Physical modelling in SystemC-WMS and real time synthesis of electric guitar effects

F. Gambini, Massimo Conti, S. Orcioni,
DEIT, Università Politecnica delle Marche, Ancona, Italy

F. Ripa, M. Caldari
Korg Italy, Osimo, Italy

max@deit.univpm.it www.deit.univpm.it

WISES 07, Madrid, June 2007

Cooperation between
Korg Italy and University of Ancona
in the design of IC
for high quality musical instruments

Outline

- Introduction
- SystemC-WMS application on sound generation and transformation
- Application to a electric guitar distortion effect generator: Fuzz-Face
- Conclusions

Sound generation and transformation physical modeling

- Many techniques used nowadays:
  - Frequency Modulation,
  - Additive or Subtractive synthesis,
  - Wavetable Synthesis based on the storage of samples of sounds of real musical instruments
- Physical modeling identifies the physical mechanism that generates the sounds, starting from the physical laws that are on the basis of the instrument.
- Physical modelling
  - Allows good sounds
  - Gives a natural and intuitive interaction between the player and sound generator
  - High computational complexity

Motivations for SystemC-AMS

- ***SystemC*** is an emerging simulation environment used to provide a virtual model of a digital IP required to verify in short time the design specifications.
- SystemC is open
- Implementation of digital and analog parts (DAC, RF front-end ...) in the same chip requires a ***mixed-signal simulation***.
- Analog and mixed-signal extensions are currently missing in SystemC

SystemC-WMS

✓Definition of a methodology for the description of analog blocks using ***instruments and libraries provided by SystemC***.

✓***No new simulation kernel*** is introduced.

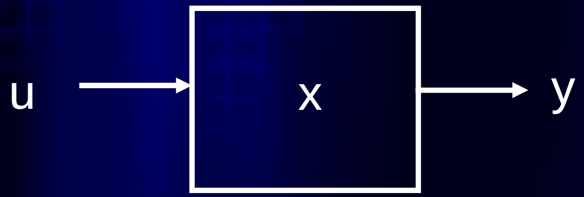
✓Therefore analog and digital modules are ***easily simulated together***.

✓analog system can be described either at a ***low or a high level*** using analog macromodels.

✓We think that a high level macromodel should be used in a system level simulation, using a SPICE-like simulator only when a detailed analysis is required.

The Problem

Implementation in SystemC of continuous time systems described by a signal flow graph (SFG) and by a system of non-linear ordinary differential equations of the type



$$\begin{cases} \dot{x} = f(x,u) \\ y = g(x,u) \end{cases}$$

x : state, y: output, u: input vector

Detailed description or macromodel of the analog circuit

Solution is obtained using the explicit Euler formula

$$\begin{cases} x(t+dt) = x(t) + dt \cdot f(x(t),u(t)) \\ y(t) = g(x(t),u(t)) \end{cases}$$

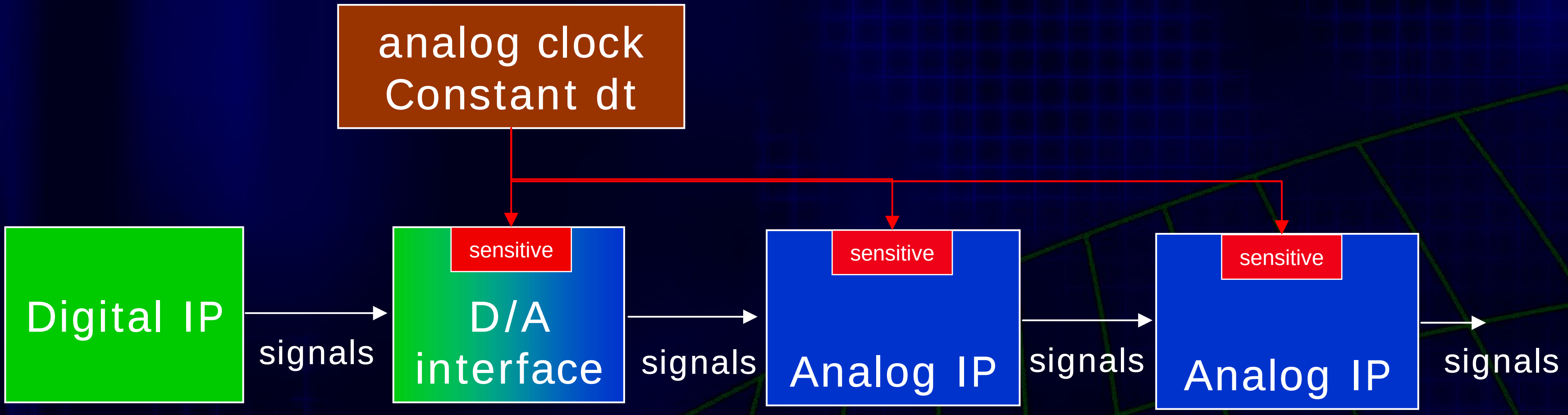
Methodology

Different possible time step implementations in SystemC:

***Global constant time step for all the analog processes*.**

(synchronous, dt constant)

This “analog simulation clock” activates all the analog blocks.
This solution is the simplest, but very inefficient since it requires a very small time step to reach an accurate solution.

