Model Name:	Client Concurrent Simulation Model
Model version Number:	2.0
Authors:	M. Maksimov; V. Bandura
Author’s Organization:	Taganrog Radio-Technical University
Date:	March 26, 2003

INTRODUCTION

This model represents the client part of a symmetrical linking model.

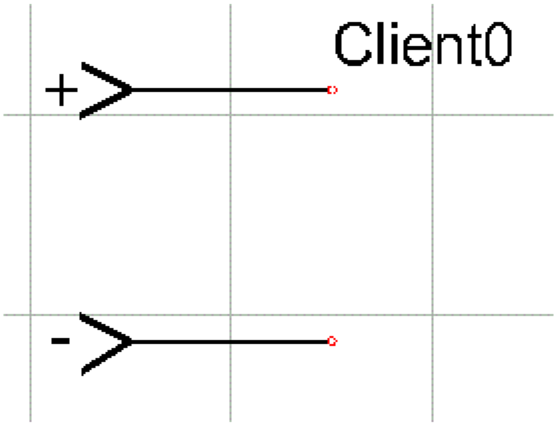

Fig 1. Pictorial representation of CConcurrentSim model

BRIEF DESCRIPTION OF MODEL

The model represents the client part of a symmetrical linking model. A symmetrical linking model is intended for simulating different parts of large circuits on separate computers, connected by a network. These parts are combined by the linking models. The server and the client parts with the same endpoint value represent one connection.

The communications between a server and a client part may be realized using RPC (remote procedure call, in this case workstations may be connected by a network) or using a pipe (a section of shared memory that processes use for communication).

This version of model is intended for simulation of linear circuits that were distributed only on two parts.

MODEL VALIDITY RANGE AND LIMITATIONS

Because of synchronization, the time step while simulating, must be constant and the same in all parts of the combined circuit. Also, the simulation start time and the end time must be the same.

List of Model Terminals With Connectivity Information

Pin Designation	Description
Terminal –	Negative terminal in connection to the server
Terminal +	Positive terminal in connection to the server

LIST OF MODEL PARAMETERS

Parameter Name	Description	Default Value	Units
Rst	Impedance of the stabilizing element	1	Om
InterVTB communications	Type of interprocess communications used in this model: RPC (remote procedure call) or pipes	using RPC	
Computer name	For RPC communications: the unique name of the computer, where the server is located	Bvv	
EndPoint	For RPC communications: unique value to identify the location of the connection	2290	
Protocol	For RPC communications: TCP/IP or UDP/IP	TCP/IP	
PipeName	For pipe communications: unique name of the pipe	pipe0	
Auto adjustment of Rst	Type of auto adjustment of the stabilizing resistance (Rst)	after 4 iterations	
Tolerance	Maximal difference between transmitted to the server voltage and accepted from the server voltage	7e-9	Volts

LIST OF OUTPUT VARIABLES

VARIABLE NAME	DESCRIPTION	UNITS
Voltage	Voltage across this element	V
Current	Current through this element	A
Resistance	Impedance of the stabilizing element	Om
Iterations	Number of iterations needed to achieve required convergence	

MATHEMATICAL DESCRIPTION OF MODEL

This model represents one part of the general coupling scheme:

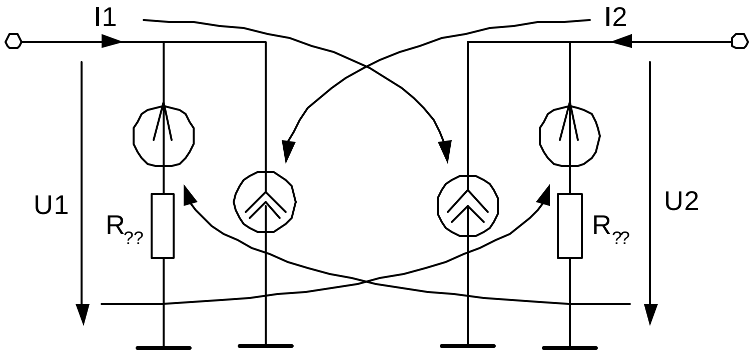

Fig.2. The linking mechanism

By each iteration, across (U_1) and through (I_1) variables are transmitted from server to client and from client to server. Then these variables are calculated on both sides and transmitted again. There is no difference between a server model and a client model circuit representation, but the server model is initiating the communicatio n process.

