INERTIA

Universal movement simulator

Manual Revision 1.0

NEW SYSTEMS INSTRUMENTS

SPECIFICATIONS

Size	14HP
Depth	25mm
Power Consumption	+12V 72mA −12V 72mA
Timing Range	50s–7ms (low), 7Hz–50kHz (high)
Tuning Accuracy	5 octaves
Self-Oscillation Range	−6V–+6V peak
Full I/O Range	−9.5V–+9.5V peak
Input Impedance	50kΩ
Output Impedance	150Ω
Output Drive	2kΩ (min), 20kΩ+ (ideal)

INSTALLATION

Before installing the module, make sure the power is off. Attach the power cable to the module and to the bus. Double check the alignment of the red stripe (or the brown wire for a multicolor cable) with the markings on the module and the bus. The red stripe should correspond with −12V, as is standard in Eurorack. Check the documentation of your bus and power solution if you are unsure. Screw the module to the rails of the case using the provided screws. (M2.5 and M3 size screws are provided.)

New Systems Instruments modules all have keyed headers and properly wired cables. But please remember to double check the other side of the cable for proper installation with the bus. Additionally, if using a different power cable, note that not every company wires modular power cables such that the red stripe will align properly with a keyed header. While our modules are reverse polarity protected as much as is practical, it is still possible that you could damage the module, your power supply, or another module by installing the power cable improperly.

Lastly, please fully screw down the module before powering on your case. The electronics are potentially sensitive to shorts, and if the module is not properly attached to a case, there is a risk of contact with conductive or flammable matter.

OVERVIEW

Inertia produces a rate limited output signal from an input signal. In musical applications this is commonly known as *slew limiting*. However, unlike most slew limiters, Inertia can give the output signal *momentum*, a tendency to continue in the current rate and direction of motion. Momentum is ubiquitous in physical systems. It lets the finch briefly fold back its wings while bobbing upward, lets the waves on the beach slide up the shore, and keeps the planets moving around the sun.

Inertia allows you to control rising and falling rates and momenta separately, so the output can rise slower or faster than it falls, and it can have a strong tendency to continue moving in one direction, but a weak tendency (or no tendency at all) in the other direction.

While these controls are simple and intuitive, Inertia is extremely versatile. Inertia can be set up as an envelope generator, as an oscillator, as a resonant filter, as a frequency divider, as an LFO, etc.

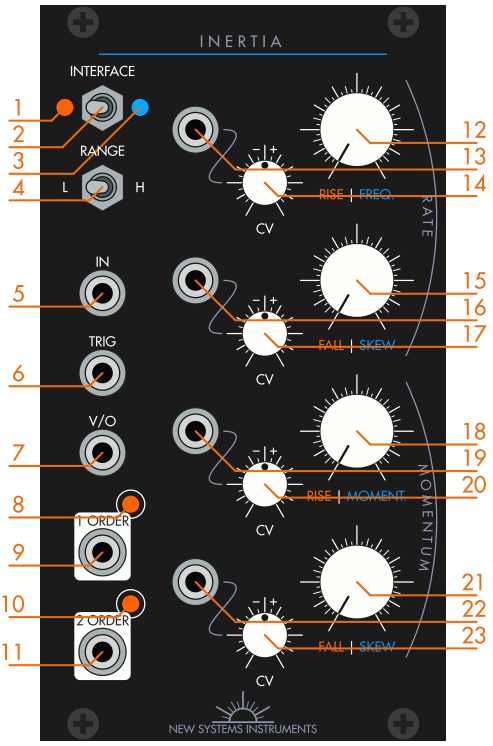
HOW TO READ THIS MANUAL

This manual is intended to be an in depth resource for continuing exploration as you continue your journey through sound and synthesis. While Inertia's controls are simple, this manual provides a deep analysis of how those controls perform in a wide variety of contexts and applications. It is not at all required for you to read the whole manual before using the module. Read according to your own learning style. I recommend reading over the Overview and Interface section, then going through Quick Start and trying out some patches. From there, browse through the rest of the sections and read what interests you. The sections on the Core of Inertia at the beginning of the manual will give you an in-depth understanding of how Inertia behaves in general, but they don't address particular applications. The sections following that are all devoted to various applications, and can be read when you need more information than is provided by the Quick Start section. If you already understand a lot of the underlying theory, you can look through the Model and Parameters sections at the end and find the equations that govern the module's behavior, but these sections assume a lot of prior knowledge.

A note on the mathematics: many sections of this manual contain mathematical equations. These equations aid understanding, but where possible the text was written in such a way that you should be able to skip past them and still understand the basics. So read through with confidence, even if you don't understand the math yet. There are a few sections specifically focused on mathematics where that is not the case, but you can safely skip over these sections, too.

I do recommend you make an effort to learn and understand the mathematics of synthesis. Just like music theory is the language of music, mathematics is the language of synthesis. You can be an excellent musician without knowing any music theory, and an excellent synthesist without knowing any mathematics. But it's very difficult to speak precisely about music without music theory, and it's very difficult to speak precisely about synthesis without mathematics. Think of it as a tool to help you learn from others and teach them in turn.

INTERFACE



1. Rise/Fall Indicator – Signals that the module is in rise/fall mode, and the left/orange text indicates the function of the controls.

2. Inerface Switch – Switches the module between rise/fall mode and skew mode. In rise/fall mode, you control rise and fall independently. In skew mode, you control the value of rise + fall with one knob, and the skew between them with the other knob.

3. Skew Indicator – Signals that the module is in skew mode, and the right/blue text indicates the function of the controls.

4. Range Switch – Switches between L (low, CV) and H (high, audio) range.

5. Input – Unless the trigger is active, output follows the input value according to rate and momentum.

6. Trigger Input – When the trigger input sees a rising edge, Inertia behaves as if the input value were at 5V, until the output value goes past 5V. *Note that the output will only exceed 5V if there is some rise momentum.*

7. Volt per Octave Input – Adjusts the rate up or down one octave for each volt of CV.

8. First Order Level – Indicates the value of the first order output, with brightness proportional to the positive value.

9. First Order Output – An output with a customary, first order exponential rate of motion. When filtering, this output gives a gentle −6 dB/oct. slope.

10. Second Order Level – Indicates the value of the second order output, with brightness proportional to the positive value.

11. Second Order Output – This output is a little smoother than the first order output. When filtering, this output gives a −12 dB/oct. slope.

12. Rise/Frequency – In rise/fall mode, controls the rate of the output when it is rising. In skew mode, controls the total rate (frequency) of the output.

13. Rise/Frequency Modulation – In rise/fall mode, CV input to control the rate of the output when it is rising. In skew mode, CV input to control the total rate (frequency) of the output.

14. Rise/Frequency Attenuverter – Attenuverter for rise/frequency CV input.

15. Fall/Skew – In rise/fall mode, controls the rate of the output when it is falling. In skew mode, controls the skew between the rise and fall rates, such that the total rate of a rise/fall cycle is preseved. Positive (right) values shorten the fall time and lengthen the rise time, whereas negative (left)

values shorten the rise time and lengthen the fall time. Tuned such that positive (right) values are frequency stable enough to be able to usefully shape the waveform without overly detuning the frequency.

16. Fall/Skew Modulation – In rise/fall mode, CV input to control the rate of the output when it is falling. In skew mode, CV input to control the skew between rise and fall rates. Positive values shorten the rise time and lengthen the fall time, whereas negative values shorten the rise time and lengthen the fall time.

17. Fall/Skew Attenuverter – Attenuverter for fall/skew CV input.

18. Rise/Momentum – In rise/fall mode, controls the amount of momentum on a rising output. In skew mode, controls the total momentum.

19. Rise/Momentum Modulation – In rise/fall mode, CV input to control the amount of momentum on a rising output. In skew mode, CV input to control the total momentum.

