Received: 8 October 2022 | Accepted: 19 October 2022

DOI: 10.1002/pamm.202200262

Towards neural network-based numerical friction models

Kerstin Vater[1,*], **Merten Stender**[1], and **Norbert Hoffmann**[1,2]

[1] Hamburg University of Technology, Department of Mechanical Engineering, Am Schwarzenberg-Campus 1, 21073, Hamburg, Germany

[2] Imperial College London, Department of Mechanical Engineering, Exhibition Road, London, SW7 2AZ, UK

Friction contacts can be found in almost all mechanical systems and are often of great technical importance. However, they are usually difficult to describe, and their behavior and influence on the whole system are hard to predict accurately. Modern product design and system operation strongly benefit from numerical simulation approaches today, but reliable friction models still represent a major challenge in this context.

To tackle this problem, we employ neural network regression to capture the characteristics of frictional contacts and make them accessible for numerical methods in a minimal intrusive fashion. In particular, we test our approach using a Finite Element model of a 2D cantilever beam subject to stick-slip vibrations induced by a moving conveyor belt at its free end. As a reference solution, we perform a transient analysis based on a simple analytical friction model, where the kinetic friction force only depends on the normal load and the relative sliding velocity. We take the same friction model, add some artificial noise to mimic uncertainties coming with experimental measurements, and pick a limited set of data points to train a regression neural network. The machine learning friction model is then deployed in the Finite Element code to predict the kinetic friction force acting on the beam tip during the slip phases.

The deflection curves obtained by the transient numerical analysis using the new neural network friction model agree well with the reference solution based on the underlying analytical model. The results indicate that data-driven approaches may also be capable of capturing more complex frictional contacts, including effects of temperature, humidity, and load history. The trained neural network friction models can then be employed in numerical simulations in a minimally intrusive manner. This approach opens up new possibilities to predict individual mechanical system behavior as accurately as possible.

1 Introduction

The behavior of frictional contacts within a mechanical system is usually difficult to describe in a formal way. That is because it depends on many parameters: the relative sliding velocity between a contact pair and the local contact pressure are the most common factors. But temperature and humidity might also have an impact, as well as the load history and wear. The complex nature of friction makes it challenging to build a proper representation of such contact in numerical structure simulation. Analytical models are usually simple but inaccurate in one way or another.

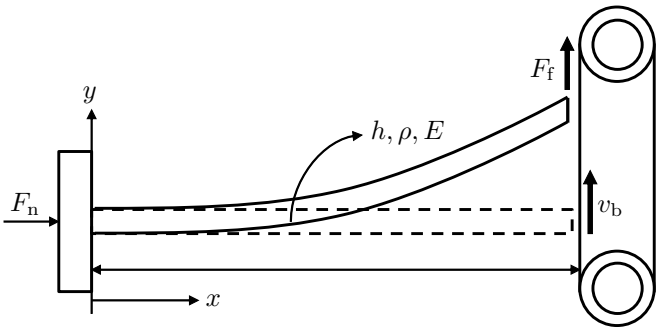
The latest developments in the field of machine learning provide a remedy to this problem. Consider the case where it is possible to measure all significant parameters during an experiment or real-work operation of a system and feed the data of many contact incidents to a neural network. Then, it would be possible to make profound predictions on the contact behavior within a numerical simulation based on Finite Elements (FE), for instance. And that, in turn, enables system design improvements, or estimations on the maintenance interval in the context of digital twins, for example.

To this end, we establish a proof of concept that neural network-based friction models are capable of enhancing the accuracy of numerical analyses. We introduce a simple time-dependent two-dimensional test case and set up a FE model in Section 2. In Section 3, we generate friction data based on an analytical model with added artificial noise to mimic genuine measurement data. We perform hyperparameter optimization studies to decide on the neural network model to fit the friction data, train the network and evaluate performance metrics. Finally, in Section 4, we plug the neural network friction model into the FE algorithm and run the transient test case.

