UNIVERSITY OF SOUTH CAROLINA

Multi-Event Model Updating for Ship Structures

NAVSEA NAVAL SEA SYSTEMS COMMAND

Introduction

Next-generation naval ships subjected to impact and fatigue events will benefit from condition assessment technology and the ability to react appropriately. When implemented properly a digital twin model of a naval ship or ship structure can be used for informed response management that will increase ship lifespans, maintenance intervals, and survivability. These models include fatigue and load, which will be used to make decisions across the structure's timescales (Real-Time to life span). The main obstacle that occurs with multi-model data assimilation is the vast amount of data that needs to be updated into multiple models and linked to the structure's existing condition to calculate its remaining life. This data comes from a wide range of sources and locations, including strain measurements from physical sensors that are attached to a ship's structure and 3D scans from aerial drones. This research presents a methodology that updates multiple damage cases (fatigue and plastic deformation impact) into a single FEA model. This work uses a scaled model of a structural ship component subjected to representative wave loadings.

Methodology

Thrust 1: Develop a 1-D test structure.
This thrust will develop a 1-D structural testbed ("beam model") of a ship to produce simplified data sets that will be used for validating multi-model data assimilation algorithms.

Thrust 2: Develop tools for multi-model data assimilation.
Thrust 2 will investigate the assimilation of data into multiple models with the goal of developing a well-informed digital twin of a naval structural system.

Thrust 3: Investigate multi-timescale decision-making.
This thrust will investigate the use of the updated models for decision-making on varying timescales from real-time (impact) responses to life-cycle decision-making (fatigue).

References

A. Downey, C. Hu, S. Laflamme, Optimal sensor placement within a hybrid dense sensor network using an adaptive genetic algorithm with learning gene pool, Structural Health Monitoring (2017) 147592171770253doi:10.1177/1475921717702537.

A. Downey, J. Hong, B. Joyce, J. Dodson, C. Hu, S. Laflamme, Methodology for real-time state estimation at unobserved locations for structures experiencing high-rate dynamics, in: Structural Health Monitoring 2017, Destech Publications, 2019

J. Yan, X. Du, A. Downey, A. Cancelli, S. Laflamme, L. Leifsson, A. Chen, F. Ubertini, Surrogate model for condition assessment of structures using a dense sensor network, in: H. Sohn (Ed.), Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2018, Vol. 10598, SPIE, 2018, pp. 10598–9. doi:10.1117/12.2296711.

Thrust 1 Results

Research outcomes for Thrust 1:

Thrust 1 consisted of designing a test structure to generate distribution A data. The test structure is shown below in Figure 1 where figure 2a) is the original plan and shows how the test structure could be deployed in a water-tunnel like setting to collect data. Figure 2b) shows how the test structure was updated to allow for this testing at the UofSC. However, due to lab closures, the test structure was modeled in finite element analysis software (FEA).

