Sound Synthesis Based on Ordinary Differential Equations

Nikolaos Stefanakis,* Markus Abel,† and André Bergner**

*UP Transfer GmbH
University of Potsdam
Am Neuen Palais 10
14469 Potsdam, Germany
nstefana@ics.forth.gr
†Ambrosys GmbH
Albert-Einstein-Str. 1–5
14467 Potsdam, Germany
markus.abel@ambrosys.de
**Native Instruments GmbH
Schlesische Straße 29–30
10997 Berlin, Germany
andre.bergner.0@gmail.com

Abstract: Ordinary differential equations (ODEs) have been studied for centuries as a means to model complex dynamical processes from the real world. Nevertheless, their application to sound synthesis has not yet been fully exploited. In this article we present a systematic approach to sound synthesis based on first-order complex and real ODEs. Using simple time-dependent and nonlinear terms, we illustrate the mapping between ODE coefficients and physically meaningful control parameters such as pitch, pitch bend, decay rate, and attack time. We reveal the connection between nonlinear coupling terms and frequency modulation, and we discuss the implications of this scheme in connection with nonlinear synthesis. The ability to excite a first-order complex ODE with an external input signal is also examined; stochastic or impulsive signals that are physically or synthetically produced can be presented as input to the system, offering additional synthesis possibilities, such as those found in excitation/filter synthesis and filter-based modal synthesis.

Ordinary differential equations (ODEs) and time-difference equations are the two most common mathematical tools used for the analysis and modeling of dynamical processes of the real world. They are of primary interest in physics, with applications in qualitative analysis of dynamical systems (e.g., by dimension, Lyapunov exponents, or entropies), in signal prediction, and in classification (Packard et al. 1980; Gouesbet and Letellier 1993; Baake et al. 1992).

The techniques to solve these problems are usually subsumed under the term "reconstruction," since one is faced with the question of how to determine a system given the data, i.e., how to solve the inverse problem (Kantz and Schreiber 1997). In music, dynamical systems have been proposed as a means for modeling and synthesizing natural sounds. The particular theory is appealing when applied to the domain of speech, as well as to that of many physical instruments exhibiting nonlinear behavior, and their resynthesis has not been solved in a satisfactory manner by linear methods. Work by Axel Röbel (2001) is an excellent paradigm of how this method can be applied to the resynthesis of sustained sound, and Bernd Schoner (1996) has extended the research to a case where the dynamic model is designed to accept an input signal as well.

At the same time, dynamical systems are attractive from the perspective of synthesis, i.e., for the generation of completely artificial structures. The concepts of chaos and fractals have inspired researchers to exploit nonlinear dynamical systems for sound synthesis and musical design since the days of analog synthesizers (Slater 1998). Indeed, simple and deterministic dynamic models prove to be very efficient synthesis engines, producing surprisingly complex behavior (Argyris, Faust, and Haase 1995). Researchers have exploited this type of behavior in connection with parametric control of nonlinear maps (Pressing 1988; Manzolli 1993; Damiani et al. 2000), or in connection with physical models (Lindemann 1988; Rodet and Vergez 1999a, b) as a means to extend the user-defined timbral

Computer Music Journal, 39:3, pp. 46–58, Fall 2015
doi:10.1162/COMJ_a_00314

parameters of these models and to create routes to chaos.

This article presents a general framework for sound synthesis based on the numerical integration of a system of coupled nonlinear ODEs, which can be driven by an external input signal. We believe that, although the mathematical and physical aspects of such systems are more or less known, their potential in sound synthesis has not yet been fully addressed. Furthermore, even when ODEs have been exploited for such purposes in prior work, to our knowledge none has revealed the significant advantage that arises when using ODEs in the complex domain rather than in the real domain. As will be shown, a first-order complex ODE offers a set of control parameters that are much more meaningful, both physically and functionally, in comparison with a real ODE. We start the discussion with a linear first-order ODE with a time-dependent term that can be analytically "tuned" to respect desired amplitude and temporal characteristics. We then introduce obvious nonlinear terms for enriching the bandwidth of the generated sounds and for producing dynamic behavior that is more complex. As will be shown, a system of coupled deterministic ODEs becomes a powerful synthesis engine with both additive synthesis and nonlinear synthesis capabilities. At the same time, use of external input in the ODEs allows the implementation of forms of synthesis such as excitation/filter synthesis and filter-based modal synthesis. The rest of the article conveys the necessary technical information associated with ODE sound synthesis. Finally, sound samples are available online at mitpressjournals.org/doi/suppl/10.1162/COMJ_a_00314.

General ODE Framework

Our synthesis platform is formulated in terms of the general framework of N first-order complex ODEs (CODEs)

$$\begin{aligned}\dot{y}_1 &= f_1(t, y_1, y_2, \ldots, y_N, x_1),\\ \dot{y}_2 &= f_2(t, y_1, y_2, \ldots, y_N, x_2),\\ &\vdots\\ \dot{y}_N &= f_N(t, y_1, y_2, \ldots, y_N, x_N),\end{aligned} \tag{1}$$

with state-space coordinates $y_1, y_2, \ldots, y_N \in \mathbb{C}$. Here, the overdots denote differentiation with respect to time t, i.e., $\dot{y} = dy/dt$, and $x_1, x_2, \ldots, x_N$ denote a real or complex external input signal. There is a significant advantage in using a complex variable here: $y = re^{j\theta}$, which means that y provides simultaneous information about the phase and the envelope at each time instant. The parameters of the nth oscillator are encoded inside the definition of the standard function f_n. Given $f_n(\cdot)$ and an initial condition $y_{0,n}$ for all $n = 1, 2, \ldots, N$, the dynamical system above can be numerically integrated, returning N different complex outputs.

$$\begin{aligned}\hat{y}_1(t) &= \int_0^t f_1(t, y_1, y_2, \ldots, y_N, x_1)dt,\ \ y_1(0) = y_{0,1},\\ \hat{y}_2(t) &= \int_0^t f_2(t, y_1, y_2, \ldots, y_N, x_2)dt,\ \ y_2(0) = y_{0,2},\\ &\vdots\\ \hat{y}_N(t) &= \int_0^t f_N(t, y_1, y_2, \ldots, y_N, x_N)dt,\ \ y_N(0) = y_{0,N}.\end{aligned} \tag{2}$$

A sound signal $s[k] = s(kT)$ can then be acquired at a sampling rate $F_s = 1/T$ as a mixture of all oscillator outputs at each time index k as

$$s[k] = \sum_{n=1}^{N} \mu_n \mathrm{Re}\{\hat{y}_n[k]\}, \tag{3}$$

where $\hat{y}_n[k] = \hat{y}_n(kT)$, μ_n is the user-defined gain of the nth oscillator, and $\mathrm{Re}\{\cdot\}$ denotes the real part of a complex number. At first glance, the approach presented reveals a connection to additive synthesis; each equation in the system of ODEs can be associated with a different mode of vibration and a synthesis result can be obtained as a superposition of modes. On the other hand, in the context of nonlinear dynamics, the same system is expected to be capable of producing quasiperiodic and even chaotic behavior. It will be shown in the course of this article that the addition of simple nonlinear terms can provide an efficient way to enrich the bandwidth of the generated sounds and that nonlinear coupling of ODEs is the cause of frequency modulation.

