Reference Manual

STIM file format + Generator

Daniele Linaro[a,b], Jõao Couto[a,b], Michele Giugliano[a-d]

[a]Theoretical Neurobiology and Neuroengineering Laboratory,
Department of Biomedical Sciences, Univ. of Antwerp, B-2610 Wilrijk, Belgium
[b]Neuro-Electronics Research in Flanders (NERF), B-3001 Leuven, Belgium

[c]Department of Computer Science, Univ. of Sheffield, S1 4DP Sheffield, UK
[d]Brain Mind Institute, EPFL, CH-1015 Lausanne, Switzerland

Introduction

This document constitutes a reference for the STIM file format, developed by M. Giugliano, as a symbolic *meta-description* for the computer-synthesis of a class of digital waveforms, as employed in the LCG command-line software suite by Linaro, Couto, and Giugliano (2013). It also documents the standard ANSI-C library and executable of the project ***Create Stimulus***[1], which support and implement the interpreter of the meta-description, which convert it into an actual time series data. In the STIM format, a generic digital **waveform** is considered as the **juxtaposition** of a set of elementary **subwaveforms** (i.e., templates), and/or of their algebraic composition, and individual parameters and features of each subwaveform are represented as a string of characters. Such a symbolic representation consists of a piece-wise parametric decomposition of a waveform and it has the advantage of a compact description, independent of waveform length, discretization/sampling interval, and D/A quantization resolution. Other advantages are the limited disk or memory occupancy, the ease of on-the-fly editing, of automatic scripting, and of the definition by ad-hoc graphical user interfaces (i.e., waveform editors - see the MATLAB toolbox accompanying the package).

Waveforms and subwaveforms

The ultimate aim of the ANSI-C library and executable~~s~~ binaries, provided as stand-alone or embedded into LCG, is to convert the STIM meta-description written as a text file into a sequence of numeric samples (i.e., double precision floating point). This sequence is made available in the memory or on disk for subsequent digital-to-analog (D/A) conversion in a stimulus-response acquisition system, e.g., to impose a particular evolution to a time-varying physical variable, such as the current to be injected in an electronic circuit at a certain node. Such a sequence of samples is generically referred to as a **waveform**. Within the STIM format description, waveform design / storage / synthesis / etc. do not require actual data samples (e.g., one for each sampling interval). Instead, a **higher level abstract description** is employed**, specified as a string of characters**. This string fully specifies the desired waveform. Each string comes always in multiple of 12 numerical elements, each separated from each other by an arbitrary number of spaces or tab characters, and followed by a carriage return. STIM files are standard text files, whose editing can be performed very quickly through a text editor (e.g., emacs, vim, WordPad), by any user familiar with the STIM format. Each block of 12 elements is called a **subwaveform** and it refers conventionally to a time interval where some of the properties of the overall waveforms are stationary (e.g., the amplitude, the frequency of an oscillation, the standard deviation, etc.). An example of an **elementary subwaveform** is provided below, where the name of each field has been indicated explicitly (e.g., DURATION, CODE, etc.).

[**DURATION**   **CODE**   ***P₁***   ***P₂***   ***P₃***   ***P***   ***P₁***   ***FIXSEED***   ***MYSEED***   ***SUBCODE***   ***OPERATOR***   ***EXPON***]

[1] It is based on the earlier ***Generate Trial***, developed for the *MeeDuck* and *MauQuack* toolboxes, 2001-2005 authored by M. Giugliano and M. Arsiero at the University of Bern, Switzerland.
http://danielelinaro.github.io/dynclamp/
http://www.ua.ac.be/michele.giugliano

A **waveform** is always subdivided in a series of temporal intervals i=1,2,3,..., each lasting **T_i** and each isomorphic to a specific **subwaveform template**. A subwaveform may be ***elementary*** (e.g. a **DC** value lasting for **T** = 3 seconds, a **sinusoidal** oscillation lasting **T** = 5 seconds and with frequency 50Hz, a realization of a Gauss-distributed **random** signal lasting **T** = 3 seconds with mean 0 and unitary standard deviation, etc.), or it may result from the ***algebraic composition*** of elementary subwaveforms (e.g. 5 seconds of a random signal, whose mean is modulated in time as a sinusoidal oscillation).

Let us for simplicity discuss the elementary subwaveforms first. We consider a sample waveform composed of 4 subwaveforms: **each subwaveform is specified as a string of 12 numbers [. . . . .]**, so in total the waveform requires **4 x 12 numbers**: **[. . . . .] [. . . . .] [. . . . .] [. . . . .]**. The brackets have been used here only for clarity, and the carriage return separating subsequent subwaveforms has been omitted.

The first element out of each 12 numbers block is called **DURATION** and encodes the temporal duration **T** of the subwaveform. Then, it is possible to calculate the overall duration of any waveform, by simply adding together the first elements of each group of 12 numbers, e.g.:

[T_1 . . . .] [T_2 . . . .] [T_3 . . . .] [T_4 . . . .], $\mathbf{T_{total}} = \mathbf{T_1} + \mathbf{T_2} + \mathbf{T_3} + \mathbf{T_4}$.[2]

The second element, named **CODE**, is a numerical identifier to refer to an existing set of the subwaveform templates:

[T_1 code$_1$ . . . .] [T_2 code$_2$ . . . .] [T_3 code$_3$ . . . .] [T_4 code$_4$ . . . .].

