WCT Tutorial for the Time Reversal Simulation

Wavenology, Inc.
Aug., 2012
Introduction

The time reversal method utilizes the reciprocity of wave propagation in a time-invariant medium to find the shape and the location of an object.

The focusing quality in the time-reversal method is decided by the size of the effective aperture of transmitter-receiver array. This effective aperture includes the physical size of the transmitter-receiver array and the effect of the environment. A complicated background will create so-called multipath effect and can significantly increase the aperture of transmitter-receiver array.

In this tutorial, we will demonstrate how to simulate time-reversal cases in Wavenology EM package.
Table of Content

• Basic setup scheme
• Transmitter & Receivers
• Result verification by Snapshot
• Demo Cases
  – 1) detect a sphere behind walls
  – 2) detect an object in a chamber by antenna array

Note: due to the large space requirement (large aperture size) for time-reversal cases, it is suggested to use 64bit OS to run time-reversal simulations.
Basic Setup Scheme

The basic idea of time-reversal method is based on the scattered fields from the induced current on target. A time-reversal simulation will include following steps:

1) Simulate two cases to obtain the transient fields on receivers. These two cases are almost the same. The only difference is one case has the target (total field case). Another has not the target (incident field case).

2) Setup transmitter in two cases. It can be ideal dipole source or transmitting antenna.

3) Setup receiver array in two cases. The receiver can be point observer or receiving antenna.

4) Calculate the transient scattered field from the data obtained from these two cases.

5) Build a new case (reverse-radiation) based on the incident field case. But this case will not include the original transmitter. And the original receivers will be converted to be transmitters. The transient excitation pulse on the new transmitters comes from (4).

6) Define snapshot (recommend 2D face snapshot) to record how the EM wave propagates in the simulation case.

7) Run the case and check the result (transient snapshot result).
Transmitter & Receivers

For time-reversal cases, the transmitter can be ideal dipole source or antenna. The receiver can be ideal point observer and antenna.

For point receiver in incident & total field case, user need to convert the receiver to ideal dipole source in the reverse-radiation case.

For antenna in incident & total field case, user need to convert the receiving port on the antenna to transmitting port in the reverse-radiation case.

Wavenology EM package provides menus to simplify the conversion of point receiver-to-dipole source, and loading multiple excitation signals. The demo cases show the detail operations on these menus.
Because the time-reversal method is used to locate the target. Therefore, the simulation result will be verified by checking whether there is an energy focus on the target area.

This kind of focus can be easily distinguished by a transient 2D field snapshot.
Case (1): Detect a PEC Sphere behind a Wall

This case shows how to setup a time-reversal case by ideal dipole source(s).

Note:
1. Due to the time-reversal is an approximate method, it cannot focus on a small target. It is suggested that the target size is larger than one wavelength.
2. The focus quality of the time-reversal method depends on the aperture size of the re-radiation transmitter array, i.e., larger size of the array can obtain better focus.
3. The clutter between the re-radiation transmitter array and the target will significantly improve the quality of focus. If there is not clutter between the re-radiation transmitter array and the target, sometimes there is not focus.
In this case, we will use one dipole source and eleven point observers to work as transmitting system.

- **Ideal dipole**
- **Observers**
- **Ideal dipole with loaded excitation pulse**

Transmitter & receivers in the **Incident field & total field cases**

Transmitters in the **reverse-radiation case**
Steps:

1. Create a new folder as single_wall_single_target. Then use WCT GUI create a new project, save it under the sub-folder tot with name 1.

2. Define a new material: cement
3. Define the target: PEC sphere (unit: mm)
4. Define clutter: wall (unit: mm)
The position of *Sphere* & Wall is shown in the following Figure.
5. Setup project unit, background, boundary, excitation pulse, mesh and simulation time.

For the unit, we use default setting. The background material is default material, air.

