# High-rate Structural Health Monitoring: Part-II Embedded System Design

Austin Downey[1,2], Jason Bakos[3]

[1]Department of Mechanical Engineering
[2]Department of Civil and Enviormental Engineering
[3]Department of Computer Science and Engineering
University of South Carolina, Columbia, SC 29208

**ABSTRACT**
Engineering systems designed to operate in extreme dynamic and complex environments face stringent timing deadlines when it comes to implementing actions decided upon by an autonomous decision-maker. Importantly, the short time scales of the considered applications require the deterministic transfer between the different domains of a control scheme; mainly the domains related to decision-maker, model-updating, feature extraction, and signal filtration. Moreover, the development of these embedded systems are limited by the Size, Weight, Power, Cost, and Cooling (SWaPC2) of their parent applications. Part II of this III part tutorial will provide attendees with an organized discussion of the challenges related to the deterministic transfer of knowledge across controller domains considering the required stringent latencies. This tutorial will cover the basics of high-rate machine learning (HRML) and real-time modeling techniques deployed to heterogeneous computing systems that are likely to include real-time operating systems (RTOS), field programmable gate arrays (FPGA), and shared memory frameworks for data acquisition systems. Experimental results from the deployment of benchmark algorithms to realistic hardware will be discussed.

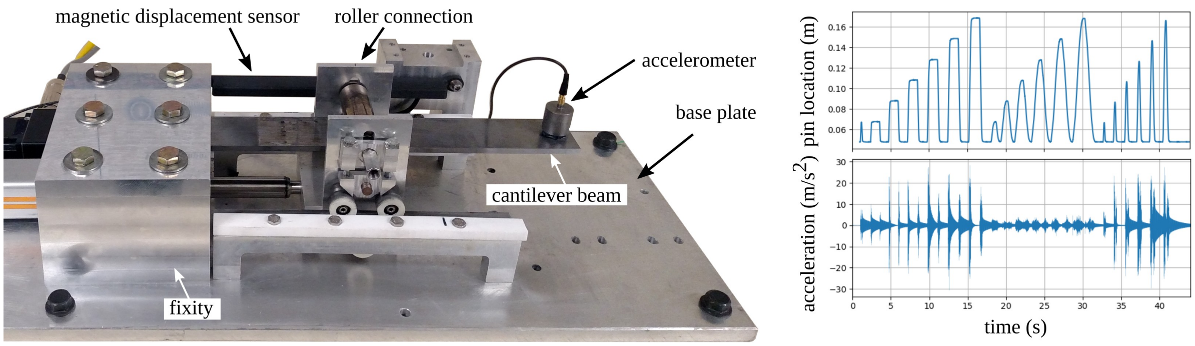
**Keywords:** hardware, high-rate, sensing, edge computing, shock

## INTRODUCTION

Enabling real-time artificial intelligence for active structures operating in high-rate dynamic environments would unleash numerous smart structure applications. To meet the high-rate challenge, these solutions must be capable of meeting the 1 ms latency from event detection to decision-making [1]. The damage detection and prognostics algorithms designed to run on the embedded hardware that makes up these systems must be co-designed with the hardware to ensure the required information makes it to the automated real-time decision-maker on time. These strict latency constraints necessitate the need for selecting the correct computing environments for the embedded system.

The selection of optimal computing hardware for the embedded system is also driven by the latency requirements of the high-rate mission. Hardware must have sufficient processing power and memory bandwidth to facilitate the required algorithms. However, the over-deployment of resources will result in less-efficient systems with excessive power consumption and thermals that must be accounted for. Thereby increasing system weight and space.

This extended abstract presents preliminary results with respect to algorithm latency and hardware selection for algorithm deployment. Both challenges are presented within the context of enabling real-time artificial intelligence for active structures. The contribution of this work is the consideration of these constraints within the future development of embedded systems.



(a) (b)

