The Journal of Open Source Software

bridge-sim: A Python library for concrete slab bridge simulation

Jeremy Barisch-Rooney[1, 2]

1 TNO **2** University of Amsterdam

DOI: 10.21105/joss.02496

Software

- Review
- Repository
- Archive

Editor: Patrick Diehl

Reviewers:

- @skreimeyer
- @manparvesh

Submitted: 06 July 2020
Published: 20 July 2020

Summary

`bridge-sim` is an open-source library for Python that provides a high-level API for building linear 3D models of concrete slab bridges, running simulations of the generated models with OpenSees (McKenna, 2011), and generating time series and graphics of the responses e.g. Figure 1. `bridge-sim` has been written with extensibility in mind such that adding support for another finite element program other than OpenSees is possible.

`bridge-sim` makes the process of generating time series data via simulation of concrete slab bridges much faster and easier than with OpenSees alone (OpenSees has no knowledge of concepts such as bridges, traffic, temperature, or even how to generate a mesh). `bridge-sim` accomplishes this by providing a high-level API with classes such as `Bridge`, `Material` and `Vehicle`. Functionality includes generation of finite element models based on a `Bridge` specification including mesh generation, traffic flow generation, settlement of piers, simulation of temperature effect and concrete shrinkage and creep. More detail of the provided functionality is given in the list below. One of the generated models has been validated against sensor data collected from bridge 705 in Amsterdam, a plot comparing responses from linear simulation to static load tests (a truck parked on the bridge) is shown in Figure 2.

- **Model file generation:** generating a model file for OpenSees from a high-level description of geometry, material properties and boundary conditions.
- **Pier settlement:** simulation of settlement of piers by applying a displacement load to the central node of one or more supporting piers.
- **Traffic flow:** generation of a traffic flow based on a Poisson process and simulation of the response of the bridge to the traffic based on superposition of a number of one-time simulations.
- **Temperature load:** simulation of temperature effect based on linear thermal expansion $\Delta L = \alpha L \Delta T$.
- **Shrinkage & creep:** generation of responses due to shrinkage and creep of concrete using EuroCode 2 models (EN, 2004) .
- **Plots & animations:** support for collecting and displaying simulation results including generation of animations e.g. Figure 1.
- **In development:** after discussion with TNO, work has begun to add support for running non-linear simulations, and for generating FE model files for girder bridges.

