Rank Reduction of LSTM Models for Online Vibration Signal Compensation on Edge Computing Devices

Josh McGuire; Department of Mechanical Engineering

Joud N. Satme; Department of Mechanical Engineering

Daniel Coble; Department of Mechanical Engineering

Austin R.J. Downey, Department of Mechanical, Civil and Environmental Engineering

Jason Bakos, Department of Computer Science and Computer Engineering

Ryan Yount; Department of Mechanical Engineering

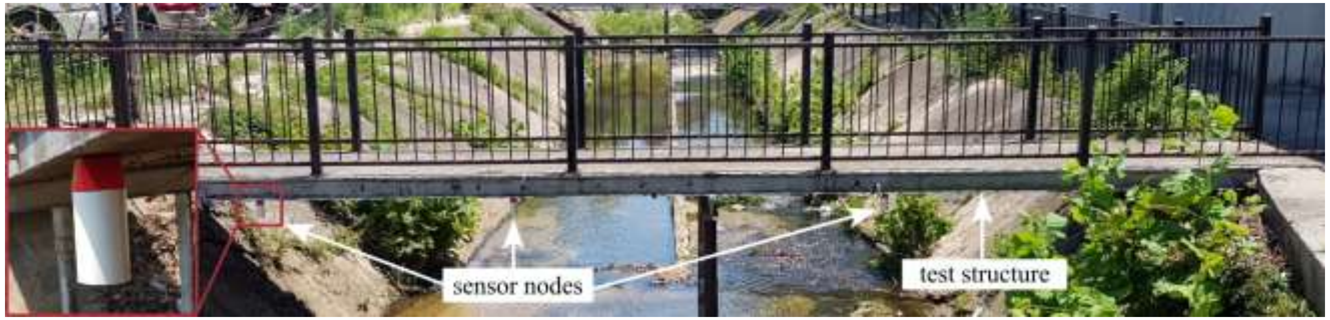
Arion Pons, Department of Mechanical and Maritime Sciences, Chalmers University

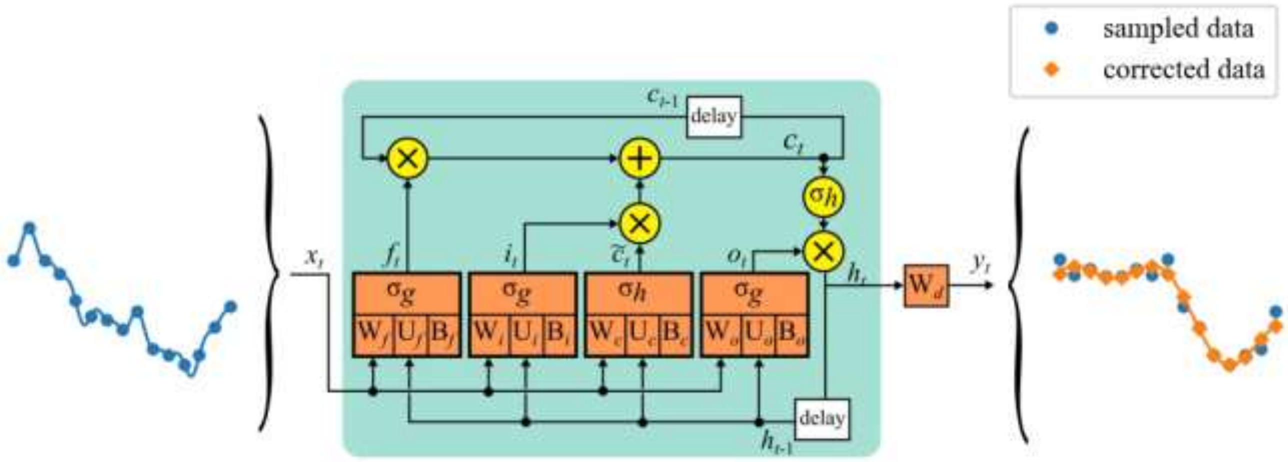
UNIVERSITY OF South Carolina

Molinaroli College of Engineering and Computing

Outline

- Background
  - Structural health monitoring
  - UAV-deployable sensor package
  - Vibration signal compensation
  - Edge machine learning
  - Model compression
  - Long Short-Term Memory
- Methodology
  - Degrees-of-freedom decomposition
  - Collecting training data
  - Bench top experiment
  - Training the model
  - Rank reduction
- Results and discussion
  - Model performance
  - Future work





Background

Structural Health Monitoring

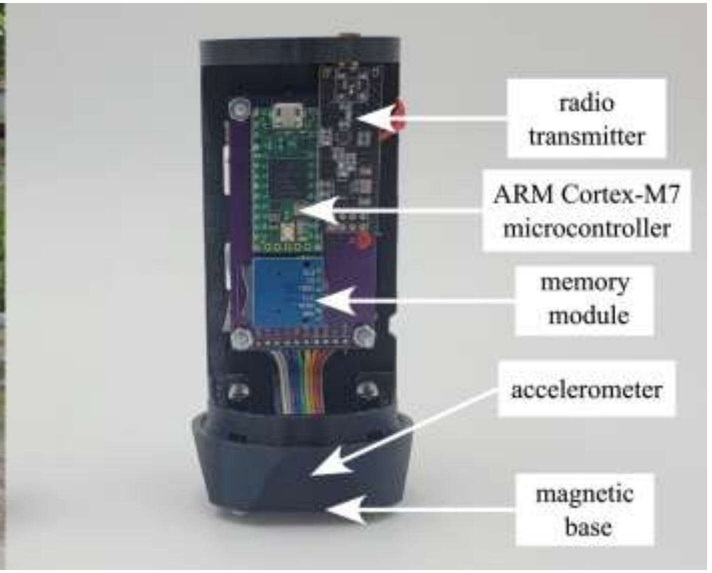
Structural Health Monitoring (SHM) is the process of assessing the integrity of structures in real time. It has the following benefits:

- Enables early warning for structural failure
- Provides insights into how the structure responds to changing conditions
- Collects data to inform future designs