Although the complex ODE framework is of primary interest in this article, it is also worth considering an identical mathematical framework in terms of real ODEs, in which case $y_1, y_2, \ldots, y_N \in \mathbb{R}$. The real ODE (RODE) framework can be exploited for designing envelopes, and these envelopes can be used for modulating a deterministic or stochastic input signal. The use of a noise signal as input is particularly interesting in the so-called subtractive-synthesis approach. For synthesis of drum sounds, for example, a CODE and a RODE can be used, respectively, for designing deterministic and stochastic components that can be combined to produce a more complete timbre of the percussive instrument, similar to that which can be found in the work of Aramaki and colleagues (2006).

Linear ODE with Time-Dependence

As a starting point, in this section we introduce a simple linear ODE that includes time dependence as a means for achieving desired temporal and amplitude characteristics. Our approach has the *gammatone* function (Cooke 1993) as a prototype function. Gammatones are used in auditory modeling for constructing time-domain filters that approximate the frequency resolution of the human cochlea. For our purposes, we consider a slightly modified gammatone function, defined strictly for $t \geq 0$ in the complex domain as

$$y(t) = \alpha(t+\epsilon)^b e^{(\sigma + j\omega)t}. \tag{4}$$

Here α and b are the amplitude and the order of the modified gammatone, $\omega = 2\pi f$ and σ are the radial frequency and decay rate, and ϵ is a small positive constant that is necessary for reasons that will become apparent later. Whereas a modified gammatone of order $b = 0$ is essentially an exponentially decaying complex harmonic, orders of b greater than zero are excellent examples of sounds characterized by slow amplitude buildup. The gammatone functions exhibit one global maximum, and the time t_p, when this occurs, varies in proportion to the value of b. For this reason, the value b can be used as a parameter for controlling the attack time in both the CODE and the RODE framework. Now, an equivalence between the analytical function of the modified gammatone and a dynamical system relies on the observation that the first derivative of $y(t)$ in Equation 4 can be expressed as

$$\dot{y} = \frac{by}{t+\epsilon} + (\sigma + j\omega)y. \tag{5}$$

Forward integration of this dynamical system, subject to a properly defined initial condition, is expected to reproduce a signal that respects the characteristics of the original gammatone function. In the form of Equation 5 one can already observe a usable mapping between the ODE parameters and the classic sound-design parameters of decay rate σ and frequency f. Also, the role of the time-dependent term $by/(t+\epsilon)$ in controlling the attack time of the synthesized signal can easily be interpreted. We illustrate this by showing that any first-order complex ODE, such as that shown in Equation 5, can be decoupled into two simpler ODEs where the frequency and the amplitude are decoupled. Keeping in mind that y is a complex variable ($y = re^{j\theta}$), Equation 5 can be decomposed into polar form as

$$\dot{r} = \left(\sigma + \frac{b}{t+\epsilon}\right) r, \tag{6}$$

$$\dot{\theta} = \omega, \tag{7}$$

where $r(t)$ is equal to $|y(t)|$ and denotes the instantaneous amplitude or the envelope of the signal, and $\dot{\theta}(t)$ is the instantaneous radial frequency. In the current form, it is assumed that b has only a real part, so it has no influence on the frequency of oscillation. Assuming that $\sigma < 0$, it becomes obvious from Equation 6 that while $\sigma + b/(t+\epsilon) > 0$ the envelope of $r(t)$ will have a positive slope, and after the time $t = t_p$ the slope of the envelope will become negative.

In the Appendix we show that given a value of σ and ϵ, a sound with user-defined attack time t_p and intensity r_p can easily be synthesized by setting the value of b to $-\sigma(t_p + \epsilon)$ and setting the initial condition so that it satisfies

$$|y(0)| = \frac{r_p}{r_p^*}, \tag{8}$$

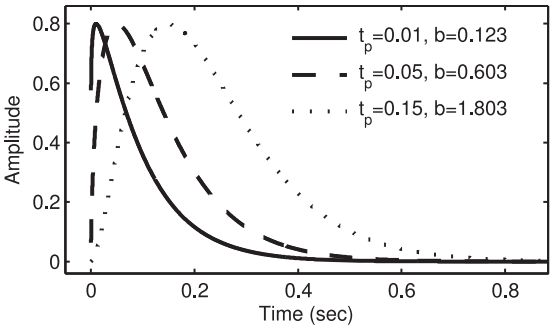


Figure 1. Synthesis result using Equation 9 with the values of b *and* r_0 *derived analytically as a function of the desired attack time* t_p *and peak amplitude* r_p. *The three envelopes reach the same peak value of 0.8 but at different attack times. Observe the variance in the initial condition* r_0 *for each curve.*

where r_p^* is an amplitude normalization constant derived from the analytical solution of the ODE. Among the many possible physical meanings of the term "intensity," we have the ability here to associate it directly with the peak value of the envelope of the synthesized signal $r_p = r(t_p)$. Equation 8 actually dictates that the phase of the initial condition is an additional free parameter that either can be defined by the user or can vary automatically according to some arbitrary criteria. For example, if the real part of the synthesized signal is used for playback, we could set the initial condition to have only an imaginary part. This can ensure that there will be no audible clicks in the rendered signal. Alternatively, the phase of the initial condition can vary according to the value of the previous sample, thereby ensuring continuity in the case of a succession of different percussive events. In Figure 1 we present an example with the RODE

$$\dot{r} = -12r + \frac{b}{t+\epsilon}r,\ r(0) = r_0, \tag{9}$$

with $\epsilon = 2.72 \cdot 10^{-4}$, the desired peak amplitudes of $r_p = 0.8$, and the desired attack times of $t_p = 0.01$, 0.05, and 0.15 seconds. In this, as well as in following examples, the ODEs are integrated forward in time using the built-in MATLAB function ode45. The value of b and the initial condition r_0 are defined analytically as a function of r_p and t_p. It can be observed that the actual attack times and peak amplitudes coincide exactly with the desired values, whereas the initial condition r_0 varies as necessary to achieve the desired amplitude and temporal characteristics of r_p and t_p.

