# Implementing Machine Learning in Earthquake Engineering

Cristian Acevedo
Civil and Environmental Engineering
Stanford University, Stanford, CA 94305, USA

**Abstract**—**The use of machine learning across many fields has seen a rise in recent years, from life and physical sciences to finance and athletics. Within the physical sciences, it is just starting to see some implementation in the field of Earthquake Engineering. The objective of this paper is to implement machine learning to Earthquake Engineering data to create more literature in the field. In particular, this project aims to implement predictive models to properly capture the residual displacement of a structure caused by an earthquake using acceleration data. Current methods, which involve double integration of the acceleration data with a combination of baseline correction and filtering, do not do a good job at capturing residual displacements. For this reason, machine learning was investigated as a possible alternative to numerical integration. The results showed that Feedforward and Recurrent Neural Networks are not able to pick up the residual displacement. In addition, it was found that ground displacement was an important feature to get reasonable results. More research needs to be done on this topic before discarding neural networks as a possible solution for obtaining residual displacements from acceleration data from an earthquake.**

**Index Terms**—Acceleration, integration, displacements, earthquake, engineering, feedforward, recurrent, neural networks

## INTRODUCTION

The use of accelerometers in the field of Earthquake Engineering is very important as it helps engineers understand and quantify the magnitude of seismic forces on structures. The ground accelerations of an earthquake are typically recorded at different locations away from the fault and are used in numerical models by engineers to develop better methods for design. Similarly, structures are typically instrumented at floor levels of buildings, in order to monitor the acceleration forces which can then be used to validate numerical models. An important aspect of these accelerometers, is that the acceleration data can be use to monitor the floor displacements of the building by double integrating it numerically. Since various components from a building have a certain damage threshold displacement, obtaining the displacements can help evaluate and determine whether the displacements were large enough to cause significant non-visible damage on a building (i.e., within the walls or locations that are hard to spot with the naked eye). With that said, double integrating acceleration data can be a tricky task as the data needs to be baseline corrected and filtered to remove noise and get reasonable results. Finding the right filter and base line correction is a blind trial and error task as the resulting displacement data cannot be compared to actual displacement readings. Furthermore, displacement sensors have not seen the same application as accelerometers in building monitoring due to their cost and application to large-scale structures since stationary reference points are required [1][2].

For this reason, this project aims to investigate machine learning as a possible alternative to integrate acceleration data to capture the residual displacements. The current literature is limited to earthquake ground motion analysis and simulation, earthquake early warning, and some damage classification [3][4][5]. As a starting point, neural networks were implemented in the study to quantify its versatility and appropriateness for this problem. In particular, two types of neural networks were implemented: 1) Feedforward Neural Networks (FFNN) due to its capability of modeling relationships from input and output without cycles 2) Recurrent Neural Networks (RNN) due to its versatility for time sequences [6]. Both neural networks are implemented and compared to numerical double integration.

## DATA USED

Data was obtained from the experimental results of a 2-story wood-frame Unibody house [7]. The structure was tested on an earthquake simulator or shake table under seven different ground motions or earthquakes (see Fig. 1). The 1989 Loma Prieta Earhtquake (GM) was used and scaled at various amplitudes, i.e., 0.4, 1, 1.5, 2.3, 3.0, 1.5, 4.5. Each floor of the house was instrumented with 15 accelerometers, one for each direction of motion,i.e., horizontal, parallel to direction of motion, perpendicular, and vertical. Only the horizontal, parallel to the direction of motion and the vertical accelerations were considered for this study as inputs. The floor and ground displacements were measured using string potentiometers as seen in Fig. 1. In this case, the ground displacement and the target floor displacements in the direction of motion were considered. Each sensor recorded a total of 12,000 data points.