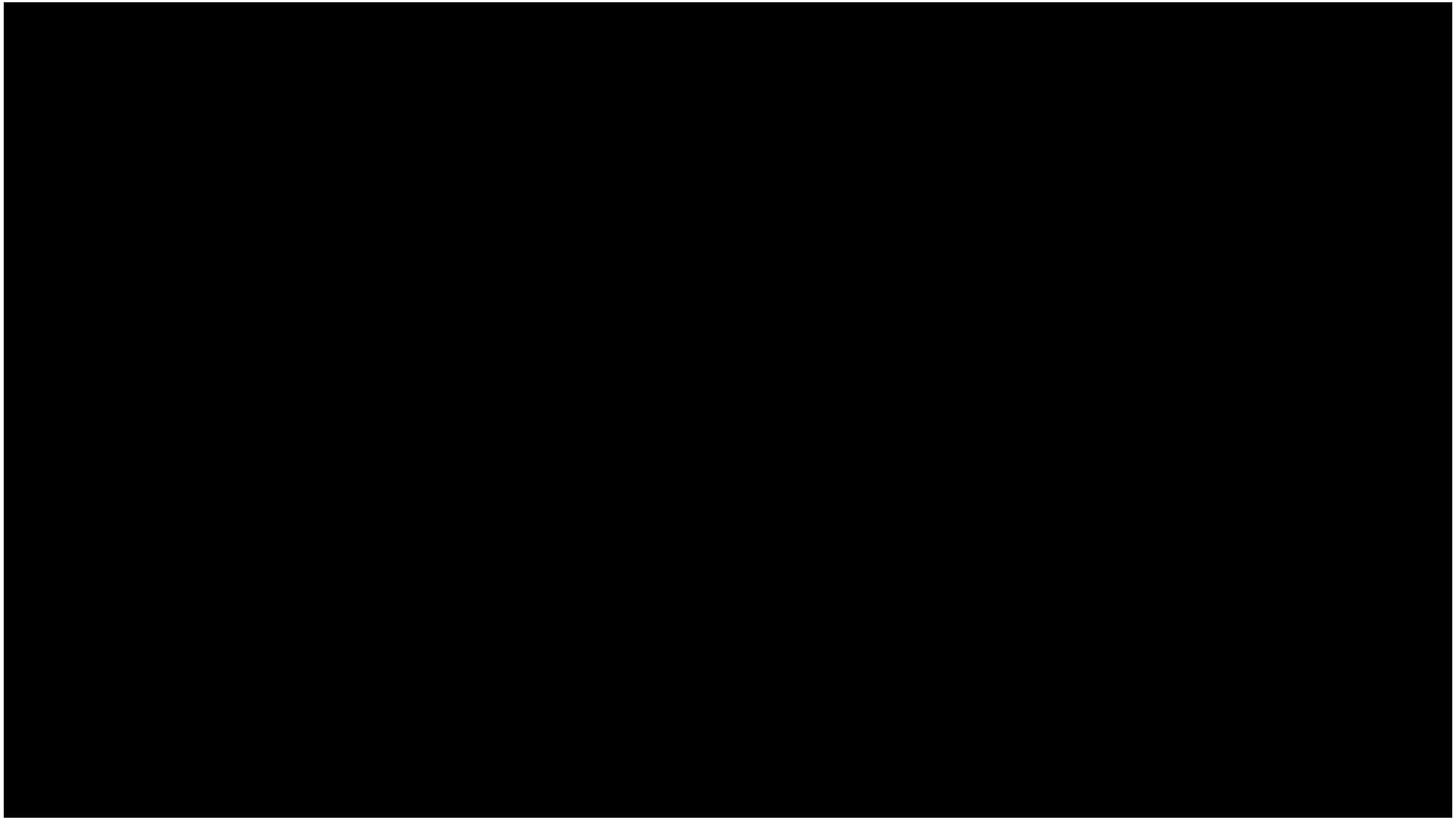
UAV-Deployable Sensor Package

Brooklyn SHM sensing node features:

- **Architecture:** 1x Arm Cortex-M7 at 600MHz with FPU, 1024 KB memory
- **Sensors:** MEMS accelerometer for vibration sensing
- **Deployment:** Magnetic base enables it to be attached to the structure using a drone



Sensor deployment and retrieval mission

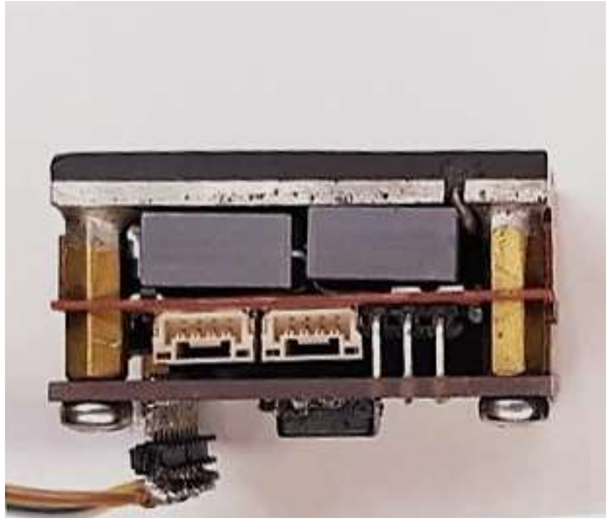


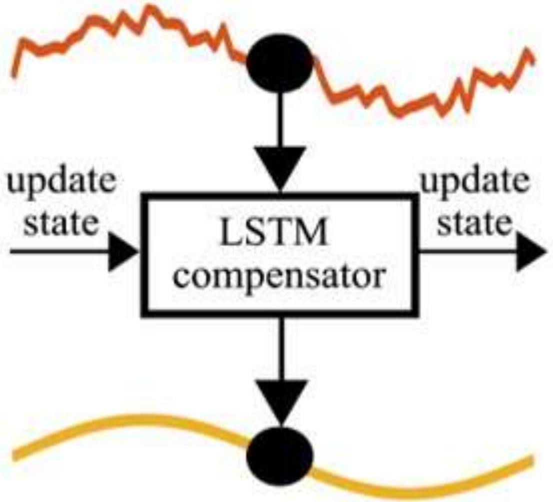
Vibration Signal Compensation

Challenges facing SHM sensing nodes:

- **Transmissibility loss:** low-frequency vibration information may struggle to reach the sensing node through the attachment point.
- **Cost:** SHM nodes need to be cost-effective, leading to sensor quality compromises

LSTM-based signal compensators have been shown to be effective at mitigating these problems.





