Embodied Cognitive Science of Music

Modeling Experience and Behavior in Musical Contexts

Inaugural-Dissertation
zur
Erlangung der Doktorwürde
der Philosophischen Fakultät
der Universität zu Köln

Lüder Schmidt

Köln 2010

Contents

1 Introduction **1**

2 Cognitive Science of Music (CSM) **5**

2.1 State of CSM 10

2.2 Criticism of “Music Cognition” 11

3 Embodied Cognitive Science of Music **13**

3.1 Role of the Body 15

3.1.1 Spatial Hearing: Shape of the Body, Active Motion, Modality Interaction, and Neural Plasticity 15

3.1.2 Formation of Conceptual Structure: Evidence from Cognitive Linguistics 20

3.1.3 Expressive Movement 27

3.1.4 Interactive Technology: New Interfaces 30

3.1.5 Remarks 31

3.2 Notions of Embodiment 32

3.3 Agents as a Modeling Framework 35

3.3.1 Tropistic Agent 36

3.3.2 Hysteretic Agent 38

3.3.3 Knowledge-Level Agent 39

3.3.4 Complete Agent 41

3.4 Conclusion 44

4 Theory of Dynamic Systems **45**

4.1 Spring-Mass Pendulum 48

4.1.1 No Damping 53

4.1.2 Weak Damping 56
4.1.3 Critical / Strong Damping 59
4.1.4 Negative Damping: $\beta < 0$ 63
4.1.5 Intermediate Summary – Harmonic Oscillator 65
4.2 Resonance Model Analysis 66
4.2.1 Quality Management 74
4.3 van-der-Pol Oscillators 77
4.4 Gesture Analysis 87

5 Dynamic Systems: A Framework for CSM? 91
5.1 Agents as Dynamical Systems 91
5.2 Dynamical Turing Machine 96
5.3 Benefits of Dynamic Systems 99

6 Musical Robotics 101
6.1 Animated Sound Installations 104
6.2 Interaction in Musical Contexts 105
6.3 Human Music Performing Capabilities 109
6.4 Social Interaction and Synchronization in Musical Contexts 111
6.5 Communication of Musical Expression 112
6.6 Conclusion 113

7 Khepera III 115
7.1 Khepera III: Technical Description 117
7.2 Khepera III: Pd Interaction 122
7.3 Low Level C Programming 127
7.3.1 Decoding Messages: Command Table and Command Parser 127
7.3.2 Retrieving IR Data: `getproxir` → `proxIR` 130
7.3.3 Integrating the Communication Protocol: `kh3_proximity_ir` 131
7.3.4 Interacting with the dsPIC: `kh3_sendcommand` and `kh3_getcommand` 132

8 Synchronization 137
8.1 Synchronization: Technical Notion 139
8.2 Synchronization without Periodicity 142
8.3 Observing Synchronization 145
8.4 Entrainment and Interaction 150
8.4.1 Text of the Proposal 150

A “Robots can't . . . ”? 155

B Mathematical Supplements 157
B.1 Critical Damping, Initial Conditions 157
B.1.1 Position 1, Velocity 1 157
B.1.2 Position 1, Velocity -1 158
B.1.3 Position -1, Velocity -1 158
B.1.4 Position -1, Velocity 1 158
B.2 Strong Damping, Initial Conditions 159
B.2.1 Position 1, Velocity 0 159
B.2.2 Position 1, Velocity 1 160
B.2.3 Position 1, Velocity -1 160
B.2.4 Position -1, Velocity -1 160
B.2.5 Position -1, Velocity 1 161

C Khepera III: Driving Circles 163
C.1 Code Listing `k3_circle_test.c` 165

D Khepera III: Motion Tracking 171

E Khepera III: C-Source for Pd 173

Bibliography 187

List of Figures

2.1 Cognitive Science 6

3.1 Spatial Hearing 18
3.2 Notions of Embodiment 33
3.3 Cart in Maze 36

4.1 No Damping 56
4.2 Weak Damping 59
4.3 Critical Damping 62
4.4 Strong Damping 63
4.5 Negative Damping 64
4.6 Phase Shift Forced Oscillations 70
4.7 Standard Resonance Curves 71
4.8 Effective Resonance Curve 73
4.9 Curious Resonance Curve 75
4.10 Two trajectories of the van der Pol oscillator for $\gamma = 0.3$. 80
4.11 Oscillations of van der Pol oscillator with $\gamma = 0.3$. 81
4.12 Two trajectories for $\gamma = 5$. 82
4.13 Oscillations for $\gamma = 5$. 83
4.14 Oscillations for $\gamma = 10$ and $\gamma = 20$. 83
4.15 Attraction to Limit Cycle 84
4.16 Schematic Face 88

5.1 Agent-Enviroment Coupling 92
5.2 Coupling Nervous System – Body – Environment 93
5.3 Mutual coupling of within-agent sub-systems 94

6.1 Haile 106
6.2 Spherical Robot 108
6.3 Waseda Flutist 110
6.4 M[ε]X, taken from Burger / Bresin 2007 [46] 112

7.1 Khepera II Extended 116
7.2 Khepera III 118
7.3 K3 Control by Pd 123
7.4 Khepera III OSC Control 125

8.1 Phase 140
8.2 Synchronous Movement 144
8.3 Degrees of freedom for Keepon's movements. Taken from Michalowski / Sabanovic / Kozima 2007 [187], page 91. 146
8.4 Interaction(s) between one speaker and N listeners. 147
8.5 Facial display of a child interacting with Keepon, illustrating the importance of sequential analysis. Screen shots from Kozima and Michalowski's movie *Keepon dancing to Spoon's "Don't You Evah"*; for a link see Footnote 4. 148
8.6 Entrainment 151

C.1 Driving a circle with a Khepera III robot 164

D.1 MaxMSP / softVNS Patch 172

Chapter 1

Introduction

The considerations put down in the following chapters have grown out of previous work on modeling perceptual auditory processes, aimed at understanding (certain aspects of) music perception and music cognition.

This work draws on the anatomy and physiology of the auditory system, findings from psychoacoustics, and signal processing procedures (see Schmidt 2000 [241] for an introduction to basic concepts). Auditory anatomy and physiology provide hints for the architecture of a system intended to model the function of the peripheral auditory system, i.e. to produce a comparable output given the same input, for instance concerning different processing stages to be taken into account. A combination of physiological data (single and multiple cell recordings) and psychoacoustic measurements (e.g. discrimination of tones, masking patterns) is commonly used to obtain a detailed specification of the response characteristics of the peripheral auditory system. Formalisms provided by the signal processing literature allow to design appropriate procedures generating the desired output. Such a system, frequently called *auditory model* (e.g. Leman 1995 [171]), is typically implemented in the form of a filterbank. The outputs of the individual filters essentially constitute a time-varying spectral representation of the acoustic input and are sometimes interpreted as representing neural activity of the auditory nerve (as a so-called neural activation pattern, see e.g. Patterson / Allerhand / Giguere 1995 [213]).

Research on the perceptual organization of auditory input, dubbed *auditory scene analysis* (ASA) by Bregman 1990 [43], attempts to utilize rules inspired by Gestalt psychology in connection with informal applications of concepts from artificial intelligence to give an account of the way a listener arrives at a description of objects or events in the environment solely based on auditory information (for applications to music perception see Bregman 1990 [43], Chapter 5). The rules set forth by are ASA research are assumed to operate on some form of time-frequency representation of acoustical input, inviting a combination with research on auditory models as described above.

Accordingly, within *computational auditory scene analysis* (CASA) (e.g. Schmidt / Seifert / Eichert 1997 [245]; for a recent overview see Wang / Brown 2006 [300]), the attempt is undertaken to integrate rules provided by ASA with auditory models in the specification of systems actually performing tasks such as separating speech or music from noisy backgrounds or segregating different musical voices; for a system description addressing music listening see Scheirer 2000 [240].

Some problems, however, seem to arise in this approach:

- The rules of auditory scene analysis are formulated rather vaguely (e.g. Eichert / Schmidt / Seifert 1997 [76]), leaving room for situations in which competing rules may apply; little is known about the resolution of such conflicts (see van Valkenburg / Kubovy 2004 [293]). Thus, more investigation of the processes underlying phenomena described by Gestalt rules is needed; the same applies for the interaction of such processes.

- The a priori restriction to auditory data neglects the possible importance of information from other sensory domains for the phenomena to be described, as exemplified by the discussion of spatial hearing in Section 3.1.1.

- Neither the tasks solved by the systems nor the behavioral repertoire realized so far seem to be truly representative of humans behaving in musical contexts. Thus, besides neglecting intermodal interactions, there is a danger of implementing "fake functions" e.g. by over-emphasizing aspects not present to the same degree in humans or by missing aspects actually important for human behavior.

From such concerns, in the first place, the desire has arisen to turn towards artifacts operating in realistic contexts, integrating data from different modalities, and producing in real-time behaviors appropriate to the context of operation. This desire appears to converge with trends in cognitive science more generally to reconsider assumptions made about the interdependence of cognitive processes (see Chapter 2) and to take the situational context of a behaving entity more fully into account (e.g. Clark 1997 [56], 2001 [58]).

In Chapter 2, we will present a short and selective characterization of the *cognitive science of music* (CSM) and its relation to a traditional view of cognitive science (CS).

Chapter 3 will present arguments for a revision of the view of CSM described in Chapter 2 in an approach that has been called *embodied cognitive science of music* (Schmidt 2005 [242], 2007 [243]; cf. Pfeifer / Scheier 1999 [220]), take up a discussion of the term "embodiment", and introduce the notion of an *agent*.

Chapters 4 and 5 introduce some basic concepts of the theory of dynamic systems, which are discussed with respect to examples from research on rhythm perception and production, and try to embed the notion of agent presented in Chapter 3 within this theoretical framework.

Chapter 6 discusses some examples from *musical robotics*, implementing aspects of the theoretical ideas discussed so far.

A more detailed look at a specific robotic platform (Khepera III) to be integrated into musical interaction will be given in Chapter 7.

Chapter 8, finally, takes up again the discussion of the phenomenon of synchronization (started in Chapter 4), which is considered a crucial feature of interaction processes, and concludes with a proposal for future work. –

In small institutes, such as the Institute for Musicology at Cologne University, the continual engulfment of scientific personnel in administrative chores is necessarily detrimental to the scientific profile of the institution. A rare exception may be the restructuring of the curriculum concerning the cognitive science of music in the course of the implementation of the BA/MA system. The ideas presented in this text had a chance to enter into the discussions resulting in the module descriptions for the BA/MA curriculum that was launched in the winter term 2007 at the named institute.

The appendices contain technical material referenced in the text.

Chapter 2

Cognitive Science of Music (CSM)

The scope of *cognitive science* (CS) is commonly circumscribed by giving examples of research considered relevant or by listing scientific disciplines contributing to the examination of cognitive phenomena. These lists may vary somewhat, but a core membership of psychology, artificial intelligence / computer science, linguistics, neuroscience, and philosophy seems to be generally accepted (see e.g. Gardner 1985 [91], Miller 2003 [189], Wilson / Keil 1999 [318], Boden 2006 [37], Strube 2001 [264]). Interrelations between these disciplines are commonly visualized by / with reference to the so-called *cognitive hexagon*, that was sketched in a State of the Art Report for the Sloan Foundation in the year 1978 (according to Gardner 1985 [91], pages 36–37). In the graphical display, lines between the names of the disciplines are taken to represent interdisciplinary connections. In the version reproduced in Figure 2.1 (taken from Miller 2003 [189]), only those connections depicted as "strong interdisciplinary ties" in the original version (Gardner 1985 [91], page 37) are taken up; a further set of broken lines representing "weak interdisciplinary ties" establish full connectivity in the 1978 version (ibid.). The situation of cognitive science as a discipline in 1985 is characterized by Gardner as there being "as yet no agreed-upon research paradigm – no consensual set of assumptions or methods [. . .]" (ibid.). Accordingly, up to the present a definition is hardly given, and even in *The MIT Encyclopedia of the Cognitive Sciences* (Wilson / Keil 1999 [318]) an entry for cognitive science is conspicuously lacking.

Despite the absence of a common paradigm, Gardner (ibid., pages 38–45) offers a set of five "key features", two of which are considered as "core assumptions" (ibid. page 38) of CS. The core assumptions state that cognitive science is crucially involved with mental representations such as "symbols, schemas, images, ideas" (ibid., page 39) and that computers play a central role in cognitive science research

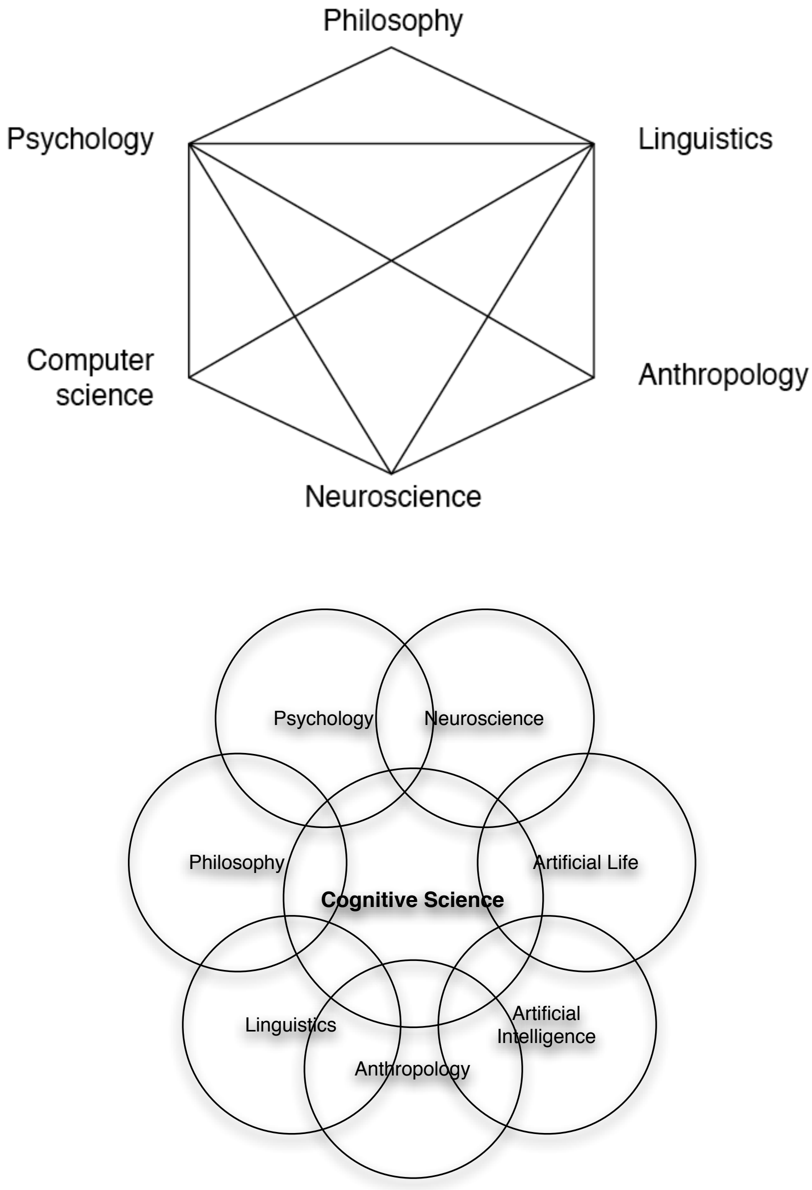


Figure 2.1: Different visualizations of cognitive science:
Top panel: *Cognitive Hexagon*, taken from Miller 2003 [189].
Bottom panel: Alternative visualization; the disciplines entered in the figure are based on Boden 2006a [37], page xxxv.

(ibid., pages 41–42). The other features concern the "De-Emphasis on Affect, Context, Culture, and History", the "Belief in Interdisciplinary Studies", and the "Rootedness in Classical Philosophical Problems" (ibid., pages 41–42). Three of these features have elicited criticism, contributing to the interest in embodied cognitive science (see below).

In an attempt to reconstruct CS as a coherent – albeit immature – scientific discipline, Barbara von Eckardt 1993 [295] identifies different sets of assumptions and related questions that characterize the domain of CS, the fundamental theoretical approach to the domain, and resulting methodological commitments. The assumptions are closely related to the features described by Gardner.

More specifically, von Eckardt spells out the framework of CS by a set of three domain-specifying assumptions (D1–D3) and two substantive assumptions (SA1, SA2), some of which are further differentiated, and eleven methodological assumptions (M1–M11). In addition, four schemata for questions to be answered by research in cognitive science are provided (for a short overview see von Eckardt 1993 [295], pages 45–56). According to the assumption D1–D3, the domain of (adult normal typical) cognitive science is formed by the human cognitive capacities (D1: Identification Assumption) which are characterized by the set of properties of being intentional, pragmatically evaluable, coherent, reliable, and productive (D2: Property Assumption), and "make up a theoretically coherent set of phenomena, or a *system*" (D3: Grouping Assumption; ibid., pages 47–48). The two substantive assumptions essentially coincide with Gardner's core assumptions, stating that "the human mind/brain is a computational device (computer)" (SA1) as well as a "representational device" (SA2; ibid., page 50).

We will not enter into a detailed discussion of the independence of the assumptions presented. As an example, we will only point out, that the independence of the two substantial assumptions seems to be subject to debate: according to Churchland / Sejnowski (1992 [55], page 62), a physical system is considered a computational system only if its states "can be seen as representing states of some other systems"; Thagard 2005 [277] quotes the slogan "No computation without representation".

The methodological assumptions M1–M3 can be taken to reflect Gardner's "De-Emphasis on Affect, Context, Culture, and History": According to M1, it is sufficient to concentrate on the individual, i.e. social and cultural contexts can be safely disregarded in the investigation of cognitive phenomena, M2 claims that cognitive capacities are sufficiently autonomous from aspects such as affect and personality to warrant independent study, and M3 assumes a partitioning of cognition into individual capacities again allowing study in isolation.

In the light of the considerations presented in the following chapters, these three assumptions appear to be not just methodological. Rather, they seem to touch upon the very nature of cognitive activity as expressed by the recurrent theme of coupled, interactive processes at various levels.

More genuinely methodological assumptions are encoded in M4–M8, dealing with the assumption of normal and typical cases of cognitive phenomena (M4, M5), sound explanatory strategies and the commitment to usual canons of scientific methodology and empirical research (M6, M7), and the necessity to integrate contributions from all "subdisciplines of cognitive science" (von Eckardt 1993 [295], page 55).

The remaining three assumptions (M9–M11) on the one hand grant a special status to the "subdiscipline" human neuroscience as providing constraints to be observed in information processing accounts of cognitive phenomena (M9), on the other hand it is assumed that information processing theories can give explanations for features that cannot be explained on the basis of neuroscientific processes (M11). Actually, this question appears to concern the integration of theoretical results from different scientific domains; the topic will briefly be taken up below.

More recently, in her monumental history of the field of cognitive science entitled "Mind as Machine", Margaret Boden (2006a,b [37, 38]) presents a detailed account of the contributions of different disciplines (taken up in the bottom panel of Figure 2.1) to the investigation of cognitive phenomena. She, too, stresses the need to *integrate* the views from these "cognitive sciences" (Boden 2006a [37], page 12), defining CS as "the study of *mind as machine*" that covers "all aspects of mind and behaviour" and draws "on many disciplines" (ibid., page 9)[1].

The need to integrate contributions from different disciplines within cognitive science expressed unanimously opens up questions concerning the relations of the disciplines taken into consideration to CS as a whole and the specific ways findings from these disciplines have bearing on each other or can be merged into coherent theoretical accounts.

The visualization in the cognitive hexagon (see Figure 2.1, top panel) may give rise to the impression that CS is *constituted* by a set of "subdisciplines" (e.g. von Eckardt 1993 [295], page 55), each belonging completely to the field, that are more or less tightly interconnected. According to this interpretation, all research within any of the subdisciplines is to be considered to belong to CS, and there will be no "cognitive science proper" set apart from the subdisciplines. Clearly, such a – purposefully exaggerated – view is inappropriate: As an example, computer science research pertaining to the optimization of industrial production will not primarily be relevant for the investigation of human cognitive capacities; furthermore, the intuitions about a specific domain of CS captured e.g. in von Eckardt's domain-specifying and substantive assumptions will be violated. Therefore, we have tried to prepare an alternative visualization (see Figure 2.1, bottom pa-

[1] In the light of this characterization, a more appropriate title for her work might have been "*Man* as Machine".

nel), conceiving of CS as an independent, self-contained field partly overlapping[2] with the partner disciplines. A problem arising with this kind of view is to single out the specific research of CS; resorting to the assumptions formulated by von Eckardt will not do as these are taken to be shared by cognitive scientists working within any of the "subdisciplines".

The way to integrate findings from different disciplines is illustrated e.g. by Gardner (1985 [91], Chapters 10–14) by way of examples describing successful research crossing disciplinary boundaries, either by institutional cooperation or by personal effort. A more precise account of how theoretical approaches can merge, e.g. how constraints from neuroscience enter into information-processing theories, nevertheless appears desirable. A viable approach may be offered by the concepts of local theory, inter-theoretic relations, and theory nets as expounded by Balzer / Moulines / Sneed 1987 [23] or Balzer 1997 [22]. A more detailed and rigorous exposition of these ideas is beyond the scope of the present text. We will, however, repeatedly come across the problem of transferring results from one scientific domain to the other, see e.g. Sections 4.2, 4.3, or 8.1. –

In two ways Boden's definition of CS appears to be broader than the one given by von Eckardt: the "computational device" is explained by von Eckardt with reference to a computer as characterized in the standard literature on computer science (von Eckardt 1993 [295], page 105)[3], whereas the term "machine" used by Boden seems to apply to a wider range of artifacts. The human cognitive capacities are traditionally taken "to refer to such activities as thinking, conceiving, and reasoning" (Reber / Reber 2001 [230], entry "cognition"), the scope of "all aspects of mind and behaviour" again including a wider range of phenomena.

The approach of an *embodied cognitive science* as discussed in the following chapters is related to the different approaches apparent in these definitions, claiming to transcend the limits set by the first definition in a substantial way (e.g. Pfeifer / Scheier 1999 [220], Pfeifer / Bongard 2006 [218]).

A common aspect of the two approaches is the reference to technical artifacts, which on the one hand provides theoretical concepts for investigations, on the other hand creates the need to be explicit in the formulation of theories which then can be implemented and tested by the design of model systems, e.g. computer simulations.

According to the stance taken here, the challenge posed by becoming explicit in theorizing about and by modelling all aspects of mind and behavior relevant to "music" should be taken as seriously in cognitive science of music (CSM) as the possible theoretical stimulations by ideas adopted from "cognitive sciences".

---

[2]Each of these attributes would require further discussion.

[3]The approaches described in a (more) recent anthology on cognitive modeling (Polk / Seifert 2002 [227]) remain within the confines staked out thus.

2.1 State of CSM

According to the ambitions implied by this view, CSM should come up with (designs for) artifacts that incorporate abilities also exhibited by humans in contexts deemed to be musical. A rough summary of research relevant to CSM – e.g. on music cognition –, however, can only point to modeling attempts that look at restricted areas of musical structure, take as input highly simplified representations of music and produce output that needs to be interpreted in terms of data gained in empirical investigations or in traditional analyses of musical structure. Converging interpretations of simulation output and empirical data are taken as evidence supporting the theoretical assumptions implemented in the model.

Typical examples of research in music cognition include investigations of attributes of local musical events – tones – such as pitch and timbre: data from psychological tests, such as (dis-)similarity judgements, are transformed into geometrical constructions, which in turn are interpreted as (models of?) internal representations of the attributes in question (e.g. on pitch see Shepard 1982 [250], on timbre Grey 1977 [104] or Donnadieu 2007 [69]). The integration of local events can be construed after similar lines: geometrical configurations which are regarded as visualizing mental representations of mutual relations of musical features are derived from judgements about elements fitting into a context, extensively described for the case of tonal organization by Krumhansl 1990 [160]. Other approaches rely on rule systems inspired by linguistic and / or gestalt psychological considerations to derive descriptions of sound scenes in general (Bregman 1990 [43]) or more specifically musical structure (most prominently Jackendoff / Lerdahl 1983 [175]), which are again interpreted as a listener's internal representations.

Corresponding modeling approaches are occasionally classified according to the well-known (well-founded?) opposition of connectionism and symbolic AI (e.g. Toiviainen 2000 [282]). Experimental work based on similarity judgement is more easily associated with neural network models as exemplified by Leman 1995 [171]: There, output of a so-called auditory model is used as input to a Kohonen self-organizing map. After a sufficient amount of training, tonal centers could be demonstrated to arise, i.e. areas within the network that responded most strongly to stimuli within a specific musical key. The topology of the tonal centers could be interpreted in terms of relationships familiar from traditional music theory (circle of fifths) and was compatible with the results of Krumhansl 1990 [160]. In a more recent example, Krumhansl / Toiviainen 2003 [161] used data derived from judgements of key distances to train a Kohonen map. The network was incorporated into a key-finding model operating on a highly reduced musical input (pitch numbers ranging from 1 to 12 and onset / offset times), whose output then was compared with judgements of musically trained listeners.

A similarly reduced input ("piano roll representation") is used by Temperley 2001 [276] in the implementation of a system based of preference rules (inspired by Jackendoff / Lerdahl 1983 [175]) to automatically provide analyses of musical pieces. Scheirer 2000 [240] integrates rules from auditory scene analysis and more general psychoacoustic data to automatically extract musical features from the output of an auditory model, calling his procedure musical scene analysis (ibid., chapter 5).

In summary, the systems described may be characterized as follows: A rather restricted set of musical features is addressed, mostly conforming to the "narrow" definition of cognitive science described above and possibly leaving out (more?) important aspects of musical experience (implementing an equivalent of von Eckardt's methodological assumptions M1 and M2). There is a tendency to study and model these features in isolation, aiming at self-contained descriptions or explanations (assumption M3). This form of particularization, however, may entail neglecting issues of the coherence and closure of the musical domain. Despite the argument of converging evidence (a variant of the methodological assumption M10), system performance remains difficult to evaluate as a model of human cognitive processes (see Wang / Brown [301] for a detailed discussion), because input as well as output is quite remote from realistic situations including musical stimulation and human music-related behavior.

For these reasons it appears desirable to integrate modeling attempts into systems, in the following chapter introduced as agents, that can exhibit more or less appropriate behavior within musical contexts.

2.2 Criticism of "Music Cognition"

Extending somewhat the reservations expressed with regard to CSM, some points of criticism against the cognitive (or cognitivistic) approach to music perception / music production or musical experience will be taken up again. Partly, at least, criticism seems related to the perpetuation of the traditional notion of cognition apparently inherent in the research described in the previous section. Cognitive psychology of music primarily dealing with abstract, intellectual(istic) features of musical structure such as tonal relationships / tonal hierarchies, timbral spaces, or grouping and segmentation within sequences of tones, is considered to be leaving out aspects that are considered more central themes of musical experience. More specifically, two broad veins of criticism can be discriminated:

1. Within part of german Musikpsychologie (psychology of music) a somewhat anti-naturalistic attitude seems to prevail. It is argued (e.g. Gembris 1999 [92]) that cognitive psychology / cognitive science of music / music cognition[4] by its very definition (see above) focusing on abstract mental processes of the individual

[4]Usually no clear distinction is drawn between: music cognition, cognitive psychology of music, and cognitive science of music.

cannot take up issues involving social and cultural aspects of musical experience. Moreover, such phenomena are considered not susceptible to the methodology of computational modeling.

2. Other aspects taken to be lacking by definition in CSM concern the widely accepted connection of musical experience with emotional processes as well as the relation of music to corporeal motion. The traditional separation of thinking, planning, and problem solving from emotional and other corporeal processes, however, has lately been rejected within areas of research as disparate (at face value) as linguistics / philosophy on one side and neuroscience on the other side. Most prominently, Damasio 1994 [65] and LeDoux 1996 [170] have demonstrated the important role of emotional processing for human decision making. As the coupling of emotional processes to specific brain structures could be shown, not only the view of rationality is challenged, but also the role of neuroanatomy for human mental structure is further established. (Regarding emotional processes connected to musical experience, see e.g. Peretz 2001 [216].) The influence of corporeal interaction with a structured environment on the formation of concepts in humans, thus shaping human thought, is extensively discussed by Lakoff and Johnson (see Section 3.1.2 for further discussion).

As will become apparent in the beginning of the following chapter, these critical remarks can be considered as specific examples for a more general set of challenges to cognitive science, for which an *embodied cognitive science (of music)* aspires to present an answer.

Chapter 3

Embodied Cognitive Science of Music

Thagard 2005 [277] summarizes problems encountered with a traditional approach to cognitive science, which he characterizes as "Computational and Representational Understanding of Mind" (abbreviated to CRUM), in a list of seven major challenges (ibid., page 140):

1. *The Brain Challenge*: CRUM ignores crucial facts about how thinking is performed by the brain
2. *The Emotion Challenge*: CRUM neglects the important role of emotion in human thinking
3. *The Consciousness Challenge*: CRUM ignores the importance of consciousness in human thinking
4. *The Body Challenge*: CRUM neglects the contribution of the body to human thought and action
5. *The World Challenge*: CRUM disregards the significant role of physical environments in human thinking
6. *The Dynamic Systems Challenge*: The mind is a dynamic system, not a computational system
7. *The Social Challenge*: Human thought is inherently social in ways that CRUM ignores

Several of these challenges will strike familiar tunes, given the recent interest in the cognitive neuroscience of music (e.g. Peretz / Zatorre 2003 [217], Levitin / Tirovolas 2009 [178]; popular introductions: Spitzer 2002 [259], Levitin 2006

[177]), music and emotion (e.g. Juslin / Sloboda 2001 [144]), or embodied music cognition and gesture research (e.g. Leman 2008 [172], Godøy / Leman 2010 [96]) and the surge of new sensor-based interactive technology (e.g. conferences on New Interfaces for Musical Expression NIME[1]).

This short sample of research already indicates that the challenges listed by Thagard cannot be considered in isolation: expressive movements are commonly seen as one aspect of emotional processes occurring in different parts of the body including the brain, are mediated via physical interaction in the world (e.g. via sensor technology) and constitute an integral part of communicative social processes. Further, it is suggested that the theory of dynamic systems (see Chapters 4 and 5) may provide a broad general framework *within which* the processes addressed can be explored and integrated or *against which* at least the claims made can be checked.

In fact, the theory of dynamical systems seems to have gained acceptance as a framework for cognitive science since the 1990s not only for the description and explanation of observed phenomena (e.g. the contributions in Port / van Gelder 1995 [228]; Ward 2001 [302], Schöner 2008 [247]) but also for the specification of systems acting within an environment (Pfeifer / Bongard 2006 [218], in particular pages 93-94).

The claim inherent in Challenge 6 raised against CRUM that dynamic systems are not computational seems to hinge on a specific interpretation of the term computation attributed to traditional cognitive science (see Anderson 2003 [7]; cf. Section 3.3.4, Footnote 30). Some tentative remarks on this topic will be offered in Section 5.2.

The stance taken here is that adopting ideas of what will be termed "embodied cognitive science" following Pfeifer / Scheier 1999 [220] (see also Clark 1999 [57]) to music research may well lead to modeling approaches integrating and extending in an embodied way previous research in music cognition. Although Pfeifer / Iida 2004 [219] now address their field as embodied artificial intelligence, we will here retain the name *embodied cognitive science of music* to stress the interest in the study of music-related human behavior and experience rather than building systems that mainly fulfill certain specifications.

[1] `www.nime.org`

3.1 Role of the Body in Cognitive Processes

We will extend the introductory remarks by discussing some examples from a growing amount of evidence that the investigation of cognitive phenomena can benefit from taking into account conditions and processes that were previously not regarded as pertinent to cognition proper: Since the 1980s, the role of corporeal interaction and *embodiment* of an agent (human, animal, or artifact) has increasingly come into focus from a wide range of perspectives.

3.1.1 Spatial Hearing: Shape of the Body, Active Motion, Modality Interaction, and Neural Plasticity

Within theories of spatial hearing, explicit reference to the human (or animal) body has always played an important role. Geometrical considerations are used to explain differences in the sound field at the eardrums depending on the direction to a sound source relative to the listener's head; these differences are interpreted as spatial cues to be evaluated by the listener's auditory system.

Early approaches, such as the "duplex theory" (see Warren 1999 [303], pages 30 – 33, for a concise overview) rely on simplifying assumptions concerning the human head. The head is construed as a sphere carrying the ears at exactly opposite positions, effects of irregularities and protrusions such as nose and pinnae are disregarded. The spatial cues attributed to these geometric properties are *interaural time differences* (ITDs) due to different distances to the ears for sound sources outside the median plane between the ears, and *interaural level differences* (ILDs) caused by shadowing effects of the head.

Problems arise, however, from the simplifications introduced. Because of the symmetrical shapes assumed, there will be multiple directions that give rise to the same values for the interaural differences: All sound sources situated in the median plane or on a surface, that for large distances from the head approaches a cone centered around the axis through the two ears, will produce the same interaural time and intensity differences; as a consequence, their directions can not be distinguished on the basis of these theoretical assumptions. The associated perceptual phenomenon is known by the name "cone of confusion" (see Blauert 1997 [35], page 179).

We will give a short description of two classical ways to amend these problems. The first approach is to take into account the effects of factors breaking the symmetries, the second integrates the effects of active head movements as a means to disambiguate otherwise ambiguous spatial cues. Following that, we will briefly discuss some recent findings in auditory physiology that seem to combine both these approaches.

Most obviously, symmetry is broken by the details of the head ignored in the assumption of spherical shape. One important factor, taken to influence the sound field at the eardrums, is the complex shape of the pinnae. The various structures of the pinnae cause multiple reflections and diffractions of the sound waves arriving at the ears introducing various delay times that in turn are the basis of a pinna-specific filtering effect (for a short review see Warren 1999 [303], page 45 – 48). More precisely, the pinnae constitute a complicated acoustical filtering mechanism characterized by an individual transfer function[2] depending on the angle of incidence. Other measures influencing the sound field at the eardrums include size and shape of the head, but also shape, material, and dimensions of the upper body and even the height of the head above the floor (e.g. Algazi / Duda / Thompson 2002 [6]; Algazi et al. 2002 [5]; Angel / Algazi / Duda 2002 [8]). Functions describing all these influences of the human body on the sound field at the eardrums are called *head-related transfer functions* (HRTFs).

Anthropometric measures taken to underlie head-related transfer functions have been incorporated in dummy-heads used for stereophonic sound recording (see Blauert 1997 [35], Chapter 4.5.2) and in manikins used for measurement purposes such as the KEMAR[3] (Knowles Electronics Manikin for Acoustic Research; see the contributions in Burkhard 1978 [48], recent specifications in [102]). In virtual reality environments, head-related transfer functions are employed to create realistic spatial impressions by presenting sound via headphones (e.g. Begault 1994 [31]).

The first discussion of *active head movement* to obtain unique directional information about a sound source is attributed by Blauert 1997 [35], page 180, to van Soest 1929 [292]. According to geometrical considerations,

> [o]ne obtains the cues for a number of lateral angles[4] for the same sound direction by turning one's head while the sound is being given. Geometrically, a sequence of lateral angles obtained in this manner completely determines a given direction [. . .]. (Wallach 1939 [297], page 270)

In the case of rotating the head around the vertical axis, lateral angle will change most (i.e. by the same amount as the angle of rotation) for sound sources located in the horizontal plane containing the ears; for sound sources placed directly above the head, rotating around the vertical axis will not change lateral angle. In general, the change of lateral angle for a given rotation around the vertical

[2]A transfer function in this context is a function that relates filter gain and phase delay to the frequency of a signal component, see e.g. Schmidt 2000 [241], Chapter 7.

[3]For an example measurement situation at the Parmly Hearing Institute see `http://www.parmly.luc.edu/parmly/behav_psych_resrch.html`.

[4]Lateral angle here refers to the angle between the direction to the sound source and the axis through the ears.

axis of the head and a certain angle of elevation (specifying the direction to the sound source relative to the horizontal plane) can be computed from geometrical considerations[5].

Using the experimental setting displayed in the top panel of Figure 3.1, the theory was tested by reverse argument: If the head of the subject is connected to the array of loudspeakers in such a way that turning the head by a certain angle will cause the location of the sound source to change by the same angle, no differences in lateral angle will arise and the sound source should be perceived directly above the head. If the connection to the loudspeaker array produces an angular displacement of the sound source less than the angle of rotation, the sound source will be expected to be perceived at an elevation between 0° and 90° according to the ratio of the angles. The test of the first condition was "successful with all observers who were able to localize sounds above under ordinary circumstances (10 out of 17)" (Wallach 1939 [297], page 272); values comparing perceived and theoretically expected angles of elevation are given in Table I (ibid.).

To evaluate the effects of head movement on the spatial cues, information about position of the head and auditory information must be combined, i.e. a motional theory of spatial hearing of the kind described here requires the integration of different sensory modalities in the formation of a percept.

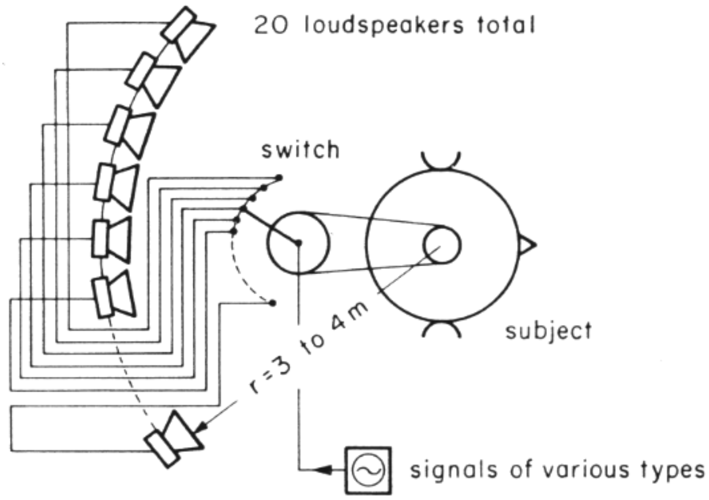
In a series of further experiments – summarized in Blauert 1997 [35], pages 189–191 – Wallach tried to determine the relative influence of postural / proprioceptive information, responses from the vestibular system, and visual cues. Strongest influences was found to be exerted by vision, followed by information provided by the vestibular system only. Immobilizing listeners' heads with respect to the torso and moving listeners passively did not alter performance in the experiments as compared to active rotation of the head.

In another set of experiments devised by Klensch 1948 [155], here reported after Blauert 1997 [35], pages 185–187, change of the sound field at the eardrums was separated from head movement by introducing a pair of funnels, that were connected to the external ear canal by rubber tubes of equal length, functioning as "mobile pinnae." Thus, it was possible to produce changing sound fields at the eardrums keeping the head immobile or moving the head without changing the sound fields. Even combinations of head / funnel movements could be produced that individually would have given rise to contradicting percepts. A series of

[5] The formula specifying lateral angle depends on the choice of coordinate system. In a coordinate system relative to the head, lateral angle ψ is related to elevation ϑ and azimuth φ as defined conventionally (see Blauert 1997 [35], page 14) according to the formula

$$\cos\psi = \cos\vartheta \sin\varphi,$$

which is equivalent to the formula $\sin(90° - \psi) = \cos\vartheta \sin\varphi$ given by Wallach 1939 [297], page 272.



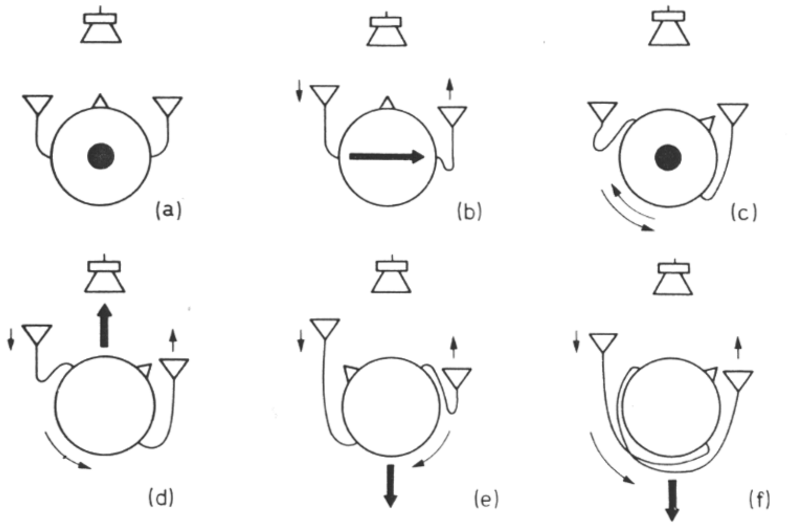


Figure 3.1: Top: Head tracking, experimental setup devised by Wallach (e.g. 1939 [297]), after Blauert 1997 [35], page 187.

Bottom: “Mobile outer ears” devised by Klensch (1948), after Blauert 1997 [35], page 186.

experimental conditions is shown in the bottom panel of Figure 3.1. Light arrows indicate direction of movement of head and funnels, bold arrows and dots mark the perceived location and direction of movement of the sound source. In the top row, the sound source is perceived to be located / moving inside the head, due to the removal of effects produced by the natural head-related transfer functions. In these experiments, too, results seem to support the basic assumptions of the motional theories of spatial hearing.

Although giving some hints as to which kinds of sensory data are combined, these early theories do not explicitly address the underlying processes, i.e. the ways in which the corresponding sensory systems are thought to interact. There is, however, a substantial body of recent research in (auditory) neurophysiology that investigates the integration of sensory data in non-human animals such as cats, barn owls, gerbils etc.

An early locus for the integration of multisensory data in the auditory system, that has attracted attention, is the dorsal cochlear nucleus (DCN) of the cat. Besides auditory nerve inputs, innervation has been shown to relate to pinna orientation (Kanold / Young 2001 [146]) as well as vestibular and further somatosensory information (Oertel / Young 2004 [209]). The DCN is assumed to process wideband spectral characteristics of the audio signal induced by head-related transfer functions (May 2000 [184]). In particular, relevant cues seem to be spectral edges (Reiss / Young 2005 [231]) or, more specifically, spectral notches related to the pinna (Imig et al. 2000 [126]). As an underlying mechanism, wideband inhibition is discussed by Hancock / Voigt 1999 [109].

The structure of neural circuitry in the DCN resembles that found in the cerebellum (Oertel / Young 2004 [209]), a possible function of which is "predicting consequences of sensory events" (ibid., page 108). Movement of the pinnae will change the HRTF. Thus, the cerebellum-like structures are hypothesized to establish a "form of sensory-motor coordination, for optimizing auditory processing. This hypothesis is similar to the hypothetical role of the cerebellum for sensory-motor coordination [...]" (Young / Davis 2002 [324], page 197).

As these findings illustrate, sensory integration not only occurs at an early stage in auditory processing, but also appears to be an integral part of the process instead of an a posteriori combination of cues independently derived in different sensory systems. Another interesting aspect is the inclusion of elements of a predictive mechanism that in addition may be shaped by experience (evidence and mechanisms of plasticity are discussed by Oertel / Young 2004 [209], pages 104–105) similar to those observed in movement control, which are discussed as central to embodied artificial intelligence by Holland 2004 [121].

As another site of sensory interaction experience-dependent neural plasticity, the inferior colliculus (IC) of the barn owl has been extensively studied. Interaural time difference has been found to be represented in the central nucleus of the inferior colliculus (ICc) (Wagner / Takahashi / Konishi 1987 [296]). Under normal conditions, this spatial representation is preserved in projections to the external nucleus of the inferior colliculus (ICx), which further projects to the optic tectum (considered homolog to the superior colliculus in mammals) to form a combined auditory / visual map of space (Brainard / Knudsen 1993 [40]). Knudsen and co-workers (Brainard / Knudsen 1993 [40]; Knudsen / Zheng / DeBello 2000 [157]; Knudsen 2002 [156]) studied the effects of barn owls wearing prismatic spectacles, that shifted the visual field to the left or right, on the maps found in the neural pathways of the owls. Best ITD responses in the optic tectum were found to change such that they corresponded to the change of visual representation. In the ICx, too, changes were found in best ITD response that could account for the changes observed in the optic tectum. The mapping of ITD in the ICc, however, was found to remain unchanged. These results indicate that in the brainstem there is a strong interaction between optical and auditory processing of spatial cues that again is modified by experience; the site of plasticity is assumed to be the external nucleus of the inferior colliculus, and visual input in these experiments appears to dominate auditory information.

These examples clearly indicate the existence of multisensory integration and experience-dependent neural plasticity in the early stages of processing of different cues related to the "auditory" perception of events in space, processes otherwise considered low-level.

3.1.2 Formation of Conceptual Structure: Evidence from Cognitive Linguistics

Whereas the examples in the previous section rather directly deal with corporeal processes that may play a role in auditory perception and thus contribute to the experience of music, further evidence for the role of the body in music-related contexts may be gained from analyzing the way humans verbalize music-related experience and musical structure. In the following, we will take up ideas from cognitive linguistics and sketch a way to extend them into music research, fully aware that a comprehensive treatment is beyond the scope of this work. Starting from detailed analyses of the use of metaphorical expressions as representative of *conceptual metaphors*, which in turn are taken to reflect underlying conceptual structure (e.g. Evans / Green 2006 [81], Chapter 9), and their role in the understanding of everyday experience, Lakoff and Johnson (1980 [164], 1999 [165]) advance the claim that a substantial part of even quite abstract human thinking is grounded in the experience of physical interaction; in more recent studies, the analysis has been extended to philosophical topics such as the foundations of mathematics (Lakoff / Núñez 2000 [166], Núñez 2004 [208]). Some examples relating to music analysis will be mentioned below.

The framework of conceptual metaphor as presented in Lakoff / Johnson 1999 [165], Chapter 4, comprises four central components:

1. the notion of conflation,
2. the theory of primary metaphor,
3. the neural theory of the formation of metaphor,
4. the theory of conceptual blending.

The idea of *conflation* is based on observations by C. Johnson on the acquisition of language by small children (Lakoff / Johnson 1999 [165], pages 46 and 48 – 49). It is hypothesized that in early childhood experiences related to different domains that occur simultaneously are not differentiated but conflated. As a paradigmatic example, the authors refer to the experience of warmth from being held giving rise to a feeling of affection; the repeated co-occurrence of these experiences is considered to form the basis of metaphorical expressions such as "a *warm* smile" (ibid. page 46, emphasis original). This kind of experience, however, may be quite different from the case of language acquisition as investigated by C. Johnson (1999 [138]) and thus C. Johnson's notion of conflation may not be appropriate as a basis for Grady's approach as claimed by Lakoff / Johnson (1999 [165]).

Primary metaphors, according to Grady (1997 [101], pages 19 – 26) arise from *conceptual binding* of co-occurring distinct aspects of what he calls *primary scenes* (subjective experiences of recurring *basic events*, ibid. page 23), possibly to be followed by a step of *deconflation* in case of very tight binding (*conflation*?). In this process, it is claimed by Lakoff / Johnson that "[. . .] everyday experience should lead to the automatic formation of hundreds of primary metaphors that pair subjective experience and judgment with sensorimotor experience" (1999 [165], page 49). Primary metaphors are taken to be "simple, atomic components" entering into more complex metaphorical structures (ibid.). They are illustrated by a representative list specifying in each case a descriptive name of the metaphor, subjective experience, sensorimotor domain, example, and primary experience. Here, we will pick out two examples that are taken up in the discussion below; the typographical conventions follow Lakoff / Núñez 2000 [166]:

> 1. Similarity Is Closeness
> Subjective Judgment: Similarity
> Sensorimotor Domain: Proximity in space
> Example: "These colors aren't quite the same, but they're *close*."
> Primary Experience: Observing similar objects clustered together (flowers, trees, rocks, buildings, dishes)

> 2. Organization Is Physical Structure
> Subjective Judgment: Abstract unifying relationships
> Sensorimotor Domain: Experience of physical objects
> Example: “How do the *pieces* of this theory *fit together*?”
> Primary Experience: Interacting with complex objects and attending to their structure (correlation between observing part-whole structure and forming cognitive representations of logical relationships)
> (Lakoff / Johnson 1999 [165], page 51)

The *neural theory of metaphor* which is developed within the framework of connectionist modeling or more specifically the *neural theory of language* (see Lakoff 2008 [163]; e.g. Feldman 2006 [83], Feldman / Narayanan 2003 [84]) aims to provide “the anatomical basis of source-to-target activations that constitute metaphorical entailments” (Lakoff / Johnson 1999 [165], page 47). Referring to investigations of motor schemas (Narayanan 1997 [196]) it is assumed that “the same neural mechanism that can control bodily movements can perform logical inferences about the structure of action in general” ([165], page 42). Neural connections across networks underlying different domains are taken to arise “during the period of conflation” (ibid.) as a result of simultaneous activation, in turn providing the basis for “metaphorical entailment”.

Conceptual blending is presented as the process by which complex metaphors are formed from primary metaphors (ibid., page 49). A conceptual blend, according to Lakoff / Núñez (2000 [166], page 48) “is the conceptual combination of two distinct cognitive structures with fixed correspondences between them”[6]; for cases in which the correspondences are established by metaphors, the term metaphorical blend is introduced (ibid.).

As an important aspect of metaphorical mappings, Lakoff and Johnson (1999 [165], pages 57-58) point out their asymmetric nature[7]: there is a source domain – in the examples given the sensorimotor domain – and a target domain – subjective experience. By the mapping, the inferential structure of the source domain is preserved within the target domain, i.e. conceptualization in the target domain according to this scheme is influenced or rather shaped by the inferential structure of the source domain.

Within this framework, metaphor is considered by Lakoff and Johnson (ibid., page 54) as embodied in three ways: correlations “arise out of our embodied functioning in the world,” “the source domain [...] comes from the body’s sensorimotor system,” and “the correlation is instantiated in the body via neural connections.”

Although there do not appear to be any explicit references to the sensory domain

[6] for a detailed discussion of conceptual blending see Fauconnier / Turner 2002 [82]

[7] also referred to as the *principle of unidirectionality*, e.g. Kövecses 2010 [159], page 7 or Evans / Green 2006 [81], pages 296-297

of hearing in the examples provided by Lakoff and Johnson nor to conceptualizations of phenomena related to music, e.g. codified in music theory, psychology of music, or research on cultural / social aspects of music, an extension of analyses in these directions may be worthwhile:

1. An intimate interdependence of the notions of similarity and proximity, reminiscent of the primary metaphor SIMILARITY IS CLOSENESS seems to be pervading a substantial part of literature within music perception / music cognition. As an early example, Stumpf (1883 [265], §§ 6-7) explicitly discusses the dependence of of judgments of distance on the (dis-)similarity of underlying sensation. According to his view, the four basic relationships of multiplicity ("Mehrheit"), amplification ("Steigerung"), similarity ("Ähnlichkeit") and fusion ("Verschmelzung") are "given with and within the momentary sensations and completely determined by them"[8] (ibid., page 97). [A judgment of, L.S.] distance is defined by Stumpf as "the inverse of the degree of similarity of two sensations or, shortly, the degree of their dissimilarity"[9] (ibid., page 122). Thus, Stumpf considers the psychological relationships between sensations as primary for conscious judgment, not (the experience of) physical relationships between objects in the world (leaving open at that point, however, the relationship between sensations and external objects / processes). This view is close to one line of positions taken in subsequent discussions e.g. of corporeal / material properties of sound (see below). Although establishing a close relationship between judgements of similarity and distance, however, in Stumpf's account the roles of source and target domains are reversed as compared to the primary metaphor cited above.

In more recent research, interpreting similarity of objects in terms of distances in some space (e.g. mathematical space of features), i.e. conceptualizing similarity (target) in terms of distance (source), forms the conceptual basis for the procedure of multidimensional scaling, which aims at producing a spatial configuration in which the distances between points representing objects reflect the (dis-)similarities between these objects according to some psychological measure (e.g. direct rating). Approaches utilizing multidimensional scaling have been applied to the investigation of tonal relationships (e.g. Shepard 1982 [250], Krumhansl 1990 [160], Krumhansl / Toiviainen 2003 [161]) or timbre (e.g. Grey 1977 [104], Wessel 1979 [316], Donnadieu 2007 [69]).

Neural network modeling approaches featuring the Kohonen self-organizing map implicitly rely on the interpretation of similarity as closeness: According to the description of the Kohonen algorithm given by Rojas (1992 [236], Algorithm 15.1), the euclidean distance between an input vector and the vectors of weights of the units of the network yields the criterion which of the units' weights to update.

[8] "[...] mit den augenblicklichen Empfindungen, in ihnen, und durch sie völlig determinirt [!] uns gegeben [...]"

[9] "[Der Begriff der Distanz] bedeutet in seiner allgemeinsten Fassung den reciproken Wert des Ähnlichkeitsgrades zweier Empfindungen oder kürzer den Grad ihrer Unähnlichkeit"

Regarding the input vectors as vectors of features of the objects to be classified by the network will again invite the interpretation of distances among input vectors as well as between input vectors and weight vectors as measures of dissimilarity. In this sense, the mapping of similarity to distance and the conceptualization of similarity in terms of closeness can be said to be "built into" the Kohonen map. For examples using the Kohonen map to investigate the cognitive structure of tonality see Leman 1995 [171] or again Krumhansl / Toiviainen 2003 [161].

Within research on perceptual organization of sound (e.g. Bregman 1990 [43]), the principles of proximity and similarity are invoked as rules underlying certain grouping phenomena. The rules are assumed to operate on some kind of time-frequency representation of auditory input, and both principles are sometimes used to refer to the same signal property, e.g. the more or less pronounced agreement in frequency. I.e., it sometimes appears not to be easy to differentiate between the notions of proximity and similarity in these contexts. Accepting the primary metaphor Similarity Is Closeness, however, this should not come as a surprise, because the conceptual structure associated with Similarity will not differ much from that associated with Closeness / Proximity.

2. The second primary metaphor quoted above, Organization Is Physical Structure, may be underlying concepts discussed in music theory / composition and aesthetic reasoning about music. Here, we will only briefly mention two examples:

- In his *Harmonielehre*, Arnold Schönberg (1922 [246]) explicitly and repeatedly states that "the tone is the material of music"[10] from which any musical piece must be constructed.

- Adorno (1948/1978 [2], e.g. pages 38–42) discusses the notion of musical material[11] emphasizing the need to take into account historical processes in addition to constructive details. Seen from the perspective of primary metaphor, he appears to be refuting a literal interpretation of the metaphor of material using fixed mappings to physical properties of sound, a tendency he ascribes to contemporary psychology of music / tone psychology (ibid., page 39).

3. The two examples discussed so far illustrate that primary metaphors may be underlying conceptualizations of musical phenomena and might even be responsible for confusion and debate if the metaphorical nature and the resulting structure of arguments are not considered with sufficient care. However, they constitute rather abstract and general ways of reasoning, not specific to the domain of hearing or to music.

[10] "Das Material der Musik ist der Ton [. . .]" (page 15); "Noch einmal: der Ton ist das Material der Musik." (page 17)

[11] e.g. "Material der Musik" (page 38)

The relation of not sound-related terminology to auditory phenomena has been extensively discussed within 20th century psychology of music with regard to what has been called corporeal or *material properties of sound*[12]. For example Rich (1916 [234]) discusses attributes such as small/sharp/pointed/high or large/massive/voluminous/low and their correlations; Hornbostel (1926 [122], pages 707–709) takes up among others extension, weight, and density[13] as static properties and movement, height, and distance[14] as more kinematic attributes[15]. Albersheim (1939 [3], Chapter 6) gives an overview of material properties used to describe sound and discusses their interrelationships; in addition, he offers a discussion of spatial properties of tones that is taken up as main focus and extended in Albersheim 1974 [4]. The "tone body"[16], its properties, and their consequences for compositional practice are discussed by Dräger 1952 [72].

