A General Family of Morphed Nonlinear Phase Oscillators with Arbitrary Limit Cycle Shape

Mostafa Ajallooeian[a,*], Jesse van den Kieboom[a], Albert Mukovskiy[b], Martin A. Giese[b], Auke J. Ijspeert[a]

[a] *Biorobotics Laboratory, School of Engineering, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland*
[b] *Section for Computational Sensomotorics, Department of Cognitive Neurology, Hertie Institute for Clinical Brain Research & Center for Integrative Neuroscience, University Clinic Tübingen, Germany*

Abstract

We present a general family of nonlinear phase oscillators which can exhibit arbitrary limit cycle shapes and infinitely large basins of attraction. This general family is the superset of familiar control methods like PD-control over a periodic reference, and rhythmic Dynamical Movement Primitives. The general methodology is based on morphing the limit cycle of an existing phase oscillator with phase-based scaling functions to obtain a desired limit cycle behavior. The introduced methodology can be represented as first, second, or $n-$th order dynamical systems. The elegance of the formulation provides the possibility to define explicit arbitrary convergence behavior for simple cases. We analyze the stability properties of the methodology with the Poincaré-Bendixson theorem and the Contraction Theory, and use numerical simulations to show the properties of some oscillators that are a subset of this general family.

Keywords:
nonlinear oscillator, limit cycle shape, convergence behavior, Poincaré-Bendixson theorem, Contraction Theory.

1. Introduction

Nonlinear oscillators have been widely used in different fields ranging from abstract to applied sciences and engineering [1]. Their capability of entrainment, synchronization and smooth modulation of their output signal makes them appropriate tools for modeling natural phenomena and for control [1–4].

One challenge in the field of nonlinear oscillators is how to design oscillators with desired limit cycle shapes [4]. One key application of such oscillators is in system and controller design. The benefit of using nonlinear oscillators in controller design is that the control references can be coded into dynamical systems, and if done properly, properties like smoothness, continuity and stability after state perturbation are preserved. Now if the desired control references are functions of desired shapes or structures, then there is a need for oscillators capable of having arbitrary limit cycle shapes.

For example, nonlinear oscillators have been extensively used to model animal locomotion [5–8], or create controllers for robot locomotion [9–13]. But due to the lack of a concrete, general and easy-to-use methodology, many of the aforementioned approaches had to utilize either 1) oscillators which exhibit simple predetermined limit cycles, e.g. with the shape of a sine-wave, and consequently could not accurately encode the complex/desired control trajectories into oscillatory dynamics [14–17]; or 2) oscillators augmented with functions to shape the output signal ($f(y)$ with y being the oscillator output and f the shaping), which are not appropriate for state feedback integration (no smooth output dynamics) [18–20]; or 3) complex connectionist oscillators which are rather hard to design [21–24]; or 4) property and order-specific oscillators like rhythmic Dynamical Movement Primitives which only give one possible solution to the problem (forced second order linear ODE) [25, 26]. This motivates us to propose a simple methodology which results in a general family of nonlinear phase oscillators exhibiting arbitrary limit cycle shapes, with a wide choice of oscillator properties including the order of the dynamical system and the convergence behavior.

*EPFL STI IBI BIOROB, INN 235, Station 14, CH-1015 Lausanne, Switzerland. Tel: +41 21 693 26 76. Email: mostafa.ajallooeian@epfl.ch

Nonlinear oscillators with arbitrary limit cycle shape can be built with recurrent neural networks (RNN) [27–34]. However, it is not trivial to calculate the connection weights, analyze the asymptotic stability of the resulting network, and predict the behavior of it [30]. Nevertheless, there are reservoir computing approaches to RNN [35] which provide ways to create nonlinear oscillators without complicated training. Wyffels et al. [24] presented a randomly connected RNN with a linear readout which can encode arbitrary limit cycles, and used ridge regression to find the readout weights, yet the problem of asymptotic stability remains unsolved.

Some researchers have tried to create nonlinear oscillators with an arbitrary limit cycle shape by using fitting tools on data-driven generated vector fields. Okada et al. [36] presented a method to represent the desired trajectory as a limit cycle and define a corresponding vector field in the vicinity of each data point directed towards the limit cycle. They use a polynomial approximation of the vector field to create the dynamical system encoding the desired limit cycle. Similarly, Ajallooeian et al. [37] use the desired trajectory as the limit cycle and define margins of attraction based on the Poincaré-Bendixson theorem. They map the desired periodic trajectory on the limit cycle of a stable oscillator (and vice versa) to strengthen the stability properties, and employ feedforward neural networks to create the maps. Both [36] and [37] create dynamical systems which are trained to be (asymptotically) stable in the margins defined around the limit cycle, but not necessarily outside of these margins.

Righetti et al. [13] presented Adaptive Frequency Oscillators and used a pool of them to create nonlinear oscillators with an arbitrary limit cycle shape. They utilize a dynamic estimation of the phases of Fourier harmonics constructed from a pool of adaptive Hopf oscillators [38], and build the desired dynamics. The output of their model is the weighted sum of local oscillators, which makes the integration of the proprioceptive feedback difficult (also known as the credit assignment problem [39]). The final output shape of such system is limited, unless an infinite number of oscillators is used. Also a modified version of this approach was presented in [40] to use a single oscillator and an adaptive Fourier series.

One other interesting approach to create custom nonlinear oscillators is to use a virtual linear spring and periodically force it such that the desired output is generated. Ijspeert et al. [41–43] presented rhythmic Dynamical Movement Primitives (DMP) which implements this idea. Rhythmic DMP gives a nice formulation of a phase oscillator with an arbitrary limit cycle shape, and was later coupled with Adaptive Frequency Oscillators [44]. Here in this paper we give a general representation which includes rhythmic DMP as an example realization.

We take inspiration from the works mentioned above, especially from our previously collaborated works [37] and [41–43], and present a general way to convert an existing phase oscillator to any desired nonlinear phase oscillator with well defined, controllable properties. We believe that our contribution is three fold: 1) We present a *general*, *simple* and *systematic* way to design nonlinear phase oscillators. As a consequence, we obtain not a single, but a family of nonlinear phase oscillators exhibiting desired limit cycle shapes. This gives researchers the ability to easily design custom nonlinear phase oscillators. 2) The general methodology we present here can be taken as a unifying view on well known techniques including PD control and rhythmic Dynamical Movement Primitives. This enables the comparison of these methods under a same conceptual framework. 3) The mathematical formulation gives room to introduce a simple nonlinear oscillator which can have a *custom convergence behavior*, independent of the other properties like arbitrary limit cycle shape. This is a useful property providing the possibility to define the way to converge to the limit cycle, which is potentially useful for initiation in many cyclic motion control problems.

The rest of this document is organized as follows: The design methodology is explained in Section 2. Example realizations of the introduced methodology is given in Section 3. Extension to multi-dimensions is explained in Section 4. We then analyze the stability conditions in Section 5. Section 6 details how arbitrary convergence behavior can be obtained. Section 7 discusses building nonlinear phase oscillators from data. Section 8 describes the applications of the morphed oscillators, and we summarize and discuss our work in Section 9.

2. Methodology

There are many nonlinear oscillators with different characteristics, including Hopf oscillator [1], van der Pol oscillator [1], Fitzhugh-Nagumo oscillator [45], and many more, and each of these oscillators exhibit a different limit cycle shape. However, it is not trivial to design an oscillator with a *desired* limit cycle shape. We propose a systematic way to design phase oscillators (oscillators with an explicit phase variable, such as a Hopf oscillator expressed in polar coordinates, for instance) with arbitrary limit cycle shape.

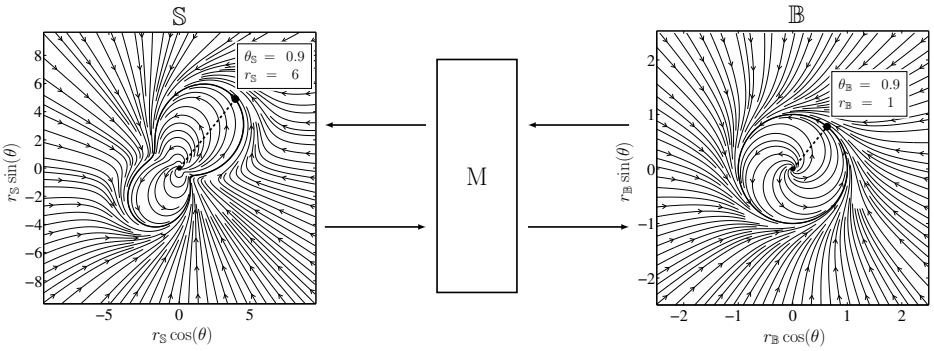

Figure 1: Example of a mapping between the desired oscillator (left) and the base oscillator (right). The main idea is to select an existing structurally stable oscillator, like the Hopf oscillator (right), and find a mapping (M) which shapes the limit cycle of this oscillator to a desired one (left). A phase-based scaling function $f(\theta)$ can implement M to map points from $\mathbb{B}$ to $\mathbb{S}$ and vice versa. For example, for the two points shown, the scaling function is $f(0.9) = \frac{6}{1}$.

One can take a simple phase oscillator, and try to morph this oscillator's limit cycle to obtain a desired one. So a mapping which takes the states of this oscillator and modifies them such that the outcome has the desired property is needed. This mapping can be arbitrarily complex, but in case of the phase oscillators, we show that scaling the radial state depending on the phase value is sufficient to modify the limit cycle shape.

Our methodology consists of an oscillator called the *base oscillator* (in phase plane $\mathbb{B}$, real plane $\mathbb{R}^2$), a *desired oscillator* (in phase plane $\mathbb{S}$, real plane $\mathbb{R}^2$) and a mapping between them (M). We aim to find invertible M such that any path in $\mathbb{B}$ is mapped to a unique path in $\mathbb{S}$ and vice versa. Consequently, with the correct choice of M, the limit cycle of the base oscillator in $\mathbb{B}$ will be mapped to the desired one in $\mathbb{S}$.

Let us assume that the desired limit cycle is defined by the mapping $\Xi_{\mathbb{S}} : \theta_{\mathbb{S}} \rightarrow r_{\mathbb{S}}$, with $\theta_{\mathbb{S}}$ and $r_{\mathbb{S}}$ respectively being the angle and radius in polar coordinates. Also assume a base oscillator for which its limit cycle can be defined in polar coordinates by the mapping $\Xi_{\mathbb{B}} : \theta_{\mathbb{B}} \rightarrow r_{\mathbb{B}}$. Now if the oscillator in $\mathbb{S}$ has a phase always equal to the base oscillator's phase, i.e. $\theta_{\mathbb{S}} = \theta_{\mathbb{B}} = \theta$, then the radius of the limit cycle of the oscillator in $\mathbb{S}$ can be defined with a *phase-based scaling* (also see Figure 1) :

$$r_{\mathbb{S}} = f(\theta) r_{\mathbb{B}} \tag{1}$$

where f is a scaling function that scales based on the phase value. So a state $\{\theta, r_{\mathbb{S}}\}$ in $\mathbb{S}$ corresponds to a state $\{\theta, \frac{r_{\mathbb{S}}}{f(\theta)}\}$ in $\mathbb{B}$, and vice versa.

The mapping from space $\mathbb{B}$ to $\mathbb{S}$ can be written as $M(\theta, r) = (\theta, rf(\theta))$, and calculating the Jacobian determinant of M gives $|J| = f(\theta)$. So if $\forall\theta : f(\theta) \neq 0$ and f is C^1 differentiable, then M defines a C^1-diffeomorphism [1] between $\mathbb{B}$ and $\mathbb{S}$. This means that each state in $\mathbb{S}$ corresponds to a *unique* state in $\mathbb{B}$ and vice versa. Having the diffeomorphism property, an intuitive way to form the desired limit cycle system in $\mathbb{S}$ is:

---

1. Take the current state $\{\theta(t), r_{\mathbb{S}}(t)\}$ in $\mathbb{S}$
2. Map this state to $\mathbb{B}$ using $r_{\mathbb{B}}(t) = \frac{r_{\mathbb{S}}(t)}{f(\theta(t))}$
3. Use the dynamical system of the base oscillator in $\mathbb{B}$ to calculate $r_{\mathbb{B}}(t + \Delta t)$ and $\theta(t + \Delta t)$
4. Map $r_{\mathbb{B}}(t + \Delta t)$ back to $\mathbb{S}$ by $r_{\mathbb{S}}(t + \Delta t) = f(\theta(t + \Delta t)) r_{\mathbb{B}}(t + \Delta t)$
5. Do $t \xleftarrow{update} t + \Delta t$ and continue from 1.

With the above algorithm we can mathematically write the process of the limit cycle generation as an iterative map, i.e. a discrete-time dynamical system (the continuous-time description will be given later):

$$\theta(t + \Delta t) = \theta(t) + \int_t^{t+\Delta t} D_{\mathbb{B},\theta}\left(\theta(t)\right) dt \tag{2}$$

$$r_{\mathbb{S}}(t + \Delta t) = f(\theta(t + \Delta t)) \left(\frac{r_{\mathbb{S}}(t)}{f(\theta(t))} + \int_t^{t+\Delta t} D_{\mathbb{B},r}\left(\frac{r_{\mathbb{S}}(t)}{f(\theta(t))} \right) dt \right) \tag{3}$$

where $D_{\mathbb{B},\{r,\theta\}}(.)$ are the differential equations of the base oscillator ($\dot{\theta} = D_{\mathbb{B},\theta}(\theta), \dot{r}_{\mathbb{B}} = D_{\mathbb{B},r}(r_{\mathbb{B}})$). Since the base oscillator is a phase oscillator ($\dot{\theta} = \omega = \text{const}$) we have:

$$D_{\mathbb{B},\theta}(.) = \omega = 2\pi\vartheta \tag{4}$$

where ϑ is the oscillator's frequency. Equations (2,3) introduce a general way to convert a chosen base oscillator to one with a desired limit cycle. The shape of the limit cycle in $\mathbb{S}$ is $\Xi_{\mathbb{S}}(\theta) = \Xi_{\mathbb{B}}(\theta)f(\theta)$.

2.1. Continuous-time dynamical system

The oscillators obtained by Equations (2,3) are with discrete-time updates. A continuous-time dynamical system form can be obtained by calculating the limits of the forward differences of those equations when $\Delta t \to 0$. For θ we simply have (using Euler approximation for integration):

$$\dot{\theta}(t) = \lim_{\Delta t \to 0} \frac{1}{\Delta t}\left(\theta(t) + \left(\int_t^{t+\Delta t} D_{\mathbb{B},\theta}\left(\theta(t)\right) dt\right) - \theta(t)\right) = D_{\mathbb{B},\theta}(\theta(t)) = \omega = 2\pi\vartheta \tag{5}$$

which is the definition of the phase dynamics of the base oscillator in $\mathbb{B}$. For $\dot{r}_{\mathbb{S}}(t)$ we have:

$$\dot{r_{\mathbb{S}}}(t) = \lim_{\Delta t \to 0}\left(\frac{r_{\mathbb{S}}(t)}{f(\theta(t))}\frac{f(\theta(t+\Delta t)) - f(\theta(t))}{\Delta t} + \frac{f(\theta(t+\Delta t)D_{\mathbb{B},r}(\frac{r_{\mathbb{S}}(t)}{f(\theta(t))})\Delta t}{\Delta t}\right) \tag{6}$$

which gives:

$$\boxed{\dot{r_{\mathbb{S}}}(t) = \frac{r_{\mathbb{S}}(t)}{f(\theta(t))}\dot{f}(\theta(t)) + f(\theta(t))D_{\mathbb{B},r}\left(\frac{r_{\mathbb{S}}(t)}{f(\theta(t))}\right)} \tag{7}$$

with $\dot{f}(\theta(t)) = \frac{f(\theta(t))}{d\theta(t)}\frac{d\theta(t)}{dt} = \omega f'(\theta(t))$. Equation (5) along with (7) gives a general way to obtain a desired continuous-time phase oscillator. The shape of the limit cycle can be arbitrarily defined by $\Xi_{\mathbb{S}}(\theta) = \Xi_{\mathbb{B}}(\theta)f(\theta)$ (either $f(\theta)$ is defined and then $\Xi_{\mathbb{S}}(\theta)$ is derived, or vice versa). The convergence behavior is defined by the dynamics of the base oscillator. We call the set of Equations (5,7) the *original form*.