The list of possible numerical values for the field **CODE**, is reported below:

- **1)** DC, constant value, subwaveform
- **2)** Ornstein-Uhlenbeck stochastic process realization subwaveform
- **3)** Sinusoidal subwaveform
- **4)** Square wave, with zero mean subwaveform
- **5)** Saw tooth wave, with zero mean subwaveform
- **6)** Sine subwaveform with frequency sweep subwaveform
- **7)** Ramp subwaveform, increasing or decreasing slope subwaveform
- **8)** Poisson / regular unipolar square pulses subwaveform
- **9)** Poisson / regular unipolar exponentially decaying pulses subwaveform
- **10)** Poisson / regular bipolar square pulses (zero mean) subwaveform
- **11)** Uniformly distributed stochastic process realization subwaveform
- **12) Alpha function subwaveform**

The extension of the library to other elementary subwaveforms is straightforward by adding more templates and using unique integer identifiers. A variety of waveforms can be made by combining arithmetically these 11 elementary types.

The remaining elements of each 12-elements block have context-dependent meanings, i.e., depending on the **CODE** and on other considerations. In the next section, we examine elementary subwaveforms, their parameters, and their encoding into the corresponding 12-elements (sub)strings.

[2] In LCG, when performing a voltage or current clamp experiment, T_{total} is used to determine the duration of the recording.

Elementary subwaveforms

The overall purpose of this meta-representation is compression: we aim at specifying/storing only the ***parameters*** necessary to identify the actual subwaveform, without indicating explicitly the entire time series, sample by sample. For instance, a sinusoidal waveform can be described by 5 numbers (i.e., its duration, amplitude, offset, frequency, and phase). While decreasing costs of storage will make disk or memory occupancy may be of limited interest, the impact of a compact description to the fast design and modification of a new waveform is obvious.

DC elementary subwaveforms

The DC elementary subwaveform is defined by just **two** parameters: its **duration** and **amplitude**, according to the following mathematical expression:

$$f(t) = \boldsymbol{P_1} \quad \text{for all times } t \in [t_0\,;\, t_0 + \boldsymbol{T}]$$

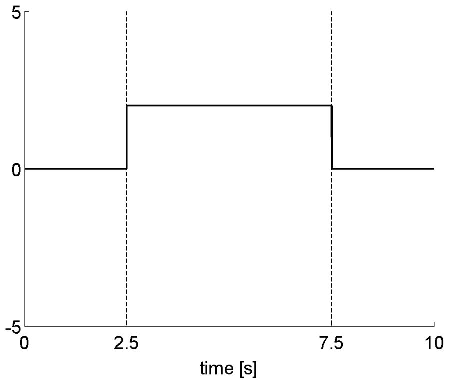
The string that identifies this DC elementary subwaveform consists in 12 distinct numerical values, but only 2 of them are relevant to parametrize the actual trace. Let's ignore ***EXPON*** for a moment, and let's indicate by **-** , the elements that are irrelevant for the identification of the DC elementary subwaveform:

[***T***	1	***P_1***	-	-	-	-	-	-	-	-	***EXPON***]

The first field contains the **duration (in seconds)** of the subwaveform, while the second field encodes its identifier (i.e., "1" corresponds to the DC type). The third field, named ***P_1*** takes the meaning of the DC subwaveform parameter, i.e., its **amplitude value (in arbitrary units)**. The remaining fields needs to be specified as they cannot be left blank, but their actual numerical value (i.e., set here to 0 by convention) is ignored.

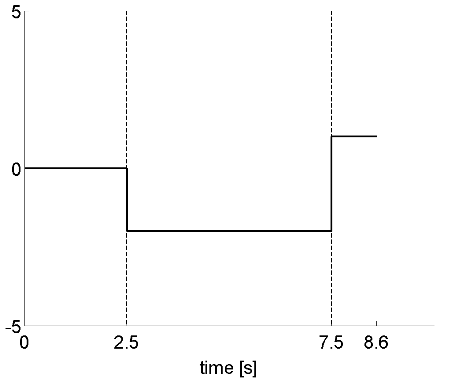
Example 1:

2.5	**1**	**0.0**	0	0	0	0	0	0	0	0	1
5.0	**1**	**2.0**	0	0	0	0	0	0	0	0	1
2.5	**1**	**0.0**	0	0	0	0	0	0	0	0	1



Example 2:

2.5	**1**	**0.0**	0	0	0	0	0	0	0	0	1
5.0	**1**	**-2.0**	0	0	0	0	0	0	0	0	1
1.1	**1**	**1.0**	0	0	0	0	0	0	0	0	1



Ornstein-Uhlenbeck elementary waveforms

The next elementary subwaveform is a realization of a stochastic process. The underlying mathematical details requires some mathematical background that is not covered here (but see the books by Cox & Miller, or by Papoulis, on random variables and stochastic processes). The process considered here is the **Ornstein-Uhlenbeck's**, sometimes referred to as **colored noise**, or as "exponentially-filtered" white Gaussian noise. A family of noisy subwaveforms realizations of this process can be iteratively synthesized considering the following stochastic differential equation:

$$df(t) = -f(t)\, dt / \mathbf{P_3} + dt\, \mathbf{P_1} / \mathbf{P_3} + \xi_t\, \mathbf{P_2} (2dt / \mathbf{P_3})^{1/2} \qquad \text{for all times } t \in [t_0\, ; t_0 + \mathbf{T}]$$

We indicated by *dt* the sampling interval (in *ms)* and by ξ_t the value returned at each call by a routine for the computer-generation of a Gauss-distributed pseudo-random numbers, with zero-mean and unitary variance. Internally, the library does not implement this specific iterative equation but an alternative one that allows for improved numerical precision. From mathematical considerations, it follows that for a certain discrete sequence of pseudo-random numbers ξ_t, f(t) is a realization of a stochastic process whose amplitude is Gauss-distributed with **mean** and **standard deviation** equal to $\boldsymbol{P_1}$ and $\boldsymbol{P_2}$, at the steady-state respectively. The spectral properties of this stochastic process are entirely specified by $\boldsymbol{P_3}$, which is the time-constant of the exponential decay of the autocorrelation function of f(t). This parameter is sometimes called **correlation length** of the process and it is always specified here in **ms**.