In this case, we use 0\textsuperscript{th} order BHW wideband signal. And we set up the size of sphere about one wavelength of the frequency which has the maximum energy in this wideband excitation. 30 cm sphere -> 1\lambda\approx30 cm\rightarrow f_{\text{max\_energy}}\approx1 GHz.

In our setup, we set $f_{\text{max}}=2.76$ GHz.
Sampling density is set to be 14 points per wavelength.

Because for time-reversal case, we only want to check the transient result. So, we manually set up the simulation time window as 30 ns.

For the FDTD method, a high accurate scattered field can be obtained by incident & total field cases if these two cases use the same $\Delta t$ and mesh.

Here, we set $\Delta t$ as 0.0085 ns.

![Edit Existing Source dialog box](image)
7. Setup point observers.

Eleven observers in a line with a distance of 120 mm. Recording E field only.

Obv_001 is for monitoring only. It is not a part of reverse-radiation array.
8. Simulate the case to obtain the total field on observers.
9. Setup a case to obtain the incident field on observers.

   9.1 Create a new sub-folder under single_wall_single_target as inc. Use SaveAs function to save the total field case to inc sub-folder with name 1.

   9.2 Remove PEC sphere, or change the material of sphere as background material air.

   9.3 We need to let this case has the same mesh as the total field case. But due to the sphere does not exist. If we use Automatic meshing method, we cannot generate the same mesh as the total field case. Thus, we need to load the mesh directly from the total field case.

   Switch to “User defined” mesh option, press “Load” button

   In file dialog, goto single_wall_single_target\tot\1_res, there is a file “1_grid.m”, which has the mesh information of the total field case, load this mesh into this incident field case.

Note: WCT GUI generate the case mesh automatically after the simulation or preprocessing.
10. Simulate this incident field case to obtain the incident field on observers.
11. Calculate the scattered field from the total field case and the incident field case.

\[ E_{\text{sc}} = E_{\text{tot}} - E_{\text{inc}} \]

\( E_{\text{tot}} \) is the total field, from the result on the observers in the total field case. \( E_{\text{inc}} \) is the incident field, from the result on the observers in the incident field case.

In this case, the \( z \) polarized dipole source makes the \( z \) component in the case is the dominant component. Therefore, in our time-reversal case, we will use \( E_z \) to reverse-radiate only.

The transient received signal on receiver can be obtained by two methods:
1) Export from 2D result canvas.
2) Directly use the simulation result data files.
A. Export Data from 2D result canvas. For example, we will export Ez on all receivers.

Double click this tree-node

On 2D canvas, right click mouse to get this menu. Export all data curves.
Data format of export data file, it is compatible for *Matlab* ASCII data file.

%Wave Computation Technologies simulation waveform data, version 3.0
%created at 08/23/12 21:55:00 in GMT
%===================================
%Title    Observer Ez transient
%X-Lin     1e-009   Unit:ns
%Y-Lin     1 Unit:V/m
%Sub-Title Obv_(01)
1766 1766 %Size Nx=Ny
0.0000000e+000 0.0000000e+000
1.7000000e-002 0.0000000e+000
3.4000000e-002 0.0000000e+000
5.1000000e-002 0.0000000e+000
6.8000000e-002 0.0000000e+000
8.5000000e-002 0.0000000e+000
1.0200000e-001 0.0000000e+000
1.1900000e-001 0.0000000e+000
1.3600000e-001 0.0000000e+000
1.5300000e-001 0.0000000e+000
1.7000000e-001 0.0000000e+000

%Sub-Title Obv_(02)
1766 1766 %Size Nx=Ny
0.0000000e+000 0.0000000e+000
1.7000000e-002 0.0000000e+000

How many data (rows) in this curve + real data. 1\textsuperscript{st} column is time, 2\textsuperscript{nd} column is magnitude. For this curve, there are 1766 recorded data. It means the 1766 rows after the line “1766 1766 %Size Nx=Ny“ is the real data for curve 1.