Regarding the origins of the terminology under discussion, two extreme positions can be distinguished:
In keeping with the position of Stumpf referred to above, Hornbostel states with respect to the attribute of extension of tones

> This impression, too, is immediately acoustically given, not mediated by experience within other sensory domains.[17] (Hornbostel 1926 [122], page 708)

A contrasting view is expressed by Révész, in this case concerning the low-to-high characterization of musical pitch: according to him, this way of description is based on the experience of resonance to low tones in lower parts of the body and to high tones in upper parts[18] (Révész 1946 [233], pages 76–77).

4. The pertinence of metaphorical thought to the understanding of music is further illustrated by recent work in music theory and analysis. In particular, Larson & Johnson (2002 [169]), Johnson & Larson (2003 [139]), Spitzer (2004 [260]), and Zbikowski (2002 [325], 2008 [326], 2009 [327]) incorporate the notion of conceptual metaphor as cross-domain mapping and further ideas from cognitive linguistics / cognitive semantics into their analytical frameworks, which are exemplified by concrete musical compositions from different styles and epochs. As examples for

[12]e.g. "Materielle Eigenschaften", e.g. Albersheim 1939 [3], Chapter 6

[13]"Ausdehnung.Gewicht.Dichte.", Hornbostel 1926 [122], page 708

[14]"Bewegung.Höhe.Distanz.", Hornbostel 1926 [122], page 707

[15]"Ruhende Erscheinung" vs. "Bewegungseindruck und seine Richtung", Hornbostel 1926 [122], page 707

[16]"Tonkörper", Dräger 1952 [72]

[17]"Auch dieser Eindruck ist unmittelbar akustisch gegeben, nicht durch Erfahrung anderer Sinne vermittelt."

[18]"Meiner Ansicht nach verdanken die Ausdrücke ≪hoch≫ und ≪tief≫ innerhalb der akustischen Sphäre ihr Entstehen den *Lokalisationseindrücken der Schallvibrationen im Körper.* [. . .]"

metaphorical cross-domain mappings, Zbikowski (2002 [325], pages 65–72) discusses several conceptualizations of musical pitch, e.g. in terms of age, size of physical objects, or the vertical dimension from high to low. The role of the latter mapping, in particular, is further investigated in the context of common descriptions of pitch relations (e.g. musical "gestures"; ibid., pages 66–67) and – in combination with the idea of conceptual blending – as a means for musical "text-painting" (2002 [325], Chapter 2; 2009 [327]). The motivation offered by Zbikowski for the establishment of the conceptual metaphor PITCH RELATIONSHIPS ARE RELATIONSHIPS IN VERTICAL SPACE resembles the argument by Révész mentioned above: lower frequency tones are experienced to resonate in the chest while for high frequency tones "the sound source seems located nearer our head"[19] (2002 [325], page 69). However, although arguments for the plausibility of metaphorical mappings are advanced and on the whole a forceful case for an embodied understanding of music is made, there is no mention of specific ways in the sense of the framework presented in the beginning of this section that complex metaphorical systems reflecting the conceptualization of phenomena related to sound and music are established.

These remarks may suffice as a motivation to further pursue the investigation of conceptual metaphor and related ideas with regard to (the modeling of) musical behavior and experience. As the topic of metaphor is increasingly taken up in the context of music-related movement and musical gesture (e.g. Jensenius et al. 2010 [137]), this work will eventually be relevant for and benefit from the research addressed in the following section.

To conclude the section, two topics implicit in the discussion above will be addressed explicitly: Firstly, the notion of conceptual metaphor may appear appealing in light of the examples given above and seems to be accepted to some extent in the context of music analysis. Nevertheless, this approach may run counter to the search for musical features, i.e. structural properties within the domain of music, underlying certain types of conceptualization and thus constraining (metaphorical) mappings.[20] Secondly, within cognitive linguistics there appears to be a strong focus on the *learning* of metaphorical mappings (cf. Lakoff 2008 [163], who stresses Hebbian learning), which may be constrained by corporeal properties of the learning individual (agent) and regularities of the (physical and social) environment. This stance is reflected by reference to the neural network modeling attempts mentioned above and e.g. by Snyder's treatment of musical metaphor as a phenomenon of long-term memory (Snyder 2001 [254], Chapter 9). On the other hand, certain cross-domain mappings have been considered not to be mediated by individual experience. As a recent example for such a posi-

[19]but note here the reference to *corporeal sensation of vibration* on the one hand and *sound source localization* on the other

[20]For a recent overview regarding the extraction of music- and movement-related features see Camurri / Volpe 2011 [52]; cf. Lakoff's (1990 [162]) discussion of the conceptualization of time.

tion, Jackendoff / Lerdahl (2006 [131], page 65) postulate an instinctive "[...] ability of dancers to convert musical into gestural shape and [...] of performers following a conductor to do the reverse (though it can be refined by training)" along with an "[e]qually instinctive [...] ability of audiences to interpret these relationships". More generally, Seitz (e.g. 2005 [249]) emphasizes the role of inborn mechanisms underlying *basic metaphors* taken to precede conceptual metaphors in human development and discusses these with respect to evolutionary, developmental, and neuropsychological evidence. Thus, future tasks in the investigation of cross-domain metaphorical mappings involving the domain of music will have to include the disentanglement of aspects due to learning / individual acquisition from those imparted to humans by their biological nature, placing the enterprise in an interdisciplinary field comprising disciplines such as computational cognitive neuroscience, developmental psychology, comparative musicology, ethology, and evolutionary biology. It is in this context that robotic technology may come to play a role by giving the opportunity to provide partly controllable environments for sound- or music-related interaction. Moreover, a more systematic investigation of music-related terminology, a discussion of notions such as *concepts*, *domains*, *conceptual structure* or *mental spaces* with regard to music and a stronger integration with research on musical gestures will be necessary.

3.1.3 Expressive Movement

The human body as a medium of communication and interaction in musical contexts has gained ever increasing interest: Body movements and processes are investigated as a means to convey expressive or emotional aspects; more fundamentally, within the Musical Gestures Project [194] music-related bodily movements are considered to be intimately connected to the formation of musical concepts and the organization of musical behavior.

Fundamental to the use of corporeal cues for the transmission of emotional and expressive contents is the observation of processes commonly occurring in connection with subjective emotional experiences. At least the following kinds of cues have been investigated:

- physiological signals, such as heart beat rate, electromyogram (EMG), skin conductance
- facial configuration (e.g. position and movement of lips and eyebrows)
- small scale movements / gestures (e.g. finger / hand / arm)
- whole body cues (posture, large scale gestures)
- agent-environment relationships (stationary / mobile; occupations of space; movement patterns)

Automatic recognition of processes related to emotional experience and generation of displays exhibiting emotional expression has been investigated with respect to the facilitation of interaction between humans and different kinds of artifacts. Strong fields of interest are human-computer interaction (HCI, e.g. K.L. Norman 2008 [206]) and human-robot interaction, but even for the "design of everyday things" (e.g. D.A. Norman 2004 [205]) emotion-related processes are taken into consideration.

Picard (1995 [222], 1997 [223]) presents an extensive overview over the field of *affective computing*. As an example, Picard / Vyzas / Healey 2001 [224] examine the benefit of several cues based on physiological reactions for the recognition of a person's emotional state. As physiological signals, they measured electromyogram, blood volume pressure, respiration, skin conductance, and the rate of heart beat derived from blood volume pressure. From these signals, ten physiology-dependent features were defined based on smoothing and averaging, normalization on a daily basis, and signal differences; four of the features concerned the spectral analysis of the respiratory signal. In addition, six statistical features such as mean and standard deviation of the raw signals were taken into account. To these features, special techniques from pattern classification (e.g. Fisher projection, sequential floating forward selection, see Duda / Hart / Stork 2001 [74]) were applied to determine which of them were contributing to the discrimination of emotional states. Of all 16, five robust features (related to spectral characteristics of respiration, change of skin conductance, and change of heat beat rate) were always found to be effective, whereas mean heart beat rate, mean skin conductance, and mean respiration turned out to be useless. Related considerations will be of interest when similar signals are utilized within interactive settings to generate or investigate expressive or emotional musical performances (e.g. Camurri et al. 2007 [50], Camurri / Volpe 2011 [52]).

The relation of facial activity to emotional experience and other emotion-related processes has been studied extensively during the last decades, e.g. by Paul Ekman and many co-workers; a framework for the quantification of facial movement was developed by Ekman and Friesen in the 1970s (frequently quoted is Ekman / Friesen 1982 [77]). Results of these and related studies have been taken up in the design of artifacts intended to engage observers in (social) interaction, e.g. by displaying emotional facial expressions.

A well-known robotic example, Kismet, designed by Breazeal (e.g. 2002 [42]), consists of a head-like application equipped with eyes, eyelids, eyebrows, lips, and ears that is mounted in such a way that the head can turn, move forward / backward, and can be lowered / raised. The robot is endowed with a control system that allows the extraction of emotional cues e.g. from prosodic characteristics of vocal input and calculates an internal emotional state which forms the basis for the generation of appropriate behavior including expressive facial configurations, head posture and acoustic output.

Combining data on facial expression, musicians' expressive movements during performance (Dahl / Friberg 2007 [64]) and emotional cues extracted from acoustic input based on the analysis of musical performances (see e.g. Juslin 2001 [143] for a review), Mancini / Bresin / Pelachaud 2006 [181] specify the architecture for a software agent displaying an animated face. The goal of this system is to exhibit in real-time facial expressions matching the emotional coloring of the input, e.g. in order to provide feedback to performers trying to communicate specific emotions.

A platform aimed at "multimodal analysis and processing of nonverbal expressive gesture in human movement and music signals" (Camurri et al. 2005 [53], page 48), EyesWeb[21], was developed at the InfoMus Lab of the Department of Communication, Computer and System Science (DIST), University of Genova. EyesWeb provides the possibility to integrate different kinds of data such as audio, video, and various types of (on-body) sensor signals. Modules are included to enable communication using established music related devices e.g. via MIDI or Open Sound Protocol (OSC), to analyze incoming data, and to process output data (sound, video) in real-time based on the analyses. Video data is processed on different levels: on the low level, kinematic parameters[22] of a performer's movements are extracted (e.g. the "barycenter of a performer's silhouette" (ibid.)); at mid-level, "expressive cues such as body contraction / expansion [. . .]" are computed. (For a more detailed description of various cues, see e.g. Camurri et al. 2004 [51].)

According to Camurri et al. (2005 [53], page 49), "EyesWeb is the basic platform of the European Union Information Society Technologies (IST) Multisensory Expressive Gesture Applications (MEGA) project" and was also adopted within further EU projects.

As an ambitious attempt at integrating acoustical, optical, and physiological data, the Premio Paganini Experiment was launched in 2006 (Camurri et al. 2007b [49]): The experiment aims to investigate communication of emotion both among musicians and between performers and audience in a concert situation. To this

[21] `www.eyesweb.org`

[22] The term *kinematic* is used in the sense introduced in standard physics textbooks relating exclusively to positions, velocities, and accelerations e.g. of particles, as opposed to the term *dynamic* that also refers to forces / physical laws describing the relations between the kinematic variables by way of equations of motion. I was alerted to this distinction in the discussion following a performance of par_cho|r:fugue by Christoph Lischka and Frank Gratkowski (see Figure 6.2). By some of the observers, delays in the change of direction etc. of the spherical robot were attributed to the mass of the sphere, whereas according to the explanation given afterwards, they were caused by the processing speed of the robot's control. Data from optical observation only inform about (changes in) position and speed; the ascription of dynamic properties, such as inertia, force, tension (e.g. "Laban's effort space", Camurri et al. 2004 [51], Figure 1) constitutes already an interpretation of kinematic data. This distinction should be kept in mind when dealing with standard terminology describing expressive motion.

end, audio data, video data recorded from different angles simultaneously, and EMG and electrocardiogram (ECG) data from pairs of professional violinists playing a Bach canon was collected in varying constellations (e.g. with and without allowing the musicians to see each other). A large database was compiled for the study of gesture-emotion relations in the musical performances addressing
– intra-personal synchronization regarding the performers
– inter-personal synchronization among performers
– inter-personal synchronization between performers and audience.

(For a discussion of problems concerning the evaluation of synchronization see e.g. Pikovsky / Rosenblum / Kurths 2001 [225], Section 1.2 and Chapter 6.)

More information on music-related research on (expressive) gesture can be obtained from the recent book by Leman (2008 [172]), in particular Section 2.3. Indeed, judging from the subdivisions and headings in this section, research on embodied music cognition for Leman appears to coincide very nearly with gesture research, subsuming “Physics-Based Sound Modeling” (with a focus on players' movements in the production of sound), “Motor Theory of Perception”, and “Cognitive Neuroscience Research” as sub-headings within Section 2.3.2 “Gesture Modeling”; see also ibid., page 167: “Mediating Gestures (the embodied cognition approach).”

A recent tentative approach to analyze gestures within the framework of the theory of dynamical systems (Wenger / Copeland / Schuster 2007 [315]) will be addressed in Section 4.4.

3.1.4 Interactive Technology: New Interfaces

A related but quite different role is assigned to bodily movements in the context of new ways of sound production: New interfaces and instruments are developed that take different kinds of body-related signals as variables in the generation process providing new and enhanced possibilities for interaction between humans and technical artifacts. In principle, all of the technical applications and physical signals mentioned in the previous section can be employed in the design of interfaces or *Multimodal Interactive Expressive Environments (MIEEs)*, introduced as a “framework for mixed-reality applications in the performing arts [...]” by Camurri et al. 2005 [53] (see also Leman 2008 [172], pages 176–182). The broad range of possible devices is illustrated e.g. by the series of conferences on New Interfaces for Musical Expression (NIME, `www.nime.org`) – understanding musical behavior, however, doesn't appear to be a primary goal of this work.

3.1.5 Remarks

The examples presented in the previous sections were intended to give an impression of different ways the human body can be of importance for experience and behavior in music-related contexts. Starting from quite disparate fields of inquiry, perspectives seem to become increasingly integrated, as evidenced particularly by the research on expressive movement discussed in Section 3.1.3. Nevertheless, the systems designed – with the possible exception of Kismet – mainly react to body-based signals of components (e.g. humans) *outside* the system in the sense that the systems are intended to operate with these components in place. Thus, our preference will be to regard these systems as disembodied.

To gain some more insight concerning criteria for *embodied* systems, we will next turn to a discussion of different *notions of embodiment*.

3.2 Notions of Embodiment

The term *embodiment* is intuitively appealing, but difficult to capture precisely. Ziemke (2001 [328]) presents a summary of previous attempts to find a definition, proposed with regard to embodied cognition and autonomous agent research and offers a quasi-hierarchical set of notions of embodiment (see Figure 3.2) to capture the discussion so far.

The notions are, in the order of increasing restrictiveness:

- structural coupling
- historical embodiment
- physical embodiment
- organismoid embodiment
- organismic embodiment.

The presentation suffers slightly from presupposing a view that restricts cognition to living organisms, which leads to blurring the concepts cognition, life, and embodiment in the explanations given.

The most permissive notion, structural coupling, is adopted from Maturana & Varela [183]. It is provided to allow for agents without a body in any obvious sense (software agents, alife creatures) to be included in the discussion. Taking this notion to be more inclusive than physical embodiment, however, seems to be misleading:

Agents as programs, i.e. as structures in a purely logical sense, do not interact with their environment and therefore do not exhibit any form of coupling. For structural coupling to arise, the agent program must be implemented and running, i.e. it must be realized by some physical processes. These processes, however, need not be tied to any specific and readily identifiable material substrate. This seems to me to be the main aspect of the characterization that there is no body "in the usual sense". The distinction drawn here between logical structure and realizing physical processes may reflect Maturana & Varela's differentiation of organization and structure:

"By *organization* we mean those relations that must be given among the constituent parts of something so that it will be recognized as a member of a certain class.

By the *structure* of something we mean the constituent parts and the relations that in a concrete manner constitute a certain unit and realize its organization." (Maturana & Varela [183], translated from the German edition, page 54)

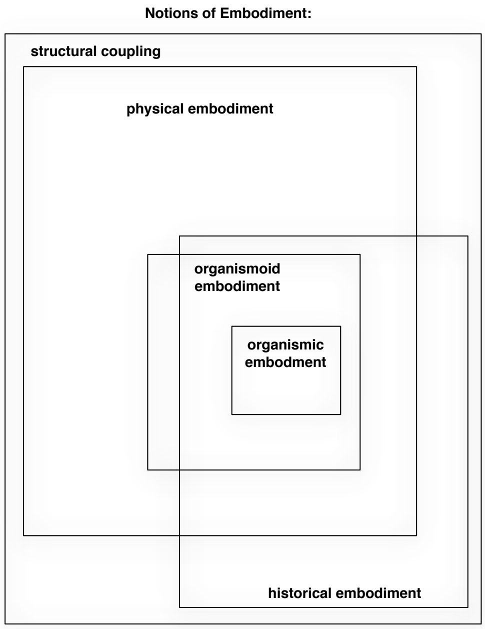


Figure 3.2: Notions of embodiment, redrawn after Ziemke [328]

By the notion of *historical embodiment* the observation is captured that not only the present structure is of importance for the way agent and environment interact, but that also the course of previous mutual influences may have contributed to the present form of coupling. This means that among others phenomena of learning, development, adaptation (on the side of the agent) as well as structuring of the environment can be taken into account.

As indicated above, the differentiation of physical embodiment from the notion of structural coupling seems unclear. As a possible way for a system to qualify as physically embodied, Ziemke quotes Brooks' requirement of the presence of sensors and actuators integrated into a material structure; a first attempt to delimit the body of a living organism might be to take the "biological skin-bag" (Clark [59], page 5) as a boundary.

The remaining two notions, organismoid and organismic embodiment, again direct attention to the specific ways agent-environment coupling is implemented, i.e. to the presence and location / morphology of sensors and actuators as well as the integration of their respective activity into coherent and well-adapted behaviors. Whereas organismoid (organism-like) embodiment is intended to be applied to both artificial and living systems, organismic embodiment is reserved for the latter – which leaves the problem of distinguishing the living from the non-living / artificial. The discussion presented by Ziemke (2001 [328], Section 4.5) leaves room for further debate (for some preliminary remarks see Appendix A).

In this discussion, there is a strong focus on the individual agent and its coupling to a more or less unspecified environment; the role of the environment is discussed with regard to shaping the specific coupling. However, some more consideration might be given to the role of the environment as *part of* processes mediating agent-environment interaction, as well as to the presence of other (similar) agents within the environment.

As an example for the first aspect, the role of artifacts has long been considered as an indicator of human capabilities and the organization of human behavior in general archaeological research as well as the archaeology of music (e.g. Leroi-Gourhan 1964/1988 [176]; Morley 2003 [193]; for a popular overview see Mithen 2005 [190]).[23] More recently, the role of artifacts (and routines including other agents) has been addressed in cognitive science with respect to everyday situations (Norman 1993 [204]), working environments (Hutchins 1995a [123], 1995b [124]), or the enhancements of human capabilities and experience afforded by the development of new technology (Clark 2003 [59]).

Regarding the second aspect, the role of the presence and recognition of other agents has been emphasized e.g. in the context of cultural learning (Tomasello / Kruger / Ratner 1993 [286], Tomasello 2000 [285]).

According to the interpretation offered here, embodiment in any of the senses discussed above is tightly connected to some sort of physical processes going on, problems to be solved including:

- What kinds of processes are involved?
- What are the ways of (re-)configuring these processes?
- What are the mutual interactions between processes?
- How can the coherence of processes be defined in a way that reflects the identity of an agent?

Some general aspects will turn up in the following discussions of agents and agent-environment interaction. Spelling out the details, however, can be considered the major challenge to an embodied cognitive science of music.

[23] As an aside, a tentative observation based so far on rather restricted reading, will be added: The development of the human vocal apparatus has been investigated extensively with regard to articulatory capabilities required for the production of speech sounds and musical vocalizations (Fitch 1994 [87]; Morley 2003 [193], Chapter 4.1). Another aspect of the human voice not found in other animals (Tembrock 1996 [275], pages 33–34) does not appear to have attracted as much attention: the sexual dimorphism regarding vocal fundamental frequency, generating the need for humans to generalize over frequency in auditory perception. These considerations will have to be followed up and discussed elsewhere.

3.3 Agents as a Modeling Framework

The entities discussed above regarding embodiment were referred to as *agents*. The notion of agent has become increasingly important within artificial intelligence since the early 1990s, supported by the spread of internet applications based on software agent technology (see Russell / Norvig 2003 [238], Chapter 1.3.9). The definitions of the term "agent" remain rather broad: the most general characterization offered by Russell / Norvig (2003 [238], page 4) is

> An **agent** is just something that acts [...]. But computer agents are expected to have other attributes that distinguish them from mere "programs," such as operating under autonomous control, perceiving their environment, persisting over a prolonged time period, adapting to change, and being capable of taking on another's goals.

No restrictions on "something" are introduced, nor is there any delimitation of "acting". The second sentence, however, clarifies the main intended application area to be treated in the book, namely software agents.

A little more specific is the definition given by Russell / Norvig 2003 [238], Chapter 2, dedicated to the introduction of *intelligent agents*:

> An **agent** is anything that can be viewed as perceiving its **environment** through **sensors** and acting upon that environment through **actuators**.

Taken together, these characterizations contain with one exception all elements considered essential for an agent by Franklin / Graesser 1997 [89][24] in analysis of different agent definitions:

An agent is an entity that can be distinguished from an environment and is endowed with the ability to act within / upon the environment without explicit directing by any other entity, can take in information from the environment (sense the environment), and persists over some time. Franklin and Graesser's requirement that an agent acts "so as to effect what is senses in the future" is explicitly called into question by Russell / Norvig 2003 [238], page 32.

For the sake of explicitness, we will describe and discuss three different kinds of agents introduced by Genesereth / Nilsson 1986 [93], Chapter 13, that appear to capture in a concise way ideas extended in later work in artificial intelligence, e.g. in the textbooks by Russell / Norvig 2003 [238] or Nilsson 1998 [202]. The ideas presented will show up again in later sections.

[24]here referring to the online version retrieved from `http://www.msci.memphis.edu/~franklin/AgentProg.html`, date of retrieval 2005-06-11

All three types of agent are illustrated by the example of a cart moving in a 2-dimensional maze, shown in Figure 3.3. The maze consists of a 3×3 array of cells that are vertically and horizontally connected to their neighbors; the cart can move from cell to cell along these connections. In one of the cells a bar of gold is contained which the cart can take up when in the same cell. A task for the cart may be to start from the upper left cell, find and take up the gold bar to take it to the lower right cell (exit) and put it down again. To achieve this task, the cart must be able to sense whether or not the bar of gold is in the same cell and to take it up resp. put it down.

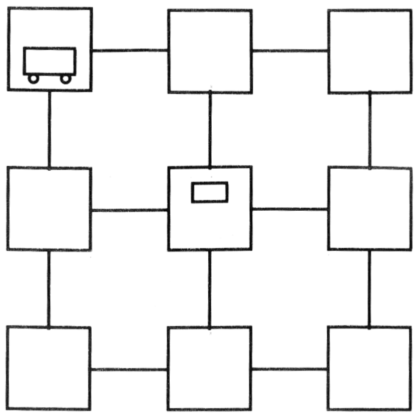


Figure 3.3: Example Cart, from Genesereth / Nilsson 1986 [93] page 309.

The first agent introduced that might achieve this task is called tropistic agent.

3.3.1 Tropistic Agents: Acting in an Environment

Tropistic agents, according to the characterization given, are agents, "whose activity at any moment is determined entirely by their environments at that moment" (Genesereth / Nilsson 1986 [93], page 307).

The environment is assumed to be in any one of a set **S** of states. To influence the agent's activity, the states must somehow be taken into account, i.e. they must be perceived. Possibly, "due to sensory limitation" (page 308), not all states can be distinguished, so that all states belonging to a partition `t` of **S** give rise to the same perception. In the example of the cart in a maze, all states with the cart in one specific cell and the gold bar in another cell will lead to the same percept. Already at this point, it is obvious that the characterization quoted above actually is incomplete: The activity is influenced entirely by the environment, *given a particular set of sensory capabilities.* In other words, the *structural coupling between agent and environment* is implicitly part of this definition of an agent.

Sensory capabilities in this scheme are described by a sensory function `see` which maps states to partitions of indistinguishable states, i.e.

$$\texttt{see} : \mathbf{S} \to \mathbf{T},$$

where $\mathbf{T}$ is the set of all the partitions.

Possible activities of the agent (in the cart example: move right / left / up / down, take up / put down gold bar) are included in the description as a set $\mathbf{A}$. Since states can not be distinguished, the action performed is conceptualized not as depending on the state of the environment but on the corresponding partition into which the state is mapped by the function `see`. Thus, a function determining the action is defined as a mapping[25]

$$\texttt{action} : \mathbf{T} \to \mathbf{A}.$$

The effect on the environment, i.e. the state of the environment *after* the action of the agent has been performed, is captured by another function mapping a combination of action and (initial) state to another (final) state. This function, called effectory function, is specified as

$$\texttt{do} : \mathbf{A} \times \mathbf{S} \to \mathbf{S}.$$

By including the function `do` in the definition of the agent, which describes the change of the environment, aspects of the environment are incorporated in the definition, i.e. structural coupling between agent and environment according to this definition is mutual.

Taking these specifications together, the tropistic agent is abstractly defined as a 6-tuple (page 308)

$$\langle \mathbf{S}, \mathbf{T}, \mathbf{A}, \texttt{see}, \texttt{do}, \texttt{action} \rangle.$$

Note that in this definition nothing specific is said about the nature of the states and actions, nor are there any restrictions on the realization of the functions introduced. Thus, nothing prevents application of this scheme to robotic artifacts or bees collecting honey. Not also that the tropistic agent fully qualifies as embodied qua structural coupling.

An aspect considered problematic may arise in the software realization of the cart example: Starting from an initial state $\texttt{s}_0$, to this would be applied the function `see`, resulting in a partition $\texttt{t}_0$ characterizing perception:

$$\texttt{t}_0 = \texttt{see}(\texttt{s}_0).$$

[25]The introduction of the set $\mathbf{T}$ does not appear imperative: The action function could be defined on the set of states, e.g. as a composition of the functions `see` and `action` as given here:

$$\texttt{directaction} : \mathbf{S} \to \mathbf{A}, \quad \texttt{directaction} = \texttt{see} \circ \texttt{action}.$$

Next, $\mathtt{t}_0$ would be used to determine an appropriate action $\mathtt{a}_0$ according to

$$\mathtt{a}_0 = \mathtt{action}(\mathtt{t}_0).$$

Performing the action finally would lead to a new state $\mathtt{s}_1$ of the environment:

$$\mathtt{s}_1 = \mathtt{do}(\mathtt{a}_0, \mathtt{s}_0),$$

which then could be passed to the function `see` again, leading to a cycle of function applications inducing an ordered set of states traversed and actions performed by the agent.

But this cyclical and serial execution of functions is due to the realization of the agent software, it is in no way prescribed by the general definition of the tropistic agent.

One aspect lacking in the definition of the tropistic agent is the potential to change behavior: each time identical states (ore states belonging to the same state partition induced by the sensory function) are encountered, the action function will select the same activity[26]. To amend this deficit, Genesereth and Nilsson introduce the hysteretic agent.

3.3.2 Hysteretic Agent: Retaining Information

The *hysteretic agent*, introduced in Chapter 13.2 of Genesereth / Nilsson 1986 [93] extends the tropistic agent by introducing a set of internal states **I** of the agent to provide "the ability to retain information internally" (page 311). In the cart example, internal state is illustrated as knowledge about the cell the cart is currently occupying (specified as a numerical label – 1 to 9 – or as a combination of two of the characters `A`, `B`, `C` indicating row and column of the cell, such as `AA` for the upper left cell).[27]

For internal state to take effect upon agent activity, action selection must be made dependent on internal state. Therefore, the function action is modified to include internal state:

$$\mathtt{action} : \mathbf{I} \times \mathbf{T} \rightarrow \mathbf{A};$$

the functions `see` and `do` remain unchanged.

[26] reminiscent of insect-like behavior; an illustrating discussion of the behavior of the sphecoid wasp is given by Dörner 2001 [70], pages 89 – 91

[27] Genesereth / Nilsson 1986 [93], page 312, introduce the assumption that the agent can go from one internal state to any other internal state "in a single step". Firstly, the need for this assumption is not motivated; secondly, it seems to be in conflict with the cart example, because the cart can not move from state `AA` to state `CC` without passing at least three intermediate states; thirdly, the introduction of steps at this stage limits the generality of the definition.

In addition, the internal state needs to change according to present state and external influence, which is described by an update function

$$\texttt{internal} : \mathbf{I} \times \mathbf{T} \to \mathbf{I}.$$

With these extensions included, the definition of the hysteretic agent will be an 8-tuple

$$\langle \mathbf{I}, \mathbf{S}, \mathbf{T}, \mathbf{A}, \texttt{see}, \texttt{do}, \texttt{internal}, \texttt{action} \rangle.$$

Again, nothing specific has been said about the included sets and functions (but see footnote 27). Thus, a very general framework to describe agents has been provided by Genesereth and Nilsson. It is the task of the researcher / agent designer to come up with appropriate sets of states and actions as well as sensory / effectory functions and functions describing change of state – and it seems to be on this task that classical artificial intelligence and embodied cognitive science / embodied artificial intelligence diverge. This impression is supported by the discussions presented in Section 3.3.4 and Chapter 5.1.

3.3.3 Knowledge-Level Agents: Taking Care of Time?

As a third category of agents, Genesereth / Nilsson 1986 [93] introduce the *knowledge-level* agents, thereby taking a step in the direction of classical artificial intelligence. These agents are explicitly designed to abstract from physical details of the realization. For an illustration, we will extensively quote from the description given:

> Intelligence appears to be a phenomenon that transcends implementation technology, such as biology or electronics. Consequently, we want a design in which physical detail is abstracted away.
>
> In this section, we examine a conceptualization of agents, called the *knowledge level*, in which all excess detail is eliminated. In this abstraction an agent's internal state consists entirely of a database of sentences in predicate calculus, and an agent's mental actions are viewed as inferences on its database. At this level, we do not specify how the beliefs are physically stored, nor do we describe the implementation of the agent's inferences.
>
> Genesereth / Nilsson 1986 [93], page 314

It is the characterization of intelligence in the beginning of this quote that proponents of embodied cognition / embodied cognitive science / embodied artificial intelligence are skeptical about.

Another point of criticism, e.g. with regard to the computational assumption, concerns the reference to predicate calculus as a description of the agent's internal state. We will have a closer look at this aspect, feeling somewhat uneasy with predicate calculus as a description for inherently temporally structured situations.

Formally, the knowledge-level agent is defined as a modified hysteretic agent, replacing the set **I** of internal states by a set **D** of databases consisting of "sentences in predicate calculus" to describe the agent's internal state. As a consequence, the action function must be modified as a function

$$\mathtt{action} : \mathbf{D} \times \mathbf{T} \to \mathbf{A},$$

(**T** and **A** are defined as before), and the update function for internal state (`internal`) is replaced by a database update function

$$\mathtt{update} : \mathbf{D} \times \mathbf{T} \to \mathbf{D}.$$

Thus, the knowledge-level agent is defined as an 8-tuple (ibid., page 314)

$$\langle \mathbf{D}, \mathbf{S}, \mathbf{T}, \mathbf{A}, \mathtt{see}, \mathtt{do}, \mathtt{database}, \mathtt{action} \rangle.$$

As long as only databases of the form

$$\Delta = \{\mathtt{Cart(AA)}\}, \quad \Delta \in \mathbf{D}$$

are considered, asserting that the example agents is situated in the upper left cell of the maze, there are no problems.

Uneasiness, however, arises when the formalism is extended to describe the behavior of the agent. As an example, consider the sentences (ibid., page 316)

$$\begin{aligned}
\mathtt{Cart(AA)} \wedge \mathtt{Gold(IC)} &\;\Rightarrow\; \mathtt{Must} = \mathtt{R} \\
\mathtt{Cart(AA)} \wedge \mathtt{Gold(SC)} &\;\Rightarrow\; \mathtt{Must} = \mathtt{I} \\
\mathtt{Cart(AA)} \wedge \mathtt{Gold(EW)} &\;\Rightarrow\; \mathtt{Must} = \mathtt{R}
\end{aligned}$$

These sentences express that for different combinations of internal (`Cart(AA)`) and external (`Gold(·)`) states, different actions (`I`, `R`) are to be performed (`Must`). "$\Rightarrow$" and "=" have been introduced as logical symbols by Genesereth / Nilsson 1986 [93], page 18 resp. 85, 89. Nevertheless, here they suggest a different interpretation:
"=" seems to be used as an assignment operator (as common in programming languages as C or Java), not as a logical predicate;
the use of "$\Rightarrow$" in this context gives the impression of a "triggering operator": whenever the antecedent holds true, the action specified in the consequent – here an assignment operation – is to be executed.

A logical interpretation of these sentences does not quite comply with this reading: Assuming `Cart(AA)` and `Gold(IC)` to be true, the first statement is only true if also `Must` = `R` is true, specifying the adequate action for the situation described by the antecedent.

But since `Gold(IC)` and `Gold(SC)` are mutually exclusive, the antecedent of the second statement is false rendering the statement as a whole true even if the inappropriate action specification `Must` = `I` is true.

Thus, unless imposing that all sentences constituting the behavioral description of the agent be simultaneously true, i.e. interpreting the set of sentences as a *conjunction* of these sentences, the inappropriate action `I` is logically compatible with the situation $\texttt{Cart(AA)} \wedge \texttt{Gold(IC)}$. From the point of view of logic, this is not a problem, but may seem awkward when dealing with situations inherently characterized by temporal ordering. This issue becomes more interesting when function execution is not as neatly ordered as in the example of the tropistic cart discussed above, page 38; see also Chapter 8.4.

These remarks are just intended as an indication of the desirability of a formalism that explicitly takes into account the aspects of time and ordering.

3.3.4 Complete Agents: The Fungus Eater as a Hysteretic Agent

The notion of the *complete agent* is discussed by Pfeifer / Scheier 1999 [220], in particular Chapter 4.1, as a basic and central concept of embodied cognitive science. The complete agent according to this presentation is characterized by a set of requirement, which will be discussed with regard to the hysteretic agent described above, Section 3.3.2. Complete agents are considered "[...] complete, because they incorporate everything required to perform actual behavior" (ibid., page 81).

As an alternative name for the complete agent, Pfeifer and Scheier introduce the term *Fungus Eater* in homage to Masanao Toda: In the early 1960, the psychologist Toda developed a fictitious scenario of a robotic agent feeding on fungi that is sent to an inhospitable planet (called Taros) to collect valuable ore; the aim of this scenario was to find a set of properties and capabilities of the agent to be successful on such a mission (see Toda 1982 [279], Chapters 6 and 7). In the words of Pfeifer / Scheier 1999 [220], page 83, Toda's Fungus Eater approach captures "the main intuitions underlying the embodied cognitive science framework."

The five requirements treated by Pfeifer / Scheier 1999 [220], in the order presented in Chapter 4.1, are

- self-sufficiency

- autonomy
- situatedness
- embodiment
- adaptivity

The criterion of *self-sufficiency* refers to the capability of the agent to "sustain itself over extended periods of time" (page 85). This is illustrated with an agent's need to take care of its energy supply – by taking in food for a biological agent or electrical charge for a robotic device – or to avoid damage, keep up operating temperature etc. Aspects involved with self-sufficiency pertain to the agent itself, not only the environment. Therefore, the definition of an agent complying with this criterion must include internal state.

One problem discussed by Pfeifer / Scheier in the context of self-sufficiency is the necessity that can arise to provide different processes taking care of different aspects of the agent's needs, such as obstacle avoidance and monitoring of battery charge of a robotic agent: With an increasing number of aspects to be taken into account, a growing amount of time will be required for any serial execution of related functions in the way indicated in Section 3.3.1[28]. Assuming parallel operation of the required processes as suggested by the principle of loosely coupled parallel processes (discussed extensively in Chapter 11 of Pfeifer / Scheier 1999 [220]), on the other hand, the coordination of these processes may necessitate special measures to determine agent activity, i.e. special care must be taken in the realization of the effectory function.

A basic way to combine different processes operating in parallel to determine agent behavior was suggested by Braitenberg 1984 [41], vehicles 1 – 3: Here, the speed of the motors of small vehicles is directly set by the readings of sensors present on the vehicle (e.g. light sensor); in the presence of different kinds of sensors, such as light, temperature, oxygen (Braitenberg 1984 [41], page 12) operating in parallel, the sensor values are simply superposed.

A rather more sophisticated, classic approach was presented by Brooks 1985 [44]: In the so-called subsumption architecture, aspects of the agent's behavior are realized by functions at different levels, low level functions realizing basic behaviors (e.g. obstacle avoidance, Brooks 1985 [44], Figure 2) and higher levels adding more complex aspects to the behavior (e.g. exploration, ibid.). Higher level functions are combined with lower levels such that they "can subsume the roles of lower levels by suppressing their outputs" (ibid., page 1).

[28] This is actually the mode of operation of the control programs for the Khepera III robot described in Chapter 7, even in cases instantiating autonomous activity.

In both examples, one goal is to devise an appropriate effectory function for the resp. agent; in the higher levels of the second example, the update function for internal state, too, is addressed.

The requirements of *autonomy* and *situatedness* are discussed as complementary by Pfeifer / Scheier 1999 [220], pages 89 – 91: Situatedness establishing the dependence of an agent's activity on its environment is seen as limiting the agent's autonomy interpreted as freedom of control. Interpreting control more restrictively as determining the course of the agent's activity by processes other than its own sensory, effectory, and update functions, the examples given as control could better be interpreted as structural coupling. Both requirements are fulfilled by tropistic and hysteretic agents as defined above.

The criterion of *embodiment* here seems to coincide with the notion of *physical embodiment* discussed above, i.e. there needs to be some material structure that can be contained in a finite volume with a closed surface (referred to as "biological skin-bag" above). It is at this point that embodied cognitive science diverges from classical artificial intelligence, given the explicit goal of abstracting from physical detail quoted above. Nevertheless, the framework definition of the hysteretic (or even tropistic) agent is not violated by this requirement. Physical properties of agents are explicitly acknowledged by Genesereth / Nilsson, page 316, as important components of e.g. of effectory and sensory functions by pointing out the difficulty to change hardware implementations for desired changes of behavior.

Adaptivity, as the last of the criteria for the complete agent, was presented above as the motivation to extend the tropistic agent to the hysteretic agent including internal state. Pfeifer / Scheier 1999 [220], pages 93 – 94, discuss four aspects of adaptivity:

- The first aspect, *evolutionary adaptation*, actually does not pertain to the specification of the individual agent; rather, it points at the way the agent's design is conceived.

- *Physiological* and *sensory adaptation*, illustrated by sweating as a reaction to external temperature or widening / contraction of the pupils according to light changes, can be considered as activities of the agent to be described by the effectory function.

- Learning, i.e. retention of information in order to modify behavior in otherwise similar situations, is related to internal state.

From these considerations we may conclude that it is mot a new general framework that is provided by the notion of the complete agent. Rather, even a complete agent seems to be a special case of the hysteretic agent as defined by Genesereth / Nilsson 1986 [93]. What is radically different from approaches based on classical artificial as indicated above, however, is the way to spell out the components of the agent, exploiting instead of abstracting from details of real-world physical[29] interactions[30]. On the one hand, this poses the challenge to incorporate a wide range of research into agent design; one consequence has been the growing interest in the theory of dynamical systems (discussed in Chapters 4 and 5). On the other hand, the chance is offered to put theories within these research fields to real-world tests by building artifacts incorporating the principles proposed, thus eventually stimulating further theoretical development.

3.4 Conclusion

As will be evident from the discussion presented so far, we cannot offer a coherent view of an embodied cognitive science of music. Instead, we have tried to give some motivation for adopting an embodied perspective focusing on a particular approach to modeling.

In the following chapters, we will turn to considerations of the proposed general framework, present some concrete examples and discuss ideas regarding possible future work.

[29] including chemical, biological, and social

[30] The remarks offered here seem to fit well with the opinion expressed by Anderson (2003 [7]), page 95, that research in embodied cognition is mainly directed against the commitment to the explicit (declarative) representation of knowledge by sentences in predicate logic (e.g. Nilsson 1991 [201]).

Chapter 4

Theory of Dynamic Systems: Some Simple Examples

Up to this point, the conceptual framework of embodiment has mostly been discussed in rather informal terms. Here, we will try to take some first steps towards a more formal way of talking about the notions of embodiment described above.

A very broad and general mathematical framework for the description of the behavior of agents (natural and artificial) within their environment is provided by the theory of dynamical systems. Aspects of this theoretical framework have been exploited at least since the late 1940s, prominently exemplified by Norbert Wiener's *Cybernetics: or Control and Communication in the Animal and the Machine* [317], first published in 1948. Although often mainly associated with the principle of negative feedback,[1] in hindsight cybernetics may possibly be characterized as the application of the theory of dynamical systems to a restricted class of problems, namely the regulation of processes with the aim to keep operation within certain pre-defined bounds.

Indeed, it was under the heading of cybernetics that W. Ross Ashby (1956, 1960 [17, 18]) discussed the properties of dynamical systems and developed ideas that are taken up in recent research on agent - environment interaction (e.g. Beer 1995a,b, 2000, 2003 [27, 28, 29, 30]; Pfeifer / Scheier 1999 [220]; Pfeifer / Bongard 2006 [218]; Steels 1996 [261]). Ashby explicitly rejected the reduction of cybernetics to the principle of feedback at the expense of more general notions such as the mutual influence of multiple coupled variables described in [18], Section 4/11.

Ideas related to cybernetics and the theory of dynamical systems have been applied to a wide spectrum of areas: To the general public, effects that are related

[1] As an example, Arbib [11], p. 87 talks about "the crucial cybernetic concept of (negative) feedback."

to special mathematical properties of dynamical systems have been effectively introduced in the context of "strange" (i.e. unexpected and/or unpredictable) phenomena discovered in physical systems that are now known under the name of (deterministic) chaos (see Ott 2002 [211]; for a popular overview Gleick 1998 [94]).

More relevant for the cognitive science of music are applications within the fields of neuroscience / brain theory (e.g. Arbib 1972,1989 [10, 11]), biology (Bertalanffy 1969 [33]; Ellner / Guckenheimer 2006 [78]), and various sub-fields of psychology including developmental psychology (e.g. Thelen 1995 [278]; Bertenthal 2007 [34]) and social interaction (Rodgers / Johnson 2007 [235]).

Within cognitive science, the "dynamical systems approach" has been put in opposition to the doctrine of "computational and representational understanding of mind (CRUM)" (Thagard 2005 [277]), most vigorously by van Gelder (e.g. van Gelder 1997 [290]; for more examples see also Port / van Gelder 1995 [228]; Thagard 2005 [277], especially Chapter 12). The claim of incompatibility of dynamical systems and computational approaches will be taken up below (Section 5.2).

A thorough introduction to the theory of dynamical systems is obviously beyond the scope of this presentation and the capabilities of its author. We will restrict ourselves to the discussion of two textbook examples to illustrate some fundamental properties of dynamical systems: The simple free *harmonic oscillator* with and without damping, exemplified by the spring-mass pendulum, will be used to introduce basic notions such as variables, parameters, phase space and to demonstrate the application of the canonical first order formulation to system descriptions involving higher order temporal derivatives of variables. In addition, the free and forced harmonic oscillator will be addressed in relation to current models of perception and production of rhythmic pulse (Section 4.2).

In contrast to the passive harmonic oscillator, the *van der Pol oscillator* is an active system that exhibits a type of behavior called *self-sustained oscillation*. The principles discussed are illustrative of features underlying the general theory of synchronization (for a comprehensive overview see Pikovsky et al. 2001 [225]; condensed version Rosenblum / Pikovsky 2003 [237]), which again as a phenomenon is taken up in the context of rhythm perception and production (see contributions in Desain / Windsor 2000 [68]), attentional dynamics (Large / Jones 1999 [168]; Jones 2004 [141]; Drake / Jones / Baruch 2000 [73]) as well as larger scale phenomena of emotional (Hatfield 1994 [114]) and more general social interaction (e.g. Kendon 1970, 1990 [153, 154]; Leman 2008 [172]).

As a final introductory example, recent research on gestural dynamics will be discussed to illustrate the potential for psychological research and associated problems of data analysis.

– The rather lengthy discussion of well known textbook examples in the following may appear tedious. Nevertheless, as illustrated by the analysis presented in Section 4.2, it seems worthwhile to spend some time on the basic assumptions and technicalities involved with the dynamical systems approach. As will be argued, the approach can provide metaphorical inspiration, which however may be misleading without careful interpretation of the assumptions tacitly acknowledged; for thinking through of the consequences some technical skills will be required. –

As may already be apparent from the discussion so far, the term “system” is used in differing – sometimes even ambiguous – way: in contexts closer to technical or empirical research, such as Arbib 1989 [11] or the contributions in Nehaniv / Dautenhahn 2007 [197] (but also in some philosophically inclined texts: Glymour 1997 [95]; Clark 2001 [58], Chapter 7) “system” is taken to refer to (collections of) real-world entities or phenomena under investigation; in more mathematically oriented discussions (but also: Ashby [17, 18], Beer [27, 28, 30]) “system” refers to the mathematical formalism such as differential / difference equations set up to describe (aspects of) the behavior of real-world entities / phenomena.

Here, we will try to keep to the first convention, regarding the mathematical formalism as a description of a “real system” (e.g. Arbib 1989 [11], p. 90). Arbib includes in the “description of any real system” (ibid.) five elements, here adapted for the case of a continuous time description:

1. The set of inputs. These are variables describing the environment and believed to affect system behavior of interest.

2. The set of outputs. The outputs are those of the variables describing the system that are chosen to be observed or believed to “act upon the environment to yield observable changes in the relationship between the system and the environment” (ibid.)

3. The set of states. The explanation given by Arbib is somewhat unclear. More precisely, the set of states may be taken as the set of possible assignments of values to the system’s variables; the current state of the system at time t is given by the ordered set of values of the system’s variables at time t.

4. A function describing the change of the system’s state. In the theory of dynamic systems (e.g. Anosov 1988 [9]; Ott 2002 [211]; Ashby 1960 [18]) this function is canonically assumed to be a (set of coupled) first order ordinary differential equations.

5. The output function. “The function that determines what output the system will yield with a given input when in a given state” (ibid.)[2]

[2]Note the shift in meaning of the term “system” implied by this last quote.

To this list we will add – in accordance with e.g. Ashby 1960 [18]; Arbib 1972 [10]; and Beer 1995a,b [27, 28] – the set of parameters characterizing the system. The significance of the system's parameters will be illustrated in the example of the spring-mass pendulum; the distinction between system variables and system parameters will be taken up in the discussion of agent-environment interaction below (Section 5.1).

In terms of mathematical formulae, the behavior of a "real system" within the framework of the theory of dynamical systems will thus be described by an equation as follows:

$$\frac{d\vec{X}}{dt} = \vec{F}(\vec{X}, \vec{u}) \tag{4.1}$$

where the components of the vector $\vec{X}$ are the variables of the system, the components of $\vec{u}$ represent the system's parameters, and the components of the vector $\vec{F}$ are arbitrary functions of the components of $\vec{X}$ and $\vec{u}$. Obviously, the number of dimensions of $\vec{F}$ must match the dimensionality of $\vec{X}$.

Before refining this scheme to explicitly address system-environment interaction, we will proceed to illustrate it by turning to our first textbook example.

4.1 The Spring-Mass Pendulum

The spring-mass pendulum is a simple mechanical device taken to consist of two main components: a spring that is at one end attached to a rigid, immobile support and at the other end to a mass that is free to move in one dimension. There is a stable equilibrium position, which an undisturbed pendulum will remain in, and a slightly disturbed pendulum eventually will return to. Displacement of the pendulum is typically measured in relation to the equilibrium position: designating displacement by $x(t)$ the value $x(t) = 0$ will correspond to the equilibrium position, positive values will correspond to deviation to one side, negative values to the other side of the equilibrium position.

Any displacement of the pendulum from equilibrium will lead to the spring exerting a force on the mass counteracting this displacement. This force F_s depends on the stiffness k of of the spring (also called spring constant) and is assumed to be directly proportional to the negative displacement of the pendulum, i.e.

$$F_s = -kx(t); \tag{4.2}$$

this idealized relationship is known as *Hooke's law*.

If the pendulum is released from a non-equilibrium position, it will start to move towards equilibrium; the ensuing movement pattern depends on the conditions surrounding the pendulum. In real-world settings the movement of the pendulum will be obstructed by interactions with the media surrounding the body of the pendulum such as gases / fluids the pendulum is moving in or surfaces it is sliding on. These interactions are usually summarized by the notion of a *frictional* force F_f, which is assumed to be directly proportional in amount to the velocity and acting in the opposite direction. Introducing the frictional proportionality constant r and the first temporal derivative $\frac{dx(t)}{dt} = \dot{x}(t)$ for the velocity of pendulum movement, the frictional force is given by

$$F_f(t) = -r\dot{x}(t).$$

Finally, taking into account Newton's third axiom relating the total force $F_{total}(t)$ acting on a body of mass m to the acceleration $\frac{d^2x(t)}{dt^2} = \ddot{x}(t)$ experienced by that body, the so called equation of motion of the pendulum is obtained by summing up the forces $F_{total}(t) = F_s(t) + F_f(t)$ leading to the second order differential equation

$$m\ddot{x}(t) = -r\dot{x}(t) - kx(t) \tag{4.3}$$

or

$$\ddot{x}(t) = -\frac{r}{m}\dot{x}(t) - \frac{k}{m}x(t). \tag{4.4}$$

So far, the system description is given in terms of one single time-dependent variable $x(t)$ – displacement from equilibrium – and its first and second temporal derivatives. To transform this formulations into the canonical first order description of a dynamical system, the standard device is to introduce a new variable $y(t)$ such that

$$y(t) = \dot{x}(t), \tag{4.5}$$

which implies

$$\dot{y}(t) = \ddot{x}(t).$$

Reverting the order in equation (4.5) and substituting $\dot{y}(t)$ for $\ddot{x}(t)$ as well as $y(t)$ for $\dot{x}(t)$ leads to a new system description consisting of two first order differential equations:

$$\begin{aligned} \dot{x}(t) &= y(t) \\ \dot{y}(t) &= -\frac{k}{m}x(t) - \frac{r}{m}y(t) \end{aligned}$$

Since the first derivative of the variable $x(t)$ depends on the variable $y(t)$ and vice versa, these two equations are said to be coupled.

As the last step demonstrates, by the seemingly simple trick of introducing an appropriate new variable, the original second order differential equation has been reduced to a coupled set of first order equations. This procedure can in principle be extended to any number of "primary" variables and any order of derivatives of these variables present in the equations to start with. Although this may lead to an inflation of the number of variables and possibly the loss of straightforward interpretations of the variables, a decisive advantage is the applicability of mathematical results proven for first order systems, i.e. the applicability of the mathematical theory of dynamical systems (e.g. Anosov / Arnold 1988 [9]).

For the case of the spring-mass pendulum an immediate consequence can be drawn: the new variable introduced is the velocity of pendulum movement. The inclusion of velocity as a variable describing the system means, that for specifying the current state of the system it is necessary to determine both position and velocity – which fits well with the observation that just by finding the pendulum in the equilibrium position we cannot tell whether it will remain there, move to the right or to the left (if supported horizontally).

Taking $x(t)$ and $y(t)$ to be the components of a two-dimensional vector $\vec{X}(t) = (x(t), y(t))$, the state of the system will be fully described by a vector in a two-dimensional space which is commonly called *phase space* (or also state space[3]); in a graphical representation the first component – displacement in the example – will be represented on the horizontal axis, the second component – velocity – on the vertical axis. The system description is transformed into a vectorial first order differential equation involving the state vector and its first temporal derivative:

$$\dot{\vec{X}}(t) = \vec{F}(\vec{X}(t)) = \begin{pmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{r}{m} \end{pmatrix} \cdot \vec{X}(t), \tag{4.6}$$

i.e. the function $\vec{F}(\cdot)$ in this case turns out to be a matrix multiplication.

To fully accord with the general system description of equation (4.1), the system's parameters should be taken up explicitly. In this case, the parameters are the values of the spring constant k, the mass of the pendulum m, and the frictional constant r. These values, which are taken to be constant, characterize on the one hand the material properties of the system's parts – mass and spring stiffness as captured by m and k. On the other hand, even properties pertaining to the environment are addressed by the system parameter r. In other words, even in this very small scale example, the difficulty of drawing a clear-cut distinction between system and environment becomes obvious.

The parameters of the system can also be represented in vectorial notation: Assigning the components $u_1 = m$, $u_2 = k$, and $u_3 = r$, the vector of parameters $\vec{u}$

[3]The expressions *phase space* and *state space* are commonly treated as synonyms, e.g. Norton 1995 [207], p. 51 or Pikovsky et al. 2001 [225], p. 29. In classical mechanics, the phase space comprising the generalized coordinates and related momenta is distinguished from the *space of configurations* taking up only the generalized coordinates (e.g. Goldstein 1985 [100], p. 33 / p. 274; Scheck 2003 [239], pp. 33). The term state space is justified here because "the complete dynamical description of a mechanical system is [. . .] given by a point in such a space" (Goldstein [100], p. 274).

will be

$$\vec{u} = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} = \begin{pmatrix} m \\ k \\ r \end{pmatrix}$$

and the complete system description will be

$$\dot{\vec{X}}(t) = \vec{F}(\vec{X}(t), \vec{u}) = \begin{pmatrix} 0 & 1 \\ -\frac{u_2}{u_1} & -\frac{u_3}{u_1} \end{pmatrix} \cdot \vec{X}(t) \tag{4.7}$$

(For ease of interpretation in the following we will continue to use m, k, r instead of u_1, u_2, u_3.)

Treating the parameters as constants and taking into account the linearity of matrix multiplication, we notice that the differential equation (4.6) resp. (4.7) falls under the category of homogeneous linear differential equation with constant coefficients, commonly written in the form

$$\dot{\vec{X}}(t) - \begin{pmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{r}{m} \end{pmatrix} \cdot \vec{X}(t) = 0. \tag{4.8}$$

Therefore, a solution may be obtained by assuming $x(t)$ to be of the form

$$x(t) = e^{\lambda t},$$

where λ is a constant to be determined according to conditions the solution must obey, which entails by equation (4.5) that even

$$y(t) = \lambda e^{\lambda t}$$

holds and therefore a solution of the vectorial equation will be of the form

$$\vec{X}(t) = \begin{pmatrix} e^{\lambda t} \\ \lambda e^{\lambda t} \end{pmatrix}.$$

(For a rigorous mathematical treatment see e.g. Heuser 1995 [117] or Aulbach 2004 [20].)

In this general form of solution, the value of λ has to be determined according to the specific problem under consideration. Inserting the general solution into equation (4.6) yields

$$\lambda\vec{X}(t) - \begin{pmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{r}{m} \end{pmatrix} \cdot \vec{X}(t) = \left(\lambda\mathbf{I} - \begin{pmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{r}{m} \end{pmatrix}\right) \cdot \vec{X}(t) = 0$$

where $\mathbf{I}$ denotes the two-dimensional identity matrix $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$.

This equation will be fulfilled when the determinant of the matrix

$$\lambda\mathbf{I} - \begin{pmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{r}{m} \end{pmatrix} = \begin{pmatrix} \lambda & -1 \\ \frac{k}{m} & \lambda + \frac{r}{m} \end{pmatrix}$$

is equal to 0:

$$\begin{vmatrix} \lambda & -1 \\ \frac{k}{m} & \lambda + \frac{r}{m} \end{vmatrix} = 0,$$

which in turn holds for the solutions of the so called *characteristic equation*

$$\lambda(\lambda + \frac{k}{m}) + \frac{r}{m} = \lambda^2 + \frac{k}{m}\lambda + \frac{r}{m} = 0. \tag{4.9}$$

(For a justification of the last steps, textbooks on linear algebra such as Hirsch / Smale 1974 [118], which is considered a classic in the field, can be consulted.)

