Substructure module 3 – OpenSees, MATLAB, and C++

Xu Huang

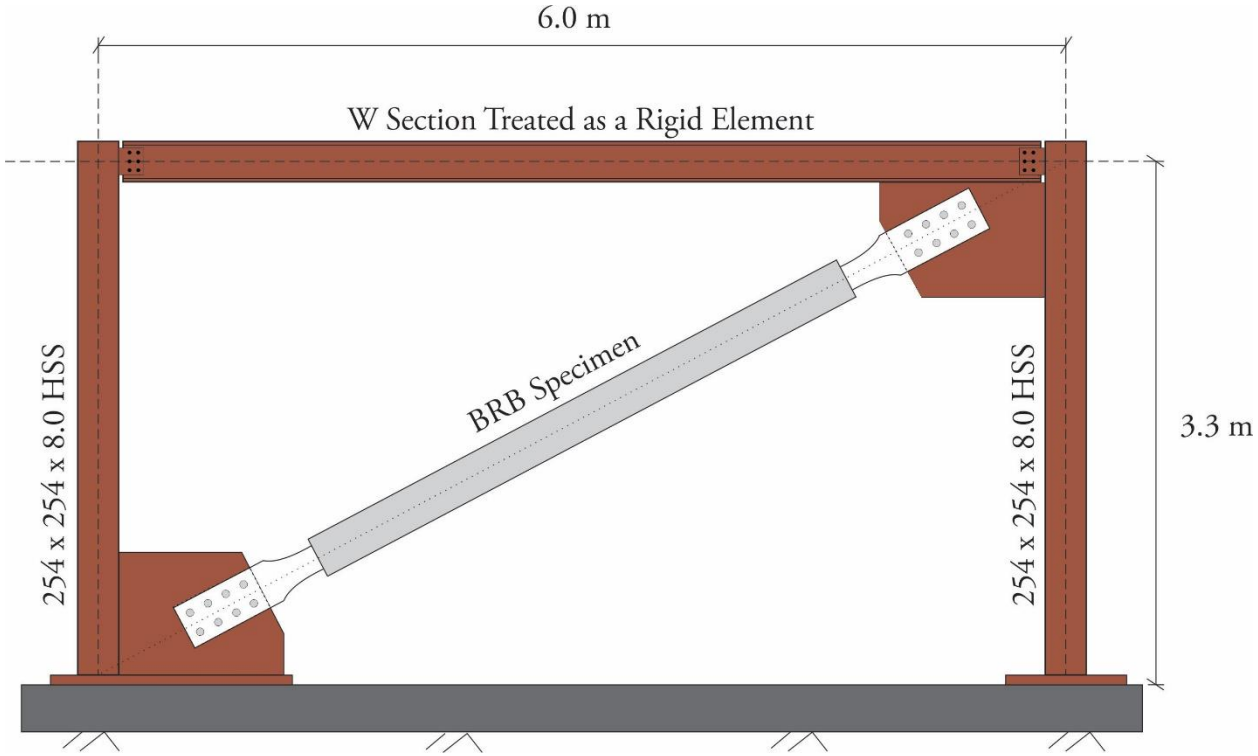
Civil Engineering
UNIVERSITY OF TORONTO

Outline

- **Example Structure**
  - Structural configuration
  - OpenSees Model
- **Simulation Method**
  - Decomposition
  - Integration module - OpenSees
  - Communication overview
- **Substructure modules**
  - OpenSees
  - MATLAB
  - C++