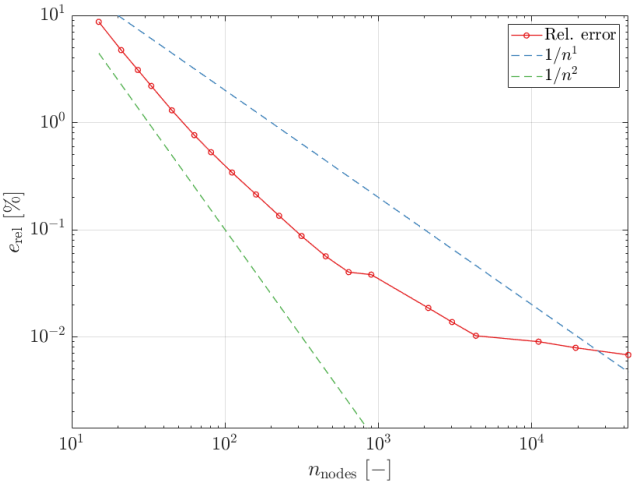
2 Cantilever Beam Finite Element Model

The test case under investigation is a simple cantilever beam [1, 2] with the specifications given in Figure 1a. We choose parameter values such that the Euler-Bernoulli beam theory applies. Thus, for each deflection value of the beam tip, there is an equivalent transversal point load value. The transient FE algorithm exploits this relationship to determine the static friction force during the stick phases. The beam is subjected to a time-dependent point load at its free end, which results from frictional contact with a conveyor belt moving with the specified velocity. The given configuration is supposed to show the stick-slip phenomenon, where the beam tip sticks to and slides on the belt in an alternating fashion. The kinetic friction force acting on the beam tip during the slip phases is obtained from a neural network model that reproduces an exponential-type analytical formulation. A FE analysis is run using either model and the time-dependent results are compared.

* Corresponding author: e-mail kerstin.vater@tuhh.de, phone +49 40 42878 2993



(a) Cantilever beam model with $L = 2\,\mathrm{m}$, $h = 0.01\,\mathrm{m}$, $\rho = 1000\,\mathrm{kg\,m^{-3}}$, $E = 200\,\mathrm{GPa}$, $v_b = 0.02\,\mathrm{m\,s^{-1}}$, $F_n = 100\,\mathrm{N}$



(b) Convergence of the deflection error at the beam tip with increasing number of degrees of freedom

2.1 Mesh convergence study

We use MATLAB's Partial Differential Equation Toolbox [3] to set up a transient 2D plane-stress FE model of the cantilever beam depicted in Figure 1a. Since we aim to make statements on the capability of different numerical friction models, we need to make sure that the numerical error introduced by the spatial discretization does not affect the results of the respective FE analyses substantially. Therefore, the element size for the beam model ideally emerges from a mesh convergence study. For this purpose we consider a static load case, where the beam is simply bend up by a transversal point force $P = 50\,\mathrm{N}$ at the free end. The quantity of interest is defined as the maximum deflection of the beam tip. As we can refer to the Euler-Bernoulli beam theory, we are able to compute an error value

$$e = \frac{|\delta_{\mathrm{ref}} - \delta_{\mathrm{num}}|}{|\delta_{\mathrm{ref}}|} \tag{1}$$

between the FE result δ_{num} and the reference deflection value based on analytical considerations [4]

$$\delta_{\mathrm{ref}} = \frac{4PL^3}{Eh^3}\ . \tag{2}$$

Figure 1b reveals the behavior of the error value e as the number of computational nodes in the mesh n_{nodes} increases as a consequence of repeated h-refinement. We choose a target element edge length of $h_{\mathrm{max}} = 0.02\,\mathrm{m}$ corresponding to a relative error value of about 0.04%. Further mesh refinements do not change the result significantly. Hence, the FE model of the cantilever beam is able to make statements on the quality of different friction models.