The canonical evolution in Equation (7) is $\frac{r_{\mathbb{S}}(t)}{f(\theta(t))}\dot{f}(\theta(t))$, meaning that the amplitude of the oscillations generated by $\dot{f}(\theta)$ are magnified by $\frac{r_{\mathbb{S}}(t)}{f(\theta(t))}$ (Figure 2-top, dashed blue lines). So for large $r_{\mathbb{S}}(t)$ values, the amplitude of the oscillations will be largely magnified. If this effect is undesirable, then one can compensate for this effect by using the following radial differential equation:

$$\boxed{\dot{r_{\mathbb{S}}}(t) = \Xi_{\mathbb{B}}(\theta(t))\dot{f}(\theta(t)) + f(\theta(t))D_{\mathbb{B},r}\left(\frac{r_{\mathbb{S}}(t)}{f(\theta(t))}\right)} \tag{8}$$

where $\Xi_{\mathbb{B}}(\theta(t))$ is the shape of the base oscillator's limit cycle. We call the set of Equations (5,8) the *compensated form*. State-time evolution and phase portraits of both original and compensated forms are depicted in Figure 2, for same desired limit cycle. As shown, the difference between these two forms is in their convergence behavior.

As Figure 2 depicts, for the compensated form, the $r_{\mathbb{S}}$ state can become negative (Figure 2-top, at $t \simeq 0.6$) even starting from positive initial values. This means that negative radius values are meaningful without shifting to the antiphase state. In this paper we term the space formed by $\{\theta, r_{\mathbb{S}}\}$ as *extended polar coordinates*, and discuss this more in Section 5 when analyzing stability conditions.

The obtained oscillator forms are first order dynamical systems. Positions $r_{\mathbb{S}}(t)$ and velocities $\dot{r}_{\mathbb{S}}(t)$ are continuous if not directly perturbed. If f is additionally C^2 differentiable, then accelerations $\ddot{r}_{\mathbb{S}}(t)$ are continuous as well. In general, if f is C^n differentiable, consequently $r_{\mathbb{S}}^{(n)}(t)$ values are continuous, unless directly perturbed.

Table 1 summarizes the needed formulae to create continuous-time morphed oscillators, and also gives a simple example. One only needs to define a base oscillator, and choose the desired limit cycle $\Xi_{\mathbb{S}}(\theta)$ to obtain the compact equations of the desired morphed oscillator.

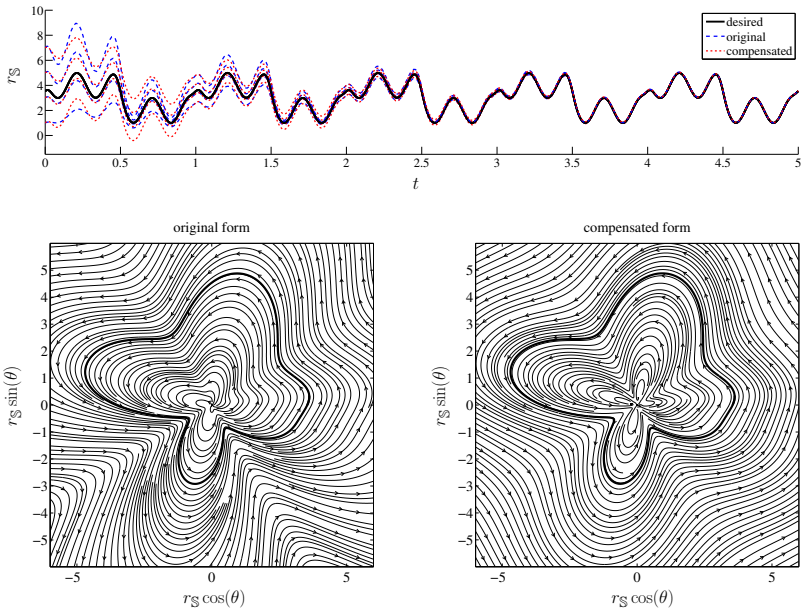

Figure 2: Comparison between the original and the compensated forms. The base oscillator used is an amplitude controlled oscillator with $\mu = 1$ and $\gamma = 1$ (please refer to Table 2 for the base oscillator's equations). The desired limit cycle is $\Xi_{\mathbb{S}}(\theta) = 3 + \tanh(5\sin(\theta)) + \cos(4\theta + 1)$. The contours of the original form are similar to the limit cycle and scaled by $\frac{r_{\mathbb{S}}(t)}{f(\theta(t))}$, while the contours of the compensated form become circular when they are away from the limit cycle. The phase portraits are only for $r_{\mathbb{S}} > 0$.

Note: From now on and for the rest of this paper: 1) We will use the compensated form (Equations (5,8)) to extend for higher dynamical system orders and dimension. It is possible to rewrite the same procedures for the original form in the same way; 2) We remove the time indexing "$.(t)$" for brevity, and will only mention time if there is an explicit time dependency. So terms like $r_{\mathbb{S}}(t)$, $\theta(t)$ and $f(\theta(t))$ will respectively become $r_{\mathbb{S}}$, θ and $f(\theta)$; 3) All the phase portraits are illustrated only for $r_{\mathbb{S}} > 0$, unless mentioned otherwise.

2.2. Second order dynamical system

Equations (5, 8) define a first order dynamical system. Equation (5) drives the phase dynamics while Equation (8) controls the radial dynamics. If one desires to directly control acceleration, or create the possibility to add feedback mechanisms on the acceleration dynamics, then a second order dynamical system is needed. One can rewrite Equation (8) for a new velocity state ($v_{\mathbb{S}}$), and take the velocity profile, instead of position profile, as the limit cycle. The desired velocity limit cycle is $\dot{\Xi}_{\mathbb{S}}(\theta)$, so:

$$
\begin{aligned}
f_v(\theta) &= \frac{\dot{\Xi}_{\mathbb{S}}(\theta)}{\Xi_{\mathbb{B}}(\theta)} = \frac{\dot{\Xi}_{\mathbb{B}}(\theta)}{\Xi_{\mathbb{B}}(\theta)} f(\theta) + \dot{f}(\theta) \\
\dot{v_{\mathbb{S}}} &= \Xi_{\mathbb{B}}(\theta)\dot{f}_v(\theta) + (f_v(\theta) + \delta)\, D_{\mathbb{B},r}\left(\frac{v_{\mathbb{S}}}{f_v(\theta) + \delta}\right)
\end{aligned}
\tag{9}
$$

Table 1: Continuous-time first order morphed oscillators. Two form of oscillators is obtained: the original form, and the compensated form. $\Xi_{\mathbb{S}}(\theta)$ is the desired limit cycle, $\Xi_{\mathbb{B}}(\theta)$ is the limit cycle of the base oscillator, $f(\theta)$ is the phase-based scaling function, ω is the oscillation frequency multiplied by 2π, γ controls the rate of convergence, θ is the phase of the oscillator, and $r_{\mathbb{S}}$ is the radial state and also the desired output of the system which converges to the desired limit cycle $\Xi_{\mathbb{S}}(\theta)$.

	Original form	Compensated form
Morphed oscillator equations	$\dot{\theta} = \omega$ $\dot{r_{\mathbb{S}}} = \frac{r_{\mathbb{S}}}{f(\theta)}\dot{f}(\theta) + f(\theta) D_{\mathbb{B},r}\left(\frac{r_{\mathbb{S}}}{f(\theta)}\right)$	$\dot{\theta} = \omega$ $\dot{r_{\mathbb{S}}} = \Xi_{\mathbb{B}}(\theta)\dot{f}(\theta) + f(\theta) D_{\mathbb{B},r}\left(\frac{r_{\mathbb{S}}}{f(\theta)}\right)$
Example: desired limit cycle $\Xi_{\mathbb{S}}(\theta) = e^{sin(\theta)}$ and base oscillator $D_{\mathbb{B},r}(r) = \gamma(\mu - r)$ (so $f(\theta) = \frac{\Xi_{\mathbb{S}}(\theta)}{\Xi_{\mathbb{B}}(\theta)} = \frac{1}{\mu}e^{sin(\theta)}$)	$\dot{\theta} = \omega$ $\dot{r_{\mathbb{S}}} = \omega\cos(\theta) r_{\mathbb{S}} + \gamma(e^{sin(\theta)} - r_{\mathbb{S}})$	$\dot{\theta} = \omega$ $\dot{r_{\mathbb{S}}} = \omega\cos(\theta) e^{sin(\theta)} + \gamma(e^{sin(\theta)} - r_{\mathbb{S}})$

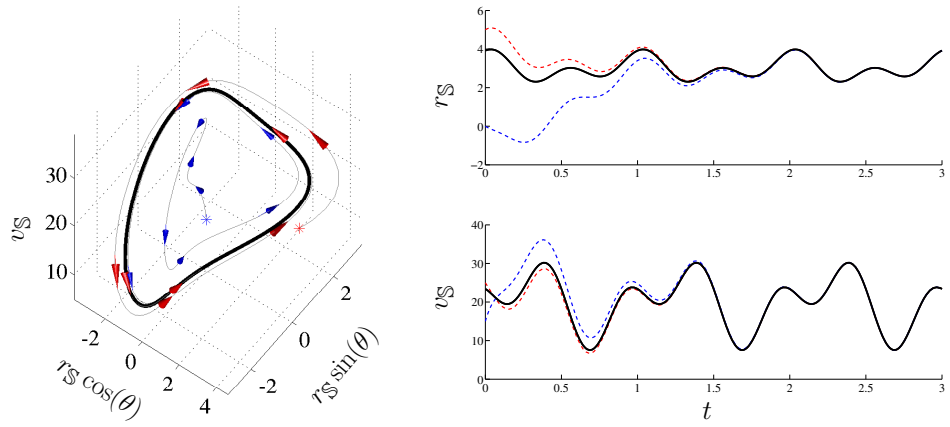

Figure 3: An example second order system. The desired limit cycle is $\Xi_{\mathbb{S}}(\theta) = 3 + 0.5(\cos(\theta) + \sin(2\theta + 1))$. The base oscillator is defined by $\dot{r}_{\mathbb{B}} = 2\pi\vartheta\cos(\theta) + 2 + \sin(\theta) - r_{\mathbb{B}}$, which has the limit cycle $\Xi_{\mathbb{B}}(\theta) = 2 + \sin(\theta)$. Consequently $f(\theta) = \frac{\Xi_{\mathbb{S}}(\theta)}{2+\sin(\theta)}$. For this figure $\gamma = 10$, $\vartheta = 1$ and $\beta = 20$. Asterix $(*)$ symbols indicate the initial points in the trajectories. Blue and red trajectories are the same for state-time and phase plots.

where a constant $\delta > -\min_{\theta}(f_v(\theta))$ is added to $f_v(\theta)$ to keep the scaling function strictly positive. The oscillator obtained by Equations (5,9) will follow a desired velocity profile, but yet we need an equation for $\dot{r}_{\mathbb{S}}$. Additionally, unwanted position offsets (errors on $r_{\mathbb{S}} = \int v_{\mathbb{S}}\, dt$) are not forgotten in the system defined by Equation (9). We add a phase-based attractor force field to $\dot{v}_{\mathbb{S}}$ to damp the unwanted position offsets, and attract to the desired limit cycle in $\mathbb{S}$, which is $\Xi_{\mathbb{S}}(\theta) = \Xi_{\mathbb{B}}(\theta)f(\theta)$. By doing so we obtain:

$$\dot{v}_{\mathbb{S}} = \Xi_{\mathbb{B}}(\theta)\dot{f}_v(\theta) + (f_v(\theta) + \delta)\, D_{\mathbb{B},r}\left(\frac{v_{\mathbb{S}}}{f_v(\theta) + \delta}\right) + \beta\left(\Xi_{\mathbb{B}}(\theta)f(\theta) - r_{\mathbb{S}}\right) \tag{10}$$

$$\dot{r}_{\mathbb{S}} = v_{\mathbb{S}} - \delta\Xi_{\mathbb{B}}(\theta) \tag{11}$$

where β determines the strength of the attractor force field defined by $\Xi_{\mathbb{B}}(\theta)f(\theta) - r_{\mathbb{S}}$ term. Also since a δ offset is added to f_v, the velocity limit cycle is having an offset of $\delta\Xi_{\mathbb{B}}(\theta)$, which is subtracted from $v_{\mathbb{S}}$ in Equation (11). Equations (5,10,11) together give a general second order phase oscillator which exhibits a desired limit cycle behavior. It should be noted that as long as the base oscillator's limit cycle is circular $\dot{\Xi}_{\mathbb{B}}(\theta) = 0$, and Equations (10,11) can be written as a simpler form where $f_v(\theta) = \dot{f}(\theta)$ and $\dot{f}_v(\theta) = \ddot{f}(\theta)$. Figure 3 illustrates an example state time evolution and phase portrait of a compensated second-order system.