20. Rise/Momentum Attenuverter – Attenuverter for rise/momentum CV input.

21. Fall/Skew – In rise/fall mode, controls the amount of momentum on a falling output. In skew mode, controls the skew between rise and fall momenta. A positive (right) value lowers the rise momentum, while a negative (left) value lowers the fall momentum.

22. Fall/Skew Modulation – In rise/fall mode, CV input to control the amount of momentum on a falling output. In skew mode, CV input to control the skew between rise and fall momenta. A positive value lowers the rise momentum, while a negative value lowers the fall momentum.

23. Fall/Skew Attenuverter – Attenuverter for fall/skew CV.

QUICK START

Attack–release envelope: Set Inertia's INTERFACE to rise/fall mode (left/orange), set RANGE to L, turn FALL momentum all the way down, send a trigger or gate to the TRIG input, and take the output from the 1 ORDER output. RISE rate controls the attack of the envelope and FALL rate controls the release of the envelope, with right being faster and left slower. RISE momentum will affect the level and shape of the attack. For a less traditional but equally useful envelope, use the 2 ORDER output.

Attack–release–sustain–release envelope: Set Inertia's INTERFACE to rise/fall mode (left/orange), set RANGE to L, turn FALL momentum all the way down, send a gate to the INPUT and take the output from the 1 ORDER output. RISE rate controls the attack of the envelope and FALL rate controls the release of the envelope, with right being faster and left slower. The sustain and release levels of the envelope are set by the incoming gate signal. RISE momentum will cause the attack to rise above the sustain level, before falling back to this level at the release rate, thus allowing you to set different attack and sustain levels. Use RISE momentum CV for accents. For a less traditional but equally useful envelope, use the 2 ORDER output.

LFO: Set Inertia's INTERFACE to skew mode (right/blue), set RANGE to L, turn momentum SKEW to 12 o'clock, and turn MOMENT. all the way to the right. Take the output from 1 ORDER. FREQ. will control the frequency of the LFO, and SKEW controls whether it's skewed left or right. 2 ORDER gives a second LFO, delayed 45° in phase from the first.

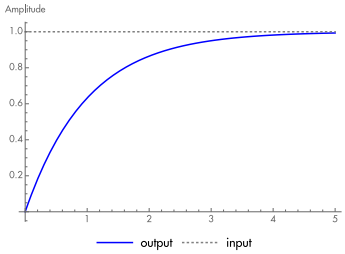
VCO: Set Inertia's INTERFACE to skew mode (right/blue), set RANGE to H, put a pitch CV into V/O, turn momentum SKEW to 12 o'clock, and turn MOMENT. all the way to the right. Take the output from either 1 ORDER or 2 ORDER. FREQ. will control the frequency of the VCO, while SKEW controls the skew. In the center, Inertia produces a sine wave, while towards the two ends Inertia produces a harmonically rich waveform, similar to a sawtooth wave.

Resonant low-pass filter: Set Inertia's INTERFACE to skew mode (right/blue), set RANGE to H, put an audio signal into INPUT, and take the output from 2 ORDER. FREQ. will control the frequency of the filter cutoff, while MOMENT. controls the resonance of the filter. Rate SKEW will set a different filter cutoff for the rising and falling parts of the waveform, which will change the waveshape of the resonance as well as change the cutoff frequency in a way that depends on the input waveshape. Momentum SKEW will limit the resonance in a way that depends on the input waveshape. With complex input signals, such as the output of the Harmonic Shift Oscillator, this waveshape-dependent filtering creates incredibly rich, evolving outputs.

THE CORE OF INERTIA: EXPONENTIAL MOTION

Inertia produces first and second order exponential motion, with controllable momentum, in response to an input signal. These opening sections on the Core of Inertia will explain what that means.

While linear motion moves equal *absolute* distances in equal times, exponential motion moves equal *proportional* distances in equal times. To choose which type of motion you want, you have to decide whether proportions of distances are important, or whether absolute distances are important. It's also important to know a little about natural phenomena and how different kinds of signals are represented in a modular system—for that see the following section. In the remainder of this section, we'll look at some important properties of exponential motion.



An Exponential Approach

Exponential motion moves from point a to point b, a total distance of d, in such a way that the amount of time it takes to move from a to $a + 1/2\ d$ is the same as the amount of time it takes to move from $a + 1/2\ d$ to $a + 3/4\ d$, which is the same amount of time it takes to move from $a + 3/4\ d$ to $a + 7/8\ d$, etc. That is, moving over each half of the remaining distance takes the same amount of time as moving over the previous half distance (and the same is true for all other proportions). Linear motion, on the other hand, would move over a distance d in such a way that any ratio of the total distance, say $1/3\ d$, would take the same amount of time to traverse, regardless of whether it happens closer to a or closer to b.

Because exponential motion covers equal proportional distances in equal times, the (linear) *rate* at which an exponential signal moves must be faster when it's further from its goal, but slower as it gets closer to its goal. If we represent the position of the moving object at time t as $y_1(t)$, the rate at which it's currently moving as $y_1'(t)$, and the place it's moving toward as $x(t)$, then we can express the relationship between rate and distance like this:

$$y_1'(t) = \omega(x(t) - y_1(t))$$

(If you're wondering why we chose y_1 instead of y, see the section on Second Order Exponential Motion below.)

All this equation says is that the rate at which an exponential signal moves is proportional to the distance between it and the place it's trying to move toward, scaled by some factor ω. With longer distances and/or a bigger ω scaling factor, the signal moves faster. With shorter distances and/or a smaller ω scaling factor, the signal moves slower.

Because it goes slower in proportion to its closeness, by the time a signal would reach $x(t)$ it would have zero motion ($x(t) - y_1(t) = 0$). Therefore, strictly speaking, *exponential motion will never reach its goal*, and $y_1(t)$ will never equal $x(t)$. But more loosely, exponential motion often gets close enough quickly enough that conventionally we will say it has "reached" its destination, even if this is not strictly true.

The philosopher Zeno of Elea, in the 5th century BCE, tried to prove that our perceptions of motion are illusory by making an argument about proportional motion, but he didn't consider the difference between linear and exponential motion. Imagine Achilles is in a race with a tortoise, which has a head start. In order for Achilles to catch up, he first has to close half the distance, during which time the tortoise moves just a little further, and Achilles once again has to close half the distance, but the tortoise has moved still further, etc. While philosophically there are some more

complex factors involved (having to do with the nature of infinity and its role in the relationship between the continuous and the discrete), generally the resolution of this paradox comes from understanding the nature of *linear* motion, which we assume is how Achilles is moving. In linear motion, it is not the *proportion* of the distance traveled which is related to time, but the absolute magnitude of this distance. And so, while Achilles always has to cover half the distance between him and the tortoise, as he gets closer to the tortoise these half distance proportions are absolutely smaller, and therefore take less time to cover. That is, *proportionally* linear motion speeds up as it approaches its goal, while *absolutely* linear motion remains at a constant speed. With exponential motion, this proportional speed up doesn't happen. Each half distance takes the same amount of time, and so Achilles never overtakes the tortoise, no matter how slowly it moves. *Proportionally* exponential motion moves at a constant rate, while *absolutely* exponential motion slows down.

This relationship between distance and rate means that exponential signals and linear signals respond very differently to different magnitudes of distance. For example, if we are dealing with a linearly moving pitch CV, the amount of time it takes to move one semitone is always the same, so moving over a major third (four semitones) takes 4 times the time it takes to move 1 semitone. In contrast, if we decide that getting within 1/32 of a semitone is close enough, then exponential motion over a major third takes only 7/5 (1.4) times as much (1/2 semitone + 1/4 + 1/8 + 1/16 + 1/32 vs. 2 semitones + 1 + 1/2 + 1/4 + 1/8 + 1/16 + 1/32). When the distances are larger, linear motion means much more time, while exponential motion means only a little more time. Of course, if all that matters is the proportional distance–that we get twice as close, for example–then no matter what the total distance, this happens in the same amount of time. But even when the absolute distance is what matters, longer distances don't take much more time with exponential motion.