Edge Machine Learning

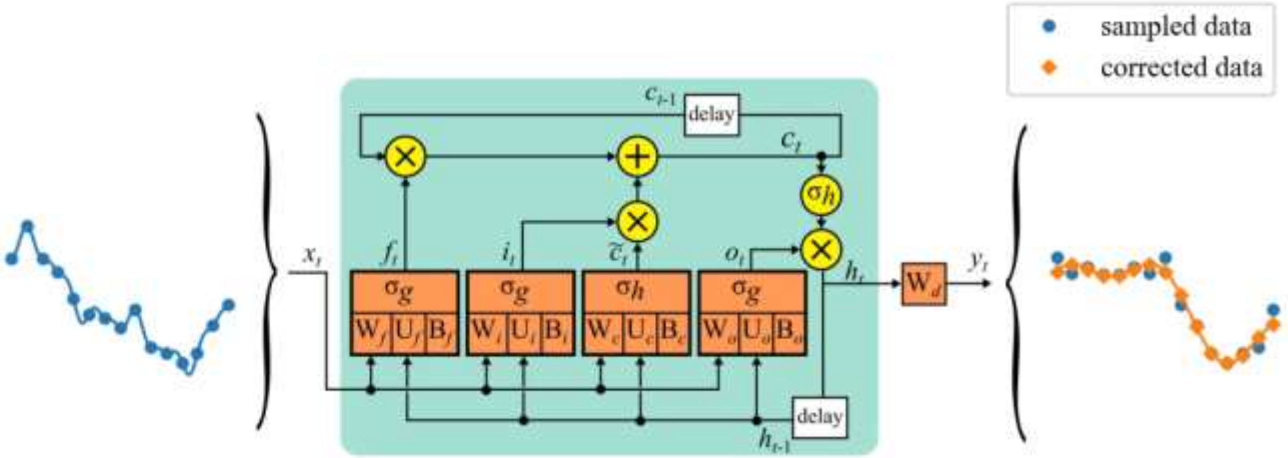
Why process data at edge?

- **Efficiency:** IoT devices like SHM sensing nodes are responsible for producing much of the world's data. Off-line processing puts a strain on cloud computing infrastructure.
- **Reduce transmission:** SHM sensing nodes have limited battery capacity. Processing data on edge saves power by minimizing radio use.
- **Location:** Some sensing nodes need to be placed in areas where constant communication is not possible.

Model Compression

Why compression is important:

- **Memory footprint:** Edge devices are often equipped with minimal memory, making the savings provided by compression essential.
- **Latency:** Compression may allow inference to be completed faster.
- **Complexity:** Compression enables models initially too complex for constrained edge devices to be deployed.



Long Short-Term Memory

- LSTM is a form of recurrent neural network that uses a gated structure to determine what information to retain and "forget".
- The complex structure of LSTM makes it an ideal candidate for model compression.

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f),$$

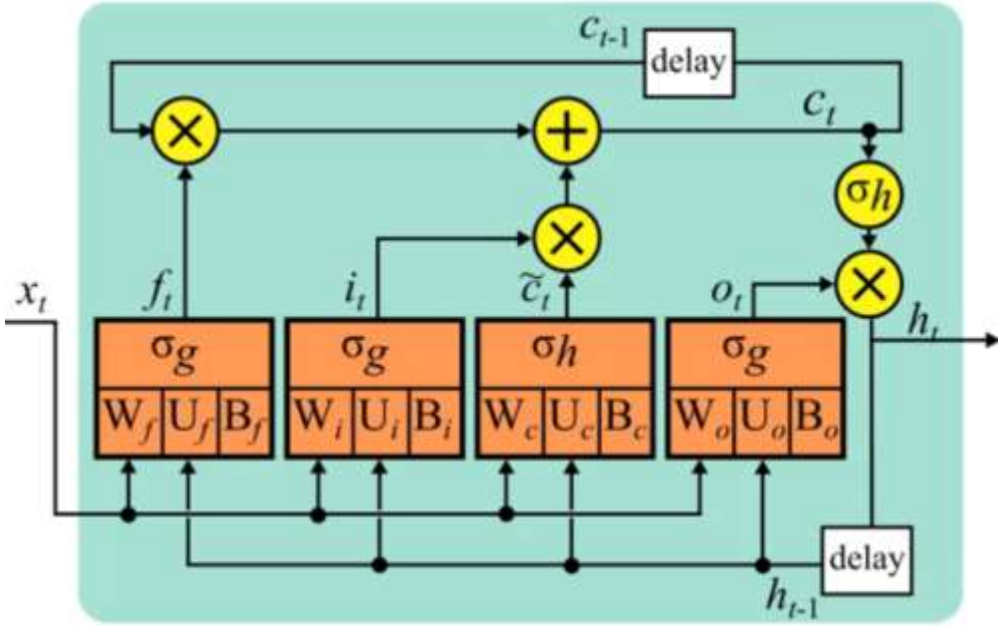
$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i),$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o),$$

$$\bar{c}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c),$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \bar{c}_t,$$

$$h_t = o_t \circ \tanh(c_t).$$



f: forget gate
i: input gate
o: output gate
c: carry state
h: hidden state / inference output

$\circ$: element-wise multiplication
σ: sigmoid activation function

Long Short-Term Memory

- Typical LSTM inference requires four separate weight matrices (assuming W and U are concatenated) – one for each gate. This leads to four separate matrix-vector multiplications being required for inference.
- These weight matrices can be combined to enable inference to be performed in a single matrix-vector multiplication.
- The consolidated weight matrix simplifies the model compression process, as only one matrix has to be compressed.