2.3. n−th order dynamical system

The idea of extending to a second order system can be generalized to create an n−th order dynamical system. The n−th order radial state is controlled by the base oscillator. All radial differential equations are augmented by attractors from states which are one order lower ($\dot{r}^m_{\mathbb{S}}$ is augmented with an attractor on $r^{(m-1)}_{\mathbb{S}}$):

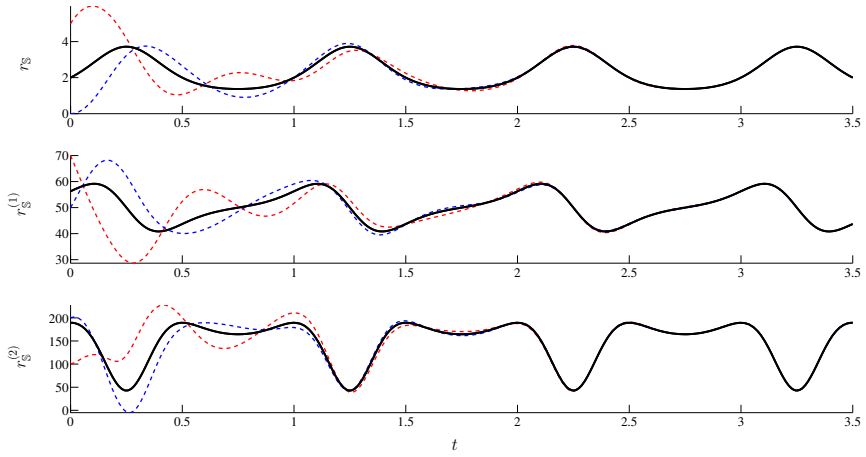

Figure 4: Example state-time evolution of a 3rd order system. The base oscillator is defined with $\dot{\theta} = 2\pi\vartheta$, $\dot{r}_{\mathbb{B}} = \gamma \tanh(\mu - r_{\mathbb{B}})$. The desired limit cycle is $\Xi_{\mathbb{S}}(\theta) = 1 + e^{\sin(\theta)}$. For this example $\vartheta = 1$, $\mu = 1$, $\gamma = 10$, $\beta_1 = \beta_2 = 50$, $\delta_1 = 50$ and $\delta_2 = 150$.

$$
\begin{aligned}
\dot{r}_{\mathbb{S}}^{(n-1)} = \Xi_{\mathbb{B}}(\theta)\dot{f}_{(n-1)}(\theta) + \left(f_{(n-1)}(\theta) + \delta_{n-1}\right) D_{\mathbb{B},r}\left(\frac{r_{\mathbb{S}}^{(n-1)}}{f_{(n-1)} + \delta_{n-1}(\theta)}\right) + \beta_{n-1}\left(\Xi_{\mathbb{B}}(\theta) f_{(n-2)}(\theta) + \delta_{n-2} - r_{\mathbb{S}}^{(n-2)}\right) \\
\dot{r}_{\mathbb{S}}^{(n-2)} = r_{\mathbb{S}}^{(n-1)} - \delta_{n-1}\Xi_{\mathbb{B}}(\theta) + \beta_{n-2}\left(\Xi_{\mathbb{B}}(\theta) f_{(n-3)}(\theta) + \delta_{n-3} - r_{\mathbb{S}}^{(n-3)}\right) \\
\dot{r}_{\mathbb{S}}^{(n-3)} = r_{\mathbb{S}}^{(n-2)} - \delta_{n-2} + \beta_{n-3}\left(\Xi_{\mathbb{B}}(\theta) f_{(n-4)}(\theta) + \delta_{n-4} - r_{\mathbb{S}}^{(n-4)}\right) \\
\vdots \\
\dot{r}_{\mathbb{S}}^{(1)} = r_{\mathbb{S}}^{(2)} - \delta_2 + \beta_1\left(\Xi_{\mathbb{B}}(\theta) f(\theta) - r_{\mathbb{S}}\right) \\
\dot{r}_{\mathbb{S}} = r_{\mathbb{S}}^{(1)} - \delta_1
\end{aligned}
\tag{12}
$$

where $f_{(m)}(\theta) = \frac{d^m \Xi_{\mathbb{S}}(\theta)}{dt^m} \frac{1}{\Xi_{\mathbb{B}}(\theta)}$, $\delta_m > -\min_{\theta}\left(f_{(m)}(\theta)\right)$, and β_m is the strength of the force field on the m-th order which damps the unwanted offsets of $r_{\mathbb{S}}^{(m-2)}$. $r_{\mathbb{S}}, r_{\mathbb{S}}^{(1)}, \ldots, r_{\mathbb{S}}^{(n-1)}$ are system states corresponding respectively to position, velocity, acceleration, and so on. Figure 4 depicts an example state-time evolution for a 3rd order system.

3. Realization

The previous section gave a general methodology to design morphed phase oscillators with arbitrary limit cycle shapes. The obtained forms represent a general family of morphed oscillators that are parametrized by the radial equation of the base oscillator $D_{\mathbb{B},r}(.)$. Realizations are obtained by using desired base oscillators, defining $D_{\mathbb{B},r}(.)$.

Tables 2-3 show first and second order realizations of this methodology with different base oscillators. First a base oscillator is chosen, and then Equation (8) or Equations (10,11) are used to obtain the morphed oscillator. For all the examples given $\dot{\theta} = 2\pi\vartheta$, γ controls the rate of convergence, and μ corresponds to the radius of the limit cycle in $\mathbb{B}$.

Figure 5 illustrates phase portraits of these oscillators as well as their state-time evolution (both for first order realizations). The desired limit cycle $\Xi_{\mathbb{S}}(\theta)$ is chosen to be the same for all the examples to enable comparison. One should keep in mind that although $\Xi_{\mathbb{S}}(\theta)$ is the same for all the examples, the corresponding f functions are not necessarily the same, because $\Xi_{\mathbb{B}}(\theta)$ is different for different base oscillators (and we know $f(\theta) = \frac{\Xi_{\mathbb{S}}(\theta)}{\Xi_{\mathbb{B}}(\theta)}$).

The basin of attraction in examples (I) and (IV) is the whole state space, while this is not true for examples (II) and (III). Example (II) uses a Hopf oscillator as base, which has two limit cycles at $r_{\mathbb{B}}(\theta) = \pm\sqrt{\mu}$, and states who enter the basin of attraction of the unwanted limit cycle at $r_{\mathbb{B}}(\theta) = -\sqrt{\mu}$ will not converge to the desired limit cycle. Finally, for example (III), the solution diverges initiating with $r_{\mathbb{S}}|_{t=0} = -2$. We will discuss the stability conditions in Section 5 and see when the oscillator in example (III) gets unstable.

Table 2: Example first order realizations with different base oscillators.

Base oscillator	**Compensated first order realization**
(I) Amplitude controlled oscillator $\dot{r}_{\mathbb{B}} = \gamma\,(\mu - r_{\mathbb{B}})$	$\dot{r_{\mathbb{S}}} = \mu\dot{f}(\theta) + \gamma\,(\mu f(\theta) - r_{\mathbb{S}})$
(II) Hopf oscillator $\dot{r}_{\mathbb{B}} = \gamma(\mu - r_{\mathbb{B}}^2)r_{\mathbb{B}}$	$\dot{r_{\mathbb{S}}} = \sqrt{\mu}\dot{f}(\theta) + \gamma\left(\mu - \left(\frac{r_{\mathbb{S}}}{f(\theta)}\right)^2\right)r_{\mathbb{S}}$
(III) Logarithmic saturated oscillator $\dot{r}_{\mathbb{B}} = \gamma\tanh(\mu - r_{\mathbb{B}}^2)\log(1 + r_{\mathbb{B}}^2)$	$\dot{r_{\mathbb{S}}} = \sqrt{\mu}\dot{f}(\theta) + \gamma\dot{f}(\theta)\tanh\left(\mu - \left(\frac{r_{\mathbb{S}}}{f(\theta)}\right)^2\right)\log\left(1 + \left(\frac{r_{\mathbb{S}}}{f(\theta)}\right)^2\right)$
(IV) Morphed oscillator $\dot{r}_{\mathbb{B}} = \omega\cos(\theta) + 2 + \sin(\theta) - r_{\mathbb{B}}$	$\dot{r_{\mathbb{S}}} = (2 + sin(\theta))\,\dot{f}(\theta) + f(\theta)\left(\omega\cos(\theta) + 2 + \sin(\theta) - \frac{r_{\mathbb{S}}}{f(\theta)}\right)$

Table 3: Example second order realizations with different base oscillators.

Base oscillator	**Compensated second order realization**
(I) Amplitude controlled oscillator $\dot{r}_{\mathbb{B}} = \gamma\,(\mu - r_{\mathbb{B}})$	$\dot{v_{\mathbb{S}}} = \mu\ddot{f}(\theta) + \gamma\left(\mu\dot{f}(\theta) + \mu\delta - v_{\mathbb{S}}\right) + \beta(\mu f(\theta) - r_{\mathbb{S}})$ $\dot{r_{\mathbb{S}}} = v_{\mathbb{S}} - \mu\delta$
(II) Hopf oscillator $\dot{r}_{\mathbb{B}} = \gamma(\mu - r_{\mathbb{B}}^2)r_{\mathbb{B}}$	$\dot{v_{\mathbb{S}}} = \sqrt{\mu}\ddot{f}(\theta) + \gamma\left(\mu - \left(\frac{v_{\mathbb{S}}}{\dot{f}(\theta)+\delta}\right)^2\right)v_{\mathbb{S}} + \beta(\sqrt{\mu}f(\theta) - r_{\mathbb{S}})$ $\dot{r_{\mathbb{S}}} = v_{\mathbb{S}} - \sqrt{\mu}\delta$
(III) Logarithmic saturated oscillator $\dot{r}_{\mathbb{B}} = \gamma\tanh(\mu - r_{\mathbb{B}}^2)\log(1 + r_{\mathbb{B}}^2)$	$\dot{v_{\mathbb{S}}} = \sqrt{\mu}\ddot{f}(\theta) + \gamma(\dot{f}(\theta) + \delta)\tanh\left(\mu - \left(\frac{v_{\mathbb{S}}}{\dot{f}(\theta)+\delta}\right)^2\right)\log\left(1 + \left(\frac{v_{\mathbb{S}}}{\dot{f}(\theta)+\delta}\right)^2\right) + \beta(\sqrt{\mu}f(\theta) - r_{\mathbb{S}})$ $\dot{r_{\mathbb{S}}} = v_{\mathbb{S}} - \sqrt{\mu}\delta$
(IV) Morphed oscillator $\dot{r}_{\mathbb{B}} = \omega\cos(\theta) + 2 + \sin(\theta) - r_{\mathbb{B}}$	$f_v(\theta) = \frac{\omega\cos(\theta)}{2+\sin(\theta)}f(\theta) + \dot{f}(\theta)$ $\dot{v_{\mathbb{S}}} = (2 + sin(\theta))\,\dot{f}_v(\theta) + (f_v(\theta) + \delta)\left(\omega\cos(\theta) + 2 + \sin(\theta) - \frac{v_{\mathbb{S}}}{f_v(\theta)+\delta}\right) + \beta\left((2 + \sin(\theta))f(\theta) - r_{\mathbb{S}}\right)$ $\dot{r_{\mathbb{S}}} = v_{\mathbb{S}} - (2 + \sin(\theta))\delta$

3.1. Equivalence

Here we will show how certain realizations of the morphed oscillators represent familiar trajectory control/generation methods. The first example in Table 2 uses an amplitude controlled oscillator as base. Assuming $\mu = 1$, the first example implements a position PD-controller with a desired periodic reference defined as $\Xi_{\mathbb{S}}(\theta) = \Xi_{\mathbb{B}}(\theta)f(\theta) = f(\theta)$. The normalized proportional gain for this controller is γ. So with the right representation, position PD-control over a periodic reference is a subset of the general family generated by the compensated form.

The first example in Table 3 implements a second order dynamical system which represents a form of the rhythmic Dynamical Movement Primitives [26]. Dynamical Movement Primitives (DMP) are robust movement generators that are commonly used for motor control. The rhythmic DMP is formulated as:

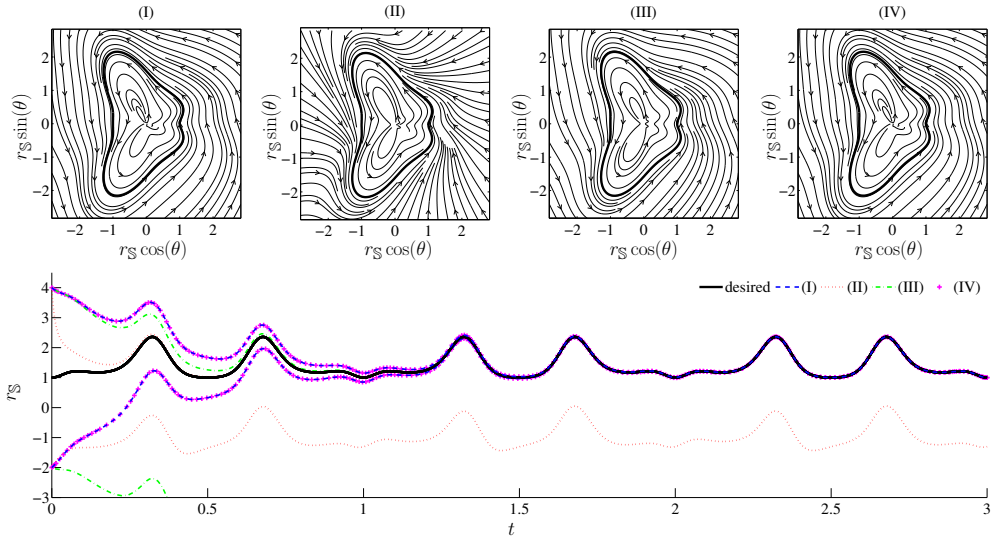

Figure 5: First order realization examples with four different base oscillators: (I) amplitude controlled oscillator, (II) Hopf oscillator, (III) logarithmic saturated oscillator, (IV) morphed oscillator. The shape of limit cycle is the same for all examples, and the difference is in the convergence behavior. The initial state value in the bottom plot, $r_{\mathbb{S}} = 4$, is corresponding to $r_{\mathbb{S}}\cos(\theta) = 4, r_{\mathbb{S}}\sin(\theta) = 0$ in the top plots ($\theta|_{t=0} = 0$). The basin of attraction of the desired limit cycle for (I) and (IV) is the whole state space, while this is not true for (II) and (III). The basin of attraction for these oscillators are later defined in Table 4. The behavior of examples (I) and (IV) are the same. This is because example (IV) uses a morphed oscillator as base which has been obtained by morphing an amplitude controlled oscillator, the base of (I). So (IV) is identical to having an amplitude controlled oscillator as base, and then morphing it twice using two phase-based scaling functions.

$$\tau\dot{z} = \alpha_z(\beta_z(g - y) - z) + F, \quad \tau\dot{y} = z \tag{13}$$

$$\tau\dot{\theta} = 1 \tag{14}$$

$$F = \tau^2\ddot{y}_{des}(t) + \tau\alpha_z\dot{y}_{des}(t) + \alpha_z\beta_z y_{des}(t) - \alpha_z\beta_z g$$
$$= \tau^2(\frac{1}{\tau})^2 y''_{des}(\theta) + \tau(\frac{1}{\tau})\alpha_z y'_{des}(\theta) + \alpha_z\beta_z y_{des}(\theta) - \alpha_z\beta_z g \tag{15}$$

where τ is the cycle period divided by 2π, and F is the nonlinear forcing term to shape the limit cycle such that the output y_{des} is obtained. Rewriting the transformation system (Equation (13)) with expanded F and simplifying gives:

$$\dot{z} = \tau\ddot{y}_{des}(\theta) + \frac{\alpha_z}{\tau}(\tau\dot{y}_{des}(\theta) - z) + \frac{\alpha_z\beta_z}{\tau}(y_{des}(\theta) - y), \quad \dot{y} = \frac{1}{\tau}z \tag{16}$$

and by assuming $r_{\mathbb{S}} = y$, $v_{\mathbb{S}} = \frac{1}{\tau}z$ and $f(\theta) = y_{demo}(\theta)$, the above is identical to a second-order realization with a unit radius amplitude controlled oscillator as base ($\mu = 1$), $\gamma = \frac{\alpha_z}{\tau}$ and $\beta = \frac{\alpha_z\beta_z}{\tau^2}$. So with the right representation, rhythmic DMPs (with the latest representation in [26]) are a subset of the general family generated by the compensated form.

4. Extension to higher dimensions

The methodology presented in Section 2 gives a systematic way to create nonlinear oscillators with one dimensional outputs $r_{\mathbb{S}}$. In this section we explain how multidimensional oscillators can be created out of these one dimensional oscillators. Extending to high dimensions expands the scope which the introduced oscillators can be applied, including coupled synchronized high dimensional movements needed in applications like robotics [46, 22], or in modeling of Central Pattern Generators [3, 4] .