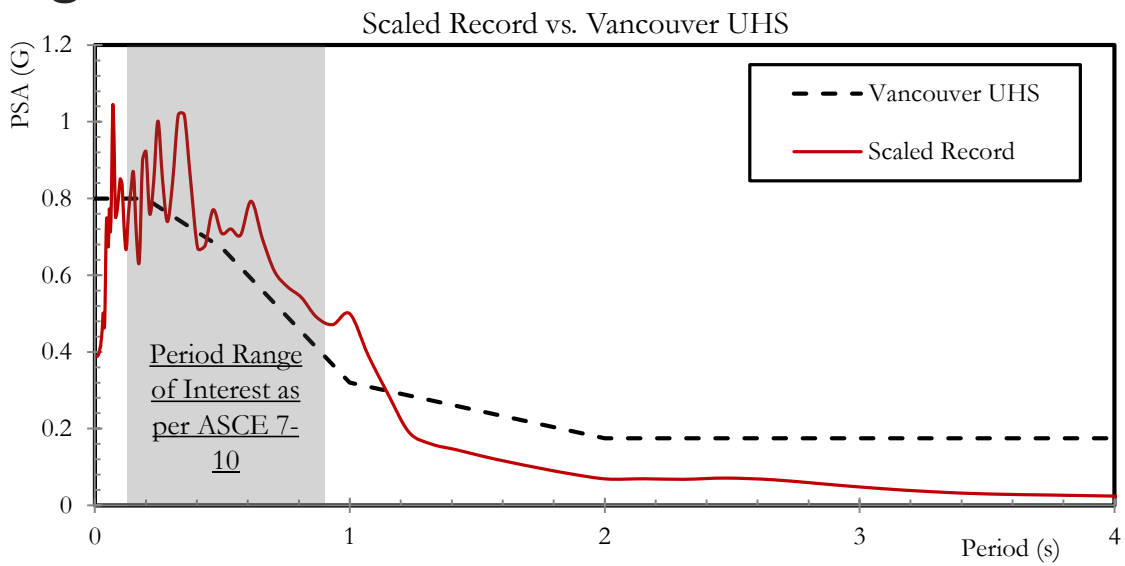
Example Structure

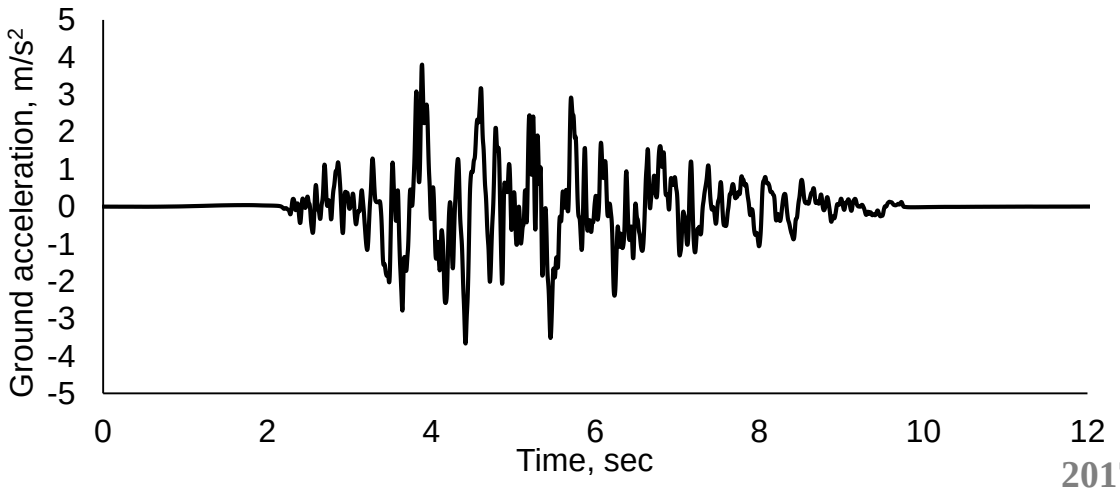
- **Structural Configuration**



Example Structure

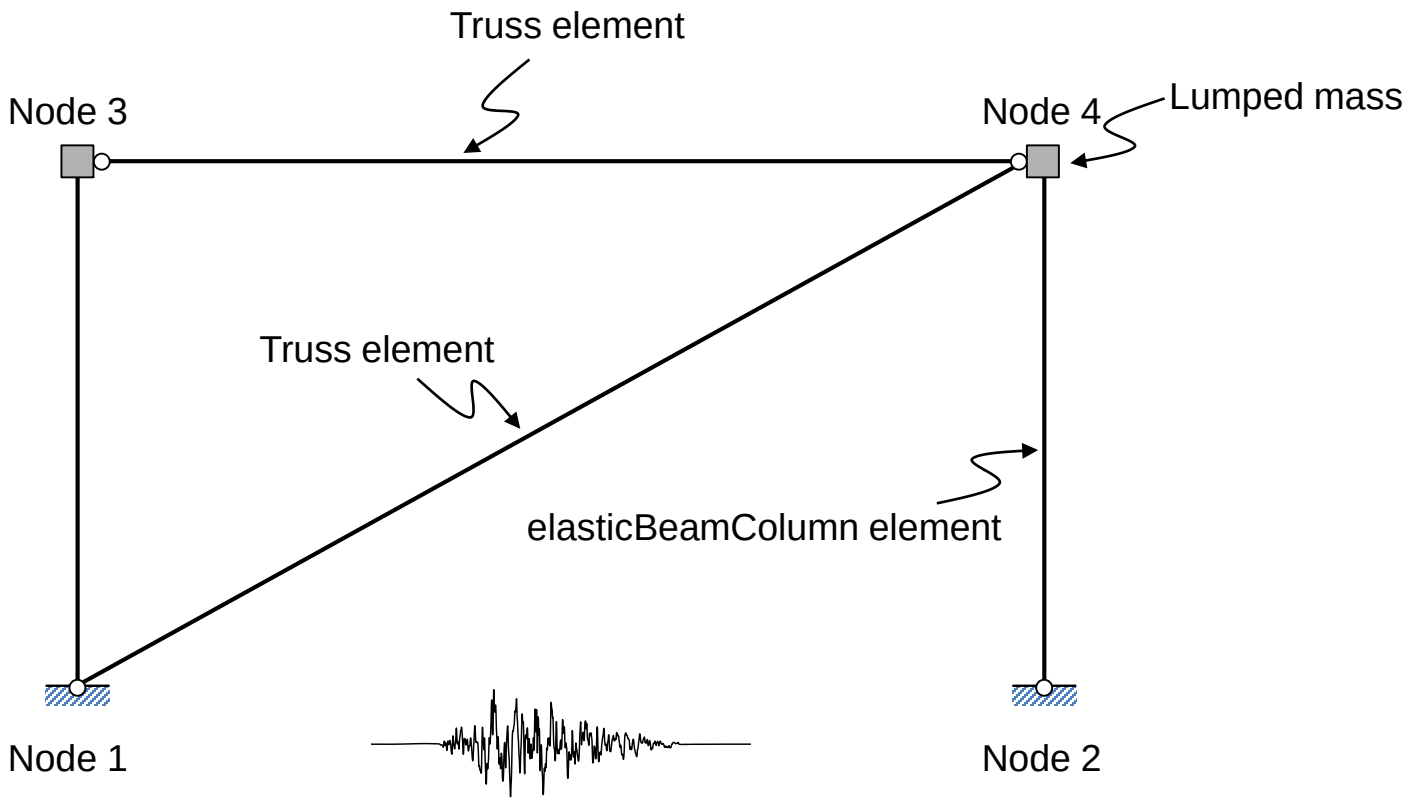
- Loading





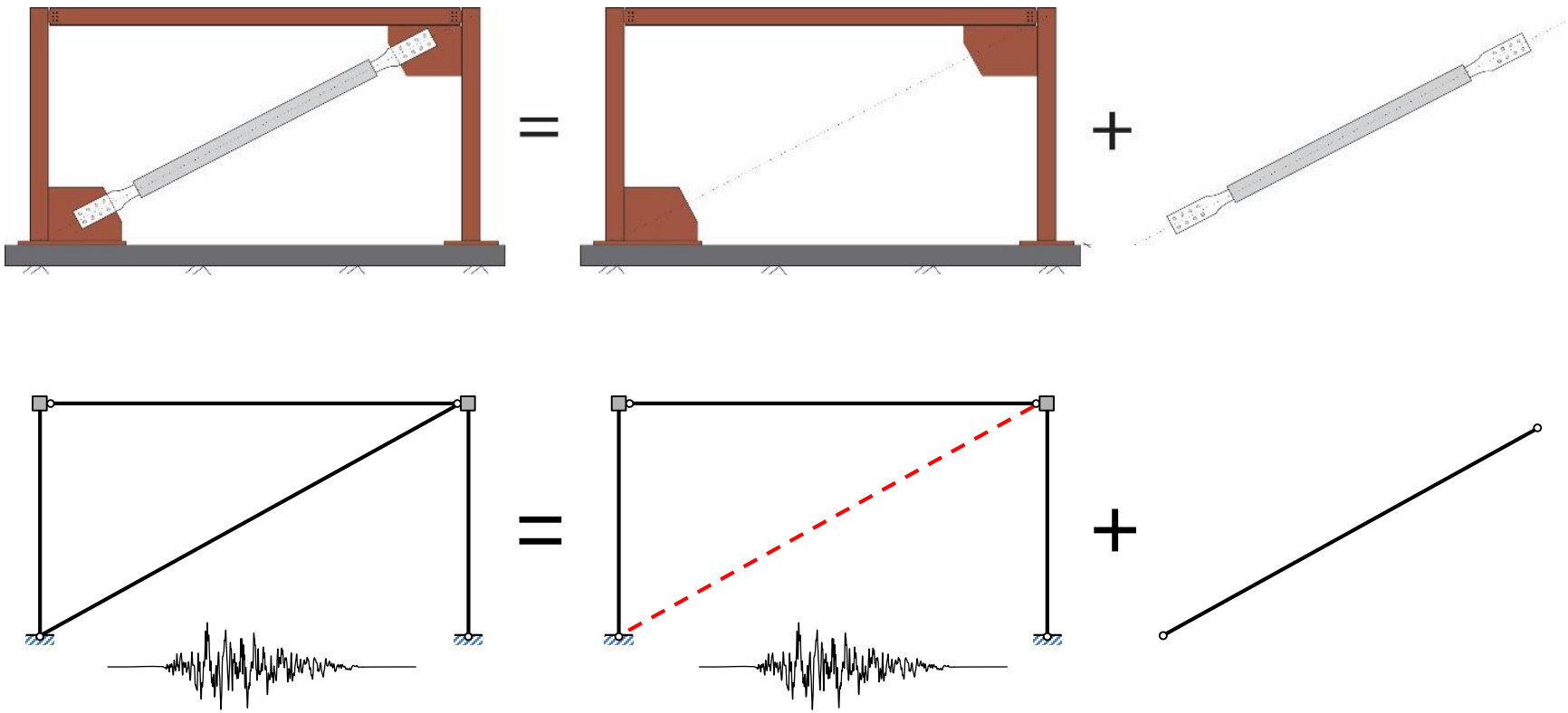
Example Structure

- **OpenSees model**



Simulation Method

- **Decomposition**

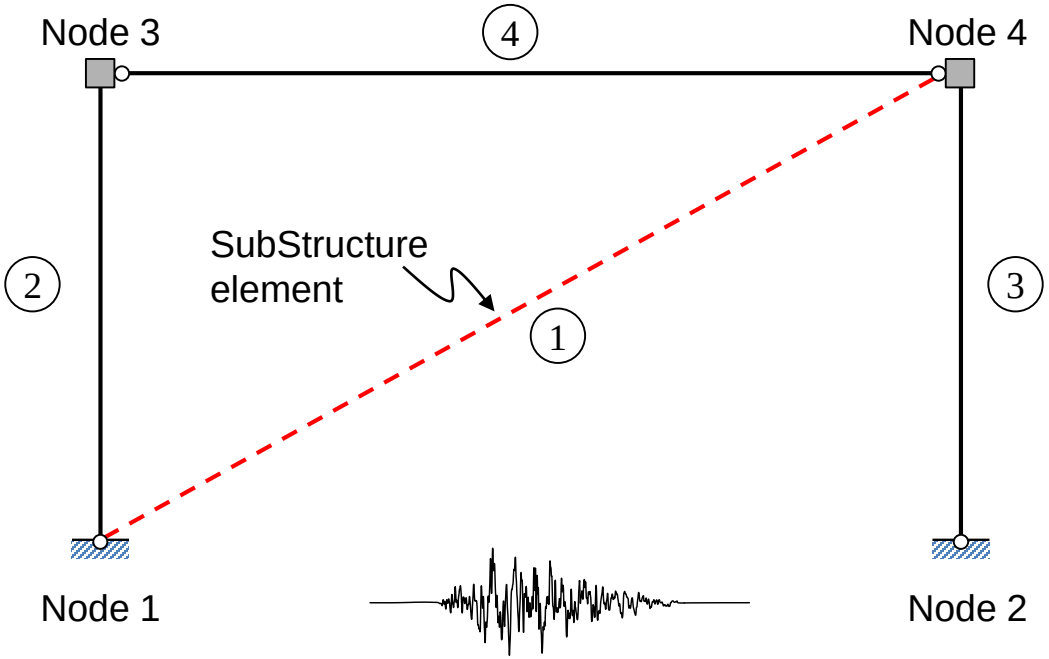


Complete model | Integration module | Substructure module

Simulation Method

- **Integration module - OpenSees**

$$m\ddot{u} + c\dot{u} + \sum_{n=2}^{4} r_n + r_1 = f(t)$$



Simulation Method

❑ Integration module – OpenSees

- Complete model

```
puts "Define Elements";

element corotTruss 1 1 4 1 ; # BRB defined as a truss element
element elasticBeamColumn 2 1 3 $Ac $E $Ic $TFrame; # Column Element Between Nodes 1 & 3
element elasticBeamColumn 3 2 4 $Ac $E $Ic $TFrame; # Column Element Between Nodes 2 & 4
element corotTruss 4 3 4 2 ; # Beam Element Defined
```