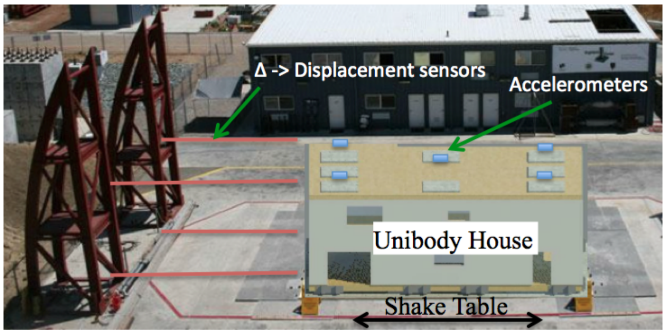



Fig. 1. Experimental Data

One limitation on the data is the fact that it is limited and the residual displacement is only observed on the largest earthquake the house was exposed to. Fig. 2 shows an example of ground motion 5 (GM5) horizontal and vertical acceleration as well as the ground displacement measured.

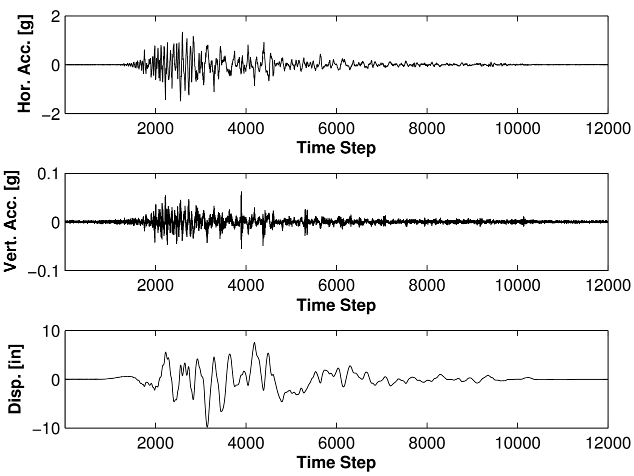



Fig. 2. Earthquake 5 (GM5) Horizontal and Vertical Acceleration and Ground Displacement

## Machine Learning Implementation

Neural networks were investigated for this project using Matlab libraries. Since these are time sequences, Feedforward Neural Networks (FFNN) and Recurrent Neural Networks (RNN), respectively, were particularly implemented. FFNN are characterized by a series of layers where the input is mapped to hidden layers and those layers are mapped directly to the output. Meanwhile, RNN are similar to FFNN except they have a recurrent tap delay. Fig. 3 shows a schematic process of the two types of layers implemented.

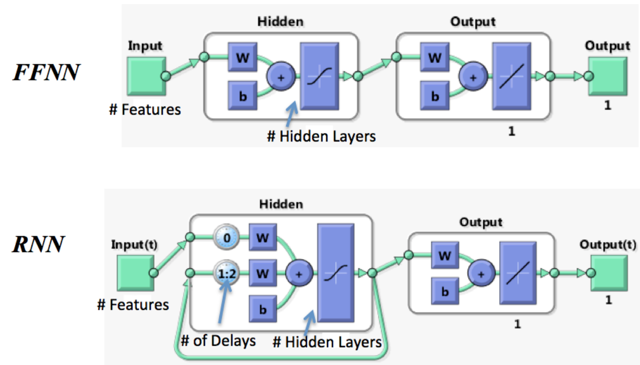



Fig. 3. Schematic Flowchart of FFNN and RNN

## Features

A total of eight features were considered for this project. These features were selected based on data that can be collected from an earthquake in real life scenarios. These features are shown on Table 1. The power spectral densities and ratio of acceleration frequencies were derived using the fast fourier transform.

TABLE 1
Unibody Special Fastener Schedule

| Structure (Floor of Interest | Ground Motion |
|---|---|
| Acc. of the floor | Acceleration of the Ground |
| Vertical acc. of the floor | Vertical acc. of the ground |
| Ratio of floor Fz / ground Fz | Ground displacement |
| Power spectral density of floor acc. | Power spectral density ground acc. |

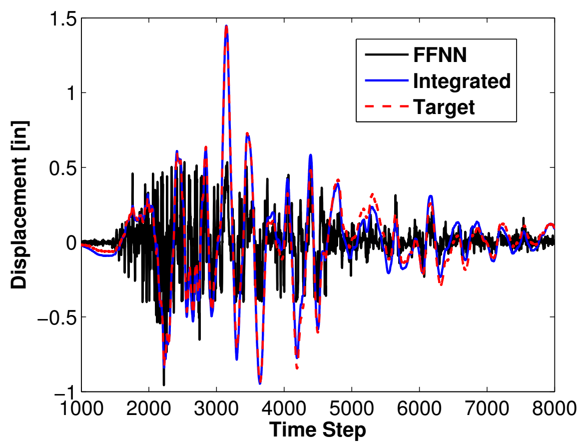
Note: Power spectral density was computed for acceleration in direction of motion only.