When auto adjustment of Rst is absent, the simulation will process by 1 iteration at each time step. It is useful for linear circuits.

When auto adjustment of Rst is present only at the first step, the simulation will process by 1 iteration at each time step (except for the first time step, when Rst will be adjusted). It is useful for working with an equipment.

When auto adjustment of Rst is present at each time step, the simulation will process by the number of iterations, required to achieve the specified tolerance. At each time step, after given number of iteration the auto adjustment of Rst will be made. This mode is useful for simulating all circuits with linear elements with the highest precision.

TESTBENCH

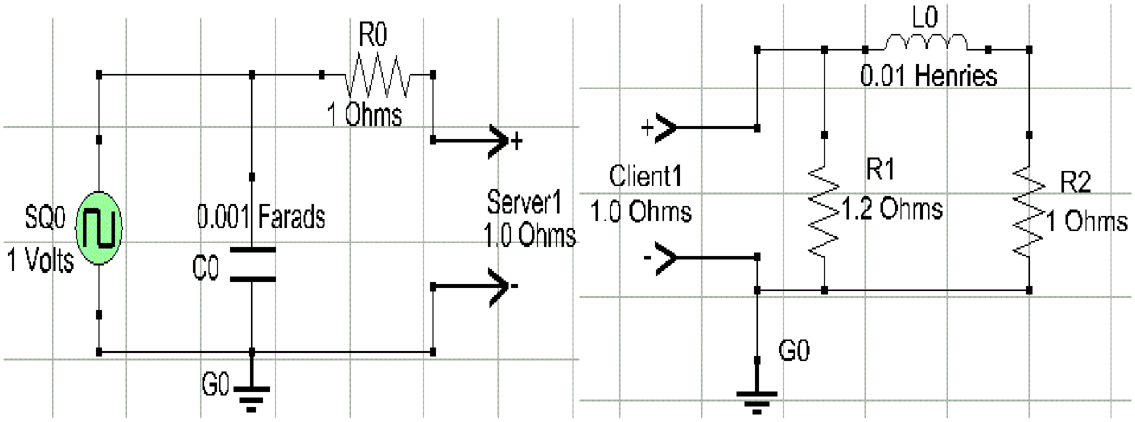

Fig 3. Test Bench for the concurrent simulation models

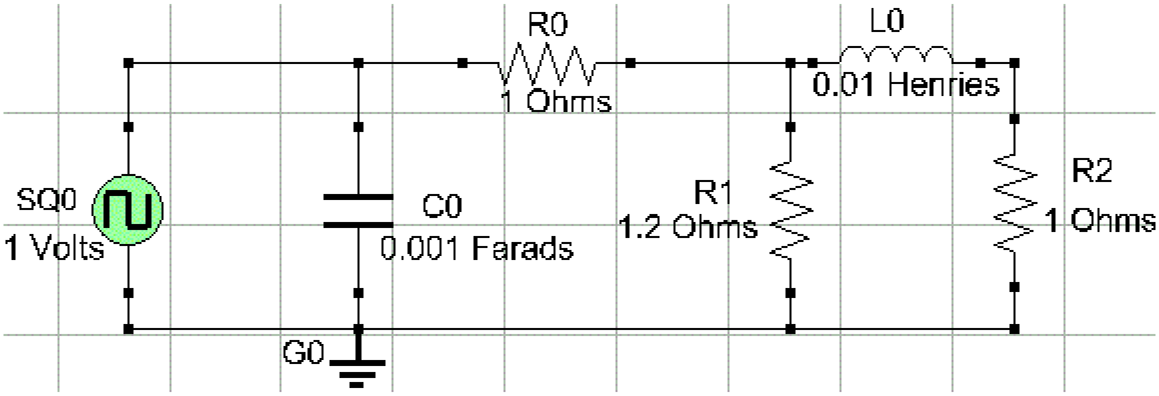

Fig.4. Test bench for verifications of results

VERIFICATION

Put two circuits into two sessions of VTB, as shown in the figure 3. The frequency of the SS0 source is 10 Hz. Set time step 0,0001 second and simulation stop time the same at both sides. Set the following parameters for the concurrent simulation models:

- initial value of the stabilizing resistance Rst=1.0 Oms (as shown in the fig.3);
- auto-adjustment of Rst – after 4 iterations;
- interVTB communications using RPC, the same endpoint and protocol at both sides (Server0 and Client0), the ServerName for the Client0 is the name of the computer, where the Server0 is running.

