ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF ARTS AND SOCIAL SCIENCES

INTRODUCING MEMO: REAL TIME SOUND SYNTHESIS AS A TOOL FOR INTERACTIVE MUSIC SYSTEM DEVELOPMENT

M.A. THESIS

Mehmet KORKMAZ

Music M.A. Programme

JULY 2020

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF ARTS AND SOCIAL SCIENCES

INTRODUCING MEMO: REAL TIME SOUND SYNTHESIS AS A TOOL FOR INTERACTIVE MUSIC SYSTEM DEVELOPMENT

M.A. THESIS

Mehmet KORKMAZ
(409171017)

Music M.A. Programme

Thesis Advisor: Assistant Prof. Gökhan DENEÇ
Thesis Co-Advisor: Dr. Konstantinos VASILAKOS

JULY 2020

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ SOSYAL BİLİMLER ENSTİTÜSÜ

MEMO: EŞ ZAMANLI SES SENTEZİ KULLANAN ETKİLEŞİMLİ DOĞAÇLAMA SİSTEMİ

YÜKSEK LİSANS TEZİ

Mehmet KORKMAZ
(409171017)

Dr. Erol Üçer Müzik İleri Araştırmalar Merkezi

Müzik Yüksek Lisans Programı

Tez Danışmanı: Dr. Öğr. Üyesi Gökhan DENEÇ
Eş Danışman: Dr. Konstantinos VASILAKOS

TEMMUZ 2020

Mehmet Korkmaz, an M.A. student of ITU Graduate School of Arts and Social Sciences with 409171017 student ID, successfully defended the thesis/dissertation titled "INTRODUCING MEMO: REALTIME SOUND SYNTHESIS AS A TOOL FOR INTERACTIVE MUSIC SYSTEM DEVELOPMENT" which was prepared after having fulfilled the requirements specified in the associated legislations, before the jury whose signatures are stated below.

Thesis Advisor: **Assistant Prof. Gökhan Deneç** ....
Istanbul Technical University

Co Advisor : **Dr. Konstantinos Vasilakos** ....
Istanbul Technical University

Jury Members: **Prof. Dr. Can Karadoğan** ....
Istanbul Technical University

Assoc. Prof. Barkın Engin ....
Bahcesehir University

Assoc. Prof. Burak Tamer ....
Bahcesehir University

Date of Submission : 15 June 2020
Date of Defense : 16 July 2020

Ode to the victims of COVID-19.

FOREWORD

I would like to express my sincere appreciation to my advisor Assistant Prof. Gökhan Deneç, my co-advisor Dr. Konstantinos Vasilakos, Dr. Jeremy Woodruff and Dr. Reuben De Lautour for their encouragements and supports throughout the period of writing this thesis. Their guidance had been the most useful.

Also, I would like to express my deepest gratitude to my friend Bartu Çankaya, who helped me with his insights and ideas during the development process of my novel improvisation system *MEMO*.

July 2020 — Mehmet Korkmaz
(Sound Artist & Performer)

TABLE OF CONTENTS **Page**

ABBREVIATIONS

AM : Amplitude Modulation
CA : Cellular Automata
GUI : Graphical User Interface
IRCAM : Institute for Research and Coordination in Acoustics and Music
MIDI : Musical Instrument Digital Interface
MC : Multi Channel
NDWR : Norwestdeutscher Rundfunk
RTF : Radiodiffusion Télévision Française

LIST OF FIGURES

Page

INTRODUCING MEMO: REAL TIME SOUND SYNTHESIS AS A TOOL FOR INTERACTIVE MUSIC SYSTEM DEVELOPMENT

SUMMARY

This thesis is concerned with the development and the roles of my interactive music system *MEMO* that was developed in Max software. Using the sound synthesis methods that I replicated in a novel and a customized way, such as live sampling, granular synthesis, time stretching, and embodying a statistical algorithm called game of life, *MEMO* is a system for real-time sonic deviation, which is responsive to the performer's output and it is able to engage in a dialogue with the performer or can be used solely as an instrument as well. In this thesis, with a practice-based research approach undertaken, I have reflected my own process of programming and explained the internal dynamics of *MEMO* by indicating the patching process in Max software and referring to the theories of interactive music systems as well as sound synthesis methods. This thesis has three main research concerns:

- Determination of *MEMO's* stance in the current literature of interactive music systems and defining the relationships with the selected novel music systems.
- The implementation of real-time sound synthesis methods as a model for interactive music systems,
- Developing scenarios to control the system with fluency,
- Development of a novel improvisation system and offer a self-reflective approach for designing interactive music systems.

Throughout the course of this thesis, the background that led to this research, my aims as a performer and developer, the key theories of interactivity as well as the degrees that interactivity can occur, the evolution and a brief history of the sound synthesis methods that I have used in my system, my main methodological approach, the aesthetic need that this system addresses, the roles of *MEMO* and the technical process of developing *MEMO* are discussed in detail. Also, this thesis will conclude with a discussion about the future directions that *MEMO* can embody.

MEMO: EŞ ZAMANLI SES SENTEZİ KULLANAN ETKİLEŞİMLİ DOĞAÇLAMA SİSTEMİ

ÖZET

Bu tez, Max isimli müzik ve görsel programlama yazılımında geliştirilen etkileşimli müzik doğaçlama ve performans sistemi *MEMO'*nun kurgulanmasını ve üstlenebileceği rolleri konu almaktadır. Özgün bir şekilde kurgulanan, eş zamanlı örnekleme, granüler sentezleme, zaman gerimi ve istatistiki bir algoritma olan "game of life" kavramını kullanan bir sistem olan *MEMO,* eş zamanlı sessel yineleme gerçekleştiren ve performansçının çıktısına tepki veren ve aynı zamanda enstrüman olarak da kullanılabilen bir etkileşimli müzik sistemdir. Pratik bazlı araştırma yaklaşımı üstlenilen bu tezde, programlama sürecine, *MEMO*'nun iç dinamiklerine ve aynı zamanda etkileşimli müzik sistemleri ve ses sentezleme teorilerine değinildi.
Bu tez süresince, bu araştırmaya yol açan sebeplerden, bir performansçı ve geliştirici olarak bu bağlamdaki hedeflerimden, etkileşim teorilerinden, ses sentezleme yöntemlerinin evriminden ve tarihçelerinden, ana metodolojik yaklaşımımdan, bu sistemin ele aldığı estetik ihtiyaçtan, *MEMO*'nun rollerinden ve aynı zamanda teknik gelişim sürecinden ayrıntılı olarak bahsedilmiştir. Bu tezde temel amaç, doğaçlama performanslar için tasarlanan ve etkileşimli bir müzik sistemi olan aynı zamanda gerçek zamanlı ses sentez yöntemlerini kullanan ve eş zamanlı örneklemeye dayanan *MEMO*'nun yaratıcı gelişim süreçlerini yansıtmaktır. Bu tezden en iyi verimi alabilmek için ses sentezleme yöntemleri ve Max yazılımı ile ilgili aşinalık esastır. Sistemin geliştirilme sürecinin yanı sıra, etkileşimli müzik sistemlerinin kavramları ve rolleri ile ilgili temel teoriler açıklanmış ve bu teoriler icracı / geliştirici olarak kendi pratiğimle ilişkilendirilmiştir. Bu tezdeki ana araştırma yöntemi, *MEMO*'nun estetik altyapısına, teknik sorunlarına ve geliştirme sürecine ışık tutan, uygulamaya dayalı, kendini yansıtıcı bir araştırma metodolojisine dayanmaktadır. Bu araştırmanın beklenen katkıları aşağıdaki gibidir;
- *MEMO*'nun interaktif müzik sistemlerinin mevcut akademik ortamındaki duruşunun belirlenmesi ve *MEMO* ile seçilen yeni interaktif müzik sistemleri arasındaki ilişkinin tanımlanması.
- Ses sentezi yöntemlerinin, icracı ve bilgisayar arasında bir diyalog izlenimi elde etmek amacıyla ve etkileşimli müzik sistemi geliştirilmesi için bir model olarak uyarlanması.
- Bu sistemin kontrolü açısından özerklik ve bağımlılık arasındaki ideal dengenin bulunması.
- Etkileşimli müzik sistemi geliştiriciliği için öz yansıtan bir yaklaşımın sunulması.
Bu araştırma, özgün interaktif müzik sistemi *MEMO'*nun gelişimini ve tasarım sürecini yansıtarak literatüre katkı sağlamaktadır. Metodoloji bölümünde, bu süreç temelde yatan estetiklerden başlayarak Max yazılımındaki işlemlerine kadar aşamalarıyla açıklanmıştır. Ayrıca, yapılan literatür araştırması neticesinde *MEMO* bir sistem olarak etkileşimlilik açısından Robert Rowe'un öne sürdüğü score/performance-driven modeli, Todd Winkler'ın conductor modeli ve Joel Chadabe'nin fly-by wire modelleriyle ilişkilendirilmiştir. Ayrıca, etkileşimlilik kavramının tanımının öznelliği ve bu konudaki farklı görüşler tartışılmıştır. Bunların yanı sıra *MEMO,* Giacomo Lepri'nin *InMuSIC* sistemi, IRCAM tarafından geliştirilen *OMax* sistemi*, Ben Carey'nin _derivations* sistemi, Michael Musick, Jonathan P. Forsyth ve Rachell Bittner tarafından geliştirilen *The Harmonically Ecosystemic*

Machine; Sonic Space No. 7 sistemi, François Pachet'in *Continuator* sistemi ve Rodrigo Constanzo'nun *The Party Van* sistemiyle çalışma prensipleri açısından karşılaştırılmıştır. Bu karşılaştırmaların sonucu olarak *MEMO*, temel olarak eş zamanlı ses sentezi ve eş zamanlı sampling kullanımlarından dolayı en çok Rodrigo Constanzo'nun *The Party Van* ve Ben Carey'nin *_derivations* sistemleriyle ilişkilendirilmiştir. Ek olarak,eEstetik açıdan *MEMO*'nun geniş bir yelpazede müzikal tarz sunduğu tartışılmıştır. Bu müzikal tarzlar, minimalist, gamelan, ambient ve elektro-akustik olarak tanımlanmıştır. *MEMO*'yu oluşturan her bir modül için estetik eşleştirme yapılmış, hafıza modülünün, kontrpuan bazlı eşlik yarattığından, granüler sentezleme modülünün zengin ses dokuları oluşturduğundan, game of life modülünün evrimsel yapısı nedeniyle sürpriz ritimler oluşturduğundan, poliritim sentezleme modülünün Gamelan müziğinde, görülen çoklu ritimleri oluşturduğundan ve zaman gerimi modülünün ise yavaş gelişen uzun ses dokuları yarattığından bahsedilmiştir. Ayrıca, kullanımı açısından *MEMO*'nun iki farklı role sahip olduğu anlaşılmıştır. Bu roller; enstrüman ve eşlikçi olarak açıklanabilir. *MEMO,* herhangi bir enstrümandan eş zamanlı ses girdisi almayarak, hafızasına ses yüklenerek bu sesi sentezleyen bir enstrüman olarak kullanılabilir. Bu kullanım MIDI kontrolörüne yapılan parametre eşleştirmeleri sayesinde kolaylaştırılabilir. *MEMO*'nun eşlikçi olarak kullanımı ise, icracının enstrüman çıktısını *MEMO*'nun girdisine göndermesiyle ve *MEMO*'nun da eş zamanlı olarak aldığı sesleri o anda sentezleyerek değişime uğratmasıyla sağlanabilmektedir. Bu etki tepki süreci bir diyalog izlenimi yaratmaktadır. Ayrıca MIDI kontrolörü sayesinde icracı her an sisteme müdahale edebilmekte ve performansın akışını şekillendirebilmektedir. Kontrol senaryosu olarak iki senaryo geliştirilmiş olup, enstrüman modundayken hafıza uzunluğunun uzun tutulmasının müzikal çeşitliliği artırdığı gözlemlenmiştir. Ayrıca eşlikçi modundayken, hafıza modülünün en fazla sekiz saniye olarak ayarlanmasının sistemle icracının etkileşim ve diyalog şansını artırdığı gözlemlenmiştir. *MEMO*, Hafıza Modülü, Granüler Sentezleme Modülü, Game of Life Modülü, Poliritmik Sentez Modülü, Zaman Gerimi Modülü ve Granüler Geciktirme Modülü ile toplam altı ses sentezleme yönteminin bir araya gelmesiyle oluşmuş bir sistemdir. Bu modüller obje bazında Max yazılımında özgün bir şekilde geliştirilmiştir. Sistemdeki sinyal akışı paralel zincirlemeyle sağlanmıştır. Sinyal ilk olarak Hafıza modülüne gitmekte sonrasında Granüler Sentezleme, Zaman Gerimi, Poliritmik Sentez Modülü ve Game of Life modüllerine paralel olarak iletilmektedir. Bunlara ilave olarak Game of Life ve Granüler Sentezleme modülleri ses çıkışlarını Granüler Geciktirme modülüne iletmektedir.

