Study on Perception-Action Scheme for Human-Robot Musical Interaction in Wind Instrumental Play

Klaus Petersen

Graduate School of Advanced Science and Engineering

Waseda University

A thesis submitted for the degree of

Doctor of Philosophy

Version 2011-02-09

Acknowledgements

First and foremost I would like to thank my supervisor Professor Atsuo Takanishi. It has been an honor to be his Ph.D. student. He has taught me in many ways, how good robotics research is done. I appreciate all his contributions of time and ideas to make my Ph.D. experience productive and stimulating. The joy and enthusiasm he has for his research was contagious and motivational for me, even during tough times in the Ph.D pursuit.

Throughout the graduation process I have been strongly supported from my advisory Professors Prof. Mitsuo Umezu, Prof. Masakatsu Fujie and Prof. Shuji Hashimoto. Towards the completion of this work, they provided me with many ideas and corrections, that contributed very much to quality and relevance of this work. I would like to sincerely thank them for their on-going support.

I would like to express my deepest gratitude to my Ph.D. advisor Dr. Jorge Solis. Without his continuous support throughout these 4 years, this work would not have been possible. I especially appreciate his effort and patience in times, when the production of the results, that we had set out to achieve, became more difficult than expected. He has constantly helped me with good ideas and sincere advice, that have proven to be essential for the completion of this thesis.

In the process of finalizing this work, I have received valuable support from Dr. Kotaro Fukui. He has provided a great amount of help and patience, to find and correct flaws in the thesis drafts. Furthermore, he has helped me with the formal submission of several important Japanese forms, which I would not have been able to correctly file without his support. I would like to express my gratitude to him.

Two people, who were absolutely indispensable for completing this thesis, were the laboratory's two secretaries, Mrs. Hisako Ohta and Ms. Terumi Itou. Although I was notoriously late with filing documents, they not only always made great effort to correct my mistakes, but they also always have treated me with the greatest friendliness and respect. Their outstanding work is not only essential to me, but for the whole laboratory.

I was very lucky to be part of an outstandingly pleasant, productive and interesting team within the laboratory. The flute team has always provided me with an excellent environment to do my research work. Every time I had any question or problem, I was strongly supported in a friendly and efficient way. Takeshi Ninomiya, Testuo Yamamoto, Masaaki Takeuchi, Shinpei Ishikawa, Takafumi Kusano and Junpei Kashiwakura have been great company and great scientific support during these 4 years. I especially want to thank Yoshihisa Sugita and Kenji Ozawa for making huge efforts to finally successfully make the flutist robot perform well again. Thank you very much for the good work!

For several essential parts of this thesis I have worked together with the professional musicians Prof. Fujita and Prof. Miyazaki. I want to express my deepest gratitude to them to make the effort to come to laboratory so often, to perform many, sometimes unsuccessful experiments. I hope we can continue to work together in the future, to further improve the flute team's musical robots and the interaction system.

Apart from the people named above, also other members of the laboratory have provided constant support to me during the Ph.D. pursuit. Especially Yohan Noh, Zhuohua Lin, Qing Shi, Nobotsuna Endo and Aiman Moussa have always been there for me to talk things over or to consult with each other in difficult situations. Satoru Shouji, who is not directly a laboratory member, but has spent considerable

time here working on the airway robot, has always been wonderful company to talk, discuss research and have dinner.

I want to thank also the other research staff in Takanishi laboratory, who have not been mentioned above so far, Hiroyuki Ishii, Salvatore Sessa, Massimiliano Zecca, Hideki Kondo and Kenji Hashimoto. Their experienced support has been very valuable at all times during the Ph.D. pursuit. I also want to express my gratitude to all the other members of Takanishi Laboratory, who make with their hard work and kind nature this laboratory a good place to work at.

Special thanks go out to the GCOE Global Robot Academia, who has provided me with considerable funding for several years. I want to express my deepest gratitude to GCOE-GRA secretary Miwa Komatsu for her constant support, and for explaining things to me all over again, when I for once again couldn't figure out the correct procedure to fill in my documents, spend budget etc. Thank you very much for your patience.

For the first three years I spent in Japan, I was supported by a full-time Monbukagakusho scholarship provided by the Japanese Ministry of Education (MEXT). Life and work in Japan was very comfortable with this generous support. For the first year, a certain amount of his scholarship was provided by the German Academic Exchange Agency (DAAD).

My friends here in Tokyo, have always been there for me, also in times, when living and working in a foreign country was difficult for me. I want to express my deepest gratitude to all of you. I would like to especially thank a few friends, who have been there from the beginning, when I first moved here: Mari Yamada, Maria Suzukawa, Koichi Atsumi, Kong Pahurak and Neda Firfova, you have always been there, when I needed friendship and advice. From Germany, mainly via Skype, my great friend Tobias Schlueter has massively supported me during these years with his friendship and his sometimes critical, but always accurate advice. Thank you so much for that.

Finally, I would like to deeply thank my dear parents, for their endless patience, valuable advice, love and encouragement. Without your relentless support this work would not even have been started. Thank you.

Abstract

In recent years, there have been many attempts, to introduce robots to areas where they come into direct contact with people. Through the approach presented in this thesis, an anthropomorphic robot is to be enabled to act in the same environment as its human partner with the intention to perform a certain task. In contrast to interaction through predefined actions, the robot is to dynamically adapt to the human partner behavior to reach a common goal.

A class of robots, that perform a very rich communication with humans are musical performance robots. Anthropomorphic musical performance robots have the ability to mechanically emulate the human way of playing a musical instrument. In particular, a Musical-based Interaction System (MbIS) that allows human musicians to interact with the Flutist Robot WF-4RV is proposed. The system focuses on musical interaction with the purpose of creating novel ways of musical expression. The system consists of several modules with different tasks, for each task, there is one specific module that analyses the output from the camera and microphone of the robot and maps the extracted information to parameters that modify the musical performance. From the point-of-view of the user the interaction system is separated into two stages, that accommodate for users of different skill levels.

The basic level interaction stage uses audio and visual analysis, to allow the user control aspects of the robot performance. The translation of the input provided by the user and the modulation of the robot parameters is conditioned by the real-time state of the robot. In the extended level interaction interface, my goal is to give the user

the possibility to interact with the robot more freely (compared to the basic level). To achieve this, I propose a simple learning system that allows the user to link instrument gestures with musical patterns. Here, the correlation of sensor input to sensor output is not fixed.

To verify the proposed contribution of the described interaction system to the research field of Human-Robot Interaction (HRI), several experiments were performed. With these experiments the conceptual functionality of the interaction system as an interface for musical interaction with a performance robot is demonstrated. The results show that the proposed system introduces a novel action-perception scheme with learning capabilities that enables wind instrument playing anthropomorphic robots to dynamically adapt their behavior, depending on the human partner's intentions for achieving a common goal: a natural and intuitive musical interaction.

Contents

List of Figures

Chapter 1

Introduction

1.1 Human-Robot Interaction Research

1.1.1 Robots in a Human Environment

Since the 1960's industrial production processes have become more and more automatized. The development of industrial robots has helped to improve production quality and production speed in many industry areas. Rapid development has taken place in the car industry, where from the late 1970's production processes like welding and painting of car parts has been strongly accelerated by the use of robotic arms equipped with the respective production tool. In other areas like the IT industry, robots rapidly assemble complex electronic devices with high accuracy. Through the use of vision analysis systems and other sensory equipment, production quality can be asserted to a high level.

More recently, there have been several attempts to introduce robots also in areas where they come into direct contact with people. Apart from scientific research, one of the starting points of this development was the toy industry in the 1980's. Further development has established robots in households to perform everyday tasks like dust cleaning. In places more inviting to innovation such as trade fairs or art exhibition spaces, robots have been used as visitor guides and information terminals. Furthermore, development efforts worldwide are intending to push robot technology to perform medical treatment tasks and patient care. With the prospect of elderly societies in several developed countries, one

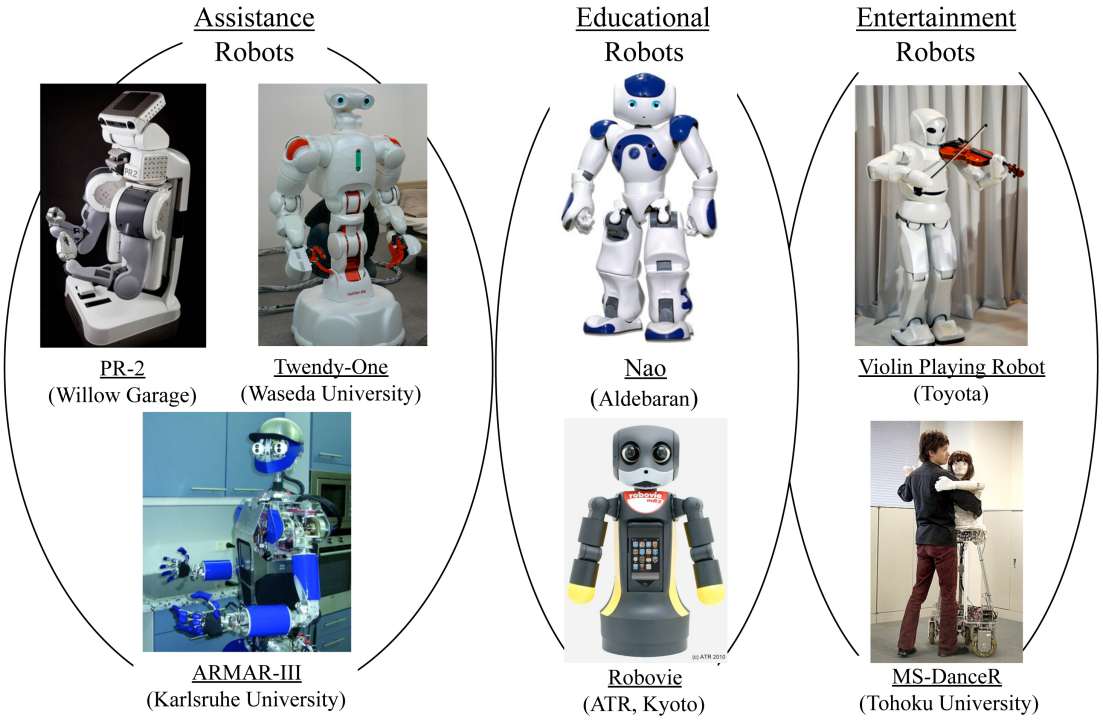

Figure 1.1: Overview of different classes of anthropomorphic robots, that are capable of human-robot interaction. Musical performance robots belong to the class of entertainment robots.

important future field of application of robot technology is in health and welfare care.

As soon as a robot is expected to operate in an environment that is shared with people, the robot needs to be able to interact with humans. Sensory units allow the robot to find its area of operation and define its task space. It needs to be able to detect the proximity of people and give humans the possibility to operate it through an appropriate interface. The people being in the same environment with the robot, regardless if they passively or actively use the device, need to have the possibility to control the machine's actions. In case of emergency, a mechanism to immediately secure the operation of the robot needs to be provided.

The amount of operability of a robot depends on the character of the interface that is used for interaction. The more natural interaction the interface allows, the

easier it is for unskilled users to interact with the robot. The optimum solution is completely human-like interaction with the machine. Recently there have been many research efforts to develop interfaces which allow natural communication between human and robot. The development of such an interface has shown to be a challenging task. Robots that integrate into human environments in many cases are very complex devices. To use these devices without initialization or regular adjustment by an engineer, requires a user interface that makes a large amount of parameters accessible in an abstracted, user-friendly way. This requires careful considerations not only from a technical point-of-view, but also from a social-scientific perspective. In Fig. 1.1, an overview of different types of anthropomorphic robots capable of human-robot interaction are displayed. Displayed in the group of assistance robots, the robot PR2 has been developed by the company Willow Garage as an universal purpose robot, that is able to perform autonomous navigation in complex environments and can execute object manipulation tasks in cooperation with human users ([2]). The research connected to the development of this robot focuses on simultaneous localization and mapping (SLAM)-based navigation and human-robot interaction. The anthropomorphic robot Twendy-One, developed at Waseda University, is also able to achieve various tasks in common surroundings such as a living room or kitchen. It is equipped with force compliant arms, that allow it to safely interact with humans. One of the research goals to enable to robot to perform tasks with human-like dexterity ([3]). ARMAR-III has been developed at the Technical University of Karlsruhe. Specialized on operation in a kitchen environment, it uses machine vision techniques to manipulate common kitchen objects like cereal boxes, plates and glasses. Using its force compliant arms, it is able to insert items into a dishwasher. The robot is used for various research purposes, including the development of robust image processing algorithms and motion planning methods ([4]). The company Aldebaran has developed the small-scale anthropomorphic robot Nao as a tool for research and educational purposes. Although the company itself does not perform actual scientific research with the robot, several universities use Nao to do experiments in the fields of human-robot interaction, robot vision, motion planning etc. ([5]). Based on a similar concept, the robot Robovie has been developed at

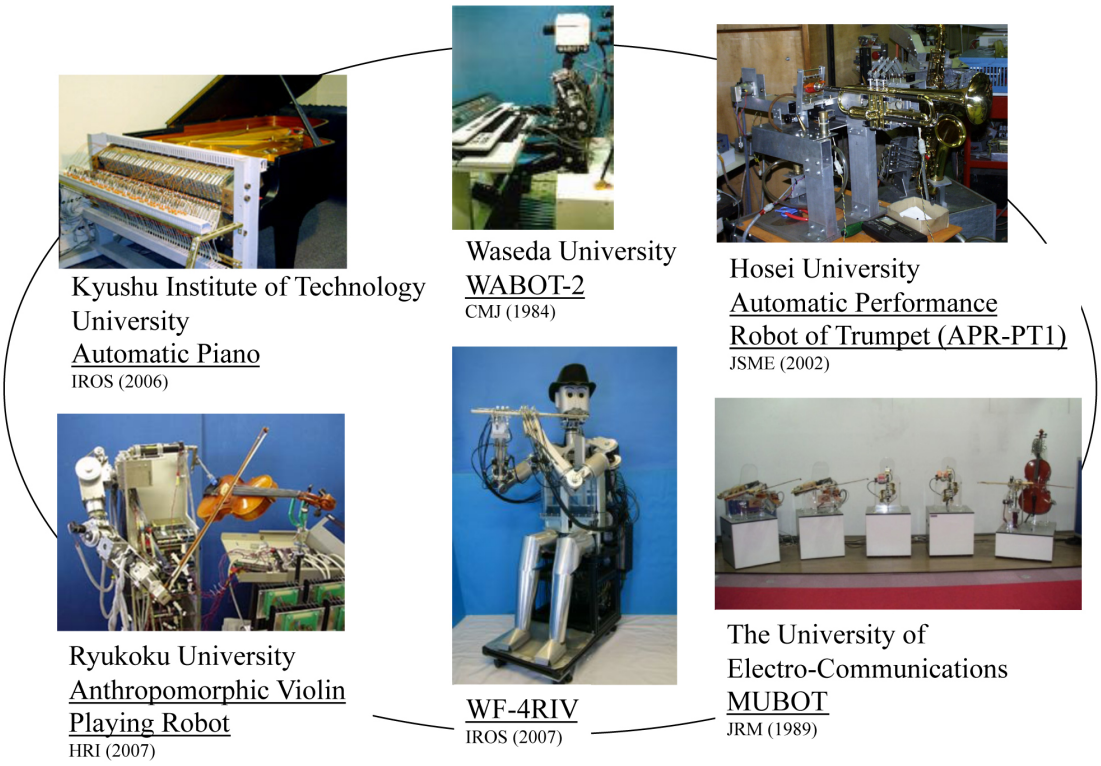

Figure 1.2: Anthropomorphic musical performance robots can be grouped into two subtypes: Passive musical performance robots and active musical performance robots

Advanced Telecommunications Research Institute International (ATR) as educational platform for machine learning and human-robot interaction research ([6]). Two entertainment robots shown displayed in Fig. 1.1 are the violin playing robot, that was developed by Toyota Motors ([7]) and the dancing robot created at Tohoku university ([8]). Whereas the research connected with the violin playing robot focuses on the reproduction of human motor skills, the dancing robot is to interact naturally with humans during a dance performance.

1.1.2 Musical Performance Robots

Robots, that perform a very rich communication with humans are musical performance robots. Anthropomorphic musical performance robots the ability to

mechanically emulate the human way of playing a musical instrument. These technically very complex robots reach high performance levels, that are comparable to the skill level of professional human musicians. A feature that in most cases they still lack however, is the ability to interact with other musicians. Playing a fixed, invariable sequence, they might be able to perform together with other players, but as soon as there is a spontaneous variation in the musical performance, the human and the robot performance will be de-synchronized.

Similar to a rough classification, that we attempted for robots capable of human-robot interaction we also divide musical performance robots into two groups. Passive musical performance robots do not interact with their musical partners, but play a static, predefined performance. The purpose of these robots is to play an instrument at a high level of dexterity. A passive musical performance robot may be anthropomorphic, but generally the focus of its design is functionality. In spite of this design premise, passive musical performance robots may be extended with perceptional equipment (sensors) to enable communication with its musical partner.

The purpose of active musical performance robots is inter-play with human musicians or other robot musicians to create a lively, natural performance. In order to convey a more human-like feel to their performance, active musical performance robots may have an anthropomorphic body. This is supported by the assumption, that the more the human body is emulated, the more the robot will be able to integrate into a human environment. This principle however, applies not only to the additional capabilities provided by an anthropomorphic robot, but also includes the physical constraints that are inherited with the human-like design structure. The necessity to adjust inter-play with the robot to these human-like physical constraints, automatically gives a more natural feel to the interaction.

Examples of anthropomorphic musical performance robots and their classification are shown in Fig. 1.2.

The saxophone robot APR-SX2, developed at Hosei University, is an example of a passive musical performance robot. APR-SX2 is able to play the saxophone at the level of an intermediate to advanced human player. A complex mechanical lever system has been constructed to enable the robot to strike all keys of a tenor

saxophone. The application focus of the robot lies on performance dexterity, it has only limited anthropomorphic characteristics.

At Ryokoku University the anthropomorphic violin playing robot has been developed. This robot has a total of 7 degrees-of-freedom, with the goal of studying the artificial reproduction of the bowing motion of a violin player. This musical performance robot has not yet been developed as far, as that it can play complete tunes. In the present research stage it is able to play single notes without any interaction capabilities. An important part of its design concept is the integration of the japanese Kansei into its performance. This means that the robot is to be equipped with a human-like sensitivity to add expressive nuances to its play, according to the musical context.

At Georgia University of Technology the percussion performance robot Haile has been developed. Haile can be classified as simplified active performance robot, as it can acoustically interact with its fellow musicians to a certain degree. However, the research focus in the realization of this performance interface is different from the approach I present in this thesis. In this thesis, I introduce an interaction system that is to provide a human-like interface for an anthropomorphic musical performance robot. Haile's design is not anthropomorphic. Its mechanical design is relatively simple, and cannot be enabled to play different instruments apart from the drum.

The two further introduced active musical performance robots are the WABOT-2 piano playing robot and the Waseda Flutist No. 4 Refined IV (WF-4RIV). WABOT-2 has been developed in 1984 at Waseda University. It is the first in a line of anthropomorphic musical performance robots constructed at Waseda University. WABOT-2 is a able of limited interaction with other human performers, by following the voice of a singer. It acquires information about the musical score to be performed by capturing notes with a video camera and processing the image data.

The development of musical performance robots at Waseda University has been continued with the construction of the Waseda flutist robot. The WF-4RIV is the predecessor of the interaction system, that I propose in this thesis. This robot is capable of limited interaction with human musicians: it is able to track the gaze of a musician partner. Furthermore it can offline-process musical audio

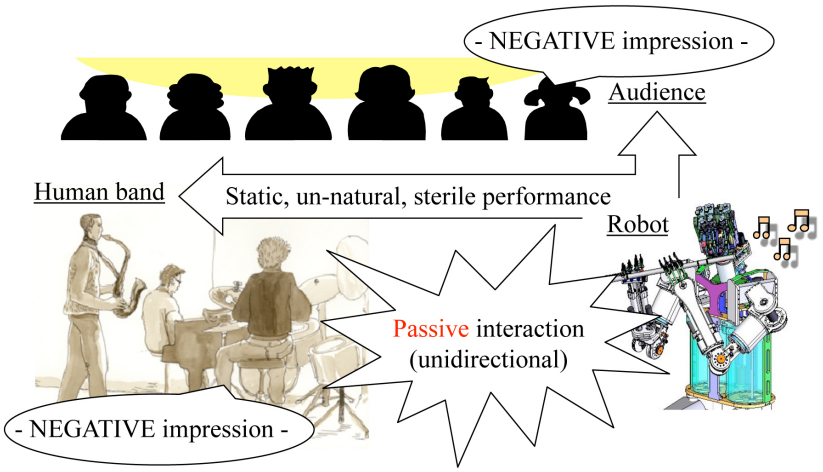

Figure 1.3: If the robot plays together with a human band without interacting with the other musicians, the result is a static, unnatural and sterile performance. This gives a negative, unsatisfactory impression on the human musicians and the audience.

data and extract certain performance parameters, in order to apply these to its own performance.

1.2 Research Background

The research on musical robots has a long tradition at Waseda University. Nearly thirty years ago the late professor Ichiro Kato has developed the humanoid pianist robot WABOT-2 ([9]), that was able to play a piano song from a score of notes. After the successful development of the piano robot, the research on an anthropomorphic flutist robot has been started ([10]). In order to better understand the human flute performance, several mechanical and cognitive improvements have been done towards understanding the dexterity of humans to play wind instruments and proposing novel ways of musical expression. Thanks to the bio-inspired design of the mechanically simulated organs of the robot, the most recent version, the Waseda Flutist No. 4 Refined IV (WF-4RIV) is considered to play the flute similar to the performance of an intermediate human player. This has been verified by a survey as well as through the use of an evaluation function to com-

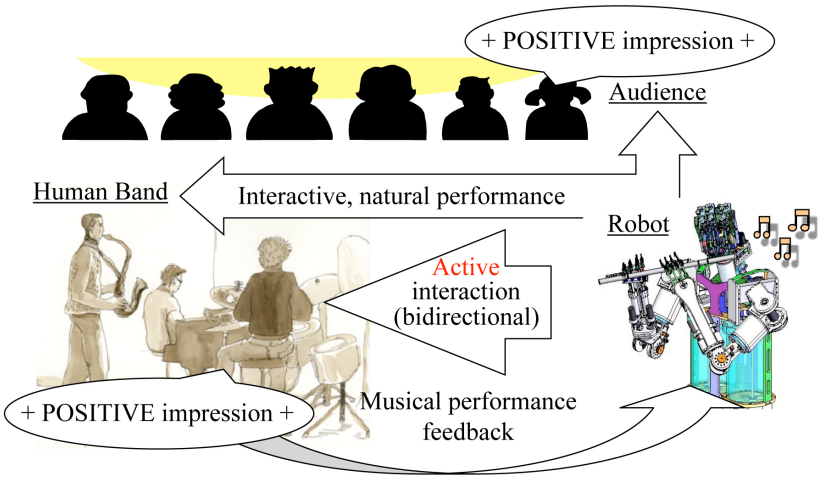

Figure 1.4: If the robot is able to react to musical and visual feedback from the human musicians, the result is an inter-active performance. Such a performance gives the positive impression of a natural performance to the other musicians and the audience.

pare the performance quality of a professional flute player and the flutist robot ([11]). In addition, as an approach towards enhancing its cognitive capabilities, an automatic melody recognition system and face tracking vision system were implemented, to enable the robot to interact with flutist beginners ([12]). Through experimental results it was shown, that the flutist robot is able to interact with flutist beginners under controlled experimental conditions (acting as a musical tutor). Considering these results, it was proposed, that in order to extend the interaction capabilities of the flutist robot with musical partners, further research efforts should be done ([10]).

In this further research efforts I try to integrate the flutist robot with a human band. The goal is to try to give the robot the interactive capabilities to actively play together with human musicians or other musical performance robots. As shown in Fig. 1.3, if the robot completely lacks interactivity with its musical environment, the result is a static, unnatural and sterile performance. Such a performance especially during a longer presentation will not make a stimulating, entertaining impression on the audience. A passive performance is predictable and does not contain improvised or situation-based alterations. If the robot is

able to perceive information from its environment, it can adjust to the musical concept of its partnering musicians (Figure 1.4). The higher the degree to which this is achieved, the more lively, natural and human-like the whole performance is going to become. A natural performance will give a positive impression on the human band, as well as the audience.

1.3 Research Objective and Motivation

Through the approach presented in this thesis, I try to enable an anthropomorphic robot to act in the same environment as its human partner with the intention to perform a certain task. In contrast to interaction through predefined actions, the robot is to dynamically adapt to the behavior of the human partner to reach a common goal. Through perceiving information via various communication channels, the robot should be aware of its situation within a certain environment and act accordingly (perception-action scheme). By the development of a multi-modal interface with teach-in capabilities, I would like to use anticipated communication patterns related to the target task to provide natural interaction to the human partner of the robot.

In this thesis I would like to attempt to prove the hypothesis, that the proposed perception-action scheme-based, human-robot interaction system is suitable to enable natural, task-oriented collaboration with an anthropomorphic robot (Figure 1.5).

The implementation of a perception-action scheme means, that the robot is able to perceive actions within its environment through various communication channels, process this information and provide feedback to the environment. If this methodology is implemented well enough, communication with the robot will be perceived by the human interaction partner as natural communication. For this reason, it is desirable to realize the concept of perception and action, in order to mimic human behavior. In this thesis, I applied this principle to a musical context, but I also attempt to develop an interaction system, that is extendable to other application areas (such as kansei engineering, communication science, music composition). Besides working on basic perception modules for visual

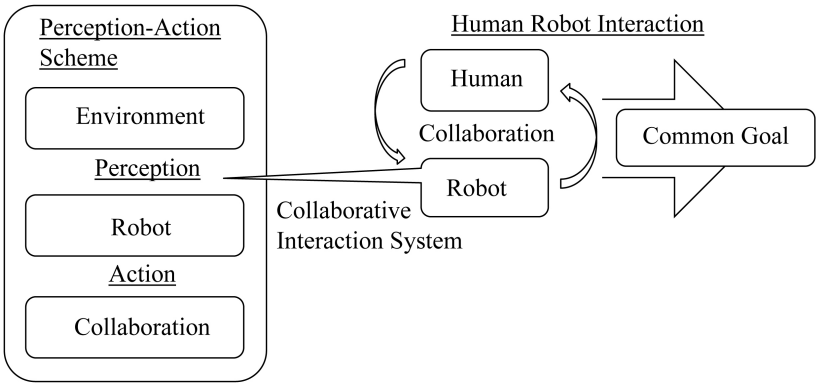

Figure 1.5: In order to integrate a robot into a human environment, the robot needs to be able interact with its human user. As a concept to enable such an interaction, I propose a perception-action scheme as shown in this figure.

and acoustic analysis, I put special consideration to the translation of sensory information into action.

Regarding this translation, I especially focus on two points: First, although I mimic parts of the human body with an anthropomorphic robot, when controlling the robot, I still need to take into account the specific mechanical structure of the robot. This needs to be reflected in the conception of the interaction system and the way perceived information is translated into action. Second, I consider that humans, when communicating naturally, do not always translate information instantly into action, but they also use their memory, to delay or alter action. By implementing a teach-in concept into one part of the interaction system, that I propose in this thesis, I attempted to account for this behavior.

In particular, I propose a Musical-based Interaction System (MbIS) that allows human musicians to interact with the Waseda Flutist Robot No. 4 Refined V (WF-4RV). The system focuses on musical interaction with the purpose of creating novel ways of musical expression. The anthropomorphic robot is not merely used as an instrument-playing machine, but it is enabled to act as a musical partner. The WF-4RV achieves this by processing acoustic as well as visual sensor input and by mapping those incoming inputs into musical outputs. The system has three specific novelties that separate it from other musical interac-

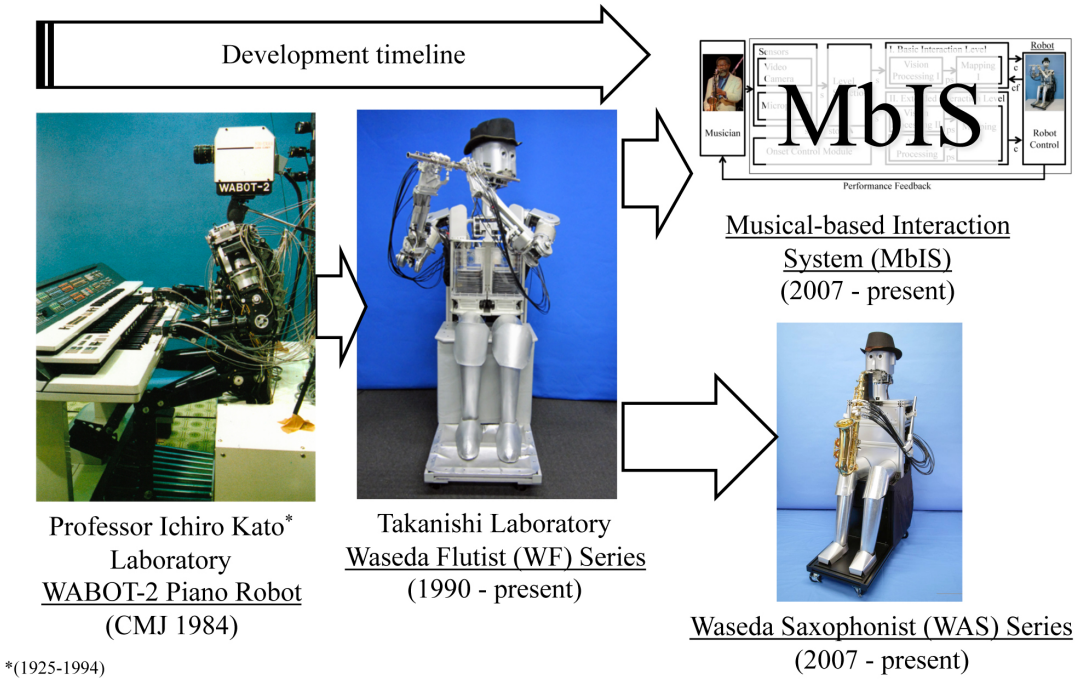

Figure 1.6: This figure shows the development timeline of musical performance systems at Waseda university. The research on the WABOT-2 piano robot, started by late Prof. Ichiro Kato, has been continued in Takanishi laboratory with the development of the flutist robot (WF series) and saxophone robot (WAS series). The general concept of the Musical-based Interaction System is complimentary to the developed robotic systems, in order to enable them to interact with human musicians.

tion systems: It does not only use acoustic sensor data to render new musical content, but also has a vision recognition system to react to gestures by other musicians. The music is performed by a anthropomorphic robot that is designed to emulate the human way of playing the flute as closely as possible. This incorporates that the robot has, like a human player, certain physical limitations that need to be taken into account within the interaction system. As the interaction with a complex robot like the WF-4RV is considered to be a challenge for a musician, the interaction system is adjustable to a beginner and an advanced skill level. Through these novel developments, I try to create a natural interaction system for an anthropomorphic robot entity that conveys a human-like feeling. Fig. 1.6 shows a timeline, where the proposed MbIS fits into the family of musical performance robot systems, that have been developed at Waseda University.

The system consists of several modules with different tasks, for each task, there is one specific module that analyses the output from the camera and microphone of the robot and maps the extracted information to parameters that modify the musical performance. From these parameters MIDI data is generated and sent to the robot. The robot itself has a motor control module that receives the MIDI information and adjusts the movement of the motors accordingly. The separation of the user interface into two stages takes place in the sound and video analysis part of the system.

The basic level of interaction uses visual analysis: Virtual Faders and buttons sense the instrument movements of the robots partner musician. The data sent from the input sensor processing to the robot control module is modulated by the physical state of the robot. I use histogram-based audio analysis to verify, if the performance output of the robot is according to the intended result. In the extended level interaction interface, my goal is to give the user the possibility to interact with the robot more freely (compared to the beginner level). To achieve this, I propose a simplified learning (teach-in) system that allows the user to link instrument gestures with musical patterns. Here, the correlation of sensor input to sensor output (mapping strategy) is not fixed. Furthermore, I allow for more degrees-of-freedom in the instrument movements of the user. As a result this level is more suitable for advanced level players. For this task I use a particle filter-based instrument gesture detection system. A Bayesian mapping algorithm

is employed in order to ensure, that if the teaching musician does not account for all combinations of instrument orientation and musical output in the teaching phase, in the performance phase the robot will automatically play the most closely matching answer modulation to a given instrument state.

To verify the proposed contribution of the described interaction system to the research field of HRI, several experiments were performed. With these experiments the conceptual functionality of the interaction system as an interface for musical interaction with a performance robot is to be demonstrated. To evaluate the system, musicians of different skill levels were asked to interact with the robot, following their personal intention for musical expression. I analyzed the result of these experiments practically by recording incoming sensor data and musical output. This recorded data was examined in order to investigate the accuracy and robustness of the proposed method.

I evaluated the system in terms of the practicability of the HRI concept by surveying the personal performance experience of the human partner musicians. The proposed experiments were based on a comparative, technical and non-technical user-oriented evaluation to verify the contribution of the proposed interaction system to the HRI research field. I regard the system as being functional in respect to its purpose from a technical as well as from a user-experience point of view. Therefore, I consider the system as a whole, due to its novel components, as a conceptual as well as a practical contribution to the HRI field.

1.4 Related Research

Automatic music performance machines have been developed since the early 19th century. These first musical robots were mostly mechanical piano players like Fourneauxs Pianista or Voteys Pianola ([13]). Though the principle of tone generation when automatically playing an instrument has not changed much since then, in former times, the ways to make a performance sound more natural were very limited. Since in the 1960s and 1970s, the possibility of electronic triggering and transduction of music, as well as music synthesis, has technologically advanced rapidly. In the 1980s first musical robots were constructed and have since then become a more and more popular research topic. Anthropomorphic, robotic

	Haile	WF-4RIV
Creator	Georgia Tech. 2007	Waseda Univ. 2007
Instrument	Drum	Flutist
Anthropomorphic	no	yes
No. DOFs	2-DOFs	41-DOFs
Multi-instrument	no	yes

Figure 1.7: A comparison of the technical characteristics of Haile, developed at Georgia Institute of Technology and WF-4RIV developed at Waseda University.

string instruments like the Guitar-Bot ([14]) were created. GuitarBot is only one of several robots developed within the cooperation project LEMUR ([15]). Recently LEMUR has worked together with the well-known musician Pat Metheny to create a robotic orchestra, called the orchestrion. This large-scale robotic orchestra contains among others the TibetBot (Tibetan Singing Bowls robot), !rBot (Novel percussive instrument robot), ForestBot (Tree-like percussive instrument robot), ModBot (Simple, modular one-actuator instrument robot). At the University of Electro-Communications the MuBot string instrument robot series has been developed ([16]), that has more recently resulted into an anthropomorphic violin playing robot ([17]). From Hosei University results about an automatic trumpet performance robot ([18]) and an automatic saxophonist robot ([19]) have been published. A more exotic musical performance robot are Trimpins robotic turntables ([13]), that use control over the rotational speed of a record and the position of the tone-arm to rearrange the musical content of a record. If more than one of these robotic turntables are used synchronously, complex re-arrangements of recorded material is possible.

The approach of processing musical audio input by matching it to predetermined musical content has been developed by several researchers ([20]). Various techniques to perform artificial composition and improvisation have been proposed. The work that might be closest to the approach presented in this thesis is the work by Weinberg ([21]). The percussion robot Haile is able to tune in to the rhythm of a partner musician and within a certain limit varies its performance to display improvisation capabilities. One of the main differences between WF-4RV and the Haile are the level of complexity of the sound generation mechanisms of the robots (Figure 1.7). Considering required mechanical dexterity, flute playing on the one hand requires the accurate synchronization of several organs to produce sound. Drum playing on the other hand, is in comparison a relatively simple process and requires only the control of the hand motion. To perform rhythm analysis Haile utilizes external Max/MSP objects that are mainly based on, similar to my method, measuring volume peaks in a certain frequency range. In my approach I developed software components in C++, that I performance optimized for my purpose. In particular, special image processing methods (motion perception-based tracking, particle filter-based tracking) and audio processing algorithms (histogram-based rhythm and melody tracking) were implemented to work computationally efficiently.

Weinberg has concentrated on the interaction between human musicians and his music robot. His robot can actively adjust to the play of partner musicians, imitating their behavior of creating a rhythm. His work uses an approach of analyzing the recorded music data and extract information about the current musical situation by applying a rhythmic rule-set. Although I also base my aural interaction on imitating a human musician, I do so by comparing musical input with prerecorded sequences in the library using a histogram method. A further substantial difference is that my system also involves visual processing. I combine audio-visual perception in a two levels of interaction system. This enables human-robot communication that is more natural and complete than previously introduced systems. From the musical point of view our system allows adjustment of the flutist robots performance according to the musicians musical intentions.

1.5 Contents Overview

I laid out the remainder of the thesis as follows (1.8):

1. **Chapter 2 - Musical-based Interaction System (MbIS) Overview** In this chapter I introduce the interaction system presented in this thesis. I describe the hardware platform on which the system was implemented, the flutist robot WF-4RV. I consider the application of the Waseda Flutist Robot in different musical contexts. To integrate the WF-4RV into an active performance environment with human players, I propose a skill-dependent Musical-based Interaction System (MbIS). Experimental results, that explain the capability of the robot to play an interactive performance are shown and evaluated.

2. **Chapter 3 - Image Processing Module Implementation** The MbIS is divided into several modules. To allow the robot to do visual communication with its human partners I implemented an image processing module. The basic level interaction module uses motion tracking to create virtual faders and virtual buttons that can be used by a musician similar to traditional studio controllers. In the extended level interaction system, I apply particle filter-based histogram tracking to follow instrument movements with more degrees of freedom. In an experimental section I evaluate the functionality of the image processing modules.

3. **Chapter 4 - Audio Processing Module Implementation** In this chapter, I introduce the audio processing module of the MbIS. For both skill levels, after a basic audio signal analysis, histogram-based pattern library matching is applied. In the basic level interaction mode, recorded audio material is searched for known rhythmical content. In the extended level interaction mode, rhythmical patterns as well as melodic content are evaluated. I verify the effectiveness of both methods experimentally.

4. **Chapter 5 - Mapping Strategy Implementation** In this section I describe the mapping strategy implementation of the Musical-based Interaction System (MbIS). The mapping module is responsible for converting

the information from the sensor processing system into parameter modulation of the musical performance of the flutist robot. I chose different mapping strategies for the beginner level interaction system and the advanced level interaction system, in order to accommodate for different user skill levels. I present the results of experiments to evaluate the functionality of the mapping strategy module for various user inputs.

5. **Chapter 6 - Evaluation of the Proposed Interaction System from a HRI Research-oriented Perspective** This chapter describes the previously introduced system from an universal HRI research-orientated point-of-view. To clarify the advantages of the implemented interaction system, I compare passive and interactive operation of the robot. I perform an analysis of the restricted parameter adjustment within the physical limitations of a robot and an analysis of the variable sensor mapping of the advanced level interaction system, through experiments under realistic circumstances. I evaluate the basic level interaction system as well as the extended level interaction system from a non-technical user perspective.

6. **Chapter 7 - Conclusion** In this chapter I show a concluding evaluation of the results and considerations in this thesis. I suggest future works as a perspective to continue the propose research path.

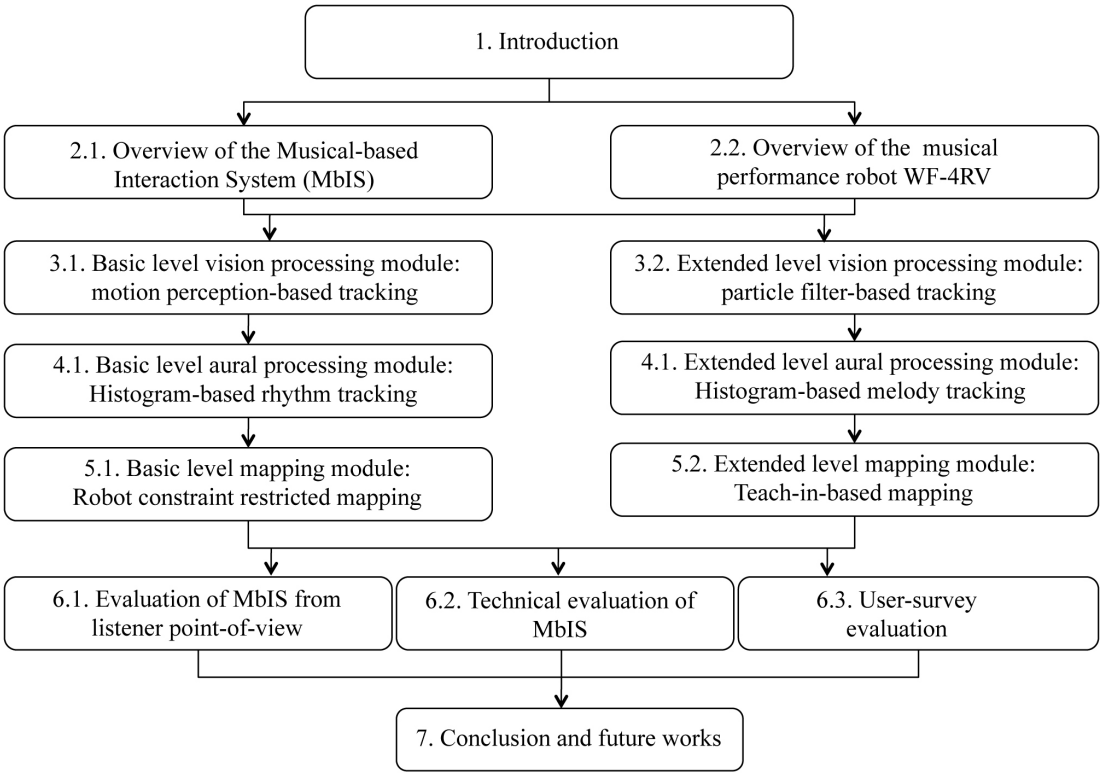

Figure 1.8: An overview of the contents of the thesis is shown in this chart.