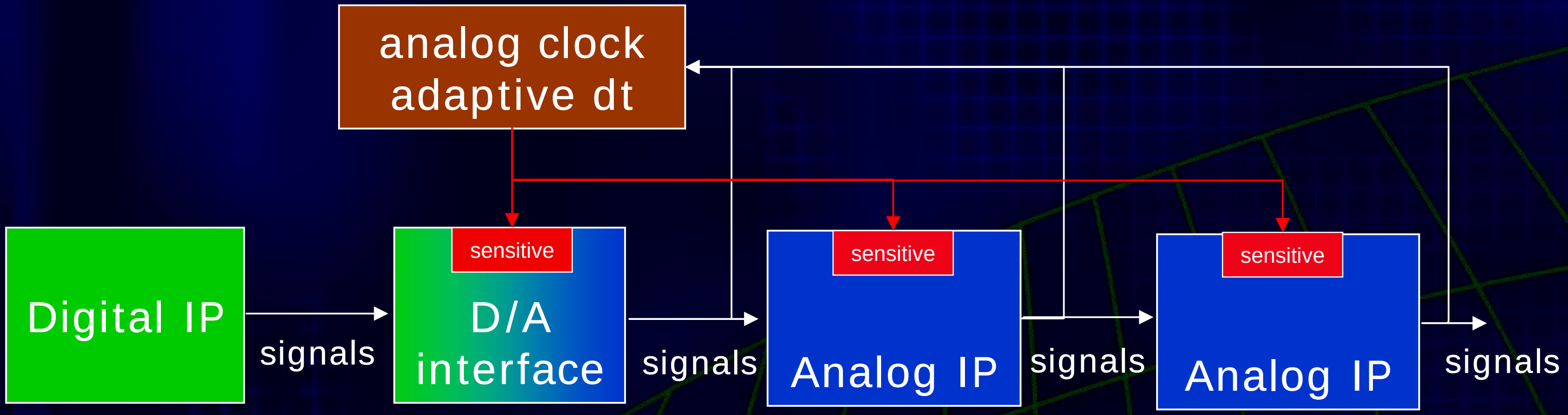


Methodology

***Global adaptive time step for all the analog processes*.**

((synchronous, adaptive dt)

A new global thread is necessary to calculate at each time instant the smallest time step required to guarantee the required accuracy. This thread acts as an analog simulation kernel. This solution is inefficient when analog blocks with different time constants are present in the same system.

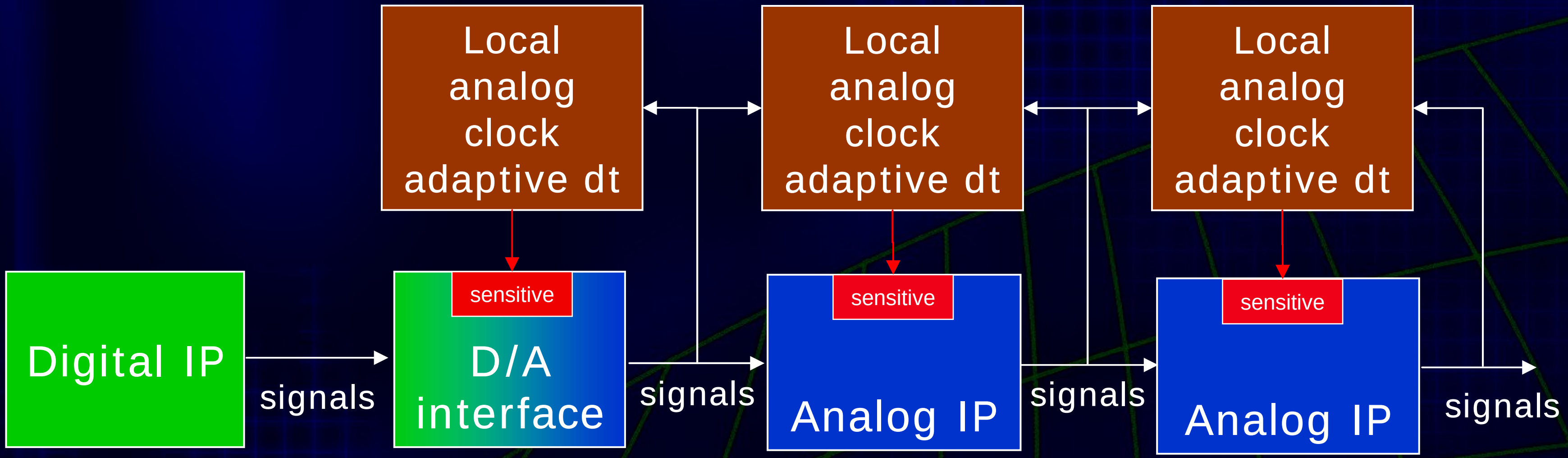


Methodology

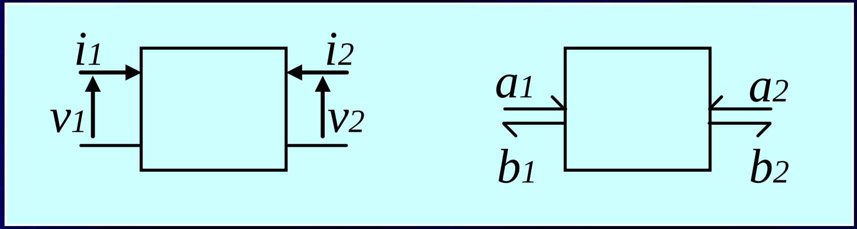
Local adaptive time step

((asynchronous, adaptive dt)

Each analog process calculates its adaptive time step, and it passes this time step to the blocks to which its outputs are connected. The analog inputs activate the process, that calculates the new state and output values. Then it sends them to the connected analog blocks after the minimum between its time step and the time steps received from the input blocks.



Systems non representable with Signal Flow Graph

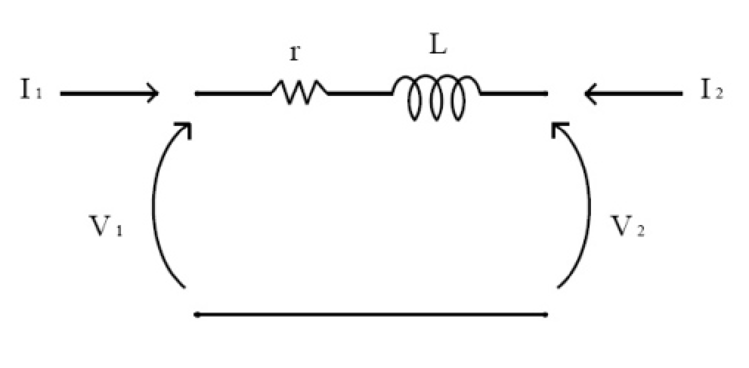


Variable transformation using incident and reflected waves a,b

$$a_j = \frac{1}{2}\left(v_j / \sqrt{R_j} + i_j \sqrt{R_j}\right)$$

$$b_j = \frac{1}{2}\left(v_j / \sqrt{R_j} - i_j \sqrt{R_j}\right)$$

Example



$$v_L = L\frac{d}{dt}i_L$$
$$v_1 = v_L + R\cdot i_L + v_2$$
$$i_1 = i_L$$
$$i_2 = -i_L$$