2.2 Transient Finite Element algorithm

The transient FE algorithm that is able to compute complex friction phenomena again is build from functions in MATLAB's PDE Toolbox. Figure 2 gives an overview of the iterative process. Within the time-stepping loop, nodal displacements u^i and velocities v^i are computed for the current time step $i + 1$. Subsequently, a reaction force is derived from the transversal deflection of the beam tip $u^i_{y,x=L}$ based on Equation (2) with $P = F_{\mathrm{react}}$. This reaction force is equivalent to the static friction force acting on the free end as long as the beam tip sticks to the conveyor belt. The relative sliding velocity in the contact interface results from the velocity of the beam tip according to $v_s = v^i_{y,x=L}$. During the stick phases, this value equals zero by definition. Assuming that the beam currently sticks to the belt, the static friction force F_{react} is compared to the critical static friction force

$$F_{\mathrm{crit}} = \mu_s F_n \tag{3}$$

depending on the normal force F_n and the static friction coefficient μ_s. In case the current static friction force does not exceed the critical value, the algorithm proceeds using a displacement boundary condition (BC) that enforces a deflection of the beam tip according to the movement of the conveyor belt within the next time step:

$$u^{i+1}_{y,x=L} = v_b \Delta t\ . \tag{4}$$

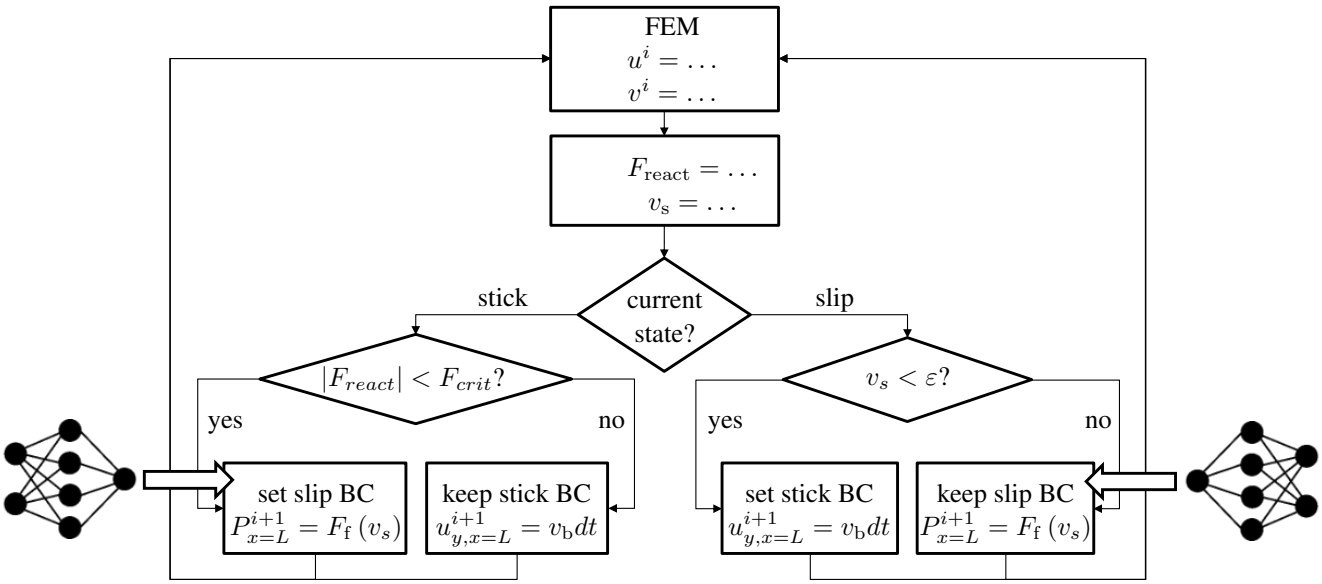


Fig. 2: Transient Finite Element algorithm. The boundary conditions need to be adapted according to the current state of the contact pair (stick or slip). The kinetic friction force $F_{\mathrm{f}}(v_{\mathrm{s}})$ is predicted by a neural network model.

Once the maximum static friction force is reached within the contact interface, the beam tip breaks loose from the belt and starts sliding downwards. The kinetic friction force F_{f} counteracting this movement is subject of a load BC replacing the displacement BC during the stick phase:

$$P^{i+1}_{y,x=L} = F_{\mathrm{f}}(v_{\mathrm{s}}) \ . \tag{5}$$

The friction force BC in Equation (5) is retained until the relative sliding velocity between the beam tip and the belt falls below a numerical threshold value ε. At this point, the algorithm returns to the stick state, where the displacement BC according to Equation (4) applies again. Consequently, there are few isolated points in the FE algorithm, where the numerical friction model and the neural network are employed, respectively. These are the computation of the kinetic friction force for the load BC during the slip phases. Thus, the data-driven friction model developed in the following Section 3 may be plugged into any existing FE code in a minimal intrusive manner.