MEMO yapısı gereği bir makine öğrenim yöntemi gibi entelektüel bir otomatik öğrenim metodu içermediğinden, *MEMO*'nun etkileşimlilik derecesinin bazen şansa kalabildiği anlaşılmıştır. Bu şans unsuru, performans sırasında hem pozitif hem de negatif etki sağlayabilir. Pozitif etkisi, icracıya sürprizler yaratarak doğaçlamayı şekillendirebilmesi, negatif etkisi ise performans anında icracının karşılaştığı senaryoyu telafi etmek zorunda kalabilmesi olarak tanımlanmıştır.

MEMO'nun temelde sunduğu her şey, icracının doğaçlama materyalinin bir yansıması ve yeniden düzenlenmesi niteliğindedir. Gelecekteki araştırmalarda, tamamen özerk bir karar verme sisteminin entegrasyonu, bilişsel bir süreç olarak doğaçlamanın incelenmesi ve makine öğrenimi algoritmaları ile insan beynindeki doğaçlama sürecinin dijital mecrada taklit edilmesiyle gerçekleştirilebilir. Bu fikirle, *MEMO'*daki etkileşim derecesi büyük ölçüde geliştirilebilir. Makine öğreniminin entegrasyonunun yanı sıra, bu çalışmanın gelecekte ele alabileceği bir başka yönü ise, *MEMO*'nun kendini yenileyen ekosistemik algoritmalarındaki modüllerin geliştirmeleridir. Bu geliştirmeler, *MEMO*'nun tamamen icracı lehine çalışan itaatkar bir sistem haline geleceği ya da tam tersi, dinleyen, tepki veren ve kendi malzemesini yaratan tamamen özerk bir diğer müzisyen haline gelebileceği şekilde gerçekleştirilebilir.

MEMO'nun yaratım ve tasarlama sürecini yansıtmak, bu sistemin rollerine ve karşılaşılan teknik konulara değinilen bu tezde temel olarak etkileşimli müzik sistemi geliştirme literatürüne kişisel bir bakış açısı katma amacıyla katkıda bulunulmaya çalışılmıştır.

1.INTRODUCTION

In this thesis, the main aim is to reflect the creative processes of development of *MEMO* which is an interactive music system for improvisation and based on live-sampling that uses real-time sound synthesis methods. Some familiarity with regards to audio synthesis methods and Max software is required to get the best out of this thesis. Apart from the process of development, the key theories will be explained, with regards to the concepts and the roles of interactive music systems and relate these theories with my practice as a performer/developer. The main research method in this thesis will be based on a practice based, self-reflective research methodology that will shed light on the aesthetic reasons, technical issues and the process of development of *MEMO*. The expected contributions of this research are as follows;

- Determining *MEMO's* stance in the current academic landscape of interactive music systems as well as defining the relationship between *MEMO* and the selected novel interactive music systems.
- Adaptation of sound synthesis methods as a model for interactive music system development to obtain an impression of a dialogue between the performer and the computer.
- Finding the ideal balance between the autonomy and dependence with regards to the control of this system.
- Offering a self-reflective approach for interactive music system development.

As a performer with electric guitar and electronics I have been inspired by musicians such as Robert Fripp, Bill Frisell and Brian Eno, who experimented with sampling techniques and created new ways of performing using tape machines, samplers and various other effect units. Having experimented with the concept of live sampling for a long time, I used myriad of pedals and effect units in my solo performances to replicate the experiments of above-mentioned musicians. With this initial practice, I decided to develop a system in Max Software which is a programming environment for signal processing, to use the computer as a tool for improvisation, since the ready-made pedals and effect units only worked in a way that they were meant to be and there was no option for the user to alter the way they operate. One of the handiest tools that I got to experiment with was a looper/sampler unit which is Line6-DL4. I mostly used this unit to lay down textural sounds that I accompany when I am playing and re-sample during the course of a performance and repeat the process of sampling, enabling the creation of new musical materials. This process of re-sampling as I go, sparked the idea of creation of a system that can both navigate by itself through pre-determined scenarios and be conducted by the performer as well. With its modules such as; Buffer, Granular Synthesizer, Game of Life (CA), Polyrhythm Generator, Time Stretch and Granular Delay, *MEMO* offers an interactive environment in which the material played by the performer is stored in the main buffer and re-iterated through the aforementioned modules.

1.1. Structure of the thesis

This thesis is divided into five main chapters; Introduction, Literature Review, Methodology, Developing *MEMO* and Conclusion. Having talked about my personal motivations and the nature of *MEMO* as well as my main research issues in this chapter, I will be talking about the concept of interactivity, the roles of interactive music systems, novel improvisation systems and sound synthesis methods in Literature Review chapter. Besides, the main methodological approach with regards to the research of this project, my creative processes and aesthetic motivations as well as *MEMO's nature* and the way *MEMO* relates to the theories will be discussed in Methodology chapter. Moreover, in the fourth chapter of this thesis which is Developing *MEMO*, the development phase of *MEMO* will be discussed in detail. Also, a discussion on how this system was developed on Max software, the most useful control strategies and the roles that *MEMO* can undertake as an interactive music system will take place. Lastly, in Conclusion chapter I will be talking about the outcomes of this study as well as future directions that this system can embody. Also, a video excerpt that reflects the behavior and the musical merits of *MEMO* can be found in the appendix of this thesis.

2.LITERATURE REVIEW

This chapter is concerned with the creative practices of Improvisation Systems under the context of Human Computer Interaction in real time performances. The main concerns of this diverse research and practice field are the development of novel musical systems that are responsive, generative or interactive in terms of their engagement with the performer.

2.1.Defining The Roles of an Interactive Music System

According to Robert Rowe's Taxanomy of Interactive Computer Music systems, there are three main concerns regarding the roles of Interactive Music Systems:

1. *Score - driven / performance-driven:* In this model the system either reacts to the performer based on pre-determined scenarios and probabilistic methods or with spontaneity.
2. *Transformative / Generative / Sequenced:* This model is related to the way a system operates. To give an example, a system can operate in a transformative behavior and it is not mostly driven by the performance, on the contrary the generative or sequenced nature of the system drives the performance.
3. *Instrument / Player Paradigms:* These paradigms are related with whether a system acts as an instrument or an improvising agent (Rowe, 1993).

Stockhausen's *Mikrophonie* I (1964) can be given as an example for the mixture of score-driven, transformative and instrument paradigm with the instruments such as tam-tam, 2 microphones, 2 filters and 6 performers since it employs both performers and electronics controllers operating based on the score of the piece. Another example of the same interaction model would be Boulez's *Anthemés* (1997) for violin and electronics.

Another classification with regards to the levels of interactivity in computer music systems came from Todd Winkler in 1998 and he proposes the levels of interaction as;

1. *The Conductor Model:* Everything is controlled by a single source and musical responses are dictated by the executor. This is similar to the Rowe's taxonomy of Interactive Computer Music systems exemplified above; "Score-driven systems".

2. T*he Chamber Music Model:* Similar to a string quartet where the roles are switched between the players. In this case, the control of the system can be traded between the player and the system. Thus, this model can be also be perceived as a hybrid model.
3. *The Improvisation Model:* A jazz ensemble playing through the chord changes can be given as an example for this model. This is also similar to Robert Rowe's "Score-driven" classification (Winkler, 1998).

Besides Winkler's classification, Orio and Wanderley suggest a list of contexts in Interactive Computer Music:

1. *Note-level control* (as a musical instrument)
2. *Score-level control* (conductor)
3. *Sound processing control* (generative unit)
4. *Traditional HCI Contexts* (using a mouse, keyboard etc.)
5. *Interactive Multimedia Installations* (Space oriented)
6. *Interaction in Dance* (Embodiment)
7. *Control of computer games* (Using a joystick) (Orio et.al., 2002.)

These classifications reflect the levels of interaction as possible models in interactive music systems.

Apart from the classifications from different perspectives I stated above with regards to interaction models in performance systems, a study by Thomas Ciufo outlines the possible "control strategies" in novel improvisation systems in which he poses the following issues:

1. Defining the desired relationship with the instrument or the system: Is the system going to be an extensible instrument or will it be another performer/improviser or something else?
2. Determining the desired control level over the system: Should the control level over the system be a democracy, dictatorship or something hybrid?
3. Designating the effects of the control system: (Should the control systems impact the overall performance direction on a micro or macro level or something in between?)
4. Defining the response stimuli of the system: What is the desired level of explicit control and how much unpredictability should the system offer?
5. Introducing control strategies: If the system works in conjunction with controllers, what are the primary issues of performability?

6. Finding the sweet spot: How can we achieve a balance between extensibility and playability?

With these issues that I referred to Thomas Ciufo's article "*Design Concepts and Control Strategies for Interactive Improvisational Music Systems*", he argues that considering these questions will greatly affect the design, realization and outcomes of Interactive Improvisational Systems (Ciufo, n.d.). Also, regarding the behaviors of such systems, Ciufo highlights two opposing polarities, that are; deterministic and indeterministic. Joel Chadabe describes these two opposing sides of operation models as, deterministic being defined by complete predictability acting fully under the scenarios set by the developer and indeterministic being fully unpredictable to which the performer reacts and vice versa (Chadabe, 1983).

2.2.Definition of Interactivity

Another notable topic under the context of Interactive Computer Music Systems is the definition of "interactivity" itself. The definition of "interactivity" has had different perspectives in the literature. Robert Rowe defines an interactive music system as *"one whose behavior changes in response to musical inputs"* with three different stages; sensing, processing and response (Rowe, 1993). Garth Paine, in his article *"Interactivity, where to from here?"* infers from Rowe's definition that the term "musical" refers to an awareness of a system that makes sense of what is being played (Paine, 2002). So, in this definition the system has to make sense of what is being played so as to be defined as "interactive".