$$W = \begin{bmatrix} W_i & W_f & W_c & W_o \\ U_i & U_f & U_c & U_o \end{bmatrix},$$

$$b = \begin{bmatrix} b_i \\ b_f \\ b_c \\ b_o \end{bmatrix},$$

$$y = \begin{bmatrix} x \\ h \end{bmatrix},$$

$$\begin{bmatrix} z_i \\ z_f \\ z_c \\ z_o \end{bmatrix} = Wy + b,$$

$$\begin{bmatrix} i_t \\ f_t \\ \bar{c}_t \\ o_t \end{bmatrix} = \begin{bmatrix} \sigma(z_i) \\ \sigma(z_f) \\ \tanh(z_c) \\ \sigma(z_o) \end{bmatrix}.$$

The Driving Challenge

- The matrix-vector product Wy is the driving computation in the LSTM in terms of cost.
- We can make LSTMs fit on smaller processors and run faster if we can reduce the complexity of this matrix-vector multiply.

$$\boxed{Wy} + b$$

$$W = \begin{bmatrix} W_i & W_f & W_c & W_o \\ U_i & U_f & U_c & U_o \end{bmatrix} \qquad y = \begin{bmatrix} x \\ h \end{bmatrix}$$

Quantization

Quantization is the process of converting floating-point values to lower-precision fixed-point values.

Advantages:

- Significant reduction in memory footprint
- Preserves dense matrices
- May improve latency if platform lacks a performant FPU

Disadvantages:

- Inference still takes the same amount of adds and multiplies
- For non-parallel architectures with a performant FPU, latency is not improved

8.114	4.626
1.231	3.993

8	5
1	4

Pruning

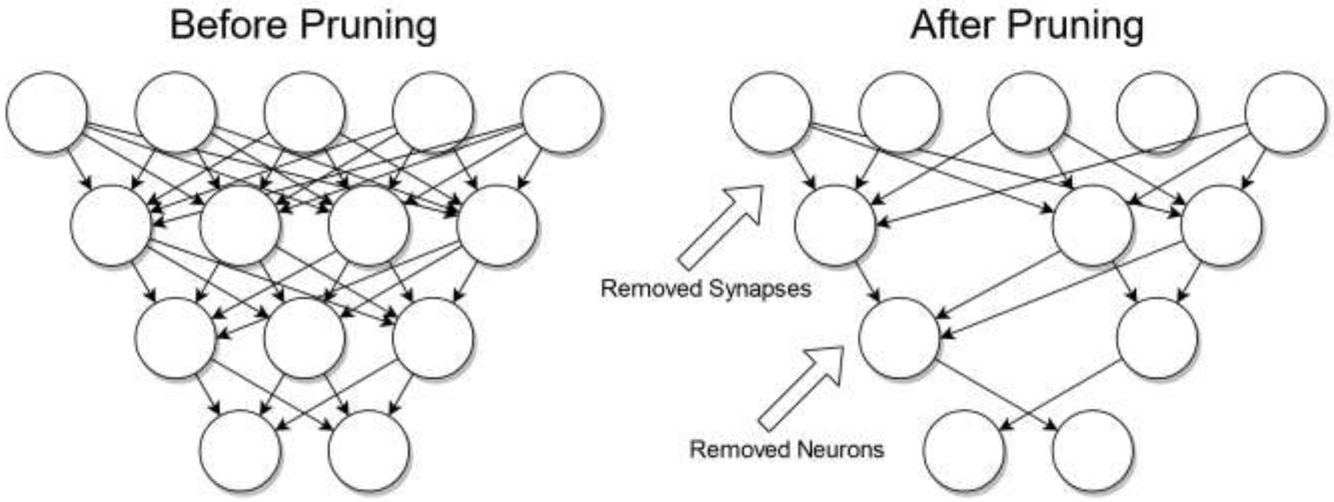
Pruning is the process of eliminating weights close to zero.

Advantages:

- Control over error threshold

Disadvantages:

- Sparse matrices are less efficient on general-purpose computers due to cache issues



Low-Rank Approximation

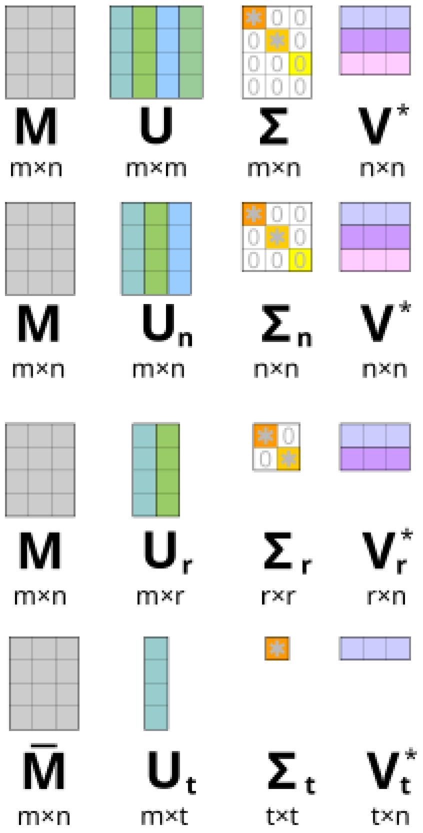
Low-rank approximation is the process of representing the information in a matrix with another matrix of a lower rank.

Advantages:

- Preserves dense matrices
- Control over error threshold
- Both saves space and improves latency

Disadvantages:

- Direct use of singular value decomposition (SVD) only provides benefits after over half the ranks are removed, which may not be possible



Methodology

Stating the Challenge

Problem: In traditional low-rank approximation, the truncated SVD is used directly for inference. This means we only save space after more than half the ranks of *W* in the LSTM are removed, which may be impractical.

Question: Can we create a more efficient way to store the information in *W,* enabling the deployment of LSTMs on smaller edge devices?

Degrees-of-Freedom Decomposition

Mathematical Reasoning:

- For a matrix with rank $r<\min(m,n)$, all rows exist in the span of only r ‘basis’ rows
- The remaining $m-r$ rows may be written as a transform of these ‘basis’ rows.

Using this information, we construct the following two-step process:

$$Ax_1 = Bx,$$
$$Ax_2 = Cx_1,$$
$$Ax = P\begin{bmatrix} Ax_1 \\ Ax_2 \end{bmatrix}$$

Here, B is an $r\times m$ matrix, C is $(n-r)\times r$, and P is a permutation matrix used to chose the r basis rows. Now, instead of storing U, Σ, and V^T , we only need to store B and C.