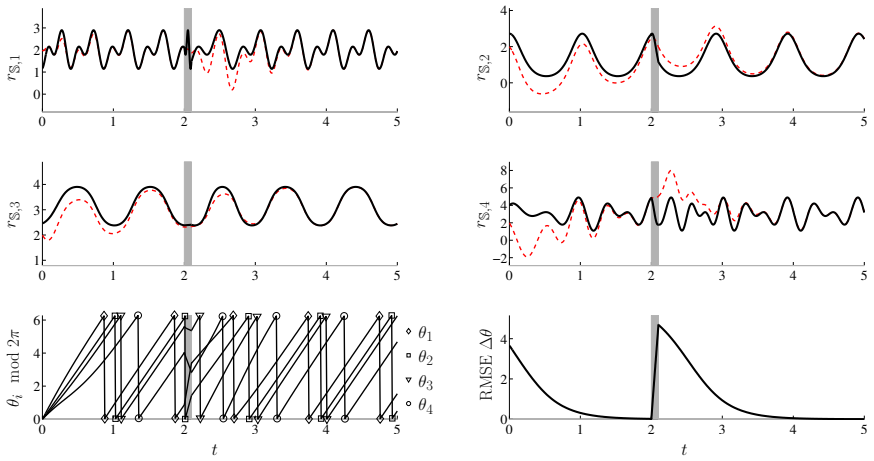

Figure 6: A coupled four dimensional system of second order morphed oscillators. The desired phase differences, with respect to θ_1, are $\{0, \pi/3, \pi/2, \pi\}$. The base oscillators are respectively the ones in Table 3. The desired limit cycles are $\Xi_{\mathbb{S},1}(\theta) = 2 + \cos(3\theta) + sin(\theta - 1)$, $\Xi_{\mathbb{S},2}(\theta) = e^{\cos(\theta)}$, $\Xi_{\mathbb{S},3}(\theta) = \pi + \tanh(\sin(\theta - 1))$, and $\Xi_{\mathbb{S},4}(\theta) = \sin(2\theta) + cos(3\theta) + 3$. The top four plots depict the evolution of the output state $r_{\mathbb{S},i}$ over time, for each dimension. The black references are $\Xi_{\mathbb{S},i}(\theta)$ and the dashed red trajectories are the outputs of each dimension $r_{\mathbb{S},i}$. The bottom-left plot shows the evolution of the phases of the oscillators. All phases are initialized as 0, and they get coupled around $t = 1.5$. The phases are perturbed for $t \in [2, 2.1]$. The bottom-right figure shows the root mean square error between the vector of phases and the vector of desired phase differences. One can see that phases get coupled again after the perturbation at about $t = 4$. For this example all $\gamma_i = 10$, $\beta_i = 20$, $\delta_i = 10$, $\vartheta = 1$, and $c_{ik} = 1$.

Several approaches including [37, 8, 13] use coupled oscillators to create multidimensional nonlinear oscillators. One can use diffusive phase coupling [47] to create the multidimensional system, so Equation (5) changes to:

$$\dot{\theta}_i = \omega + \sum_{k=1}^{N} c_{ik} \sin(\theta_k - \theta_i - \phi_{ik}) \tag{17}$$

where θ_i is the phase state of the $i-$th oscillator, c_{ik} is the coupling strength between $i-$th and $k-$th oscillators, and ϕ_{ik} is the desired phase difference between them. The diffusive schema in Equation (17) couples the phase dynamics, and this coupling is independent of the output states $r_{\mathbb{S},i}$. Implementing coupling schema which are affected by the $r_{\mathbb{S},i}$ states is possible, and is application specific. An example can be found in [8].

Figure 6 illustrates an all-to-all coupled four dimensional system where each oscillator is a second-order morphed oscillator. A different limit cycle shape is chosen for each dimension, and these are depicted in the top four plots. The bottom-left plot shows the phase-time evolution, and phase differences can be seen with the overlaid markers. All phases are initialized with 0 value, and they are also perturbed for $t \in [2, 2.1]$. The system gradually gets synchronized after the perturbation, as illustrated in the bottom-right plot in Figure 6.

5. Stability

This section is dedicated to stability analysis of the introduced morphed oscillators. We first discuss the stability of a one dimensional first order system using the Poincaré-Bendixson theorem [48]. Analysis of $n-$th order systems, $n > 1$, is complex as the Poincaré-Bendixson theorem is not valid anymore, and we briefly discuss these systems using Contraction Theory [49]. Finally we include the stability of the multidimensional system.

Before going into the stability analysis, we would like to describe the space formed by $\{\theta, r_{\mathbb{S}}\}$. One can simply assume that $\theta \in [0, 2\pi)$, $r_{\mathbb{S}} \in \mathbb{R}^+$, and by doing so $\{\theta, r_{\mathbb{S}}\}$ forms the standard polar coordinates. However we additionally want to be able to analyze the system when $r_{\mathbb{S}} < 0$ (without jumping to the antiphase state $\{\theta+\pi, |r_{\mathbb{S}}|\}$). One can assume that $\theta \in [0, 2\pi)$, $r_{\mathbb{S}} \in \mathbb{R}^+$ forms a manifold, and $\theta \in [0, 2\pi)$, $r_{\mathbb{S}} \in \mathbb{R}^-$ forms a second manifold, and these two manifolds are connected when $r_{\mathbb{S}} = 0$. The resulting manifold can be chosen to be a 2-manifold, as illustrated in Figure 7, to describe the space formed by $\{\theta, r_{\mathbb{S}}\} \in [0, 2\pi) \times \mathbb{R}$. We call this representation the *extended polar coordinates* where negative radius values are meaningful. The chosen 2-manifold in Figure 7 is only an arbitrary representation, and any other representation which defines an orientable 2-manifold is valid.

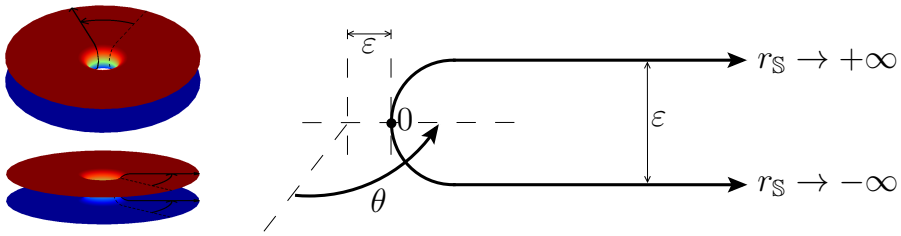

Figure 7: A representative 2-manifold formed by $\theta \in [0, 2\pi)$, $r_{\mathbb{S}} \in \mathbb{R}$. When the value of $r_{\mathbb{S}}$ crosses zero the system state goes from one disk to another. ε is a positive constant. The pulley-like 2-manifolds on the left are obtained by rotational sweep of the coordinates shown in right.

5.1. One dimensional first order system

Analyzing the limit cycle properties of a dynamical system is not an easy task. The analysis becomes possible when one is looking for the existence of limit cycles in a bounded region of a phase plane, where the Poincaré-Bendixson's theorem can be utilized. The same theorem can also be used the analyze the asymptotic stability properties of a limit cycle system. The original form of this theorem is:

Theorem (Poincaré-Bendixson [48])**.** *Every nonempty, compact ω-limit set of a C^1 planar flow that does not contain an equilibrium point is a (nondegenerate) periodic orbit.*

A simpler interpretation of this theorem is: given a differential equation in the plane, assume $\zeta(t)$ is a solution curve which stays in a bounded region. Then, if there is no equilibrium point in this region, $\zeta(t)$ converges for $t \to +\infty$ to a periodic trajectory. Now if there is only one periodic orbit in this region, the asymptotic stability of the corresponding limit cycle in this bounded region is assured. Poincaré-Bendixson's theorem also holds for every orientable 2-manifold for which the Jordan curve theorem [50] holds. The extended polar coordinates defined by $\{\theta, r_{\mathbb{S}}\}$ is an orthogonal coordinate system, and the manifold created by $\{\theta, r_{\mathbb{S}}\} \in \mathbb{R}^2$ is an orientable 2-manifold satisfying Jordan curve theorem. To employ Poincaré-Bendixson's theorem on our problem, we need to define a bounded region around the desired limit cycle $\Xi_{\mathbb{S}}(\theta)$. We define the **L**ower and **U**pper bounds around $\Xi_{\mathbb{S}}(\theta)$ as:

$$\begin{aligned} g_L &: \theta \longmapsto g_L(\theta) \; ; \; g_L(\theta) < \Xi_{\mathbb{S}}(\theta) \\ g_U &: \theta \longmapsto g_U(\theta) \; ; \; g_U(\theta) > \Xi_{\mathbb{S}}(\theta) \end{aligned} \tag{18}$$

For the upper bound, vectors pointing inwards the bounded region are defined as $\mathbf{p} = \{\dot{g}_U(\theta), -\dot{\theta}\}$, and for the lower bound they are $\mathbf{p} = \{-\dot{g}_L(\theta), \dot{\theta}\}$ (assuming clockwise phase evolution, i.e. $\dot{\theta} > 0$). The system dynamics on these bounds are also defined as:

$$\begin{aligned} &\text{original form, Equation (7)} : \mathbf{d} = \{\dot{\theta}, \frac{g_\iota(\theta)}{f(\theta)}\dot{f}(\theta) + f(\theta)D_{\mathbb{B},r}(\frac{g_\iota(\theta)}{f(\theta)})\}, \iota \in \{L, U\} \\ &\text{compensated form, Equation (8)} : \mathbf{d} = \{\dot{\theta}, \Xi_{\mathbb{B}}(\theta)\dot{f}(\theta) + f(\theta)D_{\mathbb{B},r}(\frac{g_\iota(\theta)}{f(\theta)})\}, \iota \in \{L, U\} \end{aligned} \tag{19}$$

To have the condition that no flow leaves the bounds we need (with $\langle ., . \rangle$ operator being the inner product):

$$\langle \mathbf{p}, \mathbf{d} \rangle > 0 \tag{20}$$

To utilize the Poincaré-Bendixson's theorem, it is needed to define bounds such that no flows leaves the enclosed area. Figure 8 gives an idea about how to define the bounds. For the original form we define:

$$g_{L|U}(\theta) = \Xi_{\mathbb{S}}(\theta) + \kappa f(\theta) = (\Xi_{\mathbb{B}}(\theta) + \kappa) f(\theta); \kappa \lesseqgtr 0 \tag{21}$$

Rewriting Equation (20) and dividing by the positive term $\dot{\theta} f(\theta)$ gives (by definition $f(\theta) > 0$ and we have already assumed $\dot{\theta} > 0$):

$$\text{lower bound:} \quad D_{\mathbb{B},r}(\Xi_{\mathbb{B}}(\theta) + \kappa) > \dot{\Xi}_{\mathbb{B}}(\theta); \kappa < 0 \tag{22}$$

$$\text{upper bound:} \quad D_{\mathbb{B},r}(\Xi_{\mathbb{B}}(\theta) + \kappa) < \dot{\Xi}_{\mathbb{B}}(\theta); \kappa > 0 \tag{23}$$

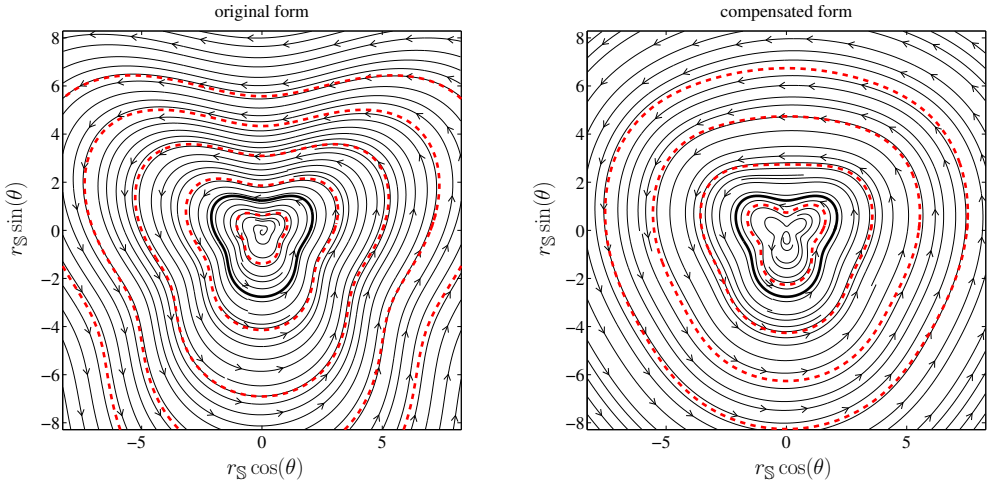

Figure 8: Poincaré-Bendixson bounds (dashed red) for the first order original and compensated forms. The bounds of the original form are scaled versions of the limit cycle, while the bounds of the compensated form are similar to the limit cycle shape when near it, and become circular when far. For this example $\Xi_{\mathbb{S}}(\theta) = 2 + \sin(\theta)\tanh(\cos(2\theta))$, and the base is an amplitude controlled oscillator $\vartheta = 1$, $\mu = 1$ and $\gamma = 0.2$. The trajectories of the system dynamics (solid gray) enter the bounds (dashed red) and do not leave them.

Now if the conditions in Equations (22-23) are met for all possible margins in a bounded region, then the desired limit cycle is asymptotically stable. So the basin of attraction for the original form is defined by the bounds:

$$\begin{aligned} \kappa_L \in \mathbb{R} \mid \forall\theta \in \mathbb{R}\ \&\ \forall\kappa,\ \kappa_L < \kappa < 0 : D_{\mathbb{B},r}(\Xi_{\mathbb{B}}(\theta) + \kappa) > \dot{\Xi}_{\mathbb{B}}(\theta) \\ \kappa_U \in \mathbb{R} \mid \forall\theta \in \mathbb{R}\ \&\ \forall\kappa,\ 0 < \kappa < \kappa_U : D_{\mathbb{B},r}(\Xi_{\mathbb{B}}(\theta) + \kappa) < \dot{\Xi}_{\mathbb{B}}(\theta) \end{aligned} \tag{24}$$

The lower and upper bounds for the compensated form in Equation (8) are not same as the ones of the original form. This is depicted in Figure 8. What happens with the compensated form is that the dynamics contours are like the shape of limit cycle for states near it, but become circular for states far from limit cycle. So we define:

$$g_{L|U}(\theta) = \Xi_{\mathbb{S}}(\theta) + \kappa = \Xi_{\mathbb{B}}(\theta)f(\theta) + \kappa; \kappa \lessgtr 0 \tag{25}$$

and by using the same procedure used for the original form we get:

$$D_{\mathbb{B},r}(\Xi_{\mathbb{B}}(\theta) + \frac{\kappa}{f(\theta)}) \gtrless \dot{\Xi}_{\mathbb{B}}(\theta); \kappa \lessgtr 0 \tag{26}$$

The term $\frac{\kappa}{f(\theta)}$ is strictly positive when $\kappa > 0$ and strictly negative if $\kappa < 0$. So again the basin of attraction is defined by the bounds:

$$\begin{aligned} \kappa_L \in \mathbb{R} \mid \forall\theta \in \mathbb{R}\ \&\ \forall\kappa,\ \kappa_L < \kappa < 0 : D_{\mathbb{B},r}(\Xi_{\mathbb{B}}(\theta) + \frac{\kappa}{f(\theta)}) > \dot{\Xi}_{\mathbb{B}}(\theta) \\ \kappa_U \in \mathbb{R} \mid \forall\theta \in \mathbb{R}\ \&\ \forall\kappa,\ 0 < \kappa < \kappa_U : D_{\mathbb{B},r}(\Xi_{\mathbb{B}}(\theta) + \frac{\kappa}{f(\theta)}) < \dot{\Xi}_{\mathbb{B}}(\theta) \end{aligned} \tag{27}$$

Table 4 gives examples of the basin of attraction for different realizations. The first four examples are the same morphed oscillators as in Table 2. The fifth example is with a base oscillator that has one stable limit cycle at $r_{\mathbb{B}} = \mu$ and two unstable ones at $r_{\mathbb{B}} = 0$ and $r_{\mathbb{B}} = 2\mu$. For the first four examples, the upper bound of the basin of attraction tends to infinity. The basin of attraction for the fifth example is bounded by the two other unstable limit cycles. In general, if the base has multiple limit cycles, the designer should choose which limit cycle is used for morphing, and the other limit cycles will then limit the span of the basin of attraction.

