equation 2.3), the expected position of the first overtone is estimated as twice the peak frequency. To account for variations in whistle models and recording conditions, the search is not restricted to a single frequency bin. Instead, a frequency window around the expected overtone position is defined. The width of this window depends on the size of the fundamental search region and a scaling factor (*overtoneBorderMultiplier*), ensuring that the overtone search remains robust even when the fundamental frequency fluctuates.

Within this window, the system searches for the strongest spectral peak. If the magnitude of this peak exceeds the average spectral magnitude of the current frame, the presence of the first overtone is considered confirmed. This confirmation indicates that the detected fundamental peak is not an isolated tonal artifact but part of a harmonic structure char-

acteristic of a whistle. If no sufficiently strong overtone is found, the whistle candidate is treated as uncertain.

The harmonic validation described above is incorporated into the confidence calculation presented below, where confirmed overtones lead to a higher confidence value.

The estimation of ambient noise, which influences the final confidence through the dampening mechanism, is described separately in Section 2.5.

Voice Activity Suppression. High-pitched human voices, particularly from children or referees, frequently overlap with the fundamental frequency range of whistles and therefore pose a significant risk for false detections. To mitigate this effect, the system includes an optional mechanism that suppresses whistle candidates exhibiting speech-like characteristics. This mechanism relies on the Zero Crossing Rate (ZCR), which provides a lightweight indicator of voiced speech. For each whistle candidate, the ZCR is evaluated over the corresponding time window. If the ZCR pattern is consistent with speech rather than with a tonal whistle, the candidate is discarded before the confidence calculation. This prevents the system from interpreting short, high-pitched vocalizations as whistle events and increases robustness in acoustically challenging environments such as crowded sports halls.

2.3.3. Confidence Calculation

The confidence value c_{pm} quantifies how likely a detected spectral peak corresponds to a whistle. The amplitude of the whistle candidate is denoted by A, and the maximum magnitude of the current frame by A_{max}. The ratio A/A_{max} therefore weights the confidence according to the relative strength of the detected peak.

In addition to the amplitude, the local spectral gradients around the peak are evaluated. The gradients to the left and right of the peak, normalized by A_{max}, are denoted as γ_{max} and γ_{min}. These terms capture how steeply the spectrum rises and falls around the detected frequency: sharp, narrow peaks produce large gradients, whereas broadband or noisy signals result in flatter slopes. This makes the gradients a reliable indicator of tonal structure.

The harmonic validation described in the previous section is incorporated into the confidence calculation. A whistle candidate is considered *confirmed* if (1) the fundamental peak exceeds the average of the historical maximum magnitudes, and (2) the strongest peak in the overtone search window exceeds the mean spectral magnitude of the current frame. If both conditions hold, the harmonic structure is considered reliable.

The strategy to compute the confidence of the model-based detection is shown in Equation 2.5.

$$c_{pm} = \begin{cases} \frac{A}{A_{max}} \cdot \max\left(\left|\frac{\gamma_{max}}{A_{max}}\right|, \left|\frac{\gamma_{min}}{A_{max}}\right|\right), & \text{if the whistle candidate is confirmed} \\ \frac{A}{A_{max}} \cdot \min\left(\left|\frac{\gamma_{max}}{A_{max}}\right|, \left|\frac{\gamma_{min}}{A_{max}}\right|\right), & \text{otherwise.} \end{cases} \tag{2.5}$$

This formulation ensures that confirmed whistle candidates receive a higher confidence, while uncertain detections are penalized. The resulting value c_{pm} is already normalized and is passed directly to the hybrid fusion module, where it is combined with the neural network confidence.

2.4. Neural Network

In addition to the conventional audio processing methods described in the previous section, we employ a neural network to further improve the robustness of whistle detection. Both approaches operate on the same frequency-domain representation, namely the 513 log-scaled magnitudes obtained from the 1024-point FFT as defined in Equation 2.1. This ensures that the physical model and the neural network process identical input information while exploiting complementary strengths: the physical model incorporates domain knowledge about harmonic structure and noise characteristics, whereas the neural network learns discriminative spectral patterns directly from data.

The network is designed to run in real time on the NAO V6 robot, which imposes strict computational constraints. To meet these requirements, the architecture remains compact while still capturing the characteristic spectral signatures of whistles. A high-level overview of the model is shown in Figure 2.3. The architectural details are described in Subsection 2.4.1, followed by the training procedure in Subsection 2.4.2. Finally, Subsection 2.4.3 outlines the fine-tuning steps applied to the neural network.

2.4.1. Architecture

The neural network is designed to operate reliably under the real-time constraints of the NAO V6 robot, which requires that whistle detection fits into a 30 ms control cycle while all other robot tasks continue to run. To meet these requirements, the architecture remains compact while still capturing the characteristic spectral patterns of whistles.

The input to the network is identical to the representation used by the conventional audio processing pipeline: the 513 log-scaled magnitudes obtained from the 1024-point FFT as defined in Equation 2.1. This ensures that both approaches operate on the same frequency-domain information, enabling a consistent fusion of their confidence values. No additional normalization is applied, as the logarithmic scaling already stabilizes the dynamic range and highlights tonal structures.

The architecture consists of four convolutional blocks of two different types, followed by a dense output layer. The first three blocks share the same structure: a one-dimensional convolution, batch normalization, a leaky Rectified Linear Unit (ReLU) activation with a negative slope of $\alpha = 0.1$, max-pooling, and dropout with a rate of 0.3. These blocks use a kernel size of 5 and a stride of 2, which reduces the length of the spectral feature map while expanding the receptive field. The number of filters increases from 32 to 64 and finally to 128, allowing the network to extract increasingly complex spectral patterns.

The fourth block differs from the previous ones and consists only of a convolutional layer with an Exponential Linear Unit (ELU) activation. This block uses 64 filters, a kernel size of 5, and a stride of 1. The ELU activation provides a smooth nonlinearity without requiring batch normalization, reducing computational cost while still enabling the network to model subtle variations in the whistle spectrum [19]. The output of this block is flattened and passed to a dense layer with a single sigmoid unit, producing the neural network confidence value c_{nn}.

A high-level overview of the architecture is shown in Figure 2.3. The diagram illustrates the sequence of convolutional blocks, the reduction of the spectral feature-map length through striding and pooling, the increasing number of filters across blocks, and the final dense layer that maps the extracted features to a confidence value.

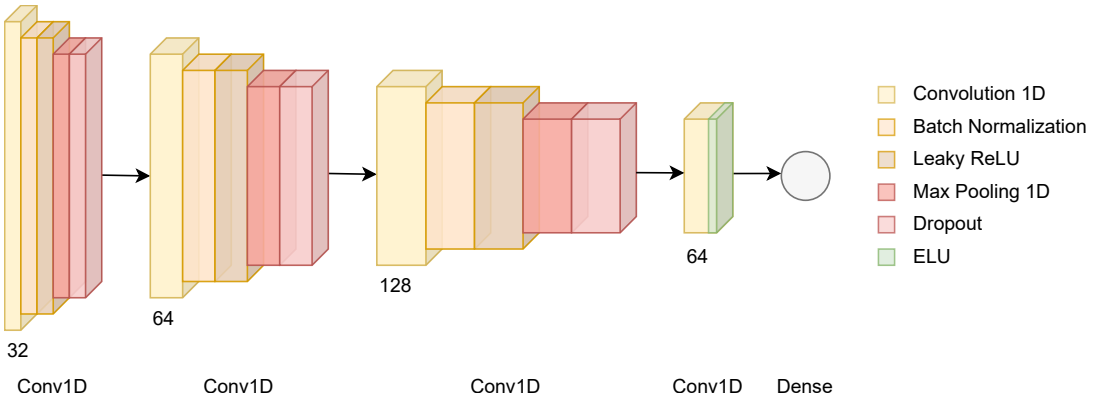


Figure 2.3.: High-level overview of the proposed whistle recognition neural network. The numbers indicate the number of filters in each convolutional block, and each downward step represents a reduction of the feature-map length caused by striding and max-pooling.

2.4.2. Training

The neural network was trained on the frequency-domain representation data described in Section 2.2, using the manually curated training and validation split. In contrast to the stress-test dataset, which is replayed in the simulator to ensure reproducibility, the training and validation data were extracted directly from log files and separated manually

to avoid temporal correlations and to ensure that the validation set contains challenging and previously unseen acoustic conditions.

Handling Class Imbalance

The dataset exhibits a natural imbalance of approximately 20:1 between non-whistle and whistle samples. To prevent the network from converging toward a trivial majority-class solution, we applied both oversampling and undersampling strategies, as described in Section 2.2. Oversampling was performed using a variant of Synthetic Minority Over-Sampling Technique (SMOTE) [20], specifically KMeansSMOTE, which first clusters the minority class using MiniBatchKMeans with $n_{\text{clusters}} = 32$ and then synthesizes new whistle samples within each cluster. This approach preserves the local structure of the minority class and avoids generating unrealistic examples.

Undersampling was implemented using a fast approximation of Edited Nearest Neighbors (ENN) [21] based on the FLANN library[22]. For each sample, the 256 nearest neighbours were queried, and samples whose neighbourhood was inconsistent with their label were removed. This procedure smooths the decision boundary and reduces noisy or ambiguous majority-class instances. Both steps were applied sequentially, resulting in a balanced and cleaner training set. The resampling steps were implemented using the Imbalanced-Learn library [23], which provides efficient and well-tested variants of SMOTE and ENN. This sequential combination follows the well-established idea of coupling SMOTE-based oversampling with ENN-style undersampling [24].

Training Configuration

The network was trained for a maximum of 45 epochs with a batch size of 2048. No additional normalization was applied beyond the logarithmic magnitude scaling of the FFT input, as this representation already stabilizes the dynamic range and highlights tonal structures. The optimizer used was Ranger [25], which combines Rectified Adam (RAdam) with the Lookahead mechanism to improve training stability and convergence. RAdam was configured with a learning rate of 10^{-3}, a warmup proportion of 0.25, a minimum learning rate of 10^{-6}, and a total step count of 20 000. Lookahead was applied with a synchronization period of 6 and a slow step size of 0.5, improving training stability and convergence.

The loss function was implemented as part of the `WhistleDetectorLoss` class. Although the final loss is based on binary cross-entropy between predicted and true confidence values, the class additionally computes precision, recall, accuracy, F-score, and a combined score across ten thresholds in the range $[0.1, 0.99]$. The threshold that maximizes the combined score is selected dynamically during training. These metrics are used for monitoring and model selection but do not directly influence the gradient.

Model selection was performed using a validation-based checkpointing strategy. The `ModelCheckpoint` callback monitored the validation F-score and stored only the best model. Early stopping with a patience of 20 epochs was applied to prevent overfitting and to restore the best-performing weights. All checkpoints were cleared before training to ensure a clean run.

Architecture Search

In addition to optimizing the training procedure, we also evaluated whether the network architecture itself could be improved automatically. For this purpose, a Bayesian optimization was performed over a search space that included the number of convolutional layers, filter sizes, kernel sizes, stride configurations, dropout rates, and activation functions. Each candidate architecture was trained and validated using the same procedure as the manually designed model.

The automatically learned architectures were generally smaller and computationally more compact than the final hand-crafted design. However, despite their reduced complexity, these architectures consistently showed lower robustness when evaluated on the diverse set of whistle and non-whistle sounds present in the dataset. In particular, they were more sensitive to different types of acoustic disturbances and failed to generalize reliably across the variety of challenging negative examples.

Based on these observations, the manually designed architecture was retained for all subsequent experiments, as it provided a more stable balance between efficiency and robustness.

Training Environment and Model Exploration

Training was conducted locally on two development machines: a Lenovo P53 laptop equipped with an NVIDIA RTX 3000 GPU and a Lenovo X1 Extreme featuring an NVIDIA GTX 1050 Ti Max-Q. Several architectural variants were explored during development, including deeper and smaller networks, wider convolutional layers, and models trained with curriculum learning. Although curriculum learning improved early convergence, it did not outperform the final architecture in terms of robustness and generalization. The chosen model provided the best trade-off between accuracy, noise robustness, and real-time feasibility on the NAO V6 platform.

Augmentation Strategy

No explicit data augmentation was applied. While KMeansSMOTE introduces synthetic samples for balancing the minority class, no additional spectral or temporal augmentations were used. This decision was motivated by the availability of real-world noise conditions

in the dataset and by the observation that artificial distortions can alter the harmonic structure of whistles in ways that do not reflect realistic acoustic environments.

The fine-tuning procedure applied after collecting additional RoboCup data is described in Subsection 2.4.3.

2.4.3. Fine Tuning

After the initial training of the neural network, the model is further refined using an iterative fine-tuning procedure. The goal of this process is not to retrain the entire network, but to adapt the decision-making component of the model to challenging acoustic conditions that were not fully represented in the original training set. This approach ensures that the recognizer improves over time while maintaining the stability and generalization capabilities of the base model.

Hyperparameter Optimization

For each negative dataset listed in the configuration, a small-scale hyperparameter optimization is performed before fine-tuning. This step adjusts the most influential training parameters:

- *learning_rate*
- *min_lr*
- *total_steps*
- *warmup_proportion*

The optimization is carried out using Bayesian search, where each candidate configuration is evaluated on a validation set that combines all previously processed training and validation data. This ensures that the selected hyperparameters improve the model not only on the current dataset but also in the broader context of all accumulated data.

Selective Fine Tuning of the Dense Layer

Following hyperparameter optimization, the model is fine-tuned on the respective negative dataset. To preserve the learned spectral representations, all layers except the final dense layer are frozen, ensuring that only the decision-making component is updated. This strategy reduces the risk of overfitting and allows the network to adapt specifically to challenging negative examples without altering the underlying feature extraction.

After each fine-tuning round, the updated model is evaluated against a baseline that contains all previously used training and validation data. Although the performance metrics (precision, recall, F-score, and the combined score) are compared to this baseline,

the updated model is not discarded if it performs worse. Instead, every iteration is fully logged, including both improvements and degradations, since later updates may compensate for earlier declines or improve different aspects of the model's behavior. Regardless of the outcome of the comparison, the newly trained model always becomes the baseline for the next iteration.

At the end of the entire procedure, the collected statistics allow the user to decide whether the final model should be kept or discarded based on the overall development across all iterations.

Iterative Improvement Across Datasets

This process is repeated for each negative dataset. In contrast to a strictly performance-driven update strategy, the model is not only advanced when a fine-tuning step yields an improvement. Instead, every iteration is carried out and the resulting model is saved, even if a temporary degradation occurs. This prevents the system from discarding potentially valuable adaptations and ensures that difficult or rare negative examples can still be learned over time.

Across multiple iterations, the model gradually becomes more robust, especially against challenging or ambiguous non-whistle sounds. Importantly, the network is never retrained from scratch, only the dense layer is adjusted, ensuring that the core spectral representation learned during the initial training remains stable while still allowing targeted adaptation to new negative data.

Data Augmentation and Balancing During Fine Tuning

Whenever possible, the negative datasets are augmented using the same over- and undersampling strategies described in Section 2.2. This increases the diversity of challenging examples and stabilizes the fine-tuning process. However, this kind of augmentation is not always feasible. For datasets that contain exclusively negative samples, oversampling cannot be applied, and undersampling may remove too many instances. In such cases, the fine-tuning step is performed without augmentation.

Overall, this fine-tuning strategy enables the model to adapt incrementally to new acoustic conditions while preserving the robustness and generalization of the original network.

2.5. Ambient Noise Detection

Whistle recognition becomes particularly challenging in environments with loud and irregular background noise, such as cheering crowds or children. With increasing volume, a growing number of spectral components exceed the amplitude of the whistle itself, mak-

ing it progressively more difficult to distinguish the whistle from the background. At sufficiently high noise levels, even humans are no longer able to reliably identify a whistle.

To quantify the amount of background noise, the system evaluates how many frequency bins exceed a predefined amplitude threshold. Let $\#f$ denote the number of all frequency bins of the magnitude spectrum and $\#f_c$ the number of bins whose amplitude exceeds the noise limit. The noise confidence σ is defined as:

$$\sigma = \frac{\#f_c}{\#f} \tag{2.6}$$

This measure reflects how strongly the spectrum is affected by broadband noise. In quiet environments, only a few bins exceed the threshold, resulting in a small value of σ. In contrast, cheering crowds or children produce many high-energy components across the spectrum, leading to a large value of σ and thus a reduced reliability of whistle detection.

Since whistle recognition is performed independently for each microphone channel, the noise level is also computed per channel. The resulting noise confidence is later used in the hybrid fusion stage (Section 2.6) to attenuate the overall whistle confidence when the background noise is high.

2.6. Hybrid Structure

Combining all acquired information is an essential part of every ensemble-based system. Our hybrid approach integrates the strengths of the neural network and the physical model by fusing their confidence values and subsequently adapting the result to the current acoustic environment. The process consists of two stages: (1) fusion of the whistle confidences and (2) dampening based on the ambient noise level.

$$c_{nn/pm} = \frac{w_{nn} \cdot c_{nn} + w_{pm} \cdot c_{pm}}{w_{nn} + w_{pm}} \tag{2.7}$$

The fused confidence $c_{nn/pm}$ is computed as a weighted average of the neural network confidence c_{nn} and the physical model confidence c_{pm}. The weights w_{nn} and w_{pm} determine which component is considered more reliable in the current situation. This allows prior knowledge to be incorporated directly into the fusion process, as shown in Equation 2.7.

$$c = c_{nn/pm} \cdot (1 - (w_\sigma \cdot \sigma)) \tag{2.8}$$

To ensure reliable whistle recognition under varying noise conditions, the fused confidence is attenuated using the ambient noise estimate σ introduced in Equation 2.6. The factor w_σ controls how strongly the noise level influences the final confidence. As shown

in Equation 2.8, higher noise levels reduce the confidence proportionally, preventing false positives in acoustically challenging environments.

2.7. Evaluation

The evaluation of our whistle recognition approach is structured into two main parts. First, we compare our method with existing solutions used by other SPL teams as well as with our previous neural network-based recognizer, see Subsection 2.7.1. This comparison provides a direct assessment of how the new approach performs under realistic conditions and in relation to established frameworks.

Second, we analyze how the individual components of our system influence the overall performance. By evaluating the physical model, the neural network, their combination, and the additional noise dampening mechanism, we can quantify the contribution of each part, described in Subsection 2.7.2. This analysis includes an ablation study on the laboratory and event datasets, allowing us to understand how selectively enabling or disabling components affects precision and recall under different acoustic conditions.

2.7.1. Comparison to State-of-the-Art Solutions

The first part of the evaluation compares our whistle recognition approach with three existing solutions: the frameworks of B-Human, the HULKs, and our previous neural network-based recognizer. The experimental setup is shown in Figure 2.4. Each approach runs on a separate robot positioned along the touchline with the center line as the midpoint. All robots are exposed to several noise sources, both close and far, without any whistle being blown. Here, *close* refers to noise sources placed directly in front of or around the robot, while *far* corresponds to the opposite side of the SPL field (approximately $6 \times 9\,\mathrm{m}$), representing the maximum practical distance within the field dimensions. This setup allows us to measure how often each system falsely detects a whistle.

Table 2.3 summarises the False Positive Rate for the tested noise types. The values show clear differences between the approaches. The B-Human recognizer produces false detections for several noise sources, including both levels of the cordless drill, the vacuum cleaner at close range, and the angle grinder at close range. This pattern indicates that the system reacts to a range of broadband and high-energy noise sources and does not suppress these disturbances reliably.

The HULKs' recognizer shows no false detections for any of the tested noise types. The same holds for our new hybrid approach, which also reports a False Positive Rate of 0.0 across all noise conditions. Both systems therefore avoid false detections in this part of the evaluation.

The old neural network approach performs better than B-Human for most noise types but still produces false detections for the angle grinder at close range. This suggests that

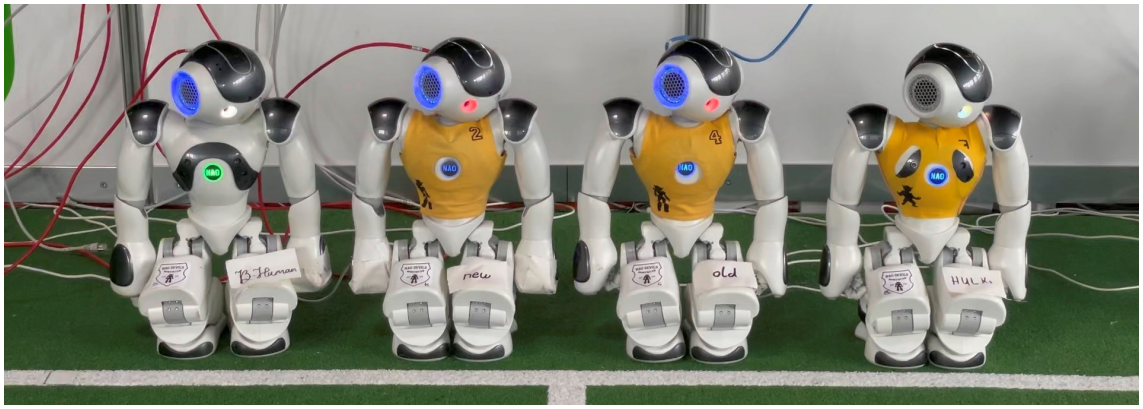


Figure 2.4.: Whistle comparison evaluation setup. Frameworks from left to right B-Human, the method presented here, our old approach and HULKs. First published in [1].

the old model is affected by high-energy broadband noise that overlaps with the frequency range of whistles.

Table 2.3.: A comparison of how different noise from different approaches is detected as a whistle (False Positive Rate). Lower is better.

	B-Human	HULKs	Old Approach	Our
Cordless Drill Level 1	0.2	0.0	0.0	0.0
Cordless Drill Level 2	0.2	0.0	0.0	0.0
Vacuum Cleaner (close)	0.6	0.0	0.0	0.0
Angle Grinder (far)	0.0	0.0	0.0	0.0
Angle Grinder (close)	1.0	0.0	0.8	0.0

To evaluate the True Positive Rate, we blew a whistle ten times at varying intensities and positions on an SPL field, first in a quiet environment and then with additional noise. The results are shown in Table 2.4. In a quiet environment, all approaches except B-Human achieve detection rates of 0.9 or 1.0. During the experiments, we observed that B-Human occasionally misses weaker whistles, which is consistent with the lower value in this setting.

When noise is present, the differences between the approaches become more apparent. With a vacuum cleaner placed on the far side of the field, the detection rate of the HULKs' recognizer decreases from 0.9 to 0.7. For B-Human, the value increases slightly from 0.6 to 0.7. Our old approach maintains a detection rate of 0.9, while our new hybrid method decreases from 1.0 to 0.9, resulting in the same value as the old approach under these conditions.

For the angle grinder at the far side of the field, B-Human, the old approach, and our new method all achieve a detection rate of 1.0. The HULKs' recognizer, however, drops to

0.2, indicating that its behavior, which avoids false detections in the previous experiment, also leads to missed whistles in this scenario.

Table 2.4.: A comparison of how different noise affects the performance of varying whistle recognition frameworks (True Positive Rate). Higher is better.

	B-Human	HULKs	Old Approach	Our
None	0.6	0.9	0.9	**1.0**
Vacuum Cleaner (far)	0.7	0.7	**0.9**	**0.9**
Angle Grinder (far)	**1.0**	0.2	**1.0**	**1.0**

Overall, the comparison shows that our new hybrid approach achieves a combination of properties that are distributed across the other systems. It avoids false detections in the noise scenarios tested, similar to the HULKs' recognizer, while maintaining high detection rates for actual whistles, comparable to our previous neural network. At the same time, it does not exhibit the drop in sensitivity observed for the HULKs' approach under certain noise conditions.

2.7.2. Influence of the detection components

The second part of the evaluation investigates how the individual components of our whistle recognition system contribute to the overall performance. To this end, we compare five configurations: our old neural network approach, the physical model alone, the neural network alone, the combination of both, and finally the full hybrid system including the ambient noise dampening mechanism. This allows us to quantify the contribution of each component and to understand how they interact under different acoustic conditions.

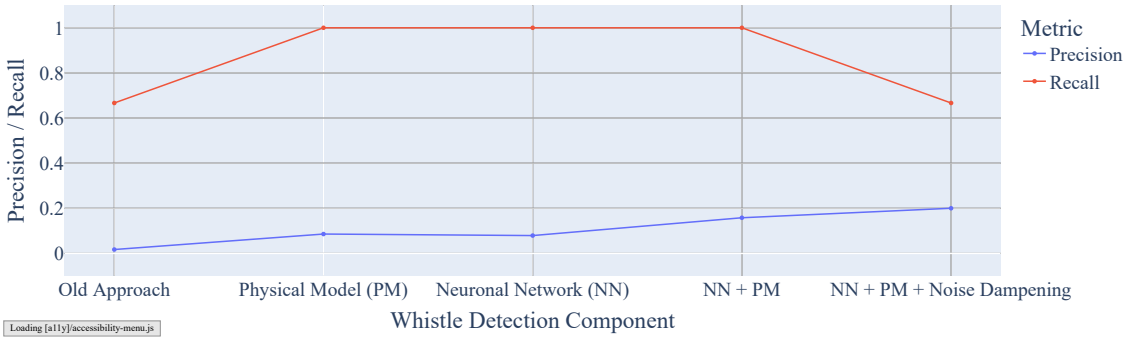


Figure 2.5.: Comparison of the influence of the different components of our approach on precision and recall during the Maker Faire Rome plus our old system since it proved to be promising in the previous tests. First published in [1].

Figure 2.5 shows the results for the Maker Faire Rome dataset. As described in Section 2.2, this dataset is particularly challenging due to the cheering crowd directly sur-

rounding the field. In this environment, our old approach achieves a precision close to zero, indicating that it frequently misclassifies crowd noise as whistles. Its recall remains moderate, meaning that it still detects a portion of the actual whistles but produces many false positives.

Both the physical model and the new neural network achieve a higher precision than the old approach, although the improvement remains modest in this dataset. Their recall increases noticeably compared to the old approach, indicating that both components detect whistles more reliably in this noisy environment. When the two methods are combined, the precision increases further, suggesting that their strengths complement each other in this environment.

The full hybrid system, which additionally includes the ambient noise dampening mechanism, achieves the highest precision among all configurations. Its recall is lower than that of the intermediate combinations, reflecting a more conservative detection behavior that prioritises avoiding false positives in very noisy segments. At the same time, its recall remains comparable to that of the old approach.

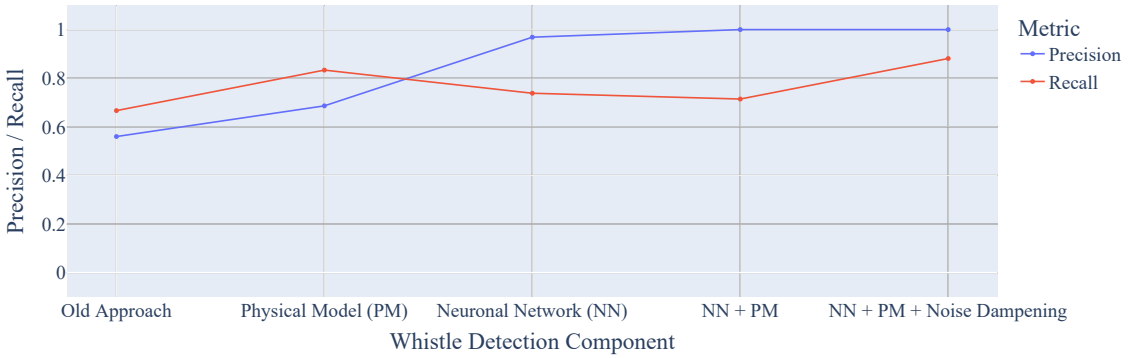


Figure 2.6.: Comparison of the influence of the different components of our whistle recognition on precision and recall under lab conditions plus our old system since it proved to be promising in the previous tests. First published in [1].