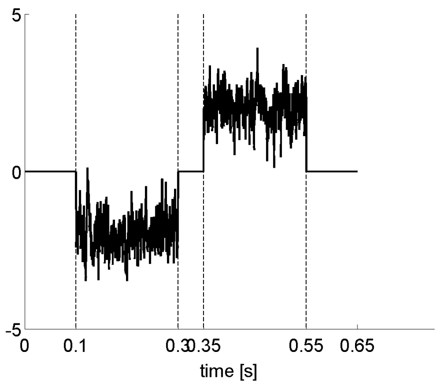
As for the DC subwaveform, from the point of view of the user, only a limited number of parameters must be specified. In this case, $\boldsymbol{P_1}$, $\boldsymbol{P_2}$, and $\boldsymbol{P_3}$ are required. The Orstein-Uhlenbeck process can be identified by the following string

[***T***	**2**	$\boldsymbol{P_1}$	$\boldsymbol{P_2}$	$\boldsymbol{P_3}$	-	-	**FIXSEED**	**MYSEED**	-	-	***EXPON***]

For this particular subwaveform, given the internal use of a random number generator, there are other two additional parameters under the user control: **FIXSEED** and **MYSEED**. When **FIXSEED** is set to 1 (i.e., its accepted values are "1" or "0", as a logical variable), the seed of the internal random number generation algorithm will be temporarily initialized to the integer specified by **MYSEED**. In order to confine the effect of **FIXSEED** only to the current subwaveforms and not the subsequents, upon completion of the subwaveform synthesis, the original seed at that point is restored. The consequences of several parameters choices are reported in the examples.

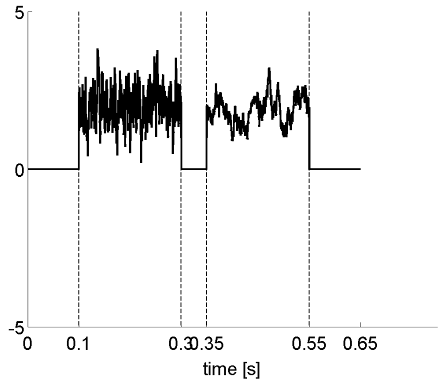
Example 3:

0.1	1	0.0	0	0	0	0	0	0	0	0	1
0.2	**2**	**-2.0**	0.5	1	0	0	0	0	0	0	1
0.05	1	0.0	0	0	0	0	0	0	0	0	1
0.2	**2**	**2.0**	0.5	1	0	0	0	0	0	0	1
0.1	1	0.0	0	0	0	0	0	0	0	0	1



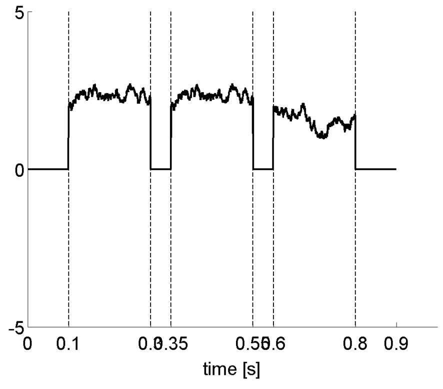
Example 4:

0.1	1	0.0	0	0	0	0	0	0	0	0	1
0.2	**2**	2.0	0.5	**1**	0	0	0	0	0	0	1
0.05	1	0.0	0	0	0	0	0	0	0	0	1
0.2	**2**	2.0	0.5	**10**	0	0	0	0	0	0	1
0.1	1	0.0	0	0	0	0	0	0	0	0	1



Example 5:

0.1	1	0.0	0	0	0	0	0	0	0	0	1
0.2	**2**	2.0	0.5	100	0	0	**1**	**21**	0	0	1
0.05	1	0.0	0	0	0	0	0	0	0	0	1
0.2	**2**	2.0	0.5	100	0	0	**1**	**21**	0	0	1
0.05	1	0.0	0	0	0	0	0	0	0	0	1
0.2	**2**	2.0	0.5	100	0	0	**0**	**0**	0	0	1
0.1	1	0.0	0	0	0	0	0	0	0	0	1



Sinusoidal elementary subwaveforms

The family of sinusoidal subwaveforms considered here, is represented by the following mathematical formula:

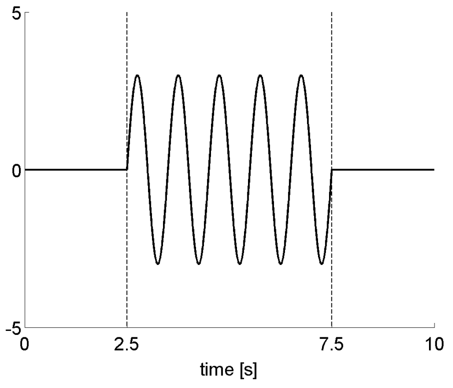
$f(t) = \mathbf{P_4} + \mathbf{P_1} \sin[2 \pi \mathbf{P_2} (t - t_0) + \mathbf{P_3}]$ for all times $t \in [t_0 ; t_0 + \mathbf{T}]$