Joel Chadabe however, describes an interactive instrument as; when a performer engages with such an agent/instrument with which the performer shares controls of the music through algorithms and seems to be in a dialogue where the agent/instrument creates unpredictable information reacted by the performer and performer creates control information reacted by the agent/instrument, this agent/instrument can be referred to as interactive. With this idea, Chadabe describes a feedback loop between the agent/instrument and the performer and he refers to this as an "interactive instrument". Besides this definition, Chadabe exemplifies interaction models as;

1) Simple Model: This model can be described as a basic action/reaction type of interaction, for instance using a mouse or a keyboard.

2) Fly-by wire: In this model, Chadabe exemplifies the operation of a Boeing 777 plane where the interface of the plane assists the pilot to operate the plane.
3) Interactive Model: Here, Chadabe describes this model as instruments having their own thoughts and reacting autonomously to each other.
4) Daily life model: As we interact with the winds and waves of life, the currents do not respond to what we do but share control with us instead (Chadabe, 1983).

Computer game designer Chris Crawford (2005) suggests that the nature of interactivity is a subjective one; as he explains, there can be high, moderate, low or even zero interactivity degrees. Thus, we can solve the subjective nature of interactivity by accepting that anything can be interactive and the degree of it can be left as a subjective concern.

2.3. Interactive Music Systems

Having discussed the interactivity concept in the literature, it is a vital part of this chapter to discuss the novel interactive music systems that were developed through the performer and the developer perspective.

One of the most notable interactive music systems developed in recent years is *InMuSIC* (Lepri, 2016) system which is a multimodal interactive system for live electroacoustic improvisation, developed particularly for clarinet and live electronics by Giacomo Lepri, this system can be defined as an interactive performance environment which generates musical responses by analyzing the performer's bodily actions through (upper-body motion tracking) as well as the sonic parameters of the sounds played by the performer. After the analysis, various decision-making processes negotiate between the system's internal stochastic processes and the analyzed performer's behavior. *The Harmonically Ecosystemic Machine; Sonic Space No. 7,* which was developed by Michael Musick, Jonathan P. Forsyth and Rachel Bittner, is a result of a collaboration between music technology researchers focused on harmonic accompaniment generation through finite state transducers (FSTs). This system is an installation-based system and it interprets music that occurs in the physical space of a room through a harmonic perspective. The analysis of musical information is achieved by Music Information Retrieval (MIR) Basically this system creates musical outputs based on the participants actions in a physical space (Musick et.al., 2015). Another notable interactive music system is *OMax* (IRCAM, 2015) which was developed by

IRCAM. This system was created based on a research on stylistic modeling by Gerard Assayag and Shlomo Dubnov and a computer improvisation research by G. Assayag, M. Chemillier and G. Bloch. *OMax* (Assayag et.al 2015) was developed on OpenMusic and Max softwares. It is basically a software environment which is able to learn the performer's style of playing in real time and accompanies interactively, which gives the impression of human - machine co-improvisation. *OMax* carries out analyses on the instrument signal and extracts an event-based performance data. Besides this, an algorithm called *Yin* which was developed by Alain de Cheveigne and Hideki Kawahara is used to execute the pitch-MIDI conversion, translating the acoustic signal to MIDI data which then processed by machine-learning methods. With these methods, *OMax* system develops improvisational responses and the performer seems to be in a conversation with himself/herself. Francois Pachet's *Continuator (*Pachet, 2002*)* system learns the behavior of a performer and generates interactive responses by using augmented Markov chain models. Pachet states that the purpose of *Continuator* system is to enhance the technical ability of a performer by integrating a stylistically consistent music system which automatically learns the musical material played by the performer. Pachet also states that the *Continuator* is in between two classes of what he refers to as traditionally incompatible music systems; interactive and imitative. Continuator extracts the midi information of what is played and re-models it by using Markov chains which leads to the reorganization of the pitches played by the performer but played back in the same manner as the performer. The *_derivations (*Carey*, 2012)* software is an interactive performance system, which was developed by saxophonist and electro-acoustic improviser Ben Carey. This system mainly uses live-sampling and timbral matching techniques to create improvisatory responses to an improvising performer, enabling an iterative and improvised musical dialogue between the computer and the performer.

Besides this, another improvisation system based on live sampling and sonic deviation *The Party Van* (Constanzo, 2014) was developed by improviser and computer musician Rodrigo Constanzo. *The Party Van* samples the live input of a performer in real time and sends the sampled material to a number of modules such as slicer, granular synthesizer, pattern generator, stutter, shuffler, chopper, bit reducer and reverb. With these synthesis modules, The Party Van creates a vast opportunity to deviate a number of musical parameters; pitch, texture, timbre and time. This system supports MIDI controller mapping which enables the performer to make use of different combinations of modules while performing. The trombone player, composer, improviser and audio programmer George Lewis created *Voyager* system (Lewis, 2000) which he describes as a "virtual improvising orchestra" in which a performer cannot directly control the system but can influence its attitude. Voyager system analyses the input of a performer, converts it to MIDI, which is then transferred to 64 asynchronously operating single voice MIDI controlled "players" all of which generate music in real time alongside with the performer. George Lewis (2000) describes his perception of *Voyager* system as: "*I conceive a performance of Voyager as multiple parallel streams of music generation, emanating from both the computers and the humans—a non-hierarchical, improvisational, subject-subject model of discourse, rather than a stimulus-response setup."* As a model for Voyager, George Lewis exemplifies a Javanese Gamelan ensemble where there are no hierarchies between the players and the music serves a holistic purpose. Computer music pioneer Robert Rowe (1993) developed *Cypher* system, which listens to a musical performance, analyzes real-time and generates responsive outputs In an interview by Lev Koblyakov, Rowe states that he has two main motivations for developing *Cypher* system; using the computer to aid the compositional processes and training computers to listen and react to musical inputs. *Cypher* system can be modified with regards to its responsiveness by re-defining the interactive mappings of the system which allows the performer to affect the way *Cypher* responds to musical input.

Apart from the discussion of various point of views with regards to the nature of interactivity, its degrees and possible concerns related to the development of novel improvisation systems as well as notable projects realized in this context, this chapter will be followed by an overview and discussion about audio signal processing methods. The reason for this is that, *MEMO* mainly uses the methods mentioned below as its main operation principle. Later in Chapter 4, the application of these methods in Max software environment for development of *MEMO*, will be discussed in detail.

2.4. Audio Signal Processing Methods

2.4.1.Sampling

The first attempts on sampling started with the tape experiments which became a compositional tool in the second half of the twentieth century for the post-war electronic musicians (Manning, P. n.d.). After the devastating Second World War, with the help of advanced technologies compared to the pre-war era, there was an artistic revival in Europe. Two broadcasting networks in Europe took the initiative; Radiodiffusion Télévision Française (RTF) which was located in Paris and Norwestdeutscher Rundfunk (NWDR) in Cologne. The French school focused on "Musique Concrete" and the German school pursued what they called "Elektronische Musik". These two pioneer movements in electronic music had strong disagreements with regards to their approaches. Pierre Schaeffer who was an electronic engineer did his first work "*Etude Aux Chemins de Fer"* at which he manipulated the train sounds that he recorded on his tape machine. The main aim was to manipulate the sounds recorded on the tape machine through montaging, speed shifting and reverse playback of the tape. Also, during 1950s, in Cologne similar experiments were carried out by Karlheinz Stockhausen at Westdeutcher Rudfunk Studio however he was working both with purely generated tones besides the recorded sounds. Around this time Stockhausen composed *Gesang Der Jünlinge* in which he used human voice with electronically generated tones bringing together the two opposing worlds of *German Elektronische Music* and *French Musique Concrete*. Although these early experiments of sampling focused on compositional aspects of music, later it pawed the way for musicians such as Terry Riley, Steve Reich, Robert Fripp, Brian Eno and myriad of other musicians to use the tape recorder as a performance tool.

2.4.2. Live sampling

Apart from its compositional uses, sampling was also used as a tool for improvisation and performance. This practice was initially experimented in the 70s through reel to reel tape recorders. Apart from Brian Eno, Terry Riley and Steve Reichs experiments, one of the main pioneers of this was Robert Fripp's *Frippertronics* which uses an analog delay system created with two reel to reel tape machines, one of which records the musical input and the other plays back the recorded material feeding back to the initial tape recorder and the delay time can be controlled by setting a distance between the two machines. Frippertronics was used in Brian Eno & Robert Fripp's album *No Pussyfooting* (1973). Figure 2.1 shows an example of reel to reel tape recorder setup for Frippertronics.

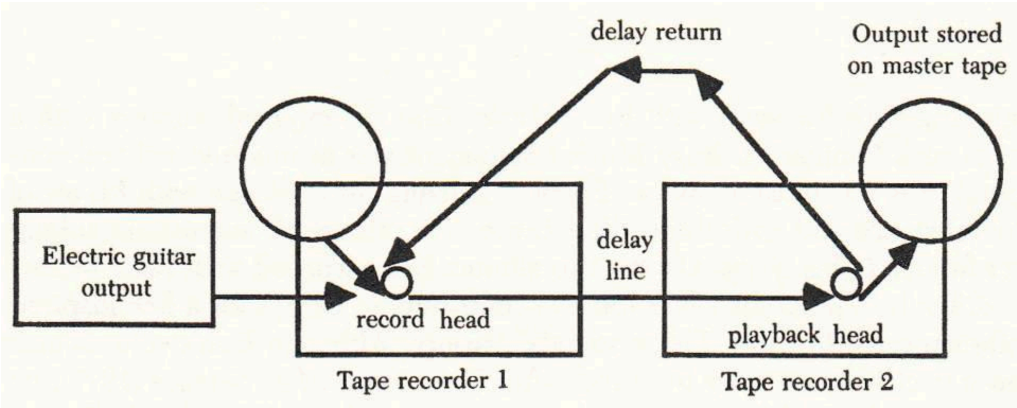

Figure 2.1: Frippertronics diagram

2.4.3.Granular synthesis

Granular synthesis was initially proposed by the physicist Dennis Gabor (1947) in his theory of hearing at which he described the sound particles as *acoustical quanta*. Being a relatively recent development in sound transformation, Granular Synthesis was initially experimented by the composer and mathematician Iannis Xenakis in 1971 and in 1978 by Curtis Roads. Xenakis, when he discovered Gabor's research, he was inspired to apply the theory musically by chopping a magnetic tape in small fractions and rearranging them. Having attended one of Xenakis' seminar on Granular Synthesis, Curtis Roads began experimenting with the idea on computer. One of the most significant features of this technique is that it offers a huge span of sonic material from an audio sample, ranging from the very micro sound level to rich textures of individual grains (Roads, 2001). In this synthesis method, the sounds are broken into tiny grains and then reorganized with alterable sizes, onsets, envelopes creating various timbres and temporalities (De Poli, G. et.al 1991).
In 1988, Barry Truax offered a real-time implementation of granular synthesis by using DMX-1000 Digital Signal Processor (Wallraff, 1979) which is one of the earliest DSPs. Truax' main principle for real-time granulation is that the incoming audio signal is read through a buffer, enabling the processing of the signal in real time.

2.4.3.1. Quasi-synchronous granular synthesis

In this method, the grains are organized as separate voices and each voice generates grains with a changing synthesis period, determining the time delay between their onsets. Besides, the proportion of the synthesis period and grain length determine the amount of grain overlap. In this case, when the grain length is lower than the synthesis period, grain overlap does not occur which causes Amplitude Modulation (AM) since the audio stream is interrupted. At Figure 2.2, the waveforms of grain streams are displayed.