The results for the laboratory dataset are shown in Figure 2.6. In this more controlled environment, all components of the new system achieve higher precision and recall than the old approach. The new neural network reaches a higher precision than the physical model, indicating that it rejects non-whistle events more effectively. The physical model, in turn, achieves a higher recall than the neural network, suggesting that it is more sensitive to weaker whistles. Their combination yields a detector with very high precision and a recall that remains close to that of the individual components.

When all three components are combined, the recall increases further while the precision remains at the maximum level. This shows that the ambient noise dampening mechanism

reduces false positives without limiting the system's ability to detect whistles under quieter laboratory conditions.

Overall, the ablation study shows that each component contributes in a distinct way to the final performance. The physical model improves detection stability in noisy environments, the neural network increases precision by rejecting non-whistle events more effectively, and the noise dampening mechanism reduces false positives in acoustically challenging segments. The full hybrid architecture combines these effects and achieves a balanced trade-off between precision and recall across the tested datasets.

3. Whistle Localization

Robots are designed to serve a specific purpose, such as playing soccer in our case. However, in addition to their main task, they have to react to specific events, such as a whistle. With our whistle recognition [1], we demonstrated a robust method for detecting whistles reliably. This resulted in the necessity of discarding whistles from neighbouring fields since reacting to referee commands from other fields is not beneficial for the game. Whistle recognition was first introduced in the SPL in 2015 to indicate the start of a game. By now, a whistle indicates the beginning of a kickoff, a goal, and the end of the game [26]. In addition, official RoboCup events provide multiple soccer fields close to each other for one league, with different leagues possibly nearby. With the implemented time delay of the game controller signal sent to the robots, recognising and reacting to the whistle is necessary. However, falsely detecting a whistle from another field can currently only be noticed when the expected game controller signal is missing after the set delay. So, the problem addressed here is not the recognition of whistle events but distinguishing between relevant whistles and whistles from other fields that can be labelled as noise.

The described task contains two parts: estimating the sound source's direction and distance, described in Section 3.3 and Section 3.4. This information, combined with knowledge about the soccer field and the robot's location, can be used to decide whether the detected whistle is relevant. In Section 3.1, we give an overview of existing solutions for Sound Source Localization (SSL). One of the most common difficulties in distance estimation is overfitting to a specific room. This is unfortunate for soccer robots which are supposed to play everywhere, just like humans. Therefore, the goal is to make the localization of the whistle as independent from the room as possible. The issue is also described in detail in Section 3.1, which leads to the dataset described in Section 3.2.

Another main criterion for RoboCup teams is the runtime. To maintain a fluent game, added tasks should not block or slow down the motion, vision or behavior. Additionally, according to the rules of the SPL league, all robots have to be autonomous and are not supposed to be altered, so they must use the integrated hardware of the NAO robot. So, in the evaluation, Section 3.5, not only the performance in terms of correct detections is analyzed, but also if the execution time still fits in the set timeframe of 30 ms for one cognition frame.

3.1. Related Work

SSL has been investigated in a wide range of contexts, but most established approaches are designed for human speech. Consequently, they operate in the spectral range of voiced speech [27–30], which spans approximately 50 Hz to 450 Hz with harmonics up to 4 kHz [16]. In the SPL league, however, robots do not process speech; instead, audio perception focuses exclusively on whistle detection. Whistles exhibit fundamental frequencies between 2 kHz and 3.5 kHz with only a few overtones, resulting in a signal structure that differs substantially from speech. Combined with the NAO robot's compact and acoustically suboptimal microphone arrangement, see Section 1.1, this creates a setting that diverges from classical microphone-array scenarios.

3.1.1. SSL on the NAO Platform

Li et al. [28] present one of the few SSL approaches explicitly developed for the NAO robot. Their method estimates the direct path using phase and level differences between two diagonal microphone pairs in the Short-Time Fourier Transform (STFT) domain. The authors identify several hardware-related challenges: the microphones are placed close to internal cooling fans, which introduce stationary noise; the available processing power is limited; and the built-in microphones provide only modest signal quality. Although newer NAO versions employ omnidirectional microphones, the physical layout and spacing remain unchanged.

To address these limitations, Li et al. outsource computation to an external machine and incorporate visual information. Outsourcing is prohibited by current SPL rules [26]. Vision, in contrast, is permitted but impractical for whistle localization. As discussed in Section 1.1, the NAO camera has a limited field of view and, due to the robot's height and orientation constraints, cannot simultaneously observe the game situation and the referee. Moreover, whistles do not necessarily produce visible cues, making vision an unreliable modality for this task. Their evaluation also focuses on a single human speaker positioned at table height, which does not reflect SPL-specific requirements such as whistle localization near floor level or dynamic in-game conditions.

3.1.2. Machine-Learning Approaches

Kushwaha et al. [31] propose a Convolutional Recurrent Neural Network (CRNN)-based system for distance estimation and SSL. A central challenge in their work is overfitting to specific acoustic environments. Two of their datasets are synthetic, and several real datasets contain only a single room or church. Other datasets include more rooms but without describing their acoustic properties. Since room geometry and materials strongly influence reverberation [32], models trained on limited room diversity tend to generalize

poorly. This observation is relevant for whistle-based SSL, where robots operate in a wide variety of indoor environments with minimal calibration.

3.1.3. SSL in Domestic Robotics

A substantial body of SSL research originates from domestic service robotics, where environments are comparatively static and predictable [27, 33]. Cho et al. [27] accelerate Steered-Response Power-Phase Transform (SRP-PHAT) by clustering the search space and evaluate their method in a classroom. Sewtz et al. [33] combine audio and vision in conversational scenarios with two speakers within the robot's field of view. These systems achieve good performance but rely on assumptions that do not transfer to robot soccer use cases, where robots must localize short whistle signals under rapid motion, changing orientations, and minimal calibration.

3.1.4. Room-Acoustic Cues and DRR-Based Methods

Room-acoustic information can support localization, particularly in reverberant environments. In psychoacoustics, the Direct-to-Reverberant energy Ratio (DRR) is known to be one of the dominant cues for human distance perception [34]. Although our work does not aim to model human perception, this connection illustrates why DRR frequently appears in technical SSL literature and why room-dependent propagation effects are relevant for whistle localization.

Zohourian and Martin [30] show that the DRR provides a useful cue for several SSL tasks. Their later work [35] demonstrates that accurate inter-microphone distances and HRTF-based modeling significantly influence localization accuracy, especially for binaural systems.

Building on this line of research, Zohourian [36] investigates distance estimation using binaural hearing aids and deep neural networks. Their results show that DRR-based cues remain informative even under strong room-dependent variability, provided that the model learns the underlying microphone relationships directly from data. This supports our decision to rely on a learning-based approach that implicitly captures the NAO's microphone geometry and its room-dependent propagation characteristics, rather than depending on explicit HRTF measurements or manual calibration.

3.1.5. Relevance to This Work

Existing SSL approaches either rely on speech-specific features, assume static indoor settings, or require hardware configurations not available on the NAO. In contrast, our goal is to perform whistle localization under SPL constraints: no external computation, no reliance on vision, minimal calibration, and highly dynamic environments. These require-

ments motivate the development of a learning-based method tailored to the NAO platform and the acoustic characteristics of whistle signals.

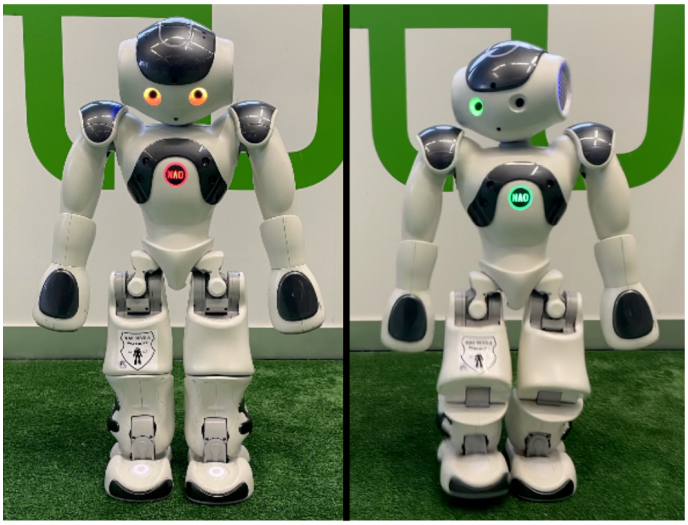


Figure 3.1.: The robot is in a penalised position (left) and walking (right). First published in [2].

3.2. Dataset

Accurate whistle localization requires training data that provide not only the whistle signal itself but also reliable spatial labels. In contrast to whistle detection, localization depends on inter-microphone phase differences and on the acoustic characteristics of the NAO head, including its HRTF. Furthermore, as discussed in Section 3.1, localization models are prone to overfitting to specific rooms. A dedicated dataset covering a wide range of environments, distances, and angles is therefore essential.

This section describes the recording setup, preprocessing steps, labeling procedure, evaluation dataset, and final dataset composition used for training the direction and distance models.

3.2.1. Recording Setup

All recordings for the whistle localization dataset were conducted on NAO V6 robots. The hardware characteristics of the platform, including the microphone configuration and the acoustic properties of the robot head, are described in Section 1.1 and are therefore not repeated here. These properties are relevant because the HRTF of the NAO head directly shapes the spatial cues contained in the recorded whistle signals.

To ensure stable and reproducible recording conditions, all robots were placed in the penalised position, shown in Figure 3.1 (left). This posture prevents any movement of the head or body and therefore avoids inaccuracies in the spatial labels. Before each session,

the microphone status of the robot was verified. Only robots with at least three functioning microphones were used, and recordings with more than one defective microphone were discarded. In total, 142 recording sessions were collected using 15 different robots acquired between 2019 and 2023. Four sessions originate from robots with a single defective microphone. These were retained to increase the diversity of the dataset and to test the effect it has on the recognition.

The whistle was blown by two persons of similar height of around 170 cm, one male and one female. A hand-held whistle with an integrated air reservoir was used, ensuring that the whistle sound was generated independently of lung capacity. The loudness of the whistle was varied naturally by pressing the air reservoir with different force and speed, resulting in realistic variations in sound pressure level. For each recording position, the whistle was blown five times at different heights to capture variations in the vertical placement of the sound source. These heights spanned the range from an outstretched arm above the head to an outstretched arm next to the body, resulting in three intermediate positions at eye level, chest level, and hip level. These five whistle heights are illustrated in Figure 3.2. The figure provides a visual reference for the vertical positions used during data collection and highlights the range of typical whistle heights observed in RoboCup matches. This accounts not only for how a referee holds the whistle but also for differences in referee height.



Figure 3.2.: Illustration of the five whistle heights used during data collection.

As shown in Figure 3.7, the whistle emits sound through a single opening, producing a directed acoustic signal. To capture this directionality, the whistle was rotated and tilted differently for each of the five blows at a given position, as the tilt of the whistle in a RoboCup game is inherently random.

The spatial coverage around the robot depended on the setup and the geometry of the recording environment. In large open spaces with precisely placed floor markings and a single recording robot, primarily the Arena (sometimes referred to as *Lab* in the label files), the robot was positioned in the center circle, and the person holding the whistle walked around it in discrete angular steps. This allowed a full 360 ° coverage with a 22.5 ° angular resolution. When the setup changed to four recording robots (one in the center circle, one on the penalty cross, one between the two, and one on the goal line), walking around the robots was no longer suitable. As a result, the robots were rotated instead, and the angular resolution was adjusted from 22.5 ° to 45 °, as placing the robots accurately enough for the finer resolution would not have been feasible.

The data recording in the MMI Lab followed the same pattern as the multi-robot setup in the Arena, but with only one robot. This setup was chosen to use the available space efficiently and to enable recordings at larger distances. Both the reverberant and the curtain-damped configurations of the MMI Lab were recorded with a 45° angular resolution at distances of 1.0 m, 2.0 m, 4.5 m, 5.0 m, and 6.0 m.

In narrow environments such as hallways, staircases, or the basement (which is also shaped like a hallway), the available space restricted the feasible recording method. Additional constraints included whether floor markings were reasonable, how much the setup would disturb other employees, and how accurately the robot could be positioned for the chosen angular resolution. In these cases, the robot was rotated instead of the person walking around it, and a 90 ° angular resolution was used. This approach was also applied in rooms where the robot needed to be placed on a table, such as the meeting room, the conference room, the office, or the lecture hall.

The distance between the robot and the whistle was kept constant within each recording session. Depending on the room size and geometry, distances ranged from 0.7 m in the Arena or outdoors to more than 15 m in the hallway. In some scenarios, multiple robots were placed in a row to record several distances simultaneously. This setup was particularly useful in large indoor environments and ensured consistent spatial alignment across distances.

Table 3.1 summarises all recorded rooms, the corresponding distances, and the angular resolutions used in each environment.

3.2.2. Labeling and Labeltool

The goal of the whistle localization dataset is to provide reliable ground-truth labels for the direction, distance, and room in which each whistle event was recorded. These labels form the basis for training and evaluating the neural networks described in Section 3.3 and Section 3.4. Since localization performance is highly sensitive to the quality and con-

Table 3.1.: Overview of recorded rooms, distances, and angular resolutions.

Room	Distances [m]	Angular Resolution
Arena (Lab, regular grid)	0.7, 1.9, 3.1, 4.5, 6.0, 7.7, 9.0, 10.0	45°
Arena (Lab, fine grid)	1.9, 4.5, 6.0, 10.0	22.5°
MMI Lab (reverberant)	1.0, 2.0, 4.5, 5.0, 6.0	45°
MMI Lab (with curtains)	1.0, 2.0, 4.5, 5.0, 6.0	45°
Hallway upstairs	3.0, 15.3	90°
Hallway downstairs	12.4	90°
Basement	7.0	90°
Staircase	1.0, 2.5	90°
Meeting Room	1.6, 3.5	90°
Conference Room	4.1	45°
Office	1.3	45°
Audimax Hamburg	5.4, 8.5	90°
Outdoor Field (Rome)	0.7	90°

sistency of the labels, particular care was taken to design a transparent and reproducible labeling process.

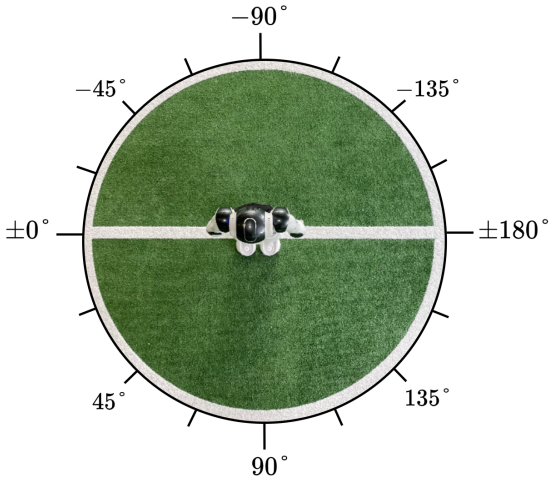


Figure 3.3.: NAO robot with azimuth labels. First published in [2].

Label definition and structure. Each whistle event in the dataset is annotated with three labels:

- **Direction label:** The azimuth angle of the whistle source relative to the robot's forward-facing direction, see Figure 3.3. The angles follow the conventions defined

during data collection (see Subsection 3.2.1) and are stored as a single scalar value per event. The angle sequences for each recording session were manually documented during data collection and transferred into a Python dictionary used by the labeling tool.

- **Distance label:** The distance between the robot and the whistle source, expressed in meters. Distances are assigned strictly on a per-event basis. In most recordings, the distance remains constant throughout the file, so all events receive the same numerical value. However, some sessions, particularly in the evaluation dataset, contain multiple distance segments within a single recording.
- **Room label:** A categorical label indicating the recording environment, such as *Arena (Lab)*, *MMI Lab (reverberant)*, or *Hallway upstairs*. These labels are used to analyze potential room-specific biases, as discussed in Section 3.1.

All three labels are constant within a whistle event and do not vary across the frames that compose the event. This ensures that the neural networks learn from stable, event-level annotations rather than frame-level fluctuations.

Whistle event extraction. Before assigning labels, we extract the whistle events from the continuous audio recordings. We use the whistle detection network described in Chapter 2 with a deliberately low detection threshold to ensure that the direct sound at the onset of each whistle is captured reliably. All frame-level preprocessing steps, like framing, windowing, FFT computation, and the retention of the 513 positive-frequency bins, follow the pipeline described in Subsection 2.2.1. Only localization-specific adaptations are applied during event extraction.

For each recording, we identify the longest detected whistle sequence as the reference sequence X_{ref}. We compute the attention length as

$$a_{\text{len}} = |X_{\text{ref}}| \cdot a_{\text{mul}},$$

with an attention multiplier of $a_{\text{mul}} = 1.25$. We then group frames into a whistle event as long as the number of consecutive non-whistle frames remains below a_{len}. This ensures that each event forms a coherent temporal unit while preserving the direct sound that is essential for localization.

Preprocessing and data representation. For localization, both magnitude and phase are extracted from the 513 positive-frequency bins of each microphone channel. In contrast to whistle detection, which uses only the log-magnitude representation, localization models require phase information to compute inter-microphone phase differences. The FFT symmetry and all frame-level preprocessing steps follow Subsection 2.2.1 and are not repeated here.

For each whistle event, we store two complementary representations:

1. **Time-domain frames:** The raw audio frames used by the whistle detector. These allow future work to apply alternative preprocessing or feature extraction techniques.

2. **Frequency-domain representation:** The magnitude and phase values of the 513 positive-frequency bins, which form the primary input for the localization networks.

Temporal augmentation. To increase robustness against small variations in the detected onset of a whistle, we apply a temporal augmentation step to each extracted event. After forming the whistle event X_{w}, we generate two additional variants by shifting the event window forward by one and by two frames, respectively. Together with the original event, this results in three temporally aligned versions of the same whistle. This augmentation ensures that the localization models remain stable even if the whistle detector identifies the onset slightly later than the true physical onset. We use this augmentation exclusively for distance estimation.

Label assignment and tooling. The labeling tool assigns direction, distance, and room labels to each extracted whistle event by looking up metadata associated with the filename of the recording. Three lookup tables provide this metadata:

- **Direction labels:** A list of azimuth angles, one entry per whistle event, reflecting the angular sequence used during the recording session.

- **Distance labels:** A list of distances, also one entry per whistle event. In most recordings, all entries are identical because the distance remained constant. However, some sessions contain multiple distances within a single file. This occurs when the robot was not restarted between distance changes, resulting in several distance blocks being stored in the same audio file. The labeling tool therefore uses the full per-event distance list rather than assuming a single value.

- **Room label:** A single categorical identifier describing the recording environment, shared by all events in the file.

All metadata was documented during data collection and transferred into the lookup tables used by the tool. After whistle events are extracted, the tool verifies that the number of events matches the lengths of the direction and distance lists as a sanity check. Each event is then assigned its corresponding direction, distance, and room label, ensuring that the final dataset remains consistent with the original recording protocol.

3.2.3. Evaluation Dataset

In addition to the training and test subsets described above, a separate evaluation dataset was recorded to assess the generalization of the localization models under conditions that were not part of the original data collection. This evaluation dataset consists of three components: (i) standing recordings on the RoboCup field, (ii) walking-in-place recordings on the same field layout, and (iii) additional recordings in previously unseen indoor environments. All recordings in this evaluation dataset were created after the models were trained and therefore provide an external benchmark for assessing real-world generalization.

Field recordings. A dedicated recording setup was created on the RoboCup field using three robots (K, B, and V) placed at fixed positions, as shown in Figure 3.4. Around each robot, the whistle was blown at predefined positions derived from the official field dimensions. The sound source was held at the five whistle heights described in Subsection 3.2.1. These recordings were performed in two variants: standing and walking-in-place.

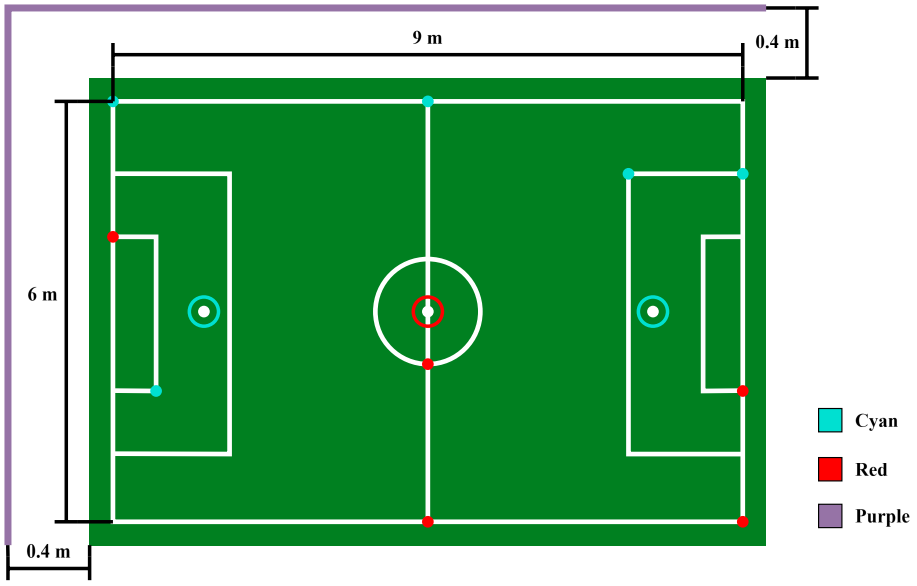


Figure 3.4.: Soccer field according to the rules [26] with walls coloured in purple. The circles mark the positions of the robots K, B, and V, while the dots mark the positions of the sound sources. Red points were used for both standing and walking-in-place recordings, while cyan points were used only for the standing evaluation.

Standing evaluation. The standing evaluation follows the same recording principles as described in Subsection 3.2.1, with the difference that the whistle was blown ten times per position and height. A total of ten sound-source positions was used: five red positions that were also included in the walking-in-place evaluation, and five additional cyan positions that were recorded only in the standing evaluation, as their inclusion

during walking-in-place would have made the procedure disproportionately complex and error-prone. The robots were placed at the center circle (red circle) and at the two penalty marks (cyan circle), and their fixed posture allowed multiple recordings to be taken in parallel. The sound source was positioned at the locations shown in Figure 3.4, comprising the lower T-junction, the intersection of the center circle and halfway line, the lower right corner of the field, the right goal post, and the upper left goal post (red dots), as well as the lower left goal-area corner, the upper left corner, the second T-junction, the upper right penalty-area corner, and the intersection of the penalty area and goal line on the right side (cyan dots).

Walking-in-place evaluation. The walking-in-place evaluation uses the same field layout as the standing evaluation but introduces controlled robot motion. During each whistle event, the robot executed a walking-in-place motion while the head performed a horizontal sweep from $-90\,^\circ$ to $+90\,^\circ$ at a fixed elevation of $17\,^\circ$ and a velocity of $80\,^\circ\,\mathrm{s}^{-1}$. In practice, the robot inevitably exhibits small unintended rotations and slight lateral drift during this motion, even when supported. These effects introduce additional sources of uncertainty that are not present in the standing evaluation. Because each sound-source position required a separate recording session, the combination of robot motion, head movement, and repeated session restarts would otherwise have amplified pose inconsistencies and made the evaluation substantially more labour-intensive. The reduced set of five red positions was therefore chosen deliberately: the goal of the walking-in-place evaluation was not to achieve dense spatial coverage, but to obtain a coarse indication of whether models trained exclusively on stationary data generalise to moving conditions, until data generated under real on-field operation becomes available.

Angle computation. For all field-based evaluations, the direction labels were computed analytically based on the exact robot and sound-source coordinates shown in Figure 3.4. The angles were calculated using GeoGebra to ensure consistent geometric reference values across all recordings. This computation does not eliminate placement inaccuracies; instead, such inaccuracies were minimised during data collection by carefully controlling how the whistle was held and oriented at each position. The analytically derived angles were then used as direction labels for the evaluation dataset.

Recordings in previously unseen rooms. To further evaluate the generalization of the localization models, additional recordings were created in indoor environments that were not part of the original dataset. The recording procedure followed the same principles as the main dataset: the robot was placed in the penalised position, the whistle was blown at five heights, and the sound source was positioned at discrete angular intervals around the robot. The angles used in these recordings cover the full $360\,^\circ$ range in steps of $45\,^\circ$ (i.e. $0\,^\circ$,

±45 °, ±90 °, ±135 °, and ±180 °). These environments were intentionally chosen because they differ acoustically from all rooms in the training and test datasets, making them valuable for assessing robustness to unseen reverberation patterns and room geometries.

Purpose of the evaluation dataset. The evaluation dataset provides an external benchmark for assessing the generalization of the localization models. It enables a systematic analysis under idealised (standing), dynamic (walking-in-place), and acoustically unfamiliar (unseen rooms) conditions. Together, these recordings form a comprehensive basis for evaluating the real-world performance and generalization capabilities of the proposed whistle localization system.

3.2.4. Dataset Composition

The complete whistle localization dataset consists of all recordings collected across the different environments described in Subsection 3.2.1. After labeling and event extraction, the data were split into three disjoint subsets: training, test, and evaluation. This split follows the structure of the publicly released dataset on TUDOdata [37], where the raw audio files (`.wav`) are organised into three top-level folders:

- **train/** – recordings used for training the direction and distance models,
- **test/** – recordings used for hyperparameter tuning and model selection,
- **eval/** – recordings used exclusively for the final evaluation.

Each folder contains the original session-level recordings. The labeling tool described in Subsection 3.2.2 processes these files and converts them into event-level samples with direction, distance, and room labels. This ensures that the publicly released dataset can be reproduced exactly from the raw audio.

Dataset size. The training set contains 125 recording sessions with a total of 15 312 whistle events. The test set consists of 12 sessions with 1272 events, and the evaluation set contains seven sessions with 1050 events. Only five of these evaluation sessions are included in the public release. Two sessions recorded in the Arena could not be published because speech from bystanders was unintentionally captured, and releasing these recordings would violate personal rights.

Rooms and distances. An overview of the rooms and distances represented in each subset is provided in Table 3.2–Table 3.4. To ensure readability, the training, test, and evaluation subsets are listed separately.

Table 3.2.: Training set: rooms and distances.

Rooms	Arena; Audimax Hamburg; Basement; Hallway upstairs; MMI Lab (reverberant); MMI Lab (with curtains); Meeting Room; Staircase
Distances [m]	0.7, 1.0, 1.6, 1.9, 2.0, 2.5, 3.0, 3.1, 3.5, 4.5 5.4, 6.0, 7.0, 7.7, 8.5, 9.0, 10.0, 15.3

Table 3.3.: Test set: rooms and distances.

Rooms	Arena; Hallway downstairs; MMI Lab (reverberant); MMI Lab (with curtains); Outdoor Field (Rome)
Distances [m]	0.7, 1.9, 3.1, 4.5, 5.0, 7.7, 9.0, 12.4

Special cases. Most recordings contain a single fixed distance per session. However, some sessions in the evaluation set include multiple distance segments within the same file. This occurred when the robot was repositioned without restarting the recording. The labeling tool handles these cases by assigning a distance value to each whistle event individually, ensuring that the final dataset remains consistent and fully traceable.

A total of four sessions were recorded with one defective microphone. These sessions are included intentionally to increase the diversity of the dataset and to evaluate the robustness of the localization models under realistic hardware conditions.

Overall, the dataset provides a broad and well-balanced distribution of rooms, distances, and angular resolutions, forming a comprehensive basis for training and evaluating the whistle localization models presented in the following sections.