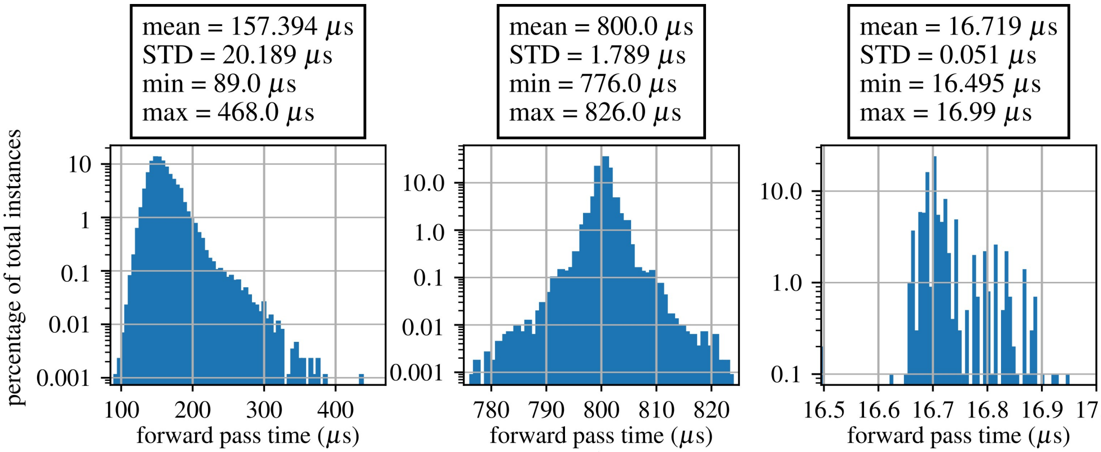
Figure 1: The DROPBEAR setup and data used in this work as a stand-in for a high-rate structural system, showing: (a) the testbed, and; (b) dynamic pin location and acceleration data collected from the testbed.

## BACKGROUND

This work uses a dataset of progressive structural changes (i.e. degradation) that was developed using the Dynamic Reproduction of Projectiles in Ballistic Environments for Advanced Research (DROPBEAR) testbed shown in Figure 1. DROPBEAR was initially built and validated by Joyce et al. [2]. It features a movable roller boundary condition attached to a linear actuator that is used to simulate damage to the structure. The advantage of the DROPBEAR testbed is that it is capable of repeatedly altering test parameters as the parameter can be changed during a test, allowing researchers to gain better insights into the system's temporal responses. Datasets developed by this system have been used extensively for studying the challenge of high-rate structural health monitoring [1]. This work focuses on using the DROPBEAR testbed to investigate real-time state estimation of the structure. First, a data-driven approach based on Long short-term memory (LSTM) deep learning models [3] is used to investigate the impact of different operating systems and hardware on latency constraints. Second, the effects of computational hardware selection and algorithm scalability are investigated using a physics-based model updating scheme [4]. For brevity, details about the discussed methodologies and their respective implementation are omitted and the reader is encouraged to see prior works.

## ANALYSIS

Figure 2 reports the results from a timing study for LSTM models built to ingest the acceleration data from the DROPBEAR testbed (shown in the bottom-right of Figure 1) and returns the inferred displacement of the pin in the DROPBEAR testbed (shown in the top-right of Figure 1). Figure 2(a) reports the timing distribution of the LSTM model solved for on a general purpose operating system (GPOS). In this work, a Window's 10 desktop machine with an Intel® Core© i7-10700K processor with a 3.80 GHz base clock and 64 GB of RAM. Figure 2(b) reports the timing study for the LSTM model running on a 1.33 GHz Dual-Core Intel Atom (E3825) utilizing the NI-Linux real-time operating system (RTOS). Lastly, Figure 2(c) reports the timing study for an LSTM model running on a field programmable gate array (FPGA).



(a) (b) (c)

Figure 2: Timing distributions for an LSTM model running on the DROPBEAR data.

Figure 2 shows the difference in timing distributions between the three methods for the LSTM-based state estimator. The GPOS (Figure 2(a)) has a long tail to the right which demonstrates the uncertainty in latency that is present on a GPOS. In contrast, the timing distribution for the RTOS (Figure 2(b)) is tightly bound in a Gaussian distribution around the mean forward pass time. Lastly, the FPGA-based implementation has only limited jitter in the latency due to the inherent deterministic timing of the FPGA. The jitter that is present in the FPGA implementation is a result of the MicroBlaze soft-core processor that is implemented in the FPGA. An important note for these results is that while the GPOS results are significantly faster than that of the RTOS, this is due to the hardware each model was run on and not the chosen operating system. However, the timing distribution is a function of the chosen operating system.

The selection of appropriate hardware when scaling algorithmic systems for deployment is discussed next. Figure 3 demonstrates the need to consider hardware constraints when developing systems. For this work, a model order reduction technique [4] is used to reconstruct the physical system using only a subset of the system's modes to reduce computation time. Figure 3(a) shows an algorithmic overview of the proposed system while Figure 3(b) reports on the performance space for the algorithm implemented on an Intel® Core© i7-10700K processor. Note the large performance discontinuity around 250 nodes where pre-discontinuity the incremental cost per finite element analysis (FEA) model, represented by the slope in the line, is substantially less than that post-discontinuity. This is in addition to the baseline offset present.