In the previous expression, ***P₁*** represents the amplitude of the sinusoid (i.e., half of the peak-to-peak amplitude), ***P₂*** is the frequency, expressed in Hz (i.e., because *t* is supposed to be expressed in seconds), ~~and~~ ***P₃*** is the phase, expressed in radian~~t~~s, and ***P₄*** is a constant offset. As for the previous cases, ~~the~~ such a subwaveform can be expressed by the meta-parameters involved:

[***T***	3	***P₁***	***P₂***	***P₃***	***P₄***~~-~~	-	-	-	-	-	***EXPON***]

Example 6:

2.5	1	0.0	0	0	0	0	0	0	0	0	1
5.0	3	**3.0**	**1**	**0**	**0**	0	0	0	0	0	1
2.5	1	0.0	0	0	0	0	0	0	0	0	1



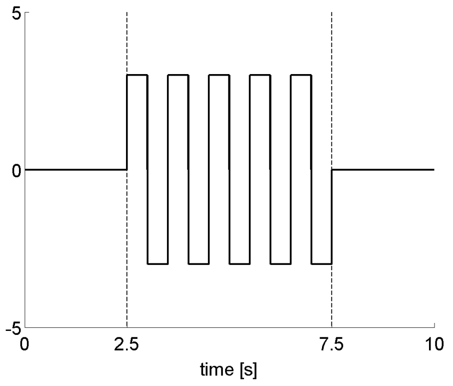
Square elementary subwaveforms

A square wave is a periodic alternation of two DC values, lasting each a given fraction of the period (i.e., the duty cycle of the square-wave). Similarly to the sinusoids, here we considered a definition of square-wave characterized by a zero mean. In addition, we assume that the duty-cycle is referred to the positive amplitude. Therefore, the parameter ***P₁*** controls the maximal absolute value of the square-wave, whose resulting peak-to-peak amplitude is **2 *P₁***. The frequency of oscillations is specified in Hz by ***P₂***, while ***P₃*** sets the duty-cycle of the positive epochs as a percentage of the period (**1/ *P₂***). The general description of these elementary subwaveform can be clarified by outlining the meta-parameters involved.

[***T***	4	***P₁***	***P₂***	***P₃***	-	-	-	-	-	-	***EXPON***]

Example 7:

2.5	1	0.0	0	0	0	0	0	0	0	0	1
5.0	4	**3.0**	**1**	**50**	0	0	0	0	0	0	1
2.5	1	0.0	0	0	0	0	0	0	0	0	1



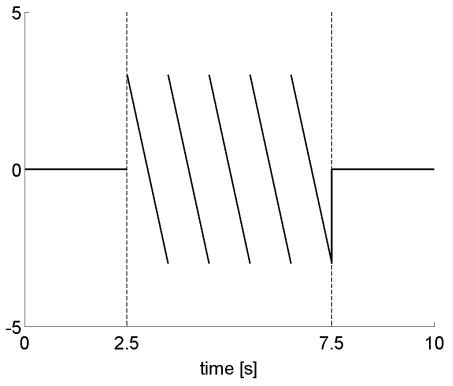
Saw-tooth elementary subwaveforms

A saw-tooth waveform is defined as the time-integral of a square wave, and it is a periodic alternation of oblique lines characterized by two slopes, lasting each a given fraction of the period (i.e., the duty cycle). Similarly to the sinusoids, here we considered a definition of sawtooth characterized by a zero mean and a duty-cycle related to the positive slope. Therefore, the parameter ***P₁*** controls the maximal absolute value, whose resulting peak-to-peak amplitude is 2 ***P₁***. The frequency of oscillations is specified in Hz by ***P₂*** while ***P₃*** sets the duty-cycle of the positive slope epochs as a percentage of the period (1/ ***P₂***). The general description of these waveform can be clarified by outlining the meta-parameters involved.

[***T***	5	***P₁***	***P₂***	***P₃***	-	-	-	-	-	-	***EXPON***]

Example 8:

2.5	1	0.0	0	0	0	0	0	0	0	0	1
5.0	5	**3.0**	**1**	**0**	0	0	0	0	0	0	1
2.5	1	0.0	0	0	0	0	0	0	0	0	1



Sine-wave elementary subwaveforms with linear frequency sweep (i.e., chirp)

A sine waveform with frequency sweep is defined as a sinusoidal function with an oscillation frequency that is linearly changing in time. It is represented by the following generic mathematical formula:

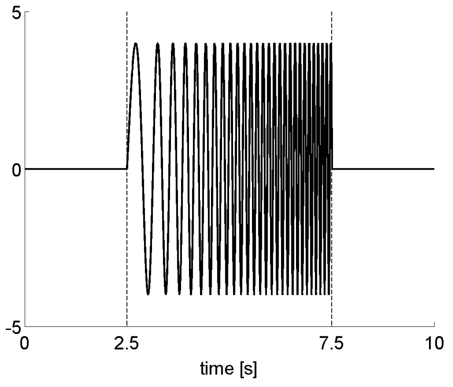
$$f(t) = \mathbf{P_1} \sin\{2\pi \quad [\mathbf{P_2} + 0.5\,(\mathbf{P_3} - \mathbf{P_2})\,(t - t_0) / \mathbf{T}] \quad (t - t_0)\} \qquad \text{for all times } t \in [t_0 ; t_0 + \mathbf{T}]$$