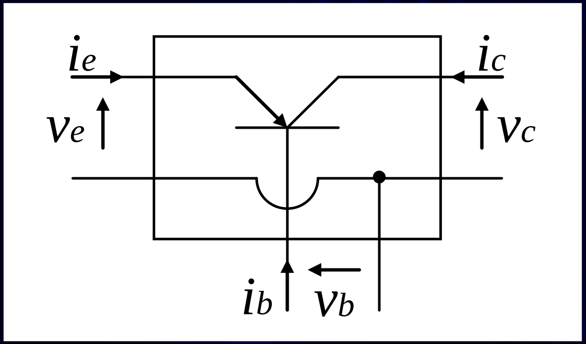
$$\longrightarrow$$

$$b_1 = a_1 - \frac{x}{L}$$
$$b_2 = a_2 + \frac{x}{L}$$
$$\dot{x} = 2\cdot\left(a_1 - a_2\right) - x\frac{\left(2+R\right)}{L}$$

Current and Voltages

Incident and Reflected waves a,b

Example: pnp BJT



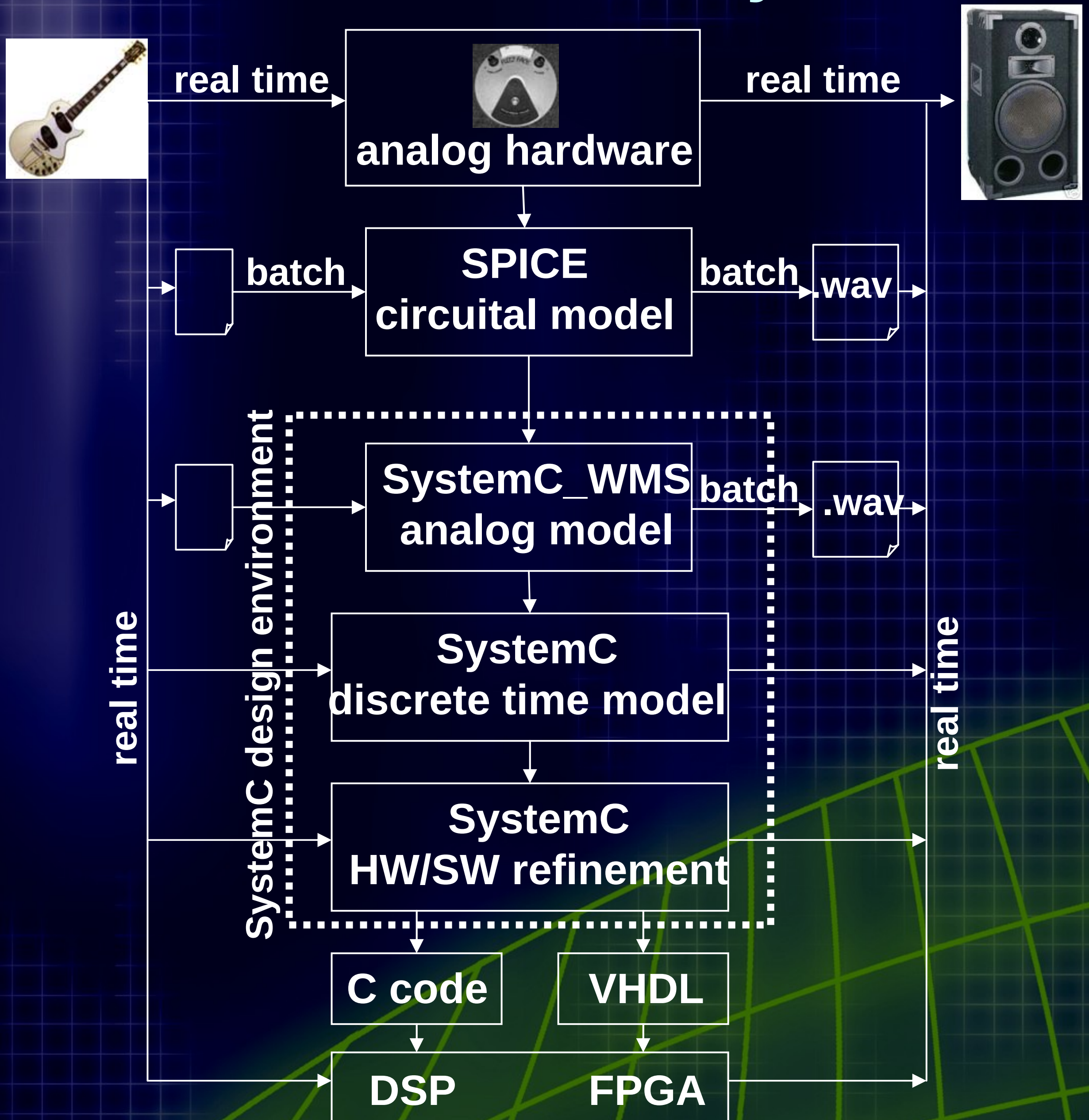
$$a_b = \frac{1}{2}\left(v_b / \sqrt{R} + i_b \sqrt{R}\right); \quad a_c = \frac{1}{2}\left(v_c / \sqrt{R} + i_c \sqrt{R}\right); \quad a_e = \frac{1}{2}\left(v_e / \sqrt{R} + i_e \sqrt{R}\right)$$

$$b_b = \frac{1}{2}\left(v_b / \sqrt{R} - i_b \sqrt{R}\right); \quad b_c = \frac{1}{2}\left(v_c / \sqrt{R} - i_c \sqrt{R}\right); \quad b_e = \frac{1}{2}\left(v_e / \sqrt{R} - i_e \sqrt{R}\right)$$

$$i_b = (1-\alpha_F)I_{EC} + (1-\alpha_R)I_{CC}; \qquad i_c = I_{CC} - \alpha_F I_{EC}; \qquad i_e = I_{EC} - \alpha_R I_{CC}$$

$$I_{EC} = I_{E0}\left(\exp\left(\frac{v_e - v_b}{v_t}\right) - 1\right); \qquad I_{CC} = I_{C0}\left(\exp\left(\frac{v_c - v_b}{v_t}\right) - 1\right);$$

