NANOPOWER

Project number	**256959 (Grant Agreement Number)**		
Project title	**NANOPOWER -** Nanoscale energy management for powering ICT devices		
Call identifier	FP7-ICT-2009-5	**Funding scheme**	Collaborative Project
Starting date	01/08/2010	**Duration**	36 months
PARTNERS	UNIPG (Coordinator), UNI WUE, VTT, ICN, UNIGE, UNICAM		

Deliverable D1.2

Numerical simulation tools for nonlinear stochastic dynamics

Due date of deliverable	M12
Actual submission date	31. July 2011
Version	1.1
Dissemination level	RE

Short description

Numerical simulation tools for nonlinear stochastic dynamics : In the form of MATLAB codes, to be used in the design phase for WP4. Specifically these are codes for the numerical solution of nonlinear differential stochastic equations that are necessary for modeling the dynamical behavior of nonlinear oscillators in the presence of random forces.

SEVENTH FRAMEWORK PROGRAMME

Table of Content

Description of the deliverable

D 1.2) Numerical simulation tools for nonlinear stochastic dynamics : this report presents the method used to realize the MATLAB code to solve the nonlinear differential stochastic equations that are necessary for modelling the dynamical behaviour of piezoelectric nonlinear oscillators in the presence of random forces.

In this deliverable we will identify the numerical method to be implemented and we will report the MATLAB code to solve the dynamic and voltage coupling of nonlinear piezoelectric oscillators.

Progress towards objective

For the solution of stochastic differential equations we used the Euler-Maruyama method. The Euler-Maruyama method is a method for the approximate numerical solution of a stochastic differential equation. It is a simple generalization of the Euler method for ordinary differential equations to stochastic differential equations.

code

```
function [Xstd,Vstd]=em()
% Numerical simulation tools for nonlinear stochastic dynamics of piezoelectric oscillators
% em() returns the standard deviation of displacement
% and voltage produced of a piezoelectric oscillator
% subjected to exponentially correlated noise.
% The simulation is performed using the Euler-Maruyama
% method.
% The dynamic of the oscillator, the physical properties
% and the coupling characteristics can be customized modifying the code.


% Simulation parameter
% Integration step, best results with lower value (min value 10^-4)
dt=10e-4;
% Number of step to simulate
N=500000;

% Noise parameter
% Intensity of the noise
sigma=0.3;
% Exponentially correlated noise correlation time
tau=0.01;

% Mechanical/Electrical parameters
% Product between capacitance of the piezoelectric and load resistance
RC=1000;
% Account for the dissipated energy trough the electric load
```

```
Kv=0.5;
% Coupling term between voltage and displacement
Kc=0.5;
% Damping factor (viscous damping)
damp=0.5;

% Time vector resulting from time step and number of step
t=(0:N-1)*dt;  % time vector

% We assume to explore a two parameter represented by the vector A and B.
A=-1:0.1:1;
B=0.1:0.1:1;

% Initialize vector for standard deviation of displacement
Xstd=zeros(length(A),length(B));
% Initialize vector for standard deviation of displacement
Vstd=zeros(length(A),length(B));

k=0;
% Cycle trough parameter A
for a=A
        k=k+1;
        q=0;
        % Cycle trough parameter B
        for b=B
                % Add other for cycle for each further parameter to explore

                % Display actual computing parameter
                disp([a b]);

                q=q+1;

                epsilon=zeros(1,N); % Vector to store generated noise
                dW=sqrt(dt)*randn(1,N); % Brownian increment
                randn('state', sum(100*clock)); % Initialize random number generator (keep seed constant to generate the same noise each iteration)
                % Generation of exponentially correlated noise
                for g = 1:N-1
                        epsilon(g+1) = epsilon(g)*(1-dt/tau) + randn*dt;
                end
                epsilon = epsilon/std(epsilon); % Normalization of noise standard deviation to 1 (the noise will be multiplied by sigma)
                epsilon = epsilon - mean(epsilon); % Normalization of noise mean to 0

                X=zeros(1,N);   % Initialization of position vector
                V=zeros(1,N);   % Initialization of Voltage vector
```

```
            X(1)=0; % Set initial position
            V(1)=0; % Set initial voltage
            Y=0; % Set initial velocity

            % Define the force function relative to the potential used (in this example duffing
potential)
            forcefun=@(x)((a.*x-b.*x.^3));

            % Solve the set of nonlinear stochastic differential equation
            for i=1:N-1
                X(i+1)=Y*dt+X(i);
                Ynew=Y+(forcefun(X(i))-damp*Y-Kv*V(i))*dt+sigma*epsilon(i)*dW(i);
                V(i+1)=V(i)+(Kc*Y-V(i)/RC)*dt;
                Y=Ynew;
            end
            % At this step the vectors X and V contain the simulated position and voltage
produced for each step

            % Compute the standard deviation of the position, we remove the first unreliable
points
            Xstd(k,q)=std(X(5000:N));
            % Compute the standard deviation of the Voltage, we remove the first unreliable
points
            Vstd(k,q)=std(V(5000:N));
      end
end
end
```

Conclusion

We have validated the simulation tool comparing the experimental result to the simulation whit good agreement. The MATLAB code to solve the nonlinear differential stochastic equations is reported in appendix and as **.m** file.

References

1) An Algorithmic Introduction to
   Numerical Simulation of
   Stochastic Differential
   Equations
   - Desmond J. Higham
   SIAM REVIEW
   Vol. 43, No. 3, pp. 525–546

2) Kloeden, P.E., & Platen, E. (1999). Numerical Solution of Stochastic Differential Equations. Springer, Berlin. ISBN 978-3-540-54062-5.