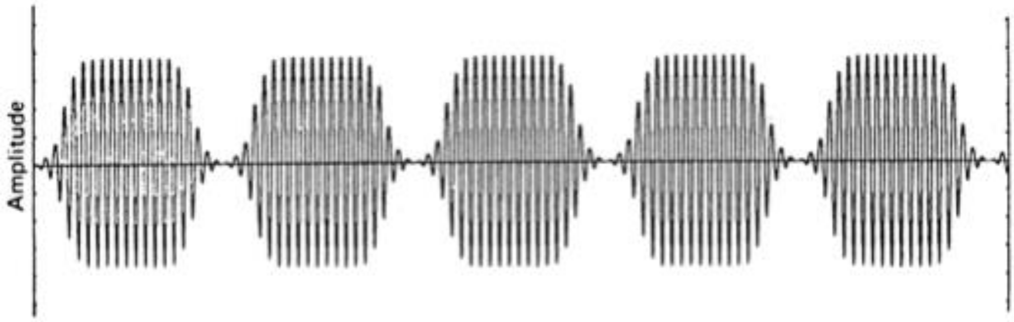

Figure 2.2: A stream of grains resulted in AM due to grain lengths lower than 50 ms.

2.4.3.2.Asynchronous granular synthesis

This synthesis method uses grains that are randomly distributed without any quasi regularity which causes asynchronicity. This method works well for clouds of grain textures. Figure 2.3 shows a waveform of asynchronous grains.

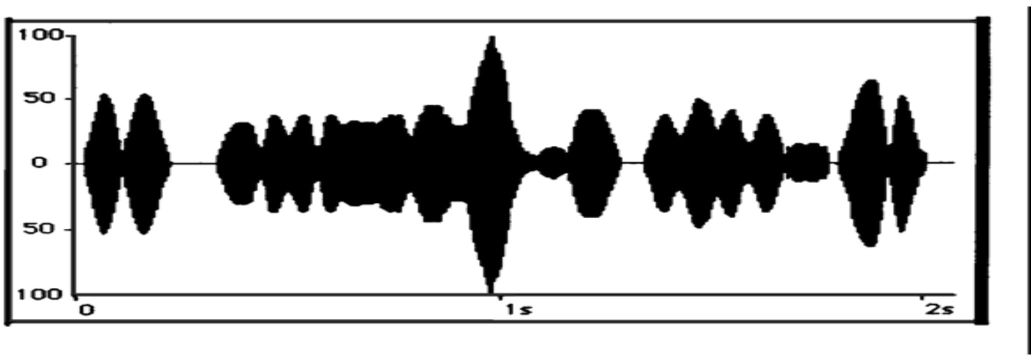

Figure 2.3: The stochastically dispersed grains without any synchronicity.

2.4.3.3.Pitch-synchronous granular synthesis

Initially experimented with speech reproduction with source/model (Moulines and Charpeniter 1990), this method uses a time-varying filter creating overlapping grain envelopes that are synced with the frequency of the grain waveform. Figure 2.4 shows an example of overlapping grain envelopes that are synchronized.

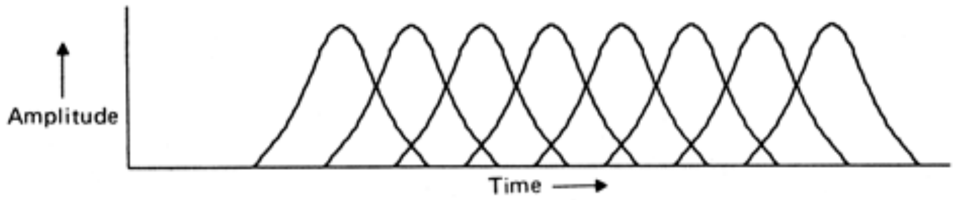

Figure 2.4: A graph of overlapping synced grain envelopes.

2.4.3.4. Polyrhythmic granular synthesis

It is possible to create polyrhythmic structures by firing individual grains at certain time intervals. Wellington-based composer, software programmer and music theorist Michael Norris created the Max external [*spindrift~*] that features standart granular synthesis parameters as well as multi-channel spatialization and polyrhythmic treatment. Basically, this external uses Max's *~poly* object to store each triggered grain and play them simultaneously at various rhythmic intervals.

2.4.4. Cellular automata

Cellular automata (CA) is a statistical generation model introduced by John von Neumann in 1960s (Miranda, E. R. 2007). Neuman's initial motivation was to develop a system in which the dwellers could reproduce themselves. However, later by following a suggestion of his friend Stanislaw Ulam, Neumann focused on a two-dimensional, distinct system in which there were 29 different states available as well as self-reproduction.

Later on, mathematician John Conway who was inspired by this idea, developed Game of Life (Berkelamp et.al, 1982) which became one of the most popular automatons ever.

Conway's Game of Life algorithm is based on the following rules: For a space that is occupied in the matrix, each cell having less than two neighbors dies, due to solitude. If a cell has four or more neighbors, it dies due to overpopulation. A cell will only survive if it has two or three neighbors. For a cell that is empty, if it is surrounded by three neighbors it becomes populated.

CA is widely used as a generative tool for sonic deviation. For example, Mara Helmut and colleagues designed a CA that is based on a real-time sound granulator. The implementation of this technique occurred by the decomposition of a sound sample into individual grains, which were assigned to 32 band-pass filters and a CA algorithm is used to control the bandwidth and main frequency values of these grains. With this method, a generative transformation of harmonic structure of the individual grains were obtained. As it takes place in *MEMO*, CA can be used as a sequencer operating based on the pre-determined rules (e.g. Game of Life), thus it be used as a sampling tool in which the portions of the sample are played based on the generative sequences created by this method. Besides this, Los Angeles based sound-artist Dillon Bastan in conjunction with Cycling 74 developed a CA algorithm based on the rules of Game of Life. This algorithm makes it possible to create sequences by scanning and playing individual chunks of sounds from a sound sample.

2.4.5. Time stretching

Time stretching has long been used as a creative tool in the field of sound design. Its main principle is based on the manipulation of temporal domain without changing the pitch information. There have been a number of approaches for the implementation of this method and one of the main methods is called Time-Domain Harmonic Scaling which was introduced by Rabiner and Schafer in 1978. Initially invented for speech processing, this method calculates the fundamental frequency of a signal through pitch-detection algorithm and then the sections are crossfaded into one another, resulting in the modulation of the time domain of the signal. Another method is called Phase Vocoding, this method was invented by Flanagan and Golden (1960) and it implements time stretching by transforming the audio signal into frequency domain by using phase information. Phase vocoding is similar to that of granular synthesis in which a signal is broken into equally calculated grains. If the extracted chunks (grains) of an audio signal has higher amounts of overlap, the output result will sound stretched in time domain. Also, researches of Gabor (1946) on *acoustical quanta* which led to the research of granular synthesis made possible to realize time stretching through granular synthesis. In this method, the frequency of grains is multiplied by the stretch factor resulting in the temporal stretch of the audio signal.

2.5. Review

In this chapter, I examined the key approaches, theories and the practices that are relevant to my novel improvisation system *MEMO*. The issues such as the roles of an interactive music system, degrees and models of interactivity and the definition of interactivity itself were key topics to discuss, since they make up a huge portion of this research. Besides these key topics, other novel improvisation systems developed by performers, improvisers and software developers I stated above were exemplified. Also, Audio Signal Processing was another key topic to discuss, since *MEMO* uses the aforementioned processing methods as the backbone of the system and realizes real-time sonic deviation through those methods. In the next chapter I will be talking about my methodology that I have undertaken with regards to this thesis and design process of my interactive music system *MEMO*.

3. METHODOLOGY

The main methodological approach undertaken in this thesis is based on a self-reflective, practice-based research approach. The main goal of this thesis is to reflect the creative processes in the practice of interactive music system design as a performer & developer with my novel improvisation system *MEMO.* Since the creative practices of interactive music system development have been mutually influenced by one another, reflecting on these creative processes, will be contributing to self-reflective research practices in this field.

As a performer of electric guitar and electronics, I have been experimenting with the concept of live sampling for a long time. During these times, I used looper pedals that have pitch-shifting and reverse playback features in my solo improvised performances. My main approach to these effect units and samplers was to use them in a way in which I would be able to be in a conversation with myself. For instance, looping a rubato melody that is around 20 seconds long, then playing it back reverse, half speed or double speed and then reacting the synthesized sounds with new musical utterances. Also, I have been using many effect units in my setup to alter the sounds that I create and this has enabled me to improvise with various sound textures and rhythms. Having performed with this method for some time, I realized a need to expand the behaviors of these sound processors and looper/samplers I discussed above, since those ready-made units had their own way of operating which created constraints with regards to expansion possibilities during the course of a performance. This was the main reason I decided to develop a fully customizable system in Max which is able to generate new and unexpected musical material by incorporating temporal and timbral modulation sequences derived from real-time sound synthesis methods in which the element of surprise is the driving force of any instance of performance with this system.

The main musical needs that this system addresses, can be explained by Mark J. Butler's (2014) term "pre-existent elements" that he uses to describe the musical structures on which the improvisational performances based. For instance, with its different modules such as, Granular Synthesizer, Game of Life (Cellular Automata), Polyrhythm Generator, Time Stretch and Granular Delay, *MEMO* offers a vast range of stylistic possibilities, these possibilities enable the creation of an arsenal of musical material to be used in the performance. Each of these modules, with their distinct and alterable behaviors can be defined under the context of “pre-existent elements”. Apart from the notion of pre-existent elements, this system is a music generator that can be used in different musical instances, such as non-idiomatic musics (free-improvisation), minimalistic musics and live electro-acoustic performances.

3.1. The Aesthetic Grounds

MEMO offers a wide range of possibilities with regards to the musical style and aesthetics. As George Lewis points out, all of the musical systems reflect the attitudes of their creators. In this case *MEMO* is a system whose aesthetics are highly influenced by minimalism, gamelan music, ambient and electro-acoustic musics. Each module in *MEMO* offers a distinct part of these influences. If we are to talk about main buffer’s aesthetic offerings, since it has the capability to record and playback at the same time with reverse / normal playback and low or high octave shifting features, using these properties real-time create a possibility to lay down the reverse notes with alterable octaves as textures accompanying to the improviser. With these working principals, the main buffer module offers a way to create contrapuntal way of interaction with the performer. Besides, the granular synthesis module is able to re-iterate small chunks of the incoming signal with randomization features. By setting the grain size, length density and pitch manually or automatically with *MEMO*’s event schedulers, this module creates lush stochastic textures as well as the distorted playback of whatever is being played to the buffer. Also, since this module also offers modulations with regards to the grain lengths, it enables a play between the micro-time scales and meso time scales. Apart from these, the Game of Life module offers sequential playback of an incoming signal.

These sequences create surprising rhythms and textures due to the generative nature of Game of Life algorithm enabling the expansions of improvised material both in temporal and textural domains. Also, Polyrhythm Generator module creates polyrhythms inspired by Gamelan ensembles and minimalist musics of Steve Reich, and Philip Glass. Last but not least, the Time Stretch module creates slowly evolving powerful layers of textures with a wide range of spectrum as well as glitchy sounds due to its auto-pitch shifting features.

