Baseband Fading Channel Simulator
For Inter-Vehicle Communication
Using SystemC-AMS

Abdelbasset Massouri
Laurent Clavier
Andreas Kaiser

supported by: ANR WASABI Project

Wireless system And SystemC-AMS: Basic Infrastructure

Partners: STMicroelectronics Grenoble, Magillem Design Services
- AMS&RF SoC virtual prototyping and validation through the development of an industrial example: Wireless Video system (WVS).
- Inter-Vehicle Communication

MOTIVATION

- What is the efficient behavioral RF model to be used in this complex application?
- Which tool/programming language should be used to model & simulate this application?
RF models and Modeling language

Application & Simulation Platform

Wireless Channel Model

Simulation results

Prospects
MODELING LANGUAGE: SYSTEMC-AMS

APPLICATION & SIMULATION PLATFORM

WIRELESS CHANNEL MODEL

SIMULATION RESULTS

SUMMARY
Questions:

- Which Signal representation of RF Signal?

- Simulations can be done using either passband or complex baseband representation.

- Passband
  - Pass-band simulations are more accurate
  - However, they consume more resources and simulation time.

- Baseband
  - Baseband models suppress the carrier frequency to trade some accuracy for a dramatic increase in execution speed (they run thousands of times faster than passband models)
  - Baseband models allow a generic channel modeling

Baseband behavioral models are used for RF devices and wireless channel
Why SystemC-AMS?
- According to previous works, it was established that SystemC-AMS is an efficient tool to deal with the described application.

What is SystemC-AMS?
- Analog/Mixed-Signal (AMS) standard of the Open SystemC™ Initiative (OSCI)
- Open Source

Model of Computation
1. Electrical Linear Networks (ELN): used to model continuous time behavior (current & voltage)
2. Linear Signal Flow (LSF): used to model continuous time behavior
3. Timed Data Flow (TDF): facilitates a very efficient simulation, as TDF models are processed at discrete time points without using the discrete-event kernel of SystemC.

TDF is the SystemC-AMS formalism used in this work.
OUTLINE

- Modeling language: SystemC-AMS
- Application & Simulation Platform
- Wireless Channel Model
- Simulation results
- Summary
• **STIMULUS**
  - Positions, speeds

• **CAR**
  - QPSK modulation, Non-linearities, …

• **Wireless Channel**
  - multipath
  - Time-varying
  - Broadcasting characteristics

  ➢ **20 wireless time-varying channel**
  - Time and memory consuming
OUTLINE

- Modeling language: SystemC-AMS
- Application & Simulation Platform
- Wireless Channel Model
- Simulation results
- Summary
What do we mean by wireless communication channel?

- Transmitter (Tx)
- Receiver (Rx)

The transmission channel comprises antennas and all objects contributing or hampering propagation between source and destination nodes.

The propagation channel excludes the antennas and expresses all wave propagation phenomena between Tx and Rx.

Transmission channel is considered in this work!!
Wireless Channel Modeling

Transmitter (Tx) and Receiver (Rx)

\[
y(t) = \left(10^{L_{dB}/10}\right) \cdot \chi \cdot \sum_{k=0}^{K-1} \tilde{g}_k(t) \cdot x \left(t - \frac{k}{f_s}\right) + n(t)
\]

\[
h(t) = \left(10^{L_{dB}/10}\right) \cdot \chi \cdot \sum_{k=0}^{K-1} \tilde{g}_k(t) \cdot \delta \left(t - \frac{k}{f_s}\right) + n(t)
\]
Node-to-Node Link

\[ h(t) = \left(10^{L_{dB}/10}\right) \chi \sum_{k=0}^{K-1} \tilde{g}_k(t) \delta\left(t - \frac{k}{f_s}\right) + n(t) \]

- **Path Loss**
  - Mean attenuation at a given distance
  - Simple
  - Short time of simulation

- **Shadowing**
  - Environment
  - Simple
  - Short time of simulation

- **Small Scale Fading : Multipath**
  - Reflection, diffraction, diffusion, refraction, ...
  - Complex
  - Memory Consuming
  - Simulation time consuming

- **AWGN**
  - Non-idealities of Antenna
  - Simple
  - Short time of simulation

**Time-varying multipath contribution will be detailed!!**
Small Scale Fading Contribution: Multipath propagation

- Tapped Delay Line (TDL)
  - Uniformly spaced model
  - FIR filter (order $K =$ number of paths)
  - Coefficients are complex Gaussian variables

- Filtered Gaussian Noise
  - 2 independent Gaussian Variables (Box-Muller method)
  - Time-varying criteria: Doppler filter

\[ h_{MPC}(t) = \sum_{k=0}^{K-1} \tilde{g}_k(t) \delta\left( t - \frac{k}{f_s} \right) \]

**Doppler Filter Design!!**
Doppler Shift?
- Motion of cars or scatterers produces Doppler shifts of incoming received waves
- Frequency shift ~ Doppler spread
- Time-varying aspect of the wireless channel is due to this physical phenomenon

Which Doppler Spectrum for Mobile Communication?
- Jakes, Flat, Gaussian, Rounded, ...