Degrees-of-Freedom Decomposition

Computing B and C

Split U into $\begin{bmatrix} U_1 \\ U_2 \end{bmatrix}$, where U_1 is $r \times r$,

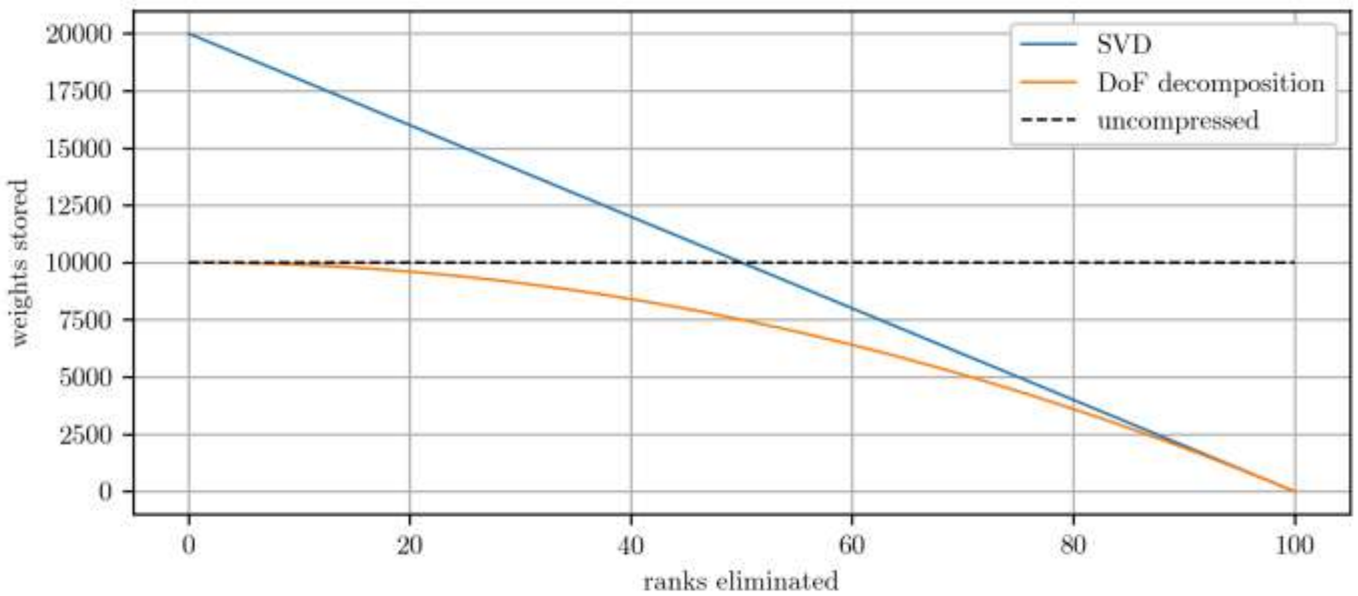
$B = U_1 \Sigma V^{\mathrm{T}}$,

$C = U_2 U_1^{-1}$.