3.2. Comparing *MEMO* with the Literature of Interactive Music Systems

Before introducing *MEMO,* locating *MEMO's* stance in the field of Interactive Music Systems by comparing to the novel interactive music systems in the literature is a vital part of this thesis. The systems that will be compared to *MEMO* are as follows: *InMuSIC* by Giacomo Lepri, *OMax* by IRCAM, *Continuator* by Francois Pachet, *_derivations* by Ben Carey and *The Party Van* by Rodrigo Constanzo. To start with InMuSIC system by Giacomo Lepri, there are two main issues with regards to this system's operations that are the multimodal analysis of the instrumentalist's upper body expressions and the analysis of the sonic parameters of the performer's output. The musical outputs are organised based on these analyses. The generation of musical materials in this system are divided into three: Synthesis (FM, additive, subtractive, and physcial models), sampling (real-time processing of pre-recorded sounds) and live processing (live sampling, live granulation, FFT analysis, resynthesis and reverberation). Besides InMuSIC system, *OMax* system operates based on stylistic modeling which uses "multi-level memory models" enabling the discovery and learning process for the system. With whis operation principle, *OMax* is able to learn the behaviour of the performer and create musical responses. Similarly, Francois Pachet's *Continuator* system operates based on the analysis of musical phrases of the performer which are transformed to MIDI, then re-organized by Markov chains. This creates an impression of human-machine co-improvisation. All these systems operate based on an analytical method, either based on the body movements of the performer or the sonic qualities of the performer's output. However, *MEMO* does not embody an analytic function or a machine learning method that makes sense of the performer's improvisatory material. As it will be furtherly discussed in the next chapter, *MEMO* mainly operates based on live-sampling of the performer's output which is processed and deviated by the system's distinct sound synthesis modules. In this sense, *MEMO* is similar to Rodrigo Constanzo's *The Party Van* which operates mainly based on live sampling. In this system the sampled material is processed by the modules such as slicer, granular synthesizer, pattern generator, stutter, shuffler, chopper, bit reducer and reverb. Besides, InMuSIC system's sound generation principles that are listed above are similar to that of *MEMO's* sound generation models.

Also, _*derivations* system by Ben Carey works mainly by live-sampling as well. However, this system has a timbral matching function that differs from *MEMO*. This function allows the system to analyze the timbre of the performer's output and re-model it with congruency, creating improvisatory responses to the performer.

3.3. Introducing *MEMO*

As the different perspectives relating to the degree and the nature of interactivity were discussed in the Literature Review chapter of this thesis, and as a result of the comprasion of *MEMO* with the novel interactive music systems listed above, it can be said that, since *MEMO* does not have an analysis method with regards to the performers output that creates autonomous decisions based on these analyses, as it is in the systems such as OMax, InMuSIC, Continuator, and _derivations, *MEMO* relates to these conceptions of interactivity as follows:
Joel Chadabe's "Fly by wire" model of interactivity through which he creates an analogy for the operation of interactive music systems and the operation of a Boeing 777 plane in which the interface of the plane assists the pilot who shares control with the system, can be another model of interactivity that describes *MEMO's* way of operation. With regards to Robert Rowe's taxonomy of interactive music systems, *MEMO* can be defined under the category of "Score/Performance driven" systems. This is due to the fact that *MEMO* is fed by the performer's live input which is stored instantaneously and processed by each module in the system which have alterable, pre-determined scenarios. An example for these scenarios would be the randomizer function in the Granular Synthesis module. Also, the frequency and stretch rate of Time Stretch module can be given as an example for these scenarios as well, since deviations of the performer's material can be affected drastically by setting or altering these parameters. Besides relating to Robert Rowe's taxanomy, *MEMO* can be situated under Todd Winkler's classification of interaction models which he defines as "Conductor Model". Although *MEMO* can be used as an autonomous agent by incorporating event schedulers (e.g. randomizer functions, automatic parameter shifters) that change the parameters automatically, the random nature of this setting might cause problems with regards to the context and credibility of an improvisatory performance. Thus, the best scenario for this model in my practice is that the incorporation of a MIDI controller,

that is able to be mapped to such parameters. This way the system can still have some autonomic behavior but can be intervened by the performer whenever necessary. *MEMO* as a whole, can be defined as a music system which can be used both as an improvising agent and as an instrument. Since this system is mainly based on live sampling and the deviations of the sampled material, after recording or loading a sample into the buffer, the performer will be able to use *MEMO* as an instrument by manually shifting its parameters via a MIDI controller. Apart from this scenario, *MEMO* is able to operate in a hybrid mode in which a MIDI controller can be used to enable/disable the modules, start/stop the randomizations and use automatic parameter shifting features by incorporating the event schedulers in *MEMO*, which makes it possible to use this system as a co-improvising hybrid agent with the performer. Since this system is meant to be a part of my practice as an improviser, my usual setup when I use *MEMO* as an improvising agent as follows:

- Fender Telecaster Electric Guitar / Classical Guitar
- APC40 MIDI Controller,
- Line6 DL4 (Delay Modeler / Sampler)

I occasionally choose to use Line6 DL4 Delay Modeler / Sampler in my setup if I want to feed *MEMO's* buffer with textural sounds rather than only my guitar input. Since Line6 DL4 has a 16 second sampler that is able to manipulate the way playback occurs, it is possible to create overlapping guitar textures and feeding these textures into MEMO's buffer will result in various other sounds with different temporalities and timbres.

3.3.1. *MEMO* as an instrument

MEMO's use as an instrument can be achieved by loading the buffer by recording the live input in which the length of the loop can be modifiable. When used as an instrument, the longer loop lengths in the buffer enable more variety with regards to the deviations of the signal by *MEMO's* modules. This is due to the fact that the buffer is constantly scanned by the *MEMO's* modules and the longer loop times create more possibilities with regards to the iterated material since the content is more varied compared to a shorter loop length. Moreover, *MEMO* works as a sampler that is processed with distinct modules such as Granular Synthesizer, Game of Life, Polyrhythm Generator, Time Stretch and Granular Delay. With these modules, MEMO is a fully functioning sound-synthesis engine that is able to generate rhythms, melodies, textures with its distinct generative methods in each module.

3.3.2. *MEMO* as an improvising agent

Apart from its use as an instrument, *MEMO* is able to operate as an improvising agent as well. Since the main operation principle in MEMO is based on live sampling, the way *MEMO* interacts with the performer is achieved by sampling the material played by the performer in real time and re-iterating them with *MEMO*'s real time sound synthesis modules that I will be elaborating in the following sections of this chapter. Each module in *MEMO* has different ways of re-iterating the material fed into the buffer, the controls of these parameters can both be done by manually with a MIDI controller or automatically with event schedulers of *MEMO.*

3.3.3. Introducing control strategies

The main aim with regards to the control of this system when designing *MEMO* was to be able to have a system that is able to carry out the sonic deviation generatively and at the same time can be controlled and conducted by the performer. This idea sparked two main control strategies that are:

Instrument Mode: As discussed above, this setting enables *MEMO* to be used as a sampler with sound synthesis modules that are working both in temporal and timbral domains. An example of this scenario can be as follows:

Loading the buffer with a selected sample, in my case I usually pick or record sounds with rich harmonics and textures. This is because when the modules in the system reiterate the buffer with their distinct algorithms, usually the results are the creation of the richer harmonics with wide range of spectrums, unique gestures and evolving textures.

Hybrid Mode: In this setting, *MEMO* acts as an improvising agent that is under control of the performer. The most flexible scenario for this mode for my practice is that I choose to set the buffer length as maximum 8 seconds long. The reason for this is that the longer gets the buffer length, the lower gets the chances of conversation and interaction since the modules might end up scanning the other end of the buffer, while the new input is fed into a quite distant point. Due to this, the most useful setting of the buffer length should be maximum 8 seconds long if a high interactivity rate of conversation is desired from the system. Since *MEMO* has its own event schedulers, it is useful to incorporate them during the course of the performance since they have considerable contributions with regards to variety. Besides, these parameters can be mapped to a MIDI controller as well, in case the performer need to intervene the process.

3.3.4.MIDI mappings

The MIDI mappings in *MEMO* can be realized by clicking assign MIDI Map button located on the bottom of the patcher. In the mapping mode, the Max toolbar as well as the objects that are available for mapping will turn blue. Also, some of the UI objects such as sliders and keyboards need to be enabled before being mapped to a MIDI controller. The mappings will be functional both in locked and unlocked modes. Deletion of a parameter's mapping can be done by clicking the parameter and hitting the "delete" key or by clicking "Delete Mapping" from the contextual menu. At Figure 3.1 mapping activation is shown.

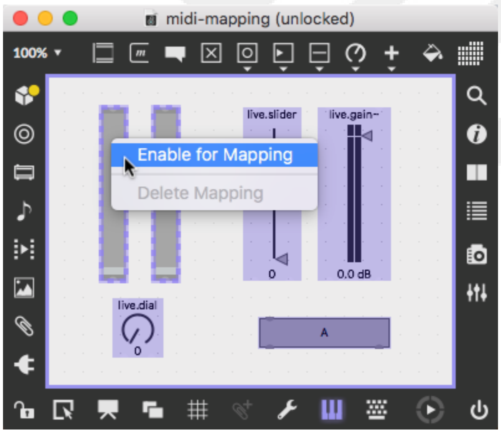

Figure 3.1: Enabling for Mapping.

Also, at Figure 3.2 MIDI Mapping assignment in *MEMO* is displayed.

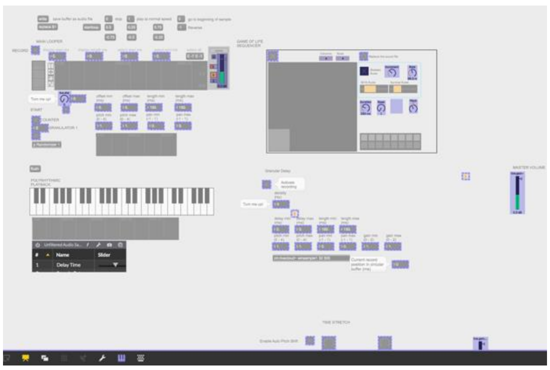

Figure 3.2: Assigning MIDI Map.

3.3.5.Reflecting on the musical output

Before explaining the patching process for the development of *MEMO* in Max software, this section will cover a short musical analysis of the two performance excerpts that are embedded in one video and that can be found in the appendix of this thesis.

The first performance excerpt starts with stochastic textures created by the Granular Synthesis and Game of Life modules, processing the electric guitar output stored in the buffer. At (00:08 – 00:11), the electric guitar creates a melodic utterance that is immediately reacted by the Time Stretch module (00:11 – 00:20) that is creating iterative glitches accompanying the melody played by the electric guitar. After this response by the system, the performer disarms the record button in the Buffer module, enabling the synthesis of the material that was left in the Buffer. Between, (00:20 – 00:30) the Time Stretch module keeps deviating the melodic utterance with auto-pitch shifting and at (00:31) the performer cranks the pitch value in the Game of Life module all the way up and the textures created by this module gain a brighter sound character. After this, at (00:38) the performer incorporates the Polyrhythm Generator module, latching grains from the selected waveform offset and slowly increases the number of latched grains, this creates multiple polyrhythms due to different time intervals of the latched grains.