At this stage it is worth noting additional free parameters that arise from the preceding analysis. One case is the value of ϵ, which so far has been treated as a default constant but in practice can provide interesting sound-morphing capabilities. By looking at the expression of the modified gammatone function, ϵ is a default time offset that is translated to an amplitude offset at $t = 0$. Its value can therefore be varied as a means to affect the amplitude offset at $t = 0$. Additional sound-design capabilities arise by considering b to have an imaginary part $b = b_R + jb_I$, over and above its real part. The imaginary part can be used to enforce a kind of time-dependent frequency variation in the system. The reader can verify that in this case, Equation 7 will be modified in the form $\dot{\theta} = \omega + b_I/(t+\epsilon)$, leaving unaffected, however, the expression of the amplitude in Equation 6.

Nonlinearities and Their Role in the Synthesis Process

The dynamical system presented in Equation 5 can be solved analytically, which means that we know exactly what maximum amplitude is achieved and what the attack time will be. This allows us to establish a precise one-to-one mapping between the desired temporal and amplitude characteristics of the trajectory and the ODE coefficients. Nevertheless, the sonic capabilities of this simple oscillator are relatively poor, at least outside the context of an additive-synthesis approach, where many partials would be added together to produce a sound that is more complex. This section considers the addition of nonlinear terms as a means for enriching the outcome of the ODE, while highlighting the connection to other forms of synthesis, such as nonlinear synthesis.

Dependence of Frequency and Decay on Amplitude

A very basic type of nonlinearity that can be considered is in the form $|y|^m y$, with $m \in \mathbb{N}^+$. The dynamical system of Equation 5 can thus be enriched

as

$$\dot{y} = (\sigma_0 + j2\pi f_0)y + b\frac{y}{t+\epsilon} + c|y|^m y, \tag{10}$$

providing the additional user-defined parameters $c \in \mathbb{C}$ and m. Rewriting $c = c_R + jc_I$, it can be observed that c_R and c_I will influence the dynamics of the system in completely different manners. Again, this can be better understood by decoupling the amplitude and frequency of the complex variable:

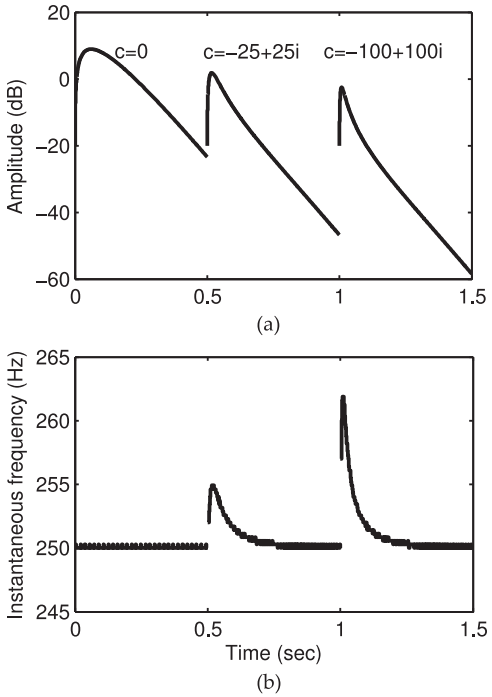
$$\dot{r} = \left(\sigma_0 + b_R\frac{1}{t+\epsilon} + c_R r^m\right) r \tag{11}$$

$$\dot{\theta} = 2\pi f_0 + c_I r^m, \tag{12}$$

where we can see that c_I influences the frequency of the system but, at least in the current context, it has no effect on the amplitude of the oscillator, whereas c_R influences both characteristics, as $\dot{\theta}$ is inevitably coupled to the envelope r. Obviously, for $f = 0$ and for real b and c, the system of Equation 10 represents a RODE. In the more general case of a CODE, the nonlinearity thus introduced makes the frequency and decay depend on the amplitude. The first effect is well known in the physics of percussion and string instruments as "tension modulation" (Tolonen, Välimäki, and Karjalainen 2000; Avanzini, Marogna, and Bank 2011); the instantaneous frequency increases instantly at the onset of the note before converging to its nominal value. Also, several commercial synthesizers are equipped with "time-varying pitch" or "pitch-bend" mechanisms, forcing the pitch to vary in time according to, for example, a user-defined envelope. On the other hand, there are also real-world examples of decay depending on amplitude. For example, Daniel Martin reports a double decay profile in piano tones (Martin 1947). Another related example from physics is the "hardening spring," where the stiffness of the spring is not constant but increases with displacement (Strogatz 1994).

The following series of simple examples shows that the simple nonlinear dynamical system of Equation 10 can produce a rich variety of decaying and sustained sounds. As a first example, consider

Figure 2. Effect of increasing the nonlinearity in Equation 13, shown in the amplitude of the signal (a) and in the frequency (b). Note that amplitude is scaled logarithmically in (a).

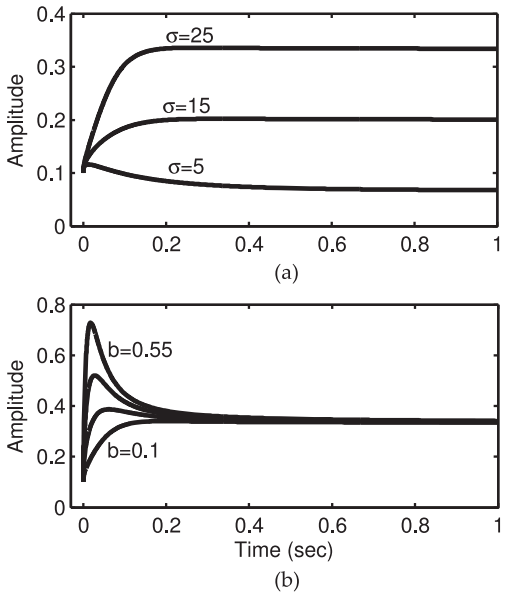


the CODE

$$\dot{y} = (-12 + j2\pi 250)y + 0.75\frac{y}{t+\epsilon} + c|y|\,y, \tag{13}$$

with initial condition $y_0 = 0.1$ and parameter c taking values from the set $\{0, -25 + 25j, -100 + 100j\}$. The characteristics of the generated signal are shown for different values of $c \in \mathbb{C}$ in terms of amplitude (in dB) in Figure 2a and in terms of frequency in Figure 2b. We deliberately plotted the amplitude variation using a logarithmic scale to show the influence of the nonlinearity. We note that exponential decay corresponds to a constant dB/sec rate, which means that, in a logarithmic scale, one should observe a straight line. This is indeed the case for all three envelopes but only after some considerable amount of time after the onset, when the influence of the time-dependent and the nonlinear term is negligible in comparison to the other term. Obviously, the influence of term $c|y|\,y$ is stronger at higher amplitudes, and it vanishes completely as the amplitude decreases, in which case the decay curve is completely governed by the value of σ. In this transition phase, the nonlinearity offers

Figure 3. Sustained oscillations corresponding to different values of σ (a) and to different values of b *(b).*



interesting possibilities for morphing the envelope; for the second and third curve one can notice a steeper decay immediately after the oscillation reaches the peak amplitude. Additional effects worth mentioning are the decrease of both the attack time and the peak amplitude r_p with decreasing c_R. In Figure 2b we used a simple zero-crossing algorithm to plot the variation of instantaneous frequency against time for the same three dynamical systems. The dependence of frequency on amplitude is obvious for the nonzero values of the parameter c.