3 Neural Network-based Friction Model

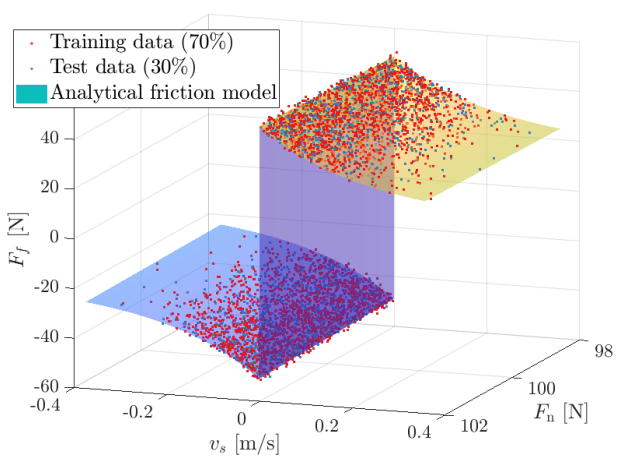
In order to set up a neural network-based numerical friction model, we leverage MATLAB's Statistics and Machine Learning Toolbox [5], that provides comprehensive functionality fit friction data from experiments or real-time system operation. This paper, though, focuses on synthetic data based on an analytical friction model for comparative purposes. In this way it is possible to make statements on the performance of the novel neural network approach.

3.1 Training and validation data generation

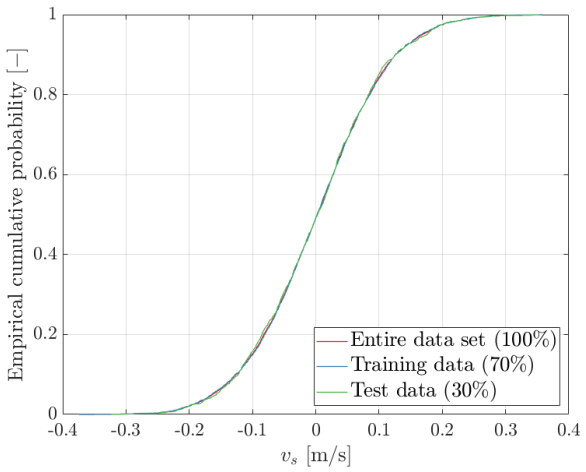
We use a simple exponential-type friction model [6] taking into account two variables, which are the relative sliding velocity v_{s} and the normal force F_{n} between the contacting surfaces:

$$F_{\mathrm{f}}(v_{\mathrm{s}}) = \mu(v_{\mathrm{s}}) F_{\mathrm{n}} = \left[\mu_k + (\mu_s - \mu_k)\, \mathrm{e}^{-\alpha |v_{\mathrm{s}}|}\right] F_{\mathrm{n}} \ , \tag{6}$$

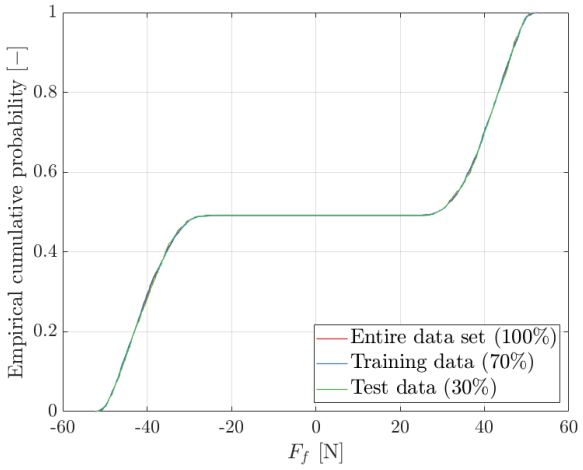
where $\mu_{\mathrm{k}} = 0.2$ and $\mu_{\mathrm{s}=0.5}$ denote the static and kinetic friction coefficient, respectively, and $\alpha = 5$. With a view to the parametrization of the FE test case, we consider $N = 5000$ equally distributed random normal force values around $F_{\mathrm{n}} = 100\,\mathrm{N}$, and a zero-mean normal distribution of associated relative sliding velocity values $v_{\mathrm{s}} \in [-0.4, 0.4]$ with $\sigma = 0.1$. The output values are computed according to Equation (6). Normally distributed noise with zero mean and standard deviation $\sigma = 0.02$ is added to F_{f} in order to mimic genuine measurement data. The friction data set is partitioned into a training set making up 70%, and a test set of 30%. Figure 3a shows the surface plot of the exponential-type friction model, as well as the training and test data samples. Figures 3b and 3c provide deeper insights into the distribution of values in the resulting data sets. The normal force F_{n} and the sliding velocity v_{s} act as inputs (features) for a regression neural network, whereas the kinetic friction force F_{f} is the output variable.