The second performance excerpt starts with another melodic utterance by the electric guitar (01:31) followed by the muted notes (01:36 - 01:45) and these notes are processed by the Time Stretch and Game of Life modules (01:50 – 02:04), enriching the timbres of the muted notes with various pitches. Following this interaction with the system, between (02:05 – 02:22), the electric guitar plays another melody and this time the modules don’t react to the melody since they are scanning a distant point in the waveform and still processing the initial melodic utterance. After this, at (02:42) the Polyrhythm Generator Module is incorporated and with the polyrhythms, the second excerpt comes to the end

4. DEVELOPING *MEMO*

As the developer of this system, my initial aim was to create a hybrid instrument/agent which seems to be in a dialogue with the performer during the course of a performance. This impression of a "dialogue" is obtained by live-sampling and its deviation with real time sound synthesis methods that I will discuss below. *MEMO* is comprised of six different modules: Buffer, Granular Synthesizer, Game of Life (Cellular Automata), Polyrhythm Generator and Time Stretch as well as a Granular Delay module which are constructed in parallel chains. These modules can be controlled in real-time by the performer as well as *MEMO's* event schedulers. Figure 4.1 shows the signal flow as well as the components of *MEMO*.

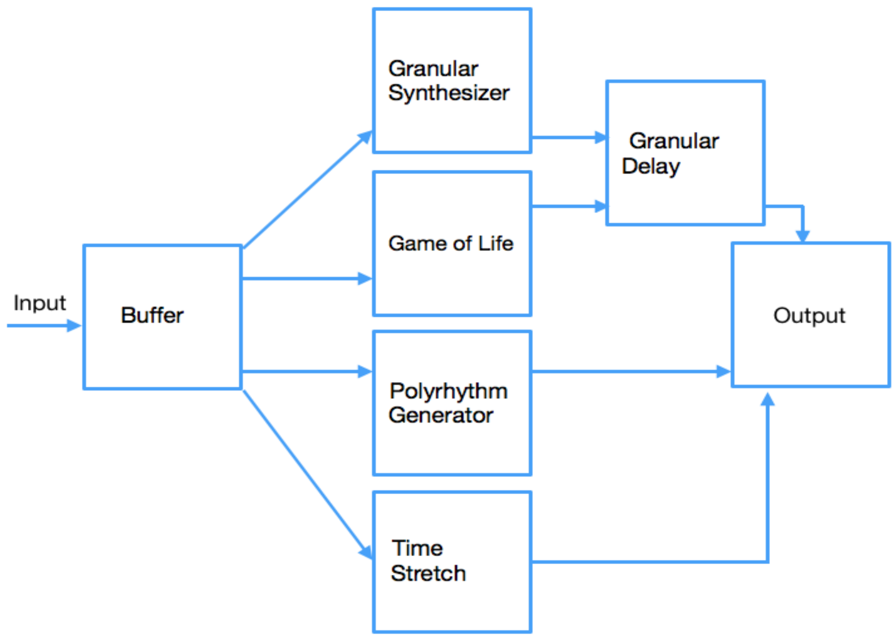

Figure 4.1: Components of *MEMO*

4.1. Buffer Module

The main operation principle in *MEMO* is based on live-sampling. This was achieved with the *adc~* object that can receive real-time audio signal and store it to the *buffer~* object through *record~* object. After the signal is received, *groove~* object is able to playback the material recorded on the buffer real-time. Also, the waveform of the recorded signal can be monitored through *waveform~* object. The parameters such as playback speed and direction can be controlled through messages sent to the *sig~* object which converts numeric information into audio signals and this object is connected to the third inlet of the groove~ object, enabling the manipulation of pitch content and playback direction. Figure 4.2 shows the patching view of *MEMO*'s main buffer.

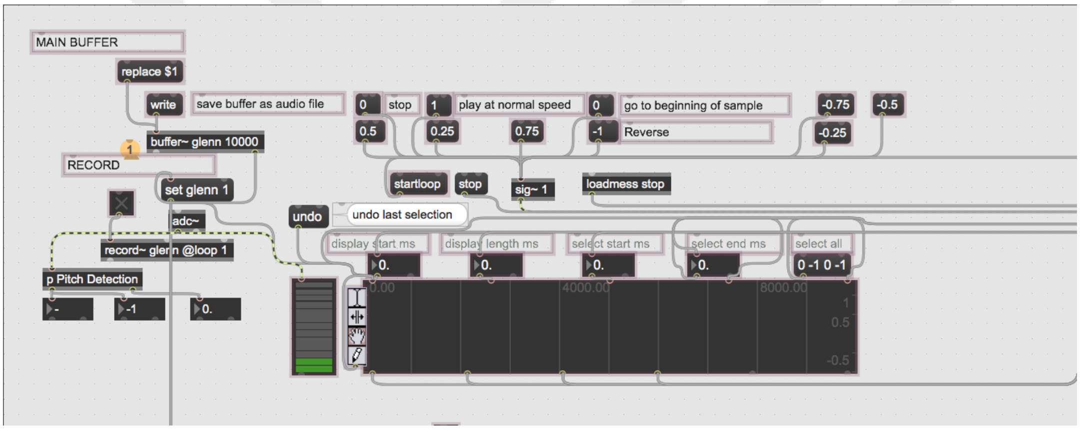

Figure 4.2: Patching view of main buffer in *MEMO*.

4.2. Granular Synthesizer Module

The granular synthesis algorithm in *MEMO* uses "*petra"* package developed by Circuit Music Labs[1]. This module uses *cm.buffercloud~* object that is a polyphonic granulator for mono or stereo audio files and it is able to read from the *buffer~* object in real time. Also, the windowing function can be changed by loading a sample into window buffer which allows the alteration of grain envelopes. The parameters such as minimum/maximum grain offset, grain length, pitch, pan and gain can be modulated simultaneously while the object is reading the buffer. Also, this object receives four different arguments that are; name of the *buffer~* object in which the sample is read,

[1] http://circuitmusiclabs.com/projects/petra-for-max/

name of the window sample, maximum cloud size (the maximum amount of overlapping grains) as well as maximum grain length. The playback of the grains can be realized in two different ways:

- Sending a bang and triggering a single grain by connecting the bang object in the first inlet of the *cm.buffercloud~* object.
- Connecting the *phasor~* object to the first inlet of the *cm.buffercloud~* object. The *phasor~* object generates sawtooth waves enabling sample-accurate control and with precise timings. As the frequency value in the *phasor~* object is increased, the number of grains will increase in direct proportion to frequency value.

Besides these properties, the direction of the waveform scanning can be randomized through *drunk* object that outputs random numbers within a step range. Figure 4.3 shows the Randomizer algorithm's patching view.

Figure 4.3: Patching view of Randomizer algorithm.

The patching view of Granular Synthesizer Module in *MEMO* is shown below at Figure 4.4.

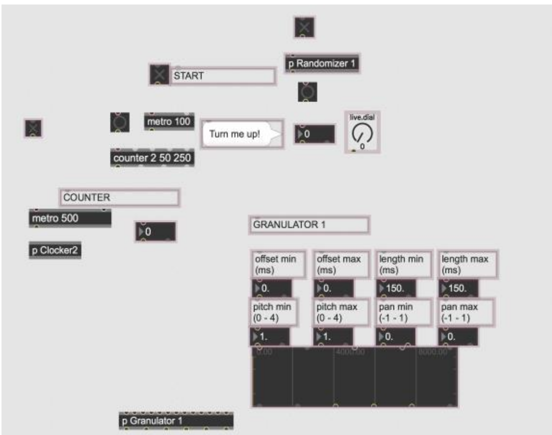

Figure 4.4: Patching view of Granular Synthesizer module in *MEMO*.

4.3. Game of Life Module

This module was developed for Max, by Los Angeles based sound-artist *Dillon Bastan* in conjunction with *Cycling 74* as a part of the package “*mc.movement.studies*”[2]. The algorithm creates sequences between 0-1 that are the grid values. After the sequences occur, new values are calculated based on the Game of Life rules, enabling the birth or death of the new cells according to the rules. Basically, this module sequences individual sections of an audio sample received by the buffer. The grid values determine the starting positions of playback. The signal flow occurs as follows: First, the playback sequences are created by *db.mc.periodiconeshot~* sub patcher developed by Los Angeles based sound artist Dillon Bastan. This sub patcher regularly outputs a single one-shot phasor according to 0/1 (off/on) values. The decision with regards to whether the module will play the sequence or not is based on the data received from Game of Life Rules that are 0 or 1 values (off/on). These values are received by *route* object which are sent to *zl.lookup* list after the values sent to the list, *sel* object selects the values that are 1 and dumps the zeros then sends an integer to the *oneshotphasor~* object. Figure 4.5 shows the patching view of db.mc.periodiconeshot~ algorithm.

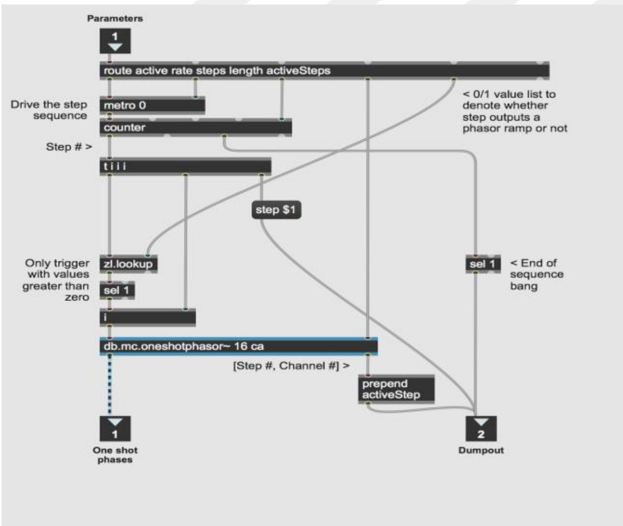

Figure 4.5: Patching view of *db.mc.periodiconeshot~* algorithm.

After the integer is sent to the *oneshotphasor~* object, the signal is transferred to *MC2DWavetablePlayback~* subpatcher. Here, the playback of the sample chunks are

[2] https://cycling74.com/articles/an-interview-with-dillon-bastan

realized through *2d.wave~* object and this object is able to fire individual audio segments in determined lengths, with various starting positions. The length and the pitch of the audio are controlled through *p playback_length* and *p playback_params* subpatchers and the signals from these subpatchers are transferred to db.mc.periodiconeshot~ subpatcher. The playback length and the parameters are shown below at Figure 4.6.

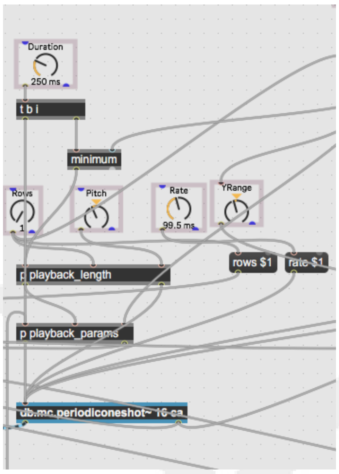

Figure. 4.6: Signal flow of playback length and playback parameters.

The patching view of 2d.wavetable playback algorithm can be seen below at Figure 4.7.