Weights stored: $r \times m + (n - r) \times r$

Multiplies: $mn - (m - r)(n - r)$

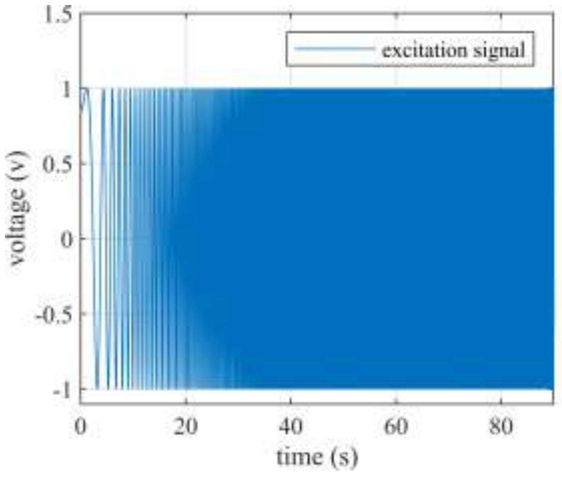
Adds: $(mn - n) - (m - r)(n - r)$

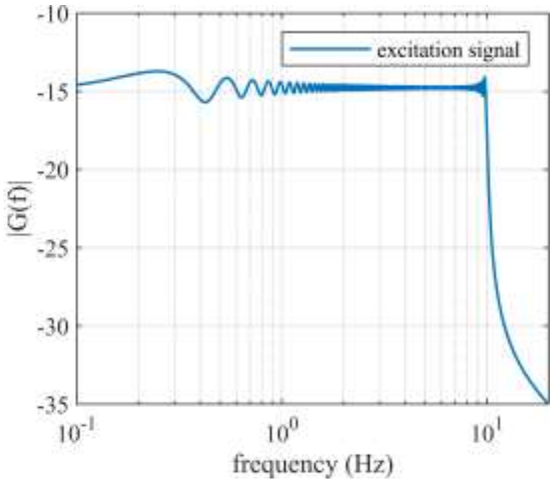


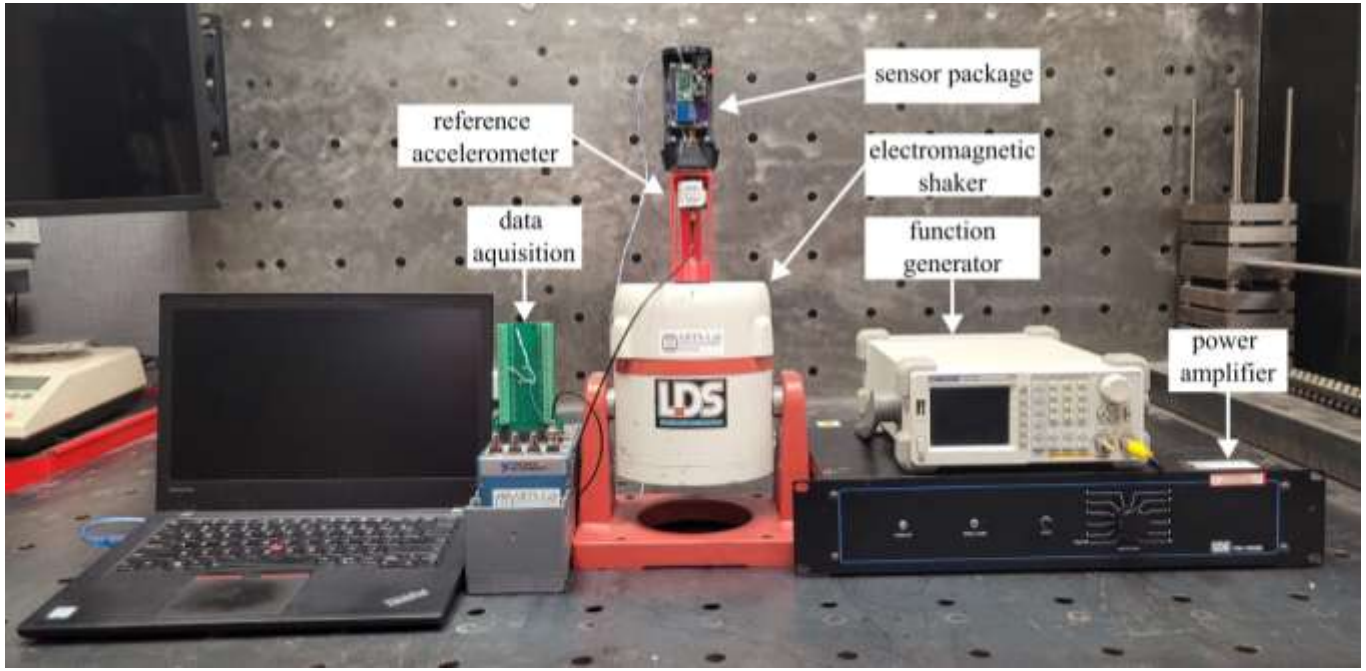
Using the DoF Decomposition, we see both memory and computational savings immediately, rather than after a set number of reductions.

Collecting Training Data

- A function generator is connected to an electromagnetic shaker.
- The sensor package is attached to a higher-quality reference accelerometer.
- The electromagnetic shaker was excited with frequency sweeps from 1-10 Hz.
- Phase between the two signals was aligned through interpolation.