3.3. Whistle Direction

The goal of the direction estimation module is to determine the azimuth angle of the whistle source relative to the robot. In contrast to general SSL scenarios, elevation cues are negligible in RoboCup environments, as the whistle is always held approximately at human torso or head height. The azimuth convention used in this work is illustrated in Figure 3.3.

Table 3.4.: Evaluation set: rooms and distances (publicly released sessions).

Rooms	Arena; Conference Room; Office
Distances [m]	0.8, 1.3, 3.0, 3.2, 3.3, 4.1, 4.4, 4.6, 5.4, 7.2, 7.8, 8.0, 8.3

3.3.1. Direct Sound and Reverberation

Accurate direction estimation relies primarily on the direct sound component. As shown in Figure 3.5, the direct path (a) carries the spatial cues required for localization, while reflections (b) introduce misleading phase and level differences. Based on measurements conducted in our Arena, we found that the first two frames after whistle onset contain predominantly direct sound. To compensate for occasional detection inaccuracies, we use the first three frames (93 ms) for direction estimation. This ensures that the model receives a stable representation of the direct sound while minimising the influence of reverberation.

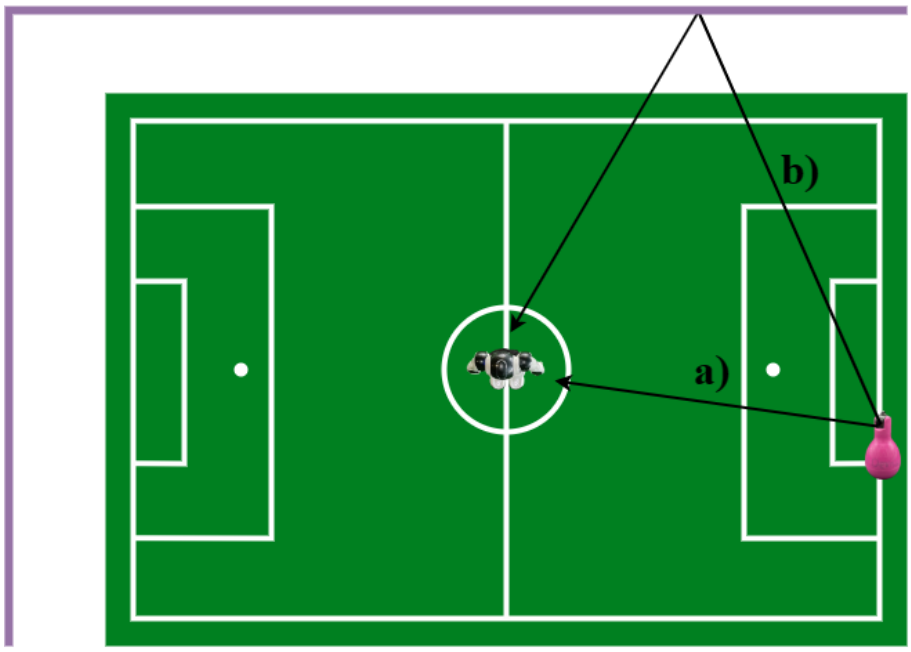


Figure 3.5.: Illustration of direct sound (a) and reflected sound (b) in our Arena (Lab). The NAO robot is positioned in the center circle, and the L-shaped walls surrounding the field are shown in purple. First published in [2].

3.3.2. Input Representations

The NAO robot is equipped with four microphones arranged asymmetrically around the head, which introduces characteristic inter-microphone phase and level differences. This is shown in Figure 1.2 in Section 1.1. These differences encode the direction of arrival and form the basis of our localization approach.

For each whistle frame, we extract two complementary representations:

- **Phase input (frequency domain):** The phase of the STFT, with shape $(513, 4)$, capturing inter-microphone phase differences. Prior work [27, 28] has shown that phase information is highly informative for azimuth estimation.

- **Level input (time domain):** The amplitude levels of the raw audio frame, with shape $(1024, 4)$. These levels encode Inter-Microphone Level Differences (ILD), which provide additional directional cues, especially for higher frequencies and near-field effects.

3.3.3. Neural Network Architecture

The direction estimation network processes the phase and level inputs using two parallel convolutional backbones. Each backbone extracts a 256-dimensional feature vector from its respective input. The architecture is illustrated in Figure 3.6.

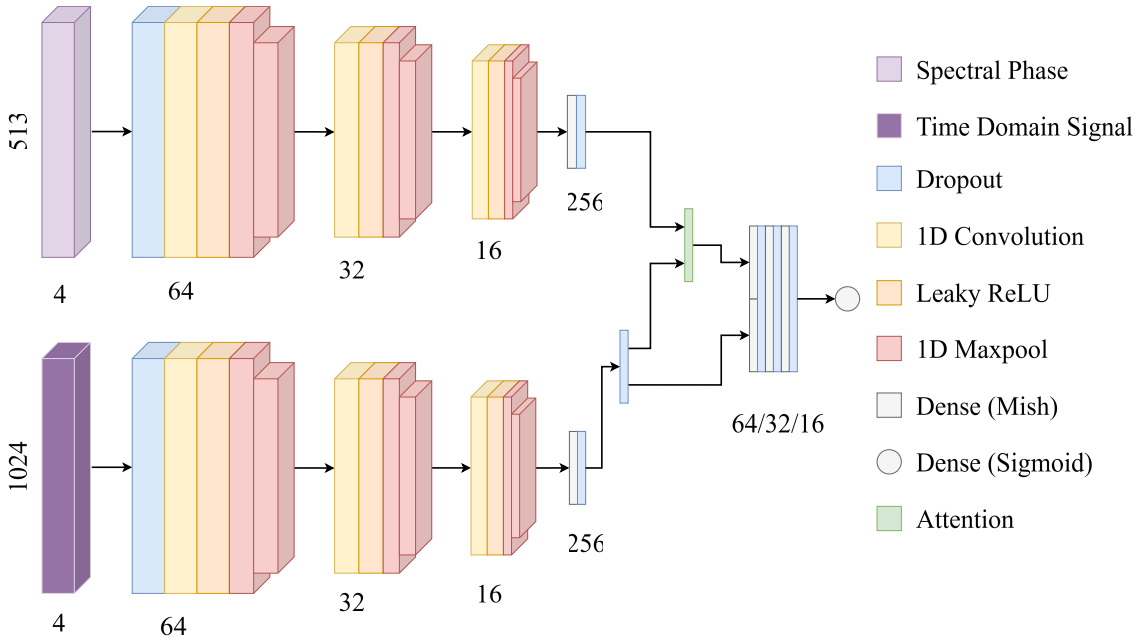


Figure 3.6.: Overview of our proposed whistle direction estimation neural network architecture. The numbers indicate the filter count of the convolution or the number of dense units, and a step indicates that the input is reduced by half.

Phase backbone. The phase backbone processes the $(513, 4)$ input using a sequence of 1D convolution layers with Leaky ReLU activations, interleaved with max-pooling. The output is flattened and passed through a dense layer with Mish activation [38], producing a 256-dimensional feature vector.

Level backbone. The level backbone mirrors the phase backbone but operates on the $(1024, 4)$ time-domain amplitude input. Its output is also a 256-dimensional feature vector.

Attention-based fusion. To refine the phase features using level information, we apply a softmax-based attention mechanism. A dense layer with softmax activation produces

a 256-dimensional weight distribution, which is multiplied element-wise with the phase feature vector. This allows the model to emphasise phase components that are consistent with the level-based cues.

Regression head. The fused feature vector is concatenated with the level backbone output and passed through three fully connected layers (64, 32, 16 units) with Mish activation and dropout. The final output layer uses a sigmoid activation, scaled to the range $[0, 360]^\circ$, yielding a single azimuth prediction. Since the dataset stores azimuth angles in the range $[-180, 180]^\circ$, all labels are converted to the interval $[0, 360]^\circ$ before training. This conversion is consistent with the unit-circle representation used in the loss function (see Subsection 3.3.4) and avoids discontinuities at the wrap-around boundary.

3.3.4. Loss Function

To address the slight imbalance in the distribution of direction labels, we adopt the shrinkage loss proposed by Lu et al. [39]. Instead of comparing angles directly, we convert both predicted and ground-truth angles into unit vectors on the circle as shown in Equation 3.1.

$$\mathbf{v} = (\cos\theta, \sin\theta) \tag{3.1}$$

The angular error is computed as the normalised angle between the two vectors. This representation avoids discontinuities at the wrap-around boundary (e.g. 5° vs. 355°) and ensures smooth gradients across the entire range.

The shrinkage formulation down-weights easy samples and emphasises difficult ones, improving robustness under imbalanced training conditions.

3.3.5. Training Procedure

Each whistle consists of multiple frames, but the network processes one frame at a time. To ensure that only the direct sound is used for training, we restrict sampling to the first frames of each whistle event, as described in Section 3.2. From these initial frames, we select up to three per whistle, and shuffle them. This avoids redundancy while guaranteeing that all training samples originate from the direct-sound region and are less affected by reverberation.

Training is performed using the Nadam optimiser with a learning rate of 3×10^{-4} and a batch size of 256. We train for up to 2500 epochs with early stopping based on the median validation loss. The model is checkpointed whenever the validation performance improves.

Although the dataset includes distance soft labels and an optional confidence-learning loss, these components are not used in the current version of the direction model.

3.3.6. Discussion

The combination of phase and level information enables robust azimuth estimation across a wide range of environments. The attention mechanism helps mitigate the effects of reverberation and microphone variability, while the unit-circle loss formulation ensures stable optimisation.

Challenges remain in scenarios with strong reflections or highly asymmetric whistle orientations, as illustrated in Figure 3.7. Nevertheless, the proposed architecture achieves reliable performance within the computational constraints of the NAO platform.

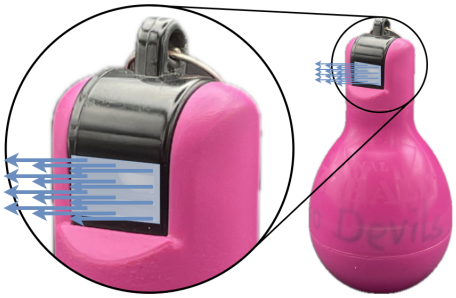


Figure 3.7.: Hand whistle used in the SPL, showing the directional opening that shapes the emitted sound field. First published in [2].

3.4. Whistle Distance

Estimating the distance of a whistle is the second component of the SSL task. While direction estimation determines whether a whistle originates from the robot's own field, distance estimation provides an additional cue for assessing its relevance. In contrast to azimuth estimation, distance perception is inherently more challenging. Humans themselves rely heavily on vision for accurate distance judgements, and auditory distance perception is known to be imprecise and strongly influenced by room acoustics and reverberation [34]. For the NAO robot, these challenges are further amplified by its limited field of view and restricted ability to visually track the referee during gameplay, as described in Section 1.1. Consequently, distance estimation must rely solely on acoustic features that can be extracted efficiently and robustly during operation.

3.4.1. Distance Binning and Problem Formulation

Rather than predicting a continuous distance value, we discretise the distance into five bins following the approach of Zohourian et al. [36]. The bin boundaries are chosen as 2 m, 4.5 m, 6.75 m, and 9 m, resulting in five distance classes that provide a finer resolution than a simple near–far distinction while avoiding unrealistic expectations of precise regression. These boundaries were chosen to reflect the most relevant distance ranges on a standard

SPL field. Distances up to 2 m correspond to situations where the referee stands very close to the robot, while 4.5 m and 6.75 m cover typical midfield and penalty-area positions. The final boundary at 9 m separates whistles originating from the opposite half of the field. This binning therefore captures the distances that are most informative for in-game decision making without requiring unrealistic regression accuracy. To account for the gradual nature of distance perception, we use soft labels obtained by Gaussian smoothing of the bin boundaries.

Unlike direction estimation, where only the first frames of a whistle are informative, distance estimation benefits from analysing a longer temporal window. An analysis of the training and test datasets revealed that 18 frames capture the majority of a whistle event while providing sufficient temporal context for robust distance estimation. Therefore, the distance model operates on 18-frame spectrogram segments.

3.4.2. Input Features

The distance estimation network uses two complementary input representations:

- **Spectrogram input:** An 18-frame log-amplitude spectrogram with shape $(18, 513, 4)$, where each frame corresponds to the magnitude of the STFT of the four microphones. This representation captures the temporal decay and spectral structure of the whistle, both of which correlate with distance.
- **DRR:** A frequency-dependent DRR feature with shape $(513, 4)$, computed from the first three frames (direct sound) and the subsequent 15 frames (reverberation). The DRR is defined as

$$\mathrm{DRR}(f) = 20 \log_{10}\left(\frac{D(f)}{R(f)}\right) \tag{3.2}$$

  where $D(f)$ and $R(f)$ denote the accumulated direct and reverberant energy at frequency f. This definition follows standard practice in room acoustics and auditory distance perception [34].

The spectrogram captures the temporal evolution of the whistle, while the DRR provides a physically grounded measure of distance that is robust to variations in whistle loudness and microphone gain.

3.4.3. Neural Network Architecture

The distance estimation network consists of two parallel backbones followed by a shared classification head, as shown in Figure 3.8.

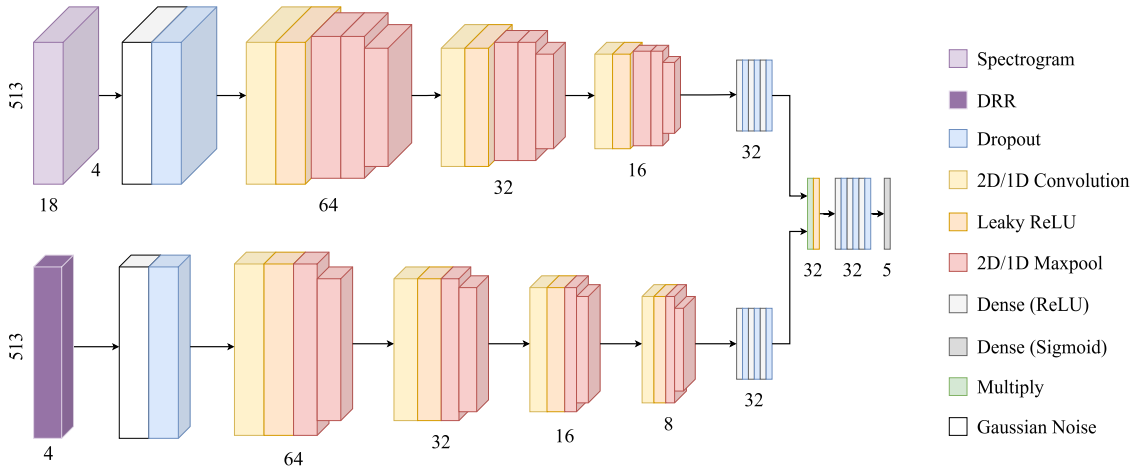


Figure 3.8.: Overview of our proposed whistle distance estimation neural network architecture. The numbers indicate the filter count of the convolution or the number of dense units, and a step indicates that the input is reduced by half.

Spectrogram backbone. The spectrogram input is processed by a 2D convolutional network. A Gaussian noise layer and channel-wise dropout provide robustness against microphone variability and environmental noise. Four convolutional blocks (64, 32, 16, and 8 filters) with Leaky ReLU activation and max-pooling extract increasingly abstract spectral–temporal features. The output is flattened and passed through three dense layers (64, 32, 16 units) with L1 regularisation and dropout, resulting in a compact feature vector.

DRR backbone. The DRR input is processed by a 1D convolutional network with a structure analogous to the spectrogram backbone. Gaussian noise and dropout are applied to the input, followed by four Conv1D–MaxPool blocks and three dense layers with L1 regularisation. This backbone extracts frequency-dependent cues related to the direct-to-reverberant energy ratio.

Fusion and classification head. The feature vectors from both backbones are concatenated and passed through three dense layers (32 units each) with dropout. The final output layer uses a sigmoid activation and produces a five-dimensional soft probability vector corresponding to the distance bins defined in Subsection 3.4.1. The bias of the output layer is initialised according to the empirical class distribution in the training dataset to stabilise early training.

3.4.4. Loss Function

To train the model, we use an adapted version of the focal loss [40] that accounts for the soft-label structure of the distance bins. For each bin, we define

$$p_t = 1 - \frac{|y - \hat{y}|}{\max(|1 - y|, |y|)} \tag{3.3}$$

where y is the soft target label and $\hat{y}$ is the predicted probability. The final loss is computed as

$$\mathcal{L} = -\alpha_t (1 - p_t)^\gamma \log(p_t + \varepsilon) \tag{3.4}$$

with $\gamma = 2$, $\alpha_t = 0.75$ for positive bins, and $\alpha_t = 0.25$ otherwise. This formulation emphasises difficult samples and reduces the influence of bins that are already predicted well.

3.4.5. Training Procedure

Only whistle events with at least 18 frames are used for training. Longer whistles are truncated to 18 frames, while shorter ones are discarded. For each training step, the generator extracts the 18-frame spectrogram, computes the DRR feature, applies optional augmentations (noise, scaling, and simulated microphone dropout), and constructs the corresponding soft distance label. We experimented with additional augmentation strategies, including additive noise, temporal scaling, and frequency perturbations. However, these augmentations did not improve training stability or validation performance and were therefore discarded. Only the simulated microphone dropout was retained, as it consistently improved robustness without degrading accuracy.

Training is performed using the Nadam optimiser with a learning rate of 3×10^{-4} and a batch size of 32. Early stopping is applied based on the mean validation loss, with a patience of 100 epochs. The model weights are checkpointed whenever the validation performance improves.

3.4.6. Discussion

The combination of spectrogram and DRR features enables robust distance estimation across a wide range of environments. The DRR backbone provides physically interpretable information that correlates strongly with distance, while the spectrogram backbone captures temporal and spectral cues that complement the DRR. The adapted focal loss ensures stable training despite the imbalanced distribution of distance classes.

Challenges remain in highly reverberant environments or when whistle orientation varies strongly, but the proposed architecture achieves reliable performance within the computational constraints of the NAO platform and provides a meaningful basis for distinguishing relevant from irrelevant whistle events.

3.5. Evaluation

This evaluation examines how reliably the proposed whistle localization system performs under controlled laboratory conditions and how well it meets the real-time requirements of RoboCup gameplay. We begin by outlining the evaluation setup in Subsection 3.5.1 and then present the first stage of the evaluation, the laboratory analysis. To capture the different operational conditions, the laboratory evaluation is divided into two parts: a standing evaluation (Subsection 3.5.2) and a walking-in-place evaluation (Subsection 3.5.3). The section continues with an analysis of the runtime performance in Subsection 3.5.4 and concludes with the in-game evaluation presented in Subsection 3.5.5.

3.5.1. Evaluation Setup

All evaluations were performed on NAO V6 robots running the NaoDevils framework. This ensures that the localization system is assessed under realistic computational load, including concurrent processes such as walking control, obstacle detection, and path planning. Audio-isolated tests were not conducted, as they do not reflect operational conditions during RoboCup matches.

Two laboratory scenarios were evaluated:

Standing evaluation. The robot was placed in the penalised position, where the head remains fixed and no self-generated robot noise (e.g., motor or joint noise) is present. The sound source was moved between whistles according to the spatial schema defined in Section 3.2. For each position and height, ten whistle events were recorded. This setup isolates the influence of room acoustics, microphone characteristics, and whistle orientation.

On-the-spot walking evaluation. To assess robustness under motor noise and head motion, the robot executed a walking-in-place motion. During each whistle event, the head performed a horizontal sweep from $-90\,^\circ$ to $+90\,^\circ$ at a fixed elevation of $17\,^\circ$ with a velocity of $80\,^\circ\,\mathrm{s}^{-1}$. The whistle source remained stationary during each individual whistle but was repositioned between whistles. This setup introduces several sources of uncertainty: delayed sensor readings, mechanical play in the head joints, and the challenge of assigning the correct head orientation to the corresponding audio frame.

Across all environments, typical background noise included room reverberation, the robot's cooling fan, occasional quiet conversations, and door sounds. These noises were either filtered out by the whistle detector or did not significantly affect localization.

Microphone conditions. Most evaluations were performed with robots equipped with four functioning microphones. A small number of tests involved robots with one defective

microphone. While the system can still operate in this configuration, direction estimates may flip by approximately 180 ° due to missing inter-microphone phase differences. For this reason, quantitative results focus on robots with all microphones operational.

3.5.2. Standing Evaluation: Direction and Distance

The standing evaluation provides a baseline for localization accuracy under stable conditions without robot-induced noise or head motion. The robot remained in the penalised position, and the sound source was moved between whistles according to the spatial schema defined in Section 3.2. For each position and height, ten whistle events were recorded.

Table 3.5 summarises the direction and distance errors obtained in this scenario. Direction estimation is consistently accurate across all tested positions. The median direction error ranges from 2 ° in the conference room to 12 ° at the field positions, indicating that the model provides reliable estimates when all four microphones are functioning and no head motion is present.

The office environment shows higher mean errors and a larger standard deviation than the other locations. This behavior is consistent with the stronger early reflections and the more confined room geometry. Small deviations in how the whistle was held, such as being positioned next to the body rather than directly above the marked point, also introduce minor systematic offsets in the ground-truth labels.

Distance estimation remains stable across all positions. The median normalised distance error is at or below one bin (0.25), and the mean error stays below 0.23 for all environments. The remaining variation is primarily caused by small inconsistencies in the whistle-holder's posture and arm position, as described in Section 3.2.

Table 3.5.: Direction and distance errors in an evaluation with standing robots, the letters in brackets describe which robot was used for the evaluation. $\tilde{x}$, $\bar{x}$, and σ denote the median, mean error, and standard deviation. The distance error is normalized on the maximum deviation represented by four distance bins. Therefore, a normalized distance error of 0.25 corresponds to a deviation by one bin.

	Direction Error			Normalized Distance Error		
	$\tilde{x}$	$\bar{x}$	σ	$\tilde{x}$	$\bar{x}$	σ
Penalty mark (right) [B]	12 °	14 °	13 °	0.0	0.16	0.22
Kick-off point [K]	12 °	16 °	19 °	0.25	0.21	0.17
Penalty mark (left) [V]	12 °	14 °	15 °	0.0	0.17	0.21
Conference Room [B]	2 °	4 °	5 °	0.0	0.16	0.23
Office [E]	5 °	14 °	24 °	0.25	0.23	0.18

3.5.3. Walking-in-Place Evaluation: Direction and Distance

The walking-in-place evaluation introduces motor noise, head motion, and timing uncertainties. These factors increase the complexity of the localization task. Table 3.6 summarises the direction and distance errors obtained under these conditions.

For robots with four functioning microphones, direction estimation remains consistent across most positions. Median errors lie between 8.5 ° and 15 °, and the increased mean and standard deviation reflect the additional uncertainties introduced by head motion and motor noise. The variability is particularly visible at positions with stronger reflections or less favourable head orientations, where individual outliers increase the mean error.

Robots with one defective microphone show substantially degraded performance. The missing channel leads to ambiguous inter-microphone phase differences, causing the estimated direction to flip by approximately 180 ° in many cases. This results in a median error of 58 ° and a mean error of 72.5 ° for robot K, confirming that reliable direction estimation requires all four microphones.

Distance estimation remains stable even under walking conditions. The median normalised distance error stays at or below one bin (0.25) for all positions, and the mean error remains below 0.34. This behavior is expected, as the distance model relies on longer temporal windows and the DRR feature, which is less sensitive to head motion and motor noise.

Table 3.6.: Direction and distance errors in an evaluation with walking robots, the letters in brackets describe which robot was used for the evaluation. $\tilde{x}$, $\bar{x}$, and σ denote the median, mean error, and standard deviation. The distance error is normalized on the maximum deviation represented by four distance bins. Therefore, a normalized distance error of 0.25 corresponds to a deviation by one bin.

	Direction Error			Normalized Distance Error		
	$\tilde{x}$	$\bar{x}$	σ	$\tilde{x}$	$\bar{x}$	σ
Goal (left) [B]	15 °	40.2 °	69.6 °	0.25	0.25	0.0
Center circle [B]	11.5 °	11.8 °	6.4 °	0.25	0.14	0.15
T-junction [B]	8.5 °	9.3 °	6.6 °	0.0	0.14	0.22
Goal (right) [B]	10.5 °	14.3 °	10.3 °	0.25	0.25	0.0
Corner (right) [B]	13.5 °	17.8 °	17.3 °	0.25	0.47	0.25
Overall (3 mics) [K]	58 °	72.5 °	51.2 °	0.25	0.34	0.22

3.5.4. Runtime Evaluation

Real-time performance is essential for deployment in RoboCup matches. The upper runtime limit for whistle localization is 30 ms, as described in Section 1.2. Runtime was

measured using the internal timing utilities of the NaoDevils framework, which buffer and smooth execution times over approximately ten frames. The reported values therefore represent a smoothed average.

Across all evaluations, the complete processing pipeline, including whistle detection, direction estimation, and distance estimation, required approximately 25 ms per frame. Only rare outliers exceeded the 30 ms threshold. Since localization is only executed when a whistle is detected, these occasional peaks do not cause frame drops or interfere with other robot processes. The system therefore meets the real-time requirements for RoboCup gameplay.

3.5.5. In-Game Evaluation

While the laboratory evaluations provide controlled conditions for analysing localization accuracy, they do not fully reflect the acoustic and geometric complexity of real RoboCup matches. The in-game evaluation therefore examines how the localization system performs under actual match conditions, using log data recorded during tournament games. The log files contain a wide range of information recorded during gameplay, including localization data, behavior states, sensor readings, and internal debugging information. For the in-game evaluation, only the components relevant to whistle localization were used. Importantly, the localization results shown in this section are not stored in the logs themselves. Instead, the same localization code that runs on the robot is executed during log replay, and the NaoDevils simulator visualises the robot's self-localised pose together with the predicted whistle direction and distance, represented as an arrow originating from the robot. Since the pose originates from the robot's own localization module, as implemented in the Nao Devils code release [14], and not from ground-truth measurements, both the robot position and the inferred relative whistle position are subject to uncertainty.

Evaluation setup. Only kick-off whistles were used as reference events, because this is the only situation in which the referee's position is reliably predictable in practice. The SPL rules specify that, at kick-off, the referee stands at the T-junction on the side opposite the game controller. Although this position is defined in the rule book, referees typically align themselves only approximately, and the effective sound source may vary depending on how the whistle is held (e.g., at the side of the body, in front, behind the back, or above the head). With an average arm length of roughly 75 cm, this can shift the actual sound source by up to one metre relative to the idealised reference point.

While the rules also describe referee positions for other match events, such as standing behind the goal when signalling a scored goal, these positions are not applied consistently in practice. In contrast, the kick-off position is the only one that is both rule-defined and reliably followed during real matches, which makes kick-off whistles the only suitable ground-truth reference for evaluation.

Most evaluations were performed using the goalkeeper (player number 1), as this robot typically occupies a stable position at kick-off and receives fewer penalties, resulting in a consistent geometric reference. One exception is RoboCup 2024 on Field D, where robot 7 was used instead.