Chapter 2

Musical-based Interaction System (MbIS) Overview

2.1 Introduction

The WF-4RV is equipped with sensors that allow it to acquire information about its environment. Due to the human-like design of the robot, it emulates two of the human's most important perceptual organs: the eyes and the ears. Therefore, two miniature video cameras and a stereo microphone were integrated into the head mechanism of the robot. A major part of the typical performance of a jazz composition is based on improvisation. Musicians take turns in playing solos based on the harmonies and rhythmical structure of a song. Upon finishing his solo section, one musician will give a acoustic or visual signal to designate the next soloist. A Musical-Based Interaction System (MbIS) is proposed, to allow the interaction between the flutist robot and musicians based on two levels of interaction. The flutist robot WF-4RV has been developed for several years, by several generations of students. As an inevitable result of this process, the overall structure of the robot electronics and software implementation has become too complex to be easily extendible and understandable. To improve this situation, in the course of this thesis, besides focusing on the interaction system, I completely rewrote the motor control software of the robot. I also remade the complete electronics and wiring system of the robot, to make it less space-consuming and more reliable.

2.2 Waseda Flutist Robot 4 Refined V

2.2.1 State of the Art of the Flutist Robot

The anthropomorphic flutist robot ([22], [12], [23]) has been developed to study the methodology of human motor control ([11]), to introduce novel ways of man-machine interaction and to propose interesting applications for humanoid robots (i.e. musical tutor, etc.). After initial stages of research the flutist robot has been capable of basic playing techniques ([22]), the sound production quality (fingering, breath control etc.) being similar to a flutist beginner's. After further developments, extended technical skills ([24]), which are typically practiced by intermediate level flutists (i.e. vibrato, etc.), have been successfully implemented. More recently, in [24], the Waseda Flutist Robot No.4 Refined III (WF-4RIII) has been presented. This version of the flutist robot has automatically generated musical expressiveness by modeling the performance of a professional flutist using a Neural Network (NN). In this way, individual musical sound characteristics such as slightly varying note duration and vibrato amplitude were reproduced. Mechanically, the WF-4RIII has been developed with 43-DOFs in total. Compared to previous versions the mechanical design of the lung system was improved and a human-like vocal cord designed. These modifications have allowed very accurate control of note duration and vibrato. As result, the expressiveness of the musical performance of the flutist robot, which also plays an important role during interaction with humans, has been significantly enhanced ([23]). During the development process there has been constant cooperation with professional flutist players to verify the performance level of the flutist robot ([10]).

Thanks to these recent improvements on the WF-4RIV, the flutist robot is capable of a performance similar to an intermediate player. From the interaction point of view, in [25], an auditory feedback system has been developed. For this purpose, the perceptual capabilities of the flutist robot have been enhanced. In particular, the flutist robot has been enabled to recognize melodies by using Hidden Markov Models [26], evaluate the performances of flutist beginners and detecting and track human faces [12]. However, the interaction between the flutist robot and the musical partner have been too limited (several restrictions were

considered regarding the conditions of the interaction). Therefore, I considered to perform improvements on the perceptual capabilities of the robot.

Building upon this achievement, further research has been focused on improving the capabilities of the robot to play together with its musical partner more interactively. As a main goal, I plan to enable musical robots to actively cooperate with a human band, thus creating novel ways of musical expression. For this purpose, I considered it necessary to enhance the perceptual abilities of the flutist robot to process musical information from the aural and visual perceptual channels.

2.2.2 The Mechanical Hardware of WF-4RV

The Waseda Flutist Robot WF-4RV is an anthropomorphic musical performance robot. It consists of several separate components, that in conjunction give the robot the ability, to emulate human flute playing. In total, the WF-4RV is composed of 41-DOFs (degrees of freedom).

The air pressure to play the flute is produced by a 2-DOFs lung mechanism. The lung system of the robot has been designed as imitation of a human lung. The breathing cycle can be adjusted to control the flow of the air beam from the mouth of the robot to the mouthpiece of the flute. The lung system is composed of two airtight acrylic cases. Each of these cases contains a bellow controlled by a crank mechanism, to pull and push air in and out. As the lung system is airtight, the bellows have no contact with the acrylic surface during the breathing cycle. This minimizes the inhalation time, decreases the mechanical noise and increases the quantity of air being exhaled from the lips.

A special mechanism was designed to mimic the human vocal cord. During flute play, the vocal cord modulates the air flow from the lung with variable amplitude and frequency. In musical terms, the sound effect that is produced through the regular (e.g. sinusoidal) manipulation of the air-flow is referred to as vibrato.

The tonguing mechanism and lips mechanism are key mechanical components of WF-4RV to generate a natural flute sound. They are employed to improve the clarity of the produced sound and to enhance the quality of the transitions

between different notes. The lips mechanism of the WF-4RV is composed of 3-DOFs to accurately control the shape of the air beam. The lips themselves are made from a thermoplastic rubber called Septon ([27]). Septon has the advantage, that it is as elastic as rubber while it can be modeled like plastic. Using Septon, we can easily create different shapes to produce more human-like lips. Additionally, other kinds of lip forms can be designed so that the flutist robot can play other wind instruments just by changing the lip part.

A mechanism emulating the human tongue is located inside the oral cavity. This construction has 1-DOF and is controlled using a DC motor. The motor axis is connected to the tongue via a directly attached link. As a result of this new design of the oral cavity and tonguing mechanism, the WF-4RV can emulate double tonguing. This effect is achieved by obstructing the air flow using the front and back tip of the tongue. In order to hold the flute in the correct position to play, the robot has humanoid arms and hands. Its fingers can open and close the keys of the flute in 8 Hz. The flute is attached to the right hand with a fixture. In total, the arm system has 7-DOFs, divided into upper arm mechanism with 4-DOFs and the lower arm mechanism with 3-DOFs.

The two video cameras of the robot are attached to a 3-DOFs eye system, that allows to pan and tilt the camera view. A face mask has been designed to be fixed to the head mechanism of the robot, enhancing the human-like appearance of WF-4RV.

2.2.3 Motor Control System

WF-4RV consists of 41-DOFs with 41 active joints that are controlled by one servo motor each. Each of these motors is connected to a central unit, the robot control PC. The robot control PC contains six Interface PCI-6205 counter boards to record the signal from the encoder units of the servo motors. The motion of each motor is controlled by one motor driver. The flutist robot contains four types of motor drivers: Maxon motor drivers, TITech motor drivers, Tokuden motor drivers and Yasukawa AC motor drivers.

Three Interface digital-to-analog (PCI-3300) converter boards send low current analog voltages to the motor drivers, which convert these low current voltages

into high current voltages to drive the servo motors. In case of the Maxon motor driver and the Yasukawa AC drivers, the motor driver is also connected with the encoder output of the corresponding motor. Each joint of the robot, except for the finger joints, is equipped with an optocoupler to allow for calibration of the zero position of the servo. The output from these optical sensors is connected to a digital I/O board (Interface PCI-2000) in the robot control PC.

In the course of the research for this thesis the wiring of the robot has been renewed. In case of the old wiring, the motor connections of the robot (motor drive voltage, encoder signal, calibration sensor) were connected to a set of boxes, located behind the computer. These boxes, containing the motor drivers to control the robot, were connected to the counter, digital I/O and D/A hardware of the control computer. Summarized the following disadvantages of this design have been observed: A spacious location is required for the operation of the robot. Transport is complicated due to the large number of external connections, boxes and other items. The very large amount of wiring is prone to damage, short-circuits and connection failures.

Therefore a new wiring system has been introduced. The motor drivers, except the AC drivers, were moved into a compartment below the robot (into the seat of the robot). Previous wiring was completely removed and the new wiring attached using safe and affordable components. The remaining peripherals, which were not inserted into the robot structure itself (robot control PC etc.), were placed inside a wheeled 19" rack mount, placed behind the flutist robot. The following advantages were achieved with the new connection concept: The size of the robot peripherals reduced to a third. Through the reduced overall wire length, the robot is less prone to connection problems. Better transportability has been achieved due to less peripherals and fixed installation within rack mount.

2.2.4 Vision and Audio Processing Hardware

In comparison to the passive performance hardware (Figure 2.1), the hardware system of WF-4RV has been significantly improved. The vision and audio processing systems are separated from the robot control PC (Figure 2.2). The audio

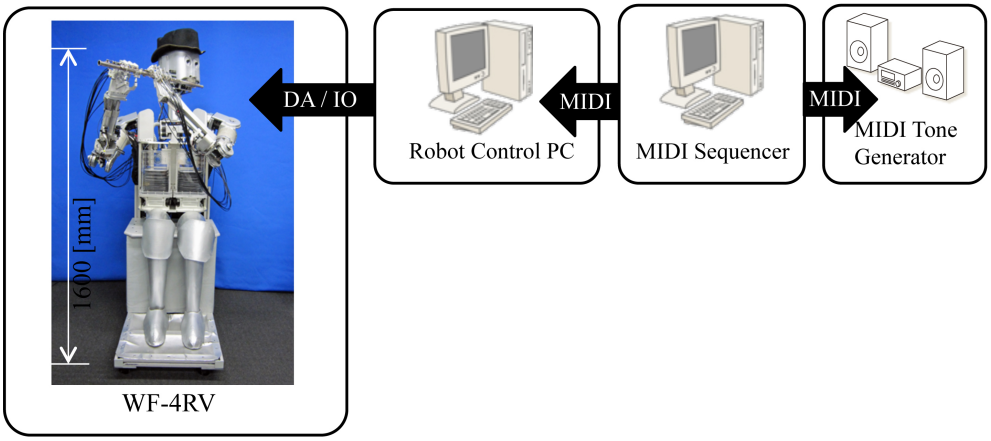

Figure 2.1: The figure shows the hardware control system used in the passive performance setup. A MIDI sequencer feeds MIDI data to the robot control PC, that controls the servos of WF-RV. A MIDI tone generator has been added to provide background accompaniment.

/ visual processing PC is connected to the robot control PC by a TCP / IP network.

Two Point Grey Firefly cameras integrated in the head mechanism of the robot, obtain video data at a resolution of 480 x 640 pixels and a sampling speed of 30 frames per second. The cameras are connected to the processing computer by two IEEE1394 connections.

Sound data is sampled at 44.1 kHz with a resolution of 16-bit. A Soundman OKM II stereo microphone, connected to an Echo Digital Audio Audiofire 4 interface through a pre-amplifier adapter Soundman A3, is used to record WF-4RVs partner musicians performance. The left and right part of the stereo mircophone are attached to the sides of the head of the robot for stereo-acoustic perception. The audio interface is also connected to the audio-processing PC by a IEEE1394 connection.

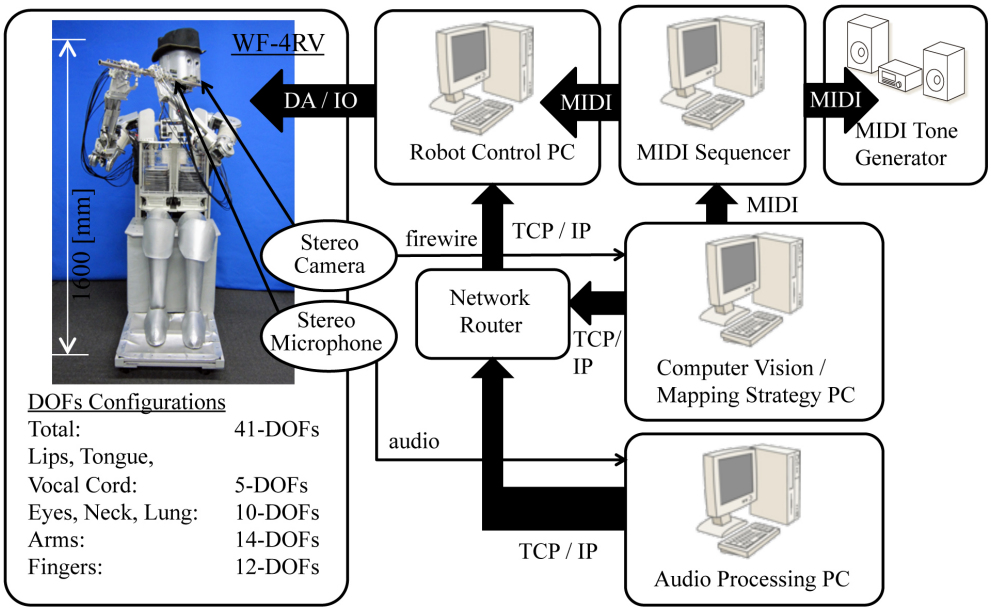

Figure 2.2: This figure shows an overview of the hardware of the interactive performance setup. A computer vision and mapping strategy subsystem as well as an audio processing unit have been added. All components are connected over a TCP/IP network.

2.2.5 Implementation of Servo Motor Control and Initialization

To provide more flexibility and future expandability, the software controlling the basic functions of the flutist robot has been re-written in the course of this research. The tasks of the control software of WF-4RV can be summarized in the following points: The robot control software has several tasks. Regarding the control of the motors of the robot, it is to perform manual and automatic initialization of the servo motors. The position of the servo motors is to be controlled, with the possibility to generate movement patterns and sequences from the position states of the motors. Transitions from one pattern to the next are achieved by continuous position control. Lung, vocal cord, tonguing, lips and fingers, are controlled through special purpose interfaces, that make accurate adjustment more easy. If needed inverse kinematics calculation can be performed within these modules. Pattern and sequence states, as well as all other state parameters of the robot can be saved, loaded and edited. Through MIDI and middleware communication, the robot control software can communicate with program sections that are located on different computer units, such as the mapping module of the MbIS.

The purpose of the WF-4RV's controller module is to control the active joints of the flutist robot, in order to produce the flute sound. From a user point-of-view, the state of each joint can be accessed separately. A joint can be initialized manually (by moving the joint to the initialization position by hand), or it can be initialized through an automatic procedure. In the automatic procedure, a constant command voltage moves the motor of a joint automatically to its initialization point. The initialization point is the position where the optocoupler connected to the joint changes its state. All joints expect the fingers are equipped with such optocouplers. In case of the fingers, the joints are moved to an initialization position (all flute keys closed state) by hand and the joints then manually initialized. Force initialization means that the current position of a motor encoder is defined as zero position joint. In case of failed initialization or hardware problems, de-initialization of a joint or re-initialization are also possible.

After a joint has been initialized, it is position controlled by the motor control thread. The thread calls the control routine of each initialized joint of the robot every 5ms. To provide as exact timing as possible, it polls timing information from the CPU performance counter.

In a previous version of the robot control program, it was attempted to run the position control of the motors in a separate thread each. This attempt was not successful due to the lack of thread prioritization mechanisms in Microsoft Windows. However, to synchronize all motor controllers in a single thread does not seem to lead to any performance deficiency. When all joints are initialized and controlled synchronized, a 5 ms timing cycle can be maintained.

The controller itself, that is used to adjust the position of each joint is a PID (Proportional Integral Differential) controller. The parameter settings of each joint have been adjusted and these settings saved in the program code. Careful adjustment of the parameters has led to satisfactory control results, so that a normal PID controller is considered to be sufficient for the accurate positioning of the flutist robot joints.

A set of joint positions is saved as a pattern. A pattern therefore contains the encoder values of a selected number of joints. When generating a pattern the user selects which joint encoder values will be part of the pattern. Motor encoder values of the current state of the robot, as well as motor encoder target values can be inserted into a pattern. Patterns are assigned a name and can be saved, loaded and deleted. Therefore manual teach-in can be performed in order to generate a pattern. This is important in case a new pattern is to be created in an easy manner. The procedure to achieve this is displayed in Fig. 2.3.

To move the robot from one joint configuration to another, the pattern generator of the program linearly interpolates between the current joint states of the robot and the target position saved in a pattern. The speed with which the robot moves to the joint positions that are saved in a pattern can be selected by the user. Patterns are saved as text files, that can be edited. Therefore manual manipulation of the patterns is possible. The format of these text files is displayed in Fig. 2.4.

Patterns can be grouped to be executed sequentially. For each pattern in such a sequence, the execution speed can be selected. When a sequence is finished and

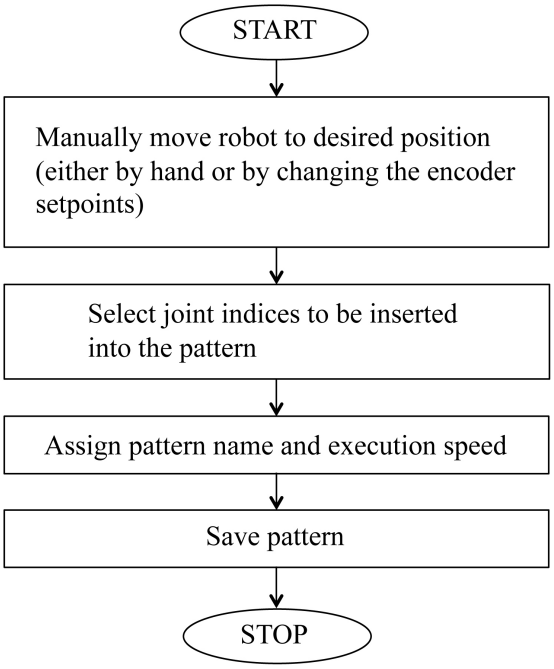

Figure 2.3: To create a new pattern the user has to follow a teach-in procedure to manually set the encoder positions, define pattern name and execution speed, and finally save a new pattern.

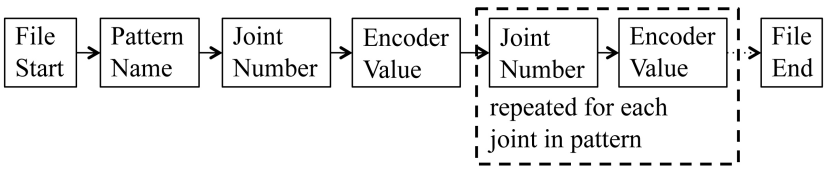

Figure 2.4: The figure displays the format of a pattern file. Besides the pattern name, the file contains the encoder data for each joint position that is to be saved in the pattern file.

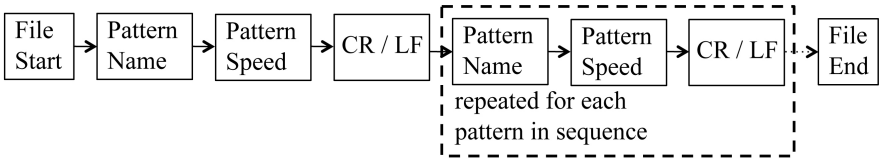

Figure 2.5: The purpose of a sequence file is to create a chain of patterns, that can be used to continuously control the robot. Additionally to the name of each contained pattern, also the transition speed from one pattern to the next is recorded.

its last pattern has been reached it automatically stops. The user has the option to loop a sequence, so that it runs continuously until it is manually terminated. The format of the sequence files is displayed in Fig. 2.5. Sequence speed and sequence selection can be modified by commands through the middleware connection of the program. Details of the middleware implementation are explained later on in this chapter.

2.2.6 Novel Lung and Breathing Control Implementation

In principle the control of the lung movement, the vocal cord movement, the tonguing movement and the finger movement can all be controlled through the general motor control dialog, the pattern generator and the sequence generator. To make the usage of these components easier for the user, several special purpose dialogs have been developed.

The lung controller dialog contains buttons to directly start and stop the movement of both lung motors. It also has input fields to set upwards and downwards movement speed. The vocal cord controller enables the user to set open and close position of the vocal cord, directly switch between these two positions and control vibrato amplitude and frequency. The tonguing controller contains input fields to set zero and on-position as well as the duration of the tonguing mechanism. A button allows the user to test the functionality of the tonguing movement. The finger movement controller assigns open and close encoder values for each finger. For each note that can be played by the flutist robot, finger position, lip position and lung speed can be set separately in the note control

dialog. Each of the special purpose controllers run in their own thread. Accurate thread timing can be monitored by the user in the thread master timing dialog.

We have experimented with two different methods to realize continuous breathing speed. Due to the construction of the lung mechanism, if we control the rotation of the motors of the lung at a constant speed, the plates which push the air up in the lung cylinders do not move accordingly, but accelerate and decelerate. The relationship of the rotation angle of the lung motor axis and the position of the lung plates can be calculated using forward-kinematics. To control the breathing speed to a constant value (compute the required rotation angle of the motor axis from the target position of the lung plates), an inverse-kinematics calculation has been employed.

Apart from the inverse-kinematics method, an additional way to compensate the previously described inconstant breathing speed has been proposed. The amount of air being pumped by the lung mechanism was measured for one breathing cycle. To measure this air-flow I inserted an air-flow meter between the lung and head mechanism of the robot. I recorded the air-flow data during a breathing cycle and saved it into a data file. Each time the software is started, this data file is read, and from the contained information a compensation function is calculated. The algorithm is applied as is displayed in Fig. 2.6. This purpose of this compensation function is to provide a steady air-stream to produce a flute sound of constant volume, that is independent from the fill level of the robot's lung.

Despite the described efforts, practical experiments with both setups, the inverse-kinematics method and the look-up table compensation, show that both methods do not provide sufficient results for the tone production equal to an intermediate human player. Therefore, I tried to compensate the inconsistency of the air-flow, by manipulating the lung speed manually. I did this by applying a piece-wise linear compensation function. Using this function, and by continuously improving its parameters with each experiment, I was able to achieve constant breathing air-flow.

Interaction with human musicians requires the flutist robot not only to play a carefully optimized performance. The robot also needs to be able to adjust its play to the actions of its musical partner, without diminishing its performance

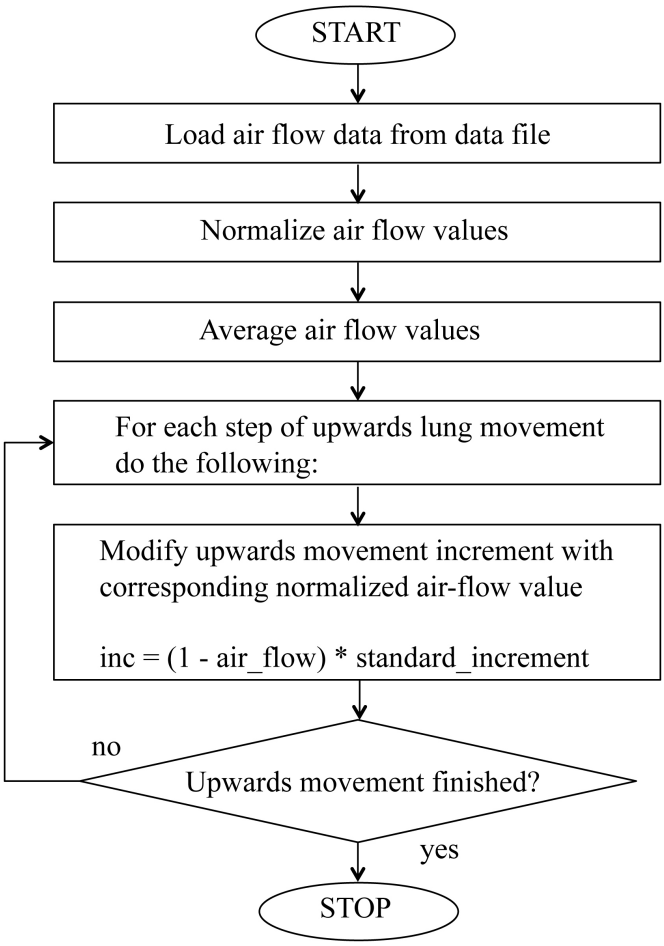

Figure 2.6: Constant air flow at the mouthpiece of the flute robot is essential for a good tone generation. The air-flow at the mouthpiece was measured in direct control mode of the lung (no inverse kinematics), and the measured data was used to compensate the lung speed to achieve a constant air-flow as shown in the algorithm diagram above.

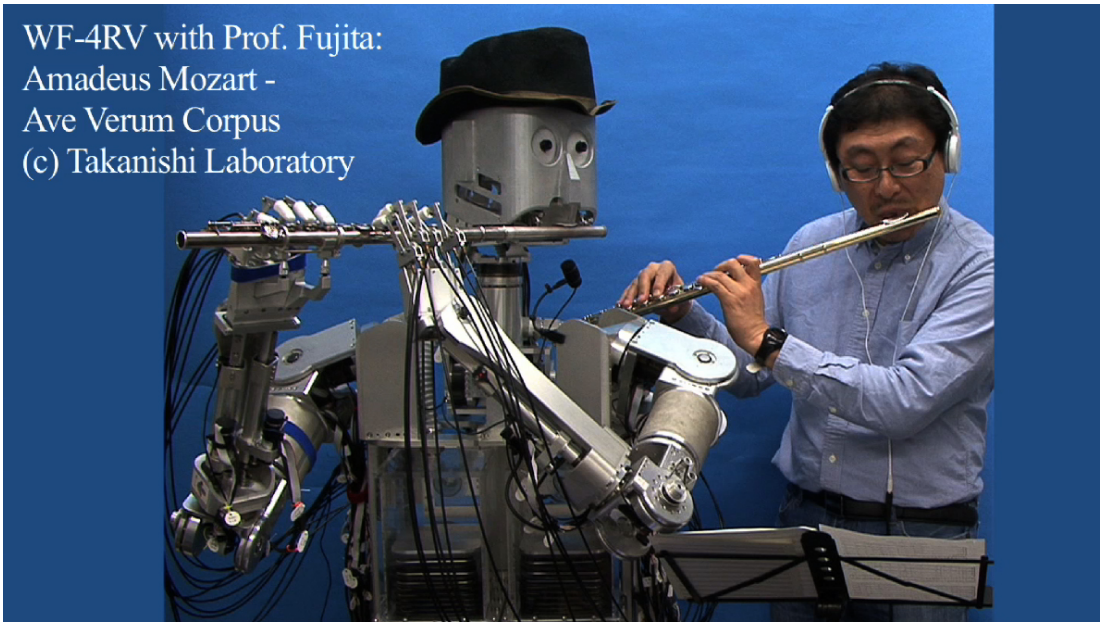

Figure 2.7: WF-4RV and professional flutist player Professor Fujita performed Mozart's Ave Verum Corpus in a duet to evaluate the functionality of the air-flow control.

quality. In first interaction experiments, I found, that especially for more complex melody sequences, the performance quality is not sufficient, using the tone generation method explained so far. To evaluate this capability I asked a professional flutist player to suggest an exemplary score. The flutist suggested the piece *Ave Verum Corpus* by Wolfgang Amadeus Mozart. The piece was performed as a duet performance with the professional flutist player and WF-4RV as shown in Fig. 2.7.

After performing several performance experiments with WF-4RV playing the suggested test piece, the professional flutist concluded, that the tone-to-tone transitions were not sufficiently clean, especially in cases of jumps from high notes to low notes (intervals of half an octave or more). As a solution he suggested to adjust the air-flow envelope for each tone separately.

An adjustment of the note envelope for each tone during a performance requires very fast and accurate control of the air-flow. To do this adjustment the possibility of direct control of the air-flow strength by modulation the lung speed

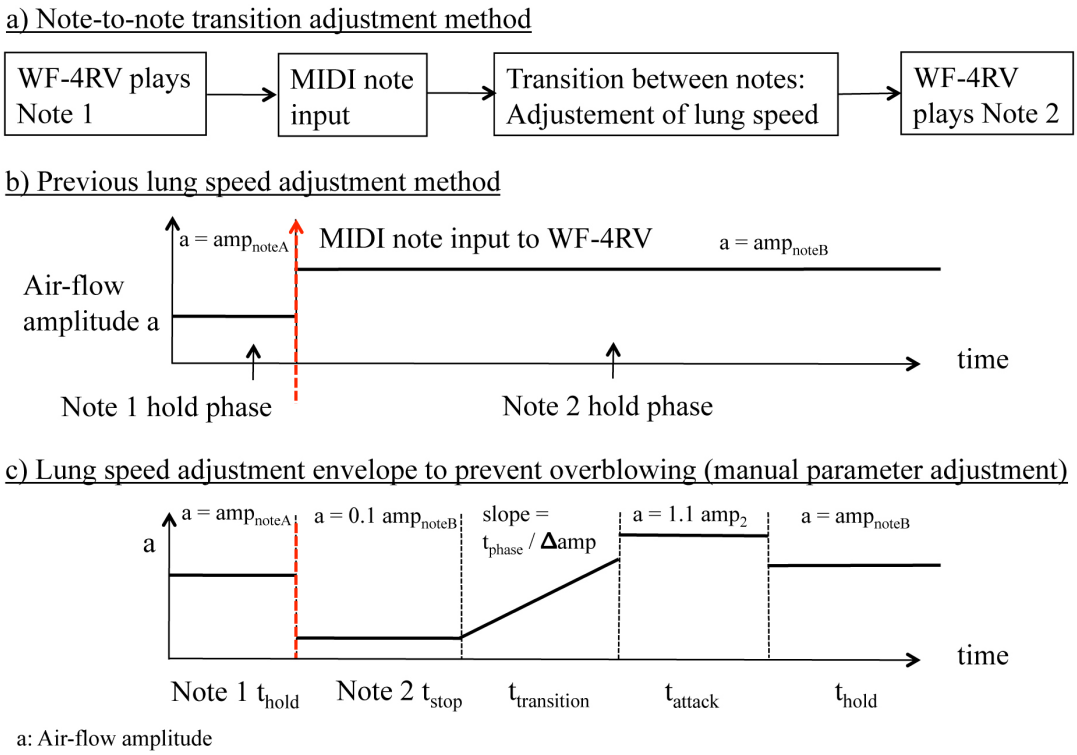

Figure 2.8: In a) the principle of the transition from one note to the next is shown based on the MIDI input to WF-4RV. b) shows the adjustment of lungs speed depending to the currently played note, but without air-flow control envelope. In c) the lung speed adjustment envelope to prevent overblowing is shown.

of the robot was considered. Note-to-note transitions during a performance are triggered by MIDI note data input to the motor control system of the robot as shown in Fig. 2.8a. So far, although different lung speeds were adjusted for different notes, speed levels were switched directly from one note to the next (Figure 2.8b). To provide a gradual adjustment of the air-flow, I implemented a transitions curve as shown in Fig. 2.8c. The shape of this envelope curve is similar to an Attack-Decay-Sustain-Release (ADSR, [28]) curve that is used to shape volume and filter envelopes in electronic music synthesizers.

In an experiment using the musical score, that was suggested by the professional flutist, I evaluated the performance of controlling the air-flow envelope

directly by adjusting the lung speed. I modified the WF-4RV motor control software in order to use different speed parameters for each note to be played. In the following experimental section of this chapter, I will present results of these experiments that show, that the note-to-note transition clearness has been significantly improved.

2.2.7 MIDI / Middleware Communication Management

The WF-4RV control software communicates with the image processing software, the audio processing software and the music sequencer through two different interfaces. Through these communication interfaces the software can receive control commands to trigger notes and change performance parameters, such as the vibrato frequency. The interface also can be used to send status information about the robot (such as the lung fill status of the robot) to other software components. The MIDI (Musical Instruments Digital Interface, [29]) interface is a serial communication interface that allows the transmission of note and expression parameter data. Therefore, the interface is suitable for conveying musical information.

To send more complex modulation parameters, ICE (Internet Communications Engine, [30]) is used as a middleware system. This library allows to send and receive data between different computers and operating systems with a high bandwidth. The Robot Operating System (ROS [31]) is used for the internal communication between the sensor processing components. For communication with the Windows motor control software, I use the Internet Communications Engine (ICE [30]). Because so far no native Windows port of ROS has been released, two different middleware systems had to be employed.

2.3 Musical Context Considerations for WF-4RV

2.3.1 Passive Performance Setup

The flutist robot WF-4RV is able to play the flute at the level of an intermediate human player. In past performances, the robot has played with a human musician, doing a passive performance. That means that the robot was not able to actively react to the play of its human partner, but rather performing a static MIDI score from a PC sequencer, with the human player adjusting his output to the performance of the robot. The resulting flute duet is similar to the interaction of two human players considering tone quality, but as the communication between the players is only unidirectional, the overall performance is not naturally dynamic.

2.3.2 Active Performance within a Human Band

During a musical performance musicians express themselves by visual communication and acoustic communication. Fig. 2.9 shows jazz saxophonist Charles Lloyd moving his instrument up and down, according to the rhythm of the music he is playing. Saxophonist Joe Henderson uses a similar instrument gesture to give a visual cue to another musician in his band, when finishing his solo performance. The use of acoustic synchronization allows band members to perform parts of a composition together, without at the same time reading a score of notes.

Two principal ways of human musicians to interact with each other during a performance are communication through the acoustic and visual channel. Although aural exchange of information seems predominant in a musical band setup there is also a large amount of communication exchanged through visual interaction. In my research, I have examined methods to visually track the movements of the instrument of a musician performing together with the robot.

For this purpose, I propose to enable the flutist robot to interact more naturally with musical partners, by implementing a Musical-based Interaction System

Figure 2.9: The figure shows a sequence of a typical instrument movement performed by saxophonist Charles Lloyd ([1]).

(MbIS). The MbIS is to be designed in order to enhance the perceptual capabilities of the flutist robot, to process both visual and aural cues occurring during the interaction.

2.4 Development of a Skill Level-dependent Musical-based Interaction System (MbIS)

2.4.1 Requirements to the Characteristics of the MbIS

The Musical-Based Interaction System (MbIS) has been proposed, to allow the interaction between the flutist robot and musicians based on two levels of interaction (Figure 2.10). The purpose of the two-level design is to make the system usable for people with different experience levels. Two human players have to get used to each other, get used to the way they interact and musically communicate with each other: I would like to introduce the same kind of behavior for the anthropomorphic flutist robot. A musician who has no experience in playing

together with the robot might require more time to adjust to the particularities of this type of human-machine interaction. For that reason, I designed the beginner level interaction system that provides easy-to-learn controllers having a strong resemblance to established studio equipment. Considering an advanced level player, I want the robot to offer a way of interaction that satisfies more refined ways of creative expression. The second, advanced level interaction system thus allows to control the performance parameters more freely.

Considering a situation of two human musicians intending to play together, the higher skilled person would always have to adjust his way of interaction to the less skilled person. Resulting in a bi-directional process, the lower skilled person also adjusts his play in order to improve his own skills. With the approach shown here, I intend to model this kind of interaction between two human players, and apply it to the communication of WF-4RV and a human musician. At first, the human player can choose the beginner interaction level to get basic knowledge about how to control the robot. As the skill of the human musician improves, he can switch to the advanced interaction level to control the robot with greater freedom and produce more expressive musical performance.

The flutist robot WF-4RV is a complex humanoid robot. It has 41-DOFs and is therefore very complicated to control. To initialize and maintain the robot, specialized technical personnel is required. When performing together with the WF-4RV, attention needs to be paid to the physical limitations of the robot. Parameters that are modulated by the user must stay within the specifications of the device, otherwise there is a risk of hardware damage. As beginning users are usually not familiar with the operation of a complex humanoid robot, the interaction system needs to be able to scale the user input to fit with the value range for valid operation. As an example, the flutist robot has a certain maximum vibrato amplitude and vibrato frequency. If these maximum values are surpassed, the vibrato mechanism of WF-4RV might be damaged. Furthermore does the robot's lung have only a limited air volume. If this air volume is exhausted the robot needs to pause operation to inhale. During this time, the user cannot musically interact with the WF-4RV.

To implement the proposed MbIS, I present the details of the audio-visual processing module interface, that enables the robot to interact with its partners.

The motivation is to enhance the cognitive capabilities of the robot, based on the principle that, if the case of a Jazz combo is considered, not only the acoustic but also visual interaction plays a key role in communication between the musicians. A major part of the typical performance of a piece of Jazz is based on improvisation. In these parts the musicians take turns in playing solos based on the harmonies and rhythmical structure of the piece. Upon finishing his solo section one musician will give a visual or acoustic signal, a certain melody, a motion of the body or his instrument, to designate the next soloist. As such, the role of visual communication in a band or orchestra is not to be underestimated. Apart musical communication that takes place acoustically during a performance, other channels are also used to exchange information. These signals change the performance, but by themselves are not supposed to be perceived as cues by the audience. While not being actually audible they do change the course of the performance significantly ([1]). To use WF-4RV as a creative tool, and to integrate the robot into a jazz band setup, the robot needs to be able to understand such cues and process them into appropriate modulation of its performance.

2.4.2 Basic Level Interaction System

In the basic level interaction stage I focus on enabling a user who does not have much experience in communicating with the robot to understand about the device's physical limitations. We use a simple visual controller that has a fixed correlation regarding which performance parameter of the robot it modulates, in order to make this level suitable for beginner players. The WF-4RV is built with the intention of emulating the parts of the human body that are necessary to play the flute. Therefore it has artificial lungs with a limited volume. Also other sound modulation parameters like the vibrato frequency (generated by an artificial vocal chord) have a certain dynamic range in which they operate. To account for these characteristics the user's input to the robot via the sensor system has to be modified in a way that it does not violate the physical limits of the robot.

As the whole interaction system consists of several modules with different tasks, for each task, there is one specific module that analyses the output from the camera and microphone of the robot and maps the extracted information to

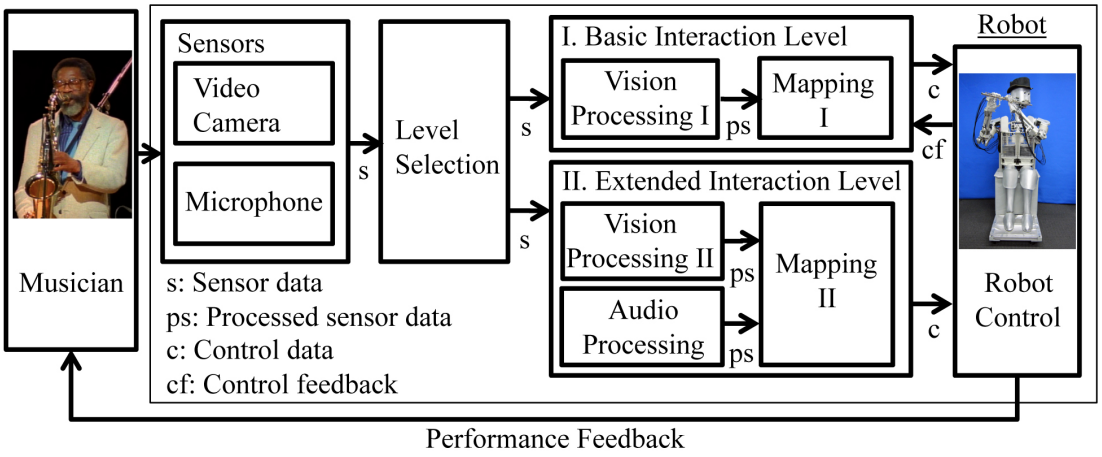

Figure 2.10: Diagram of the proposed Musical-based Interaction System (MbIS) that was implemented in the Waseda Flutist Robot WF-4RV. The system captures the performances actions of the human musician and, after the experience level selection stage, maps the processed sensor information into musical performance parameters for the robot. The robot's performance provides musical feedback back to the human musician.

parameters that modify the musical performance. From these parameters MIDI data is generated and sent to the robot. The robot motor control module receives the MIDI information and adjusts the movement of the motors accordingly. The separation of the user interface into two stages takes place in the sound and video analysis part of the system.

The basic level of interaction is composed by the following analyses: Visual Analysis (virtual Faders and buttons sense the instrument movements of the robots partner musician. Buttons switch the currently played melody pattern. The virtual fader controls the song speed) and Audio Analysis (Rhythm patterns played by the partner musician are analyzed and compared with a previously acquired library. The robot reproduces the detected rhythm pattern).

2.4.3 Extended Level Interaction System

In the extended level interaction interface, my goal is to give the user the possibility to interact with the robot more freely (compared to the beginner level). To achieve this, I propose a simplified learning (teach-in) system that allows the user to link instrument gestures with musical patterns. Here, the correlation of sensor input to sensor output is not fixed. Furthermore, I allow for more degrees-of-freedom in the instrument movements of the user. As a result this level is more suitable for advanced level players. For this task I use a particle filter-based instrument gesture detection system. A Bayesian mapping algorithm is employed in order to ensure, that if the teaching musician does not account for all combinations of instrument orientation and musical output in the teaching phase, in the performance phase the robot will automatically play the most closely matching answer modulation to a given instrument state.

The extended level of interaction requires more experience in working interactively with the robot, but also allows for more complex control of the musical performance. The user adjusts parameters by changing the orientation of his instrument. If the user is holding a wind instrument like the saxophone, he can adjust two musical parameters at the same time, one by moving the instrument sideways and the other one by bending the instrument closer or further away from the robot. These movements require more practice and are in their effect on the

performance not as easily conceivable as the techniques in the basic interaction level. One option of parameter control mapping is to let these movements modulate the amplitude of the vibrato of the flute tone. Changing vibrato amplitude is a way of enhancing performance expression, that is knowingly perceived only by more experienced musicians.

Similarly, the extended level of interaction processes the same incoming information at a higher level of perception as follows: Visual analysis (a particle filter based algorithm is used to track a musicians instrument movements. Movement information is mapped to modulate the vibrato amplitude of the robots play.) and audio analysis (rhythm and melody patterns are analyzed, matched against the robots library and, in case of successful recognition, they are reproduced).

Regarding the assignment of expression parameters of the robot to either the beginner or the advanced skill level, I would like to point out, that I implemented these two skill levels with regard to technical as well as musical conditions and requirements. The flutist robot is technically a very complex construction that requires very accurate control in every element of its body. If one part of the robot is not suitably controlled, the robot will not be able to produce any sound. As a result the ways that the performance of the robot can be controlled may be limited.