## Results

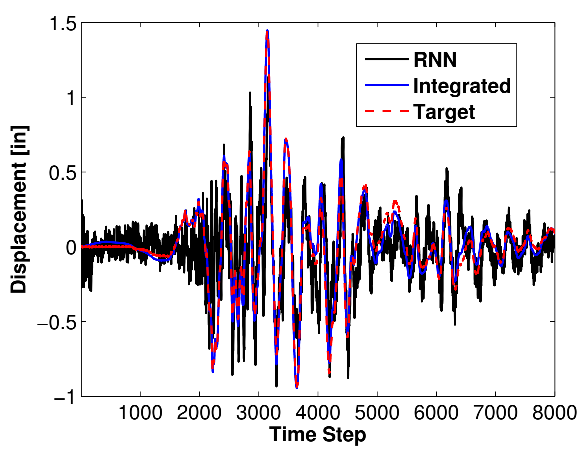
Several trial and error attempts were made to train the data with various features. Two features were considered at first, the ground acceleration and the horizontal floor acceleration. The target displacement used was the one measured by the displacement potentiometers. It was noticed that for both neural networks, the acceleration features took long to train and did not yield good results. As can be seen in Fig. 4,5,6, the output is noisy and it deviates for the test cases. The output from the neural network is compared with the result from the numerical integration. For the numerical integration procedure, the data was filtered using the butterworth function from Matlab, a second degree polynomial, and a low pass frequency of 0.07 Hz. The equation used for the integration is shown below, where $x(t) = displacement$, $t = time$, $\ddot{x}(t) = acceleration$, and $\delta t = time\ increment$.

$$\boxed{\begin{aligned} x(t) &= \int_{t=1}^{t=N}\int_{t=1}^{t=N} \ddot{x(t)}\delta t \delta t \\ &= \sum_{t=1}^{t=N}\sum_{t=1}^{t=N} \ddot{x(t)}\Delta t \Delta t \end{aligned}}$$

When adding the ground displacement as a feature, it was noticed that the neural networks behaved better. As more features were added, it was concluded that the power spectral densities did not contribute to the results; instead, it generated noise on the output. Therefore, a total of six features were used to generate the final results.