System Level Design Methodology for sound transformation based on Physical Modeling



real time

analog hardware

real time

batch

SPICE circuital model

batch

.wav

SystemC design environment

SystemC_WMS analog model

batch

.wav

SystemC discrete time model

SystemC HW/SW refinement

real time

real time

C code

VHDL

DSP

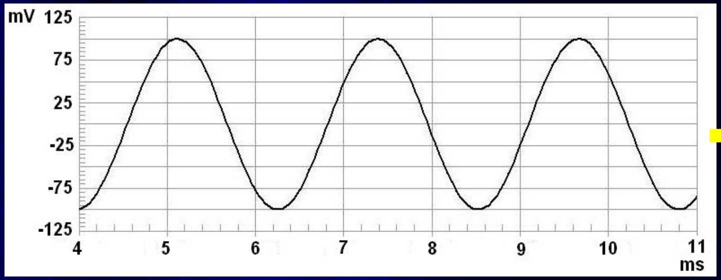
FPGA

Physical Modeling of electric guitar effect

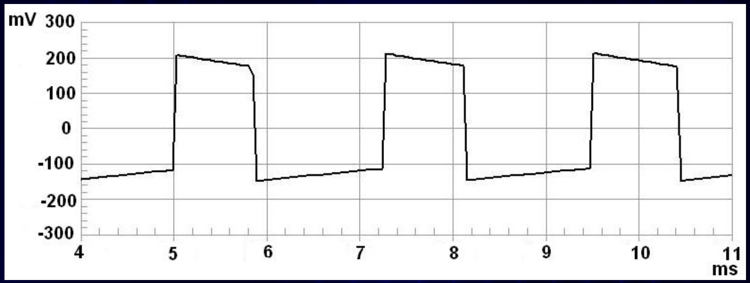
Fuzz Face

- Analog distortion electronic circuit for electric guitar
- Produced by Dallas Arbiter in 1966
- Famous in the ’60 and ’70 and used by
  Jimi Hendrix, David Gilmour, Carlos Santana, Eric Clapton,
- Widely used up to now.

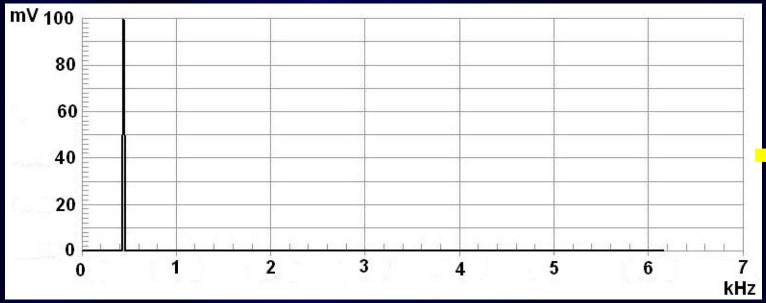
Distorsion Effect



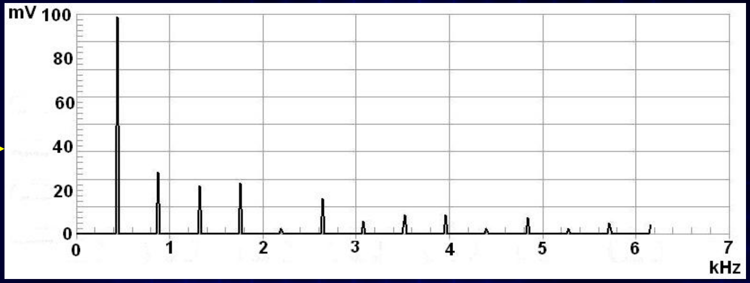
Input (t)



Fuzz Face output (t)



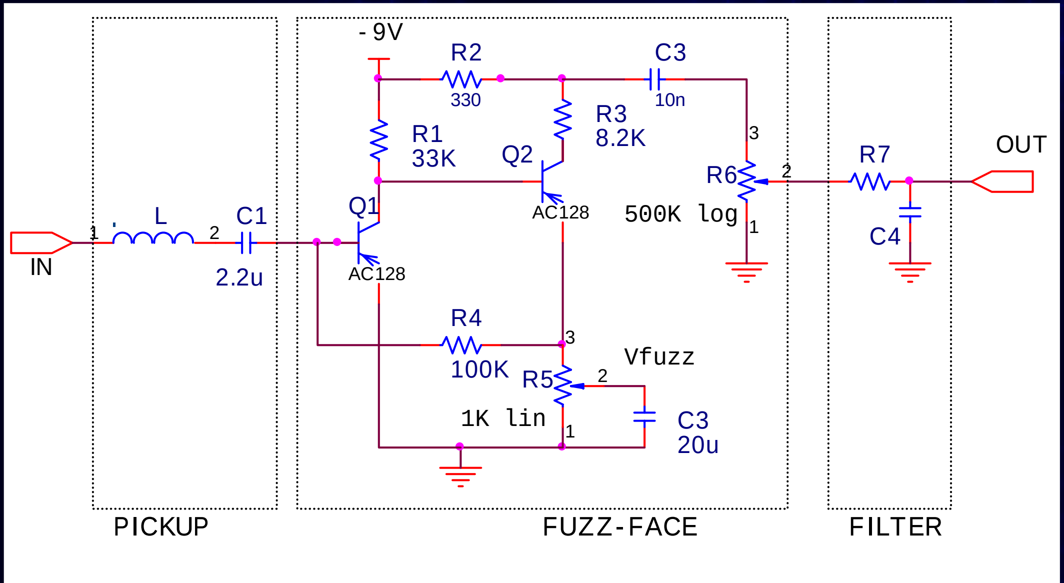
Input spectrum



Fuzz Face output spectrum

The circuit saturates for some values of the input signal, giving the effect of distorting a sinusoid into an almost square waveform, introducing new harmonic components.