Apart from decaying sounds, the nonlinearity has interesting potential in the design of sustained sounds as well. To see that, consider the RODE

$$\dot{r} = \sigma r + 0.05\frac{r}{t+\epsilon} - 75r^2, \tag{14}$$

with $\sigma \in \{5, 15, 25\}$ and the initial condition $r_0 = 0.1$. It is trivial to see here that, because σ takes positive values, the ODE reaches the fixed point $r = 75/\sigma$. The trajectories corresponding to the different values of σ are shown in Figure 3a. For an additional example, consider the RODE

$$\dot{r} = 15r + b\frac{r}{t+\epsilon} - 75r^2 \tag{15}$$

with $b \in \{0.1, 0.25, 0.4, 0.55\}$. The trajectories illustrated in Figure 3b indicate that, although the same fixed point is always reached, the value of b has a very visible effect on the attack region of the generated signal. In particular, increasing the value of b results in a decrease of the attack time but, simultaneously, in a higher peak value. This offers a rich palette for morphing the shape of the generated envelope, which in turn can be used for modulating a stationary deterministic or stochastic signal.

Frequency Modulation

In this section we examine nonlinearities of the form $(y + y^*)y$ and $(y - y^*)y$, where the asterisk superscript $(^*)$ denotes the complex conjugate. This provides further enrichment of the right-hand side of the oscillator in the form

$$\dot{y} = (\sigma_0 + j2\pi f_0)y + b\frac{y}{t+\epsilon} + d(y + y^*)y + e(y - y^*)y. \tag{16}$$

In general, d and e can be complex coefficients, but here we assume that they are both real. This ensures that the terms $d(y + y^*)$ and $e(y - y^*)$ are strictly real and imaginary quantities, respectively. Again, by decoupling the amplitude from the frequency we derive

$$\dot{r} = \left(\sigma_0 + b_R\frac{1}{t+\epsilon} + 2d\cos(\theta)r\right)r, \tag{17}$$

$$\dot{\theta} = 2\pi f_0 + b_I\frac{1}{t+\epsilon} + 2e\sin(\theta)r. \tag{18}$$

It can be seen that the two nonlinearities are transformed into terms of the form $d\cos(\theta)r^2$ and $e\sin(\theta)r$ in the expression of the amplitude and the frequency. In other words, they enforce a kind of frequency modulation upon the dynamical system, with a time-varying modulating frequency $\dot{\theta}$ and amplitude r. This is an extremely useful mechanism for increasing the bandwidth of the synthesis result, as well as for creating traditional effects such as tremolo and vibrato (Chowning 1973). It is worth noting an important difference between an $e(y - y^*)$ and a $d(y + y^*)$ term in the ODE: The first term affects only the oscillation frequency, meaning

that the reproduced envelope will coincide exactly with the envelope in the case of total absence of the nonlinearity. This basically means that when $e(y - y^*)y$ is the only nonlinear term, we can still control r_p and t_p according to the analysis presented earlier. On the other hand, a term of the form $d(y + y^*)y$ couples the amplitude to the frequency, and the analytical relation—which allows explicit control of attack time and peak amplitude—is not valid anymore.

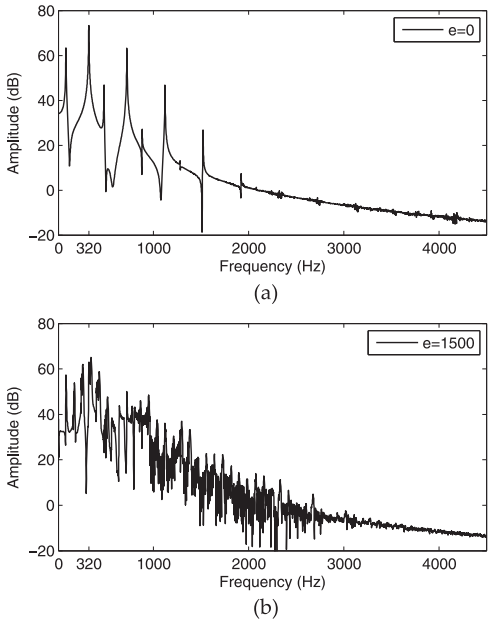
Although Equation 16 represents a case of self-modulation (i.e., the ODE modulates itself), it is obvious that the mechanism can be extended to cases of cross modulation, where two or more CODEs are nonlinearly coupled. We illustrate an example by considering a system of two CODEs bidirectionally coupled as

$$\dot{y}_1 = (-7 + j2\pi 320)y_1 + \frac{0.1}{t + \epsilon}y_1 + 1500(y_2 - y_2^*)y_1, \tag{19}$$

$$\dot{y}_2 = (0 + j2\pi 400)y_2 + e(y_1 - y_1^*)y_2. \tag{20}$$

with $e \in \mathbb{R}$ being a free parameter. The case $e = 0$ is the simplest example, in which the second CODE plays the role of the modulator and the first plays the role of the carrier. For this case, the spectral magnitude of the first CODE's output can be seen in Figure 4a. The peaks in the frequency response are in accordance with the basic theory of frequency modulation (Chowning 1973). It is worth illustrating an additional example, however, by setting $e = 1500$. In this case, the previous distinction between a modulating and a carrier CODE is less clear, because both CODEs are carriers and modulators at the same time. The spectral magnitude of the first CODE's output is shown in this case in Figure 4b, where it is obvious that a sound with a richer spectrum is produced. Of course, the example presented here can be generalized to more than two CODEs.