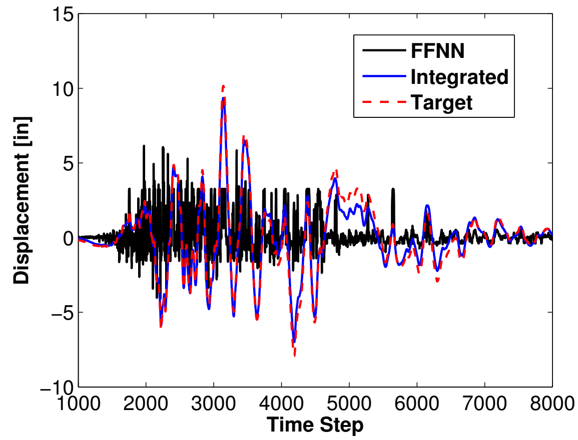

(a) FNN



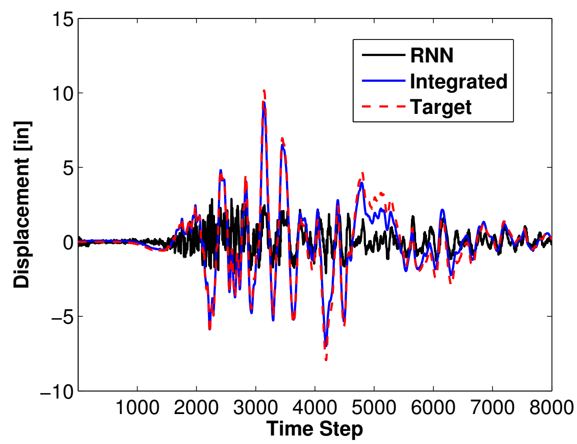

(b) RNN

Fig. 4. Training/Testing Data (GM1) with 2 Features

It is important to notice that our target is to predict GM7, especially the residual displacement. If can be seen from Fig. 4,5,6 that neither the neural networks with two features nor the numerical integration are able to pick up the resulting residual displacement. Nevertheless, the numerical integration performs better than the neural networks for this phase. In addition, when comparing FFNN and RNN, the latter does a better job in during the training of GM1 but both do not perform well for the testing data points.




(a) FNN

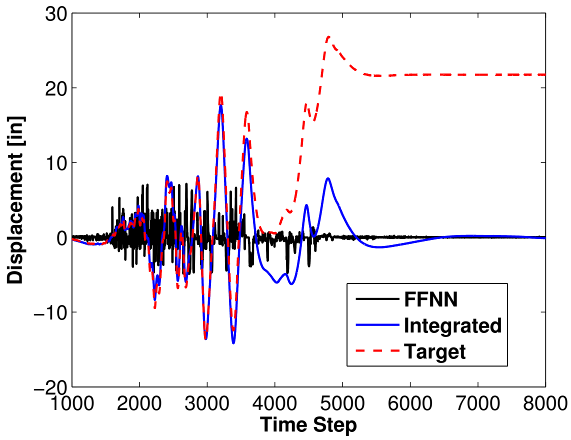



(b) RNN

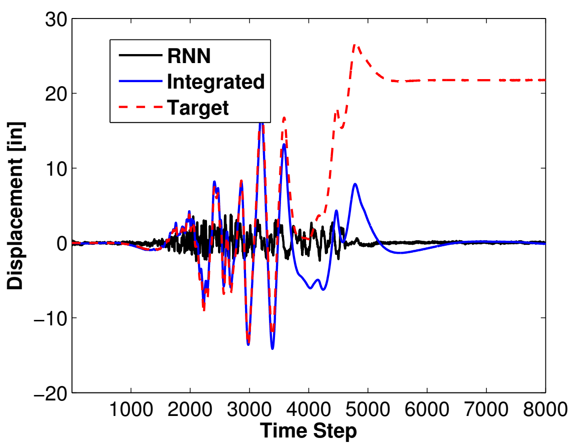
Fig. 5. Testing Data (GM5) with 2 Features

For the final neural network models, the FFNN was trained with 30% of GM1 using six features (all the ones mentioned in Table 1, except for the power spectral densities) and seven hidden layers. The RNN was also trained with 30% of GM1 using the same six features but with ten hidden layers and ten positive delays. The resulting plots can be seen in Fig. 7,8,9. The training performance was $1.42 * 10^{-4}$ for FFNN and $1.35 * 10^{-4}$ for RNN with 15 and 85 epochs respectively.

As can be seen from Fig. 7,8,9, the FFNN and the RNN do a better job at matching the target output displacement for the training and testing datas than when compared to the output generated by only two features. In fact, the match is almost perfect. This can be attributed to the fact that the ground displacement is really close to the floor displacements. On the other hand, when implemented to match GM7, since the displacement is not longer the same amplitude due to the nonlinear behavior of the structure, none of the two neural networks are able to pick up the amplitudes of the target displacement. From the plots presented, it