2.5 Experimental Evaluation of the Musical Performance System

2.5.1 Experiment Purpose

The purpose of the experiments in this section, is to show that the performance capabilities of the flutist robot WF-4RV are suitable for the robot to be used as an implementation platform of the proposed musical-based interaction system (MbIS). In case of an inter-active performance, the way in which the user will manipulate the musical performance of the robot is not to be determined beforehand. Every action is decided by the user, and therefore cannot be predicted by the robot. If the changes in the performance cannot be computed in advance, the

robot needs to able to process them in real-time. Only in this way it can react appropriately to *spontaneous* input by the musician.

In case of the basic level interaction system, the input controllers (virtual fader and buttons) give the musician various possibilities to modulate the performance parameters of the robot. The mapping module of this level of the MbIS is adjusted with feedback information from the physical state parameters of the robot. In case of the extended level interaction system, the musician can select the musical patterns to be performed by the robot using instrument gestures. This can result in very sudden switches of the musical material, that is transmitted to the robot. To play an appropriate interactive performance, the robot needs to be able to play these sudden transitions accurately.

To ensure this, the proposed air-flow envelope control system was developed. Using the envelope system, the lung speed is adjusted individually for each note to be played, in order to achieve precise note-to-note transitions. In the following experiments I compare the performance capabilities of the previous control system and the new system with note envelope control.

2.5.2 Experiment Conditions

I performed experiments using two different experimental platforms: the flutist robot WF-4RV without the envelope controlled tone generation system and the flutist robot WF-4RV with the envelope controlled tone generation system. As musical base material, I chose the classical piece *Ave Verum Corpus* by Wolfgang Amadeus Mozart. The piece was suggested as a test piece to evaluate the musical performance quality of the flute robot by the collaborating professional flutist. The note data of the song is sent directly to the motor control module of the flutist robot from the music sequencer software. The robot control system has no information about the characteristics of the music beforehand. Two phrases A and B were selected from the evaluation piece. Both phrases contain note jumps, that according to the professional flutist are, even for an intermediate human player, hard to perform. These two phrases were performed by the robot (with and without the envelope control system activated) and the output recorded.

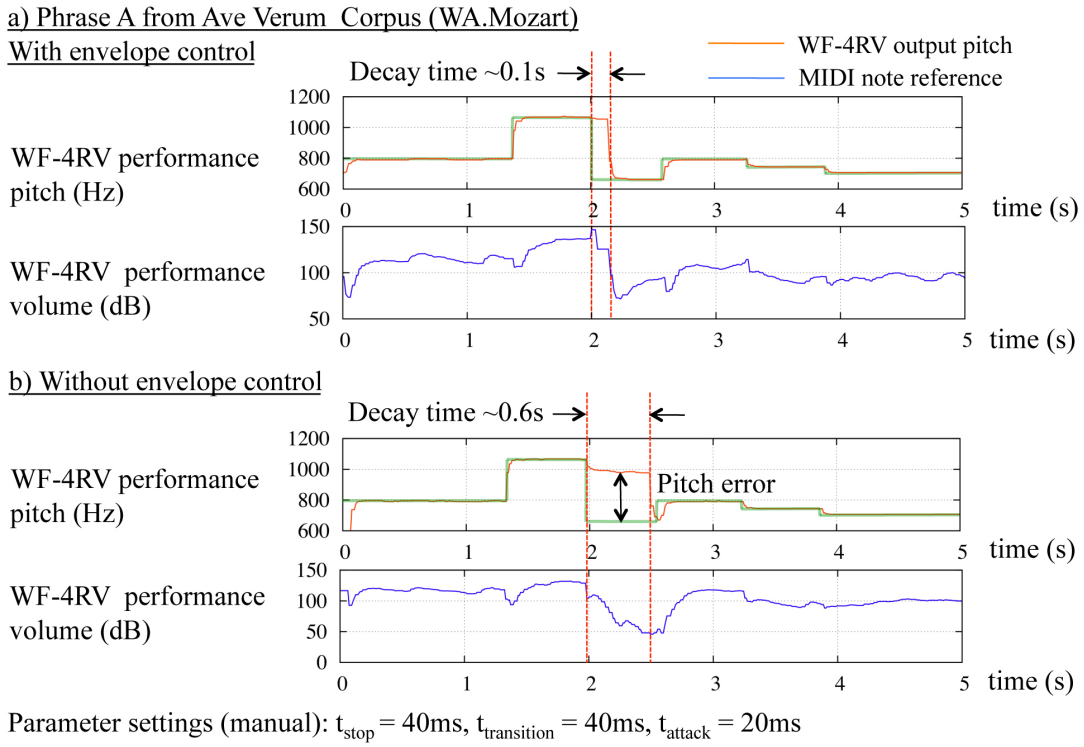

Figure 2.11: a) Shows a phrase A from *Ave Verum Corpus*, as it is played by the flutist robot, in case the air-flow envelope control is used. The pitch and volume decay time of the second note is $0.1s$ longer than decay of the reference signal. b) shows the same phrase played without air-flow control. In this case the pitch and volume decay time amounts to $0.6s$ over the reference decay time.

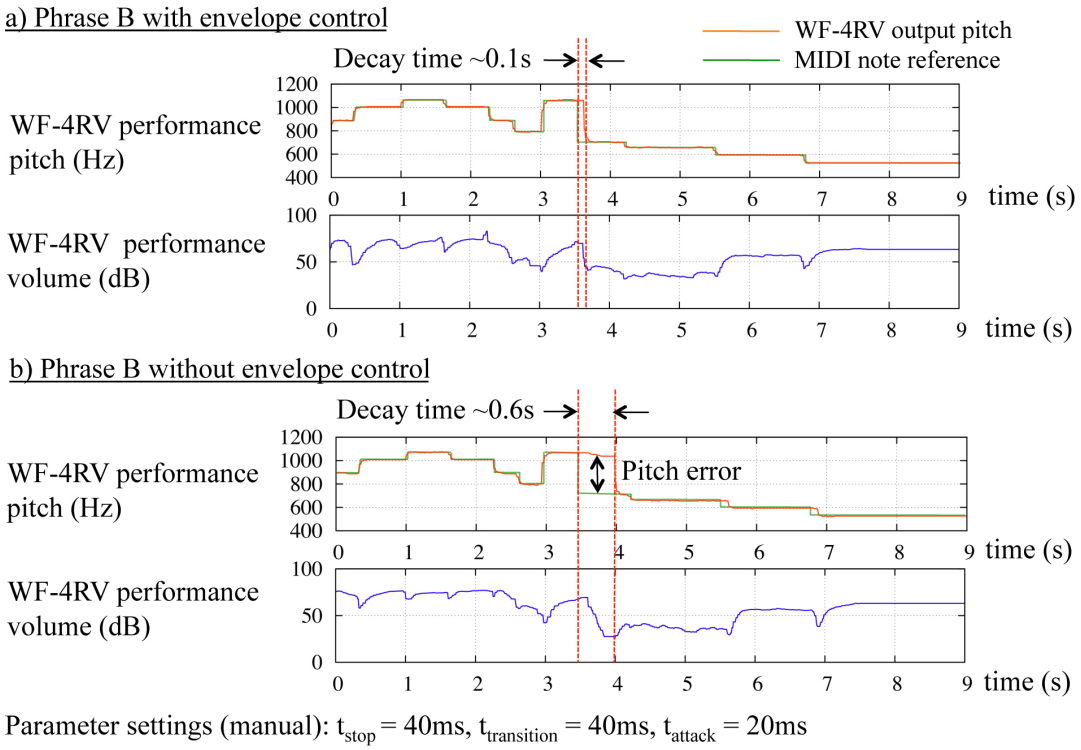

Figure 2.12: This graph shows pitch and volume plot of phrase B from *Ave Verum Corpus*. a) displays the resulting performance of WF-4RV with air-flow envelope control ($decaytime = 0.1s$). b) shows the same phrase without air-flow envelope control ($decaytime = 0.6s$). A decrease in volume of $20dB$ during the decay phase can be observed.

2.5.3 Experiment Results

I evaluated the experimental results focusing especially on two difficult phrases in the beginning and the end of the composition. The result of the experiment using phrase A is shown in Fig. 2.11. In case of phrase A both systems show an accurate performance of the phrase, except for a note jump occurring after $2s$. At the time of the note jump, the envelope controlled system is able to reproduce the notes close to the reference at $2.1s$. The system without envelope control on the other hand, is not able to perform the tone pitch control accurately. The result is a pitch curve that is very different from the reference, with the tone only being correctly produced again from $2.6s$. In case of phrase B similar results are visible. These are displayed in Fig. 2.12. The phrase is replayed similarly to the reference for both systems (from $0s$ to $3s$ and from $4s$ to $9s$), whereas the envelope controlled system outperforms the uncontrolled system for note jumps. Although the pitch reproduction accuracy is more accurate when using the envelope controlled system, there is still a slight delay of about $0.05s$ in the tone production, when a large interval occurs. This is due to the breath release time, that is inserted in-between two notes in case of note jumps. This has the purpose of avoiding overblown tones, that are visible in the pitch graph of the non-envelope controlled performance (at $3.5s$ for phrase A and at $2s$ for phrase B).

2.5.4 Discussion

The experimental results show that the tone generation is improved, when using the envelope controlled air-flow system. It is an important aspect for a spontaneous performance with the robot, that the robot is able to perform scores with note jumps well. From the recorded experimental data, I consider the performance capabilities of the robot to be suitable for an interactive performance. However, the musical material, that was used for testing was quite limited (only one exemplary tune was tested). In the following sections we apply the musical interaction system to another musical piece, the jazz standard *The Autumn Leaves*. As I will try to show later on, the system proves to be suitable for the performance of this further test tune as well. Although the new system improves the

note-to-note transitions, a certain delay of tone production can also be observed. This means, that for the system to perform well, very accurate adjustment of the tone parameters is necessary. This results in a trade-off between accurate note timing and accuracy of note-to-note transitions.

2.6 Conclusion of this Chapter

In this chapter I introduced general technical aspects of the experimental robotic system. In the first part, I described the technical setup of the flutist robot WF-4RV, specifically the mechanical hardware, with focus on the motor control electronics and software implementation. Furthermore, I explained, how the system hardware has been changed from a passive performance system to an active performance system. While working on this thesis, besides developing the various aspects of the interaction system, I also tried to improve the control electronics and software of the flutist robot WF-4RV. I rewrote the control software of the flutist robot to improve modularity and to accommodate for future extensions and improvement. I also remade the complete wiring system of the robot to reduced the power and space requirements of the robot unit. These improvements were complete within a duration of around 2 years. As a result, the flutist robot WF-4RV system is more easily initialized, safer to operate and easier to transport. By simplifying the wiring system the probability of operation failure of the robot has been reduced. I tried to put great attention to these points, as the robot has to fulfill various requirements, in order to produce a good performance.

In the second part of this chapter, I introduced the Musical-based Interaction System (MbIS). I introduced this interaction system focusing on developing a system, that is usable by users with different experience levels, is easily adaptable to different robot models, makes use of various communication channels and is able to adjust itself to the physical constraints of the robot. A system integrating all four of these points has not been developed before and is, especially in a musical context, novel and original. From a more general human-robot interaction research perspective, the system's objective is to act as a natural interaction platform between a human user and an anthropomorphic robotic entity. The system

is divided into basic level interaction for less experienced users and the extended level interaction system to accommodate for users with higher experience.

Chapter 3

Image Processing Module Implementation

3.1 Introduction

Humans communicate with each other through various perception channels, such as vision, audition and tactile feedback. Vision is one of the most important channels, through which we gather visual information about our environment and our own situation inside this environment. Using our eyes and perceptual processing connected with them inside the brain, we are able to perceive actions of other humans by studying their appearance and movements. To make a robotic system work efficiently in a human environment, it makes sense to emulate the human vision by means of a video processing system. To interact with humans, it is of advantage, if the robot is able to perceive body movements and gestures, that are performed by the robot's interaction partner. Various systems, that can track the whole human body, the human face or specific body parts, have been developed in the past ([32], [33], [34]).

Especially for our purpose of developing the Musical-based Interaction System, the detection of the instrument movements of the partner musician of the flutist robot are important. In a human band, players do not only exchange musically hearable information, but they also communicate by looking at each other and giving each other gestures with their instruments. This is specifically important in musical performances, that contain improvised content. In this case

musicians need to synchronize their play, without relying on a fixed arrangement, that defines who is going to play at which time. By using instrument gestures, the musicians can determine each other's performance role (solo player, accompaniment player) while continuing their play.

With the two vision processing algorithms that are proposed in this chapter, it was attempted, to create a perception interface for the described kind of instrument gesture communication. Regarding previous research, there has be intensive work being conducted in the field of visual gesture recognition for several years. Important state-of-the-art publications in the field are introduced in the section below (3.2). For the two different levels of the interaction system, two different vision processing algorithms are proposed. The video processing in the beginner level interaction system uses a motion area detection method to allow the user to control virtual controllers (virtual in the sense that they do not have a physical manifestation and the user manipulates them by gestures) by instrument movements. This system is designed in a musician-friendly way as the controllers resemble the virtual pendent of buttons and faders as they are used on a mixing console in a music studio environment. For this reason a musician can be expected to be familiar with the principle of operating these controls, resulting in a steep learning curve. The advanced interaction module uses a particle filter based object tracker to follow the motions of a musical instrument. As information from both of the cameras of the robot is provided, the algorithm is able to recognize three dimensional gestures.

These vision processing algorithms were chosen, to satisfy requirements that were posed to the system. It should be easy to use for the user. If possible no or a very short (one time) initialization process, should be necessary. Furthermore, we would like to, on the long term, use our interaction system in various environments, not only in the laboratory, but also at exhibitions or on a concert stage. Therefore the system should be able to cope with varying lighting environments and cluttered / changing background. As described in the previous chapter, space considerations are also important for the complete system. In case the system is to be transported to another location, the amount of computer hardware necessary to support the interaction system should be minimized. Computational efficiency of the proposed vision algorithms was therefore also considered. Connected with

this issue, processing methods were selected, that are relatively simple to be implemented, but still are suitable to support the purpose of the Musical-based Interaction System.

3.2 Related Research and State of the Art

Several methods for tracking user gestures that do not rely on a camera interface have been proposed. The BodySuit introduced by [32] registers body movements of a performer through bending sensors and converts these signals into musical performance modification parameters. Hasan ([35]) uses the principle of the Theremin instrument and other sensors to track a performers hand movements. In this work we design an interaction method for the anthropomorphic flutist player. It is an important part of our research goal to develop the different parts of the robot, resembling their human counterparts as closely as possible. This premise encourages us to recognize human gestures visually, using the two cameras built into the head of the robot.

In [36] and [37]; different kinds of tools have proposed to generate musical data from vision processing. Although two commercially available systems EyeCon ([33]) and A Very Nervous System ([38]) are generally able to output MIDI (Musical Instrument Device Interface) data from camera input, the target for these systems seem to be dance performances. In contrast we tried to optimize our system to suit the needs of a person who wants to interact with a device musically. For this reason we on the one hand designed virtual controllers similar to ones that have been used in mechanical MIDI controller devices. On the other hand we use the orientation data of the interacting musical instrument itself to control the robots performance.

Particle filtering techniques as utilized in the advanced level interaction system have been used to solve various tracking problems (according to a survey published in [39]). In [34] an approach (without the use of markers) to track the complete human body is described. In [40], hand gestures of a person are recognized using a similar principle. These works rely on tracking human body parts by analyzing their color profile. The search for the best match of an initially

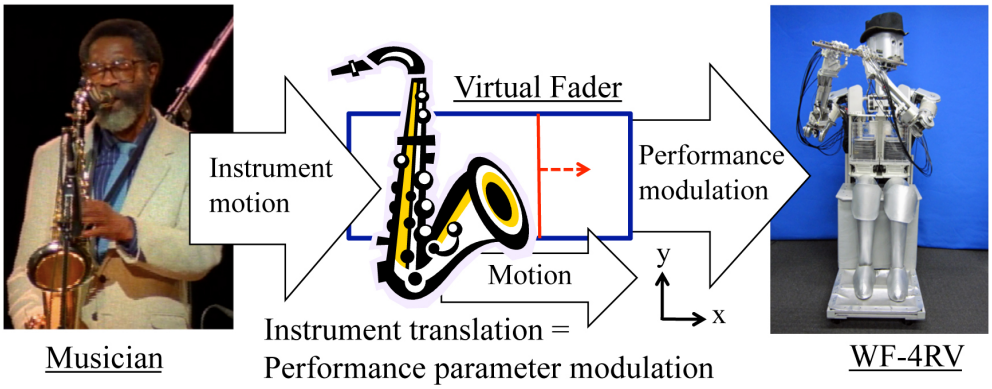

Figure 3.1: Using a virtual fader, the musician can control the performance of the flutist robot with simple instrument movements. Given a suitable mapping strategy, the motion of the instrument could control the vibrato expression of the flutist robot.

determined color profile within a certain image area is performed by a particle filter.

3.3 Basic Level: Motion Perception-based Tracking

3.3.1 Introduction

In this section we will describe the functionality of the motion tracking algorithm that we use in case of the beginner level interaction system. The method is inspired by an interface extension called Eyetoy ([41]) that the company Sony in its first version introduced about ten years ago for its gaming console Playstation. It enables players to control games by movements of their body in front of a small camera connected to the gaming console. The similarity of these games and our application with the flutist robot is that in both cases, we need to extract information about a person‘s movement in a varying environment. The general principle of functionality of the basic interaction level vision processing module is shown in Fig. 3.1.

3.3.2 Motion Perception-based Tracking System Implementation

Similar to the principles applied in the Playstation games, in our research we use only the moving parts of a video for analysis. A related method called delta framing is employed in video compression (i.e. MPEG I compression as in [42]). Thus, if we have a continuous stream of video images, for every frame we calculate a difference image with the previous frame:

$$p_r = |p_p - p_c| \tag{3.1}$$

p_r: absolute difference for the resulting pixel

p_p: pixel at the same position in the previous image

p_c: same pixel in the current image

We threshold the resulting image, and thus create a b/w bitmap of the parts in the video image, that have changed from one frame to the next:

$$p_r = \begin{cases} 0 & \text{if } p_c \leq t_r \\ 255 & \text{if } x > t_r \end{cases} \tag{3.2}$$

p_r: threshold pixel value

t_r: threshold level

p_c: same pixel in the current image

To smoothen the result we calculate a running average over several of these images [43]:

$$p_r = \alpha * p_p + (1 - \alpha) * p_c \tag{3.3}$$

p_r: average for the resulting pixel

p_p: pixel at the same position in the previous difference image

p_c: same pixel in the current image

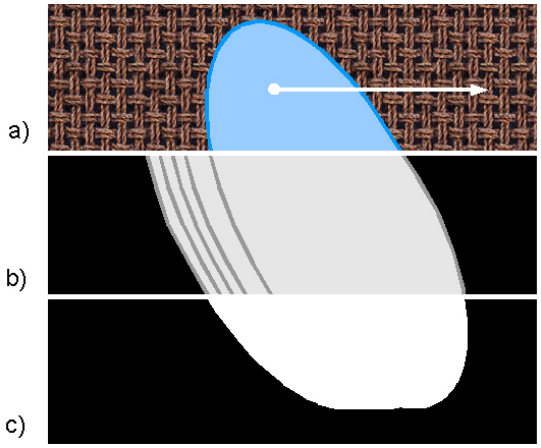

Figure 3.2: The different stages of determining the motion history of an object using the motion perception-based instrument tracking algorithm. a) shows the original (in this example synthetic) object moving over a fixed background. In b) neighboring video frames have been subtracted from each other and the subtracted images averaged over several frames to provide a motion history. c) shows the thresholded binary image resulting from these image operations.

α: Averaging factor

The different stages of this averaging process is displayed in Fig. 3.2.

In a video game the information resulting from this delta-framing method might be used to destroy enemies or do a virtual boxing fight. However, we want to use it to control musical content. In music production, composers use switches and faders to control their electronic musical instruments, so we tried to model these controls in image space. The first controller we created is a simple push-button, in functionality similar to a drum pad of an Akai MPC drum machine ([44]). These push buttons can be positioned anywhere in the video image. If a push button is triggered, a previously defined signal is sent to the flutist robot. At this stage of our research, the user can watch the input of the video camera on a monitor located beside the robot. On the screen the position of the push buttons is graphically displayed. The buttons are drawn in a semitransparent color, so the area covered by the button is clearly defined and at the same time

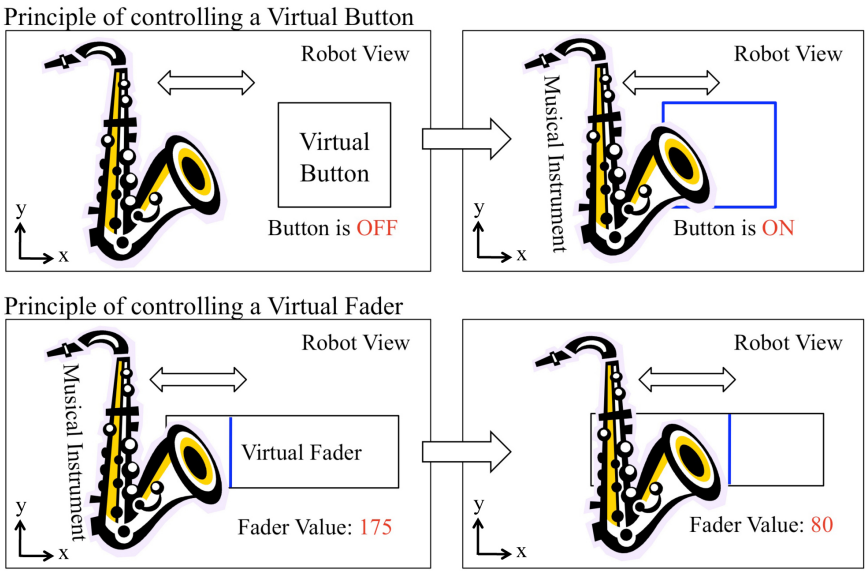

Figure 3.3: Virtual buttons and faders resemble music studio controllers, that can be manipulated by instrument motion. The buttons allow discreet ON and OFF states, triggered by the instrument moving into button area. Faders record continuous values that are adjusted by the position of the instrument.

the video image beneath can still be seen. To detect if a button is switched on or off, we employ the algorithm in Fig. 3.4.

As a second controller we implemented a virtual fader, that can be used to continuously set a performance value. In this case, the position of a fader can be changed for example by a motion of the hand. For each change of the fader position, a MIDI controller message is sent to the robot. The fader slowly resets itself to a default position after it has been manipulated. This prevents a fader from remaining in an erroneous position that might have resulted from background noise (i.e. a person moves in the background of the image, causing an undesired change of the fader position). A fader can be deliberately positioned in the image and orientated in any angle to allow the user to easily adjust it to his control requirements and physical constraints. The principle functionality of both, the virtual fader and the virtual button are displayed in Figure 3.3. To determine the position of a fader, we use the algorithm in Fig. 3.5. A second

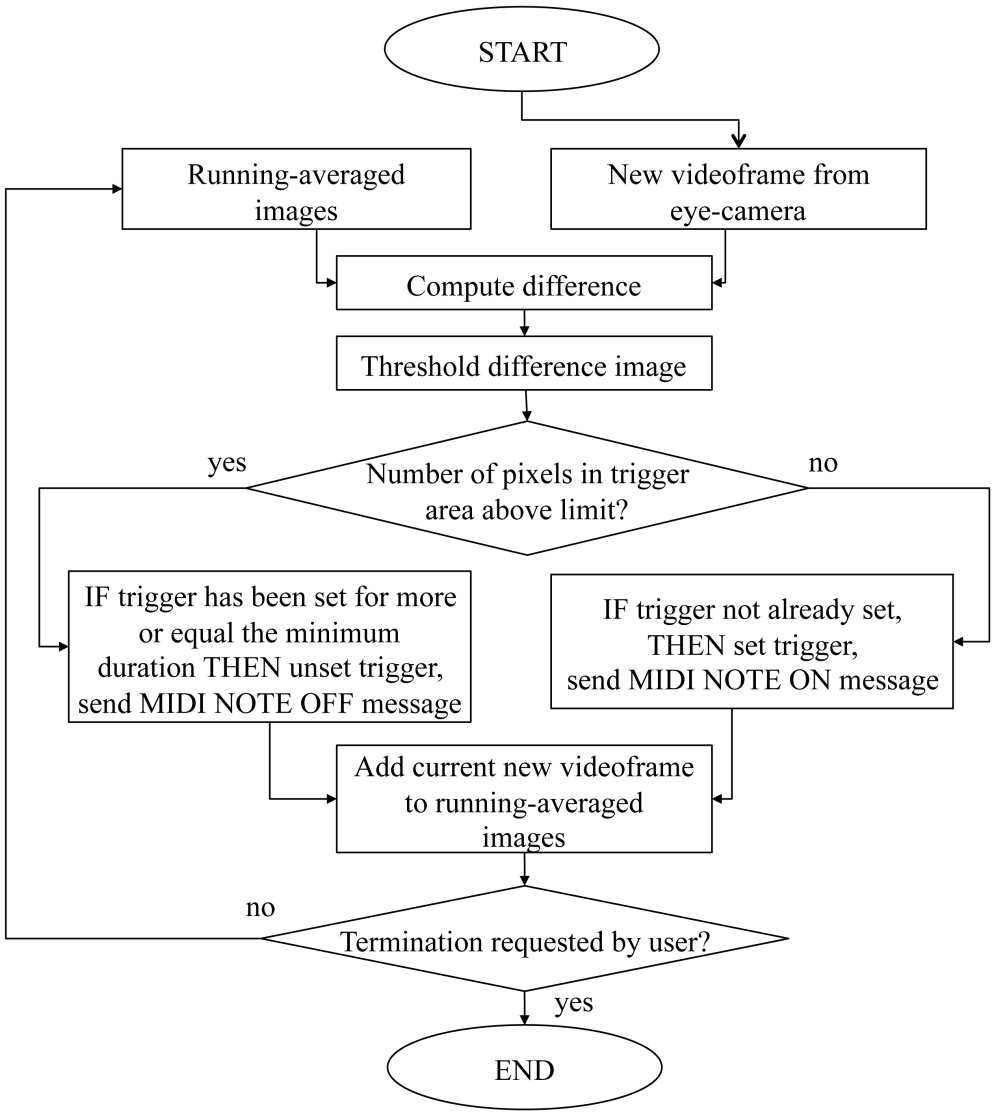

Figure 3.4: The figure displays a flow-chart of the algorithm to determine the current state of a virtual button. The decision if a button is triggered depends on the number of motion pixels detected within the button area. Depending on the application environment the trigger threshold can be adjusted.

implementation using an algebraic calculation to determine the new fader value v has also been implemented:

$$\sin\alpha = \frac{y_p - y_0}{|\underline{r}|} \Longleftrightarrow \tag{3.4}$$

$$\alpha = \arcsin\frac{y_p - y_0}{|\underline{r}|} \tag{3.5}$$

$$\cos\beta - \alpha = \frac{|\underline{v}|}{|\underline{r}|} \tag{3.6}$$

$$v = \cos\left(\beta - \arcsin\frac{y_p - y_0}{|\underline{r}|}\right)|\underline{r}| \tag{3.7}$$

$\underline{r}$: distance vector between (x_0, y_0) and (x_p, y_p)

$\underline{v}$: fader value vector

α: angle between r and the x-axis

β: orientation angle

The calculation of the control value of a virtual fader is graphically shown in Fig. 3.6. The position of the baseline (the position the fader automatically adjusts itself to, when it is not touched) of the fader can be freely defined. For some control parameters of the flutist robot a homing position at zero might not be optimal. For variations in vibrato expression, the baseline position might be at a moderate vibrato amplitude value, allowing the fader to be moved to actuate slight positive and negative alterations to the vibrato amplitude.

3.3.3 Experiment Objective

For the purpose of experimentation in case of the basic level interaction we chose the very reduced setup of two virtual buttons and one virtual fader. In this configuration we mapped the virtual fader to control the pattern play tempo of the robot. The two virtual buttons allow to switch between two melody patterns. We chose the notes of melody pattern one and the notes of melody pattern two to be within separate octaves, to be easily able to distinguish between the two when doing pitch analysis on the sound result. The song tempo is varied between

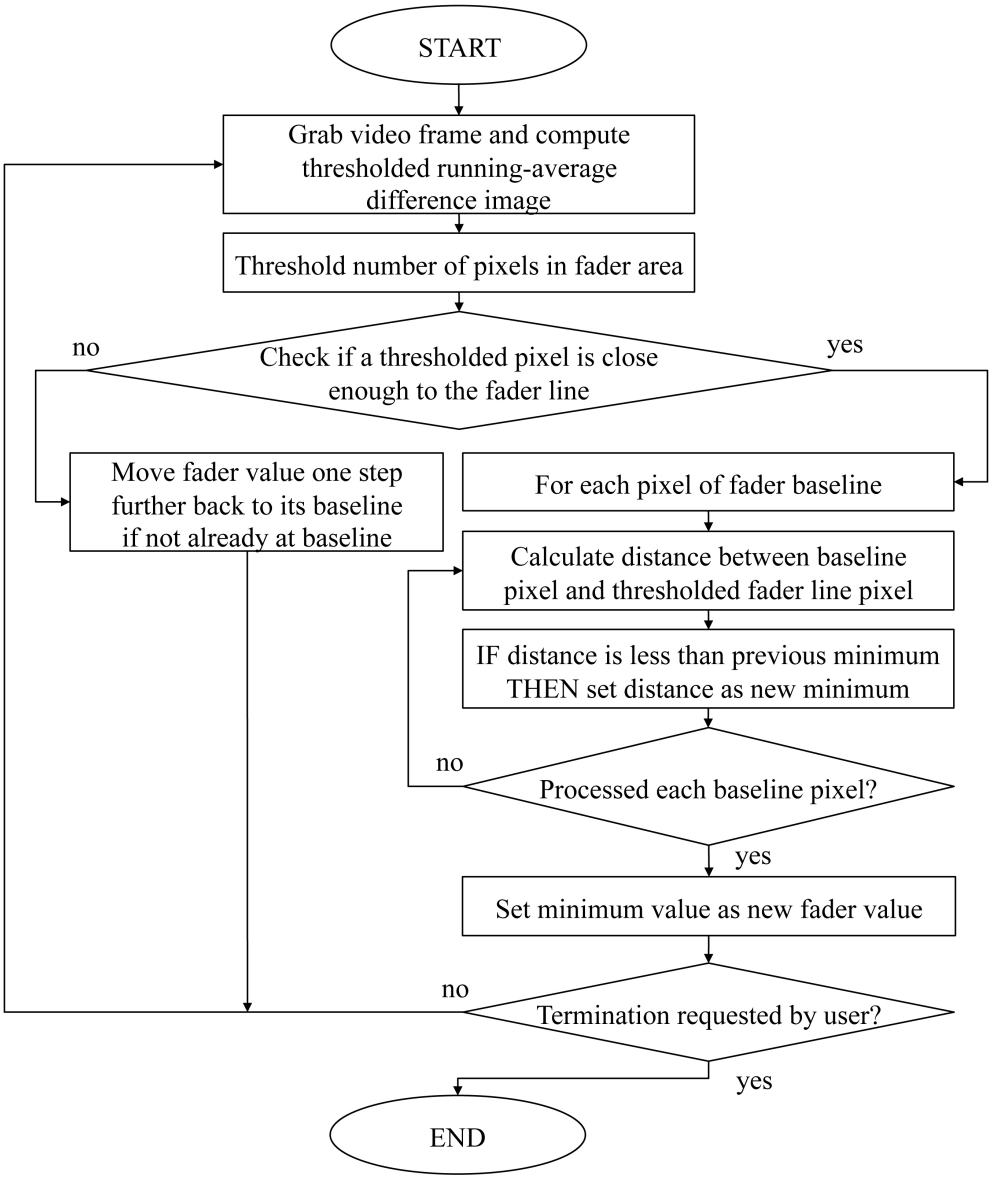

Figure 3.5: The figure displays a flow-chart of the algorithm to determine the current value of a fader. Also in case of the fader, the number and position of active pixels within the fader area determine the fader value. Additionally, to prevent the fader value from jumping, only motion pixels with a certain vicinity of the current fader position are included in the calculation.

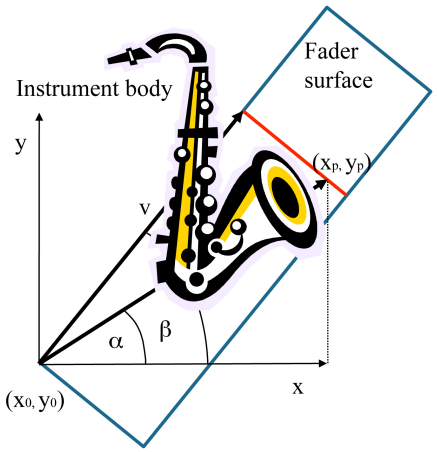

Figure 3.6: The figure outlines the functionality of the fader given a certain instrument position and fader angle. Here, β is the angle of the fader controller and α is the angle of a vector pointing to one motion pixel on the fader line.

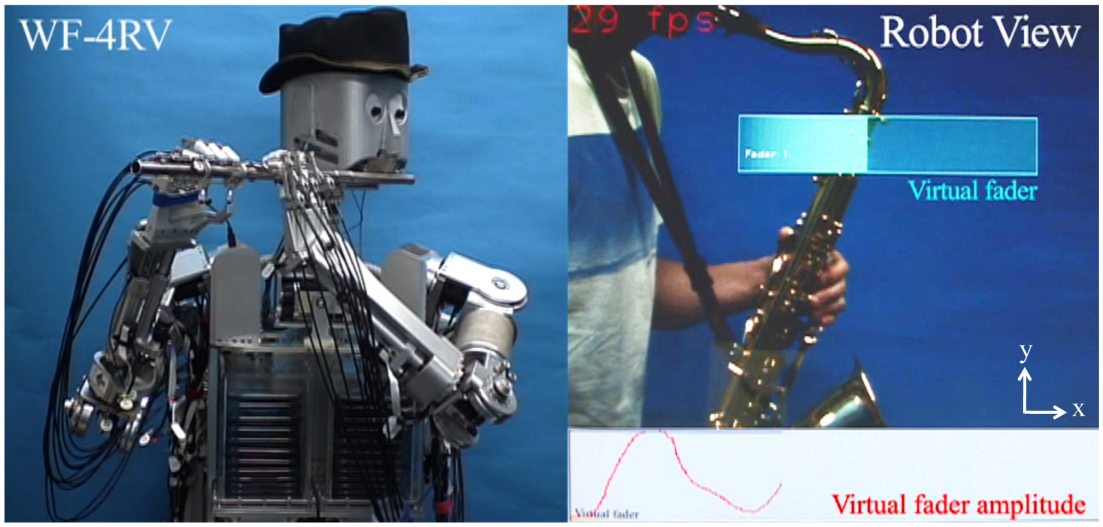

Figure 3.7: In this screenshot a saxophone player controls a single virtual fader with the cone of his instrument. The amplitude of the virtual fader shown in the small graph on the bottom changes according to the musician's movement. The left side of the picture shows the flutist robot WF-4RV.

$60bpm$ and $160bpm$ (bpm = beats per minute). Although the robot would be able to play up to much higher tempos, we chose the upper and lower limit of the song tempo to be within musically reasonable limits.

In both experiment modes the interacting person is standing in front of the robot inside the viewing angle of the robot cameras. The interacting person can see a graphical representation of the interaction interface as well as the video image recorded by the cameras on a computer screen installed beside the robot. The user, an intermediate level musician, is instructed to utilize the interaction interface to vary the flutist performance in a way that is comprehensible in a musical sense, performing more or less regular movements with his instrument.

For each of the parameters that we intend to manipulate we perform a separate experimentation run. That results in three experiments: the virtual buttons changing the current melody pattern, the virtual faders changing the song tempo and the instrument orientation varying the vibrato amplitude. For each of these experiments we record separate data, that we also analyze separately.

In case of changing melody patterns during interaction using the virtual buttons, one melody is pattern is played and once it is finished the system waits for the next event. In case of continuous control, one melody pattern is infinitely looped, as our aim is observe continuous changes on the sound output. The easiest way to do that is to keep the current melody pattern constant and to vary the expression parameters. A screenshot of a saxophone player using a virtual fader is shown in Fig. 3.7.

3.3.4 Experiment Method

We measure the status of the interaction system in two places. We monitor and save the events of a button being triggered and the continuous values of the faders when they are sent from the computer vision module to the mapping module. At the same time we record the sound output of the robot with a microphone placed in front of the robot. We records both types of information in a music recording application (Ableton Live). This is the advantage of sending the data from one module to the next in MIDI data form, with one application we can handle discreet data (MIDI notes), continuous data (MIDI controller data)

and audio information (recorded with an audio interface). The disadvantage is that the resolution of continuous controller data is limited to 8 bit. However for our purpose of qualitatively evaluating the interaction system this resolution is considered to be sufficient.

When analyzing our experiments we intentionally do not work with ground proof data coming from a third-party motion tracking device. We bear in mind that the accuracy of our system may be inferior to that of other available systems. Our intention is to create a user interface that allows spontaneous interaction with a robot. For the interacting person the feeling of spontaneousness does not so much depend on the accuracy of the algorithm but on its responsiveness and intuitiveness of use. For the data analysis we compare the system input, the movement of the user's instrument, respectively the results from the computer vision, to the sound output of the robot. We use a FFT spectral analysis algorithm to detect the pitch and amplitude of the music data. This enables us to qualitatively extract pitch / melody pattern, song tempo and vibrato amplitude from the music data.

For discrete information input in case of the virtual buttons, we separate the course of one experiment into several stages or states. For example if the user triggers button one, we enter state A (high notes melody pattern), because button one is associated to the melody pattern that lies above the pattern for button two. Respectively the same happens when touching button two, state B (low notes melody pattern) is entered. To verify the responsiveness of the interaction itself we observe in the graph if the state of the resulting sound of the robot is according to the input state. After one melody pattern has been played the system enters an idle state until the next event. When the artificial lungs of the robot are empty and the robot needs time to refill them, it is not able to generate sound for a certain duration. This state is called *breathing point*. During continuous control in case of interaction using the virtual fader, we only consider two different stages. One is the continuous control state, in which one melody pattern is looped, and the other state is again the breathing state where sound output is stopped.

The virtual buttons have a width of $80 pixel$ and a height of $60 pixel$, resulting in an area of $4800 pixel^2$. If the musician stands in front of the robot at a distance

of $3m$, this pixel are relates to an area approx. $1m^2$ in the camera plane. In case of the virtual fader, the controller has a length of $200pixel$ and a height of $60pixel$, which results in a controller area of $12 * 10^3 pixel^2$. The musician can therefore control the virtual fader in a space of $2.5m^2$.

3.3.5 Experiment Results

Regarding the basic interaction level, Fig. 3.9 and Fig. 3.8 show the outcome of our experiments. Fig. 3.8 presents the results for a musician manipulating two virtual buttons (button A and B), switching between two melody patterns. The graph is divided into several phases that resemble the activity states of the interaction system. *Low notes pattern* states are triggered by the user touching the area of button A. When the button is activated the computer vision module sends a MIDI signal to the performance control module to trigger the assigned musical pattern that is to be played by the robot. The result of the pitch analysis of the recorded output can be seen in the pitch graph of Fig. 3.8. A trigger of button A results in a low note pattern being played consisting of MIDI notes 67 (g4), 71 (b4), 74 (d5). Touching button B respectively results in notes 77 (f5), 81 (a5), 84 (c6). The last note of a pattern is continued until the next button trigger, in a state that we call *idle state*. At that time, the robot does not do anything else than hold the last note sent from the performance module. We observe regular operation of the buttons at $3s$, $8.5s$, $13.5s$, $18.3s$, $22s$ (for button A) and at $0s$, $5.5s$, $10.5s$, $18s$, $27s$ (for button B). The button trigger at $18s$ is to be especially considered, as here button B is triggered while the low notes melody is still playing. The trigger of a new melody while another one is playing makes the robot switch patterns immediately. After the switch has been conducted, operation is continued as usual. As we can see from the pitch analysis, response to button activity is fast, in the graph we observe not delay or lag. When we look at the volume plot we can identify areas, where the level suddenly drops for a certain duration. These moments are called breathing points and relate to the time when the robot's lung system is deflated and needs to pull air in order to be able to produce the air-beam necessary to generate the flute sound. As the lung breathing speed is constant we see these events regularly happening at $4s$, $15s$

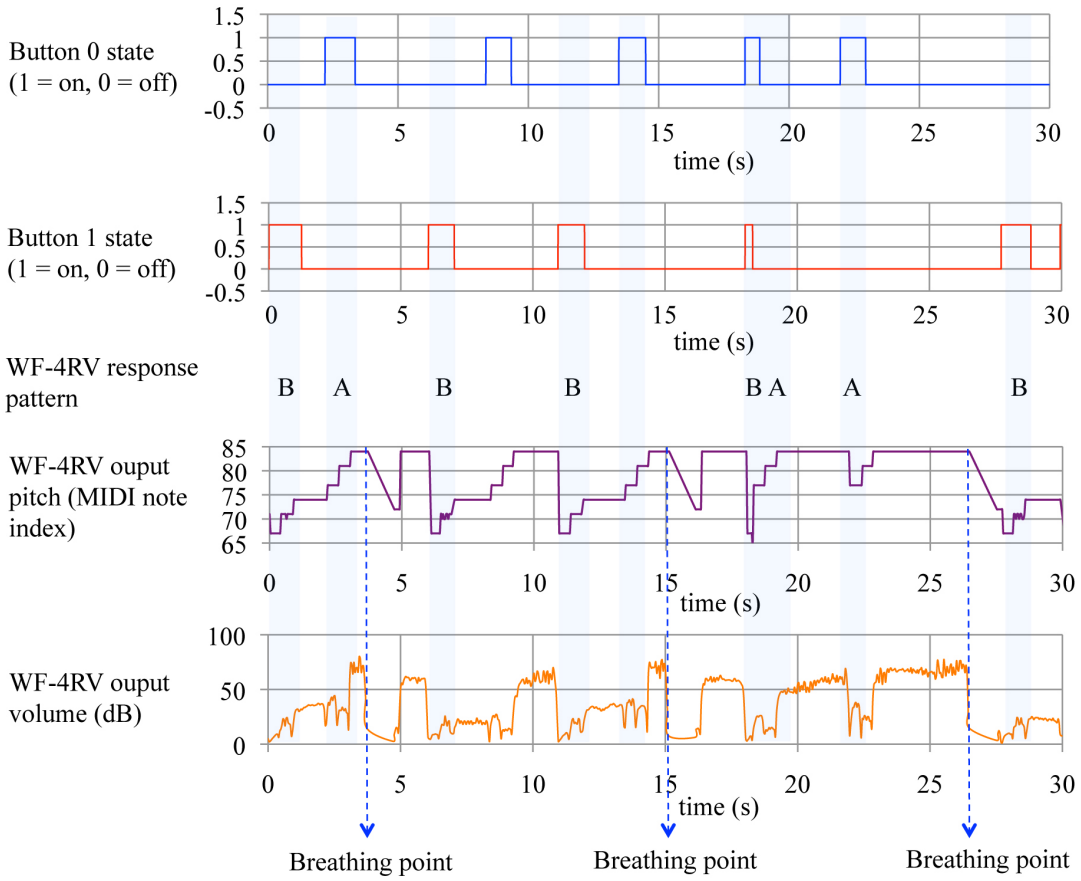

Figure 3.8: Similar to the previous graph this graph shows the relationship between the recorded user movements triggering two virtual buttons and the analysis of the sound output of the flutist robot. The resulting activity stages represent the two different melody patterns that are associated to the two buttons, an idle phase and the breathing point.

