NON-LINEAR SPRINGS FOR CYCLIC ANALYSIS OF WOODEN STRUCTURES

Giovanni RINALDIN
PhD candidate
University of Trieste
Piazzale Europa 1, 34127, Trieste, Italy
giovanni.rinaldin@phd.units.it

Claudio AMADIO
Associate Professor
University of Trieste
Piazzale Europa 1, 34127, Trieste, Italy
amadio@units.it

Massimo FRAGIACOMO
Associate Professor
University of Sassari
Piazza Duomo 6, 07041, Alghero, Italy
fragiacomo@uniss.it

Abstract

A non-linear spring for modelling of wooden structures under seismic loading condition has been implemented as an external library in OpenSees framework. This element, previously used in the FE code ABAQUS and written in Fortran 77 language, has been interfaced with OpenSees via an ad-hoc routine. The proposed element is used in static and dynamic non-linear models of wooden structures to represent the dissipative capacity of steel connections. An example of a X-lam (cross-lam) wall is presented and a comparison with available experimental results is shown. The model is built with non-linear springs elements and quad elements characterized by linear-elastic behaviour.

Keywords: cyclic hysteretic dissipation, Fortran language, non-linear springs, X-lam, wooden structures.

1. Introduction

This article presents an implementation of a new element in the OpenSees Framework. From version 2.3.0 (release tag 4554), OpenSees supports external libraries (DLL) to adding capabilities to the framework. The available Fortran wrapper has been used to interfacing OpenSees classes to Fortran structured code. The new implemented non-linear spring element models the hysteretic behaviour of a steel connector for X-lam wooden buildings, like the spring element presented in [1]. X-lam panels are schematized with Quad elements associated to elastic material, while the steel connections are modelled using the aforementioned non-linear springs. The model is based on the evidence that energy dissipation of X-lam wooden buildings subjected to earthquake excitation mainly occurs in the connections between panels and with the foundations. With this assumption, the capabilities of the proposed model can be used to carrying out static and dynamic non-linear analyses of X-lam structures with the aim to predict faithfully their seismic performance. In this paper, a first non-linear static analysis of a X-lam wall is presented, and the results are compared to experimental ones.
2. Non-linear spring

The element formulation and a brief description of the developed interface are presented in this section.

2.1 Kinematic formulation

Each metal connection (component) has been modelled as a non-linear spring with hysteretic behaviour. The actual curves have been approximated with piecewise linear laws, more specifically tri-linear curves, which have been parameterized to allow the user to fully control their shape. Three different types of curve have been developed: for angle brackets, for screws and for hold-downs. Each curve is made of several branches composing the backbone curve and the hysteretic cycle.

The non-linear spring connects two coincident points in the undeformed state, hence it has zero length. In the most general case, every spring returns to the solver the three forces that develop in its plane and the corresponding three stiffnesses (see Table 1). Although only planar springs with three degrees of freedom have been considered in this study for the sake of simplicity as this is the most important case, the theory can be easily generalized to the case of a spatial spring with six degrees of freedom.

Table 1: Basic definition of non-linear springs.

<table>
<thead>
<tr>
<th>Schematic of springs</th>
<th>Congruence equations – Spring degrees of freedom:</th>
</tr>
</thead>
<tbody>
<tr>
<td>v K_N</td>
<td>φ = φ_2 - φ_1</td>
</tr>
<tr>
<td>u φ</td>
<td>γ = u_2 - u_1</td>
</tr>
<tr>
<td>2 K_γ</td>
<td>ε = v_2 - v_1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Constitutive equations – Spring forces:</th>
<th>Nodal equilibrium equations:</th>
</tr>
</thead>
<tbody>
<tr>
<td>M = M(φ)</td>
<td>T_1 = T; T_2 = -T</td>
</tr>
<tr>
<td>T = T(γ)</td>
<td>M_1 = M; M_2 = -M</td>
</tr>
<tr>
<td>N = N(ε)</td>
<td>N_1 = N; N_2 = -N</td>
</tr>
</tbody>
</table>

2.2 Hysteretic laws

Figure 1a displays the shear force vs. shear displacement (slip) piecewise linear law used to model the behaviour of screw and angle bracket connections.