2\textsuperscript{nd} curve, same format as curve 1
B. Directly use the simulation result data files. The simulation result for each case has following structure:

*x_res* is simulation result folder. (*x* is project name)

The sub-folder “*observers*” has all results for observers
Data format of simulation data file, it is compatible for *Matlab* ASCII data file.

%Wave Computation Technologies simulation waveform data, version 1.0 ::
%Time (ns)
%frames number
12
%frame start
0
%frame end
3.0005e-008
%frame step
1.7e-011
%frame length
1766
0.0000000e+000
0.0000000e+000
0.0000000e+000
... 1766 rows for the first curve.
0.0000000e+000
0.0000000e+000
0.0000000e+000
0.0000000e+000
... 1766 rows for the second curve.
0.0000000e+000

Information for all curves.
All curves has the same length with the same uniform interval.

Note: the curve index is the same as that shown in the project tree.
Following matlab code shows how to calculate the Ez_sct from the simulation data files directly. (This matlab file for this code is single_wall_single_target\ sct_reverse_time.m)

clear all;

%%% load total field
Etot_tx = load( '\\tot\1_res\observers\1_rev_ez.txt' );
nFrame = Etot_tx( 1 );
t0 = Etot_tx( 2 );
t1 = Etot_tx( 3 );
dt = Etot_tx( 4 );
nLen = Etot_tx( 5 );
Etot_tx = reshape( Etot_tx(6:end), nLen, nFrame );

%%% load incident field
Einc_tx = load( '\\inc\1_res\observers\1_rev_ez.txt' );
Einc_tx = reshape( Einc_tx(6:end), nLen, nFrame );

%%% resampling
ND = 1;
Etot_tx = Etot_tx( 1:ND:end, :);
Einc_tx = Einc_tx( 1:ND:end, :);
dt = dt * ND;
nLen = length( Etot_tx(:, 1 ) );
t = [0:(nLen-1)] * dt;
t = t';

%%% make transient scattered field
Esct_tx = (Etot_tx - Einc_tx);
Esct_tx_1 = flipud( Esct_tx );   % reverse the result time

%%% assume you will create Z field only
fid = fopen( '\\Z_field.txt', 'w' );

fprintf( fid, '%Wave Computation Technologies simulation waveform data, version 3.0\n' );
fprintf( fid, '%created at 08/21/12\n' );
fprintf( fid, '%===================================\n' );
fprintf( fid, '%Title      User defined pulse\n' );
fprintf( fid, '%X-Lin      1    Unit: \n' );
fprintf( fid, '%Y-Lin      1    Unit: \n' );

for k = 1 : 11,
    tt = [ t Esct_tx_1(:,k) ];
    fprintf( fid, '%Sub-Title Obv_%02d\n', k );
    fprintf( fid, '%d\n', nLen, nLen );
    fprintf( fid, '%d\n', nLen, nLen );
    for j = 1 : nLen,
        fprintf( fid, '%g\n', tt( j, 1 ) );
        fprintf( fid, '%g\n', tt( j, 2 ) );
    end;
end;
fclose( fid );
12. Set up the reverse-radiation case.
   1) **SaveAs** the incident case to sub-folder “inc1” with name 1.
   2) delete the original source “source1”
   3) convert all observers to dipole source (except the observer Obv_001)

WCT GUI provides a menu to simplify this operation.
Right click mouse button on tree-node “Source”, there is menu item “Convert all observers to electrical dipole source”.

This is the converted source with z polarization.
4) Load excitations for all new sources.

WCT GUI provides a menu to simplify this operation. Right click mouse button on tree-node “Source”, there is menu item “Load Source Pulse from File”.

Select the file we create in Page 25, load the pulse for all sources.

This function uses the name of curve in the file to match the source name. WCT GUI will report how many curve has been successfully loaded.
By checking the source pulse, all source pulse become “User defined”
13. User can modify the computation size, mesh and time to reduce simulation time. But this is an optional setup, user can still keep all setups.

