

Humanoid Robots

Applied Principles and Research

Paula Sutton

Humanoid Robots: Applied Principles and Research

Edited by **Paula Sutton**



New Jersey

Published by Clanrye International,
55 Van Reypen Street,
Jersey City, NJ 07306, USA
www.clanryeinternational.com

Humanoid Robots: Applied Principles and Research

Edited by Paula Sutton

© 2015 Clanrye International

International Standard Book Number: 978-1-63240-300-1 (Hardback)

This book contains information obtained from authentic and highly regarded sources. Copyright for all individual chapters remain with the respective authors as indicated. A wide variety of references are listed. Permission and sources are indicated; for detailed attributions, please refer to the permissions page. Reasonable efforts have been made to publish reliable data and information, but the authors, editors and publisher cannot assume any responsibility for the validity of all materials or the consequences of their use.

The publisher's policy is to use permanent paper from mills that operate a sustainable forestry policy. Furthermore, the publisher ensures that the text paper and cover boards used have met acceptable environmental accreditation standards.

Trademark Notice: Registered trademark of products or corporate names are used only for explanation and identification without intent to infringe.

Printed in the United States of America.

Humanoid Robots: Applied Principles and Research

Preface

Advanced engineering and scientific research results and growth in the area of humanoid robotics as well as its applications have all been elucidated in this book. Humanoids are believed to transform the way we communicate with machines and possess the ability to blend smoothly into an environment already designed for humans. The information provided in this book aims at discovering the future abilities of humanoid robots by evaluating a variety of integrated research in distinct engineering and scientific areas like perception, human-robot interaction, locomotion, machine learning, adaptive behavior and neuroscience. It is a practical book as it lays stress on important information for the people engaged in the fields of robotics, computational methods, artificial and cognitive science as well as other scientific fields directly or indirectly connected with the applications and advancement of future humanoid robots. The information in this book is contributed by experts holding significant research and development experience, patents as well as publications in the field of humanoid robotics.

The researches compiled throughout the book are authentic and of high quality, combining several disciplines and from very diverse regions from around the world. Drawing on the contributions of many researchers from diverse countries, the book's objective is to provide the readers with the latest achievements in the area of research. This book will surely be a source of knowledge to all interested and researching the field.

In the end, I would like to express my deep sense of gratitude to all the authors for meeting the set deadlines in completing and submitting their research chapters. I would also like to thank the publisher for the support offered to us throughout the course of the book. Finally, I extend my sincere thanks to my family for being a constant source of inspiration and encouragement.

Editor

Contents

	Preface	VII
	Part 1 Periodic Tasks and Locomotion Control	1
Chapter 1	Performing Periodic Tasks: On-Line Learning, Adaptation and Synchronization with External Signals Andrej Gams, Tadej Petrič, Aleš Ude and Leon Žlajpah	3
Chapter 2	Design of Oscillatory Neural Network for Locomotion Control of Humanoid Robots Riadh Zaier	29
Chapter 3	Autonomous Motion Adaptation Against Structure Changes Without Model Identification Yuki Funabara, Yoshikazu Yano, Shinji Doki and Shigeru Okuma	49
	Part 2 Grasping and Multi-Fingered Robot Hand	61
Chapter 4	Grasp Planning for a Humanoid Hand Tokuo Tsuji, Kensuke Harada, Kenji Kaneko, Fumio Kanehiro and Kenichi Maruyama	63
Chapter 5	Design of 5 D.O.F Robot Hand with an Artificial Skin for an Android Robot Dongwoon Choi, Dong-Wook Lee, Woonghee Shon and Ho-Gil Lee	81
Chapter 6	Development of Multi-Fingered Universal Robot Hand with Torque Limiter Mechanism Wataru Fukui, Futoshi Kobayashi and Fumio Kojima	97
	Part 3 Interactive Applications of Humanoid Robots	109
Chapter 7	Affective Human-Humanoid Interaction Through Cognitive Architecture Ignazio Infantino	111