Figure 4. Example of frequency modulation. Spectral magnitude for the output of the CODE in Equation 19 when e = 0 *in (a) and* e = 1500 *in (b).*



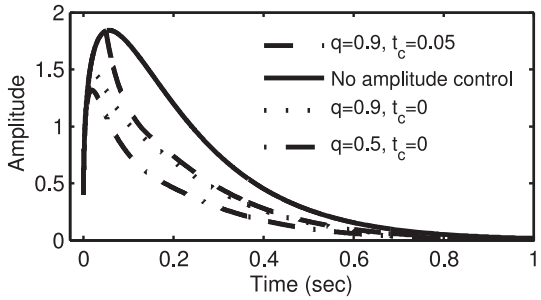
Amplitude Control

Nonlinearities connected to the amplitude of the oscillation not only make the trajectory unpredictable, but they contain the risk of making the oscillation unstable. Instability issues can be partly addressed by restricting the values of the ODE coefficients. For example, under the condition that $\sigma, c_R \leq 0$, the oscillator will be always stable as long as $d = 0$. But when the amplitude is coupled to the frequency, as for example in Equation 17, stability analysis becomes a complicated task. Inevitably, if we use a greater number of coupled ODEs, thereby increasing system complexity, either we run out of theoretical tools for such an analysis or the required mathematics start to become intractable. We believe that a more practical way to deal with problems of instability is to apply control to the dynamical system. In the general context of control theory, a controller manipulates the inputs to a system according to some reference in order to obtain a desired effect on the output of the system (Chernousko, Ananievski, and Reshmin 2008). Consequently, we implement an amplitude control mechanism to stabilize the system: We add a term to the right-hand side that acts according to a user-defined threshold $q > 0$ as

$$g(y) = p(E(y) - q + |E(y) - q|)H(t - t_c)y, \tag{21}$$

where $p \leq 0$ and $t_c \geq 0$ are user-defined parameters, $E(y)$ is a measure of the local energy of the oscillation, and $H(\cdot)$ represents the Heaviside step function. The mechanism is activated in the ODE only when both the local energy of oscillation

Figure 5. Synthesized envelope with and without amplitude control for different values of the threshold q *and the attack time* t_c *in Equation 22.*



exceeds the threshold q and the time t is greater than the attack time parameter t_c. Oscillation above the maximum power q is penalized by generation of additional damping. One possibility for $E(y)$ is, of course, the envelope of the signal, $E(y) = |y|$, but one could also use the absolute value of the real part, i.e., $E(y) = |\mathrm{Re}\{y\}|$. Not only can this mechanism prevent instability, but it can also help to avoid audio clipping by restricting the dynamic range of the synthesized signal to a range within $\pm q$. Apart from that, this approach offers sound-morphing capabilities that, from a functional point of view, are similar to classical dynamic compression in audio engineering: Parameters p, q, and t_c can be used to constrain the dynamic range, similarly to the control parameters "ratio," "threshold," and "attack" found in most dynamic compression units (Izhaki 2012).

In Figure 5 we illustrate an example with a fixed value of $p = -15$, using the RODE

$$\dot{r} = -6r + \frac{0.35}{t+\epsilon} r - 15(r - q + |r - q|)H(t - t_c)r, \tag{22}$$

with initial condition $r(0) = 0.4$ and different values for the threshold q and the attack time t_c. As can be observed, the mechanism can be used to affect both temporal and amplitude characteristics of the trajectory.

CODEs with External Input

In the examples we have seen so far, we have assumed that there is a nonzero initial condition that perturbs the dynamical system from rest to produce a signal. Indeed, the initial condition is the simplest control stream required for triggering an ODE, and it is straightfoward to associate the initial condition with the velocity message of a MIDI controller. But from another point of view, any type of audio signal, synthetic or physical, presented as external input in the ODE, can be used to cause the dynamical system to evolve in time.

Stochastic CODE

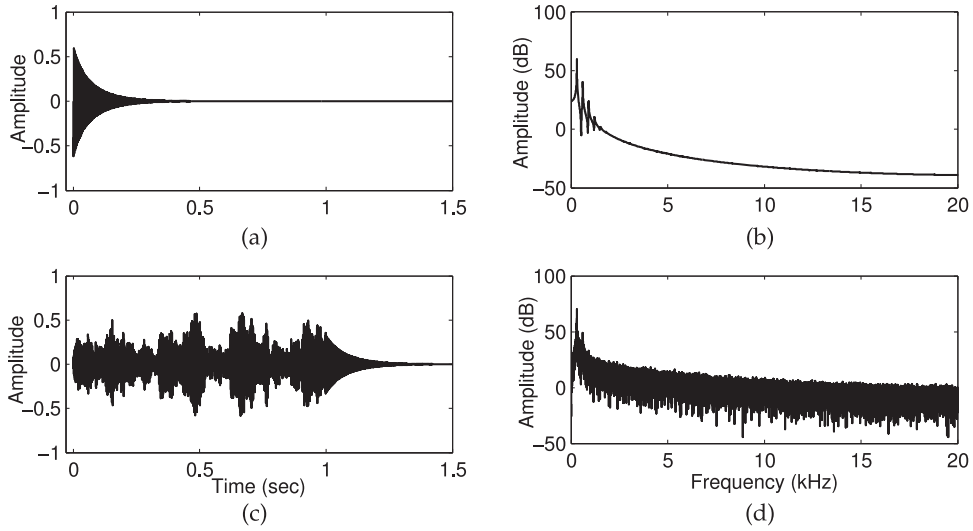
Using noise for sound synthesis purposes is definitely not a new concept. Noise is the easiest mean for synthesizing broadband signals, and subtractive synthesis is probably the most representative example (Cook 2002). From the point of view of a subtractive synthesis method, a RODE can be used to design an envelope for windowing a noise signal. Additionally, a noise signal can be presented as input to the CODE, thus turning a deterministic ODE into a stochastic one. Such an approach has implications for excitation/filter synthesis (Laroche and Meillier 1994), where the deterministic part of the CODE plays the role of the resonant filter and the input signal plays the role of the "exciter." In this context, we choose to illustrate an example by considering a nonlinear CODE without time dependency

$$\begin{aligned}\dot{y} = {} & (-11 + j2\pi 300)y + (-8 + j25)\,|y|\,y \\ & + 750(y + y^*)y + kx,\end{aligned} \tag{23}$$

where $x \in \mathbb{R}$ is a stationary, Gaussian white-noise sequence with standard deviation equal to one, and $k \in \mathbb{R}$ is the coefficient that weights the external input. To facilitate comparison, we consider a similar CODE without any external forcing ($k = 0$), excited by the initial condition $y_0 = 1i$. Figure 6 shows the signal characteristics of the time and frequency domains for the signal produced by the CODE with noise input ($k = 300$) and without noise input ($k = 0$).