*$\mathbf{P_1}$ represents the **amplitude** of the sine, $\mathbf{P_2}$ the **starting** instantaneous frequency and $\mathbf{P_3}$ the **final** instantaneous frequency, both expressed in Hz. The general description of these waveform can be clarified by outlining the meta-parameters involved.*

[***T***	6	***P₁***	***P₂***	***P₃***	-	-	-	-	-	-	***EXPON***]

Example 9:

2.5	1	0.0	0	0	0	0	0	0	0	0	1
5.0	6	4.0	1	10	0	0	0	0	0	0	1
2.5	1	0.0	0	0	0	0	0	0	0	0	1



Ramp elementary subwaveforms

A ramp subwaveform is defined as a linearly increasing (or decreasing) function of time. It is represented by the following generic mathematical formula:

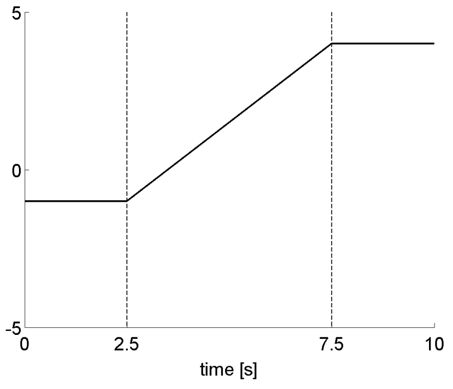
$$f(t) = f_{start} + (\mathbf{P_1} - f_{start})\ (t - t_0) / \mathbf{T} \qquad \text{for all times } t \in [t_0 ; t_0 + \mathbf{T}]$$

*$\mathbf{P_1}$ represents the final value reached by the ramp after **T** seconds, starting from f_{start}. Such a starting value is however not specified explicitly by it is set by default to 0 in case no subwaveform is preceding the ramp. Otherwise, f_{start} is set to the last numerical sample of the previous subwaveform, with the aim of guaranteeing continuity (of the first type). The general description of these subwaveform can be clarified by outlining the meta-parameters involved*

[***T***	7	***P₁***	-	-	-	-	-	-	-	-	***EXPON***]

Example 10:

2.5	1	-1.0	0	0	0	0	0	0	0	0	1
5.0	7	4.0	0	0	0	0	0	0	0	0	1
2.5	1	4.0	0	0	0	0	0	0	0	0	1



Poisson / Regular pulses elementary subwaveforms

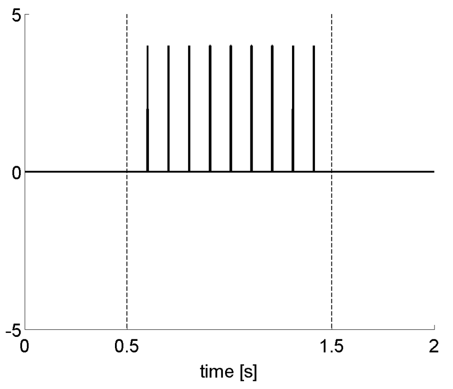
This family of elementary subwaveforms consists in a (ir)regular occurrence of pulses of amplitude ***P_1***. When the frequency of such events, specified in Hz by ***P_2***, is negative, then the intervals between successive pulses are constant and equal to 1/***P_2***. When ***P_2*** is positive, such intervals are randomly distributed according to an *exponential distribution*, so that the events will be occurring as in a (realization of a) Poisson point process. However, as opposed to a formal Poisson process, the duration of each event is non-zero and it is specified in *ms* by ***P_3***. The last determines the duration of individual pulses in the case of **unipolar** and **bipolar** pulses trains, but it takes the meaning of a time constant of the exponential decay phase, when the pulses are neither unipolar nor bipolar (see the list at the beginning of this section). The general description of these waveform can be clarified by outlining the meta-parameters involved.

[***T***	**8/9/10**	***P_1***	***P_2***	***P_3***	-	-	**FIXSEED**	**MYSEED**	-	-	***EXPON***]

By setting the field **FIXSEED** to 1 (i.e., its accepted values are “1” or “0”, as a logical variable), the seed of random number generation algorithm will be temporarily initialized to the integer specified by **MYSEED**, instead of evolving independently.

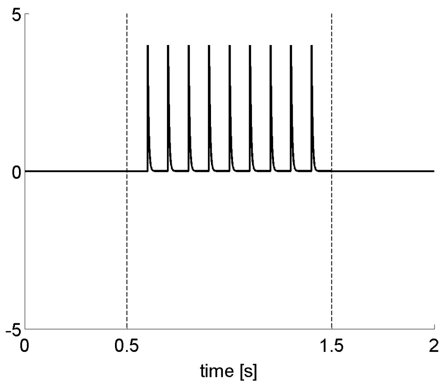
Example 11:

0.5	1	0.0	0	0	0	0	0	0	0	0	1
1.0	8	4.0	-10	1.5	0	0	0	0	0	0	1
0.5	1	0.0	0	0	0	0	0	0	0	0	1



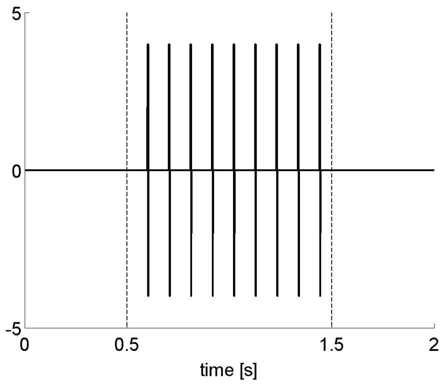
Example 12:

0.5	1	0.0	0	0	0	0	0	0	0	0	1
1.0	9	4.0	-10	5	0	0	0	0	0	0	1
0.5	1	0.0	0	0	0	0	0	0	0	0	1



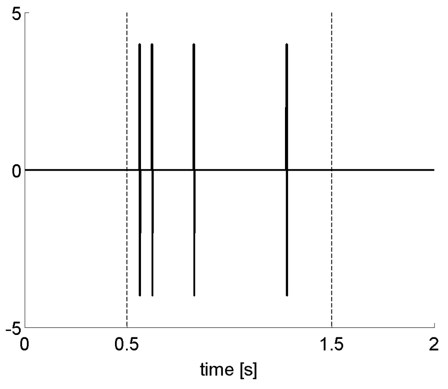
Example 13:

0.5	1	0.0	0	0	0	0	0	0	0	0	1
1.0	10	4.0	-10	5	0	0	0	0	0	0	1
0.5	1	0.0	0	0	0	0	0	0	0	0	1



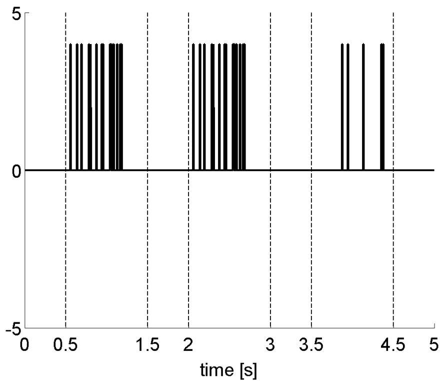
Example 14:

0.5	1	0.0	0	0	0	0	0	0	0	0	1
1.0	10	4.0	10	5	0	0	0	0	0	0	1
0.5	1	0.0	0	0	0	0	0	0	0	0	1



Example 15:

0.5	1	0.0	0	0	0	0	0	0	0	0	1
1.0	8	4.0	10	5	0	0	**1**	**43**	0	0	1
0.5	1	0.0	0	0	0	0	0	0	0	0	1
1.0	8	4.0	10	5	0	0	**1**	**43**	0	0	1
0.5	1	0.0	0	0	0	0	0	0	0	0	1
1.0	8	4.0	10	5	0	0	**0**	**0**	0	0	1
0.5	1	0.0	0	0	0	0	0	0	0	0	1



Uniformly distributed noisy elementary waveforms

This elementary subwaveform is a realization of a stochastic process, where samples are generated according to a uniform distribution, whose mean and standard deviation are specified by the user. This realization is synthesized by repeated calling to the library function *drand49()*:

$$f(t) = \boldsymbol{P_1} + \boldsymbol{P_2}\, (12)^{1/2} \,(\, \xi_t - 1/2\,) \qquad \text{for all times } t \in [t_0\, ;\, t_0 + \boldsymbol{T}]$$

We indicated by ξ_t the value returned at each call by *drand49()* that generates at each call a pseudo-random number distributed uniformly between 0 and 1 and thus mean and variance of ξ_t are 0.5 and *(1 / 12)*, respectively. f(t) is a realization of a stochastic process whose amplitude is stationary and uniformly-distributed with **mean** and **standard deviation** given by $\boldsymbol{P_1}$ and $\boldsymbol{P_2}$. The spectral properties of this stochastic process are however determined by the sampling rate.

As for the Ornstein-Uhlenbeck subwaveform, only $\boldsymbol{P_1}$, and $\boldsymbol{P_2}$ need to be specified. As for the other subwaveforms, the uniformly distributed process can be identified by the following string

[***T*** 11 $\boldsymbol{P_1}$ $\boldsymbol{P_2}$ - - - **FIXSEED** **MYSEED** - - ***EXPON***]

For this particular subwaveform, given the internal use of a random number generator, there are other two additional parameters under the user control: **FIXSEED** and **MYSEED**. When **FIXSEED** is set to 1 (i.e., its accepted values are “1” or “0”, as a logical variable), the seed of the internal random number generation algorithm will be temporarily initialized to the integer specified by **MYSEED**. In order to confine the effect of **FIXSEED** only to the current subwaveforms and not the subsequents, upon completion of the subwaveform synthesis, the original seed at that point is restored.

Alpha function elementary subwaveforms

This elementary subwaveform represents a bi-exponential function of the form:

$$f(t) = \boldsymbol{P_5} + \boldsymbol{P_1} \, [- \exp(-(t-\boldsymbol{P_4})/\boldsymbol{P_2}) + \exp(-(t-\boldsymbol{P_4})/\boldsymbol{P_3}) \,]/K \qquad \text{for all times } t \in [t_0 \,; t_0 + \boldsymbol{T}]$$

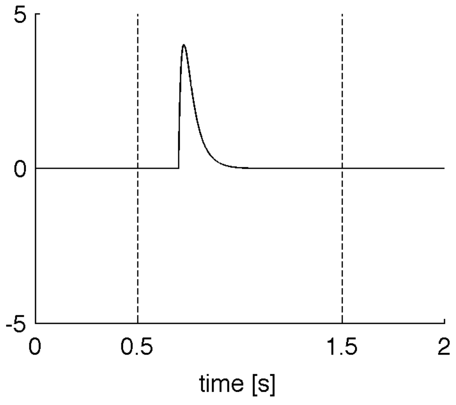
where $\boldsymbol{P_1}$ is the maximum value of the function, $\boldsymbol{P_2}$ and $\boldsymbol{P_3}$ are the rise and time constants, respectively, $\boldsymbol{P_4}$ is a delay and $\boldsymbol{P_5}$ is a constant offset. Additionally, K is a normalization factor that ensures that the maximum value assumed by the function is indeed $\boldsymbol{P_1}$.