Chapter 8	Speech Communication with Humanoids: How People React and How We Can Build the System Yosuke Matsusaka	129
Chapter 9	Exoskeleton and Humanoid Robotic Technology in Construction and Built Environment T. Bock, T. Linner and W. Ikeda	153
Chapter 10	Implementation of a Framework for Imitation Learning on a Humanoid Robot Using a Cognitive Architecture Huan Tan	189
Chapter 11	User, Gesture and Robot Behaviour Adaptation for Human-Robot Interaction Md. Hasanuzzaman and Haruki Ueno	211
Chapter 12	Learning Novel Objects for Domestic Service Robots Muhammad Attamimi, Tomoaki Nakamura, Takayuki Nagai, Komei Sugiura and Naoto Iwahashi	239
Chapter 13	A Multi-Modal Panoramic Attentional Model for Robots and Applications Ravi Sarvadevabhatla and Victor Ng-Thow-Hing	259
Part 4	Current and Future Challenges for Humanoid Robots	277
Chapter 14	Rob's Robot: Current and Future Challenges for Humanoid Robots Boris Durán and Serge Thill	279
	Permissions	
	List of Contributors	

Part 1

Periodic Tasks and Locomotion Control

Performing Periodic Tasks: On-Line Learning, Adaptation and Synchronization with External Signals

Andrej Gams, Tadej Petrič, Aleš Ude and Leon Žlajpah
*Jožef Stefan Institute, Ljubljana
 Slovenia*

1. Introduction

One of the central issues in robotics and animal motor control is the problem of trajectory generation and modulation. Since in many cases trajectories have to be modified on-line when goals are changed, obstacles are encountered, or when external perturbations occur, the notions of trajectory generation and trajectory modulation are tightly coupled.

This chapter addresses some of the issues related to trajectory generation and modulation, including the supervised learning of periodic trajectories, and with an emphasis on the learning of the frequency and achieving and maintaining synchronization to external signals. Other addressed issues include robust movement execution despite external perturbations, modulation of the trajectory to reuse it under modified conditions and adaptation of the learned trajectory based on measured force information. Different experimental scenarios on various robotic platforms are described.

For the learning of a periodic trajectory without specifying the period and without using traditional off-line signal processing methods, our approach suggests splitting the task into two sub-tasks: (1) frequency extraction, and (2) the supervised learning of the waveform. This is done using two ingredients: nonlinear oscillators, also combined with an adaptive Fourier waveform for the frequency adaptation, and nonparametric regression¹ techniques for shaping the attractor landscapes according to the demonstrated trajectories. The systems are designed such that after having learned the trajectory, simple changes of parameters allow modulations in terms of, for instance, frequency, amplitude and oscillation offset, while keeping the general features of the original trajectory, or maintaining synchronization with an external signal.

The system we propose in this paper is based on the motion imitation approach described in (Ijspeert et al., 2002; Schaal et al., 2007). That approach uses two dynamical systems like the system presented here, but with a simple nonlinear oscillator to generate the phase and the amplitude of the periodic movements. A major drawback of that approach is that it requires the frequency of the demonstration signal to be explicitly specified. This means that the frequency has to be either known or extracted from the recorded signal by signal

¹ The term “nonparametric” is to indicate that the data to be modeled stem from very large families of distributions which cannot be indexed by a finite dimensional parameter vector in a natural way. It does not mean that there are no parameters.

processing methods, e.g. Fourier analysis. The main difference of our new approach is that we use an adaptive frequency oscillator (Buchli & Ijspeert, 2004; Righetti & Ijspeert, 2006), which has the process of frequency extraction and adaptation totally embedded into its dynamics. The frequency does not need to be known or extracted, nor do we need to perform any transformations (Righetti et al., 2006). This simplifies the process of teaching a new task/trajectory to the robot. Additionally, the system can work incrementally in on-line settings. We use two different approaches. One uses several frequency oscillators to approximate the input signal, and thus demands a logical algorithm to extract the basic frequency of the input signal. The other uses only one oscillator and higher harmonics of the extracted frequency. It also includes an adaptive fourier series.