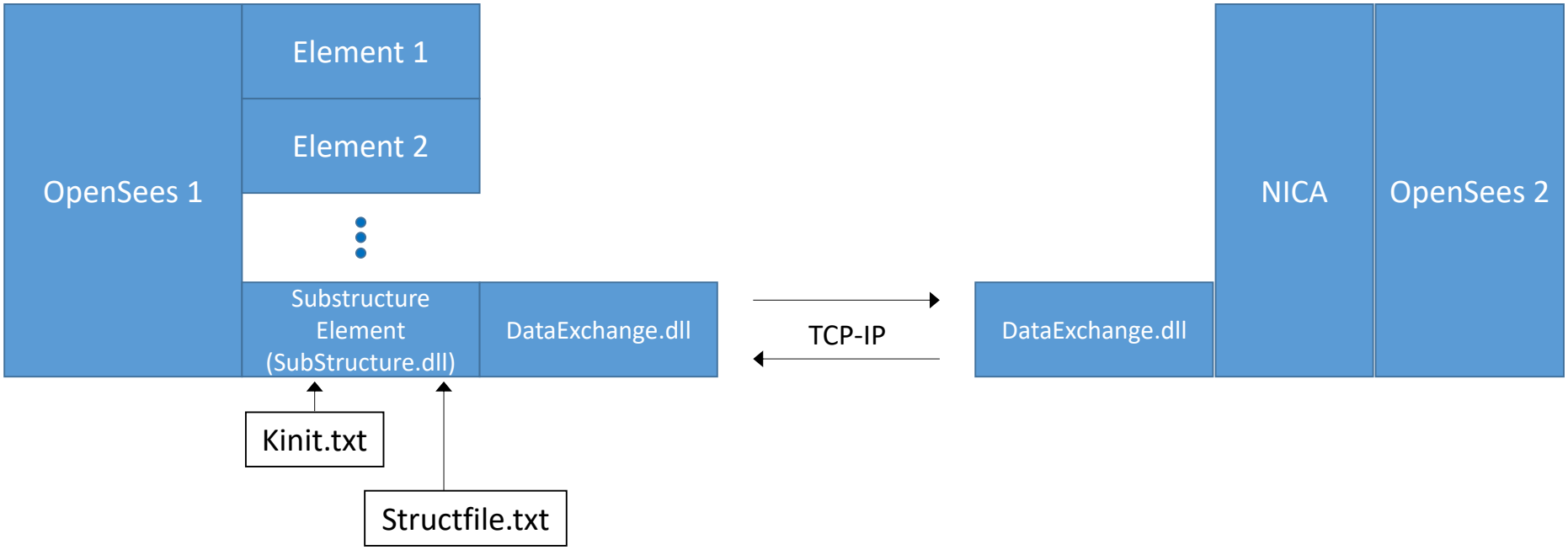
- Integration model

```
puts "Define Elements";

element SubStructure 1 -file Structfile.txt -Kinit Kinit.txt
element elasticBeamColumn 2 1 3 $Ac $E $Ic $TFrame; # Column Element Between Nodes 1 & 3
element elasticBeamColumn 3 2 4 $Ac $E $Ic $TFrame; # Column Element Between Nodes 2 & 4
element corotTruss 4 3 4 2; # Beam Element Defined
```