To solve this quadratic equation and for the discussion of the solutions it is common practice to introduce two new constants:

The symbol ω_0 used for the first constant to be introduced is almost universally accepted in physics textbooks:

$$\omega_0^2 = \frac{k}{m}, \qquad \text{or} \quad \omega_0 = \sqrt{\frac{k}{m}}.$$

Some variation regarding the symbol to be used for the second constant, however, can be observed. We will introduce the constant β as follows:

$$2\beta = \frac{r}{m}, \tag{4.10}$$

the factor 2 is included for ease of calculation.

The first of these constants is related to the stiffness of the spring and inertia of the pendulum, the second is related to friction and again pendulum inertia.

With these definitions substituted, the quadratic equation (4.9) becomes

$$\lambda^2 + 2\beta\lambda + \omega_0^2 = 0$$

with the two solutions

$$\lambda_{1,2} = -\beta \pm \sqrt{\beta^2 - \omega_0^2}. \tag{4.11}$$

In the following, we will discuss four different cases according to values assumed by the constant β as compared to the constant ω_0.

4.1.1 No Damping: $\beta = 0$

β attains the value 0 only if there is no friction between the pendulum and its environment, i.e. if the movement of the pendulum is not reduced by interaction with the environment.

In this case, the two solutions defined by equation (4.11) of the the characteristic equation will be purely imaginary numbers:

$$\lambda_{1,2} = \pm\sqrt{-\omega_0^2} = \pm\, i\,\omega_0. \tag{4.12}$$

As there are two distinct values for λ, a complete solution of the differential equation (4.6) will consist of an appropriate superposition of the two possible solutions, i.e. $\vec{X}(t)$ will be of the form

$$\vec{X}(t) = c_1 \begin{pmatrix} e^{i\omega_0 t} \\ i\omega_0 e^{i\omega_0 t} \end{pmatrix} + c_2 \begin{pmatrix} e^{-i\omega_0 t} \\ -i\omega_0 e^{-i\omega_0 t} \end{pmatrix}. \tag{4.13}$$

The complex constants c_1 and c_2 will be specified according to requirements to be fulfilled by the solution.

A first usual requirement is that a physical phenomenon be described by real numbers. Therefore, c_1 and c_2 will have to be chosen in such a way that both components of $\vec{X}(t)$ turn out to be real-valued. To evaluate the consequences of this requirement, we will make use of

- Euler's formula: $e^{i\alpha} = \cos(\alpha) + i\sin(\alpha)$ and $e^{-i\alpha} = \cos(\alpha) - i\sin(\alpha)$
- the representation of a complex number c as $c = a + ib$, where a and b are real numbers; a is called the real part of c, b is called the imaginary part. For two complex numbers to be equal, the real and imaginary parts must be equal. The constants c_1 and c_2 will be re-written as
  $c_1 = a_1 + ib_1$,
  $c_2 = a_2 + ib_2$.

Taking these remarks into account, equation (4.13) can be re-written as follows:

$$\begin{aligned}
\vec{X}(t) &= \begin{pmatrix} c_1(\cos\omega_0 t + i\sin\omega_0 t) + c_2(\cos\omega_0 t - i\sin\omega_0 t) \\ \omega_0(ic_1(\cos\omega_0 t + i\sin\omega_0 t) - ic_2(\cos\omega_0 t - i\sin\omega_0 t)) \end{pmatrix} \\
&= \begin{pmatrix} (a_1 + ib_1)(\cos\omega_0 t + i\sin\omega_0 t) + (a_2 + ib_2)(\cos\omega_0 t - i\sin\omega_0 t) \\ \omega_0((ia_1 - b_1)(\cos\omega_0 t + i\sin\omega_0 t) - (ia_2 - b_2)(\cos\omega_0 t - i\sin\omega_0 t)) \end{pmatrix} \\
&= \begin{pmatrix} a_1\cos\omega_0 t + ia_1\sin\omega_0 t + ib_1\cos\omega_0 t - b_1\sin\omega_0 t \cdots \\ \cdots + a_2\cos\omega_0 t - ia_2\sin\omega_0 t + ib_2\cos\omega_0 t + b_2\sin\omega_0 t \\ \omega_0(ia_1\cos\omega_0 t - a_1\sin\omega_0 t - b_1\cos\omega_0 t - ib_1\sin\omega_0 t \cdots \\ \cdots - ia_2\cos\omega_0 t - a_2\sin\omega_0 t + b_2\cos\omega_0 t - ib_2\sin\omega_0 t) \end{pmatrix}
\end{aligned}$$

$$= \begin{pmatrix} a_1 \cos\omega_0 t + a_2 \cos\omega_0 t - b_1 \sin\omega_0 t + b_2 \sin\omega_0 t \cdots \\ \cdots + i(a_1 \sin\omega_0 t - a_2 \sin\omega_0 t + b_1 \cos\omega_0 t + b_2 \cos\omega_0 t) \\ \omega_0(-a_1 \sin\omega_0 t - a_2 \sin\omega_0 t - b_1 \cos\omega_0 t + b_2 \cos\omega_0 t \cdots \\ \cdots + i(a_1 \cos\omega_0 t - a_2 \cos\omega_0 t - b_1 \sin\omega_0 t - b_2 \sin\omega_0 t)) \end{pmatrix}$$

$$= \begin{pmatrix} (a_1 + a_2)\cos\omega_0 t - (b_1 - b_2)\sin\omega_0 t + \cdots \\ \cdots i((a_1 - a_2)\sin\omega_0 t + (b_1 + b_2)\cos\omega_0 t) \\ \omega_0(-(a_1 + a_2)\sin\omega_0 t - (b_1 - b_2)\cos\omega_0 t \cdots \\ \cdots + i((a_1 - a_2)\cos\omega_0 t - (b_1 + b_2)\sin\omega_0 t)) \end{pmatrix}.$$

From both components of the last line we can see that the imaginary part of the solution disappears if

$$\begin{aligned} a_1 - a_2 &= 0 \\ b_1 + b_2 &= 0 \end{aligned}$$

or

$$\begin{aligned} a_1 &= a_2 \quad = a \\ b_1 &= -b_2 = b, \end{aligned}$$

so that

$$\begin{aligned} c_1 &= a + ib \\ c_2 &= a - ib \end{aligned}$$

and c_2 is the complex conjugate of c_1 and vice versa.

Choosing c_1 and c_2 according to these requirements, the solution reduces to

$$\begin{aligned} \vec{X}(t) &= \begin{pmatrix} 2a\cos\omega_0 t - 2b\sin\omega_0 t \\ -2a\omega_0 \sin\omega_0 t - 2b\omega_0 \cos\omega_0 t \end{pmatrix} \\ &= A \begin{pmatrix} \cos\omega_0 t \\ -\omega_0 \sin\omega_0 t \end{pmatrix} + B \begin{pmatrix} \sin\omega_0 t \\ \omega_0 \cos\omega_0 t \end{pmatrix} \end{aligned} \tag{4.14}$$

with $A = 2a$ and $B = -2b$.

This final version again consists of two linearly independent parts whose components exhibit – as will have been expected – sinusoidally oscillating behavior. The oscillation frequency is given by

$$f_{osc} = \frac{\omega_0}{2\pi},$$

i.e. it is related to the system parameters spring stiffness and mass of the pendulum.

The constants A and B need to be determined e.g. by reference to initial conditions, i.e. the state of the system at a given time t: If e.g. the pendulum is released from displacement 1 with 0 velocity at time $t = 0$, then

$$\vec{X}(0) = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = A \begin{pmatrix} 1 \\ 0 \end{pmatrix} + B \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

and by comparison of vector components

$$\begin{aligned} A &= 1 \\ B &= 0, \end{aligned}$$

therefore

$$\vec{X}_{Ex1}(t) = \begin{pmatrix} \cos \omega_0 t \\ -\omega_0 \sin \omega_0 t \end{pmatrix}. \tag{4.15}$$

Now we are in the position to introduce the next concept central to the applications of the theory of dynamical systems:
The example solution equation (4.15) of the differential equation (4.6) can be interpreted as the description of a curve in the space of possible system states, the phase space of the system. Any such curve is called a *trajectory* of the system. Each point on the curve corresponds to some value(s) of time t, and the coordinates of any such point are the values of the system variables at the respective time(s) – the plural is included because any point in the example can be reached at multiple times as will immediately be demonstrated: because both components of the solution are periodic time functions with a common period T, each point on the curve (and each corresponding system state) will be taken up repeatedly after time intervals T, $2T$, $3T$, etc. This means that the curve exhibited must be a closed curve (as long as no jumps are permitted), or more generally: A system characterized by a periodic (and continuous) sequence of states will exhibit a closed (phase space) trajectory.

The closed curve described by our example solution equation (4.15) is of a rather simple elliptic shape since both components vary sinusoidally with the same period; a graphical representation for the value $\omega_0 = 1$ (in arbitrary units) is given in Figure 4.1, the horizontal axis representing excursion from the equilibrium position and the vertical axis velocity of pendulum movement.

Note that the apparent shape of the closed curve depends on the scaling of the axes.

Since in the situation discussed here there is no friction – or more generally any interaction with the environment – to slow down the movement, and therefore even the maximum size of the excursions will not decrease, a system in this case is said to exhibit no damping. In most real world situations, however, some interaction will tend to slow down movement leading to a gradual decay, which will more appropriately captured by one of the following cases to be discussed:

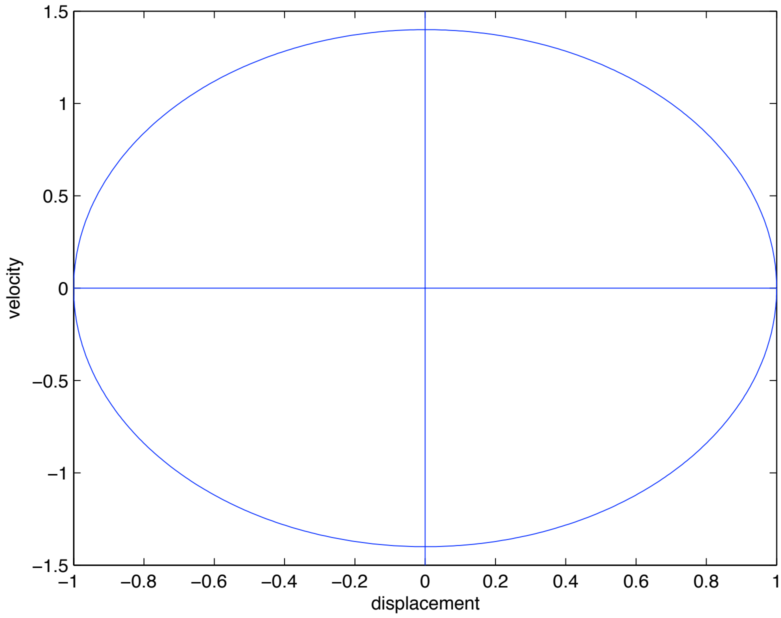


Figure 4.1: The case of no damping, $\beta = 0$. In this case, the trajectory is a closed elliptic curve in phase space. Closed trajectories describe periodic system behavior, in this case sinusoidal oscillations of the pendulum.

- weak damping, characterized by $0 < \beta < \omega_0$,
- critical damping: $\beta = \omega_0$,
- strong damping: $\beta > \omega_0$.

Some remarks will also be offered for the case of so called negative damping $\beta < 0$.

4.1.2 Weak Damping: $0 < \beta < \omega_0$

By assuming a small positive value for the constant β, i.e. a small friction parameter r considered in relation to the mass m of the pendulum and the stiffness k of the spring, the qualitative behavior of the system changes substantially.

With $0 < \beta < \omega_0$ the solutions $\lambda_{1,2}$ of the characteristic equation (4.9) will be complex numbers with non-zero real and imaginary parts; the real part is just $-\beta$, the imaginary part is given by $\sqrt{\omega_0^2 - \beta^2}$. We will introduce the abbreviation

$$\kappa = \sqrt{\omega_0^2 - \beta^2}, \tag{4.16}$$

pointing out that $\kappa < \omega_0$.

The values $\lambda_{1,2}$ can be given as

$$\lambda_{1,2} = -\beta \pm i\kappa, \tag{4.17}$$

and the solution (4.13) of the differential equation describing the system will be modified to

$$\vec{X}(t) = c_1 \begin{pmatrix} e^{-\beta t + i\kappa t} \\ (-\beta t + i\kappa t)e^{-\beta t + i\kappa t} \end{pmatrix} + c_2 \begin{pmatrix} e^{-\beta t - i\kappa t} \\ (-\beta t - i\kappa t)e^{-\beta t - i\kappa t} \end{pmatrix}. \tag{4.18}$$

Again, the requirement of a real valued system description leads to restrictions the constants c_1 and c_2 must fulfill. To spell out the consequences for this case, we will make use of the results of the previous case ($\beta = 0$) by first noticing that it is possible to factor out the common term $e^{-\beta t}$ in equation (4.18), such that

$$\vec{X}(t) = e^{-\beta t} \left(c_1 \begin{pmatrix} e^{+i\kappa t} \\ \lambda_1 e^{+i\kappa t} \end{pmatrix} + c_2 \begin{pmatrix} e^{-i\kappa t} \\ \lambda_2 e^{-i\kappa t} \end{pmatrix} \right),$$

(for shortness using λ_1, λ_2 as specified in equation (4.17)).

Since the factor $e^{-\beta t}$ is real valued it can be disregarded concerning the specification of c_1, c_2 to obtain a real valued solution. The form of the first component within the parentheses is the same as on the right-hand side in equation (4.13) and therefore will be real-valued if the same relations for c_1 and c_2 derived from that equation hold. Thus, we assume

$$\begin{aligned} c_1 &= a + ib \\ c_2 &= b - ib \end{aligned}$$

as above, which leads to the following form of the solution (4.18):

$$\vec{X}(t) = e^{-\beta t} \begin{pmatrix} 2a\cos\kappa t - 2b\sin\kappa t \\ (\lambda_1 + \lambda_2)(a\cos\kappa t - b\sin\kappa t) + i(\lambda_1 - \lambda_2)(b\cos\kappa t + a\sin\kappa t) \end{pmatrix}.$$

The steps leading to the first component are exactly the same as illustrated above, the steps leading to the second component just involve multiplication, sorting of terms, and factoring out.

This result can be further simplified because

$$\begin{aligned} \lambda_1 + \lambda_2 &= -2\beta \\ \lambda_1 - \lambda_2 &= 2i\kappa \end{aligned}$$

resulting in

$$\vec{X}(t) = e^{-\beta t} \begin{pmatrix} 2a\cos\kappa t - 2b\sin\kappa t \\ -2a(\beta\cos\kappa t + \kappa\sin\kappa t) - 2b(-\beta\sin\kappa t + \kappa\cos\kappa t) \end{pmatrix},$$

which reduces to

$$\vec{X}(t) = e^{-\beta t}\left(A\begin{pmatrix}\cos\kappa t\\ -\beta\cos\kappa t - \kappa\sin\kappa t\end{pmatrix} + B\begin{pmatrix}\sin\kappa t\\ -\beta\sin\kappa t + \kappa\cos\kappa t\end{pmatrix}\right). \quad (4.19)$$

As in equation (4.14), this solution contains two linearly independent oscillatory terms, whose components vary sinusoidally with the frequency $f_{osc} = \frac{\kappa}{2\pi}$. As evident from equation (4.16), this frequency is lower than in the case with zero friction – which does not come as a surprise because friction slows down the movement of the pendulum. The phase relation between the components of the oscillatory terms is somewhat more complicated than in the case of equation (4.14) depending on the relation between β and κ resp. ω_0 or more fundamentally on the intrinsic relations between the system's parameters.

The most important difference compared to equation (4.14), however, is expressed by the common factor $e^{-\beta t}$, which describes a gradual decrease of the width of oscillation in both components, eventually approaching a state of rest in the equilibrium position. This change entails that the system behavior is no longer periodic and the trajectory in phase space turns into a curve spiralling towards the origin as will be illustrated by another example.

Again, we consider the case of releasing the pendulum from position 1 with 0 velocity at time $t = 0$, i.e. starting from initial state $\begin{pmatrix}1\\0\end{pmatrix}$. The constants A and B then need to fulfill the following equation:

$$\begin{pmatrix}1\\0\end{pmatrix} = A\begin{pmatrix}1\\-\beta\end{pmatrix} + B\begin{pmatrix}0\\\kappa\end{pmatrix}.$$

From the first component, it can immediately be seen that

$$A = 1,$$

and, as a consequence, the second component yields

$$0 = -\beta + \kappa B$$

or

$$B = \frac{\beta}{\kappa}.$$

The example solution therefore is

$$\vec{X}_{Ex2}(t) = e^{-\beta t}\begin{pmatrix}\cos\kappa t + \frac{\beta}{\kappa}\sin\kappa t\\ -(\kappa + \frac{\beta^2}{\kappa})\sin\kappa t\end{pmatrix}. \quad (4.20)$$

In Figure 4.2, this solution is plotted for two different values of β indicated in the legend, retaining the value $\omega_0 = 1.4$ applied in Figure 4.1.

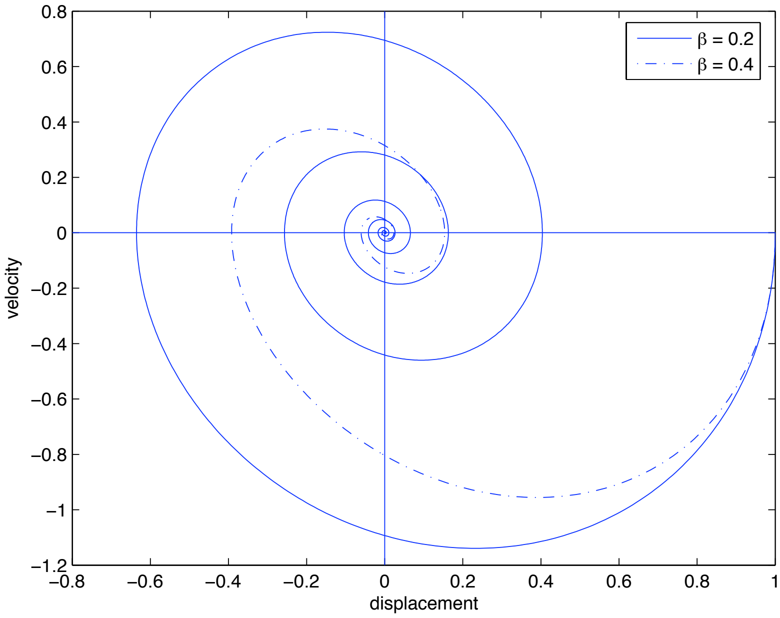


Figure 4.2: Weak damping, $0 < \beta < \omega_0$. The trajectory is no longer closed, i.e. system behavior is no longer periodic. Here, a gradually decaying oscillatory movement is described.

Comparing Figures 4.1 and 4.2, the qualitative change of the trajectories' shape can clearly be noticed: Changing the value of β from 0 to a small positive value will change the closed elliptic trajectory into an inward spiral, further increasing β will result in a steeper inward movement and fewer turns. Thus, inspection of the systems trajectories may reveal even some quantitative information about system parameters. However, we will point out that time is only implicit in this kind of graphical representation, so that the duration of a process cannot immediately be inferred from the corresponding (part of a) trajectory. Specifically, time is not linearly represented by the length of the part of a trajectory: in Figure 4.2, both the solid and the dash-dotted curves reflect the same duration.

4.1.3 Critical Damping: $\beta = \omega_0$ and Strong Damping: $\beta > \omega_0$

In introductory texts in mathematics (e.g. Aulbach 2004 [20], Chapter 6.5) and (theoretical) physics (e.g. Greiner 1984 [103], Section 23), these two cases are typically treated separately reflecting the different mathematical techniques employed to obtain a solution and the different resulting mathematical forms of those solutions. Qualitatively (i.e. regarding the overall shape of the trajectories), however, the cases don't appear to differ greatly. Therefore, we will only discuss the case $\beta = \omega_0$ as this will be taken up again in Section 4.2 and merely state and graphically represent the solution for the case $\beta > \omega_0$ to support the claim of qualitative similarity.

If $\beta = \omega_0$, the term $\sqrt{\beta^2 - \omega_0^2}$ vanishes; as a consequence the two solutions $\lambda_{1,2}$ of the characteristic equation (4.9) given by eq. (4.11) will coincide and become

$$\lambda_{1,2} = -\beta,$$

yielding *one* solution of the form

$$\vec{X}_1(t) = A \begin{pmatrix} e^{-\beta t} \\ -\beta e^{-\beta t} \end{pmatrix}. \tag{4.21}$$

One solution, however, will in general be insufficient to accommodate an arbitrary initial state $\vec{X}(0)$ as can easily be demonstrated by the initial state $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$: Setting $t = 0$ in equation (4.21) leads to

$$\vec{X}_1(0) = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = A \begin{pmatrix} 1 \\ -\beta \end{pmatrix}.$$

The first component of this equation requires $A = 1$, the second component on the other hand leads to the incompatible requirement $A = 0$. For this reason a second, linearly independent[4] solution is needed.

It can be shown (e.g. Aulbach 2004 [20]; Heuser 1995 [117]) that a suitable solution is obtained by setting the first component to $x(t) = te^{-\beta t}$; because the second component $y(t)$ is the first temporal derivative of $x(t)$, i.e.

$$y(t) = \dot{x}(t) = e^{-\beta t} - \beta t e^{-\beta t} = (1 - \beta t)e^{-\beta t},$$

the second solution will be specified by

$$\vec{X}(t) = B \begin{pmatrix} te^{-\beta t} \\ (1 - \beta t)e^{-\beta t} \end{pmatrix}, \tag{4.22}$$

and a general solution of our differential equation will be of the form

$$\vec{X}(t) = A \begin{pmatrix} e^{-\beta t} \\ -\beta e^{-\beta t} \end{pmatrix} + B \begin{pmatrix} te^{-\beta t} \\ (1 - \beta t)e^{-\beta t} \end{pmatrix}. \tag{4.23}$$

[4]which in this case simply means that it cannot be obtained by multiplying the first solution with some constant factor

With this solution the initial state $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ can again be expressed by an appropriate choice of A and B as in the previous cases: The constants will be determined by the following (set of) equation(s):

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix} = A \begin{pmatrix} 1 \\ -\beta \end{pmatrix} + B \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

The first line states that

$$A =,$$

which in turn leads to

$$B = \beta.$$

In the case $\beta = \omega_0$, the complete mathematical description of the behavior of a spring-mass pendulum released from rest in the position 1 is given by

$$\vec{X}_{Ex3}(t) = \begin{pmatrix} e^{-\beta t} \\ -\beta e^{-\beta t} \end{pmatrix} + \beta \begin{pmatrix} te^{-\beta t} \\ (1-\beta t)e^{-\beta t} \end{pmatrix}. \tag{4.24}$$

The corresponding trajectory is plotted (red curve) along with the trajectories for four additional initial states[5] in Figure 4.3. In the figure, the initial states are marked with asterisks from which the trajectories are "emanating"; again, the value $\omega_0 = \beta = 1.4$ has been retained as well as the range of values for t.

It can be observed that none of the trajectories crosses the vertical axis, which means that the pendulum does not go past the equilibrium position. Rather, in all examples shown a gradual decrease in speed and approach towards the equilibrium position is seen. Therefore, in the case of $\beta = \omega_0$ the pendulum no longer exhibits oscillatory behavior, which is characterized by repeated change of direction of movement and passing an equilibrium position in both directions.

As the case $\beta = \omega_0$ constitutes a boundary condition for the parameter (combination) β:

- for $\beta < \omega_0$ oscillatory behavior is observed,
- for $\beta \geq \omega_0$ there will be no oscillatory behavior,

this case is commonly labeled *critical damping*.

In the case $\beta > \omega_0$, which is referred to as *strong damping*, there will again be two different solutions $\lambda_{1,2}$ of the characteristic equation (4.9):

$$\begin{aligned} \lambda_1 &= -\beta + \sqrt{\beta^2 - \omega_0^2} = -(\beta - \sqrt{\beta^2 - \omega_0^2}), \\ \lambda_2 &= -\beta - \sqrt{\beta^2 - \omega_0^2} = -(\beta + \sqrt{\beta^2 - \omega_0^2}), \end{aligned}$$

[5]The mathematical description of the additional trajectories is deferred to Appendix B.

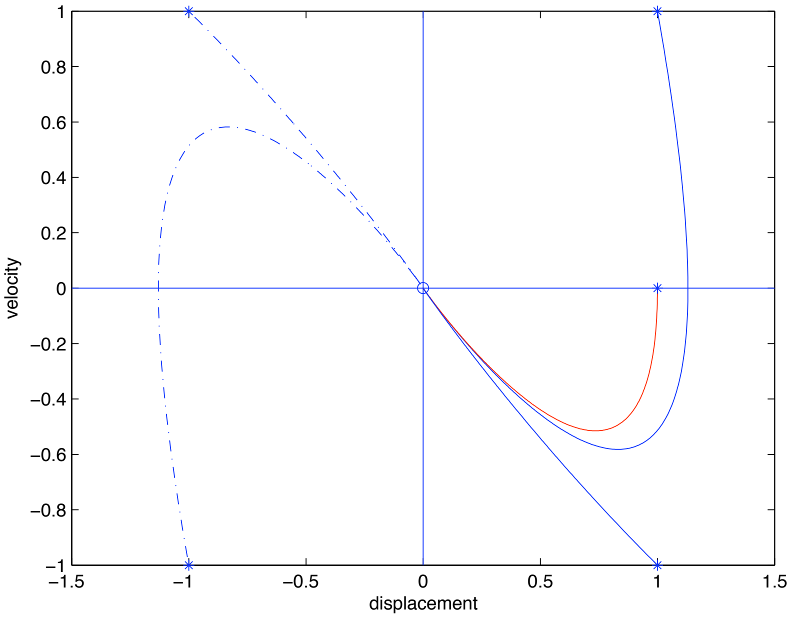


Figure 4.3: The case of *critical damping* $\beta = \omega_0$. The trajectories are shown for five different initial states marked by asterisks in the drawing; the time span shown is the same for all curves.

and the corresponding solution of the differential equation is

$$\begin{aligned} \vec{X}(t) &= A \begin{pmatrix} e^{-(\beta - \sqrt{\beta^2 - \omega_0^2})t} \\ -(\beta - \sqrt{\beta^2 - \omega_0^2}) e^{-(\beta - \sqrt{\beta^2 - \omega_0^2})t} \end{pmatrix} \cdots \\ & \quad \cdots + B \begin{pmatrix} e^{-(\beta + \sqrt{\beta^2 - \omega_0^2})t} \\ -(\beta + \sqrt{\beta^2 - \omega_0^2}) e^{-(\beta + \sqrt{\beta^2 - \omega_0^2})t} \end{pmatrix} . \end{aligned}$$

Since $\sqrt{\beta^2 - \omega_0^2} < \beta$, all exponents in this equation will be negative for positive values of t, giving rise to a continual decrease in the absolute value of all components of the solution with increasing t, regardless of the choice of A and B. Thus, as in the case $\beta = \omega_0$, a gradual approach to the equilibrium state (zero position and zero velocity) will be characteristic of the spring-mass pendulum with $\beta > \omega_0$ as illustrated by the five trajectories of Figure 4.4. For this figure, the same initial states were chosen as in Figure 4.3, the value of ω_0 and the time span are the same as in all previous examples, and β is set to 1.8.

As claimed above, the shapes of the trajectories in Figure 4.4 are very similar to those of Figure 4.3, which can be considered indicative of qualitatively similar system behaviors.

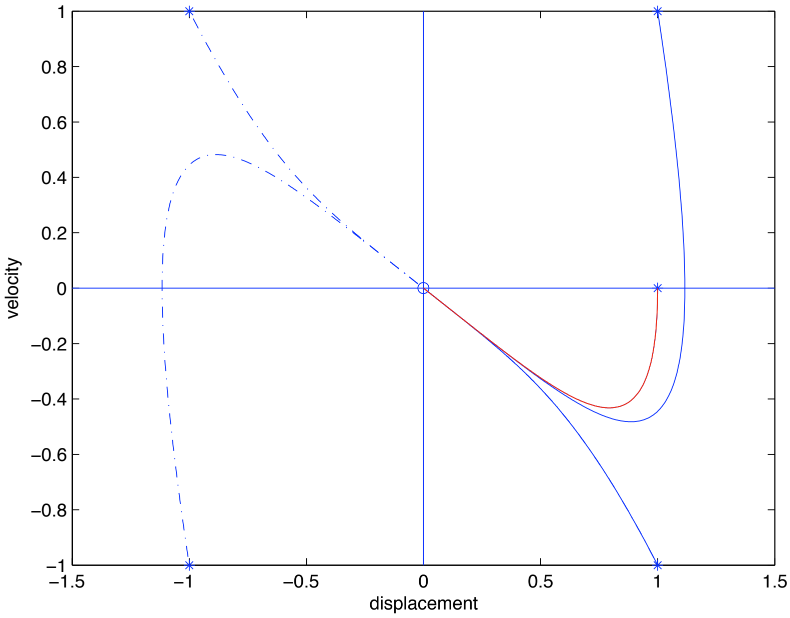


Figure 4.4: The case of *strong damping* $\beta > \omega_0$. The trajectories very much resemble the trajectories for critical damping and identical initial states.

4.1.4 Negative Damping: $\beta < 0$

The case of *negative damping* is usually not treated in the context of mechanical systems but will be included here as a reference for later discussion.

Changing the sign of the frictional constant in the differential equation describing system behavior means that the interaction described will not counteract movement of the pendulum but rather support, i.e. speed up movement. In the case of negative damping we will therefore expect gradually increasing movement as opposed to the cases of weak and critical / strong damping.

The formal treatment can be reduced to the results of the previous sections by introducing a new constant $\nu = -\beta$ and replacing β as appropriate in the equations given above. We will distinguish two cases in analogy to the treatment of damping described above:

- $0 < |\beta| < \omega_0$ or $0 < \nu < \omega_0$ and
- $|\beta| = \omega_0$ or $\nu = \omega_0$,

the case $|\beta| > \omega_0$ resp. $\nu > \omega_0$ will be assumed to be qualitatively similar to the case $\nu = \omega_0$.

For $0 < \nu < \omega_0$, we can directly modify equation (4.20) to obtain a solution for initial state $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$:

$$\vec{X}_{Ex4}(t) = e^{\nu t} \begin{pmatrix} \cos \kappa t - \frac{\nu}{\kappa} \sin \kappa t \\ -(\kappa + \frac{\nu^2}{\kappa}) \sin \kappa t \end{pmatrix};$$

because of equation (4.16) and $\nu^2 = \beta^2$, the value of κ will be the same as above.

For $\nu = \omega_0$, equation (4.24) will be modified in the following form:

$$\begin{aligned} \vec{X}_{Ex5}(t) &= \begin{pmatrix} e^{\nu t} \\ \nu e^{\nu t} \end{pmatrix} - \nu \begin{pmatrix} t e^{\nu t} \\ (1 + \nu t) e^{\nu t} \end{pmatrix} \\ &= e^{\nu t} \begin{pmatrix} 1 - \nu t \\ -\nu^2 t \end{pmatrix}, \end{aligned}$$

again giving a solution for initial state $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$.

The corresponding trajectories are displayed in the two panels of Figure 4.5.

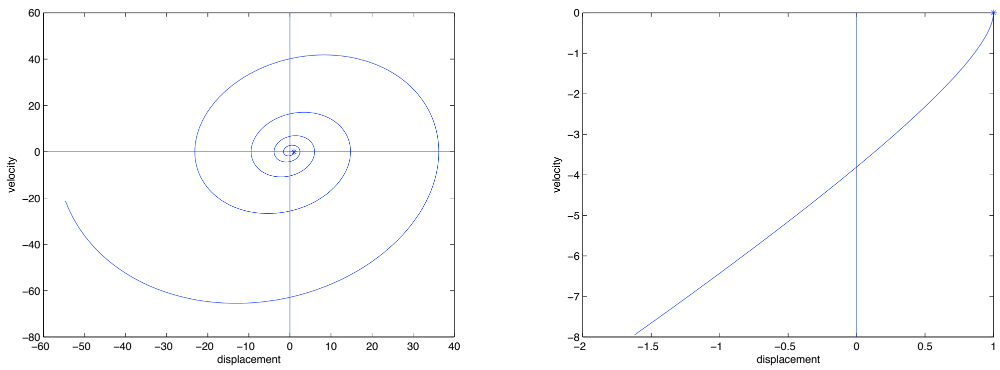


Figure 4.5: The case of *negative damping* for $\beta = -0.2$ (left panel) and $\beta = -1.4$ (right).

In the left panel, corresponding to $\nu = 0.2$ resp. $\beta = -0.2$, oscillatory behavior with increasing maximum displacement / velocity can be noticed, represented by a trajectory spiralling outward.

In analogy to the cases of critical and strong damping, for $\nu \geq \omega_0$ resp. $\beta \leq -\omega_0$, oscillations will no longer be observed. Instead, the pendulum will start from the initial state and acquire an ever increasing negative velocity and displacement; the corresponding trajectory approaches a straight line leading to the lower left.

These kinds of behavior will not be observed in passive mechanical systems; rather, some active elements feeding energy into the system will be required.

4.1.5 Intermediate Summary – Harmonic Oscillator

To avoid getting lost in details, some general traits observed in the discussion of the spring-mass pendulum will be summarized and somewhat extended.

1. Any system, whose (second order) equation of motion can be stated in the form
$$\ddot{x}(t) + 2\beta\dot{x}(t) + \omega_0^2 x = 0,$$
where β and ω_0 are any constant functions of the system's parameters (and therefore may be considered "higher order" parameters themselves) is called a harmonic oscillator. Because this equation is of the same form as equation (4.4) (with the adequate definitions of β and ω_0), the results of the preceding sections can immediately be extended to any system that can be considered a harmonic oscillator with due change of interpretation of the system variables and parameters. In particular, any such system exhibits the same sort of oscillating or decaying / growing behavior as discussed above.

2. The momentary state of a system at time t can be represented as a point in phase space / state space corresponding to an assignment of values to the system variables at time t. The sequence of states traversed by the system over some time span is represented by a curve in phase space called trajectory or orbit. In particular, closed trajectories correspond to periodic system behavior.

3. The global shape of the trajectories – reflecting global properties of system behavior – may change according to the values of system parameters. In the discussion of the spring-mass pendulum, changes depending on the value β were demonstrated.

4. The variable $x(t)$ was introduced as representing the excursion of the pendulum from an equilibrium position. A more general concept is the notion of an equilibrium state represented by a "point of equilibrium" in phase space. A state of equilibrium is characterized by the system not changing its state over time. Therefore, in an equilibrium state the temporal derivative of the state vector must vanish, which for the spring-mass pendulum (and more generally any two-dimensional linear system) is the case for the state $\vec{X} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ as can be verified by insertion of this state vector in equation (4.6).

   The character of this equilibrium state, however, will change with the value of β:

- For $\beta > 0$, any initial condition will result in the system eventually approaching the equilibrium point (as reflected by the trajectories spiralling inward). This kind of equilibrium state is called stable equilibrium.
- For $\beta = 0$, every initial condition will lead to a system behavior represented by a trajectory that remains in a bounded region containing the initial state and the equilibrium state. Even this kind of equilibrium is considered stable, sometimes also called neutral (Guckenheimer / Holmes 1983 [107], p. 4).
- For $\beta < 0$, the trajectory representing system behavior will eventually leave any bounded region for initial states not equal to the equilibrium state; this is an example of unstable equilibrium.
- Both stable and unstable equilibrium can be further differentiated: For $0 < |\beta| < \omega_0^2$, i.e. for decreasing ($\beta > 0$) and increasing ($\beta < 0$) oscillatory behavior, the equilibrium point is called a stable resp. unstable focus; for $|\beta| > \omega_0$ the point is called a stable resp. unstable node (e.g. Ellner / Guckenheimer 2006 [78], Chapter 5.3.2).

The change of stability conditions due to the variation of system parameters is treated in bifurcation theory. For an accessible treatment of some standard forms of bifurcations, see Strogatz 1994 [262], Chapter 3. For a more formal discussion of equilibrium and criteria for different types of equilibrium see Arbib 1989 [11], section 3.2; Guckenheimer / Holmes 1983 [107], Chapter 1; Arnold / Il'yashenko [16], Chapter 1, §4.

It is these kinds of general considerations that contribute substantially to the appeal of the dynamical systems approach.

4.2 Why Music Researchers Should Care: An Analysis of the Resonance Model of Pulse Perception

The spring-mass pendulum is frequently taken up as a first step in the discussion of more complex devices that are part of the movement system of the (human) body. For example Arbib 1989 [11], Section 3.2, discusses the muscle as a spring-mass system integrated into the motion control of the human arm; similar ideas show up in contributions dedicated to the investigation of rhythm perception and production such as Eck / Gasser / Port 2000 [75]; Peper / Beek / Daffertshofer 2000 [215]; Todd / Lee / O'Boyle 2002 [281].

In a widely received paper, van Noorden and Moelants (1999 [291]) attempt to relate temporal preferences observed in the study of pulse perception and pulse production, e.g. by tapping in synchrony with an isochronous pulse pattern, to dynamical (physical) properties of the human body. More specifically, they propose a model based on the oscillatory behavior of a forced harmonic oscillator, which is most notably characterized by the phenomenon of *resonance*. The observed preference for an inter-tap interval of 0.5 seconds resp. a tapping frequency of 2 Hz is explained in terms of the so-called resonance frequency of an oscillatory formed by the different parts of the human body.

In our analysis we will completely disregard the experimental data adduced by van Noorden / Moelants and instead concentrate on the technical and conceptual specification of the resonance model.

A forced harmonic oscillator can be treated as an example of a dynamical system that is subject to external influences. Taking up our example of a spring-mass pendulum this is said to be forced if a temporally varying force is exerted on the pendulum; the resulting oscillation pattern is called forced oscillation.

In the mathematical description, an external force $F_{ext}(t)$ will be entered as an additional term on the right hand side of the equation of motion (4.3)

$$m\ddot{x}(t) = -r\dot{x}(t) - kx(t) + F_{ext}(t) \tag{4.25}$$

because of Newton's third axiom requiring that $F_t(t) = F_s(t) + F_f(t) + F_{ext}(t)$. In standard textbook discussions – which provide the standard description of the resonance phenomenon – $F_{ext}(t)$ is assumed to vary sinusoidally with a frequency $f = \frac{\omega}{2\pi}$, i.e. to be of the form

$$F_{ext}(t) = E\cos\omega t.$$

Entering this into equation (4.25) and dividing by m leads to

$$\ddot{x}(t) = -2\beta\dot{x}(t) - \omega_0^2 x(t) + \eta\cos\omega t$$

with β and ω_0 as defined above and $\eta = \frac{E}{m}$.

Using again the substitution (4.5), this equation will be transformed into the first order system

$$\begin{aligned} \dot{x}(t) &= y(t) \\ \dot{y}(t) &= -\omega_0^2 x(t) - 2\beta y(t) + \eta\cos\omega t \end{aligned}$$

or in vectorial form

$$\dot{\vec{X}}(t) = \begin{pmatrix} 0 & 1 \\ -\omega_0^2 & -2\beta \end{pmatrix} \vec{X}(t) + \eta \begin{pmatrix} 0 \\ \cos\omega t \end{pmatrix},$$

i.e. the system description, equation (4.6), is extended by an additive term that is explicitly time dependent.

Note again that in this formulation nothing particular is said about the kind of interaction and the way force is exerted on the pendulum – this kind of interpretation is part of the task of relating the mathematical description to the phenomena to be described.

A solution of this differential equation is commonly obtained by assuming the pendulum (oscillator) to perform sinusoidal oscillations with the frequency f of the external force but allowing for phase differences, i.e. the first component of the solution is assumed to be of the form

$$x(t) = A\cos(\omega t + \varphi)$$

and therefore

$$\begin{aligned} y(t) &= -\omega A \sin(\omega t + \varphi) \\ \dot{y}(t) &= -\omega^2 A \cos(\omega t + \varphi), \end{aligned}$$

so that the vectorial equation will be

$$\begin{pmatrix} -\omega A \sin(\omega t + \varphi) \\ -\omega^2 A \cos(\omega t + \varphi) \end{pmatrix} = \begin{pmatrix} -\omega A \sin(\omega t + \varphi) \\ -\omega_0^2 A \cos(\omega t + \varphi) + 2\beta\omega A \sin(\omega t + \varphi) + \eta \cos \omega t \end{pmatrix}.$$

The first component will be true for any choice of A and φ, therefore only the second component needs to be evaluated.

Using the trigonometric identities for sums of angles for sine and cosine functions, the second component can be re-written as

$$\begin{aligned} -\omega^2 A(\cos\omega t \cos\varphi - \sin\omega t \sin\varphi) &= -\omega_0^2 A(\cos\omega t\cos\varphi - \sin\omega t \sin\varphi)\cdots \\ &\quad \cdots + 2\beta\omega A(\sin\omega t\cos\varphi + \cos\omega t\sin\varphi) + \eta\cos\omega t. \end{aligned}$$

Bringing all terms to the left-hand side and collecting terms multiplying $\cos\omega t$ and $\sin\omega t$ resp. leads to

$$\begin{aligned} \cos\omega t(-\omega^2 A\cos\varphi + \omega_0^2 A\cos\varphi - 2\beta\omega A\sin\varphi + \eta)\cdots & \\ \cdots + \sin\omega t(\omega^2 A\sin\varphi - \omega_0^2 A\sin\varphi - 2\beta\omega A\cos\varphi) &= 0 \end{aligned}$$

This equation will only be true for all t if the factors multiplying $\cos\omega t$ and $\sin\omega t$ vanish, i.e. if the equations

$$\begin{aligned} A\cos\varphi(\omega_0^2 - \omega^2) - 2\beta\omega A\sin\varphi + \eta &= 0 \\ A\sin\varphi(\omega^2 - \omega_0^2) - 2\beta\omega A\cos\varphi &= 0 \end{aligned} \tag{4.26}$$

both hold.

The phase relation between the external force and the resulting forced oscillation can be derived from the last line bringing the second term to the right-hand side and dividing by $(\omega^2 - \omega_0^2)$

$$A \sin \varphi = \frac{2\beta\omega A \cos \varphi}{\omega^2 - \omega_0^2} \tag{4.27}$$

and then dividing by $A \cos \varphi$:

$$\tan \varphi = \frac{\sin \varphi}{\cos \varphi} = \frac{2\beta\omega}{\omega^2 - \omega_0^2}. \tag{4.28}$$

The relation between phase shift φ and frequency $f = \frac{\omega}{2\pi}$ of the external force is shown in Figure 4.6 for a resonant frequency $f_0 = \frac{\omega_0}{2\pi} = 2$ Hz and $\beta = 0.2$; all values of phase shift being negative indicates a phase *lag* for all values of f. Phase lag for a given resonance frequency according to this equation depends only on the frequency of the external force and the friction-related constant β but is independent of the strength (amplitude) of the external force. In particular, for *all* values of β, phase lag for $\omega = \omega_0$ – the frequency of the external force being equal to the resonance frequency of the oscillating system – is $\frac{\pi}{2}$, i.e. corresponding to a quarter of a period of the external force or 125 ms for a period of 500 ms.

To determine the amplitude A of the forced oscillations, equation (4.27) will be used to replace $A \sin \varphi$ in equation (4.26):

$$A \cos \varphi(\omega_0^2 - \omega^2) - \frac{(2\beta\omega)^2 A \cos \varphi}{\omega^2 - \omega_0^2} + \eta = 0,$$

which can be transformed to

$$A \cos \varphi \frac{(\omega_0^2 - \omega^2)^2 + (2\beta\omega)^2}{\omega_0^2 - \omega^2} = -\eta$$

or

$$A \cos \varphi = -\eta \frac{\omega_0^2 - \omega^2}{(\omega_0^2 - \omega^2)^2 + (2\beta\omega)^2} = \eta \frac{\omega^2 - \omega_0^2}{(\omega_0^2 - \omega^2)^2 + (2\beta\omega)^2}.$$

This can in turn be re-inserted in equation (4.27) to yield

$$A \sin \varphi = \eta \frac{2\beta\omega}{(\omega_0^2 - \omega^2)^2 + (2\beta\omega)^2}.$$

The absolute value of A can now be determined by summing the squares of the last two equations[6] and taking the square root; the value is given as a function

[6] making use of the trigonometric identity $\sin^2 \varphi + \cos^2 \varphi = 1$

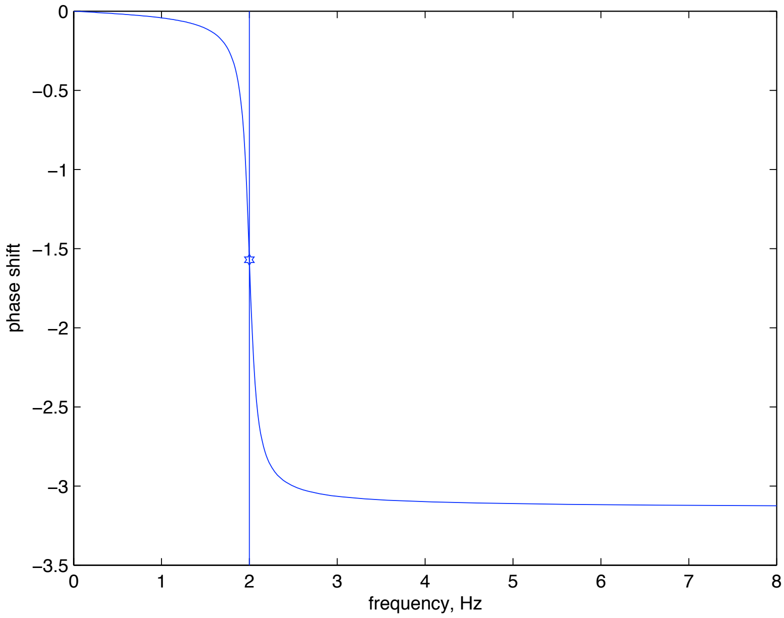


Figure 4.6: Phase shift for forced oscillations as compared to phase of the external force $F_{ext} = E \cos \omega t$.

of angular frequency ω:

$$
\begin{aligned}
A(\omega) &= \eta \sqrt{\frac{(\omega^2 - \omega_0^2)^2 + (2\beta\omega)^2}{((\omega_0^2 - \omega^2)^2 + (2\beta\omega)^2)^2}} \\
&= \eta \frac{\sqrt{(\omega^2 - \omega_0^2)^2 + (2\beta\omega)^2}}{(\omega^2 - \omega_0^2)^2 + (2\beta\omega)^2} \\
A(\omega) &= \eta \frac{1}{\sqrt{(\omega^2 - \omega_0^2)^2 + (2\beta\omega)^2}} \qquad (4.29)
\end{aligned}
$$

Equation (4.29) is essentially the mathematical description of the *resonance curve* van Noorden / Moelants are starting from. It is characteristic of the shape of the curve and of the values of amplitude of the forced oscillations that for low and high frequencies $\frac{\omega}{2\pi}$ relatively low amplitudes are encountered; around the resonant frequency $\frac{\omega_0}{2\pi}$ there is a maximum in the values of $A(\omega)$. The maximum becomes more pronounced for small β – i.e. the amplitude of forced oscillations at the resonant frequency becomes greater with low damping – and is nearly levelled out for high values of β. Resonance curves for $\beta = 0.2$ and $\beta = 4$ and a resonant frequency of $\frac{\omega_0}{2\pi} = 2$ Hz have been plotted in Figure 4.7; scaling is the same for both curves and has been adjusted so that the maximum value for $\beta = 0.2$ is 1.

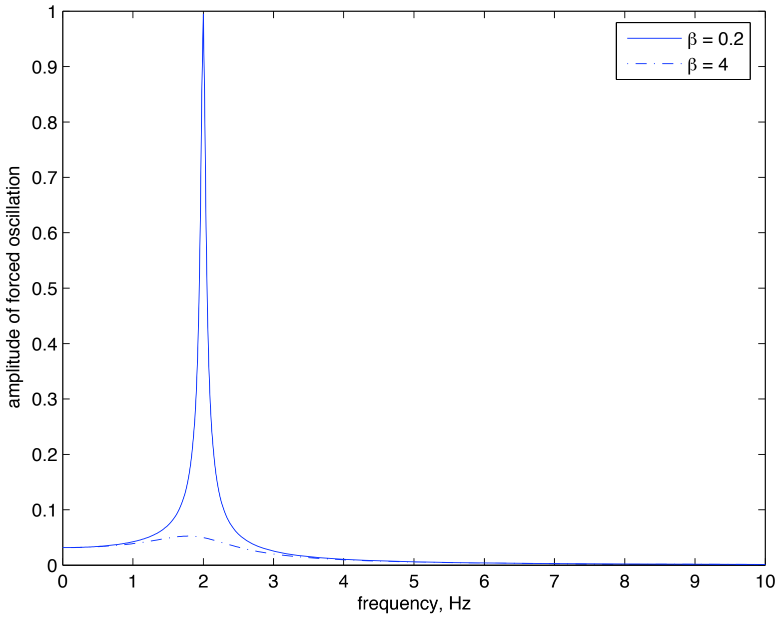


Figure 4.7: Standard resonance curves for two different values of β and a resonance frequency of $\frac{\omega_0}{2\pi} = 2$ Hz.

For their model van Noorden / Moelants assume a damped forced harmonic oscillator with a resonance frequency $f_{=}\frac{\omega_0}{2\pi} = 2$ Hz, whose relation between amplitude and frequency is described by equation (4.29). From this amplitude function they subtract the amplitude function of an oscillator with the same ω_0 but for the case of critical damping, i.e. for the case $\beta = \omega_0$ discussed above.

The amplitude function $A_{crit}(\omega)$ to be subtracted reduces to

$$\begin{aligned}
A_{crit}(\omega) &= \eta\frac{1}{\sqrt{(\omega_0^2 - \omega^2) + (2\omega_0\omega)^2}} \\
&= \eta\frac{1}{\sqrt{\omega_0^4 - 2\omega_0^2\omega^2 + \omega^4 + 4\omega_0^2\omega^2}} \\
&= \eta\frac{1}{\sqrt{/omega_0^4 + 2\omega_0^2\omega^2 + \omega^4}} \\
&= \eta\frac{1}{\sqrt{(\omega_0^2 + \omega^2)}} \\
&= \eta\frac{1}{\omega_0^2 + \omega^2}.
\end{aligned}$$

Therefore, the formula for the effective resonance curve $A_{eff}(\omega) = A(\omega) - A_{crit}(\omega)$ in terms of angular frequency is given by

$$A_{eff}(\omega) = \eta \left(\frac{1}{\sqrt{(\omega_0^2 - \omega^2)^2 + (2\beta\omega)^2}} - \frac{1}{\omega_0^2 + \omega^2} \right).$$

For better comparison, this formula is recast in terms of frequency:

$$A_{eff}(f) = \frac{\eta}{4\pi^2} \left(\frac{1}{\sqrt{(f_0^2 - f^2)^2 + \frac{\beta^2}{\pi^2} f^2}} - \frac{1}{f_0^2 + f^2} \right),$$

in contrast to

$$\tilde{A}_{eff}(f) = frac\eta 4\pi^2 \left(\frac{1}{\sqrt{(f_0^2 - f^2)^2 + \frac{\beta^2}{\pi^2} f^2}} - \frac{1}{\sqrt{f_0^4 + f^4}} \right)$$

as van Noorden / Moelants' equation (3) should read with the normalization conventions adopted here.

The difference is explained as follows: the constant β of van Noorden / Moelants, equation (1), corresponds to $\frac{\beta^2}{\pi^2}$ here. Therefore, the value β_{crit} given in equation (2) of van Noorden / Moelants should be

$$\beta_{crit} = \frac{\omega_0^2}{\pi^2} = \left(\frac{\omega_0}{\pi}\right)^2 = \left(\frac{2\pi f_0}{\pi}\right)^2 = (2f_0)^2 \neq 2f_0^2$$

The overall shape of the curve does not change much: Figure 4.8 shows the "effective resonance curve" in the corrected version (solid line) and according to the formula given by van Noorden / Moelants, plotted as a function of period instead of frequency[7] of the external force; the value $\frac{\beta^2}{\pi^2}$ is set to 0.2. Possibly, the slight differences will not produce any significant changes in the fit between theoretical values and empirical data.

Besides revealing a minor technical flaw, however, the detailed analysis of the model presented here makes obvious a conceptual problem: A resonance curve describes an aspect of the behavior of a dynamical system with a specified, fixed setting of system parameters. The "effective resonance curve", specified by the difference in amplitude between a – presumably weakly, i.e. $\beta < \omega_0$ – damped forced harmonic oscillator and the self-same harmonic oscillators with damping set to the critical value $\beta = \omega_0$, is part of the description of the behavior of a dynamical system one of whose parameters – β – is assigned two values simultaneously. That is: the "effective resonance curve" describes the behavior of a system that cannot exist.

[7]see remark in the next section

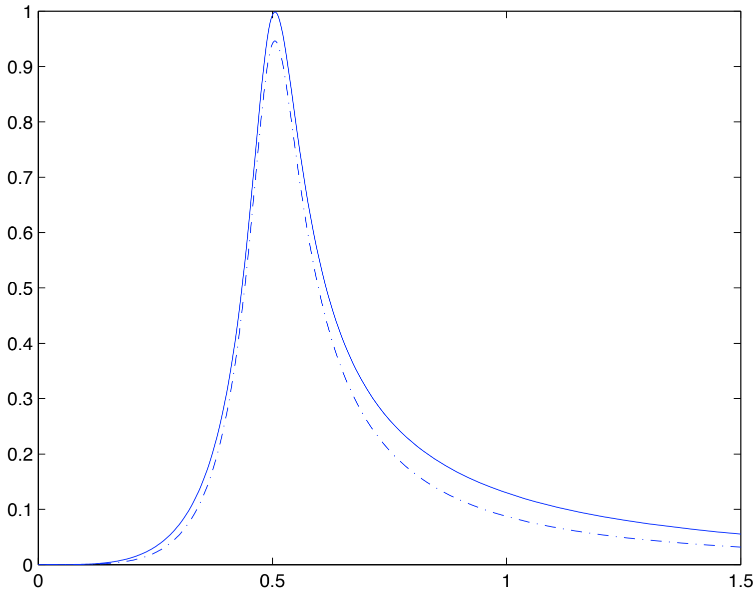


Figure 4.8: The *effective resonance curve* in the corrected version (solid line) and as specified by van Noorden / Moelants 1999 [291] (dash-dotted) as a function of oscillation period. The value $\frac{\beta^2}{\pi^2}$ (or β in the terminology of van Noorden / Moelants) is set to 0.2.

But not only the concept of "effective resonance" appears to be problematic: As demonstrated above, equation (4.28) and Figure 4.6, an inherent feature of the (resonating) forced spring-mass pendulum is a phase lag between the external force and the resulting oscillation, which for the resonance frequency of the system amounts to a quarter oscillation period. Therefore, tapping in synchrony with an external pulse can hardly be explained as a resonance phenomenon: assuming a resonance frequency of 2 Hz – as shown above – a time lag of 125 ms between stimulating pulse and resulting tap should have to be expected. In the context of the design of a robotic drummer intended to tap along in synchrony with an external beat, related problems are taken up by Crick / Munz / Scassellati 2006 [63].

To a certain degree, the problems pointed out here are imported into the work of Toiviainen / Snyder 2003 [284], presented under the title *Tapping to Bach: Resonance-Based Modeling of Pulse*. Although no explicit reference to the mechanisms underlying oscillatory behavior of the human body is included in the formalism developed, the observed phenomenon of "preferred pulse period" is related to the "characteristic oscillation of human limbs" ([284], page 49). Further, in the dynamic system (here used in the sense of mathematical formalism) proposed, a weighting function is included which "approximates the resonance curve obtained by van Noorden and Moelants (1999)" ([284], page 68). Their dynamical system captures the temporal development of the activation strength – somewhat misleadingly labeled "resonance value" – of possible modes of tapping to the musical excerpts presented.