As for the other subwaveforms, the alpha function subwaveform can be identified by the following string

[***T***	**12**	$\boldsymbol{P_1}$	$\boldsymbol{P_2}$	$\boldsymbol{P_3}$	$\boldsymbol{P_4}$	$\boldsymbol{P_5}$	-	-	-	- ***EXPON***]

Example 16:

0.5	1	0	0	0	0	0	0	0	0	0	1
1	12	4	15	50	200	0	0	0	0	0	1
0.5	1	0	0	0	0	0	0	0	0	0	1



The EXPON parameter

To increase the number of possible waveforms to be generated, each elementary subwaveform can be altered by a **real positive exponent**. In other words, when **EXPON** is different from 1, the corresponding subwaveform is modified as

$$f(t) \rightarrow f(t)^{\textbf{EXPON}} \qquad \text{for all times } t \in [t_0 \,; t_0 + \boldsymbol{T}].$$

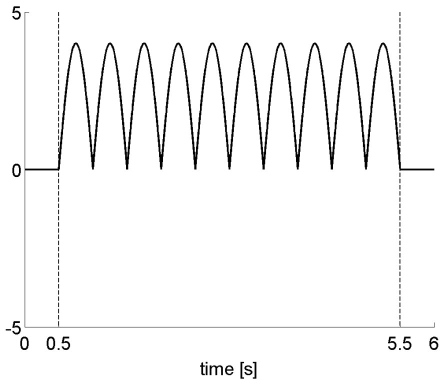
As a consequence, it is now possible to obtain the full class of polynomials in t, by composing several elementary ramp subwaveforms, and appropriately using positive integer exponents for each of them.

*When the parameter EXPON is **1**, then no alteration is performed. When EXPON is **-1**, the transformation obtained is the absolute value, point by point, of the original elementary subwaveform:*

$f(t) \rightarrow \mathbf{|}\, f(t) \,\mathbf{|}$ for all times $t \in [t_0\, ;\, t_0 + \mathbf{T}]$.

Example 16:

0.5	1	0.0	0	0	0	0	0	0	0	0	1
5.0	3	4.0	1	0	0	0	0	0	0	0	-1
0.5	1	0.0	0	0	0	0	0	0	0	0	1

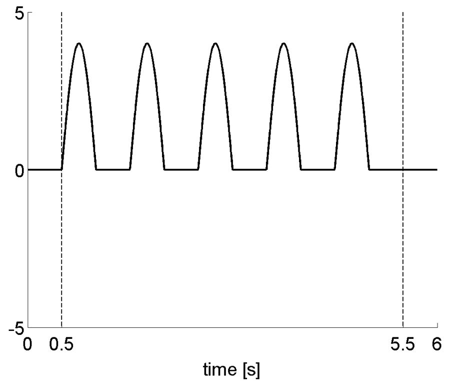


*It might at first appear counterintuitive that the last settings does not result into the inverse of the subwaveform (i.e., 1 / f(t)). As it will be introduced in the following chapter, the "/" operator can be specified when dealing with **composite** subwaveforms. Finally, when EXPON is **0**, the transformation obtained is the positive part, point by point, of the original elementary subwaveform (i.e., the identity operator where the points are positive, and zero otherwise):*

$f(t) \rightarrow \mathbf{[}\, f(t) \,\mathbf{]}^{+}$ for all times $t \in [t_0\, ;\, t_0 + \mathbf{T}]$.

Example 17:

0.5	1	0.0	0	0	0	0	0	0	0	0	1
5.0	3	4.0	1	0	0	0	0	0	0	0	0
0.5	1	0.0	0	0	0	0	0	0	0	0	1


5
0
-5
0
0.5
5.5
6
time [s]

Composite subwaveforms

A given number of elementary subwaveforms may be also combined algebraically to generate a novel subwaveform. This newly created subwaveform can be juxtaposed in a sequence as for any other elementary subwaveform. For instance, a sinusoidal waveform *with non-zero mean* or with *a time-varying amplitude* are not elementary, but they can be described in terms of a composition of elementary subwaveforms. In those examples, *summation* or *multiplication* operators are required to act on the subwaveform to produce the desired output (i.e., by a constant DC or by a ramp, respectively).

To introduce our convention, behind the algebraical combination of several elementary subwaveforms, and its order of priority, we consider an example. Let A(t), B(t), C(t) and D(t) be 4 elementary subwaveforms **of the same duration T**. A combined subwaveform is always the result of **equally long elementary subwaveforms**. A combination F(t) is then defined as

$$F = [(A \bullet B) \blacklozenge C\,] \blacksquare D$$

where the order of A, B, C and D is fixed and where the arithmetic operators indicated by ●, ◆, and ■, can be any **arithmetic operator {+, -, x, /}**. The sequence by which the operands are combined (i.e., the parentheses expressing **priority of evaluation**) is not arbitrary but fixed as shown above. The rule of combining subwaveforms each other **from left to right**, **two at the time,** stems from simplicity and efficiency as in a **single-accumulator arithmetic-logic unit**.