Day2: 9:00 – 9:45, Integration Module 1 – OpenSees, Pedram Mortazavi

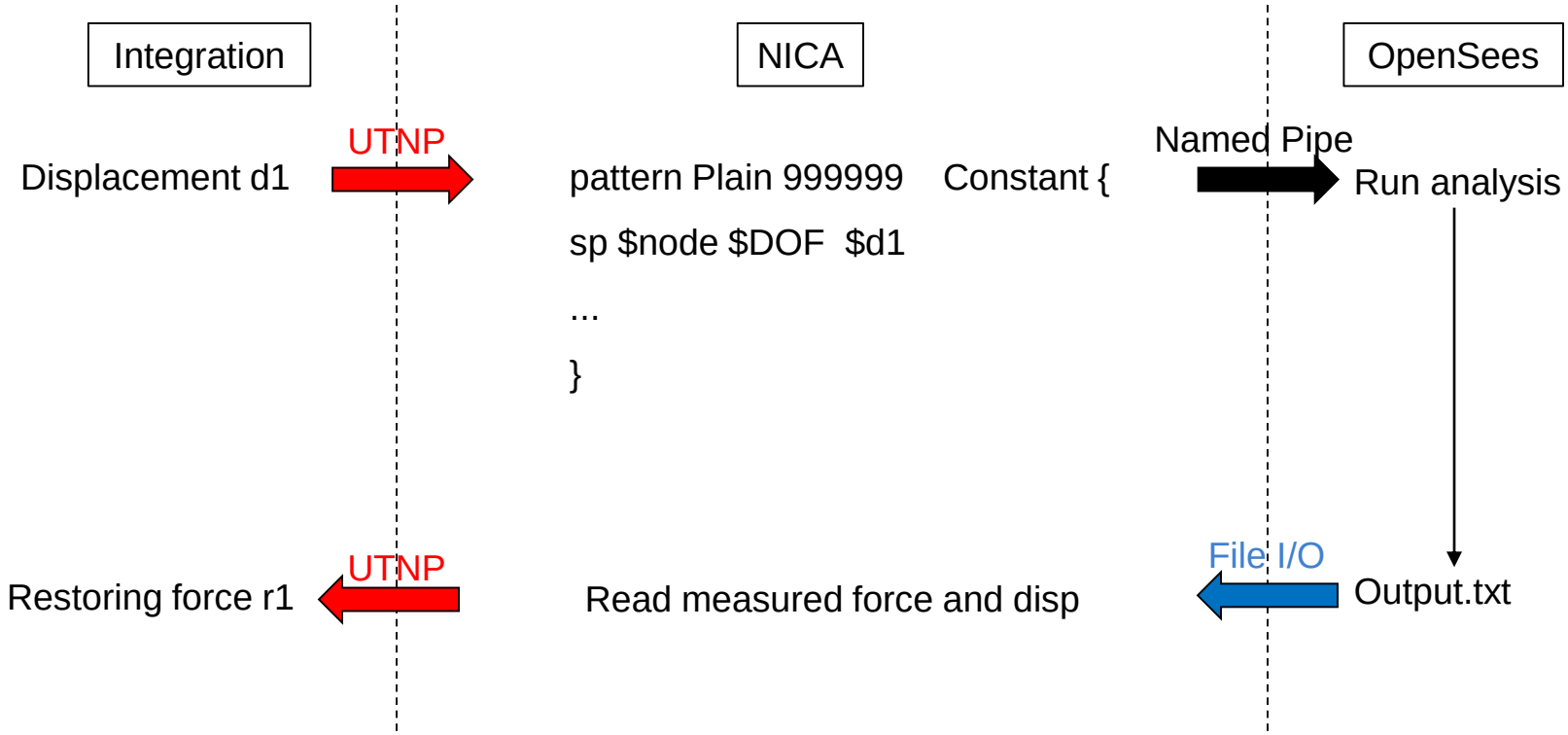
Simulation Method

- **Communication overview**
  - OpenSees to OpenSees



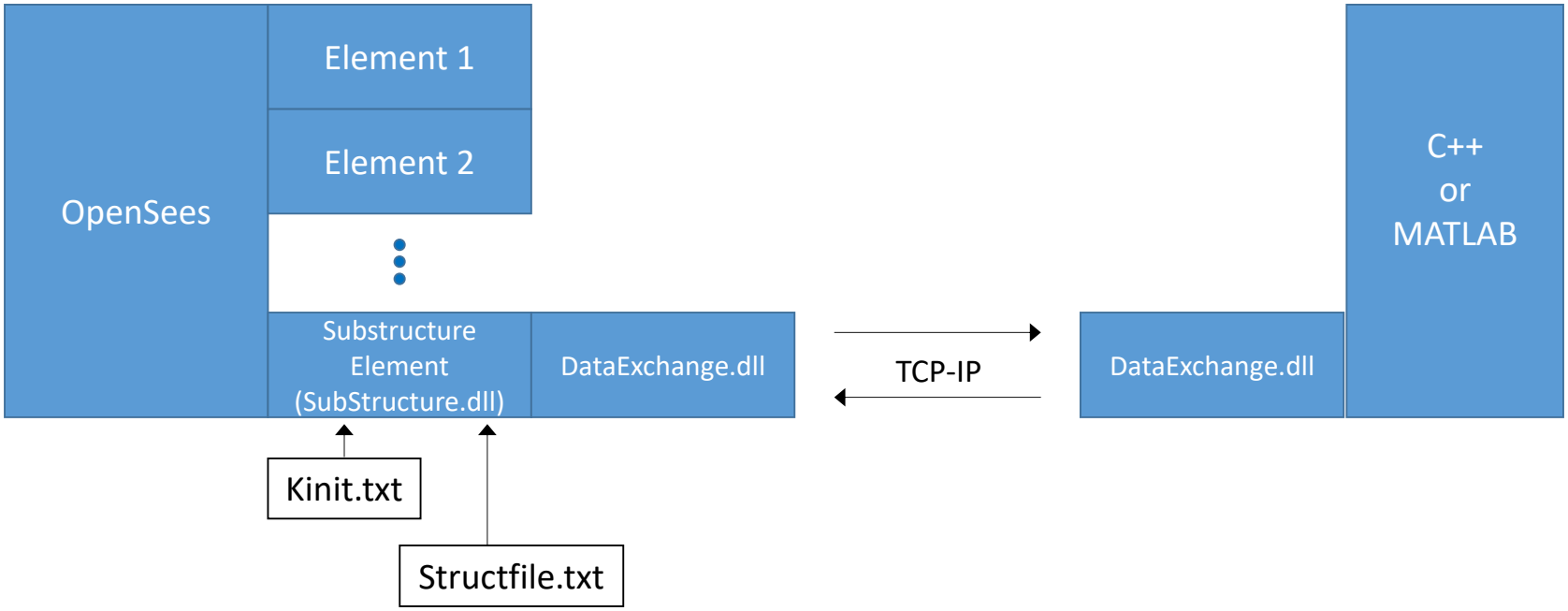
Simulation Method

- **Communication overview**
  - OpenSees to OpenSees



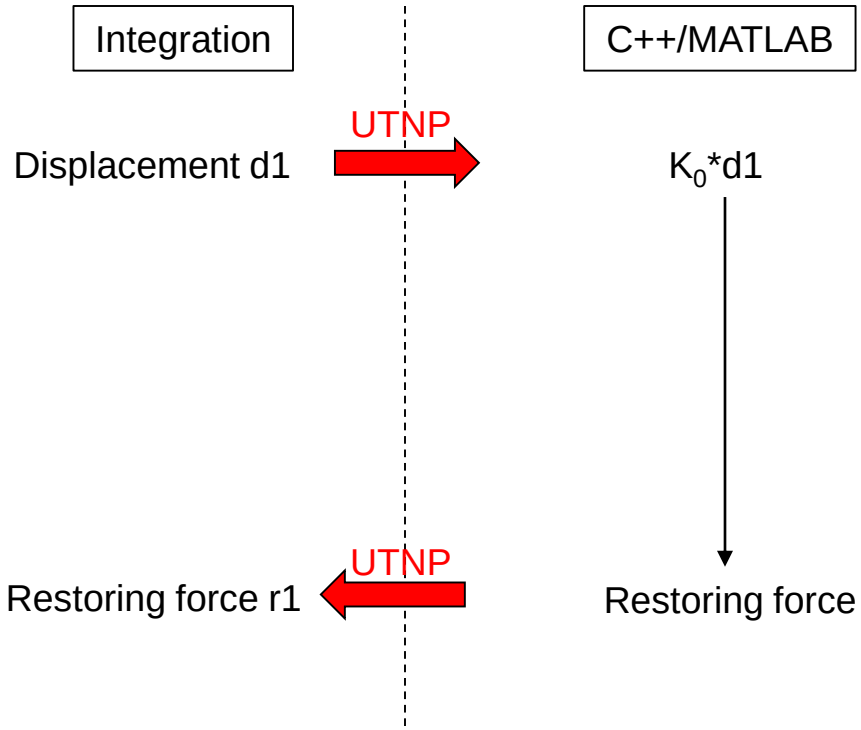
Simulation Method

- **Communication overview**
  - OpenSees to C++/MATLAB



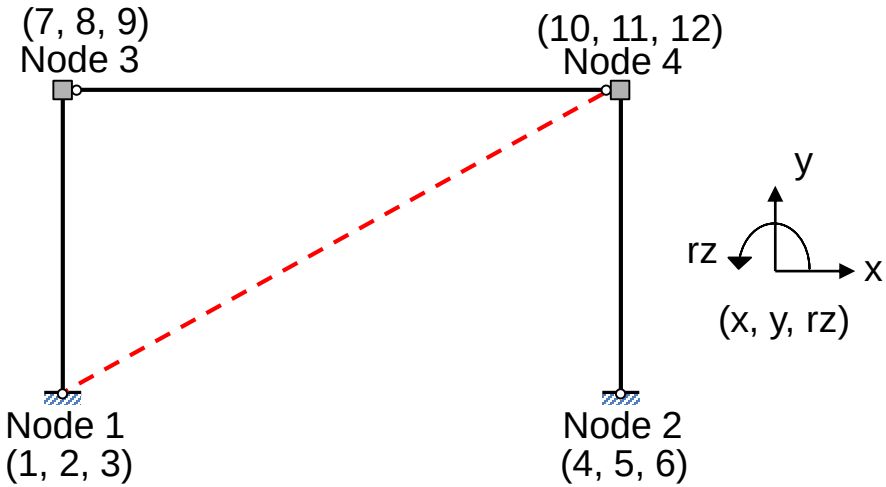
Simulation Method

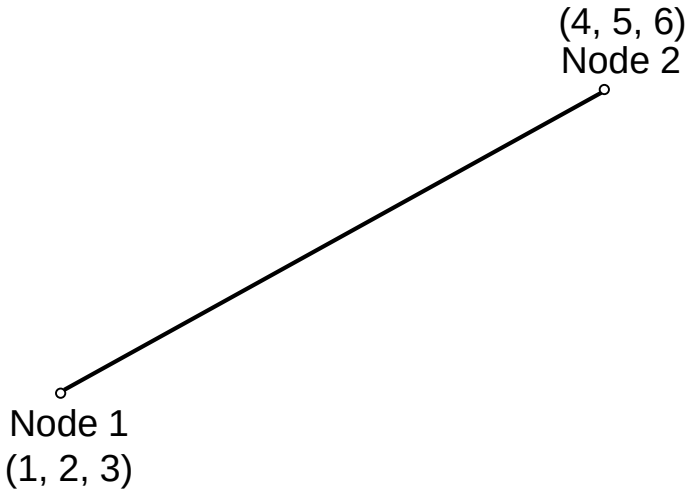
- **Communication overview**
  - OpenSees to C++/MATLAB



Substructure modules

- **Communication configuration**
  - Port number
  - Interface nodes and DOFs





Integration module			Substructure module	
Node 1	1	⟷	1	Node 1
	2	⟷	2	
	3	⟷	3	
Node 4	10	⟷	4	Node 2
	11	⟷	5	
	12	⟷	6	

Substructure modules

- **OpenSees**
  - Example folder

« UT-SIM ▸ deliverable ▸ OpenSees - OpenSees ▸ NICA

Search NICA

Name	Date modified	Type	Size
DataExchange.dll	2015-05-11 3:37 PM	Application extens...	13 KB