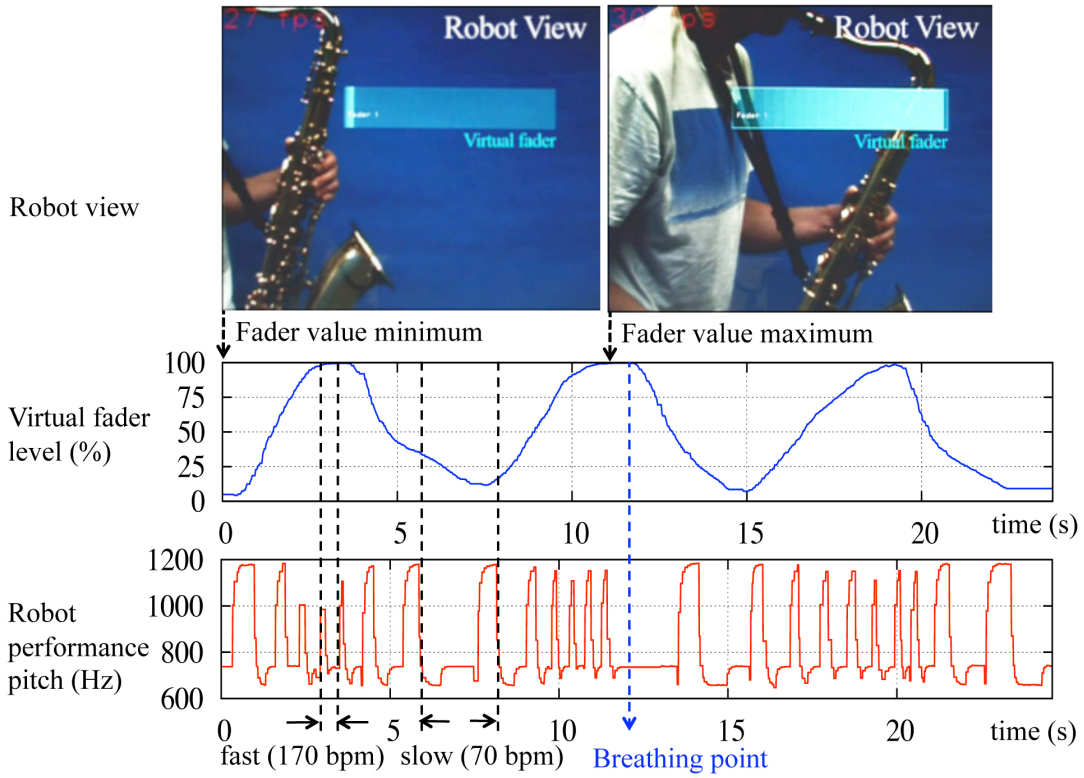

Figure 3.9: The graph shows the relationship between the recorded user movements adjusting a virtual fader and the analysis of the sound output of the flutist robot. According to the amplitude of the fader (here displayed as relative percent value), the tempo of the robot performance changes with a range of a maximum of $170bpm$ to a minimum of $70bpm$.

and $26s$ in the graph. The duration of one breathing phase is $\approx 10s$ long, with the 'breathing in' taking approximately $1s$.

Fig. 3.9 shows results of the experiment to evaluate the functionality of the virtual fader in order to adjust the musical interaction between the human player and the robot. The virtual fader manipulates the speed of the pattern currently being played by the robot when moving a virtual fader in level 1 interaction mode. Movements are executed in different tempos, From $5s$ to $10s$ slow movement is performed, whereas the time from $17s$ to $30s$ is dominated by faster changes. Fader movement relates to changes in song tempo as can be observed in the pitch graph. The higher the fader value the closer the tones recorded in the pitch graph move together on the timescale. When closely observing the tones in the pitch plot we see that a note duration in case of the furthest fader movement ranges from about $0.2s$ long, which relates to $170bpm$ in song tempo, and $1s$ for the fader position closest to its baseline, which relates to around $70bpm$. These values comply with what we previously determined as the highest and lowest tempo values for our pattern. We conclude, that the pattern play tempo does change as intended in compliance with fader movement. This is true for the whole duration of the measurement except for the breathing points that occur in the same way as in the previous experiment. The time between two breathing points is again approximately $12s$ and their duration is $1s$. This is also reflected in the volume plot, that although it shows changing volumes for the different tones of the played pattern, at the breathing times the volume drops significantly.

As a main difference to the virtual buttons, during control using the virtual fader there are only two possible states, the continuous control state where the fader movement adjusts the song speed, and the breathing point where the robot's lungs are refilled. It remains to note, that the breathing tempo of the lung does not change when the fader volume changes. The speed for the upwards movement of the lung stays the same during the whole experiment. Regarding the irregular volume values for different notes, due to the nature of sound generation at the flute mouthpiece, with constant air pressure from the lung, the output volume varies from note to note. As a drawback, between the recorded fader values and the sound output slight lagging can be observed. This is a result of the running average we apply to the digitized video input. Depending on the application of

the system the amount of averaging can be varied. Basically the averaging can be regarded as a low-pass filter: The more averaging, the smoother the recorded data will be, but as a trade-off, the response time of the system will be slower. However, we made the practical experience that this error does not strongly affect the applicability of the system.

3.4 Extended Level Interaction System: Particle Filter-based Tracking

3.4.1 Introduction

With our second vision processing approach, used in the advanced interaction level, we concentrate on creating a visual interface, that enables a human to control the robot through gestures with his musical instrument as freely as possible. A further emphasis here is to create an interface, that allows on the one hand to control the musical expression of the robot accurately, but also robustly and in a computationally efficient way. In the past we have exhibited the robot on several occasions and we found that each place had very different lighting and background conditions. Our gesture detection method should not only work in an optimized laboratory environment, but also in real environments; thus, we need to find a way to cope with these varying ambiances. There might be a constantly changing background (i.e. people passing by, stopping to watch the robot perform) and below-optimum lighting sources (i.e. stage lighting facing into the cameras of the robot). Another difficulty is that the humanoid head of the robot can move during a performance. This requires our image processing algorithm to adapt rapidly to fundamental changes of background (fundamental in contrast to a small object being moved in the background). Particle filter-based tracking has been reported to achieve reliable and robust tracking results in various environments. For this reason we chose this algorithm as tracking method for the advanced interaction level. The basic principle of the particle filter-based tracking is shown in Fig. 3.10.

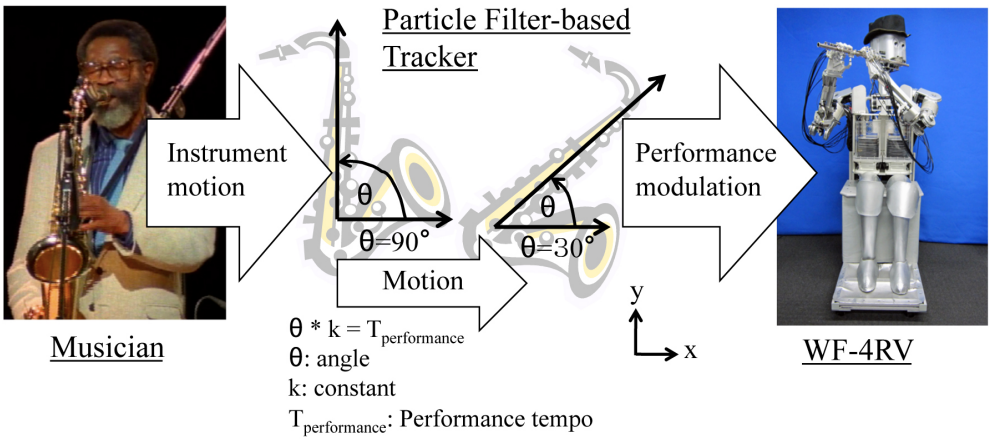

Figure 3.10: Another way of controlling the expression of a musical performance robot is the use of a particle filter-base color histogram tracker. In this application the inclination of the instrument of the musician is tracked, by following the color profile of the musician's hands holding the instrument.

3.4.2 Mathematical Background

The particle filter is used to predict the object positions in our video sequence. These object states behave like a hidden Markov process. That means that we have a sequence of unobserved (hidden) states x_k from time $k = 0$ to $k = n$ ($x_{0:k}$) and parallel to these a sequence of observed states y_k ($y_{0:k}$). The relationship of these unobserved and observed states is characterized by two probability density functions ([45]):

- $p(x_k|x_{k-1})$ for $k >= 0$ (one actual state x_k only depends on the state one time-step prior in time, k indexes the time-step)
- $p(y_k|x_k)$ for $k >= 0$ (an observation y_k only depends on the current state x_k)
- the sequence has an initially known distribution $p(x_0)$

We use the probability density function to describe the development of our system as it contains all the necessary information about the probability distribution of

possible states in a convenient form. We have

$$\mathrm{E}(x) = \int_{-\infty}^{\infty} x\, f(x)\, dx \tag{3.8}$$

to calculate an estimation $\mathrm{E}(X)$ of a state variable x from the probability density function $f(x)$. In order to do our object tracking we want to know which state an object is likely to have, given a certain observation. We gain information about this state from evaluating the posterior density function $p(x_{0:k}|y_{1:k})$. Particularly for the particle filter, the posterior density function is approximated by generating a set of random samples and assigning weights to these random samples according to their importance (definition of the pdf for a discrete random variable from [46]).

$$p(x_{0:k}|y_{1:k}) \approx \sum_{i=1}^{N_s} w_k^i \delta(x_{0:k} - x_{0:k}^i) \tag{3.9}$$

where $\{x_{0:k}^i, i = 1, ..., N_s\}$ are N_s random samples with associated weights $\{w_k^i, i = 1, ..., N_s\}$. We wrote $y_{1:k}$ (index starting at 1) as the initial state x_0 is known and needs not be observed, thus there is no y_0.

The weights w_k^i are normalized such as that

$$\sum_{i=1}^{N_s} w_k^i = 1 \tag{3.10}$$

which means that

$$w_k^i = \frac{w_n^{*i}}{\sum_{i=1}^{N_s} w_n^{*i}} \tag{3.11}$$

with $\{w_n^{*i}, i = 1, ..., N_s\}$ as the non-normalized weights.

We pull these samples from a proposed probability density function (PDF) $q(x_{0:k}^i|y_{1:k})$ that characterizes our assumption about the posterior state ([47]). The ratio

$$w_n^{*i} \propto \frac{p(x_{0:k}^i|y_{1:k})}{q(x_{0:k}^i|y_{1:k})} \tag{3.12}$$

thus describes how well our current random draw $x_{0:k}^i$ fits the the real state. Furthermore, as both pdfs should describe the same reality we have

$$\int p(x_{0:k}^i|y_{1:k}) dx_{0:k}^i \approx \int q(x_{0:k}^i|y_{1:k}) dx_{0:k}^i \tag{3.13}$$

As we are dealing with a sequence of video frames (resembling our Markov chain), we need to find a rule how to augment each weight when transitioning to a new time-step [47]. By (first for non-discrete states x) factorizing q in

$$q(x_{0:k}|y_{1:k}) = q(x_{0:k}|x_{0:k-1}y_{1:k})q(x_{0:k-1}|y_{1:k-1}) \tag{3.14}$$

and using Bayes' estimation on our prior like

$$p(x_{0:k}|y_{1:k}) = \frac{p(y_{1:k}|x_{0:k})p(x_{0:k})}{p(y_{1:k})} \tag{3.15}$$

we finally get (now for the discreet case)

$$w_k^{*i} = w_{k-1}^{*i} \frac{p(y_{1:k}|x_{0:k}^i)p(x_{0:k}^i|x_{k-1}^i)}{q(x_{0:k}^i|x_{0:k-1}^i, y_{1:k})} \tag{3.16}$$

By setting the proposal distribution [48]

$$q(x_{0:k}^i|x_{0:k-1}^i, y_{1:k}) := p(y_{1:k}|x_{0:k}^i) \tag{3.17}$$

we reach

$$w_k^{*i} = w_{k-1}^{*i} p(y_{1:k}|x_{0:k}^i) \tag{3.18}$$

, which means that we modify the particle weight from a previous frame only with the likelihood $p(y_{1:k}|x_{0:k}^i)$. In case of the proposed color histogram tracking method we calculate a new weight by multiplying its old value with the measure of similarity between the color histogram of its new (determined by Gaussian distribution) position and the original color histogram of the tracked object, determined at initialization. To measure this likelihood we apply the Bhattacharyya [49] coefficient of similarity between histograms,

$$\rho[p^i, q] = \sum_{u=1}^{m} \sqrt{p_u^i q_u} \tag{3.19}$$

with p^i being the histogram of one seeded particle, q resembling the object color histogram sampled at initialization and m expressing the histogram size indexed by u.

Although we adapt new object coordinates only from the particle with the highest likelihood, as the particle filter method works recursively, the information about the other particles are not lost. A particle with an initially lower than

maximum likelihood is not abandoned, but can still propagate to gain more likelihood. However, research on particle filters has shown that in case all particles are kept for the whole tracking run, all particles but one tend to be degraded to probabilities close to 0. There are several ways to counteract this behavior [47]. We have chosen the method of re-sampling. After each new predict-update cycle, particles with a probability lower than a certain threshold are exchanged for newly initialized particles. This threshold as well as the optimum number of particles to be used have been determined manually.

3.4.3 Implementation

We propose to implement color histogram matching ([50]) and particle tracking ([47]) to follow the movement of a musical instrument. The combination of the two methods is an established way to follow an object with a certain color profile. We propose this method, because compared to the previously proposed motion perception-based tracking using the virtual faders and buttons, it allows more freedom of movement for the user. In case of the virtual faders and buttons, the controller areas are fixed in the video image and cannot be moved during the performance. The tracking algorithm proposed in this section gives the user the possibility to move into any area of the image, while still being able to control the robot with his instrument movements.

The system is initialized manually by defining the starting positions of the player's hands (Figure 3.11). For the computation of the 3D data, the algorithm makes use of a stereometry mapping technique. We do not calculate a complete depth map of the scene, but limit our matching to the four patches found by the particle tracker. One might argue that we could also use 2D techniques to compute a depth image: as we try to emulate human perception, we prefer using two cameras and stereo pair matching. The proposed technique also saves resources due to the limited number of points being calculated.

To begin with, the user marks the initial location of the image areas to be tracked. Four areas are selected: One patch for each hand in the left eye camera and one patch in the right eye camera. A color histogram is generated from each of the selected image areas. Once a new video frame is acquired we find

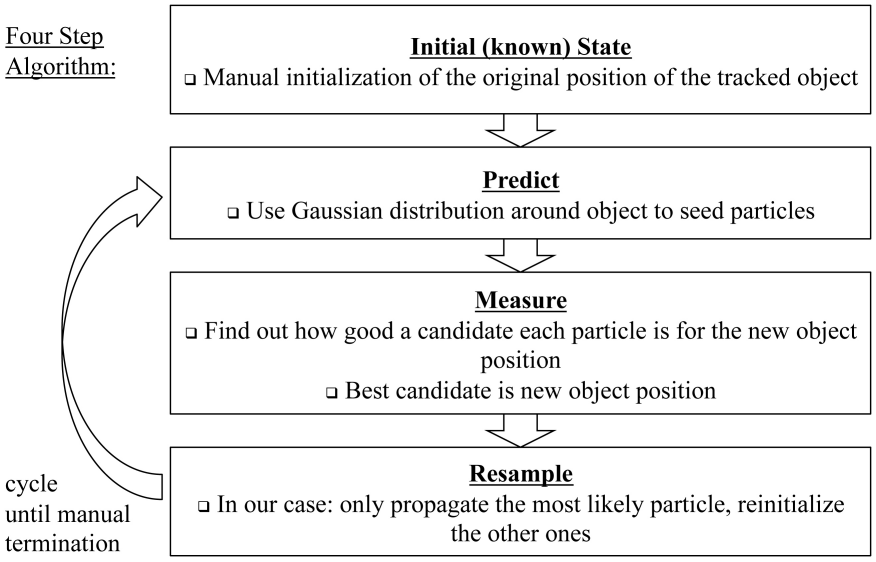

Figure 3.11: The figure shows the cycles of our particle filter system. The system is initialized with a known state. The following state is predicted by seeding particles and the new state of the object measured by comparing the candidate particles. The re-sampling stage re-initializes a certain amount of particles to avoid degradation.

a random distribution of locations around the previous location of each of the patches. For each of these particles a histogram is generated and compared with the initialization histogram. The particle with the most similar color profile is the new position of the tracked patch. Particles with less likelihood are also saved, each weighted according to its likelihood. The more particles are being used the more accurate the method becomes.

After we found the x-y-axis coordinates of the image patches, we now calculate the distance of a patch from the camera. To achieve this we examine the difference between the x-position of the patch in the left camera image to the x-position in the right camera image. The larger the difference, the closer the patch is located to the camera. Without calibration we cannot determine the absolute distance of the patch from the camera. However, if we work with relative values to compute only changes in orientation and position, this is not necessary.

We model the shape of the saxophone as a line. The hands of the player are located on two spots on that line. The average of the position of the hands is recorded as the center position of the saxophone. Similarly we deal with the orientation: We consider a line drawn from the center of one hand to the center of the other. The inclination of the line is the orientation of the instrument. There is no ambiguity about the position, as normally a player would not hold the instrument upside down. From the 2D coordinates of the four hand particles we calculate the relative position, inclination and rotational angle of the instrument. To compute the depth values of both hands we use a z-transformation:

$$\Delta x_p = abs(x_{pl} - x_{pr}), z = \frac{1}{\Delta x_p}\alpha \tag{3.20}$$

Δx_p is the distance between the x-axis of the patch in the left camera image (x_{pl}) and the right camera image (x_{pr}). Accordingly z denominates the z-axis of the patch. We use α as a constant to adjust the value of Δz for further calculations.

Inclination and rotational angle are obtained by transforming the Cartesian coordinates resulting from the object tracking into a cylindrical system. Δx, Δy,

and Δz denote the vector between the saxophone player's hands.

$$\Delta x = x_1 - x_0 \tag{3.21}$$

$$\Delta y = y_1 - y_0 \tag{3.22}$$

$$\Delta z = z_1 - z_0 \tag{3.23}$$

$$\theta = arctan(\frac{\sqrt{\Delta x^2 - \Delta y^2}}{\Delta z}) \tag{3.24}$$

$$\phi = arctan(\frac{\Delta y}{\Delta x}) \tag{3.25}$$

for $\Delta x \wedge \Delta y > 0$.

We use a cartesian coordinate system parallel to the view plane of the robot to calculate roll (ϕ), pitch (θ) and yaw (ρ) inclination of the instrument.

$$\phi = arctan(\frac{\Delta y}{\Delta x}), \theta = arctan(\frac{\Delta y}{\Delta z}), \rho = arctan(\frac{\Delta x}{\Delta z}) \tag{3.26}$$

The flutist robot performance can be controlled by MIDI data. Two kinds of MIDI data are evaluated by the robot: A MIDI note (containing pitch and note duration information) triggers the tone generation routines of the robot. As the flute is a monophonic instrument, only one tone can be played at a time. The robot evaluates MIDI controller data as continuous information. Tone volume and vibrato amplitude are manipulated by sending MIDI controller data to the robot.

3.4.4 Experiment Objective

In this section I introduce an experiment, that was conducted, in order to evaluate the technical functionality of the particle filter-based tracking algorithm. The purpose of the experiment is to confirm, if the the proposed algorithm works as has been intended with the development. In case of the extended interaction level vision processing system this means, that by using the proposed method,

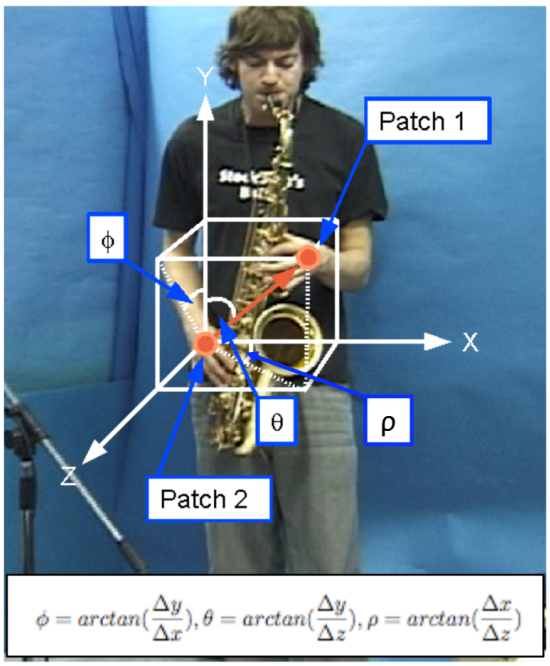

Figure 3.12: The orientation angles of the instrument are calculated from the two patch positions. In this picture, a saxophone is shown as an example. The method itself is applicable to other kinds of woodwind instruments as well.

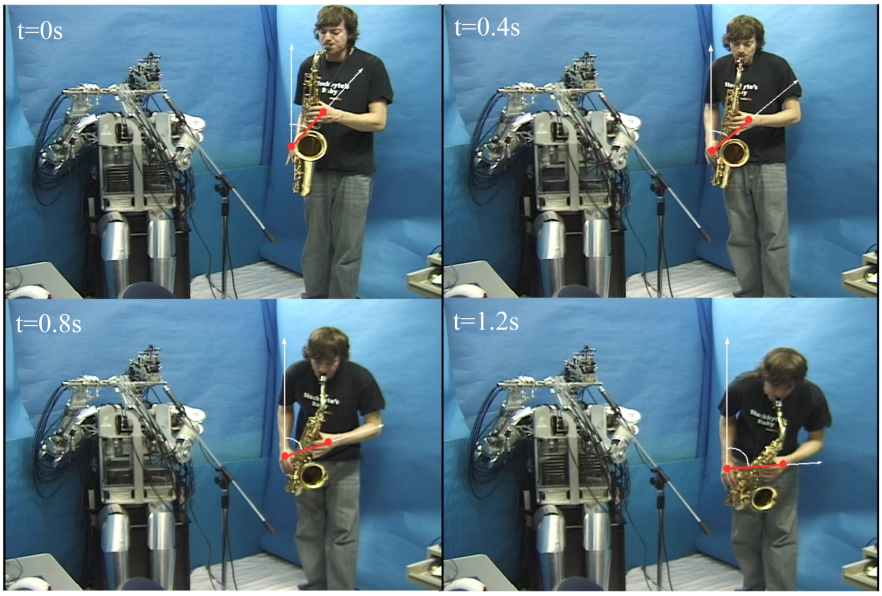

Figure 3.13: The orientation of the instrument is altered by leaning forward with the upper body. This movement is typical for giving a cue in a band performance.

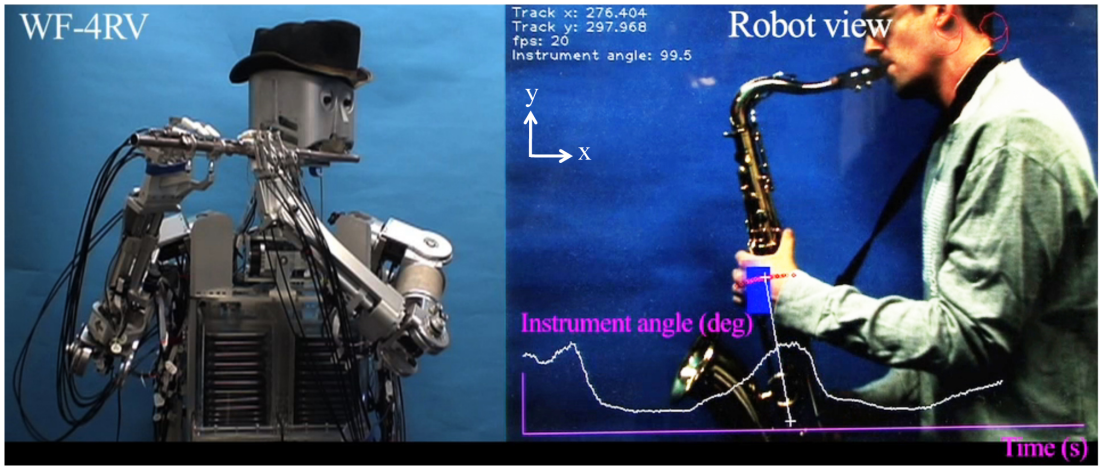

Figure 3.14: The figure shows a screenshot of an experiment in which the particle filter-based color tracker is used to follow the instrument angle of a human saxophone player. In the lower right graph the the currently detected angle is plotted. On the left side the flutist robot WF-4RV is shown.

the flutist robot is able to track the instrument movements of a human wind-instrument performer. These movements might then be mapped to modulate the performance parameters of the robot. In comparison to the basic interaction level vision processing system, the particle filter-based tracking gives the partner musician of the robot the possibility to perform movements with a higher degree of freedom. As a result, the musician can move his instrument anywhere in the image space, while the robot tracks the instrument orientation. In the experiments we did not use simulated input to the tracking system, but asked a human musician to perform typical instrument movements in front of the robot camera. As I will show in the further chapters, I would like to evaluate the system particularly for the case of natural and intuitive interaction within a human band. The suitability of the proposed particle filter-based tracking to be integrated into such a system is to be demonstrated with the following experiment.

3.4.5 Experiment Method

In case of the particle tracking algorithm, rather than proposing an experiment that is very complex (e.g. integrating the system in its current form into a human band), we decided to perform basic experiments to explore the capabilities of our idea. In the experiment the vibrato amplitude of the flute player is manipulated by changing the saxophone orientation. We considered that the absolute positioning of the saxophone relative to the robot does not have to be evaluated: the orientation of the saxophone is calculated from the absolute position values of the hands. Thus, is the absolute position of the instrument calculated as the mean of the hand positions. The saxophone can be rotated by the player, in terminology changing ‘pitch‘, ‘roll‘ and ‘yaw‘. Fig. 3.12 shows a 3-dimensional representation of the detection method of the saxophone orientation. Fig. 3.13 shows a typical movement sequence, that is performed during control of the flutist robot using a wind-instrument. An image of the robot view of such an interaction is shown in Fig. 3.14. For the experiment we asked a human wind-instrument player (saxophonist) to stand in front of the robot and perform instruments that, he considers typical during a musical band performance. After initializing the system to recognize the color histogram of the hands of the musicians, the robot used the

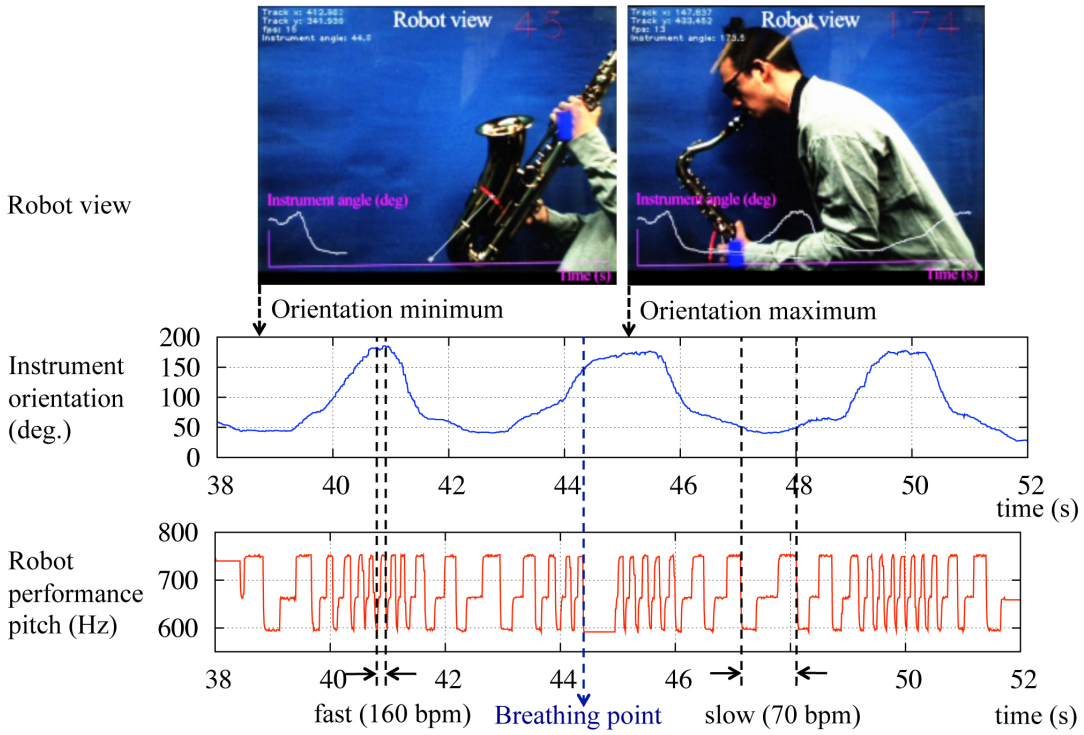

Figure 3.15: The graph shown in this figure displays the modulation of the performance tempo of the flutist robot proportional to the inclination angle of the instrument of the musician. At at orientation of 180 degree the performance speed is set to $160bpm$. An orientation of 50 degree relates to a tempo of approximately $70bpm$.

particle filter-based tracker to follow the musician's instrument orientation. The angle of the instrument orientation is directly translated into modulation of the flutist robot's performance tempo. I recorded the musical output of the flutist robot and, in order to evaluate the result result of the experiment, analyzed its pitch.

3.4.6 Experiment Results

The experimental results for the advanced interaction level are displayed in Fig. 3.15. Movements of the instrument in front of the robot camera is recorded as orientation values that change the vibrato amplitude of the flute sound. To keep

the analysis as simple as possible only one tone is played, as can be observed from the horizontal line of the pitch plot. The average volume of the sound output becomes higher with less vibrato as the amount of air streaming through the glottis mechanism of the robot, that controls the vibrato, is at these times higher and produces louder volume. However, the vibrato oscillates over this average value and changes its amplitude according to the orientation value calculated by the particle tracking algorithm. At $16s$ we observe a maximum of approximately 100 controller tics that relates to a vibrato amplitude of $10dB$ in the volume plot. A minimum at $25s$ accordingly produces a very low vibrato amplitude. At $9.5s$, $19s$ and $29s$ we, in compliance with the previous experiments, find breathing points. The distribution of the breathing points is equidistant and of equal duration. The re-filling period of the lung is the only time that the tone pitch is not reported as constant. At that time, when the robot does not produce any sound, the reported pitch values result from the environment noise recorded by the microphone.

3.5 Discussion

The results for the motion perception-based tracking as well as for the particle filter-based tracking, show that the movements of a musical wind-instrument can be tracked using the proposed method. In the result graphs we see that motion of the instrument of the experiment subject proportionally relates to the musical output of the flutist robot. However, both methods have their advantages and disadvantages. The virtual fader and virtual buttons on the one hand, require no initialization procedure and work very reliably. On the other hand, as they there position on the screen is fixed, they do not allow for much freedom of movement by the musician. In case of users with little experience level, this can be of advantage, but for users with more experience, I proposed the particle filter-based tracker. As described before, this tracker follows the hands of the musician by evaluating color histograms. The result graphs show the relationship of changing the inclination of the musical wind-instrument of the experimental subject with the musical output of the robot. Although this method requires initialization before being used, it provides more flexibility of movement to the musician. For the calculation of the instrument angle, relative position values

are used. Therefore, the user can translate the instrument to any position in the image space, without producing wrong results. The experiments show, that the requirements to the tracking algorithms were satisfied: With both methods the motion of a wind-instrument can be sufficiently tracked, to modulated a musical parameter of the flutist robot. All the experiments were performed under a single controlled environment. So far I conducted the experiments under constant lighting, in a quiet environment with static background (laboratory space). The application of the proposed algorithms in a realistic stage environment has so far not been evaluated.

3.6 Conclusion of this Chapter

Two image processing methods for the musical based interaction system were introduced. The suitability for basic interaction using direct mapping has been shown. Vision is one channel for information exchange between a musician and the flutist robot. In the experiments presented in this chapter, the two proposed vision processing algorithms for the basic and extended level of interaction are shown to be a suitable suitable to calculate information about the musician's instrument motion. As performance material a short phrase from the jazz standard *The Autumn Leaves* has been used. The virtual buttons gave a musician the possibility to trigger two different melody patterns. The virtual fader and the particle filter-based tracker were installed to continuously regulate performance parameters of the flutist robot continuously. In both experiments the performance tempo of the robot is controlled from $70bpm$ to 170 for a duration of approx. $20s$. Detected orientation angles of the wind-instrument range from approx. $-90°$ to $90°$. In the following chapter, I will introduce an audio processing system, that uses a second communication channel, the acoustic channel. In the interaction system, information from both these channels is fused and concerted into musical performance parameters. Both proposed methods are novel and original in their implementation, and have been specifically developed for this thesis.

Chapter 4

Audio Processing Module Implementation

4.1 Introduction

In the previous chapter, the vision processing modules of the Musical-based Interaction System have been introduced. In this chapter we would like to propose a module, to process interaction information through the auditorial communication channel (see Figure 2.10). In human communication acoustic interaction is important for various reasons, one very important being the transmission of information through human language. In the context of the application to the MbIS, we propose a acoustic processing module, that allows the robot to analyze the melodic and rhythmical content of its partner musician's performance.

The proposed algorithm is able to perceive rhythm and melody information from the acoustic input to the microphones of WF-4RV. The extracted data is associated with a library of patterns, that is determined by the user before starting the performance with the robot. To perform the basic frequency analysis of the audio signal, regular filtering, Fast Fourier Transformation and peak analysis is applied. To compare perceived musical content with library patterns we use a histogram matching method.

There has been various research performed regarding acoustic processing algorithms, very strongly also related to the field of music. Important scientific works in this field are introduced, in section 4.2. The basic audio processing method

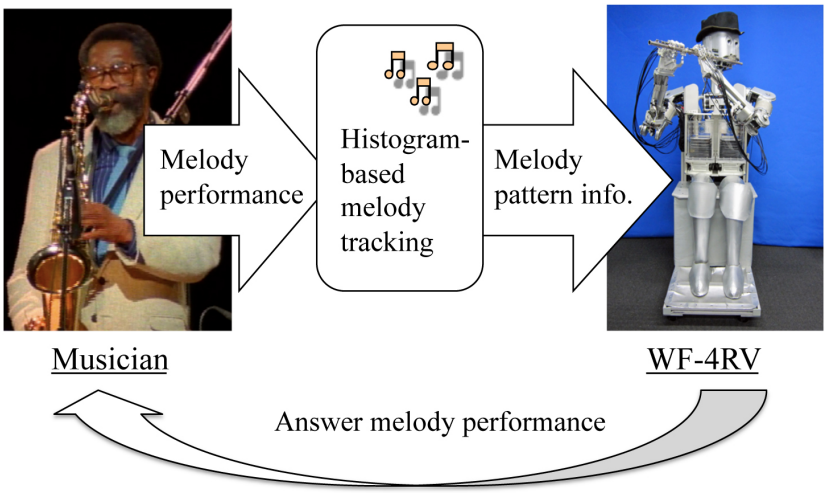

Figure 4.1: The figure shows the functional principle of the histogram-based melody tracking. Input to the melody tracking module is the sound data of the performance played by the musician. Output of the module is the index of the detected melody pattern. After the musician plays a melody, this melody is classified, if it matches one of the patterns in the library of the robot. If a matching pattern is found, this pattern is mapped to a specific response by the flutist robot.

that we use is very similar to what is proposed in other scientific work. What is specific to our implementation, is the way, that we compare existing musical pattern information (rhythmic and harmonic information) to currently recorded material. To do this, we create histograms, containing the note timing values for rhythmic tracking, and the note pitch information for harmonic tracking. These histograms are created for the real-time detected musical information and the information contained in the musical pattern library of the robot. To evaluate a currently played musical pattern, the histogram of the pattern is compared with the histograms contained in the library, using a special comparison coefficient (the specific application of this coefficient will be described further below in section 4.3.1).

We chose this approach for the acoustic processing module, because it seems especially suitable to be used within our interaction system. The rhythm as well

as the harmony detection algorithm do not need to be initialized specifically for one user. This makes the method to be comfortably usable, also by users with little experience in interacting with the flutist robot. The histogram method is relatively easy to implement and computationally efficient (Operation time of O(1)). The required result of a pattern comparison, that corresponds to a detection match can be gradually adjusted, so that the method is adjustable to environments, that are more noisy. In this case the required matching accuracy would be reduced (which also has an influence on the false positives, though), so that the impact of the environment conditions is compensated (in terms of true positives). The general principle of functionality of the histogram-based melody tracking is shown in Fig. 4.1.

4.2 Related Research

Extensive research has been done in the real-time rhythm extraction and the real-time detection of harmonic structures within audio recordings. Various methods have been proposed to detect transients in musical data, a requirement for performing rhythmic analysis. Klapuri ([51]) uses division into frequency bands and linear regression to detect the starting point of a rhythmical feature. Division into high and low-energy peaks in addition to timing criteria ([52]) is applied to allow application on polyphonic sound data. Rao-Blackwellian Models ([53]), Online Onset Detection Models ([54]) and Brownian Motion Models ([55]) are commonly used to extract tempo and structural information.

A popular approach for harmonic analysis (phonic transcription) is the so called Blackboard Method. Originally not designed for audio analysis it has been applied in this area by various researchers ([56], [57], [58], [54], [59], [60]). The method is somewhat similar to the algorithm described here as it also uses a ‘knowledge source‘ (which in our case consists of a library of possible melody patterns) to reinforce recognition. Multiple model based techniques have been proposed using approaches ranging from spectral template matching ([61], [62]) to sequential Monte Carlo methods ([63], [64], [65]).

The idea of using histograms for characterizing audio material has been used mainly in the context of efforts to archive large amounts of musical data ([66],

[67], [68]). Based on this previous research we try to exploit the technique as a simple method to characterize and match smaller pieces of harmonic and rhythmic information.

4.3 Basic Level Interaction System: Histogram-based Rhythm Detection

4.3.1 Implementation

In this section we will describe the rhythm identification algorithm's principle of operation. Our purpose is to extract real-time, rhythmic information from the recorded sound data. The analysis result is matched with a library of timing patterns that are saved as previous knowledge in the robot. The algorithm determines the best matching pattern and passes this information on to the mapping module, in order to generate an output performance by the robot.

The performance of an instrumentalist is recorded and the data directly streamed to the analysis algorithm. In the beginner level interaction mode, because we are interested in the rhythm information contained in the acquired data, we examine it for timing characteristics. In the sound waveform separate notes are represented as distinguishable amplitude peaks. We isolate these peaks by thresholding, as it is shown in Eq. (4.1).

$$a_t = \begin{cases} 0 & \text{if } i_t \leq m \\ i_t & \text{if } i_t > m \end{cases} \tag{4.1}$$

a_t: thresholded sound wave value

m: threshold level

i_t: input sound wave

It is to note, that we consider only the case of *portato* play. If the interacting musician plays legato, separate notes cannot be reliably identified by the algorithm. Also we do not divide into frequency bands, as we restrict our approach to monophonic signals.

The duration of one tone impulse naturally is longer than a certain minimum time. In order to prevent very short noise peaks from falsely triggering the threshold we smoothen the sound wave with a running average calculation (Eq. (4.2)). This computation acts, from a signal processing point of view, similar to a low-pass filter ([43]):

$$p_r = \alpha * p_p + (1 - \alpha) * p_c \tag{4.2}$$

p_r: average for the resulting pixel

p_p: pixel at the same position in the previous difference image

p_c: same pixel in the current image

α: averaging factor

A gate control in a music studio works similarly. Instead of a running average such a gate effect uses adjustable attack (the time a certain amplitude needs to be retained for the gate to open) and release (the time the gate is open after the amplitude has dropped below the threshold) times. In case of our threshold filter, using the averaging calculation, we have equal attack and release times. This is sufficient for our approach as we do not intend to generate well-sounding output (as is the intention of a gate effect in music production) but to suitably preprocess the data for further analysis (Figure 4.2).

The rhythm patterns have a certain length. To identify the most recently played pattern we do not need to analyze all of the previous sound input. We rather use a window that always contains only the most up-to-date part of the recorded music information. This window continuously slides forward as new data is acquired. The size of the window is the length of the longest rhythm pattern in the robot‘s pattern library. Regardless which pattern is currently played by the interacting musician, it will always completely fit inside the window.