Last, we should note that with exponential motion, although the rate is faster when the signal first begins to approach the input, and slower as it gets closer to it, it doesn't actually matter where the signal started, only where it is and where it is going. That is, exponential motion has *no memory*. If we begin at a certain distance, the first half distance takes a certain amount of time, and the next half distance takes the same amount of time. If we begin instead after we've traversed that half distance, again the next half distance takes the same amount of time. No matter where or when we are, exponential motion will cover that half distance in the same time as all other half distances. Momentum will change this, adding a tendency to continue in the current rate and direction of motion, which is a sort of memory.

EXPONENTIAL AND LINEAR SCALING IN PERCEPTION, PHYSICS, AND SYNTHESIS

As explained in the previous section, an exponential signal covers *equal proportions in equal time*, while a linear signal covers *equal distance in equal time*, but when do you care about equal distance and when equal proportions? While the previous section gives us abstract considerations to think about artistic intent, this section gives some concrete considerations: how perception, physical sound sources, and modular signal levels work with exponential or linear scales.

Lets begin by examining the perception of two things: pitch and loudness.

Pitch is very strongly perceived to exist on an exponential scale. That is, equal proportions between two frequencies results in the perception of equally spaced notes. For example, one frequency at 880Hz and another at 440Hz are in the ratio 880/440 = 2/1. This is an octave, and this distance sounds the same as the distance between 660Hz and 330Hz, even though the absolute (linear) distance between the one pair is 440Hz, and the absolute distance between the other pair is 330Hz.

Loudness, on the other hand, has a relatively weak exponential perception. Current research seems to indicate that we perceive loudness as a power curve $p^{0.3}$, which is not particularly exponential in

its shape. However, because our hearing is capable of precisely judging the amplitude of a huge range of signals, it is nevertheless convenient to think of sound pressure level on an exponential scale. Further, the common use of the decibel exponential scale means that there are strong cultural associations between exponential differences in loudness and "equal" loudness differences.

Perception is certainly one factor in the creation of music, but we should remember that we are not passive listeners, but actively take in what we're hearing, compare it to other natural and constructed sounds we have heard, reach conclusions, make judgements about it, experience associations with the memory of other events, and have an emotional response. Regardless of how our perceptions work, perceptions are not just images to experience but windows through which we look at the physical world.

Physical phenomena of pitch and amplitude are various, and contain both linear and exponential scaling. However, the types of phenomena that we usually look at in a musical context tend to happen on exponential scales. Anything which involves both force and momentum, such as the elasticity of a steel string, will have an amplitude that changes over time at an exponential rate. Further, anything that loses energy from fixed events over time, such as the decay of a reverberant signal in a room or the hollow body of an instrument, will have an amplitude that operates on an exponential scale. Pitch on the other hand is musically and perceptually exponential, but physically more often takes a linear scale. Uniform motion on a string, or the elongation of a resonant chamber such as in a trombone, increases pitch linearly with linear changes in distance. Players will generally compensate for this. But pitch curves resulting from the stretching of an elastic body, such as a guitar string plucked hard or a struck drum head, tend to have this extra length, and hence change in pitch, operate exponentially. Changes in timbre over time generally follow the curves of changes in amplitude. For example, in a steel string higher frequencies decay faster than lower frequencies, but both decay at an exponential rate.

In a modular system, communicating a signal always involves two moments: generating the signal in one module, and interpreting the signal in a second module. These don't necessarily have to be abstractly "matched," but if the ways in which signals are both generated and interpreted are understood properly, then it becomes possible to exercise artistic intent in matching and "mismatching" signals.

While there are other kinds of curves that can be used, in practice almost every jack interprets its input either linearly or exponentially. In particular, pitch and rate controls are most commonly interpreted using a scale that is already exponential: volt per octave. But some pitch inputs are linear: linear FM inputs, the stride input of the Harmonic Shift Oscillator, and some but not most filter inputs. Additionally, there are two common types of VCAs: exponential and linear. A linear VCA will interpret its signal without alteration, while an exponential VCA will interpret its signal using a scale that is already exponential. Two kinds of signals and two kinds of interpretation gives four possibilites: (1) *linear signals into linear inputs* give linear values, (2) *exponential signals into linear inputs* give exponential values, (3) *linear signals into exponential inputs* give exponential values, and (4) *exponential signals into exponential inputs* curve the signal twice, giving "double exponential" values.

Using linear inputs is straightforward: the curve works without alteration. Linear signals give linear curves and exponential signals give exponential curves.

Exponential inputs are generally used either for a purely musical purpose (volt per octave) or to try and give a linear modulation (such as a linear envelope) an exponential character. There are several problems with the latter approach. First, a real exponential signal approaches its target forever, getting continually closer, but a linear signal has to begin and end somewhere in a fixed amount of time. This leads to one of two problems: either the exponential curve is much too sharp, or it doesn't get close enough to its goal (VCAs that don't close). Second, linear signals with exponential inputs don't do much to model physical phenomena. In particular, real exponential upward and

downward motion generates curves that are opposites of each other, which a linear signal into an exponential input will not do. Last, the continuously oscillating form of an exponential curve (as we will see in the following sections) is a sine wave, which bears little resemblance to the shape of a triangle wave fed into an exponential input.

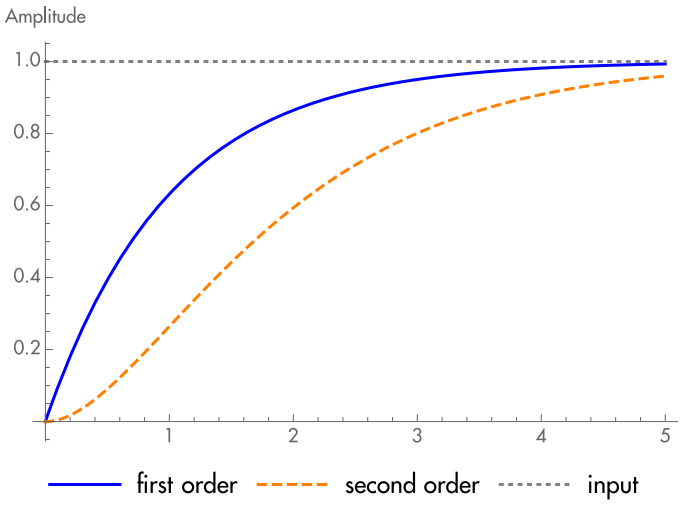
Double exponential signals are not particularly natural, but they can be musically useful—for example, to make different ranges of filter sweeps happen in relatively similar times, or to set a relatively constant portamento slide time, regardless of the value of the two pitches.

Filter sweeps are a special case. When modeling the changing timbre of a decaying signal, arguably an exponential sweep (an exponential curve into a linear input) is the correct choice. But in nature timbre does not generally change over time in the way the timbre of a filter changes. In the rare cases where actual filter sweeps are found in nature—in speech or a brass instrument with a mute, for example—the motion of the filter is usually more or less intentionally controlled, although often these phenomena are linear—for example the radius of an open mouth, or the area a mute leaves open, is correlated linearly with filter frequency. At this point in time, filter sweeps are more strongly associated with electronic instruments than anything else. Historically, almost all envelope generators have been exponential. Most filters tend to have exponential frequency inputs, leading to a characteristic double exponential filter sweep sound, but a few notable exceptions have had linear frequency inputs, giving regular single exponential filter sweeps. These different envelope shapes contribute a lot to the sound of a given filter, and this is often the reason an exact copy of a filter nevertheless doesn't "sound right." Here's a short list of the envelope curves of various well-known filters: Moog, transistor ladders, and diode ladders (double exponential); Korg MS series (double exponential); Oberheim SEM (double exponential); Roland SH-101, Juno 60, Jupiter 6, etc. (single exponential); CEM or SSM based, Prophet 5, Oberheim Matrix, etc. (double exponential); Yamaha CS series (single exponential).