NICA.cfg	2017-03-25 2:26 PM	CFG File	1 KB
NICA.exe	2015-06-20 8:35 PM	Application	171 KB
Sub.S2(BRB).tcl	2016-11-20 9:49 AM	TCL File	5 KB
Units.tcl	2016-10-11 2:03 AM	TCL File	3 KB

Substructure modules

- **OpenSees**
  - NICA.cfg

NICA.cfg - Notepad

File Edit Format View Help

```
# Connection port to NICA
Port = 8090

# Control node numbers
# Note: The sequence of nodes should be consistent with Nodes in SimCor.
MDL_Node = [1 2]

# Model dimension
# Note: This tag is used only for OpenSees and Abaqus. Ignore for other analysis types.
#       In the current version, 2D 3DOF system and 3D 6DOF system are supported.
MDL_Dim = 2

# Effective DOFs in control point.
# Note: 1.The sequence of DOFs should be consistent with sequence of DOFs in UI-SimCor.
#       2.Use one line per each controlled node.
#       3.For 2D 3DOF model, use DOF 1, 2, 6 (not 1, 2, 3) for x, y, rz DOFs.  It is set as x,y,z,Rx,Ry,Rz
EFF_DOF =
1 1 0 0 0 0
1 1 0 0 0 0

# Model file name (include extension)
MODEL = Sub.S2(BRB).tcl
```