Acoustic conditions across tournaments. The acoustic environments differed substantially between the tournaments:

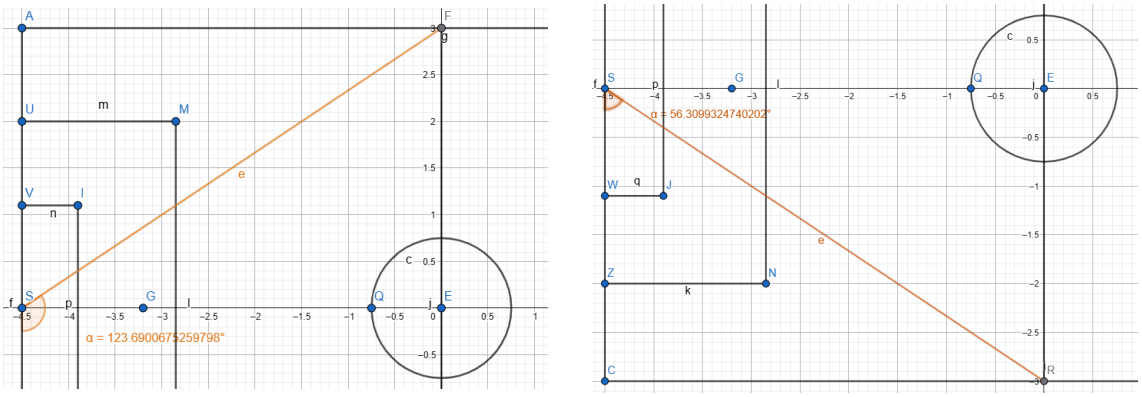
- **RoboCup 2023 (Exhibition Center Bordeaux):** Large indoor exhibition hall with a window front, producing reflections and reverberation.
- **RoboCup 2024:** Two distinct acoustic environments were used:
  - *Field C (outdoor field in a tent):* Tent structure with wind noise, reverberation, and variable reflection patterns.
  - *Field D (Indoor Sportcentrum Eindhoven):* Large indoor sports hall with a prominently loud ventilation system, which dominated the acoustic environment.
- **RoboCup 2025 (indoor, Brazil):** Low ceiling; Field D located directly in a corner, causing strong wall reflections; Field B also close to walls but less extreme.

Ground-truth definition. The ground-truth labels for the in-game evaluation were computed using the robot's self-localised pose at the time of each whistle event. The corresponding localisation data are listed in the appendix (see, e.g., Table A.5). For the schematic construction shown in Figure 3.9, however, the idealised referee and goalkeeper positions were used to illustrate how distances and angles are computed.

In this idealised geometry, the goalkeeper is assumed to be located at $(x = -4.5, y = 0)$ and the whistle at the T-junction at $(x = 0, y = \pm 3)$. This corresponds to a distance of 5.4 m and results in target angles of 123.69° or 56.3°, depending on whether the robot is positioned on the home or away side. A robot orientation of 0 ° denotes facing the opponent's goal line, and the directional convention follows the dataset definition: 0 ° corresponds to the robot's right-hand side, with angles increasing counterclockwise.

Error categories for interpretation. Since no exact ground-truth robot poses are available, the evaluation is qualitative. Directional predictions are described using the following categories:

- *close to the expected direction* $- > < 15\,^\circ$
- *moderate deviation from the reference direction* $- > 15$–$45\,^\circ$
- *clear deviation from the reference* $- > > 45\,^\circ$



Left side (game controller view)

Right side (game controller view)

Figure 3.9.: Idealised ground-truth geometry for kick-off whistles on both field sides, generated with GeoGebra.

- *left/right mirroring*
- *front/back mirroring*
- *approximate 180° reversal of the direction*

These numerical thresholds reflect the uncertainties inherent in the in-game setup: the referee does not stand exactly at the idealised T-junction position, the effective sound source can shift depending on arm posture, and the robot's pose is based on self-localisation rather than external ground truth. Mirroring categories are assigned based on geometric symmetry: *left/right mirroring* corresponds to a reflection across the robot's forward axis, *front/back mirroring* corresponds to a reflection across the lateral axis (x-axis), and an *approximate 180° reversal* represents a point reflection, where the predicted direction is rotated by roughly 180° around the robot. These definitions ensure that systematic geometric confusions are distinguished from ordinary angular deviations.

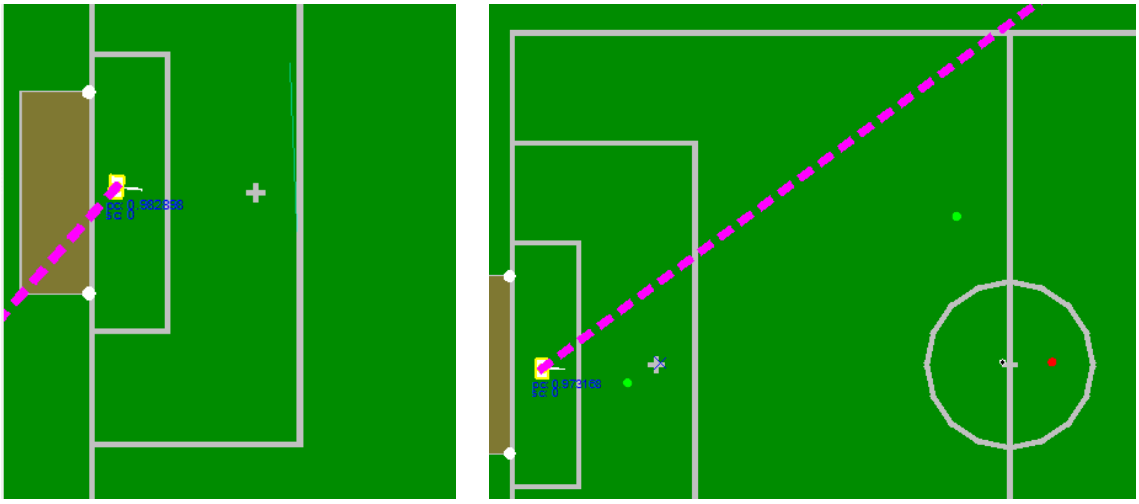
For distance interpretation, we refer to the methodology described in Subsection 3.5.2 and Subsection 3.5.3.

RoboCup 2025. For RoboCup 2025, the in-game evaluation covers matches on Field D and Field B, both of which were acoustically challenging indoor environments (see acoustic overview above). The predictions for all kick-off whistles recorded for robot R are summarised in Table 3.7 and Table 3.8. Using the error categories defined in this section, the 22 whistle events show a broad distribution across correct estimates, moderate deviations, and systematic mirroring effects.

Across both matches, **36.4 %** of the predictions were classified as *close to the expected direction*. A further **9.1 %** showed a *moderate deviation*, while **18.2 %** were judged as a *clear deviation from the reference*. Systematic mirroring effects occurred frequently:

9.1 % exhibited *left/right mirroring*, **18.2 %** showed *front/back mirroring*, and **9.1 %** corresponded to an *approximate 180° reversal*. Distance predictions predominantly fell into the larger distance intervals, consistent with the tendencies observed in the laboratory evaluations in Subsection 3.5.2 and Subsection 3.5.3. Across all 22 whistle events, **86.4 %** of the distance estimates deviated by only one interval from the reference value (error 0.25), while **9.1 %** matched the correct interval exactly (error 0). Only a single event (**4.5 %**) showed a deviation of two intervals (error 0.5). Given the width of the distance bins and the fact that the true in-game distance lies near the boundary between two intervals, deviations of 0.25 are not unexpected and can still be regarded as acceptable for in-game conditions.

Two representative examples of the simulator visualization for Field D are shown in Figure 3.10.



Whistle Nr. 1 (Field D) Whistle Nr. 2 (Field D)

Figure 3.10.: Log visualizations of two kick-off whistles on Field D (RoboCup 2025).

RoboCup 2024. RoboCup 2024 featured two distinct acoustic environments: an outdoor field in a tent (Field C) with wind noise and variable reflection patterns, and an indoor sports hall (Field D) dominated by a prominently loud ventilation system. The in-game evaluation includes both settings, with results summarised in Table 3.9 and Table 3.10.

On Field D, two whistle events (Nr. 7 and Nr. 8) occurred during the half-time break and correctly pointed towards a whistle on a neighbouring field. Two additional whistle events (Nr. 5 and Nr. 9) could not be evaluated due to missing detections or unfavourable game situations. Since events Nr. 7 and Nr. 8 do not correspond to kick-off situations on Field D and events Nr. 5 and Nr. 9 lack usable localization data, all four were excluded from the qualitative evaluation but are retained in the table for completeness.

Table 3.7.: In-game whistle localization results for the NAO robot Redskull [R] (player number 1) during RoboCup 2025 Round 4 against B-Human on Field D. The table lists the predicted whistle direction, the estimated distance interval, the calculated ground-truth, the error and the interpretation at the time of each whistle event. The horizontal separator marks the half-time break.

Nr.	Direction	Distance	Ground-truth		Error		Category
1	312 °	6.75–9 m	129.7 °	5.2 m	177.7 °	0.25	180 °
2	117 °	6.75–9 m	125.9 °	5.2 m	8.9 °	0.25	close
3	143 °	6.75–9 m	131.0 °	5.2 m	12.0 °	0.25	close
4	163 °	6.75–9 m	122.3 °	5.2 m	40.7 °	0.25	moderate
5	331 °	6.75–9 m	61.0 °	5.2 m	90.0 °	.025	∼ FB-mirror
6	101 °	6.75–9 m	53.5 °	5.2 m	47.5 °	0.25	wrong
7	10 °	6.75–9 m	64.2 °	5.2 m	54.2 °	0.25	wrong
8	112 °	6.75–9 m	54.8 °	5.2 m	57.2 °	0.25	LR-mirror
9	50 °	6.75–9 m	57 °	5.3 m	7.0 °	0.25	close
10	307 °	6.75–9 m	52.2 °	5.2 m	105.2 °	0.25	FB-mirror

Across all evaluable whistle events (13 in total), **7.7 %** of the predictions were classified as *close to the expected direction*, while **15.4 %** showed a *moderate deviation*. Clear deviations occurred in **38.5 %** of the cases. Systematic mirroring effects were also present: **7.7 %** exhibited *left/right mirroring*, and **30.8 %** showed *front/back mirroring*. No approximate 180° reversals were observed.

Regarding distance, **7.7 %** of the estimates matched the reference interval exactly (error 0), while **38.5 %** deviated by one interval (error 0.25) and **53.8 %** by two intervals (error 0.5). Given the coarse distance binning and the challenging acoustic conditions, deviations of one interval can still be regarded as acceptable, whereas larger deviations mainly reflect the unstable localization behaviour on the outdoor field.

Overall, the results reflect the challenging and heterogeneous acoustic conditions of the two fields, with the indoor Field D producing noticeably more stable direction and distance estimates than the outdoor Field C.

RoboCup 2023. The matches in the Exhibition Center Bordeaux took place in a large indoor hall with reflections. The in-game whistle localization results for robot K are summarised in Table 3.11. Across all ten kick-off whistles, **30 %** of the predictions were classified as *close to the expected direction*, while **50 %** showed a *moderate deviation*. Clear

Table 3.8.: In-game whistle localization results for the NAO robot Redskull [R] (player number 1) during RoboCup 2025 quarter final against B-Human on Field B. The table lists the predicted whistle direction, the estimated distance interval, the calculated ground-truth, the error and the interpretation at the time of each whistle event. The horizontal separators mark the breaks (referee timeout and half-time).

Nr.	Direction	Distance	Ground-truth		Error		Category
1	106 °	2–4.5 m	117 °	7.1 m	11.0 °	0.5	close
2	139 °	6.75–9 m	56.2 °	5.3 m	82.8 °	0.25	LR-mirror
3	359 °	6.75–9 m	53.8 °	5.2 m	54.8 °	0.25	wrong
4	37 °	6.75–9 m	50.4 °	5.3 m	13.4 °	0.25	close
5	326 °	6.75–9 m	62.8 °	5.3 m	96.8 °	0.25	FB-mirror
6	131 °	6.75–9 m	116.9 °	7.1 m	14.1 °	0.0	close
7	49 °	6.75–9 m	45.6 °	5.3 m	3.4 °	0.25	close
8	234 °	6.75–9 m	52.6 °	5.2 m	178.6 °	0.25	180 °
9	137 °	6.75–9 m	116.9 °	7.1 m	20.1 °	0.0	moderate
10	182 °	6.75–9 m	50.9 °	5.2 m	131.1 °	0.25	wrong
11	240 °	2–4.5 m	135.1 °	5.2 m	104.9 °	0.25	FB-mirror
12	119 °	6.75–9 m	114.5 °	5.2 m	4.5 °	0.25	close

Table 3.9.: In-game whistle localization results for the NAO robot Antman [A] (player number 7) during RoboCup 2024 Round 5 against B-Human on Field D. The table lists the predicted whistle direction, the estimated distance interval, the calculated ground-truth, the error and the interpretation at the time of each whistle event. The horizontal separator marks the half-time break. Two whistle events could not be evaluated: the first whistle was not detected, and the second produced no usable localization due to the game situation.

Nr.	Direction	Distance	Ground-truth		Error		Category
1	174 °	6.75–9 m	161.5 °	3.2 m	12.5 °	0.5	close
2	257 °	6.75–9 m	133.4 °	4.2 m	123.6 °	0.5	FB-mirror
3	113 °	6.75–9 m	133.6 °	4.4 m	20.6 °	0.5	moderate
4	72 °	6.75–9 m	108.3 °	3.6 m	36.3 °	0.5	moderate
5	-	-	-	-	–	–	
6	256 °	6.75–9 m	128.9 °	5.1 m	127.1 °	0.25	wrong
7	130 °	6.75–9 m	≈110 °	>9 m	20.0 °	0.25	moderate
8	101 °	6.75–9 m	≈110 °	>9 m	9.0 °	0.25	close
9	-	-	-	-	–	–	
10	339 °	6.75–9 m	56.4 °	4.5 m	77.4 °	0.5	FB-mirror
11	317 °	6.75–9 m	342 °	2.9 m	25.0 °	0.5	FB-mirror
12	170 °	6.75–9 m	42.8 °	4.4 m	127.2 °	0.5	wrong

Table 3.10.: In-game whistle localization results for the NAO robot Kronos [K] (player number 1) during RoboCup 2024 Round 3 against Bembelbots on Field C (Outdoor-Field). The table lists the predicted whistle direction, the estimated distance interval, the calculated ground-truth, the error and the interpretation at the time of each whistle event. The horizontal separators mark the breaks (half-time and robot reboot).

Nr.	Direction	Distance	Ground-truth		Error		Category
1	218 °	2–4.5 m	127.9 °	5.4 m	90.1 °	0.25	wrong
2	277 °	6.75–9 m	63.1 °	7.1 m	146.1 °	0.0	∼ FB-mirror
3	254 °	2–4.5 m	40.8 °	5.4 m	146.8 °	0.25	wrong
4	123 °	2–4.5 m	48.6 °	5.0 m	74.4 °	0.25	LR-mirror
5	258 °	6.75–9 m	122.6 °	5.9 m	135.4 °	0.25	wrong

deviations occurred in **10 %** of the cases, and **10 %** of the predictions exhibited *left/right mirroring*, with no front/back mirroring or approximate 180° reversals observed.

Regarding distance, all predictions deviated by one interval from the reference value (error 0.25). Given the coarse distance binning and the reverberant indoor environment, such deviations are not unexpected and can still be regarded as acceptable for in-game conditions.

Overall, the results reflect the acoustically challenging nature of the large indoor hall, but also indicate that the system can still produce direction estimates that are qualitatively usable under such conditions.

Table 3.11.: In-game whistle localization results for robot K (player number 1) during RoboCup 2023 Round 2 against B-Human on Field E. The table lists the predicted whistle direction, the estimated distance interval, the calculated ground-truth, the error and the interpretation at the time of each whistle event. The horizontal separator marks the half-time break.

Nr.	Direction	Distance	Ground-truth		Error		Category
1	120 °	6.75–9 m	53.6 °	5.3 m	66.4 °	0.25	∼ LR-mirror
2	38 °	6.75–9 m	49.1 °	5.3 m	11.1 °	0.25	close
3	357 °	6.75–9 m	57.5 °	5.1 m	60.5 °	0.25	wrong
4	358 °	6.75–9 m	41.4 °	5.3 m	43.4 °	0.25	moderate
5	116 °	6.75–9 m	132.2 °	5.2 m	16.2 °	0.25	moderate
6	115 °	6.75–9 m	127.3 °	5.2 m	12.3 °	0.25	close
7	114 °	6.75–9 m	132 °	5.3 m	18.0 °	0.25	moderate
8	101 °	6.75–9 m	122.1 °	5.1 m	21.1 °	0.25	moderate
9	162 °	6.75–9 m	126.3 °	5.2 m	35.7 °	0.25	moderate
10	146 °	6.75–9 m	134 °	5.3 m	12.0 °	0.25	close

Summary. The in-game evaluation demonstrates that the localization system can produce correct directional estimates under real match conditions, but also exhibits systematic error patterns. Across all tournaments, the most frequent sources of error were mirroring effects, which is consistent with both the symmetric microphone geometry and the challenging acoustic environments described in Section 3.5. The qualitative results from RoboCup 2023 (Table 3.11), RoboCup 2024 (Table 3.9 and Table 3.10), and RoboCup 2025 (Table 3.7 and Table 3.8) show that the system reliably identifies the general direction of the whistle in a substantial fraction of cases, while moderate deviations and mirroring effects remain common under difficult acoustic conditions.

Distance predictions were dominated by the larger distance intervals, and the majority of estimates deviated by only one interval from the reference value (error 0.25). This behaviour is consistent with the tendencies observed in the laboratory evaluations in Subsection 3.5.2 and Subsection 3.5.3. Across all tournaments, the distribution of distance estimates was noticeably skewed towards the upper distance bins, reflecting both the coarse binning and the fact that many in-game whistle events occurred at distances near the boundary between adjacent intervals.

Overall, the results highlight the importance of evaluating localization models not only in controlled laboratory settings but also under diverse real-world acoustic conditions, where reflections, background noise, and localization uncertainty interact in complex ways.

4. Broken Joint Detection

Humanoid robots operating in dynamic environments must not only perceive their surroundings and execute complex behaviors, but also continuously assess their own physical condition. In the SPL, this requirement is particularly pronounced: robots are exposed to frequent collisions, repeated stand-up motions, and continuous walking, all of which contribute to mechanical wear. A single malfunctioning joint can severely impair a robot's ability to play, increase the risk of further damage, or even force the robot to leave the field. Reliable self-diagnosis is therefore essential for maintaining both performance and hardware longevity.

Traditional approaches to fault detection in robotics often rely exclusively on proprioceptive sensor data, such as joint currents, stiffness values, and angle errors. While these signals provide valuable information, they do not capture all relevant aspects of a robot's condition. In practice, many mechanical defects, such as rattling or spinning gears, produce characteristic acoustic signatures that humans can easily recognise. Inspired by this observation, we extend classical fault detection with an audio-based anomaly detection module, enabling the robot to identify faults that may not yet manifest clearly in sensor readings.

The proposed system follows the principles of FDIR. It integrates multiple sensory modalities, enriches them with contextual information about the robot's current situation, and triggers appropriate behavioral adaptations to prevent consequential damage. This multimodal approach allows the robot not only to detect faults more reliably, but also to react intelligently by adjusting its behavior, for example by refusing unsafe stand-up motions or reducing stiffness in damaged joints.

To develop and evaluate this system, two datasets were created: an audio dataset containing diverse examples of normal and defective joint sounds, and an evaluation dataset combining audio and sensor data recorded both in laboratory settings and during real RoboCup matches. These datasets form the basis for training the audio anomaly detection network, validating the sensor-based detection, and assessing the overall performance of the FDIR pipeline.

This chapter is structured as follows. Section 4.1 summarises relevant work on fault detection and recovery in robotics. Section 4.2 describes the datasets used for audio-based and sensor-based detection. The audio anomaly detection and classification pipeline is detailed in Section 4.3, followed by the sensor-based detection approach in Section 4.4.

The integration of both modalities through a rule-based residual assessment framework is presented in Section 4.5. Section 4.6 discusses how the system prevents consequential damage by adapting the robot's behavior. Finally, Section 4.7 evaluates the performance, robustness, and runtime characteristics of the proposed FDIR system.

4.1. Related Work

Detecting, isolating, and recovering from faults is essential for both industrial and space robotics [41–43]. In industrial settings, the primary goals are minimizing downtime and reducing maintenance costs [42], whereas in space missions, fault detection and recovery are mission-critical, as physical repairs may be infeasible or impossible [43]. Khalastchi et al. [44] emphasize that these challenges are fundamental to robotics in general, independent of the specific domain.

Classical Fault Detection and Isolation/Identification (FDI) approaches are commonly grouped into model-based, data-driven, and knowledge-based methods [42, 44]. A central concept across these categories is the generation of *residuals*, i.e., signals that quantify the discrepancy between expected and observed system behavior. The foundational work by Chow and Willsky [45] formalized residual generation through analytical redundancy, introducing parity relations and observer-based techniques that remain influential today. Residuals can be derived via parity equations, Kalman filtering, or observer schemes [46], and they form the basis for detecting deviations caused by sensor, actuator, or process faults.

While classical residuals rely on proprioceptive measurements, recent work has explored augmenting them with data-driven features. In our case, we enrich the residual space with audio-based anomaly indicators. This choice is motivated by the fact that mechanical degradation, such as gearbox damage, often produces characteristic acoustic signatures, making audio a valuable additional information source. Unlike vibration or temperature sensing, which are commonly used in industrial diagnostics, these modalities are not or not directly available on our platform, as temperature can only be inferred indirectly via current draw. Consequently, audio provides the most informative complementary signal for detecting faults in our system.

For audio anomaly detection, One-Class Neural Networks (OC-NN) [47] are particularly suitable, as they learn compact representations of normal operation and flag deviations without requiring fault examples. This is essential in robotics, where fault data are scarce because robots are not intentionally damaged and repairs are costly or sometimes impossible. OC-NN architectures typically combine an encoder with a lightweight one-class objective [47], making feature extractors such as ResNet [48] a natural choice.

In our implementation, we adopt only the one-class loss formulation proposed in the OC-NN preprint [47]. The open-source implementation provided by the authors served

as a practical reference for integrating this loss into our pipeline, but the remaining components, including the audio encoder architecture and the autoencoder-based feature extraction, are based on our work described in Chapter 2 and Chapter 3. Thus, OC-NN contributes the conceptual and mathematical foundation for the loss function, while the overall model design and training procedure follow our established methodology.

Although neural-network-based FDI systems can outperform rule-based approaches in detection accuracy [46], they lack the transparency, determinism, and exception-handling capabilities required for safety-critical and embedded robotic systems. Hybrid architectures therefore offer a favorable trade-off: learned residuals provide sensitivity to subtle anomalies, while rule-based reasoning ensures interpretability, real-time guarantees, and predictable behavior, an approach also reflected in space-grade FDIR systems [41, 43]. Following this insight, we combine audio-based neural residuals with classical sensor-based ones inside a rule-based FDI framework.

Detecting a fault alone is insufficient for safe operation. Robots must adapt their behavior to avoid further damage, especially when mechanical components are compromised. Spacecraft fault management systems therefore integrate detection, isolation, and recovery into full FDIR pipelines [41, 43]. Inspired by these architectures, we extend our FDI system with a recovery component based on our *KeyFrameEngine* (see Section 5.2). This framework enables two key capabilities:

1. a diagnostic routine that evaluates joint health under controlled motion, and
2. an alternative stand-up strategy that avoids stressing damaged joints, preventing cascading failures.

4.2. Dataset

Datasets are a critical component of detection systems, particularly when neural networks are involved. Although FDIR has been studied in several domains, it is not yet widely explored in humanoid robotics, where platform-specific characteristics play a major role. Robots differ in the placement and quality of their microphones, the number and type of joints, and the materials used in their mechanical structure. As a result, transferring datasets between platforms is difficult, and reliable fault detection requires data recorded directly on the target robot.

For this work, all data were collected using our NAO V6 robots, described in Section 1.1. The joint defects represented in our datasets arise exclusively from naturally occurring wear and tear in our robots. We did not intentionally damage any robots to create specific fault types, and robots used by other teams at RoboCup events are typically in excellent condition, since teams bring their most reliable units. As a result, the distribution of defects in our datasets reflects the realistic failure modes encountered dur-

ing long-term operation rather than artificially induced damage. The goal was to create datasets that reflect realistic operating conditions and cover the range of joint defects typically encountered in the SPL. Since the audio-based and sensor-based components of our detection system address different aspects of the problem, we created two separate datasets that allow us to analyze these components both independently and in combination. The audio-only dataset enables an isolated evaluation of the anomaly detection network, similar in spirit to an ablation study, revealing how much information can be extracted from audio alone. The second dataset, which integrates audio and joint sensor data, is used to evaluate the behavior of the complete FDIR pipeline and to study the interaction between both modalities under realistic operating conditions.

The first dataset, introduced in Subsection 4.2.1, contains audio recordings of normal and defective joint behavior. This dataset is used to train, validate, and test the audio anomaly detection network described in Section 4.3. Only audio data are required for this part of the system, and the dataset structure follows the conventions established in our previous work on whistle detection, Section 2.2, and localization, Section 3.2. This choice allowed us to reuse existing processing pipelines within our framework, avoiding unnecessary recomputation and ensuring consistency across perception modules.

The second dataset, described in Subsection 4.2.2, combines audio data with synchronised joint sensor readings. This dataset is essential for evaluating the complete FDIR pipeline, since the final system must process both modalities together. The synchronisation is based on the timing information available in the robot logs, which reflect the behavior of the system during real operation. Although the exact timing accuracy depends on how the data are provided by the robot, the logs capture the temporal relationships between audio and sensor signals sufficiently well for evaluating the detection and recovery behavior.

Together, these two datasets provide the foundation for training the audio-based detection module, validating the sensor-based detection, and assessing the performance of the integrated system under laboratory conditions and during real RoboCup matches.

4.2.1. Audio Anomaly Detection Dataset

The audio anomaly detection dataset is used to train, validate, and test the audio-based detection module described in Section 4.3. It focuses exclusively on acoustic information and contains recordings of both normal operation and clearly audible joint defects. All recordings were made on NAO V6 robots using the four integrated microphones, with a sampling rate of 22 050 Hz, as configured in our framework.

Defect types and recording environments. To capture a representative range of audible defects, we selected four robots that exhibited distinct joint failures. These include a defective head yaw joint and multiple shoulder defects affecting either the pitch or roll

axis. The recordings were collected in different acoustic environments, such as laboratory rooms, other indoor spaces, and a children's play setting. In addition to defective examples, the dataset includes challenging negative examples from real-world scenarios, previously used in Chapter 2 and Chapter 3, including recordings from the Maker Faire in Rome and a test match in Eindhoven with a referee whose high-pitched voice previously caused false detections in whistle recognition. This combination of positive and negative examples is intended to improve the robustness of the anomaly detection network in realistic conditions.

Table 4.1.: Overview of robots and defect types included in the audio anomaly detection dataset.

Robot	Defect type	Recording environment	Number of recordings
D	Head yaw joint	Laboratory rooms	8
E	Shoulder pitch joint	Laboratory rooms	3
P	Shoulder pitch joint	Children's play environment	12
R	Shoulder roll joint	Multiple rooms	5
		Total number of recordings	30

Preprocessing. Each recording is processed using the same audio preprocessing pipeline as in our whistle detection and localization work, see Subsection 2.2.1. We reuse this established pipeline without modification to ensure consistency across perception modules and to avoid redundant implementations within our framework. The annotation tool exports the resulting log-magnitude spectra, and each labelled sample corresponds to one time window represented as a NumPy array of shape $(513, 1)$ together with a binary label indicating whether the window contains a defective joint sound.

Annotation process. To obtain labelled training data, we use an interactive annotation tool that supports both manual and model-assisted labelling. At the beginning of the project, all annotations were created manually, since no trained model was available. Once the first set of labelled examples had been collected, an initial model was trained and integrated into the tool. From that point on, the tool displayed preliminary anomaly predictions that annotators could accept, refine, or reject. This iterative process allowed the model to improve gradually while ensuring that all labels remained under human supervision. The model is described in Section 4.3.

Tool functionality. The tool visualises the log-amplitude spectrogram and allows annotators to listen to short audio segments to verify the acoustic characteristics of each window. Regions can be labelled manually or by correcting the model's suggestions, and additional segments can be marked when necessary. All four microphone channels are supported, and

the tool exports the final labels together with the corresponding spectral features. Each labelled sample corresponds to one time window and is stored as a NumPy array with shape $(513, 1)$ for the spectral features and a binary label indicating whether the window contains a defective joint sound.

