Noise

Grame

December 14, 2009

name	Noise
version	1.1
author	Grame
license	BSD
copyright	(c)GRAME 2009

1 Presentation of the "noise.dsp" Faust program

This program describes a white noise generator with an interactive volume, using a random function.

1.1 The random function

The `random` function describes a generator of random numbers, which equation follows. You should notice hereby the use of an integer arithmetic on 32 bits, relying on integer wrapping for big numbers.

1. Input signal: none
2. Output signal:
$$y(t) = r_1(t)$$
3. Internal signal:
$$r_1(t) = 12345 \oplus 1103515245 \odot r_1(t-1)$$

1.2 The noise function

The white noise then corresponds to:

1. Input signal: none
2. Output signal:
$$y(t) = 4.65661 \cdot 10^{-10} \cdot r_1(t)$$

1.3 Just add a user interface element to play volume!

Endly, the sound level of this program is controlled by a user slider, which gives the following equation:

1. Input signal: none
2. Output signal:
$$y(t) = p_1(t) \cdot r_1(t)$$
3. User interface element:
$$\text{"Volume"} : \quad u_{s1}(t) \quad \in \quad [\,0,1\,] \quad \text{(default value} = 0)$$
4. Parameter signal:
$$p_1(t) = 4.65661 \cdot 10^{-10} \cdot u_{s1}(t)$$

2 Block-diagram schema of process

This process is illustrated on figure 1.

3 Notice of this documentation

You might be careful of certain information and naming conventions used in this documentation:

- This documentation was generated with Faust version 0.9.9.6b15mdoc, on December 14, 2009.
- Eventual sub-block-diagrams may be found in the "svg" sub-directory (only top-level block-diagrams are represented in this documentation).
- Warning: symbolic names eventually used inside block-diagrams have NO direct relation with signal names used in formulas ("$x(t)$", "$y(t)$", ...). Moreover, the computation may be simplified and reorganized.
- $\forall\, s(t) \in \mathbb{S},\ s(t<0) = 0$.
- The middle dot operator "$\cdot$" denotes multiplication in formulas.
- The circled plus operator "$\oplus$" denotes an integer addition.
- The circled dot operator "$\odot$" denotes an integer multiplication.
- $y(t)$ denotes an output signal.
- $p_i(t)$ denote parameter signals (running at "block rate").
- $u_{si}(t)$ denote user interface signals of sliders.
- $r_i(t)$ denote recursive signals (delayed as $r_i(t-d)$).

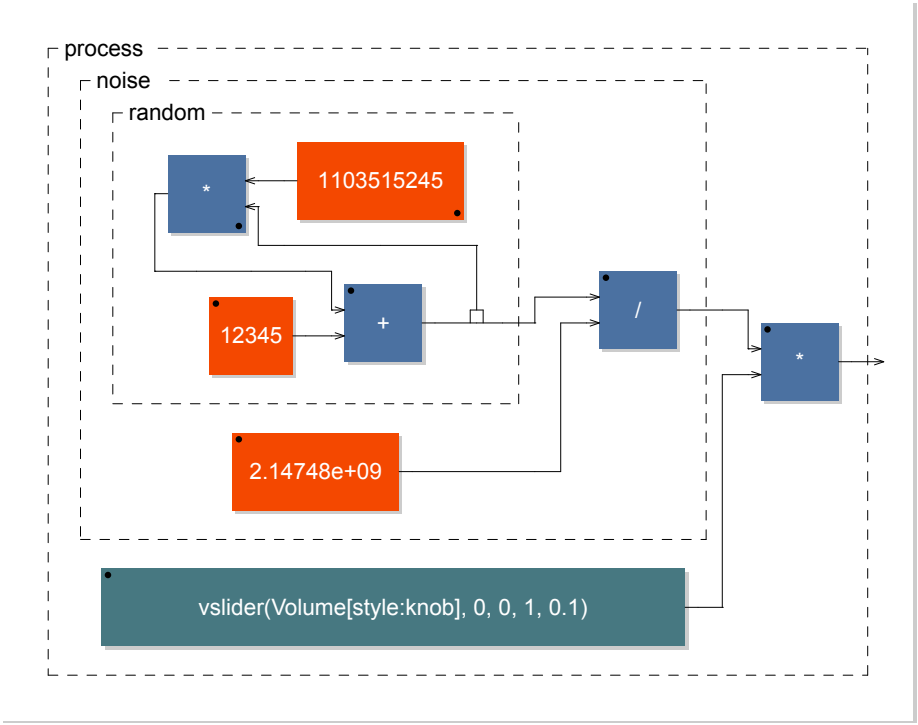

Figure 1: block-diagram of `process`

4 Listing of the input code

The following listing shows the input Faust code, parsed to compile this mathematical documentation.

Listing 1: `noise.dsp`

```
1  declare name      "Noise";
2  declare version   "1.1";
3  declare author    "Grame";
4  declare license   "BSD";
5  declare copyright  "(c)GRAME 2009";
6
7
8  //-----------------------------------------------------------------
9  // Noise generator and demo file for the Faust math documentation
10 //-----------------------------------------------------------------
11
12
13 <mdoc>
14 \section{Presentation of the "noise.dsp" Faust program}
15 This program describes a white noise generator with an interactive volume, using a random
       function.
16
```

```
\subsection{The random function}
The \texttt{random} function describes a generator of random numbers, which equation follows. You should notice hereby the use of an integer arithmetic on 32 bits, relying on integer wrapping for big numbers.
<equation>random</equation>

\subsection{The noise function}
The white noise then corresponds to:
<equation>noise</equation>
</mdoc>

random = +(12345)~*(1103515245);
noise  = random/2147483647.0;

<mdoc>
\subsection{Just add a user interface element to play volume!}
Endly, the sound level of this program is controlled by a user slider, which gives the following equation:
<equation>process</equation>
</mdoc>

<mdoc>
\section{Block-diagram schema of process}
This process is illustrated on figure 1.
<diagram>process</diagram>
</mdoc>

process = noise * vslider("Volume[style:knob]", 0, 0, 1, 0.1);

<mdoc>
\section{Notice of this documentation}
You might be careful of certain information and naming conventions used in this documentation:
<notice>

\section{Listing of the input code}
The following listing shows the input Faust code, parsed to compile this mathematical documentation.
<listing>
</mdoc>
```