What is generally observed from Figure 6c is that noise makes the system oscillate with random variations in the amplitude. Note that in this example, the noise input stops at exactly $t = 1$ sec, and after this point the system decays smoothly to silence. Remembering that white noise has a flat power spectrum, one can observe in Figure 6d

Figure 6. Example of a deterministic and a stochastic CODE corresponding to Equation 23 in the text. Time- and frequency-domain output of the CODE is shown without any stochastic excitation (a, b) and with stochastic input (c, d). The noise input stops at t = 1 *second.*



that the stochastic CODE preserves the same shape in the frequency response as the deterministic CODE (with $k = 0$), but the spectral peaks are now superimposed on an audible noise floor. Observe, however, that integration acts as a smooth filter with an attenuation of −3dB per octave, i.e., it turns white noise into pink noise. In general, the stochastic CODE acts in this case as a filter with resonant frequency and bandwidth determined by the values of f_0 and σ, respectively.

We have observed that the stochastic CODE offers an interesting potential for synthesizing the timbres of wind instruments, especially in combinations with simple types of nonlinearity, such as self-modulation of the form $(y + y^*)y$, resulting in the appearance of frequency partials that are harmonically related to the fundamental frequency. The reader is invited to listen to the sound examples (mitpressjournals.org/doi/suppl/10.1162/COMJ_a_00314/) in which stochastic CODEs are integrated into several different cases of nonlinearity.

CODE Excited by Impulsive Input

Among the most interesting types of signals presented as external input in the ODEs are impulsive sounds. A system consisting of CODEs that are excited by impulsive input is conceptually similar to filter-based modal synthesis (Cook 2002); each CODE can be seen as a resonant filter that is responsible for the generation of a single mode of vibration. The excitation signal can, of course, be processed using basic rules to model different plucking conditions and strengths.

A particularly interesting case appears when the impulsive excitation is produced naturally, by the interaction of the user with one or more physical objects. Obviously, percussionists playing their instruments are the most representative example of such interaction in music, but even simple day-to-day objects that do not belong to a particular category of musical instruments can be used as the source of impulsive excitation. In combination with classical signal-processing techniques, such as onset detection and classification, such objects can be transformed into real-time musical control interfaces to produce the control stream required for triggering a sound-synthesis engine (Stefanakis, Mastorakis, and Mouchtaris 2014). The synthesized sound can then be used to augment or to completely replace the physical sound. In the context of a system of ODEs that accepts external input, the physical sound may be directly used as the control stream required for triggering the synthesis engine.

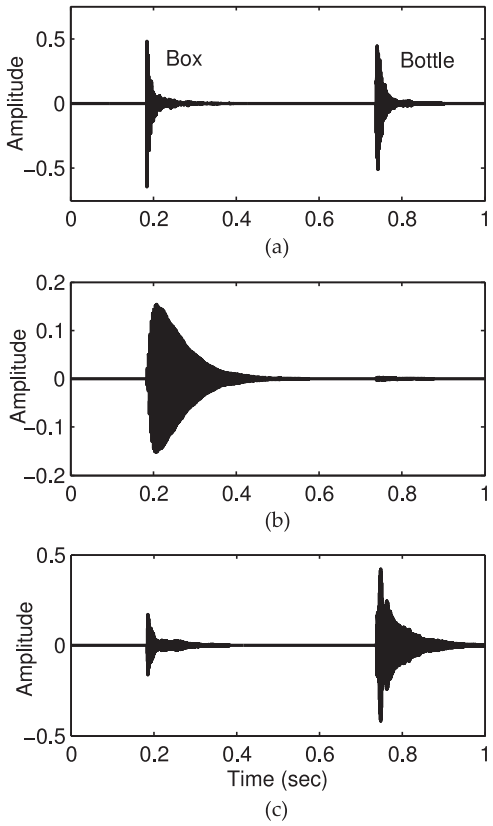


Figure 7. Example of two CODEs (Equations 24 and 25) excited by the same impulsive input. The plots show the external input signal (a), the output of the first CODE (b), and the output of the second CODE (c).

We present an example based on a recording that involves two simple objects, previously used in the work of Stefanakis, Mastorakis, and Mouchtaris (2014): a box made of recycled paper and an empty bottle made of glass. The short impulsive sound produced by striking each object was recorded with a microphone, and the recorded signal was passed as external input into a system consisting of the low-frequency and high-frequency CODEs

$$\dot{y}_1 = (-16 + j2\pi 300)y_1 + j135\,|y_1|\,y_1 + 150x, \quad (24)$$

$$\dot{y}_2 = (-18 + j2\pi 3000)y_2 + j45\,|y_2|\,y_2 + 1500x, \quad (25)$$

with $x \in \mathbb{R}$ again denoting the external input signal. Figure 7 shows time-domain plots of the excitation signal and of the real part of the two CODEs' output. Observe that the response of each CODE is significantly different in terms of amplitude. In particular, the first CODE is insensitive to bottle excitation, while the second CODE reacts more weakly to box excitation. This happens because the two objects produce inherently different sonic timbres; the energy is distributed at the lower frequencies for the box but at the higher frequencies for the bottle. It can thus be seen that there is an equivalence between sound synthesis and filtering in this example: the CODEs act as bandpass filters with central frequencies equal to their fundamental frequencies of 300 and 3000 Hz. This behavior is interesting in the sense that the spectra of the excitation signal can qualitatively affect the dynamics of the synthetic signals, which can be exploited for achieving variability in the synthesis process.

The possibility of triggering a synthesis process by using a physical signal is very interesting, but it is also worth noting that the presented CODE could also exploit information related to an onset-detection and classification algorithm. For example, onset detection could provide the necessary resetting of time required in the presence of time-dependent terms in the ODES, while a class label associated with percussionist gestures could be used to activate or deactivate a particular CODE, depending on a user-defined association between gestures and CODEs.