The model is defined by 9 independent input parameters and uses 32 state variables per cycle. They are:

1. Elastic stiffness;
2. Yielding force;
3. First inelastic stiffness (hardening branch);
4. Peak strength;
5. Second inelastic stiffness (softening or hardening branch);
6. \( K_{sc} \) factor: it sets the unloading stiffness of branches #4 and #50, which is obtained by multiplying the elastic stiffness by this factor \( K_{sc} \);
7. \( RC \) parameter: it sets the lower limit of branches #5 and #40 by multiplying the force value \( F \) attained before entering the unloading path by this parameter \( RC \);
8. \( SC \) parameter: it sets the lower limit of branches #4 and #50 by multiplying the force value attained before entering the unloading path \( F \) by this parameter \( SC \);
9. Ultimate displacement, \( D_\sigma \): when this value is attained, a brittle failure occurs.
Figure 1b displays the axial force vs. axial displacement piecewise-linear relationship used to model the behaviour of the hold-down connector. The same parameters as before were used. Four additional parameters are used to calibrate the strength and stiffness degradation, which are explained in the following section.

2.2.1 Stiffness and strength degradation

Stiffness and strength degradations have been implemented in the model as they are both important features of timber connections. A degradation of stiffness proportional to the maximum displacement attained during the load history has been assumed for the last unloading branches #5 and #50 (after the pinching effect) for both spring models. This effect has been taken into account with equation (1).

\[
k_{\text{deg}} = k_{\text{el}} \left[ 1 - \frac{D_{\text{max}}}{D_{\text{ult}}} \right] (1 - d_{\text{sf}})
\]

where:
- \(k_{\text{deg}}\) = degraded stiffness;
- \(k_{\text{el}}\) = elastic stiffness;
- \(D_{\text{max}}\) = maximum displacement attained during the load history;
- \(D_{\text{ult}}\) = ultimate displacement;
- \(d_{\text{sf}}\) = stiffness degradation parameter.

The strength degradation depends on the energy dissipated and on the maximum displacement attained during the load history. Due to the complexity of evaluating the dependence of the strength degradation on both these quantities, three calibration parameters (one linear and two exponential) have been introduced. The adopted relationship is reported in (2).

\[
\Delta d = \gamma \cdot d_{\text{el}} \left( \frac{E_{\text{dis}} - E_{\text{dis(A)}}}{E_{\text{dis}}} \right)^{\alpha} \left( \frac{D_{\text{max}}}{D_{\text{ult}}} \right)^{\beta}
\]

where:
- \(\Delta d\) = additional displacement at reloading;
- \(\gamma\) = linear parameter;
- \(d_{\text{el}}\) = displacement at yielding force;
- \(\alpha\) = exponential degradation parameter;
\[ E_{\text{dis}} = \text{dissipated energy}; \]
\[ E_{\text{dis}(A)} = \text{dissipated energy at the beginning of unloading path}; \]
\[ D_{\text{max}} = \text{maximum displacement attained during the loading history}; \]
\[ D_{\text{ult}} = \text{ultimate displacement}; \]
\[ \beta = \text{exponential degradation parameter}. \]

2.3 Interfacing OpenSees

The Fortran wrapper included in OpenSees version 2.3.0 (release tag 4554) has been used to interface the structured code previously existing for the solver ABAQUS from Simulia. The wrapper provides all the necessary types (objects) and API interfaces (methods) to handle the new element, and is compiled with C calling conventions. The used platform is Windows, the IDE is Visual Studio 2008 and the compiler is Intel Fortran 11. The code is finally compiled as Dynamic Link Library (DLL).