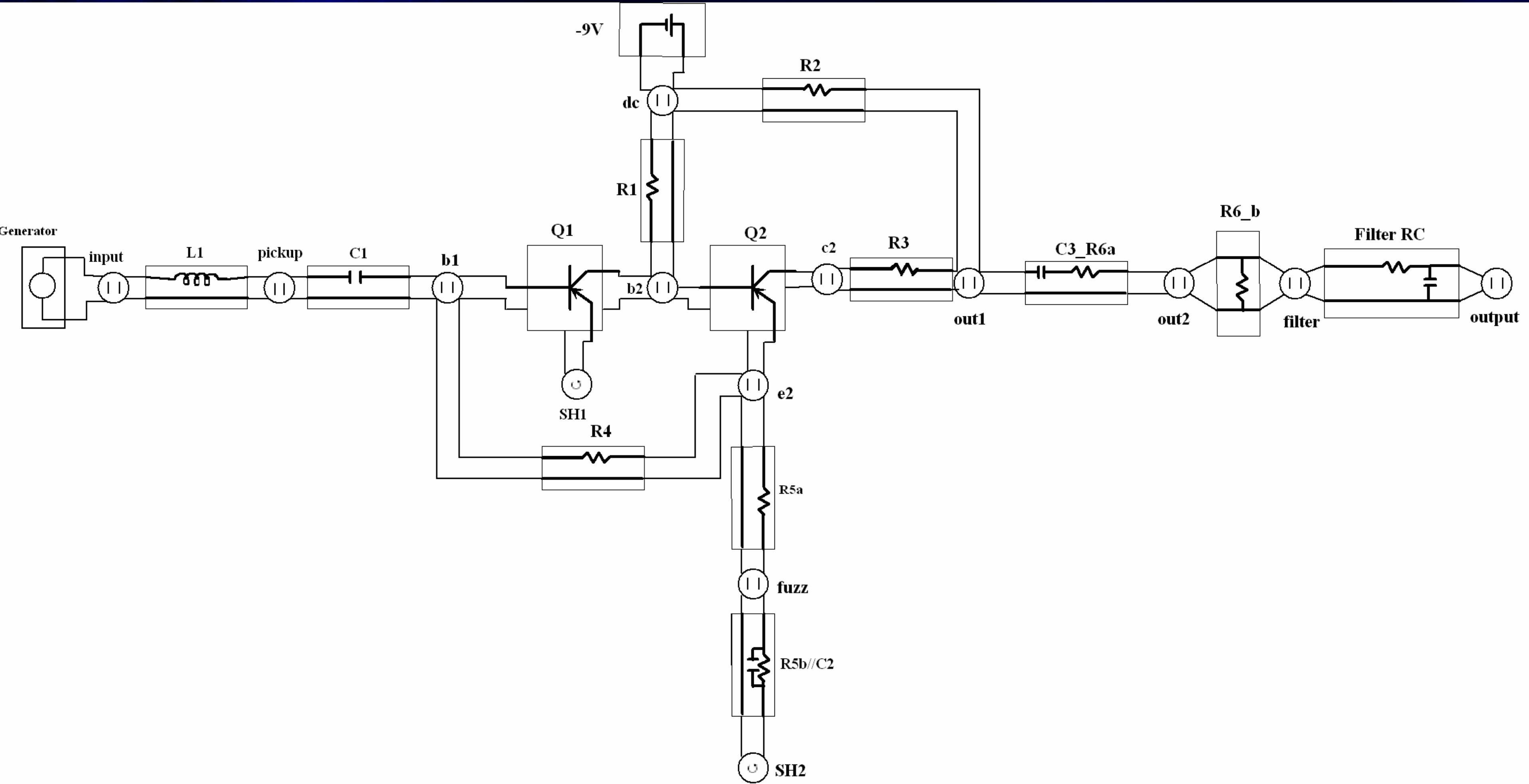
Fuzz-Face circuit



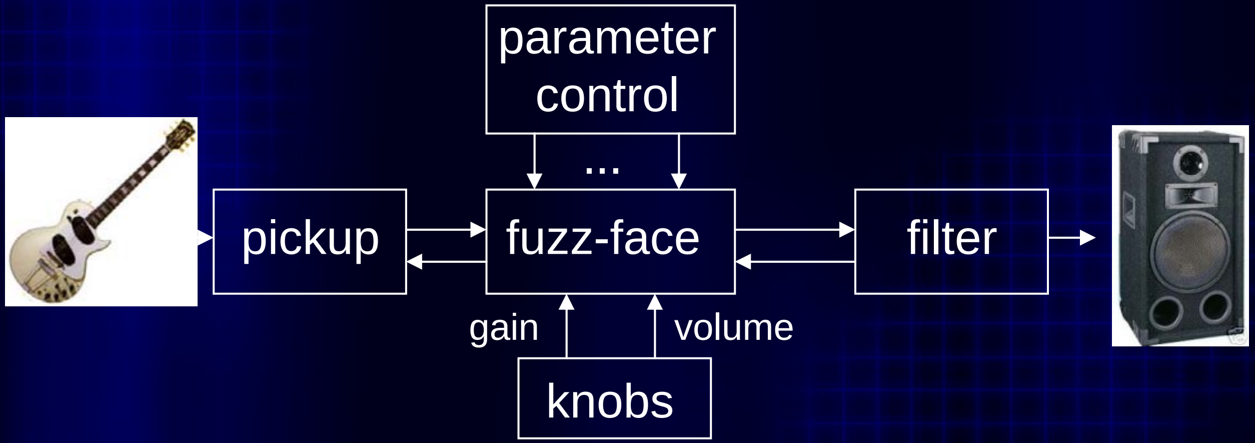
- The original and the actual analog circuit use two coupled germanium transistors, that are preferred due to their better "sound performances" to the cheaper silicon transistors.
- The germanium transistors suffer from
  - strong dependence on temperature and humidity,
  - high variability of the parameters,
  - high probability of breaking during soldering,
  - high cost due to obsolete germanium technology.
- Advantage of a digital implementation based on physical modeling