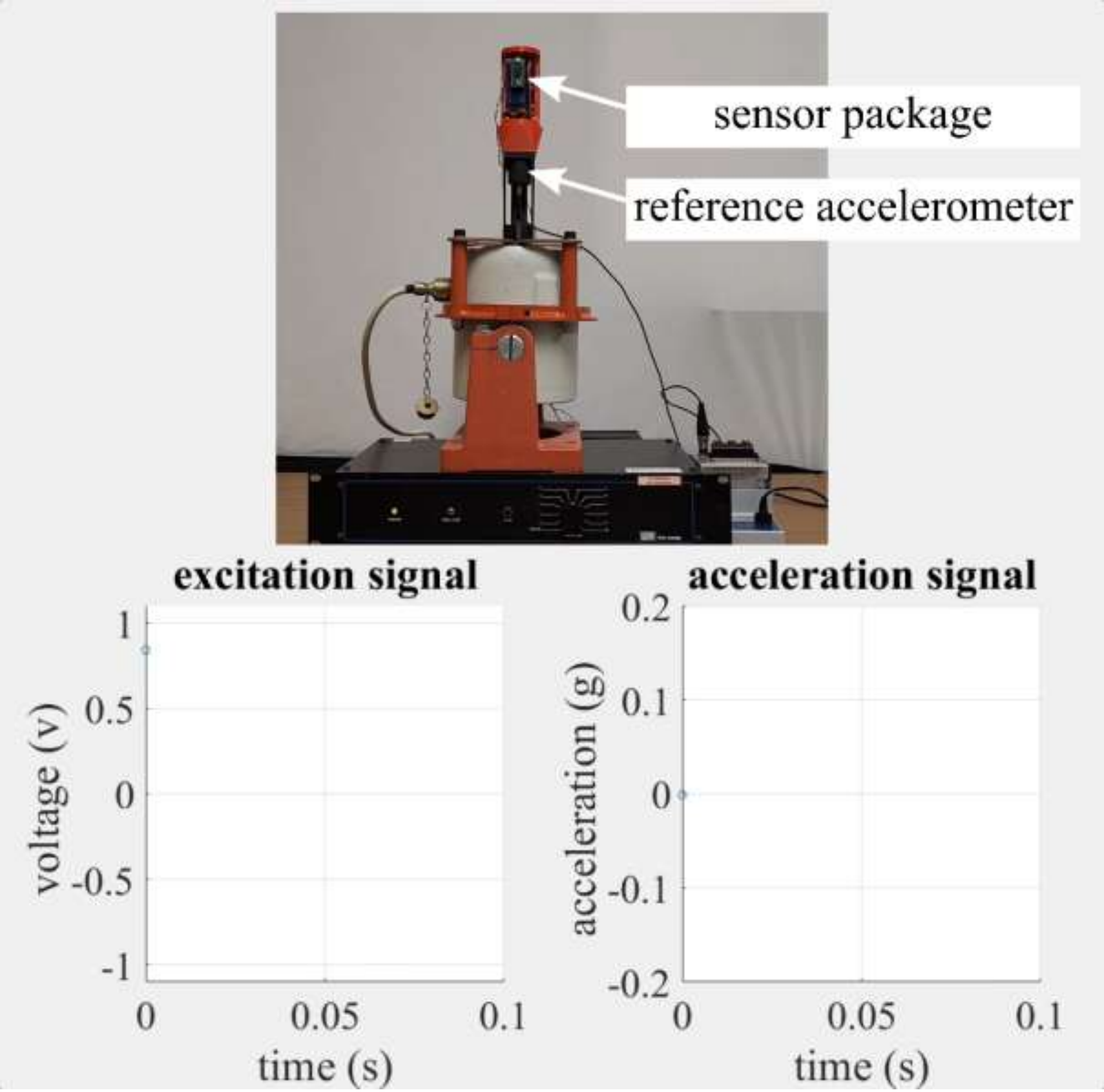






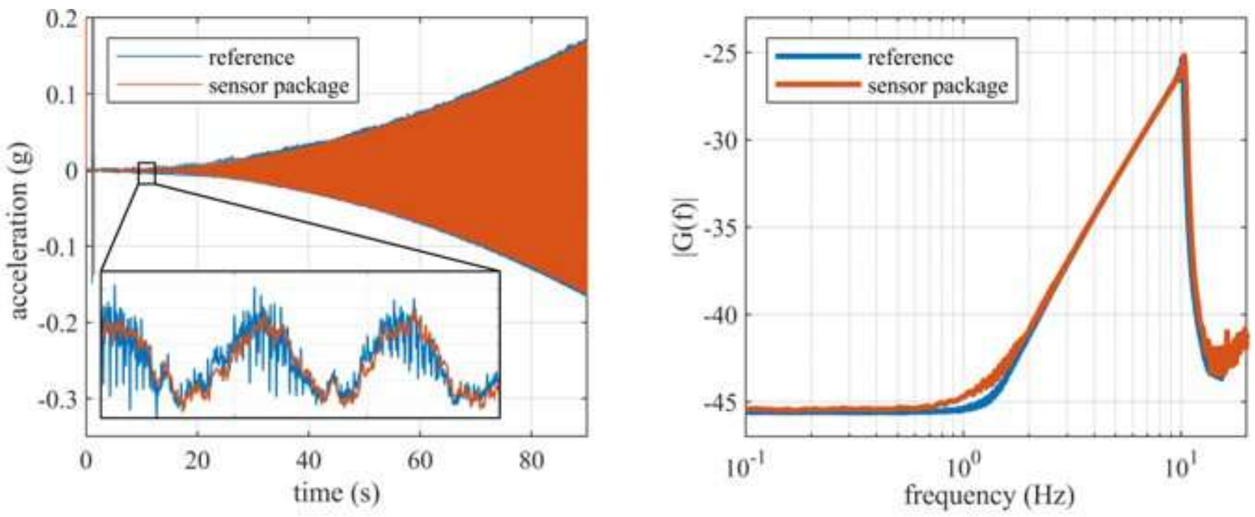
Bench top experiment

- Chirp excitation is fed into the electromagnetic shaker using an analog output module
- A data acquisition is used to record reference acceleration
- A digital trigger is set to synchronize both the reference accelerometer and sensor package
- Various dynamic ranges were used to expand the training range of the LSTM model



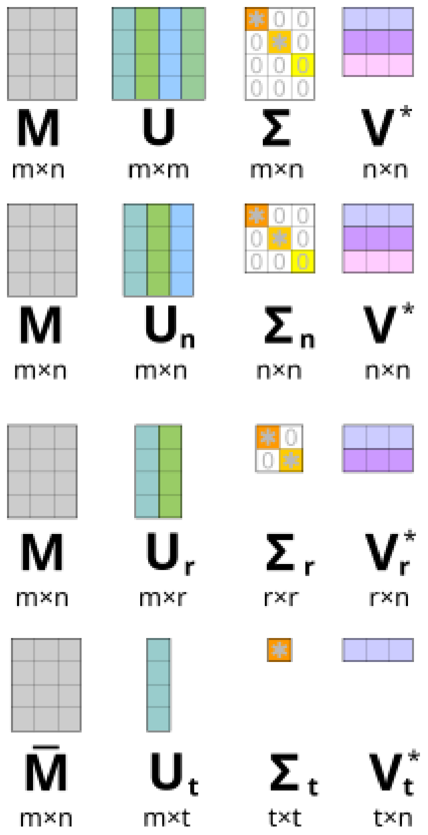
Training the Model

- **Model hyperparameters:** 50 unit, 1 input LSTM connected to a dense layer.
- Package data is fitted to the reference data in the time domain.
- Windowing is employed to reduce overfitting.
- Validated on a testing dataset.



Rank Reduction

- To prove the efficacy of the DoF decomposition, we employ the truncated SVD for rank selection.
- Weight matrix rank was reduced to 41 from 51.
- The B and C matrices for the DoF decomposition were then calculated.