THE CORE OF INERTIA: SECOND ORDER EXPONENTIAL MOTION

In Inertia, first order exponential motion follows some target input, whereas second order exponential motion follows first order exponential motion, which then follows some input. In this section we will look at second order exponential motion in its relationship to an input and to first order motion.

First, the relationship between a first order exponential and its input is the same as the relationship between first order and second order. So when first order lags behind its input a certain amount, second order will lag behind first order in the same way; while first order exponential motion limits the amplitude of higher frequencies at −6 dB/oct. (see the section on Continuous Response and Filtering below), second order exponential motion limits the amplitude of higher frequencies in first order motion by −6 dB/oct. Second order exponential motion will therefore lag the input by a larger phase difference than first order, and it will filter the input at −12dB/oct. instead of −6dB/oct.



First and second order approaches

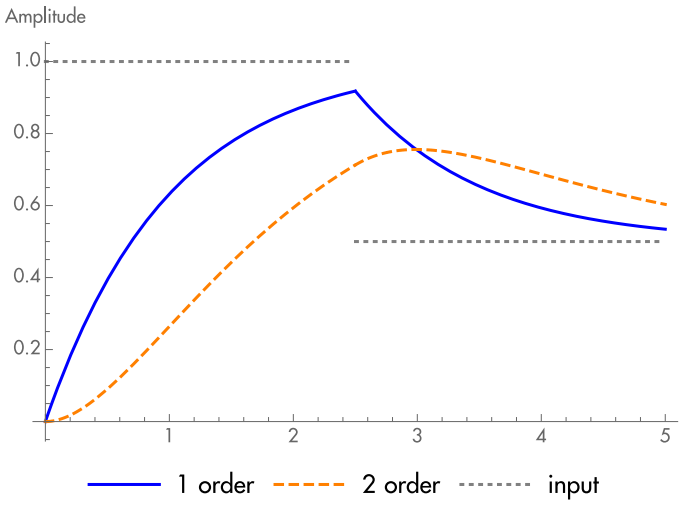
We can use the equation for the rate of first order exponential motion from the Exponential Motion section to get the relationship between first and second order motion.

$$y_2'(t) = \omega(y_1(t) - y_2(t))$$

$$y_2'(t) = \omega(x(t) - y_2(t)) - y_1'(t)$$

The first equation just reiterates what we have said above: second order exponential motion follows first order, and so the rate of second order motion is proportional to the distance between the first and second order outputs. The second equation tells us that we can think about the dependence of second on first order motion in terms of rate instead: the rate of second order motion is proportional to the distance between the input and output, and inversely proportional to the rate of first order motion. So while the rate of second order motion is generally proportional to the distance between input and output, as in first order exponential motion, second order exponential motion moves a little slower when first order is moving faster, and a little faster when first order is moving slower.

Whereas first order exponential motion is completely characterized by the input and the output (it has no memory), second order exponential motion is also dependent on the state of first order motion, which you can think about as position (the first equation above), or rate (the second equation). This extra term is a sort of memory, so if the input is suddenly moved, the first order output immediately starts moving toward that input at an exponential rate, but the behavior of the second order output depends on how close the first order output was to the input before the input changed. In particular, if the second order output is in between the first order output and the input, the two outputs will be moving in opposite directions until the second order catches up with the first order output.



First order responds immediately; second order continues upward until it reaches first order

We can also characterize second order exponential motion without reference to first order, but the equation is a bit more complicated.

$$y_2''(t) = \omega^2(x(t) - y_2(t)) - 2\omega y_2'(t)$$

In this equation, the *acceleration* $y_2''(t)$, or the *change in rate*, depends on the distance between output and input, scaled by ω^2, and on the current rate, scaled by 2ω. The dependence of the acceleration on the current rate is another way of characterizing the memory of the system.

In nature, a second order exponential output is often found in driven systems, that is, in systems where there is a constant application of force, such as a bowed string or a woodwind or brass instrument—in contrast to systems where there is a single impulse of force, like a plucked string. Note however that in musical contexts, often the application of force is deliberately controlled by the performer in order to create the desired envelope. Second order systems are also found in most acoustic filters, and (with momentum) in natural systems where harmonic (periodic) motion is involved, such as the motion of the planets or the waves of the ocean.

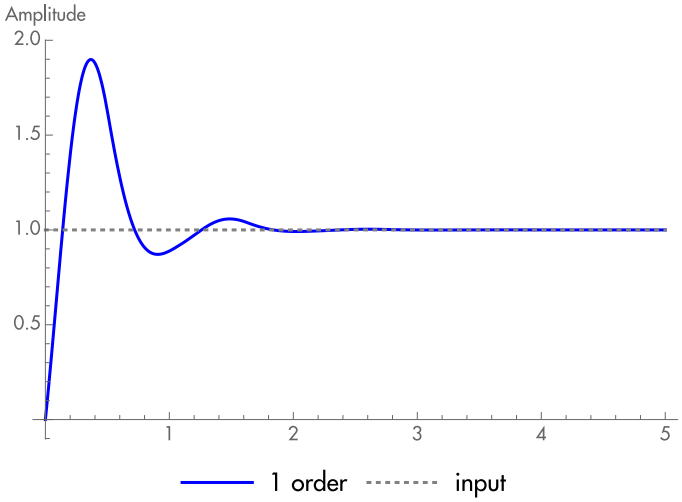
In electronic music, second order exponential motion is technically present internally in certain circuits, but only in Inertia do we have second order exponential motion fully accessible for creative use. Note in particular that the "double exponential" curve that results from plugging an exponential envelope into an exponential input is a totally different curve than that produced by a second order exponential function.

THE CORE OF INERTIA: MOMENTUM

In Inertia, momentum is the propensity for the outputs to continue moving in the same direction and rate, or alternately, the resistance to forces which would change the direction or rate of motion. With no momentum, first order exponential motion has no memory; what it will do is totally defined by the current output and the input. Momentum adds a memory, such that the output "remembers" the direction and rate at which it's going and "wants" to continue.

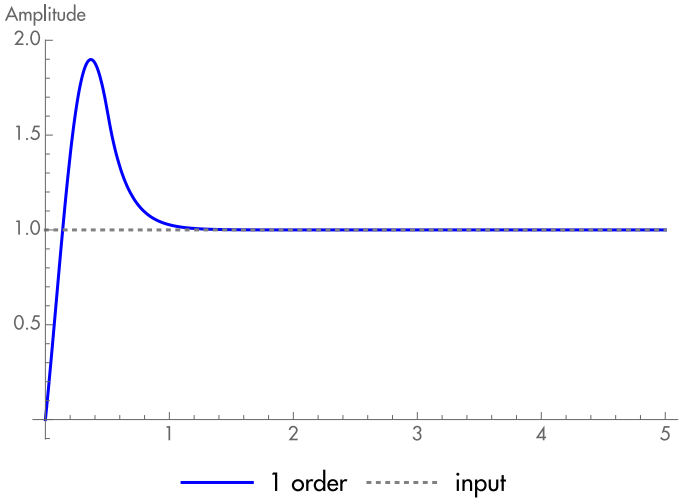
With exponential motion, the output slows down, or decelerates, just the right amount such that as the output approaches the input, the speed approaches zero, and consequently the output never reaches the input. Momentum lowers this rate of deceleration, such that when the output reaches

the input, it still has a speed. Consequently, the output keeps moving past the input, but as it passes the input, it eventually slows to zero and starts accelerating in the other direction, back toward the input it just passed. Again, when the output reaches the input, it still has a speed, continues past the input, reverses direction, etc.