14. Define 2D face snapshot to record the result.

15. Simulate the case.
16. Check the result by watching the snapshot.
Change frame index to watch how wave propagate.
A strong focus on sphere can be seen at frame #1121
Case (2): Detect a Object in a Chamber by an Antenna Array

This case shows how to setup a time-reversal case by antenna array. We will demo how to locate a target in the chamber by time-reversal method.

The simulated chamber model composed by an PIFA Antenna Array

An reference physical Chamber composed by an Bowtie Antenna Array
The antenna array is built on a rectangular chamber with 5 PCB panels (the material of PCB board is FR4). The chamber is open at the +z direction.

The chamber size is $10 \times 10 \times 10$ cm$^3$, filled by a fluid($\varepsilon_r=5$, $\sigma=0.01$ S/m). There are 8 Planar Inverted F Antennas (PIFA) fabricated on each panel, totally 32 antennas in the chamber.
Because this chamber model is complicated. Thus, in this demo, we will not discuss how to build this model and just use this model.

Assume that we already have the chamber filled with fluid, we will show how to modify this model to simulate a time reversal-case.
Similar to Case (1), we need to build three cases: the total field case, the incident field case and the reverse-radiation case.

Steps:

1. Assuming we have the chamber model case, we will build the total field case firstly.
   - **SaveAs** the chamber case to folder “time-reversal-chamber\tot\” with name “pifa_chamber”.

2. Define material for the target and the clutter.
Material for clutter
3. Insert a cubic target in the chamber.
4. Insert four dielectric planes to work as clutter.

Four planes have been added.
5. Define the transmitting and receiving antennas. In this case, we use lumped port as the feeding port and receiving ports. Thus, there is one transmitting port and 31 receiving port.
transmitting port

receiving port
6. Set up simulation pulse, mesh density and time.

Here, we use 1\textsuperscript{st} BHW wideband pulse as the source pulse. Due to the antenna radiation freq. is about 2.7 GHz, we need to make the excitation pulse has a characteristic freq. around 2.7 GHz.
# Project Design

## Time Window [ns]
- **User defined**
- **Automatic**

- **End Time**: 20
  - **Energy decay [dB]**: -60
  - **Energy variation [dB]**: -30

## Delta Time [ns]
- **User defined**: 0.0002
- **Automatic**

## Receiver Recordings
- **Recordings times per**
  - 20
- **Recording Interval (Unit: # time steps)**
  - 26

## Snapshot Recordings
- **Recordings times per**
  - 10
- **Recording Interval (Unit: # time steps)**
  - 53
8. Simulate the case to obtain the transient total voltage on the receiving antennas.

Note: in WCT GUI, there is scattered voltage on the transmitting & receiving ports.

For receiving port, it has not excitation. Thus, the total voltage on port is the same as the scattered voltage.

For this time-reversal case, in order to distinguish the voltage on the receiving port in the two cases (with-target & without-target), we call the scattered voltage for with-target case is the total voltage, the scattered voltage for with-target case is the incident voltage, in this demo.
9. Setup a case to obtain the incident voltage on receiving antennas.
   9.1 Create a new sub-folder under “time-reversal-chamber” as inc. Use SaveAs function to save the total voltage case to inc sub-folder with name “pifa_chamber”.
   9.2 Change the material of the target as fluid (oil).
   9.3 We set this case having the same mesh as the total voltage case, through loading the mesh directly from the total voltage case. Detail information please refer to Page 18.

10. Simulate this incident voltage case to obtain the incident voltage on the receiving antennas.

11. Calculate the scattered voltage from the total voltage case and the incident voltage case.

   \[ V_{sc} = V_{tot} - V_{inc} \]

   \( V_{tot} \) is the total voltage, from the result on the receiving antennas in the total voltage case. \( V_{inc} \) is the incident voltage, from the result on the receiving antennas in the incident voltage case.

   The transient voltage on the receiving antennas can be obtained by two methods:
   1) Export from 2D result canvas.
   2) Directly use the simulation result data files.
A. Export Data from 2D result canvas.

