This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

SAND2023-06682C

# Machine learning for event detection in non-smooth initial value problems

**B. Evan Saunders*, Andrew Steyer*, and Robert J. Kuether***

** Sandia National Laboratories, Albuquerque, NM 87123, USA.*

## 1. Introduction

Non-smooth and discontinuous nonlinearities occur in a wide variety of engineering applications. The most common environments include dynamical systems subject to contact or friction effects. The accurate modeling and simulation of such systems, namely accurate resolution near the times of non-smoothness or discontinuity, are critical because these types of systems feature discontinuous bifurcations, which are not found in smooth systems and which can lead to significant changes in system dynamics [1]. Grazing bifurcations, for example, occur when a system comes into contact with a soft or rigid wall, and a simple periodic motion may be transformed into multi-harmonic, quasiperiodic, or even chaotic motion with no warning. Common simulation approaches involve time-stepping a differential equation solver with or without a sub-routine to detect and locate events, but these can often be slow or inefficient due to increased overhead or requirements tailored to a specific type of system. In this work, we describe a method for event detection in ordinary differential equation (ODE) initial value problem (IVP) solvers using neural networks. The approach herein differs from that of [2], where the authors compute entire solutions of IVPs with an artificial neural network, since our approach uses neural networks to help reconstruct the IVP solution only at points of nonsmoothness. To the best of the authors' knowledge, the application of neural networks in this way and for this purpose is novel. Simple 1-dimensional systems with contact or friction are used in this work, but successful implementation can serve as a foundation to solve more complex systems in higher dimensions or with many points of contact or friction.

## 2. Neural network formulation

The following exemplar problem of a ball bouncing inelastically on a flat surface is used to illustrate our methods. The vector $(u, v)$, where $u$ is the ball's position and $v$ is the ball's velocity, are defined by:

$$\begin{aligned} \begin{pmatrix} \dot{u} \\ \dot{v} \end{pmatrix} &= \begin{pmatrix} v \\ -g \end{pmatrix} \; if \; u > 0 \\ v^{+} &= -Rv^{-} \; if \; u = 0 \end{aligned} \tag{1}$$

where $g$ is the gravitational constant and $R$ is the coefficient of restitution. We define the following event function for the bouncing ball problem:

$$E(u, v) = u \tag{2}$$

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

and let Σ denote the event surface, such that $\Sigma = \{(u,v): u = 0\}$. Let $\varphi = \varphi(u,h)$ denote a one-step IVP solver of order $p$ for integrating smooth initial value problems (IVPs) in the following sense:

$$\varphi(x_0,h) = x(t_0+h;t_0,u_0) + O(h^{p+1}), \quad h \ll 1 \tag{3}$$

where $x = x(t;t_0,x_0)$ denotes the solution to a smooth IVP $\dot{x} = f(x,t), x(t_0) = x_0$. We use a fixed time-step $\Delta t$ for integrating the bouncing ball, except at points where this time-step causes the numerical solution to cross the event surface. However, our method can be extended to an adaptive stepsize for improved efficiency. We construct a neural network model $M = M(u,v)$ for detecting events in time-integration of the bouncing ball problem, where:

$$M(u,v) = \begin{cases} 1 \; if \; E\left(\varphi\big((u,v),\Delta t\big)\right) \le 0 \\ 0 \; else \end{cases} \tag{4}$$

The model $M$ is 1 if an event has occurred and 0 otherwise. To train the model, we sample the event surface $\Sigma_N = \{\sigma_j : j = 1,\dots,N\}$ and let $\Delta T_K = \{\Delta t_j : j = 1,\dots,K\}$ denote sample time-steps where $\Delta t_1 < \Delta t_2 < \cdots < \Delta t_K$. We then define the input training set $X$ for our neural network model as follows:

$$X = \varphi(\Sigma_N, -\Delta T_K) = \{X_{j,k} : X_{j,k} = \varphi(\sigma_j, \Delta t_k), j = 1,\dots,N, k = 1,\dots K\} \tag{5}$$

The output training set $Y$ for our neural network model is then defined as follows:

$$Y = \left\{Y_{j,k} : Y_{j,k} = \begin{cases} 1 \; if \; E\left(\varphi\big(X_{j,k},h\big)\right) \le 0 \; for \; some \; h \in \Delta T \\ 0 \; else \end{cases}, j = 1,..,N, k = 1,\dots K\right\} \tag{6}$$

The input data sets are divided into subsets $X = XTrain \cup XTest$ and $Y = YTrain \cup YTest$, where $XTest, YTest$ are the testing data sets that contain 10% of the elements of $X, Y$ with the remaining 90% of the elements being in training data sets $XTrain, YTrain$. We implement a neural network classifier using the Julia Flux library. The network has 5 densely connected hidden layers with *relu* activation functions and we use the *logitbinarycrossentropy* loss function. The network is trained using the ADAM method for $10^6$ iterations, at which point the trained network is 100% accurate on the testing data sets, i.e., $M(x) = y$ is correct for all $(x,y) \in XTest \times YTest$.