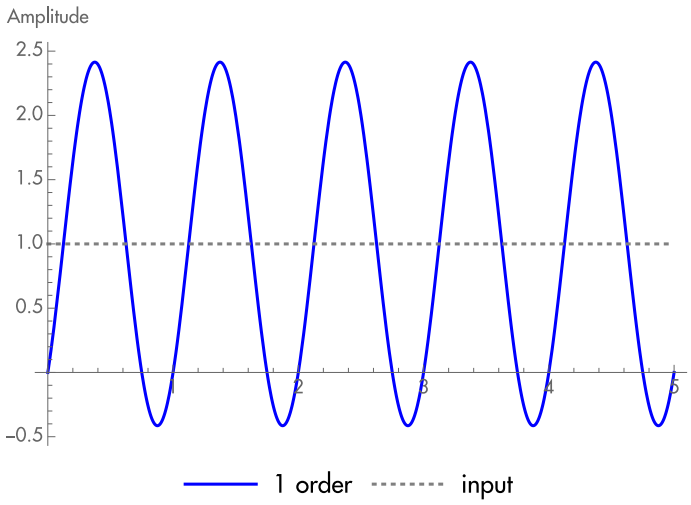
Symmetric momentum causing a ring or woggle

At control rates, this produces a "woggle," where the output settles on a value only after a time. At audio rates, this produces a ring or resonance, where the system tends to amplify and lengthen one particular frequency. However, because in Inertia you can adjust both the rate and the momentum independently, you can have the output overshoot the input in one direction, but not the other.



Momentum on rise, no momentum on fall

If we continue to add momentum, lowering further the rate of deceleration, eventually we reach a value at which the output never settles on the input, but oscillates indefinitely. In Inertia, this happens in rise/fall mode when both RISE and FALL momenta are all the way up, or in skew mode when MOMENT. is all the way up and SKEW is centered. This is a ring or a woggle that continues forever, giving you self-oscillation centered on the input value, or 0 when there is no input.



Momentum causing self-oscillation

In two dimensions, the same thing is occurring in the orbits of the planets around the sun. Gravitational attraction pulls each planet towards the sun, but the momentum of each planet causes it to overshoot and continue its course. If something removed that momentum—for example, if there was enough matter in space to cause significant friction—the planets would settle like the woggle, or in two dimensions a spiral, towards the sun at the center and destroy themselves. This relationship between earthly and celestial periodic motion was a great source of wonder before Isaac Newton proposed the mechanisms that united these two spaces. "The music of the spheres" expressed the medieval recognition of this homology. The math is the same, and if we think about music as the general science of repetition, music is more than a metaphor for what the planets do.

With momentum, the equations for the first and second order accelleration become:

$$y_1''(t) = \omega^2(x(t) - y_1(t)) - 2\omega(1-m)y_1'(t) + \omega x'(t)$$

$$y_2''(t) = \omega^2(x(t) - y_2(t)) - 2\omega(1-m)y_2'(t)$$

Where $y_{1,2}''(t)$ is acceleration/deceleration, $y_{1,2}'(t)$ is rate, $y_{1,2}(t)$ is position, $x(t)$ is input, ω is the rate scaling factor and m is the momentum, ranging from 0 to 1, where 0 is no momentum and 1 is self-oscillation.

From these equations, we might note that the rate term disappears entirely from the second order equation when momentum is at 1. Then we have a very similar equation to the equation for first order exponential motion. Whereas a rate that is proportional to the distance between input and output gives exponential motion, an accelleration that is proportional to the distance between input and output gives sinusoidal oscillation. This illustrates the deep connection between exponential motion and sine waves. Exponential motion connects time with frequency.

In nature, everything that exists has a momentum. In fact, we might define momentum in nature as the ability to interact with other objects, and so if anything had no momentum, by that very fact it could not affect the universe at all or be detected by any means. When natural objects of different momenta undergo exponential motion, they move just in the way Inertia models.

It should be noted, however, that the scale by which Inertia understands momentum and the scale by which physics understands it are different. Real exponential motion is the result of the balanced interaction of a certain amount of momentum and a certain amount of a damping force, like fric-

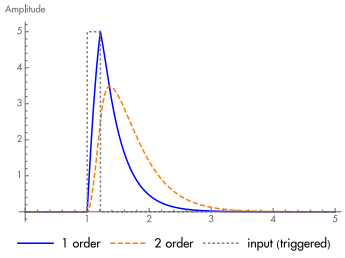
tion. Inertia models that balance as zero, and self-oscillation as 1, so zero momentum is not really the abscence of momentum, just a particular, balanced quantity of momentum.

Rarely do real objects follow perfectly exponential paths, although perfect exponentials show up in some of the math. More often, objects will follow exponential paths with a little momentum, as in Inertia's model. This allows Inertia to produce extremely natural envelopes, similar to those one often finds when physical processes, such as light filaments and vactrols, are integrated into envelope followers. At the same time, Inertia allows the modeling of all kinds of physical phenomena, and is not dependent on the physical characteristics of some particular process.

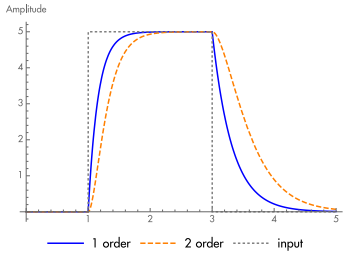
ENVELOPES AND STEP RESPONSE

When Inertia is provided with a value that suddenly shifts, it produces a curved output in response, which in synthesis is known as a function, envelope, or contour.

Usually, an envelope will be created in rise/fall mode, in order to get precise and independent control over the rise and fall times. (Although occasionally it may be important to vary the total length of the envelope independently of the ratio between the rise and fall components, in which case skew mode can be used.) An envelope is generally created by sending a gate signal to IN, to TRIG, or to both these inputs. A rising edge to TRIG will raise the input to 5V until the output exceeds 5V, at which point the trigger shuts off and the input will be whatever signal is present on IN, or 0V if no signal is present. Since a triggered envelope must rise above the same 5V value that forms its input, this requires a little rise momentum. While more complex uses are possible, TRIG can be used to create an envelope which decays immediately with no sustain (an attack–release or AR envelope), while IN will sustain for as long as the gate is held (an attack–sustain–release or ASR envelope).



An AR envelope



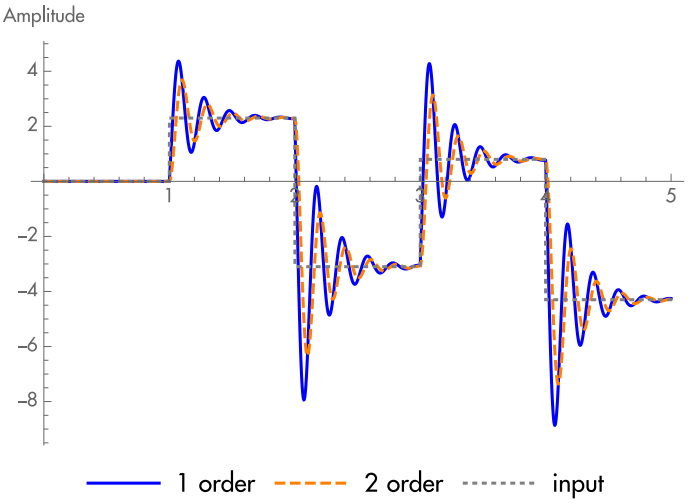
An ASR envelope

The addition of momentum has two effects on the envelope. On the one hand, momentum will cause the curve to rise above or fall below the input signal. On the other hand, momentum will cause the signal to move a little faster. Rise momentum is useful to add a little bite to an envelope's rise, producing a shape similar to a traditional ADSR envelope. Fall momentum can be useful to give more abruptness to an envelope's decay. Both rise and fall momentum together will produce a "woggle" type of effect. Adding CV to the rise momentum produces a very natural accent.