Each positive edge of the thresholded sound wave in the time window represents a rhythmic pulse. To characterize the timing of this sequence of pulses as a whole we calculate the time differences between adjacent pulses. Utilizing this information we can construct a histogram, with one bin representing one certain

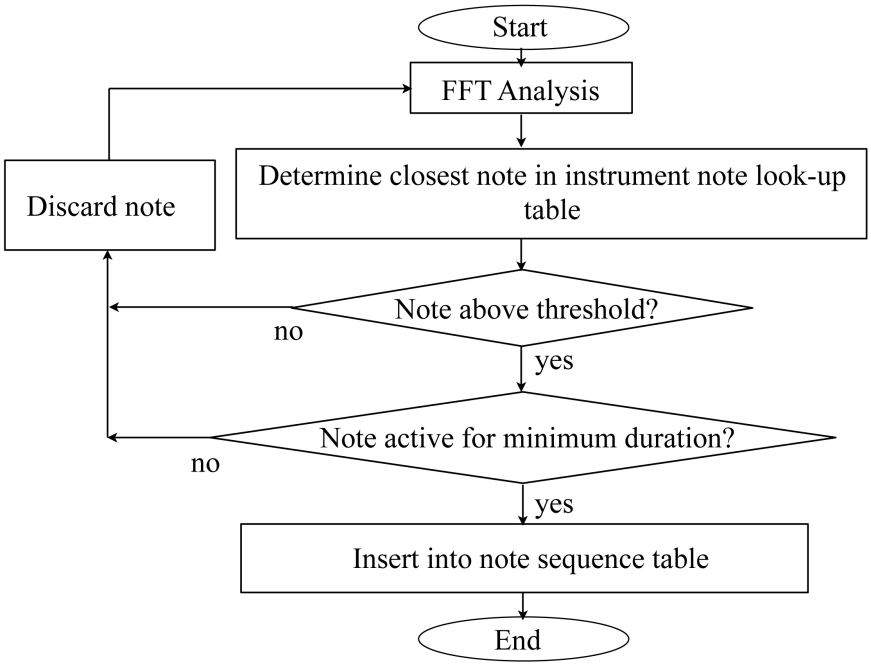

Figure 4.2: This image displays a flow-chart of the note perception algorithm. After a transformation into frequency-amplitude space, note-value, minimum volume and minimum duration are confirmed.

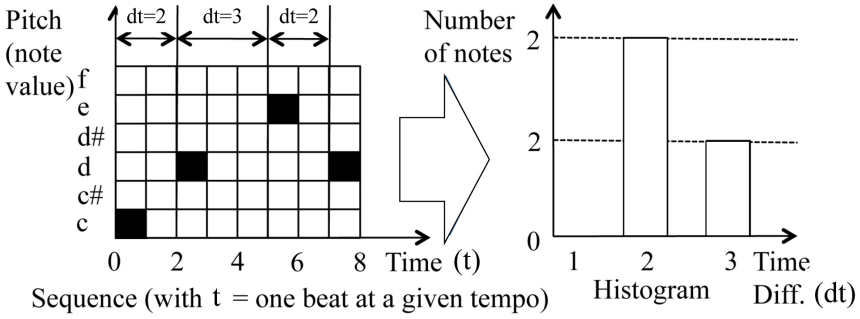

Figure 4.3: This figure shows the concept of creating a histogram from the rhythm information within a score. Each column in the histogram represents a certain difference in note-onset time. In the example two notes with a onset-difference of $2T$ and one note with a delta of $3T$ are inserted into the sequence.

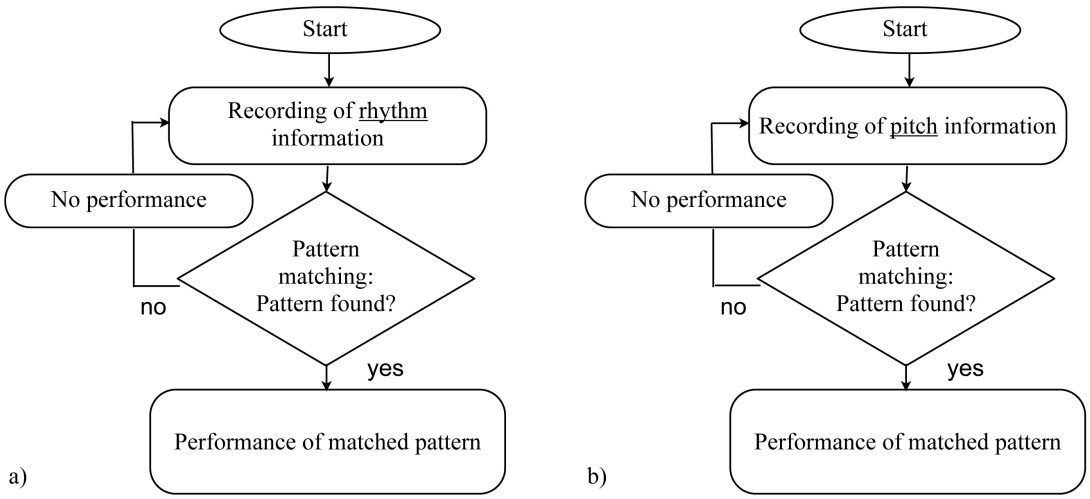

Figure 4.4: These flow charts show the experimental procedures for a) rhythm matching and b) melody matching. In case of the successful recognition of a pattern, the matched pattern is repeated by the flutist robot.

time difference. Both axis of the histogram are normalized, with the result that the maximum and minimum bin of the histogram always relates to maximum and minimum pulse delta time. This histogram is then compared to the histograms of the timing patterns in the library of the robot. The similarity between two histogram is determined using the Bhattacharyya [49] coefficient (Eq. (4.3)).

$$\rho[p^i, q] = \sum_{u=1}^{m} \sqrt{p_u^i q_u} \tag{4.3}$$

with p^i being the histogram of one library pattern, q resembling the sampled rhythm pattern and m expressing the histogram size. The sum is indexed by u (Figure 4.3). To prevent patterns from being falsely detected we apply a threshold to the similarity coefficient. If the result of the pattern comparisons falls below this threshold the robot does not recognize the input as a known rhythm. The result of the rhythmic analysis is the best match from the rhythm pattern library.

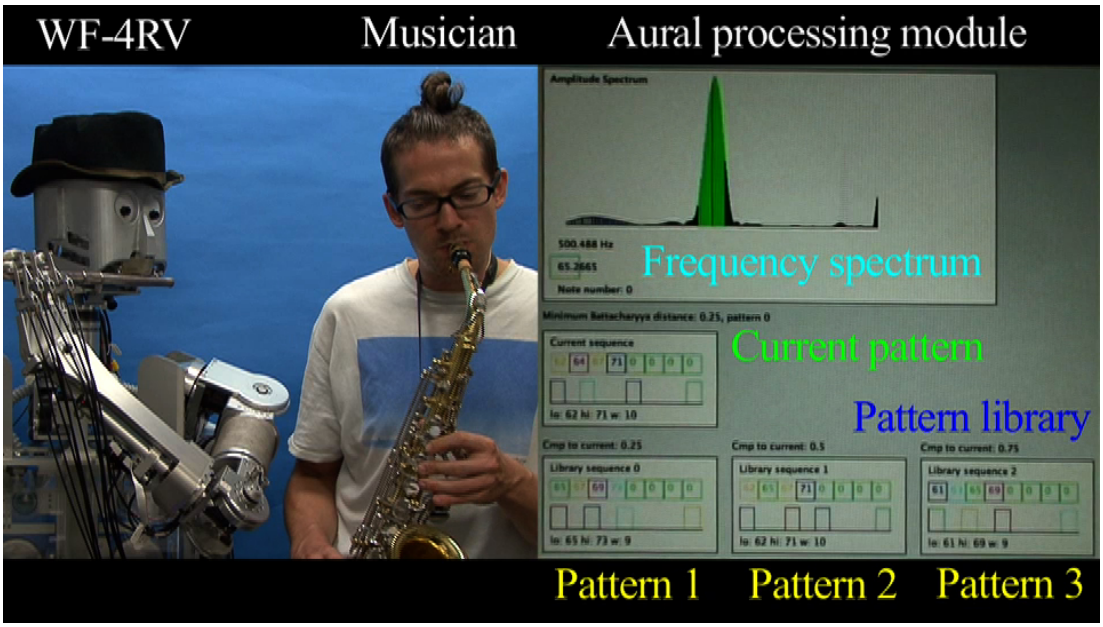

Figure 4.5: The screenshot shows a typical configuration of the audio processing module. The pattern, that is currently played by the human musician (left image), is compared to the pattern saved in the pattern library (right image). In this example the musician is performing pattern 2.

4.3.2 Experiment Objective

The purpose of our experiments is to show how well a user can express his musical intention using the provided interaction setup. The interaction system itself resembles a closed control loop, with the robot on one side and the human musician on the other side. The fact that decides about the quality of the output is how responsive the robot is to the actions of its interaction partner. In the same way, naturally it also depends on the skill level of the human player. To accommodate for these parameters our interaction system is separated into beginner level interaction module and advanced level interaction module. For each of these, we propose separate experiments. Although we examine the two modules in different experiments we in principle use the same setup and evaluation method for both (Figure 4.4).

In case of the beginner level interaction interface, the robot is programmed with a library of three rhythm patterns. We chose these patterns specifically to be easily discernible to prevent false recognition. For the current state of our research, our purpose is to verify the functionality and also practicability of the system as such. In the discussion we would like to point out, that there are still several issues with the algorithm that might be improved. Different rhythm and melody patterns can lead to identical histograms and therefore lead to false positive recognition. The FFT processing algorithm needs to be carefully adjusted to the average volume of the output of the performer. In the experiments these settings were manually adjusted by the user.

4.3.3 Experiment Method

The experiment itself consist of a human saxophone player situated in front of the robot at a distance of about 2 meters. The human musician has knowledge about the three patterns that are contained in the robot‘s library. The three patterns are melody excerpts from the jazz standard *The Autumn Leaves* (pattern 1, 2 and 3). With the start of the experiment, the subject will begin to deliberately play one of these rhythm patterns on a single note. He will repeat playing this rhythm until the robot responds. After that he will, again randomly, choose the next pattern. This procedure is repeated for several times. The responses

of the robot as well as the play by the musician that triggered a response are recorded. The two recorded sound waves are examined using a FFT spectral analysis algorithm to find pitch and amplitude of the music data. The gain of the microphone used to record the audio information was set to $+6dB$. The size of the FFT window was adjusted to $256samples$ with an overlap of $32samples$. Looking at the resulting graph we have the possibility to analyze the robot's choice of a response. Also the time-relationship of input and output are to be examined. The quality of the response can be characterized by how quickly (time difference between musician's first complete play of the pattern and the robot's response) it is produced and how accurately (compliance of the response timing pattern with the input pattern). Fig. 4.5 shows a saxophone player experimenting with the histogram-based melody tracking system. The right hand side of the image shows a screenshot of the tracking software's user interface during the experiment.

4.3.4 Experiment Results

As can be examined in Fig. 4.6, Pattern 1 consists of one bar of four equally spaced quarter notes. The instrumentalist plays this pattern alternating between the notes c and d at a tempo of 80bpm. After the first pattern has been performed (from $0s$ to $3s$), the robot answers by reproducing the same pattern (in beginner level interaction mode pre-programmed to alternate between c and d). The duration between the beginning of the instrumentalist's question and the robot's answer is approximately $3s$. The robot continues playing the last tone of the rhythm pattern until the next successful recognition. During this time the robot is in an idle state. The purpose of playing the last tone of a pattern continuously, is to give the interacting partner of the robot a feeling of the breathing cycles of the robot. When we look at the volume plot we can identify areas, where the level suddenly drops for a certain duration. These moments are called breathing points and relate to the time when the robot's lung system is deflated and needs to pull air in order to be able to produce the air-beam necessary to generate the flute sound. As the lung breathing speed is constant we see these events regularly happening at $7.5s$ and $14s$ in the graph. The duration of one breathing phase

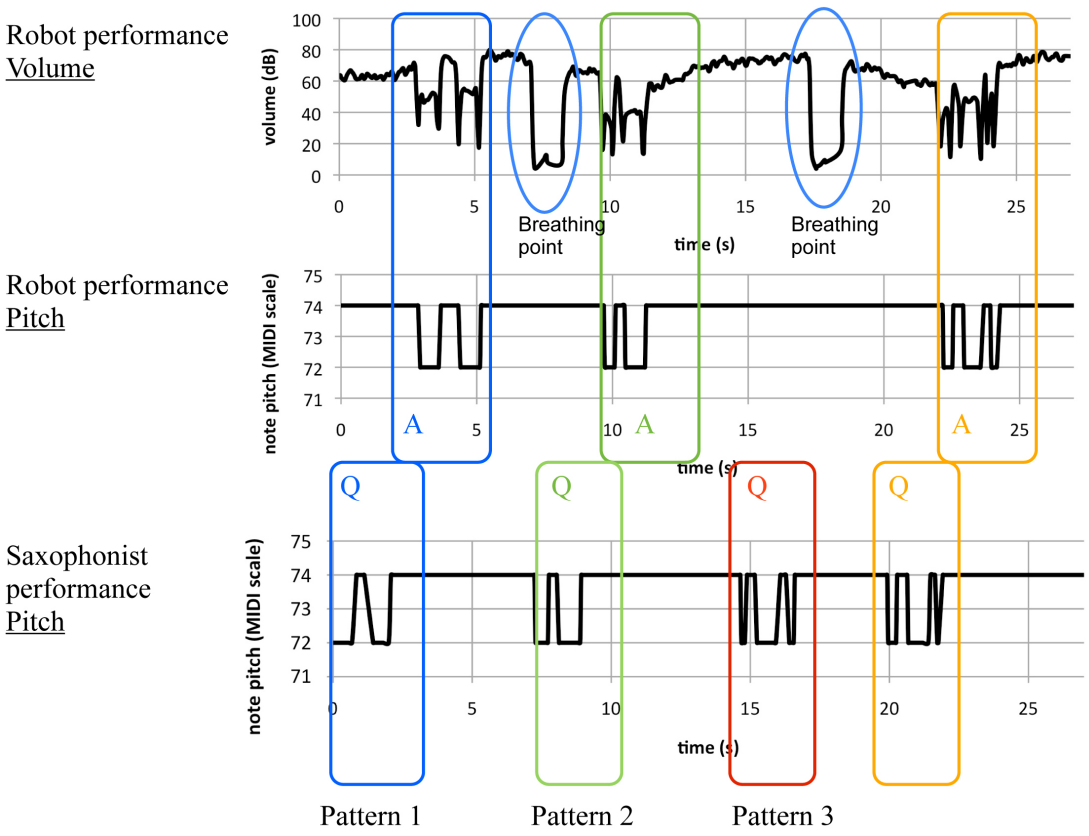

Figure 4.6: Recorded input and output of the advanced level interaction system. The top graph shows the amplitude plot of the flute robot response (A). In the middle the pitch analysis of this response is displayed. The bottom graph details the amplitude plot of the question (Q) by the robot's partner musician. This question acts as the musical target pattern for the flutist robot.

is $\approx 10s$ long. At $7.0s$ the musician starts playing rhythm pattern 2. This pattern is more complicated than the first one, consisting of one bar, containing two eighth notes and two quarter notes. The answer of the robot is observed after the pattern has been finished, at $\approx 9.5s$. The duration between the last tone of the question pattern and the robots answer is $\approx 0.1s$. The third rhythm pattern is the most complicated of the three sequences used in this experiment. Besides a quarter and an eighth note it contains a triplet. The instrumentalists plays it two times, at $14.5s$ and $19.8s$. When played for the second time it is correctly recognized by the aural processing algorithm and the robot gives a response at $24.5s$.

In this section we have introduced the beginner level interaction system audio processing module. With our method we are able to extract note and timing information from the audio input of WF-4RV. This data is compared to rhythm patterns stored in a user generated library. We use a histogram comparison technique to determine the similarity between a recorded patten and a library pattern. In the experiments we tested the detection capabilities of the proposed method. The results show that the algorithm succeeded in matching rhythm pattern played by a human musician, with a set of a rhythm pattern stored in the pattern library of the flutist robot.

4.4 Extended Level Interaction System: Histogram-based Melody Detection

4.4.1 Implementation

Our approach for the aural processing of the advanced level interaction aims to create an interface that extends the amount of freedom a player has in controlling the robot. At the same time the usage of the system becomes more demanding for the human player in terms of skill level. However, it also allows a wider scope of musical expressiveness. This advanced level approach analyzes the pitch components of the recorded sound. In the following we describe the harmonic component analysis. An overview of the structure of the system can be seen in Figure 4.7

Pitch information is recovered from the input data stream by applying a discrete Fast Fourier Transformation (FFT). As we sample sound data in windows of 1024 samples we apply the Hann windowing function (Eq. (4.4)) to smoothen spectral leakage.

$$w_i = a_i 0.5 \left(1 - \cos\left(\frac{2\pi i}{N-1}\right)\right) \tag{4.4}$$

w_i: resulting amplitude for sample i

a_i: input amplitude indexed with i

N: number of samples in the window

Similar to the timing analysis, we apply a running average to adjacent frequency spectra and perform thresholding operations to reduce noise. If the threshold amplitude is retained by one or more peaks of the spectrum for long enough not to be suppressed by the low-pass filter, the peak with the highest amplitude is identified as lead-frequency. A recently registered pitch frequency is approximated by the twelve-tone system note with the closest frequency. The value of this note is queued into the sequence window.

When looking for harmonic information we look into the past only for the number of notes contained in the longest library pattern. The note information in the sequence window is gathered by generating a histogram from the pitch values. Again we match this histogram to the library histogram in order to find the best match. Information regarding which pattern was recognized is then forwarded to the mapping module.

4.4.2 Experiment Objective

We perform the same method of experimentation for evaluating the aural recognition algorithm of the advanced level interaction system as in case of the rhythm recognition. In this case additionally to the three rhythm patterns, three harmony structures are contained in the robot‘s library. During the experiment, the saxophone player performs deliberate combinations of one timing and one harmony pattern. The human musician‘s play and the recognition response of the robot

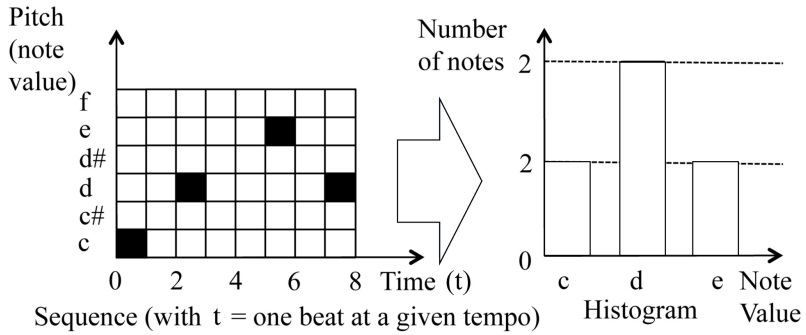

Figure 4.7: Similarly to the rhythm detection method, note pitch values are inserted into the histogram in order to compare currently played pattern and library patterns. In case of this example, there are two notes with a pitch of c, two with a pitch of d and one with a pitch of e.

are recorded and analyzed in the same way as in the previous experiment. We chose all patterns to have the same length of 1 bar. This makes the recognition of the end of one pattern more easy.

The activity in the interaction system regarding the aural processing is measured at two points. Additionally to the microphones that are integrated in the head of the robot, we place a studio microphone in front of the saxophone player. In case saxophonist and flutist robot play simultaneously, with this method we are able to separate both sources. We consider that the music data which is loudest in one recording, originates from the closest sound source. Data from both microphones is recorded in a digital audio workstation (DAW) application. It is analyzed using a utility software programmed in the scripting language MATLAB.

4.4.3 Experiment Method

For the advanced level interaction we used the same setup of microphones as for the previous experiment. Now the robot‘s library contains, not only pure rhythm patterns, but pattern that contain melodic content in addition to the rhythmical information. In the experiment we asked a wind-instrument player (saxophonist), to perform a sequence of these patterns in front of the robot. I try to evaluate the

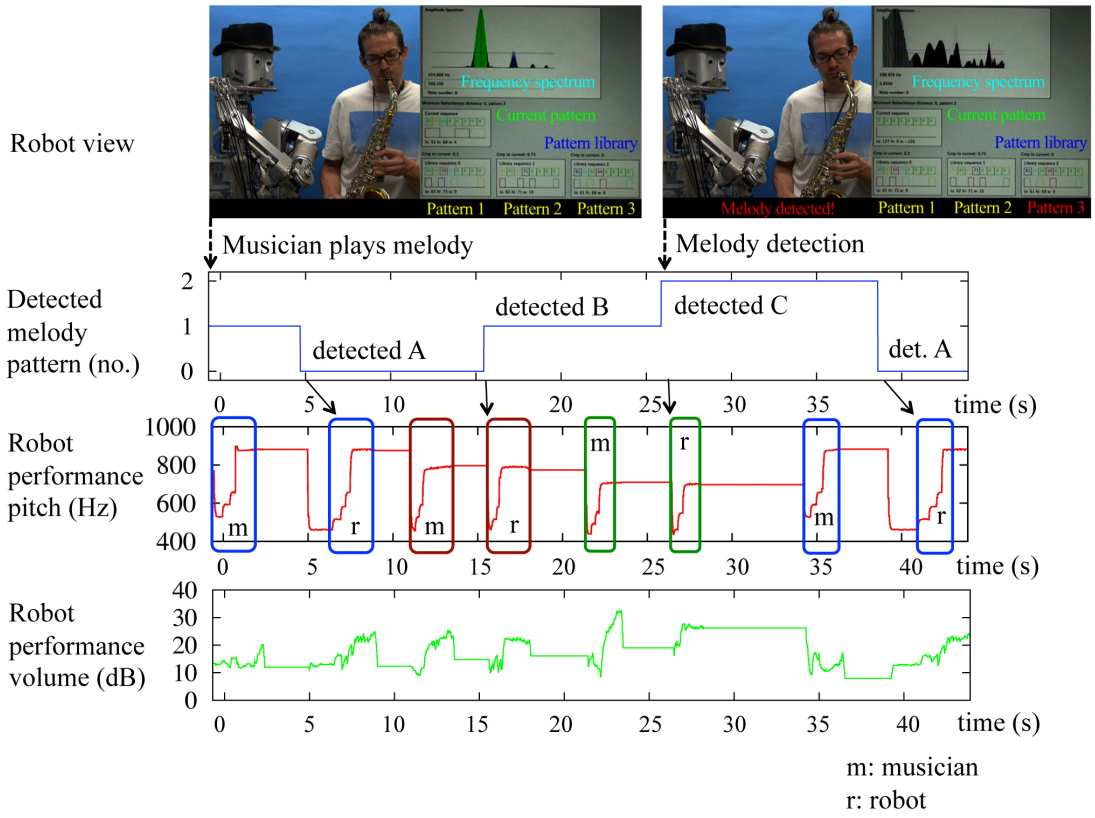

Figure 4.8: The graph shows a sequence of question and answer play between a human musician and the flutist robot. If a pattern m played by the human musician is detected by the melody recognition system, the phrase is repeated by the flutist robot (r). Three patterns A, B and C are performed.

functionality of the melody detection algorithm by analyzing the response of the robot to this musical input. In the experiment I applied a simple direct mapping of detected melody pattern and the output pattern of the robot. If one pattern is detected the robot repeats the same pattern. As evaluation of the experimental result, I compare, if input and output pattern are the same. Also I try to verify accurate detection by checking the index of the detected melody pattern.

4.4.4 Experiment Results

The musician plays the first pattern at $30s$. The robot's answer is recorded at $36s$. This sequence is repeated for the three different patterns at $41s$, $52s$ and $64s$. With the respective answers of the flutist robot at $46s$, $56s$ and $70s$. In the graph displaying the detected melody pattern index, at the indicated times, the detection of melody patterns A, B and C is confirmed. The detection curve changes at the moment, the robot recognizes a new melody pattern, when it has been played by the musician. As the lung status is reset each time one pattern has been played, there are no breathing points visible, as have been visible in case of continuous (performance of a looped pattern) performance.

4.5 Discussion

In this section we have introduced an aural pattern matching algorithm. Using this method we were able to match melody patterns played by a human musician with a set of library patterns. In the experiments the flutist robot WF-4RV verified successful recognition by repeating the pattern that was matched to the musician's play. WF-4RV is therefore able to engage in a simple form of question and answer play with a human musician. In case a pattern fails to pass the similarity threshold level that was described earlier, it will not be recognized. If this threshold is high, the human musician has to be very precise in his performance. If the threshold is low, false positive recognitions can occur. We experimentally chose a threshold that lead to satisfying results in experimental environment. All experiments were performed inside the laboratory. That means, that we used a

very control and quiet environment, with little noise. In a realistic stage environment, there might be noise from the audience, that needed to be taken into account when adjusting the aural tracking system. We documented results for both, the beginner level and the advanced level interaction system. The results show, that, given controlled experimental conditions, our method does lead to the intended outcome.

For our experiments we chose rhythm and melody patterns that are uniquely identifiable by using the histogram technique. If two rhythm pattern contain the same notes (same length) but in a different order, their histograms will be the same. Incorrect recognition by our algorithm might be the result. Regarding harmonic pattern we have in principle the same problem. However considering short sequences in terms of harmony, the order of the notes might not play a very important role. As long as the notes of two patterns with the same histogram are situated in the same scale, the pattern reproduced by the robot might not be the same as the input, but it will harmonically fit. The problem of patterns that have the same histogram might occur more often, if there are more than the so far evaluated three patterns in the library. So far we experimented only with these patterns, because we considered it a sufficient number for a study of the interaction system, that can be performed by non-technical personnel.

4.6 Conclusion of this Chapter

Information from the audio processing stage is the second channel of communication, that I use in the musical-based interaction system. As has been shown in the experiments and described in the discussion section, the system can detect melody and rhythm patterns performed by the partner musician of the robot and compare these to a pattern library saved in the robot. The information about the most fitting pattern is propagated to the mapping section of the MbIS. Here it is fused with information from the other communication channel, the computer vision module, that has been described in the previous chapter. I developed the acoustic processing system specifically for the MbIS, during the preparation of this thesis. Novelty and originality especially lies in the method of comparing melody and rhythm sequences using histograms. Histogram matching is often

used in application to image processing as shown in the previous chapter. I tried to take the approach from image processing, where it is used to characterize the colorization of an object, and use it in audio processing. Here we do not use colors to setup the histogram, but rhythm and melody information. In the experiments of this chapter we evaluated the capability of this method to classify rhythm and melody patterns. For each case (rhythm and melody) we used a library of three patterns. In our experiments, the recognition technique showed a detection rate of 100% in case the musician played the pattern exactly as had been recorded in the library of the robot. In the rhythm tracking experiment an example of a pattern that was not correctly detected can be observed.

Chapter 5

Mapping Strategy Implementation

5.1 Introduction

So far we have mainly focused on the visual and acoustic sensor processing. In this chapter we want to concentrate specifically on the last stage of the robot's performance system for translating sensor data into musically meaningful performance modulation parameters (in the following, this part of the system will be referred to as the mapping module). As an example, instrument movements as shown in Fig. 2.9 are to be analyzed by the sensor processing system and then translated into musical output by the mapping module. The mapping module of the proposed MbIS consists of two levels, one stage for interacting with players of a beginner skill level and one stage for more advanced players.

In the basic level interaction stage we focus on enabling a user who does not have much experience in communicating with the robot to understand about the device's physical limitations. We use a simple visual controller that has a fixed correlation regarding which performance parameter of the robot it modulates, in order to make this level suitable for beginner players. The WF-4RV is built with the intention of emulating the parts of the human body that are necessary to play the flute. Therefore it has artificial lungs with a limited volume. Also other sound modulation parameters like the vibrato frequency (generated by an artificial vocal chord) have a certain dynamic range in which they operate. To

account for these characteristics the user's input to the robot via the sensor system has to be modified in a way that it does not violate the physical limits of the robot. To modulate the robot's performance parameters we use a motion tracking algorithm to detect a partner musician's instrument movements. For this purpose we introduced the previously describe, specialized virtual fader and virtual button controllers.

In the extended level interaction interface, our goal is to give the user the possibility to interact with the robot more freely (compared to the beginner level). To achieve this we propose a teaching system that allows the user to link instrument gestures with musical patterns. Here, the correlation of sensor input to sensor output is not fixed. Furthermore, we allow for more degrees-of-freedom in the instrument movements of the user. As a result this level is more suitable for advanced level players. We use the robot's instrument gesture detection system that we have introduced previously. A Bayesian mapping algorithm is employed in order to ensure, that if the teaching musician does not account for all combinations of instrument orientation and musical output in the teaching phase, in the performance phase the robot will automatically play the most closely matching answer modulation to a given instrument state.

Our new approach brings two significant novelties compared to how we have dealt with sensor input in previous work. First, we now work with the acquired sensor data conditionally by getting feedback from the body of the robot (e.g. state of the lung) and using this feedback to modulate the influence of the user interaction on the performance parameters. Second, in the advanced interaction level, we allow the user to teach the robot how one or more sensor values modify one or more performance parameter values.

5.2 Related Research

Learning-teaching techniques similar to the method proposed in this thesis have, in various ways, been introduced in robot control. In [69] oral expression and sensory inputs are mapped to control the motor of a robotic arm. The approach uses information gained from camera images and microphone input to set up Hidden Markov Models (HMMs). These models contain a state-space representation

of the generated training data. This information is used to execute tasks that consist of combinations of the different teaching situations.

In [70], a similar approach by grouping sensor inputs (vision, sound, force) into clusters has been proposed. These clusters are mapped to objects, and over time build an object state history. From the information contained in this history an action-space is created. This action-space is used as a look-up table to determine a suitable action given a certain object situation.

Actions in a game taught by human demonstrators are used in [71] to attain knowledge for human-robot interaction. From this information the introduced robotic system creates a Bayesian network. Recorded patterns are clustered and represented in a graph. From this pattern graph the robot is able to choose which action applies to which game situation and is therefore able to actively play a game together with humans.

Also in navigation related publications ([72]) particle filter / HMM-based methods are utilized to reliably determine object and action-space. Especially if a robot uses more than one sensor (more than one camera, laser-scanners, audio-visual input, etc.) this method can greatly reduce the effort to find the set of data that well represents the current state of a robot's environment.

Specifically, the approach presented in this thesis is adapted from a problem setting, in which a robotic arm is taught how to empty a glass of water into a sink ([73]). The robot learns this movement by demonstration, taking the glass from a specific location. The goal is that, with changing initial location, the robot autonomously is able to find the right way to empty the glass without assistance. Similar to the other approaches referenced above this application uses a state-space table to record the instructions during the teaching phase. During the execution, sequential Bayesian filtering is employed to adapt the learnt data to the current problem environment.

5.3 Basic Level Interaction System: Direct Mapping

5.3.1 Implementation

The purpose of the basic level interaction system mapping module is to translate the actions of the user that are recorded through the virtual buttons and faders into musical output (Figure 5.1). This output is to make musical sense in the way that the user can express himself as freely as possible, while at the same time respecting the physical limitations of the robot. One important limitation of the WF-4RV flutist robot is the restricted air volume that can be contained by the lung. Similar to the human breathing the robot is only able to produce sound for a certain duration, until the lung is empty. The robot has also further limitations, like a maximum playing speed and maximum modulation speed performance parameters like the vibrato frequency.

When receiving data about the robot's partner musician from the vision processing system, we can map this data directly onto a musical performance parameter. In case of receiving a continuous value from a virtual fader controller, this relationship can be formulated as shown below:

$$A(t) = k * I(t) \tag{5.1}$$

This equation contains the constant k representing a scaling factor to resize the sensor (virtual fader) value $I(t)$ to an appropriate output value A. Using information about the maximum and minimum value emitted from this controller, we can condition k accordingly, so that the output value A does not exceed the acceptable range for the performance parameter.

$$A(t) = k(t) * I(t) \tag{5.2}$$

with

$$k(t) = \begin{cases} k & \text{if } t < T_{Breathing} \\ 0 & \text{if } t \geq T_{Breathing} \end{cases} \tag{5.3}$$

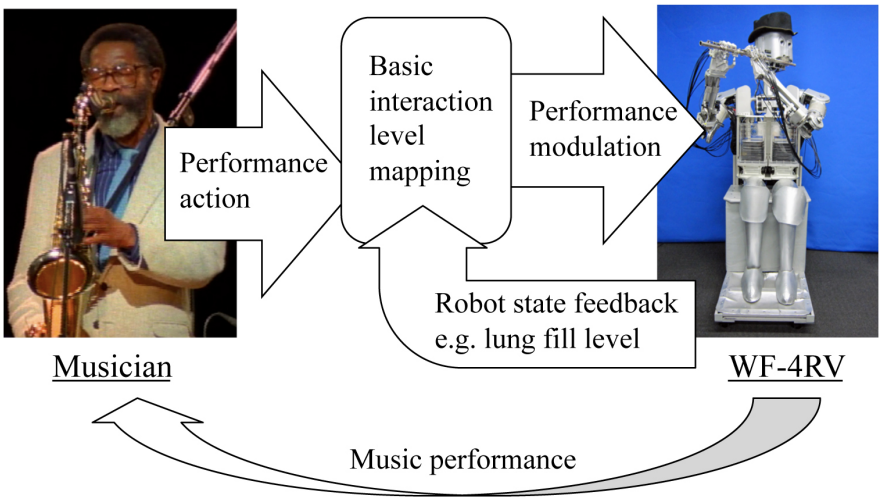

Figure 5.1: The figure shows the functional principle of the basic interaction level mapping system. The module has two input channels: Information about the musician's instrument movements (performance action) and feedback information about the robot's physical state. A performance action of the musician is mapped into performance modulation. The mapping itself is modulated by state feedback from the flutist robot. In the presented application, the lung fill level of the robot regulates the value of the virtual fader. The fader value is faded-out in case of a lung fill level above 80%.

Some limitations of the robot however are not time-constant. The capability of the robot to create an air-beam in order to play the flute, depends on the air volume left in the lung. Taking this into account we add time-dependence to k.

$T_{Breathing}$ indicates the time duration of air remaining in the lung, enabling a tone to be be produced. The equation expresses that the intended output of the flute robot is to be conditioned with the fill-status of the lung. If the lung becomes empty, the equation constant $k(t)$ is set to 0, resulting in $A(t)$ to become 0 as well. As result the flute robot does no produce sound output.

For this reason it is necessary to include the time-factor t into Eq. (5.2). During a musical performance, there are several physical characteristics of the flutist robot that are continuously changing. We need to condition $k(t)$ in a way, that it accounts for the time dependent limitations of the robot, such as previously described, the lung. One option is to simply switch $k(t)$ to 0, when the lung reaches the volume limit and needs to be refilled.

5.3.2 Robot Constraint Restricted Mapping

The flute robot is a complex mechanical construction, built in order to emulate the human way of playing the flute. It thus naturally bears similar limitations to a human. It is important that when controlling or interacting with the flutist robot the user does not drive the robot into these limitations, risking physical damage of the machine and leading to unnatural performance behavior. This is achieved by the implementation of a Constraint Restricted Mapping (CRM) approach.

As the switch from a normal performance to a breathing break is very abrupt, this method might not be satisfactory in a musical sense. Musical progressions are normally characterized by smooth transitions or intentionally inserted breaks at certain points. If a musician needs to interrupt his performance as a result of physical constraints, that would, under normal circumstances, give the impression of an unsatisfactory presentation to the audience. As every human has various bodily constraints, these need to be integrated in the mode of performance in a way, that is as little as possible perceivable by the audience. To address this issue in a natural way we applied a method to provide smoother outline to the

switching edges. Using a digital low-pass filter on the time-dependent conditioning as indicated in equation (5.2), we get smoother outlines for the switching of the performance states (normal play / interrupted play due to lung-refill).

The modulation envelope that results from this method of smoothing is similar to Attack-Decay-Sustain-Release (ADSR, [28]) curves used in electronic music synthesizers. If we vary the parameters of the low-pass filter we can change the slope of the attack curve. This enables us to adjust the smoothing in a more human-like fashion. We implemented the digital low-pass filter as a Finite Impulse Response (FIR, [74]) filter. This filter achieves a similar effect to value-averaging by chaining delay stages and scaling stages. Considering a queue of values (in this case the time-dependent values of $k(t)$), fractions of previous values are fed-back to the current value and added. Depending on the number of these delay-elements the low-pass / averaging effect becomes stronger.

Through the implementation of this method in the mapping module of the beginner level interaction system we can guarantee that the robot's partner musician can control the robot safely (within the robots value constraints). Using the time-dependent scaling parameter and the smoothing filter, the robot will automatically adjust the sensor input from the vision system to account for the system's mechanical properties (Figure 5.2).

Although the above principle was so far introduced using a virtual fader as controller source, it also applies to the adjustment of data from a virtual button. In an interaction setting using virtual buttons, the user might be able to switch between various melody patterns. If during the performance of one melody pattern the volume drops due to the necessity of a lung-refill, this might seem unnatural to the audience. A resolution to this problem is to calculate, if the melody pattern to be played fits into the time remaining until the next lung-refill; and in case it does not apply the proposed fade-out effect to the sequence tempo (slow the sequence down) using the low-pass filter to smoothly finish the last note of the pattern still fitting into the lung cycle (Equation (5.4), (5.5)).

$$A(t) = k(t) * I(t) \tag{5.4}$$

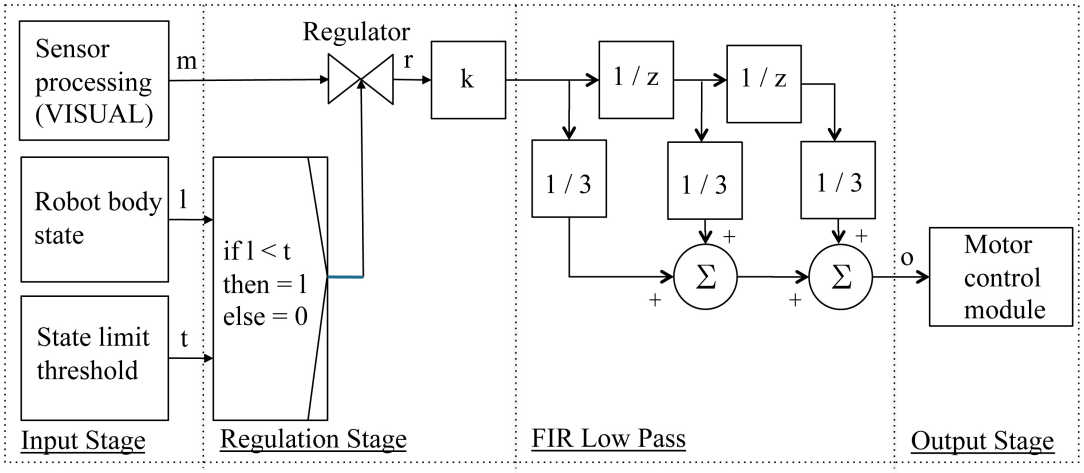

Figure 5.2: Block diagram of the beginner level mapping method. m denotes the movement value output from the sensor processing module, l the detected lung fill level (Robot body state), t the fill level threshold, c the fill level control value, r the movement data regulated by the fill level controller and o the filtered musical parameter modulation output.

with

$$k(t) = \begin{cases} fadeout & \text{IF sufficient residual air volume} \\ play & \text{IF residual air volume not sufficient} \end{cases} \tag{5.5}$$

Here (Equation (5.4), (5.5)), we apply the same type of filter as described in order to fade-out a note in case of the virtual fader scenario. As the musical material is very similar we considered that the filter can be used in the same way. The calculation of the length of the musical pattern to be played needs to be taken into special consideration. Depending on the tempo of the song adding the note values / breaks of one pattern together will result in the pattern length. However, in case the tempo can be manipulated by the robot's partner musician, we need to take into account the range of adjustment that is possible and add this amount to the pattern length as a tolerance margin.

5.4 Extended Level Interaction System: Bayesian Filtering-based Mapping

The mapping approach for the advanced level interaction system, is based on the assumption that the partner musician of the robot is a player of advanced skill level. As a result the method leaves more space for free control of the robot. The goal of our approach is to implement this technique into our musical interaction setup to create sensible musical output. In contrast to the previous approach, this time we do not use the virtual faders and buttons as input source, but the particle filter-based tracker. In two phases, the teaching phase and the performance phase we try to enable the robot to estimate the song state according to the input received from the vision and audio processing system. If the robot knows the current state of the song, it will be able to play an appropriate reaction to the human partner musician's actions. A deliberate number of input parameters (e.g. horizontal and vertical instrument orientation), is to be mapped to a deliberate number of output parameters (e.g. vibrato amplitude, played note value). This should be done without the teaching musician having to account for all possible state combinations. Using a particle filter, even if during the performance an unknown state combination is given to the robot, it is to automatically play the most closely matching answer modulation. The principle of this extended level interaction is shown in Fig. 5.3.

At first, in the teaching phase, the teacher fills up the state-space table with information on how to relate instrument orientation changes to performance modulation. Although the instrumentalist may spend a long time teaching, this information will probably not be complete. That means that there are states of the instrument configuration that are not accounted for in the table.

In the performance phase, the robot reacts to the movements of the musician in order to reproduce the previously learnt behavior. To relate a configuration of the instrument (orientation) to a correct modulation, the robot uses a particle filter (Bayesian filter). In Fig. 5.4, the robot takes the data from the vision processing, seeds particles in the state-space table (e.g. in the button states and fader states column) and selects the most closely related particle. The modulation that this particle relates to (in the table), is played by the flutist robot.