## 3. Results

To test our method, we integrate the bouncing ball IVP using the trained event detection neural network $M$ "neural network approach" and compare to results using a "simple approach", where a trial solution is computed at each time-step and an event is detected if the first component of the trial solution is nonpositive. In the neural network methods when an event is detected, the bisection method with absolute and relative tolerance of $10^{-10}$ is used to approximate the exact time when the event

occurs and the velocity is instantaneously updated as $v \leftarrow -Rv$ at these times. In the simple approach, MATLAB *ode45* with its event location feature is used. We use $R = 0.9$, $u_0 = 10.0$, and $g = 9.81$ for the constant values. In Figure 1 and Table 1, the event times and solution values for both the neural network approach and simple approach were effectively identical.

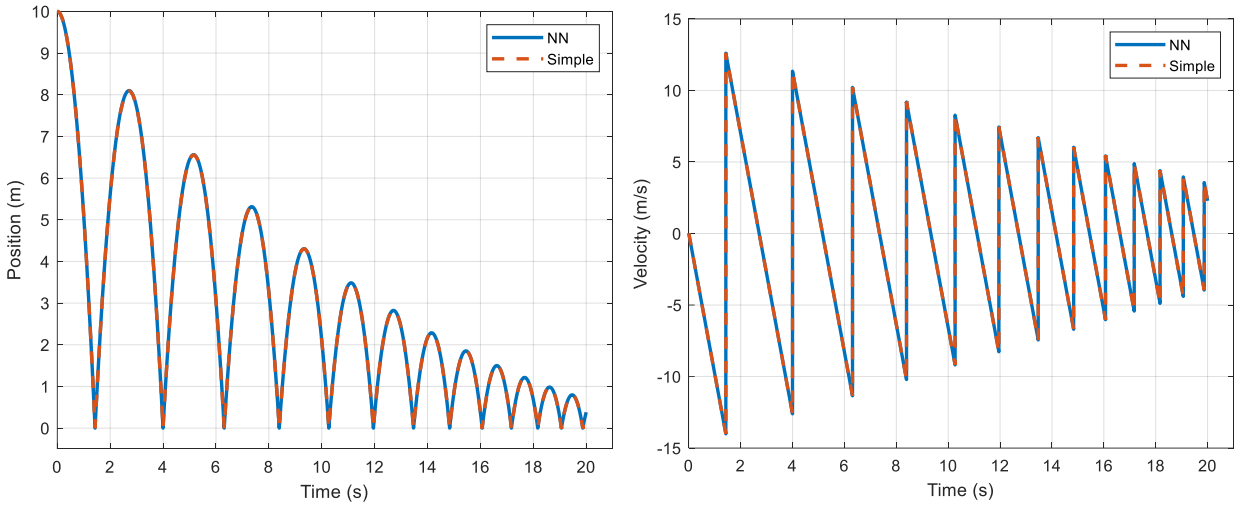


Fig 1. $u\ vs.\ time$ (left) and $v\ vs.\ time$ (right) for a solution computed using a neural network or a simple approach to determine event times.

Table 1: Comparison of event times between the two simulation approaches.

| Neural network | Simple approach |
|---|---|
| 1.427843122926641 | 1.427843122927093 |
| 3.997960744196168 | 3.997960744195887 |
| 6.311066603338677 | 6.311066603337761 |
| 8.392861876565325 | 8.392861876565391 |
| 10.266477622465255 | 10.266477622470353 |
| 11.952731793775415 | 11.952731793784908 |
| 13.470360547954783 | 13.470360547967886 |
| 14.836226426716886 | 14.836226426732534 |
| 16.065505717603227 | 16.065505717620926 |
| 17.171857079401377 | 17.171857079420576 |
| 18.167573305020159 | 18.167573305040214 |
| 19.063717908075276 | 19.063717908097843 |
| 19.870248050826223 | 19.870248050849643 |

## 4. Conclusions

Preliminary results indicate the neural network can accurately predict the event times for the bouncing ball problem. In the conference presentation, additional results will be presented for more complex 1D dynamical systems. These results indicate the neural network approach can be extended to more complex system with higher dimensions or multiple points of contact and friction.


## Acknowledgments

Supported by the Laboratory Directed Research and Development program at Sandia National Laboratories, a multi-mission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC (NTESS), a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration (DOE/NNSA) under contract DE-NA0003525. This written work is authored by an employee of NTESS. The employee, not NTESS, owns the right, title and interest in and to the written work and is responsible for its contents. Any subjective views or opinions that might be expressed in the written work do not necessarily represent the views of the U.S. Government. The publisher acknowledges that the U.S. Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this written work or allow others to do so, for U.S. Government purposes. The DOE will provide public access to results of federally sponsored research in accordance with the DOE Public Access Plan. SAND2023-06682C.


## References


[1] Leine, R.I. and Nijmeijer, H., *Dynamics and bifurcations of non-smooth mechanical systems (Vol. 18)*, Springer Science & Business Media, 2013.

[2] Chen, Ricky TQ, Brandon Amos, and Maximilian Nickel. "Learning Neural Event Functions for Ordinary Differential Equations." *International Conference on Learning Representations*. 2020.