Our approach is loosely inspired from dynamical systems observed in vertebrate central nervous systems, in particular central pattern generators (Ijspeert, 2008a). Additionally, our work fits in the view that biological movements are constructed out of the combination of "motor primitives" (Mataric, 1998; Schaal, 1999), and the system we develop could be used as blocks or motor primitives for generating more complex trajectories.

1.1 Overview of the research field

One of the most notable advantages of the proposed system is the ability to synchronize with an external signal, which can effectively be used in control of rhythmic periodic task where the dynamic behavior and response of the actuated device are critical. Such robotic tasks include swinging of different pendulums (Furuta, 2003; Spong, 1995), playing with different toys, i.e. the yo-yo (Hashimoto & Noritsugu, 1996; Jin et al., 2009; Jin & Zacksenhouse, 2003; Žlajpah, 2006) or a gyroscopic device called the Powerball (Cafuta & Curk, 2008; Gams et al., 2007; Heyda, 2002; Petrič et al., 2010), juggling (Buehler et al., 1994; Ronsse et al., 2007; Schaal & Atkeson, 1993; Williamson, 1999) and locomotion (Ijspeert, 2008b; Ilg et al., 1999; Morimoto et al., 2008). Rhythmic tasks are also handshaking (Jindai & Watanabe, 2007; Kasuga & Hashimoto, 2005; Sato et al., 2007) and even handwriting (Gangadhar et al., 2007; Hollerbach, 1981). Performing these tasks with robots requires appropriate trajectory generation and foremost precise frequency tuning by determining the basic frequency. We denote the lowest frequency relevant for performing a given task, with the term "basic frequency".

Different approaches that adjust the rhythm and behavior of the robot, in order to achieve synchronization, have been proposed in the past. For example, a feedback loop that locks onto the phase of the incoming signal. Closed-loop model-based control (An et al., 1988), as a very common control of robotic systems, was applied for juggling (Buehler et al., 1994; Schaal & Atkeson, 1993), playing the yo-yo (Jin & Zacksenhouse, 2002; Žlajpah, 2006) and also for the control of quadruped (Fukuoka et al., 2003) and in biped locomotion (Sentis et al., 2010; Spong & Bullo, 2005). Here the basic strategy is to plan a reference trajectory for the robot, which is based on the dynamic behavior of the actuated device. Standard methods for reference trajectory tracking often assume that a correct and exhaustive dynamic model of the object is available (Jin & Zacksenhouse, 2002), and their performance may degrade substantially if the accuracy of the model is poor.

An alternative approach to controlling rhythmic tasks is with the use of nonlinear oscillators. Oscillators and systems of coupled oscillators are known as powerful modeling tools (Pikovsky et al., 2002) and are widely used in physics and biology to model phenomena as diverse as neuronal signalling, circadian rhythms (Strogatz, 1986), inter-limb coordination (Haken et al., 1985), heart beating (Mirolo et al., 1990), etc. Their properties, which include robust limit cycle behavior, online frequency adaptation (Williamson, 1998)

and self-sustained limit cycle generation on the absence of cyclic input (Bailey, 2004), to name just a few, make them suitable for controlling rhythmic tasks.