Table 4: The basin of attraction of first order realizations using different bases.

base oscillator	basin of attraction bounds parameters	
	original	compensated
$\dot{\theta} = \omega$ $\dot{r}_{\mathbb{B}} = \gamma(\mu - r_{\mathbb{B}})$	LB: $\kappa \to -\infty$ UB: $\kappa \to +\infty$	
$\dot{\theta} = \omega$ $\dot{r}_{\mathbb{B}} = \gamma(\mu - r_{\mathbb{B}}^2) r_{\mathbb{B}}$	LB: $\kappa = -\sqrt{\mu}$ UB: $\kappa \to +\infty$	LB: $\kappa = -\sqrt{\mu} \min_{\theta}(f(\theta))$
$\dot{\theta} = \omega$ $\dot{r}_{\mathbb{B}} = \gamma \tanh(\mu - r_{\mathbb{B}}^2) \log(1 + r_{\mathbb{B}}^2)$	LB: $\kappa = -2\sqrt{\mu}$ UB: $\kappa \to +\infty$	LB: $\kappa = -2\sqrt{\mu} \min_{\theta}(f(\theta))$
$\dot{\theta} = \omega$ $\dot{r}_{\mathbb{B}} = \omega \cos(\theta) + 2 + \sin(\theta) - r_{\mathbb{B}}$	LB: $\kappa \to -\infty$ UB: $\kappa \to +\infty$	
$\dot{\theta} = \omega$ $\dot{r}_{\mathbb{B}} = -\gamma\, r_{\mathbb{B}}(\mu - r_{\mathbb{B}})(2\mu - r_{\mathbb{B}})$	LB: $\kappa = -\mu$ UB: $\kappa = +\mu$	LB: $\kappa = -\mu \min_{\theta}(f(\theta))$ UB: $\kappa = \mu \min_{\theta}(f(\theta))$

5.2. One dimensional second order system

Analyzing the asymptotic stability of a general nonlinear second order system is not trivial, and the Poincaré-Bendixson theorem cannot be used anymore as it is only valid for phase planes, and not volumes. We utilize Contraction Theory [49] to show the stability conditions of our 2nd+ order systems. Contraction Theory characterizes the system stability by the behavior of the differences between solutions with different initial conditions. If these differences vanish exponentially over time, all solutions converge towards a single trajectory, independent from the initial states. In this case, the system is called globally asymptotically stable. For a general dynamical system of the form $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t)$ assume that $\mathbf{x}(t)$ is one solution of the system, and $\tilde{\mathbf{x}}(t) = \mathbf{x}(t) + \delta\mathbf{x}(t)$ a neighboring one with a different initial condition ($\delta\mathbf{x}(t)$ is also called virtual displacement). It can be shown that any nonzero virtual displacement decays exponentially to zero over time if the symmetric part of the Jacobian of $\mathbf{f}$ is uniformly negative definite. In this case, it can be shown that the norm of the virtual displacement decays at least exponentially to zero, for $t \to \infty$ [49]. Namely, $\|\delta\mathbf{x}\| \leq \exp \int_0^t \lambda_{max}(\mathbf{x}, t)dt \|\delta\mathbf{x}_0\|$, where $\lambda_{max}(\mathbf{x}, t)$ is the largest eigenvalue of $\frac{1}{2}(\frac{\partial \mathbf{f}}{\partial \mathbf{x}} + \frac{\partial \mathbf{f}}{\partial \mathbf{x}}^T)$, so that the sufficient contraction condition is: $\lambda_{max}(\mathbf{x}, t) \leq -b < 0$ (for some $b > 0$) uniformly in a contracting region.

One can write the infinitesimal virtual displacements of the second order compensated form as (since θ is indifferent and $f(\theta)$ is bounded, the phase-dependent terms can be considered as inputs over time):

$$\frac{d}{dt}\begin{bmatrix}\delta v_{\mathbb{S}} \\ \delta r_{\mathbb{S}}\end{bmatrix} = \mathbf{J}\begin{bmatrix}\delta v_{\mathbb{S}} \\ \delta r_{\mathbb{S}}\end{bmatrix} = \begin{bmatrix}D'_{\mathbb{B},r}(\frac{v_{\mathbb{S}}}{f_v(\theta)+\delta}) & -\beta \\ 1 & 0\end{bmatrix}\begin{bmatrix}\delta v_{\mathbb{S}} \\ \delta r_{\mathbb{S}}\end{bmatrix} \tag{28}$$

where $\mathbf{J}$ is the unsymmetrized Jacobian of Equations (10,11). This has a symmetrized Jacobian:

$$\mathbf{J}_{sym} = \begin{bmatrix}D'_{\mathbb{B},r}(\frac{v_{\mathbb{S}}}{f_v(\theta)+\delta}) & (1-\beta)/2 \\ (1-\beta)/2 & 0\end{bmatrix} \tag{29}$$

Since here the symmetrized Jacobian has its eigenvalues of the opposite signs, we apply the linear local coordinate transform for variational displacements, obtaining the new coordinates: $\delta v_{\mathbb{S}}$ and $\delta z_{\mathbb{S}} = \sqrt{\beta}\, \delta r_{\mathbb{S}}$ (for $\beta > 0$). Now the symmetrized Jacobian is

$$\mathbf{J}_{sym} = \begin{bmatrix}D'_{\mathbb{B},r}(\frac{v_{\mathbb{S}}}{f_v(\theta)+\delta}) & 0 \\ 0 & 0\end{bmatrix} \tag{30}$$

which has a zero eigenvalue and the other one is $D'_{\mathbb{B},r}(\frac{v_{\mathbb{S}}}{f_v(\theta)+\delta})$. So the second order system is contracting when $D'_{\mathbb{B},r}(.) < 0, \beta > 0$. With the above conditions, the compensated second order form is partially contracting towards the desired limit cycle and the phase subsystem is indifferent.

5.3. One dimensional n−th order system

The stability analysis of the $n-$th order system ($n > 2$) is similar to the one for a 2nd order system. Again writing the infinitesimal virtual displacements give:

$$\frac{d}{dt}\begin{bmatrix}\delta r_{\mathbb{S}}^{(n-1)}\\ \delta r_{\mathbb{S}}^{(n-2)}\\ \vdots\\ \delta r_{\mathbb{S}}^{(1)}\\ \delta r_{\mathbb{S}}\end{bmatrix} = \begin{bmatrix} D'_{\mathbb{B},r}\left(\frac{r_{\mathbb{S}}^{(n-1)}}{f_{(n-1)}+\delta_{n-1}}\right) & -\beta_{n-1} & 0 & \dots & 0 & 0 & 0\\ 1 & 0 & -\beta_{n-2} & \dots & 0 & 0 & 0\\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots\\ 0 & 0 & 0 & \dots & 1 & 0 & -\beta_1\\ 0 & 0 & 0 & \dots & 0 & 1 & 0 \end{bmatrix}\begin{bmatrix}\delta r_{\mathbb{S}}^{(n-1)}\\ \delta r_{\mathbb{S}}^{(n-2)}\\ \vdots\\ \delta r_{\mathbb{S}}^{(1)}\\ \delta r_{\mathbb{S}}\end{bmatrix} \tag{31}$$

and again using the new coordinates: $\delta r_{\mathbb{S}}^{(n-1)}$, $\delta z_{\mathbb{S}}^{(n-2)} = \sqrt{\beta_{n-1}}\,\delta r_{\mathbb{S}}^{(n-2)}$, $\delta z_{\mathbb{S}}^{(n-3)} = \sqrt{\beta_{n-2}}\,\delta r_{\mathbb{S}}^{(n-3)}$, $\dots$, $\delta z_{\mathbb{S}} = \sqrt{\beta_1}\,\delta r_{\mathbb{S}}$, the $n-$th order system is partially contracting when $D'_{\mathbb{B},r}\left(\frac{r_{\mathbb{S}}^{(n-1)}}{f_{(n-1)}+\delta_{n-1}}\right) < 0$ and $\forall i : \beta_i > 0$.

5.4. Multidimensional system

The stability of the multidimensional system created in Equation (17) should be analyzed from two aspects: (I) asymptotic stability of the phase coupling, (II) asymptotic stability of each dimension under the coupling. Since phases are independent of the radial values, the multidimensional system is a hierarchy, and the asymptotic stability of phase coupling can be analyzed independently [49]. To address (I), it can be shown that as long as for every loop in the coupling graph passing through oscillators $i_1, i_2, i_3, \dots, i_m, i_1$:

$$\phi_{i_1 i_2} + \phi_{i_2 i_3} + \dots + \phi_{i_m i_1} = 2k\pi, k \in \mathbb{Z} \tag{32}$$

and with the additional condition $\forall i, j : c_{ij} \geq 0$, the phase differences $\theta_i - \theta_j$ will asymptotically converge to the desired phase differences ϕ_{ij}. One can introduce the potential function (for the coupling dynamics in Equation (17), with the change of variables $\theta_i \leftarrow \theta_i - \omega t$):

$$U(\boldsymbol{\theta}) = -\sum_{i=1}^{N}\sum_{j=1}^{N} c_{ij}\cos(\theta_j - \theta_i - \phi_{ij}) \tag{33}$$

which gives: $\dot{\theta}_i = -\frac{\partial U}{\partial \theta_i}$ and the potential $U(\boldsymbol{\theta})$ has minima at $\forall i, j : \theta_i = \theta_j - \phi_{ij} + 2k\pi, \forall k \in \mathbb{Z}$. Since $\frac{dU}{dt} = \sum_{j=1}^{N}\frac{\partial U}{\partial \theta_j}\frac{d\theta_j}{dt} = -\sum_{j=1}^{N}(\frac{\partial U}{\partial \theta_j})^2$, then $U(\boldsymbol{\theta})$ plays a role of Lyapunov's function, proving the asymptotic stability. Now since the phase differences are consistent (Equation (32)), the system remains synchronized.

The multidimensional system is a hierarchically coupled system, where the radial dynamics depends on the phase dynamics but the phase dynamic does not depend on the radial dynamics, and the phase dynamics is contracting by itself. As long as the phases are synchronized, $\forall i : \dot{\theta}_i = \omega = 2\pi\vartheta$, then all oscillators will converge to their limit cycles. An asynchrony, e.g. $\theta_j - \theta_i \neq 2k\pi + \phi_{ij}$, can introduce a perturbation on the radial dynamics for the $i-$th dimension. Let us assume that this perturbation, at phase θ_i, is quantified as $u_i(\theta_i)$. As long as this perturbation does not push the $i - th$ oscillator out of its basin of attraction, it will eventually be forgotten, and the $i-$th oscillator will converge to its limit cycle. So in the case where the basin of attraction is the whole state space, the whole multidimensional system is asymptotically stable. For the case where the basin of attraction is limited, if the sum of the coupling weights is small enough, $u_i(\theta_i)$ will be small enough, and the system will remain in the basin of attraction. One can simply choose to have all the coupling weights equal and small enough to ensure stability.

6. Arbitrary convergence behavior

The previous sections introduced a methodology to design morphed phase oscillators with arbitrary limit cycle shapes, and analyzed their stability. The convergence behavior of these morphed oscillators is determined by the choice of the base oscillator, and cannot be explicitly defined. However it is useful to have an oscillator which can exhibit a given desired convergence behavior. Examples can be when a certain path should be followed to reach the limit cycle [51], or if phase-dependent convergence behavior is of interest.

To obtain such oscillator, we modify the case where a unit radius amplitude controlled oscillator is used as the base oscillator ($\Xi_{\mathbb{B}}(\theta) = 1$) and an original first order realization is applied. We first need to find the analytical solution for the case where an amplitude controlled oscillator is used as the base. For the phase equation we can simply write:

$$\theta(t) = \omega t + C_\theta,\ C_\theta = \theta(0) \tag{34}$$

To find the analytical solution for the radius equation, we define $g = f(\theta)$, and we rewrite the morphed amplitude controlled oscillator (first example in Table 2) as (for simplicity, and without loss of generality, we assume $\mu = 1$):

$$\dot{r}_{\mathbb{S}} + \left(\gamma - \frac{\dot{g}}{g}\right) r_{\mathbb{S}} = \gamma g \tag{35}$$

This is a first order differential equation of the form:

$$\dot{r}_{\mathbb{S}} + p r_{\mathbb{S}} = q \tag{36}$$

with $p = \gamma - \frac{\dot{g}}{g}$ and $q = \gamma g$. This form has the general solution: $r_{\mathbb{S}} e^{\int p dt} = \int q e^{\int p dt} dt + C_r$. Solving the integration with respective p and q expressions gives (with $g = f(\theta)$):

$$r_{\mathbb{S}}(t) = f(\theta(t)) + C_r f(\theta(t)) e^{-\gamma t},\ C_r = \frac{r_{\mathbb{S}}(0)}{f(\theta(0))} - 1 \tag{37}$$

The analytical solution shows that the oscillator converges to the desired limit cycle behavior $\Xi_{\mathbb{S}}(\theta) = f(\theta)$ when $t \to \infty$. The convergence behavior is an exponential decay which is shaped by f. Now if we use a custom convergence function h instead of f to define the convergence behavior, we can modify Equation (37) to represent the new desired solution:

$$r_{\mathbb{S}}(t) = f(\theta(t)) + C_r h(t) e^{-\gamma t},\ C_r = \frac{r_{\mathbb{S}}(0) - f(\theta(0))}{h(0)} \tag{38}$$

To obtain the dynamical system yielding this desired solution we perform the steps done to obtain Equation (37) *backwards*. If we multiply both sides of Equation (38) with $\frac{e^{\gamma t}}{h}$ we have (again $g = f(\theta)$): $r_{\mathbb{S}} \frac{e^{\gamma t}}{h} = e^{\gamma t} \frac{g}{h} + C_r$, which can be rewritten as:

$$r_{\mathbb{S}} \frac{e^{\gamma t}}{h} = \int \frac{e^{\gamma t}}{h} \left(\gamma g + \frac{\dot{g}h - \dot{h}g}{h}\right) dt + C_r \tag{39}$$

Now if we define $e^p = \frac{e^{\gamma t}}{h}$ we obtain the general coefficients $p = \gamma - \frac{\dot{h}}{h}$ and $q = \gamma g + \frac{\dot{g}h - \dot{h}g}{h}$. With p and q defined, Equation (39) is a solution of a first order differential equation of the form defined in Equation (36). We obtain:

$$\dot{r}_{\mathbb{S}} + \left(\gamma - \frac{\dot{h}}{h}\right) r_{\mathbb{S}} = \gamma g + \frac{\dot{g}h - \dot{h}g}{h} \tag{40}$$

and we can simplify and write it as an ordinary differential equation for radius (with $g = f(\theta)$):

$$\boxed{\dot{r}_{\mathbb{S}} = \dot{f}(\theta) + \left(\gamma - \frac{\dot{h}(\theta,t)}{h(\theta,t)}\right)(f(\theta) - r_{\mathbb{S}})} \tag{41}$$

Equation (41) along with the phase Equation (5) gives a first order morphed phase oscillator which the shape of its limit cycle is defined by f and the shape of its convergence by h. As the desired solution in Equation (38) shows, the steady state solution does not depend on h and is only defined by f. The function $h(\theta, t)$ can be a function of phase, time, or both, and depends on the application. One can argue that the term $\gamma - \frac{\dot{h}(\theta,t)}{h(\theta,t)}$ can be generally written as $\gamma(\theta, t)$ which means having a phase - and/or time - dependent convergence rate. This is correct, however a representation like $\gamma(\theta, t)$ will not explain what the convergence behavior will be.