Ln 1, Col 1

Substructure modules

- **OpenSees**
  - Sub.S2(BRB).tcl

Constraints command

Numberer command

System command

Test command

Algorithm command

Integrator command

Analysis command

Analyze command

Sub.S2(BRB).tcl

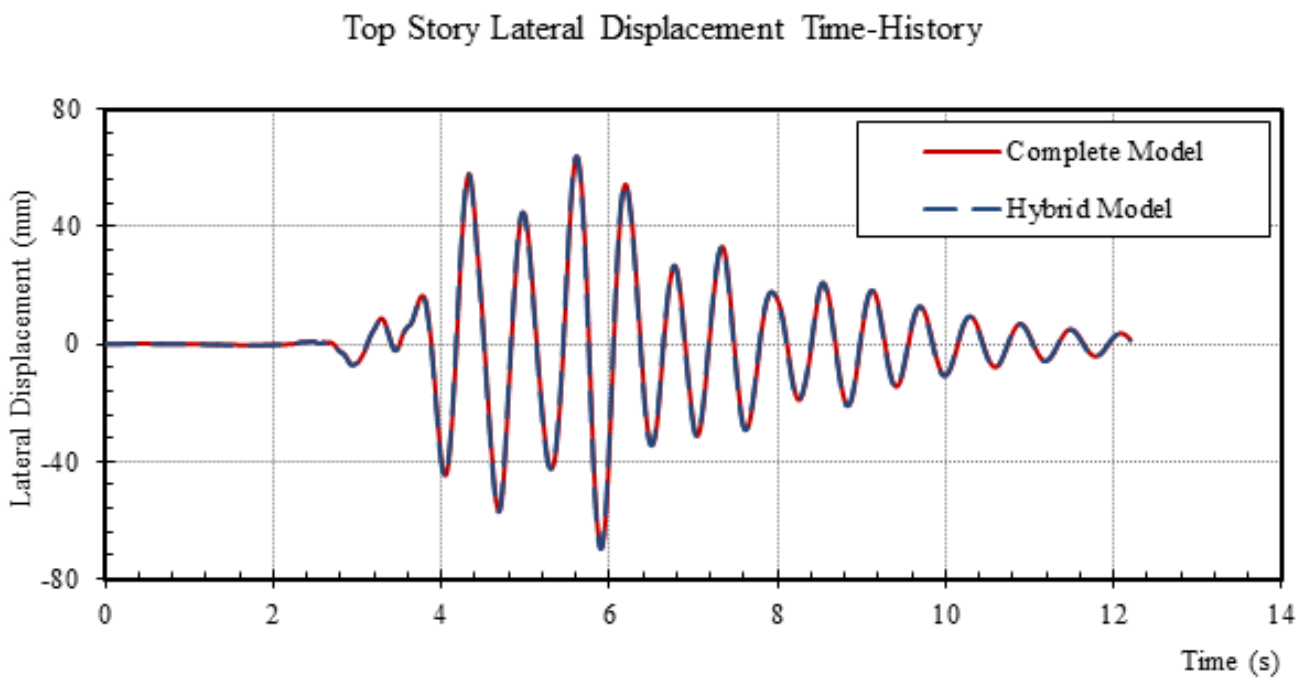
```
# ------------------------------------------------------------------------------------------
#   MODEL INPUTS
source Units.tcl;  # Source the Unit Conversion file

model basic -ndm 2 -ndf 3; # 2D model, 3 DOFs per node

set Lc [expr 3.3*$m]; # set the length of columns to 3.3 meters
set Lb [expr 6.0*$m]; # set the length of the beam to 6.0 meters
set E [expr 200000*$MPa]; # Modulus of Elasticity of Steel
set Fy [expr 300*$MPa]; # Yield Stress of Steel Material
set Lbr [expr { sqrt( $Lc*$Lc + $Lb*$Lb ) }]; # Length of the brace
set Abr [expr 1024*$mm2]; # Area of the BRB steel core

# ------------------------------------------------------------------------------------------
puts "Building Model Geomtry";
node 1 0 0;
node 4 $Lb $Lc;

# ------------------------------------------------------------------------------------------
puts "Defining Material/Sections";
set Pb [expr $Fy*$Abr]; # Brace Yield Force
set Kb [expr $E*$Abr]; # Brace stiffness in local coordinates
set b 0.01;
set R0 18;
set cR1 0.925;
set cR2 0.15;
set a1 0.0;
set a2 1.0;
set a3 0.0;
set a4 1.0;
set sigInit 0.0;
set matTag 1;
uniaxialMaterial Steel02 $matTag $Pb $Kb $b $R0 $cR1 $cR2 $a1 $a2 $a3 $a4 $sigInit;
section Uniaxial 1 1 P;   # BRB Section Defined

# ------------------------------------------------------------------------------------------
puts "Define BRB Element in HS";
element corotTruss 1 1 4 1; # BRB defined as a truss element
```

Substructure modules

- **OpenSees**
  - How to run simulation
    - Run NICA.exe. In NICA program window, it shows ‘waiting for connection’.
    - Run OpenSees integration module.
    - Once the connection between the integration module and substructure modules is established, the ‘Press Enter to continue’ message will appear in NICA command window. Click on the NICA command prompt and press Enter.