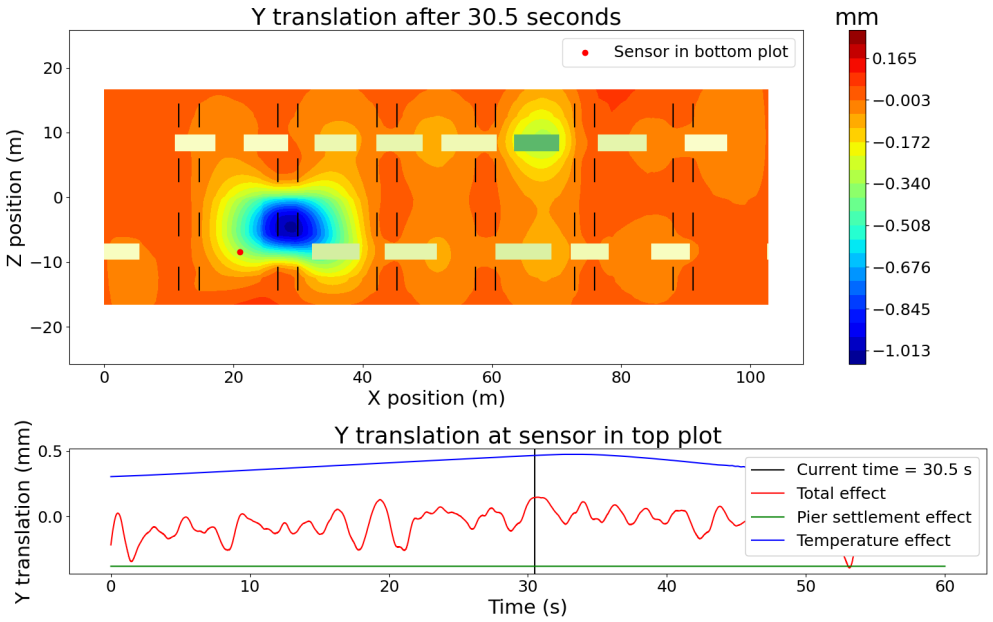

Figure 1: The top plot is a contour plot of vertical translation responses. The rectangles are vehicles on the bridge. One supporting pier has been settled by 1 mm. The bottom plot shows a time series of responses from a vertical translation sensor due to traffic, pier settlement and temperature effect.

Statement of need

The probability of a deteriorating bridge to fail increases over time until it is no longer considered safe for use. Maintenance of a bridge is typically carried out when something goes wrong or according to a preventative maintenance schedule based on expert knowledge, neither approach making the best use of limited maintenance resources. Many bridges in Europe are reaching the end of their design life because of the post-war surge in bridge contruction, yet maintenance can extend the life of a bridge. Sensor data can be used to detect damage in real-time without the delay or cost of a real-time maintenance check.

One of the biggest problems for research into damage detection of concrete slab bridges is data collection from the structure in damaged state (Worden & Manson, 2006). This is because bridges are expensive structures that we are simply not allowed to damage, except as they are being decommisioned but then traffic is no longer permitted on the bridge. To circumvent this issue a lot of research is based on numerical simulations. However models used in such research are typically not built for re-use, and so researchers must spend time creating models from the low-level building blocks of nodes and forces. `bridge-sim` addresses the need for a high-level API for data collection from concrete slab bridge simulation.