SystemC-WMS low level implementation of the Fuzz-Face circuit

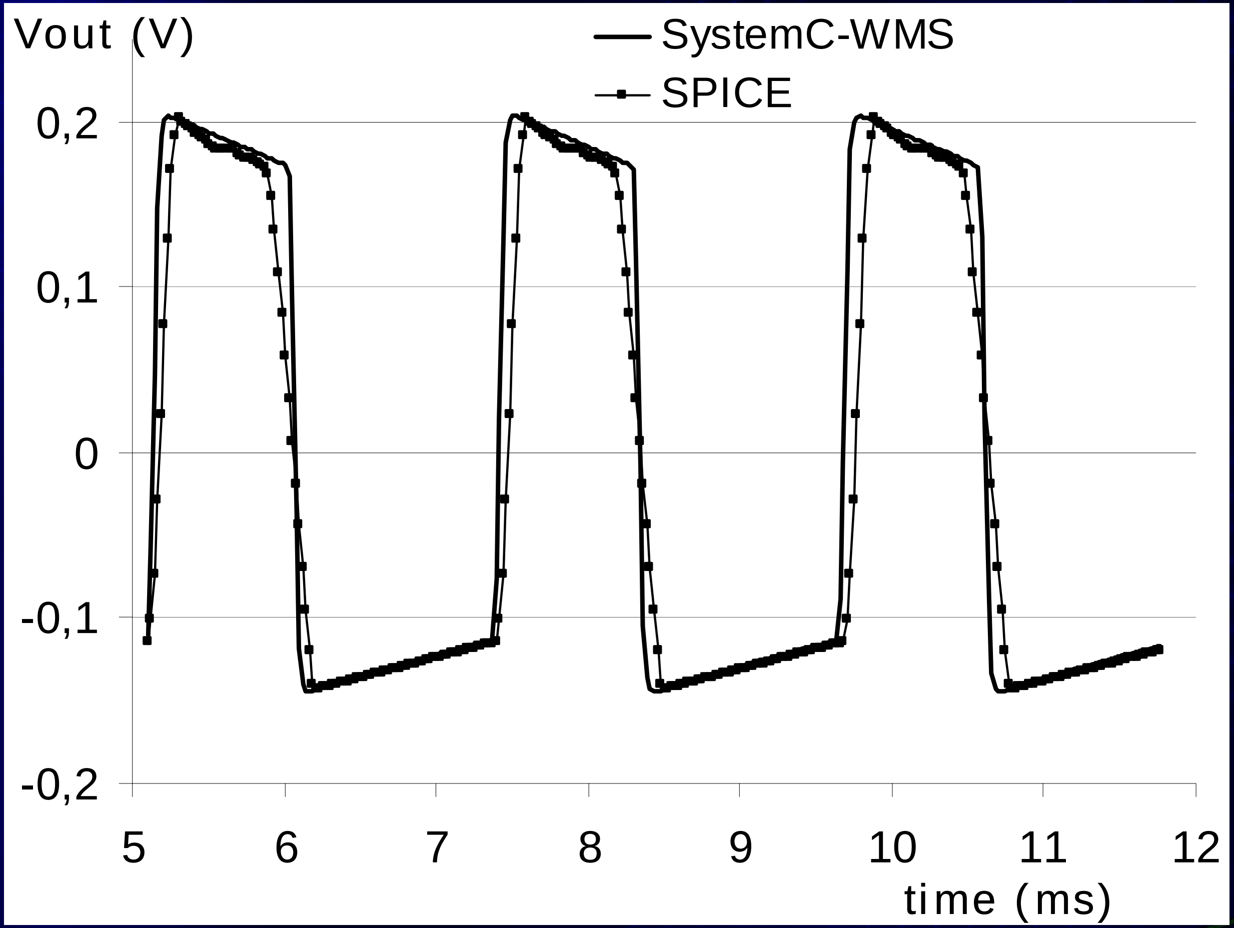
(15 modules, 14 channels)



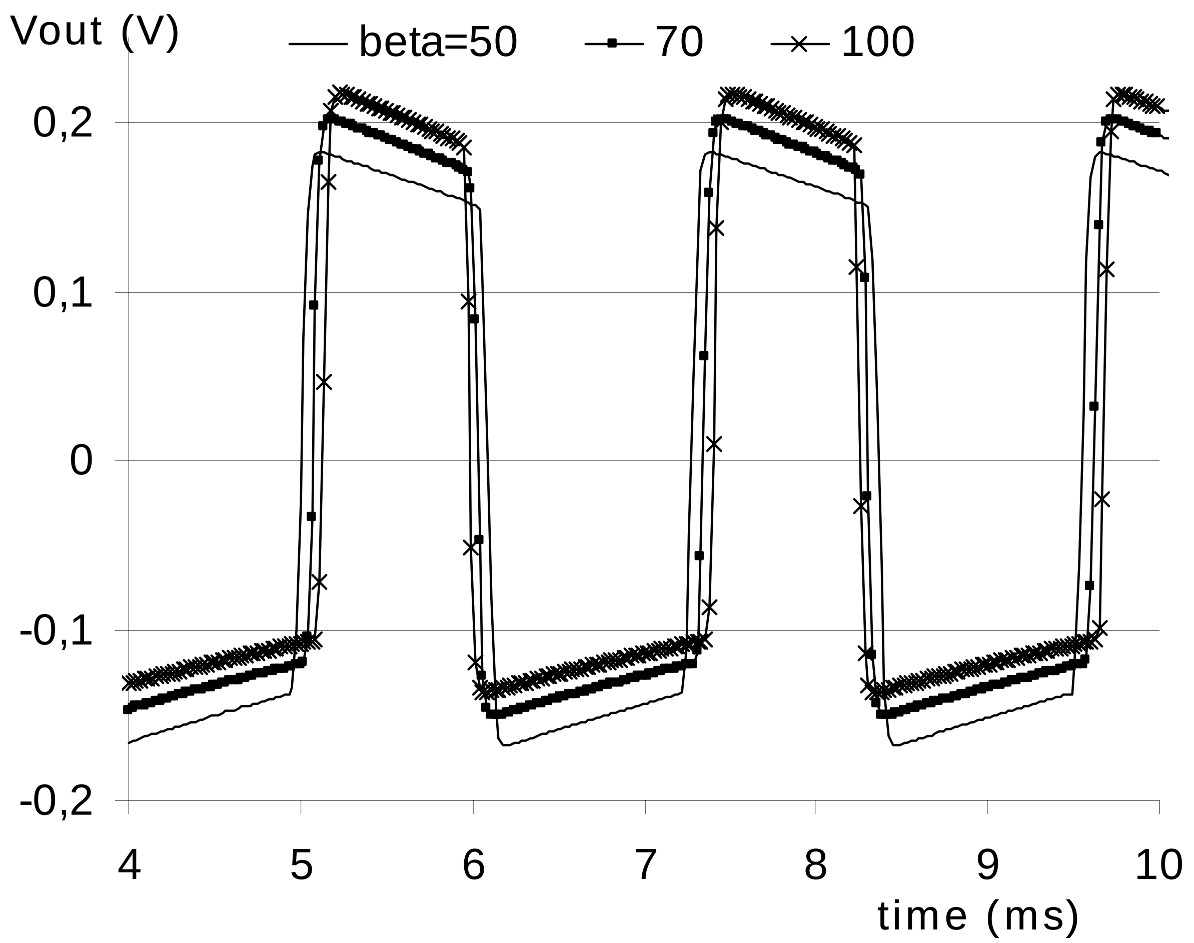
Analog description in SystemC-WMS using macroblocks



- The macroblock description of the circuit consists of 5 SystemC-WMS analog modules.
- The transistor control block is used to fix the technological parameter of the transistors.
- The knobs block is used to change the gain and the volume, that are changed by the electric guitar player during its performance.