Although the phenomenon of resonance as such will be considered inappropriate to explain the observed phenomena, the importance of resonance will not be denied. Rather, it seems necessary to look for a way to integrate resonance-related properties into an active system that provides the possibility for synchronization without time lag.

A first step into this direction, which is currently investigated by van Noorden (e.g. presentation at ISSSM 2007, private communication), may be the adoption of the van der Pol oscillator to be discussed as the next textbook example of a dynamical system.

4.2.1 Another Reason to Care: Quality Management

Another reason to care about technical details of the formal model descriptions employed should be the goal to keep up the quality of scientific publications. As an example, we will present the unlucky fate of the formula for the "effective resonance curves" in two articles that appeared in *Music Perception* in 2005 and 2006 – *Music Perception* is considered an internationally high-ranking journal within music research.

Before turning to the details, however, we will point out a minor aspect concerning the presentation of the resonance model in all cited publications: The formula for the resonance curves is specified in terms of oscillation *frequency* f; graphically, on the other hand, the curves are plotted representing amplitude vs. oscillation *period* T. Therefore, a quick "eye-balling" check of plausibility is made more difficult because the formula first needs to be transformed into a function of period (starting from the corrected version):

$$A(T) = \frac{T^2 T_0^2}{\sqrt{(T_0^2 - T^2)^2 + \beta T^2 T_0^4}} - \frac{T^4 T_0^4}{T^4 + T_0^4}.$$

This can of course be done "in the head" but requires more effort than simply reading the formula and comparing the expected shape of the curve with the graphical representation.

In fact, this is not only an issue of graphical representation but may well influence the interpretations of the model: Moelants / van Noorden 2005 [191], page 429, refer to the "typical long tail towards slow periodicities". This long tail, however, is an effect of representation due to the inverse relationship between frequency f and period T, $f = \frac{1}{T}$. Plotted as a function of frequency as in Figure 4.7, there will instead be a typical long tail towards high frequencies, i.e. "fast periodicities".

In Moelants / van Noorden 2005 [191], the formula for the effective resonance curve is given as follows:

$$A(f) = \frac{1}{\sqrt{(f_0^2 - f^2)^2 + \beta f^2}} - \frac{1}{\sqrt{f_0^4 - f^4}}.$$

It will be immediately obvious that the second term in this formula is problematic: for $f = f_0$ this term is not defined; for $f > f_0$, the term becomes purely imaginary. Plotting the resonance curve according to this formula – e.g. using the Matlab `fplot` command – leads to the curious display of Figure 4.9 and an error message stating that the imaginary part has been omitted in the plot.

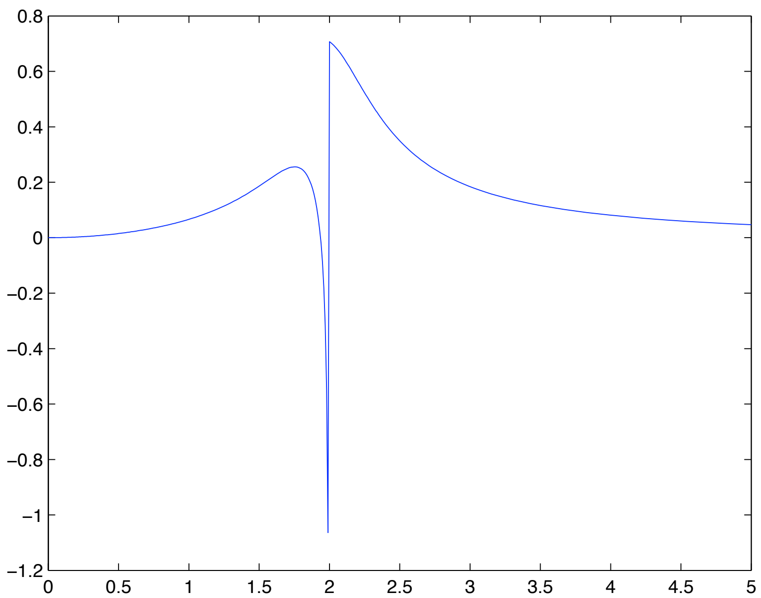


Figure 4.9: Resonance curve as a function of frequency plotted according to the formula given in Moelants / van Noorden 2005 [191].

Even worse than just replacing a plus sign by a minus sign, a completely mutilated version of the formula appears in McKinney / Moelants 2006 [185]:

$$A(f) = \frac{1}{\sqrt{(f_0 - f^2) + \beta f^2}} - \frac{1}{\sqrt{f_0^4 - f^4}},$$

i.e. in addition to the plus-minus exchange, in the first term at two places squares have been forgotten; the term under the square root should be $(f_0^2 - f^2)^2 + \beta f^2$.

Especially in a community usually not dedicated to the manipulation of mathematical formulae, more care should be given to the preparation of such expressions for printing. Otherwise, technical deficits of this kind will unnecessarily inhibit adequate understanding of the models proposed and thus stand in the way of fruitful scientific discussions.

4.3 Self-Sustained Oscillations: van der Pol Oscillators

The (concept of) the van der Pol oscillator was originally developed in the 1920s in the context of oscillatory electrical circuits. A characteristic aspect of the circuits studied is that they contain active elements such as voltage/current sources[8] which are set up in a way to maintain oscillations at a steady amplitude. Depending on the details of the circuitry, the shape of the produced oscillations can vary from nearly sinusoidal to so-called relaxation oscillations (see below).

The van der Pol oscillator has since gained importance as a "paradigmatic model of oscillation theory and nonlinear dynamics" (Pikovsky / Rosenblum / Kurth 2001 [225], page 6) – discussed extensively e.g. by Guckenheimer / Holmes 1983, Chapter 2.1 or Hirsch / Smale 1974, Chapter 10 – and as a model of systems exhibiting synchronizing behavior in technical, chemical, and biological contexts. As an example Pikovsky / Rosenblum / Kurth (2001 [225], Chapter 3.3.3) describe the work of van der Pol and van der Mark modeling the human heart by three coupled oscillators. In the 1960s activity of nerve cells was described with reference to the van der Pol oscillator by Fitzhugh and Nagumo (see e.g. Wang 1999 [299]), these considerations where integrated into artificial neural networks in the 1990s (e.g. Wang 1993 [298]). More recent discussions of synchronization in biochemical processes including as one possible mechanism the van der Pol oscillator is presented e.g by Nagano (2003 [195] et passim).

Here we will discuss the van der Pol oscillator as a mechanical system, taking up the description of the spring-mass pendulum presented in the preceding sections. Although no mechanism for the required active components will be proposed, we might speculate about biochemical processes underlying muscle contraction or nervous activity as sources of energy.

To keep the amplitude of oscillation constant, a system needs to achieve two complementary processes: If the amplitude exceeds the required value, it must be reduced; if the amplitude falls below the value required, it has to be increased. As we have seen in the discussion of the spring-mass pendulum, reduction or increase of oscillation amplitude can be influenced by the choice of the constant $\beta = \frac{r}{m}$ multiplying the velocity in the system's equation of motion: the sign of β indicates whether amplitude will be reduced (positive sign) or increase (negative); the absolute value of β will influence the rate of decrease / increase.

Starting from this observation, a way to construct a system description leading to the desired description of system behavior might be to replace the frictional

[8]We have not consulted the original papers of van der Pol and coworkers yet. For references to their work and example circuit schemes see Pikovsky / Rosenblum / Kurth 2001 [225] or Kanamaru 2007 [145].

constant r in the equation of motion

$$m\ddot{x}(t) + r\dot{x}(t) + kx = 0$$

by a function that is related to the amplitude of oscillation.

Since amplitude is a global – or at least non-local – characteristic of the oscillation pattern referring to an extended time span but only local values related to single values of t enter the equation of motion, amplitude will not be used directly. Instead, a function is chosen which is related to the absolute value of displacement, i.e. the absolute value of the variable $x(t)$, treating negative and positive values of $x(t)$ symmetrically.

The most simple function to achieve this is a function of the square of $x(t)$, i.e. the constant r will be replaced by a function $f(x(t)^2)$.

For this function to be positive when $x(t)$ exceeds a certain value $\hat{x}$ and to become negative when $x(t)$ falls below $\hat{x}$, it is sufficient to choose

$$f(x(t)^2) = r\varepsilon\left(x(t)^2 - \hat{x}^2\right),$$

where r and ε are non-negative constants.

Using this substitution, the modified equation of motion becomes

$$m\ddot{x}(t) + r\varepsilon\left(x(t)^2 - \hat{x}^2\right)\dot{x}(t) + kx(t) = 0,$$

which can be transformed into

$$\ddot{x}(t) + \frac{r\varepsilon}{m}\left(x(t)^2 - \hat{x}^2\right)\dot{x}(t) + \omega_0^2 x(t) = 0 \tag{4.30}$$

and further

$$\ddot{x}(t) + \frac{r\varepsilon}{m}\hat{x}^2\left(\frac{x(t)^2}{\hat{x}^2} - 1\right)\dot{x}(t) + \omega_0^2 x(t) = 0 \tag{4.31}$$

or

$$\ddot{x}(t) - \frac{r\varepsilon}{m}\hat{x}^2\left(1 - \frac{x(t)^2}{\hat{x}^2}\right)\dot{x}(t) + \omega_0^2 x(t) = 0. \tag{4.32}$$

Setting $\frac{r\varepsilon}{m} = 2\mu$ and $\frac{1}{\hat{x}^2} = \gamma$, the last equation turns out to be of the same form of the so called van der Pol equation as given by Pikovsky / Rosenblum / Kurth 2001 [225], page 177. (Slightly different formulations are presented resp. in Ott 2002 [211], page 11; Haken 1983 [108], page 132; Guckenheimer / Holmes 1983 [107], page 44 resp. 67-68; Kanamaru 2007 [145]; ....)

Inspecting equation (4.30), the constant β as defined by equation (4.10) in the context of the spring-mass pendulum is replaced by the function

$$\beta(x(t)) = \frac{r\varepsilon}{2m}\left(x(t)^2 - \hat{x}^2\right).$$

For $\varepsilon = 0$ this function vanishes and the system will exhibit the behavior of an undamped harmonic oscillator, i.e. perform sinusoidal oscillations with constant amplitude.

For very small positive values of ε, typically indicated as $0 < \varepsilon \ll 1$, the oscillation pattern will be expected to be similar to the case $\varepsilon = 0$. As discussed above, however, there will be regions in phase space for which $\beta(x(t))$ will be positive, leading to a decrease in amplitude roughly represented by a trajectory turning toward the origin, and other regions for which $\beta(x(t)) < 0$ leading to increasing amplitude and trajectories turning outward. To be a little more specific about these regions – although not in a formal manner – we will transform the second order differential equation into a coupled system of first order equations using the same substitution as in the case of the spring-mass pendulum specified by equation (4.5).

The system accordingly will be

$$\begin{aligned} \dot{x}(t) &= y(t) \\ \dot{y}(t) &= -\frac{r\varepsilon}{m}\left(x(t)^2 - \hat{x}^2\right) y(t) - \omega_0^2 x(t) \end{aligned}$$

and in vectorial notation

$$\begin{aligned} \dot{\vec{X}}(t) &= \begin{pmatrix} y(t) \\ -\frac{r\varepsilon}{m}\left(x(t)^2 - \hat{x}^2\right) y(t) - \omega_0^2 x(t) \end{pmatrix} \\ &= \begin{pmatrix} 0 & 1 \\ -\omega_0^2 & \frac{r\varepsilon}{m}\hat{x}^2 \end{pmatrix} \cdot \begin{pmatrix} x(t) \\ y(t) \end{pmatrix} - \frac{r\varepsilon}{m} \begin{pmatrix} 0 \\ x(t)^2 y(t) \end{pmatrix}. \end{aligned} \tag{4.33}$$

Due to the nonlinear term $x(t)^2 y(t)$ this equation can no longer be represented as a matrix multiplication with a constant matrix; another, but related, consequence of this nonlinearity is the fact that the techniques employed above to obtain a solution of the differential equation are no longer applicable. No general techniques to solve a nonlinear differential equation exist, therefore we will rely in the following on numerical schemes implemented in mathematical software packages such as Matlab.

To obtain some illustrating examples (see Figures 4.10, 4.12), trajectories were plotted for different values of the combination of constants $\gamma = \frac{r\varepsilon}{m}$ and the initial states $\begin{pmatrix} 0.1 \\ 0.2 \end{pmatrix}$ (blue curves) resp. $\begin{pmatrix} 3 \\ 3 \end{pmatrix}$ (red curves), again retaining the value $\omega_0 = 1.4$. For the numerical procedure using Matlab the (system of) differential equation(s) was coded as

```
dy(1) = y(2);
dy(2) = -gamma*(y(1)^2-1)*y(2) - 1.4*y(1);
```

`gamma` was set to 0.3 resp. 5.

The case $\gamma = 0.3$ is displayed in Figure 4.10. The beginning parts of the trajectories can clearly be distinguished: The solid trajectory starting from state $\begin{pmatrix} 0.1 \\ 0.2 \end{pmatrix}$ begins spiraling outward, but the successive turns of the spiral get closer and closer so that the trajectory remains confined to a bounded area and eventually appears to enter a closed curve.

In a complementary way, the dash-dotted trajectory starts spiraling inward but again successive turns get successively closer and the inward movement appears to be bounded by the same closed trajectory bounding the outward movement of the blue trajectory.

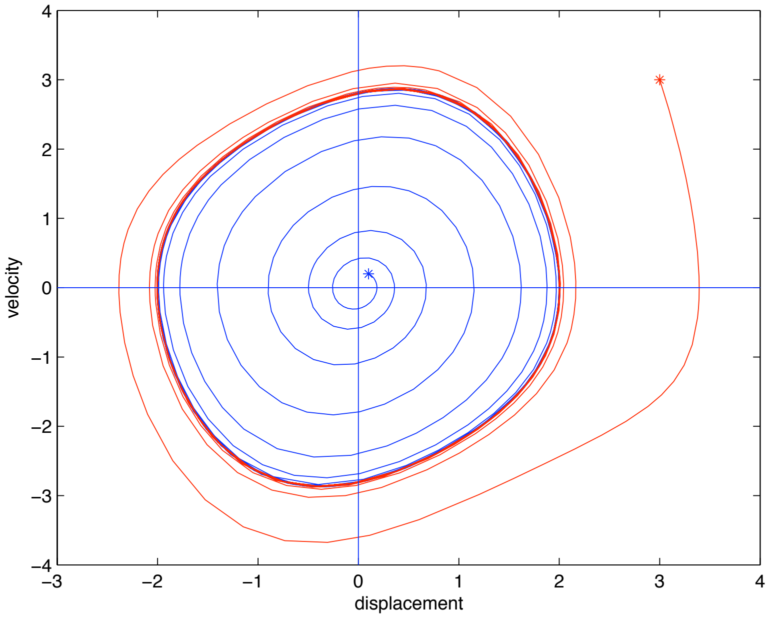


Figure 4.10: Two trajectories of the van der Pol oscillator for $\gamma = 0.3$.

It can be shown by detailed mathematical analysis (see Guckenheimer / Holmes 1983 [107], in particular Chapter 2.1. for details and further references) that for a system described by the van der Pol equation, a closed trajectory exists and that any trajectory describing system behavior will eventually come arbitrarily near to this closed trajectory.

Such a trajectory, which can only be observed in nonlinear systems, is called an *attracting limit cycle*.

Since the trajectories approach a closed trajectory we will expect a periodic system behavior to arise after some time, and because the shape of the limit cycle is nearly circular, the oscillations will be expected to be nearly sinusoidal.

To test these expectations, the temporal behavior of the system has been plotted in Figure 4.11 for both initial states.

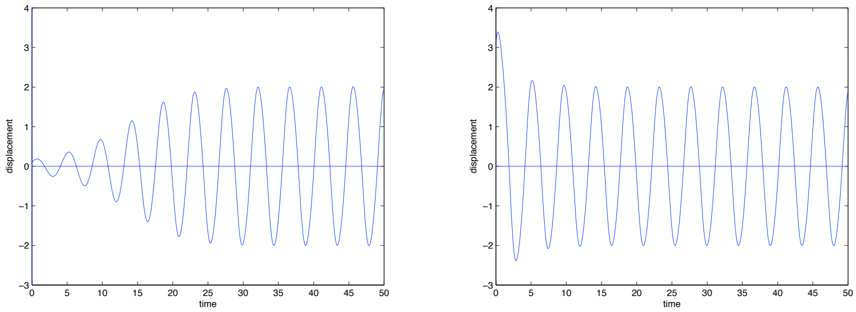


Figure 4.11: Oscillations of van der Pol oscillator with $\gamma = 0.3$.

In the left panel, corresponding to the initial state $\begin{pmatrix} 0.1 \\ 0.2 \end{pmatrix}$, an oscillation pattern is observed that starts with a low amplitude but in the course of seven oscillation cycles builds up to a steady amplitude of about 2 (arbitrary units). In the right panel corresponding to initial state $\begin{pmatrix} 3 \\ 3 \end{pmatrix}$, a higher amplitude at the beginning is reduced to the same amplitude of about 2 in the course of 2 oscillation cycles. In both panels the same number of oscillations in the same time span is displayed, i.e. for both cases the oscillation frequency is identical. The shape of the oscillations seems to be nearly sinusoidal as expected.

Although not formally shown, we can thus qualitatively observe the desired system behavior specified in the beginning of this section.

For the case $\gamma = 5$, the deviations from the undamped harmonic oscillator will be substantially greater. Therefore we can no longer expect the same shape of trajectories and oscillation patterns. Some qualitative properties, nevertheless, remain unchanged, as illustrated by Figure 4.12.

Again the trajectories have been plotted for initial conditions $\begin{pmatrix} 0.1 \\ 0.2 \end{pmatrix}$ and $\begin{pmatrix} 3 \\ 3 \end{pmatrix}$ for the same time span as in Figure 4.10. Even in this case it can be noticed that the trajectories seem to approach a closed curve after some time, i.e. even for $\gamma = 5$, the system behavior is characterized by an attracting limit cycle. Although the shape of this closed trajectory is quite different from the nearly circular shape shown in Figure 4.10, it is again indicative of periodic system behavior.

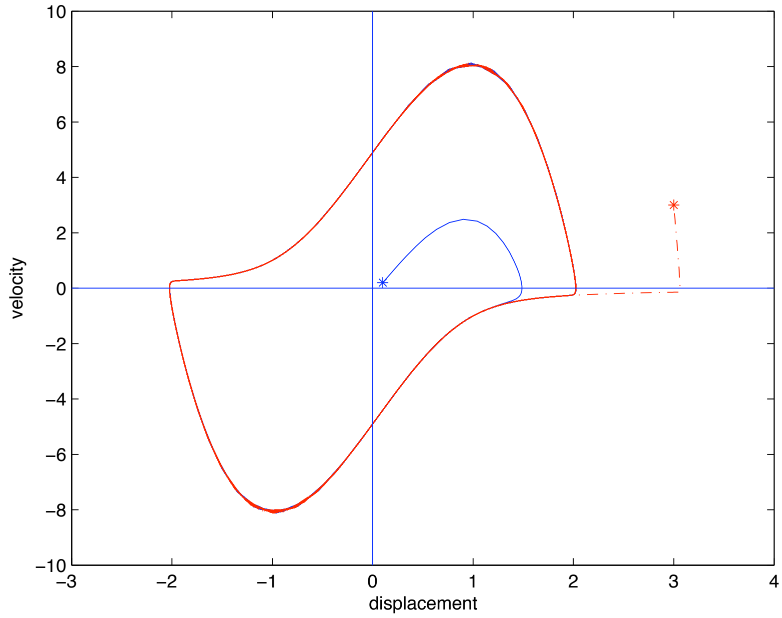


Figure 4.12: Two trajectories for $\gamma = 5$.

For both initial conditions oscillation patterns are displayed in Figure 4.13. For initial state $\begin{pmatrix} 0.1 \\ 0.2 \end{pmatrix}$, again a build-up of amplitude to the value 2 is observed (left panel), whereas for initial state $\begin{pmatrix} 3 \\ 3 \end{pmatrix}$, again amplitude decreases from an initially higher value. Even the oscillation period is again the same for both initial conditions once the beginning section is over; for $\gamma = 5$ the period of oscillation is somewhat higher than for $\gamma = 0.3$.

Although still highly symmetrical – reflecting the symmetrical shape of the limit cycle – the shape of the oscillation pattern has changed dramatically as compared to Figure 4.11: The previously smooth curve is acquiring sharp corners and the gradual change of slope is replaced by an alternation of sections with steep rise or fall and sections with more “gentle” slopes. Such an oscillation pattern is called *relaxation oscillation*. The relaxation characteristics (sharp corners, alternating steep and gentle slopes) will become more pronounced with increasing values of γ and the oscillation period will increase; the amplitude, however, will remain unchanged as illustrated in Figure 4.14 for the values $\gamma = 10$ (left panel) and $\gamma = 20$ (right).

Since the amplitudes of oscillation of the van der Pol oscillator diminish or increase until some stable value is reached, the van der Pol oscillator is said to

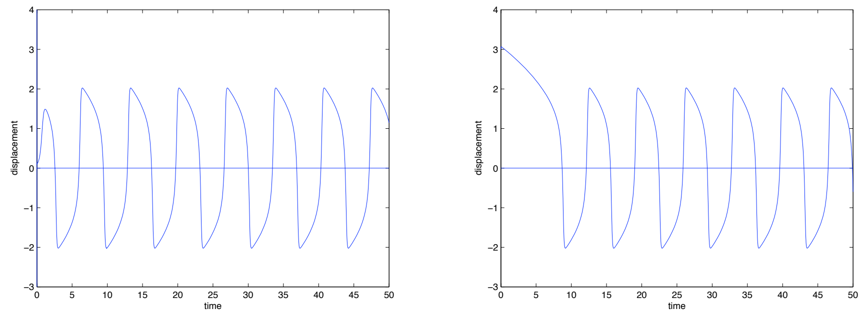


Figure 4.13: Oscillations for $\gamma = 5$.

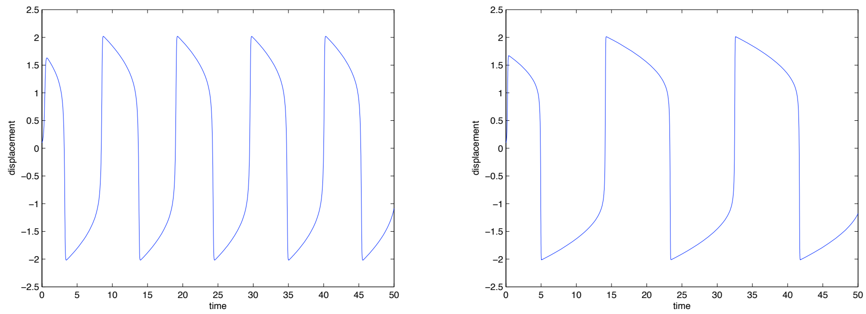


Figure 4.14: Oscillations for $\gamma = 10$ and $\gamma = 20$.

exhibit *self-sustained oscillations*. Self-sustained oscillations play a crucial role in the discussion of synchronization phenomena, as e.g. presented in Pikovsky / Rosenblum / Kurth 2001 [225]. As these are intimately related to the existence of attracting limit cycles, which can only arise in nonlinear systems, systems exhibiting synchronizing behavior must necessarily be nonlinear (as already pointed out by Wiener 1961 [317], Chapter X).

Figure 4.15 has been included to illustrate the fact that any initial conditions of the van der Pol oscillator (here again with $\gamma = 5$) will give rise to trajectories that are attracted to the limit cycle.

The remarks offered here remain rather descriptive: Even for such a simple system as the van der Pol oscillator, the task to show the existence of an (attracting) limit cycle involves a formidable – at least for the "non-initiated" – amount of mathematical preparation because general properties of two-dimensional systems of differential equations are required; even harder according to the relevant textbook sources (e.g. Guckenheimer / Holmes 1983 [107], page 44) will be the task to establish the uniqueness of the limit cycle and the global property of attraction

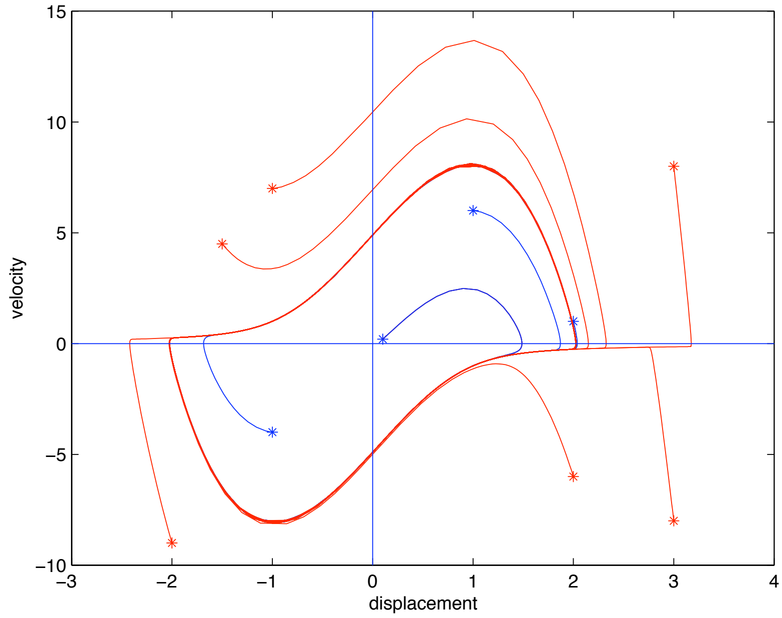


Figure 4.15: Some trajectories to illustrate attraction towards the limit cycle for different initial states and $\gamma = 5$.

– with the exception of the unstable equilibrium state $\begin{pmatrix} 0 \\ 0 \end{pmatrix}$.

Besides providing a more principled reason why resonance of a linear system such as the harmonic oscillator will not suffice as an explanation for synchronizing behavior[9], with the attracting limit cycle we have seen an example of the more general phenomenon known as an *attractor*.

According to Norton (1995 [207], page 56), "there is no general agreement on the precise definition of an attractor, but the basic idea is straightforward". He suggests the following way of characterizing an attractor:

- An attractor **A** is a closed (proper) subset of phase space.
- All initial states (points in phase space) sufficiently close to **A** correspond to trajectories approaching **A** as time t increases.

[9]But note that this is just a negative criterion: By making use of the general scheme illustrated above to obtain solutions for linear systems of differential equations, it can be shown that closed trajectories can only be neutrally stable but cannot exhibit the attracting property of a limit cycle – but this does not establish the existence of limit cycles in any specific nonlinear system.

- Trajectories corresponding to initial states that lie in (are members of) **A** will be confined to **A**.
- There are no proper subsets of **A** for which the last two conditions hold.

No general criterion of "sufficiently close" is offered; all points in phase space fulfilling this criterion, i.e. all points in phase space lying on trajectories eventually approaching an attractor are said to form the *basin of attraction* of that attractor.

As stable equilibrium points are complemented by unstable equilibrium points, attracting limit cycles are complemented by repelling limit cycles resp. attractors by *repellors*. A detailed geometrical – graphical – presentation of these ideas is offered in Abraham / Shaw 1982 [1].

Some final remarks are in place regarding yet another use of the concept of resonance in the domain of embodied music cognition: Leman 2008 [172], Chapter 3.5, discusses "Culture as Resonance System". Although the concept of resonance in this case is introduced as "a powerful metaphor" (ibid., page 69), some care needs to be taken for such a metaphorical use to be fruitful and to insure explanatory value.

The situation encountered in this context is in a way reverse to that encountered in cognitive linguistics (see Section 3.1.2): Whereas in the work documented e.g. in Lakoff / Johnson 1980 [164], 1999 [165], Lakoff / Núñez 2000 [166], Núñez 2004 [208] established metaphors are investigated in order to analyze the metaphorical mapping from a source domain to a target domain and to explicate the way the conceptual structure of the source domain influences the conceptualization of the target domain, the goal here is to establish a mapping from a source domain with a (in principle) well known conceptual structure to provide a scientific description (and explanation) of phenomena observed in the target domain.

Following a discussion by Hempel 1977 [115], Chapter 6, we would prefer to describe this situation as an instance of the use of *analogy* common e.g. in physics: Formal similarities observed in different domains are take as a justification to utilize results obtained within one (source) domain to find a description of the other (target) domain by careful re-interpretation of physical laws, for instance in the well-known electro-mechanical analogies. The use of analogy in this way can be beneficial in at least two ways: work can be saved by transferring formal structure to the target domain, and consequences derived from the formal description can serve as a stimulation for further investigation of the target domain. As Hempel points out (ibid., page 160), however, care needs to be taken because stating an analogy between the domains requires that the laws obtaining in the target domain be known; in that sense, the use of analogy does not provide additional knowledge of the target domain.

The source domain addressed by Leman is a mathematically inclined physics as evidenced by the use of terms such as energy, resonance (physics) and attractor, focus point (theory of dynamical systems). The indiscriminate combination of these concepts, however, indicates a still insufficient conceptual analysis of the source domain (e.g. placing concepts with a strong connotation of linear systems theory – resonance – side by side with concepts mainly associated with nonlinear theory of dynamical systems – attractor); an example of the consequences of a lacking analysis of the conceptual structure of the source domain has been discussed with regard to the resonance model of van Noorden and Moelants.

The scheme suggested by Leman appears to be an excellent example of what was called metaphorical (maybe better: analogical) inspiration above. The ideas should be pursued with a detailed analysis taking into account the technicalities of the source domain to develop the full descriptive and explanatory potential of the approach.

4.4 Dynamical Analysis of Gestures

As a final introductory example of a dynamical system, we will turn towards a approach presented by Wenger / Copeland / Schuster 2007 [315]. We will not do justice to their work, however: although explicitly acknowledging the importance of the questions taken up, we will leave out the discussion concerning statistical testing of hypotheses related to the model proposed, which is a core concern of their article. Instead, we will concentrate on a presentation of their system description; the formulation offered here is slightly modified to accord with the preceding sections.

The example is included for the following reasons:

- The system description given employs a formalism similar to that discussed in the preceding sections. In this case, however, it does not involve any low level physical processes. Rather, the task of magnitude estimation to be analyzed is situated within psychophysics, as already announced in the title *Gestures as Psychophysical Judgements*.
- From a psychophysical perspective, the approach seems to take up ideas developed by Lakoff / Johnson 1980, 1999 [164, 165] and Lakoff / Núñez 2000 [166] relating the conceptualization of space to corporeal movements.
- An explicit goal of the approach involves the development of a methodology that allows to take intrinsic temporal relations of observed psychological processes into consideration (see Wenger / Copeland / Schuster 2007 [315], page 246). We may speculate that this approach – including the statistical considerations omitted here – might add perspective to the analysis of (musical) gestures as discussed e.g. by Jensenius 2006 [133].
- Explicitly taking into account input to a dynamical system, the example serves as an introduction to considerations taken up in the next chapter (Chapter 5.1).

The following task was analyzed in Wenger / Copeland / Schuster 2007 [315]: Subjects were asked to estimate the vertical size of a pair of eyes and the horizontal size of the mouth of a schematic drawing of a face (see Figure 4.16, left panel). Stimuli with varying sizes for both features were presented on a computer screen. Responses were generated by moving the cursor in a two dimensional display showing a pair of axes. The horizontal axis corresponded to the (horizontal) size of the mouth, the vertical axis to the size of the eyes; thus, any point in the plane represented a simultaneous judgement of both features considered. Not only the final cursor position was recorded but the complete movement from an initial position at the origin of the coordinate system.

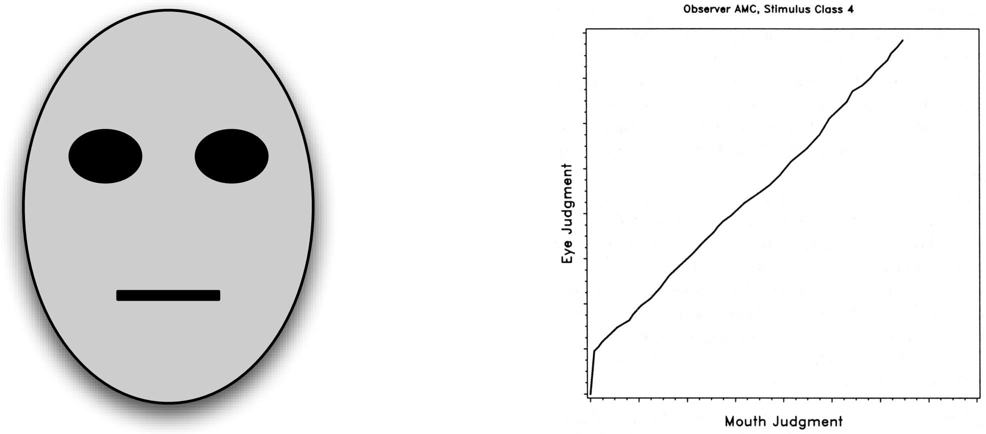


Figure 4.16: Left: Schematic drawing of a face presented as a stimulus for the judgement of facial attributes, redrawn after Wenger / Copeland / Schuster 2007 [315], page 261.
Right: Example trajectory showing the judgement of one stimulus, from Wenger / Copeland / Schuster 2007 [315], page 265.

The trajectories gathered in this way were analyzed as outputs of a dynamical system described as follows:

The system is assumed to comprise a number of processing channels described by a set of variables denoted by the vector $\vec{X}(t)$. The temporal development of the system is accordingly described by a (set of) differential equation(s) of these variables. In addition, a set of inputs influencing the system is taken up in the system description, here denoted as $\vec{V}$.

To capture the influence of the inputs on system development, the vector function $\vec{F}$ specifying the first derivatives of the variables must be a function of both the variables and the inputs, i.e. the differential equation describing the system must be of the form

$$\dot{\vec{X}}(t) = \vec{F}(\vec{X}(t), \vec{V}(t)). \tag{4.34}$$

The output of the system, in this case taken to be a trajectory produced as a response to stimulus presentation and denoted by the vector $\vec{Y}(t)$, is also considered to be described as a function of the variables and the inputs:

$$\vec{Y}(t) = \vec{O}(\vec{X}(t), \vec{V}(t)). \tag{4.35}$$

The temporal development of the variables is taken to be independent of the outputs, i.e. the inputs to the system are considered not to be changed by system outputs, reflecting the experimental setup.

The number of channels assumed was two, one for processing each of the facial features. Likewise, the input was assumed to consist of the coding of the two physical measures to be judged. Therefore, both $\vec{X}(t)$ and $\vec{V}(t)$ were taken to be two-dimensional vectors.

The question to be investigated was whether there is any cross-channel interaction in the system[10], i.e. whether eye size influences the judgement of mouth size and vice versa.

To develop hypotheses that could be tested, further assumptions concerning the system (description) were introduced:

Both $\vec{F}(\vec{X}(t), \vec{V}(t))$ and $\vec{O}(\vec{X}(t), \vec{V}(t))$ were assumed to consist of two parts depending only on the variables resp. the inputs, i.e.

$$\begin{aligned}\vec{F}(\vec{X}(t), \vec{V}(t)) &= \vec{A}(\vec{X}(t)) + \vec{B}(\vec{V}(t)) \quad \text{and} \\ \vec{O}(\vec{X}(t), \vec{V}(t)) &= \vec{A}(\vec{X}(t)) + \vec{B}(\vec{V}(t)).\end{aligned}$$

The functions $\vec{A}(\cdot)$, $\vec{B}(\cdot)$, and $\vec{C}(\cdot)$ were further assumed to constitute matrix multiplications with constant matrices

$$\mathbf{A} = \begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} b_1 & b_2 \\ b_3 & b_4 \end{pmatrix}, \quad \mathbf{C} = \begin{pmatrix} c_1 & c_2 \\ c_3 & c_4 \end{pmatrix},$$

and $\vec{D}(\cdot)$ was assumed to be the constant function

$$\vec{D}(\vec{V}(t)) = \vec{\varepsilon} = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \end{pmatrix}.$$

With these assumptions, the differential equation describing system development will be of the form

$$\dot{\vec{X}}(t) = \begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix} \cdot \vec{X}(t) + \begin{pmatrix} b_1 & b_2 \\ b_3 & b_4 \end{pmatrix} \cdot \vec{V}(t). \tag{4.36}$$

Here, the elements of $\mathbf{A}$ can be interpreted as rates of change of the variables in the absence of input; the diagonal elements a_1 and a_4 of the matrix relate to within-channel processes, the off-diagonal elements a_2 and a_3 describe cross-channel interactions (coupling of the variables). The elements of $\mathbf{B}$ describe the strength of influence of the inputs on the rate of change; again the diagonal elements (b_1, b_4) relate to within-channel processes and the off-diagonal elements (b_2, b_3) to cross-channel interactions.

The output according to the assumptions made is

$$\vec{Y}(t) = \vec{X}(t) + \vec{\varepsilon}.$$

[10]or, in the terminology used above, whether there is any coupling between the variables

Given such a system description and a constant input $\vec{V}_0$ as characteristic for the single trial situations, we can easily find an equilibrium state $\vec{X}_{eq}$ for this situation: setting $\dot{\vec{X}}(t) = \vec{0}$, equation (4.36) will become

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix} \cdot \vec{X}_{eq} + \begin{pmatrix} b_1 & b_2 \\ b_3 & b_4 \end{pmatrix} \cdot \vec{V}_0$$

and therefore

$$\begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix} \cdot \vec{X}_{eq} = - \begin{pmatrix} b_1 & b_2 \\ b_3 & b_4 \end{pmatrix} \cdot \vec{V}_0,$$

which can be solved, provided $a_1 a_4 - a_2 a_3 \neq 0$ to yield the constant vector[11]

$$\vec{X}_{eq} = -\frac{1}{a_1 a_4 - a_2 a_3} \begin{pmatrix} a_4 & -a_2 \\ -a_3 & a_1 \end{pmatrix} \cdot \begin{pmatrix} b_1 & b_2 \\ b_3 & b_4 \end{pmatrix} \cdot \vec{V}_0.$$

For the experimental situation described, the state of the system will be expected to approach this equilibrium state with increasing values of t, and as a consequence, the output of the system, too, will approach a constant state

$$\vec{Y}_{eq} = \vec{X}_{eq} + \vec{\varepsilon}.$$

An output trajectory exhibiting this kind of behavior is shown in Figure 4.16, right panel.

In the system description of equation (4.36), the matrices **A** and **B** represent independent possible sources of cross-channel interaction, depending on the values of the off-diagonal elements. In their statistical evaluation, Wenger / Copeland / Schuster 2007 [315] compared the fit of the collected trajectories to four different scenarios:

1. Off-diagonal elements of **A** and **B** are zero, i.e. there is no cross-channel interaction in the system.

2. off-diagonal elements of **A** are zero and off-diagonal elements of **B** are non-zero. This is interpreted as an example of early interaction because incoming data is distributed to the processing channels without prior processing.

3. Off-diagonal elements of **A** are non-zero and off-diagonal elements of **B** are zero. This is interpreted as late interaction because cross-channel influence of the inputs is mediated by the changing of the variables.

4. The last scenario with non-zero off-diagonal elements for both **A** and **B** is interpreted as constituting both early and late interaction.

In the experiment reported, data conformed best to a system description according with the first scenario.

[11]The matrix $\frac{1}{a_1 a_4 - a_2 a_3} \begin{pmatrix} a_4 & -a_2 \\ -a_3 & a_1 \end{pmatrix}$ can be shown to be the inverse of matrix **A**.

Chapter 5

Dynamic Systems: A Framework for CSM?

5.1 Agents as Dynamical Systems

The theory of dynamical systems has repeatedly been proposed as a theoretical framework to describe the interactions between an agent and its environment (e.g. Arbib 1972 [10]; Beer 1995a,b [27, 28]; Beer 2003 [30]; Steels 1996 [261]; van Gelder 1997 [290]; Pfeifer / Scheier 1999 [220]; Pfeifer / Bongard 2006 [218]); related ideas can be traced back at least to the work of Ashby in the 1950s (van Gelder 1997 [290], Section 6.11; Boden 2006a [37] chapter 4.viii.c; Beer 1995a [27], page 181). We will take up and slightly extend the semi-formal formulation of Beer 1995a,b [27, 28], conforming to the presentation of dynamical systems given in Chapter 4, and try to relate the notions of embodiment discussed above to this framework.

Following Ashby, Beer distinguishes between agent and environment as two mutually coupled dynamical systems (see figure 5.1). The agent is referred to with the symbol $\mathcal{A}$, the environment with $\mathcal{E}$. Agent and environment are essentially treated symmetrically, the main focus, however, rests in accordance with the discussion presented in Section 3.2 on the agent.

Whereas in Figure 5.1, no regard is taken to the internal organization of the agent, Beer 2003 [30] differentiates further, separating the agent's nervous system and body (see Figure 5.2). This may be appropriate for the applications described, evolving a control architecture for a given agent morphology. However, in general we would advocate a more integrative view allowing for various sub-systems mutually influencing each other, in order to avoid unnecessary initial commitments regarding influences to be taken into account (see the questions posed at the end of Section 3.2). The illustration given in Figure 5.3 was inspired by Figure 4/15/1 in Ashby 1960 [18].

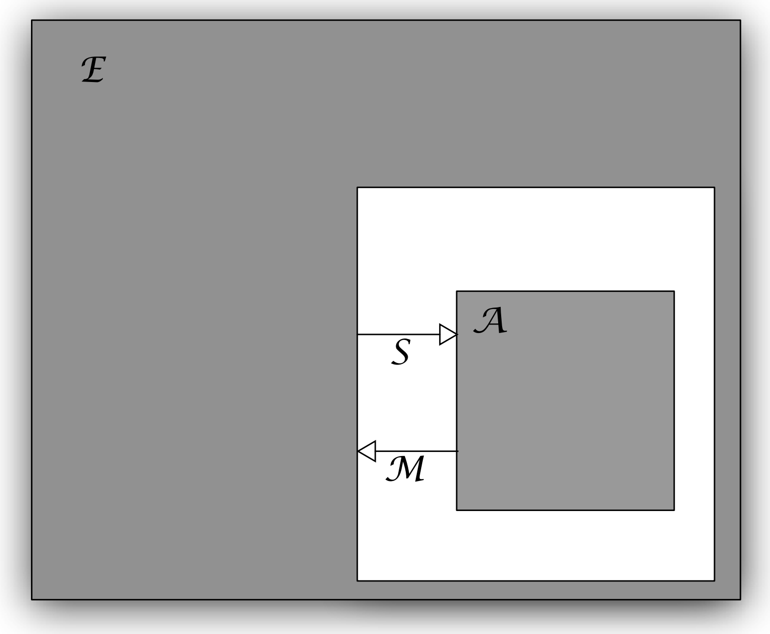


Figure 5.1: Coupling of agent and environment, redrawn after Beer 1995a [27], see also Arbib 1972 [10], page 58

In accordance with the general way of describing a dynamical system introduced in the previous chapter, the present state of the agent is assumed to be decribed by a set of time-dependent variables, in vectorial notation given as $\vec{x}_{\mathcal{A}}$; in the same manner, the environment is described by a set of variables $\vec{x}_{\mathcal{E}}$.

In addition, as already mentioned on Page 48, Beer introduces as a second set of more stable quantities to characterize both agent and environment, respectively, the parameters denoted by $\vec{u}_{\mathcal{A}}$ for the agent, and $\vec{u}_{\mathcal{E}}$ for the environment. These may refer to material properties of (parts of) an agent such as density, elasticity, number of neurons / synapses, and are intended to describe persistent structural properties of agent and environment. As argued below, however, a clear-cut distinction between state variables and parameters will be difficult to draw.

The coupling of agent and environment is introduced into Beer's scheme by two functions describing the mutual influences: the "motor" function $M(\vec{x}_{\mathcal{A}})$ takes as arguments the agent's state variables to specify the influence exerted on the environment by the agent, the "sensory" function $S(\vec{x}_{\mathcal{E}})$ captures the influences on the agent produced by environmental conditions.

The time course of agent and environment states (i.e. agent and environment behavior) is formally described by two coupled sets of coupled differential equations,

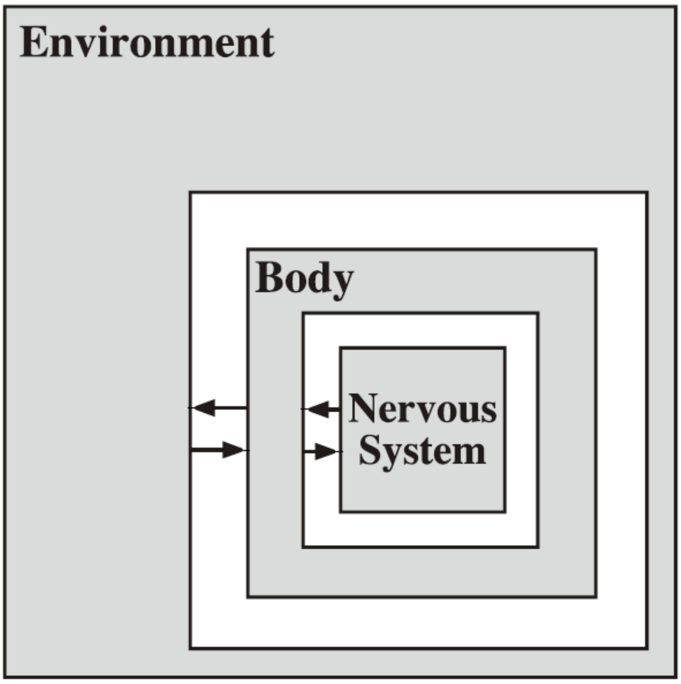


Figure 5.2: Coupling of agent body / nervous system and environment, from Beer 2003 [30]

the equations of motion of agent and environment, resp.:

$$\begin{aligned} \dot{\vec{x}}_{\mathcal{A}} &= \mathcal{A}(\vec{x}_{\mathcal{A}}; S(\vec{x}_{\mathcal{E}}); \vec{u}_{\mathcal{A}}) \\ \dot{\vec{x}}_{\mathcal{E}} &= \mathcal{E}(\vec{x}_{\mathcal{E}}; M(\vec{x}_{\mathcal{A}}); \vec{u}_{\mathcal{E}}). \end{aligned}$$

To allow for a better comparison of this scheme with the notion of the hysteretical agent described in Section 3.3.2, we will introduce the following sets:

$X_{\mathcal{A}}$: The set of possible states of the agent or its state space, corresponding to the set of internal states **I** of the hysteretic agent.

$X_{\mathcal{E}}$: The set of possible states of the environment that corresponds to the set of external states **S** of the hysteretic agent.

$S(X_{\mathcal{E}})$: The image of the set of external states under the function S will correspond to the set **T** induced on the set **S** of the hysteretic agent by the function `see`; again due to the nature of the sensory function, different states in $X_{\mathcal{E}}$ may be mapped onto the same value of the function.

$M(X_{\mathcal{A}})$: The image of the set of agent states under the motor function represents the set of possible actions to be performed by the agent in its environment, corresponding to the set **A** of the hysteretic agent.

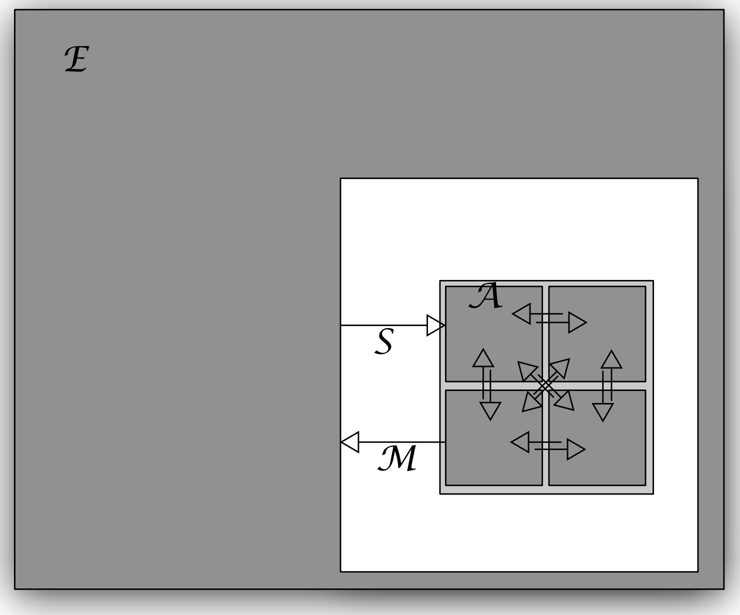


Figure 5.3: Mutual coupling of various within-agent sub-systems, inspired by Ashby 1960 [18], Figure 4/15/1

Although the function M as given here does not contain any *explicit* dependency on the elements of $X_{\mathcal{E}}$, in contrast to the function action of the hysteretic agent being defined over $\mathbf{I} \times \mathbf{T}$, there is nevertheless an implicit dependency mediated by the agent's equation of motion.

Furthermore, the agent's equation of motion describes the change of internal state of the agent, replacing the function `internal` of the hysteretic agent; the environment's equation of motion replaces the function `do` with even stronger formal similarity. Thus, although there are some technical differences between the two descriptions due to the choice of formalism, the notion of the hysteretic agent as defined by Genesereth / Nilsson 1986 [93] and the scheme of agent-environment coupling described in terms of dynamical systems theory by Beer 1995a [27] seem to be quite compatible.

As presented, Beer's scheme is well adapted to the example application of developing an insect-like walking agent: the parameters are partly fixed by the design of the material body, partly determined as free parameters in an artificial neural network used to control leg movements. Adjustment of the network parameters is achieved by genetic algorithm search in a process external to the dynamics of agent-environment interaction as described by the equations of motion. Thus, the agent's description does not seem to allow for any persistent changes due to interaction or for a history in terms of the notions of embodiment.

To incorporate an agent's history, i.e. to open up the scheme for the description of adaptive properties, at least some of the parameters need to be subject to temporal change instead of being considered as constants. This change will be mediated by the state variables $\vec{x}_{\mathcal{A}}$ and partly by the values of the other parameters, i.e. $\vec{u}_{\mathcal{A}} = \vec{u}_{\mathcal{A}}(\vec{x}_{\mathcal{A}}, \vec{u}_{\mathcal{A}})$. The time change of the parameters should be captured by the dynamical description of the agent, too, so that the agent's equation of motion is modified to

$$(\dot{\vec{x}}_{\mathcal{A}}, \dot{\vec{u}}_{\mathcal{A}}) = \mathcal{A}(\vec{x}_{\mathcal{A}}; S(\vec{x}_{\mathcal{E}}); \vec{u}_{\mathcal{A}}).$$

Thus, from a formal point of view, there will be no distinction between state variables and parameters, and it turns out to be a matter of perspective and convenience which characterizing quantities are regarded as state variables or parameters.

One criterion at hand may be the time scale of change. In the context of neuroscience, Arbib (1972 [10], page 66) proposes a separation according to the time scales of short term vs. long term memory, taking instantaneous nervous activity as representing state, more slowly changing synaptic strengths as an example of system parameters. A quite different range of time scales is discussed by Pfeifer / Bongard (2006 [218], Chapter 3.5): "here-and-now" refers to an agent's short-term behavioral mechanisms, the ontogenetic time scale to development and learning within an individual agent's lifetime, and even the evolutionary (phylogenetic) perspective is taken into consideration.

These remarks illustrate that the notions of embodiment found in the discussion of embodied cognition / embodied cognitive science (see Section 3.2) can be accommodated within the framework of dynamical systems theory or the discussions around it. This can be taken as an indication that the approach of embodied cognitive science does not raise any new foundational issues for cognitive science beyond those hiding behind the contrasting definitions of the field, which concern among others the interpretation of the term *computation* (see e.g. [99]) and the applicability of the theory of dynamical systems wearing a deterministic hat (Prigogine 1997 [229]).

Important contributions, however, can be expected in the following ways:

- Attention is drawn to aspects of cognitive processes that were previously neglected. This leads to new ways of understanding observed phenomena within the "old" framework, i.e. the theoretical foundation is explored more thoroughly and more fully.

- This is supported and facilitated by new theoretical and technological means, in turn giving rise to new developments.

- Of special interest to the realm of music (and other fields which are similarly considered to be cultural phenomena), interaction is taken up not just as an interesting artistic feat, but as a fundamental condition.
- More pointedly, taking into account developmental time scales and the mutual influence of different agents, the domain of musical behavior and experience along these lines will be construed as a process of continual cultural autoconfiguration.
- In the design of interactive artifacts, theoretical work can be integrated and put to a test under increasingly realistic conditions. New problems, however, arise in evaluating the performance of these artifacts.

5.2 The Turing Machine as a Dynamic System. Some Naïve Speculations

The speculations offered in this section are motivated on the one hand by claims within cognitive science that dynamical systems are not computational (see Thagard's Challenge 6 quoted in the beginning of Chapter 3), on the other hand by the identification of computation with the operation(s) of Turing Machines (TMs). One main proponent of the incompatibility of a dynamic systems approach and a computational approach to cognitive science is van Gelder (e.g. van Gelder 1997 [290])[1], similar opinions are expressed by Thelen: "[. . .] developmental data are compelling in support of these new anticomputational views" (Thelen 1995 [278], page 76).

More recently, Trevarthen 2002 [287] disputed the benefit of modeling approaches based on robotic or other computational devices for the investigation of (human) behavior and experience on the ground that living systems "are complex dynamic systems" (ibid., page 1 of the pdf version) exhibiting a set of properties that may never be captured by computational machines.

One source of dissatisfaction with the TM approach to computation is made explicit by Wegner and Goldin. In a row of papers (e.g. Goldin / Wegner 2002 [98], 2005 [99]; Wegner / Goldin 2003 [306]), they develop the idea that because for a TM to arrive at some result the input must be completely specified at the beginning of the computational process, the TM cannot serve as a sufficient framework to encompass interactive computational processes commonly encountered in interactive applications. They argue, therefore, that an alternative paradigm of computation – interactive computation – is required to extend the computational power of TMs.

[1] For a critical discussion of van Gelder's position see Glymour 1997 [95].

Construing interactive processes within the theory of dynamic systems as suggested in the previous section would again indicate that a traditional view of computation is inappropriate for a dynamical systems approach to cognitive science.

We are not in a position to discuss the computational power of either TMs or alternative paradigms of computation. Instead, we will take a look at processes underlying the operation of TMs, trying to give some preliminary remarks on their relation to cognitive processes.

This attempt may appear to take up the position formulated by Beer (1995a,b [27, 28]; 2003 [30]) that TMs are examples of dynamic systems. However, as mentioned above (page 47) Beer takes the term system to refer to the mathematical description, whereas we again will try to stick to the interpretation referring to the entity to be described.

In a typical textbook context, a TM is defined as a formal mathematical structure. As an example, Partee / ter Meulen / Wall (1990 [212], page 510) define a TM as a quadruple $\langle K, \Sigma, s, \delta \rangle$ where

K is a finite set of states of the machine,

Σ is a finite set of symbols forming the alphabet read by the machine,

$s \in K$ is a special element of K, the initial state of the machine, and

δ is a function $\delta: \; K \times \Sigma \to K \times (\Sigma \cup \{L, R\})$ specifying a new state of the machine in combination with an action to be taken when a specific symbol is observed.

The function δ can be given by a list of rules consisting of an initial state, observed symbol, new state, and symbol to write resp. movement to perform (L, R: left or right movement); the form employed by Partee / ter Meulen / Wall is

$$(q_i, a_j) \to (q_k, X_l)$$

where $q_i, q_k \in K$ are two possibly identical states, $a_j \in \Sigma$ is a symbol observed by the machine and $X_l \in \Sigma \cup \{L, R\}$ represents one of the actions of writing a symbol of the alphabet, moving left, or moving right.

As this presentation illustrates, even though the definition is intended to specify a mathematical formalism, it is not easy to formulate without reference to quite concrete physical operations or processes.

Although these processes can hardly be omitted from the definition / specification of a TM, they are not explicitly taken up for discussion – at least in textbook contexts; discussions instead are mainly concerned with manipulations of symbol strings that can be achieved once a specific TM is set up.