Different kinds of oscillators exist and have been used for control of robotic tasks. The van der Pol non-linear oscillator (van der Pol, 1934) has successfully been used for skill entrainment on a swinging robot (Veskos & Demiris, 2005) or gait generation using coupled oscillator circuits, e.g. (Jalics et al., 1997; Liu et al., 2009; Tsuda et al., 2007). Gait generation has also been studied using the Rayleigh oscillator (Filho et al., 2005). Among the extensively used oscillators is also the Matsuoka neural oscillator (Matsuoka, 1985), which models two mutually inhibiting neurons. Publications by Williamson (Williamson, 1999; 1998) show the use of the Matsuoka oscillator for different rhythmic tasks, such as resonance tuning, crank turning and playing the slinky toy. Other robotic tasks using the Matsuoka oscillator include control of giant swing problem (Matsuoka et al., 2005), dish spinning (Matsuoka & Ooshima, 2007) and gait generation in combination with central pattern generators (CPGs) and phase-locked loops (Inoue et al., 2004; Kimura et al., 1999; Kun & Miller, 1996).

On-line frequency adaptation, as one of the properties of non-linear oscillators (Williamson, 1998) is a viable alternative to signal processing methods, such as fast Fourier transform (FFT), for determining the basic frequency of the task. On the other hand, when there is no input into the oscillator, it will oscillate at its own frequency (Bailey, 2004). Righetti et al. have introduced adaptive frequency oscillators (Righetti et al., 2006), which preserve the learned frequency even if the input signal has been cut. The authors modify non-linear oscillators or pseudo-oscillators with a learning rule, which allows the modified oscillators to learn the frequency of the input signal. The approach works for different oscillators, from a simple phase oscillator (Gams et al., 2009), the Hopf oscillator, the Fitzhugh-Nagumo oscillator, etc. (Righetti et al., 2006). Combining several adaptive frequency oscillators in a feedback loop allows extraction of several frequency components (Buchli et al., 2008; Gams et al., 2009). Applications vary from bipedal walking (Righetti & Ijspeert, 2006) to frequency tuning of a hopping robot (Buchli et al., 2005). Such feedback structures can be used as a whole imitation system that both extracts the frequency and learns the waveform of the input signal.

Not many approaches exist that combine both frequency extraction and waveform learning in imitation systems (Gams et al., 2009; Ijspeert, 2008b). One of them is a two-layered imitation system, which can be used for extracting the frequency of the input signal in the first layer and learning its waveform in the second layer, which is the basis for this chapter. Separate frequency extraction and waveform learning have advantages, since it is possible to independently modulate temporal and spatial features, e.g. phase modulation, amplitude modulation, etc. Additionally a complex waveform can be anchored to the input signal. Compact waveform encoding, such as splines (Miyamoto et al., 1996; Thompson & Patel, 1987; Ude et al., 2000), dynamic movement primitives (DMP) (Schaal et al., 2007), or Gaussian mixture models (GMM) (Calinon et al., 2007), reduce computational complexity of the process. In the next sections we first give details on the two-layered movement imitation system and then give the properties. Finally, we propose possible applications.

2. Two-layered movement imitation system

In this chapter we give details and properties of both sub-systems that make the two-layered movement imitation system. We also give alternative possibilities for the canonical dynamical system.

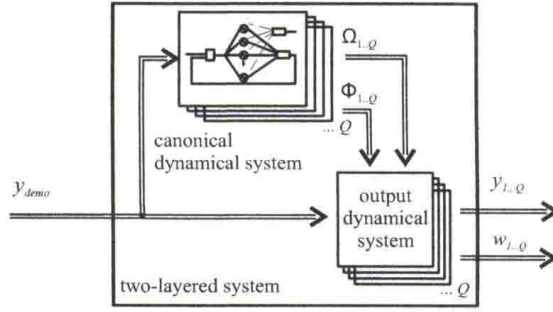


Fig. 1. Proposed structure of the system. The two-layered system is composed of the *Canonical Dynamical System* as the first layer for the frequency adaptation, and the *Output Dynamical System* for the learning as the second layer. The input signal $y_{demo}(t)$ is an arbitrary Q -dimensional periodic signal. The Canonical Dynamical System outputs the fundamental frequency Ω and phase of the oscillator at that frequency, Φ , for each of the Q DOF, and the Output Dynamical System learns the waveform.