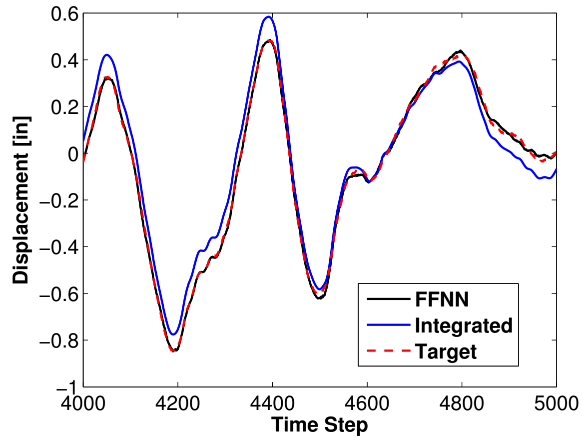



(a) FNN

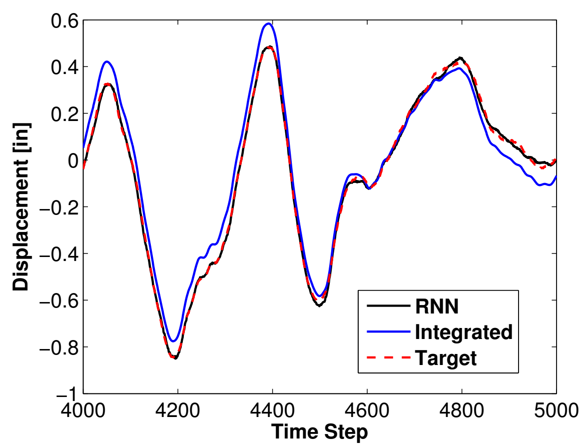



(b) RNN

Fig. 6. Testing Data (GM7) with 2 Features




(a) FNN Close-up




(b) RNN Close-up

Fig. 7. Training Data (30%)/Testing Data (70%) of GM1 with 6 Features

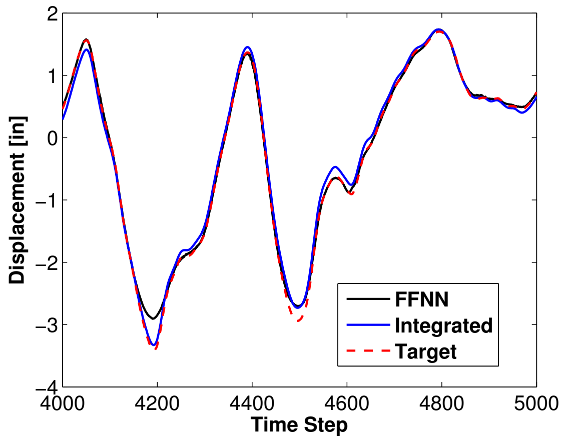
can be seen that FFNN performs slightly better than RNN. This is due to the variability of the performance. More research is needed to determined which method is better. Nevertheless, on thing is clear, neither are able to outperform the numerical double integration in GM7. Two last additional trials was tested for RNN, one was where the delay and hidden layers were increased to 20 trained with 30% percent of the data GM1 and using 6 features and the last one involved using 70% of the GM1 data using 2 features with 20 delay and hidden layers. No improvement was noticed and training took several hours.

## Conclusion

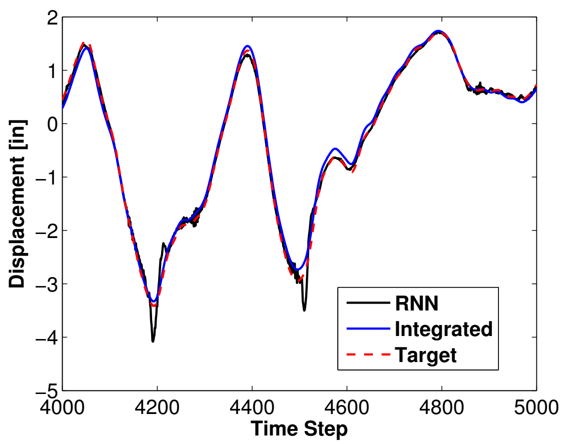
Two types of neural networks were investigated and implemented to solve the problem of residual displacements. The results showed that FFNN nor RNN were able to pick up the residual drift generated by GM7. Both neural networks were trained with only 30% percent of 12000 points for the smallest magnitude earthquake and based on the results it was sufficient to predict a good portion of the rest of the ground motions up to GM6. More training epochs and data should be implemented to see if better performance is reached. Results also suggest that features need to be selected carefully to avoid noise in the target output. The ground displacement was an important feature. In the case of this structure, the ground displacement was very close to the floor displacements since the structure was very rigid. The structure becomes more flexible on GM7. Double numerical integration is still a more practical way to compute the displacement from accelerations even though it does not capture the residual displacements, but it's able to yield reasonable target displacements when there are not residual displacements as seen in the figures presented.