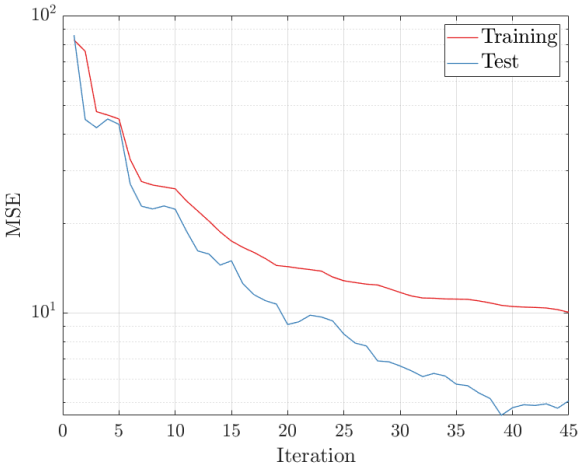
(a) Data samples based on an exponential-type friction model with added noise. The $N = 5000$ examples are split 70/30 into a training and a validation set.



(b) Empirical Cumulative Distribution Functions of feature v_s in the full, training, and test data set



(c) Empirical Cumulative Distribution Functions of output F_f in the full, training, and test data set



(d) Training history plot showing the loss values (Mean Squared Error) over the number of iterations of the LBFGS optimizer. The algorithm converged after 45 iterations.

3.2 Neural network regression model

Regarding the neural network type and architecture to fit the two-variable regression problem described in Section 3.1, it is a common choice to employ a feedforward fully-connected network for this kind of problem. Concerning appropriate values for the number of hidden layers, the number of neurons in each layer, the activation functions, as well as the regularization term strength λ, hyperparameter optimization studies are performed. The results suggested using three fully connected layers with $s = 100$ neurons each, and hyperbolic tangent activation functions, and $\lambda = 0.004$. The neural network structure is depicted in Figure 4. In order to train the network, we employ a limited-memory Broyden-Fletcher-Goldfarb-Shanno (LBFGS) quasi-Newton optimization algorithm [7], that minimizes the mean squared error (MSE) on the training data. MATLAB's Machine Learning Toolbox provides a function named `fitrnet` for that purpose. With a view to the training history plot in Figure 3d, it can be determined that the MSE finally converges and the optimization process is terminated after 45 iterations at a test loss value of $\mathrm{MSE}_{\mathrm{test}} = 4.52$. The coefficient of determination (R-squared) on the test data yields a remarkably high value of $R^2_{\mathrm{test}} = 0.9974$.

4 Results

Prior to the actual FE analysis deploying the neural network friction model, the performance of the fit model is assessed in greater detail. To this end, Figure 5a reveals how close the predicted test set response values are to the true responses over the entire value range. Outliers can be identified especially for high values of the kinetic friction force F_{f} associated with sliding velocities close to the jump in sign at $v_{\mathrm{s}} \approx 0$. Figure 5b shows the analytical and the estimated friction force curves.