Figure 1 shows the structure of the proposed system for the learning of the frequency and the waveform of the input signal. The input into the system $y_{demo}(t)$ is an arbitrary periodic signal of one or more degrees of freedom (DOF).

The task of frequency and waveform learning is split into two separate tasks, each performed by a separate dynamical system. The frequency adaptation is performed by the *Canonical Dynamical System*, which either consists of several adaptive frequency oscillators in a feedback structure, or a single oscillator with an adaptive Fourier series. Its purpose is to extract the basic frequency Ω of the input signal, and to provide the phase Φ of the signal at this frequency.

These quantities are fed into the *Output Dynamical System*, whose goal is to adapt the shape of the limit cycle of the Canonical Dynamical System, and to learn the waveform of the input signal. The resulting output signal of the Output Dynamical System is not explicitly encoded but generated during the time evolution of the Canonical Dynamical System, by using a set of weights learned by Incremental Locally Weighted Regression (ILWR) (Schaal & Atkeson, 1998).

Both frequency adaptation and waveform learning work in parallel, thus accelerating the process. The output of the combined system can be, for example, joint coordinates of the robot, position in task space, joint torques, etc., depending on what the input signal represents.

In the next section we first explain the second layer of the system - the output dynamical system - which learns the waveform of the input periodic signal once the frequency is determined.

2.1 Output dynamical system

The output dynamical system is used to learn the waveform of the input signal. The explanation is for a 1 DOF signal. For multiple DOF, the algorithm works in parallel for all the degrees of freedom.

The following dynamics specify the attractor landscape of a trajectory y towards the anchor point g , with the Canonical Dynamical System providing the phase Φ to the function Ψ_i of the control policy:

$$\dot{z} = \Omega \left(\alpha_z (\beta_z (g - y) - z) + \frac{\sum_{i=1}^N \Psi_i w_i r}{\sum_{i=1}^N \Psi_i} \right) \quad (1)$$

$$\dot{y} = \Omega z \quad (2)$$

$$\Psi_i = \exp(h(\cos(\Phi - c_i) - 1)) \quad (3)$$

Here Ω (chosen amongst the ω_i) is the frequency given by canonical dynamical system, Eq. (10), α_z and β_z are positive constants, set to $\alpha_z = 8$ and $\beta_z = 2$ for all the results; the ratio 4:1 ensures critical damping so that the system monotonically varies to the trajectory oscillating around g - an anchor point for the oscillatory trajectory. N is the number of Gaussian-like periodic kernel functions Ψ_i , which are given by Eq. (3). w_i is the learned weight parameter and r is the amplitude control parameter, maintaining the amplitude of the demonstration signal with $r = 1$. The system given by Eq. (1) without the nonlinear term is a second-order linear system with a unique globally stable point attractor (Ijspeert et al., 2002). But, because of the periodic nonlinear term, this system produces stable periodic trajectories whose frequency is Ω and whose waveform is determined by the weight parameters w_i .

In Eq. (3), which determines the Gaussian-like kernel functions Ψ_i , h determines their width, which is set to $h = 2.5N$ for all the results presented in the paper unless stated otherwise, and c_i are equally spaced between 0 and 2π in N steps.

As the input into the learning algorithm we use triplets of position, velocity and acceleration $y_{demo}(t)$, $\dot{y}_{demo}(t)$, and $\ddot{y}_{demo}(t)$ with *demo* marking the input or demonstration trajectory we are trying to learn. With this Eq. (1) can be rewritten as

$$\frac{1}{\Omega} \dot{z} - \alpha_z (\beta_z (g - y) - z) = \frac{\sum_{i=1}^N \Psi_i w_i r}{\sum_{i=1}^N \Psi_i} \quad (4)$$

and formulated as a supervised learning problem with on the right hand side a set of local models $w_i r$ that are weighted by the kernel functions Ψ_i , and on the left hand side the target function f_{targ} given by $f_{targ} = \frac{1}{\Omega^2} \ddot{y}_{demo} - \alpha_z (\beta_z (g - y_{demo}) - \frac{1}{\Omega} \dot{y}_{demo})$, which is obtained by matching y to y_{demo} , z to $\frac{\dot{y}_{demo}}{\Omega}$, and \dot{z} to $\frac{\ddot{y}_{demo}}{\Omega}$.