Synthesis Based on a System of CODEs

In this section we present a generic mathematical formulation that includes all the synthesis mechanisms presented so far, while revealing additional forms of interaction that become possible when considering multiple coupled CODEs. We begin the analysis here by rewriting the system of N CODEs shown at the beginning of the article in a more compact form:

$$\dot{\mathbf{y}} = F(\mathbf{y}, \mathbf{x}, t), \ \mathbf{y}(0) = \mathbf{y}_0, \quad (26)$$

using the vector notation $\dot{\mathbf{y}} = [\dot{y}_1,\ \dot{y}_2, \ldots, \dot{y}_N]^T$, $\mathbf{y} = [y_1,\ y_2, \ldots, y_N]^T$, $\mathbf{y}_0 = [y_{01},\ y_{02}, \ldots, y_{0N}]^T$, and $\mathbf{x} = [x_1,\ x_2, \ldots, x_N]^T$. We also need the notation

$$\mathbf{y}_E^m = [|y_1|^m, |y_2|^m, \ldots, |y_N|^m]^T \quad (27)$$

for the mechanisms of tension modulation and nonlinear decay, as well as

$$\mathbf{y}_c = [g(y_1, q_1, t_{c,1}, t), \ldots, g(y_N, q_N, t_{c,N}, t)]^T, \tag{28}$$

with

$$g(y_n, q_n, t_{c,n}, t) = (E(y_n) - q_n + |E(y_n) - q_n|)H(t - t_{c,n}), \tag{29}$$

for the mechanism of amplitude control with user-defined thresholds q_n and attack times $t_{c,n}$. Based on these mathematical structures, Equation 26 can now be written as

$$\dot{\mathbf{y}} = \left(\mathbf{A} + \frac{1}{t+\epsilon}\mathbf{B}\right)\mathbf{y} + \left(\sum_{n=1}^{N} \mathbf{C}_n \mathbf{y}_E^{m_n} + \mathbf{D}(\mathbf{y} + \mathbf{y}^*) + \mathbf{E}(\mathbf{y} - \mathbf{y}^*) + \mathbf{P}\mathbf{y}_c\right) \circ \mathbf{y} + \mathbf{K}\mathbf{x}. \tag{30}$$

Here $\circ$ denotes the Hadamard product. All bold capital letters are $N \times N$ coefficient matrices carrying the synthesis control parameters.

Based on the analysis in the previous sections, the sound-morphing potential of each term in Equation 30 can already be imagined. In the general case, the coefficient matrices are fully populated $N \times N$ complex matrices. Diagonal terms are responsible for processes that are intrinsic to each CODE, whereas nondiagonal terms in the matrices are responsible for coupling between CODEs. The diagonal terms of $\mathbf{A}$ carry the decay rate and fundamental frequency of each CODE, and the nondiagonal terms are responsible for additive coupling. Matrices $\mathbf{C}_n$, $n = 1, 2, \ldots, N$ are responsible for inserting tension modulation and nonlinear decay into the CODEs. Nondiagonal terms of $\mathbf{C}_n$ can be exploited to make the decay rate or the frequency of a particular CODE depend on the amplitude of another CODE, a scenario that has not been explicitly presented in this article but obviously has some interesting potential. Matrices $\mathbf{D}$ and $\mathbf{E}$ are intended here for frequency modulation, and we recommend that they be restricted to the real domain. The same holds for matrix $\mathbf{P}$, whose primary role is to insert amplitude control into the system. Again, nondiagonal terms could produce interesting effects, with the amplitude of one CODE controlling the amplitude of another, but further discussion of these effects is beyond the scope of this article. Also, the same equation can be enriched with power-dependent frequency-modulation mechanisms, for example, with a term of the form $j\sum_{i=1}^{I} \mathbf{E}_i \mathbf{y}_I^{p_i} \circ \mathbf{y}$, where the $\mathbf{E}_i$ are again $N \times N$ real matrices, $p_i \in \mathbb{N}^+$, and

$$\mathbf{y}_I^{p_i} = [\mathrm{Im}\{y_1\}^{p_i}, \mathrm{Im}\{y_2\}^{p_i}, \ldots, \mathrm{Im}\{y_N\}^{p_i}]^T, \tag{31}$$

with $\mathrm{Im}\{\cdot\}$ denoting the imaginary part of a complex number. This design exploits different powers of the modulator through the parameter p_i, which has a significant impact on the spectral content of the generated oscillation. The system of Equation 30 is integrated forward in time, returning the complex-valued vector $\hat{\mathbf{y}}(t) = [\hat{y}_1(t), \ldots, \hat{y}_N(t)]^T$, and the final output of the synthesis process can be obtained as a mixture of all N components, as in Equation 3.

Real-Time Capability

In the previous discussion we have shown results for a prototypical MATLAB implementation of the ODEs. Now we briefly comment on an implementation suitable for real-time integration. In addition, we tested whether the coefficients in the equations can be modified in real time, opening up the possibility of producing different sounds by a simple "tuning" of the coefficients.

For numerical integration,we used the odeint framework (Ahnert and Mulansky 2011; see also www.odeint.com), now part of the Boost template library (www.boost.org). It is extremely flexible, and it is used in many scientific and industrial projects. On the audio side, we interfaced odeint with the PortAudio library. Development was done under Linux (Ubuntu) and MacOSx using the GNU compiler g++ with C++11 support. We were able to integrate systems of up to twelve components with up to eight additive constituents on the right-hand side. Because the integrated signal can be highly nonlinear and may have very steep slopes, we used the Runge-Kutta 45 method with adaptive step-size control. On a dual-core i5 CPU, real-time

execution was feasible without problems. For the purpose of preliminary testing of the limits of the integration method, we added increasingly complex terms, such as an increasing number of polynomials of correspondingly higher order and sinusoidal external driving. Both testing scenarios eventually led to buffer underflow when the integration could not fill the PortAudio I/O buffer fast enough. We could not find any systematic behavior in this study, since the whole program ran on a Linux kernel with quite a few background processes that had higher priority than our software. For quantitative testing, one would probably need to run systematic tests keeping track of the PortAudio I/O buffer and the state of the integration. In any case, systems with few components, as in Equation 10, pose no problem for a real-time application using a high-level programming language and state-of-the-art numerical evolution. With optimization of the numerics, the performance is sure to improve.

Conclusion

Ordinary differential equations have been studied for centuries as a means to model complex dynamical processes of the real world. Nevertheless, their application to sound synthesis has not yet been fully exploited. In this article we have shown that a system of first-order complex ODEs is a powerful sound-synthesis platform for the generation of both decaying and sustained oscillations. We have introduced a simple time-dependent term for controlling the attack time of the oscillation, and we have revealed obvious nonlinear and coupling terms with which the system gains nontrivial sound-morphing capabilities—for example, pitch bend, nonlinear decay, and complex forms of frequency modulation. The ability of the dynamical system to accept external input is also an important property; stochastic or impulsive signals that are physically or synthetically produced can be used as input to the CODEs, offering additional synthesis possibilities, similar to excitation/filter and filter-based modal synthesis.

Whereas other types of dynamical systems (e.g., those based on difference equations) are known for their ability to synthesize complex sounds, a system of CODEs has the important advantage of providing parameters that are both physically and functionally meaningful for controlling the synthesis process. Finally, the framework presented here has an obvious connection to reconstruction and to analysis/synthesis approaches, where the ODE parameters can be automatically optimized so that the synthetic sounds match the timbre of natural sounds used as input in the optimization stage. The authors are currently validating this potential of CODEs in the context of percussive instruments.