Double click this tree-node and export the data

Exported data file format is described on page 22.
B. Directly use the simulation result data files.

The simulation result for each case has following structure:

- **x_res** is simulation result folder. *(x* is project name)

The sub-folder **"Lump_ports"** has all results for lumped ports, the scattered voltage has the file name **“x_lumped_port_sct_volt_tran.txt”**.

Data file format is described on page 24.
Following matlab code shows how to calculate the Vsct from the simulation data files directly. (This matlab file for this code is time-reversal-chamber\sct_reverse_time.m)

clear all;

%%% load incident field
Etot = load( './tot\pifa_chamber_res\lump_ports\pifa_chamber_lumped_port_sct_volt_tran.txt' );
nFrame = Etot( 1 );
t0 = Etot( 2 );
t1 = Etot( 3 );
dt = Etot( 4 );
nLen = Etot( 5 );
Etot = reshape( Etot(6:end), nLen, nFrame );

%%% load total field
Einc = load( './inc\pifa_chamber_res\lump_ports\pifa_chamber_lumped_port_sct_volt_tran.txt' );
Einc = reshape( Einc(6:end), nLen, nFrame );

%%% resampling
ND = 1;
Etot = Etot( 1:ND:end, : );
Einc = Einc( 1:ND:end, : );
dt = dt * ND;
nLen = length( Etot(:, 1) );
t = [0:(nLen-1)] * dt;
t = t';

Esct = Etot - Einc; %%% make transient scattered field
ids = [1 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 4 5 6 7 8 9 ];
%%% need mapping, because the export function of WCT use ASCII sequence. (Till current version 1.6)

for k = 1 : 32,
    tt = [ t flipud(Esct(:,k)*1e4) ];
    str = sprintf( 'pifa_chamber_user_lport_pulse_%0d.txt', ids(k) );
    fid = fopen( str, 'w' );
    fprintf( fid, '%Wave Computation Technologies simulation waveform data, version 3.0\n' );
    fprintf( fid, '%created at 11/06/10 04:21:37 in GMT\n' );
    fprintf( fid, '%Title User defined pulse\n' );
    fprintf( fid, '%X-Lin 1 Unit: \n' );
    fprintf( fid, '%Y-Lin 1 Unit: \n' );
    fprintf( fid, '%Sub-Title User defined pulse\n' );
    fprintf( fid, '%Size Nx=Ny\n', nLen, nLen );
    for j = 1 : nLen,
        fprintf( fid, '%g\n', tt( j, 1 ), tt( j, 2 ) );
    end;
    fclose( fid );
End;

Note: This code generate 32 data files. Each file is corresponding to the scattered voltage of one port.
12. Set up the reverse-radiation case.
   1) SaveAs the incident case to sub-folder “inc1” with name “pifa_chamber”.
   2) delete the original excitation port “5”
   3) convert all receiving ports to excitation port, set up the excitation pulse of port as “User defined” and load it from the data file.

(The code in page 48 generates the scattered voltage for each port, named as “pifa_chamber_user_lport_pulse_xx.txt”. xx is the name of port.)
13. Define 2D face snapshots to record the result.

<table>
<thead>
<tr>
<th>Plane Location (mm)</th>
<th>Name</th>
<th>Normal</th>
<th>Display Components</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lower Corner (x, y, z)</td>
<td>Name 1</td>
<td>Normal Z</td>
<td>Ex, Ey, Ez</td>
</tr>
<tr>
<td>Higher Corner (x, y, z)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Plane Location (mm)</th>
<th>Name 2</th>
<th>Normal X</th>
<th>Display Components</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lower Corner (x, y, z)</td>
<td>Name 2</td>
<td>Normal Z</td>
<td>Ex, Ey, Ez</td>
</tr>
<tr>
<td>Higher Corner (x, y, z)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
14. Simulate the case.

15. Check the result by watching the snapshot.

It can be seen, there is a strongest focus (frame 1650) at the position of target.