The code provides the solver with the tangent stiffness matrix and all the reactions for both nodes in the plane where the spring acts. All the needed parameters are read after the “element” instruction in the input file, followed by the name of the subroutine (coincident with the name of the DLL), which is “woodspr”. The parameters read are:

1. springTag: integer representing the ID of the spring;
2. node1 and node2: integer parameters that indicate connectivity;
3. nProps: number of parameters used as input to be read;
4. nSvars: number of state variables needed in the subroutine;
5. sprDOFs: degrees of freedom for the springs, numbered in OpenSees notation;
6. jelemID: internal ID of the spring, used for ABAQUS subroutine compatibility.

After this data, the input parameters (Props) are read.

The Fortran spring subroutine used previously in ABAQUS is included in the code entirely via the INCLUDE instruction. The main subroutine inside it, called “UEL”, is called when the calculation of stiffness and residual forces is needed (when “isw” flag is equal to “ISW_FORM_TANG_AND_RESID”).

3. Numerical analysis

3.1 Calibration

Each type of spring was calibrated on the experimental results reported in [2] for X-lam connections.

![Figure 2. Calibration of angle bracket springs (a.) and of hold-down springs (b.)](copyright IVALSA-CNR)
The calibration was done by following the steps listed herein after:

1. the yielding and peak force were extracted from the experimental results;
2. the elastic stiffness was estimated once a good fit of the yielding displacement was obtained;
3. the hardening stiffness of the plastic branch was chosen on the basis of the backbone curve;
4. the other parameters, such as the stiffness and the strength degradation factors, were evaluated in an iterative way until a goof fit between the experimental curve and the model was obtained.

As can be seen in Figure 2, each calibration shows a good approximation of the experimental result on the single connector. This fact is fundamental in order to extend results to entire buildings.

To speed up the calibration process of the spring components on the experimental results, the software So.ph.i. (acronym for SOftware for PHenomenological Implementations) has been developed using the Visual Basic .NET language [3]. So.ph.i. allows the user to visualize the results of the calibration made upon an experimental data set of a certain component [4] (Figure 3). In addition, So.ph.i releases an input data file in the right format that will be used in the user subroutine implemented for cyclic modelling of the corresponding spring component. An automated calibration procedure based on EN 12512:2001 [5] has been implemented in So.ph.i. This allows the user to obtain automatically elastic stiffness and yielding force values according to the code.

Furthermore, strength degradation was calibrated with a genetic algorithm developed at the University of Trieste, Department of Civil Engineering and Architecture. The obtained parameters are assumed as constant for a certain type of connector, and they have not been changed in other calibration for the same connection.

In Table 2 the mean characteristics of single component calculated from tests presented in [1] is shown. These values have been used in all subsequent analyses.

![Figure 3: Software So.ph.i. v.4.0 for automated calibrations, freely available at http://giovanni.rinaldin.org](http://giovanni.rinaldin.org)
Table 2. Calibration of mean values obtained from experimental results on X-Lam connections

<table>
<thead>
<tr>
<th>Parameter description</th>
<th>Angle bracket: shear</th>
<th>Angle bracket: tension</th>
<th>Hold-down: tension</th>
<th>Hold-down: shear</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elastic stiffness $k_e$ [kN/m]</td>
<td>1.78</td>
<td>2.76</td>
<td>4.82</td>
<td>0.99</td>
</tr>
<tr>
<td>1st inelastic stiffness $k_{pe}$ [kN/m]</td>
<td>0.22</td>
<td>0.41</td>
<td>0.69</td>
<td>0.12</td>
</tr>
<tr>
<td>2nd inelastic stiffness $k_{pe}$ [kN/m]</td>
<td>-0.9</td>
<td>-0.86</td>
<td>-0.96</td>
<td>-0.8</td>
</tr>
<tr>
<td>Yielding force $F_y$ [kN]</td>
<td>23.77</td>
<td>19.23</td>
<td>40.3</td>
<td>9.79</td>
</tr>
<tr>
<td>Maximum force $F_{max}$ [kN]</td>
<td>27.71</td>
<td>23.51</td>
<td>48.33</td>
<td>13.88</td>
</tr>
<tr>
<td>Unloading ratio $SC$ [%]</td>
<td>0.86</td>
<td>0.99</td>
<td>0.95</td>
<td>0.88</td>
</tr>
<tr>
<td>Reloading ratio $RC$ [%]</td>
<td>0.66</td>
<td>0.88</td>
<td>0.86</td>
<td>0.66</td>
</tr>
<tr>
<td>Ultimate Displacement $D_u$ [mm]</td>
<td>64</td>
<td>24</td>
<td>30.2</td>
<td>48</td>
</tr>
</tbody>
</table>