Set output method at both sides to data file (for example, OutputFile1.xls and OutputFile2.xls).

At first, run the simulation at the server side. Then, run the simulation at the client side. Wait for the simulation results are prepared.

Then, compare obtained results with results of simulation the circuit in the fig.4. All elements in the circuit at the fig.4 have the same parameters as the corresponding elements in the circuit at the fig.3 with the exception of the circuit at the fig.4 is not broken into two parts (therefore, there are no linking models need).

For example, compare the voltage across R0 in both cases:

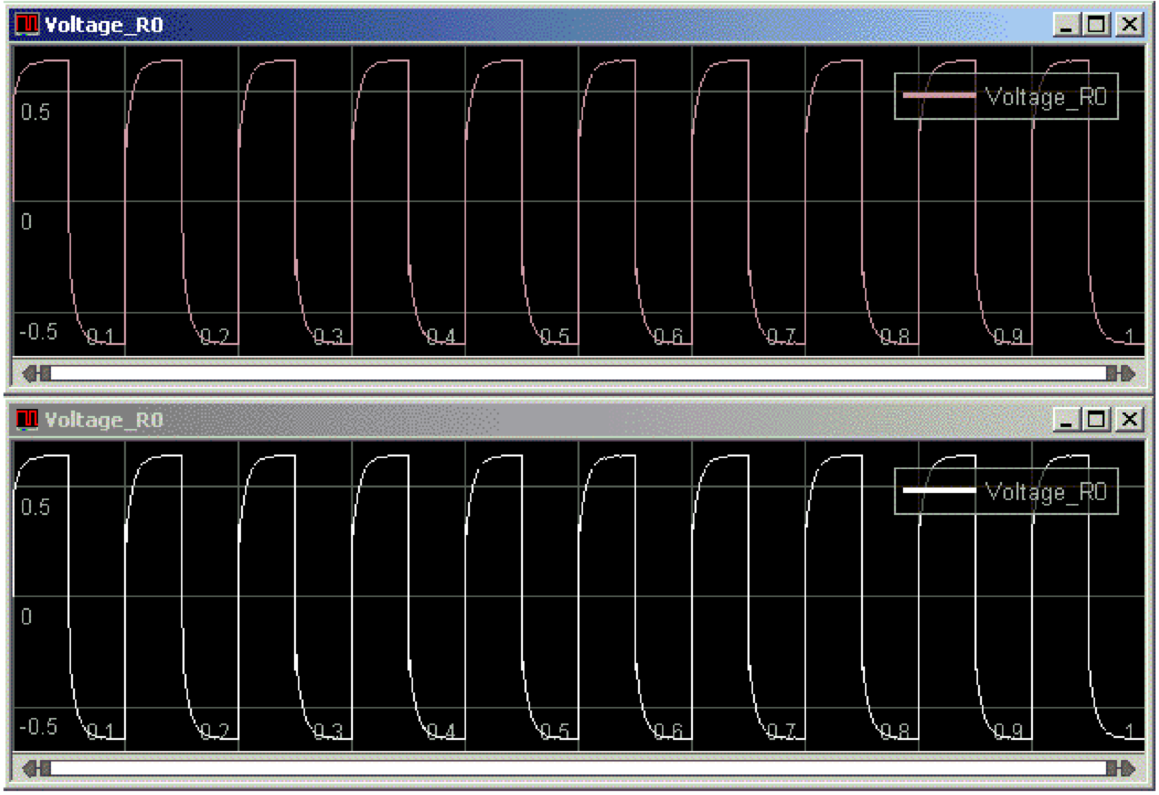

Fig.5. The voltage across R1 in the circuits at the fig.3 and at the fig.4

As you see in the fig.5, the result of simulation in both cases is the same. The difference does not exceed 7e-9 volts by default (as set for the Client0 and the Server0 elements).