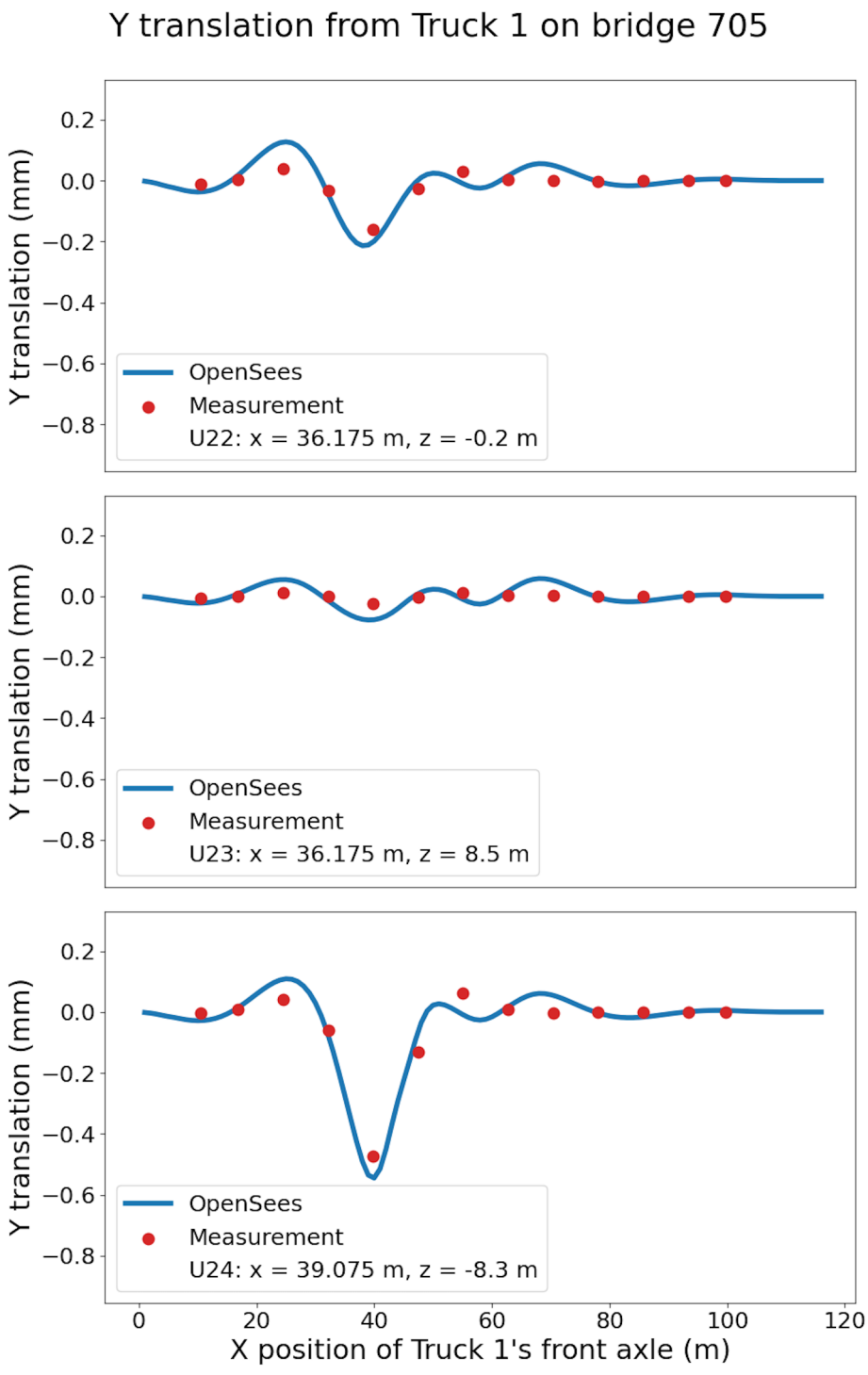

Figure 2: Comparison of vertical translation responses from linear simulation with `bridge-sim` and measurements collected in real life. The real bridge which is modeled and from which sensor measurements were taken is bridge 705 in Amsterdam. The x-axis in each plot shows the longitudinal position of the front axle of a truck parked on bridge 705. The y-axis shows the vertical translation from a sensor due to the truck's weight.

Usage example

To give the reader an idea of the level of abstraction that is provided by the `bridge-sim` library a code snippet is provided. The snippet is correct at the time of writing, but may be subject to change. In this snippet a contour plot of vertical translation responses is generated from a custom `Vehicle` placed on a provided model (`Bridge`) of bridge 705 in Amsterdam.

```
import matplotlib.pyplot as plt
from bridge_sim import bridges, configs, model, plot, sim

config = configs.opensees_default(bridges.bridge_705(msl=0.5), shorten_paths=True)
point_loads = model.Vehicle(
    load=[5000, 4000, 4000, 5000, 7000],
    axle_distances=[2, 2, 2, 1],
    axle_width=2.5,
    kmph=20,
).point_load_pw(time=3.5, bridge=config.bridge, list=True)
responses = sim.responses.load(config, model.RT.YTrans, point_loads)
plot.contour_responses(config, responses, point_loads)
plot.top_view_bridge(config.bridge, piers=True)
plt.savefig("example.pdf")
```

Acknowledgements

Support from TNO is acknowledged during the genesis of this project.

This software would not have been possible if not for a number of open source projects. In particular the contributors and authors of OpenSees (McKenna, 2011) deserve thanks.

References

EN, B. (2004). 1-1. Eurocode 2: Design of concrete structures–part 1-1: General rules and rules for buildings. *European Committee for Standardization (CEN)*.

McKenna, F. (2011). OpenSees: A framework for earthquake engineering simulation. *Computing in Science & Engineering*, *13*(4), 58–66.

Worden, K., & Manson, G. (2006). The application of machine learning to structural health monitoring. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *365*(1851), 515–537.