Dataset characteristics. The resulting dataset provides a diverse and realistic basis for training the audio anomaly detection network. It covers multiple defect types, acoustic environments, and background noise conditions, and is designed to support robust detection performance in real RoboCup scenarios.

4.2.2. FDIR Evaluation Dataset

The second dataset is used to evaluate the complete FDIR pipeline under realistic operating conditions. It contains synchronised audio and sensor data extracted from full robot logs recorded during laboratory experiments and RoboCup matches. These logs include all information available on the NAO V6 platform, such as audio streams, joint sensor readings, IMU and FSRs data, camera images, stiffness and current values, internal behavior states, game controller messages, localization estimates, perceived obstacles, and team communication. Although not all of these data sources are used directly by the FDIR system, they provide a comprehensive representation of the robot's state and environment.

Timestamp-based synchronisation. Audio and sensor data are synchronised using the timestamps embedded in the logs. The cognition process, which includes audio processing, operates at 30 Hz, while the motion process updates joint sensor information at approximately 83 Hz. The timestamp-based synchronisation ensures that both modalities can be aligned with sufficient accuracy for evaluation purposes.

Laboratory recordings. The dataset consists of two parts. The first part was recorded in our laboratory arena and includes both quiet conditions and scenarios with artificially introduced noise. Six robots (A, M, N, P, Q, R) were used for recordings with noise, and five robots (A, N, P, Q, R) participated in recordings without noise. The robots performed typical motions such as walking, falling, standing up, and interacting with the ball. Two additional test games were recorded, one with noise sources such as a vacuum cleaner, a drill driver, and metal impacts, and one without noise.

RoboCup German Open 2025 recordings. The second part of the dataset consists of logs recorded during the RoboCup German Open 2025. These logs were collected during three complete and two partial soccer games. Only our own robots were analyzed, but the matches were played against teams such as B-Human, Bemblebots, Berlin United,

HULKs, and SPQR. The following robots and game halves were included in the dataset. Unless stated otherwise, both halves of each match are used.

- **A**ntmanntman: B-Human (first half), Bemblebots, Berlin United, SPQR
- **D**aredevil: B-Human, Berlin United, SPQR
- **F**rogthor: B-Human, Berlin United, SPQR
- **J**arvis: B-Human, Berlin United
- **N**ova: B-Human
- **Q**uicksilver: B-Human (first half), Berlin United, SPQR
- **U**ltron: B-Human, Berlin United, SPQR
- **V**enom: HULKs
- **W**onderwoman: Bemblebots (first half), HULKs (second half), SPQR (first half)
- **X**avier: Bemblebots (first half), Berlin United, SPQR (first half)

Dataset size and defect diversity. In total, the event dataset comprises 67 logs with approximately 490 minutes of gameplay data. These logs contain a diverse set of joint defects, including temporarily faulty joints, limited usability due to mechanical wear, and fully broken joints. The variety of robots, opponents, and match situations ensures that the dataset covers a broad spectrum of real-world conditions.

Export format for evaluation. For the evaluation, the relevant information is exported from the logs into a CSV file using the BrokenJointEvaluator module. Each entry contains two distinct timestamps:

1. the event timestamp, which marks the moment when the underlying fault first becomes observable in the sensor data, and
2. the detection time, which corresponds to the moment when the FDIR system reaches a final classification decision.

In addition, each entry includes the affected joint, the internally stored fault type, and the evaluated joint state. This CSV format provides a compact and consistent representation of the information required for analysing both the correctness and the responsiveness of the complete FDIR system.

4.3. Audio-Based Detection and Classification

The audio-based detection module identifies characteristic acoustic signatures that occur when a joint is damaged. Mechanical defects such as spinning or rattling gears generate distinct vibration patterns that propagate through the robot's structure and are captured by the NAO's four microphones. Our approach consists of two stages: an anomaly detection neural network and a lightweight classifier that distinguishes between different defect types using prior knowledge. Figure 4.2 and Figure 4.3 illustrate typical spectrograms of rattling and spinning gearboxes, while Figure 4.1 shows the architecture of the anomaly detection network. The entire audio pipeline operates at 30 Hz on the NAO robot.

4.3.1. Neural Network

The first stage of the audio pipeline is an anomaly detection neural network that evaluates short audio segments for indications of abnormal mechanical behavior. Each audio frame consists of 1024 samples recorded at 22 050 Hz. After applying a Hamming window, the signal is transformed into the frequency domain using the Fourier transform. Due to symmetry, only the first 513 frequency bins are retained, and their magnitudes are converted to a logarithmic scale using the operation defined in Equation 2.1.

The resulting log-spectral representation serves as the input to a compact convolutional neural network inspired by the ResNet architecture [48]. The network comprises four residual blocks with varying kernel sizes, followed by a dense layer with a sigmoid activation that outputs the confidence that the current frame contains an acoustic anomaly. Residual connections improve gradient flow and stability, while the small model size ensures real-time inference on the NAO's hardware.

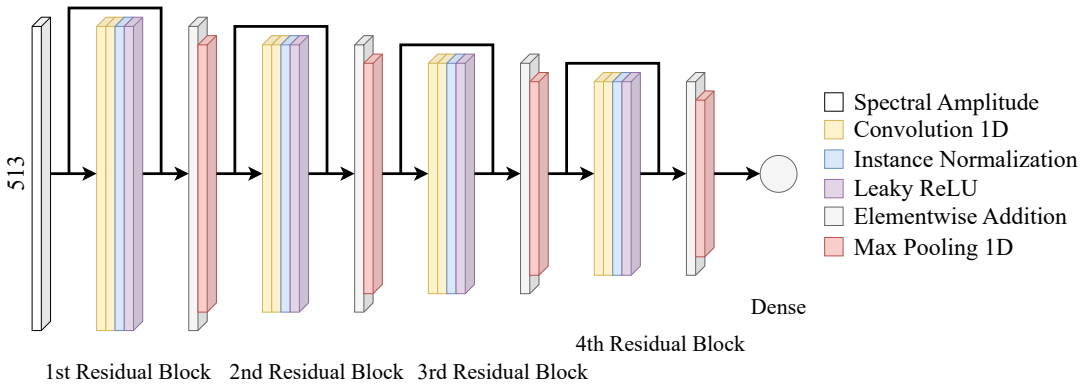


Figure 4.1.: Overview of our anomaly detection neural network architecture. Numbers indicate the size of the input, arrows show the connections between the blocks, and each step represents a halving of the input size. First published in [3].

To address the strong class imbalance in the dataset (approximately 8:1 between normal operation and defective joints), we apply a combination of ENN [21] and the SMOTE [20].

This approach proved effective in our whistle recognition, in Chapter 2, and ensures that the anomaly detector generalizes well across different robots and acoustic environments.

Training is performed using the Ranger optimizer, which combines RAdam with the lookahead mechanism, as described in Chapter 2. Early stopping is applied to prevent overfitting. The final model is exported to TensorFlow Lite for deployment on the NAO robot, enabling efficient inference within the 30 Hz cognition thread. The complete implementation is available in our audio processing tools [18].

4.3.2. Enrichment with Prior Knowledge

While the neural network identifies whether an anomaly is present, it does not distinguish between different types of mechanical defects. To classify detected anomalies into *rattling* or *spinning* gearboxes, we use a lightweight, knowledge-driven approach based on the ZCR. Figure 4.2 and Figure 4.3 show representative spectrograms from our dataset that illustrate the acoustic differences between both defect types.

The ZCR is defined as the number of sign changes in the audio signal normalized by the frame length of 1024 samples [49]. A high ZCR indicates a broadband and irregular signal, whereas a low ZCR corresponds to a more periodic structure. We classify an anomaly as rattling if the ZCR exceeds the threshold $\tau = 0.8$, and as spinning otherwise. This threshold was determined empirically and is high enough to suppress human speech, which also exhibits elevated ZCR values [50], while reliably separating the two defect types.

This hybrid approach combines the strengths of machine learning and classical signal processing: the neural network provides robust anomaly detection, while the ZCR-based classifier offers an interpretable and computationally efficient way to distinguish between defect categories.

4.3.3. Runtime and Deployment Considerations

The audio-based detection pipeline operates within the 30 Hz cognition thread of the NAO robot. To ensure real-time performance and avoid interference with whistle recognition, the anomaly detection network is executed only when no whistle candidate is active. Whistle detection has higher priority, as missed or false whistle events directly affect gameplay, whereas joint defects evolve gradually and do not require immediate reaction within a single frame.

Separating whistle and broken-joint analysis prevents overlapping FFT windows and reduces the risk of false positives. Whistle candidates produce strong spectral peaks and elevated ZCR values, which could otherwise be misinterpreted as mechanical anomalies. By evaluating joint-related audio only during whistle-free intervals, the system maintains high robustness even in noisy environments.

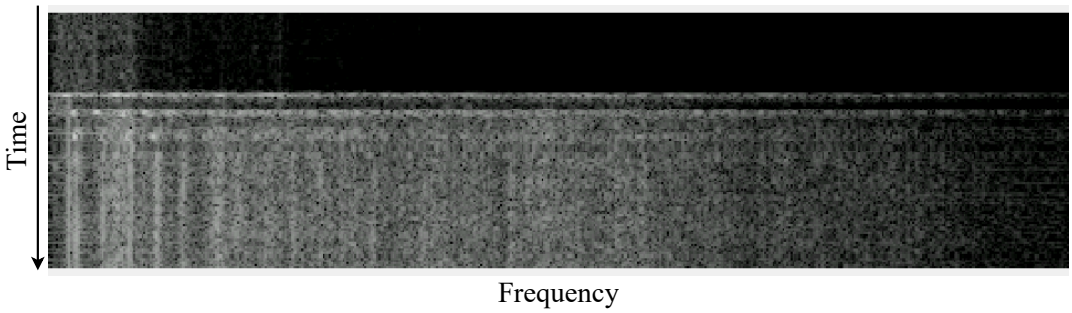


Figure 4.2.: Spectrogram illustrating the characteristics of silence (upper part) and a rattling gearbox (lower part), generated using the log file of the NAO robot Proteus [P]. First published in [3].

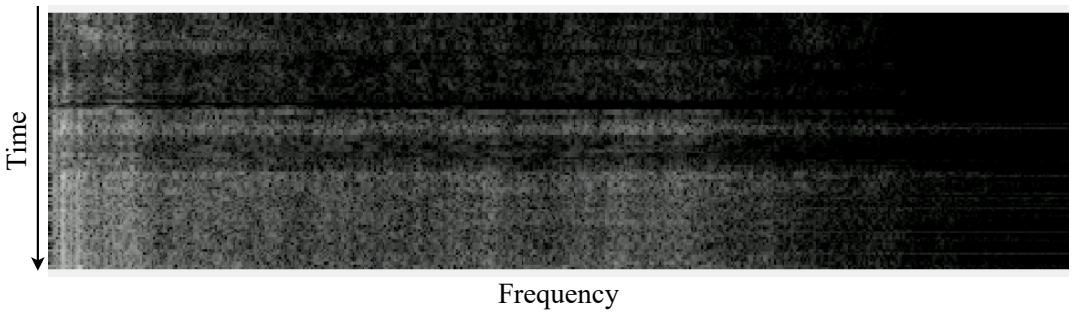


Figure 4.3.: Spectrogram illustrating the characteristics of silence (upper part) and a spinning gearbox (lower part), generated using the log file of the NAO robot Redskull [R]. First published in [3].

The TensorFlow Lite model enables low-latency inference on the NAO's limited CPU resources. Combined with the lightweight ZCR-based classification, the architecture remains fully real-time capable without blocking other cognition modules. This design ensures that whistle detection retains the highest priority, while audio-based fault detection operates opportunistically in the background, providing reliable classification without compromising gameplay performance.

4.4. Sensor-Based Detection

The sensor-based detection module evaluates the mechanical condition of each joint using only proprioceptive measurements available on the NAO robot. The method relies on five quantities that are continuously provided by the robot's hardware: the motor current, the commanded stiffness, the deviation between commanded and measured joint angle, the average joint play estimated for each joint, and the information whether the robot is currently standing on its feet. These signals allow the system to distinguish between different fault mechanisms without requiring external sensing.

The joint angle deviation serves as the primary residual. It reflects how well a joint follows the commanded position and is therefore sensitive to both mechanical failures and external disturbances. However, due to wear and tear, the NAO's plastic gearboxes introduce an increasing amount of mechanical play over time. To avoid false detections caused by this natural degradation, the angle deviation threshold is dynamically adjusted using the average joint play. This value is estimated per joint during walking, where the continuous motion provides sufficient excitation to measure the effective backlash. If the robot has not walked long enough, a conservative static value is used instead.

Based on these measurements, the detector distinguishes four types of residuals:

Malfunction. A malfunction is detected when a joint does not produce any measurable motor current despite a sufficiently high commanded stiffness. If, in addition, the joint fails to reach the desired position and the angle deviation exceeds the threshold corrected by the average joint play, the joint is classified as malfunctioning. This situation typically corresponds to a motor that is turned off or unable to generate torque.

Stuck Joint. A stuck joint is characterized by a high motor current and a high commanded stiffness, while the joint still fails to reach the requested position. In this case, the joint actively attempts to move, but the motion is blocked. The cause may be internal (e.g., a damaged gear) or external (e.g., contact with another robot or a goal post). The sensor-based detector cannot distinguish these cases on its own, and the classification is refined later during residual assessment.

General Problem. A general problem is detected independently of the motor current. If both the commanded stiffness and the angle deviation exceed their respective thresholds, the system assumes that the joint is unable to follow the command reliably. This category captures faults that do not fit the malfunction or stuck-joint patterns, such as slipping gears or intermittent mechanical issues.

Frozen Load. The FrozenLoad residual is triggered when the sensor values of a joint remain unchanged for an unusually long period of time. This behavior indicates that the joint is no longer providing valid proprioceptive feedback. The most common cause is a loss of connection in the flat ribbon cable, for example when an arm has detached and the joint electronics are no longer reachable. However, short communication interruptions can produce similar symptoms, which is why the required duration for declaring a FrozenLoad state was chosen conservatively. Unlike the other residuals, FrozenLoad does not describe a mechanical fault but a loss of sensor data, which must be handled separately in the rule-based assessment.

Each joint uses its own stiffness and angle deviation thresholds, which were determined through a combination of controlled tests and analysis of extensive real-world data. To minimize false detections during matches, all thresholds were chosen conservatively. This ensures that only persistent and clearly observable deviations trigger a fault, while short-term fluctuations or minor disturbances are ignored.

Together, these three residual types form the basis for the subsequent rule-based assessment described in Section 4.5.

4.5. Residual Assessment

The residual assessment module combines sensor-based and audio-based residuals with contextual information from the robot's environment. This fusion enables a robust classification of joint conditions, even in situations where individual residuals may be ambiguous. The assessment is rule-based and fully configurable, allowing each joint to be evaluated according to its mechanical role, typical failure modes, and situational constraints.

4.5.1. Rule-Based Evaluation Framework

Each joint is associated with a set of residual types generated by the sensor-based and audio-based detection modules. To determine the most relevant residual, we apply a priority ordering defined as

`stuckJoint` > `spinningGear` > `rattlingGear` > `frozenLoad` > `malfunction` > `gearProblem`.

This ordering reflects the severity and reliability of the underlying detection mechanisms. For example, a `stuckJoint` indicates an immediate mechanical blockage, whereas `gearProblem` represents a more general and less specific deviation.

To classify the selected residual, we use a rule-based framework that incorporates environmental information such as whether the robot is upright, whether obstacles are nearby, which motion is currently active, or whether the robot is penalized. These rules are derived from empirical observations during RoboCup games and laboratory tests and are encoded in a configurable behavior file. If no rule can confidently determine the joint state, the system assigns the state *undecided*, indicating that additional observations are required.

4.5.2. Diagnostic Test Motions

To resolve ambiguous cases, the system can execute diagnostic test motions. These motions sweep selected joints through their full controllable range to gather additional information. For example, the `testHeadYaw` routine moves the head yaw joint from its minimum to maximum angle, while arm and leg routines actuate all joints of the respective limb. Separate routines exist for upright and lying configurations to account for restricted ranges of motion. These diagnostic motions are only executed when the robot is not walking and when they do not interfere with gameplay.

4.5.3. Short-Term and Long-Term Memory

To ensure stable classification, the system maintains a memory structure for each joint. Short-term memory filters out transient fluctuations by requiring a residual to persist for a minimum duration before it is considered relevant. If the residual disappears before reaching this threshold, it is discarded. Long-term memory stores persistent or critical faults until they are explicitly resolved by a rule. This prevents oscillations between states and ensures that serious defects remain visible to the behavior system.

4.5.4. Joint State Categories

Based on the rule evaluation and memory system, each joint is assigned one of the following states:

- **healthy** — no relevant residuals are present.
- **undecided** — insufficient information is available.
- **temporarilyFaulty** — a transient or minor deviation is detected.
- **temporarilyBlocked** — the joint is obstructed by the environment or current motion.

- **limitedUsability** — the joint is degraded but still usable.
- **lostConnection** — sensor data is frozen or unavailable.
- **broken** — a severe mechanical defect is detected.

4.5.5. LED Feedback

To provide immediate visual feedback, the robot's eye LEDs encode the joint state using the color scheme in Table 4.2.

Color	RGB	Joint State
violet	(0.5, 0.0, 1.0)	broken
white	(1.0, 1.0, 1.0)	temporarilyFaulty
cyan	(0.0, 1.0, 1.0)	limitedUsability
yellow	(0.5, 0.5, 0.0)	lostConnection

Table 4.2.: LED color coding for joint state feedback.

4.5.6. Global Joint State Determination

Finally, the system computes a global joint state for the robot. If audio-based detection identifies a defect but no sensor-based residuals are present, the global state is set to *audioOnly*. If no residuals are detected, the robot is considered *fine*. Otherwise, the global state is *broken*. This global decision is used by the recovery framework to adapt the robot's behavior accordingly.

4.6. Prevention of Consequential Damage

Detecting a fault is only the first step. A robot must also adapt its behavior to prevent further damage. Based on the assessed joint states, we adjust both the affected joint and the robot's overall behavior to ensure safe operation.

To make the resulting behavior transparent and to summarize the safety logic applied across different joint categories, Table 4.3 provides an overview of the joint states and the corresponding actions taken by the robot. The table abstracts the decision process and highlights the fundamental principles that guide the prevention of consequential damage. The detailed handling of arm and leg joints is described in the following subsections.

4.6.1. Arm Joints

If an arm joint is classified as *faulty* or *broken*, its stiffness is set to zero to avoid additional mechanical stress or sounds. A *faulty* joint indicates that the motor is no longer producing

Table 4.3.: Overview of joint categories, detected states, and corresponding safety actions.

Joint category	State	Resulting action
Arm	healthy	normal operation
Arm	limited	joint used with caution, stand-up allowed
Arm	faulty	stiffness set to zero, stand-up restricted depending on motion
Arm	broken	stiffness set to zero, stand-up disallowed if shoulders required
Leg	healthy	normal operation
Leg	limited	walking and stand-up allowed with reduced confidence
Leg	faulty	walking disabled, safe sitting or lying position enforced
Leg	broken	no locomotion or stand-up, robot remains in safe posture

torque, whereas a *broken* joint corresponds to a spinning gearbox with minimal or no friction.

Shoulder joints require special handling. If a shoulder joint is *faulty* or *broken*, the robot refuses to perform a front stand-up motion, as this movement relies heavily on both shoulders and would likely cause further damage. A new front stand-up motion was developed and successfully used during RoboCup 2025. Based on our observations, the motion is likely to enable recovery even with a defective shoulder joint, as several robots with significantly worn or loosened arm joints were still able to perform it reliably. However, the motion was not explicitly tested with a fully defective shoulder joint, as it was created during the competition and not all robots were available for systematic evaluation. Since RoboCup 2025 was also the final event using the NAO platform, the motion was not further integrated as a long-term recovery strategy. When lying on its back, the robot only refuses to stand up if the arms are not positioned behind the torso, since this configuration is essential for a safe and stable back stand-up motion.

Other arm joints, with the exception of the elbow roll, are less critical for basic mobility and stand-up motions. In these cases, the affected joint is simply turned off, and no additional behavioral adjustments are required.

4.6.2. Leg Joints

Leg joints are essential for locomotion and for all stand-up motions. If a leg joint is classified as *faulty* or *broken*, the robot can no longer walk or stand up safely. In such cases, the robot transitions into a safe state: it either sits down if currently standing or, if lying on the ground, refuses to stand up. The robot remains in this safe position until external assistance is provided.

This behavior prevents further mechanical damage and ensures that the robot does not attempt motions that it can no longer execute reliably.

4.7. Evaluation

This evaluation addresses three main objectives. First, we assess the performance of the audio anomaly detection module, which serves as an early indicator of mechanical issues. Second, we evaluate the complete FDIR pipeline with respect to its ability to detect and classify joint defects under laboratory and real-world conditions. Third, we analyze the timing behavior of the system, that is, how long it takes from the first occurrence of a fault to the final classification decision. Together, these analyzes provide a comprehensive view of the robustness, reliability, and practical applicability of our approach.

4.7.1. Experimental setups

The evaluation is based on the two datasets introduced in Subsection 4.2.2. The first dataset contains recordings from controlled laboratory experiments, including both quiet conditions and scenarios with artificially introduced noise. The second dataset consists of logs collected during the RoboCup German Open 2025 and represents realistic match conditions with crowd noise, collisions, and varying levels of mechanical wear.

For the evaluation, we use only the synchronised audio and joint sensor data exported by the BrokenJointEvaluator module[18]. This ensures that both laboratory and competition recordings are processed in a consistent format, independent of the additional information contained in the full robot logs. The two setups therefore provide complementary conditions: controlled and repeatable laboratory data, and diverse real-world data from competitive matches.

4.7.2. Evaluation methodology

Building on the two experimental setups described in Subsection 4.7.1, all evaluations are carried out using exported CSV logs that contain the detected joint states, the corresponding ground truth annotations, and timing information. These CSV files are generated by the BrokenJointEvaluator module and provide a compact representation of the information required for evaluating the broken joint detection pipeline.

A dedicated Python notebook implements the complete evaluation workflow. It extracts the first occurrence of each detected fault, aligns predictions with manually annotated labels, and computes both binary and multi-class metrics, including precision, recall, accuracy, and confusion matrices for all severity levels.

In addition, the notebook analyzes detection timings by measuring the delay between the first occurrence of a fault and the final decision of the FDIR system. This provides insight into the responsiveness of the system under both controlled and real-world conditions.

The audio-only anomaly detection module is evaluated independently using labelled audio data from the robots Proteus and RedSkull, as these were the only robots for which

isolated audio annotations were available. Binary classification metrics are computed to assess the performance of the audio-based component separately from the full FDIR pipeline.

4.7.3. Audio anomaly detection

We first evaluate the audio anomaly detection module, which uses an empirically determined threshold of 0.1. Under silent laboratory conditions, the system achieves a precision of 96.9 %, a recall of 53.3 %, and an accuracy of 75.4 %. Under less controlled conditions, the precision decreases to 91.0 %, the recall to 39.9 %, and the accuracy to 62.4 %. Across both environments, this results in a combined precision of 93.9 %, a recall of 45.9 %, and an accuracy of 68.4 %.

These results show that the audio module operates conservatively: it produces very few false positives, but this comes at the cost of a lower recall, particularly in noisy environments. Such behavior is appropriate for a real-time robotic system, where false alarms would trigger unnecessary shutdowns and disrupt gameplay, whereas a slightly delayed detection does not change the underlying mechanical state. Within the FDIR pipeline, the classification of anomaly types using the ZCR approach achieves an accuracy of 100 %.

4.7.4. FDIR detection performance

The next step evaluates the complete FDIR system with respect to its ability to detect whether a defect is present. Table 4.4 summarises the precision, recall, and accuracy for both evaluation environments.

Table 4.4.: Detection performance of the FDIR system.

Metric	Laboratory	German Open 2025
Precision	100.0 %	95.2 %
Recall	87.5 %	76.9 %
Accuracy	92.9 %	91.5 %

The system achieves high precision in both settings, indicating that false alarms are rare. The recall is lower, particularly under real-world conditions, which reflects the conservative behavior of the detection pipeline. Despite this, the overall accuracy remains above 90 % in both environments, showing that the system reliably identifies the presence of mechanical defects.

4.7.5. FDIR classification performance

To evaluate the classification of joint states, we analyze the confusion matrices for both environments. The laboratory results in Table 4.5 show that healthy joints are consistently classified correctly. Most misclassifications occur between the *faulty* and *broken* categories, which is expected since both represent severe mechanical issues and differ mainly in the presence of characteristic noise patterns.

Table 4.5.: Confusion Matrix Lab Evaluation

		Predicted			
		healthy	**faulty**	**limited**	**broken**
True	**healthy**	6	0	0	0
	faulty	0	1	0	3
	limited	1	0	1	0
	broken	0	0	0	2

The real-world evaluation in Table 4.6 shows a similar pattern. Healthy joints are classified correctly in almost all cases, with only a single misclassification as *limited*. This misclassification was caused by an external collision during a stand-up attempt, which produced a noise pattern similar to a rattling gearbox. Since joints classified as *limited* typically remain functional, this error is not critical.

Table 4.6.: Confusion Matrix RoboCup German Open 2025

		Predicted			
		healthy	**faulty**	**limited**	**broken**
True	**healthy**	55	0	1	0
	faulty	3	10	1	6
	limited	3	0	2	1
	broken	0	0	0	0

The most notable misclassifications involve joints labeled as *faulty* that were predicted as *healthy*. All three cases originate from a single robot whose wrist joint had been switched off and mechanically fixed with tape. Since the wrist does not normally move during gameplay and the taped joint produced no measurable deviation, the system classified it as healthy even though the joint was known to be defective. This misclassification is unrelated to acoustic noise, which only affects the distinction between the *limited* and *broken* categories. Similarly, one joint labeled as *limited* was classified as *broken*. Since

both states correspond to reduced functionality and differ mainly in the presence of noise, this misclassification is acceptable and does not compromise safety.

4.7.6. Detection timing

Finally, we analyze the time required for the system to reach a final decision after the first occurrence of a fault. Table 4.7 summarizes the minimum, maximum, mean, and median detection times for each joint state.

Table 4.7.: Detection timings RoboCup German Open 2025.

State	min [s]	max [s]	mean [s]	median [s]
broken	1.15	407.81	120.82	55.45
limited	1.01	1.78	1.25	1.1
blocked	1.04	22.58	2.03	1.15
faulty	3.30	4.80	3.50	3.30

The system typically reaches a decision within one to five seconds. The only outliers occur in the *broken* category, where some joints were initially classified correctly but later reclassified due to unrelated noise events occurring long after the fault first appeared. These late reclassifications do not affect safety, as both *faulty* and *broken* states trigger conservative behavior.

Overall, the timing results indicate that the system reacts quickly enough to prevent consequential damage while remaining robust against transient noise.

5. Motion

While Chapter 4 focused on detecting joint defects and understanding the mechanical limitations caused by wear, joint play, and sensor uncertainty, this chapter addresses the complementary question of how a robot should move in a way that prevents additional damage and remains functional for as long as possible. The goal of the motion framework presented here is to increase the robustness of the robot's movements such that failures occur less frequently and worn-out robots remain usable for a longer time.

Among all additional motions, the stand-up routine is the most critical. A robot that repeatedly fails to stand up loses valuable playing time, risks further falls, and is exposed to increased mechanical stress. Every fall accelerates wear, and every unsuccessful stand-up attempt increases the likelihood of additional damage. Therefore, robustness, stability, and adaptability are more important than achieving the fastest possible stand-up motion.