An ASR envelope with rise momentum

Using Inertia as a slew limiter for discrete steps, for example from a sample and hold circuit, behaves exactly the same as using Inertia for producing an envelope. The jumps in the input value are smoothed out and a curve results.

In step responses, Inertia's momentum controls can be used to produce effects that are generally found only on specialty single function modules. In particular, the woggle effect of combined rise and fall momentum, when paired with a random sample and hold input, produces a meandering contour that is useful for adding cv with a particularly random sound.



Woggled contour from a sample and hold

TECHNICAL DETAILS OF THE STEP RESPONSE

For a step response—where the input moves instantly from one value to another—these are the equations for when momentum is negligible:

$$y_1(t) = x(t) - (x(t) - y_1(0))e^{-\omega t}$$

$$y_2(t) = x(t) - (x(t) - y_2(0))(1 + \omega t)e^{-\omega t}$$

Where $x(t)$ is the input, $y_1(t)$ and $y_2(t)$ are the first and second order outputs, ω is the rate at which the output approaches the input, and t is time. Conventionally, step response equations are expressed such that $t = 0$ the instant just before the value of the input was changed.

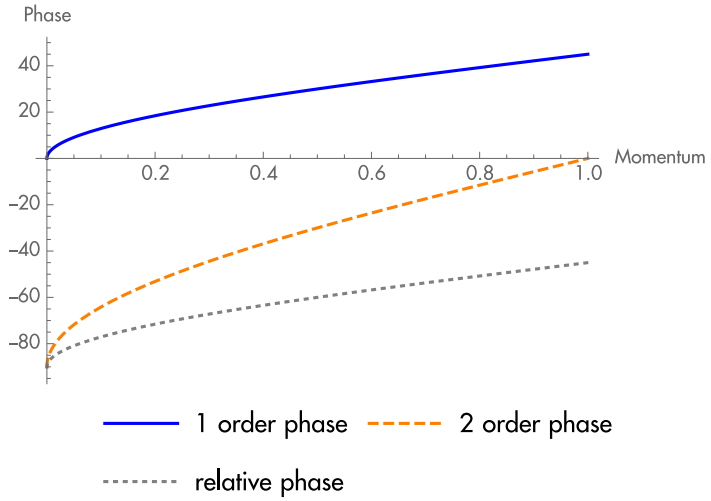
These equations help to illustrate the difference between first and second order motion. While first order is a pure exponential, $e^{-\omega t}$, second order is also scaled by a linear factor, $1 + \omega t$.

With momentum, these equations become:

$$y_1(t) = x(t) - (x(t) - y_1(0))e^{-(1-m)\omega t}\sqrt{\frac{2}{2-m}}\cos(\sqrt{1-(1-m)^2}\omega t + \arcsin(\sqrt{\frac{m}{2}}))$$

$$y_2(t) = x(t) - (x(t) - y_2(0))e^{-(1-m)\omega t}\frac{\cos(\sqrt{1-(1-m)^2}\omega t - \arcsin(1-m))}{\sqrt{1-(1-m)^2}}$$

That is, when momentum is significant, the primary differences between first and second order motion are phase and amplitude. As momentum ranges from 0 to 1, the phase of $y_1(t)$ ranges from 0 to +45 degrees, while the phase of $y_2(t)$ ranges from −90 to 0 degrees. The total phase difference between the two terms, then, will range from a 90 degree or quadrature difference with no momentum, to a 45 degree difference with momentum.



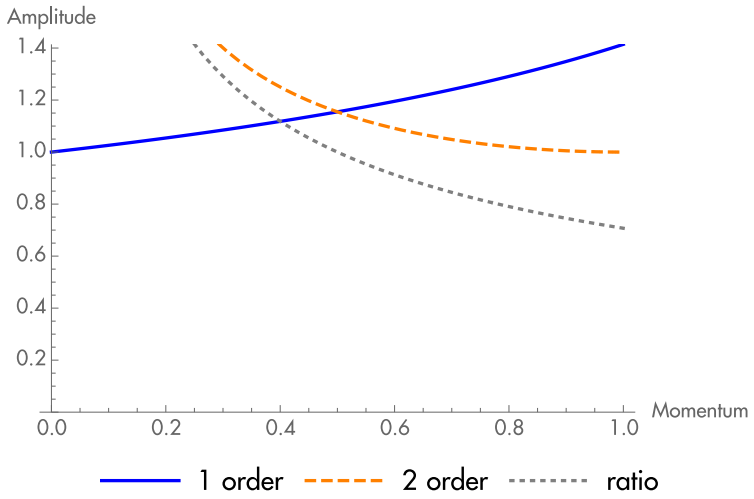
Phase and Momentum in Step Response

In the case of amplitude, we can see that the amplitude factor will range from 1 to $\sqrt{2}$ in the case of $y_1(t)$. In the case of $y_2(t)$, as the momentum approaches full, the amplitude approaches 1. However, as the denominator and the momentum approach zero, the denominator approaches zero at

the same time as the phase approaches −90°, and consequently the cosine approaches zero. At some point, then, we need a different form of the equation:

$$y_2(t) = x(t) - (x(t) - y_2(0))e^{-(1-m)\omega t}\sqrt{\frac{2}{2-m}}(\cos(\sqrt{1-(1-m)^2}\omega t + \arcsin(\sqrt{\frac{m}{2}})) + \omega t \operatorname{sinc}(\sqrt{1-(1-m)^2}\omega t))$$

We can see from this form that the difference in phase and amplitude between a first and second order step response is due to the combination of the oscillation with the linear term ωt multiplied by a sinc function. As momentum gets closer to zero, the sinc function becomes increasingly closer to unity, in which case this equation reduces to the equation for second order motion without momentum described above.



Amplitude and Momentum in Step Response

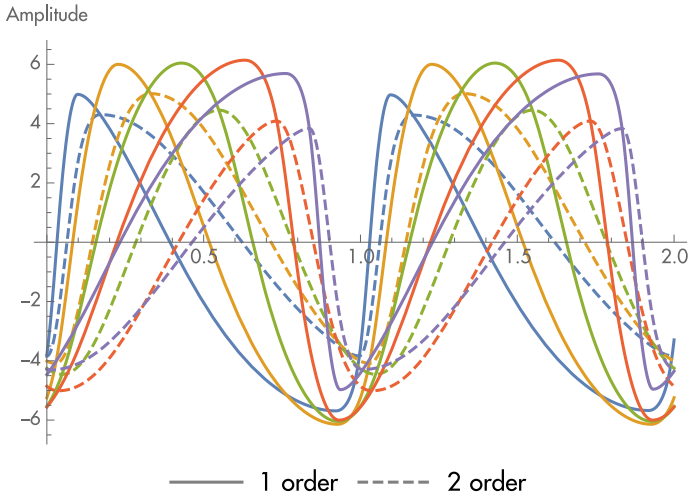
OSCILLATION

With the momentum all the way up, instead of the output gradually settling on the input value, the amplitude will increase to about ±6V in a perpetual orbit around the input value, or around 0V if there is no input. At low frequencies, this creates a periodic modulation source, or LFO. At audio frequencies, this creates a series of waveforms ranging from a pure sine wave to a very bright waveform, depending on skew.