The processes involved in a computation following a TM description rely on the following components:

symbols: In the context of an operating computational device, symbols need to be some sort of physical configurations; they will be considered external to the system as long as different computational processes can be instigated by providing different combinations of symbols without changing the setup of the TM.

reading: To read a symbol, the internal processes of the system performing the computation need to be affected in some way by encountering the physical configuration of its environment formed by the symbol.

recognition: Recognition of the symbol seems to entail reproducible effects upon encountering specific symbol configurations.

decision: The decision what action to perform and what state to enter according to the TM rules entails that the temporal development of the system given a momentary state is determined by the influence of the external encountered symbol configuration.

writing: By writing a symbol, the system brings about a change of the environmental physical configuration.

Thus, in order to perform a computation, an adequate configuration of relevant physical components is required. Thereby, results are produced as an outcome of a chain of processes which are determined lawfully by the encounter of external physical configurations and by internal state of the system.

In standard computing machinery, components are carefully designed drawing on expert (electrical engineering) knowledge. In particular, input devices are set up to provide a set of clearly distinguishable physical configurations (such as key strokes) to influence processes going on within the system.

For most practical purposes, physical aspects of those processes can be disregarded. For instance, a programmer does not need to worry about electrical current flowing in the computer; attention can be focused to a restricted set of the system's variables.

In a way, the use of up-to-date technology masks the problems inherent in the basic processes required. This may be illustrated by taking up Turing's idea of a human computer performing the operations required in a computation (e.g. Turing 1950 [289]; more vividly described by Feynman 1996 [86]): The human's abilities to read, recognize, and write the involved symbols is taken for granted – but needs to be contrasted with the difficulties of building machinery e.g. able to recognize hand-written characters.

More fundamentally, the recognition and manipulation of symbols appears to generate doubts about the compatibility of computational and dynamic systems approaches to cognitive science (see e.g. van Gelder's discussion of the Watt governor). The association of arbitrary symbols with an interpretation has been discussed under the label of *symbol grounding problem* (e.g Harnad 1990 [113]). More recently, there have been attempts to argue for the grounding of symbols in perceptual experience (Barsalou 1999 [25]); the emergence and grounding of symbols and symbolization has been been attacked within robotics research (e.g. Sinha 2001 [253], Vogt 2003 [294], Inamura et al. 2004 [127] – at the moment only to be mentioned). We take these efforts as a vague indication that the use and manipulation of symbols may not fall outside the domain of dynamic systems.

Given the possibility to set up a dynamic system to perform the computations prescribed by a specific TM description, the question remains whether any basic components beyond those needed for a TM specification will be required to give a description of a dynamic system (even if a description based on such notions may turn out utterly awkward with regard to practical applications). Another question is whether the incorporation of further basic processes – possibly so far unknown or taking up the idea of morphological computation (e.g. Paul 2006 [214]) – will lead to an increase in the power of TMs, and necessitate at the same time a corresponding adjustment of the theory of dynamic systems.

These remarks are by no means to be considered as conclusive or complete, they represent rather a first attempt to fixate some ideas that were stirred up in dealing with the purported incompatibilities and that will have to be worked out properly.

5.3 Benefits of the Dynamic Systems Approach

As described in the beginning of Chapter 4, ideas inspired by or pertaining to the theory of dynamic systems can be traced within a broad scope of fields related to the investigation of (musical) experience and behavior. To sum up, these fields include:

- the investigation of rhythmic behavior and synchronization; some aspects were discussed in Chapter 4; see also contributions in Desain / Windsor 2000 [68]

- more specifically, the dynamic theory of rhythmic attention as developed by Large / Jones 1999 [168], Jones et al. 2002 [142], Jones 2004 [141]

- the investigation of human development as exemplified by the work of Thelen, e.g. Thelen 1995 [278]; see also Bertenthal 2007 [34], Trevarthen 2002 [287]; more technologically oriented applications can be found in the realm of developmental robotics, e.g. Lungarella et al. 2004 [179], Metta et al. 2004 [186]

- aspects of neural network modeling, e.g. Elman 1995 [79], Grossberg 1995 [105]
- phenomena of social or cultural interaction, e.g. Strogatz 2003 [263], Rodgers / Johnson 2007 [235], Leman 2008 [172]
- more generally, problems of self-organization, extended to the evolution of life have been addressed within this framework, e.g. Haken 1983 [108], Kauffman 1993 [152]

A strong appeal by the approach was formed by informal aspects: The interpretations of first order systems of differential equations as vector fields in state space and of solutions to these equations as trajectories in state space provides for a geometric point of view (Strogatz 1994 [262], Section 2.1; Abraham / Shaw 1982 [1]...), allowing to gain insight into qualitative properties of system behavior even if no exact solution is available.

Moreover, notions such as trajectory, fixed point (stable or unstable), attractor, or bifurcation serve as inspiration to think about certain aspects of system behavior such as qualitative changes during the course of development or the emergence of (cultural) phenomena (e.g. Leman 2008 [172], Section 2.3).

More specific technical notions have been taken up in theories and models implemented: The concept of self-sustained oscillators has been explored in the context of rhythm perception and production e.g. by Beek / Peper / Daffertshofer 2000 [26], Peper / Beek / Daffertshofer 2000 [215]. In the theory of rhythmic attention presented by Large and Jones (e.g. Large / Jones 1999 [168], Jones 2004 [141]), the notions of order parameter (e.g. Haken 1983 [108]), generalized phase as taken up in the technical discussion of synchronization by Pikovsky / Rosenblum / Kurths 2001 [225], and a specific concept of phase adjustment play an important role.

As argued above, some sound understanding of the fundamental technical concepts should be acquired in order to obtain full benefit offered by the framework and to avoid getting carried away by qualitative inspiration, losing sight of limitations and implications possibly standing in contrast to the phenomena investigated.

Finally, the framework proposed should not only serve to integrate and make explicit common aspects within research addressed, but also to increase the degree of precision of doubts and reservations.

Chapter 6

Musical Robotics

To an increasing degree, robotic applications are leaving the confines of industrial production and enter into everyday life (e.g. Bar-Cohen / Hanson 2009 [24]). Toy robots as Sony's Aibo have gained some popularity, household aids such as lawn mowers and vacuum cleaners are marketed, and the possibilities of intelligent supports are being investigated.

A natural part of the functionality of these applications concerns the capacities to interact within an intended environment, including humans (and other animals) present, giving rise to the research area of human-robot interaction. For recent topics and challenges see e.g. the sections on "theme and topics" and "tutorials and special sessions" of the IEEE RO-MAN 2006 website[1].

At the same time, robots are employed in the investigation of certain aspects of behavior, e.g. imitation and social learning (Nehaniv / Dautenhahn (eds.) 2007 [197]) or human development in the RobotCub project[2].

The use of robots for ethological research is discussed by Webb 2001 [304]; Möller 2006 [192] presents a popular introduction to the field of biorobotics.

The term *musical robotics* was introduced by the author into German music research and musicological education in a seminar given in the summer term 2006 at the Institute of Musicology, University of Cologne, as a first attempt to get an impression of the use of robotic technology in the context of music with a special focus on the investigation of music-related behavior.

As pointed out in recent overviews (e.g. Kapur 2005 [147], Solis / Takanishi 2007 [257]), there is a long history of robotic applications in musical contexts. Well known early examples are the human-shaped instrument-playing automata

[1] RO-MAN 06: The 15th IEEE International Symposium on Robot and Human Interactive Communication "Getting to Know Socially Intelligent Robots," `http://ro-man2006.feis.herts.ac.uk/`

[2] `http://www.robotcub.org`

built in the 18th century by Vaucanson and Jacquet-Droz: In the late 1730s, Vaucanson presented an automatic flute player[3] and a tambourine player; during the late 1760s, Jacquet-Droz produced among other automata a female musician (*Musicienne*) robot playing keyboard[4].

Focusing on more recent advances, Kapur 2005 [147] describes devices such as

- piano playing mechanisms,
- apparatus for the manipulation of turntables,
- percussion robots for both membranophones and idiophones,
- machinery for plucking and bowing strings,
- robots playing wind instruments.

Two of the examples mentioned in Kapur's overview are described in detail in the Winter Issue 2006 of the Computer Music Journal on robot musicians (Weinberg / Driscoll 2006a [309], Solis et al. 2006 [256], see below).

The systems discussed so far are alternatively addressed as *robotic musical instruments* (Kapur 2005 [147]), *musical performance robots* (Solis / Takanishi 2007 [257]) or *robot musicians* (e.g. Solis / Takanishi 2007 [257], Sobh et al. 2003 [255], Hoffman / Weinberg 2010 [119]). Although differing in emphasis, a common aspect seems to be captured by Kapur's definition of a robotic musical instrument as "sound-making device[s] that automatically create[...] music with the use of mechanical parts, such as motors, solenoids and gears" (2005 [147]), i.e. mechanical sound generation is considered a constitutive feature. Thus, robotic musical applications are set apart from systems relying on purely electronic means of sound generation.

A main focus in the development of (humanoid or human-like) musical performance robots is the investigation of human motor control capabilities in instrumental playing (Solis / Takanishi 2007 [257]), whereas a strong motivation in the design of robot musicians is the extension of traditional musical performances by enhanced mechanical and computational capacities and thus giving rise to new forms of (robotic) musicianship (e.g. Sobh et al. 2003 [255], Weinberg / Driscoll 2006 [309], Hoffman / Weinberg 2010 [119]).

[3]For a short description and a drawing of the mechanism see Solis/Takanishi 2007 [257] or `http://www.francoisjunod.com/automates/eightennth/im_vaucanson/futeplayer.htm` (checked 2010-06-30)

[4]A short technical description along with photographs and a short video clip of the Musicienne playing can be found at `http://www.francoisjunod.com/automates/eightennth/im_jaquetdroz/musicienne_uk.htm` (checked 2010-06-30)

The prospects of using recent robotic and information processing technology in the service of musical and more generally artistic activity appear to be an underlying theme in the special session on *Robotics in Music and Art* of the Ro-Man 2010 conference [258]. Obviously, the design of any system that qualifies in this context requires proficiency in various technical fields such as electrical and mechanical engineering, computer science, acoustics as well as music theory (e.g. Kapur 2005 [147]), signal processing, and music information retrieval. Moreover, system design may be informed by research on cognitive processes underlying human musical behavior. Within research on music cognition, on the other hand, the use of robotic applications is not yet well-established (cf. the contributions for the 2009 ESCOM conference). This reluctance will partly be due to the technical requirements and practical skills needed, but also rests on principled doubts about scientific and computational approaches to the phenomena of music cognition (e.g. Trevarthen 2002 [287]). Here, we will argue that the investigation of music-related human behavior and experience will profit in various ways from research utilizing and relating to robotic applications in musical and artistic contexts.

Notwithstanding the terminological differentiations mentioned above, at the moment I am not aware of any extensive systematic treatment of music-related robotic applications in terms of embodiment and related notions. One attempt, however, to characterize human-machine environments including robots for dance and musical performances in a two-dimensional scheme that appears to be related to notions of embodiment discussed by Ziemke and Chrisley (Ziemke 2001 [328], Chrisley / Ziemke 2002 [54]) is presented by Suzuki & Hashimoto 2004 [268].

The first dimension, represented on the horizontal axis of their Figure 1 (ibid., page 658), is referred to by the contrasting labels environment-oriented vs. object-oriented. These are intended to capture the degree to which the interactive features of the system are concentrated so as to give the impression of one coherent object such as an autonomously interacting robot or are freely placed within the environment in which the interaction is to take place. Placement within the environment is considered regarding both sensory devices and actuators so that there is actually a gradual transition from interactive systems such as provided by the EyesWeb platform in combination with sound and video processing applications to standalone robotic systems. This dimension can be taken to be related to the degree of embodiment of a system. As an example for an installation that uses both robotic behavior and components such as video screens and loudspeakers placed in the environment for mediating interaction, they discuss the visitor robot that interacted with people walking around at the *Arti Visive 2* exhibition in Genova, 1998 (see also Suzuki et al. 2008 [267] for a discussion of the emotional system involved in the internal state of a robot).

The dimension represented on the vertical axis captures the degree of direct vs. indirect response of the system. Indirect response, in this context, is taken to be mediated by the internal state of the system and regarded as a measure of autonomy.

Systems instantiating high degrees of both object orientation and indirect response are considered to be rare by Suzuki and Hashimoto: "No existing system or approach, as far as we know, attempts to construct an autonomous musical robot" (Suzuki / Hashimoto 2004 [268], page 658).

In the following, we will briefly present some examples of robotic musical applications. Instead of following the characterization scheme proposed by Suzuki and Hashimoto, grouping here is motivated by a focus on the intended context of the systems, granting that there will not be any sharp boundaries between the groups.

6.1 Animated Sound Installations

As a first example of what may be called animated sound installation Mark Polishook's *Robots in Residence* will be mentioned, which was set up at the Center for Advanced Visualization and Interaction (CAVI), Aarhus, Denmark, in 2004[5]. The installation consisted of three light sculptures resembling the outlines of human bodies that were suspended from a circular ring. Different parts of the sculptures were connected via strings to a single two-wheeled LEGO RCX robot moving on the floor within the circle and animating the "bodies". Visitors were asked to communicate with the "robots" by sending email messages, which on the one hand were used in the generation of a soundscape and to influence the movements of the LEGO robot, on the other hand answers to the messages were produced by a chatbot. Sound generation and robot control is realized using the SuperCollider sound programming environment.

As the visitor robot mentioned above, the installation clearly does not qualify as robotic musical instrument in the sense defined by Kapur 2005 [147] and apparently adopted for the robotic musician issue of Computer Music Journal, because sound generations is achieved electronically and the robot's task is to produce physical movement not related to sound generation in response to the stimulation received.

Robotic installations that do comply to Kapur's definition are produced by the artists of LEMUR (League of Electronic Musical Urban Robots)[6]. Installations can comprise large numbers of robotic components spread over a room and integrated into a MIDI-controlled network. The components can be specifically designed for certain sound generating structures such as the *GuitarBot* plucking strings and controlling their pitch or the *TibetBot* activating Tibetan singing bowls, others may also be mounted e.g. on parts of buildings to produce sound as the *ModBot*

[5] A movie can be seen at the following website: `http://robots.polishook.org/`
[6] `http://www.lemurbots.org/`

(for descriptions of *GuitarBot*, *TibetBot*, *ModBot*, and some further components see Singer et al. 2004 [252], Singer / Feddersen / Bower 2005 [251]; for a demonstration see e.g. the LEMURtron movie available from the LEMUR website, link in Footnote 6).

MIDI control opens up to the interactive possibilities offered by programming environments such as Max/MSP, allowing for interaction based on camera tracking and sound input, e.g. in a performance together with musicians playing traditional musical instruments. Dedicated "interaction stations" were under development as of 2005 (Singer / Feddersen / Bower 2005 [251], page 52).

6.2 Interaction in Musical Contexts

The aspect of interaction between a human performer and robotic sound generating equipment in live performance situations typically provides a more restricted setting. System requirements are dictated by the performer's need for reliable, rapid, and adaptive performance.

In the development of performance system, the performer's intuitions serve as a heuristic for the detection of processes underlying interactive processes. As an example, Kapur / Singer 2006 [148], Kapur et al. 2006 [150], and Kapur et al. 2007 [149] report efforts to realize a "one-man performance system" (Kapur et al. 2006 [150]) for Indian music, striving to integrate a human performer playing an electronically modified / enhanced sitar (*ESitar*) with a set of specifically adapted drumming mechanisms (*MahaDeviBot*, Kapur et al. 2007 [149], page 239) based on the technology of LEMUR's *ModBot*. Interaction with the performer is achieved by sensors located in the sitar for fret detection and thumb pressure, accelerometers to be placed on the human body and in the instrument are utilized to register movement, and acoustical data is recorded by a pickup on the instrument's bridge. To generate drumming patterns during performance, software was developed to record and analyze pre-recorded music; the patterns are stored in a database and selected during performance by means of techniques for music information retrieval (Kapur / Singer 2006 [148], Kapur et al. 2007 [149], pages 239-240). Cues reported to be used for database retrieval are data from thumb pressure measurement, further cues still to be developed and implemented; interaction with the database is based on the programming language *ChucK* (ibid.).

Further advanced interactive robotic musical applications are the robot drummer *Haile* (see Figure 6.1) and the marimba player *Shimon*, both developed by Gil Weinberg's group at the Center for Music Technology of the Georgia Institute of Technology.

Haile incorporates a sophisticated drumming mechanism (e.g. Weinberg / Driscoll

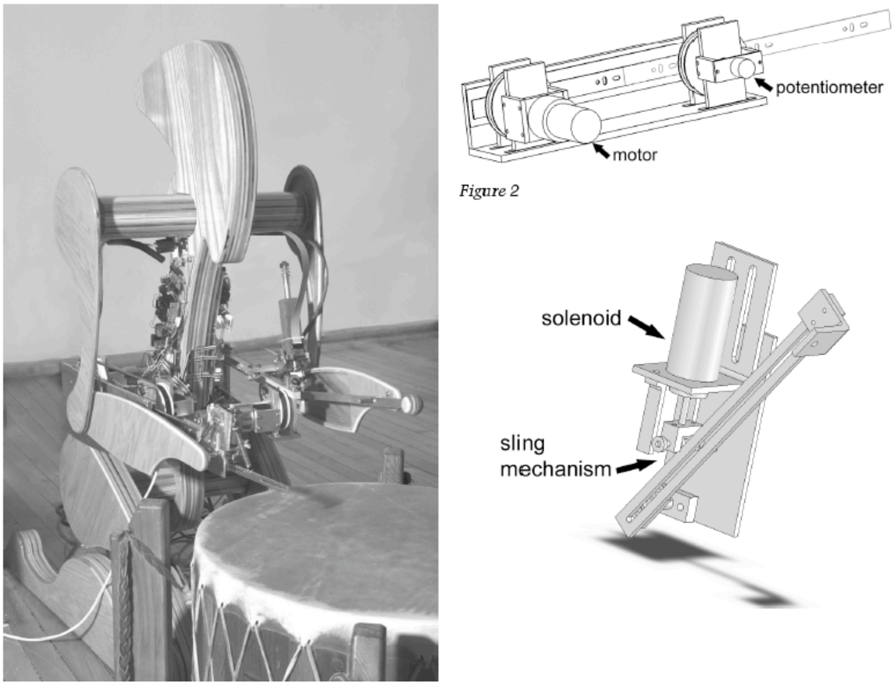


Figure 6.1: The percussionist robot Haile, from Weinberg / Driscoll 2006a [309]

/ Parry 2005 [312], Weinberg / Driscoll 2006a [309]): The left arm is designed to perform fast notes, whereas the right arm is intended “to perform larger and more visible motions that produce louder sounds [. . .]” (Weinberg / Driscoll 2006a [309], page 32). By varying e.g. striking position, velocity of stroke, or duration of contact with the drum, it is possible to produce notes with different pitches, timbres, and volumes. More recently, the mechanism has been modified to play melodic phrases within one octave on a xylophone (Weinberg / Driscoll 2007b [311]).

Besides the optical cues exhibited by the robot, interaction with (human) performers is based on live acoustic input that can be analyzed as to pitch, volume, and aspects of rhythm. The generation of responses can be based on stochastic modifications of “perceived” patterns or involve algorithms for improvisation based on note onset times (Weinberg / Driscoll / Parry 2005 [312]); in combination with the xylophone playing ability, a genetic algorithm based on “a human-generated phrase population” (Weinberg et al. 2007 [313]) was employed, taking into account both audio data and MIDI signals produced by the performers.

The anthropomorphic shape imparted to *Haile* does not appear to be related to the mechanical functioning of the robot. Nevertheless, it was considered to encourage interaction with the robot by participants questioned in “user study” (Weinberg / Driscoll 2006a [309], page 40).

Goals of the *Haile* project concern according to Weinberg / Driscoll / Parry 2005 [312] issues of the mechanics of the robot, perceptual abilities such as the extraction of musically meaningful features from input data, and social interactions with the robot in a musical performance. As an educational tool, they hope to provide an "environment that would allow students to learn about the connection between mathematics, physics, technology and music through programming and composing for Haile" (ibid.).

In the more recent development of the robotic marimba player *Shimon*, the interactive quality of visible sound producing movements has been a major aspect (Hoffman / Weinberg 2010 [119]), motivating a novel way of motion control that incorporates principles previously explored e.g. in the context of animation or theater performance, such as the principle of anticipatory action[7]. Additionally, features of turn-taking based on musical cues have been implemented in the improvisation system to achieve satisfactory interactive (social) performance capabilites (Weinberg et al. 2009 [308, 314], Weinberg / Blosser 2009 [307]). On an artistic level, these considerations address aspects that in more general investigations of human behavior have been discussed under the heading of communicative musicality (Trevarthen / Malloch 2002 [288], Malloch / Trevarthen 2009 [180]).

A quite different form of interaction between a musical performer and a robot based on musical communication is implemented in the performance `par_cho|r:fugue` by Christoph Lischka and Frank Gratkowski (see Figure 6.2).

The robot consists of a spherical shell of about 60cm diameter, covered with an especially durable rubber coating. Inside the shell, two motors are mounted for forward/backward resp. left/right movement; also contained is a loudspeaker system for sound transmission. Motor controllers and sound system are connected to an external computer via a wireless Bluetooth connection.

The computer is running audio analysis and processing software, taking in data from a microphone placed in the room.

In the performance from which the picture of Figure 6.2 is taken, a bass-clarinetist (F. Gratkowski) played improvisations, embedding tonal patterns that control the robot's movements in the music; the commands are extracted and sent to the computer via the bluetooth connection. Apart from processing delays (see the remark in Footnote 22, Chapter 3), according to the explanations given following the performance the robot's movement reactions can be considered as rather direct in terms of the characterization scheme of Suzuki and Hashimoto.

In addition, the music is recorded and transformed electronically and the resulting audio data is streamed to the robot's sound system for replay, turning the robot into a moving sound source.

[7] For a first evaluation see Hofman / Weinberg 2011 [120].

Figure 6.2: Spherical robot in a performance of par_cho|r:fugue by Christoph Lischka and Frank Gratkowski, Cologne University, Department of Musicology, December 5th, 2005

From the point of view of an observer, the spherical robot acts as an interaction partner, responding with movement as well as musical utterances to the instrumentalist's play, especially since the control signals are hardly recognized as such by the audience. Even though the shape of the robot is quite regular, by moving around during the performance it seemed to capture the attention of the audience.

6.3 Human Music Performing Capabilities

By turning to the next group, we are shifting focus to some extent away from mainly artistic applications.

Waseda University, Japan, has a long history of building robots to implement capabilities found in humans. Development of components used in the well-known WABOT-2 robot[8] originated in the late 1960s; the concrete project of building a keyboard playing robot was commenced in 1980. In 1985, a modified version of the WABOT-2, the WASUBOT, was presented in a public performance at the Expo '85 world's fair in Tsukuba, Japan.

The WABOT-2 comprises anthropomorphic arms and legs to play the upper keyboards, bass keyboard, and expression pedal of an electronic organ. A vision system was designed to recognize printed music, a conversation system including speech recognition and synthesis was set up for human-robot interaction, and a singing voice tracking system was implemented to enable automatic accompaniment of a singer. According to Kato et al. 1987 [151], page 144, "playing a keyboard instrument was set up as intelligent work which the WABOT-2 aimed to realize, since an artistic activity like playing a keyboard instrument would require human-like intelligence."

Another long-term and high-level project, the Waseda Flutist Robot was launched in 1990[9]. In figure 6.3, a version of 2006 is displayed.

Explicitly, the main goal of the project is to understand the motor processes underlying human flute playing (Solis et al. 2006 [256], page 12). As subsidiary goals, interaction between robot and human and teaching the flute by means of the robot are mentioned.

Over the years, an increasing number of details reflecting aspects of human anatomy relevant for flute playing have been incorporated in the Waseda Flutist. Most obviously related to the generation of sound are the shape and motility of lips and tongue and those aspects of the respiratory apparatus that are regulating the flow of air or are involved in the production of vibrato. Requirements pertaining to the posture of the flutist are reflected in the degrees of freedom provided for head, neck, and arm movement. The robot's eyes are mobile and equipped with a camera vision system with the intention "to maintain visual contact with the audience" (ibid., page 17).

A major challenge addressed by Solis et al. is the evaluation of the quality of the robot's musical performance. In the attempt reported, the recording of a professional flutist's performance was analyzed to extract timing, pitch, and volume

[8]The informations given here are taken from Kato et al. 1987 [151].

[9]For an overview over the development of the Waseda Flutist Robot, see the project website `http://www.takanishi.mech.waseda.ac.jp/research/flute/`

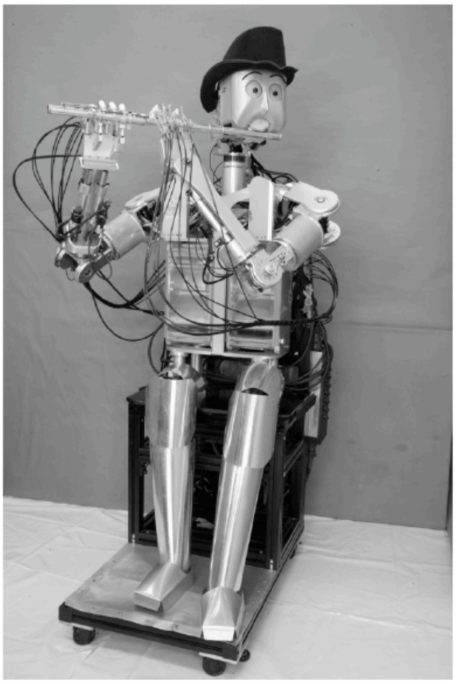


Figure 6.3: Waseda University flutist robot WF-4RII, taken from Solis et al. 2006 [256]

data to be converted into a MIDI score; further player-specific data concerned features such as breathing, tonguing, or vibrato. In the robot's performance, these data are combined with a set of robot data defining position and activity of the robot's components for the notes to be played.

For the comparison of professional player and robot performances, recordings were analyzed with regard to acoustical features such as fundamental frequency ("pitch") or spectral centroid and roll-off (analyses shown ibid., pages 24-25). A good agreement can be observed regarding the features related to pitch and overall intensity, whereas the robot lacks variability in timbre-related and dynamic features in comparison with human performance (ibid., page 23).

6.4 Social Interaction and Synchronization in Musical Contexts

The temporal coordination of agents' activities, for instance referred to by the concepts of synchronization and turn-taking, is considered to play an important role in (human) social interaction (e.g. Kendon 1970 [153]; Condon / Ogston 1971 [62]; Hatfield / Cacioppo / Rapson 1994 [114]; Part V in Nehaniv / Dautenhahn 2007 [197]).

Within rhythmic musical performances, the requirement of synchronization between performers appears obvious and places tight constraints on the temporal properties of interactive devices such as those mentioned above.

In the approach to a drumming task for their robot Nico, Crick / Munz / Scasselati 2006 [63] make use of the "musical environment" as a "relatively well-structured and constrained" (ibid.) domain to investigate aspects of synchronization in a musical performance.

The robot's task is to drum along with human performers, adapting its drumming speed based on acoustical and optical information. The beat performed by the humans is extracted from the acoustical input by an analysis of intensity, more specifically by searching for changes in the stream of smoothed absolute sample values subjected to a threshold function (ibid., page 3). To extract timing information from optical input, the points in time are located when a local minimum in the vertical position in the course of a conducting movement is reached (ibid.).

Timing of the robot's movement is mediated by a set of "attentional" self-sustained oscillators,[10] set to a base frequency of 60 bpm, that attune to the beat pattern detected (ibid., page 4). The use of self-sustained oscillators is motivated by the theory of dynamic attending by Large and Jones; drumming errors are reported to parallel results in tapping experiments by Drake / Jones / Baruch 2000 [73].

One problem to be solved by the robot is to initiate the beating movement in advance for the beat to occur at the correct time. To achieve this, learned forward models are employed (for a detailed discussion see Sun / Scassellati 2004 [266]; cf. Grush 2004 [106]).

The need for the robot to distinguish its own beats from beats produced by the other players is treated as a special case of the more general problem for an agent to recognize the effects of its own actions in the environment. An approach based on temporal closeness is discussed by Gold / Scassellati 2005 [97]; however, this may lead to mistakes if the timing of the agent's movement does not conform to the prediction based on the forward model (Crick / Munz / Scassellati 2006 [63], page 6).

[10] See Section 4.3 for a discussion of the standard example of a self-sustained oscillator.

Another growing field for the investigation of human-robot interaction involving rhythmic movement and synchronization to music is provided by the development of dancing robots or robotic dance partners. Some examples are presented by Aucouturier 2008 [19] or Or 2009 [210]; see also Chapter 8.

6.5 Communication of Musical Expression

A first step in the direction of using robots to explore musical behavior was taken by Burger (2007 [45]). Stimulated by our seminar on musical robotics, she developed an experimental setup to investigate the communication of musical expression by means of a small mobile robot named M[ε]X (see Figure 6.4).

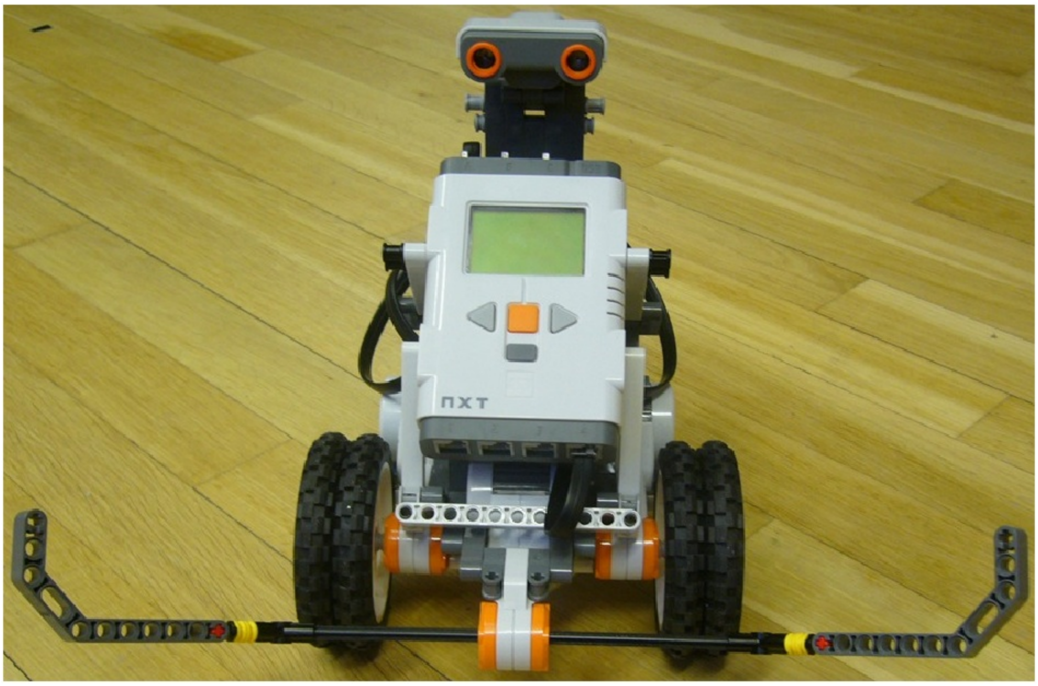


Figure 6.4: M[ε]X, taken from Burger / Bresin 2007 [46]

The robot was built using the LEGO Mindstorms NXT kit. It is driven by a pair of motors activating the front wheels and supported by a third wheel at the rear. A third motor is employed to raise and lower two arm-like extensions at the front. Thus, movement can be characterized by a combination of overall movement pattern in the room and waving of the arms.

The movement patterns to be performed were designed to incorporate on the one hand cues derived from the movement of instrumentalists in musical performance (Dahl / Friberg 2007 [64]), on the other hand the overall shape of movements was checked against the shapes of objects taken to correlate with specific emotional impressions (Isbister et al. 2006 [130]).

Three movement patterns were provided to incorporate cues for the emotions happy, angry, and sad. The happy pattern, as an example, was characterized by fast, fluent, and regular motion and a large amount of gesture, a rounded shape was implemented as a movement pattern based on circles.

In two experimental sessions (at KTH in Stockholm and Cologne University), subjects were presented the robot's movements and asked to rate the degree to which an observed movement pattern was considered angry, happy, or sad.

There were two experimental conditions: In the first condition, only the robot movements were shown, in the second condition, robot movements were accompanied with a musical excerpt intended to convey the same emotion as implemented in the movement pattern.

Under all condition in both experimental sessions, the movement pattern implementing the sad cues was clearly rated highest on the scale for sad; the picture for the other two emotions is less clear. Some reasons for the latter result were considered (Burger 2007 [45], pages 95-98), including similar speed and amount of movement in the movement patterns and environmental conditions (slippery floor) reducing the differences of the patterns.

6.6 Conclusion

As the examples presented in this highly selective review illustrate, the study of robotic applications in musical contexts or *musical robotics* offers ample opportunity to bring together artistic, technological, and scientific approaches to the study and pursuit of musical activity. Studies pertaining to such areas as interaction in classical musical performance situations, behavior in more open artistic contexts, expressive musical movement or "musical" aspects of general social interaction will benefit from taking into account robotic applications as a basis for modeling attempts and means to create more realistic experimental environments.

In particular, the use of robotic applications poses a challenge to become explicit about processes and conditions underlying behavior in music-related contexts to a higher degree than in "disembodied" computer modeling if it is aimed at providing the system with adequate perceptual and motor capabilities to interact in a satisfying manner with human partners. Thus, theoretical descriptions and analyses of the behaviors in question are put to a rigorous test of functionality and completeness, taking into account environmental conditions whose significance might otherwise be missed. Moreover, the integration of different modalities will be an indispensable requirement in such an approach. Besides bringing some more realistic flavor into the task of modeling music-related behavior, robotic systems may on the other hand offer the opportunity to abstract from realistic settings by using simplified or reduced / stylized body shapes, movement patterns or interactional features. Such settings may thus provide a basis for testing the validity of generalized measures characterizing movements and thereby contribute to the investigation of musical gestures and the foundation of music-related metaphors. Novel environments or interaction partners made available by robotic applications may provide a chance to uncover human behavioral patterns that are otherwise not elicited in natural / cultural environments, possibly contributing to the disentanglement of inborn and acquired responses in sound- and music-related contexts. Again, this could shed some light onto the foundations of metaphorical understanding of music.

Besides issues of scientific, technological, and artistic design of the systems, challenges remain concerning the development of methodology for an empirical investigation of the interactive processes taking place within these human-machine contexts (cf. Burger / Schmidt 2009 [47] and references given there). The more regular the structures of the environment, the easier it will be to define some measure of system performance. The question remains to what extent a task-oriented approach such as advocated within human-computer interaction (e.g. Norman 2008 [206]) captures the essential features of the interaction processes and thus is adequate for this domain.

Despite all efforts that have recently been invested into the development of musical robots, however, the challenge of an autonomous musical robot – or, in the terminology introduced earlier, a complete agent engaged in music-related behavior – remains.

Chapter 7

Khepera III: Practical Aspects

As a first step towards the investigation of robotic artifacts within artistic, sound related contexts, it was decided as a part of the project "Artistic Interactivity in Hybrid Networks"[1] (subproject C10 of the collaborative research project SFB/FK 427 "Media and Cultural Communication"[2], funded by the national german research foundation DFG) to utilize a pair of Khepera II robots[3]. One aim is the investigation of patterns of interaction at least at a rather small scale and to gain some insight into the technological issues relating to problems of human-animat (-artifact) interaction in general.

The Khepera II robot is a popular device widely used for the investigation of behaviors such as obstacle avoidance or wall / line following (see examples in Pfeifer / Scheier 1999 [220], Chapter 5) or the implementation of the so-called Braitenberg vehicles (Braitenberg 1984 [41]; Pfeifer / Scheier 1999 [220], Chapter 6). Even evolutionary techniques have been attacked (Nolfi / Floreano 2001 [203]), and not least the Khepera II platform has functioned as an educational tool for robotics (Ichbiah 2005 [125], page 420). As a consequence, a substantial body of project descriptions and applications for the Khepera II is available on the internet.

As a rather advanced example for the use of the Khepera II platform, an application presented by Webb / Reeves / Horchler 2003 [305] can be considered: Here, the Khepera II – which in its standard configuration is not equipped for outdoor operation – is supplemented with sound sensors and a chassis with an extra controller and motor moving on so-called whegs (rotating sets of legs – see Figure 7.1). This setup was devised to test principles of cricket phonotaxis relating the orienting behavior of female crickets towards males making use of acoustic signals to the layout of their auditory and nervous systems.

[1] `http://www.uni-koeln.de/phil-fak/muwi/c10/`
[2] `http://www.fk-427.de/` – only in german
[3] Built by K-Team Corporation, `http://www.k-team.com`.

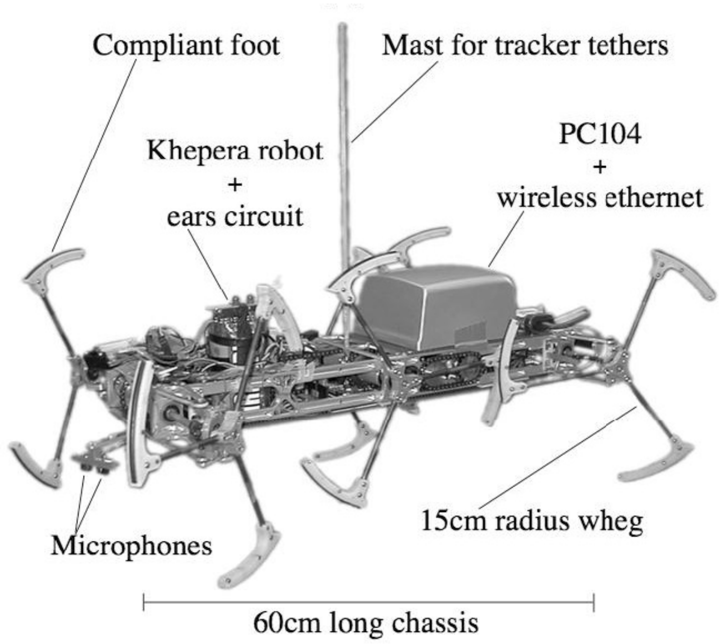


Figure 7.1: Khepera II robot extended for an outdoor test, implementing theoretical assumptions about cricket phonotaxis. Described by Webb / Reeve / Horchler 2003 [305].

The choice of the Khepera II was at least partly motivated by the popularity of the platform (see Schmidt 2005 [242]). At the time of the beginning of our project, however, K-Team Corporation released a successor to the Khepera II, the Khepera III robot. This new device offers greatly enhanced possibilities for the development of control structures and interactive applications, but also confronts the developer with a new set of challenges, e.g. dealing with the use of embedded Linux systems, but partly also resulting from insufficient documentation and the lack of example applications.

As reported elsewhere (Schmidt 2005 [242], 2007 [243]; Schmidt / Seifert 2006 [244]), a technical goal of our project is to provide an interface for the Open Sound Control (OSC) protocol to access the control of the Khepera III robot. The OSC protocol was designed to connect sound programming applications via standard internet connections providing a flexible framework to specify messages sent between the applications (e.g. Wright / Freed / Momeni 2003 [323], Wright 2005 [322], current state of the protocol specification [321, 320]). The protocol has been implemented within popular sound programming environments such as Max/MSP, Pure Data (Pd) and SuperCollider, but even in general purpose programming languages such as Java or C++. Thus, an OSC interface could enable access to the Khepera III independent of the application / programming language used.

Of particular interest for the implementation and investigation of interaction based on sound-related movements / gestures may be the possibility afforded by an OSC interface to integrate work presented by Jensenius: tools are developed for the analysis of (musical) gestures – to be run within the Max/MSP/Jitter environment – along with the discussion of relevant features for the analysis and appropriate formats for streaming related data (e.g. Jensenius / Godøy / Wanderley 2005 [136], Jensenius 2006 [133], Jensenius et al. 2007a,b [134, 135]).

At the time of writing, the goal of providing an OSC interface has not been achieved yet. A preliminary solution using a network connection via a Pd patch, however, can be presented and will be described in some detail below. This description will include aspects of the robot's control architecture, the knowledge of which is a prerequisite for setting up an OSC interface. Mastery of these steps should render the design of the OSC interface a technicality, albeit probably still rather time consuming.

We will start with a superficial technical description of the Khepera III, next turn to the Pd application for interacting with the robots and finally take an intense look at the underlying (low level) control programs.

7.1 Khepera III: Technical Description

The Khepera III is a small, circular mobile robot running on two wheels and a sliding support. The diameter is about 130 mm, the height about 70 mm and the weight without extensions amounts to ca. 690 g[4]. Different view of the Khepera III are presented in figure 7.2, the top row showing a prototype and the bottom row the current commercially available version.

In its basic configuration, the Khepera III is equipped with two motors with associated controllers, a ring of 9 infrared (IR) sensors attached to the bottom layer of the robot's internal structure, another ring of ultrasonic (US) sensors attached to the second layer and an additional pair of IR sensors pointing downward (called ground sensors). Communication with and control of these devices is mediated by a dsPIC 30F5011 microprocessor. (For a textbook description of hardware devices for mobile robots see e.g. Jones / Flynn / Seiger 1999 [140] or Nehmzow 2000 [198].)

[4]All technical details if not explicitly stated otherwise are taken from the Khepera III User Manual (Lambercy / Bureau [167], the version referred to here is dated 2007-03-12) and the specifications published on the website of K-Team Corporation, `http://www.k-team.com`. The material referred to can be retrieved in the latest revision from this site.

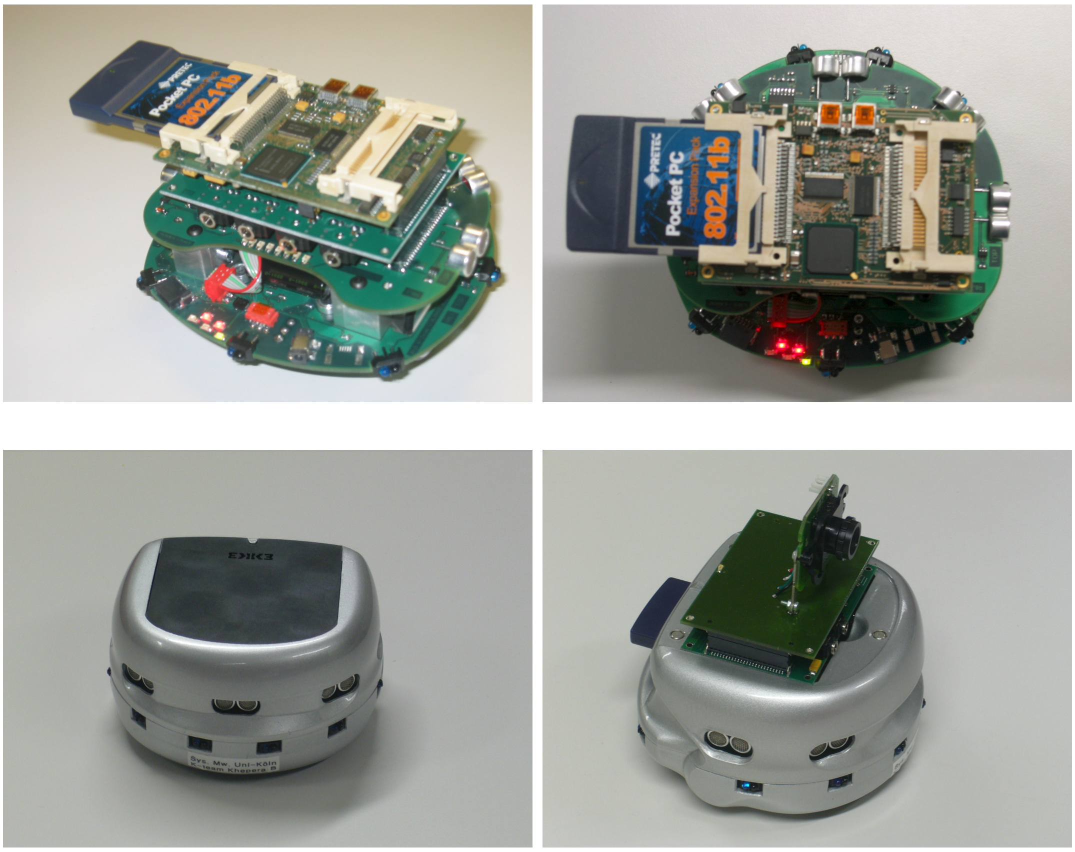


Figure 7.2: Khepera III mobile robot.
Top row: Prototype, still without chassis.
Bottom row: Left panel standard form; right panel with KoreSound card and USB camera.

More specifically, each of the motors is equipped with an incremental shaft encoder producing 16 pulses per revolution of the motor axis; since the motor is connected to the wheels with a 43.2:1 reduction (i.e. 43.2 motor axis revolution corresponding to 1 revolution of the wheel), one revolution of the wheel will correspond to 691.2 pulses produced by the shaft encoder. Because 55 pulses are stated to correspond to 10 mm of distance covered by the robot (Khepera III User Manual [167], Section 3.2), the diameter of the wheels can be calculated to be 40 mm.[5]

The incremental encoder together with a PIC 18F4431 microprocessor (technical

[5]691.2 pulses correspond to $2\pi r$, therefore $\frac{691.2}{2\pi} = 110.01$ pulses correspond to r. On the other hand 110 pulses correspond to 20mm, therefore the radius r equals 20 mm and the diameter will be 40 mm.
The value of 22 pulses per 1 mm robot movement given in the online specification does not appear to be compatible with the User Manual / the actual size of the wheels.

data in [188]) reading the encoder pulses and controlling the pulse width of a 20 MHz pulse, which provides the electrical power of the motor, allows for two different modes of motor control: control of position making use of the number of pulses registered (which can be converted into distances according to the calculations indicated above) and control of speed measuring the number of pulses in time; the maximum speed is specified as 1 m/s.[6] For both control modes, different options and values (including proportionality constants for PID controllers) can be set. The motor controllers act as I2C slave devices (for a detailed description of the I2C bus see [221]).

Even the IR sensors provide two modes of measurement: Every sensor consists of one receiver for infrared radiation and one emitter. In the mode using only the receiver part of the sensors, the intensity of infrared radiation present in the robot's environment will be measured. This mode, which is called *ambient ir* measurement, may e.g. be used to implement heat following or avoiding behavior as classically described by Braitenberg 1984 [41], Vehicles 2 – 4. In the mode referred to as *proximity ir* measurement, the IR emitters are used to obtain a comparison between ambient infrared radiation and radiation reflected from surfaces nearby. More specifically, the value returned from the sensor is the difference in intensity measured between the conditions with IR emitter turned on and IR emitter turned off. Thus, proximity IR measurement does not directly provide a measure of distance to an object: For identical surface conditions, the sensor reading will increase when the distance is diminished; for different surfaces, however, sensor readings will differ according to the reflecting / absorbing properties of the surfaces even if distance is kept constant. According to the online specification, the measuring range for proximity IR measurement is up to 25 cm; the measurement is mainly intended for the implementation of obstacle avoidance (Khepera III User Manual, Section 3.3).

The 11 sensors are read consecutively, starting at the rear left and going clockwise along the ring to the sensor pointing straight backwards, taking the right and left ground sensors last; it is in this order in which the sensor readings are returned as a list when the built-in commands for retrieval of sensor data are issued (see below). The time to jump from one sensor to the next is specified as 3 ms, thus each sensor is read every 33 ms. In its present state, the Khepera III User Manual does not give any precise interpretation of the IR sensor values returned.[7]

Longer range distance measurements (20 cm to 4 m) are performed by the ultrasonic sensors. In the default operation mode of the Khepera III, only the sensor pointing forward is active. Due to the low speed of sound (as compared to the speed of light), ultrasonic measurement requires inherently longer time than IR

[6] resp. 0.5 m/s following the online specifications

[7] Even the format of the numbers returned is not specified coherently: according to Section 3.3, the values are 12 bit numbers (without type specification), in Appendix A describing the built-in commands the values are defined as 10 bit numbers ...

measurement: for a maximum distance of 4 m to an obstacle, i.e. a traveled distance of 8 m for a reflected ultrasonic sound burst, and a speed of sound of 320 m/s, the required time for one measurement is at least 25 ms – thus, in situations requiring rapid interaction, ultrasonic measurement shouldn't be employed too frequently.

In the basic configuration of the Khepera III, the dsPIC 30F5011 processor operates as an I2C master device controlling the motor controllers and reading the sensors. To access the various control functions and to retrieve sensor data, a communication protocol has been implemented consisting of single character commands, which can be followed by additional parameters. Commands are entered as capital letters producing an effect on the robot and evoking a response on the command line. As a simple example, entering the command 'A' will result in the robot entering the "Braitenberg mode," which instantiates some form of wandering behavior including obstacle avoidance – no detailed description is given in the User Manual – and returning the small letter 'a' on the command line.

As another example, which will be taken up below, consider the command 'N': entering 'N' will result in the retrieval of the current proximity measurements of the IR sensors. The answer displayed on the command line consists of the small letter 'n' followed by eleven numbers representing the sensor readings in the order described above and another number specifying the relative time stamp, i.e. the value of the "relative time counter" (User Manual, page 36) indicating the time of measurement.

Further commands are used to configure the mode of operation of the Khepera III, to set various options for the motor controllers or the desired speed / position values, or to retrieve other data from the robot (see Appendix A of the User Manual for a complete list).

To access the command line e.g. via a serial connection, a terminal emulator should be running on the host computer and be connected to a serial port, which can be connected to the RS 232 connector of the KoreConnect adapter intended for operation of the Khepera III in the basic configuration without a KoreBot extension.

The major innovation introduced with the Khepera III robot is the possibility to connect a KoreBot board via the KB-250 extension bus, i.e. by "stacking" the KoreBot board on top of the robot.

The main component of the KoreBot board is an Intel PXA255 XScale processor running at 400 MHz with 60 MB RAM and 32 MB flash memory. The PXA255 processor was developed for "handheld computing applications" ([128], page 1-1) supporting an ARM embedded Linux operating system (for technical details concerning the processor see PXA255 User's / Developer's Manual [129, 128]). In addition, the KoreBot board provides a PCMCIA slot to which in our case a standard wireless network card is connected. In Figure 7.2, top row, the topmost level of the robot is formed by another version of the KoreBot board carrying two PCMCIA slots; in the actual version – Figure 7.2, bottom row – the layout has been changed allowing to stack further extensions on top of the KoreBot board, which is situated within the robot's chassis.

When the KoreBot board is mounted on the Khepera III robot, the dsPIC microcontroller running the communication protocol switches to the I2C slave mode. As a consequence, the control commands described above can no longer be entered directly. Instead, commands have to be transmitted to the dsPIC processor via the I2C bus. For this purpose, a C library containing functions implementing the low level communication with the I2C devices is provided by K-Team Corporation. The latest version of this library is libkorebot-1.10; in the following, we are referring to libkorebot-1.9.1[8].

Access to the ARM Linux command line can again be established via a serial connection as described above but now using the RS 232 connector of the KoreConnect adapter intended for use with the KoreBot board. Alternatively, standard (wireless) network connections can be used such as telnet / ssh resp. ftp / sftp.

The first of the new challenges mentioned above that are posed by the Khepera III platform more precisely concerns the setup of a cross compilation toolchain for the ARM Linux system running on the KoreBot board, i.e. an environment that allows to compile executable programs (e.g. from C or C++ source code) for the ARM Linux system on a different – e.g. Mac OS X or Windows – platform. Here, we will refrain from further discussion of this topic, "simply" asserting that we have successfully compiled the C code discussed below for the Khepera III robot extended with a KoreBot board.

The other challenge, the still incomplete and at times – at least apparently – inconsistent state of the Khepera III documentation has been illustrated in the discussion.

Further extensions available for the Khepera III and displayed in Figure 7.2, lower right panel, include the KoreSound card providing audio input and output and a USB camera.

[8]The libraries can be retrieved from the K-Team ftp server: `http://ftp.k-team.com/korebot/libkorebot/`.

7.2 Khepera III: Pd Interaction

We will start our discussion of interaction with the Khepera III robot with the description of a pure data (pd) application providing a "high level" interface. The patch shown in Figure 7.3 was originally prepared by Tobias Grewenig and Ralf Baecker within the research project mentioned above; some additions and corrections were introduced by the present author.

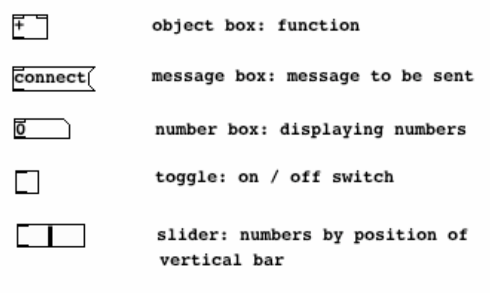
In the figure, four regions enclosed by polygons are displayed. These regions correspond roughly to different types of functionality:

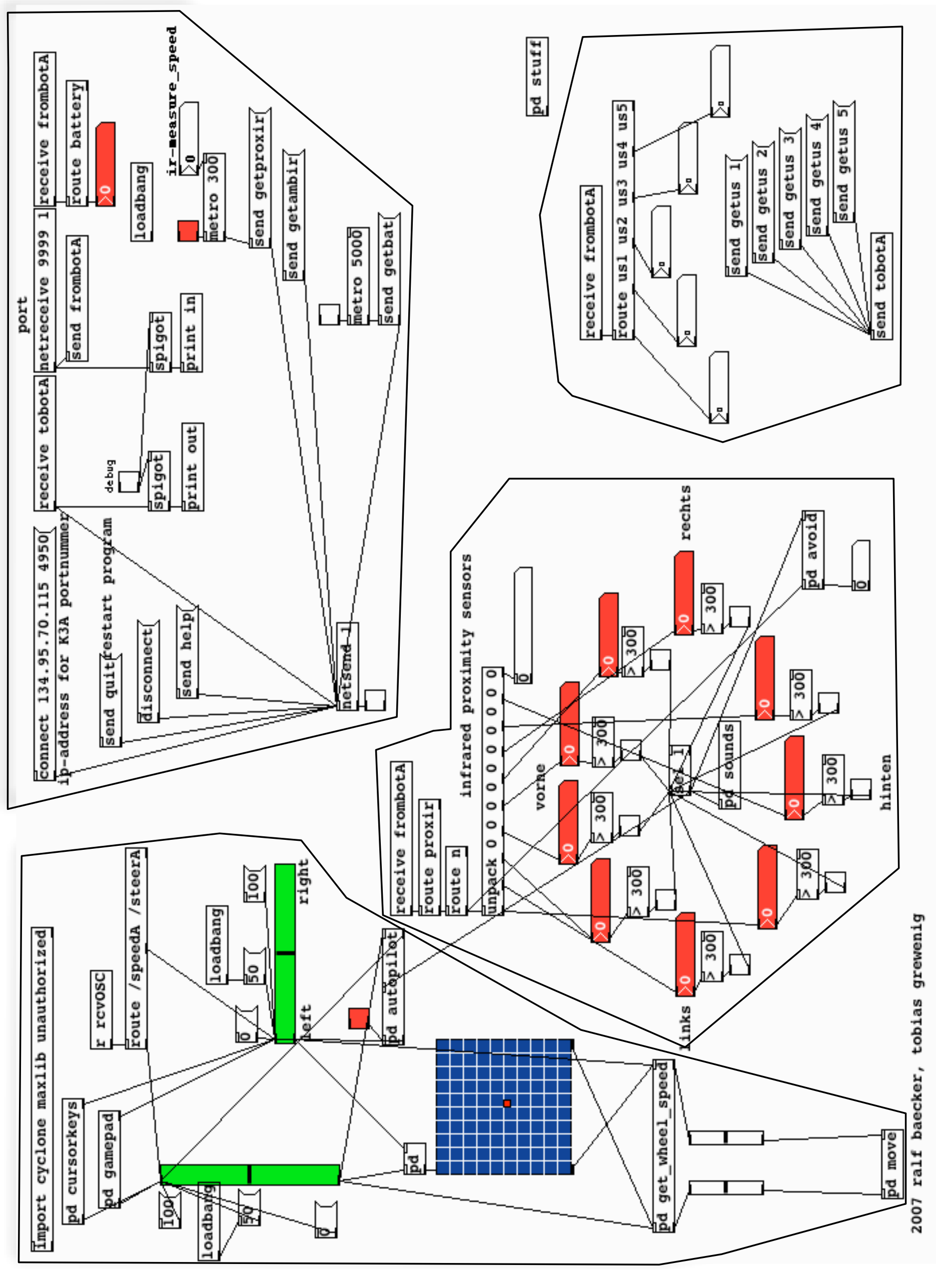
1. In the upper right area, functions pertaining to network interaction are collected;
2. the area on the left is related to movement control of the robot;
3. functions in the lower right area address the ultrasonic sensors and display data obtained from US measurement;
4. the functions collected in the lower middle evaluate data retrieved from the infrared sensors.