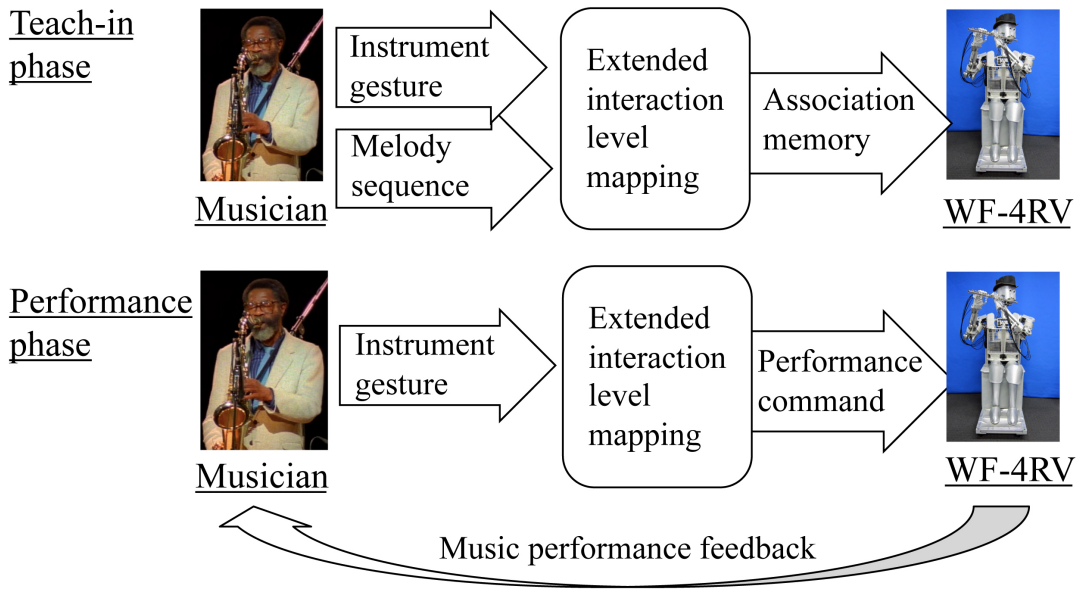

Figure 5.3: The figure shows the functional principle of the extended interaction level. In a teach-in phase the musician creates an association between an instrument gesture and a melody sequence. As input the orientation of the instrument of the musician and the melody performed by the musician is processed. The output is a state-space representation, that in the performance phase is used to map the input of instrument gestures by the musician to performance output commands to the robot.

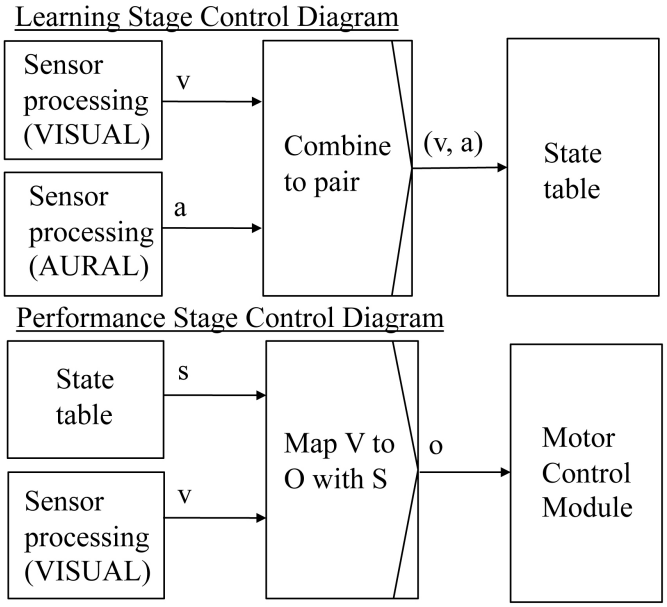

Figure 5.4: Block diagram of the advanced level mapping method. a) shows the teach-in phase signal flow. I denotes the detected instrument motion, N the detected note or rhythm sequence. b) displays the signal flow in the performance phase. Additionally S denotes a state from the state table here (could be also expressed as an (I, N) tuple).

A Bayesian filter represents the PDF $p(\underline{x}_k|\underline{z}_{k-1})$ of state $\underline{x}_k$ given observation $\underline{z}_{k-1}$ were k is the discretized time. Specifically for the particle filter this PDF is defined through a set of N_s random measurements $\underline{s}_k^i$ with weights π_k^i. In this case, the current observation $\underline{X}_k$ is given by

$$\underline{X}_k = \sum_{i=1}^{N_s} \pi_k^i \underline{s}_k^i \tag{5.6}$$

and the PDF $p(\underline{x}_k|\underline{z}_{k-1})$ can be approximated as ([47])

$$p(\underline{x}_k|\underline{z}_{k-1}) \approx \sum_{i=1}^{N_s} \pi_k^i \delta(\underline{x}_k - \underline{s}_k^i) \tag{5.7}$$

with

$$\sum_{i=1}^{N_s} \pi_k^i = 1 \tag{5.8}$$

δ denotes the Dirac delta function. $\delta(\underline{x}_k - \underline{x}_k^i)$ represents the deterministic relationship between the random samples and the actual state. Both of the above equations show, how we gather random samples, assign them with weights and construct an approximation of our actual state. The random samples (or particles) are compared to the input gathered from the sensor processing modules. We use this information to find the closest match of the state-space entries and the sensor values. We restrict the algorithm to a limited set of samples N_s to keep the computational effort manageable / real-time.

5.5 Synchronization of Robot and Human Performance by Onset Control

Additionally to fluent interaction during the performance, based on the basic and extended level interaction system, a synchronized start and end of the performance is very important. As an extension of the previously introduced interaction system components I used two virtual buttons (using the motion perception-based tracking) to give the performer the possibility to control the onset of the flutist robot performance. A screenshot of a saxophonist using this setup is shown in

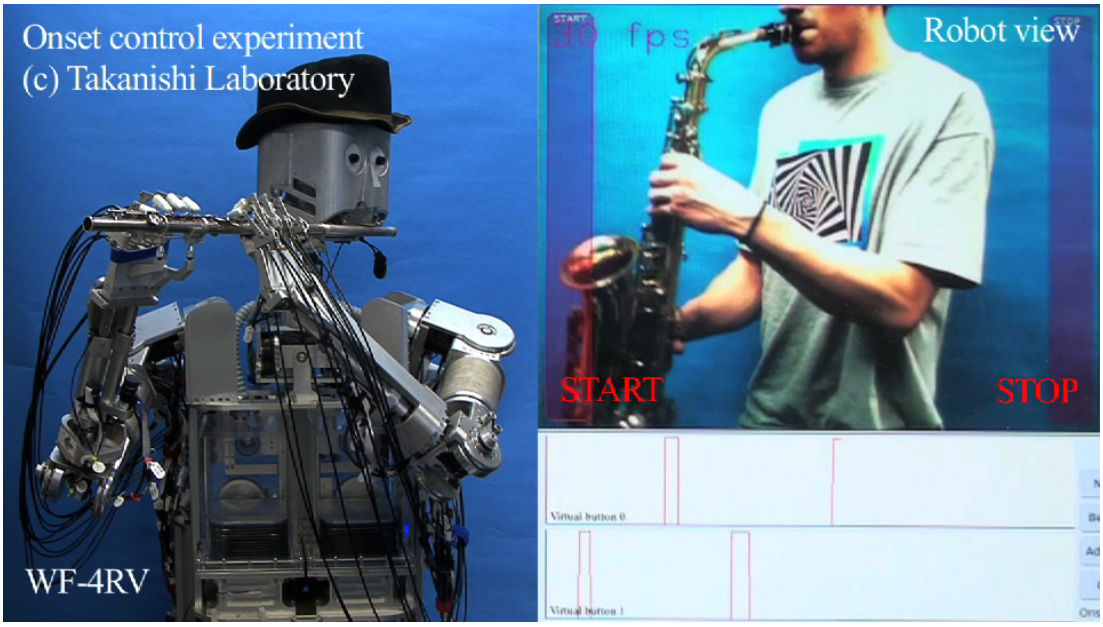

Figure 5.5: In this screenshot, the setup of the onset experiment is shown.

Fig. 5.5. Fig. 5.6 shows the MbIS, extended with the onset control system module.

I evaluated the functionality of the onset control system experimentally. An amateur saxophonist was asked to use the system to start and stop the performance of a phrase from *Ave Verum Corpus*. *Ave Verum Corpus* is a classical composition by Wolfgang Amadeus Mozart. The piece was suggested as an exemplary score for the experiment by a professional flutist, who I collaborated with for this research. A graph of these results is shown in Fig. 5.7. The graph shows triggering of the *virtual start button* at $24s$, $28s$, $31s$, $33.5s$ and $38s$. The corresponding stop button triggers can be observed at $26.5s$, $27.5s$, $32.5s$, $36s$ and $40.5s$. Between the triggering of the start button and the actual start of the performance of the flutist robot, a delay of $0.1s$ can be observed.

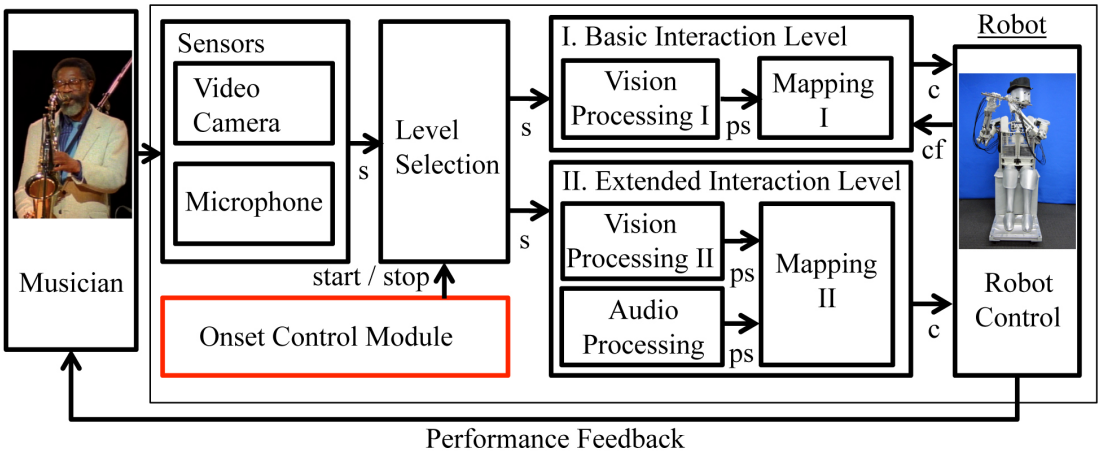

Figure 5.6: This figure shows the MbIS extended with the onset control module. It has been integrated as module that provides start and stop cues to the level selection module. As with the level selection module the user can select an interaction level according to his amount of interaction experience, he can with the onset module, control the flow of the performance in an additional way.

5.6 Experimental Evaluation

5.6.1 Experiment Purpose

In the following section I propose experiments to evaluate the functionality of the basic and the extended level mapping module. The task of the mapping module is to relate input from the sensor processing modules (vision processing and audio processing) to musical performance output played by the robot. The mapping module has two different interaction levels to accommodate for performance with players with different experience levels.

In the proposed experiments, I intend to evaluate the mapping module as such, without any data from the vision or audio processing modules. Data from the sensor analysis modules might also be used for testing. However in this case, human input is necessary to generate test data. Human input is in this case not suitable, as a human is not able to produce exactly the same movements for every experimental run. As we would like to run experiment series with

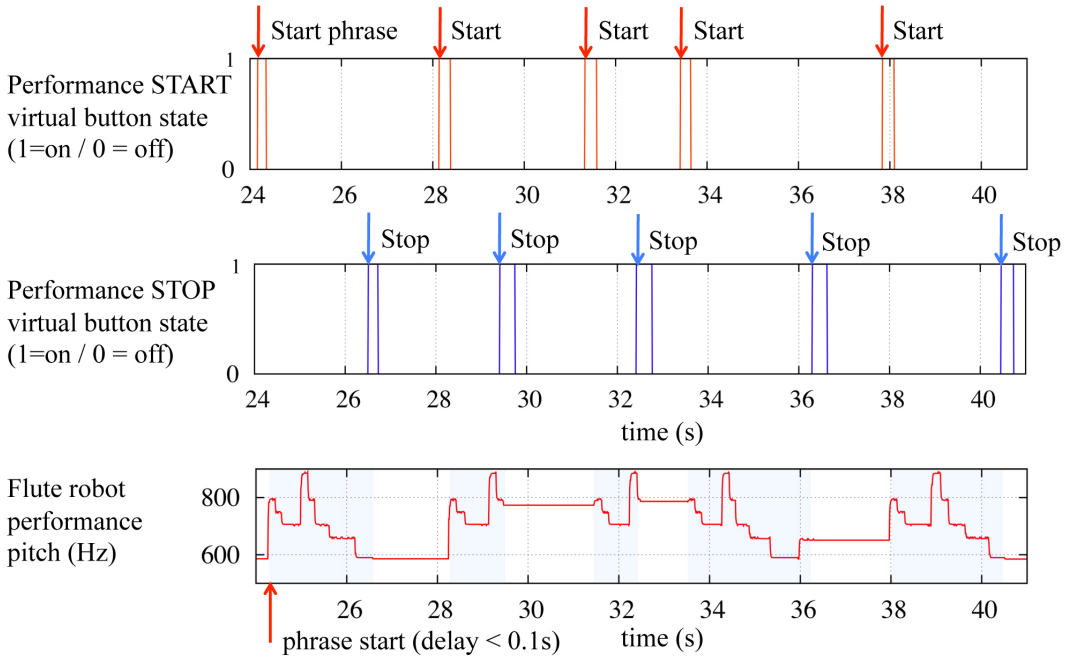

Figure 5.7: The result of the onset control experiment is shown in this graph. The graph on the top of the figure shows the state of the virtual button to control the start of the performance. If this button is triggered, the flute robot performance is started. The *Performance stop virtual button state* graph shows the state of the stop button to stop the performance. The is a delay of $0.1s$ between the start trigger and the start of the performance.

reproducible data, I generate simulated input for the mapping module. In the experiments I would like to test if a given input to the mapping module results in the output, that I intended, when I developed the module. In case of the basic interaction level mapping module this means, that I would like instrument motion data to be modulated by a physical state parameter of the robot. The result of this calculation is to be mapped to parameter of the robot performance. As an example, input from a virtual fader (as proposed in the vision processing chapter) might be modulated by the lung fill level of the flutist robot. This modulated data might then control the tempo of a musical pattern performed by WF-4RV.

Different human users provide very different input to the mapping module. So, in the experiment, input wave-forms to the module were varied to find out if the module works appropriately for different input scenarios. Fade-out curves were exchanged to evaluate their effect on the performance output.

In case of the extended level interaction module, the teach-in and the performance phase are to be evaluated. Both these phases are tested separately in two experimental runs. In the performance phase the association table data from the teach-in phase is used. In the experiments I try to confirm, that the instrument orientation - melody pattern association, that was generated in the teach-in phase is accurately reproduced in the performance phase. As different users might assign different combinations of instrument orientation and melody pattern, in the different experiments, such varying combinations were applied to the system.

5.6.2 Experimental Conditions

To evaluate the basic interaction level mapping module experiment, experiment runs with different input wave-forms and varying fade-out curves were performed (Figure 5.8). I used two types of input wave-forms: Sinusoidal input waves and rectangular waveforms to represent different types of human movement. These waveforms were applied to the mapping module at three different frequencies: 0.1 Hz, 1 Hz and 10 Hz. The simulated input was to be modulated inside the mapping modules by the lung fill level state of the flutist robot. This data was then output to the motor control of the robot in order to modulate the performance tempo.

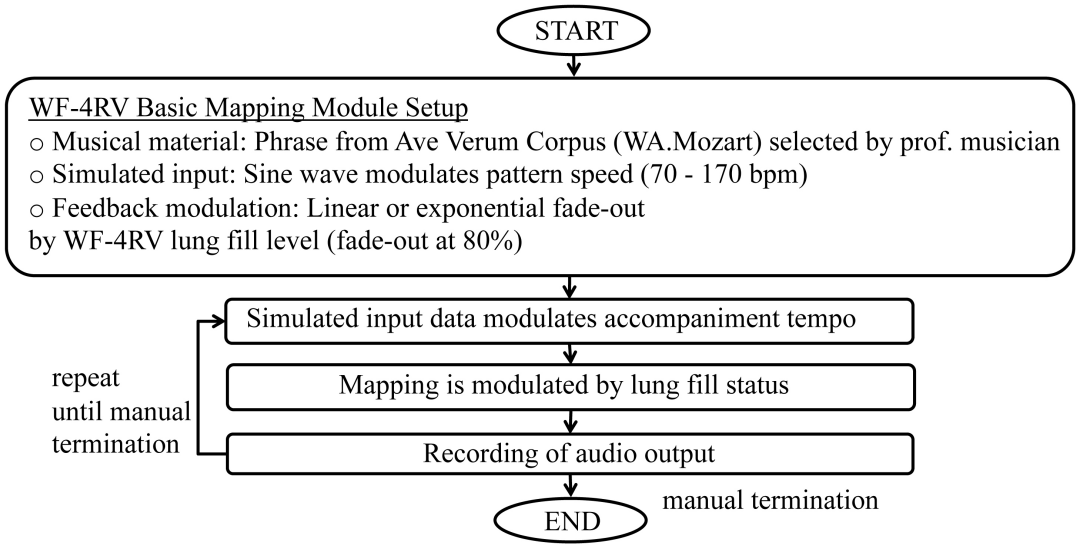

Figure 5.8: The flow-chart shows the experimental method, that was used to evaluate the basic level of interaction mapping module. Input to the mapping module are different types of waveforms (sinusoidal, rectangular) with different frequencies (0.1 Hz, 1 Hz, 10 Hz). The feedback from the lung fill status of the lung modulates the input signal with a linear or exponential fade-out curve.

As exemplary musical score an excerpt from the piece *Ave Verum Corpus* by Wolfgang Amadeus Mozart was used. The tempo of the phrase was modulated in a range of 70 bpm to 170 bpm. To realize the fade-out, two different fade-out curves were applied. An exponential curve and a linear fade-out curve.

To evaluate the extended level interaction system, varying instrument orientation data was generated by a simulated input sine wave of 0.2 Hz (Figure 5.9). The range of a typical instrument movement of a human player varies between -90 and $+90°$. To emulate this movement range, the sine-wave was programmed to oscillate within this range. In the teach-in phase a user manually assigned one of 3 different melody patterns to an instrument position. Two experiments with different assignments were performed. In the performance phase, the same simulated input is used to trigger the different patterns, by changing instrument orientation. As musical evaluation material 3 different phrases A, B and C from the previously mentioned *Ave Verum Corpus* were used.

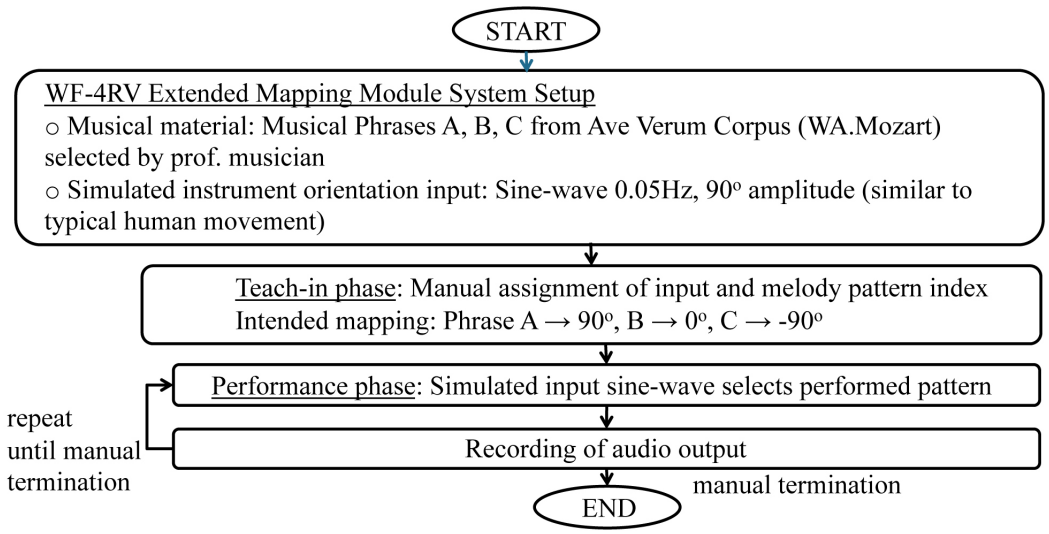

Figure 5.9: The experimental method to evaluate the extended level of interaction is shown here. Three musical phrases from the piece Ave Verum Corpus by W. A. Mozart are mapped to different instrument positions that are simulated using a 0.05 Hz sine-wave as an input. In the performance phase, such a sine-wave is used to simulate input from the vision processing to trigger the previously taught-in musical phrases.

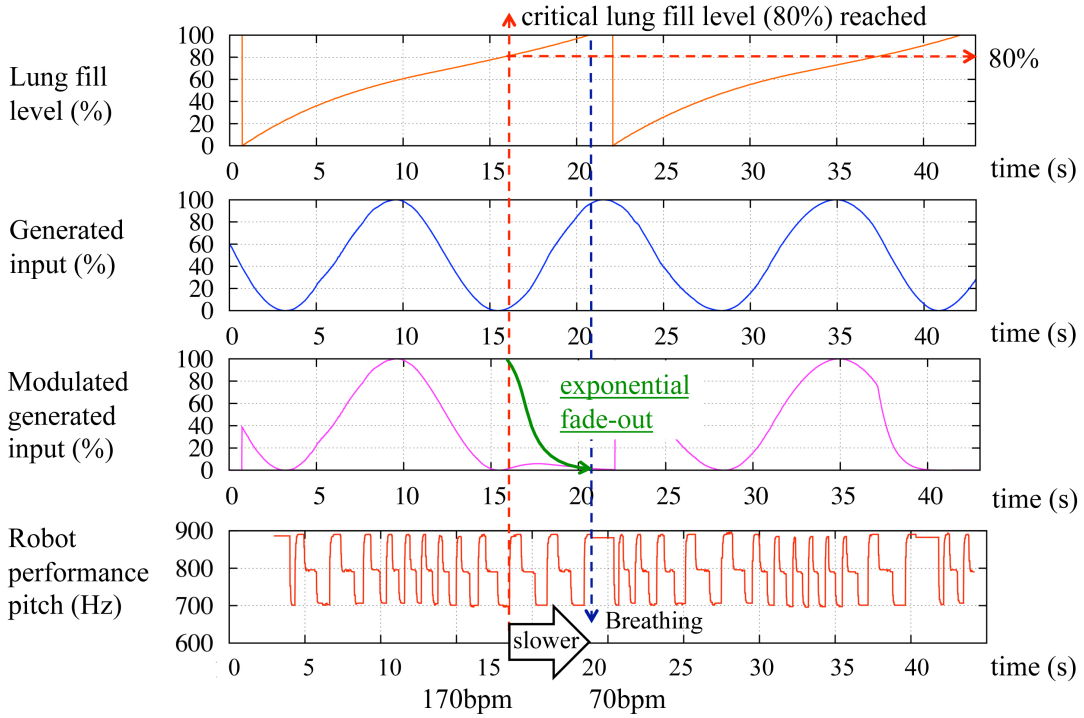

Figure 5.10: Basic interaction level mapping module experiment result graph with sinusoidal input wave (0.1 Hz) and an exponential fade-out curve $f(t) = input * (k * 1/e^(T-t))$ ($input$: simulated module input, k: constant, $T-t$: time from air consumption limit 80%).

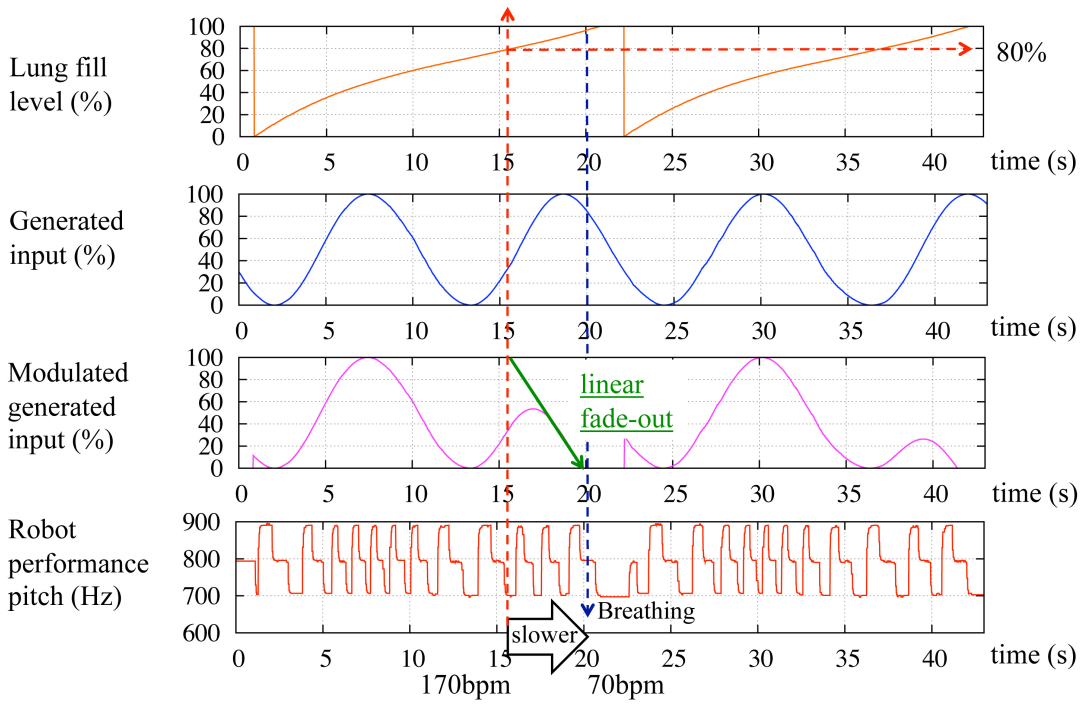

Figure 5.11: Basic interaction level mapping module experiment result graph sinusoidal input wave (0.1 Hz) and linear fade-out curve $f(t) = input*(k*(T-t))$ (*input*: simulated module input, k: constant, $T-t$: time from air consumption limit 80%).

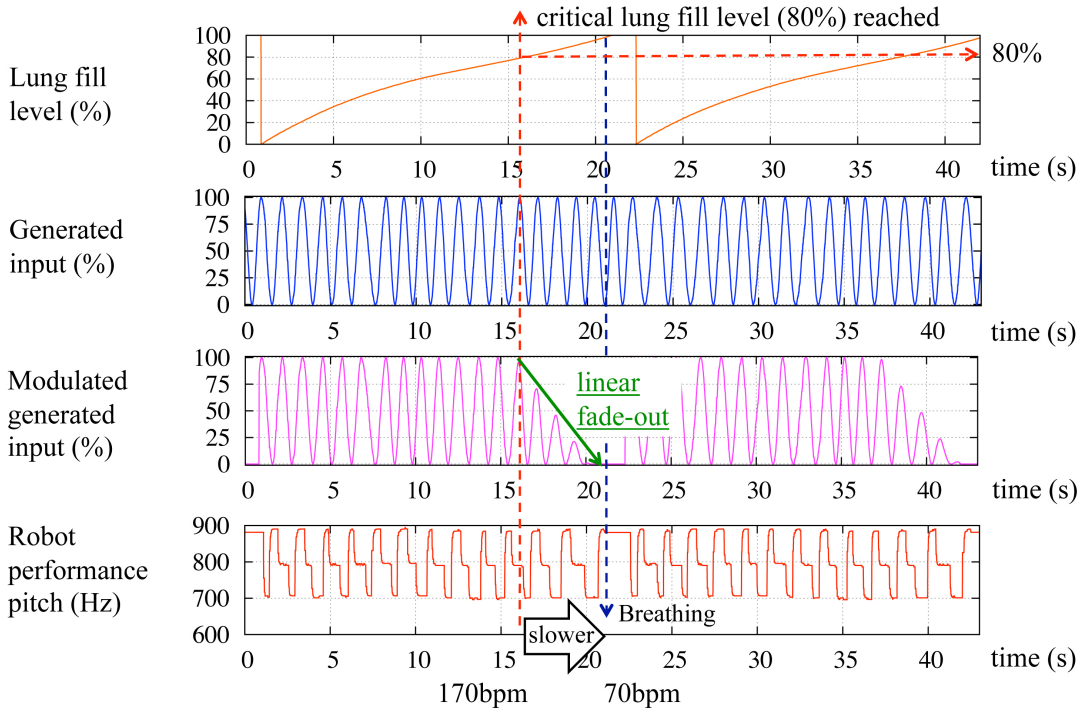

Figure 5.12: Basic interaction level mapping module experiment result graph sinusoidal input wave (1.0 Hz) and linear fade-out curve $f(t) = input * (k * (T - t))$ ($input$: simulated module input, k: constant, $T - t$: time from air consumption limit 80%).

5.6.3 Experimental Results

In this section I describe the results of the previously explained experiment runs. The results for the basic level interaction system are shown in Fig. 5.10 to Fig. 5.12. I show only the results for three selected experiments, that I consider to be representative for the overall result. In all graphs the results for two breathing cycles of the robot are shown. In the first experiment (Fig. 5.10) a sine-wave of frequency $0.1Hz$ was applied as input to the basic interaction level mapping module. An exponential curve was chosen as fade-out characteristic. The lung fill level limit was set to 80%. The result shows direct control of the robot performance tempo from $0s$ to $15s$ and $22s$ to $36s$. Afterwards ($15s$ to $21s$ and $36s$ to $41s$) input from the virtual fader is modulated by the lung fill level, which results into an exponential fade-out of the robot performance tempo. In the following experiment (Fig. 5.11) an input sine-wave with the same frequency was chosen, but this time with a linear fade-out curve. From $0s$ to $15s$ and $22s$ to $41s$ the virtual fader input signal is directly translated into tempo modulation of the robot performance. The linear fadeout, that is controlled by the lung fill-level takes place from $15s$ to $20s$ and $36s$ to $41s$ (lung fill level limit: 80%). Breathing points can be observed at $20s$ and $40s$. In the third experiment (Fig. 5.12), that I would like to describe in detail, the input sine-wave frequency has been set to $1Hz$. This frequency is to emulate faster instrument movements, that might be performed by a human musician. The unmodulated control of the performance tempo can be observed in the graph from $1s$ to $15s$ and $22.5s$ to $36s$. The linear fade-out takes place from $15s$ to $21s$ and from $36s$ to $40s$. There are breaks in the breathing cycle to re-fill the lung from $0s$ to $1s$ from $21s$ to $22s$. As the oscillation of the input sine-wave is quite fast compared to the performance tempo of the phrase played by the robot ($70bpm$ to $170bpm$), in the pitch graph, variations of the performance tempo cannot clearly be seen.

The results for the extended interaction level mapping module experiments are shown in Fig. 5.13 to Fig. 5.16. For each teach-in phase, three association steps, involving three instrument gestures / positions and three manually triggered melody patterns are shown. To simulate reproducible user input, an input sine-wave of 0.05 Hz was used. Fig. 5.13 shows the result for the teach-in phase

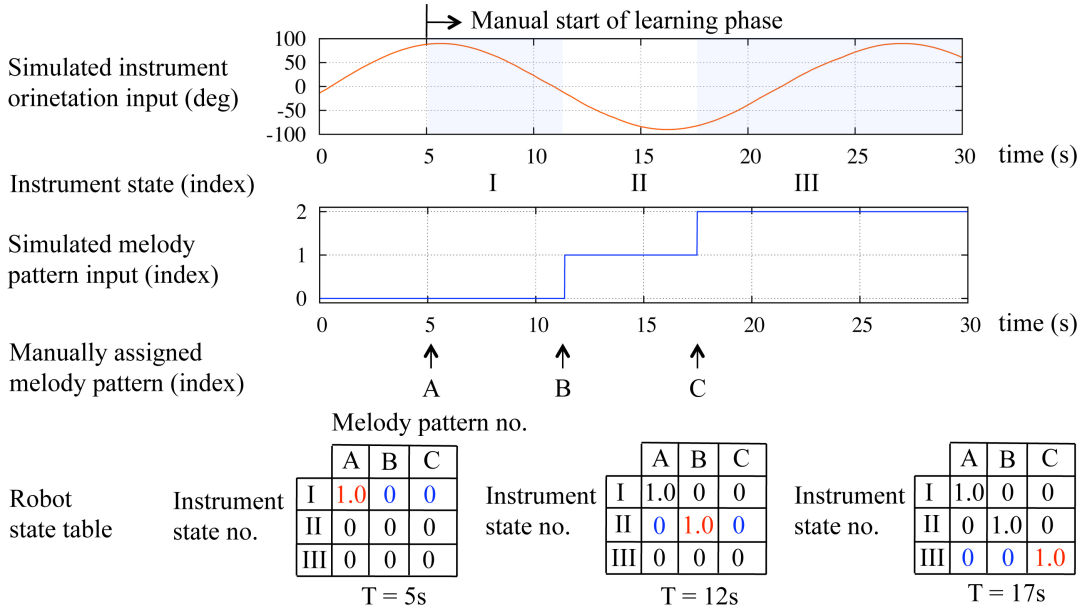

Figure 5.13: Extended interaction level mapping module teach-in phase experiment 1.

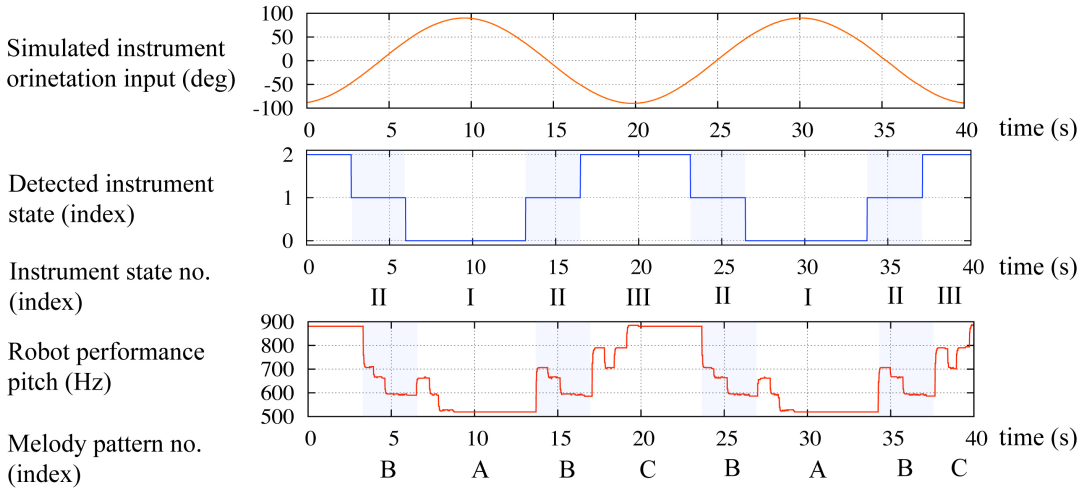

Figure 5.14: Extended interaction level mapping module performance phase experiment 1.

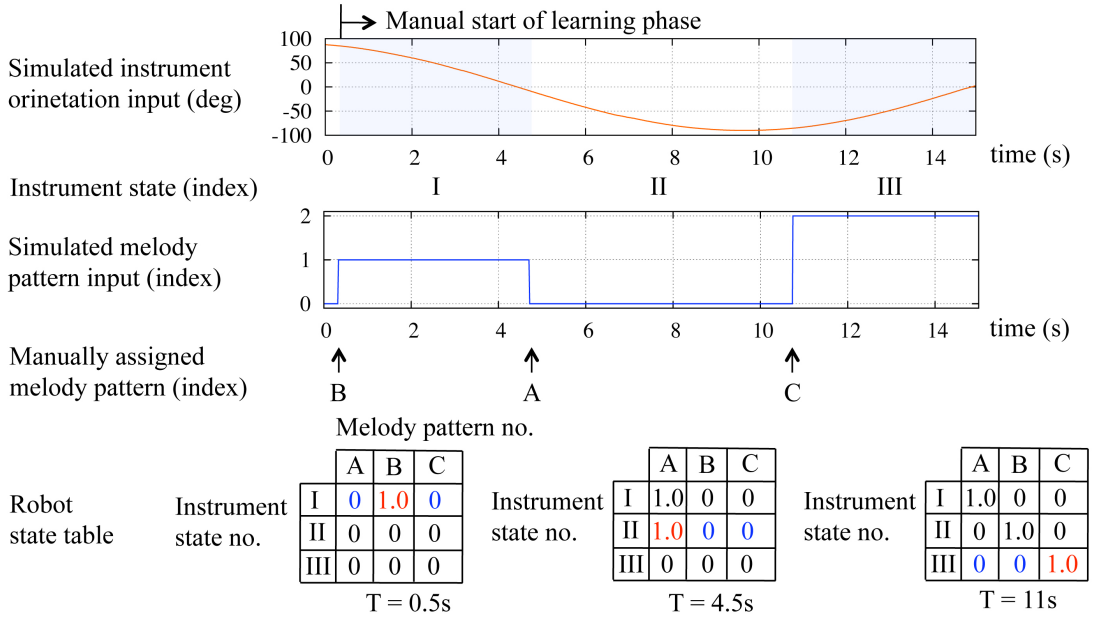

Figure 5.15: Extended interaction level mapping module teach-in phase experiment 2.

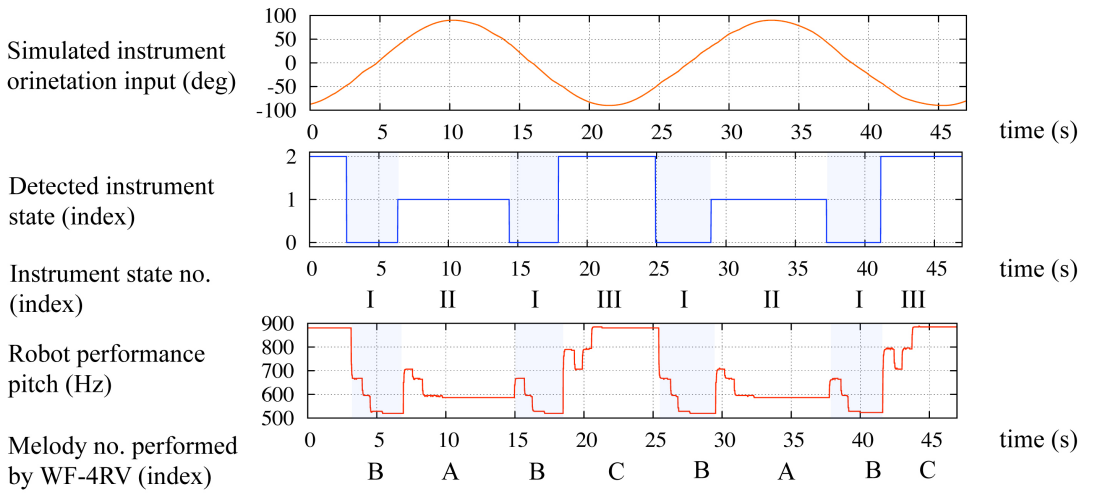

Figure 5.16: Extended interaction level mapping module performance phase experiment 2.

of the first experiment for the extended level. The teach-in phase is started by a manual melody trigger input from $5s$. An instrument angle of approx. $90°$ was associated to melody pattern A. At $12s$ the next melody pattern B was manually triggered and associated to an instrument angle of $0°$. The last melody pattern C was assigned to an instrument orientation of $-90°$ at $17s$. This association information was saved in an association matrix as shown at the bottom of the graph. In the performance phase as shown in Fig. 5.14, the experiment is evaluated for two cycles of the input orientation sine-wave. Each time the input orientation value moves to a new instrument state, the associated melody pattern is performed by the flutist robot. At $2.5s$ the robot detects instrument state II and therefore plays melody pattern B, that was associated in the previous teach-in phase. At $6s$, instrument state I is detected and melody pattern B is played. At $17s$, instrument state III is detected and melody pattern C is performed. The graph shows a regular sequence of the performed melody patterns with the progression of the input sine-wave.

In Fig. 5.15, another example of a teach-in phase is shown. In this result, a different assignment between instrument orientation and melody pattern is shown. At $0.5s$, melody pattern B is associated with an instrument orientation angle of $90°$. At $4.5s$, an angle of $0°$ is assigned with melody pattern A, and at $11s$, $-90°$ are associated with melody pattern C. The resulting association matrix is again shown at the bottom of the figure. Fig. 5.16 shows the result of the teach-in phase. The assignments, that were defined in the association matrix are reproduced. At $2.5s$, instrument state I is detected and melody pattern B is played. At $6s$ instrument orientation II is detected and melody pattern A is reproduced. At $18s$, instrument state III is observed and correctly associated to perform melody pattern C.

5.6.4 Discussion

In the experiments I tried to evaluate in how far, the proposed mapping module functions according to the purpose of its development. In case of the basic interaction level system the mapping module is to transform data from the vision processing module to modulation of musical performance parameters, using

feedback data about the physical state of the robot. In the experiments for the basic interaction level mapping module, I show that sensor input can be directly translated into performance parameters and from a certain threshold level be modulated by the lung fill level of the flutist robot. The experiments were performed for different wave-forms and different wave frequencies. As successful transformation was observed in case of all wave-forms and frequencies, I assume that the mapping works for typical human instrument movements (wave-forms and frequencies were chosen to emulate such typical movements). The feedback modulation takes place as a fade-out. For this fade-out two different forms were applied. With both wave-forms a complete fade-out within the critical lung fill level takes occurs. In case of the exponential curve the fade-out is much quicker than in case of the linear curve. For further experiments involving human interaction partners I favor the linear fade-out curve, as it allows for a more gradual fade-out of the modulation parameter. In case of the faster input oscillations, the result was not clearly observable in the robot performance pitch graph. The performance tempo was in these cases slow, compared to the oscillation of the input sine-wave. The higher frequency input waves were chosen to test the functionality of the systems for possible inputs that might in principle occur, but in the practical case, the sine-wave at a frequency of around 0.1 Hz most closely models the typical human instrument movement. We will see this also in the next chapter, where experimental results from interaction with human players are shown.