Oscillations are generally created in skew mode so as to have separate control over the frequency and shape of the waveform. Since the momentum must be 100% on both rise and fall for stable oscillations, this means that the momentum SKEW must be zero, or vertical, while MOMENT. should be all the way up. The frequency and skew of the oscillation are then controlled by the rate FREQ. and SKEW controls, respectively. While usually it is more useful to have independent control over the frequency of an oscillation, you may sometimes wish to have independent control over rise and fall time, instead. In this case, switch to the rise/fall mode and set both RISE and FALL momenta to maximum to start oscillation.

Inertia produces oscillations as skewed sine waves, with potentially different rise and fall times. When the rise and fall times are the same, the result is a pure sine wave. As the rise and fall times become increasingly different, however, the result approaches a straight rising or falling edge, fol-

lowed by a curved falling or rising half sine wave. This is conventionally known as a sigmoid waveform. At audio rates, a sigmoid wave is similar to a sawtooth wave, but with a bit more emphasis on the low end. It could be described as warmer and less nasal than a sawtooth, while retaining the same level of brightness.



Output Waveforms

In oscillation, the second order lags behind the first order by 45 degrees. Due to the lower amplitude of second order, first and second order become equal values at the peaks and troughs of the second order waveform.

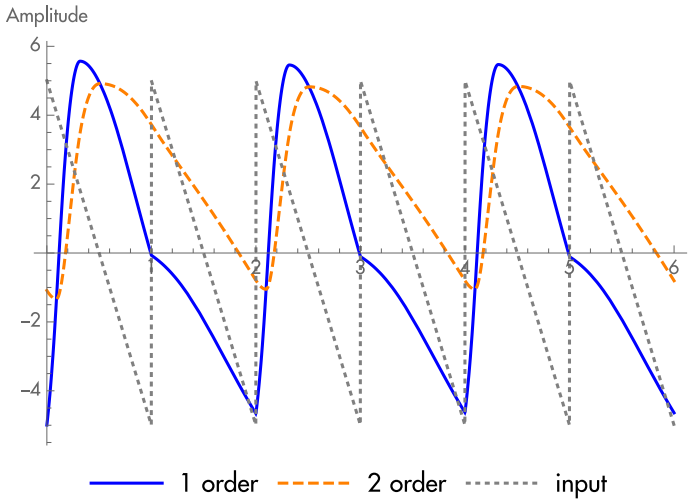
In skew mode, the frequency of oscillation is relatively pitch stable, meaning that for a given frequency setting, changing the skew with either cv or with the rate knob won't affect the frequency, and changing the frequency won't affect the skew. At audio rates, because the spectrum of the waveform is related to the wave shape, voltage contours can be used to shape the evolution over time of the spectrum produced by Inertia, thus giving an important parameter for sound design. There are two limits to pitch stability, however. Because of the physical limitations of the analog circuit, the pitch can be tuned for stability in only one direction. The direction for which Inertia is tuned is right/positive, and so anything from a sine wave to a rising sigmoid is pitch stable. Usually, the pitch is still detuned less that a semitone in the left/negative direction. Second, with CV it is possible to adjust skew beyond its maximum setting, which will detune the waveform. As with many other analog circuits, these imperfections can actually be very useful for sound design. In particular, the slight detune of the waveform along with the change of its waveshape mimics the way in which a drum head changes both pitch and timbre after it is struck.

SYNC AND FREQUENCY DIVISION

Unlike a traditional oscillator, Inertia has an input signal, and apart from Inertia's self-generated motion, a signal to either the IN or TRIG input will cause Inertia to move. This motion acts as a force to synchronize and combine Inertia's oscillation with an input signal.

To set up a frequency division patch, set up Inertia as an oscillator, and set up another oscillator (or a second Inertia) at a higher frequency. Plug the other oscillator into the IN jack. As you adjust the frequency of Inertia, it will latch on to certain ratios of the input signal. To get a more pronounced effect, turn the momentum down a little and tune the momentum SKEW to match the part of the waveform you want Inertia to latch on to. For example, a rising saw with a hard falling edge could be synced with Inertia by skewing the momentum in the right/positive direction, causing inertia to

have more falling than rising momentum. This will reduce the self-motion and make Inertia more firmly grab on to the falling edge of the saw input.



1:2 frequency division of a sawtooth wave

Synchronization occurs in Inertia because of interaction between the input and the momentum of the output. When the input is moving in the same direction as the output, the speed of the waveform is boosted. But when the input is moving in the opposite direction, the output slows down and loses amplitude. As an analogy, when pushing someone on a swing, the swing forms a pendulum and has a natural oscillating frequency with a certain momentum. You can push once a cycle, or you can push every other cycle, every third cycle, etc., but if you push at some in between rate, sometimes you'll end up pushing against the swing and slowing it down. Only when the period of the pushing is an exact multiple of the period of the swinging will oscillations build up, or conversely, only when the period of the swinging is an exact division of the period of the pushing will it oscillate.

Inertia will oscillate at frequencies that are a whole integer division of the input frequency, f, $f/2$, $f/3$ etc. This is the reverse of the conventional harmonic series. The frequencies are closer together the lower they are, culminating in an octave leap to the input frequency. In terms of conventional intervals, the inverted harmonic series gives unity (for example, C5), then $\downarrow 8^{va}$ (C4), $\downarrow 8^{va} + \mathrm{P5}$ ($\downarrow 2 \times 8^{va} \uparrow \mathrm{P4}$, F3), $\downarrow 2 \times 8^{va}$ (C3), $\downarrow 2 \times 8^{va} + \mathrm{M3}$ ($\downarrow 3 \times 8^{va} \uparrow \mathrm{m6}$, Ab2), $\downarrow 2 \times 8^{va} + \mathrm{P5}$ ($\downarrow 3 \times 8^{va} \uparrow \mathrm{P4}$, F2), $\downarrow 2 \times 8^{va} + m7$ ($\downarrow 3 \times 8^{va} \uparrow \mathrm{M2}$, D2), $\downarrow 3 \times 8^{va}$ (C2), etc. Depending on which notes are emphasized, this can sound like the notes of a suspended fourth with the input as the root, a minor chord with the input frequency on the 5th, or possibly a diminished chord located one whole step above the input. In practice, which of these are emphasized depends on the musical context, the relative frequencies of Inertia and its input, and the interaction between the wave shape of the incoming signal and Inertia's controls.

CONTINUOUS INPUTS AND FILTERING

Inertia is a type of *slew limiter*. While slew limiting is commonly thought of as a way to slow down or create smooth transitions between an input and an output signal, it has a more exact meaning. Slew limiting places constraints on the output signal such that it will only move at a certain maximum rate. Any input below this rate is reproduced unchanged except for a phase shift, as the output takes some small amount of time to catch up to the input. As described in the section on Exponential Motion above, Inertia generates exponential motion, which is the same as saying that Inertia limits the maximum *exponential* rate of motion that will be produced. So if an input signal moves

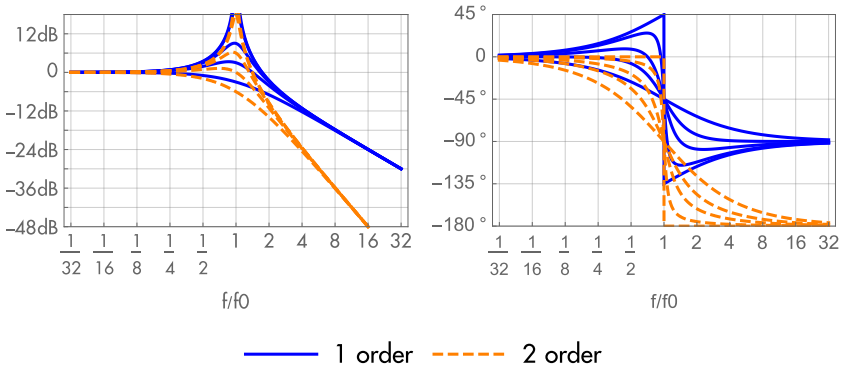
half the distance to its destination in a certain time t_1, but the rate on Inertia is set such that it can move half the distance to its destination in a time $t_2 < t_1$, then the output of Inertia will move more slowly than the input. On the other hand, if the rate on Inertia is set such that it can move half the distance to its destination in time $t_2 > t_1$, then Inertia will just reproduce the input with a little delay, but with no change to the amplitude or rate of motion.