To ensure a stable system, the fixed-point $f(\theta) - r_{\mathbb{S}}$ in Equation (41) should be attractive. This means the condition $\gamma - \frac{\dot{h}(\theta,t)}{h(\theta,t)} \geq 0$ should be satisfied. Moreover, since $h(\theta, t)$ is in the denominator in Equation (41), $h(\theta, t)$ should not have a zero-crossing.

Figure 9 shows examples of the arbitrary convergence behavior. The top plot in Figure 9 depicts the effect of perturbation on different h functions. The left plot in Figure 9 shows the phase portrait of a system with four different convergence behaviors. As it is clear, the limit cycle shape is not affected by the choice of the convergence behavior.

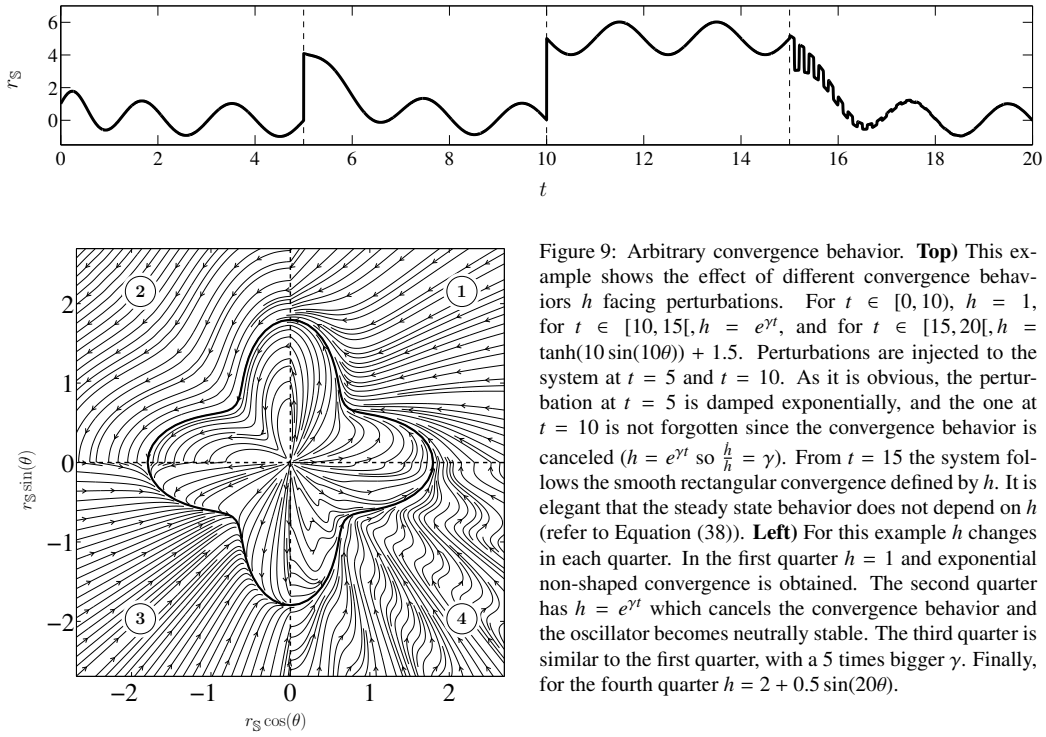

Figure 9: Arbitrary convergence behavior. **Top)** This example shows the effect of different convergence behaviors h facing perturbations. For $t \in [0, 10)$, $h = 1$, for $t \in [10, 15[, h = e^{\gamma t}$, and for $t \in [15, 20[, h = \tanh(10\sin(10\theta)) + 1.5$. Perturbations are injected to the system at $t = 5$ and $t = 10$. As it is obvious, the perturbation at $t = 5$ is damped exponentially, and the one at $t = 10$ is not forgotten since the convergence behavior is canceled ($h = e^{\gamma t}$ so $\frac{\dot{h}}{h} = \gamma$). From $t = 15$ the system follows the smooth rectangular convergence defined by h. It is elegant that the steady state behavior does not depend on h (refer to Equation (38)). **Left)** For this example h changes in each quarter. In the first quarter $h = 1$ and exponential non-shaped convergence is obtained. The second quarter has $h = e^{\gamma t}$ which cancels the convergence behavior and the oscillator becomes neutrally stable. The third quarter is similar to the first quarter, with a 5 times bigger γ. Finally, for the fourth quarter $h = 2 + 0.5\sin(20\theta)$.

7. Learning

This section will explain how the scaling function f can be created from data points. We will additionally explain how the convergence behavior h can be fitted from data when Equation (41) is of interest. All the given descriptions are for one-dimensional cases, and extension to multi-dimensional cases is done by just repeating the same process for all dimensions and setting correct phase differences ϕ_{ij}.

7.1. Learning the scaling function

Let us assume that one dimensional input data is given as $\{t_i, y_i\}, i = 1 \dots N$, where t_i is the sample time, y_i is the desired output, and N is the number of data points. This data vector should represent a periodic activity. We first need to extract the frequency of oscillation, which methods like discrete Fourier transform or cross correlation can be used. After the frequency ϑ is determined, we create the phase data as $\theta_i = 2\pi\vartheta t_i$.

We can then use the values of y_i as the desired radius of the limit cycle. We add a constant offset δ_0 to y_i values in case they include negative values. This is due to the fact that f should be a positive function (other than the case where a base oscillator with a linear $\dot{r}_{\mathbb{B}}$ equation is used), and the same δ_0 should be subtracted when reading the output of the system. Consequently the data describing the scaling function f is defined as (with $\delta_0 > -\min_i(y_i)$):

$$f : \theta_i \longmapsto \frac{y_i + \delta_0}{\Xi_{\mathbb{B}}(\theta_i)},\ i = 1 \dots N \tag{42}$$

The dataset given in Equation (42) can be used to create any function approximator describing f, as long as it keeps the periodicity with a period of $1/\vartheta$. As recommended by [26], the shaping function can be modeled with normalized weighted periodic Gaussian-like bases, known as von Mises basis functions:

$$f(\theta) = \frac{\sum_{k=1}^{K} w_k \psi_k(\theta)}{\sum_{k=1}^{K} \psi_k(\theta)}, \quad \psi_k(\theta) = e^{\frac{1}{\sigma_k}(\cos(\theta - c_k) - 1)} \tag{43}$$

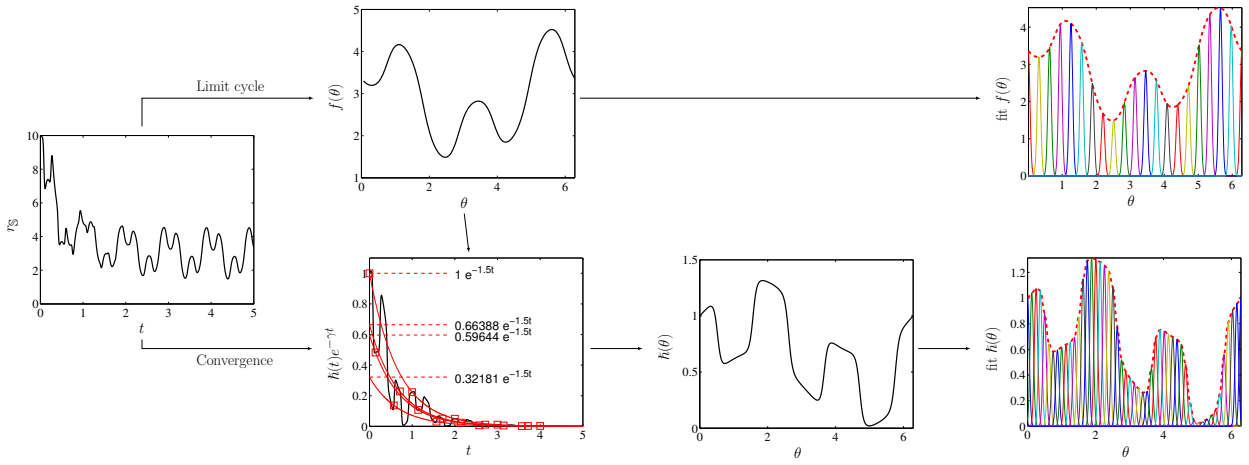

Figure 10: Learning the limit cycle and the convergence behaviors from an input data vector. The last cycle of the input data is used to extract the desired limit cycle and to build $f(\theta)$. Knowing $f(\theta)$, the convergence data is obtained by Equation (45), and is sampled at $2k\pi + \theta_o$ phases. The obtained sampled points are used to find an exponential fit. As the figure shows, different samples for different phase offsets θ_o result in the same convergence factor of $\gamma = 1.5$. After that the periodic part of the convergence function can be extracted. Both $f(\theta)$ and $\hbar(\theta)$ can be modeled with von Mises bases. This example uses 20 bases for $f(\theta)$ and 50 bases for $\hbar(\theta)$. The oscillator generating the input data for this example is described with $\gamma = 1.5$, $\Xi_S(\theta) = f(\theta) = 3 + \arctan(\sin(3\theta + 4)) + \cos(\theta)$, and $h(\theta) = 2 + \sin(\theta) + \tanh(3\cos(3\theta))$.

where ψ_k is a von Mises basis centered at phase c_k and σ_k determines the span. If f is modeled so, then there are powerful tools like locally weighted regression [52] to find the w_k parameters in an $O(KN)$ procedure. One additional benefit of using von Mises bases is that the resulting f function is smooth, i.e. it is C^∞ differentiable. This means that, even using the first order realization, all position, velocity, acceleration, etc states will be continuous, until perturbed.

We like to mention that one nice outcome of using a mixture of periodic basis functions to model f is that they can represent f in terms of simple movement/motor primitives, which can be linked to more biological explanation of how movements are coded and generated [53–55]. Rhythmic Dynamical Movement Primitives [26] are one example of such, and the framework here is a superset of rhythmic DMPs, and the notion of movement primitives apply as long as the f function is accordingly represented.

7.2. Learning the convergence function

In case where a morphed oscillator having a desired limit cycle shape but also with an explicit desired convergence behavior is of interest, one would use Equations (5,41) to model it. Let us assume that the given data $\{t_i, y_i\}, i = 1 \dots N$ describes the oscillation behavior from an initial condition y_0 which convergences to a periodic behavior[1]. We can simply take the last periods of oscillation (where the oscillator is already converged enough with respect to an error measure), and use this part to model f (Section 7.1). Knowing f, and by utilizing Equation (38), we have:

$$y_i + \delta_0 = f(\theta_i) + \frac{y_0 - f(\theta_0)}{h(0)} h(t) e^{-\gamma t} \tag{44}$$

which can be rewritten as:

$$\hbar(t, \theta) e^{-\gamma t} = \frac{h(t)}{h(0)} e^{-\gamma t} = \frac{y_t + \delta_0 - f(\theta_i)}{y_0 + \delta_0 - f(\theta_0)} \tag{45}$$

where $\hbar(t, \theta)$ is the normalized convergence function. If $\hbar$ is desired to be a function of time, then γ can chosen arbitrarily and then $\hbar$ is numerically obtained, and a function approximation tool of choice can be used to model $\hbar(t)$.

The other case is where $\hbar$ is meant to be a periodic function of phase $\hbar(\theta)$. So the term $\frac{y_t + \delta_0 - f(\theta_i)}{y_0 + \delta_0 - f(\theta_0)}$ should describe a pure periodic behavior multiplied by the exponential decay $e^{-\gamma t}$. This needs a correct estimation of γ. It is difficult, and

[1] Our approach here is limited to having one single example. If multiple examples are given, one can first apply the process explained for one example, and then use the median of the resulting parameters.

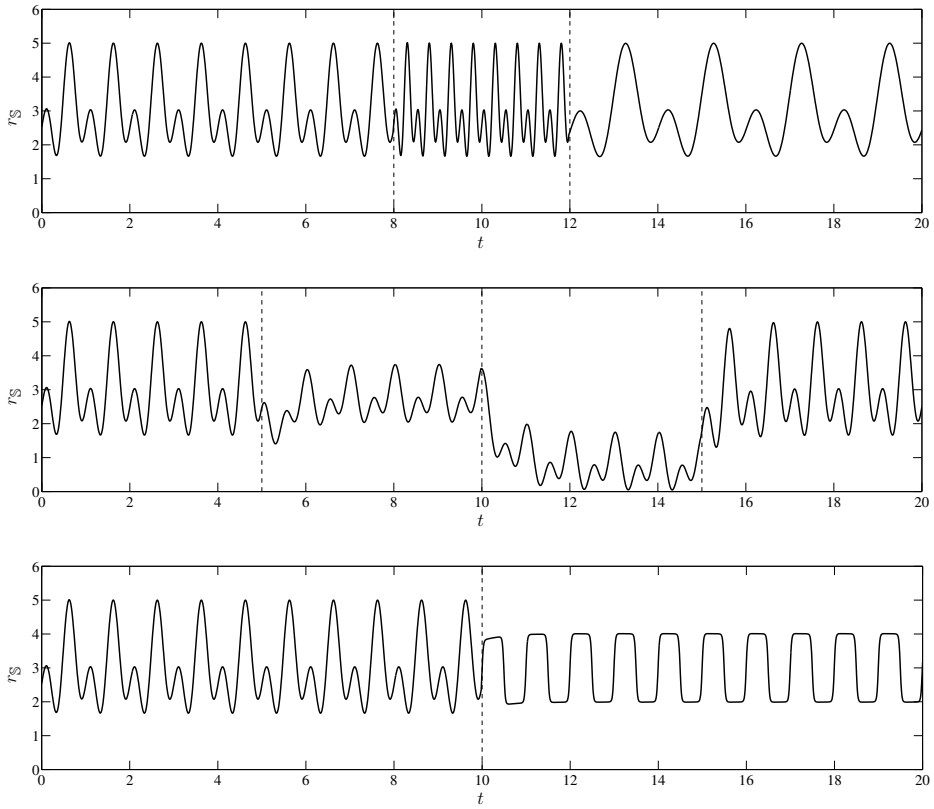

Figure 11: Online modulation of the limit cycle behavior. Top) modulation of frequency to $\vartheta = 2$ and then to $\vartheta = 0.5$. Middle) Modulation of amplitude by $a = 0.5$ around midpoint $g = 2.5$, then adding an offset of $o = -2$, and finally going back to the initial limit cycle. Bottom) Swapping the limit cycle with a new one. For all the figures $\gamma = 2$, and except the top figure, the value of γ determines the modulation/switching duration.

can also be imprecise, to estimate the shape of $\hbar(\theta)$ and the convergence factor γ at the same time. The solution is to calculate γ without knowing the form of $\hbar(\theta)$. Since $\hbar(\theta)$ is a non-damped periodic function, its relative displacement in $2k\pi\vartheta + \theta_o, k \in \mathbb{Z}$ phases is zero. This means that if we sample $\hbar(t)e^{-\gamma t}$ in $k\vartheta + \frac{\theta_i}{2\pi\vartheta} + t_o$ time stamps, or $2k\pi + \theta_o$ phases, then the sampled data fits on a pure exponential decay $\alpha e^{-\gamma t}$ (t_o and θ_o are arbitrary offsets). Finally, γ can simply be estimated by fitting $\alpha e^{-\gamma t}$ on the newly sampled data, e.g. by least squares. After γ is estimated, data describing periodic phase-dependent function $\hbar$ is:

$$\hbar : \theta_i \mapsto \frac{y_t + \delta_0 - f(\theta_i)}{y_0 + \delta_0 - f(\theta_0)} e^{\gamma \frac{\theta_i}{2\pi\vartheta}} \tag{46}$$

and again von Mises bases with locally weighted regression can be used to model this data, like what was done for f. Figure 10 depicts the procedure to extract the limit cycle and convergence behavior from a given data vector.