In case of the extended interaction system the mapping module is to allow the user to assign a relationship between instrument orientation and musical output patterns in a teach-in phase. In a performance phase this assignment data is to be used to give the user the possibility to select output patterns using movements of his instrument while playing. In the experiments I show that associations that have been assigned in the teach-in phase are correctly recalled in the performance phase. This is shown for two example assignments and simulated performance inputs. The assignments however are very simple. Three instrument states are uniquely assigned to three melody patterns. The is no interference in the assignments (Two melodies assigned to the same orientation area). This makes the probability distribution in the assignment table very simple. In case

of the experiments presented in this section I accounted for a limited case of assignment complexity, because I considered it sufficient, in order to show that the system is suitable to be used for the interaction with a human player. In case of the interaction with a human player, in the scope of this thesis, I chose to limit the experiments to a simple setup, that is easily understandable for musicians without any engineering background.

5.7 Conclusion of this Chapter

As a result we expect improved performance capabilities of the robot. Compared to the static, relatively simple mapping we have done so far, much more complex interplay with the robot is possible.

Also collaborative performance of more that one musician, with a variety of virtual buttons and faders to control the robot's output, might be reasonable. Similar to the beginner level interaction system mapping module, also here we need to keep the physical limitations of the robot in mind. To do this, after the teaching phase we analyze the state-space table for invalid parameter state-effects (the effect of a certain song condition on the modulation) to be directed to the robot. If one of these invalid state-effects is found, it is replaced with the closest valid effect state.

In the experimental section we have shown, that the basic interaction level works according to its development purpose. It converts the instrument motion input by the musician into parameter modulation of the flutist robot performance. At the same time the mapping is adjusted by the physical state of the robot. In our experiments this physical state is the fill-level of the lung of the robot. We tested several input waves (sine and square waves with frequencies ranging from 0.1 Hz to 10 Hz) and two different fade-out curves (exponential and linear fade-out curve). As a result, after consultation with a professional, we chose a linear fade-out curve as optimum for using the basic interaction mapping module in case of a real interaction as described in the next chapter.

In case of the extended interaction system, we verified the functionality of the teach-in phase and the performance phase by using various input combinations. The instrument orientation motion was simulated using a 0.1 Hz sine-wave. The

input of 3 melody patterns was triggered manually. We conclude from the experimental results, that the use of the extended interaction system mapping module is valid for the real interaction experiments with human musical partners.

Chapter 6

Evaluation of the Proposed Interaction System from a HRI Research-oriented Perspective

6.1 Introduction

In this chapter I would like to describe the previously introduced system from a universal HRI Research-orientated point-of-view. An analysis of the restricted parameter adjustment within the physical limitations of a robot and an analysis of the variable sensor mapping of the extended level interaction system are performed through experiments under realistic circumstances. I first show a comparative evaluation of a passive and an interactive performance. This evaluation is done in two ways: A performance index for both performance methods is calculated and compared. The impression on listeners is evaluated through a survey. In a technical evaluation, I demonstrate the system's functionality by presenting data from experiments with the interaction system. Finally, the basic level interaction system as well as the extended level interaction system are evaluated from a non-technical user perspective. As a conclusion, the proposed interaction system is considered from a HRI point-of-view.

One application of the research proposed in this thesis is to enable the robot to interact with other players by perceiving information from its environment. From a more general Human Robot Interaction-oriented perspective, a system is to be

developed, that is able to recognize and create information that is transmitted during human communication. This information is contained in several channels such as acoustic expression or visual gestures.

For the implementation of this system an application to the musical context was chosen. To perform a piece of music, musicians need to constantly synchronize their play through several communication channels. To provide an entertaining experience to the audience this interaction needs to be flexible and at the same time exact. Therefore musical interaction is considered to be very rich in information. As the communication is focused on playing a piece of music and in this sense orientated on a fundamental scheme (the musical score, chord progression etc.), recurring patterns and structure provide an environment of limited complexity.

In the previous chapters we have introduced the vision sensor processing module, the audio processing module and the mapping strategy module of the proposed interaction system. I have shown in experiments that in a musical context, the system gives musicians the possibility to manipulate certain performance parameters of a complex humanoid robot. With the basic level interaction system and the extended level interaction system I accommodate for the specific experience level of the different users. In this section I would like to show experiments, that were performed to validate the results of the research, the ability of the robot to reliably detect, analyze and answer to information provided through several communication channels from its musical partner is to be documented.

In the first part of the chapter I perform the comparative analysis, in order to determine the different characteristics (by calculation and listener survey) of the passive and active performance system. It follows a technical system evaluation from interaction experiments in realistic situation oriented circumstances. Restricted parameter adjustment within physical limitations, as well as the analysis of variable sensor mapping of the advanced level of interaction, are evaluated. In the third part of the chapter, I look more closely at the experimental system evaluation from the non-technical user perspective. I perform a user survey for the beginner level interaction system and the advanced level interaction system.

From the results of these comparative, technical and human-point-of-view related evaluations I draw the conclusion, that the proposed interaction concept

is applicable for musical applications, and with adjustments also in a more general HRI context.

6.2 Comparative Evaluation of the HRI Interface: Comparison of Passive and Interactive Musical Performance

6.2.1 Objective

In this section I describe how passive and interactive musical performance with WF-4RV was evaluated. In the first part, I introduce a performance index, that has the purpose of mathematically characterizing the synchronicity of movements of the musical partner and the performance of the robot. In the second part, the results of a user survey to characterize, how the two performance modes are perceived by the listener. In a third part, I draw preliminary conclusion from these results. The result from this section will later on be evaluated in resume together with the technical demonstration and the user-perspective validation.

Both, the quantitative and the qualitative evaluation rely on the same musical material. The jazz standard The Autumn Leaves has been chosen as experimental material, because it a well-known piece of music, that can be musically appreciated my non-musicians and musicians alike. It does not require a very high instrument skill level to be performed, and thus allows a broader range of experimental subjects to interact with the flutist robot.

Images showing a musician during passive and active performance are shown in Fig. 6.1 and Fig. 6.2.

6.2.2 Performance Index: Method and Results

I created the performance index as a mean to quantitatively characterize the differences between the passive and the interactive (using the MbIS) performance of the robot. When considering the movements of a musician during a musical performance, as has been already considered in the first chapter, the synchronicity to the musical expression of the performer is high. If the musician plays very lively

Figure 6.1: In this screenshot a passive performance between the flutist robot WF-4RV and a human musician is displayed. The robot does not react to the play of the human musician. It plays a static score that is being sent from a MIDI sequencer.

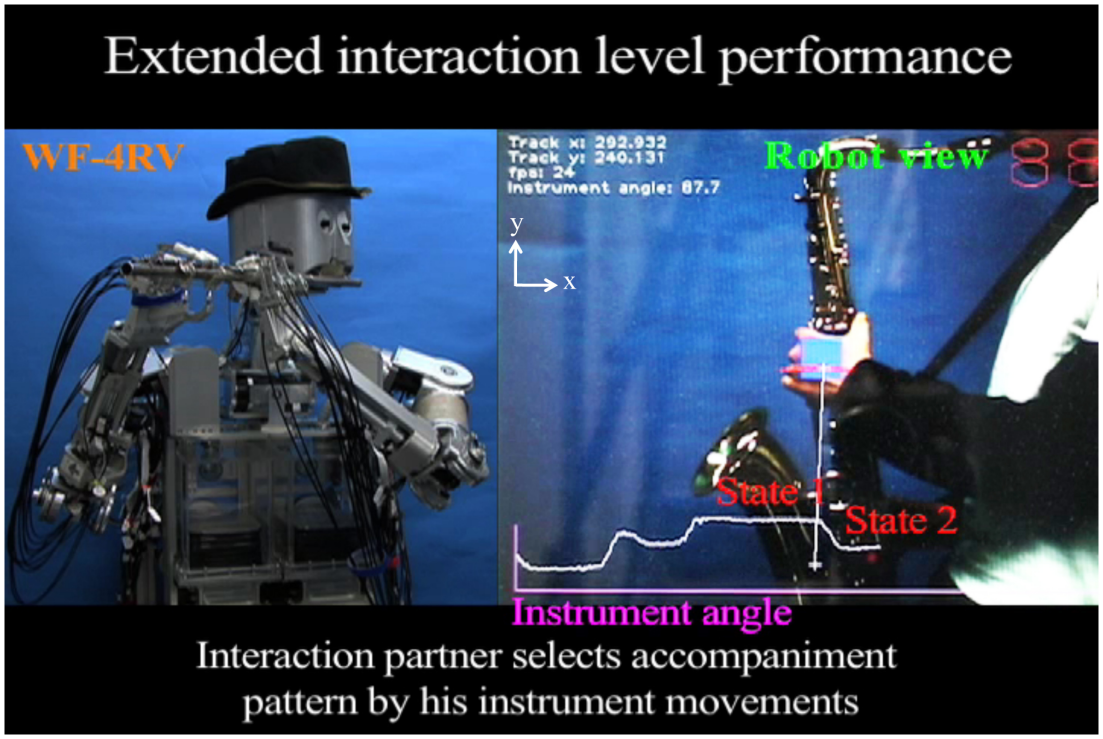

Figure 6.2: In this screenshot an inter-active performance with the flutist robot is displayed. Here the robot reacts to movements of the musician's instrument (the musician's hands respectively) and maps these into performance modulation.

musical material, the movements might be very strong, in case of quieter material, feeling might be expressed through slower motions.

The performance index was proposed based on the consideration, that strong change of musical material correlates to strong instrument motion. In order to measure this relationship I propose the following equation:

$$p = \frac{1}{T} \int_{t=0}^{T} \left| \frac{d\underline{i(t)}}{dt} \right| \left| \frac{d\underline{r(t)}}{dt} \right| dt \tag{6.1}$$

p: Performance index

T: Performance duration

$\underline{i(t)}$: Interaction-partner-state vector at time t

$\underline{r(t)}$: Robot-state vector at time t

In this equation I calculate the performance index p as an integration of the time differential of the interaction-partner-state vector $\underline{i(t)}$ and the time differential of the robot-state vector $\underline{r(t)}$. I integrate over the time parameter t with an integration interval equal to the duration of the recorded performance data T. The performance factor is normalized by factor $\frac{1}{T}$ to allow the comparison of performances of different durations.

To apply the performance index evaluation, I recorded a passive performance of the jazz composition The Autumn Leaves and an inter-active performance of the same piece. I divided the melody of the theme of the song into three different parts, that I refer to as melody patterns *A*, *B*, *C*. At the same time, I also characterize the instrument motion of the musician with three states *I*, *II*, *III*. Each of these states corresponds to a certain instrument angle interval.

- state *I* relates to $0° - 30°$
- state *II* relates to $30° - 60°$
- state *III* relates to $60° - 90°$

Trial	p (passive)	p (active)	T
1	0.32	0.76	50s
2	0.44	0.74	50s
3	0.30	0.83	50s
4	0.37	0.84	50s
5	0.47	0.89	50s

Table 6.1: The performance index result table is shown here. A higher index relates to a better action-reaction result. p (passive) describes the performance index for a passive performance, p (active) shows the result for an inter-active performance. Five trials have been conducted. The result shows a higher average performance index of 0.82 (variance 0.13) for the active performance, compared to a lower average index of 0.35 (variance 0.15) for the passive performance.

Using the integral the temporal changes of instrument state related to changes of the currently played melody pattern are summed. The result of this sum is a measure of the synchronicity of the instrument motions and the musical performance of the flutist robot.

I used the instrument tracking method of the advanced level interaction system to determine the instrument motion and the histogram-based melody tracking to identify the melody patterns.

The results shown in Fig. 6.1 express, that for the passive performance the relationship between motion and performed melody pattern is less strong than in case of the inter-active performance. To show that the performance index evaluation works reliably, we performed five trials of the same experiment (performed by the same player) and compared the resulting performance indices. Variances for both passive and active performance indices range to approximately 0.1. The average performance index for the passive performance amounts 0.35 and the according value for the active performance results in 0.82.

Adjective pair	Enquired characteristic
interesting / boring	Does the performance catch the listener's attention?
beautiful / awful	Is the performance overall perceived as pleasant?
melodic / atonal	Does the performance make a well composed impression on the listener? Are there atonalities?
rhythmical / at random	Is the performance rhythmically ordered and orientated to a constant beat?
homogenous / varied	Can the performance be characterized as diversified or is it rather monotonous?
natural / artificial	Does the performance convey a human like touch?
emotional / rational	Are the elements in the performance that make the user feel in a certain way / has an influence on the mood of the user?

Table 6.2: The table explains the adjective pairs used for the listener survey.

6.2.3 Evaluation by Listener Survey: Method

Additionally to the quantitative evaluation with the performance index, I also performed qualitative validation of the distinction between passive and active performance by a user survey. The user survey was done using the same musical material as introduced in the previous section: A passive and an active performance movie of the jazz piece The Autumn Leaves.

As experimental subjects, I chose 15 amateur musicians and 2 professional musicians. I created a questionnaire, that included 7 adjective pairs.

This method of posing a questionnaire was derived from a method proposed in [75].

The purpose of the survey was to find the impression of the two performance modes on the listener. This impression was characterized by the adjective pairs shown in Tab. 6.2. For each adjective pair, the survey subject was asked to

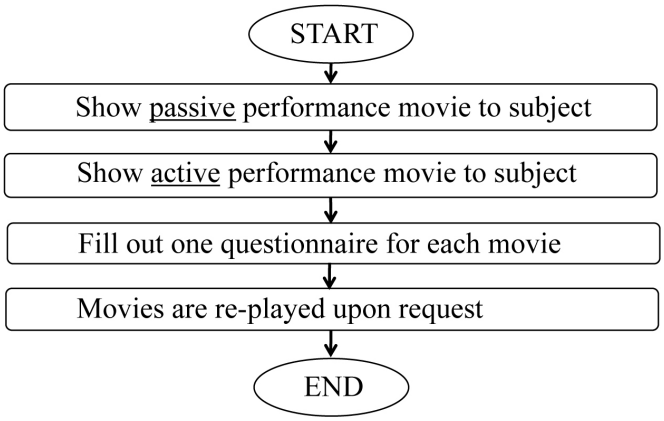

Figure 6.3: The methodology of the listener point-of-view survey is displayed in this flow diagram. The passive and active performance movie are shown, then the survey subject is asked to fill-out the questionnaire form.

express to which degree the performance could be classified by a certain adjective by selecting an integer number between 1 and 5. Applied to the adjective pair interesting / boring, a 1 would account for a completely boring performance and a 5 for a really interesting one. If a listener was indecisive on which adjective to choose, he could choose a 3 to emphasize neither one of both adjectives.

A flow chart of the experimental method of the survey is shown in Fig. 6.3.

6.2.4 Evaluation by Listener Survey: Results

For the amateur musicians I achieved the results shown in Fig. 6.4a. 15 Amateur musicians were asked to characterize the difference between the active and the passive performance video, that they were shown, using 7 different adjective pairs. The result of the survey shows, that in case of the adjective pairs *interesting / boring*, *homogenous / varied*, *natural / artificial* and *emotional / rational*, there is a significant difference (p-test result > 0.05) between the active and the passive performance. For the pair *interesting / boring*, the listener judged the active performance to be more attention attracting than the passive performance. The average score for the active performance here is 4.5 and 2.8 for the passive performance. For the adjective pair *homogenous / varied*, the amateur musician

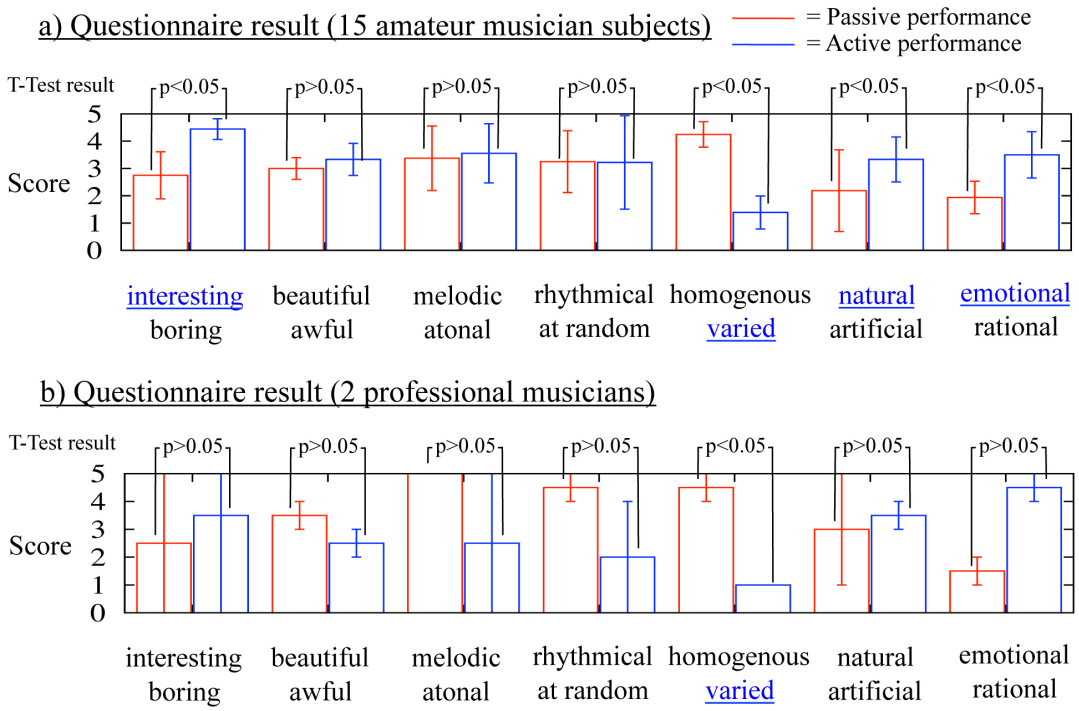

Figure 6.4: In the two graphs above, the results of the listener survey to compare the active and passive performance are shown. In a) the averaged questionnaire scoring by the amateur musicians is shown. b) shows the survey results for the professional musicians. The T-test results framed with a green rectangle point to an adjective category for which there is a significant difference between the result for the basic and extended level system. Red boxes show the scoring for a passive performance and blue boxes display the results for an active performance.

listeners gave an average score of 4 to the passive performance and a score of 1. The active performance is judged to be more varied than the homogenous passive performance. In case of the adjective pair *natural / artificial*, a score 2.5 was reached for the active performance and a score of 2 for the passive performance. The active performance is therefore judged to be more natural than the passive performance. The adjective pair *emotional / rational* shows a higher score of 3.5 for the active performance, compared to 3.5 for the active performance. A stronger emotional expression was attributed to the active performance than to the passive performance. In case of the adjective pairs *beautiful / awful*, *melodic / atonal* and *rhythmical / at random*, there is no significant difference to be noted between the passive and the active performance.

The results of the professional musician survey are shown in Fig. 6.4b. Two professional musicians were asked to characterize the difference between the active and the passive performance video, that they were shown, using 7 different adjective pairs. Only in case of the adjective pair *homogeneous / varied* the result of the p-test id > 0.05 and therefore shows a significant difference. The remaining pairs (*interesting / boring*, *beautiful / awful*, *melodic / atonal*, *rhythmical / at random*, *natural / artificial*, *emotional / rational*) are not attributed to be significantly different for the active and the passive performance by the professional musicians.

6.2.5 Discussion

For the comparative evaluation of a passive performance and an active performance from a listener point-of-view, I performed a quantitative evaluation using the performance index, and a qualitative evaluation performing a listener survey. The performance index shows a higher average (0.82) for the active performance evaluation. The passive performance average results amounts in only 0.35. The performance index characterizes the amount of synchronized motion and change of the musical performance. The result values confirm, that in the active performance, the musical content changes in accordance to the movements of the instrument of the musician. This leads to the conclusion that there is a stronger relationship between musician physical action and musical reaction, in case of

the active performance. As the active by conception and technical realization has been developed to generate a musical performance, in which this relationship is very strong, this result is according to our intended outcome.

However, with the proposed method, I can only characterize physical and musical expression of the experimental subject to a certain degree, as far as the vision and aural processing methods proposed in this thesis allow it. The functionality of these modules has been described in the previous sections. A musician's expressive behavior during a performance is very rich in information content. There might be parts of that information that express synchronicity (or even non-synchronicity), that are not sufficiently be detected by the proposed sensor processing technics, but still should have influence on the performance index. Therefore is the performance, a quantitative way of expressing the action-reaction relationship within an inter-active musical performance, but it needs to be supplemented by further experimental proof.

To provide this further experimental proof, I performed a listener survey, in which experimental subjects were ask to evaluate an active and a passive performance. Especially in case of the survey with 15 amateur musician subjects, the survey shows promising results. The active performance scored significantly higher (with the result of the p-test > 0.05) for classifications *interesting*, *varied*, *natural* and *emotional*. As I have already shown in the results regarding the performance index, the active performance shows a higher degree of relationship between visible actions by the musician and musical performance output. This results in the impression of a more interesting and varied performance on the listener viewer of the performance. In the Introduction of this thesis it was stated that the active performance makes a more natural impression to the listener and the musical performer. Given, that this statement is true, the result for the adjective pairs *natural / artificial* and *emotional / rational* can be explained. The active performance was attributed a higher score for naturalness and emotionality than the passive performance. An explanation might be, that the additional physical movement, resulting in stronger synchronicity of the two performers, shows more human-like features, that a static performance without further exchange of information might not display. This leads the listener / viewer

to the conclusion that the active performance is more natural (human-like) and conveying more emotional content than the passive performance.

With the proposed survey method, I try to qualitatively characterize the impression of the active and passive performance on the user. However, I provide to the survey subjects only a limited way of describing their impression. The real judgement of the subjects might be significantly more diverse, than can be expressed in the proposed adjective pairs. As a result, to a certain degree, the result does sufficiently characterize the active and passive performance situations. For further insight on its impact on the user, the technical evaluation and the evaluation from the user point-of-view were performed.

6.3 Technical System Evaluation from Interaction Experiments in Realistic Circumstances

6.3.1 Objective

In this section I describe the technical evaluation of the interaction system under realistic circumstances. With realistic circumstances I intend to express, that I did not set any specific environment conditions to improve the performance of the interaction system. I tried to test the system in a similar situation as it might be used in a real, public performance situation. Although the tests were still carried out in the laboratory, I did not take any special effort in adjusting lighting, background, environment noise level etc. Intentionally, I did not perform a quantitative evaluation of the performance system by comparing the input values from the different sub-modules to ground-truth data, but concentrated on demonstrating the functionality by using the system in a realistic way. Experiments were done with both the basic interaction system and the extended interaction system.

6.3.2 Analysis of Constraint-Restricted Mapping (CRM) within Physical Limitations: Experiment Method

In the following I show the technical evaluation of the basic interaction level system, using the robot Constraint Restricted Mapping (CRM). Purpose of the

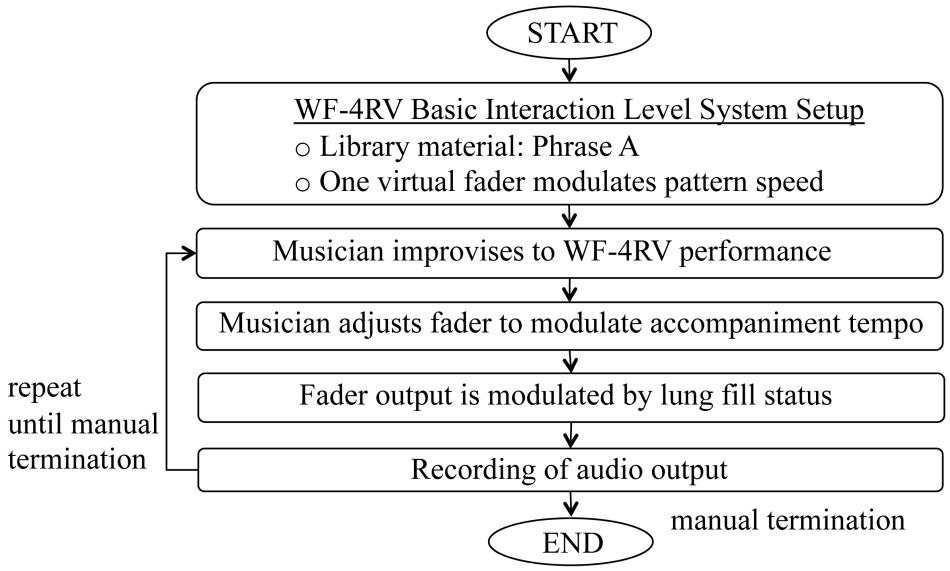

Figure 6.5: The technical experiment for the basic level of interaction was performed as shown here.

evaluation is the demonstration of the technical functionality of the basic interaction system. In the experimental setup, an intermediate level saxophone player improvises over a repetitive musical pattern from the theme of the jazz standard piece *The Autumn Leaves* (the primary exemplary piece used throughout this thesis). By moving his instrument, the musician can adjust the tempo of the sequence, that is performed by the flutist robot. While the musician controls the performance of the robot, his input is modulated by the physical state of the robot. As the relevant state parameter can be deliberately selected by the user, I chose the state of the robot's lung to modulate the values, that are transmitted from the sensor processing module.

In case of the basic level interaction experiment, the robot is controlled by one virtual fader. This fader is used to continuously control the speed of a predefined sequence that is played by the flutist robot. The output of the sensor processing system determining the value of the virtual fader is conditioned by the lung movement of the robot. I use the Constraint Restricted Mapping (CRM) to continuously reduce the speed of the performed pattern, when we reach a fill-level of 80% of the robot's lung. In order to perform the experiment, an intermediate

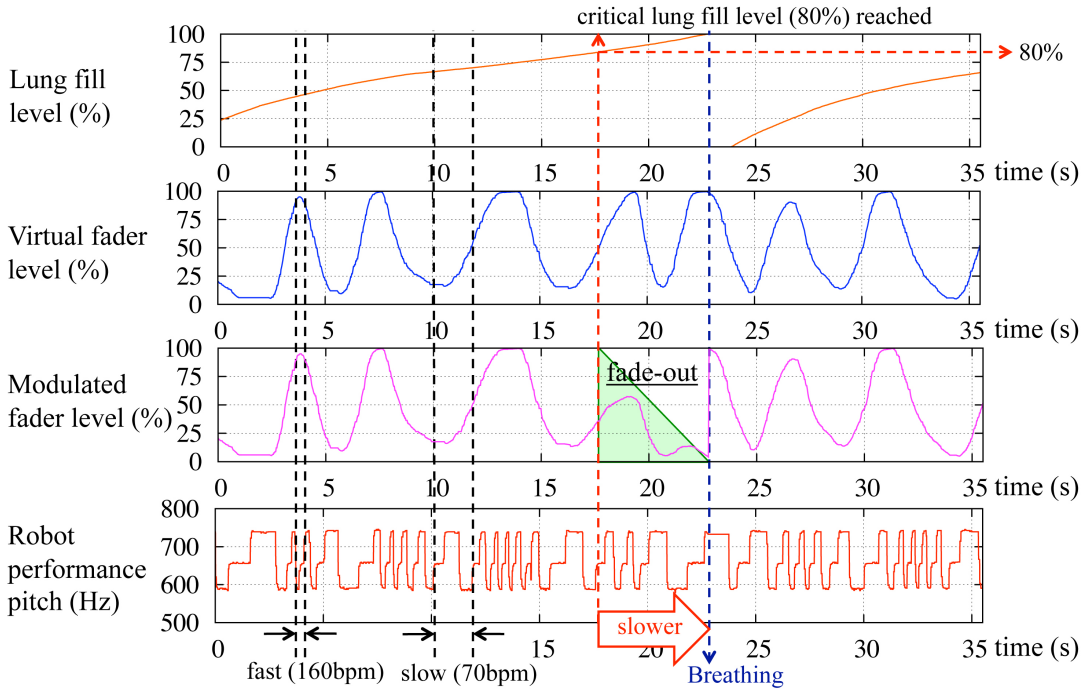

Figure 6.6: In the beginner level interaction system, the user controls the tempo of a pattern performed by the robot. The lung fill level plotted in the top graph, modulates the input data from the virtual fader resulting in the robot performance displayed by the pitch and the amplitude curve.

saxophone player stood in front of the robot (within the viewing angle of the robot's cameras). After introducing the functionality of the basic level interaction system to the player I recorded data of the resulting performance with the robot. A flow-chart of the experiment procedure is displayed in Fig. 6.5.

To achieve quantitative results for the first level interaction system we performed the experiment with an intermediate flutist player. Fader movements control the tempo of the tone sequence that is performed by the robot. If the amount of air remaining in the lung reaches a certain limit (in this experiment 80% of the lung volume, see lung fill level graph), the fader value transmitted to the robot is faded-out (using the low-pass filter previously described).

6.3.3 Analysis of Constraint-Restricted Mapping (CRM) within Physical Limitations: Experiment Results

A graph of the result of the experiment is shown in Fig. 6.6. At $23.5s$ and $24s$ the robot refills its lungs for a duration of approx. $0.5s$. These breathing points have a time-distance of approx. $40s$ (in the displayed graph one breathing cycle is displayed). During the breathing points no sound is produced by the robot. The fader value actually transmitted to the robot (Modulated fader value) is faded out before the lung is completely empty. This adjustment can be observed at $17.5s$-$22.5s$ in the fader value plot, the modulated fader value plot and the robot output volume plot. As the fader value is faded-out rapidly, the resulting performance tempo of the robot decreases from fast ($160bpm$) to slow ($70bpm$). This variation can been seen in the pitch plot in the robot performance pitch plot.

Musicians can engage in interplay without having to consider about the physical restrictions of the anthropomorphic robot. The constraint restricted mapping (CRM) makes the robot aware of its own limitations and able to adjust its performance accordingly, similar to how a human player might act. I show in my experiments, that this principle is applicable to simple improvisational play together with a musician partner. As mentioned previously, I try to achieve musical expression to the degree that is possible, given the physical limitations of the flutist robot. As first approach, in this research, the experiments were limited to changes of the song tempo and vibrato amplitude. These expression parameters are relatively simple in respect to the complex parameter variations, which a human musician applies, while performing a musical piece. However, to produce the flute sound, accurate control of all degrees of freedom of the robot is necessary. If this control is not accurate enough, the sound production will fail. For this reason I limited the parameter modulations presented in these experiments to relatively simple conditions.

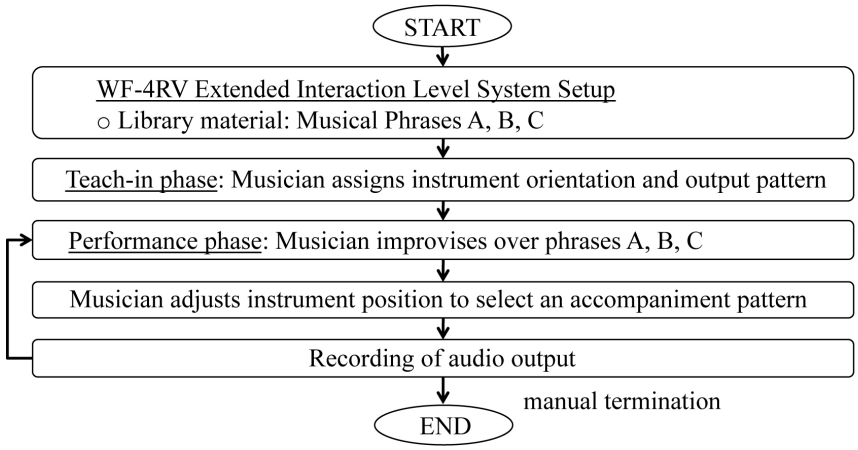

Figure 6.7: The experiment sequence for the advanced level of interaction.

6.3.4 Analysis of Interactive Mapping (IM) of the Extended Level Interaction System: Method and Results

In the following section I describe my approach to evaluate the technical functionality of the extended interaction system. As purpose of the experimental procedure I demonstrate the technical functionality of the extended interaction system. Similar to the technical evaluation of the basic level interaction system, I concentrate on a demonstration of the qualitative functionality of the system under realistic circumstances, rather than calculating quantitative error values for the separate system components. In the experimental setup, an intermediate-level (instrument skill level) saxophone player controlled the robot for improvisation of the theme on the piece *The Autumn Leaves*. The musician controls the robot using the interactive mapping (IM) of the extended level interaction system. In a teaching phase the musician associates instrument angles and melody patterns to form a state association table in the memory of the flutist robot. In the performance phase the musician uses this state association table to recall musical patterns by his instrument motion during improvisation.

In the experiment for the advanced level interaction system I try to confirm that, using the interactive mapping (IM) module, a musician of intermediate skill

level has the possibility to teach the robot how to relate instrument motion with the variation of musical parameters.

The experiment has two phases, the teaching phase and the performance phase. In the first phase the interacting musician teaches a movement-performance parameter relationship to the robot. In this particular case we relate one of three melody patterns to the inclination angle of the instrument of the robot's partner musician. From this information the robot builds a state-space table, that relates instrument angles to musical patterns. In the second stage the interaction partner controls the robot with these movements. Using the proposed particle Bayesian network-based filtering I search for the instrument angle in the state-space table that most likely represents the current state. When a match is determined, the robot plays the musical pattern that relates to the current instrument angle. The transition of the teaching phase to the performance phase is defined by the number of melody patterns associated by the robot. In case of this experiment, the switch occurs after 3 melody patterns have been recorded. A flow-chart of the experiment procedure is displayed in Fig. 6.7.

The experiment was performed by the intermediate-level flute player. After introducing the functionality of the system to the player, he performed one teaching phase and the following performance phase. In the following I show and evaluate an excerpt of the data recorded from the interaction of the intermediate level player with the system.

The recorded data for the teaching phase can be seen in Fig. 6.8. In the first part (from $T = 0s$), the instrument player moves his instrument to an angle of approximately 125° (state I) and plays melody pattern A. The flutist robot confirms the detection of the pattern by repeating the melody. This is displayed in the *Robot performance pitch graph* and marked with m (musician) and r (robot). The association of pattern A and instrument state I is written to the robot state table. At $T = 18s$ the player changes his instrument position to approximately 100° (state II) and plays the next melody pattern, which is recognized and confirmed as melody pattern B. The association of state II and pattern B is memorized in the robot state table. At last ($T = 22s$), the instrumentalist moves his instrument to state III (approximately 75°, and plays melody pattern C. The association of instrument state III and melody pattern C is saved in the association table.

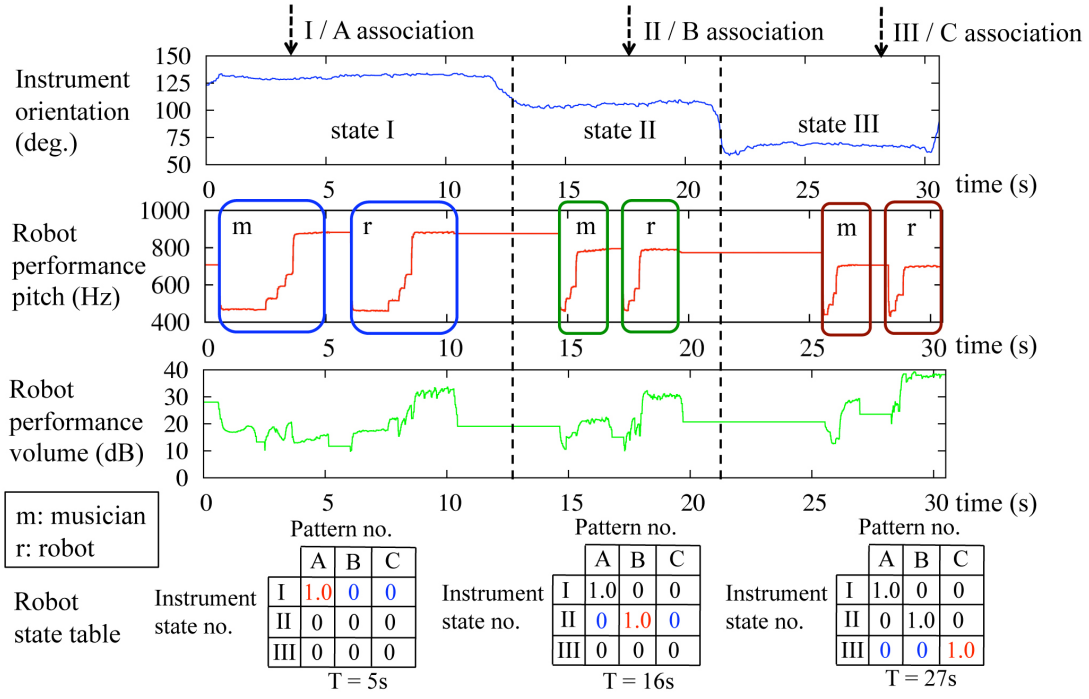

Figure 6.8: In the extended level interaction system's teach-in phase the user associates instrument motion with melody patterns. A melody pattern m performed by the musician is repeated by the robot for confirmation r. The robot state table shows the association that is being set-up the teach-in system.

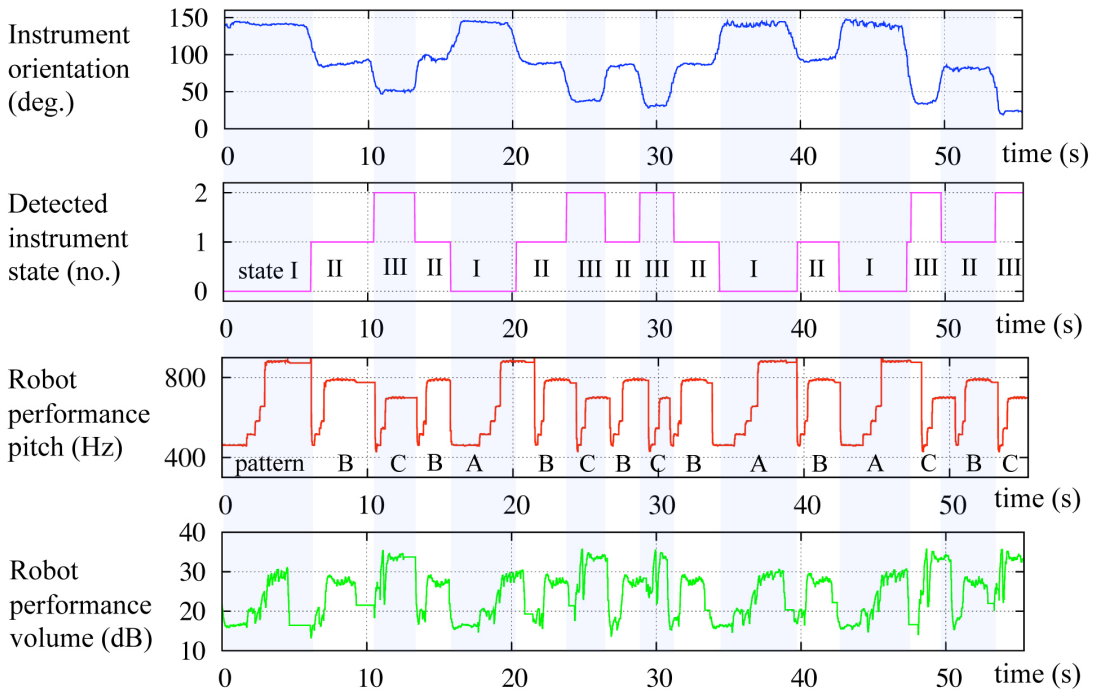

Figure 6.9: In the extended level interaction system's performance phase the user controls the robot's output tone by changing the orientation of his instrument. In the graph the detected instrument orientation, the associated musical pattern and the output of the robot are shown.

The results for the performance phase of the extended level interaction experiment is shown in Fig. 6.9. In the teaching phase the musician associated three melody patterns A, B, C, to instrument states I, II, III. In the performance phase he recalls the melody patterns in order to build an accompaniment for an improvisation. In the graph, each time the musician shifts his instrument to a new angle (Instrument orientation graph), the detected instrument state changes. As a result of this change the robot plays the answer melody that was associated in the teaching phase. This happens several times in the displayed graph. At $15s$ the musician moves his instrument to an angle of 150° (state I) and the robot immediately plays the associated answer melody (pattern A). At $20s$ he shifts the instrument to 100° and triggers melody pattern B. When moving the instrument to 50° at $23s$, the robot answers with melody pattern C. It remains to note, that, after one pattern has been performed the robot automatically resets its lung until the next pattern is commanded. These short breathing spots can be seen throughout the *Robot performance volume* plot, notably at $t = 5s$, $t = 9s$ or $t = 13s$.

6.3.5 Discussion

With the experiments performed in this section, I tried the technical validity of the basic and extended level of interaction. This was to be achieved not on a sub-module level, as was done in the previous sections, but using the system as a whole. Therefore the input was also provided by a real musician user, to see if the system is suitable to work in a realistic context. In this case realistic context means, that the characteristics of the user input are the same as user input, that would occur during a stage performance in front of an audience. In case of the basic level interaction system, the user generated a regular sinusoidal input waveform with a frequency of around $0.1Hz$ and an amplitude of around 0% to 100%. This input data was translated into proportional modulation of the performance tempo of the flutist robot. As soon as the fill level of the lung of the flutist robot reached a level of 80%, the lung fill level started to adjust the modulation of the performance by the musician and force a fade-out to the performance tempo of the robot from $170bpm$ to $70bpm$. As I showed in the experimental results

graph, the intended outcome of the experiment can be verified by analyzing the experimental data. However, the proposed experiment covers only one case of a constraint restricted mapping (CRM) configuration. This configuration was suggested a conceptual test of the basic level interaction system by the professional musician, I worked together with, when planning the presented experiments. Also have experiments only been performed for relatively simple types of user input. In a realistic performance, more extreme movements and musical expression than proposed here might occur. As a result the basic interaction level can be characterized as functional from the technical point-of-view for a certain performance scenario. The certainty, that the system works with every possible input data, cannot be achieved. The system can only be evaluated on a case-by-case basis. I tried to choose the situation proposed here as an example for a typical scenario. To further classify, in how far this makes sense also from the point-of-view of a real user, I try to explore in the next section.