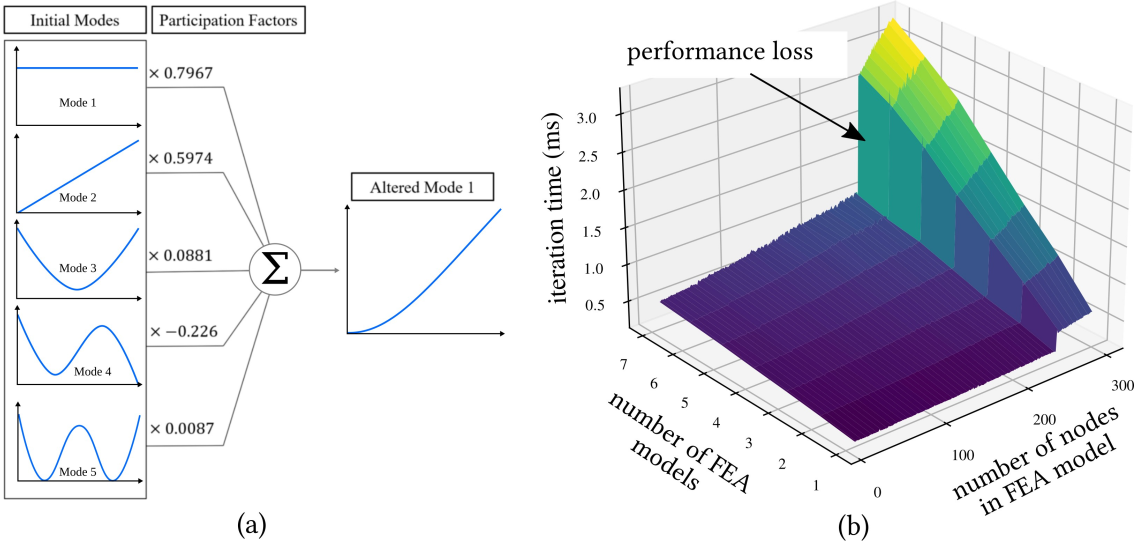


Figure 3: Real-time physics-based model updating of the DROPBEAR system showing: (a) an overview of how reduced order modeling can be achieved through calculating modal contributions, and; (b) timing results for a number of modes showing an extreme discontinuity in algorithm performance with scaling due to hardware limitations.

## CONCLUSION

This work presented two examples that demonstrate the importance of considering the potential hardware in the design of algorithms for embedded deployment on high-rate systems. This need for hardware/software co-design was first demonstrated through a study of timing distributions for general-purpose operating systems (GPOS), real-time operating systems (RTOS), and field programmable gate arrays (FPGA). In decision-making algorithms, the consistent timing of sub-systems is of critical importance to ensure that the best possible actions are taken. The second example demonstrates the need to consider the hardware for future deployments when developing systems with stringent timing constraints, particularly when systems are intended to be scaled.

## ACKNOWLEDGEMENTS

This material is based upon work supported by the Air Force Office of Scientific Research (AFOSR) through award no. FA9550-21-1-0083. This work is also partly supported by the National Science Foundation grant numbers 1850012, 1937535, and 1956071. Any opinions, findings, and conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation or the United States Air Force.

## REFERENCES

[1] Jacob Dodson, Austin Downey, Simon Laflamme, Michael D. Todd, Adriane G. Moura, Yang Wang, Zhu Mao, Peter Avitabile, and Erik Blasch. “High-rate structural health monitoring and prognostics: An overview”. In *Data Science in Engineering, Volume 9*, pages 213–217. Springer International Publishing, oct 2021.

[2] Bryan Joyce, Jacob Dodson, Simon Laflamme, and Jonathan Hong. “An experimental test bed for developing high-rate structural health monitoring methods”. *Shock and Vibration*, 2018:1–10, jun 2018.

[3] Joud Satme, Daniel Coble, Braden Priddy, Austin R.J. Downey, Jason D. Bakos, and Gurcan Comert. “Progress towards data-driven high-rate structural state estimation on edge computing devices”. In *Proceedings of the ASME 2022 International Design Engineering Technical Conferences*, 2022.

[4] Seong Hyeon Hong, Claire Drnek, Austin Downey, Yi Wang, and Jacob Dodson. “Real-time model updating algorithm for structures experiencing high-rate dynamic events”. In *ASME 2020 Conference on Smart Materials, Adaptive Structures and Intelligent Systems*. American Society of Mechanical Engineers, sep 2020.