When a subwaveform is composite and not elementary, it requires more than 12 numbers (i.e., actually an integer multiple of 12, as if it was a juxtaposition of different elementary subwaveform). Let's indicate by **N** the **number of elementary subwaveforms to be composed**. Each of the set of 12 numbers (i.e., N x 12 in total), must share an identical **CODE**. This is however not set to a positive number (i.e., "1" for DC, "2" for the Ornstein-Uhlenbeck, "3"...), but instead it is set to **-N**. The identify of each composing elementary subwaveform is then encoded into a new field, the 10th, called **SUBCODE**. This field takes its meaning only for composite subwaveforms. Another rule is that only the first set of 12 numbers indicates the duration **T**: all the others (i.e., N - 1) needs to have the first-field set to **0**, by convention. In this way, the overall duration of a waveform, consisting of the juxtaposition of elementary and composite subwaveforms can still be inferred with the rule already specified.

By inspection of the expression reported above for F, it is possible to state that, with the exception of the very first elementary subwaveform (i.e., A, in the above example), each elementary subwaveforms can be accompanied to the preceding operator (i.e., ●, ◆, and ■). Such an information is transferred into the notation and strictly be used when the composition, two subwaveforms by two is computed. Another field of the meta-description for each of the (N - 1) sets of 12 elements, the 11th - named **PRECOP**, encodes by a number every operation as follows:

1. SUMMATION, +
2. MULTIPLICATION, x
3. SUBTRACTION, -
4. DIVISION, \

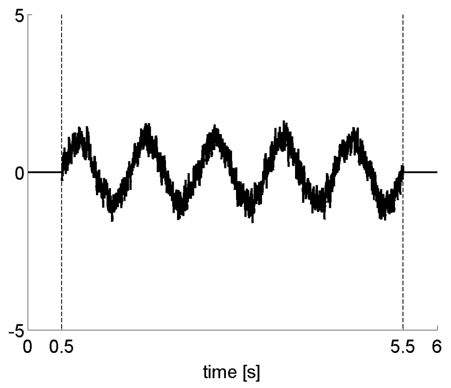
Below we indicate an example, with **N = 5** and where only part of the information has been filled in, while outlining the important parameters that control the combinations:

F = { [(A + B) x C] / D } - E for all times $t \in [t_0 ; t_0 + \boldsymbol{T}]$.

[**T**	**-5**	***P₁***	***P₂***	***P₃***	***P***	***P₁***	***FIXSEED***	***MYSEED***	***A-SUBCODE***	***-***	***EXPON***]
[**0**	**-5**	***P₁***	***P₂***	***P₃***	***P***	***P₁***	***FIXSEED***	***MYSEED***	***B -SUBCODE***	***+OPERATOR***	***EXPON***]
[**0**	**-5**	***P₁***	***P₂***	***P₃***	***P***	***P₁***	***FIXSEED***	***MYSEED***	***C -SUBCODE***	***xOPERATOR***	***EXPON***]
[**0**	**-5**	***P₁***	***P₂***	***P₃***	***P***	***P₁***	***FIXSEED***	***MYSEED***	***D -SUBCODE***	***/OPERATOR***	***EXPON***]
[**0**	**-5**	***P₁***	***P₂***	***P₃***	***P***	***P₁***	***FIXSEED***	***MYSEED***	***E -SUBCODE***	***-OPERATOR***	***EXPON***]

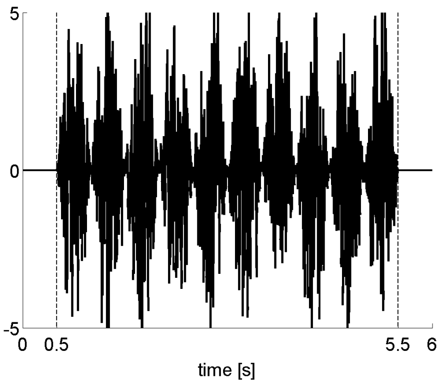
Example 18:

0.5	1	0.0	0	0	0	0	0	0	0	0	1
5.0	-2	1.0	1	0	0	0	0	0	3	0	1
0.0	-2	0.0	0.2	5	0	0	0	0	2	1	1
0.5	1	0.0	0	0	0	0	0	0	0	0	1



Example 19:

0.5	1	0.0	0	0	0	0	0	0	0	0	1
5.0	-2	1.0	1	0	0	0	0	0	3	0	1
0.0	-2	0.0	2	5	0	0	0	0	2	2	1
0.5	1	0.0	0	0	0	0	0	0	0	0	1



Stand-alone software version

A stand-alone version of “Create Stimulus” implementation based on ANSI C is provided. This is a command-line executable , to be invoked by the prompt of an operating system shell and requiring a variable number of arguments. These are reported below:

```
verb{0,1} outbinfile{0,1,-1} srate[Hz] fname1 [fname2 [fname3 [fname4 ...[fnameNchan]]]]
```

The first input argument is a logical flag (i.e., *verb*, taking values *0* or *1*) and it controls the verbosity of the output. When it is disabled (i.e., *verb* = *0*) then no diagnostic output is produced. The second argument is a flag controlling whether an output data binary-file must be generated and written on file (i.e., *outbinfile*, taking values *0, 1,* or -*1*).

Binary file-format

The file-format is binary, with the following fields:

```
sampling_rate (double)

N (unsigned long int) - number of simultaneous waves (channels)

M (unsigned long int) - number of samples per wave (channel)

sample_1, first wave

sample_2, first wave

sample_3, first wave

:

sample_M, first wave

sample_1, second wave

sample_2, second wave

sample_3, second wave

:

sample_M, second wave

:

:

sample_M, N-th wave
```