The extended interaction level has been evaluated using a similar approach as the previously applied technique for the basic interaction level. The extended level of interaction was to be evaluated on an overall level, including the functionality of all involved modules (sensor processing, mapping etc.). The input that was used to verify, if the result data of the interaction with the extended level is according to the implementation intention, was generated by a human musician. In the previous section I used artificially generated input to show the functionality of the basic and extended level mapping module. In this chapter, the intention behind the proposed experiments is to show, if the system works in a realistic context. This makes it necessary to use realistic, human input to evaluate the system.

In the teach-in phase, the musician assigned three different melody patterns to three different instrument angles. This assignment was intentionally defined by the musician, using a melody pattern-based question-and-answer teach-in process similar to the approach shown in Chapter 4. The association table was built from two parameters (angle, musical pattern). The experiments show that after a-priori definition of the melody pattern library and connected instrument gestures, the robot is able to reproduce these pattern from the musician's movements in the performance-phase. In the experiments we have worked under a controlled

experimental environment with limited complexity. Using the proposed mapping strategy, more complicated applications, in which the state-space table contains more parameters (distance of the player to robot, angle, vibrato amplitude, musical pattern etc.) are possible. The complete teach-in procedure takes a during of about $30s$. In the later performance phase, the assignment that has been set in the teach-in phase is recalled by the musician, in order to create an interactive performance based on the musical fragments from *The Autumn Leaves*. The results show an association between instrument angle and musical pattern output as assigned in the teach-in phase.

In the extended interaction level experiment, I evaluated in the system in a way, that I considered a typical use-case scenario. However this describes only one of these scenarios. To a certain degree this leads to the conclusion, that as the system can be used in the presented case, it is also suitable for similar performance situations. To further evaluate the system for suitability to fundamentally different scenarios, the proposed experiments might not be sufficient. With the experiments in the following section I try to show, that at least for typical usage by a human the musician, the system leads to a good impression on the user.

6.4 Experimental System Evaluation from the Non-Technical User Perspective

6.4.1 Objective

In this section, I show experiments and their results that qualitatively document the usability of the system. Goal of the development of the Musical-based Interaction System is to provide the user with an intuitive, natural interface to interact with a musical robot. To find out about the acceptance of the system and its general usability, the system needs to be tested with a variety of users, the experience and comments of the users need to be recorded and these results need to be analyzed. Users were asked to do a musical performance with the flutist robot, first using the basic level interaction system and second using the extended level interaction system. I asked the users to fill out a questionnaire for each of these performances to characterize their experience with the system. This

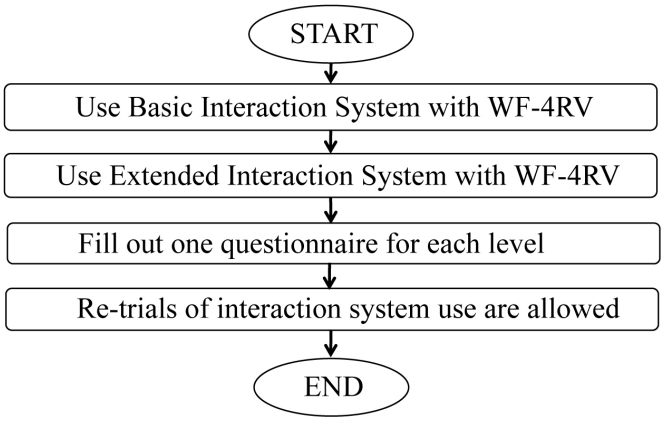

Figure 6.10: User survey method

experimental method was applied for professional musicians as well as amateur musicians and the result statistically analyzed. With the results I try to show that the system provides a natural user experience to users of different experience levels.

6.4.2 Experiment Method

In Fig. 6.10 the experimental method of the user survey is shown. Purpose of the experiment is a survey of the user evaluation of the system during practical operation of the basic and extended level interaction system. For the evaluation of the basic interaction system, I analyzed the the qualitative perception of the basic interaction system with a questionnaire poll of 17 subjects (amateur musician and professional musicians). For the evaluation of the extended interaction system I analyzed the qualitative perception of the extended interaction system with a questionnaire poll of 17 subjects (amateur musicians and professional musicians).

In the survey experiment I asked 15 male amateur musicians and 2 male professional musicians to use the basic interaction system with WF-4RV and to use the extended interaction system with WF-4RV. For each interaction level one questionnaire needed to be filled out. Re-trials of the interaction system use were allowed. Each questionnaire consisted of 8 pairs of adjectives, similar to the

Adjective pair	Enquired characteristic
natural / artificial	Does the user perceive the interaction with the robot as natural or artificial?
fast / slow	Is the overall response time of the system perceived as fast or also?
free movement / constrained movement	Is the expressive movement required in order to control the robot performance perceived as free or constrained?
rhythmical / at random	Is the performance of the robot that results from the interaction perceived as rhythmically aligned or randomly arranged?
emotional / rational	Does the resulting performance allow for emotional expression of the user's mental state?
expressive / unexpressive	Is the interaction with the robot perceived as expressive or unexpressive?
easy / difficult	Did the user find the interaction with the robot easy or difficult?
system sufficient / insufficient	Is the scope of the system perceived as sufficient or insufficient?

Table 6.3: 8 Pairs of adjectives were chosen to allow users to characterize the usability of the basic and extended interaction level.

approach proposed in [75]. As a score system, I asked each user to assign a score from 1 to 5 for each adjective pair (Table 6.3).

For each adjective pair, the survey subject was asked to express to which degree the performance could be classified by a certain adjective by selecting an integer number between 1 and 5. Applied to the adjective pair natural / artificial, a score of 1 would account for a very natural, human-like interaction and a score of 5 for a very machine-like, static one. If a user was indecisive on which adjective to choose, he could choose a score of 3 to emphasize neither one of both adjectives.

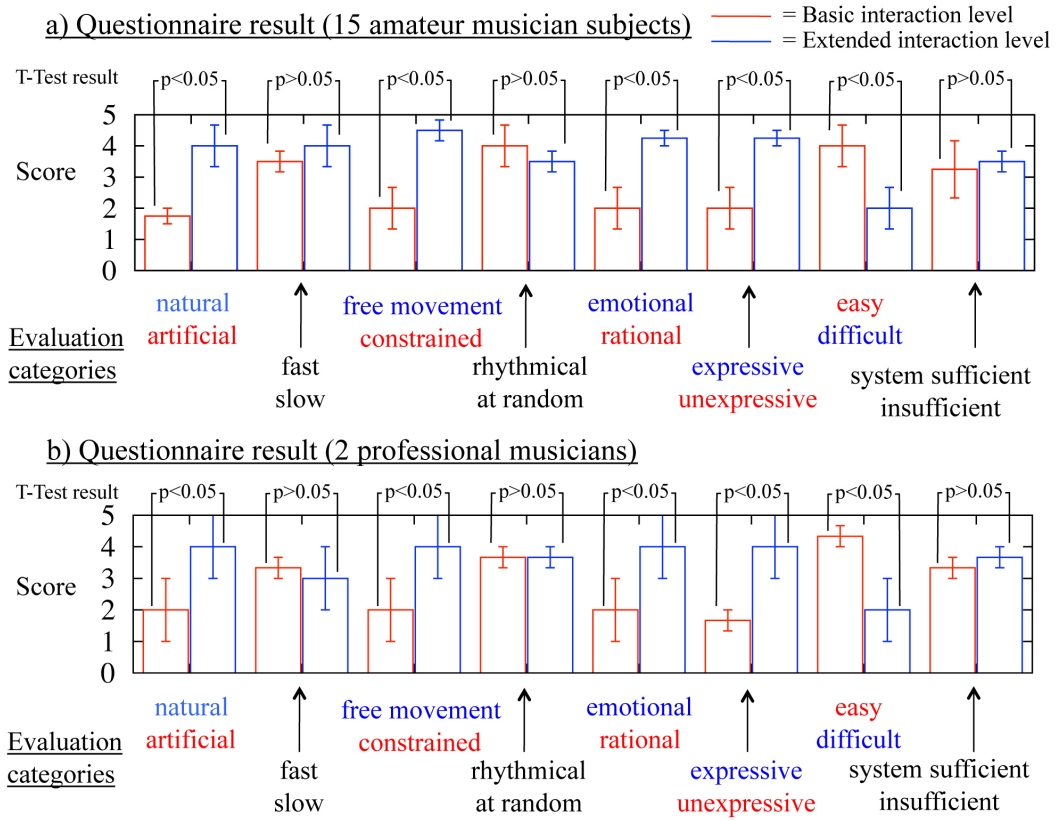

Figure 6.11: This figure shows the results of the user survey for the basic interaction system and the extended interaction system. In a) the averaged questionnaire scoring by the amateur musicians is shown. b) shows the survey results for the professional musicians. The T-test results framed with a green rectangle point to a adjective category for which there is a significant difference between the result for the basic and extended level system. Red boxes show the scoring for the basic interaction level and blue boxes display the results for the extended interaction level.

6.4.3 Results

The results of the user survey for the active and passive interaction system are shown in Fig. 6.11. 15 Amateur musicians were asked to characterize the differences between the use of basic level of interaction and the extended level of interaction. The users were asked to attribute 8 pairs of adjectives to the two levels after interacting with the robot using each of these levels. The result of the survey shows, that in case of the adjective pairs *natural / artificial*, *free movement / constrained*, *emotional / rational*, *expressive / unexpressive*, *easy / difficult*, there occurs a significant difference (p-test result > 0.05) between the basic and extended interaction level.

For the pair natural / artificial, the amateur users in average gave a score of 1.7 to the basic and a score of 4 to the advanced interaction level. A similar result was achieved for the adjective pair *free movement / constrained* with a score of 2 for the basic interaction level and 4.5 for the advanced interaction level. In case of the adjective pair *emotional / rational*, the basic level scored 2 and the extended level 4. The basic interaction level was attributed with a higher score of 3.8, than the extended interaction level with 2 for the adjective pair *easy / difficult*.

Furthermore, I asked 2 professional musicians to use the basic and extended interaction level and attribute their impression with the previously described adjective pairs. The outcome of the experiment is similar to the results for the amateur musicians. In case of the adjective pair *natural / artificial*, the basic interaction level scored 2, whereas the extended interaction level achieved 4. For the adjective pair *free movement / constrained* a score of 2.2 was attributed to the basic level of interaction and a score of 4.5 was attributed to the extended interaction level. In case of the adjective pair *emotional / rational* the professional musicians in average evaluated the basic level with a score of 2.5 and the extended level with a score of 4. A score of 4.5 for the basic and 2.5 for the extended interaction system were attributed for the adjective pair *easy / difficult*.

6.4.4 Discussion

The intention of the employed user-study was to evaluate in which way the usage of the basic and extended interaction level was characterized by the user. The survey subjects were asked to provide such a qualitative characterization by giving scores to a number of adjective pairs, as has been described in the previous section. I found a significant difference in score attribution in case the five adjective pairs *natural / artificial*, *free movement / constrained*, *emotional / rational*, *expressive / unexpressive* and *easy / difficult*. As has been mentioned previously for the survey in the lister point-of-view evaluation section, these adjectives were chosen to find an approximate characterization of the impression of the interaction on the user. The real *feeling* a user has during the interaction, is difficult to be measured and displayed as numerical representation. Therefore, I chose to show this approach as a result that is closely related to the actual average impression a typical user has during the interaction with the system. The results are congruent for both the amateur musicians and the professional musicians. However, it has to be taken into consideration, that the number of survey subject was less in case of the professional musicians (two subjects) than in case of the amateur musicians (15 subjects).

Compared to the basic interaction level the extended interaction level was attributed a higher score for adjective pairs *natural / artificial*, *free movement / constrained*, *emotional / rational*, *expressive / unexpressive*. The greater amount of freedom of movement in case of the extended level of interaction is directly related to the visual processing method chosen for this level. The particle filter-based tracking allows instrument movements in the complete image space of the robot cameras, whereas in case of the motion perception-based tracking, the tracking area is limited to the area occupied by the virtual controllers. On the other side, as the virtual controllers are unmovable and provide a fixed reference for the user, the basic interaction level might also be easier to use. This is expressed by the user evaluation of the adjective pair *easy / difficult*. As a further fundamental approach to make the interaction with the robot easier in the basic interaction level, there is the robot state feedback concept. Using this approach, the user is not directly confronted with the technical complexities of the anthropomorphic

robot. Furthermore, the extended level of interaction offers the teach-in system to provide more freedom of expressiveness to the user. However that also means that the usage of the system in extended level mode becomes more difficult. The scores attributed for the adjective pair *expressive / unexpressive* further confirm this assumption. As average impression for the amateur musician survey subjects as well as the professional musician subjects, the extended interaction level was evaluated to be more natural and more emotional in its usage. This might be related to the additional freedom of expression given by the teach-in system and the use of the particle filter-based tracking. As a completely human-like interaction would grant a very high degree of autonomy to the musician partner of the robot, the greater the number possible free decisions in the course of a performance are and the more easily they are accessible, the more might the interaction itself be classified as natural. As a result such a natural interaction gives the musician freedom for emotional expression.

6.5 Conclusion of this Chapter

In this chapter the interaction system has been evaluated in three different categories. The difference between a passive performance and an active performance was analyzed. Through the calculation of a performance index and the application of a user study, I concluded that to a certain degree, the active performance is more similar to a performance between humans, than a passive performance between robot and human. This led to the next section, in which I displayed experimental results to demonstrate the technical functionality of the basic and extended level of interaction. Although these results did not cover all possible cases of usage of the system, I decided to further evaluate the system with the user survey shown in the next section. These user survey results show, that the interaction system levels are characterized differently by the amateur and professional musician users. The basic level interaction system on the one hand, is evaluated to be more constrained and in general provide a more artificial feel, but is easy to use. The extended level interaction system on the other hand is more complicated in its usage, but due to its greater flexibility leads to more natural and expressive performance.

These diverse results show, that an interaction system to suit all types of users can not easily be developed. However, the results, that are displayed in the listener and user survey show, that the general concept of the proposed system points into the right direction. Although the musical application is a specialized case of human-robot-interaction, it contains many characteristics (such as communication through different perception channels etc.) that occur also in other types of such interaction. This allows the consideration, that a similar system with slight modification might also be applied to other cases of personal robotics.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

In this thesis I propose an approach to a perception action scheme-based human-robot robot interaction system, that is to allow natural and intuitive collaboration between a human and a robot. I would like to present the following conclusions that show that, to a certain degree, the intended result of developing such a system has been achieved.

1. **The anthropomorphic flutist WF-4RV in an interactive musical context** - To prove the concept of operation of this interaction system, I implemented it for a musical context. In this musical context, the flutist robot WF-4RV is to play together and interact with human musicians in a natural way. The research is based on the fundamental consideration, that an inter-active performance of a robot with human musicians, is more enjoyable for the audience, than a passive, non-interactive performance. A passive performance therefore leads to a static, unnatural impression on the listener, whereas the inter-active performance is perceived as natural and human-like. I tried to prove this statement with the listener survey performed in the last chapter. In the experiment, the active performance scored significantly higher (with the result of the p-test > 0.05) for classifications *interesting*, *varied*, *natural* and *emotional*. I also introduced a performance index, that the active performance has a stronger relationship

($p_i = 0.82$) between visible actions by the musician and musical performance output than the passive performance ($p_i = 0.35$).

2. **Configuration of the active performance system of the WF-4RV** - Interaction with human musicians requires the flutist robot not only to play a carefully optimized performance. The robot needs to be able to adjust its play to the actions of its musical partner, without diminishing its performance quality. An adjustment of the note envelope for each tone during a performance requires very fast and accurate control of the air-flow. I achieved this by directly controlling the breathing speed of the flutist robot depending on the performed note. Using this envelope control system, decay times for large note-intervals (more than 6 half-tones) was reduced from $0.6s$ to $0.1s$.

3. **Development of the Musical-based Interaction System (MbIS)** - To give the robot the possibility to interact with its partner musicians, I developed an interaction system, that is to allow human-like usability of complex robot hardware. The interaction system is able to processes information from multi-modal communication channels. Within the research focus of this thesis, visual and aural sensor processing was implemented. The input from the sensor processing system is mapped to musical performance parameters of the robot using two different strategies. Which one of these strategies is being used, can be selected by the user, depending on the experience level of the user.

4. **Detection of visual cues** - In the application proposed in this thesis I focused on visual and acoustic communication. The visual information is pre-processed before being sent to the mapping module, where it is combined with information from the acoustic processing to modify the performance parameters of the flutist robot. When musicians interact with each other in a band environment, they use visual as well as aural cues to synchronize their performance. The most common visual cue performed especially by wind instrument players, are instrument gestures like leaning the instrument

into the direction of another player. I developed the visual processing systems (motion perception-based tracking and particle filter-based tracking) to be able to detect this kind of movement.

5. **Detection of acoustic cues** - As aural cues musicians mainly use commonly known melody and rhythm patterns. If one musician plays one of these known patterns, the other musicians in the band are informed about the current song context. In this thesis I proposed histogram-based melody and rhythm-based tracking, to find such known patterns within a sequence played by a partner musician of the robot.

6. **Evaluation of the MbIS** In the experiments section of the thesis I evaluated the overall performance of the system in two different ways. First, the system overall functionality was technically validated. This achieved not on a sub-module level, but using the system as a whole. Therefore the input was also provided by a real musician user, to confirm that the system is suitable to work in a realistic context. Second, I performed a survey to evaluate the system from a user perspective. As a result of this survey, compared to the basic interaction level the extended interaction level was attributed a higher score for adjective pairs *natural / artificial*, *free movement / constrained*, *emotional / rational*, *expressive / unexpressive*. The results, that were reached in the user survey show, that the general concept of the proposed system points into the right direction. To a certain degree, the overall hypothesis I proposed in this thesis has been confirmed. The proposed perception-action scheme-based, human-robot interaction system is suitable to enable natural, task-oriented collaboration with an anthropomorphic robot, given the controlled experimental environment described in the previous sections.

7. **Application to a general HRI context** - The Musical-based Interaction System (MbIS) was designed with the purpose to create an interaction system that is generally applicable to a general Human-Robot-Interaction context. The experimental results show, that the system works in a musical

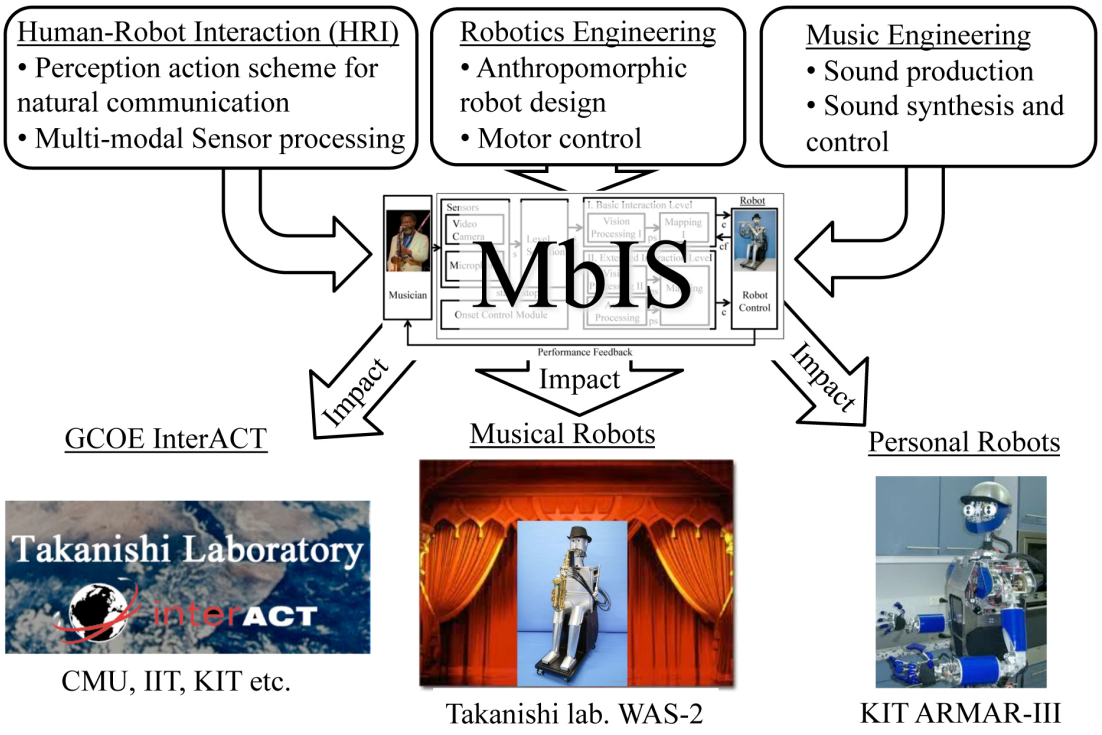

Figure 7.1: The figure shows an overview of the influences from various fields, that are incorporated by the musical-based interaction system (inputs into the MbIS unit). In the lower part of the figure, possible further impact areas of the developed concept are displayed.

context. Although the musical application is a specialized case of human-robot-interaction, it contains many characteristics (such as communication through different perception channels etc.) that occur also in other types of such interaction. This allows the consideration, that a similar system with slight modification might also be applied to other cases of personal robotics.

This research, due to the interdisciplinary approach among human-robot interaction, music engineering and robot engineering as well the potential application of the novel perception-action scheme to other musical robotic platforms, has the intention to contribute to the better understanding of the basic requirements to enhance the human-robot musical interaction. In particular, further possible

applications of the proposed perception-action scheme to other different kinds of entertainment robots with slightly modifications are expected. In order to better understand possible strategies to modify the behavior of the robot to further strengthen the human-robot interaction, a teach-in method has been conceived by implementing a memory model, which allows the robot to save a correlation between a melody pattern that is performed by the musician and the musician's instrument movement. In this thesis try the knowledge of mechanisms of musical interaction and to contribute to the previously mentioned fields as well as to other related fields such as kansei engineering, communication science, music informatics. An overview over possible impacts areas of the musical-based interaction system is shown in Fig. 7.1. As one possible further application, the Global Center of Excellence (GCOE) and the interACT program are shown. These are both international collaboration programs, in which active exchange of knowledge between research institutions is encouraged. Within these frameworks, the proposed interaction system might be applied to a wide range of other personal robot systems.

7.2 Future Work

In this final section, I would like to propose several improvements to the proposed interaction system. With these improvements the functionality of the MbIS could be improved to enhance the performance of the system and to make it suitable for a wider range of application in personal robotics.

1. **Audio processing algorithms** - Generally regarding all proposed sensor processing components, I would like to do further work to enhance the algorithms and improve the robustness of sensor processing. Regarding the audio processing, the pattern library size is now limited to three melody patterns. I would like to extend the size of this library to, depending on the musical material used, allow for more distinguished detection. The low level routines of the audio processing also leave room for improvement. In future works, I would like to improve the implemented algorithm to do faster pitch detection. When using the onset detection system, I detected

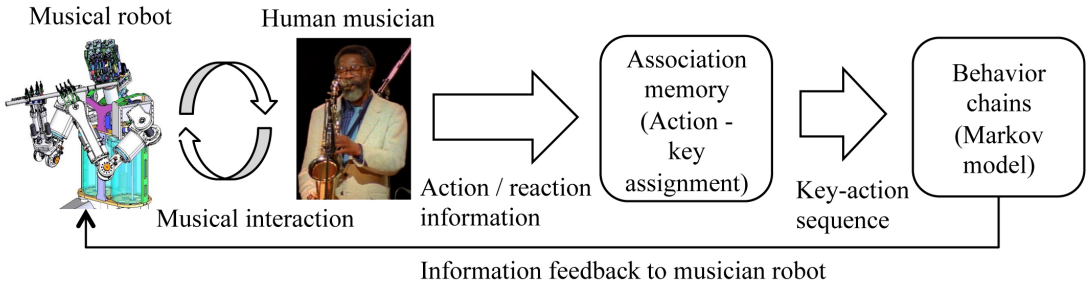

Figure 7.2: For future development a system for online-learning of the performance behavior of the human musician is planned.

a time difference between the detection of the onset control signal and the actual performance start ($\Delta t = 0.1s$). This might be due to the slow speed of the audio processing.

2. **WF-4RV performance quality** - The performance quality of WF-4RV using the improved lung control algorithm has been qualitatively evaluated on a preliminary scale. In further experiments and analysis the improvement of the note-to-note transition quality should be more accurately recorded and compared to the previous sound quality. An evaluation function might be employed to numerically express the improvement. So far the use of an additional mechanical valve mechanism has been considered, but not implemented. In further research such a **speak-on** valve could be added to the robot, to provide even better air-flow envelope control.

3. **Further evaluation of the MbIS** - Qualitative and quantitative evaluation of the interaction system have been performed in various ways. Nevertheless I would like to do further evaluation of the proposed method. I would like to apply the system not only to jazz, but also to other musical styles. A classical piece of music (*Ave Verum Corpus* by Wolfgang Amadeus Mozart) to which the MbIS might be applied has recently been suggested by the professional flutist, that I worked together with for the research in this thesis.

4. **Extension of the mapping modules** - I would like to enhance the functionality of both, the basic interaction level mapping module as well as the extended level mapping module. In the basic level, so far only the lung fill level status has been used as feedback from the physical state of the robot. In future work I would like to uses further feedback channels from the mechanical structure of the robot. Furthermore I would like to extend the teach-in approach of the extended level interaction system to enable the flutist robot to actively learn the performance behavior patterns of its partner musician.

5. **Application to realistic performance environments** - So far the musician robot has mainly provided background accompaniment for the improvisation of the musician. In future works I would like to extend the system in order to allow the flutist robot to take the lead performance as well. The original goal of the Musical-based Interaction System is to allow an interactive performance in front of an audience, on a stage. Because the robot cannot be easily transported, I have so far not advanced to that level, but I would like to test the system under realistic circumstances on a real stage.

6. **Application to general HRI context** - From a more general HRI context point-of-view, I would like to extend the application of the system to other musician robots. The design of the interaction system is a general purpose concept. Therefore, I would like to attempt to implement the system with minor modifications to further personal robotics platforms.

Appendix A

Microphone and Camera Specifications

A.1 Preamplifier A3 Characteristics

Input	impendance 1 kOhm
Output	impedance 47 kOhm
Frequency response	20 Hz - 50 kHz (+0 / - 3dB)
Frequency response with filter	300 hz - 50 kHz (+0 / -3 dB)
Noise Level (A-weigthed)	47 kOhm loaded max 30 V
Max. output voltage	1,5 V eff.
Max. output voltage with ATT.	1.0 V eff
Battery	6 V (4LR44 o.)
Battery life time	50 h (OKM I), 100 h (OKMII)

A.2 OKM II Characteristics

Frequency range	20 Hz to 20 kHz
Channel Balance	typ. ¡ 1,0 dB
S/N ratio	re 1 Pa approx. 61 dB
Max. SPL (with A 3 Adapter)	108 dB
Max SPL (with A 3 Adapter 20 dB att.)	125 dB
Operating voltage	max. 10 V

A.3 Firefly Camera (FFMV-03M2C)

Image Sensor Type	1/3 progressive scan CMOS
Image Sensor Type	Global shutter using Micron TrueSNAP technology
Image Sensor Model	Micron MT9V022
Maximum Resolution	752 (H) x 480 (V)
Pixel Size	6.0 m x 6.0 m
Analog-to-Digital Converter	On-chip 10-bit ADC
Video Data Output	8 and 16-bit digital data
Image Data Formats	Y8, Y16 (monochrome), 8-bit and 16-bit raw Bayer data (color models)
Digital Interface	6-pin IEEE 1394a for camera control, video data, power
Transfer Rates	400 Mb/s
Maximum Frame Rate	752x480 at 61 FPS, 320x240 at 112 FPS (region of interest) 320x240 at 122 FPS (2 x 2 pixel binning)
Partial Image Modes	pixel binning and region of interest modes via Format7
General Purpose I/O Ports	7-pin JST GPIO connector, 4 pins for trigger and strobe, 1 pin +3.3 V, 1 Vext pin for external power
Gain Control	automatic / manual, 0 dB to 12 dB
Shutter Speed	automatic / manual, 0.12 ms to 512 ms
Gamma	0 to 1 (enables 12-bit to 10-bit companding)
Synchronization	via external trigger, software trigger, or free-running*
External Trigger Modes	IIDC v1.31 Trigger Modes 0 and 3
Power Requirements	8 to 30 V via IEEE-1394, less than one (1) Watt
Dimensions (LxWxH)	24.4 x 44 x 34 mm
Mass	37 g (including tripod adapter)
Camera Specifications	IIDC 1394-based Digital Camera Specification v1.31
Memory Storage	three memory channels for user configurable power-up settings
Lens Mount	CS-mount (5mm C-mount adapter included) M12 microlens mount2
Compliance	CE, FCC Class B, RoHS
Operating Temperature	0 to 45C
Storage Temperature	-30 to 60C

References

[1] J. Fordham, "Jazz," *DK Publishing (Dorling Kindersley), 1993*, 1993.

[2] R. B. Rusu, G. Bradski, R. Thibaux, and J. Hsu, "Fast 3d recognition and pose using the viewpoint feature histogram," in *International Conference on Intelligent Robots and Systems*, (Taipei, Taiwan), 2010.

[3] T. Sugaiwa, Y. Yamaguchi, H. Iwata, and S. Sugano, "Dexterous hand-arm coordinated manipulation using active body-environment contact," in *International Conference on Intelligent Robots and Systems*, pp. 4168 –4173, 2009.

[4] T. Asfour, F. Gyarfas, P. Azad, and R. Dillmann, "Imitation learning of dual-arm manipulation tasks in humanoid robots," in *International Conference on Humanoid Robots*, pp. 40 – 47, 2006.

[5] T. Hester, M. Quinlan, and P. Stone, "Generalized model learning for reinforcement learning on a humanoid robot," in *International Conference on Robotics and Automation*, pp. 2369 –2374, 2010.

[6] H. Ishiguro, T. Ono, M. Imai, T. Maeda, T. Kanda, and R. Nakatsu, "Robovie: an interactive humanoid robot," *The Industrial Robot Journal*, vol. 28, pp. 498 – 504, 2001.

[7] Y. Kusuda, "Toyota's violin-playing robot," *The Industrial Robot Jornal*, vol. 35, pp. 504–506, 2008.

[8] T. Takeda, K. Kosuge, and Y. Hirata, "Hmm-based dance step estimation for dance partner robot ms dancer," in *International Conference on Intelligent Robots and Systems*, pp. 3245 – 3250, 2005.

[9] I. Kato, "The robot musician wabot-2," *Robotics, Vol. 3(2), pp. 143-155*, 1978.

[10] J. Solis, K. Chida, K. Suefuji, K. Taniguchi, S. Hashimoto, and A. Takanishi, "The waseda flutist robot wf-4rii in comparison with a professional flutist," *Computer Music Journal*, vol. 30, pp. 127–151, 2006.

[11] J. Solis, K. Taniguchi, T. Ninomiya, and A. Takanishi, "Understanding the mechanisms of the human motor control by imitating flute playing with the waseda flutist robot wf-4riv," *Mechanism and Machine Theory (Special Issue on Bio-Inspired Mechanism Engineering)*, vol. 44, pp. 527–540, 2008.

[12] J. Solis, K. Suefuji, K. Taniguchi, and A. Takanishi, "Towards an autonomous musical teaching system from the waseda flutist robot to flutist beginners," in *International Conference on Intelligent Robots and Systems - Workshop: Musical Performance Robots and Its Applications*, pp. 24–29, 2006.

[13] A. Kapur, "A history of robotic musical instruments," in *International Conference on Computer Music*, pp. 10–15, 2005.

[14] E. Singer, K. Larke, and D. Bianciardi, "Lemur guitarbot: Midi robotic string instrument," *Conference on New Interfaces for Musical Expression*, pp. 3188–3191, 2003.

[15] E. Singer, "Lemurs musical robots," in *Conference on New Interfaces for Musical Expression*, pp. 181–184, 2004.

[16] M. Kajitani, "Development of musician robots in japan," in *Australian Conference on Robotics and Automation*, 1999.

[17] K. Shibuya, "Analysis of human kansei and development of a violin playing robot," *Conference on Intelligent Robots and Systems - Workshop: Musical Performance Robots and Its Applications*, pp. 13–17, 2006.

[18] K. Kobayashi and S. Takashima, "The control of an automatic performance robot of trumpet - development of lip-tension control by using pressing-roller," in *Robotics Mechatronics Workshop*, pp. 2–6, 2002.

[19] T. Miyawaki and S. Takashima, "Control of an automatic performance robot of saxophone-performance control using standard midi files," in *Conference on Mechatronics and Robotics of the Robotics Society of Japan*, 2003. no. 1P1.1F.A2(1).

[20] N. Orio, S. Lemouton, and D. Schwarz, "Score following: State of the art and new developments," in *Conference on New Interfaces for Musical Expression*, pp. 36–41, 2003.

[21] G. Weinberg and S. Driscoll, "Towards robotic musicianship," *Computer Music Journal*, vol. 30, pp. 28–45, 2006.

[22] J. Solis and A. Takanishi, "Imitation of human flute playing by the anthropomorphic flutist robot wf-4rii," *The Computer Music Journal*, vol. 30, pp. 12–24, 2006.

[23] J. Solis, K. Chida, K. Suefuji, and A. Takanishi, "The development of the anthropomorphic flutist robot at waseda university," *International Journal of Humanoid Robots (IJHR)*, vol. 3, pp. 127–151, 2006.

[24] J. Solis and A. Takanishi, "Implementation of expressive performance rules on the wf-4riii by modeling a professional flutist performance using nn," *International Conference on Robotics and Automation*, pp. 2252–2557, 2007.

[25] J. Solis, K. Taniguchi, T. Ninomiya, K. Petersen, T. Yamamoto, and A. Takanishi, "Implementation of an auditory feedback control system on an anthropomorphic flutist robot inspired by the performance of a professional flutist," *Advanced Robotics Journal*, vol. 23, pp. 1849–1871, 2009.

[26] J. Solis, K. Taniguchi, T. Ninomiya, T. Yamamoto, and A. Takanishi, "Development of waseda flutist robot wf-4riv: Implementation of auditory feedback system," *International Conference on Robotics and Automation*, pp. 234–239, 2008.

[27] Kuraray Co., "Speton high performance thermoplastic rubber," 2010. [Online; accessed 27-July-2010].

[28] P. Webster and D. Williams, "Experiencing music technology: Software, data, and hardware," *Thomson Schirmer*, pp. 221–222, 2005.

[29] Wikipedia, "Midi (musical instrument digital interface)," 2010. [Online; accessed 27-July-2010].

[30] ZeroC, "Ice (internet communications engine," 2010. [Online; accessed 27-July-2010].

[31] Willow Garage, "Ros (robot operating system)," 2010. [Online; accessed 27-July-2010].

[32] S. Goto, "The case study of an application of the system 'bodysuit and robotmusic: Its introduction and aesthetics'," *International Conference on New Interfaces for Musical Expression*, pp. 292–295, 2006.

[33] F. Weiss, "Lob der anwesenheit - dance performance." Installation at Ludwigsforum Aachen, Germany, 2001.

[34] J. Deutscher, A. Blake, and I. Reid, "Articulated body motion capture by annealed particle filtering," in *Conference on Computer Vision and Pattern Recognition*, pp. 2126–2131, 2000.

[35] L. Hasan, N. Yu, and A. Paradiso, "The thermenova: A hybrid free-gesture interface," in *Conference on New Instruments for Musical Expression*, pp. 1–6, 2002.

[36] M. Wanderley and M. Battier, "Trends in gestural control of music," tech. rep., IRCAM Centre Pompidou, France, 2000.

[37] T. Winkler, "Making motion musical: Gestural mapping strategies for interactive computer music," in *International Conference of Computer Music*, pp. 261–264, 1995.

[38] D. Rokeby, "A very nervous system." Installation at Arte, Technologia e Informatica, Venice Biennale, Venice, Italy, 1986.

[39] A. Yilmaz, "Object tracking: A survey," *ACM Computing Surveys*, vol. 38, 2006. no. 13.

[40] M. Bray, E. Koller-Meier, and L. V. Gool, "Smart particle filtering for 3d hand tracking," in *Conference on Automatic Face and Gesture Recognition*, pp. 675–680, 2004.

[41] Sony Computer Entertainment, "Playstation eyetoy," 2008. [Online; accessed 27-July-2010].

[42] I. Richardson, "H. 264 and mpeg-4 video compression: Video coding for next-generation multimedia," *as book by Wiley Publishing*, 2003.

[43] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfinder: Real-time tracking of the human body," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 780–785, 1997.

[44] Akai Professional Japan, "Mpc 2000," 2010. [Online; accessed 27-July-2010].

[45] A. C. Doucet A., Godsill S., "On sequential monte carlo sampling methods for bayesian filtering," *Statistics and Computing*, vol. 10, pp. 197–208, 2000.

[46] A. Leon-Garcia, "Probability and random processes for electrical engineering," *Addison Wesley*, 1994.

[47] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear / non-gaussian bayesian tracking," *IEEE Transactions on Signal Processing*, pp. 174–188, 2002.

[48] van der Merwe R., de Freitas N., D. A., and W. E., "The unscented particle filter," *Presentation in Proceedings Advanced Neural Information Processing System Signal Processing*, 2000.

[49] K. Nummiaro, E. Koller-Meier, and L. V. Gool, "An adaptive color-based particle filter," *Journal of Image and Vision Computing*, vol. 21, pp. 99–110, 2003.

[50] R. F. David Saxe, "Toward robust skin identification in video images," *International Conference on Automatic Face and Gesture Recognition, pp. 379-384*, 1996.

[51] A. Klapuri, "Automatic transcription of music," Master's thesis, Audio Research Group, University of Tampere, Finland, 1998.

[52] S. Dixon, "Monte carlo methods for tempo tracking and rhythm quantization," *Journal of New Music Research*, vol. 30, pp. 39–58, 2001.

[53] A. Cemgil and B. Kappen, "Monte carlo methods for tempo tracking and rhythm quantization," *Journal of Artical Intelligence Research*, vol. 18, pp. 45–81, 2003.

[54] J. Bello, G. Monti, and M. Sandler, "An implementation of automatic music transcription of monophonic music with a blackboard system," *Irish Signals and Systems Conference*, pp. 217–223, 2000.

[55] J. Bello, *Towards the Automated Anaylsis of Simple Polyphonic Music: A Knowledge Based Approach.* PhD thesis, Queen Marys University, London, 2003.

[56] D. Martin, "A blackboard system for automatic transcription of simple polyphonic music," tech. rep., Media Laboratory, MIT, 1996.

[57] D. Godsmark and G. Brown, "A blackboard architecture for computational auditory scene analysis," *Speech Communication*, vol. 27, pp. 351–366, 1999.

[58] J. Bello and M. Sandler, "Blackboard system and top-down processing for the transcription of simple polyphonic music," *Digital Audio Effects Workshop (DAFx)*, pp. 55–64, 2000.

[59] G. Monti, "Signal processing and music analysis," tech. rep., Department of Electronic Engineering, Queen Marys, University London, 2000.

[60] M. Plumbley, S. Abdallah, J. Bello, M. Davies, G. Monti, and M. Sandler, "Automatic music transcription and audio source separation," *Cybernetics and Systems*, vol. 33, pp. 603–627, 2002.

[61] K. Kashino and H. Murase, "Music recognition using note transition context," *International Conference on Advanced Systems and Signal Processing*, vol. 6, pp. 3593–3596, 1998.

[62] K. Kashino and H. Murase, "A sound source identication system for ensemble music based on template matching and music stream extraction," *Speech Communication*, vol. 27, pp. 337–349, 1999.

[63] P. Walmsley, *Signal Separation of Musical Instruments - Simulation-based Methods for Musical Signal Decomposition and Transcription.* PhD thesis, Cambridge University Engineering Department, 2000.

[64] P. Walmsley, S. Godsill, and P. Rayner, "Bayesian graphical models for polyphonic pitch tracking," *Diderot Forum*, pp. 1–26, 1999.

[65] P. Walmsley, S. Godsill, and P. Rayner, "Multidimensional optimisation of harmonic signals," *European Signal and Image Processing Conference*, pp. 2033–2036, 1998.

[66] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Transactions on Speech and Audio Processing*, vol. 10, pp. 293–302, 2002.

[67] G. Tzanetakis and P. Cook, "A quick search method for audio and video signals based on histogram pruning," *IEEE Transactions on Multimedia*, vol. 5, pp. 348–357, 2003.

[68] G. Tzanetakis, A. Ermolinskyi, and P. Cook, "Pitch histograms in audio and symbolic music information retrieval," *Journal of New Music Research*, vol. 32, pp. 143–152, 2003.

[69] K. Sugiura and N. Iwahashi, "Learning object-manipulation verbs for human-robot communication," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2234–2240, 2008.

[70] T. Suzuki, S. Yano, and K. Suzuki, "Motivation oriented action selection for understanding dynamics of objects," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 846–851, 2008.

[71] H. Lee, H. Kim, K. Park, and J. Park, "Robot learning by observation based on bayesian networks and game pattern graphs for human-robot game interactions," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 319–325, 2008.

[72] R. Kurazume, H. Yamada, K. Murakami, Y. Iwashita, and T. Hasegawa, "Target tracking using sir and mcmc particle filters by multiple cameras and laser range finders," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3838–3844, 2008.

[73] S. Calinon and A. Billard, "A probabilistic programming by demonstration framework handling constraints in joint space and task space," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 367–372, 2008.

[74] L. Rabiner and B. Gold, "Theory and application of digital signal processing," *Englewood Cliffs, New Jersey: Prentice-Hall, Inc.*, 1975.

[75] C. Bartneck, D. Kulic, and E. Croft, "Measuring the anthropomorphism, animacy, likability, perceived intelligence, and perceived safety of robots," in *Workshop on Metrics for Human-Robot Interaction (HRI2008)*, pp. 37–43, 2008.