Network communication with the Khepera III is established using the Pd `netsend` and `netreceive` objects[9]. Here, they are used with a non-zero creation argument (the argument following the function name in the corresponding object boxes) specifying the network protocol to be used as the UDP (User Datagram Protocol) protocol; a zero or missing creation argument would set the TCP/IP protocol. The IP address and receiving port number of the robot need to be edited in the message box containing the `connect` command. The robot's receiving port is defined in the control program running on the KoreBot board, the IP address must of course conform to the momentary IP configuration of the KoreBot.

Data is sent to the robot in the form of messages consisting of sequences of characters (including numerals), which by the control program are interpreted as

[9] The shapes and basic functionality of the Pd boxes used in Figure 7.3 are indicated in the following figure:





Bounded regions roughly correspond to network communication (upper right), movement control (left), IR measurement (center), US measurement (lower right).
Calculation of actual speed values sent to motor controller is performed in the canvas object 'pd get_wheel_speed'.

Figure 7.3: Pd patch implementing the wireless communication with the Khepera III robot.

names and arguments of robot control commands: Whenever any of the *message* boxes containing a message that starts with `send` is activated (by clicking or by an activating impulse – bang – from another component of the patch), the part of the message following `send`, e.g. `quit` or `getambir` is transmitted to the robot by the `netsend` object.

Incoming data from the robot is received by the `netreceive` object. The first creation argument of `netreceive` sets the port of the host computer listened to; in the control program running on the KoreBot, this is defined as the sendport number. In a setting involving more than one robot, different sendport numbers are specified in the resp. control programs to keep apart data from the different sources.

Within the patch, data received from the robot is distributed using the Pd internal `send` and `receive` objects: data transmitted from the `send frombotA` object will be processed by any `receive frombotA` object in any currently opened Pd patch.

Data to be processed is selected using the Pd `route` function; the need for the consecutive `route` objects in the context of IR measurement will become clear in the discussion of low level control below

The evaluation ultrasonic measurement in the application shown is restricted to the display of sensor values; the geometrical arrangement of the number boxes in the patch reflects the placement of the corresponding US sensors on the robot.

Even the number boxes displaying the reading of the IR sensors (red boxes in the lower middle area) are arranged corresponding to the placement on the robot, discarding the data retrieved from the ground sensors. The values in this patch are used for two different purposes: whenever any sensor reading exceeds the value of 300, an activating impulse is generated. On the one hand, this is sent to the `pd sounds` canvas object triggering the playback of a short sound sample ("boing"; the functions used are encapsulated within the canvas object). On the other hand, the impulse is sent to the `pd autopilot` canvas object that implements an automatic movement control of the robot including obstacle avoidance. When the autopilot is turned on and receives an impulse resulting from an IR sensor reading higher than 300 – indicating the presence of some object in the vicinity of the robot – the robot's direction of movement is inverted until new speed and direction values are generated randomly.

When the autopilot is turned off, movement of the robot can be controlled in two different ways using this patch: the (green) slider objects are used to generate values for speed (vertical slider) and direction (horizontal slider) independently, the grid object functions as a controller resembling a joystick. The numerical values generated by the slider and grid objects range from 0 to 100; calculation of the actual values sent to the robot is encapsulated in the `pd get_wheel_speed` canvas object, in the `pd move` object the values are combined into messages with the appropriate control command `setmotspeed`.

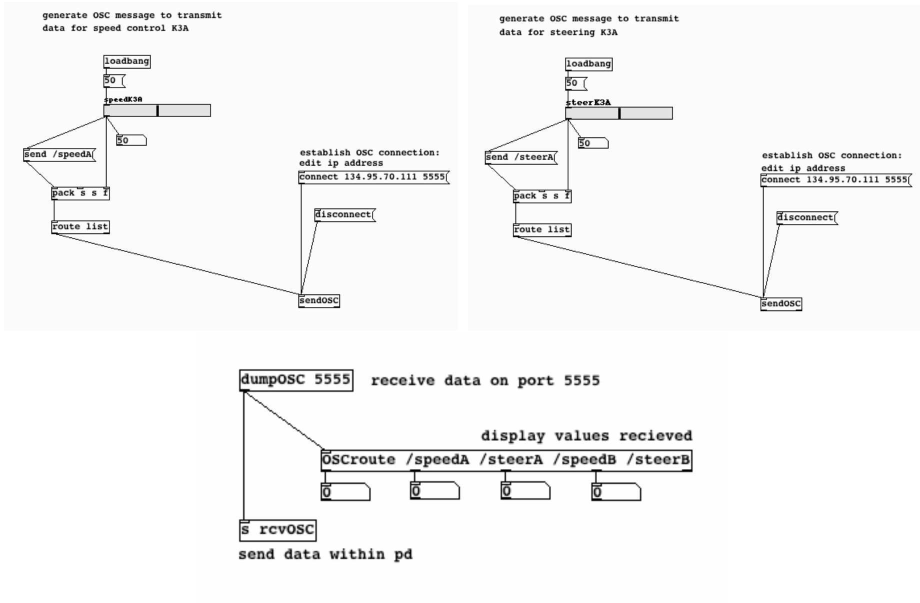


Figure 7.4: Patches for control of the Khepera III via OSC
Top: separate patches for speed and direction control
Bottom: patch receiving control data

In addition to the patch discussed, a set of small Pd applications for movement control of two Khepera III robots via OSC connections (shown in Figure 7.4) was prepared. In the patches displayed in the top row, numbers in the range 0 to 100 generated by slider objects are combined into messages with keywords indicating the intended use of the numbers; the keyword `/steerA` refers to direction control, the keyword `/speedA` to speed control, and the capital letter 'A' included in both keys indicates the robot addressed. Using the `sendOSC` object, the messages are coded according to the OSC specification and sent to the IP address and port defined in the `connect ...` message box. On the computer indicated by the IP address, the messages can be received and decoded using the `dumpOSC ...` listening to the port defined in the sending patches (see bottom row of Figure 7.4). Here, the message received is transmitted further within the Pd application using the `s rcvOSC` object[10]. A corresponding `r rcvOSC` object is included in the control patch of Figure 7.3, again using the `route` function to select data for speed and direction control according to the keywords `/speedA` and `/steerA`.

These patches were informally tested in a playful classroom situation, serving as an example to illustrate possible applications of the OSC protocol. The "slider patches" were running on five or six different laptop computers and there was a host computer running control patches for both robots. Participants were given

[10]'s' is used as an abbreviated form of 'send'; likewise 'r' is used as a shorthand for 'receive'.

the task to collaborate in controlling the robots in order to push a small box in a certain direction – collaboration of at least two "operators" is required to steer one robot because only one slider can be operated on any computer at a time. The task turned out to be tricky for the following reasons:

- since the robots look alike, the operators first need to find out which one reacts to the objects they are manipulating – and keep track of "their" robot once they found out,
- operators can not be certain whether the reaction observed was a result of their action,
- coordination with the person manipulating the other control parameter of the robot needs to be established,
- other persons might be interfering with one operator's actions,
- technical problems include reaction times of the robots and delays in the transmission of control commands.

It is intended to replace the sliders by other input devices such as sensors registering body movements or motion tracking applications as developed by Jensenius (see above) / the EyesWeb system (Camurri et al. 2007 [50]). A first motion tracking application based on the patch described has been implemented by Jochen Arne Otto (see Appendix D). More sophisticated robot behaviors will be taken into account as well.

7.3 Low Level C Programming

As repeatedly mentioned in the previous section for interaction with the Khepera III robot using the Pd patches described, a control program is required running on the KoreBot board that decodes the messages sent to the robot into appropriate control commands and corresponding arguments and on the other hand encodes the responses from the robot into messages that can be put to use within the Pd application. The control program to be discussed is based on a test application for the Khepera III provided by K-Team Corporation as part of the libkorebot libraries; the C source code is contained in the file `khepera3_test.c`. Along with the preparation of the Pd patch, this application was modified by Tobias Grewenig and Ralf Baecker to incorporate functions for network interaction and coding / decoding messages sent between computer and robot. Networking functionality was established by importing source code for the Pd `netsend` object by Miller Puckette. Some changes and additions had to be introduced by the present author: Grewenig / Baecker started from the application as provided with the libkorebot-1.8 distribution. Since the return values of some low level functions were changed from pointer to integer type with the libkorebot-1.9.1 distribution, the functions in our application had to be adapted. Even the syntax and return values of some commands of the communication protocol running on the dsPIC 30F5011 seem to have been changed so that the functions making use of these had to be revised, too. A minor change concerns the definition of different sendport numbers for different robots.

In the following, we will specifically discuss the command for retrieving an ambient IR measurement, which involves both receiving and sending data on the side of the robot, in order to gain some insight into the control architecture implemented in the libkorebot distribution. The complete listing of the code for our control application is included as Appendix E. We will only look at functions specific to the Khepera III platform, i.e. details of setting up the network connection will be left out.

7.3.1 Decoding Messages: Command Table and Command Parser

Messages sent to the robot via an existing network connection will be written into a character array. This array will be parsed – by a function defined in the file `kb_commandparser.c` of the libkorebot distribution – for names and arguments to be executed; the parsing function is invoked by the line

```
kb_parse_command( sbuf, cmds, NULL);
```

sbuf is the name of the buffer containing the data and cmds is the name of a structure defining the available command names.
More specifically, the command table cmds maps arbitrary strings to a minimum and a maximum number of arguments to be entered and the name of a function to be called. The actual function call is issued by the command parser program. The command table of our application is defined as follows:

```
/*--------------------------------------------------------------------*/
/*! The command table contains:
 * command name : min number of args : max number of args : the
 * function to call
 */
static kb_command_t cmds[] = {
 { "quit"            , 0 , 0 , quit } ,
 { "exit"            , 0 , 0 , quit } ,
 { "bye"             , 0 , 0 , quit } ,
 { "ciao_bella"      , 0 , 0 , quit } ,
 { "setcfg"          , 2 , 2 , configureOS },
 { "getrev"          , 0 , 0 , revisionOS },
 { "getbat"          , 0 , 0 , voltageBAT },
 { "rststamp"      , 0 , 0 , tstampRST },
 { "getambir"      , 0 , 0 , ambIR },
 { "getproxir"       , 0 , 0 , proxIR },
 { "getus"           , 1 , 1 , measureUS },
 { "setmotspeed"     , 2 , 2 , motSpeed },
 { "setmotmove"      , 2 , 2 , motMove },
 { "motstop"         , 0 , 0 , motStop },
 { "help"            , 0 , 0 , help } ,
 { "getallus"        , 0 , 1 , getallUS },
 { "benchmark"       , 1 , 1 , getBenchmark },
 { "reboot"          , 0 , 0 , doReboot },
 { "alive"           , 0 , 1 , alive },
 { NULL              , 0 , 0 , NULL }
};
```

As can be seen from the first four entries, any number of strings can be associated with one function name, here: quit. As also illustrated, command names can be arbitrary but should be chosen so as to indicate the function performed.

The commands defined in our example may pertain to (list not exhaustive):

– the operating system of the dsPIC processor: setcfg can be used to configure the mode of operation, e.g. the number of US sensors active; getrev retrieves the current revision of the operating system,

- further information about the current state of the robot: `getbat` in the revision addressed here retrieves the momentary battery voltage; in a more recent version, some more information including the battery temperature can be obtained,

- retrieval of sensor data: `getambir` and `getproxir` retrieve ambient resp. proximity IR measurements, `getus` retrieves the ultrasonic measurement of the sensor whose number is entered as argument, `getusall` reads all US sensors,

- setting of certain control values: for example, `setmotspeed` sets the desired speed values for the motors thus requiring two arguments; in a similar way `setmotmove` sets the desired position values,

- as an exception in this list, the command `reboot` leads to the restart of the ARM Linux operating system of the KoreBot board.

The commands possibly included in messages sent by the Pd patch are: `getproxir`, `getambir`, `getbat`, `getus 1...5`, `quit`, `help`, and `setmotspeed` – the last is used within the object `pd move` of the patch.

We will concentrate on the command `getproxir`. As can be seen in the command table, no arguments are required or allowed. Upon finding the string `getproxir` in the input array without arguments, the command parser will call the function `proxIR` defined as shown below within the control program.

7.3.2 Retrieving IR Data: `getproxir` → `proxIR`

The function `proxIR`, which is called when a message containing the command `getproxir` is received, is defined by the following code fragment:

```
/*--------------------------------------------------------------------*/
/*! proxIR retrieves proximity ir measure using kb_khepera3.c library.
 */
int proxIR( int argc, char * argv[], void * data)
{
  char irdata[512];
  char Buffer[MAXBUFFERSIZE];
  if(kh3_proximity_ir((char *)Buffer, dsPic)) {

     sprintf(irdata,"proxir %c %4.4u %4.4u %4.4u %4.4u %4.4u %4.4u %4.4u %
     4.4u %4.4u %lu\n",
    Buffer[0], (Buffer[1] | Buffer[2]<<8), (Buffer[3] | Buffer[4]<<8),
   (Buffer[5] | Buffer[6]<<8), (Buffer[7] | Buffer[8]<<8),
   (Buffer[9] | Buffer[10]<<8), (Buffer[11] | Buffer[12]<<8),
   (Buffer[13] | Buffer[14]<<8), (Buffer[15] | Buffer[16]<<8),
   (Buffer[17] | Buffer[18]<<8),
   ((Buffer[19] | Buffer[20]<<8) | (Buffer[21] | Buffer[22]<<8)<<16));
     pdsend(9999,rip,irdata);
  } else
   printf("\r\nn, error...\r\n");
}
```

First, two character arrays are declared: the character array `Buffer` will be used to store data read from the sensors, which in turn will be written into the character array `irdata` together with the keyword `proxir`. It is the array `irdata` that is sent to the computer hosting the Pd patch, where this message is selected according to the keyword by the `route proxir` object.

In the head of the if-statement, another function `kh3_proximity_ir` is called. The integer return value of this function decides which of the branches will be executed: for non-zero return values, data will be processed and sent to the host computer, for a zero return value, an error message will be printed.

The function `kh3_proximity_ir`, defined in the file `kb_khepera3.c` of the libkorebot distribution, is called with two arguments: The first argument is a pointer to the array `Buffer` defined locally within the function `prixIR`. The second argument is a pointer to the device addressed and is defined globally for the complete control program. As the name of the second argument suggests, the device addressed is the dsPIC 30F5011 processor running the communication protocol.

The handling of data retrieved by the function `kh3_proximity_ir` closely reflects the way data is returned when the command `N` is entered in the communication protocol (see above, page 120): The first element of `Buffer` – `Buffer[0]` – which encodes the small letter ‘n’ is processed independently. The following elements of `Buffer` are combined in eleven pairs, each pair encoding the reading of an IR sensor in two bytes. The bit shifting operation performed on the second element of each pair[11] points at the fact that numbers are encoded in the little endian format.

The inclusion of the element `Buffer[0]` necessitates the presence of the `route n` object in the part of the Pd patch evaluating IR sensor data.

7.3.3 Integrating the Communication Protocol: `kh3_proximity_ir`

Turning to the function `kh3_proximity_ir` will illustrate the way commands of the communication protocol are accessed by control programs running on the KoreBot board. The definition of this function including comments is presented in the following code fragment:

```
/*!
 * kh3_proximity_ir retrieves an instant IR measure.
 * \param outbuf is a buffer where the data will be stored on.
 * \param hDev is a handle to an openned knet socket (Khepera3:dsPic).
 * \return NULL or a pointer to the IR measure
 */
int kh3_proximity_ir(char *outbuf, knet_dev_t *hDev){

  int rc , i;

  /* Frame format : { Size, Command, Terminator }
   * where the command can be more than 1 byte */
  char cmd[3] = { 2, 'N', 0};

  if(hDev) {
  kh3_sendcommand( hDev , cmd );
   /* delay to ensure the correct reading of KNET_INTO pin */
usleep(K3_CMD_DELAY);
while(!kb_gpio_get(KNET_INTO));

rc = kh3_getcommand( hDev, outbuf );
return rc;
        }
  return 0;
}
```

[11] `<<` 8: shifting 8 bits – one byte – to the left

As evident from the specifier preceding the function name, the return value of the function is of type integer – but according to the last line of the initial comment it should be a NULL pointer or a pointer to the array holding sensor data. This is another example of documentation problems which led to misunderstandings and delay in setting up the Pd interaction.

As discussed in the previous section, the two arguments of the function are pointers to an array holding data and to the device data will be retrieved from, the dsPIC processor.

In the body of the function, a *command frame* `cmd` is defined in the form of a character array. The array holds as its first element the number of bytes following this element. The second element is a character specifying a command of the communication protocol, here the capital letter `N`, if required / admissible followed by any arguments of the command – none in our case. The last element of the command frame is 0, used as a terminating symbol.

The function `kh3_proximity_ir` makes use of two other functions, `kh3_getcommand` and `kh3Ñsendcommand`, which, too, are defined within the file kb_khepera3.c. As the names indicate, the first of these is used to pass the command frame to the dsPIC processor, the second retrieves the answer produced by the dsPIC and writes it to the array `Buffer` referenced by the pointer `outbuf`.

Between the passing and retrieving commands, a delay is inserted consisting of a fixed amount of time defined by the constant `K3_CMD_DELAY` and a variable part determined by the condition of a while loop. In the file `kb_khepera3.h`, the constant `K3_CMD_DELAY` is set to 300 μs, the while loop waits for a certain pin to signal readiness.

7.3.4 Interacting with the dsPIC: `kh3_sendcommand` and `kh3_getcommand`

The functions `kh3_sendcommand` and `kh3_getcommand` are "as far down" as we will have to go. All sensor related functions defined in the control program eventually make use of these functions, because the sensors are controlled by the dsPIC processor.

Although appropriate commands for motor control are available in the communication protocol, these can not be used in the manner described here: Since the dsPIC processor operates in the I2C slave mode in connection with the KoreBot board, no commands will be issued to the I2C devices controlling the motors, the PIC 18F4431 processors. Instead, special functions defined in the file `kmot.c` of the libkorebot distribution have to be employed.

The code defining `kh3_sendcommand` and `kh3_getcommand` is included as an illustration in the code fragments at the end of this section. The first argument of `kh3_sendcommand` is a pointer to the dsPIC device, the second argument points to the command frame defined within `kh3_proximity_ir`. The pointer to the array `Buffer` defined in the function `proxIR` is passed as second argument to `kh3_getcommand` eventually writing data into `Buffer`, which then can be sent to the host computer as described above.

These remarks may suffice as an overview to give an impression how interaction with the Khepera III can be implemented. For actual programming work – e.g. preparing an OSC interface as proposed above – of course more detailed work will be required, but the general framework should be clear.

Code Fragment: kh3_sendcommand

```
/*!
 * kh3_sendcommand sets a command frame to a given khepera3 device.
 *
 * Normally and end user doesn't want to use these function as they are
 * assumed as "low level functions".
 *
 * \param hDev is a handle to an opened knet socket (Khepera3:dsPic).
 * \param in is a pointer to a buffer where the command frame to be sent
 * is stored on.
 *
 * \return A value:
 *       - <0 on error (KH3_ERROR_FRMSNDERR)
 *       - >=0 on success (returns should be the size of frame)
 *
 * \remark This function requires that kb_kh3_init has been called
 */
int kh3_sendcommand( knet_dev_t *hDev, unsigned char *in )
{
 char sizeMsg;

 /* first byte in the frame is the complete frame size */
 sizeMsg = in[0];

 if( knet_llwrite( hDev, in, sizeMsg) == sizeMsg)
  return sizeMsg;
 else
 {
  KB_ERROR("knet_sendCommand", KB_ERROR_KH3FRMSNDERR);
return KH3_ERROR_FRMSND;
 }
}
```

Code Fragment: `kh3_getcommand`

```
/*!
 * kh3_getcommand gets a command frame from a given khepera3 device.
 *
 * Function flow:
 * - a) :  retrieve the first byte which is the frame size from the device
 * - b) :  retrieve the required bytes
 *
 * Normally an end user don't want to use these function as they are
 * assumed as "low level functions".
 *
 * \param hDev is a handle to an openned knet socket (Khepera3:dsPic).
 * \param out is a pointer to a buffer where the command frame
 * will be stored on.
 *
 * \return A value:
 *       - <0 on error (KH3_ERROR_FRMSZERR, KH3_ERROR_SZFMTERR)
 *       - >=0 on success (returns should be the size of frame)
 *
 * \remark This function requires that kb_kh3_init has been called
 */
int kh3_getcommand( knet_dev_t *hDev, unsigned char *out )
{
 char sizeMsg;
 int rc;

 if( knet_llread( hDev, &sizeMsg, 1 ) == 1 )
 {
  rc = knet_llread( hDev, out, sizeMsg );
if(rc == sizeMsg)
return rc;
else
{
KB_ERROR("knet_getCommand", KB_ERROR_KH3FRMSZERR, rc, sizeMsg);
return KH3_ERROR_FRMSZ;
}
 }
 else
 {
   KB_ERROR("knet_getCommand", KB_ERROR_KH3SZFMTERR);
 return KH3_ERROR_SZFMT;
 }
}
```

Chapter 8

Interaction, Synchronization, and Turn-Taking

In the previous chapters (Sections 3.1.3, 4.2, 5.3, 6.2, 6.4), we have repeatedly encountered the notion of synchronization. In music-related contexts, synchronization will most readily be associated with the precise temporal coordination of performers' activities required in ensemble performances or listeners' behaviors taking up the rhythm of the music, which may range from finger / foot tapping to elaborate figures of dance.

In the investigation of rhythm perception and production (see Bengtsson / Gabrielsson / Thorsén 1969 [32] or Fraisse 1982 [88] for a review of early work), the task of finger tapping to match a stimulus such as a periodic or isochronous beat pattern has been established as a central experimental paradigm; more recently, the task has been extended to include polyrhythmic beat patterns (Handel / Oshinsky 1981 [110], Moelants / van Noorden 2005 [191], McKinney / Moelants 2006 [185]) or tonal musical excerpts (Toiviainen / Snyder 2000 [283], 2003 [284]) as stimulus material. For the dynamical theory of attention by Large and Jones, mentioned in Sections 5.3 and 6.4, the task of finger tapping appears to be the underlying experimental test bed, too.

Tapping may be related to spontaneous periodic behaviors observed in humans. Fraisse (1982 [88], pages 151–152) describes e.g. the sucking of new-born infants, rocking, and walking as well as the associated average frequencies as rhythmic movements to be taken into account; as mentioned in the discussion of the "resonance model" (see Section 4.2), Todd / Cousins / Lee 2007 [280] relate preferred beat rates observed in human subjects to a set of anthropometric measures of their bodies. As will be described below (Section 8.1), the presence of spontaneous, self-sustained periodic movement is a prerequisite for synchronization in a certain technical sense to occur.

Adopting a broader perspective, the task of beating in synchrony with a presented pulse was addressed as a special case of synchrony in social interaction in the context of the robot Nico (Crick / Munz / Scassellati 2006 [63], see Section 6.4). In the case of Nico, spontaneous periodic movement with a preferred frequency was implemented by means of self-sustained "attentional oscillators" providing the internal time pulse of the robot. The drumming task was chosen because of the relatively high degree of regularity afforded by the musical context.

Loosening the restriction on regularity, temporal coordination and synchronization have long been recognized as important aspects of interactional social processes. Kendon 1970 [153], here quoted from the reprint 1990 [154], attributes the first description of the phenomenon of *interactional synchrony* to work of Condon and Ogston in the late 1960s (Kendon 1990 [154], page 92). To provide some further examples, he presents an analysis of the temporal alignment of speakers' as listeners' actions in a conversational situation. Based on detailed descriptions of film material, he points out the coincidence of "points of change in the flow of sound" with "points of change in the body movement" (ibid., page 93) within the speaker as well as "the boundaries of the movement waves of the listener [...] with the boundaries of the movement waves in the speaker" (ibid.). In the postscript added for the reprint, Kendon takes the occurrence of interactional synchrony as "a manifestation of attentional and affective attunement" (ibid., page 115).

Evidence for the transmission and *sharing* of emotions due to mimicry and synchronization of "movements with the facial expressions, voices, postures, movements, and instrumental behaviors of others" is discussed at length by Hatfield / Cacioppo / Rapson 1994 [114], page 10) under the label of *emotional contagion*. These ideas are taken up e.g. in the Premio Paganini experiment described shortly in Section 3.1.3 (Camurri et al. 2007b [49]). In another recent approach drawing on previous work on caretaker – infant interaction, Revel / Nadel 2007 [232] compare capabilities required for imitation learning in human infants and a specific robotic platform (ETIS), stressing the role of turn-taking and synchrony for the sharing of experience.

In the examples discussed so far, the temporal coordination of processes and/or events is referred to by the term synchronization. Nevertheless, conceptual differences can be observed that may limit the possibilities to transfer theoretical approaches from one scenario to the other. In the following, we will first present some different interpretations of the notion of synchronization and discuss specific problems raised by these interpretations. As a next step, some remarks on the observation of interactional processes involving synchronization will be offered; as a case study, reference will be made to an experiment described by Michalowski / Sabanovic / Kozima 2007 [187] addressing the facilitation of human(child)–robot interaction by synchronization between music and robot movement. Finally, we will include a project proposal taking up aspects of synchronization, turn-taking, and entrainment as an indication of possible directions for future work.

8.1 Synchronization: Technical Notion

A technical notion of synchronization is developed within the framework of the theory of dynamic systems. Here we will follow the presentation given by Pikovsky / Rosenblum / Kurths 2001 [225] (a condensed version is Rosenblum / Pikovsky 2003 [237]). A more popular overview offering a somewhat different perspective can be found in Strogatz 2003 [263].

Pikovsky / Rosenblum / Kurths (2001 [225], page 8) treat synchronization as "an adjustment of rhythms of oscillatory objects due to their weak interaction." Oscillatory objects in this context are taken to possess the properties of self-sustained oscillators such as the van der Pol oscillator described in Section 4.3. Restricting ourselves to the case of periodic oscillations, self-sustained oscillators are characterized by a stable oscillation frequency and by sustaining a constant amplitude, i.e. increasing or decreasing amplitude if at some point in time the momentary amplitude is below resp. above a certain value (cf. Figure 4.11, 4.13, or 4.14). As a consequence, the oscillatory patterns of a self-sustained oscillator will look alike independent of initial conditions once an initial time has passed, except for a possible time shift of the pattern. The trajectory associated with a self-sustained oscillator will approach a limit cycle (cf. Figure 4.10, 4.12, or 4.15).

A central concept introduced in the discussion of synchronization is the *phase* of an oscillation. Essentially, the phase is a means to relate to a specific point in time within one oscillation period, i.e. to a specific point on the limit cycle. The notion of phase is generalized from the case of sinusoidal oscillations: In the most simple form, the trajectory for a sinusoidal oscillation can be described by the coordinates

$$\begin{aligned} x(t) &= \sin(\frac{2\pi}{T}t + \varphi_0) \\ \dot{x}(t) &= \cos(\frac{2\pi}{T}t + \varphi_0), \end{aligned}$$

describing a circle around the origin in phase space.

The argument $\Phi(t) = \frac{2\pi}{T}t + \varphi_0$ is commonly called the phase of the sinusoidal oscillation. In this case, $\Phi(t)$ increases by equal amounts in equal durations, and equal differences in time / phase correspond to equal distances along the limit cycle. The last relationship will already fail to hold when the two variables are allowed to be scaled differently, e.g. if

$$\begin{aligned} x(t) &= a\sin(\frac{2\pi}{T}t + \varphi_0) \\ \dot{x}(t) &= b\cos(\frac{2\pi}{T}t + \varphi_0) \end{aligned}$$

and $b \neq a$: Again, equal differences in time will correspond to equal phase differences, but – as illustrated in Figure 8.1 – the distance between points marking states separated by equal time / phase differences along the trajectory will no longer be equal.

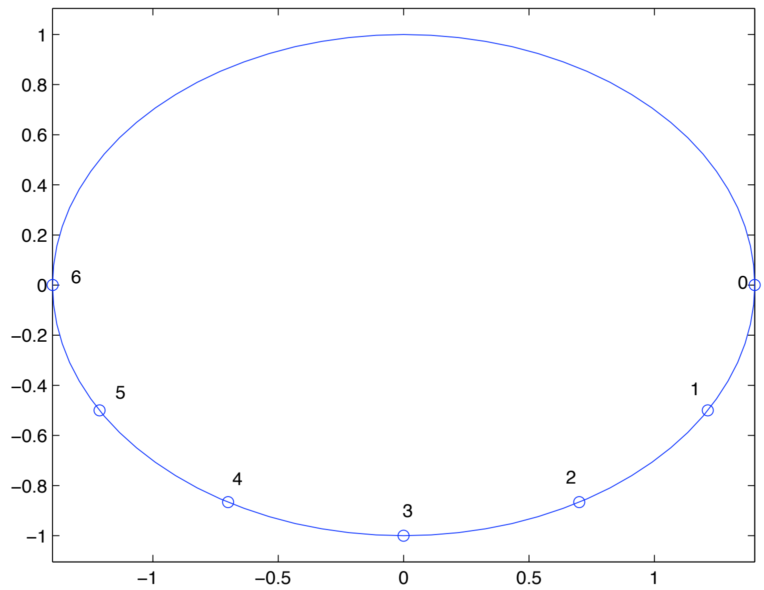


Figure 8.1: Phase: The points marked 1,2, . . . 6 correspond to time points equally spaced by a phase difference of $2\pi/12$.

The property generalized in the definition of phase for self-sustained oscillators is the fact that the phase difference corresponding to one oscillation period T is 2π, i.e. every time t increases by T, $\Phi(t)$ increases by 2π:

$$\Phi(t+T) - \Phi(t) = 2\pi.$$

To insure this property, Pikovsky / Rosenblum / Kurths (2001 [225], page 34) advance the following definition of the phase $\Phi(t)$ of a self-sustained oscillator:

$$\Phi(t) = \Phi_0 + 2\pi \frac{t - t_0}{T},$$

where Φ_0 is a constant specifying phase at time t_0, and T is the period of the self-sustained oscillation in the absence of external influences on the oscillator.

Remembering the odd shapes of the limit cycle associated with the van der Pol oscillator displayed in Figures 4.12 and 4.15, we will not expect the relation of values of $\Phi(t)$ to points on the limit cycle to be straightforward and easily seen; nevertheless, adding (an integer multiple of) 2π to the value of $\Phi(t)$ corresponding to a point on the limit cycle will produce another phase value corresponding to the same point as long as the oscillator is running free.[1]

The concept of phase is central to the investigation of synchronization because of the following observation (Pikovsky / Rosenblum / Kurths 2001 [225], Section 2.2): If a weak force is exerted on the oscillator, it will be slightly offset from its undisturbed course, i.e. the trajectory will deviate slightly from the limit cycle. Due to the attracting property of the limit cycle (i.e. to the amplitude regulating property of the oscillator), the trajectory will quickly approach the limit cycle again, leaving the amplitude and general shape of the oscillation essentially intact, but preserving eventually introduced time shifts because of the frequency stability of the oscillator. Thus, in the theoretical treatment small perturbations of the oscillator are considered as introducing phase shifts, moving the oscillator's state along the limit cycle, but leaving amplitude and shape of the oscillatory pattern unchanged[2].

Synchronization in this context is conceptualized as an effect of regularly applying small perturbations until a stable phase relationship between a perturbing system and the perturbed oscillator is established; mathematical models of synchronization e.g. seek to specify appropriate phase changes to be induced in the perturbed oscillator to achieve synchronous behavior of the two systems.

As an example, in the attentional theory of Large and Jones as presented by Jones (2004 [141], pages 52–60), the temporal development of the phase relationship between an external (disturbing) pulse and an internal (attentional) oscillator is described. Here (see ibid., Equation (1), page 56), a phase correction sinusoidally related to the phase difference between external and internal pulse is entered, increasing internal phase if the internal pulse is lagging behind and decreasing internal phase if the internal pulse is running ahead.[3]

Similar basic considerations lie at the heart of some of the mathematical models presented in Desain / Windsor (eds., 2000 [68]).

To sum up, synchronization as discussed in this section refers to the temporal alignment of repetitive, regular (periodic), self-sustained processes by weak mutual interaction.

---

[1]The use of the term phase introduced here need to be distinguished from the use as in "phase space": In "phase space", phase is used to refer to the momentary state of the system as described by the system variables and their first derivatives; in the sense of the definition given here, phase appears as a new variable which in combination with a mapping to a limit cycle in phase space can be utilized to indicate the system's state.

[2]In the words of Pikovsky / Rosenblum / Kurths 2001 [225], page 32: "Amplitude is stable, phase is free."

[3]A problem not addressed by Jones, but shown to be of some importance in the discussion of the drumming task of the robot Nico (see above page 111) concerns the mechanism to determine the phase difference, especially if the theory is put to the test in tapping tasks. The human capability to detect and discern the effects of one's own actions is apparently taken for granted.

Some further remarks are in place here:

1. The basic assumptions entering the theoretical treatment of synchronization rule out resonance as a synchronization phenomenon (see Pikovsky / Rosenblum / Kurths 2001 [225], page 15). In a resonating system, oscillation is sustained by an external driving force, and there will be no limit cycle associated with the system that the trajectory can return to after a disturbance. Thus, the idea of phase and of moving the system's state along the limit cycle will not be applicable to a resonating system. Taking up again the example of the harmonic oscillator, in the case of positive damping a slight disturbance will lead to a small deviation of the trajectory eventually entering into a different inward spiral, and in the case of no damping a disturbance will give rise to a new neutrally stable closed trajectory.

2. Pikovsky / Rosenblum / Kurths (ibid.) point out that the mere fact of synchronous variation of variables can not necessarily be interpreted as evidence of synchronization. The case is illustrated by the example of the hare-lynx cycle (ibid., page 16). In a natural setting, the numbers of hares and lynxes are observed to vary with a common period. However, it is argued that if hares and lynxes were separated, no oscillations would occur. I.e., the co-oscillations of animal numbers are interpreted as a result of the interaction of the two populations forming a common oscillating system. On the one hand, this example can be taken as just another illustration of the requirement of weakly coupled, self-sustained oscillating systems. On the other hand we should be cautioned against an uncritical adoption of results from the theoretical discussion of synchronization in interactional contexts *giving rise to* behavioral patterns that are temporally aligned. In other words, the relation of tapping / drumming tasks to temporal patterns observed in social interaction may after all need careful investigation, because the implicit assumptions underlying the models employed may not hold.

The question of periodicity will be taken up in the next section.

8.2 Synchronization without Periodicity

In order to occur simultaneously and to remain temporally aligned, processes obviously do not need to be periodic, even in contexts characterized by a clearly regular temporal structure. Strogatz (2003 [263], page 184) mentions the example of the (first) violins of an orchestra entering at the same time and remaining in synchrony while playing. Generally, the passages played and the movements executed will not be periodic, but all first violins will be performing rather similar behaviors. Extending the example beyond the violin section, the picture becomes more complicated: Trombones, flutes, and double basses will definitely show different behaviors, not necessarily playing all at the same time, and yet they are able to adjust their activities to a common temporal scheme. A theoretical approach founded on the assumption of weakly interacting periodic processes (associated with limit cycles) will not be expected to account for these phenomena without major modifications.

Even for a rather simple drumming task, it has been shown (Crick / Munz / Scassellati 2006 [63]) that in addition to a timing mechanism that may be based on self-sustained oscillators the capability to predict the duration of one's actions and the timing of the desired effects is required. In less restricted musical contexts, requirements become more complicated, including among others:

- the choice, temporal scaling, and timing of complex movement patterns,
- the prediction of points in time limiting the execution (start / termination) of behaviors, in particular:
- the anticipation of changes in the timing pattern based on external signals and expectation shaped by previous experience, which appears to go beyond the oscillator approach, and again
- the ability to recognize one's own actions and to adjust them to the context.

The problem of timing one's own actions has been addressed with regard to questions of motor control involving forward models (e.g. Sun / Scassellati 2004 [266]) and the notion of emulating sensory response based on motor control commands (e.g. Holland 2004 [121] and further references given there).

Different problems arise concerning the prediction of relevant points in time from external signals. In well structured (and well rehearsed) contexts such as a classical musical performance, the recognition of (optical or acoustical) temporal patterns will play a role, but even in less ordered situations as described by Kendon 1970 [153] or Condon / Ogston 1971 [62], close temporal alignment with respect to features such as change of movement direction without obvious cues in advance is observed. As an illustrative example, Figure 8.2 shows two screen shots from Kozima and Michalowski's movie *Keepon dancing to Spoon's "Don't You Evah"*[4]. The top picture is taken from a sequence showing both actors in what can be called parallel action: Both heads start moving at the same time in the same direction, change direction simultaneously, and come to rest again at the same time (at least, that is the impression gained at normal viewing speed). In the bottom panel, a picture taken from a sequence showing an instance of mirroring is displayed: Starting from different gaze directions, the heads begin to turn towards each other, the robot head lagging behind the human head, and again come to rest at the same time.[5]

[4] `http://www.youtube.com/watch?v=nPdP1jBfxzo&mode=user&search=`

[5] The movie appears to be carefully choreographed to the point of caricature to illustrate various issues coming up in the observation of (human-robot) interaction.

Figure 8.2: Synchronous movement.
Top: Parallel movement.
Bottom: Mirroring.

Skipping further details, there will be at least three broad aspects of synchronization to be taken into account:

- the temporal alignment of independent regular (periodic) processes
- the temporal alignment of independent processes without obvious regularity (periodicity)
- the generation of regularly varying processes as a result of interaction.

Clearly, an integration of the considerations presented here with recent research on mirror neurons and the mirror system is indicated. For a discussion of the mirror system with regard to recognizing others' intentions (empathy, simulation theory) see e.g. Arbib / Fellous 2004 [14], Arbib 2005b [13], Jeannerod 2005 [132]. The relation to rhythmic interaction and synchronization in musical contexts will be taken up at the upcoming (at the time of writing) Workshop on "Musical Movement and Synchronization" (May 3–4, 2008) and Symposium "Rhythmic Coordination in Dyads" (May 5, 2008), organized at the Max-Planck-Institute for Human Cognitive and Brain Sciences, Leipzig.

8.3 Observing Synchronization

An observational experiment addressing synchronization in an interactional situation comprising robot movements, music, and children's rhythmic behavior is reported by Michalowski / Sabanovic / Kozima 2007 [187]. In a rather free setting at an exhibition, children interacting with the "creature-like robot Keepon" (ibid., page 90) were recorded on video; the only instruction given consisted of a sign encouraging to "dance with Keepon!" (ibid., page 91)[6].

Keepon's movement patterns are governed by four degrees of freedom (see Figure 8.3): forward–backward ("nodding") and left–right ("rocking") excursions, turning around the vertical axis ("panning") and contraction along the vertical axis ("bobbing"). Face-like features are formed by two cameras for the eyes and a microphone as a nose[7]

In the experiment, Keepon was set up to derive temporal cues from visual input gained from an external camera covering Keepon's vicinity and adjust movement patterns to these cues. In addition, music was played in the area hosting Keepon. Temporal coordination between Keepon's movements and the rhythm of the music could arise due to rhythmic movement taken in by the camera.

[6] The interactional qualities of Keepon were acknowledged by the award of the *Robots at Play Prize* `http://www.robotsatplay.dk/index_eng.html` (last checked 2008-04-29).

[7] In the experiment described, on-robot cameras and microphone were not used.

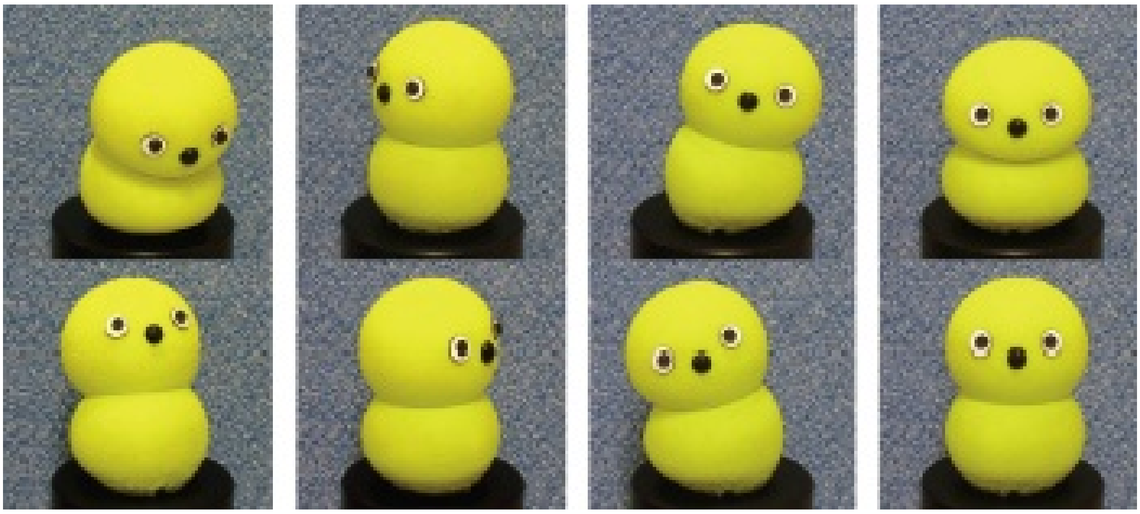


Figure 8.3: Degrees of freedom for Keepon's movements. Taken from Michalowski / Sabanovic / Kozima 2007 [187], page 91.

In an initial analysis (Michalowski / Sabanovic / Kozima 2007 [187], page 92), video segments showing children interacting with Keepon were coded according to a scheme consisting essentially of two questions: For each instance of interaction, the coder noted whether Keepon was initially in synchrony with the music and whether the interacting children (single or in groups) started to dance. The data thus obtained was subjected to a chi-square test to show that a significantly higher proportion of children started to dance when Keepon was in synchrony with the music.

This result appears to require careful interpretation. Michalowski / Sabanovic / Kozima intend to investigate rhythmic behavior as an aspect of social interaction, referring to a theoretical background of interactional synchrony (ibid., page 90). The bulk of evidence cited in favor of the role of synchronization in interactional processes by e.g. Hatfield / Cacioppo / Rapson 1994 [114], Kendon 1970 [153], Condon / Ogston 1971 [62], and Revel / Nadel 2007 [232] derives from the investigation of dyadic interactions such as infant–caretaker, speaker–listener, or husband–wife[8], possibly extended to situations such as illustrated in Figure 8.4 for the case of speaker–listener relationships; as argued above, in these contexts rhythmicity (and synchrony) may figure as an *effect* of interaction, rather than as a *cause*. In the experiment described by Michalowski / Sabanovic / Kozima, the problem addressed seems to be different from, though probably related to, the case of interactional synchrony, addressing the attractive power of observed

[8]As far as I am aware, the dominance of dyadic interactions holds in the investigation of the mirror system, too. Some pointers to work on micro-coordination in groups can be found in Collins 1988 [61], page 202.

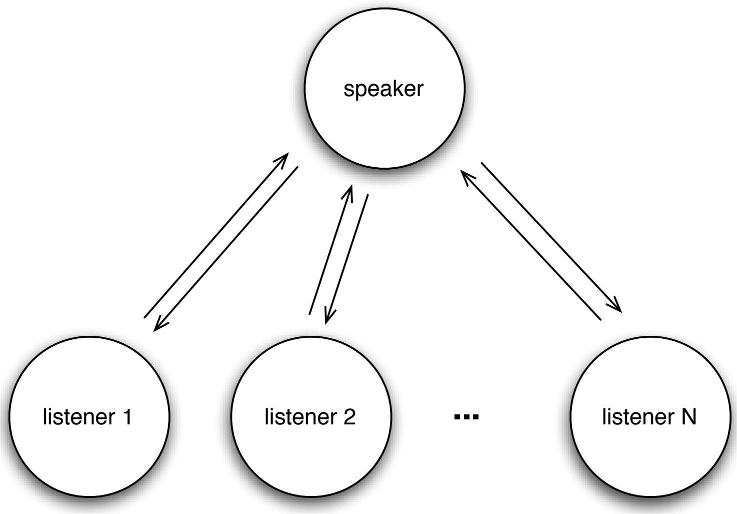


Figure 8.4: Interaction(s) between one speaker and N listeners.

(here: by the child) rhythmic coordination to join into an existing (interactional) process (here: robot-music rhythmic coordination).

In a more detailed analysis, Michalowski / Sabanovic / Kozima (2007 [187], pages 92–94) employed a more refined coding scheme, taking into account different types of behavior (including touching the robot) and information about the interacting children such as approximate age and gender. Moreover, coding was performed on a time base, opening up the possibility of sequential analysis. Although a primary interest in "the development of interaction between the children and the robot" (ibid. page 93) was expressed, however, the analyses presented are of a non-sequential type evaluating "time-budgets" (Bakeman / Gottman 1997 [21], page 7), i.e. the durations of observed behaviors under certain conditions were compared. In this type of analysis, the temporal ordering of observed behaviors is not reflected, leading to a loss of information about the time course of behavior. As an example, changing the order of the frames in Figure 8.5 might be indicative of a completely different course of interaction than the order presented here. In addition, aspects of micro-timing, deemed to be important in the study of interactional synchrony, are not addressed in these analyses. Of course, sequential analyses taking these aspects into account will significantly increase the demand on resources needed for acquisition and evaluation of the data.

A long term observational study of small children interacting with a humanoid (QRIO) dancing robot was conducted by Tanaka and co-workers (e.g. [274, 270, 273, 271, 272]). In a set of repeated sessions, the children's behavior was videotaped under three different conditions: In the first condition, the QRIO robot was set to perform a pre-programmed dance sequence to a certain

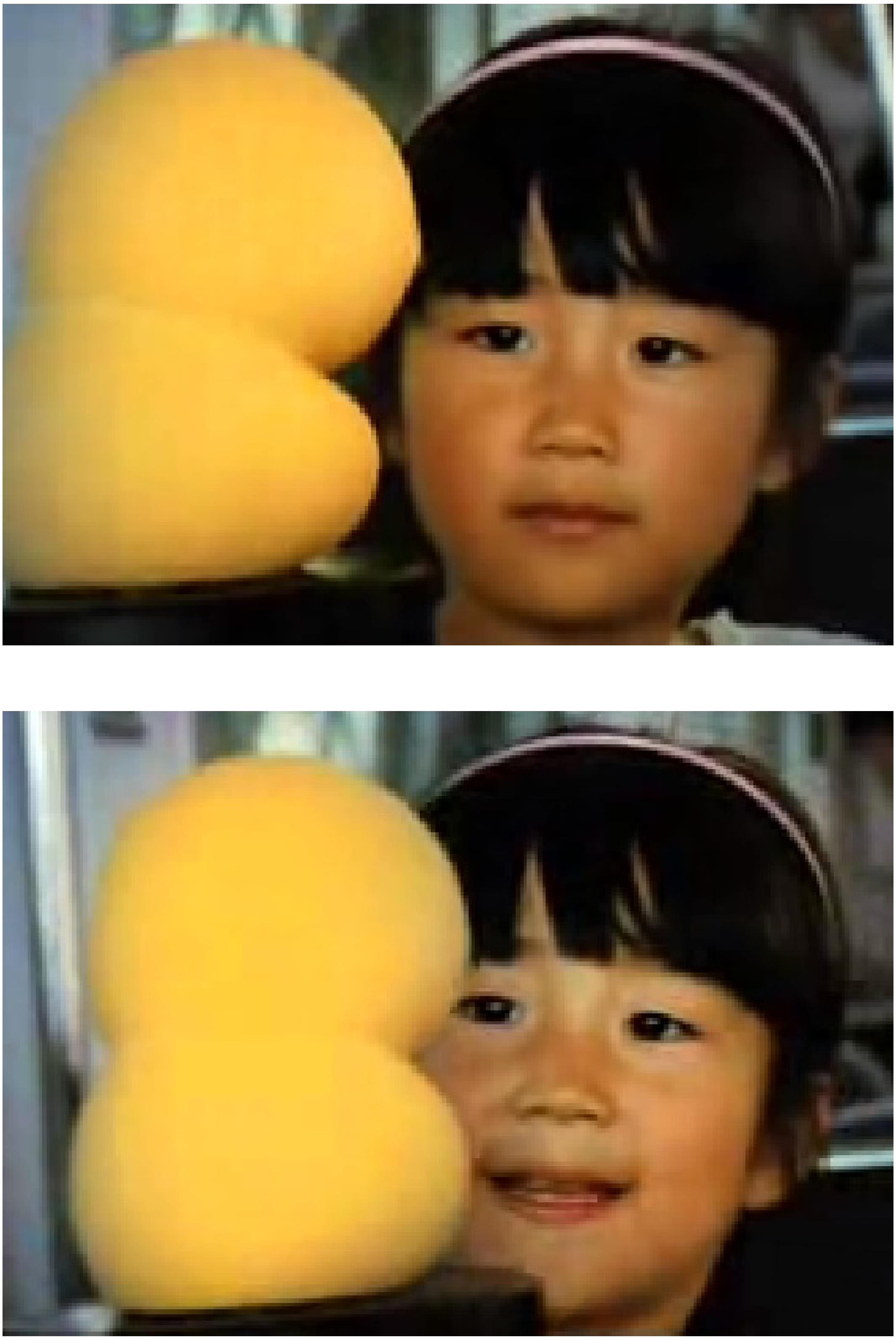


Figure 8.5: Facial display of a child interacting with Keepon, illustrating the importance of sequential analysis. Screen shots from Kozima and Michalowski's movie *Keepon dancing to Spoon's "Don't You Evah"*; for a link see Footnote 4.

song; in the second condition, the same song was played, bat the robot was set to an interactive dancing mode, responding to movement in the environment. As a third condition, intervals of playing the song without robot movements were introduced because the experimenters suspected "that the power of music was so strong that it was unclear whether QRIO had an effect of attracting children compared with the music itself" (Tanaka et al. 2006 [273], page 4 of the pdf document). The obtained video material was coded continuously by 5 coders judging on a 5 point scale whether there were "currently [...] examples of good child-robot interactions" (ibid., Figure 9). – Although a more detailed analysis of goals, experimental setup, evaluation, and results is expected to give valuable hints for the conduct of research on human-robot interaction mediated by music and dance (more generally, sound and movement), this will be left out at this point because no direct pertinence to the problem of synchronization is apparent.

A different perspective on synchronization in combination with the notions of turn-taking and entrainment will be taken in the following section.

8.4 Project Proposal: The Logic of Musical Entrainment and Interaction

The text reproduced in the following was originally prepared as a proposal for an individual project (IP), part of the collaborative research project (CRP) proposal *The Emotional and Cognitive Logic of Musical Interaction*, that was submitted to the EUROCORES LogiCCC program of the European Science Foundation (ESF) in the fall 2007. The CRP was not accepted for funding by the ESF because the aspect of developing logical concepts was not regarded to be sufficiently strong. Nevertheless, the proposal may still present ideas relevant for future work, including resources / expertise required.

The text of the proposal was prepared by U. Seifert, L. Schmidt, J. Kim, and S. Chang.

8.4.1 Text of the Proposal

Principal Investigator (PI): Uwe Seifert / Systematic Musicology, Cologne University

Abstract
Within the framework of cognitive science of music, this IP aims to investigate the logic of synchronization in musical entrainment and of turn-taking in musical interaction. Robots are used as modeling tools to study and test the temporal logical mechanisms underlying music cognition and musical behavior. This approach to music cognition is supplemented by empirical studies of human-robot interaction. The logical study of synchronization and turn-taking is based on a dynamic systems perspective combined with dynamic logics.

1.1 Aims and Objectives
A framework for music research as a science of “mind” in the methodological paradigm of cognitive science has been developed by our group (e. g. Seifert 1993 [248]; Schmidt / Seifert 2006 [244]; Schmidt 2007 [243]). This framework is theoretically based on the theory of formal systems (e. g. Enderton 2001 [80]), computability theory (e. g. Boolos / Jeffrey 2002 [39]) or equivalently the logical theory of automata (e.g. Nelson 1968 [199]). For practical music research we consider humans as well as computational systems as agents (Genesereth / Nilsson 1986 [93]; Russell / Norvig 2003 [238]; Nilsson 1998 [202]). Artificial agents are used to investigate the logic of the human mind especially mechanisms and processes underlying cognition, perception, volition, emotion, and (social) interaction (Nelson 1989 [200]; Fellous / Arbib 2005 [85]; Frijda 1996 [90]). Trevarthen (2002 [287]) criticizes the use of computational systems as modeling tools in psychology and cognitive science as a misguided mechanistic approach. Instead, he

considers adaptive and creative organisms as complex dynamic systems whose intelligence is grounded in body movements Ð especially the 'musicality' of human movement. A project on musical entrainment and interaction will have to cope with this challenge. The concepts of entrainment and interaction both seem to be essential for musical learning and understanding as well as social learning and understanding in general (Nehaniv / Dautenhahn 2007 [197]). Although intimately related these concepts will initially be investigated separately (see figure 8.6): Musical entrainment, which provides the basis for the communication of musical meaning and emotion (Clayton / Sager / Will 2005 [60]), is essentially based on synchronization, e. g. the adaptation of body movements of dancers to each other and to music (Michalowski / Sabanovic / Kozima 2007 [187]).

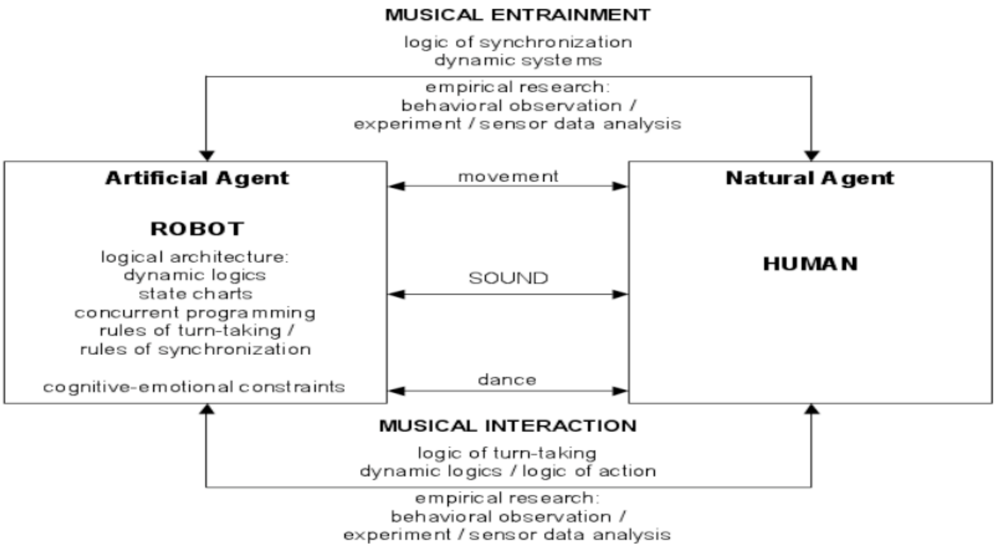


Figure 8.6: Entrainment

One main aspect of social musical interaction, which is relevant for music making (Crick / Munz / Scassellati 2006 [63]; Weinberg / Driscoll 2006 [309]; Leman 2008 [172]), is turn-taking, i. e. the mutual exchange of "question-answering". The objective of our study is a) to understand the key logical mechanisms of coupled dynamics that underlie synchronization in musical entrainment and b) to elicit the logical rules underlying turn-taking behavior in musical interaction with the ultimate aim to develop a logical architecture for understanding musical behavior and music cognition. The logic of turn-taking, in general, has mainly been studied within a logic of action from a philosophical and linguistic point of view in form of abstract static models (Lenk 1980 [174]). But human actions, especially turn-taking, are events in time and a logic of action has to be connected with a logic of time or processes (e. g. Dörner 2002 [71]) or dynamic logics. The study of musical interaction and entrainment with reactive systems seems an appropriate way to test the potential of existing dynamic or temporal logics (Harel 2004 [111]; Harel / Politi 1998 [112]; Manna / Pnueli 1991 [182]). Therefore, our principal aim is to develop and implement a logical architecture for turn-taking and synchronization. To identify logically relevant cues for formulating rules of turn-taking a category system for observational studies will be developed. To elucidate the logic of entrainment we study synchronization within the framework of dynamic systems resorting specifically to a formulation by Beer (1995a,b [27, 28]). For this very general framework to become applicable, concrete sets of variables / parameters as well as ways of coupling have to be specified. Therefore, another aim of this approach is to identify empirically possible parameters in the domain of musical rhythm and rhythmic movements.