4. Numerical example

4.1 X-lam wall

Once calibrated, the hold-down and angle bracket spring components were used to model a full-scale crosslam panel tested at CNR-Ivalsa Trees and Timber Institute [2]. The panel was connected to the foundation with three angle brackets and two hold-downs at both ends. It was made of five 17 mm thick layers of boards, for a total thickness of 85 mm. Panel dimensions were 2.95×2.95 m. The loading protocol followed during the test was the one prescribed by EN12512 [5]. The vertical load applied on the top of the specimen was 18.5 kN/m, and the connectors used were 2 hold-down HTT22 with 12 nails φ4×60 mm anchored with φ16 bolts and 4 angle brackets BMF 90×48×3×116 mm with 11 nails φ4×60 mm anchored to the foundation with φ12 bolts.

The X-lam wall is modelled with elastic Quad elements whilst every connection is modelled with non-linear springs, placed as in Figure 4a.

![Figure 4. Mesh and springs used to model the Panel (left, a.), and an example of deformed shape predicted in the numerical analysis (right, b.)](image)

The analysis is carried out imposing displacement at the top of the panel through the “sp” command. The analysis settings are reported herein after:
- Constraints: Transformation
- Test: NormDispIncr Integrator: LoadControl
- Algorithm: ModifiedNewton
- Numbered: RCM
- System: BandGeneral
- Analysis: Static.

The experimental-numerical comparison is displayed in Figure 5a, showing an overall acceptable approximation. It must be also pointed out that, unlike other software packages, no convergence problems arose at any time during the cyclic analysis carried out with the proposed model.

![Figure 5a](image)

**Figure 5. Numerical and experimental comparison on cyclic behaviour of X-lam panel (a.) and time-history of the total energy during the cyclic tests performed at IVALSA-CNR [2]**

Also the total energy was calculated and compared with the experimental one, showing that the numerical values are quite close to the ones from the test (Figure 5b).

5. Conclusions

A spring developed for the FE solver ABAQUS was implemented in the OpenSees framework through the Fortran wrapper provided in version 2.3.0. The spring allows implementation of cyclic behaviour of steel connectors in X-lam buildings, where all the dissipative capacity is assumed in the connections. The timber structure is modelled with elastic Quad elements and with the developed non-linear springs, which need to be previously calibrated on single component experimental tests.

Finally, a laboratory cyclic test of a X-lam wall was reproduced numerically in order to validate the model. The wall was tested at CNR-Ivalsa in displacement control and was connected to the ground with three angle brackets and two hold-downs. The analysis was carried out by imposing displacement at the top of the panel in accordance with the experimental protocol used. Obtained results show an overall acceptable approximation of the experimental behaviour, even in terms of total energy.

This model represents an effective tool to estimate the seismic behaviour of wooden (in particular X-lam) buildings. Thanks to the open nature of OpenSees, the model can be easily shared with other researchers at no cost. Furthermore, unlike other “private” models such as the Foschi’s [7] one which does not need calibration but is restricted to smaller models, the proposed model can be used to schematize entire buildings with different types of connections. It is the authors’ intention to make this model freely available once this work will be completed and fully validated.
As a future development, a behaviour involving more DOFs in the spring will be studied and implemented, in order to control the strength simultaneously in every direction through a resisting domain. Furthermore, the spring element will be extended to three-dimensional structures. Also dynamic capabilities of the proposed approach need further investigation. Finally, the ability of OpenSees to run in parallel on several machines will be explored. Now this capability does not support new element implementations as external DLL; as soon as this feature will be added, the model will be tested on the computer cluster available at the University of Trieste, Department of Civil Engineering and Architecture.

6. References