Locally weighted regression corresponds to finding, for each kernel function Ψ_i , the weight vector w_i , which minimizes the quadratic error criterion²

$$J_i = \sum_{t=1}^P \Psi_i(t) (f_{targ}(t) - w_i r(t))^2 \quad (5)$$

where t is an index corresponding to discrete time steps (of the integration). The regression can be performed as a *batch* regression, or alternatively, we can perform the minimization of the J_i cost function incrementally, while the target data points $f_{targ}(t)$ arrive. As we want continuous learning of the demonstration signal, we use the latter. Incremental regression is done with the use of recursive least squares with a forgetting factor of λ , to determine the parameters (or weights) w_i . Given the target data $f_{targ}(t)$ and $r(t)$, w_i is updated by

$$w_i(t+1) = w_i(t) + \Psi_i P_i(t+1) r(t) e_r(t) \quad (6)$$

² LWR is derived from a piecewise linear function approximation approach (Schaal & Atkeson, 1998), which decouples a nonlinear least-squares learning problem into several locally linear learning problems, each characterized by the local cost function J_i . These local problems can be solved with standard weighted least squares approaches.

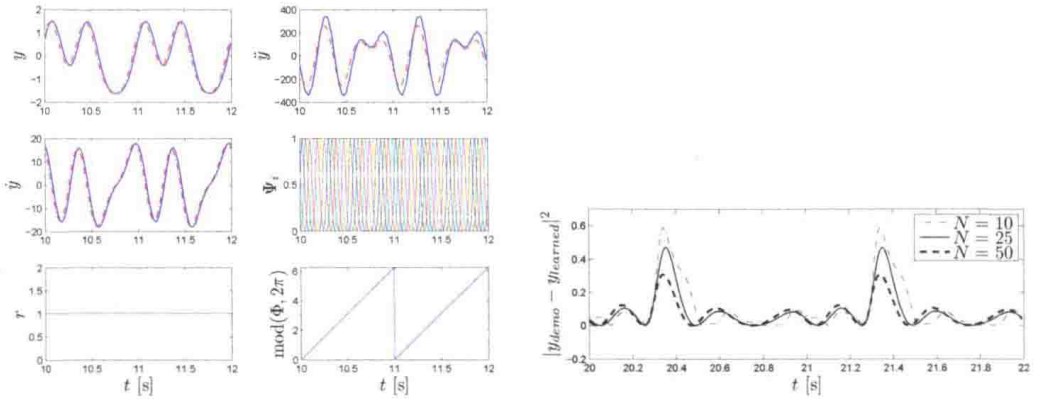


Fig. 2. *Left*: The result of Output Dynamical System with a constant frequency input and with continuous learning of the weights. In all the plots the input signal is the dash-dot line while the learned signal is the solid line. In the middle-right plot we can see the evolution of the kernel functions. The kernel functions are a function of Φ and do not necessarily change uniformly (see also Fig. 7). In the bottom right plot the phase of the oscillator is shown. The amplitude is here $r = 1$, as shown bottom-left. *Right*: The error of learning decreases with the increase of the number of Gaussian-like kernel functions. The error, which is quite small, is mainly due to a very slight (one or two sample times) delay of the learned signal.

$$P_i(t+1) = \frac{1}{\lambda} \left(P_i(t) - \frac{P_i(t)^2 r(t)^2}{\frac{\lambda}{\Psi_i} + P_i(t) r(t)^2} \right) \quad (7)$$

$$e_r(t) = f_{targ}(t) - w_i(t)r(t). \quad (8)$$

P , in general, is the inverse covariance matrix (Ljung & Söderström, 1986). The recursion is started with $w_i = 0$ and $P_i = 1$. Batch and incremental learning regressions provide identical weights w_i for the same training sets when the forgetting factor λ is set to one. Differences appear when the forgetting factor is less than one, in which case the incremental regression gives more weight to recent data (i.e. tends to forget older ones). The error of weight learning e_r (Eq. (8)) is not “related” to e when extracting frequency components (Eq. (11)). This allows for complete separation of frequency adaptation and waveform learning.