7.3. Online modulation

Different properties of the proposed morphed oscillators can be modulated online. Frequency modulation can be done by directly changing the ϑ values. This will have an immediate effect on the period of the system. Figure 11-top shows this property. Modulation of amplitude, offset and oscillation midpoint can be done by changing the f function on-the-fly. If the desired modulated output $\hat{\Xi}_{\mathbb{S}}(\theta)$ is:

$$\hat{\Xi}_{\mathbb{S}}(\theta) = a(\Xi_{\mathbb{S}}(\theta) - g) + g + o \tag{47}$$

where a is the amplitude magnification around midpoint g, and o is an added offset, then:

$$\hat{f}(\theta) = af(\theta) + \frac{(1-a)g + o}{\Xi_{\mathbb{B}}(\theta)} \tag{48}$$

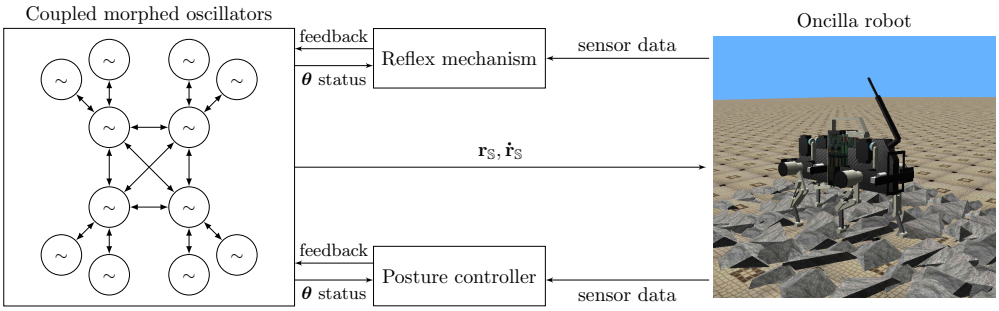

Figure 12: The control architecture based on the morphed oscillators used for rough terrain locomotion. The morphed oscillators are used to encode and generate the nominal locomotion patterns. Reflex and posture control mechanisms send feedback signals to the morphed oscillators, as additive feedbacks to $\dot{r}_{\mathbb{S},i}$. All the feedbacks are only active for certain locomotion phases, and the morphed oscillators converge back to the encoded limit cycle when the feedbacks are inactive. This effect, along the existence of the posture controller, enables stable locomotion over unperceived rough terrain. Videos can be found at `http://biorob.epfl.ch/page-89661-en.html`.

where $\hat{f}(\theta)$ is the new scaling function giving the desired modulation. Examples are given in Figure 11. The effects of these modulations are not immediate, and act as swapping the limit cycle of the system with a new one, so the system will gradually converge to the modulated limit cycle. All the above holds when f is replaced with a completely new $\hat{f}$ which can have a different shape than f, as the bottom plot in Figure 11 shows.

We like to mention that since the oscillators obtained by the introduced methodology are phase oscillators, frequency adaptation rule can easily be applied to them. With reference to [38], a frequency adaptation rule for entrainment [56] with an external periodic input can be simply written as:

$$\dot{\omega} = -\eta I(t) \sin(\theta) \tag{49}$$

where $I(t)$ is the external periodic signal and η is the adaptation rate. This means that instead of having a constant ω in the phase equation $\dot{\theta} = \omega$, an adaptive dynamics is applied to $\dot{\omega}$. Please refer to [38] for details.

8. Applications

Morphed oscillators can be applied to different problems when there is a need to encode a desired periodic pattern as a limit cycle of a dynamical system. Moreover, morphed oscillators can be designed to be globally asymptotically stable, and have a desired convergence behavior, which advocates applications requiring state feedback integration. The concept of the morphed oscillators is new, so the list applications that we present is a combination of recently implemented ones, and propositions:

Locomotion: As we know from animals, the periodic joint angle profiles during different stages of locomotion are nonlinear and can be rather complex. The reader can refer to [57] for a sample human joint angle profiles, and to [58] for a sample feline joint angle profiles. It is possible to use a simple phase generating system and apply output functions to directly obtain desired control signals (like $r_{\mathbb{S}} = f(\theta)$). This can be sufficient in cases where no feedback signal should be fed back to the oscillator (e.g. walking on a flat surface). However, as soon as there are feedback signals (e.g. for walking on irregular terrain), using an output function can lead to output discontinuities or non-smoothness ($r_{\mathbb{S}} = f(\theta) + e$ becomes discontinuous or non-smooth if e is discontinuous or non-smooth). Morphed oscillators provide a systematic way to *encode* the desired joint angles profiles in oscillator dynamics with known stability properties. This enables the control engineer to design feedback signals without the worry of pattern generation discontinuity or instability. We have recently applied morphed oscillators to the control of simulated quadrupeds locomoting over unperceived rough terrain. Figure 12 illustrates the aforementioned locomotion control scheme based on morphed oscillators, which is detailed in [59–61].

Imitation: Morphed oscillators can be used for imitation of periodic tasks. In imitation, the tutor's demonstration can be of arbitrary complexity in terms of the trajectory shape. [37, 41, 42, 62] give several examples of periodic tasks that can be imitated with a robot, including drumming. As discussed above with locomotion, such periodic tasks

can be done using phase generating systems and output functions. In case of imitation however, there are two issues using output functions: 1) Like in locomotion, adding feedback tends to output discontinuity or non-smoothness. For example in the drumming task, if the drums are repositioned, there needs to be feedback signals to adapt for proper contact. 2) Imitation involves switching between different motor tasks, and using output functions can result in discontinuities when switching. Using morphed oscillators, one can code the imitated trajectories in dynamical systems which allow for feedback integration and also make the task transitions smooth (e.g. like the limit cycle modulation in Figure 11).

Neurorobotics: Coupled morphed oscillators can be used to create high-level computational Central Pattern Generator (CPG) models [3, 8, 63]. In this case, an oscillator represents the activity of a complete oscillatory center (instead of a single neuron or a small circuit) [4]. The study in [8] shows how coupled oscillators can be used to describe swimming, walking and the transition between them in salamanders, and applies those oscillators to the control of a salamander robot. The aforementioned study uses sine-wave outputs to control the spine joints. However recent recordings from salamanders [64] show that spine movements are not purely sine-wave and have more nonlinearities. One can use the morphed oscillators instead of amplitude controlled oscillators in [8] to create a more accurate high-level model of the salamander CPGs. Moreover, using morphed oscillators allows for encoding of limb joint angle trajectories as well, which are considerably more complex than spine trajectories [64].

As another example, [65] uses time-driven pattern generators to activate a muscle model controlling the locomotion of a simulated salamander. The output patterns are used as abstract neural activation of the muscles. They only test their model for open loop pattern generation. Morphed oscillators can be applied if one desires to equip such muscle pattern generators with feedback mechanisms. Morphed oscillators will then encode the activation pattern in stable oscillators which allow for smooth feedback integration.

Synchronization and assistance: Morphed oscillators give two-layer oscillators where the phase dynamics are decoupled from the radial dynamics. Having the decoupling, one can control radial dynamics limit cycle and convergence properties separately and leave the phases for eigenfrequency control. This motivates applications like model-free tracking in assistance and rehabilitation robotics, such as locomotion support using exoskeletons [66, 67] or inter-agents phase synchronization [68] when the frequency coupling can be stably separated from radial dynamics.

Task initiation: As discussed in [51], "all periodic motions must be started in a nonperiodic way before the repeating pattern comes into play". It is important to be able to define the initiation trajectories of a periodic task, and the methods to do this with one single dynamical system are scarce [51]. As we showed in Section 6, morphed oscillators can be used to encode the desired initiation/convergence behavior into the same dynamical system that encodes the desired periodic task.

Stability analysis: The stability analysis given in Section 5 can be utilized to analyze the stability of a subset of complex systems. To give an example consider the following time-dependent system:

$$\dot{x} = -x\sin(t) + \left(1 - \frac{x^4}{e^{4\cos(t)}}\right)\left(\frac{2x}{e^{\cos(t)}} + \sin(\frac{x}{e^{\cos(t)}})\right)$$

It is not trivial to analyze the stability of this time-dependent system and describe its limit cycle properties (for example Matlab or Mathematica cannot solve this equation). However one can reformulate this system as an original morphed oscillator with $f(\theta) = e^{\cos(\theta)}$ and the base:

$$\dot{\theta} = 1, \quad \dot{r}_{\mathbb{B}} = (1 - r_{\mathbb{B}}^4)(2r_{\mathbb{B}} + sin(r_{\mathbb{B}}))$$

which has one stable fixed point at $r_{\mathbb{B}} = 1$ and an unstable one at $r_{\mathbb{B}} = 0$. This base satisfies the conditions in Equation (24) for $r_{\mathbb{B}} > 0$, and consequently, utilizing Equation (21), the system converges to the time driven limit cycle $e^{\cos(t)}$ for $x > 0$. The same analogy can be used to analyze the stability conditions of other similar dynamical systems if they can be rewritten into a morphed oscillator form. This method can be helpful in cases where classical stability analyses fall short.

One should keep in mind that morphed oscillators provide only the core building block for the above applications, and should be enriched with proper and task-specific feedback mechanisms. The role of the morphed oscillator is to encode the desired periodic trajectories into stable oscillators of desired order, which allows for feedback integration, smooth modulations and continuous transitions.

Many of the above applications can also be done applying rhythmic Dynamical Movement Primitives (DMP). So the question is why to design morphed oscillators. There are three answers to this question:

1. Rhythmic DMP is a 2nd order form of the morphed oscillators having a base with linear radial dynamics (see Section 3.1). Consequently what can be done with a rhythmic DMP can also be done with a morphed oscillator. Moreover, the idea of DMP builds upon the use of Gaussian like basis functions in order to represent the limit cycle shape *indirectly* via the forcing term, while the methodology of morphed oscillators allows for *direct* definition of the limit cycle shape using any desired tool for the representation of the shaping function (Gaussians, splines, etc).
2. Using the morphed oscillators, the system designer has the option to choose the order of the dynamical system. As it has been shown in [59], tasks like locomotion can be done with 1st order morphed oscillators, which eliminates extra numerical integrations of a 2nd order system. On the other hand, [69] shows that 3rd order dynamics can be useful for sequencing robotic tasks which need acceleration continuity. Moreover, if one looks for jerk continuity in a periodic task, as in [70], then a 4th order system should be applied.
3. The methodology of morphed oscillators allows for a variety of different convergence behaviors. One can use different bases to obtain different levels of nonlinearity in the convergence. For example, using a base with linear radial dynamics (like DMP) results in simple exponential convergence, while using a base with polynomial radial dynamics can give a convergence behavior which is weak near the limit cycle and strong when far. One should keep in mind that the convergence behavior affects the external feedback effect, and having the choice of convergence behavior gives the system designer the freedom to properly choose the needed convergence.

9. Discussion

We presented a general methodology to morph a chosen phase oscillator, which acts as a base oscillator, to an oscillator with a desired limit cycle shape. The main idea of this methodology is based on a diffeomorphic phase-based scaling map which morphs the dynamics of the base to the desired one. The given methodology creates first order oscillators, and was extended to represent second order and $n-$th order oscillators. This realizes a general and populated family of nonlinear oscillators with any desired order.

Compared to the general approach of using recurrent neural networks to create nonlinear oscillators, using the presented methodology will reduce the design/training complexity. If one desires to obtain a desired limit cycle behavior out of a recurrent neural network, then he/she should employ rather complex training techniques like backpropagation through time [71], and check for local asymptotic stability and unwanted local minima afterwards. The training technique in these approaches need to know about the internal dynamics of the network. However, for the morphed oscillators, the training process (to model the limit cycle shaping function f) only needs to know the limit cycle shape of the base oscillator, and not its transient dynamics, to be able to create the desired nonlinear oscillator. This gives the advantage that the training procedure gets reduced to a static function approximation.

Other than being an interesting mathematical challenge to create nonlinear oscillators with arbitrary limit cycle shapes, many motion control applications need such oscillators which makes this problem even more interesting. One very good example is the control of locomotion and use of nonlinear oscillators as pattern generators. This is widely known as the problem of designing Central Pattern Generator (CPG) [4] models for locomotion. As Ijspeert mentions in [4], to be able to systematically create CPGs, design of coupled nonlinear oscillator exhibiting desired limit cycles should be tackled. We believe that the approach here is general and systematic and helps to ease the design of CPGs. Moreover, since one can design globally asymptotically stable limit cycle systems with the introduced methodology, the inclusion of feedback signals will not affect the stability properties, and consequently broadens the types of feedback signals that can be included.

The nonlinear oscillators obtained by the introduced methodology are one dimensional phase oscillators, and can get phase coupled to create a multidimensional system. This is different from a recurrent neural network which is multidimensional by design. The methodology here makes the creation of a multidimensional system easy by dividing

it into low dimensional subsystems. However the radial dynamics of different dimensions are not directly coupled. This means that a perturbation on one dimension's radial state will not affect the other dimensions, other than when being explicitly coupled. This is different from a recurrent neural network where all the state dynamics are normally coupled, and this can be considered as an advantage or disadvantage depending on the application.

We also introduced the possibility to have an explicitly defined custom convergence behavior. This custom convergence behavior is apart from the choice of the limit cycle shape, and both arbitrary limit cycle shape and convergence can be obtained in one same dynamical system. This gives the possibility to include periodic tasks as well as their non-periodic initiation in one single system. This is useful in many rhythmic motor control tasks which need initiation, like locomotion. We have also explained how to learn both the limit cycle shape and the convergence behavior from given data, which makes this tool appropriate to be used for learning rhythmic tasks by imitation.

The stability analysis for the general family obtained from the introduced methodology was given. If one is looking for globally asymptotically stable limit cycle systems, the stability conditions can direct him/her to the choice of the base oscillators he/she has. More importantly, if one is forced to use a specific base (e.g. by implementation constraints), the given stability analysis can be used to know the stability bounds of the resulting system, and ensure a safe system design.

In the end, we expect the introduced methodology to ease and systematize the design of nonlinear phase oscillators for different applications including robotics and motion control. Morphed oscillator are computationally light and simple to implement, which makes them appropriate tools for robotics application with real-time constraints. We are exploiting these features, and already used morphed oscillators as the pattern generators for quadruped robots locomoting over rough terrain [59–61]. Moreover, morphed oscillators are the superset of the rhythmic DMPs, and this places the applications of DMPs within the scope of the morphed oscillators, with extended design choices.