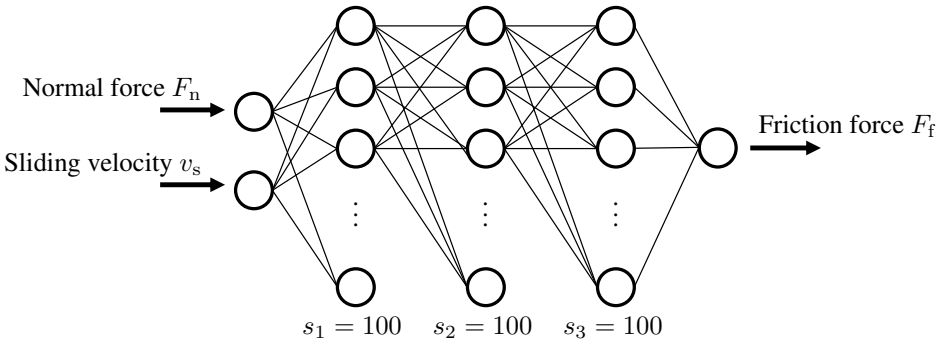
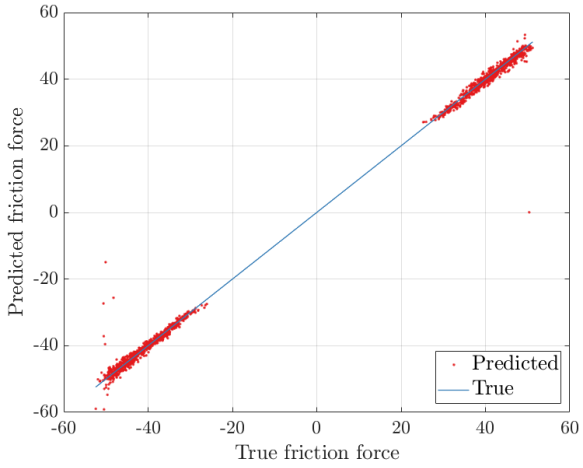
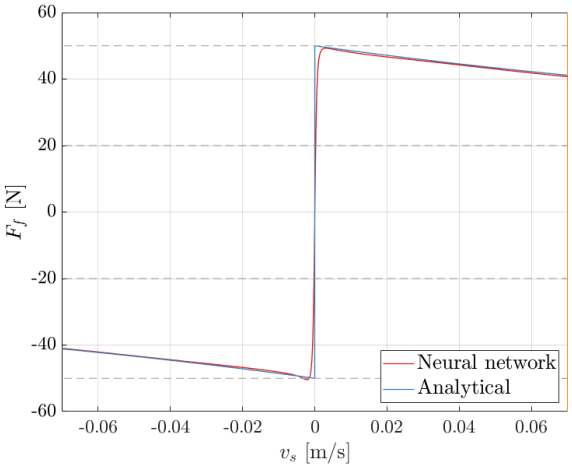


Fig. 4: Neural network architecture for regression with two inputs and one response variable featuring three fully connected hidden layers



(a) Friction force values predicted by the neural network model for the test set compared to the true response values. Outliers are particularly significant in the region corresponding to the jump in sign of the sliding velocity.



(b) Neural network friction model. The kinetic friction force estimated by the neural network is sampled for various values of the relative sliding velocity within the range of interest.

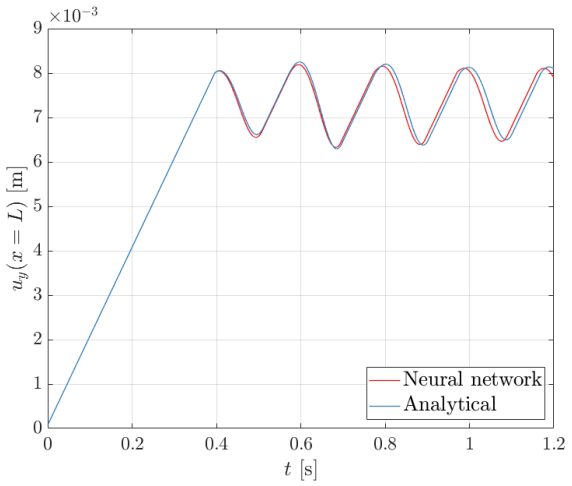
And indeed, it becomes obvious that the neural network model tends to smooth the jump discontinuity at the zero-crossing of the sliding velocity. However, it is not uncommon for discontinuities to cause difficulties in the numerical approximation of otherwise smooth functions. The most popular example is probably the Gibbs phenomenon in the context of polynomial approximation. In this case, though, the numerical threshold value $\varepsilon = 1 \times 10^{-3}\,\mathrm{m\,s^{-1}}$ in the FE algorithm 2 mitigates this problem, as the velocity range around zero is not considered, anyway. This is the stick regime by definition, where no kinetic friction force needs to be computed.