## Future Work

This was just a preliminary study incorporating machine learning to this problem, which arose from the interest of the author. More research is needed before discarding neural networks as a possible method to pick up residual displacements. More data (either experimental or simulated) should be used to generalize the model




(a) FNN Close-up




(b) RNN Close-up

Fig. 8. Testing Data (GM6) with 6 Features

and check if performance is improved. If experimental data is not available, numerical models can be used as a possible substitute to generate a large amount of data for different types of structures with different floor levels and construction materials.

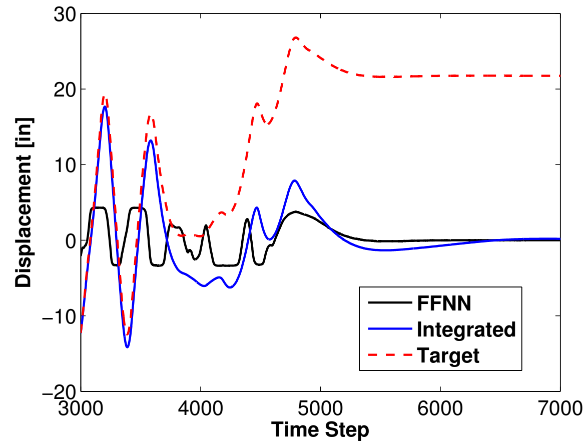
## Acknowledgment

The author is grateful to the TAs from the class whom helped with guidance. In addition, the author is thankful to the National Science Foundation through the Network for Earthquake Engineering Simulation (award NEESR-1135029), which made the data available.

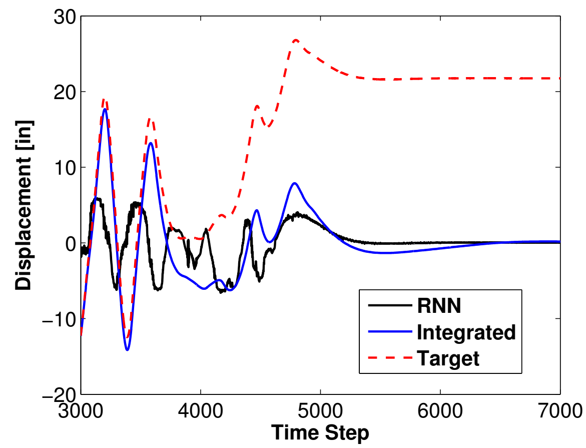
## References

[1] Nagayama, T. and Spencer, B.J. (2007) *Structural Health Monitoring*, Report No. NSEL-001. Urbana-Champaign, Illinois

[2] Park,J.W., Sim,S.H., Jung, H.J., and Spencer, B.J. (2013) *Development of a Wireless Displacement Measurement System Using Acceleration Responses*, *Journal of Sensors* ISSN 1424-8220




(a) FNN Close-up




(b) RNN Close-up

Fig. 9. Testing Data (GM7) with 6 Features

[3] Ramirez, J. and Meyer, F. (2011) Machine Learning for Seismic Signal Processing: Seismic Phase Classification on a Manifold. *10th International Conference on Machine Learning and Applications*

[4] Alimoradi, A. and Beck, J. (2014) Machine-Learning Methods for Earthquake Ground Motion Analysis and Simulation. *Journal of Engineering Mechanics*

[5] Ferguson, M. and Martin, A. (2015) Earthquake-Induced Structural Damage Classification Algorithm. *CS229 Class Project*

[6] Husken, M. and Stagge, P. (2003) Recurrent Neural Networks for Time Series Classification *Neurocomputing 50(C):223-235*

[7] Acevedo, C., Swensen, S., Jampole, E., Deierlein, G., Miranda, E., and Fell, B. *Shake Table Testing of a Full-Scale Two-Story Unibody Wood-Frame House* In Journal Review