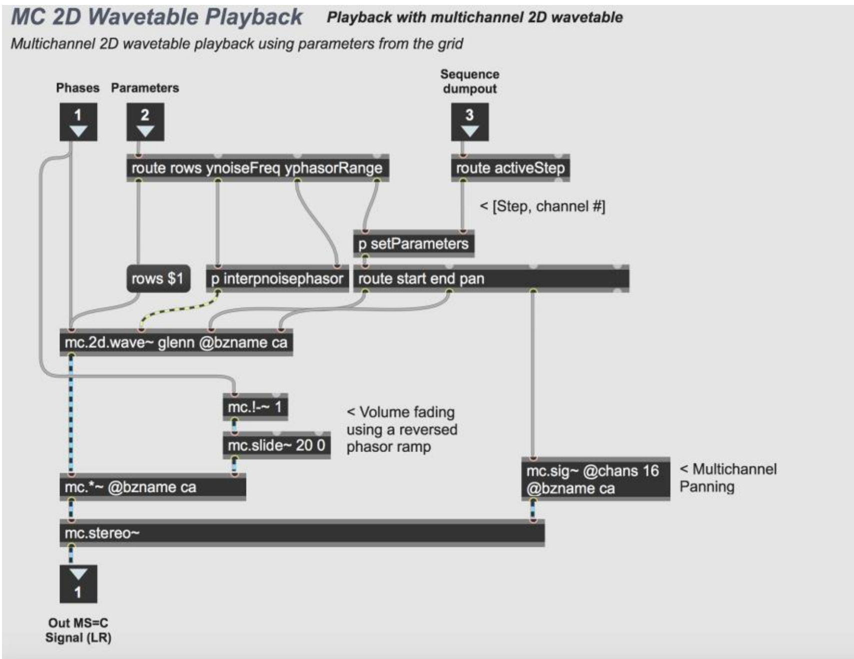

Figure 4.7: Patching view of 2d.wavetable playback algorithm.

Below, Figure 4.8 shows the GUI view of Game of Life module in *MEMO*.

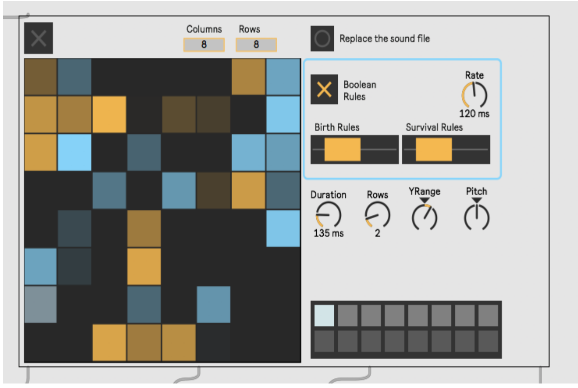

Figure 4.8: GUI view of Game of Life module in *MEMO*.

4.4.Polyrhythm Generator Module

The polyrhythmic playback algorithm in *MEMO* is realized through *spindrift~*[3] object developed by Wellington based composer, software programmer and music theorist Michael Norris. This object's working principle is based on granular synthesis, in which the small chunks of audio are played back with various grain sizes that can be determined by performer by selecting the waveform. Besides, this object is embedded into Max's *poly~* object that allows the playback of multiple streams of grains simultaneously. Inside the "*poly~ spindriftpoly"* subpatcher, there are twelve instances of *spindrift~* object and each voice has its own latching interval. By initiating each voice in *kslider* object (4 octave keyboard) various polyrhythms can be obtained. The poly~ spindriftpoly algorithm is displayed below at Figure 4.9.

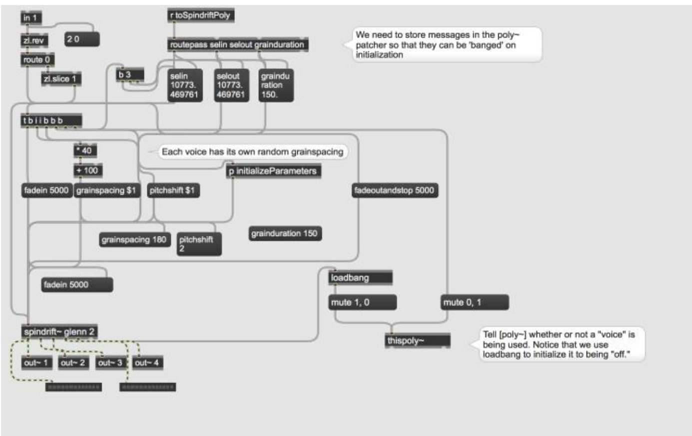

Figure 4.9: Patching view of the poly~ spindriftpoly algorithm.

---

3 http://www.michaelnorris.info/software/spindrift

Figure 4.10 shows the GUI view of Polyrhythm Generator in *MEMO*.

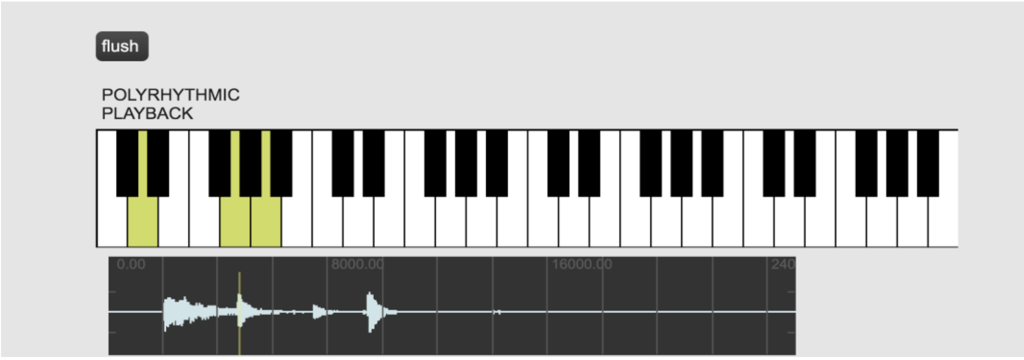

Figure 4.10: GUI view of Polyrhythm Generator in *MEMO*.

4.5. Time Stretch Module

This module uses *nw.grainphase~*[4] object that was developed by Nathan Wolek and it enables real-time application of Time Stretch by controlling grains with a phase signal that are generated by *phasor~* object. Besides, this object works in conjunction with *nw.phasorshift~* object that allows the production of overlapping grains for time-stretching and pitch-shifting algorithms. The audio content is received from the buffer~ object and the window buffer can be modified to change the envelopes. This object can receive four arguments; Frequency, Pitch, Gain and Buffer Offset. At Figure 4.11, the patching view of *nw.grainphase~* object is displayed.

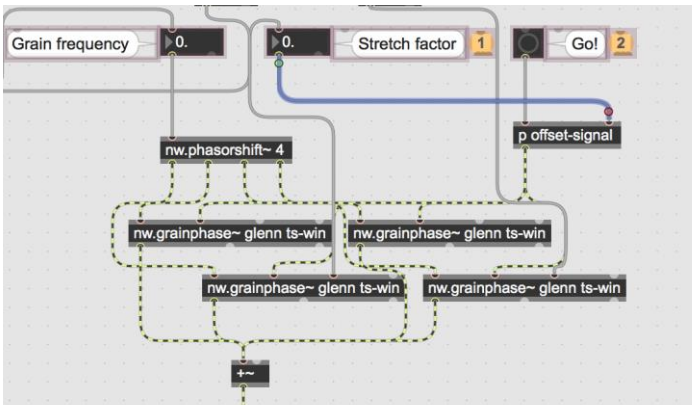

Figure 4.11: Patching view of *nw.grainphase~* object.

[4] https://github.com/nwolek/LowkeyNW

Figure 4.12 displays the patching view of Time Stretch algorithm in *MEMO*.

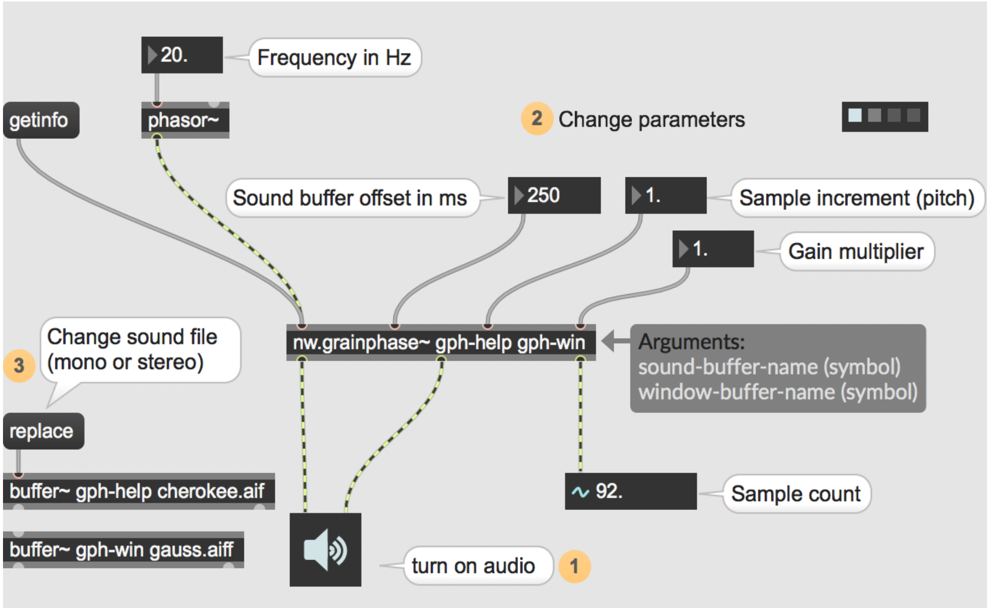

Figure 4.12: Patching view of Time Stretch algorithm in *MEMO*.

The stretch factor, which is a float value affects the linear scan of the line object, enabling *various* multiplications of the signal playback which will result in differences in length of the stretched signal.

The playback direction is determined by the sub-patcher "*offset-signal*" which enables a linear scan of the audio input by using line~ object. Also, the buffer information is determined through this sub patcher. Figure 4.13 displays the patching view of the "*offset-signal*" subpatcher.

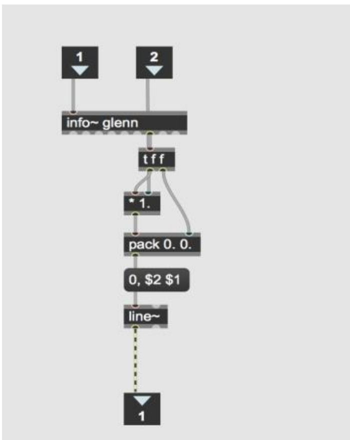

Figure 4.13: Patching view of "*offset-signal*" subpatcher.

Besides the main operation principals of Time Stretch module in *MEMO* I discussed above, Automatic Pitch Shifting is an extension that uses a *metro* object sending bangs every 200 ms, triggering the equal divisions of four octaves of an incoming note which is represented as 0.125, 0.250, 0.50, 1, 2. In this case, 1 stands for the actual octave value of a note. Moreover, *zl scramble* method randomizes the divisions I mentioned above, which creates rich stochastic textures with a wide range of sound spectrum. The patching view of Auto Pitch Shifting Extension in Time Stretch module is displayed below at Figure 4.14.

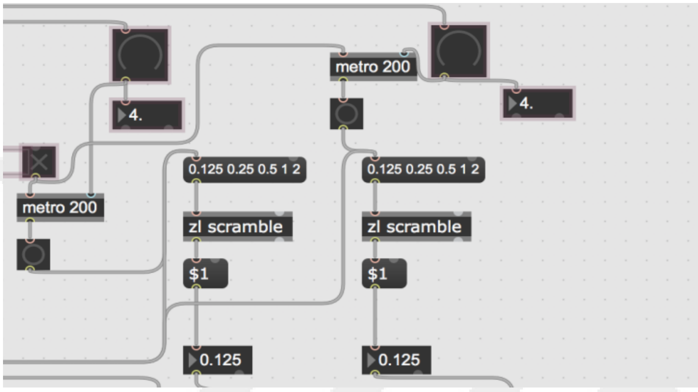

Figure 4.14: Patching view of Auto Pitch Shifting Extension in Time Stretch Module.