The motion system developed in this work focuses on three aspects of robustness:

1. preventing unnecessary falls through more stable stand-up sequences,
2. adapting to changing internal and external conditions such as floor properties, battery level, or accumulated mechanical degradation, and
3. compensating joint play and mechanical degradation during execution.

These mechanisms allow the robot to continue operating safely even when its hardware condition deteriorates. If no reliable continuation of a motion is possible, due to severe wear, blocked joints, or missing recovery strategies, the robot transitions into a safe posture or remains lying down, consistent with the behavior of the FDIR system, described in Chapter 4.

In this sense, the motion framework acts as the behavioral counterpart to the diagnostic components introduced earlier: FDIR identifies when a joint is no longer trustworthy or when certain motions should be avoided, and the motion system enforces these constraints while exploiting the robot's remaining capabilities as effectively as possible.

The remainder of this chapter is structured as follows. Section 5.1 summarises related work on stand-up motions and adaptive movement generation. Section 5.2 introduces the keyframe engine that forms the basis of our approach and is reused across diagnostic and recovery behaviors.

The stand-up routine itself, including conditional movement selection, interception strategies, and compensation of wear, is described in Section 5.3. Safety considerations

are discussed in Section 5.4, and Section 5.5 evaluates the performance of the proposed system on different surfaces and robot conditions.

5.1. Related Work

As Stückler et al. showed in their seminal work [51], enabling humanoid robots to reliably recover from falls is far from trivial. Their approach introduced one of the first general stand-up routines for kid-size humanoid robots, developed in simulation and validated under real-world conditions at RoboCup 2005.

A decade later, Jeong et al. proposed a reinforcement-learning-based method to improve existing stand-up sequences [52]. Their system uses a Gaussian Mixture Model to discretize the state space and applies Q-learning to refine hand-designed keyframes for adult-size robots. While promising, such RL approaches face a pronounced sim-to-real gap, as the learned policies depend heavily on the fidelity of the simulation.

More recently, Stelter et al. presented a spline-based stand-up framework for kid-size humanoid robots [53]. Their method replaces traditional keyframe animations with parametrized Cartesian splines and employs simulation-based parameter optimization to achieve fast and robust motions.

All of these approaches were developed for custom-built robot platforms, which simplifies accurate simulation and model identification but limits transferability to standardized platforms such as the NAO used in the SPL. Motion-tracking-based strategies, such as the sit-to-stand transfer demonstrated by Mistry et al. [54], are also not applicable to NAO robots due to fundamental anatomical and actuation differences. For example, the NAO's hip-yaw-pitch joint mechanically couples both legs, preventing independent leg placement, and its arm actuators are not strong enough to support human-like push-off strategies.

A further commonality across prior work is the reliance on simulation to overcome the reality gap before deploying motions on real hardware. For custom robots, this is feasible because their mechanical and inertial properties are known and can be modeled accurately. For the NAO platform, however, such high-fidelity simulation is not yet available, making end-to-end learned or dynamically optimized stand-up motions difficult to transfer.

For these reasons, we continue to use handcrafted keyframes as the foundation of our stand-up motions, but we introduce a fundamentally different mechanism for linking and selecting them, enabling more robust and context-aware recovery behavior on the NAO platform.

5.2. Keyframe Engine

The keyframe engine is the central component for all keyframe-based motions in our system. It provides a structured framework for executing predefined motion sequences

such as stand-up routines, fall protection, interception movements, goalkeeper actions, and cheering motions. While other motion engines handle continuous behaviors like walking, kicking, or head control, the keyframe engine is responsible for complex, discrete whole-body motions that require tight coordination of many joints.

Although the engine operates on hand-crafted keyframes, its behavior is far from static. Interpolation, stabilization based on inertial measurements, conditional branching, and online statistics enable the robot to execute motions robustly even under varying hardware conditions, joint wear, and external disturbances. In particular, the engine can adapt motion execution at runtime, abort or redirect motions when stability is at risk, and compensate for joint errors induced by mechanical play.

The engine is implemented as a modular component within the robot's motion control architecture. It receives special action requests from higher-level behavior modules and produces joint commands that are continuously adapted based on short-term sensor feedback. This design allows the keyframe engine to react to disturbances, prevent damage through fall-down protection and interception movements, and, in combination with the stand-up statistics, decide when no reliable motion remains and the robot should stay on the ground.

5.2.1. Structure

Each motion processed by the keyframe engine is represented as a sequence of keyframes. A keyframe defines target joint angles, stiffness values, interpolation parameters, and optional constraints such as stabilization requirements or fall-down protection triggers. During execution, the engine interpolates between consecutive keyframes using either linear or sinusoidal interpolation, depending on the motion specification. A phase variable tracks the progress of the interpolation and ensures smooth transitions between postures.

Sensor feedback and deviation handling To support robust execution, the engine maintains several short-term sensor buffers, including joint angles, IMU orientation, and gyroscope readings. These buffers provide a filtered estimate of the robot's current state and allow the system to detect deviations from the expected motion. If the deviation exceeds predefined thresholds, such as excessive torso inclination, unstable gyroscope readings, or blocked arm joints, the engine can pause the current keyframe, trigger fall-down protection, or switch to an interception movement.

Internal state and chaining The engine also maintains internal state variables such as the current keyframe index, the active motion identifier, and whether the motion is executed in mirrored mode. This enables the reuse of motion sequences for both sides of the robot and supports transitions between chained keyframe motions. In addition, the engine keeps track of the start motion of each chained sequence, allowing it to restart a motion chain

from its intended entry point rather than accidentally beginning with an intermediate motion.

Overall, the structure of the keyframe engine combines predefined motion descriptions with real-time sensor feedback and internal state tracking. This design allows the system to execute complex whole-body motions reliably, even under varying hardware conditions and external disturbances.

5.2.2. Conditions

A central feature of the keyframe engine is its ability to select subsequent motion fragments based on a set of predefined conditions. These conditions operate on two levels: the level of *keyframe motions*, which represent complete motion fragments, and the level of individual *keyframes* within such a motion.

Motion-level conditions Most conditional branching occurs at the motion level. Each keyframe motion may specify multiple candidate successor motions, each associated with a condition that must be satisfied for the transition to occur. Examples include `lyingOnFront`, `fallingBack`, `notOnGround`, or `useStandUpStatistic`. These conditions are used to select appropriate stand-up strategies, restart chained motion sequences from their intended entry point, or trigger interception movements when instability is detected.

During execution, all relevant motion-level conditions are evaluated using real-time sensor data. If multiple conditions are satisfied, the first matching condition in the motion specification is selected, ensuring deterministic yet flexible branching. This mechanism allows the robot to adapt its motion sequence to the current situation, for example by selecting different stand-up variants depending on the fall direction and the statistic described in Subsection 5.2.3.

Frame-level conditions Conditions at the keyframe level are used more selectively. Individual keyframes may define local transitions or fall-down protection triggers, such as `directionFront` or `directionBack`. These frame-level conditions allow the engine to react within a motion fragment when the robot becomes unstable or when a protective manoeuvre must be executed immediately. In practice, this mechanism is primarily used for fall-down protection and stabilization rather than for general motion branching.

Safety and hardware-awareness Beyond posture-based conditions, the engine incorporates checks related to arm usability, joint angle deviations, and the robot's uprightness. These conditions ensure that motions are only executed when the robot is physically capable of performing them without risking additional damage. For instance, if the arms are blocked or the torso angle deviates too far from the expected limits, the engine may pause the current keyframe, switch to a release movement, or activate fall-down protection.

Integrated decision layer The condition system therefore acts as a lightweight decision layer embedded directly into the motion execution process. By combining predefined motion structure with real-time conditional branching, it enables the keyframe engine to react to disturbances, compensate for hardware limitations, and maintain safety without requiring higher-level behavioral intervention. This integration is a key factor in the robustness of the overall motion framework.

5.2.3. Stand-Up Statistics

To improve robustness over time, the keyframe engine maintains an online statistic that tracks the success and failure rates of different stand-up variants. For each stand-up motion, the engine records both the number of attempts and the number of aborted executions. These values are updated after each stand-up attempt. Let s denote the number of started stand-up motions and a the number of aborted ones. The update rules are

$$a^{t+1} = a^t + 1 \tag{5.1}$$

$$s^{t+1} = s^t + 1 \tag{5.2}$$

From these quantities, the failure rate f is computed as

$$f = \frac{a}{s} \tag{5.3}$$

This failure rate is used as part of the condition system to select between alternative motion paths.

Role of the statistic The statistic provides a lightweight form of online adaptation that adjusts the stand-up behavior to the robot's current hardware state and the properties of the environment. Motions that repeatedly fail under the prevailing conditions, such as increased joint play, reduced battery voltage, or a softer floor, become less likely to be selected, while more reliable variants are preferred. Separate statistics are maintained for front and back stand-up routines, allowing the robot to adapt independently to different fall directions. These statistics persist across games and are only reset when a new software version is deployed, ensuring that experience accumulated under realistic match conditions is retained rather than discarded.

Contribution to robustness Although the statistic does not guarantee recovery from all failure modes, it increases the likelihood of selecting a motion that succeeds under the robot's current wear level and environmental constraints. In this way, it contributes to the overall robustness of the motion system by reducing unnecessary falls and preventing repeated execution of unsuitable stand-up strategies.

5.3. Stand-Up

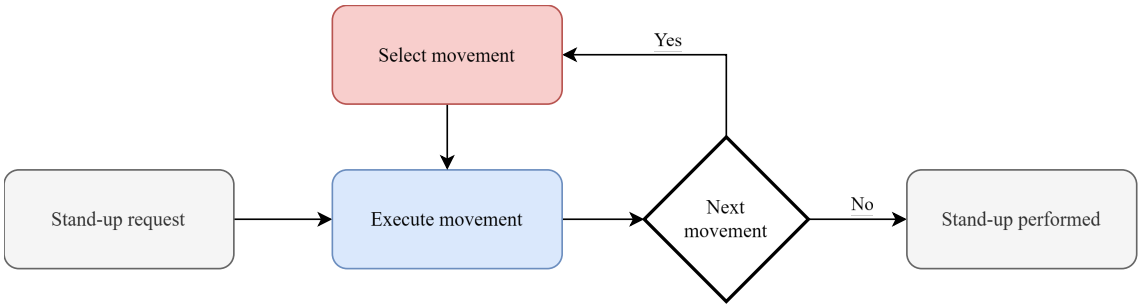


Figure 5.1.: Overview of the proposed stand-up architecture. The red box is detailed in Subsection 5.3.1, while the blue box is explained in Subsection 5.3.2 and Subsection 5.3.3. First published in [4].

Our stand-up approach consists of three interacting components:

1. a conditional movement selection mechanism,
2. interception movements for safety and recovery,
3. and a compensation mechanism for joint play.

Wear compensation is particularly important because even small deviations in hip, knee, or ankle pitch accumulate across successive keyframes and can destabilise the entire stand-up sequence.

These components are implemented using the keyframe engine introduced in Section 5.2. The keyframe engine provides the underlying execution framework, including interpolation, stabilization, fall-down protection, and the online stand-up statistics described in Subsection 5.2.3. The stand-up system builds on these capabilities by linking individual keyframe motions into adaptive stand-up routines.

The conditional movement selection forms the core of the stand-up logic. It determines which partial motion to execute next based on posture, stability, and the learned failure rates of alternative stand-up variants. Interception movements complement this mechanism by preventing damage and enabling controlled recovery when the robot becomes unstable during a stand-up attempt. Finally, the compensation mechanism adjusts

joint commands to counteract mechanical play and wear, improving the reliability of the executed motions.

Figure 5.1 illustrates how these components interact within the overall architecture. A detailed description of these three components is provided in Subsection 5.3.1 for the conditional movement selection, Subsection 5.3.2 for the interception movements, and Subsection 5.3.3 for the joint play compensation.

5.3.1. Conditional Movement Selection

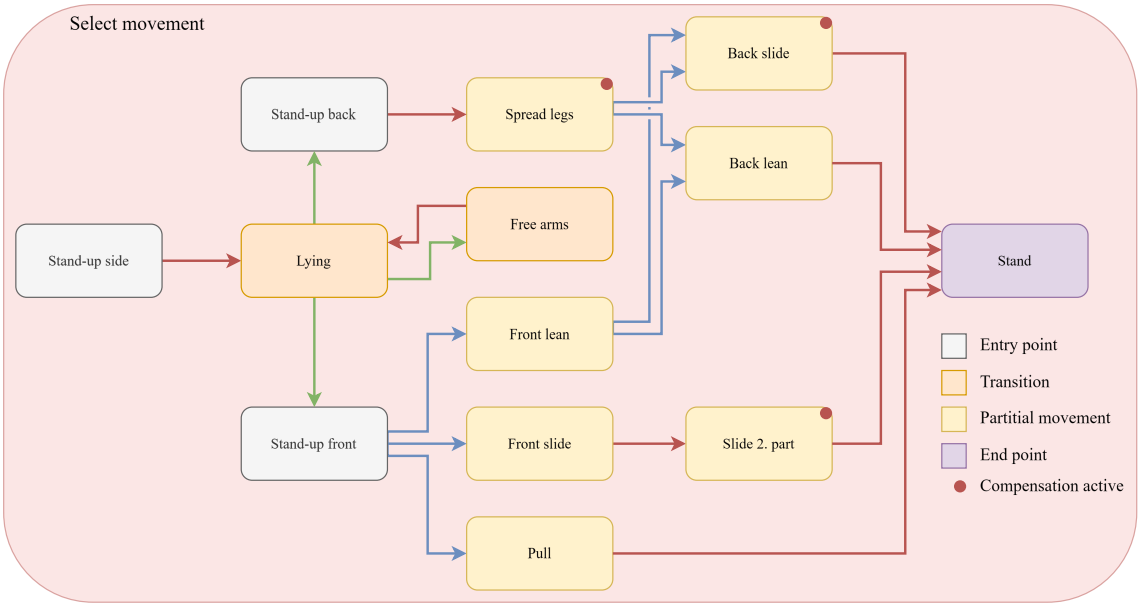


Figure 5.2.: Overview of our movement selection process, it involves forced transitions (red arrows), environmentally influenced transitions (green arrows), and learned transitions (blue arrows). First published in [4].

The stand-up routine is constructed as a sequence of partial motions that are linked through a decision structure similar to a decision tree. Each partial motion corresponds to a keyframe-based movement, and transitions between these motions are governed by the condition system of the keyframe engine described in Subsection 5.2.2. This allows the robot to adapt its stand-up behavior to its current posture, stability, and hardware condition.

We distinguish three types of transitions:

1. **Forced transitions:** These transitions are used when a specific motion must be executed to reach a valid configuration. For example, if the robot is sitting or crouching, it must transition to the *stand* motion before walking can resume. Forced transitions avoid duplicating identical keyframes across multiple stand-up variants and ensure consistency when motions are modified.

2. **Environmentally influenced transitions:** These transitions depend on the robot's physical state, such as its orientation or torso inclination. The initial stand-up motion (`standUp`) evaluates whether the robot is lying on its front, back, or side using IMU data and the `FallDownState`. The `FallDownState` representation provides a compact description of the robot's physical situation. It encodes the current body state (e.g. *upright*, *falling*, *onGround*, *flying*), the direction in which the robot is falling or has fallen (*front*, *back*, *left*, *right*), and a coarse tilt classification distinguishing between front-left, front-right, back-left, and back-right orientations. These quantities allow the stand-up system to determine the initial posture, detect unstable phases, and select appropriate stand-up variants or safety reactions. Based on this evaluation, the engine selects one of the motions `standUpFront`, `standUpBack`, or `standUpSide`. Additional conditions check whether the torso angle deviates from the expected range or whether the robot is sufficiently upright to continue. The underlying stability checks and thresholds are provided by the keyframe engine (Subsection 5.2.1).

3. **Online learned transitions:** When multiple stand-up variants exist, such as `standUpFrontLean`, `standUpFrontSlide`, `standUpFrontPull`, or `standUpFrontRoll`, the selection is based on the failure rates computed by the keyframe engine's stand-up statistics (Subsection 5.2.3). Once sufficient data has been collected, the variant with the lowest estimated failure probability is selected.

Empirical fall statistics from recent RoboCup tournaments support this design choice. During RoboCup 2023, the Nao Devils robots fell on average 57.7 times per game. This raw value does not account for differences in effective playing time or the number of robots on the field. For this reason, the statistic also reports a normalised average of 68.7 falls per game, which compensates for these factors and provides a more comparable measure across matches [55, 56]. The analysis from the RoboCup German Open 2024 shows an improved but still substantial average of 47 falls per match (non-normalised). These values indicate that the stand-up routine is triggered frequently enough for all relevant stand-up variants to be executed multiple times during a single game, allowing the online statistic to gather meaningful data and produce reliable failure-rate estimates.

Figure 5.2 illustrates a subset of the possible decision paths. To maintain clarity, transitions to interception movements are omitted; these are discussed separately in Subsection 5.3.2. The green transitions in the figure correspond to environmentally influenced decisions, which are evaluated sequentially. The first condition that evaluates to true determines the next motion, ensuring deterministic behavior while still allowing flexibility.

Safety-related conditions also influence the decision tree like structure. If the arms are blocked or if the torso angle deviates beyond a safe threshold, the keyframe engine triggers a release movement or fall-down protection (Subsection 5.2.2). These mechanisms prevent

unsafe transitions and ensure that the stand-up routine only proceeds when the robot is physically capable of doing so.

Overall, the conditional movement selection combines deterministic posture evaluation, online adaptation, and safety-aware fallback strategies. Together with the interception and compensation mechanisms described in Subsection 5.3.2 and Subsection 5.3.3, it enables robust stand-up behavior across a wide range of hardware conditions and environmental influences.

5.3.2. Interception Movement



Figure 5.3.: Robot in *sit* position. First published in [4].

The keyframe engine does not only execute predefined stand-up motions, but also monitors their execution online and triggers interception movements when a successful completion becomes unlikely. These interception mechanisms build on the same condition system that governs stand-up transitions (cf. Section 5.2) and are integrated with the stand-up statistics described in Subsection 5.2.3. In this way, the interception logic reuses existing sensing and decision infrastructure instead of duplicating stand-up specific heuristics.

A first class of interception-related motions addresses blocked arms while the robot is lying on the ground. Since several stand-up variants require large arm movements, a fall onto one arm can make the subsequent motion physically infeasible. Following the idea introduced by B-Human [12], the keyframe engine includes a dedicated *free arms* motion that lifts the hip to relieve contact forces on the arms. During the *lying* motion in Figure 5.2, the engine evaluates the arm joint errors and, if they exceed a configurable threshold, inserts *free arms* as a preparatory step before continuing with the selected

stand-up variant. This avoids repeatedly attempting a motion that is doomed to fail due to a purely kinematic constraint.

A second critical situation occurs at the beginning of the *stand-up front* motion. Here, the robot uses its arms to push the torso into a more upright configuration. If the arms cannot reach the required posture, approximately forming a T-shape relative to the torso, the subsequent keyframes cannot generate sufficient support. To avoid executing the full motion and risking an additional fall, the keyframe engine monitors the convergence of the commanded arm angles using the `armAngleReached` condition. If the arms remain blocked, the motion is aborted and the stand-up routine is restarted from *lying*, as shown in Figure 5.2. In contrast to failures detected during later stages of the stand-up process, this type of early abort does not update the stand-up statistics (Subsection 5.2.3).

The reason is that arm blockage at this stage is often caused by external disturbances rather than by the stand-up motion itself. Typical examples include another robot colliding with the robot lying on the ground or contact with a goal post. Although worn joints or increased joint play may also prevent the arms from reaching their target angles, the robot cannot reliably determine whether an arm blockage during the early stand-up phase is caused by external contact or by its own hardware condition. While the broken-joint detector can identify specific internal defects such as rattling or spinning gearboxes, it does not cover all forms of arm blockage, nor can it distinguish reliably between environmental constraints and mechanical limitations. Unlike humans, the NAO robot does not possess distributed tactile sensing (“skin”), and therefore lacks the ability to sense whether the arm is pressed against an obstacle or restricted by its own hardware. Because the cause of the blockage is ambiguous, such early aborts are not treated as failures of the stand-up variant and are excluded from the statistical update mechanism. If a genuine hardware defect is present, the broken-joint detector prevents further stand-up attempts altogether, whereas external disturbances should not penalise the corresponding motion in the long-term statistics.

In later stages of the stand-up process, the torso orientation becomes the dominant indicator for whether an interception or a full abort is required. For selected keyframes, the engine relies on a target body angle and an admissible error interval. These thresholds are derived from recorded torso-angle curves of successful and failed attempts. By analysing the time evolution of the torso angle for each relevant keyframe, an admissible tolerance range is defined, ensuring that interception is triggered only when recovery is no longer feasible.

If the measured torso angle leaves the admissible interval, the robot is considered out of balance and unlikely to complete the current stand-up motion successfully. Depending on the robot’s posture and height above the ground, different interception strategies are available. When the center of gravity is still close to the ground, the default interception is the *play dead* motion, which switches all joints to an unstiff mode. This behavior

minimises impact forces and is faster to execute than moving the joints into a specific protective posture. It therefore acts as a simple, low-level fall-down protection for early stand-up failures, but it does not replace the more comprehensive fall-down protection mechanisms that take over once the robot is upright.

If the robot has already reached a sufficiently upright configuration but has not yet entered the *stand* phase, the engine can switch from *play dead* to a dedicated *sit* motion. The idea of catching a backward stumble with a sitting movement was first proposed by B-Human [57], although it is not used in their current implementation. In our framework, the *sit* motion is triggered when the robot is upright enough to perform the movement safely, the feet still provide support, and the torso angle indicates that the current stand-up variant cannot be completed successfully. Executing *sit* reduces both the fall height and the time required to resume the stand-up routine compared to restarting from a fully lying posture.

A separate motion, referred to as "rip", is not part of the interception system but serves as a fallback mechanism when the robot can no longer continue the current stand-up motion. Unlike interception movements, which aim to stabilise the robot and allow the stand-up routine to proceed, *rip* is intentionally slow and designed to bring the robot into a safe lying posture with minimal mechanical stress. The original motivation for this motion was to provide a controlled way for the robot to return to the ground when no feasible continuation of the stand-up sequence exists, ensuring that it remains in a safe configuration rather than collapsing abruptly, hence the name "rest in peace".

In practice, *rip* has also proven useful in certain front stand-up variants. If the robot attempts to push itself upward with the arms but fails to generate sufficient momentum, the motion allows it to slide back toward the ground in a controlled manner instead of falling. Once the center of gravity is sufficiently elevated, however, the robot no longer uses *rip* and instead relies on the regular fall-down protection, as *rip* is specifically designed for low-height configurations.

Figure 5.3 illustrates the resulting *sit* posture, and Figure 5.4 summarises the corresponding decision structure. The figure distinguishes forced transitions (red arrows), environmentally influenced transitions (green arrows), and learned transitions (blue arrows), where the latter are influenced by the stand-up statistics in Subsection 5.2.3. Dashed arrows indicate transitions that return to the main stand-up decision tree in Figure 5.2. If the interception is successful, the robot continues the stand-up routine from the *sit* state. This is decided primarily based on the torso angle: as long as the upper body remains within the admissible range for a sitting configuration, the robot is considered *not lying*, and the motion is treated as successful. In this case, the time to stand up is reduced compared to restarting from *lying*, and the distance of any residual fall is smaller.

If the *sit* motion does not stabilise the robot, the engine falls back to the *lying* state in Figure 5.2 and restarts the stand-up routine. Although the *sit* interception is not

successful in all trials and its success rate depends strongly on wear and joint play, it still yields a measurable improvement for robots that can execute the motion reliably. The main challenge is to trigger the interception at the correct time: the robot must have enough time to complete the *sit* motion, but the interception should only be activated when no other safe continuation of the current stand-up variant is available.

To configure these interception points, we first determine for which keyframes an abort or interception is meaningful. For these frames, the target torso angle is measured using a debugging mode, and an admissible tolerance is derived by comparing the torso angle trajectories of successful and failed attempts. Additionally, the fall direction is taken into account, since the *sit* motion is specifically designed for backward stumbling and would be counterproductive in other directions. If the robot is no longer in an *on ground* state, is currently executing a stand-up motion, the torso angle leaves the allowed range, and the fall direction is backward, the keyframe engine triggers the *sit* interception instead of continuing the current stand-up motion.

Finally, the interception logic is integrated with other safety mechanisms of the keyframe engine. The `waitForStable` flag in selected keyframes ensures that the robot does not proceed while the torso is still oscillating, and the "flying" state, detected, among others, via the foot force sensors, overrides regular stand-up and interception behavior to guarantee a safe response when a human lifts the robot. Together with the arm blockage detection and the stand-up statistics in Subsection 5.2.3, these interception movements form a coherent safety layer around the stand-up routines, reducing unnecessary falls and limiting mechanical stress on the robot.

5.3.3. Compensation of Wear

Joint play and mechanical wear are persistent challenges in the SPL, as also noted by Fiedler et al. [58]. With increasing operating time, the NAO's joints, particularly hip, knee, and ankle pitch, develop asymmetric backlash and reduced stiffness. These effects lead to systematic deviations between commanded and measured joint angles. For highly dynamic motions such as standing up, even moderate deviations can accumulate across successive keyframes and ultimately cause instability or complete failure of the motion.

To mitigate these effects, we introduce a dedicated wear compensation mechanism that operates inside the keyframe engine during selected, critical partial movements. These partial movements are marked by red dots in Figure 5.2 and Figure 5.4, and correspond to phases in which the legs undergo large coordinated movements and the robot is most sensitive to wear-induced deviations.

Relation to B-Human. Our approach is conceptually related to the joint compensation mechanism used by B-Human [12]. Their system continuously estimates joint play and integrates this estimate into the entire motion pipeline, including walking, balancing, and

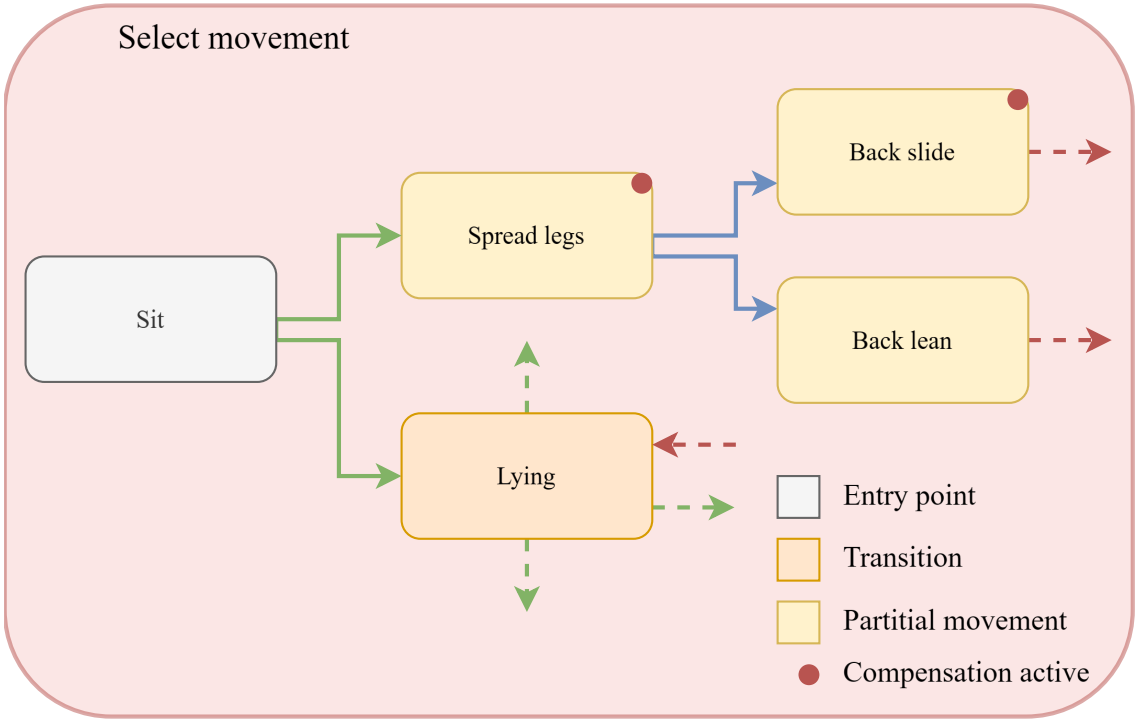


Figure 5.4.: Overview of our interception movement process, it involves forced transitions (red arrows), environmentally influenced transitions (green arrows), and learned transitions (blue arrows). The dashed lines indicate the transitions returning to Figure 5.2. First published in [4].

keyframe motions. This global strategy is highly effective in their architecture, which is designed around a unified motion framework with consistent timing and filtering.