1.2 Methodology The dynamic systems perspective serves as a means to analyze and specify the logical structuring of asynchronous concurrent processes basic to musical entrainment and interaction. As temporal aspects play a vital role for the coordination of these processes, the development and implementation of the system architecture will have to rely on concepts of dynamic logics, utilizing the state chart approach for modeling reactive systems (Harel 2004 [111]; Harel / Politi 1998 [112]; Manna / Pnueli 1991 [182]) and integrating techniques and ideas such as schemas from the field of behavior-based robotics (Arkin 1998 [15], Arbib 2004 [12]). The methodology for an empirical investigation of human-robot interaction, which is currently under development (e.g. Dautenhahn 2007a [66], 2007b [67]; Woods et al. 2006 [319]; Kooijmans et al. 2006 [158]), takes up ideas on observational and statistical methods as developed in social science especially sequential analysis and will have to be adapted to the specific situation encountered in artistic contexts. More specifically, experiments and observations on human-robot interaction in "natural" environments such as the ANIMAX multimedia theater (located near Cologne) will be carried out in the process of implementing and testing the logical architecture and developing rules for musical entrainment and interaction. A relevant category system for observational studies of musical human-robot interaction in new media art contexts has to be developed. Observer training and observer calibration using different measures of observer agreement will be used for preparing the data collection. The collection of data will be based on observer protocols from direct observations as well as video recordings and the registration of robotic and human sensor data. Sequential analysis of behavior will supplement the behavioral studies (e. g. Bakeman / Gottman 1997 [21]). The evaluation of the observations in connection with sequential analysis will be used for extracting relevant parameters and explicit logical rules of musical interaction and entrainment. These rules and parameters, which on the one hand pertain to turn-taking and synchronization in human-robot interaction but may also concern the attunement of the robot to external (acoustic / visual) stimuli, are in turn implemented on the robots and re-evaluated in the setting described above as well as related to recent empirical findings on entrainment and synchronization to rhythmic cues in music.

1.3 Work plan and deliverables/milestones Year one: We start by setting up modeling and analysis software as well as the robot systems. In parallel familiarity with the theoretical aspects of human-robot interaction will be enhanced. A deeper theoretical knowledge of behavior-based or reactive systems, their concurrent programming, dynamic logics and logic of action is acquired. At the same time pretests for observational studies are performed in connection with the development of observational categories and observer training. The search for relevant movements parameters for dynamic interaction modeling of entrainment is pursued. Milestones are a conceptual and computational working environment, the definition of a preliminary category system for the observational studies on turn-taking, and the identification of logically relevant cues for the study of synchronization.

Year two: Observational categories are refined leading to the definition of a category system for observational studies of human-robot musical interaction in connection with a further observer calibration. A next milestone, based on the logical and empirical studies of the first year, is a tentative specification of logical rules for turn-taking in musical interaction as well as logical descriptions for synchronization in musical interaction. A further step is the provisional implementation of these rules and descriptions in order to test them in observational studies on turn-taking in human-robot interaction and experiments on synchronization.

Year three: Logical, computational, and empirical research are theoretically tied together. As a milestone a definite formulation of logical rules of interaction and synchronization is put forth. Insights into the mutual dependence of turn-taking and synchronization for musical entrainment and interaction are formulated with the help of the logical framework. An outline for a logical architecture for musical entrainment and interaction as well as a methodology for music research integrating logical, computational, and empirical research is proposed as a final milestone.

1.4 Justification for budget items For this IP a PostDoc researcher, a PhD student and a programmer/technician will be needed. Requirements to be covered by the PostDoc researcher are familiarity with temporal and dynamic logics, theory of concurrent programming and reactive systems as well as a solid background in computer and cognitive science. An expertise in empirical research methods and theory is necessary for the PhD student. The tasks of the programmer/technician consist of concurrent (robot) programming, solving mechatronic problems, and hardware setup. A further student assistant is needed for technical and organizational support. For empirical research on musical interaction and entrainment, robots extended with sensors and workstations equipped with software are indispensable.

Appendix A

"Robots can't ..."?

The possibility to realize cognitive processes in computational devices is frequently debated with respect to properties found in living organisms but not in machines. As examples, we may refer to the discussion of organismoid and organismic embodiment by Ziemke 2001 [328] (see Section 3.2) or Trevarthen's (2002 [287]) remarks on the inability of robots to sing, dance etc.; an extreme position equating life and cognition ("L=C") is put forward by Heschl 1990 [116].

Reasoning appears to run along the following lines:
Some aspect of cognitive processes as observed in humans (or other living) beings is argued to be tied to one of the defining properties of living beings, say one of the criteria listed by Boden 1996 [36], and because computers, robots, or machines in general do not possess the property pointed out, they will not be able to exhibit the cognitive process / capability in question.

For a first attempt to understand the argument involved, we will introduce some abbreviations: Let $L(x)$ stand for "entity x possesses property L specific of living beings" and $C(x)$ for "entity x can exhibit a certain cognitive capability C".

The line of reasoning above may be interpreted as claiming the possession of L for an entity x, i.e. the truth of $L(x)$, to be a necessary condition for x to be able to exhibit capability C and thus for $C(x)$ to be true, or shortly: $C(x) \rightarrow L(x)$. But then, the statement that $C(x)$ cannot be true if $L(x)$ is not true, i.e. $\neg L(x) \rightarrow \neg C(x)$ is tautologically equivalent to the original claim by the law of contraposition – but nothing has been added to support the claim in the first place. As long as it cannot be shown that capability C can only be realized in one way, such claims remain persuasive but unconvincing.

Interpreting the reasoning to claim $L(x)$ to be a sufficient condition of $C(x)$, i.e. $L(x) \rightarrow C(x)$, which may be more appropriate to scientific practice, does not improve the situation: The falsity of $L(x)$ does not say anything about the truth of $C(x)$ except by the fallacy of denying the antecedent. Another problem is encountered if C refers to any species-specific capability: the lack of C in any other species would render property L to be a specific property of one species, not of any living being, tying capability C not to the general concept of living being but to some specific realization.

More effort needs to be invested in the reconstruction of possible argumentation.

Appendix B

Mathematical Supplements

B.1 Critical Damping, Initial Conditions

General Solution:

$$\vec{X}(t) = A \begin{pmatrix} e^{-\beta t} \\ -\beta e^{-\beta t} \end{pmatrix} + B \begin{pmatrix} te^{-\beta t} \\ (1-\beta t)e^{-\beta t} \end{pmatrix}$$

B.1.1 Position 1, Velocity 1

$$\vec{X}(0) = \begin{pmatrix} 1 \\ 1 \end{pmatrix} = A \begin{pmatrix} 1 \\ -\beta \end{pmatrix} + B \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Therefore,

$$\begin{aligned} A &= 1 \\ 1 &= -\beta + B \end{aligned}$$

$$B = 1 + \beta$$

$$\vec{X}(t) = \begin{pmatrix} e^{-\beta t} \\ -\beta e^{-\beta t} \end{pmatrix} + (1+\beta) \begin{pmatrix} te^{-\beta t} \\ (1-\beta t)e^{-\beta t} \end{pmatrix}$$

B.1.2 Position 1, Velocity -1

$$\vec{X}(0) = \begin{pmatrix} 1 \\ -1 \end{pmatrix} = A \begin{pmatrix} 1 \\ -\beta \end{pmatrix} + B \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Therefore,

$$\begin{aligned} A &= 1 \\ -1 &= -\beta + B \end{aligned}$$

$$B = -1 + \beta$$

$$\vec{X}(t) = \begin{pmatrix} e^{-\beta t} \\ -\beta e^{-\beta t} \end{pmatrix} + (-1 + \beta) \begin{pmatrix} te^{-\beta t} \\ (1 - \beta t)e^{-\beta t} \end{pmatrix}$$

B.1.3 Position -1, Velocity -1

$$\vec{X}(0) = \begin{pmatrix} -1 \\ -1 \end{pmatrix} = A \begin{pmatrix} 1 \\ -\beta \end{pmatrix} + B \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Therefore,

$$\begin{aligned} A &= -1 \\ -1 &= -\beta + B \end{aligned}$$

$$B = -(1 + \beta)$$

$$\vec{X}(t) = -\begin{pmatrix} e^{-\beta t} \\ -\beta e^{-\beta t} \end{pmatrix} - (1 + \beta) \begin{pmatrix} te^{-\beta t} \\ (1 - \beta t)e^{-\beta t} \end{pmatrix}$$

B.1.4 Position -1, Velocity 1

$$\vec{X}(0) = \begin{pmatrix} -1 \\ 1 \end{pmatrix} = A \begin{pmatrix} 1 \\ -\beta \end{pmatrix} + B \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Therefore,

$$\begin{aligned} A &= -1 \\ 1 &= \beta + B \end{aligned}$$

$$B = 1 - \beta$$

$$\vec{X}(t) = \begin{pmatrix} -e^{-\beta t} \\ \beta e^{-\beta t} \end{pmatrix} + (1-\beta)\begin{pmatrix} te^{-\beta t} \\ (1-\beta t)e^{-\beta t} \end{pmatrix}$$

B.2 Strong Damping, Initial Conditions

General Solution:

$$\vec{X}(t) = A\begin{pmatrix} e^{-(\beta+\gamma)t} \\ -(\beta+\gamma)e^{-(\beta+\gamma)t} \end{pmatrix} + B\begin{pmatrix} e^{-(\beta-\gamma)t} \\ -(\beta-\gamma)e^{-(\beta-\gamma)t} \end{pmatrix}$$

where $\gamma = \sqrt{\beta^2 - \omega_0^2}$

B.2.1 Position 1, Velocity 0

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix} = A\begin{pmatrix} 1 \\ -(\beta+\gamma) \end{pmatrix} + B\begin{pmatrix} 1 \\ -(\beta-\gamma) \end{pmatrix}$$

Therefore

$$\begin{aligned} 1 &= A + B \\ 0 &= -(\beta+\gamma)A - (\beta-\gamma)B \end{aligned}$$

$$A = 1 - B$$

$$\begin{aligned} (B-1)(\beta+\gamma) - B(\beta-\gamma) &= 0 \\ B(\beta+\gamma-\beta+\gamma) &= \beta+\gamma \\ B &= \frac{\beta+\gamma}{2\gamma} \end{aligned}$$

$$A = \frac{2\gamma-\beta-\gamma}{2\gamma} = -\frac{\beta-\gamma}{2\gamma}$$

B.2.2 Position 1, Velocity 1

$$\begin{pmatrix} 1 \\ 1 \end{pmatrix} = A \begin{pmatrix} 1 \\ -(\beta+\gamma) \end{pmatrix} + B \begin{pmatrix} 1 \\ -(\beta-\gamma) \end{pmatrix}$$

Therefore

$$\begin{aligned} 1 &= A + B \\ 1 &= -(\beta+\gamma)A - (\beta-\gamma)B \end{aligned}$$

B.2.3 Position 1, Velocity -1

$$\begin{pmatrix} 1 \\ -1 \end{pmatrix} = A \begin{pmatrix} 1 \\ -(\beta+\gamma) \end{pmatrix} + B \begin{pmatrix} 1 \\ -(\beta-\gamma) \end{pmatrix}$$

Therefore

$$\begin{aligned} 1 &= A + B \\ -1 &= -(\beta+\gamma)A - (\beta-\gamma)B \end{aligned}$$

B.2.4 Position -1, Velocity -1

$$\begin{pmatrix} -1 \\ -1 \end{pmatrix} = A \begin{pmatrix} 1 \\ -(\beta+\gamma) \end{pmatrix} + B \begin{pmatrix} 1 \\ -(\beta-\gamma) \end{pmatrix}$$

Therefore

$$\begin{aligned} -1 &= A + B \\ -1 &= -(\beta+\gamma)A - (\beta-\gamma)B \end{aligned}$$

B.2.5 Position -1, Velocity 1

$$\begin{pmatrix} -1 \\ 1 \end{pmatrix} = A \begin{pmatrix} 1 \\ -(\beta+\gamma) \end{pmatrix} + B \begin{pmatrix} 1 \\ -(\beta-\gamma) \end{pmatrix}$$

Therefore

$$\begin{aligned} -1 &= A + B \\ 1 &= -(\beta+\gamma)A - (\beta-\gamma)B \end{aligned}$$

Appendix C

Khepera III: Driving Circles

The movement patterns developed by Burger (2007 [45]) for the LEGO NXT robot M[ε]X contained as one major component circular segments. Originally, the patterns were implemented in the graphical programming environment NXT-G provided by LEGO.

In preparation for an observational experiment performed at the International Summer School in Systematic Musicology (ISSSM) 2007 in Ghent, the programs were re-written in the textual programming language Not eXactly C (NXC)[1][2].

One of the questions addressed in this experiment concerned the impact of the robot's shape on the outcome of observational data. It appears meaningful to re-write the programs once again in the language C for an implementation for the Khepera III robot, in order to include the Khepera III in an extended comparison.

As a building block we will present here a short program implementing movement of the Khepera III on a circle with a given radius. Since the direction of the robot's movement is determined by the speed settings of the two motor controllers, it is necessary to provide a formula calculating the appropriate relationship between motor speeds for a desired circle radius. The formula is derived from simple geometrical considerations, assuming ideal contact between robot wheels and floor, as follows (for an illustration see Figure C.1):

The desired radius r_1 of the circle is specified as the distance from the center of the circle to the wheel closer to the center (henceforth inner wheel). The distance r_2 to the farther wheel will then be

$$r_2 = r_1 + d,$$

where d is the distance between the two robot wheels, which amounts to $d = 13$ cm.

[1] `http://bricxcc.sourceforge.net/nbc/`

[2] NXC programming was done jointly using a collaborative text editing tool by B. Buch, B. Burger, S. Chang, J. Kim, J.A. Otto, L. Schmidt, and U. Seifert.

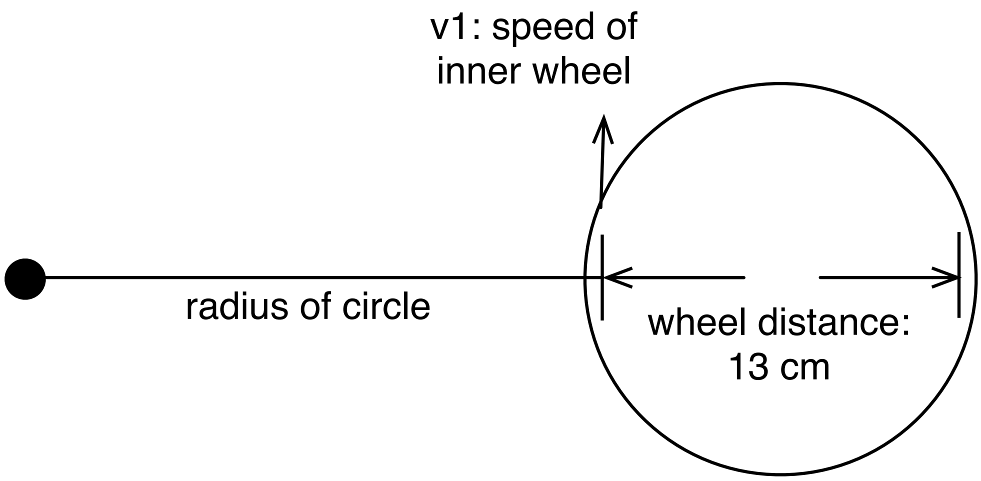


Figure C.1: Driving a circle with a Khepera III robot: The radius of the circle is measured as the distance from the center to the inner wheel, and the speed v_1 is measured for this wheel.

For the robot to stay on the circular track, both wheels will have to move around the center of the circle with a common angular velocity ω. Therefore, the speeds v_1 of the inner wheel and v_2 of the outer wheel will be

$$\begin{aligned} v_1 &= \omega r_1 \\ v_2 &= \omega r_2 \end{aligned}$$

Calculating the ratio of the speeds will cancel out the common factor ω:

$$\frac{v_2}{v_1} = \frac{r_2}{r_1} = \frac{r_1 + d}{r_1} = 1 + \frac{d}{r_1}.$$

Thus, specifying the inner radius r_1 of the circle and the speed v_1 of the inner wheel, the appropriate speed v_2 for the outer wheel to keep the robot on the desired circular course will be

$$v_2 = v_1(1 + \frac{d}{r_1}).$$

Depending on the assignment of v_1 and v_2 to the two motors, the robot will perform a left or right circular movement with the inner radius r_1.

With the remarks on Khepera III C programming given in Chapter 7 and the comments included, the following code listing should be readable.

C.1 Code Listing k3_circle_test.c

```
/* k3_circle_test: application that lets K3 move on a circle with a given radius
 * and speed for an amount of time to be entered in milliseconds. Entering 0
 * will terminate the program.
 *
 * Default value for speed: 20
 * Default value for radius: 20 cm
 *
 * Alternative values for speed and radius can be specified as command line
 * parameters:
 * ./k3_circle_test radius motspeed
 *
 * i.e. the first parameter will be interpreted as radius, the
 * second parameter - if present - as speed value.
 */

/* korebot.h contains korebot-specific definitions */
#include <korebot/korebot.h>
#include <stdlib.h>
/* macro definitions for constants needed below */
#define K3_DIAMETER 13.0
#define K3_IR_THRESHOLD 200
#define K3_TURN_SPEED 30
#define K3_TURN_DUR 250000

/* pointers for interaction with Khepera III devices, initialized in
 * function main()
 */
static knet_dev_t * mot1;
static knet_dev_t * mot2;
static knet_dev_t * dsPic;

/* function turn() makes K3 turn on the spot:
 *
 * argument = 0: turn left
 * argument != 0: turn right
 */

void turn( int dir){
  switch(dir){
  case 0:  kmot_SetPoint( mot1, kMotRegSpeed, -K3_TURN_SPEED);
```

```
          kmot_SetPoint( mot2, kMotRegSpeed, K3_TURN_SPEED);
    break;

  default: kmot_SetPoint( mot1, kMotRegSpeed, K3_TURN_SPEED);
           kmot_SetPoint( mot2, kMotRegSpeed, -K3_TURN_SPEED);
  }

  usleep(K3_TURN_DUR);
  kmot_SetPoint( mot1, kMotRegSpeed, 0);
  kmot_SetPoint( mot2, kMotRegSpeed, 0);
}



/* function k3_move_dur()  lets K3 move for time dur specified in ms and
 * motor speeds m1 1nd m2.
 *
 * A simple form of obstacle avoidance for K3 is implemented:
 * if any of the IR sensors 3, 4, 5, or 6 returns a proximity value
 * exceeding the value defined in macro K3_IR_THRESHOLD, K3 moves
 * backward 300 ms, turns on the spot away from the obstacle, and
 * resumes the original motor speed values.
 */

void k3_move_dur(int dur, int m1, int m2){
  double irsensor3 = 0;
  double irsensor4 = 0;
  double irsensor5 = 0;
  double irsensor6 = 0;
  double elapsed = 0;
  char Buffer[512];

  /* tell motor controllers to move K3 forward */
  kmot_SetPoint( mot1, kMotRegSpeed, m1);
  kmot_SetPoint( mot2, kMotRegSpeed, m2);

  dur = 1000*dur;

  while(dur > 0){
    dur -= 50000;

    if(kh3_proximity_ir((char *)Buffer, dsPic)){
      irsensor3 = (double)(Buffer[5]|Buffer[6]<<8);
```

```
      irsensor4 = (double)(Buffer[7]|Buffer[8]<<8);
      irsensor5 = (double)(Buffer[9]|Buffer[10]<<8);
      irsensor6 = (double)(Buffer[11]|Buffer[12]<<8);
      printf("Sensor3: %4.1f, Sensor 4: %4.1f, Sensor 5: %4.1f, \
             Sensor 6: %4.1f \n", irsensor3, irsensor4, irsensor5, irsensor6);

      if(irsensor3 > K3_IR_THRESHOLD || irsensor4 > K3_IR_THRESHOLD ||\
 irsensor5 > K3_IR_THRESHOLD || irsensor6 > K3_IR_THRESHOLD){
 system("cp boing1.wav /dev/sound/dsp");
 kmot_SetPoint( mot1, kMotRegSpeed, -60);
         kmot_SetPoint( mot2, kMotRegSpeed, -60);
 usleep(500000);
 turn(irsensor3 > irsensor6);
 kmot_SetPoint( mot1, kMotRegSpeed, m1);
         kmot_SetPoint( mot2, kMotRegSpeed, m2);
      }
      else{
usleep(50000);
      }
    }
    else{
      printf("error reading proximity sensors!\n");
    }
  }

  kmot_SetPoint( mot1, kMotRegSpeed, 0);
  kmot_SetPoint( mot2, kMotRegSpeed, 0);
}


int main (int argc, char *argv[])
{
  /* definitions of default values for variables */
  int motspeed1 = 20;
  int motspeed2 = 20;
  int duration = 1000;
  double radius = 20.0;

  /* the kh3_init() routine is required */
  kh3_init();

  if(argc > 1)
    radius = atof(argv[1]);
```

```
  if(argc > 2)
    motspeed1 = atoi(argv[2]);

  printf("radius: %5.1f \n", radius);

/* open various sockets and store the handles in their respective pointers:
 * the motor controllers and the dsPic processor are addressed as independent
 * I2C devices.
 *
 * mot1 points to the left motor, mot2 to the right.
 */
dsPic = knet_open("Khepera3:dsPic", KNET_BUS_I2C, 0, NULL);
mot1  = knet_open("Khepera3:mot1",  KNET_BUS_I2C, 0, NULL);
mot2  = knet_open("Khepera3:mot2",  KNET_BUS_I2C, 0, NULL);



/* initialize motor controller 1 */
kmot_SetMode( mot1, kMotModeIdle);
kmot_SetSampleTime( mot1, 1550);
kmot_SetMargin( mot1, 6);
kmot_SetOptions( mot1, 0x0, kMotSWOptWindup | kMotSWOptStopMotorBlk |\
 kMotSWOptDirectionInv);
kmot_ResetError( mot1);
kmot_SetBlockedTime( mot1, 10);
kmot_ConfigurePID( mot1, kMotRegSpeed, 400, 0, 10);
kmot_ConfigurePID( mot1, kMotRegPos, 620, 3, 10);
kmot_SetSpeedProfile( mot1, 30, 3);


/* initialize motor controller 2 */
kmot_SetMode( mot2, kMotModeIdle);
kmot_SetSampleTime( mot2, 1550);
kmot_SetMargin( mot2, 6);
kmot_SetOptions( mot2, 0x0, kMotSWOptWindup | kMotSWOptStopMotorBlk);
kmot_ResetError( mot2);
kmot_SetBlockedTime( mot2, 10);
kmot_ConfigurePID( mot2, kMotRegSpeed, 400, 0, 10);
kmot_ConfigurePID( mot2, kMotRegPos, 620, 3, 10);
kmot_SetSpeedProfile( mot2, 30, 3);


 /* make sure that no division by 0 will occur */
```

```
 if (radius < 0.001){
   printf("Radius too small!!\n");
   exit(1);
 }

 /* calculate motspeed2 based on the choice of motspeed1:
  * K3_DIAMETER represents the approximate diameter of K3 (13.0 cm)
  * 'radius' stands for the inner radius of the circle in cm
 */

     motspeed2 = (int)(motspeed1*(1+K3_DIAMETER/radius));


     /* print momentary speed for motors */
     printf("motspeed1 %u \n", motspeed1);
     printf("motspeed2 %u \n", motspeed2);



/* infinite loop, interrupted by entering 0 */
while(1){


  printf("enter duration in ms:");
  scanf("%i", &duration);
  printf("\n \n");

  if(duration){
    printf("duration: %u \n\n", duration);


     /* turn left */
    k3_move_dur(duration, motspeed1, motspeed2);
  }
  else{
    printf("bye");
    exit(0);
  }
 }

}
```

Appendix D

Khepera III Motion Tracking Using SoftVNS

Movement control of the Khepera III robot by camera based motion tracking was implmented in the patch shown on the next page.

The patch is based on a one-to-one port – with the exception of the grid object – of the Pd patch of Figure 7.3 to the MaxMSP environment, also provided by Tobias Grewening and Ralf Baecker. Motion tracking functionality has been incorporated by use of the softVNS external objects for MaxMSP by David Rokeby[1].

Here, a region of the camera image shown on the right can be selected and tracked by the video processing objects collected in the subpatch `p softvns`. x and y positions of the tracked region are mapped to the sliders for direction and speed control of the robot.

Motion tracking components of this patch were developed by Jochen Arne Otto.

Two problems were encountered with this approach:

1. Too fast movement of the selected region could lead to losing track.
2. In more advanced versions of this patch, repeatedly the connection between robot and computer got lost. Reasons for this problem are not clear yet; first ideas concern internal timing issues and the loss of data packages during transmission due to the use of the UDP network protocol.

[1] `http://homepage.mac.com/davidrokeby/softVNS.html`

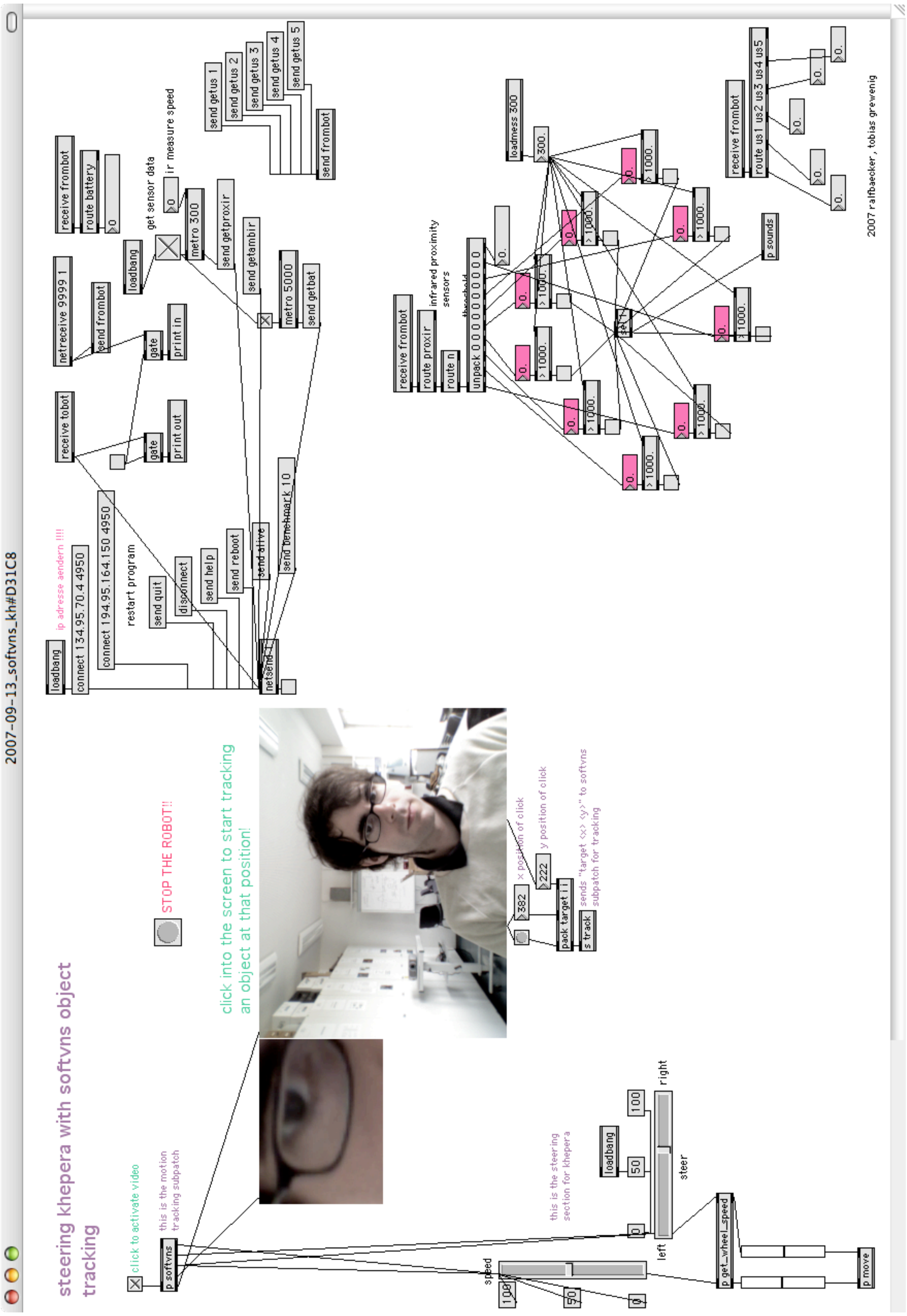


Figure D.1: MaxMSP patch using softVNS externals for movement control of the Khepera III robot by camera based motion tracking

Appendix E

Khepera III: C-Source for Network Interaction Using Pd

```
/*!
 * \file   khepera3_test.c  Khepera3 test application
 *
 * \brief
 *         This is an application that demonstrates the various khepera3
 *         commands.
 *
 *
 * \author   Arnaud Maye (K-Team SA)
 *
 * \note     Copyright (C) 2005 K-TEAM SA
 * \bug      none discovered.
 * \todo     nothing.
 * \adds     PD/UDP Ralf Baecker / Tobias Grewenig, update Lueder Schmidt
 * makes use of pdsend.c by Miller Puckette
 */

#include <korebot/korebot.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netinet/tcp.h>
```

```
#include <netdb.h>
#include <arpa/inet.h>
#define SOCKET_ERROR -1
int pdsend(int portno, char *hostname, char *msg);
#include <time.h>

#define MAXBUFFERSIZE 100
#define MYPORT 4950 // the port users will be connecting to
#define SENDPORT 9990
#define MAXBUFLEN 100
struct tm *systime;
static int quitReq = 0;
static char buf[1024];
char *rip;

/*! handle to the various khepera3 devices (knet socket, i2c mode)
 */
static knet_dev_t * dsPic;
static knet_dev_t * mot1;
static knet_dev_t * mot2;

/*--------------------------------------------------------------------*/
/*! initMot initializes then configures the motor control
 * unit.
 *
 * \return A value :
 *      - 1 if success
 *      - 0 if any error
 *
 */
int initMot(knet_dev_t *hDev)
{
  if(hDev)
  {
  kmot_SetMode( hDev , kMotModeIdle );
  kmot_SetSampleTime( hDev , 1550 );
  kmot_SetMargin( hDev , 6 );
  kmot_SetOptions( hDev , 0x0 , kMotSWOptWindup | kMotSWOptStopMotorBlk );
  kmot_ResetError( hDev );
  kmot_SetBlockedTime( hDev , 10 );
  kmot_SetLimits( hDev , kMotRegCurrent , 0 , 500 );
  kmot_SetLimits( hDev , kMotRegPos , -10000 , 10000 );
```

```
  /* PID  */
  kmot_ConfigurePID( hDev , kMotRegSpeed , 1500 , 100 , 400 );
  kmot_ConfigurePID( hDev,kMotRegPos,620,3,10);
  kmot_SetSpeedProfile(hDev,30,10);

  return 1;
  }
  else
  {
  printf("initMot error, handle cannot be null\r\n");
  return 0;
  }
}


/*--------------------------------------------------------------------*/
/*! initKH3 initialize various things in the kh3 then
 * sequentialy open the various required handle to the three i2c devices
 * on the khepera3 using knet_open from the knet.c libkorebot's modules.
 * Finaly, this function initializes then configures the motor control
 * unit.
 *
 * \return A value :
 *      - 1 if success
 *      - 0 if any error
 */


int initKH3( void )
{
  /* This is required */
  kh3_init();

  /* open various socket and store the handle in their respective pointers */
  dsPic = knet_open( "Khepera3:dsPic" , KNET_BUS_I2C , 0 , NULL );
  mot1  = knet_open( "Khepera3:mot1" , KNET_BUS_I2C , 0 , NULL );
  mot2  = knet_open( "Khepera3:mot2" , KNET_BUS_I2C , 0 , NULL );

  if(dsPic!=0)
  {
    if((mot1!=0)&&(mot2!=0))
      {
initMot(mot1);
```

```
initMot(mot2);
return 0;
      }
    else
      return -1;
  }

  return -2;

}

/*--------------------------------------------------------------------*/
/*! proxIR retrieves proximity ir measure using kb_khepera3.c library.
 */
int proxIR( int argc, char * argv[], void * data)
{
  char irdata[512];
  char Buffer[MAXBUFFERSIZE];
  if(kh3_proximity_ir((char *)Buffer, dsPic)) {

     sprintf(irdata,"proxir %c %4.4u %4.4u %4.4u %4.4u %4.4u \
                  %4.4u %4.4u %4.4u %4.4u %lu\n",
   Buffer[0], (Buffer[1] | Buffer[2]<<8), (Buffer[3] | Buffer[4]<<8),
   (Buffer[5] | Buffer[6]<<8), (Buffer[7] | Buffer[8]<<8),
   (Buffer[9] | Buffer[10]<<8), (Buffer[11] | Buffer[12]<<8),
   (Buffer[13] | Buffer[14]<<8), (Buffer[15] | Buffer[16]<<8),
   (Buffer[17] | Buffer[18]<<8),
   ((Buffer[19] | Buffer[20]<<8) | (Buffer[21] | Buffer[22]<<8)<<16));
     pdsend(SENDPORT,rip,irdata);
  } else
   printf("\r\nn, error...\r\n");
}

/*--------------------------------------------------------------------*/
/*! ambIR retrieves ambiant ir measure using kb_khepera3.c library.
 */
int ambIR( int argc, char * argv[], void * data)
{
  char Buffer[MAXBUFFERSIZE];
  char irdata[512];
  if(kh3_ambiant_ir((char *)Buffer, dsPic)) {

sprintf(irdata,"ambir %c %4.4u %4.4u %4.4u %4.4u %4.4u %4.4u\
```

```
 %4.4u %4.4u %4.4u %lu\n",
   Buffer[0], (Buffer[1] | Buffer[2]<<8), (Buffer[3] | Buffer[4]<<8),
   (Buffer[5] | Buffer[6]<<8), (Buffer[7] | Buffer[8]<<8),
   (Buffer[9] | Buffer[10]<<8), (Buffer[11] | Buffer[12]<<8),
   (Buffer[13] | Buffer[14]<<8), (Buffer[15] | Buffer[16]<<8),
   (Buffer[17] | Buffer[18]<<8),
   ((Buffer[19] | Buffer[20]<<8) | (Buffer[21] | Buffer[22]<<8)<<16));
   pdsend(SENDPORT,rip,irdata);
  } else
   printf("\r\no, error...\r\n");
}

/*--------------------------------------------------------------------*/
/*! voltageBAT retrieves the battery voltage using kb_khepera3.c library.
 */
int voltageBAT( int argc, char * argv[] , void * data )
{
  char Buffer[MAXBUFFERSIZE];
  char btxt[512];
  short argument = 0;
  if(kh3_battery_voltage((char *)Buffer, argument, dsPic)) {
                  /* argument added LS */

      sprintf(btxt,"battery %u.%u\n",
        Buffer[0], (Buffer[1] | Buffer[2]<<8), (Buffer[3] | Buffer[4]<<8),
        (Buffer[1] | Buffer[2]<<8), (Buffer[3] | Buffer[4]<<8));

        pdsend(SENDPORT,rip,btxt);
      printf ("batt string filled...\n\r");
  } else {
  printf("\r\nv, error...\r\n");
  }
}

/*--------------------------------------------------------------------*/
/*! tstampRST resets the relative time stamp using kb_khepera3.c library.
 */
int tstampRST( int argc, char * argv[], void * data)
{
  char Buffer[MAXBUFFERSIZE];

  if(kh3_reset_tstamp((char *)Buffer, dsPic))
   printf("\r\n%c\r\n", Buffer[0]);
```

```
  else
printf("\r\nz, error...\r\n");
}

/*--------------------------------------------------------------------*/
/*! revisionOS retrieves the khepera3 os version using kb_khepera3.c library.
 */
int revisionOS( int argc, char * argv[], void * data)
{
  char Buffer[MAXBUFFERSIZE]; /* buffer that handles the
                   returned data from kh3 */

  if(kh3_revision((char *)Buffer, dsPic))
   printf("\r\n%c,%4.4u,%4.4u => Version = %u, Revision = %u\r\n",
   Buffer[0], (Buffer[1] | Buffer[2]<<8), (Buffer[3] | Buffer[4]<<8),
   (Buffer[1] | Buffer[2]<<8), (Buffer[3] | Buffer[4]<<8));
  else
printf("\r\nb, error...\r\n");
}

/*--------------------------------------------------------------------*/
/*! configureOS configures various parameters inside the kh3 firmware
 *  using kb_khepera3.c library.
 *
 *  \param 1st first param (argv[1]) is the index pointing in
 *                 configuration array.
 *  \param 2nd the second param (argv[2]) is the value to store where the
 *                 index point at.
 *
 */
int configureOS( int argc, char * argv[], void * data)
{
  char Buffer[MAXBUFFERSIZE]; /* buffer that handles the returned data
                                from kh3 */
  short index; /* variable that handles index */
  short value; /* variable that handle value */

  /* Retrive the arguments from the parameter */
  index = atoi(argv[1]);
  value = atoi(argv[2]);
  printf("setcfg(%d,%d)",index,value);
  /* Configure */
  if(kh3_configure_os((char *)Buffer, index, value, dsPic) ) {
```

```
        printf("\r\n%c\r\n", Buffer[0]);
   pdsend(SENDPORT,rip,"cfg 1");

  } else {
printf("\r\nc, error...\r\n");
   pdsend(SENDPORT,rip,"cfg ");
  }
}

/*--------------------------------------------------------------------*/
/*! measureUS retrieves ultrasonic measure from a given transceiver.
 *
 *  \param 1st first param (argv[1]) is the us number to read from (1 to 5).
 *
 */
int measureUS( int argc, char * argv[], void * data)
{
  char Buffer[MAXBUFFERSIZE];
  int i;
  char ret[512];
  short usnoise; /* Noise on the given adc pin when no us is received */
  short echonbr; /* Number of echo part of this us measure */
  float usconst = 1.715;
  short usvalue; /* Variable that handle distances */
  short usampl; /* Variable than nandle amplitudes */
  short argument;   /* (re-)inserted LS*/

  argument = atoi(argv[1]); /*(re-)inserted LS*/

  usvalue= 0;
printf("Sensor Nr. %d", argument); /*changed LS*/
  if(kh3_measure_us((char *)Buffer, argument, dsPic)) /* adjusted LS*/
  {
/* Printout complete frame as received by the khepera3 */
   printf("\r\n%c", Buffer[0]);

for(i = 0; i < 22; i++)
printf(",%4.4u", (Buffer[1+i*2] | Buffer[2+i*2]<<8));
printf("\r\n");

/* We guess the echo number ( how many echo has been received from a captor ) */
echonbr = (Buffer[1] | Buffer[2]<<8);
printf("echonbr = %d\r\n", echonbr);
```

```
/* Loop as may time it is required */
for(i = 0; i < 1; i++)  /*echonbr replaced by 1 LS*/
{
/* Get the distance measure from one echo */
usvalue = (Buffer[i*8+3] | Buffer[i*8+4]<<8) * usconst; /*adjusted LS*/
usampl  = (Buffer[i*8+5] | Buffer[i*8+6]<<8); /*adjusted LS*/

/* Print out the result */
printf("Echo %d : Amplitude = %d,  Distance = %dcm.\r\n", i+1, usampl, usvalue);

}
sprintf(ret,"us%d %d\n",argument,usvalue); /*changed LS*/
pdsend(SENDPORT,rip,ret );
    return usvalue;

  }
  else
   printf("\r\ng, error...");
}

int getallUS( int argc, char * argv[], void * data)
{
  char Buffer[MAXBUFFERSIZE];
  int i;
  short usnoise; /* Noise on the given adc pin when no us is received */
  short echonbr; /* Number of echo part of this us measure */
  float usconst = 1.715;
  short usvalue; /* Variable that handle distances */
  short usampl; /* Variable than nandle amplitudes */
  short usv1,usv2,usv3,usv4,usv5;

  char usdata[512];
  usv1 = 0;
  usv2 = 0;
  usv3 = 0;
  usv4 = 0;
  usv5 = 0;

  if(kh3_measure_us((char *)Buffer,1, dsPic)) {
echonbr = (Buffer[1] | Buffer[2]<<8);
    if (echonbr >0) {
      /* usv1 = (Buffer[i*2+13] | Buffer[i*2+14]<<8) * usconst; cannot work:
```

```
                                i not initialized!! changed to: */
        usv1 = (Buffer[3] | Buffer[4]<<8) * usconst; /* in the following cases
                                changed accordingly. LS */
    }
  }

  if(kh3_measure_us((char *)Buffer,2, dsPic)) {
echonbr = (Buffer[1] | Buffer[2]<<8);
    if (echonbr >0) {
        usv2 = (Buffer[3] | Buffer[4]<<8) * usconst;
            }
  }

  if(kh3_measure_us((char *)Buffer, 3, dsPic)) {
echonbr = (Buffer[1] | Buffer[2]<<8);
    if (echonbr >0) {
        usv3 = (Buffer[3] | Buffer[4]<<8) * usconst;
    }
  }

  if(kh3_measure_us((char *)Buffer, 4, dsPic)) {
echonbr = (Buffer[1] | Buffer[2]<<8);
    if (echonbr >0) {
        usv4 = (Buffer[3] | Buffer[4]<<8) * usconst;
    }
  }

  if(kh3_measure_us((char *)Buffer, 5, dsPic)) {
echonbr = (Buffer[1] | Buffer[2]<<8);
    if (echonbr >0) {
        usv5 = (Buffer[3] | Buffer[4]<<8) * usconst;
    }
  }

  sprintf(usdata,"usall %d %d %d %d %d\n",usv1,usv2,usv3,usv4,usv5);
  pdsend(SENDPORT,rip,usdata);

}

/*-------------------------------------------------------------------*/
/*! motSpeed configures the motor controller speed in the engine
 * control unit.
 *
```

```
 *  \param 1st first param (argv[1]) is the motor1 speed.
 *  \param 2nd second param (argv[2]) is the motor2 speed.
 *
 */
int motSpeed( int argc, char *argv[], void *data)
{
  if(mot1!=0 && mot2!=0)
  {
   kmot_SetPoint( mot1 , kMotRegSpeed , -atoi(argv[1]));
   kmot_SetPoint( mot2 , kMotRegSpeed , atoi(argv[2]));
return 0;
  }
  else
return -1;
}

/*--------------------------------------------------------------------*/
/*! motMove configures the motor controller position in the
 *  engine control unit.
 *
 *  \param 1st first param (argv[1]) is the motor1 position.
 *  \param 2nd second param (argv[2]) is the motor2 position.
 *
 */
int motMove( int argc, char *argv[], void *data)
{
  if(mot1!=0 && mot2!=0)
  {
kmot_SetPoint( mot1 , kMotRegPos, atoi(argv[1]));
   kmot_SetPoint( mot2 , kMotRegPos, atoi(argv[2]));
return 0;
  }
  else
return -1;
}

/*--------------------------------------------------------------------*/
/*! motStop stops the motor  in the engine control unit.
 *
 *  \param none.
 */
int motStop( int argc, char *argv[], void *data)
{
```

```
  if(mot1!=0 && mot2!=0)
  {
kmot_SetMode( mot1 , kMotModeStopMotor );
kmot_SetMode( mot2 , kMotModeStopMotor );
return 0;
  }
  else
return -1;
}

int getBenchmark( int argc, char *argv[], void *data) {
    int i;
    char s[64];
    for(i = 0;i< atoi(argv[1]);i++) {
        sprintf(s,"benchmark bang %d\n",i);
        pdsend(SENDPORT,rip,s);
        printf("benchmark i = %d\n",i);
    }
}

int doReboot( int argc, char *argv[], void *data) {
    pdsend(SENDPORT,rip,"reboot 1\n");
    system("/sbin/reboot");
}

/*--------------------------------------------------------------------*/
/*! Quit the program.
 */
int quit( int argc , char * argv[] , void * data)
{
  quitReq = 1;
}
int alive( int argc , char * argv[] , void * data)
{
    pdsend(SENDPORT,rip,"alive 1\n");
}

int help( int argc , char * argv[] , void * data);

/*--------------------------------------------------------------------*/
/*! The command table contains:
 * command name : min number of args : max number of args : the function
 *  to call
```

```
 */
static kb_command_t cmds[] = {
 { "quit"            , 0 , 0 , quit } ,
 { "exit"            , 0 , 0 , quit } ,
 { "bye"             , 0 , 0 , quit } ,
 { "ciao_bella"      , 0 , 0 , quit } ,
 { "setcfg"          , 2 , 2 , configureOS },
 { "getrev"          , 0 , 0 , revisionOS },
 { "getbat"          , 0 , 0 , voltageBAT },
 { "rststamp"      , 0 , 0 , tstampRST },
 { "getambir"      , 0 , 0 , ambIR },
 { "getproxir"       , 0 , 0 , proxIR },
 { "getus"           , 1 , 1 , measureUS },
 { "setmotspeed"     , 2 , 2 , motSpeed },
 { "setmotmove"      , 2 , 2 , motMove },
 { "motstop"         , 0 , 0 , motStop },
 { "help"            , 0 , 0 , help } ,
 { "getallus"        , 0 , 1 , getallUS },
 { "benchmark"       , 1 , 1 , getBenchmark },
 { "reboot"          , 0 , 0 , doReboot },
 { "alive"           , 0 , 1 , alive },
 { NULL              , 0 , 0 , NULL }

};

/*--------------------------------------------------------------------*/
/*! Display a list of available commands.
 */
int help( int argc , char * argv[] , void * data)
{
  kb_command_t * scan = cmds;
  while(scan->name != NULL)
    {
      printf("%s\r\n",scan->name);
      scan++;
    }
  return 0;
}

int main( int arc, char *argv[])
{
  long c;
  int blen;
```

```
  int rsockfd;

  int tt;
  struct sockaddr_in my_addr; // my address information
  struct sockaddr_in their_addr; // connector's address information
  socklen_t addr_len;
  int numbytes;
  char sbuf[MAXBUFLEN];

  if ((rsockfd = socket(AF_INET, SOCK_DGRAM, 0)) == -1) {
    perror("socket");
    exit(1);
  }

  my_addr.sin_family = AF_INET;  // host byte order
  my_addr.sin_port = htons(MYPORT);  // short, network byte order
  my_addr.sin_addr.s_addr = INADDR_ANY; // automatically fill with my IP
  memset(&(my_addr.sin_zero), '\0', 8); // zero the rest of the struct

  their_addr.sin_family = PF_INET;
  their_addr.sin_port = htons(SENDPORT);

  if (bind(rsockfd, (struct sockaddr *)&my_addr,
   sizeof(struct sockaddr)) == -1) {

    perror("bind");
    exit(1);
  }

  addr_len = sizeof(struct sockaddr);


  printf("KCONTROL - Tobias Grewenig / Ralf Baecker 2007\n");
  printf("PD Controlled\r\n");

  if(!initKH3())
    {
      printf("KH3 Init OK...\r\n");
      c = 0;

      while (!quitReq) {

printf("%d",c++);
```

```
if ((numbytes = recvfrom(rsockfd, sbuf, MAXBUFLEN-1 , 0,
      (struct sockaddr *)&their_addr, &addr_len)) == -1) {

  perror("recvfrom");
  exit(1);
}

rip = inet_ntoa(their_addr.sin_addr);

their_addr.sin_addr.s_addr=inet_addr(inet_ntoa(their_addr.sin_addr));
sbuf[numbytes] = '\0';

if (strstr(sbuf,";")) {
  sbuf[numbytes-2] = '\0';
}
if (strlen(sbuf) > 0) {
  printf("*");
  kb_parse_command( sbuf , cmds , NULL);
  printf("\n");
}

      }
      printf("Exiting...\r\n");
    }
  else
    printf("Fatal error, unable to initialize\r\n");
  close(rsockfd);
}
```

Bibliography

[1] Abraham, Ralph H. & Shaw, Christopher D. (1982): Dynamics: The Geometry of Behavior, Part One: Periodic Behavior. Santa Cruz, CA: Aerial Press.

[2] Adorno, Theodor W. (1948 / 1978): Philosophie der neuen Musik. Frankfurt / M.: Suhrkamp.

[3] Albersheim, Gerhard (1939): Zur Psychologie der Ton- und Klangeigenschaften. Strassburg: Heitz & Co.

[4] Albersheim, Gerhard (1974): Zur Musikpsychologie. Wilhelmshaven: Heinrichshofen.

[5] Algazi, V. Ralph, Duda, Richard O., Duraiswami, Ramani, Gumerov, Nail A. & Tang, Zhihui (2002): Approximating the head-related transfer function using simple geometric models of the head and torso. In: Journal of the Acoustical Society of America 112(5): 2053 – 2064.

[6] Algazi, V. Ralph, Duda, Richard O. & Thompson, Dennis M. (2002): The Use of Head-and-Torso Models for Improved Spatial Sound Synthesis. In: 113th Convention of the Audio Engineering Society. Los Angeles, CA.

[7] Anderson, Michael L. (2003): Embodied Cognition: A field guide. In: Artificial Intelligence 149: 91 – 130.

[8] Angel, Eric J., Algazi, V. Ralph & Duda, Richard O. (2002): On the design of canonical sound localization environments. In: 113th Convention of the Audio Engineering Society. Los Angeles, CA.

[9] Anosov, D. V. & Arnold, Vladimir I., (eds.) (1988): Dynamical Systems I. vol. 1 of Encyclopedia of Mathematical Sciences. Springer.

[10] Arbib, Michael A. (1972): The Metaphorical Brain. An Introduction to Cybernetics as Artificial Intelligence and Brain Theory. New York: Wiley.

[11] Arbib, Michael A. (1989): The Metaphorical Brain 2. Neural Networks and beyond. New York: Wiley.

[12] Arbib, Michael A. (2004): Programs and Planning. In: Gregory, Richard L. (ed.): The Oxford Companion to the Mind, 2nd Edition. Oxford: Oxford University Press. 759 – 761.

[13] Arbib, Michael A. (2005b): Beware the Passionate Robot. In: Fellous, Jean-Marc & Arbib, Michael A. (eds.): Who Needs Emotions? The Brain Meets the Robot. New York: Oxford University Press. 333 – 383.

[14] Arbib, Michael A. & Fellous, Jean-Marc (2004): Emotions: from brain to robot. In: Trends in Cognitive Sciences 8(12): 554 – 561.

[15] Arkin, Ronald C. (1998): Behavior-Based Robotics. Cambridge, MA: MIT Press.

[16] Arnold, Vladimir I. & Il'yashenko, Yu. S. (1988): Ordinary Differential Equations. In: Anosov, D. V. & Arnold, Vladimir I. (eds.): Dynamical Systems I. Berlin et al.: Springer-Verlag. 1 – 148.

[17] Ashby, W. Ross (1956): An Introduction to Cybernetics. London: Chapman & Hall.

[18] Ashby, W. Ross (1960): Design for a Brain. London: Chapman & Hall. 2nd ed.

[19] Aucouturier, Jean-Julien (2008): Cheek to Chip: Dancing Robots and AI's Future. In: IEEE Intelligent Systems : 74 – 84.

[20] Aulbach, Bernd (2004): Gewöhnliche Differenzialgleichungen, 2. Auflage. Heidelberg: Spektrum Akademischer Verlag.

[21] Bakeman, Roger & Gottman, John M. (1997): Observing Interaction. An Introduction to Sequential Analysis, 2nd Edition. Cambridge: Cambridge University Press.

[22] Balzer, Wolfgang (1997): Die Wissenschaft und ihre Methoden. Grundsätze der Wissenschaftstheorie. München: Alber.

[23] Balzer, Wolfgang, Moulines, C. Ulisses & Sneed, Joseph D. (1987): An Architectonic for Science. The Structuralist Program. Dordrecht: Reidel.

[24] Bar-Cohen, Yoseph & Hanson, David (2009): The Coming Robot Revolution. Expectations and Fears About Emerging Intelligent, Humanlike Machines. New York: Springer.

[25] Barsalou, Lawrence W. (1999): Perceputal Symbol Systems. In: Behavioral and Brain Sciences 22: 577 – 660.

[26] Beek, Peter J., Peper, C. (Lieke) E. & Daffertshofer, Andreas (2000): Timekeepers versus nonlinear oscillators: how the approaches differ. In: Desain, Peter & Windsor, Luke (eds.): Rhythm Perception and Production. Lisse: Swets & Zeitlinger. 9 – 33.

[27] Beer, Randall D. (1995a): A dynamical systems perspective on agent-environment interaction. In: Artificial Intelligence 72: 173 – 215.

[28] Beer, Randall D. (1995b): Computational and Dynamical Languages for Autonomous Agents. In: Port, Robert F. & van Gelder, Timothy (eds.): Mind as Motion. Explorations in the Dynamics of Cognition. Cambridge, MA: MIT Press. 121 – 147.

[29] Beer, Randall D. (2000): Dynamical Approaches to Cognitive Science. In: Trends in Cognitive Sciences 4(3): 91 – 99.

[30] Beer, Randall D. (2003): The Dynamics of Active Categorical Perception in an Evolved Model Agent. In: Adaptive Behavior 11(4): 209 – 243.

[31] Begault, Durand R. (1994): 3-D Sound for Vitual Reality and Multimedia. Boston, MA: Academic Press.

[32] Bengtsson, Ingmar, Gabrielsson, Alf & Thorsén, Stig-Magnus (1969): Empirisk rytmforskning. En orientering om musikvetenskaplig bakgrund, utveckling och några aktuella projekt. In: Svensk tidskrift för musikforskning 51: 49 – 118.

[33] Bertalanffy, Ludwig von (1969): General System Theory. Foundations, Development, Applications. New York: George Brazillier.

[34] Bertenthal, Bennett I. (2007): Dynamical Systems: It's About Time. In: Boker, Steven M. & Wenger, Michael J. (eds.): Data Analytic Techniques for Dynamical Systems. Mahwah, NJ: Lawrence Erlbaum. 1 – 24.

[35] Blauert, Jens (1997): Spatial Hearing. The Psychophysics of Human Sound Localization. Cambridge, MA: MIT Press.

[36] Boden, Margaret A. (1996): Introduction. In: idem (ed.): The Philosophy of Artificial Life. Oxford: Oxford University Press. 1 – 35.

[37] Boden, Margaret A. (2006a): Mind as Machine. A History of Cognitive Science. vol. 1. Oxford University Press.

[38] Boden, Margaret A. (2006b): Mind as Machine. A History of Cognitive Science. vol. 2. Oxford University Press.

[39] Boolos, Gearge S. & Jeffrey, Richard C. (2002): Computability and Logic, 4th Edition. Cambridge: Cambridge University Press.