Figure 2 left shows the time evolution of the Output Dynamical System anchored to a Canonical Dynamical System with the frequency set at $\Omega = 2\pi$ rad/s, and the weight parameters w_i adjusted to fit the trajectory $y_{demo}(t) = \sin(2\pi t) + \cos(4\pi t) + 0.4\sin(6\pi t)$. As we can see in the top-left plot, the input signal and the reconstructed signal match closely. The matching between the reconstructed signal and the input signal can be improved by increasing the number of Gaussian-like functions.

Parameters of the Output Dynamical System

When tuning the parameters of the Output Dynamical System, we have to determine the number of Gaussian-like Kernel functions N , and specially the forgetting factor λ . The number N of Gaussian-like kernel functions could be set automatically if we used the locally weighted learning (Schaal & Atkeson, 1998), but for simplicity it was here set by hand. Increasing the number increases the accuracy of the reconstructed signal, but at the same time also increases the computational cost. Note that LWR does not suffer from problems of overfitting when the

number of kernel functions is increased.³ Figure 2 right shows the error of learning e_r when using $N = 10$, $N = 25$, and $N = 50$ on a signal $y_{demo}(t) = 0.65\sin(2\pi t) + 1.5\cos(4\pi t) + 0.3\sin(6\pi t)$. Throughout the paper, unless specified otherwise, $N = 25$.

The forgetting factor $\lambda \in [0, 1]$ plays a key role in the behavior of the system. If it is set high, the system never forgets any input values and learns an average of the waveform over multiple periods. If it is set too low, it forgets all, basically training all the weights to the last value. We set it to $\lambda = 0.995$.

2.2 Canonical dynamical system

The task of the Canonical Dynamical System is two-fold. Firstly, it has to extract the fundamental frequency Ω of the input signal, and secondly, it has to exhibit stable limit cycle behavior in order to provide a phase signal Φ , that is used to anchor the waveform of the output signal. Two approaches are possible, either with a pool of oscillators (PO), or with an adaptive Fourier Series (AF).

2.2.1 Using a pool of oscillators

As the basis of our canonical dynamical system we use a set of phase oscillators, see e.g. (Buchli et al., 2006), to which we apply the adaptive frequency learning rule as introduced in (Buchli & Ijspeert, 2004) and (Righetti & Ijspeert, 2006), and combine it with a feedback structure (Righetti et al., 2006) shown in Figure 3. The basic idea of the structure is that each of the oscillators will adapt its frequency to one of the frequency components of the input signal, essentially “populating” the frequency spectrum.

We use several oscillators, but are interested only in the fundamental or lowest non-zero frequency of the input signal, denoted by Ω , and the phase of the oscillator at this frequency, denoted by Φ . Therefore the feedback structure is followed by a small logical block, which chooses the correct, lowest non-zero, frequency. Determining Ω and Φ is important because with them we can formulate a supervised learning problem in the second stage - the Output Dynamical System, and learn the waveform of the full period of the input signal.