In contrast, our stand-up routine is executed in a separate keyframe engine that is not tightly integrated with the rest of the motion framework. Adopting B-Human's global joint-play estimation approach would therefore have required substantial architectural changes, as their method relies on a system-wide integration of joint-play prediction, balancing, and keyframe interpolation. Rather than restructuring the entire motion pipeline, we opted for a local, motion-specific compensation mechanism that operates only during essential partial movements and directly uses the instantaneous joint error. This design allows the compensation to remain lightweight while still addressing the wear-induced deviations most relevant for the stand-up routine.

Dynamic selection of the compensated joint. Instead of statically defining which joint should be compensated, as done in B-Human's configuration, our method dynamically selects the joint with the largest instantaneous error. For each leg, the hip pitch, knee pitch, and ankle pitch joints are evaluated, and the joint with the largest absolute deviation between measured and commanded angle is chosen as the primary source of error. This choice is recomputed at every control cycle, allowing the system to adapt to robot-specific wear patterns and to changes in the mechanical state during the motion.

The compensation is then distributed across neighbouring joints using configurable correction factors. This avoids overloading a single joint and allows the system to counteract the effect of joint play even when the problematic joint itself cannot move reliably.

Thresholds and parameter selection. To avoid premature or oscillatory compensation, the mechanism is activated only when the error of at least one joint exceeds a threshold of 8 degree. This value was determined empirically: lower thresholds caused the compensation to trigger too early, interfering with otherwise stable motions, while higher thresholds delayed the correction to a point where recovery became unreliable.

The maximum compensation applied to any joint is limited to 10 degree in general, but can be manually set for specific motions. This upper bound reflects the practical limits of our motion pipeline: larger corrections produced unnatural movements and reduced stability, whereas compensations up to 10 degree were feasible and effective across all tested robots. These values were obtained through extensive testing on multiple robots with varying degrees of wear, evaluating the stand-up performance across different stages of joint degradation and iteratively adjusting the parameters until the compensation is triggered neither too early nor too late.

Instantaneous error computation. The compensation is based on the direct difference between measured and commanded joint angles. This avoids the temporal delay that would

be introduced by additional filtering or prediction and proved essential for maintaining the responsiveness required for fast keyframe-based motions. Once a joint exceeds the compensation threshold, the error is accumulated in a short buffer to smooth out noise, and the resulting average is used to compute the correction applied to neighbouring joints.

Transition between keyframes. To avoid introducing discontinuities when switching from one keyframe to the next, the compensation is gradually reduced before a transition occurs. This is controlled by a *compensation reduction factor*, which gradually decreases the applied correction as the end of the current keyframe is approached. Without this mechanism, the compensation computed for the current posture would carry over abruptly into the next keyframe, potentially creating an artificial error at the beginning of the subsequent interpolation phase. By fading out the correction smoothly, the system preserves the benefits of instantaneous error compensation while ensuring clean transitions between keyframes.

5.4. Robot Safety

Safe execution of stand-up motions requires a coordinated interaction between the motion system and the FDIR mechanisms introduced in Chapter 4. While the previous sections focused on the design of individual stand-up motions and their supporting mechanisms, this section outlines how these components form a coherent safety architecture that ensures robust behavior despite mechanical wear, external disturbances, and sensor uncertainties. In practice, this means that the robot must never execute a motion that exceeds its mechanical capabilities or current stability limits.

5.4.1. Safety Architecture Overview

The safety concept of our framework is organised into three complementary layers:

1. **FDIR**: continuous monitoring of joint behavior, inertial measurements, motor currents, and contact states to detect abnormal conditions.

2. **Motion-Level Safeguards**: preventive mechanisms inside the keyframe engine that stop unsafe execution before instability occurs.

3. **Interception and Fall-Down Protection**: reactive mechanisms that limit the consequences of failures once instability is unavoidable.

These layers operate concurrently and exchange information through lightweight representations, enabling rapid reactions to emerging risks.

5.4.2. Interaction Between FDIR and Motion Control

FDIR continuously evaluates the robot's physical state and provides safety-relevant information to the motion system. When an anomaly is detected, the keyframe engine adapts its behavior accordingly:

- **Stuck joints**: trigger arm-blockage handling or activate wear compensation when deviations exceed the configured threshold.
- **Motor malfunction**: suppresses further motion commands and forces a transition into a safe posture.
- **Flying state**: overrides all stand-up and interception behavior to prevent dangerous movements when the robot is lifted by a human.
- **Unstable torso orientation**: activates break-up logic and selects an appropriate interception motion.

This tight coupling ensures that the keyframe engine never executes a motion blindly but continuously adapts to the robot's physical condition.

5.4.3. Safety Despite Mechanical Wear

Mechanical wear introduces systematic deviations that can accumulate during a stand-up motion. The wear compensation mechanism described in Subsection 5.3.3 reduces these deviations during critical phases by adjusting neighbouring joints based on the instantaneous error.

From a safety perspective, this mechanism contributes to:

1. **Stabilization**: reducing the likelihood that wear-induced deviations push the robot outside the admissible torso angle range.
2. **Prevention of cascading failures**: preventing small deviations from accumulating across keyframes and triggering unnecessary break-ups or falls.

The compensation is intentionally limited to selected keyframes and capped in magnitude to maintain predictable behavior.

5.4.4. Avoiding Unsafe or Unintended Movements

Several safeguards inside the keyframe engine prevent the robot from executing motions under unsafe conditions:

- **Wait-for-stable**: prevents transitions while the torso is still oscillating.

- **Arm-angle-reached**: ensures that the arms are in a safe configuration before continuing the motion.
- **Arm-blockage detection**: pauses or redirects the motion when the arms are obstructed by the environment or another robot.
- **Break-up thresholds**: abort the current motion when the torso angle leaves the admissible range for the respective keyframe.
- **Flying-state override**: suppresses all motion commands when the robot is lifted, preventing dangerous joint movements.

These mechanisms ensure that the robot does not continue a motion that cannot be executed safely.

In addition to these safeguards, the keyframe engine also verifies whether a stand-up motion is required at all and whether the correct variant has been selected. If the robot is already in a stable sitting posture, the engine transitions directly to the *stand* motion instead of executing a full stand-up routine. Similarly, if the requested stand-up variant does not match the robot's actual fall direction, the engine overrides the request and selects the appropriate motion (e.g. *standUpBack* instead of *standUpFront*). This prevents unnecessary or incorrect movements and protects the robot from avoidable mechanical stress.

While the previous safeguards aim to prevent unsafe execution, interception mechanisms handle situations in which a stand-up attempt becomes unrecoverable.

5.4.5. Fall Safety and Interception

When a stand-up attempt becomes unrecoverable, the system transitions to one of several interception motions designed to minimise impact forces and reduce fall height:

- **Play Dead**: used when the center of gravity is still low; joints are relaxed to minimise impact.
- **Sit Interception**: used when the robot is upright enough to transition into a sitting posture and stumbling backwards, reducing fall height and recovery time.
- **Rip Motion**: a controlled return to the ground when the robot cannot continue the current stand-up motion from a low-height configuration.
- **Fall-Down Protection**: activated once the robot is upright, providing high-level protection against uncontrolled falls.

The selection of the appropriate interception motion is based on torso angle, fall direction, and height above the ground, ensuring that the robot always transitions into the safest available posture.

5.5. Evaluation

In this section, we evaluate the proposed stand-up framework with respect to robustness, generalization across different floor conditions, and performance on robots with varying degrees of mechanical wear. We compare our approach to the stand-up routine included in the 2022 code release [14], which serves as a baseline. All experiments were conducted on real NAO robots.

5.5.1. Experimental Setup

We tested three floor conditions with increasing levels of compliance:

1. artificial grass in our laboratory (arena),
2. artificial grass covered by an additional carpet layer,
3. artificial grass with two layers of 4 mm pool mats underneath the carpet.

These surfaces introduce progressively softer and more irregular conditions, which are known to influence stand-up performance. We selected two worn-out robots for the experiments, with hardware quality scores of 0.0 (NAO robot Blade [B]) and 0.35 (NAO robot Cyclops [C]). The hardware quality is defined between 0 and 1 and reflects the condition of the leg joints, where 1.0 corresponds to a new robot.

Each robot performed eight stand-up attempts from both lying on the front and lying on the back for each floor condition. For the new version, we reset the stand-up statistic described in Subsection 5.3.1 between front and back evaluations to avoid interactions between the two learning processes. We counted an attempt as successful when the robot reached a stable standing posture without entering an abort state. For attempts involving the *sit* interception, we considered the attempt successful as long as the robot did not fall back to the ground.

5.5.2. Baseline: 2022 Stand-Up Routine

The baseline system provides two fixed stand-up motions from the front and one from the back. When a front stand-up fails, the baseline alternates between the two predefined motions. It does not include online learning, interception strategies, or wear compensation. This makes the baseline sensitive to floor variations and mechanical wear, but it offers a consistent reference for comparison.

5.5.3. Metrics

We evaluate the stand-up routines using the following metrics:

- **Success rate**: proportion of successful stand-up attempts.
- **Stability**: number of aborted motions or transitions into interception behaviors.
- **Safety events**: occurrences where we had to stop the test due to mechanical risk.
- **Wear sensitivity**: performance differences between robots with different hardware quality scores.

We did not use time-to-stand as a primary metric, since robustness and safety are the main focus of this work.

5.5.4. Results

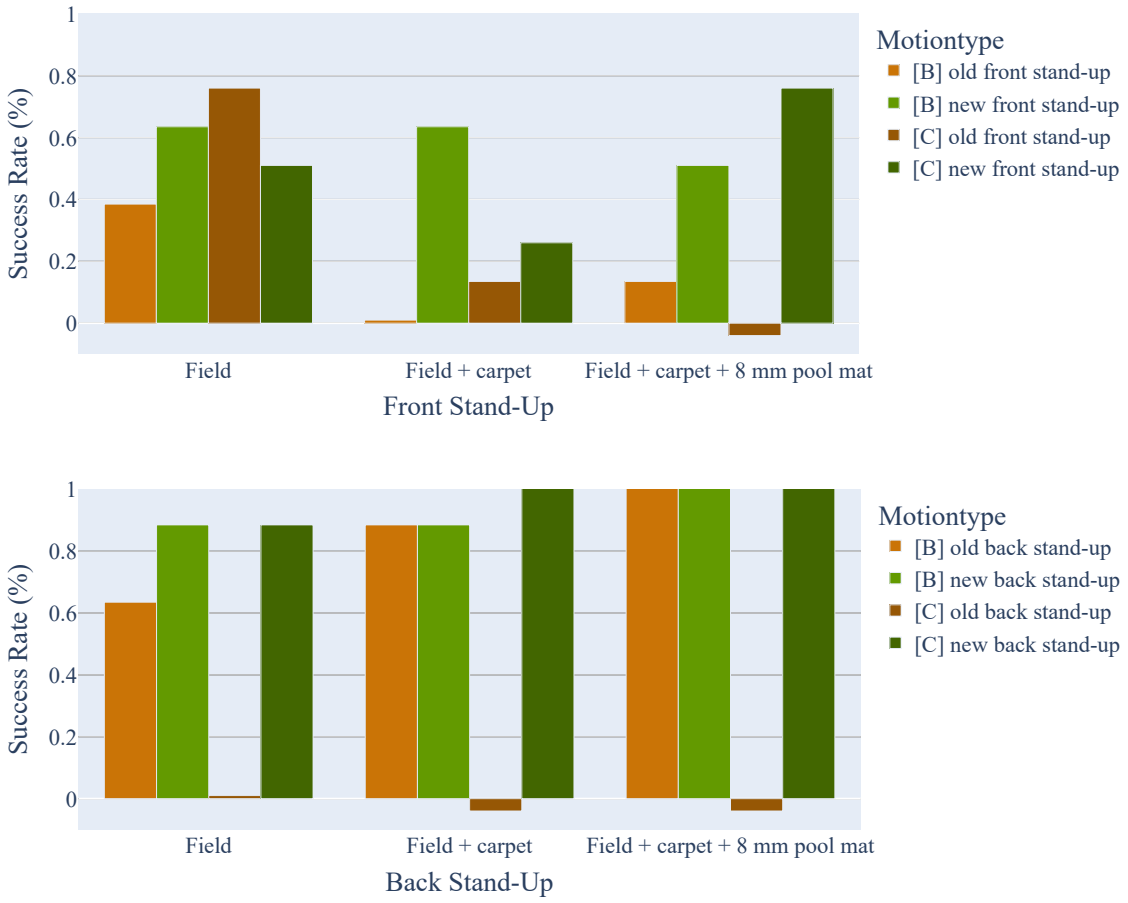


Figure 5.5.: These are the results of our tests with the new and old stand-up movement framework on two different robots. The negative value indicates that we had to stop the test because the robot broke down during it. First published in [4].

Figure 5.5 summarises the results for both robots across all floor conditions. For front stand-up motions, the new routine generally performs equally well or better than the

baseline, with one exception on the laboratory field: for robot C, the baseline achieves a higher success rate than the new version. On the softer surfaces, however, the new routine outperforms the baseline for both robots. For robot C, the baseline test on the softest surface had to be stopped early for safety reasons, which is reflected by the negative success-rate entries in Figure 5.5.

Robot B, which has a hardware quality of 0.0 and is therefore heavily worn, shows consistently lower success rates with the baseline, especially on the softer surfaces. With the new routine, its success rates increase across all floor conditions. Robot C, with a hardware quality of 0.35, performs better with the baseline on the laboratory field for front stand-up motions, but the new routine provides clear improvements on the softer surfaces. On the softest configuration, the baseline fails entirely for robot C and the test had to be stopped early, while the new routine completes the evaluation without safety issues.

For back stand-up motions, the new routine achieves consistently high success rates across all tested surfaces for both robots. On the laboratory field, it improves the performance of robot B compared to the baseline and enables robot C to stand up reliably, whereas the baseline fails for robot C. On the softer surfaces with additional layers, the success rates for robot B remain similar for both the old and the new routine, while the baseline for robot C fails in both cases and the tests had to be aborted on these configurations. In contrast, the new routine completes all evaluations for robot C without safety issues.

Across all robots and surfaces, stand-up motions from the back are more reliable than those from the front. This observation aligns with earlier findings in the literature, such as Stückler et al. [51], and reflects the mechanical limitations of the NAO arms, which must support a significant portion of the robot's weight during front stand-up motions.

5.5.5. Interpretation

The results indicate that the proposed stand-up framework improves robustness in several ways:

- **Generalization across surfaces**: the conditional movement selection adapts effectively to environmental variations, which stabilises performance on soft and irregular floors.
- **Wear tolerance**: robots with substantial mechanical degradation benefit from the joint-error compensation and the ability to avoid motion sequences that repeatedly fail.
- **Safety**: the new routine avoids dangerous failure modes observed in the baseline, particularly for heavily worn robots.

At the same time, the evaluation confirms that front stand-up motions remain more challenging than back stand-up motions. This limitation stems from the NAO's kinematics and cannot be fully mitigated by the proposed framework.

5.5.6. Limitations

The evaluation also reveals several limitations. The *sit* interception does not succeed in all cases and depends strongly on the robot's wear level. The stand-up statistic can be influenced by human interaction during testing, for example when the robot is laid down too quickly after standing up. Furthermore, we did not include new robots in the evaluation for safety reasons, although observations from recent test games suggest that the new routine performs reliably on intact hardware.

6. Conclusion

This thesis addressed four tightly connected challenges that arise when humanoid robots operate autonomously in acoustically complex and dynamically changing environments: robust whistle detection, sound source localization, fault detection and diagnosis, and adaptive motion control. Although these challenges were studied within the context of the RoboCup SPL, their relevance extends beyond this setting. They reflect fundamental requirements for mobile robots that must perceive their surroundings reliably, assess their own mechanical condition, and adapt their behavior without external assistance.

We first investigated whistle recognition, a task that is essential for rule-compliant behavior in the SPL and representative of real-time acoustic event detection under noise. The competition environment is characterised by overlapping matches, crowd noise, and high-frequency disturbances that frequently mask or imitate whistle-like signals. To address these conditions, we combined a conventional signal-processing pipeline with a compact neural network and a hybrid fusion mechanism. This approach leverages both domain knowledge and learned spectral representations, resulting in a robust and computationally efficient recogniser that operates reliably on the NAO V6 robot. The evaluation demonstrated that the hybrid system outperforms existing approaches under both laboratory and in-game conditions, particularly in scenarios with strong background noise.

Building on this capability, we extended whistle perception to include sound source localization. Determining whether a detected whistle originates from the robot's own field is essential for avoiding false reactions during matches. We developed neural network models for estimating both direction and distance using the NAO's four-microphone array. The models operate on frequency-domain representations that capture spatial information through inter-microphone differences. The evaluation across multiple tournaments showed that the system provides qualitatively reliable direction estimates under real-world conditions, despite reverberation, reflections, and heterogeneous noise profiles. While systematic mirroring effects remain a challenge due to the symmetric microphone geometry, the results demonstrate that neural network-based localization is feasible on the NAO platform and can support decision-making in dynamic environments.

The third component of this thesis focused on fault detection, isolation, and recovery. Mechanical wear is unavoidable in humanoid robots, and early detection of joint defects is essential for preventing consequential damage. We combined audio-based anomaly detection with sensor-based residuals in a rule-based FDIR framework. The audio module

identifies characteristic signatures of rattling and spinning gearboxes, while the sensor-based module evaluates joint currents, stiffness, angle deviations, and contact states. The fusion of both modalities enables a more reliable classification of joint conditions than either modality alone. The evaluation on laboratory data and logs from the RoboCup German Open 2025 showed that the system achieves high precision and responds quickly enough to prevent further damage, even under noisy and unpredictable conditions.

Finally, we addressed the question of how a robot should move once faults or unstable conditions are detected. We developed a modular stand-up framework that integrates conditional movement selection, interception mechanisms, and compensation for mechanical wear. The keyframe engine provides the underlying execution structure, while online statistics guide the selection of stand-up variants based on past success rates. Interception movements reduce fall height and prevent unnecessary failures, and the wear compensation mechanism counteracts deviations caused by joint play. The evaluation across different floor conditions and hardware states showed that the new stand-up routine improves robustness, particularly for worn robots and soft surfaces, and avoids dangerous failure modes observed in the baseline system.

Taken together, the four components of this thesis form a coherent framework for robust audio perception and fault-aware behavior on the NAO robot. The integration of whistle detection, whistle localization, multimodal FDIR, and adaptive motion control demonstrates how complementary perception and control modules can be combined to increase autonomy and resilience in real-world environments. Although the SPL has been discontinued, the methods developed in this work remain relevant for humanoid robotics more broadly. They address fundamental challenges in onboard audio processing, self-diagnosis, and adaptive whole-body motion, and they illustrate how tightly coupled perception and control can enhance the long-term operability of autonomous robots.

A. Appendix

A.1. Whistle Localization In-Game Evaluation

Table A.1.: In-game whistle localization results for the NAO robot Redskull [R] (player number 1) during RoboCup 2025 Round 4 against B-Human on Field D. The table lists the predicted whistle direction, the estimated distance interval, and the robot's self-localised pose (rotation and planar translation) at the time of each whistle event. The horizontal separator marks the half-time break, and the grey rows show the ideal reference values.

Nr.	Direction	Distance	Rotation	Translation x	Translation y	Category
Ref	123.7 °	5.4 m	0 °	-4.5 m	0 m	
1	312 °	6.75–9 m	-5.2 °	-4.3 m	0.05 m	180 °
2	117 °	6.75–9 m	-0.5 °	-4.26 m	-0.03 m	close
3	143 °	6.75–9 m	-6.3 °	-4.3 m	0.025 m	close
4	163 °	6.75–9 m	1.8 °	-4.3 m	0.086 m	moderate
Ref	56.3 °	5.4 m	0 °	-4.5 m	0 m	
5	331 °	6.75–9 m	-6.3 °	-4.3 m	0.05 m	∼ FB-mirror
6	101 °	6.75–9 m	0.7 °	-4.3 m	0.1 m	wrong
7	10 °	6.75–9 m	-9.5 °	-4.3 m	0.04 m	wrong
8	112 °	6.75–9 m	0.3 °	-4.3 m	0 m	LR-mirror
9	50 °	6.75–9 m	-1.2 °	-4.3 m	-0.08 m	close
10	307 °	6.75–9 m	2.1 °	-4.25 m	0.05 m	FB-mirror

Table A.2.: In-game whistle localization results for the NAO robot Redskull [R] (player number 1) during RoboCup 2025 quarter final against B-Human on Field B. The table lists the predicted whistle direction, the estimated distance interval, and the robot's self-localised pose (rotation and planar translation) at the time of each whistle event. The horizontal separators mark the breaks (referee timeout and half-time), and the grey rows show the ideal reference values.

Nr.	Direction	Distance	Rotation	Translation x	Translation y	Category
Ref	56.3 °	5.4 m	0 °	-4.5 m	0 m	
1	106 °	2–4.5 m	-84.8 °	-3.8 m	3.03 m	close
2	139 °	6.75–9 m	-0.8 °	-4.4 m	0.03 m	LR-mirror
3	359 °	6.75–9 m	1.3 °	-4.3 m	0 m	wrong
4	37 °	6.75–9 m	4.4 °	-4.3 m	0.03 m	close
5	326 °	6.75–9 m	-8.5 °	-4.3 m	0.09 m	FB-mirror
6	131 °	6.75–9 m	-90 °	≈ -3.2 m	≈ 3.3 m	moderate
7	49 °	6.75–9 m	8.9 °	-4.3 m	0.07 m	close
8	234 °	6.75–9 m	2.3 °	-4.25 m	-0.01 m	180 °
9	137 °	6.75–9 m	-90 °	≈ -3.2 m	≈ 3.3 m	moderate
10	182 °	6.75–9 m	3.8 °	-4.3 m	0.01 m	wrong
Ref	123.7 °	5.4 m	0 °	-4.5 m	0 m	
11	240 °	2–4.5 m	-10.8 °	-4.3 m	0.07 m	FB-mirror
12	119 °	6.75–9 m	10.6 °	-4.3 m	0.01 m	close

Table A.3.: In-game whistle localization results for the NAO robot Antman [A] (player number 7) during RoboCup 2024 Round 5 against B-Human on Field D. The table lists the predicted whistle direction, the estimated distance interval, and the robot's self-localised pose (rotation and planar translation) at the time of each whistle event. The horizontal separator marks the half-time break. Two whistle events could not be evaluated: the first whistle was not detected, and the second produced no usable localization due to the game situation.

Nr.	Direction	Distance	Rotation	Translation x	Translation y	Category
1	174 °	6.75–9 m	-1.7 °	-1.1 m	0 m	close
2	257 °	6.75–9 m	-17.8 °	-3.8 m	1.18 m	FB-mirror
3	113 °	6.75–9 m	-8.8 °	-3.6 m	0.5 m	close
4	72 °	6.75–9 m	27.2 °	-2.5 m	0.46 m	moderate
5	-	-	-	-	-	
6	256 °	6.75–9 m	45.5 °	-0.5 m	-2.06 m	wrong
7	130 °	6.75–9 m	92 °	0.6 m	-3.6 m	close
8	101 °	6.75–9 m	92 °	0.6 m	-3.6 m	close
9	-	-	-	-	-	
10	339 °	6.75–9 m	-3.3 °	-3.6 m	-0.3 m	FB-mirror
11	317 °	6.75–9 m	30.7 °	-0.65 m	-0.2 m	FB-mirror
12	170 °	6.75–9 m	12 °	-3.6 m	-0.46 m	wrong

Table A.4.: In-game whistle localization results for the NAO robot Kronos [K] (player number 1) during RoboCup 2024 Round 3 against Bembelbots on Field C (Outdoor-Field). The table lists the predicted whistle direction, the estimated distance interval, and the robot's self-localised pose (rotation and planar translation) at the time of each whistle event. The horizontal separators mark the breaks (half-time and robot reboot), and the grey rows show the ideal reference values.

Nr.	Direction	Distance	Rotation	Translation x	Translation y	Category
Ref	123.7 °	5.4 m	0 °	-4.5 m	0 m	
1	218 °	2–4.5 m	-3.65 °	-4.5 m	-0.06 m	wrong
2	277 °	6.75–9 m	90 °	≈ -3.2 m	≈ -3.3 m	∼ FB-mirror
Ref	56.3 °	5.4 m	0 °	-4.5 m	0 m	
3	254 °	2–4.5 m	14.9 °	-4.5 m	0.07 m	wrong
4	123 °	2–4.5 m	4.3 °	-4 m	0.03 m	LR-mirror
5	258 °	6.75–9 m	90 °	≈ -3.2 m	≈ 2 m	wrong

Table A.5.: In-game whistle localization results for robot K (player number 1) during RoboCup 2023 Round 2 against B-Human on Field E. The table lists the predicted whistle direction, the estimated distance interval, and the robot's self-localised pose (rotation and planar translation) at the time of each whistle event. The horizontal separator marks the half-time break, and the grey rows show the ideal reference values.