[40] Brainard, Michael S. & Knudsen, Eric I. (1993): Experience-dependent Plasticity in the Inferior Colliculus: A Site for Visual Calibration of the Neural Representation of Auditory Space in the Barn Owl. In: The Journal of Neuroscience 13: 4589 – 4608.

[41] Braitenberg, Valentino (1984): Vehicles. Experiments in Synthetic Psychology. Cambridge, MA: MIT Press.

[42] Breazeal, Cynthia L. (2002): Designing Sociable Robots. Cambridge, MA: MIT Press.

[43] Bregman, Albert S. (1990): Auditory Scene Analysis. The Perceptual Organization of Sound. Cambridge, MA: MIT Press.

[44] Brooks, Rodney A. (1985): A Robust Layered Control System for a Mobile Robot. Tech. Rep. A. I. Memo 864. Massachusetts Institute of Technology, Artificial Intelligence Laboratory. Cambridge, MA.

[45] Burger, Birgitta (2007): Communication of Musical Expression from Mobile Robots to Humans. Recognition of Music Emotions by Means of Robot Gestures. Master's thesis. KTH Stockholm.

[46] Burger, Birgitta & Bresin, Roberto (2007). Displaying expression in musical performance by means of a mobile robot. Poster presented at the Second Internation Conference on Affective Computing and Intelligent Interaction ACII 2007, Lisbon.

[47] Burger, Birgitta & Schmidt, Lüder (2009): Considerations Concerning a Methodology for Musical Robotics and Human-Robot Interaction. In: Louhivuori, Jukka, Eerola, Tuomas, Saarikallio, Suvi, Himberg, Tommi & Eerola, Päivi-Sisko (eds.): Proceedings of the 7th Triennial Conference of European Society for the Cognitive Sciences of Music (ESCOM 2009) Jyväskylä, Finland. 28 – 32.

[48] Burkhard, Mahlon D., (ed.) (1978): Manikin Measurements. Proceedings of a Conference Organized by M.D. Burkhard. Industrial Research Products, Inc.

[49] Camurri, Antonio, Castellano, Ginevra, Cowie, Roddy, Glowinski, Donald, Knapp, Ben, Krumhansl, Carol L., Villon, Olivier & Volpe, Gualtiero

(2007b): The Premio Paganini experiment: a multimodal gesture-based approach for explaining emotional processes in music performance. In: .

[50] Camurri, Antonio, Coletta, Paolo, Varni, Giovanna & Ghisio, Simone (2007): Developing multimodal interactive systems with EyesWeb XMI. In: Proceedings of the 2007 Conference on New Interfaces for Musical Expression (NIME07), New York, NY, USA. 305 – 308.

[51] Camurri, Antonio, Mazzarino, Barbara, Ricchetti, Matteo, Timmers, Renee & Volpe, Gualtiero (2004): Multimodal analysis of expressive gesture in music and dance performances. In: Camurri, Antonio & Volpe, Gualtiero (eds.): Gesture-based Communication in Human-Computer Interaction. Springer. vol. 2915 of Lecture Notes in Artificial Intelligence. 20 – 39.

[52] Camurri, Antonio & Volpe, Gualtiero (2011): Multimodal Analysis of Expressive Gesture in Music Performance. In: Solis, Jorge & Ng, Kia (eds.): Musical Robots and Interactive Multimodal Systems. Berlin / Heidelberg: Springer-Verlag. 47 – 66.

[53] Camurri, Antonio, Volpe, Gualtiero, De Poli, Giovanni & Leman, Marc (2005): Communicating Expressiveness and Affect in Multimodal Interactive Systems. In: IEEE Multimedia : 43 – 53.

[54] Chrisley, Ronald & Ziemke, Tom (2002): Embodiment. In: Nadel, Lynn (ed.): Encyclopedia of Cognitive Science. Macmillan. 1102 – 1108.

[55] Churchland, Patricia S. & Sejnowski, Terence J. (1992): The Computational Brain. Cambridge, MA: MIT Press.

[56] Clark, Andy (1997): Being There: Putting Brain, Body, and World Together Again. Cambridge, MA: MIT Press.

[57] Clark, Andy (1999): An Embodied Cognitive Science? In: Trends in Cognitive Sciences 3: 345 – 351.

[58] Clark, Andy (2001): Mindware. An Introduction to the Philosophy of Cognitive Science. Oxford University Press.

[59] Clark, Andy (2003): Natural-Born Cyborgs. Minds, Technologies, and the Future of Human Intelligence. Oxford University Press.

[60] Clayton, Martin, Sager, Rebecca & Will, Udo (2004): In Time with the Music: The Concept of Entrainment and Its Significance for Ethnomusicology. In: ESEM CounterPoint 1: 3 – 75.

[61] Collins, Randall (1988): Theoretical Sociology. San Diego: Harcourt Brace Jovanovich.

[62] Condon, W.S. & Ogston, W.D. (1971): Speech and Body Motion Synchrony of the Speaker-Hearer. In: Horton, David L. & Jenkins, James J. (eds.): Perception of Language. Columbus, OH: Charles E. Merrill Publishing Company. 150 – 184.

[63] Crick, Christopher, Munz, Matthew & Scassellati, Brian (2006): Synchronization in Social Tasks: Robot Drumming. In: Proceedings of the 15th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN 2006). University of Hertfordshire, UK.

[64] Dahl, Sofia & Friberg, Anders (2007): Visual perception of expressiveness in musicians' body movements. In: Music Perception 24: 433 – 454.

[65] Damasio, Antonio R. (1994): Descartes' Error. Emotion, Reason, and the Human Brain. New York: Putnam.

[66] Dautenhahn, Kerstin (2007): Methodology & Themes of Human-Robot Interaction: A Growing Research Field. In: International Journal of Advanced Robotic Systems 4(1): 103 – 108.

[67] Dautenhahn, Kerstin (2007): Socially Intelligent Robots: Dimensions of Human-Robot Interaction. In: Philosophical Transactions of the Royal Society B: Biological Sciences 362: 679 – 704.

[68] Desain, Peter & Windsor, Luke, (eds.) (2000): Rhythm Perception and Production. Lisse: Swets & Zeitlinger.

[69] Donnadieu, Sophie (2007): Mental Representations of the Timbre of Complex Sounds. In: Beauchamps, James W. (ed.): Analysis, Synthesis, and Perception of Musical Sounds. The Sound of Music. New York: Springer. 272 – 319.

[70] Dörner, Dietrich (2001): Bauplan für eine Seele. Reinbek: Rowohlt.

[71] Dörner, Dietrich (2002): Die Mechanik des Seelenwagens. Eine neuronale Theorie der Handlungsregulation. Bern: Huber.

[72] Dräger, Hans Heinz (1952): Begriff des Tonkörpers. In: Archiv für Musikwissenschaft 9: 68 – 77.

[73] Drake, Carolyn, Jones, Mari Riess & Baruch, Clarisse (2000): The development of rhythmic attending in auditory sequences:attunement, referent period, focal attending. In: Cognition 77: 251 – 288.

[74] Duda, Richard O., Hart, Peter E. & Stork, David G. (2001): Pattern Classification. New York: Wiley.

[75] Eck, Douglas, Gasser, Michael & Port, Robert F. (2000): Dynamics and embodiment in beat induction. In: Desain, Peter & Windsor, Luke (eds.): Rhythm Perception and Production. Lisse: Swets & Zeitlinger. 157 – 170.

[76] Eichert, Randolph, Schmidt, Lüder & Seifert, Uwe (1997): Logic, Gestalt Theory, and Neural Computation in Research on Auditory Perceptual Organization. In: Leman, Marc (ed.): Music, Gestalt, and Computing. Studies in Cognitive and Systematic Musicology. Berlin: Springer. Lecture Notes in Artificial Intelligence 1317. 70 – 88.

[77] Ekman, Paul & Friesen, W. (1982): Measuring Facial Movement with the Facial Action Coding System. In: Ekman, Paul (ed.): Emotion in the Human Face. Cambridge, UK: Cambridge University Press. 39 – 55.

[78] Ellner, Stephen P. & Guckenheimer, John (2006): Dynamic Models in Biology. Princeton: Princeton University Press.

[79] Elman, Jeffrey L. (1995): Language as a Dynamical System. In: Port, Robert F. & van Gelder, Timothy (eds.): Mind as Motion. Explorations in the Dynamics of Cognition. Cambridge, MA: MIT Press. 195 – 225.

[80] Enderton, Herbert B. (2001): A Mathematical Introduction to Logic, 2nd Edition. San Diego: Academic Press.

[81] Evans, Vyvyan & Green, Melanie (2006): Cognitive Linguistics. An Introduction. Edinburgh: Edinburgh University Press.

[82] Fauconnier, Gilles & Turner, Mark (2002): The Way We Think. Conceptual Blending and the Mind's Hidden Complexities. New York: Basic Books.

[83] Feldman, Jerome A. (2006): Fram Molecules to Metaphors. A Neural Theory of Language. Cambridge, MA: MIT Press.

[84] Feldman, Jerome A. & Narayanan, Srinivas S. (2003): Embodied meaning in a neural theory of language. In: Brain & Language 89: 385 – 392.

[85] Fellous, Jean-Marc & Arbib, Michael A., (eds.) (2005): Who Needs Emotions? The Brain Meets the Robot. Cambridge, MA: MIT Press.

[86] Feynman, Richard P. (1996 / 1999): Lectures on Computation. London: Penguin Books.

[87] Fitch, W. Tecumseh (1994): Vocal Tract Length Perception and the Evolution of Language. Ph.D. thesis. Brown University.

[88] Fraisse, Paul (1982): Rhythm and Tempo. In: Deutsch, Diana (ed.): The Psychology of Music. Academic Press. 149 – 180.

[89] Franklin, Stan & Graesser, Art (1997): Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents. In: Proceedings of the Agent Theories, Architectures, and Languages Workshop. Berlin: Springer.
`http://www.msci.memphis.edu/~franklin/AgentProg.html`

[90] Frijda, N.H. (1996): Emotions in Robots. In: Roitblat, Herbert L. & Meyer, Jean-Arcady (eds.): Comparative Approaches to Cognitive Science. Cambridge, MA: MIT Press. 501 – 516.

[91] Gardner, Howard (1985 / 1987): The Mind's New Sciene. A History of the Cognitive Revolution. Basic Books.

[92] Gembris, Heiner (1999): 100 Jahre musikalische Rezeptionsforschung. Ein Rückblick in die Zukunft. In: Musikpsychologie 14: 24 – 41.

[93] Genesereth, Michael & Nilsson, Nils J. (1986): Logical Foundations of Artificial Intelligence. San Mateo: Morgan Kaufman.

[94] Gleick, James (1998 / 1988): Chaos. The Amazing Science of the Unpredictable. London: Vintage Books.

[95] Glymour, Clark (1997): Goethe to van Gelder: Comments on "Dynamical Systems" Accounts of Cognition. In: PhilSci Archive .

[96] Godøy, Rolf-Inge & Leman, Marc (eds., 2010): Musical Gestures. Sound, Movement, and Meaning. Routledge.

[97] Gold, Kevin & Scassellati, Brian (2005): Learning about the self and others through contingency. In: AAAI Spring Symposium on Developmental Robotics. Palo Alto, CA.

[98] Goldin, Dina & Wegner, Peter (2002). Paraconsistency of Interactive Computation.
`http://www.citebase.org/abstract?id=oai:arXiv.org:cs/0207074`

[99] Goldin, Dina & Wegner, Peter (2005): Refuting the Strong Church-Turing Thesis: The Interactive Nature of Computing. Preprint.
`http://www.cs.brown.edu/~pw/`

[100] Goldstein, Herbert (1985): Klassische Mechanik, 8. Auflage. Darmstadt: Wissenschaftliche Buchgesellschaft.

[101] Grady, Joseph E. (1997): Foundations of Meaning: Primary Metaphors and Primary Scenes. Ph.D. thesis. University of California, Berkeley. Berkeley.

[102] G.R.A.S. Sound & Vibration (2007): KEMAR Manikin Type 45BA. Product Data and Specifications. G.R.A.S. Sound & Vibration. Holte, Denmark.
`www.gras.dk`

[103] Greiner, Walter (1984): Theoretische Physik Band 1: Mechanik I. Thun & Frankfurt / Main: Verlag Harri Deutsch.

[104] Grey, J. M. (1977): Multidimensional scaling of musical timbres. In: Journal of the Acoustical Society of America 61: 1270 – 1277.

[105] Grossberg, Stephen (1995): Neural Dynamics of Motion Perception, Recognition Learning, and Spatial Attention. In: Port, Robert F. & van Gelder, Timothy (eds.): Mind as Motion. Explorations in the Dynamics of Cognition. Cambridge, MA: MIT Press. 449 – 489.

[106] Grush, Rick (2004): The emulation theory of representation: Motor control, imagery, and perception. In: Behavioral and Brain Sciences 27: 377 – 442.

[107] Guckenheimer, John & Holmes, Philip (1983): Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields. New York: Springer-Verlag.

[108] Haken, Hermann (1983): Synergetics: An Introduction. Nonequilibrium Phase Transitions and Self-Organization in Physics, Chemistry, and Biology. Berlin: Springer.

[109] Hancock, Kenneth E. & Voigt, Herbert F. (1999): Wideband Inhibition of Dorsal Cochlear Nucleus Type IV Units in Cat: A Computational Model. In: Annals of Biomedical Engineering 27: 73 – 87.

[110] Handel, Steven & Oshinsky, J.S. (1981): The meter of syncopated auditory polyrhythms. In: Perception and Psychophysics 30: 1 – 9.

[111] Harel, David (2004): Algorithmics. The Spirit of Computing. Addison-Wesley. 3rd ed.

[112] Harel, David & Politi, Michael (1998): Modeling Reactive Systems with Statecharts: The Statemate Approach. New York: MacGraw-Hill.

[113] Harnad, Stevan (1990): The Symbol Grounding Problem. In: Physica D: Nonlinear Phenomena 42: 335 – 346.

[114] Hatfield, Elaine, Cacioppo, John T. & Rapson, Richard L. (1994): Emotional Contagion. Studies in Emotion and Social Interaction. Cambridge, UK: Cambridge University Press.

[115] Hempel, Carl G. (1977): Aspekte wissenschaftlicher Erklärung. Berlin: de Gruyter.

[116] Heschl, Adolf (1990): L=C. A Somple Equation with Astonishing Consequences. In: Journal of Theoretical Biology 145: 13 – 40.

[117] Heuser, Harro (1995): Gewöhnliche Differentialgleichungen, 3. Auflage. Stuttgart: B.G. Teubner.

[118] Hirsch, Morris W. & Smale, Stephen (1974): Differential Equations, Dynamical Systems, and Linear Algebra. New York: Academic Press.

[119] Hoffman, Guy & Weinberg, Gil (2010): Gesture-based Human-Robot Jazz Improvisation. In: IEEE International Conference on Robotics and Automation (ICRA2010).

[120] Hoffman, Guy & Weinberg, Gil (2011): Interactive Improvisation with a Robotic Marimba Player. In: Solis, Jorge & Ng, Kia (eds.): Musical Robots and Interactive Multimodal Systems. Berlin / Heidelberg: Springer-Verlag. 233 – 251.

[121] Holland, Owen (2004): The Future of Ambodied Artificial Intelligence: Machine Consciousness. In: Iida, Fumiya, Pfeifer, Rolf, Steels, Luc & Kuniyoshi, Yasuo (eds.): Embodied Artificial Intelligence. Berlin: Springer. 37 – 53.

[122] Hornbostel, Erich M. von (1926): Psychologie der Gehörserscheinungen. In: Bethe, A. (ed.): Handbuch der normalen und pathologischen Physiologie. Berlin: Springer. vol. XI. 701 – 730.

[123] Hutchins, Edwin (1995): Cognition in the Wild. Cambridge, MA: MIT Press.

[124] Hutchins, Edwin (1995): How a Cockpit Remebers Its Speed. In: Cognitive Science 19: 265 – 288.

[125] Ichbiah, Daniel (2005): Robots. From Science Fiction to Technological Revolution. New York: Harry N. Adams.

[126] Imig, Thomas J., Bibikov, Nikolai, Poirier, Pierre & Samson, Frank K. (2000): Directionality Derived From Pinna-Cue Spectral Notches in Cat Dorsal Cochlear Nucleus. In: Journal of Neurophysiology 83: 907 – 925.

[127] Inamura, Tetsunari, Toshima, Iwaki, Tanie, Hiroaki & Nakamura, Yoshihiko (2004): Embodied Symbol Emergence Based on Mimesis Theory. In: International Journal of Robotic Research 23(4-5): 363–377.

[128] Intel (2003): Intel PXA255 Processor. Developer's Manual. Tech. rep.. Intel.

[129] Intel (2003): Intel PXA255 Processor. User's Manual. Tech. rep.. Intel.

[130] Isbister, Katherine, Höök, Kia, Sharp, Michael & Laaksolahto, Jarmo (2006): The Sensual Evaluation Instrument: Developing an Affective Evaluation Tool. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems CHI06. Montreal. 1163 – 1172.

[131] Jackendoff, Ray & Lerdahl, Fred (2006): The capacity for music: What is it, and what's special about it? In: Cognition 100: 33 – 72.

[132] Jeannerod, Marc (2005): How Do We Decipher Others' Minds. In: Fellous, Jean-Marc & Arbib, Michael A. (eds.): Who Needs Emotions? The Brain Meets the Robot. New York: Oxford University Press. 147 – 169.

[133] Jensenius, Alexander Refsum (2006): Using Motiongrams in the Study of Musical Gestures. In: Proceedings of the 2006 International Computer Music Conference. New Orleans.

[134] Jensenius, Alexander Refsum, Camurri, Antonio, Castagné, Nicolas, Maestre, Esteban, Malloch, Joseph, McGilvray, Douglas, Schwarz, Diemo & Wright, Matthew (2007 a): Panel: The Need of Formats for Streaming and Storing Music-Related Movement and Gesture Data. In: Proceedings of the 2007 International Computer Music Conference. Copenhagen.

[135] Jensenius, Alexander Refsum, Castagné, Nicolas, Camurri, Antonio, Maestre, Esteban, Malloch, Joseph & McGilvray, Douglas (2007 b): A Summary of Formats for Streaming and Storing Music-Related Movement and Gesture Data. In: Proceedings of the 4th International Conference on Enactive Interfaces. Grenoble.

[136] Jensenius, Alexander Refsum, Godøy, Rolf-Inge & Wanderley, Marcelo M. (2005): Developing Tools for Studying Musical Gestures within the Max/MSP/Jitter Environment. In: Proceedings of the International Music Computer Conference. Barcelona. 282 – 285.

[137] Jensenius, Alexander Refsum, Wanderley, Marcelo M., Godøy, Rolf-Inge & Leman, Marc (2010): Musical Gestures: Concepts and Methods in Research. In: Godøy, Rolf-Inge & Leman, Marc (eds.): Musical Gestures. Sound, Movement, and Meaning. New York: Routledge. 12 – 35.

[138] Johnson, Christopher (1999): Metaphor vs. conflation in the acquisition of polysemy: the case of *see*. In: Hiraga, Masako K., Sinha, Chris & Wilcox, Sherman (eds.): Cultural, Psychological, and Typological Issues in Cognitive Linguistics. Amsterdam: John Benjamins Publishing Co. 155 – 170.

[139] Johnson, Mark & Larson, Steve (2003): ”Something in the Way She Moves“ – Metaphors of Musical Motion. In: Metaphor and Symbol 18(3): 63 – 84.

[140] Jones, Joseph L., Flynn, Anita M. & Seiger, Bruce A. (1999): Mobile Robots. Inspiration to Implementation, Second Edition. Natick, MA: A K Peters.

[141] Jones, Mari Riess (2004): Attention and Timing. In: Neuhoff, John G. (ed.): Ecological Psychoacoustics. Amsterdam: Elsevier. 49 – 85.

[142] Jones, Mari Riess, Moynihan, Heather, MacKenzie, Noah & Puente, Jennifer (2002): Temporal Aspects of Stimulus-Driven Attending in Dynamic Arrays. In: Psychological Science 13(4): 313 – 319.

[143] Juslin, Patrik N. (2001): Communicating Emotion in Music Performance: A Review and Theoretical Framework. In: Juslin, Patrik N. & Sloboda, John A. (eds.): Music and Emotion. Oxford: Oxford University Press. 309 – 337.

[144] Juslin, Patrik N. & Sloboda, John A., (eds.) (2001): Music and Emotion: Theory and Research. Oxford University Press.

[145] Kanamaru, Takashi (2007). Van der Pol Oscillator. `http://scholarpedia.org/article/Van_der_Pol_Oscillator`

[146] Kanold, Patrick O. & Young, Eric D. (2001): Proprioceptive Information from the Pinna Provides Somatosensory Input to Cat Dorsal Cochlear Nucleus. In: The Journal of Neuroscience 21: 7848 – 7858.

[147] Kapur, Ajay (2005): A History of Robotic Musical Instruments. In: Proceedings of the International Computer Music Conference.

[148] Kapur, Ajay & Singer, Eric (2006): A Retrieval Approach for Human/Robotic Musical Performance. In: ISMIR 2006. Proceedings of the 7th International Conference on Music Information Retrieval. Victoria, Canada. 363–364.

[149] Kapur, Ajay, Singer, Eric, Benning, Manjinder S., Tzanetakis, George & Trimpin (2007): Integrating HyperInstruments, Musical Robots & Machine Musicianship for North Indian Classical Music. In: Proceedings of the 2007 Conference on New Interfaces for Musical Expression (NIME07). New York, NY, USA. 238 – 241.

[150] Kapur, Ajay, Tzanetakis, George, Schloss, Andrew, Driessen, Peter & Singer, Eric (2006): Towards the One-Man Indian Computer Music Performance System. In: Proceedings of the International Computer Music Conference.

[151] Kato, Ichiro, Ohteru, Sadamu, Shirai, Katsuhiko, Narita, Seinosuke, Sugano, Shigeki, Matsushima, Toshiaki, Kobayashi, Tesunori & Fukisawa, Eizo (1987): The Robot Musician 'WABOT-2' (WAseda roBOT-2). In: Robotics 3(2): 143 – 155.

[152] Kauffman, Stuart A. (1993): The Origins of Order. Self-Organization and Selection in Evolution. New York: Oxford University Press.

[153] Kendon, Adam (1970): Movement coordination in social interaction: some examples described. In: Acta Psychologica. European Journal of Psychology 32: 101 –125.

[154] Kendon, Adam (1990): Conducting Interaction. Patterns of Behavior in Focused Encounters. Cambridge, UK: Cambridge University Press.

[155] Klensch, H. (1948): Beitrag zur Frage der Lokalisation des Schalles im Raum. In: Pflügers Archiv 250: 492 – 500.

[156] Knudsen, Eric I. (2002): Instructed learning in the auditory localization pathway of the barn owl. In: Nature 417: 322 – 328.

[157] Knudsen, Eric I, Zheng, Weimin & DeBello, William M. (2000): Traces of learning in the auditory localization pathway. In: PNAS 97(22): 11815 – 11820.

[158] Kooijmany, Tijn, Kanda, Takayuki, Bartneck, Christoph, Ishiguro, Hiroshi & Hagita, Norihiro (2006): Interaction Debugging: an Integral Approach to Analyze Human-Robot Interaction. In: Proceedings of the 1st Annual Conference on Human-Robot Interaction HRI06. 64 – 71.
`http://citeseer.ist.psu.edu/743683.html`

[159] Kövecses, Zoltán (2010): Metaphor: A Practical Introduction. Oxford University Press. second ed.

[160] Krumhansl, Carol L. (1990): Cognitive Foundations of Musical Pitch. New York: Oxford University Press.

[161] Krumhansl, Carol L. & Toiviainen, Petri (2003): Tonal Cognition. In: Peretz, Isabelle & Zatorre, Robert J. (eds.): The Cognitive Neuroscience of Music. New York: Oxford University Press. 95 –108.

[162] Lakoff, George (1990): The Invariance Hypothesis: is abstract reason based on image-schemas? In: Cognitive Linguistics 1(1): 39 – 74.

[163] Lakoff, George (2008): The neural theory of metaphor. In: Gibbs, Raymond W. (ed.): The Cambridge Handbook of Metaphor and Thought. New York: Cambridge University Press. 17 – 38.

[164] Lakoff, George & Johnson, Mark (1980 / 2003): Metaphors We Live By. University of Chicago Press.

[165] Lakoff, George & Johnson, Mark (1999): Philosophy in the Flesh. The Embodied Mind and Its Challenge to Western Philosophy. Basic Books.

[166] Lakoff, George & Núñez, Rafael E. (2000): Where Mathematics Comes From. How the Embodied Mind Brings Mathematics into Being. New York: Basic Books.

[167] Lambercy, Fredric & Bureau, Pierre (2007): Khepera III User Manual. Tech. rep.. K-Team Corporation. Yverdon-les-bains.
`www.k-team.com`

[168] Large, Edward W. & Jones, Mari Riess (1999): The Dynamics of Attending: How People Track Time-Varying Events. In: Psychological Review 106(1): 119 – 159.

[169] Larson, Steve & Johnson, Mark (2002): Architectural Metaphors in Music Discourse and Music Experience. In: Hertz, David M. (ed.): Yearbook of Comparative and General Literature. Toronto: University of Toronto Press. vol. 50.

[170] LeDoux, Joseph (1996): The Emotional Brain: The Mysterious Underpinnings of Emotional Life. New York: Simon & Schuster.

[171] Leman, Marc (1995): Music and Schema Theory. Cognitive Foundations of Systematic Musicology. Berlin: Springer.

[172] Leman, Marc (2008): Embodied Music Cognition and Mediation Technology. Cambridge, MA: MIT Press.

[173] LEMUR. http://www.lemurbots.org/.

[174] Lenk, Hans, (ed.) (1980): Handlungstheorien – interdisziplinär I: Handlungslogik, formale und sprachwissenschaftliche Handlungstheorien. München: Fink.

[175] Lerdahl, Fred & Jackendoff, Ray (1983): A Generative Theory of Tonal Music. Cambridge, MA: MIT Press.

[176] Leroi-Gourhan, Andreé (1964/1988): Hand und Wort. Die Evolution von Technik, Sprache und Kunst. Frankfurt / M.: Suhrkamp.

[177] Levitin, Daniel J. (2006): This is Your Brain on Music. The Science of a Human Obsession. New York: Dutton.

[178] Levitin, Daniel J. & Tirovolas, Anna K. (2009): Current Advances in the Cognitive Neuroscience of Music. In: Annals of the New York Academy of Sciences 1156: 211 – 231.

[179] Lungarella, Max, Metta, Giorgio, Pfeifer, Rolf & Sandini, Giulio (2003): Developmental robotics: a survey. In: Connection Science 15(4): 151 – 190.

[180] Malloch, Stephen & Trevarthen, Colwyn (2009): Musicality: Communicating the vitality and interests of life. In: Malloch, Stephen & Trevarthen, Colwyn (eds.): Communicative Musicality. Exploring the basis of human companionship. Oxford: Oxford University Press. 1 – 11.

[181] Mancini, Maurizio, Bresin, Roberto & Pelachaud, Cathreine (2006): From Acoustic Cues to an Expressive Agent. In: Gibet, S., Courty, N. & Kamp, J.-F. (eds.): Gesture Workshop 2005. Berlin: Springer. vol. 3881 of Lecture Notes in Artificial Intelligence. 280 – 291.

[182] Manna, Zohar & Pnueli, Amir (1991): The Temporal Logic of Reactive and Concurrent Systems. Specification. New York: Springer.

[183] Maturana, Humberto & Varela, Francisco (1984 / 1987): Der Baum der Erkenntnis. Die biologischen Wurzeln des menschlichen Erkennens. Scherz Verlag.

[184] May, Bradford J. (2000): Role of the dorsal cochlear nucleus in the sound localization behavior of cats. In: Hearing Research 148: 74 – 87.

[185] McKinney, Martin F. & Moelants, Dirk (2006): Ambiguity in Tempo Perception: What Draws Listeners to Different Metrical Levels? In: Music Perception 24(2): 155 – 166.

[186] Metta, Giorgio, Manzotti, Riccardo, Panerai, Francesco & Sandini, Giulio (2004): Development: Is it the right way towards humanoid robotics? In: .

[187] Michalowski, Marek P., Sabanovic, Selma & Kozima, Hideki (2007): A Dancing Robot for Rhythmic Social Interaction. In: Proceedings of the 2nd ACM/IEEE International Conference on Human-Robot Interaction HRI07. 89 – 96.

[188] Microchip (2003): PIC18F2331/2431/4331/4431 Data Sheet. 28/40/44-Pin Enhanced Flash Microcontrollers with nano Watt Technology, High Performance PWM and A/D. Tech. rep.. Microchip Technology Incorporated.

[189] Miller, George A. (2003): The cognitive revolution: a historical perspective. In: Trends in Cognitive Sciences 7(3): 141 – 144.

[190] Mithen, Steven (2005): The Singing Neanderthals. The Origins of Music, Language, Mind and Body. London: Phoenix.

[191] Moelants, Dirk & van Noorden, Leon (2005): The Influence of Pitch Interval on the Perception of Polyrhythms. In: Music Perception 22(3): 425 – 440.

[192] Möller, Ralf (2006): Das Ameisen-Patent. Bioroboter und ihre tierischen Vorbilder. Berlin: Spektrum.

[193] Morley, Iain (2003): The Evolutionary Origins and Archaeology of Music. Ph.D. thesis. Cambridge University.

[194] Musical. Gestures Project.
`http://www.hf.uio.no/imv/forskning/forskningsprosjekter/musicalgestures/`

[195] Nagano, Seido (2003): Receptors as a master key for synchronization of rhythms. In: Physical Review E 67(056215): 1 – 4.

[196] Narayanan, Srinivas S. (1997): Knowledge-based Action Representations for Metaphor and Aspect (KARMA). Ph.D. thesis. University of California. Berkeley.

[197] Nehaniv, Chrystopher & Dautenhahn, Kerstin, (eds.) (2007): Imitation and Social Learning in Robots, Humans and Animals. Cambridge: Cambridge University Press.

[198] Nehmzow, Ulrich (2000): Mobile Robotics: A Practical Introduction. London: Springer.

[199] Nelson, Raymond J. (1968): Introduction to Automata. New York: Wiley.

[200] Nelson, Raymond J. (1989): The Logic of Mind. Kluwer. 2nd ed.

[201] Nilsson, Nils J. (1991): Logic and artificial intelligence. In: Artificial Intelligence 47: 31 – 56.

[202] Nilsson, Nils J. (1998): Artificial Intelligence: A New Synthesis. San Francisco: Morgan Kaufman.

[203] Nolfi, Stefano & Floreano, Stefano (2000): Evolutionary Robotics. The Biology, Intelligence, and Technology of Self-Organizing Machines. Cambridge, MA: MIT Press.

[204] Norman, Donald A. (1993): Things that make us smart. Defending human attributes in the age of the machine. Reading, MA: Addison-Wesley.

[205] Norman, Donald A. (2004): Emotional Design. Why We Love (or Hate) Everyday Things. New York: Basic Books.

[206] Norman, Kent L. (2008): Cyberpsychology. An Introduction to Human-Computer Interaction. New York: Cambridge University Press.

[207] Norton, Alec (1995): Dynamics: An Introduction. In: Port, Robert F. & van Gelder, Timothy (eds.): Mind as Motion. Explorations in the Dynamics of Cognition. Cambridge, MA: MIT Press. chap. 2. 45 – 68.

[208] Núñez, Rafael (2004): Do *Real* Numbers Really Move? Language, Thought, and Gesture: The Embodied Cognitive Foundations of Mathematics. In: Iida, Fumiya, Pfeifer, Rolf, Steels, Luc & Kuniyoshi, Yasuo (eds.): Embodied Artificial Intelligence. Berlin: Springer. 54 – 73.

[209] Oertel, Donata & Young, Eric D. (2004): What's a cerebellar circuit doing in the auditory system. In: Trends in Neurosciences 27: 104 –110.

[210] Or, Jimmy (2009): Towards the Development of Emotional Dancing Humanoid Robots. In: International Journal of Social Robotics 1: 367 – 382.

[211] Ott, Edward (2002): Chaos in Dynamical Systems, Second Edition. Cambridge: Cambridge University Press.

[212] Partee, Barbara H., ter Meulen, Alice & Wall, Robert E. (1990): Mathematical Methods in Linguistics. Kluwer.

[213] Patterson, Roy D., Allerhand, Mike C. & Giguere, C. (1995): Time-domain modeling of peripheral auditory processing: A modular architecture and a software platform. In: Journal of the Acoustical Society of America 98: 1890 – 1894.

[214] Paul, Chandana (2006): Morphological Computation. In: Robotics and Autonomous Systems, Special Issue on Morphology, Control and Passive Dynamics 54(8): 619 – 630.
`http://www.mae.cornell.edu/paul/`

[215] Peper, C. (Lieke) E., Beek, Peter J. & Daffertshofer, Andreas (2000): Considerations regarding a comprehensive model of (poly)rhythmic movement. In: Desain, Peter & Windsor, Luke (eds.): Rhythm Perception and Production. Lisse: Swets & Zeitlinger. 35 – 49.

[216] Peretz, Isabelle (2001): Listen to the brain: a bioloical perspective on musical emotions. In: Juslin, Patrik N. & Sloboda, John A. (eds.): Music and Emotion. Oxford: Oxford University Press. 105 – 134.

[217] Peretz, Isabelle & Zatorre, Robert J., (eds.) (2003): The Cognitive Neuroscience of Music. Oxford: Oxford University Press.

[218] Pfeifer, Rolf & Bongard, Josh (2006): How the Body Shapes the Way We Think. A New View of Intelligence. Cambridge, MA: MIT Press.

[219] Pfeifer, Rolf & Iida, Fumiya (2004): Embodied Artificial Intelligence: Trends and Challenges. In: Iida, Fumiya, Pfeifer, Rolf, Steels, Luc & Kuniyoshi, Yasuo (eds.): Embodied Artificial Intelligence. Berlin: Springer. vol. 3139 of Lecture Notes in Artificial Intelligence. 1 – 26.

[220] Pfeifer, Rolf & Scheier, Christian (1999): Understanding Intelligence. Cambridge, MA: MIT Press.

[221] Philips Semiconductors (2000): The I^2C-Bus Specification Version 2.1. Tech. rep.

[222] Picard, Rosalind (1995): Affective Computing. Tech. Rep. 321. M.I.T. Media Laboratory Perceptual Computing Section. Cambridge, MA.

[223] Picard, Rosalind (1997): Affective Computing. Cambridge, MA: MIT Press.

[224] Picard, Rosalind, Vyzas, Elias & Healey, Jennifer (2001): Toward Machine Emotional Intelligence. In: IEEE Transactions on Pattern Analysis and Machine Intelligence 23(10): 1175 – 1191.

[225] Pikovsky, Arkady, Rosenblum, Michael & Kurths, Jürgen (2001): Synchronization. A unversal concept in nonlinear sciences. Cambridge, UK: Cambridge University Press.

[226] Polishook, Mark. http://www.youtube.com/watch?v=W-Q5m2u96sY.

[227] Polk, Thad A. & Seifert, Colleen M., (eds.) (2002): Cognitive Modeling. Cambridge, MA: MIT Press.

[228] Port, Robert F. & van Gelder, Timothy, (eds.) (1995): Mind as Motion. Explorations in the Dynamics of Cognition. Cambridge, MA: MIT Press.

[229] Prigogine, Ilya (1997): The End of Certainty. Time, Chaos, and the New Laws of Nature. New York: Free Press.

[230] Reber, Arthur S. & Reber, Emily S. (2001): The Penguin Dictionary of Psychology. London: Penguin Books.

[231] Reiss, Lina A. J. & Young, Eric D. (2005): Spectral Edge Sensitivity in Neural Circuits of the Dorsal Cochlear Nucleus. In: The Journal of Neuroscience 25: 3680 – 3691.

[232] Revel, Arnaud & Nadel, Jaqueline (2007): How to build an imitator. In: Nehaniv, Chrystopher & Dautenhahn, Kerstin (eds.): Imitation and Social Learning in Robots, Humans and Animals. Cambridge, UK: Cambridge University Press. 279 – 300.

[233] Révész, Géza (1946): Einführung in die Musikpsychologie. Bern: A. Francke Verlag.

[234] Rich, G. J. (1916): A Preliminary Study of Tonal Volume. In: Journal of Experimental Psychology 1: 13 – 22.

[235] Rodgers, Joseph L. & Johnson, Amber B. (2007): Nonlinear Dynamic Models of Nonlinear Dynamic Behaviors: Social Contagion of Adolescent Smoking and Drinking at Aggregate and Individual Levels. In: Boker, Steven M. & Wenger, Michael J. (eds.): Data Analytic Techniques for Dynamical Systems. Mahwah, NJ: Lawrence Erlbaum. 213 – 242.

[236] Rojas, Raúl (1992): Theorie der neuronalen Netze. Eine systematische Einführung. Berlin: Springer.

[237] Rosenblum, Michael & Pikovsky, Arkady (2003): Synchronization: from pendulum clocks to chaotic lasers and chemical oscillators. In: Contemporary Physics 44(5): 401 – 416.

[238] Russell, Stuart J. & Norvig, Peter (2003): Artificial Intelligence: A Modern Approach. Upper Saddle River: Prentice-Hall.

[239] Scheck, Florian (2003): Theoretische Physik 1: Mechanik. Von den Newton'schen Gesetzen zum deterministischen Chaos, 7. Auflage. Berlin et al.: Springer.

[240] Scheirer, Eric D. (2000): Music Listening Systems. Ph.D. thesis. Massachusetts Institute of Technology, Program in Media Arts and Sciences. Cambridge, MA.

[241] Schmidt, Lüder (2000): Grundlegende Konzepte der Signalverarbeitung für die Modellierung der musikalischen Hörwahrnehmung. Master's thesis. Universität Hamburg. Hamburg.

[242] Schmidt, Lüder (2005): Towards an "Embodied Cognitive Science of Music": Incorporating Mobile Autonomous Robots Into Musical Interaction. In: APSCOM05. Seoul.

[243] Schmidt, Lüder (2007): Embodied Cognitive Science as a Paradigm for Music Research. In: Lischka, Christoph & Sick, Andrea (eds.): Machines as Agency. Artistic Perspectives. Bielefeld: transcript Verlag. University of the Arts Bremen Press Series 04. 48 – 62.

[244] Schmidt, Lüder & Seifert, Uwe (2006): Musikalische Robotik: Körperlichkeit und Musik. In: Neue Zeitschrift für Musik (4): 44 – 45.

[245] Schmidt, Lüder, Seifert, Uwe & Eichert, Randolph (1997): Computational Auditory Scene Analysis as a Modeling Framework in Musicological Research on Perceptual Auditory Organization. In: Third Triennial ESCOM Conference Proceedings. ESCOM. Uppsala University.

[246] Schönberg, Arnold (1922): Harmonielehre. III. vermehrte und verbesserte Auflage. Wien: Universal-Edition.

[247] Schöner, Gregor (2008): Dynamical Systems Approaches to Cognition. In: Sun, Ron (ed.): The Cambridge Handbook of Computational Psychology. New York: Cambridge University Press. 101 – 126.

[248] Seifert, Uwe (1993): Systematische Musiktheorie und Kognitionswissenschaft – Zur Grundlegung der Kognitiven Musikwissenschaft. Bonn: Verlag für Systematische Musikwissenschaft.

[249] Seitz, Jay A. (2005b): The neural, evolutionary, developmental, and bodily basis of metaphor. In: New Ideas in Psychology 23: 74 – 95.

[250] Shepard, Roger N. (1982): Structural Representations of Musical Pitch. In: Deutsch, Diana (ed.): The Psychology of Music. New York: Academic Press. 343 – 390.

[251] Singer, Eric, Feddersen, Jeff & Bowen, Bill (2005): A Large-Scale Networked Robotic Musical Instrument Installation. In: Proceedings of the 2005 International Conference on New Interfaces for Musical Expression (NIME05). Vancouver.

[252] Singer, Eric, Feddersen, Jeff, Redmon, Chad & Bowen, Bill (2004): LEMUR's Musical Robots. In: Nagashima, Yoichi & Lyons, Michael J. (eds.): Proceedings of the 2004 Conference on New Interfaces for Musical Expression (NIME04). Shizuoka University of Art and Culture. 181 – 184.

[253] Sinha, Chris (2004): The Epigenesis of Symbolization. In: Balkenius, Christian, Zlatev, Jordan, Kozima, Hideki, Dautenhahn, Kerstin & Breazeal, Cynthia L. (eds.): Proceedings of the First International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems. vol. 85 of Lund University Cognitive Studies.
`http://www.lucs.lu.se/LUCS/085/`

[254] Snyder, Bob (2001): Music and Memory. An Introduction. Cambridge, MA: MIT Press.

[255] Sobh, Tarek M., Wang, Bei & Coble, Wurt W. (2003): Experimental Robot Musicians. In: Journal of Intelligent and Robotic Systems 38: 197 – 212.

[256] Solis, Jorge, Chida, Keisuke, Taniguchi, Koichi, Hashimoto, Shinichiro Michael, Suefuji, Kei & Takanishi, Atsuo (2006): The Waseda Flutist Robot WF-4RII in Comparison with a Professional Flutist. In: Computer Music Journal 30(4): 12 – 27.

[257] Solis, Jorge & Takanishi, Atsuo (2007): An overview of the research approaches on musical performance robots. In: Proceedings of the International Computer Music Conference, Copenhagen. 423 – 426.

[258] Solis, Jorge & Weinberg, Gil. RO-MAN 2010: Special Session on Robotics in Music and Art.

[259] Spitzer, Manfred (2002): Musik im Kopf. Hören, Musizieren, Verstehen und Erleben im neuronalen Netzwerk. Stuttgart: Schattauer.

[260] Spitzer, Michael (2004): Metaphor and Musical Thought. Chicago: University of Chicago Press.

[261] Steels, Luc (1996): Homo cyber-sapiens oder Robo hominidus intelligens: Maschinen erwachen zu künstlichem Leben. In: Maar, Christa, Pöppel, Ernst & Christaller, Thomas (eds.): DIe Technik auf dem Weg zur Seele. Forschungen an der Schnittstelle Gehirn / Computer. Reinbek: Rowohlt. 327 – 344.

[262] Strogatz, Steven (1994): Nonlinear Dynamics and Chaos. With Applications to Physics, Biology, Chemistry, and Engineering. Perseus Books.

[263] Strogatz, Steven (2003): Sync. The Emerging Science of Spontaneous Order. London: Penguin Books.

[264] Strube, Gert, (ed.) (2001): Wörterbuch der Kognitionswissenschaft. Klett-Cotta.

[265] Stumpf, Carl (1883): Tonpsychologie I. Leipzig: Hirzel.

[266] Sun, Ganghua & Scassellati, Brian (2004): Reaching through learned forward model. In: Proceedings of the 4th IEEE/RAS International Conference on Humanoid Robots, 2004. vol. 1. 93 – 112.

[267] Suzuki, Kenji, Camurri, Antonio, Ferrentino, Pasqualino & Hashimoto, Shuji (1998): Intelligent Agent System for Human-Robot Interaction through Artificial Emotion. In: Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics. 1055 – 1060.

[268] Suzuki, Kenji & Hashimoto, Shuji (2004): Robotic Interface for Embodied Interaction via Dance and Musical Performance. In: Proceedings of the IEEE 92(4): 656 – 671.

[269] Takanishi, Atsuo. Anthropomorphic Flutist Robot WF-4RIV. `http://www.takanishi.mech.waseda.ac.jp/research/flute/`

[270] Tanaka, Fumihide, Fortenberry, Bret, Aisaka, Kazuki & Movellan, Javier R. (2005): Plans for Developing Real-time Dance Interaction between QRIO and Toddlers in a Classroom Environment. In: Proceedings of the 4th IEEE International Conference on Development and Learning.

[271] Tanaka, Fumihide & Movellan, Javier R. (2006): Behavior Analysis of Children's touch on a Small Humanoid Robot: Long-term Observation at a Daily Classroom over Three Months. In: The 15th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN06). Hatfield, UK.

[272] Tanaka, Fumihide & Movellan, Javier R. (2006). How can robots predict how *good* they are interacting with humans? A way of implementing an internal reward for autinomous robots. Abstract, International Workshop on Synergistic Intelligence Dynamics at Humanoids'06, December 4, 2006.

[273] Tanaka, Fumihide, Movellan, Javier R., Fortenberry, Bret & Aisaka, Kazuki (2006): Daily HRI Evaluation at a Classrom Environment – Reports from Dance Interaction Experiments. In: Proceedings of the 1st Annual Conference on Human-Robot Interaction HRI06. Salt Lake City.

[274] Tanaka, Fumihide, Noda, Kuniaki, Sawada, Tsutomu & Fujita, Masahiro (2004): Associated Emotion and its Expression in an Entertainment Robot QRIO. In: Rauterberg, Matthias (ed.): Entertainment Computing – ICEC 2004. Berlin: Springer. Lecture Notes in Computer Science 3166.

[275] Tembrock, Günter (1996): Akustische Kommunikation bei Säugetieren: Die Stimmen der Säugetiere und ihre Bedeutung. Darmstadt: Wissenschaftliche Buchgesellschaft.

[276] Temperley, David (2001): The Cognition of Basic Musical Structures. Cambridge, MA: MIT Press.

[277] Thagard, Paul (2005): Mind. Introduction to Cognitive Science. Cambridge, MA: MIT Press.

[278] Thelen, Esther (1995): Time-Scale Dynamics and the Development of an Embodied Cognition. In: Port, Robert F. & van Gelder, Timothy (eds.): Mind as Motion. Explorations in the Dynamics of Cognition. Cambridge, MA: MIT Press. 69 – 100.

[279] Toda, Masanao (1982): Man, Robot, and Society. Models and Speculations. Boston: Martinus Nijhoff Publishing.

[280] Todd, Neil P. McAngus, Cousins, Rosanna & Lee, Christopher S. (2007): The Contribution of Anthropometric Factors to Individual Differences in the Perception of Rhythm. In: Empirical Musicology Review 2(1): 1 – 13.

[281] Todd, Neil P. McAngus, Lee, Christopher S. & O'Boyle, D.J. (2002): A sensorimotor theory of temporal tracking and beat induction. In: Psychological Research 66: 26 – 39.

[282] Toiviainen, Petri (2000): Symbolic AI Versus Connectionism. In: Reck Miranda, Eduardo (ed.): Readings in Music and Artificial Intelligence. Harwood Academic Publishers. 47 – 67.

[283] Toiviainen, Petri & Snyder, Joel S. (2000): The Time-Course of Pulse Sensation: Dynamics of Beat Induction. In: Woods, C., Luck, G., Brochard, R., Seddon, F. & Sloboda, John A. (eds.): Proceedings of the Sixth International Conference on Music Perception and Cognition. Keele: Keele University.

[284] Toiviainen, Petri & Snyder, Joel S. (2003): Tapping to Bach: Resonance-Based Modeling of Pulse. In: Music Perception 21(1): 43 – 80.

[285] Tomasello, Michael (2000): The Cultural Origins of Human Cognition. Cambridge, MA: Harvard University Press.

[286] Tomasello, Michael, Kruger, Ann C. & Ratner, Hilary H. (1993): Cultural learning. In: Behavioral and Brain Sciences 16: 495 – 552.

[287] Trevarthen, Colwyn (2002): Can a Robot Hear Music? Can a Robot Dance? Can a Robot Tell What it Knows or Intends to Do? Can it Feel Pride or Shame in Company? – Questions of the Nature of Human Vitality. In: Prince, Christopher G., Demiris, Y., Marom, Y., Kozima, Hideki & Balkenius, Christian (eds.): Proceedings of the Second International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems. Lund. vol. 94 of Lund University Cognitive Studies.

[288] Trevarthen, Colwyn & Malloch, Stephen (2002): Musicality and Music Before Three: Human Vitality and Invention Shared with Pride. In: Zero to Three : 10 – 18.

[289] Turing, Alan (1950): Computing Machinery and Intelligence. In: Mind 59: 433 – 460.

[290] van Gelder, Timothy (1997): The Dynamical Hypothesis in Cognitive Science. In: Behavioral and Brain Sciences .

[291] van Noorden, Leon & Moelants, Dirk (1999): Resonance in the perception of musical pulse. In: Journal of New Music Research 28: 43 – 66.

[292] van Soest, J.L. (1929): Richtungshooren bij sinusvormige geluidstrillingen. In: Physica 9: 271 – 282.

[293] van Valkenburg, David & Kubovy, Michael (2004): From Gibson's Fire to Gestalts: A Bridge-Building Theory of Perceptual Objecthood. In: Neuhoff, John G. (ed.): Ecological Psychoacoustics. Amsterdam: Elsevier Academic Press. 113 – 147.

[294] Vogt, Paul (2003). Anchoring symbols to sensorimotor control. `http://cogprints.org/3060/`

[295] von Eckardt, Barbara (1993): What Is Cognitive Science? Cambridge, MA: MIT Press.

[296] Wagner, Hermann, Takahashi, Terry & Konishi, Masakazu (1987): Representation of Interaural Time Difference in the Central Nucleus of the Barn Owl's Inferior Colliculus. In: The Journal of Neuroscience 7: 3105 – 3116.

[297] Wallach, Hans (1939): On Sound Localization. In: Journal of the Acoustical Society of America 10: 270 – 274.

[298] Wang, DeLiang (1993): Pattern Recognition: Neural Networks in Perspective. In: IEEE Expert .

[299] Wang, DeLiang (1999): Relaxation Oscillators and Networks. In: Webster, J.G. (ed.): Wiley Encyclopedia of Electrical and Alectronics Engineering. New York: Wiley. vol. 18. 396 – 405.

[300] Wang, DeLiang & Brown, Guy J., (eds.) (2006): Computational Auditory Scene Analysis. Principles, Algorithms, and Applications. Hoboken, NJ: Wiley.

[301] Wang, DeLiang & Brown, Guy J. (2006): Fundamentals of Computational Auditory Scene Analysis. In: Wang, DeLiang & Brown, Guy J. (eds.): Computational Auditory Scene Analysis. Principles, Algorithms, and Applications. Hoboken, NJ: Wiley. chap. 1. 1 – 44.

[302] Ward, Lawrence M. (2001): Dynamical Cognitive Science. Cambridge, MA: MIT Press.

[303] Warren, Richard M. (1999): Auditory Perception. A New Analysis and Synthesis. New York: Cambridge University Press.

[304] Webb, Barbara (2001): Can robots make good models of biological behaviour? In: Behavioral and Brain Sciences 24(6): 1033 – 1050.

[305] Webb, Barbara, Reeve, Richard, Horchler, Andrew & Quinn, Roger (2003): Testing a model of cricket phonotaxis on an outdoor robotic platform. In: Proceedings of TIMR03.

[306] Wegner, Peter & Goldin, Dina (2003): Computation beyond turing machines. In: Commun. ACM 46(4): 100–102.

[307] Weinberg, Gil & Blosser, Brian (2009): A Leader-Follower Turn-taking Model Incorporating Beat Detection in Musical Human-Robot Interaction. In: Proceedings of the 4th ACM/IEEE International Conference on Human Robot Interaction (HRI 09). La Jolla, CA. 227 – 228.

[308] Weinberg, Gil, Blosser, Brian, Mallikarjuna, Trishul & Raman, Aparna (2009): The Creation of a Multi-Human, Multi-Robot Interactive Jam Session. In: Proceedings of the International Conference on New Instruments for Music Expression (NIME 09). Pittsburg, PA.

[309] Weinberg, Gil & Driscoll, Scott (2006a): Toward Robotic Musicianship. In: Computer Music Journal 30(4): 28 – 45.

[310] Weinberg, Gil & Driscoll, Scott (2006b): Robot-Human Interaction with an Anthropomorphic Percussionist. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems CHI06. 1229 – 1232.

[311] Weinberg, Gil & Driscoll, Scott (2007b): The Design of a Robotic Marimba Player – Introducing Pitch into Robotic Musicianship. In: Proceedings of the 2007 Conference on New Interfaces for Musical Expression (NIME07). New York. 228 – 233.

[312] Weinberg, Gil, Driscoll, Scott & Parry, Mitchell (2005): Musical Interactions with a Perceptual Robotic Percussionist. In: Proceedings of the IEEE International Workshop on Robot and Human Interactive Communication (RO-MAN 2005). Nashville, TN. 456 – 461.

[313] Weinberg, Gil, Godfrey, Marc, Rae, Alex & Rhoads, John (2007): A Real-Time Genetic Algorithm in Human-Robot Musical Improvisation. In: Proceedings of the 2007 International Computer Music Conference (ICMC 2007). Copenhagen. 192 – 195.

[314] Weinberg, Gil, Raman, Aparna & Mallikarjuna, Trishul (2009): Interactive Jamming with Shimon: A Social Robotic Musician. In: Proceedings of the 4th ACM/IEEE International Conference on Human Robot Interaction (HRI 09). La Jolla, CA. 233 – 234.

[315] Wenger, Michael J., Copeland, Angelina M. & Schuster, Christof (2007): Gestures as Psychophysical Judgments. In: Boker, Steven M. & Wenger, Michael J. (eds.): Data Analytic Techniques for Dynamical Systems. Mahwah, NJ: Lawrence Erlbaum. 243 – 272.

[316] Wessel, David L. (1979): Timbre space as a musical control structure. In: Computer Music Journal 3(2): 45 – 52.

[317] Wiener, Norbert (1961): Cybernetics: or Control and Communication in the Animal and the Machine. Second Edition. Cambridge, MA: MIT Press.

[318] Wilson, Robert A. & Keil, Frank C., (eds.) (1999): The MIT Encyclopedia of the Cognitive Sciences. MIT Press.

[319] Woods, Sarah, Walters, Michael, Koay, Kheng Lee & Dautenhahn, Kerstin (2006): Comparing Human Robot Interaction Scenarios Using Live and Video Based Methods: Towards a Novel Methodological Approach. In: Proceedings of the 15th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN 2006). 51 – 58.

[320] Wright, Matthew. The Open Sound Control Specification. `http://opensoundcontrol.org/spec-1_0`

[321] Wright, Matthew. OpenSound Control Specification. `http://www.cnmat.berkeley.edu/OpenSoundControl/OSC-spec.html`

[322] Wright, Matthew (2005): Open Sound Control: an enabling technology for musical networking. In: Organised Sound 10(3): 193 – 200.

[323] Wright, Matthew, Freed, Adrian & Momeni, Ali (2003): OpenSound Control: State of the Art 2003. In: Proceeding of the 2003 Conference on New Interfaces for Musical Expression (NIME-03). Montreal.

[324] Young, Eric D. & Davis, Kevin A. (2002): Circuitry and Function of the Dorsal Cochlear Nucleus. In: Oertel, Donata, Popper, Arthur N. & Fay, Richard R. (eds.): Integrative Functions in the Mammalian Auditory Pathway. New York: Springer. 160 – 206.

[325] Zbikowski, Lawrence M. (2002): Conceptualizing Music. Cognitive Structure, Theory, and Analysis. Oxford: Oxford University Press.

[326] Zbikowski, Lawrence M. (2008): Metaphor and Music. In: Gibbs, Raymond W. (ed.): The Cambridge Handbook of Metaphor and Thought. New York: Cambridge University Press. 502 – 524.

[327] Zbikowski, Lawrence M. (2009): Music, language, and multimodal metaphor. In: Forceville, Charles J. & Urios-Aparisi, Eduardo (eds.): Multimodal Metaphor. Mouton de Gruyter. vol. 11 of Applications of Cognitive Linguistics. chap. 15. 359 – 381.

[328] Ziemke, Tom (2001): Are Robots Embodied? In: Balkenius, Christian, Zlatev, Jordan, Kozima, Hideki, Dautenhahn, Kerstin & Breazeal, Cynthia L. (eds.): Proceedings of the First International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems. Lund. vol. 85 of Lund University Cognitive Studies.
`http://www.lucs.lu.se/abstracts/LUCS_studies/LUCS85.html`