Acknowledgment

We would like to thank Alexandre Tuleu and the rest of the Oncilla team for providing us with the simulator. The research leading to these results has received funding from the European Communitys Seventh Framework Programme FP7/2007-2013 Challenge 2 Cognitive Systems, Interaction, Robotics - under grant agreement No 248311 - AMARSi.

References

[1] H. K. Khalil, Nonlinear systems; 3rd ed., Prentice-Hall, Upper Saddle River, NJ, 2002.
[2] I. Kovacic, M. Brennan, The Duffing equation: Nonlinear oscillators and their behaviour, Wiley, 2011.
[3] K. Matsuoka, Mechanisms of frequency and pattern control in the neural rhythm generators, Biological Cybernetics 56 (1987) 345–353.
[4] A. Ijspeert, Central pattern generators for locomotion control in animals and robots: A review, Neural Networks 21 (4) (2008) 642 – 653.
[5] S. H. Strogatz, I. Stewart, et al., Coupled oscillators and biological synchronization, Scientific American 269 (6) (1993) 102–109.
[6] J. J. Collins, I. N. Stewart, Coupled nonlinear oscillators and the symmetries of animal gaits, Journal of Nonlinear Science 3 (1) (1993) 349–392.
[7] M. Golubitsky, I. Stewart, P.-L. Buono, J. Collins, Symmetry in locomotor central pattern generators and animal gaits, Nature 401 (6754) (1999) 693–695.
[8] A. J. Ijspeert, A. Crespi, D. Ryczko, J.-M. Cabelguen, From swimming to walking with a salamander robot driven by a spinal cord model, Science 315 (5817) (2007) 1416–1420.
[9] J. S. Bay, H. Hemami, Modeling of a neural pattern generator with coupled nonlinear oscillators, Biomedical Engineering, IEEE Transactions on (4) (1987) 297–306.
[10] G. Taga, Y. Yamaguchi, H. Shimizu, Self-organized control of bipedal locomotion by neural oscillators in unpredictable environment, Biological cybernetics 65 (3) (1991) 147–159.
[11] K. Tsujita, K. Tsuchiya, A. Onat, Adaptive gait pattern control of a quadruped locomotion robot, in: Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on, Vol. 4, IEEE, 2001, pp. 2318–2325.
[12] M. S. Dutra, A. C. de Pina Filho, V. F. Romano, Modeling of a bipedal locomotor using coupled nonlinear oscillators of van der pol, Biological Cybernetics 88 (4) (2003) 286–292.
[13] L. Righetti, A. Ijspeert, Programmable central pattern generators: an application to biped locomotion control, in: Proceedings of the 2006 IEEE International Conference on Robotics and Automation, 2006.
[14] L. Righetti, A. J. Ijspeert, Design methodologies for central pattern generators: an application to crawling humanoids, in: Proceedings of robotics: science and systems, MIT Press Cambridge, MA, 2006, pp. 191–198.
[15] S. Rutishauser, A. Sprowitz, L. Righetti, A. J. Ijspeert, Passive compliant quadruped robot using central pattern generators for locomotion control, in: Biomedical Robotics and Biomechatronics, 2008. BioRob 2008. 2nd IEEE RAS & EMBS International Conference on, IEEE, 2008, pp. 710–715.

[16] L. Righetti, A. J. Ijspeert, Pattern generators with sensory feedback for the control of quadruped locomotion, in: Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on, IEEE, 2008, pp. 819–824.
[17] S. Pouya, J. van den Kieboom, A. Sprowitz, A. J. Ijspeert, Automatic gait generation in modular robots:to oscillate or to rotate; that is the question, in: Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on, IEEE, 2010, pp. 514–520.
[18] C. Maufroy, H. Kimura, K. Takase, Stable dynamic walking of a quadruped via phase modulations against small disturbances, in: Robotics and Automation, 2009. ICRA'09. IEEE International Conference on, IEEE, 2009, pp. 4201–4206.
[19] A. Tuleu, M. Ajallooeian, A. Sprröwitz, P. Loepelmann, A. Ijspeert, Trot gait locomotion of a cat sized quadruped robot, in: International Workshop on BioInspired Robots, 2011, pp. 1–4.
[20] M. Khoramshahi, A. Sprowitz, A. Tuleu, M. Ahmadabadi, A. Ijspeert, Benefits of an active spine supported bounding locomotion with a small compliant quadruped robot, in: IEEE International Conference on Robotics and Automation (ICRA 2013), 2013.
[21] P. Arena, The central pattern generator: a paradigm for artificial locomotion, Soft Computing 4 (4) (2000) 251–266.
[22] Y. Fukuoka, H. Kimura, A. Cohen, Adaptive dynamic walking of a quadruped robot on irregular terrain based on biological concepts, The International Journal of Robotics Research 22 (3-4) (2003) 187–202.
[23] Z. Lu, S. Ma, B. Li, Y. Wang, Serpentine locomotion of a snake-like robot controlled by cyclic inhibitory cpg model, in: Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on, IEEE, 2005, pp. 96–101.
[24] F. wyffels, B. Schrauwen, Design of a central pattern generator using reservoir computing for learning human motion, Advanced Technologies for Enhanced Quality of Life 0 (2009) 118–122.
[25] J. Nakanishi, J. Morimoto, G. Endo, G. Cheng, S. Schaal, M. Kawato, Learning from demonstration and adaptation of biped locomotion, Robotics and Autonomous Systems 47 (2) (2004) 79–91.
[26] A. Ijspeert, J. Nakanishi, P. Pastor, H. Hoffmann, S. Schaal, Dynamical movement primitives: learning attractor models for motor behaviors, Neural Computation (25) (2013) 328–373.
[27] G. Jouffroy, Design of oscillatory recurrent neural network controllers with gradient based algorithms, in: ESANN 2008, 16th European Symposium on Artificial Neural Networks, Bruges, Belgium, April 23-25, 2008, Proceedings, 2008, pp. 7–12.
[28] B. A. Pearlmutter, Learning state space trajectories in recurrent neural networks, Neural Computation 1 (1989) 263–269.
[29] S. Townley, A. Ilchmann, M. G. Weiß, W. Mcclements, A. C. Ruiz, D. H. Owens, D. Prtzel-Wolters, Existence and learning of oscillations in recurrent neural networks, IEEE Transactions on Neural Networks 11 (2000) 263–269.
[30] A. Ruiz, D. Owens, S. Townley, Existence, learning, and replication of periodic motions in recurrent neural networks, IEEE Transactions on Neural Networks 9 (4) (1998) 651 –661.
[31] K. Doya, S. Yoshizawa, Adaptive neural oscillator using continuous-time back-propagation learning, Neural Networks 2 (5) (1989) 375 – 385.
[32] Y. Kuroe, H. Lima, A learning method for synthesizing spiking neural oscillators, in: Neural Networks, 2006. IJCNN '06. International Joint Conference on, 2006, pp. 3882 –3886.
[33] Y. Kuroe, K. Miura, A method of oscillatory trajectory generation using recurrent hybrid neural networks, in: Neural Networks, 2005. IJCNN '05. Proceedings. 2005 IEEE International Joint Conference on, Vol. 2, 2005, pp. 706 – 711 vol. 2.
[34] E. Leclercq, F. Druaux, D. Lefebvre, S. Zerkaoui, Autonomous learning algorithm for fully connected recurrent networks, Neurocomputing 63 (0) (2005) 25 – 44.
[35] B. Schrauwen, D. Verstraeten, J. Van Campenhout, An overview of reservoir computing: theory, applications and implementations, in: Proceedings of the 15th European Symposium on Artificial Neural Networks, 2007, pp. 471–482.
[36] M. Okada, K. Tatani, Y. Nakamura, Polynomial design of the nonlinear dynamics for the brain-like information processing of whole body motion, in: Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on, Vol. 2, 2002, pp. 1410 – 1415 vol.2.
[37] M. Ajallooeian, M. Nili Ahmadabadi, B. N. Araabi, H. Moradi, Design, implementation and analysis of an alternation-based Central Pattern Generator for multidimensional trajectory generation, Robotics and Autonomous Systems 60 (2) (2012) 182–198.
[38] L. Righetti, J. Buchli, A. Ijspeert, Dynamic hebbian learning in adaptive frequency oscillators, Physica D 216 (2) (2006) 269–281.
[39] M. Minsky, Steps toward artificial intelligence, Proceedings of the IRE 49 (1) (1961) 8–30.
[40] T. Petri, A. Gams, A. J. Ijspeert, L. Ålajpah, On-line frequency adaptation and movement imitation for rhythmic robotic tasks, The International Journal of Robotics Research 30 (14) (2011) 1775–1788. arXiv:http://ijr.sagepub.com/content/30/14/1775.full.pdf+html.
[41] A. Ijspeert, J. Nakanishi, S. Schaal, Learning Attractor Landscapes for Learning Motor Primitives, in: S. Becker, S. Thrun, K. Obermayer (Eds.), Advances in Neural Information Processing Systems 15 (NIPS2002), 2002, pp. 1547–1554.
[42] A. Ijspeert, J. Nakanishi, S. Schaal, Learning Rhythmic Movements by Demonstration using Nonlinear Oscillators, in: Proceedings of the IEEE/RSJ Int. Conference on Intelligent Robots and Systems (IROS2002), 2002, pp. 958–963.
[43] A. Ijspeert, J. Nakanishi, S. Schaal, Movement imitation with nonlinear dynamical systems in humanoid robots, in: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA2002), 2002, pp. 1398–1403.
[44] A. Gams, A. Ijspeert, S. Schaal, J. Lenari, On-line learning and modulation of periodic movements withnonlinear dynamical systems, Autonomous Robots 27 (2009) 3–23.
[45] R. FitzHugh, Impulses and physiological states in theoretical models of nerve membrane, Biophysical Journal 1 (6) (1961) 445 – 466.
[46] S. Degallier, C. Santos, L. Righetti, A. Ijspeert, Movement generation using dynamical systems: a humanoid robot performing a drumming task, in: Humanoid Robots, 2006 6th IEEE-RAS International Conference on, IEEE, 2006, pp. 512–517.
[47] S. Strogatz, From Kuramoto to Crawford: exploring the onset of synchronization in populations of coupled oscillators, Physica D: Nonlinear Phenomena 143 (1) (2000) 1–20.
[48] J. Guckenheimer, P. Holmes, Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields, Springer-Verlag, New York, USA, 1983.
[49] W. Lohmiller, J. Slotine, On contraction analysis for non-linear systems, Automatica 34 (6) (1998) 683 – 696.
[50] W. Fulton, Algebraic topology: a first course, Vol. 153, Springer, 1995.
[51] J. Ernesti, L. Righetti, M. Do, T. Asfour, S. Schaal, Encoding of periodic and their transient motions by a single dynamic movement primitive, in: 2012 ieee-ras international conference on humanoid robots, 2012, pp. 57–64.

[52] S. Schaal, C. Atkeson, Constructive incremental learning from only local information, Neural Comput. 10 (8) (1998) 2047–2084.
[53] T. Flash, B. Hochner, Motor primitives in vertebrates and invertebrates, Current Opinion in Neurobiology 15 (6) (2005) 660 – 666.
[54] F. A. Mussa-Ivaldi, S. F. Giszter, E. Bizzi, Linear combinations of primitives in vertebrate motor control, Proceedings of the National Academy of Sciences of the United States of America 91 (16) (1994) pp. 7534–7538.
[55] K. A. Thoroughman, R. Shadmehr, Learning of action through adaptive combination of motor primitives, Nature 407 (2000) 742–747.
[56] A. Pikovsky, M. Rosenblum, J. Kurths, Synchronization: A universal concept in nonlinear sciences, Vol. 12, Cambridge university press, 2003.
[57] R. H. Miller, S. Brandon, K. J. Deluzio, et al., Predicting sagittal plane biomechanics that minimize the axial knee joint contact force during walking., Journal of biomechanical engineering 135 (1) (2013) 011007–011007.
[58] E. Hasler, W. Herzog, T. Leonard, A. Stano, H. Nguyen, In vivo knee joint loading and kinematics before and after acl transection in an animal model-temporal and angular changes in a canine model of osteoarthritis, Journal of Biomechanics 31 (3) (1997) 253–262.
[59] M. Ajallooeian, S. Pouya, A. Sproewitz, A. Ijspeert, Central pattern generators augmented with virtual model control for quadruped rough terrain locomotion, in: IEEE International Conference on Robotics and Automation (ICRA 2013), 2013.
[60] M. Ajallooeian, S. Pouya, S. Gay, A. Tuleu, A. Sprowitz, A. Ijspeert, Towards modular control for moderately fast locomotion over unperceived rough terrain, in: Dynamic Walking 2013.
[61] M. Ajallooeian, S. Gay, A. Tuleu, A. Sprowitz, A. J. Ijspeert, Modular control of limit cycle locomotion over unperceived rough terrain, in: Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on, 2013.
[62] A. Gams, S. Degallier, A. Ijspeert, J. Lenarcic, Dynamical system for learning the waveform and frequency of periodic signals–application to drumming, in: Proceedings of the 17th International Workshop on Robotics in Alpe-Adria-Danube Region (RAAD2008), Citeseer, 2008.
[63] A. H. Cohen, P. J. Holmes, R. H. Rand, The nature of the coupling between segmental oscillators of the lamprey spinal generator for locomotion: a mathematical model, Journal of mathematical biology 13 (3) (1982) 345–369.
[64] K. Karakasiliotis, N. Schilling, J.-M. Cabelguen, A. J. Ijspeert, Where are we in understanding salamander locomotion: biological and robotic perspectives on kinematics, Biological cybernetics (2012) 1–16.
[65] N. Harischandra, J.-M. Cabelguen, Ö. Ekeberg, A 3d musculo-mechanical model of the salamander for the study of different gaits and modes of locomotion, Frontiers in neurorobotics 4.
[66] R. Ronsse, N. Vitiello, T. Lenzi, J. van den Kieboom, M. Carrozza, A. Ijspeert, Adaptive oscillators with human-in-the-loop: Proof of concept for assistance and rehabilitation, in: Biomedical Robotics and Biomechatronics (BioRob), 2010 3rd IEEE RAS and EMBS International Conference on, 2010, pp. 668 –674.
[67] R. Ronsse, N. Vitiello, T. Lenzi, J. van den Kieboom, M. Carrozza, A. Ijspeert, Human-robot synchrony: Flexible assistance using adaptive oscillators, Biomedical Engineering, IEEE Transactions on 58 (4) (2011) 1001 –1012.
[68] A. Mukovskiy, J.-J. E. Slotine, M. A. Giese, Dynamically stable control of articulated crowds, Journal of Computational Science 4 (2013) 304–310. doi:10.1016/j.jocs.2012.08.019.
[69] B. Nemec, A. Ude, Action sequencing using dynamic movement primitives, Robotica 30 (5) (2012) 837.
[70] K. Petrinec, Z. Kovacic, Trajectory planning algorithm based on the continuity of jerk, in: Control & Automation, 2007. MED'07. Mediterranean Conference on, IEEE, 2007, pp. 1–5.
[71] D. E. Rumelhart, G. E. Hinton, R. J. Williams, Parallel distributed processing: explorations in the microstructure of cognition, vol. 1, MIT Press, Cambridge, MA, USA, 1986, Ch. Learning internal representations by error propagation, pp. 318–362.