Nr.	Direction	Distance	Rotation	Translation x	Translation y	Category
Ref	56.3 °	5.4 m	0 °	-4.5 m	0 m	
1	120 °	6.75–9 m	1.2 °	-4.3 m	0.03 m	∼ LR-mirror
2	38 °	6.75–9 m	5.8 °	-4.3 m	0.02 m	moderate
3	357 °	6.75–9 m	-1.9 °	-4.2 m	-0.125 m	wrong
4	358 °	6.75–9 m	13.2 °	-4.3 m	0.06 m	wrong
Ref	123.7 °	5.4 m	0 °	-4.5 m	0 m	
5	116 °	6.75–9 m	-6.3 °	-4.2 m	-0.04 m	close
6	115 °	6.75–9 m	-1.4 °	-4.2 m	-0.04 m	moderate
7	114 °	6.75–9 m	-3.9 °	-4.2 m	-0.29 m	close
8	101 °	6.75–9 m	2.6 °	-4.2 m	0.09 m	moderate
9	162 °	6.75–9 m	0 °	-4.2 m	-0.08 m	moderate
10	146 °	6.75–9 m	-6.6 °	-4.2 m	-0.21 m	moderate

List of Figures

1.1 NAO V6 [5]. 3
1.2 NAO V6 microphone positioning. First published in [2]. 4
1.3 Head joints (Yaw, Pitch) [7]. 5
1.4 Arm joint structure left arm [7]. 5
1.5 Hip joint structure [7]. 5
1.6 Left leg joint structure [7]. 5
1.7 Side view of the NAO V6 camera field of view [7]. 6
1.8 Top-down view of the NAO V6 camera field of view [7]. 6

2.1 Overview of our whistle recognition pipeline where red arrows indicate the flow of FFT information of the current audio and blue arrows indicate the confidence flow. First published in [1]. 12
2.2 Spectrogram of different whistles. First published in [1]. 14
2.3 High-level overview of the proposed whistle recognition neural network. The numbers indicate the number of filters in each convolutional block, and each downward step represents a reduction of the feature-map length caused by striding and max-pooling. 25
2.4 Whistle comparison evaluation setup. Frameworks from left to right B-Human, the method presented here, our old approach and HULKs. First published in [1]. 32
2.5 Comparison of the influence of the different components of our approach on precision and recall during the Maker Faire Rome plus our old system since it proved to be promising in the previous tests. First published in [1]. . . . 33
2.6 Comparison of the influence of the different components of our whistle recognition on precision and recall under lab conditions plus our old system since it proved to be promising in the previous tests. First published in [1]. . . . 34

3.1 The robot is in a penalised position (left) and walking (right). First published in [2]. 40
3.2 Illustration of the five whistle heights used during data collection. 41
3.3 NAO robot with azimuth labels. First published in [2]. 43

3.4 Soccer field according to the rules [26] with walls coloured in purple. The circles mark the positions of the robots K, B, and V, while the dots mark the positions of the sound sources. Red points were used for both standing and walking-in-place recordings, while cyan points were used only for the standing evaluation. 46

3.5 Illustration of direct sound (a) and reflected sound (b) in our Arena (Lab). The NAO robot is positioned in the center circle, and the L-shaped walls surrounding the field are shown in purple. First published in [2]. 50

3.6 Overview of our proposed whistle direction estimation neural network architecture. The numbers indicate the filter count of the convolution or the number of dense units, and a step indicates that the input is reduced by half. 51

3.7 Hand whistle used in the SPL, showing the directional opening that shapes the emitted sound field. First published in [2]. 53

3.8 Overview of our proposed whistle distance estimation neural network architecture. The numbers indicate the filter count of the convolution or the number of dense units, and a step indicates that the input is reduced by half. 55

3.9 Idealised ground-truth geometry for kick-off whistles on both field sides, generated with GeoGebra. 62

3.10 Log visualizations of two kick-off whistles on Field D (RoboCup 2025). . . . 63

4.1 Overview of our anomaly detection neural network architecture. Numbers indicate the size of the input, arrows show the connections between the blocks, and each step represents a halving of the input size. First published in [3]. 76

4.2 Spectrogram illustrating the characteristics of silence (upper part) and a rattling gearbox (lower part), generated using the log file of the NAO robot Proteus [P]. First published in [3]. 78

4.3 Spectrogram illustrating the characteristics of silence (upper part) and a spinning gearbox (lower part), generated using the log file of the NAO robot Redskull [R]. First published in [3]. 78

5.1 Overview of the proposed stand-up architecture. The red box is detailed in Subsection 5.3.1, while the blue box is explained in Subsection 5.3.2 and Subsection 5.3.3. First published in [4]. 94

5.2 Overview of our movement selection process, it involves forced transitions (red arrows), environmentally influenced transitions (green arrows), and learned transitions (blue arrows). First published in [4]. 95

5.3 Robot in *sit* position. First published in [4]. 97

5.4 Overview of our interception movement process, it involves forced transitions (red arrows), environmentally influenced transitions (green arrows), and learned transitions (blue arrows). The dashed lines indicate the transitions returning to Figure 5.2. First published in [4]. 101
5.5 These are the results of our tests with the new and old stand-up movement framework on two different robots. The negative value indicates that we had to stop the test because the robot broke down during it. First published in [4]. 107

List of Tables

2.1 Rule-compliant robot responses to whistle-triggered game events in RoboCup SPL. Only behaviors directly implied by the rules are listed; team-specific strategies are not included. 13

2.2 Comparison of whistle and noise frequency ranges relevant for recognition. . 20

2.3 A comparison of how different noise from different approaches is detected as a whistle (False Positive Rate). Lower is better. 32

2.4 A comparison of how different noise affects the performance of varying whistle recognition frameworks (True Positive Rate). Higher is better. 33

3.1 Overview of recorded rooms, distances, and angular resolutions. 43

3.2 Training set: rooms and distances. 49

3.3 Test set: rooms and distances. 49

3.4 Evaluation set: rooms and distances (publicly released sessions). 49

3.5 Direction and distance errors in an evaluation with standing robots, the letters in brackets describe which robot was used for the evaluation. $\tilde{x}$, $\bar{x}$, and σ denote the median, mean error, and standard deviation. The distance error is normalized on the maximum deviation represented by four distance bins. Therefore, a normalized distance error of 0.25 corresponds to a deviation by one bin. 58

3.6 Direction and distance errors in an evaluation with walking robots, the letters in brackets describe which robot was used for the evaluation. $\tilde{x}$, $\bar{x}$, and σ denote the median, mean error, and standard deviation. The distance error is normalized on the maximum deviation represented by four distance bins. Therefore, a normalized distance error of 0.25 corresponds to a deviation by one bin. 59

3.7 In-game whistle localization results for the NAO robot Redskull [R] (player number 1) during RoboCup 2025 Round 4 against B-Human on Field D. The table lists the predicted whistle direction, the estimated distance interval, the calculated ground-truth, the error and the interpretation at the time of each whistle event. The horizontal separator marks the half-time break. 64

3.8 In-game whistle localization results for the NAO robot Redskull [R] (player number 1) during RoboCup 2025 quarter final against B-Human on Field B. The table lists the predicted whistle direction, the estimated distance interval, the calculated ground-truth, the error and the interpretation at the time of each whistle event. The horizontal separators mark the breaks (referee timeout and half-time). 65

3.9 In-game whistle localization results for the NAO robot Antman [A] (player number 7) during RoboCup 2024 Round 5 against B-Human on Field D. The table lists the predicted whistle direction, the estimated distance interval, the calculated ground-truth, the error and the interpretation at the time of each whistle event. The horizontal separator marks the half-time break. Two whistle events could not be evaluated: the first whistle was not detected, and the second produced no usable localization due to the game situation. 66

3.10 In-game whistle localization results for the NAO robot Kronos [K] (player number 1) during RoboCup 2024 Round 3 against Bembelbots on Field C (Outdoor-Field). The table lists the predicted whistle direction, the estimated distance interval, the calculated ground-truth, the error and the interpretation at the time of each whistle event. The horizontal separators mark the breaks (half-time and robot reboot). 66

3.11 In-game whistle localization results for robot K (player number 1) during RoboCup 2023 Round 2 against B-Human on Field E. The table lists the predicted whistle direction, the estimated distance interval, the calculated ground-truth, the error and the interpretation at the time of each whistle event. The horizontal separator marks the half-time break. 67

4.1 Overview of robots and defect types included in the audio anomaly detection dataset. 73

4.2 LED color coding for joint state feedback. 82

4.3 Overview of joint categories, detected states, and corresponding safety actions. 83

4.4 Detection performance of the FDIR system. 85

4.5 Confusion Matrix Lab Evaluation 86

4.6 Confusion Matrix RoboCup German Open 2025 86

4.7 Detection timings RoboCup German Open 2025. 87

A.1 In-game whistle localization results for the NAO robot Redskull [R] (player number 1) during RoboCup 2025 Round 4 against B-Human on Field D. The table lists the predicted whistle direction, the estimated distance interval, and the robot's self-localised pose (rotation and planar translation) at the time of each whistle event. The horizontal separator marks the half-time break, and the grey rows show the ideal reference values. 113

A.2 In-game whistle localization results for the NAO robot Redskull [R] (player number 1) during RoboCup 2025 quarter final against B-Human on Field B. The table lists the predicted whistle direction, the estimated distance interval, and the robot's self-localised pose (rotation and planar translation) at the time of each whistle event. The horizontal separators mark the breaks (referee timeout and half-time), and the grey rows show the ideal reference values. 114

A.3 In-game whistle localization results for the NAO robot Antman [A] (player number 7) during RoboCup 2024 Round 5 against B-Human on Field D. The table lists the predicted whistle direction, the estimated distance interval, and the robot's self-localised pose (rotation and planar translation) at the time of each whistle event. The horizontal separator marks the half-time break. Two whistle events could not be evaluated: the first whistle was not detected, and the second produced no usable localization due to the game situation. 115

A.4 In-game whistle localization results for the NAO robot Kronos [K] (player number 1) during RoboCup 2024 Round 3 against Bembelbots on Field C (Outdoor-Field). The table lists the predicted whistle direction, the estimated distance interval, and the robot's self-localised pose (rotation and planar translation) at the time of each whistle event. The horizontal separators mark the breaks (half-time and robot reboot), and the grey rows show the ideal reference values. 116

A.5 In-game whistle localization results for robot K (player number 1) during RoboCup 2023 Round 2 against B-Human on Field E. The table lists the predicted whistle direction, the estimated distance interval, and the robot's self-localised pose (rotation and planar translation) at the time of each whistle event. The horizontal separator marks the half-time break, and the grey rows show the ideal reference values. 116

List of Equations

Eq. 2.1: Magnitude . . . 16
Eq. 2.2: Relation of fundamental frequency and overtones . . . 19
Eq. 2.3: Dynamic Frequency Boundaries . . . 21
Eq. 2.4: Spatial adaptation of frequency boundaries . . . 22
Eq. 2.5: Whistle Confidence Physical Model . . . 24
Eq. 2.6: Noise Confidence . . . 30
Eq. 2.7: Fused Whistle Confidence . . . 30
Eq. 2.8: Fused and dampened Whistle Confidence . . . 30
Eq. 3.1: Unit circle representation of the angle . . . 52
Eq. 3.2: Direct-to-Reverberant Ratio per frequency . . . 54
Eq. 3.3: Per-bin similarity measure for distance prediction . . . 56
Eq. 3.4: Adapted focal loss for distance estimation . . . 56
Eq. 5.1: Update of aborted stand-up motions . . . 93
Eq. 5.2: Update of started stand-up motions . . . 93
Eq. 5.3: Failure rate of a stand-up variant . . . 93

List of Abbreviations

CMOS Complementary Metal-Oxide-Semiconductor. 4

CRNN Convolutional Recurrent Neural Network. 38

DoF Degrees of Freedom. 2, 3

DRR Direct-to-Reverberant energy Ratio. 39, 54, 55, 56, 59

ELU Exponential Linear Unit. 25

ENN Edited Nearest Neighbors. 26, 76

FDI Fault Detection and Isolation/Identification. 70, 71

FDIR Fault Detection, Isolation, and Recovery. 2, 69, 70, 71, 72, 74, 75, 84, 85, 89, 103, 104, 111, 112, 122

FFT Fast Fourier Transform. 12, 15, 20, 24, 26, 44, 77, 117

FSRs Force Sensitive Resistors. 3, 74

HRTF Head-Related Transfer Function. 7, 39, 40

ILD Inter-Microphone Level Differences. 51

IMU Inertial Measurement Unit. 3, 74

OC-NN One-Class Neural Networks. 70, 71

RAdam Rectified Adam. 26, 77

ReLU Rectified Linear Unit. 25, 51, 55

SMOTE Synthetic Minority Over-Sampling Technique. 26, 76

SPL Standard Platform League. 1, 2, 3, 6, 7, 11, 13, 14, 31, 32, 37, 38, 39, 53, 54, 60, 69, 72, 90, 100, 111, 112, 118

SRP-PHAT Steered-Response Power-Phase Transform. 39

SSL Sound Source Localization. 37, 38, 39, 49, 53

STFT Short-Time Fourier Transform. 38, 50, 54

TDOA Time Difference of Arrival. 7

ZCR Zero Crossing Rate. 23, 77, 79, 85

Bibliography

[1] D. Kleingarn and D. Brämer, "Neural network and prior knowledge ensemble for whistle recognition", in *Robot World Cup*, Springer, 2023, pp. 17–28.

[2] D. Kleingarn, D. Brämer, and R. Martin, "Direction and distance estimation of whistle events on a nao robot", in *RoboCup 2024: Robot World Cup XXVII*, E. Barros, J. P. Hanna, H. Okada, and E. Torta, Eds., Cham: Springer Nature Switzerland, 2025, pp. 3–15, ISBN: 978-3-031-85859-8.

[3] D. Brämer, D. Kleingarn, and P. Reichenberg, "Multimodal neural network enriched approach for real-time fault detection, isolation and recovery on a nao robot", in *RoboCup 2025: Robot World Cup XXVIII*, to appear, Cham: Springer Nature Switzerland, 2026.

[4] D. Kleingarn and D. Brämer, "Decision tree-like dynamic conditional stand-up routines for nao robots", in *RoboCup 2024: Robot World Cup XXVII*, E. Barros, J. P. Hanna, H. Okada, and E. Torta, Eds., Cham: Springer Nature Switzerland, 2025, ISBN: 978-3-031-85859-8.

[5] *NAO V6 Datasheet*, Accessed: 2025-04-25. [Online]. Available: `https://support.unitedrobotics.group/en/support/solutions/articles/80000959718-nao-technical-specifications`.

[6] RoboCup Technical Committee, *Robocup standard platform league (nao) rule book*, 2025. [Online]. Available: `https://spl.robocup.org/wp-content/uploads/SPL-Rules-2025.pdf`.

[7] *NAO V6 Documentation*, Accessed: 2025-12-16. [Online]. Available: `http://doc.aldebaran.com/1-14/family/robots/video_robot.html`.

[8] R. T. Committee, "Robocup standard platform league (nao) technical challenges", pp. 2–7, 2023. [Online]. Available: `https://spl.robocup.org/wp-content/uploads/SPL-Challenges-2023.pdf`.

[9] A. Bonarini, D. Lavatelli, and M. Matteucci, "A composite system for real-time robust whistle recognition", in *RoboCup 2005: Robot Soccer World Cup IX 9*, Springer, 2006, pp. 130–141.

[10] K. Poore, S. Abeyruwan, A. Seekircher, and U. Visser, "Single-and multi-channel whistle recognition with nao robots", in *RoboCup 2014: Robot World Cup XVIII 18*, Springer, 2015, pp. 245–257.

[11] HULKs Team, *HULKs Code Release*, version 2022, Dec. 2022. [Online]. Available: `https://github.com/HULKs/hulk/releases/tag/coderelease2021`.

[12] B-Human Team, *B-Human Code Release*, version 2022, Oct. 2022. [Online]. Available: `https://github.com/bhuman/BHumanCodeRelease/releases/tag/coderelease2022`.

[13] HTWK Team, *HTWK Whistle Detection*, version 2022, Feb. 2022. [Online]. Available: `https://github.com/NaoHTWK/WhistleDetection`.

[14] Nao Devils Team, *Nao Devils Code Release*, version 2022, Feb. 2022. [Online]. Available: `https://github.com/NaoDevils/CodeRelease/releases/tag/CodeRelease2022`.

[15] U. Forssell and P. Lindskog, "Combining semi-physical and neural network modeling: An example of its usefulness", *IFAC Proceedings Volumes*, vol. 30, no. 11, pp. 767–770, 1997.

[16] P. Vary and R. Martin, *Digital Speech Transmission and Enhancement, 2nd Edition*. John Wiley / IEEE Press, 2023.

[17] D. Kleingarn and D. Brämer, *Whistle detection dataset*, Dataset, Version V1, 2024. DOI: `10.17877/TUDODATA-2024-M0F6BMI1`. [Online]. Available: `https://doi.org/10.17877/TUDODATA-2024-M0F6BMI1`.

[18] D. Brämer and D. Kleingarn, *Nao Devils Audio Processing Tools*, version 2024. [Online]. Available: `https://github.com/NaoDevils/AudioProcessing`.

[19] D. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)", in *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2016.

[20] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique", *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[21] D. L. Wilson, "Asymptotic properties of nearest neighbor rules using edited data", *IEEE Transactions on Systems, Man, and Cybernetics*, pp. 408–421, 1972.

[22] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration", in *International Conference on Computer Vision Theory and Application VISSAPP'09)*, INSTICC Press, 2009, pp. 331–340.

[23] G. Lemaître, F. Nogueira, and C. K. Aridas, "Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning", *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 559–563, 2017.

[24] G. E. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data", *ACM SIGKDD explorations newsletter*, vol. 6, no. 1, pp. 20–29, 2004.

[25] L. Wright, *Ranger - a synergistic optimizer.* `https://github.com/lessw2020/Ranger-Deep-Learning-Optimizer`, 2019.

[26] R. T. Committee, "Robocup standard platform league (nao) rule book", 2024. [Online]. Available: `https://spl.robocup.org/wp-content/uploads/SPL-Rules-2024.pdf`.

[27] Y. Cho, D. Yook, S. Chang, and H. Kim, "Sound source localization for robot auditory systems", *Consumer Electronics, IEEE Transactions on*, vol. 55, pp. 1663–1668, Sep. 2009. DOI: `10.1109/TCE.2009.5278040`.

[28] X. Li, L. Girin, F. Badeig, and R. Horaud, "Reverberant sound localization with a robot head based on direct-path relative transfer function", in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, Oct. 2016. [Online]. Available: `http://dx.doi.org/10.1109/IROS.2016.7759437`.

[29] M. Zohourian, G. Enzner, and R. Martin, "Binaural speaker localization integrated into an adaptive beamformer for hearing aids", *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 3, pp. 515–528, 2018. DOI: `10.1109/TASLP.2017.2782491`.

[30] M. Zohourian and R. Martin, "Binaural direct-to-reverberant energy ratio and speaker distance estimation", *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 92–104, 2020. DOI: `10.1109/TASLP.2019.2948730`.

[31] S. Kushwaha, I. Roman, M. Fuentes, and J. Bello, "Sound source distance estimation in diverse and dynamic acoustic conditions", Oct. 2023, pp. 1–5. DOI: `10.1109/WASPAA58266.2023.10248194`.

[32] H. Kuttruff, *Room acoustics*. Crc Press, 2016.

[33] M. Sewtz, T. Bodenmüller, and R. Triebel, "Sound source localization for robotic application", Jun. 2020.

[34] P. Zahorik, D. S. Brungart, and A. W. Bronkhorst, "Auditory distance perception in humans: A summary of past and present research", *ACTA Acustica united with Acustica*, vol. 91, no. 3, pp. 409–420, 2005.

[35] M. Zohourian, R. Martin, and N. Madhu, "New insights into the role of the head radius in model-based binaural speaker localization", in *2017 25th European Signal Processing Conference (EUSIPCO)*, 2017, pp. 221–225. DOI: `10.23919/EUSIPCO.2017.8081201`.

[36] M. Zohourian, J. Stinner, and R. Martin, *Speaker distance estimation using binaural hearing aids and deep neural networks*. Universitätsbibliothek der RWTH Aachen, 2019.

[37] D. Kleingarn and D. Brämer, *Whistle localisation dataset*, Dataset, Version V1, 2024. DOI: 10.17877/TUDODATA-2024-M0FDQQYQ. [Online]. Available: https://doi.org/10.17877/TUDODATA-2024-M0FDQQYQ.

[38] D. Misra, "Mish: A self regularized non-monotonic activation function", *Proceedings of the British Machine Vision Conference 2020*, 2020.

[39] X. Lu, C. Ma, B. Ni, X. Yang, I. Reid, and M.-H. Yang, "Deep regression tracking with shrinkage loss", in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 353–369.

[40] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection", in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[41] G. Aaseng, M. Do, J. Frank, C. Fry, and A. Sweet, "Integrating planning, diagnosis and execution for vehicle systems management", in *33rd International Conference on Automated Planning and Execution*, 2023.

[42] M. A. Arwin, W. A. Saheri, N. A. A. Shashoa, and A. J. Abougarair, "Overview on fault detection and isolation", in *1st International Conference on Engineering and Applied Natural Sciences*, 2022.

[43] A. Wander and R. Förstner, *Innovative fault detection, isolation and recovery strategies on-board spacecraft: state of the art and research challenges*. Deutsche Gesellschaft für Luft-und Raumfahrt-Lilienthal-Oberth eV Bonn, Germany, 2013.

[44] E. Khalastchi and M. Kalech, "On fault detection and diagnosis in robotic systems", *ACM Computing Surveys (CSUR)*, vol. 51, no. 1, pp. 1–24, 2018.

[45] E. Chow and A. Willsky, "Analytical redundancy and the design of robust failure detection systems", *IEEE Transactions on automatic control*, vol. 29, no. 7, pp. 603–614, 1984.

[46] M. Thirumarimurugan, N. Bagyalakshmi, and P. Paarkavi, "Comparison of fault detection and isolation methods: A review", in *2016 10th International Conference on Intelligent Systems and Control (ISCO)*, IEEE, 2016, pp. 1–6.

[47] R. Chalapathy, A. Menon, and S. Chawla, "Anomaly detection using one-class neural networks", Feb. 2018. DOI: 10.48550/arXiv.1802.06360.

[48] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition", in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[49] C.-h. Chen, *Signal processing handbook*. CRC Press, 1988, vol. 51.

[50] D. Shete, S. Patil, and S. Patil, "Zero crossing rate and energy of the speech signal of devanagari script", *IOSR-JVSP*, vol. 4, no. 1, pp. 1–5, 2014.

[51] J. Stückler, J. Schwenk, and S. Behnke, "Getting back on two feet: Reliable standing-up routines for a humanoid robot.", in *IAS*, Citeseer, 2006, pp. 676–685.

[52] H. Jeong and D. Lee, "Learning complex stand-up motion for humanoid robots", in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, 2016.

[53] S. Stelter, M. Bestmann, N. Hendrich, and J. Zhang, "Fast and reliable stand-up motions for humanoid robots using spline interpolation and parameter optimization", in *2021 20th International Conference on Advanced Robotics (ICAR)*, IEEE, 2021, pp. 253–260.

[54] M. Mistry, A. Murai, K. Yamane, and J. Hodgins, "Sit-to-stand task on a humanoid robot from human demonstration", in *2010 10th IEEE-RAS International Conference on Humanoid Robots*, IEEE, 2010, pp. 218–223.

[55] P. Reichenberg, "RoboCup German Open 2024 Fall Statistic".

[56] P. Reichenberg, *RoboCup Fall Statistic 2023*, Jun. 2024. [Online]. Available: `https://b-human.informatik.uni-bremen.de/public/Statistics/2023/`.

[57] B-Human Team, *Keyframe Motions - B-Human*, Dec. 2023. [Online]. Available: `https://docs.b-human.de/coderelease2023/motion/motion-keyframemotionengine/#keyframe-parameters`.

[58] J. Fiedler and T. Laue, "Neural network-based joint angle prediction for the nao robot", in *Robot World Cup*, Springer, 2023, pp. 66–77.

List of Own Publications

About the Applicant:

Kleingarn, Diana
Last name, first name

Date and place of birth

Matriculation Number

List of scientific publications produced in the context of this doctoral research

[1] D. Kleingarn and D. Brämer, "Neural network and prior knowledge ensemble for whistle recognition", in *Robot World Cup*, Springer, 2023, pp. 17–28.

[2] D. Kleingarn, D. Brämer, and R. Martin, "Direction and distance estimation of whistle events on a nao robot", in *RoboCup 2024: Robot World Cup XXVII*, E. Barros, J. P. Hanna, H. Okada, and E. Torta, Eds., Cham: Springer Nature Switzerland, 2025, pp. 3–15, ISBN: 978-3-031-85859-8.

[3] D. Brämer, D. Kleingarn, and P. Reichenberg, "Multimodal neural network enriched approach for real-time fault detection, isolation and recovery on a nao robot", in *RoboCup 2025: Robot World Cup XXVIII*, to appear, Cham: Springer Nature Switzerland, 2026.

[4] D. Kleingarn and D. Brämer, "Decision tree-like dynamic conditional stand-up routines for nao robots", in *RoboCup 2024: Robot World Cup XXVII*, E. Barros, J. P. Hanna, H. Okada, and E. Torta, Eds., Cham: Springer Nature Switzerland, 2025, ISBN: 978-3-031-85859-8.

Affidavit

Affidavit:

I declare in lieu of an oath that I have completed all parts of this thesis independently and without the unauthorized assistance of third parties, and that I have not used any aids other than those mentioned and explicitly permitted, and that I have generally conducted myself in conformity with the examination. I also declare that when using IT-/AI-supported writing tools, I have listed these tools in full in the overview of aids used with their product name.

I am aware that cheating or attempted cheating will be punished according to the examination regulations applicable to me.

Eigenständigkeitserklärung:

Ich versichere, dass ich die vorliegende Arbeit in allen Teilen selbstständig und ohne unzulässige Hilfe Dritter absolviert sowie keine anderen als die genannten und explizit zugelassenen Hilfsmittel verwendet und mich im Allgemeinen prüfungskonform verhalten habe. Ich erkläre zudem, dass ich beim Einsatz von IT-/KI-gestützten Schreibwerkzeugen diese Werkzeuge in der Übersicht verwendeter Hilfsmittel mit ihrem Produktnamen vollständig aufgeführt habe.

Mir ist bewusst, dass Täuschungen bzw. Täuschungsversuche nach der für mich geltenden Prüfungsordnung geahndet werden.

Folgende Hilfsmittel habe ich für die Bearbeitung genutzt/I used the following tools for editing:

- Microsoft Copilot
- GitHub Copilot
- Grammarly
- DeepL

Ort, Datum Unterschrift

Zu dem/r Antragsteller/in:

Kleingarn, Diana

Name, Vorname

Geburtsdatum, -ort

Matrikelnummer

Legal Notice:

The submission of a statutory declaration in lieu of an oath ("eidesstattliche Versicherung") constitutes a confirmation of the accuracy of a statement that is subject to criminal liability under Sections 156 and 161 of the German Criminal Code (StGB). The submission of a false or incomplete statutory declaration in lieu of an oath is a criminal offense.

Anyone who intentionally submits a false statutory declaration in lieu of an oath may be punished with imprisonment for up to three years or with a fine (Section 156 StGB). Negligently submitting a false statutory declaration in lieu of an oath may be punished with imprisonment for up to one year or with a fine (Section 161 StGB).

Belehrung:

Die Abgabe einer eidesstattlichen Versicherung ist eine nach §§ 156, 161 Strafgesetzbuch (StGB) strafbewehrte Bestätigung der Richtigkeit einer Erklärung. Die Abgabe einer falschen oder unvollständigen Versicherung an Eides statt ist strafbar.

Wer vorsätzlich eine falsche Versicherung an Eides statt abgibt, kann mit einer Freiheitsstrafe bis zu drei Jahren oder mit Geldstrafe bestraft werden, § 156 StGB. Die fahrlässige Abgabe einer falschen Versicherung an Eides statt kann mit einer Freiheitsstrafe bis zu einem Jahr oder Geldstrafe bestraft werden, § 161 StGB.

Die oben stehende Belehrung habe ich zur Kenntnis genommen:

Ort, Datum

Unterschrift

Statutory Declaration in Lieu of an Oath

Being aware of the significance of a statutory declaration in lieu of an oath and of the criminal liability associated with submitting a false statutory declaration, I hereby declare under oath that I have prepared the present dissertation, entitled

Robust Auditory Perception and Fault Resilience for Humanoid Robots in Competitive Environments

independently and without any impermissible external assistance. I have used no sources other than those indicated and have marked all verbatim and paraphrased quotations accordingly.

Eidesstattliche Versicherung

In Kenntnis der Bedeutung einer eidesstattlichen Versicherung und der Strafbarkeit der Abgabe einer falschen eidesstattlichen Versicherung versichere ich hiermit an Eides statt, dass ich die vorliegende Dissertation mit dem Titel

Robust Auditory Perception and Fault Resilience for Humanoid Robots in Competitive Environments

selbstständig und ohne unzulässige fremde Hilfe angefertigt habe. Ich habe keine anderen als die angegebenen Quellen benutzt sowie wörtliche und sinngemäße Zitate kenntlich gemacht.

Ort, Datum Unterschrift