Similarly we can verify the client part of the circuit. Compare the voltage across R1 in both cases:

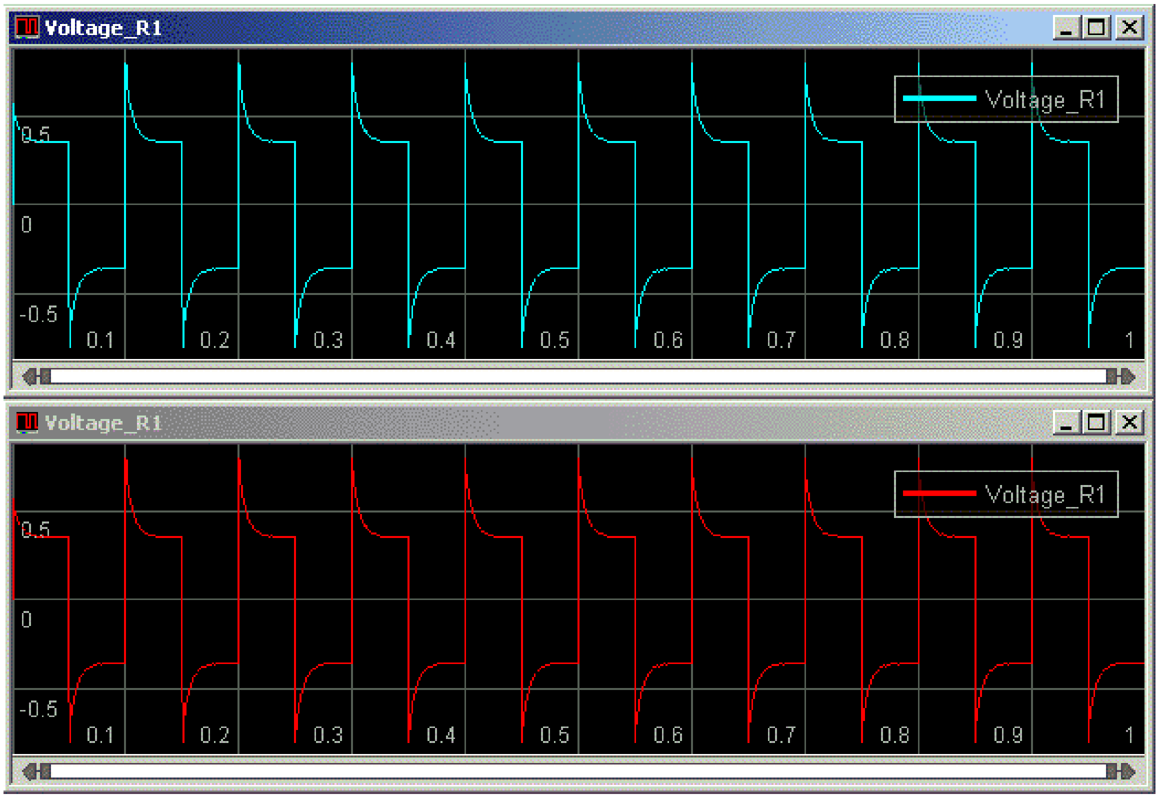

Fig.6. The voltage across R3 in the circuits at the fig.3 and at the fig.4

As you see in the fig.5 and in the fig.6, the result of simulation in both cases is the same in both parts of the circuit (in the server circuit and in the client circuit).

It is interesting to look at the changing of the stabilizing resistance at the Client0 and the Server0 elements. It is shown in the fig.7.