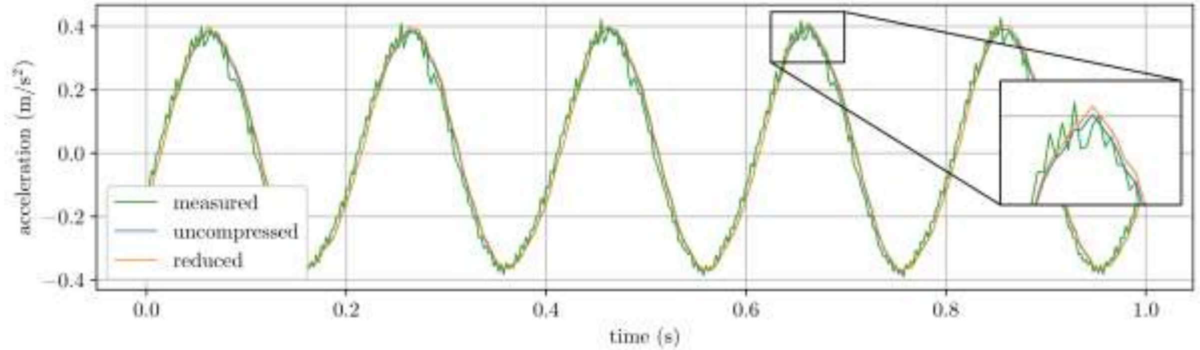


Results and Discussion

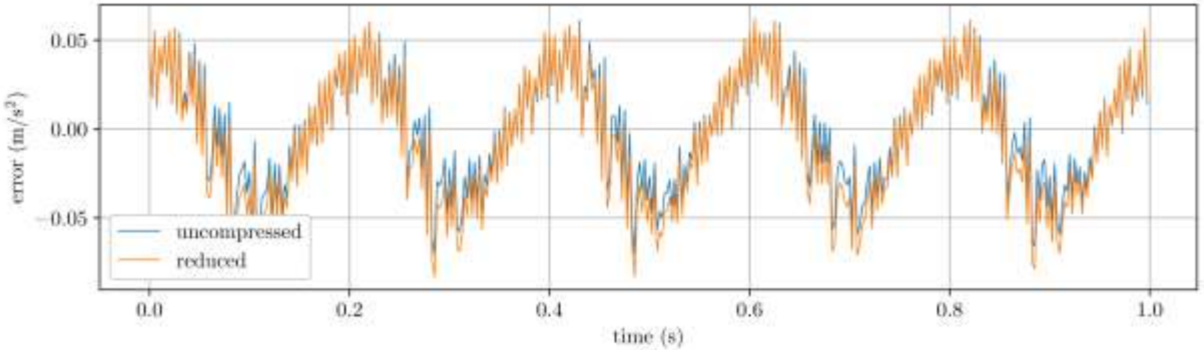
Model Performance

testing	SNR_{dB}	MSE	RMSE	MAE	TRAC	Parameters
uncompressed	14.8498	0.0005	0.0221	0.0168	0.9673	10451
compressed	14.0765	0.0006	0.0242	0.018	0.9616	8861
difference	0.7733	-0.0001	-0.0021	-0.0012	0.0057	1590
% difference	5.3466	17.7586	8.8969	7.1053	0.5925	16.4664

- SNR remained acceptable
- TRAC remained high, indicating strong similarity between the reference signal and the signal generated by the compressed model



(a)

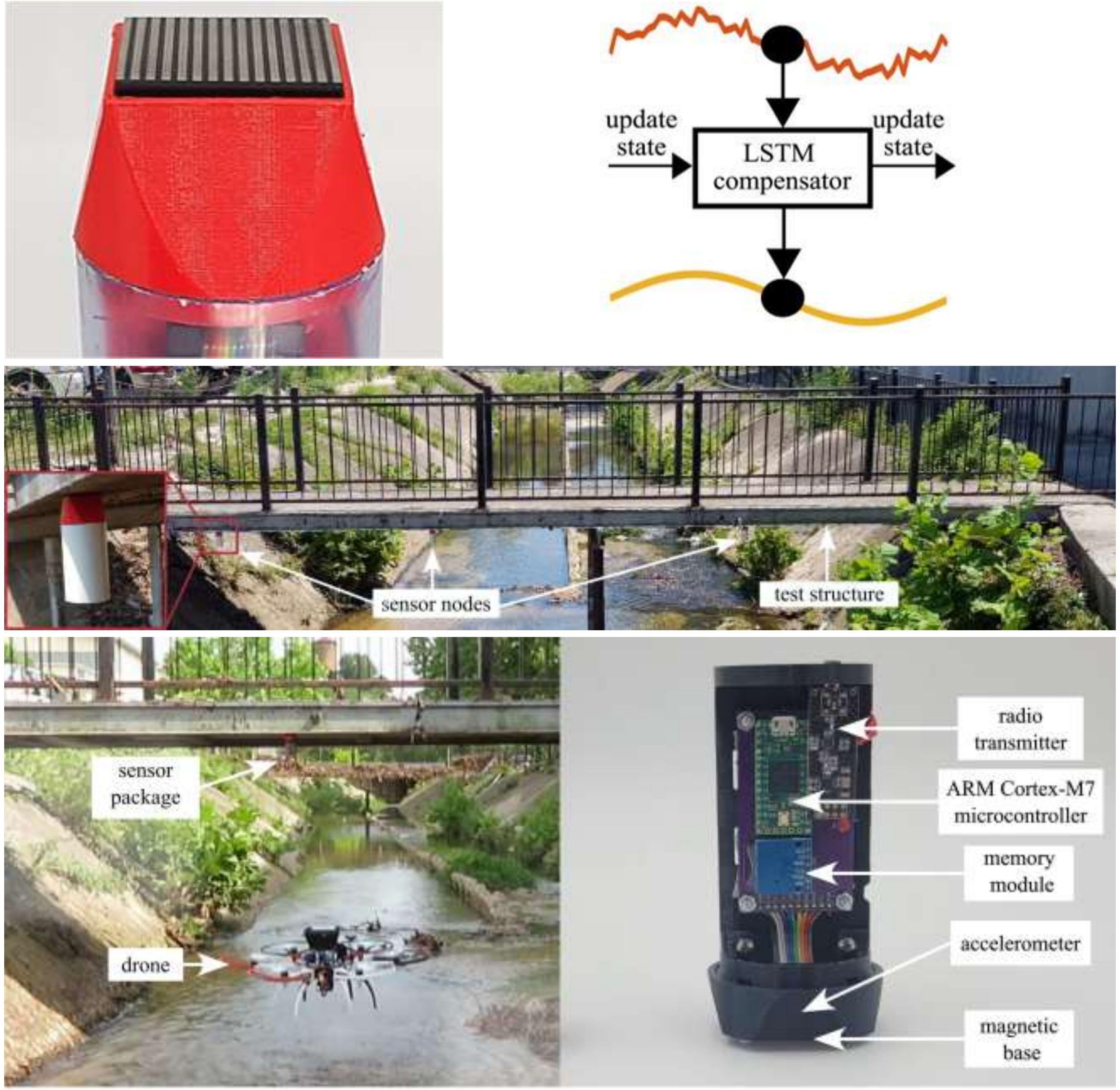


(b)

Note: Should more accuracy be desired, the DoF decomposition allows less ranks to be removed while still providing memory and computational savings over the uncompressed model.

Future Work

- **Edge Deployment:** We seek to apply this work to the first known deployment of a signal compensation model on an SHM sensing node.
- **Data Alignment:** Better techniques for aligning accelerometer signals for training will be explored.
- **Learning Rank Reductions:** Error-aware strategies for reducing ranks can provide better approximations for machine learning models than the truncated SVD.



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 (NSF) grant numbers CCF - 1956071, CMMI - 2152896, CCF-2234921 and, CPS - 2237696. Any opinions, findings, 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

Thank You for Your Time

Lab GitHub: github.com/arts-laboratory

Molinaroli College of Engineering and Computing
UNIVERSITY OF SOUTH CAROLINA