References

Ahnert, K., and M. Mulansky. 2011. "Odeint: Solving Ordinary Differential Equations in C++." In *International Conference on Numerical Analysis and Applied Mathematics*, pp. 1586–1589.

Aramaki, M., et al. 2006. "A Percussive Sound Synthesizer Based on Physical and Perceptual Attributes." *Computer Music Journal* 30(2):32–41.

Argyris, J., G. Faust, and M. Haase. 1995. *An Exploration of Chaos*. Amsterdam: North-Holland.

Avanzini, F., R. Marogna, and B. Bank. 2011. "Efficient Synthesis of Tension Modulation in Strings and Membranes Based on Energy Estimation." *Journal of the Acoustical Society of America* 131(1):897–906.

Baake, E., et al. 1992. "Fitting Ordinary Differential Equations to Chaotic Data." *Physical Review A* 45(8):5524–5529.

Chernousko, F., I. Ananievski, and S. Reshmin. 2008. *Control of Nonlinear Dynamical Systems*. Berlin: Springer.

Chowning, J. 1973. "The Synthesis of Complex Audio Spectra by Means of Frequency Modulation." *Journal of the Audio Engineering Society* 21(7):526–534.

Cook, P. 2002. *Real Sound Synthesis for Interactive Applications*. Boca Raton, Florida: CRC Press.

Cooke, M. 1993. *Modelling Auditory Processing and Organisation*. Cambridge: Cambridge University Press.

Damiani, F., et al. 2000. "A Gestural Control for a Nonlinear Sound Synthesis Method." In *Proceedings of the International Conference on Devices, Circuits and Systems*, pp. S92 1–4.

Gouesbet, G., and C. Letellier. 1993. "Global Vector-Field Reconstruction by Using a Multivariate Polynomial L2 Approximation on Nets." *Physical Review E* 49(6):4955–4972.

Izhaki, R. 2012. *Mixing Audio: Concepts, Practices and Tools*. Boca Raton, Florida: CRC Press.

Kantz, H., and T. Schreiber. 1997. *Nonlinear Time Series Analysis*. Cambridge: Cambridge University Press.
Laroche, J., and J. Meillier. 1994. "Multichannel Excitation/Filter Modeling of Percussive Sounds with Application to the Piano." *IEEE Transaction on Speech and Audio Processing* 2(2):329–344.
Lindemann, E. 1988. "Routes to Chaos in a Non-Linear Musical Instrument Model." In *Proceedings of the 84th Convention of Audio Engineering Society*, Paper 2621 (pages unnumbered). Available online at www.aes.org/e-lib/browse.cfm?elib=4792 (subscription required).
Manzolli, J. 1993. "Musical Applications Derived from FracWave Sound Synthesis Method." In *Proceedings of the 94th Convention of Audio Engineering Society*, Paper 3538 (pages unnumbered).
Martin, D. W. 1947. "Decay Rates of Piano Tones." *Journal of the Acoustical Society of America* 19(5):535–541.
Packard, N. H., et al. 1980. "Geometry from a Time Series." *Physical Review Letters* 45:712–716.
Pressing, J. 1988. "Nonlinear Maps as Generators of Musical Design." *Computer Music Journal* 12(2):35–46.
Röbel, A. 2001. "Synthesizing Natural Sounds Using Dynamic Models of Sound Attractors." *Computer Music Journal* 25(2):46–61.
Rodet, X., and C. Vergez. 1999a. "Nonlinear Dynamics in Physical Models: Simple Feedback-Loop Systems and Properties." *Computer Music Journal* 23(3):18–34.
Rodet, X., and C. Vergez. 1999b. "Nonlinear Dynamics in Physical Models: From Basic Modes to True Musical Instruments." *Computer Music Journal* 23(3):35–49.
Schoner, B. 1996. "State Reconstruction for Determining Predictability in Driven Nonlinear Acoustical Systems." Diploma thesis, Rheinisch-Westphälische Technische Hochschule, Institut für Elektrische Nachrichtentechnik, Aachen, Germany.
Slater, D. 1998. "Chaotic Sound Synthesis." *Computer Music Journal* 22(2):12–19.
Stefanakis, N., Y. Mastorakis, and A. Mouchtaris. 2014. "Instantaneous Detection and Classification of Impact Sound: Turning Simple Objects into Powerful Musical Control Interfaces." In *Proceedings of the Joint International Computer Music Conference and Sound and Music Computing Conference*, pp. 1178–1184.
Strogatz, S. 1994. *Nonlinear Dynamics and Chaos: With Applications in Physics, Biology, Chemistry, and Engineering*. Boulder, Colorado: Westview.
Tolonen, T., V. Välimäki, and M. Karjalainen. 2000. "Modeling of Tension Modulation Nonlinearity in Plucked Strings." *IEEE Transactions on Speech and Audio Processing* 8:300–310.

Appendix

We will show that the parameters involved in the definition of the dynamical system in Equation 5 can be easily adjusted to respect a user-defined attack time t_p and intensity r_p, given a desired decay rate σ. Looking back at Equation 6, we can see that at $t = t_p$ the value of r must be maximum, therefore b can be found by solving for $\dot{r} = 0$,

$$b_p = -\sigma(t_p + \epsilon). \tag{A1}$$

Although b_p now explicitly defines the attack time, the amplitude level of the synthesized signal will vary strongly as a function of $y(0)$ and σ. To control the amplitude, we can derive an estimate of the maximum amplitude given an arbitrary initial condition and use this estimate to normalize the initial condition according to the desired maximum amplitude r_p. Fortunately, Equation 6 can be solved analytically, thus providing an expression of r in terms of t as

$$\ln(r) = \sigma t + b_p \ln \frac{t + \epsilon}{\epsilon} + \ln r_0, \tag{A2}$$

where $r_0 = |y(0)|$ is the value of the envelope at $t = 0$. By replacing t with t_p and r_0 with 1 in Equation A2, we can then calculate

$$ln(r_p^*) = \sigma t_p + b_p \ln \frac{t_p + \epsilon}{\epsilon}, \tag{A3}$$

where r_p^* represents the maximum instantaneous value of the amplitude, given an initial condition with amplitude equal to 1.

The amplitude of the synthesized signal can now be controlled by the used-defined parameter r_p by setting the initial condition of our ODE as

$$y(0) = \frac{r_p}{r_p^*} e^{i\phi}, \tag{A4}$$

where ϕ can have any desired value. In other words, given this initial condition and the absence of any nonlinear terms, the synthesized signal will always reach an amplitude peak at time instant t_p and the value of the envelope at t_p will be equal to r_p.