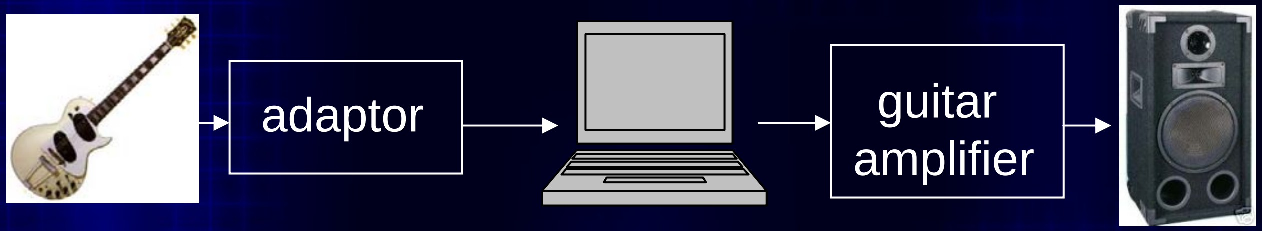


SPICE and SystemC (using a discrete time description of macroblocks) simulations of the Fuzz-Face circuit with a sinusoidal input



SystemC simulations of the fuzz-face circuit with a sinusoidal input with different values of the BJT parameter β.

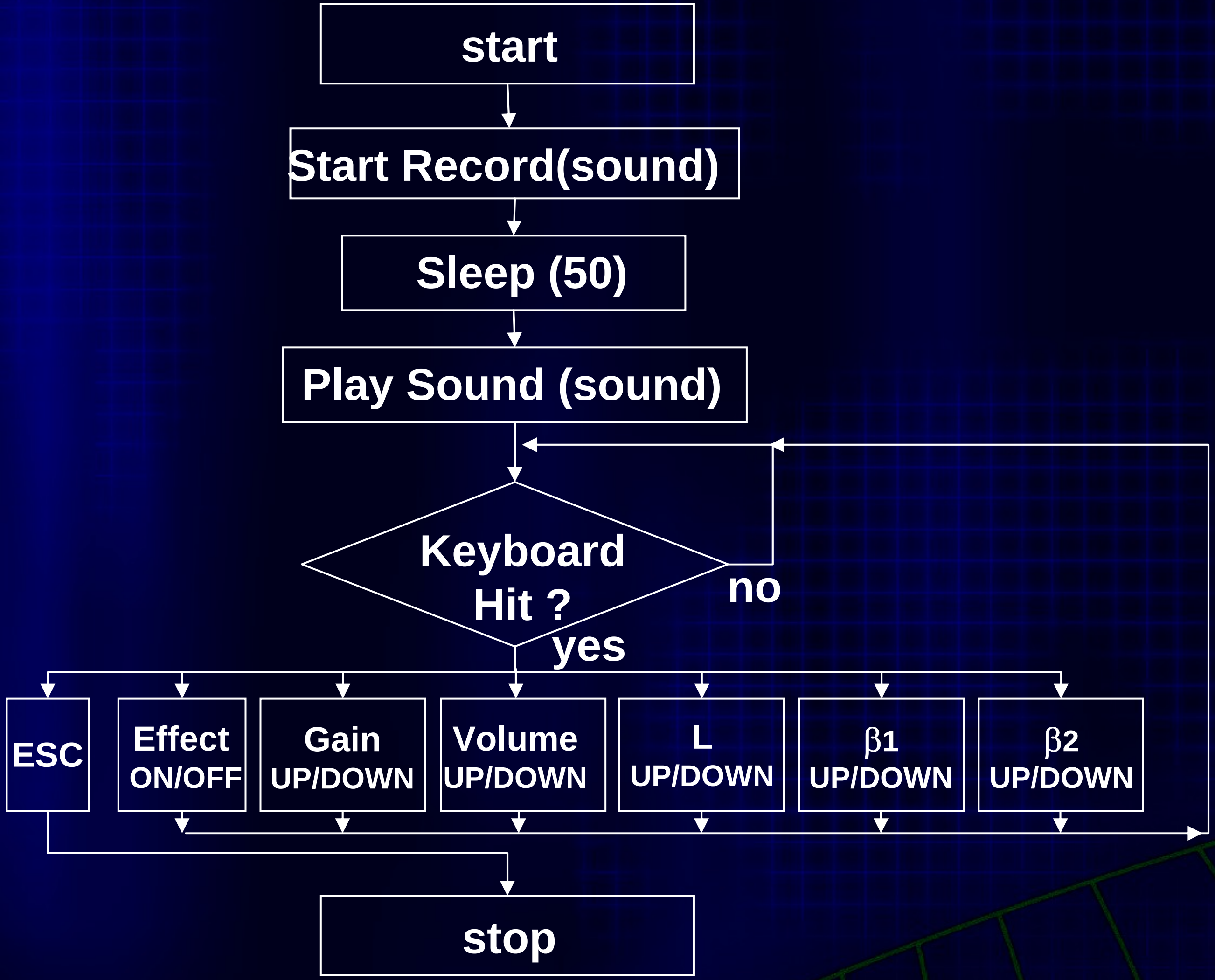
β1= β2= 50, 70, 100



adaptor

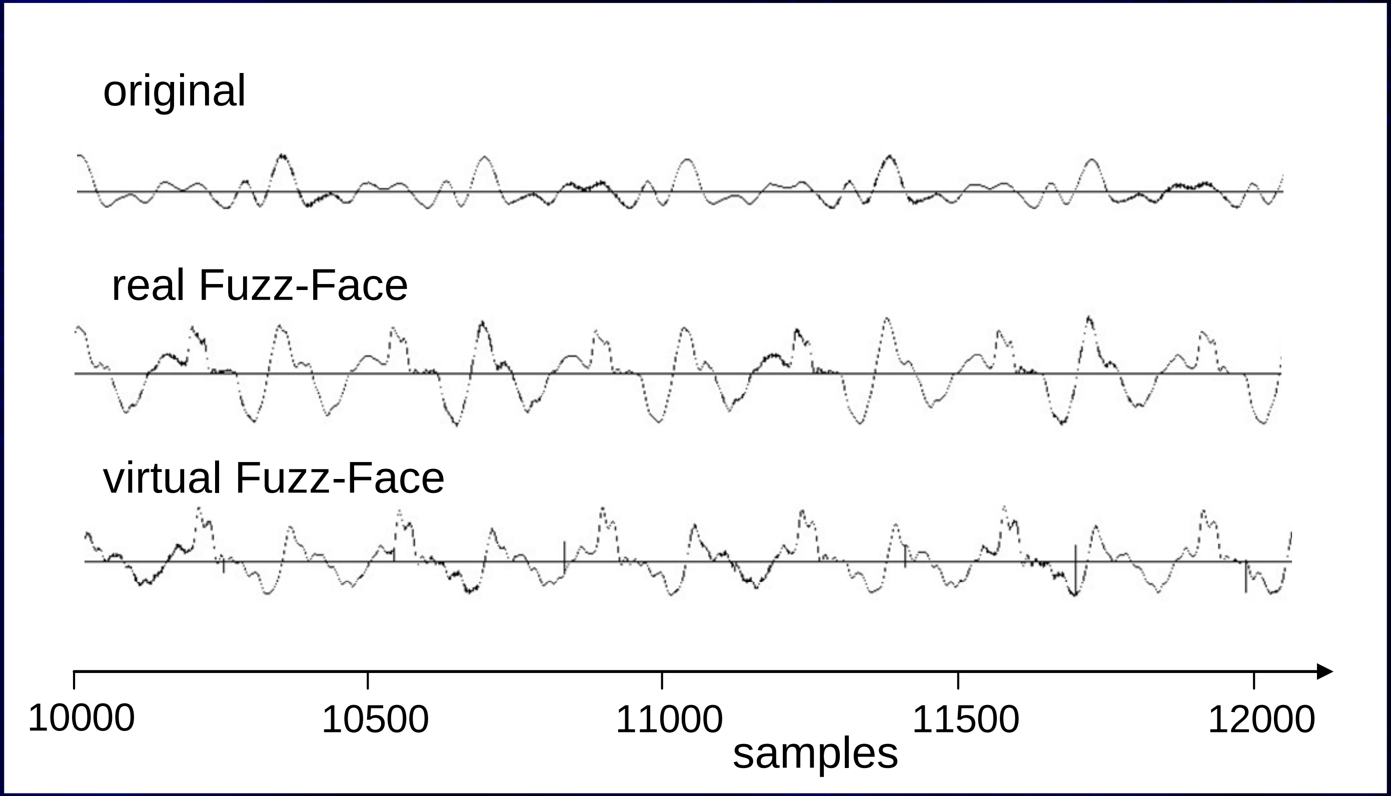
guitar amplifier

- The discrete time C++ physical model of the fuzz-face is then extracted from the SystemC environment and integrated with other libraries allowing audio real time simulations.
- The input waveform can be taken from a file or directly from the microphone input of the PC, and the output can be stored into a file or directed to the audio output of the PC.



Physical modelling allows the definition of parameters directly related to the physical mechanism the generated the distortion.

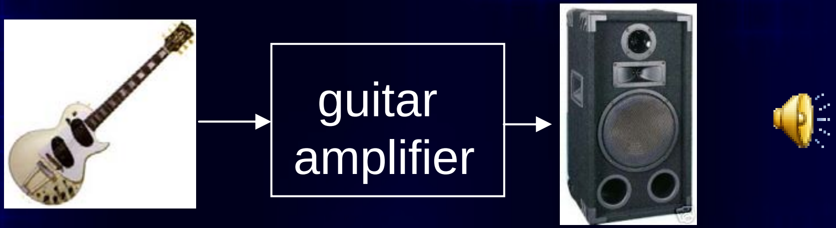
These parameters are changed in real time while the artist is playing his guitar.