4.6. Granular Delay Module

This module works as a polyphonic granulator, which has its own internal buffer and uses *cm.livecloud~* object developed by *Circuit Music Labs* as a part of *Petra* package. It is connected to the outputs of Game of Life and Granular Synthesizer modules, which are reading and processing from the main buffer in real time. There are three arguments applicable for this object:

1) Name of the *buffer~* object and its size, this contains the window sample.
2) Maximum cloud size.
3) Maximum length of grains in milliseconds.

Besides these arguments, delay time, grain length, pitch, pan and gain can be controlled through minimum and maximum float values that are connected to the designated inlets of *cm.livecloud~* object. Figure 4.15 displays the GUI view of Granular delay module in *MEMO*.

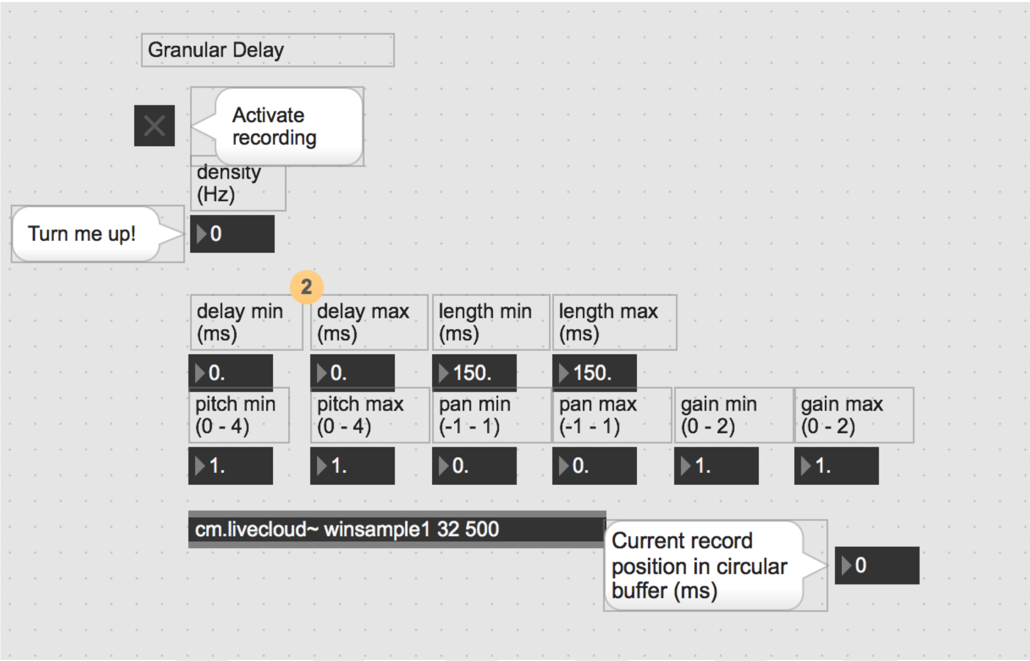

Figure 4.15: GUI view of Granular Delay module in *MEMO*.

5. CONCLUSION

In this thesis, having discussed the key theories of interactive music system design, the roles of interactive music systems, the definitions of interactivity, novel improvisation systems and audio synthesis methods, I employed a self-reflective, practice-based research approach to reflect my creative process as a performer/developer with regards to the development of *MEMO.* Also, talking about the aesthetic backgrounds, my personal motivations and roles that *MEMO* can undertake in a performance situation were crucial topics to talk about.

The main research problems in this thesis were;

- Determining *MEMO's* stance in the current academic landscape of interactive music systems as well as defining the relationship between *MEMO* and the selected novel interactive music systems;
- The adaptation of sound synthesis practices into a performance system that is using these methods real-time to give the impression of a dialogue between the performer and the computer;
- Finding the ideal balance between the autonomy and dependence with regards to the control of this system;
- Development of a novel improvisation system that offers self-reflective approaches for interactive musical performances.

The main operation principles of *MEMO*, which employs 6 different modules that are dependent on the main buffer, shows an example of how the sound synthesis methods I discussed throughout this thesis, can be used as components for an interactive music system design. Also, the ideal balance between the autonomy and the dependence has been established by the use of event schedulers such as randomizers and automatic parameter shifting methods and use of a MIDI controller that enables the performer to intervene the system in a performance instance. Moreover, the self-reflective approach undertaken in this thesis which sheds light on my aesthetic grounds and my personal taste, can be regarded as an approach to designing such interactive music systems. Besides, it was concluded that at this stage, *MEMO* has two main roles that are; being an instrument & hybrid agent. *MEMO* can be regarded as a mixture of Robert Rowe's taxonomy of "Score / Performance driven music systems", Todd Winkler's classification of "Conductor model" and Joel Chadabe's "Fly by Wire" model.

5.1. Outcomes of the Research

This research contributes to the literature by the reflection of design process as and the development of this novel interactive music system. In the Methodology chapter, this process has been explained stage by stage, starting from the underlying aesthetics to the patching processes in Max. Having developed this system, I would like to talk about the outcomes of this project as well as its advantages and disadvantages. The stance of this system in the current academic landscape of interactive music systems is a moderately interactive one. Since *MEMO'*s nature does not embody an intellectual model such as a machine learning method, *MEMO'*s interactivity degree can sometimes be left to a chance. This might create surprises during a course of a performance and as a performer with this system, I have found that this element of chance could be a spark of an inspiration to shape the way of the overall improvisatory performance. However, it can also leave the performer in an unexpected musical situation and the performer might have to overcompensate during the performance.

5.2.Future Directions

As discussed above, *MEMO's* degree of interactivity can be explained by Rowe's "Score/Performance driven" system model, Winkler's "Conductor model" and last but not least, Chadabe's interaction model "Fly-by wire" as explained in the Literature Review chapter. This interaction model does not include an autonomic intellect in which this system can have its own thoughts and own decisions. Everything that *MEMO* offers basically is a reflection and re-organization of the performer's improvisatory material. For future researches, the integration of a fully autonomous decision-making system can be taken into account by relating to the theories of improvisation and improvisation as a cognitive process. These cognitive processes can be replicated in digital domains with machine learning algorithms. With this idea, the interactivity degree in *MEMO* can be enhanced drastically. Apart from the integration of machine learning, another future direction that this study can take up is that the enhancements of the modules' of *MEMO's* generative algorithms. These enhancements can be realized in a way that either *MEMO* would become a fully obedient system that works in favor of the performer or the opposite, a fully autonomous agent that listens, reacts and creates its own material.

5.3. Final Remarks

With the purpose of reflecting my creative process of designing *MEMO*, as well as touching upon the roles of this system and the technical issues I have encountered, I attempted to contribute to the literature of interactive music system development through my own artistic perspective and this thesis can be considered as an artefact of my thoughts, inspirations and my journey with regards to the development of novel music systems that are performative and interactive.

6.BIBLIOGRAPHY

Assayag, G., Bloch, G. & Chemillier, M. (2006). OMax-Ofon. *Proceedings of the Third International Conference on Sound and Music Computing,* Marseille, France, 18-20 May.

Butler, **M. J.** (2014). *Playing with something that runs: technology, improvisation, and composition in DJ and laptop performance*. New York: Oxford University Press.

Carey, B. (2012). Designing for Cumulative Interactivity: The _derivations System, *Proceedings of the Twelveth International Conference on New Interfaces for Musical Expression,* University of Michigan, USA.

Carey, B. (2016). *_derivations and the Performer-Developer: Co-Evolving Digital Artefacts and Human-Machine Performance Practices.* (Phd thesis). University of Technology, Sydney.

Chadabe, J. (1983). Interactive Composing: An Overview, *Computer Music Journal vol.8 no. 01, 22-27*

Crawford, C. (2005). *Chris Crawford on Interactive Storytelling*. New Riders.

Ciufo, T. (n.d.). Design Concepts and Control Strategies for Interactive Improvisational Music Systems, 1-2.

De Poli, G., Piccialli, A. & Roads, C. (1991) Editors. Representations of Musical Signals. Cambridge, MA: The MIT Press.

Gabor, D. (1947). Acoustical quanta and the theory of hearing. *Nature*, 159 (40-44), 591-594.

Lepri, G. (2016). InMuSIC: An Interactive Multimodal System for Electroacoustic Improvisation. *In Other Words*, *18*(3), 5.

Lewis, G. (2000). 'Too many notes: Computers, complexity and culture in voyager', *Leonardo Music Journal*, vol. 10, 33-9.

Max (Version 8) [Computer software]. Cycling 74.

Mc.movement.studies [Computer software]. Cycling 74 & Bastan, D.

Manning, P. (n.d.). *In Electronic and Computer Music.* Oxford University Press.

Miranda, E. R. (2007). Cellular Automata Music: From Sound Synthesis to Musical Forms. In Eduardo Reck Miranda & J. A. Biles (Eds.), *Evolutionary Computer Music* (pp. 170–193). Springer London.

Musick, M., Forsyth, J. P., & Bittner, R. (2015). Building a Harmonically Ecosystemic Machine: Combining Sonic Ecosystems with Models of Contemporary Harmonic Language. 8-10.

Orio, N., Schnell, N., & Wanderley, M. M. (n.d.). Input Devices for Musical Expression: Borrowing Tools from HCI, *Computer Music Journal* vol. 26 no. 03, 62-76

Paine, G. (2002). 'Interactivity, where to from here?', *Organised Sound,* vol. 7, no. 03, 295-304.

Petra [Computer software]. Circuit Music Labs.

Roads, C. (2001) *Microsound*. Cambridge, MA: The MIT Press, 85-88.

Spindrift [Computer software]. Norris, M.

Rowe, R. (1993) *Interactive Music Systems.* MA: The MIT Press.

Truax, B. (1988) Real-time granular synthesis with a digital signal processor. *Computer Music Journal,* vol. 12 no. 02, 14-26.

Walraff, D. (1979) The DMX-1000 signal processing computer, Computer Music Journal Vol. 3, no 04, 44-49.

Winkler, T. (1998) *Composing Interactive Music, Techniques and Ideas Using Max*. Providence, MA.: The MIT Press.

Url-1: <https://www.sfu.ca/~truax/gran.html>, date retrieved 12.02.2020.

Url-2: <https://www.ultimate-guitar.com/lessons/guitar_techniques/how_to_get_frippertronics_with_delay_pedals.html>, date retrieved 12.02.2020.

APPENDICES

APPENDIX A: Video demonstration of *MEMO*

APPENDIX A

Video Excerpt: https://vimeo.com/422932296

CURRİCULUM VITAE

Name Surname : **Mehmet Korkmaz**

Place and Date of Birth : **İstanbul, 11/04/1993**

E-mail : **mehmetkrkmz93@gmail.com**

EDUCATION :

- **B.Sc.** : 2015, Marmara University, Faculty of Arts and Sciences, Department of Translation and Interpreting
- **M.A.** : 2020, Istanbul Technical University, Graduate School of Arts and Social Sciences, Centre for Advanced Studies in Music (MIAM)