Jakes Doppler Spectrum
- Spectrum
- Amplitude frequency
- Power Spectrum: « U Shape »
- \( f_d << f_s \)
- cut-off frequency is sharp
- Frequency-domain: (simple, but all the channel coefficients must be generated in the beginning of the simulation)
- Time-domain: (complex, but it has the real-time aspect of wireless communication)
### Small Scale Fading: SystemC-AMS Implementation

```c
#ifndef SMALL_SCALE_FADING_H
#define SMALL_SCALE_FADING_H

#include "compute_ssf.h"

SCA_TDF_MODULE(small_scale_fading)
{
    sca_tdf::sca_in < complex<double> > in;
    sca_tdf::sca_out < complex<double> > out;
    sca_tdf::sca_in < double > v;

    public:
        // Doppler filter parameters
        double Ap;      // Band pass ripple in dB
        double As;      // Stop band ripple in dB
        double fp;       // Band pass edge frequency (maximum doppler shift in Hz)
        double fs;       // Sampling frequency in Hz
        double fc;       // Carrier frequency in Hz

        compute_ssf *compute_ssf_inst;

    void processing(void)
    {
        this->fp = 3.6*v.read()*fc/speed_light;
        complex<double> yt = compute_ssf_inst->compute(in.read(),v.read());
        out.write(yt);
    }

    SC_CTOR(small_scale_fading)
    {
        this->Ap = 0.5;
        this->As = -80;
        this->fs = 160e6;
        this->fc = 5.9e9;
        compute_ssf_inst = new compute_ssf(Ap, As, fp, fs, fc);
    }
}
#endif // SMALL_SCALE_FADING_H
```

- **Small_scale_fading** TDF module declaration
- TDF Input & Output ports: Complex baseband signals
- TDF Input port: Vehicular Speed (used to compute Doppler shift)
- Attributes (used to compute the Doppler filter coefficients)
- **Compute_ssf** class (it implements the multipath contribution)
- Processing method (invoked at each sample time)
- Apply the multipath contribution on the input signal to get the output one
- Constructor of the small_scale_fading TDF module
- Attributes initialization and compute_ssf object instantiation
Node-to-Node Link: SystemC-AMS Implementation (Netlist)

```c
#ifndef NODE_TO_NODE_LINK_H
#define NODE_TO_NODE_LINK_H

#include "pathloss/pathloss.h"
#include "small_scale_fading/small_scale_fading.h"
#include "awgn/awgn.h"

SC_MODULE(node_to_node_link)
{
    sca_tdf::sca_in<complex<double>> in;
    sca_tdf::sca_out<complex<double>> out;
    sca_tdf::sca_in<double> d;
    sca_tdf::sca_in<double> v;
    sca_tdf::sca_signal<complex<double>> sig1;
    sca_tdf::sca_signal<complex<double>> sig2;
    pathloss *pathloss_inst1;
    small_scale_fading *small_scale_fading_inst1;
    awgn *awgn_inst1;

    SC_CTOR(node_to_node_link)
    {
        pathloss_inst1 = new pathloss("pathloss_inst1");
        small_scale_fading_inst1 = new small_scale_fading("small_scale_fading_inst1");
        awgn_inst1 = new awgn("awgn_inst1");

        pathloss_inst1->in(in);
        pathloss_inst1->out(sig1);
        pathloss_inst1->d(d);
        small_scale_fading_inst1->in(sig1);
        small_scale_fading_inst1->out(sig2);
        small_scale_fading_inst1->v(v);
        awgn_inst1->in(sig2);
        awgn_inst1->out(out);
    }

    #endif // NODE_TO_NODE_LINK_H
```
OUTLINE

- Modeling language: SystemC-AMS
- Application & Simulation Platform
- Wireless Channel Model
- Simulation results
- Summary
System Performance Simulation results

- Fading channel
- QPSK constellation
- Eye diagram

- The faster are the vehicles
- The more the system Performances are effected

- Bit Error Rate, …
Simulation Speed Enhancement

- **Linux Ubuntu machine**: 1) Dual 64 bit 2.4GHz Intel Xeon processors, 2) 12GB memory

- 5 vehicles scenario, 20 time-varying wireless channel
- TDMA protocol to avoid multi-access interferences
- Total time of simulation: 50 Time Slots (10 transmissions for each car)

![Matlab Time of Simulation](chart1)

![SystemC-AMS Time of Simulation](chart2)
OUTLINE

- Modeling language: SystemC-AMS
- Application & Simulation Platform
- Wireless Channel Model
- Simulation results
- Summary
Wireless channel was modeled and implemented using SystemC-AMS for virtual prototyping of AMS&RF SoC

SystemC-AMS is an efficient tool to simulate Heterogeneous System

TDF formalism is accurate and it speeds up simulation

- Add Channel coding and decoding processing to the current toolbox
- Perform Time-domain Equalization in order to combat Inter-Symbol Interference
- Perform High level power estimation
Thank you for your Attention