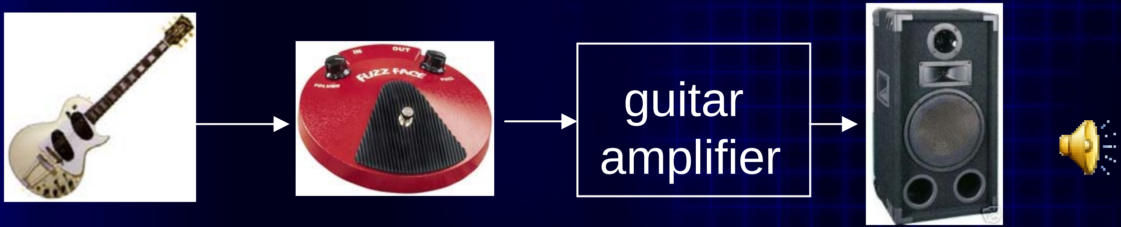
Waveforms of the electric guitar

- without distortion (original),
- after the distortion of the hardware Fuzz-Face (real Fuzz-Face)
- after the distortion obtained with software developed (virtual Fuzz-Face)

original

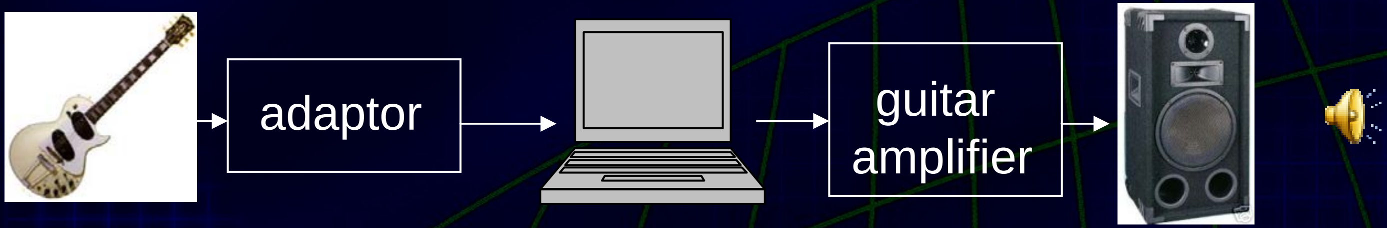

guitar amplifier

real Fuzz-Face



guitar amplifier

virtual Fuzz-Face


adaptor

guitar amplifier

- Starting from the golden model defined and verified in previous steps, the designer can simplify the algorithm, or define the architecture implementing the algorithm and the number of bits used to represent the variables.
- The next step will be the implementation in hardware using FPGA or DSP.

Conclusions

- This works presents a general methodology for the integration of physical modeling of sounds in a system level design environment using SystemC.
- The methodology has been applied in particular for physical modeling of electric guitar effects.
- The model developed has been used for a real time synthesis of a tunable electric guitar effects generator on a PC.