Momentum, a tendency towards self-motion, emphasizes Inertia's maximum rate of motion. With momentum, input motion near or above the maximum rate produces an output with more amplitude than the input, such that motion near this rate "rings" and tends to continue on its own after it is given a little push. Momentum will have less effect on motion which is slower than the maximum rate, which will only have very slight boosts in amplitude.

With an LFO input, then, Inertia can be used to slow down the quicker parts of the waveform to a certain rate, while letting other parts of the waveform through unchanged, emphasizing certain frequencies, and with self-oscillation, creating frequencies. For example, the edge of a saw LFO can be slowed down, while the ramp passes through unchanged, or a time-varying LFO can be filtered such that a certain frequency is emphasized and higher frequencies are limited. The many possibilities here are beyond the scope of this document. Carefully shaping and selecting control signals can become an alternative to traditional sequencing, where the music develops on a continuum, rather than consisting of a series of discrete notes.

At audio rates, a device which allows slower signals through but blocks faster signals is a *lowpass filter*. In other words, a slew limiter is just another name for a lowpass filter. Conventionally, however, "slew limiters" are operated below audio rate, and are more often linear than exponential, although both types exist. "Lowpass filters" on the other hand are operated at audio rate, and are virtually all exponential. Momentum, which emphasizes the cutoff frequency, is more commonly known in filtering as "resonance."

The 1 ORDER output is a first order lowpass filter with a −6dB/oct. slope, while the 2 ORDER output is a second order lowpass filter with a −12 dB/oct. slope. Momentum adds resonance such that the cutoff frequency is emphasized.

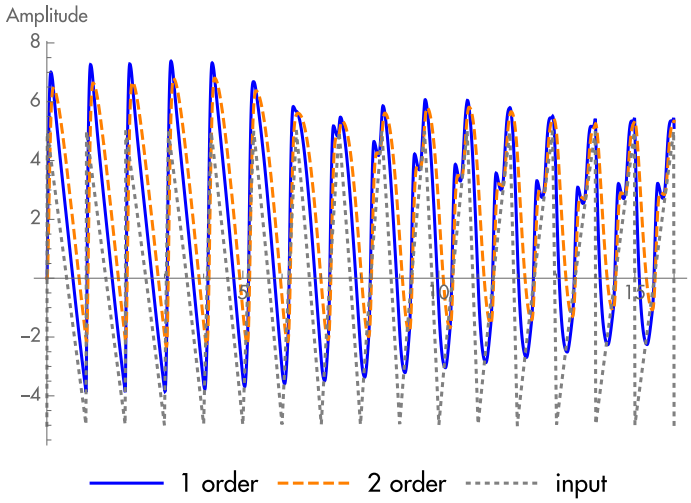


Frequency and phase response for various values of momentum

Unlike a traditional filter, Inertia has separate control over the cutoff frequency and resonance for rising and falling waveforms. With these parameters, Inertia becomes a *complex filter*, filtering a signal in a way that depends on both the frequency spectrum and the waveshape of the input signal.

To envision how this works, consider the response of Inertia to a sawtooth wave. A sawtooth wave has two pieces, one of which is extremely fast, and the other of which moves comparatively slowly. Inertia will react to each of these pieces separately, according to the frequency and resonance of

the corresponding direction. For example, for a rising saw, the slow side will be filtered by the rise section, while the fast side will be filtered by the fall section. In practice, what this means is that almost all of the filtering will be done by the fall section, and the rise section will have very little effect on the sound. If, however, we use a falling saw instead, almost all of the filtering will be done by the fall section. A triangle wave we be filtered equally by both sections. If we have a waveform which gradually changes its skew from rising to falling saw, it gradually shifts from being mostly filtered by the falling section of Inertia, to being filtered mostly by the rising section.



Waveshape-dependent filtering

Each section will filter the spectrum and create resonance, such that as a waveshape shifts, the spectrum and resonance of the filter will shift with it. For fixed waveshapes, such as the traditional subtractive synthesis waveforms, not much will change. Inertia will work as well as another filter, but its waveshape dependent filtering will result in a static sound, or a sound that responds only to direct cv control. Waveshape dependent filtering depends on interesting waveshapes, especially waveshapes that change over time, such as wavetable oscillators or the Harmonic Shift Oscillator.

MODEL

Inertia applies first and second order exponential rate limiting to an input signal, with a countervailing force of output momentum. Both of these parameters have separate values for a rising or a falling output level, giving the following parameters: input x, rate ω, comprised of rise and fall rates ω_r and ω_f, and momentum m comprised of rise and fall momenta m_r and m_f. This results in the first and second order output signals y_1 and y_2.

The y_2 output behaves like a mass–spring system, while the y_1 output behaves like the more traditional first order exponential envelope. This is reflected in the equations relating the outputs to the inputs:

$$\omega^2 y_2(t) + 2(1-m)\omega y_2'(t)/ + y_2''(t) = \omega^2 x(t)$$

$$\omega^2 y_1(t) + 2(1-m)\omega y_1'(t)/ + y_1''(t) = \omega^2 x(t) + \omega x'(t)$$

These equations give the following transfer functions:

$$\frac{Y_2(s)}{X(s)} = \frac{1}{1 + 2(1-m)\frac{s}{\omega} + \frac{s^2}{\omega^2}} = \frac{1}{(1+\frac{s}{w})^2 - 2m\frac{s}{w}}$$

$$\frac{Y_1(s)}{X(s)} = \frac{1+\frac{s}{w}}{1 + 2(1-m)\frac{s}{\omega} + \frac{s^2}{\omega^2}} = \frac{1+\frac{s}{w}}{(1+\frac{s}{w})^2 - 2m\frac{s}{w}}$$

Note that, except for the m term in the denominator, the $1 + s/w$ term in the numerator of $Y_1(s)/X(s)$ exactly cancels the square in the denominator, leaving a first order equation.

PARAMETERS

Some parameters are directly mapped to their model values. INPUT provides $x(t)$, 1 ORDER provides $y_1(t)$, and 2 ORDER provides $y_2(t)$.

In rise/fall mode:

$$\omega_r \propto 2^{V/O + 1.6(RISE + RISECV)}$$

$$\omega_f \propto 2^{V/O + 1.6(FALL + FALLCV)}$$

The base frequency with no CV and both knobs centered is around 300Hz when RANGE is set to H, and 1.5Hz when set to L.

In skew mode:

$$\omega \propto 2^{V/O + 1.6(FREQ + FREQCV)}$$

$$\omega_r = \omega + (SKEW + SKEWCV)\omega/5V$$

$$\omega_l = \omega - (SKEW + SKEWCV)\omega/5V$$

Again, the base frequencies are around 300Hz and 1.5Hz when RANGE is set to H or L, respectively.

In rise/fall mode:

$$m_r = (RISE + RISECV)/5V$$

$$m_f = (FALL + FALLCV)/5V$$

In skew mode:

$$m = (MOMENT + MOMENTCV)/5V$$

$$m_r = \begin{cases} m - (SKEW + SKEWCV)/5V & \text{if } SKEW + SKEWCV > 0 \\ m & \text{if } SKEW + SKEWCV \leq 0 \end{cases}$$

$$m_f = \begin{cases} m + (SKEW + SKEWCV)/5V & \text{if } SKEW + SKEWCV < 0 \\ m & \text{if } SKEW + SKEWCV \geq 0 \end{cases}$$