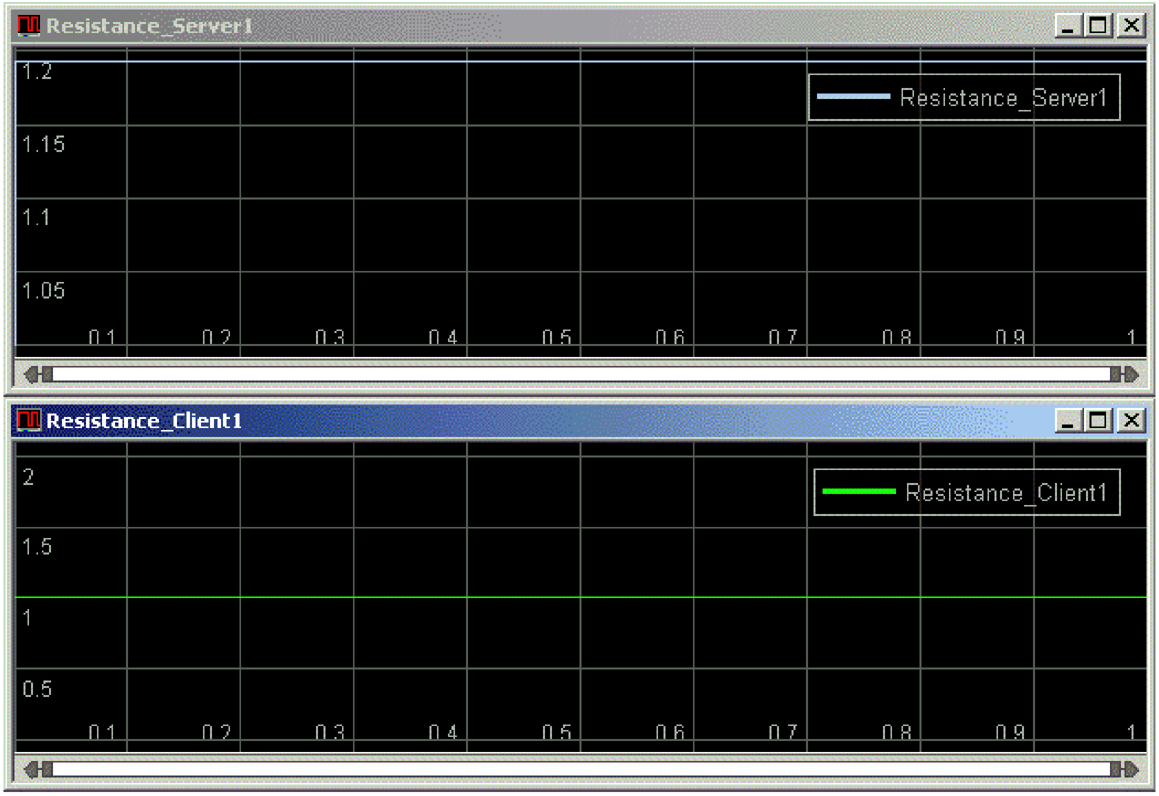

Fig.7. The changing of the stabilizing resistances and the current through D0.

As you can see, the stabilizing resistance is as adjusted once (at the first time step) and then remains constant over all the time of simulation. For all linear circuits it behaves similarly. Note, that the stabilizing resistance of the Client0 element is not adjusted at all. It is because the client part of the circuit is passive (has no sources), and its stabilizing resistance does not have an influence upon the result of simulation.

It is also interesting to look at the number of iterations and to observe, how it corresponds with the changing of a voltage, for example, of the voltage across R2. It is graphically shown in the fig.8.

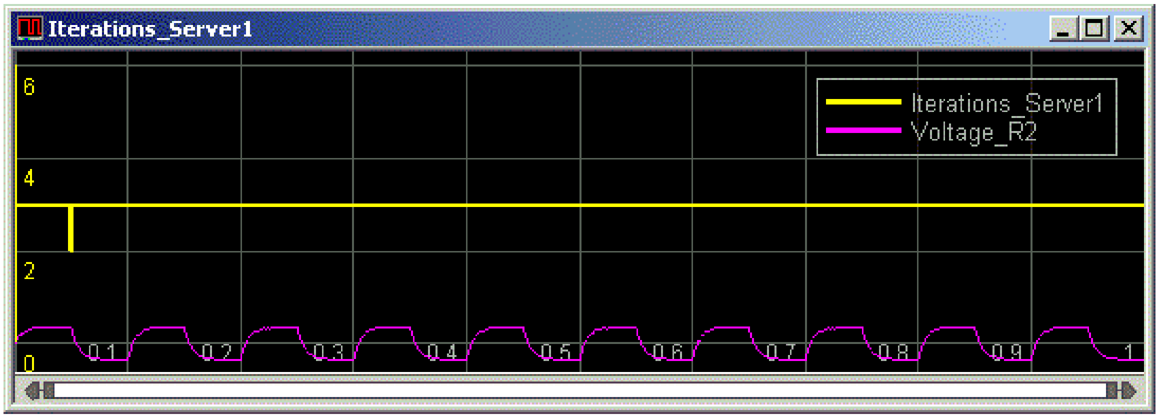

Fig.8. The number of iterations and the voltage across R2

As you can see at the fig.8, the number of iterations is equal 6 at the first time step, and then comes down to 2 or 3. It is because at the first time step the stabilizing resistance is not adjusted, and only after 4 Newton iterations it's adjusting. Then, when it is adjusted, then number of Newton iterations remains 2 or 3 for all linear circuits.