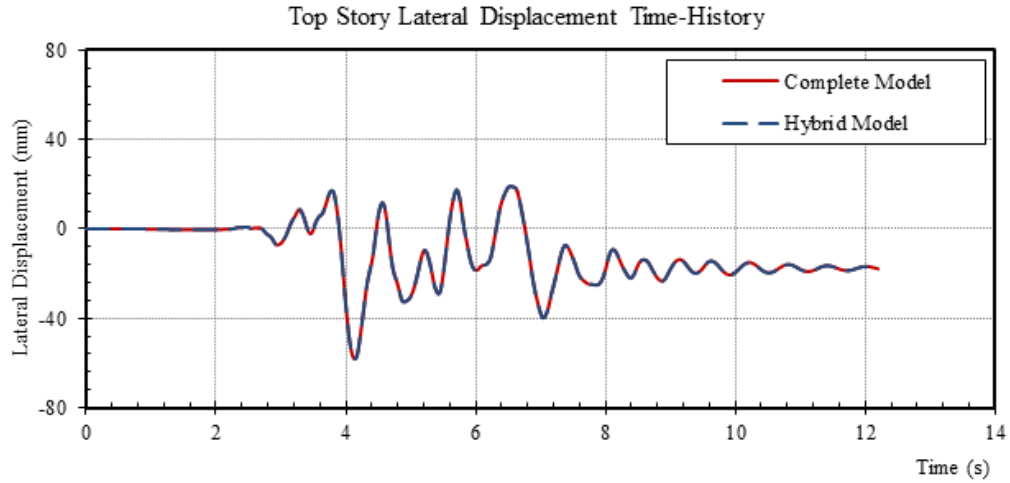
Substructure modules

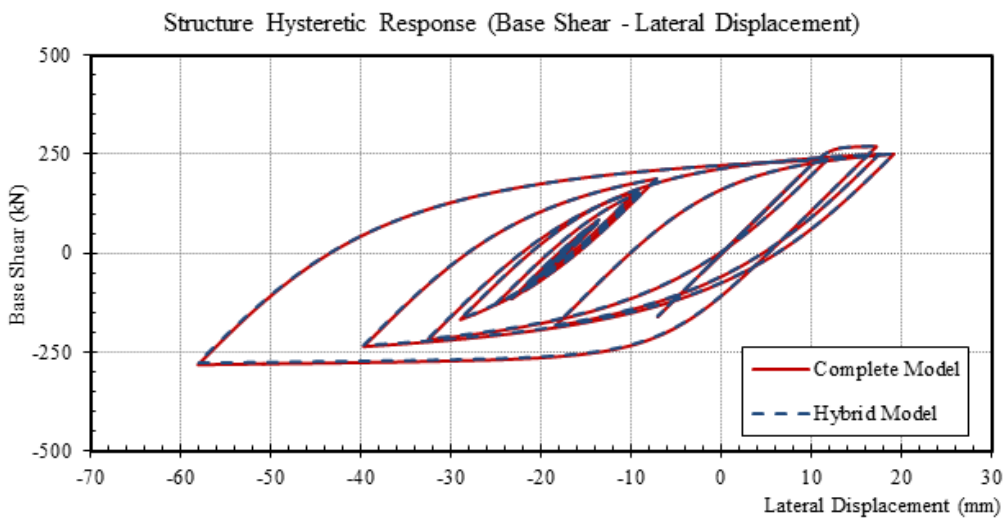
- **OpenSees**
  - Linear analysis



Substructure modules

- **OpenSees**
  - Nonlinear analysis

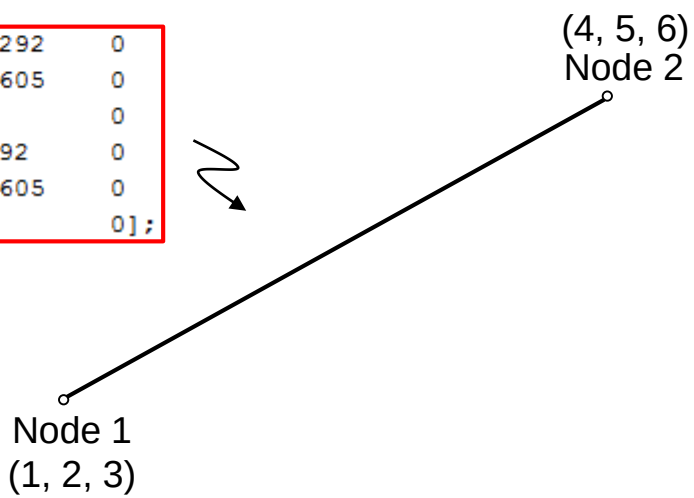




Substructure modules

- **MATLAB/C++**
  - Communication configuration (i.e. MATLAB)

```
clear all; close all; clc

loadlibrary('./DataExchange.dll', './DataExchange.h');

% define socket variables
PortNumber = 8090;
machineInetAddr = libpointer('cstring','0.0.0.0');

sockfd = 0;
flag = 2;

% define initial stiffness matrix

Kinit = [22.9621   12.62920    0 -22.9621   -12.6292    0
         12.6292    6.94605    0 -12.6292   -6.94605    0
          0         0          0   0         0          0
        -22.9621 -12.6292      0  22.9621   12.6292     0
        -12.6292  -6.94605     0  12.6292    6.94605    0
          0         0          0   0         0          0];
```



Substructure modules

- **MATLAB/C++**
  - Main loop (i.e. MATLAB)
    - Receive displacement

```
case Impose_TargetValues

    % calculate the size to be appended to the message header
    lens = calllib('DataExchange', 'indicator');

    %receive displacement from OpenSees
    rdata = libpointer('doublePtr', zeros(lens,1));
    calllib('DataExchange', 'RecvData', sockfd1, rdata, lens, TCP_IP);
    displ = get(rdata, 'value');
```

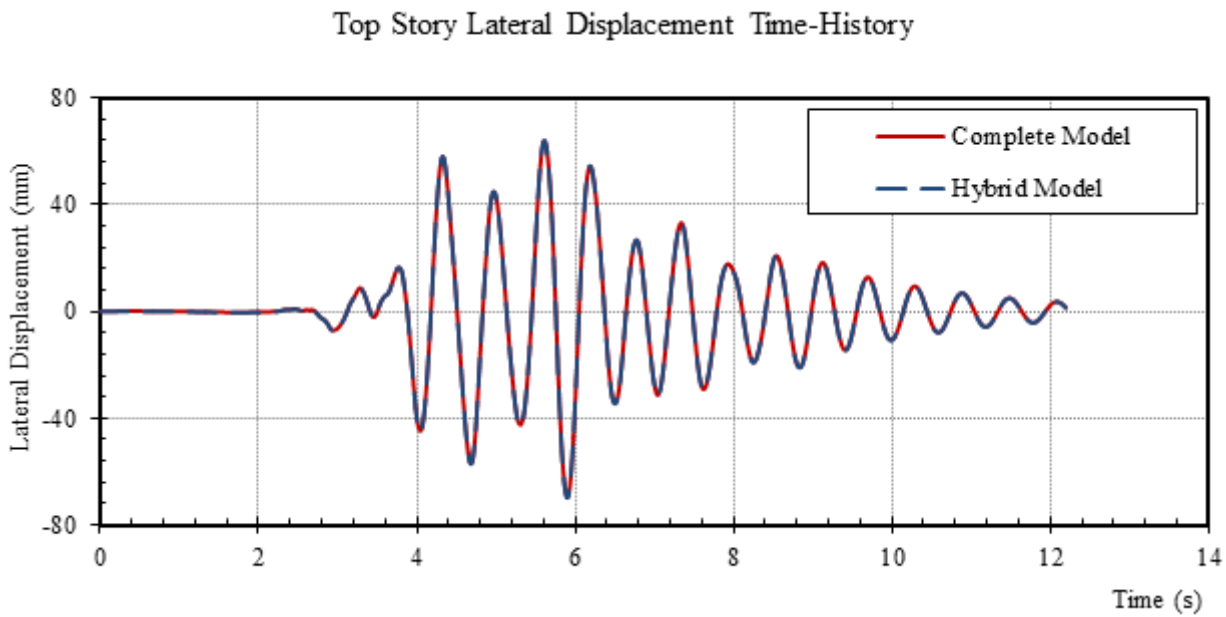
    - Send restoring force

```
case Report_Values

    % calculate the size to be appended to the message header
    lens = calllib('DataExchange', 'indicator');

    % calculate restoring force
    force = Kinit * displ;
    sdata = [displ; force];
    sdata1 = libpointer('doublePtr', sdata);

    % send force to OpenSees
    calllib('DataExchange', 'SendData', sockfd1, sdata1, lens, TCP_IP);
```

Can be replaced with the algorithm for nonlinear elements

Substructure modules

- **C++/MATLAB**
  - Linear analysis



Thanks for your attention!

Questions?