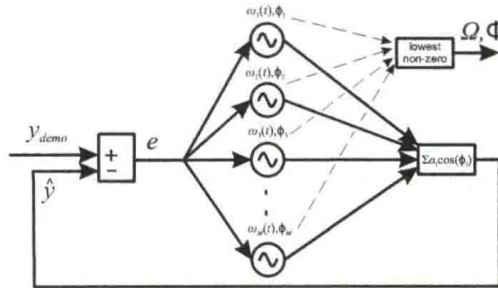


Fig. 3. Feedback structure of a network of adaptive frequency phase oscillators, that form the Canonical Dynamical System. All oscillators receive the same input and have to be at different starting frequencies to converge to different final frequencies. Refer also to text and Eqs. (9-13).

The feedback structure of M adaptive frequency phase oscillators is governed by the following equations:

³ This property is due to solving the bias-variance dilemma of function approximation locally with a closed form solution to leave-one-out cross-validation (Schaal & Atkeson, 1998).

$$\dot{\phi}_i = \omega_i - Ke \sin(\phi_i) \quad (9)$$

$$\dot{\omega}_i = -Ke \sin(\phi_i) \quad (10)$$

$$e = y_{demo} - \hat{y} \quad (11)$$

$$\hat{y} = \sum_{i=1}^M \alpha_i \cos(\phi_i) \quad (12)$$

$$\dot{\alpha}_i = \eta \cos(\phi_i) e \quad (13)$$

where K is the coupling strength, ϕ_i is the phase of oscillator i , e is the input into the oscillators, y_{demo} is the input signal, \hat{y} is the weighted sum of the oscillators' outputs, M is the number of oscillators, α_i is the amplitude associated to the i -th oscillator, and η is a learning constant. In the experiments we use $K = 20$ and $\eta = 1$, unless specified otherwise.

Eq. (9) and (10) present the core of the Canonical Dynamical System – the adaptive frequency phase oscillator. Several (M) such oscillators are used in a feedback loop to extract separate frequency components. Eq. (11) and (12) specify the feedback loop, which needs also amplitude adaptation for each of the frequency components (Eq. (13)).

As we can see in Figure 3, each of the oscillators of the structure receives the same input signal, which is the difference between the signal to be learned and the signal already learned by the feedback loop, as in Eq. (11). Since a negative feedback loop is used, this difference approaches zero as the weighted sum of separate frequency components, Eq. (12), approaches the learned signal, and therefore the frequencies of the oscillators stabilize. Eq. (13) ensures amplitude adaptation and thus the stabilization of the learned frequency. Such a feedback structure performs a kind of dynamic Fourier analysis. It can learn several frequency components of the input signal (Righetti et al., 2006) and enables the frequency of a given oscillator to converge as $t \rightarrow \infty$, because once the frequency of a separate oscillator is set, it is deducted from the demonstration signal y_{demo} , and disappears from e (due to the negative feedback loop). Other oscillators can thus adapt to other remaining frequency components. The populating of the frequency spectrum is therefore done without any signal processing, as the whole process of frequency extraction and adaptation is totally embedded into the dynamics of the adaptive frequency oscillators.

Frequency adaptation results for a time-varying signal are illustrated in Figure 4, left. The top plot shows the input signal y_{demo} , the middle plot the extracted frequencies, and the bottom plot the error of frequency adaptation. The figure shows results for both approaches, using a pool of oscillators (PO) and for using one oscillator and an adaptive Fourier series (AF), explained in the next section. The signal itself is of three parts, a non-stationary signal (presented by a chirp signal), followed by a step change in the frequency of the signal, and in the end a stationary signal. We can see that the output frequency stabilizes very quickly at the (changing) target frequency. In general the speed of convergence depends on the coupling strength K (Righetti et al., 2006). Besides the use for non-stationary signals, such as chirp signals, coping with the change in frequency of the input signal proves especially useful when adapting to the frequency of hand-generated signals, which are never stationary. In this particular example, a single adaptive frequency oscillator in a feedback loop was enough, because the input signal was purely sinusoidal.

The number of adaptive frequency oscillators in a feedback loop is therefore a matter of design. There should be enough oscillators to avoid missing the fundamental frequency and to limit the variation of frequencies described below when the input signal has many