The neural network-based numerical friction model developed in Section 3 is finally plugged into the simulation code. The FE test case described in Section 2 is run up to a final time $t_\mathrm{final} = 1.2\,\mathrm{s}$ with a step size of $\Delta t = 1 \times 10^{-4}\,\mathrm{s}$. Note that this is the (outer) time step size determining the resolution of the stick-slip phenomenon. MATLAB's `solve` function may divide the given interval $\left[t^i,\ t^i + \Delta t\right]$ further for algorithmic stability reasons and error control.

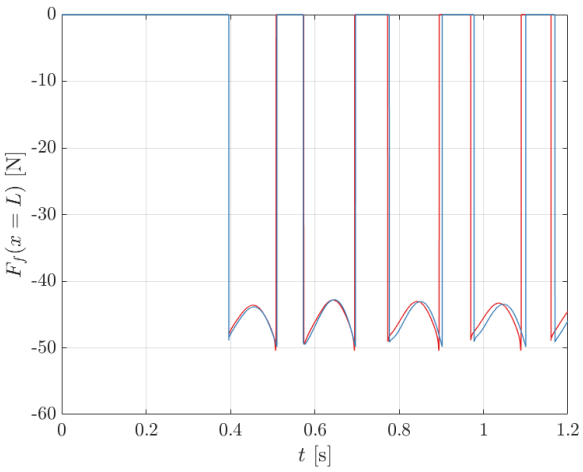
Figure 6a shows the transversal displacement of the beam tip over time as it is contacting the steadily moving conveyor belt. The red line corresponds to the FE analysis results employing the neural network-based friction model, the blue line is associated with the results obtained from the same setup but using the analytical friction model instead. The overall shape of the data driven results fit the reference solution quite well. During the (stiction only) ramp-up phase until $t \approx 0.4\,\mathrm{s}$, the algorithms are actually identical. Beyond the first break-away point, a slight phase shift builds up with increasing simulation time. The oscillation amplitude, however, stays within a refined corridor in both the neural network and the reference solution.

5 Conclusion

The present study involves a neural network-based numerical friction model to be deployed within a transient FE simulation in a minimally intrusive manner. Synthetic friction data based on an analytical reference model is designed to emulate the nature of real-world sensor data, while at the same time allowing for comparative studies between the neural network-based FE solution and the results based on the analytical friction formulation. The outcome involving a minimal test case is very satisfactory. The neural network regression model captures all features of the stick-slip motion with high accuracy. These



(a) Simulated deflection of the cantilever beam tip over time featuring the expected stick-slip motion



(b) Temporal evolvement of the kinetic friction force acting between the conveyor belt and the beam tip. The force is disabled during the stick phases, when the static friction force applies.

early results pave the way to incorporate the complex behavior of genuine frictional contacts in existing numerical simulation software.

Acknowledgements Open access funding enabled and organized by Projekt DEAL.

References

[1] H. I. Won and J. Chung, Journal of Sound and Vibration **419**, 42–62 (2018).
[2] H. I. Won, B. Lee, and J. Chung, Nonlinear Dynamics **92**(4), 1815–1828 (2018).
[3] I. The MathWorks, Partial Differential Equation Toolbox, Natick, Massachusetts, United State, 2020-2021.
[4] J. M. Gere and B. J. Goodno, Mechanics of materials (Cengage learning, 2012).
[5] I. The MathWorks, Statistics and Machine Learning Toolbox, Natick, Massachusetts, United State, 2020-2021.
[6] E. Berger, Appl. Mech. Rev. **55**(6), 535–577 (2002).
[7] J. Nocedal and S. J. Wright, Numerical optimization (Springer, 1999).