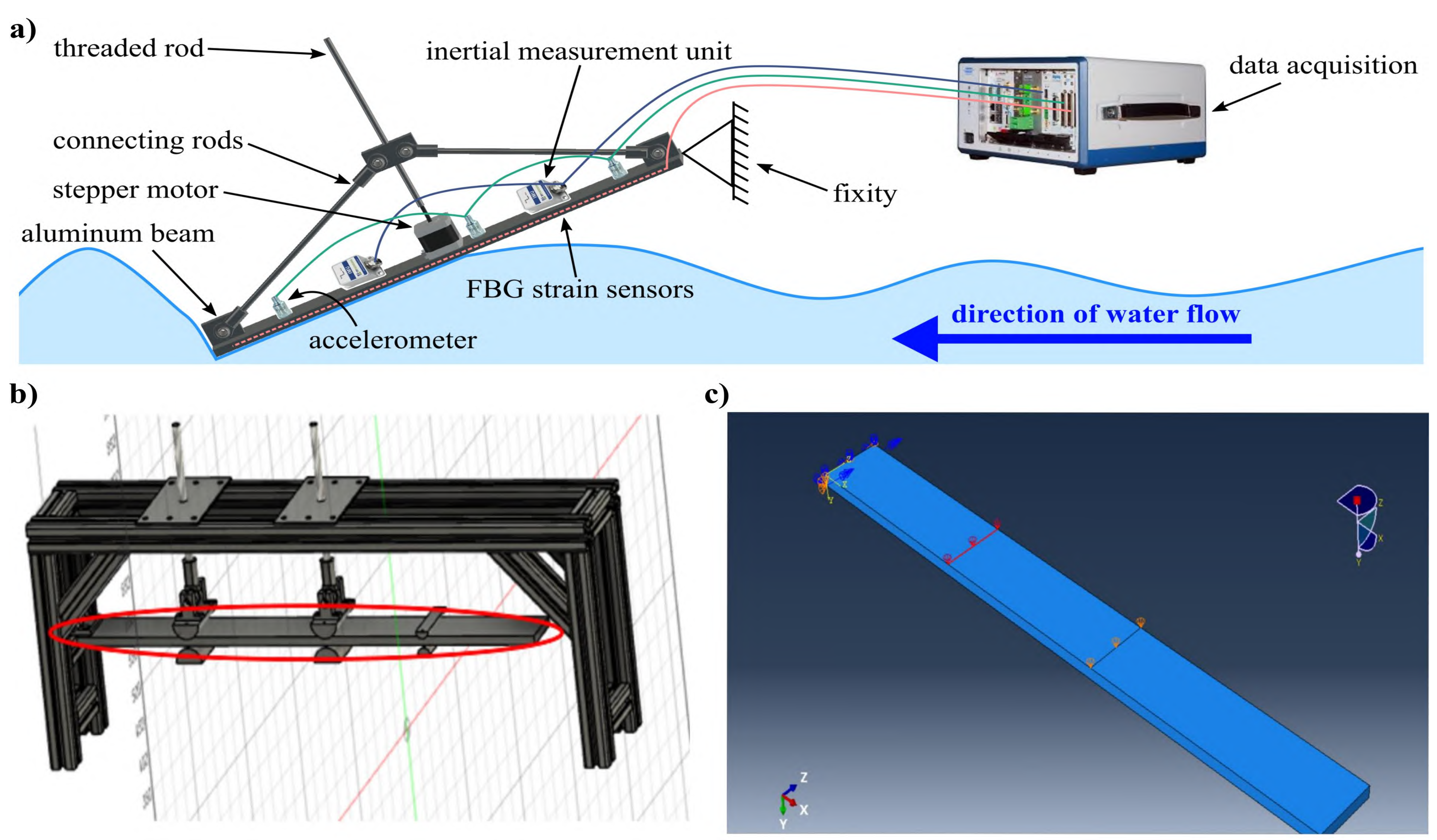


Figure 1: a) Original design with attached DAQ; b) Updated design capable of testing at UofSC, and; c) FEA model of beam.

FEA modeling for test structure

A key development was the creation of a Python code capable of building, executing, and processing FEA models. For this work, the commercial FEA code Abaqus was used as the solver. Various beam configurations were considered, including simply supported, cantilever, single and double roller supports, and beams with and without webbing. Finally, a cantilever beam with single roller support on the right-hand-side was selected, as shown in figure 2a). A fatigue crack was modeled in the beam near the left fixity. This crack was modeled as a hole in the beam, as shown below in figure 2b). The FEA model was developed, executed, and post-processed using the following processes.

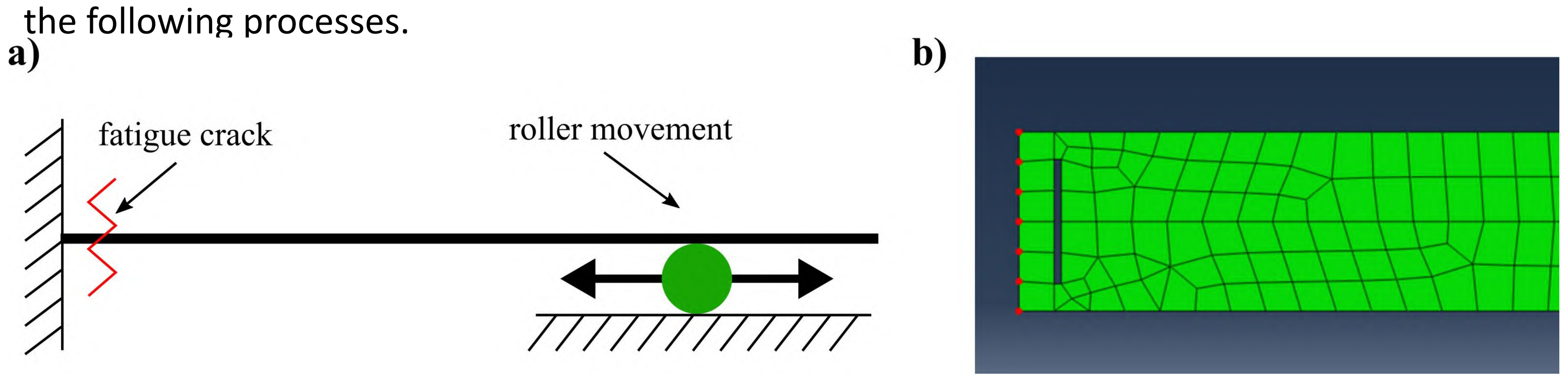


Figure 2: a) 2D representation of the final beam; b) fatigue crack modeled as a hole.

Abaqus CAE was used to create a shell model of the beam. Using both the macro manager and replay files (.rpy), the Python scripting commands for an Abaqus model were created.

1. 2. An Abaqus input file was created from the Python scripts.
2. 3. Python was used to execute the Python scripts in Abaqus via a command-line prompt.
3. 4. Python was used to post-process the data by extracting all required data from the .odb file

Thrust 2 Results

Research outcomes for Thrust 2

Thrust 2 developed a model updating framework that could update a structure for two damage cases that happen on two different time scales (fatigue damage on a long timescale and an impact on a short time scale). A variety of tests with varying fatigue crack and impact tests were conducted, a fatigue crack that grows from 2 to 20 mm and "impact" that results in roller displacement of 2 mm. Results for the change in frequency (ω) and mode shapes (ϕ) for these values are presented below in Figure 3. Note that the crack growth and impact damage both result in changes to the frequency and mode shapes.

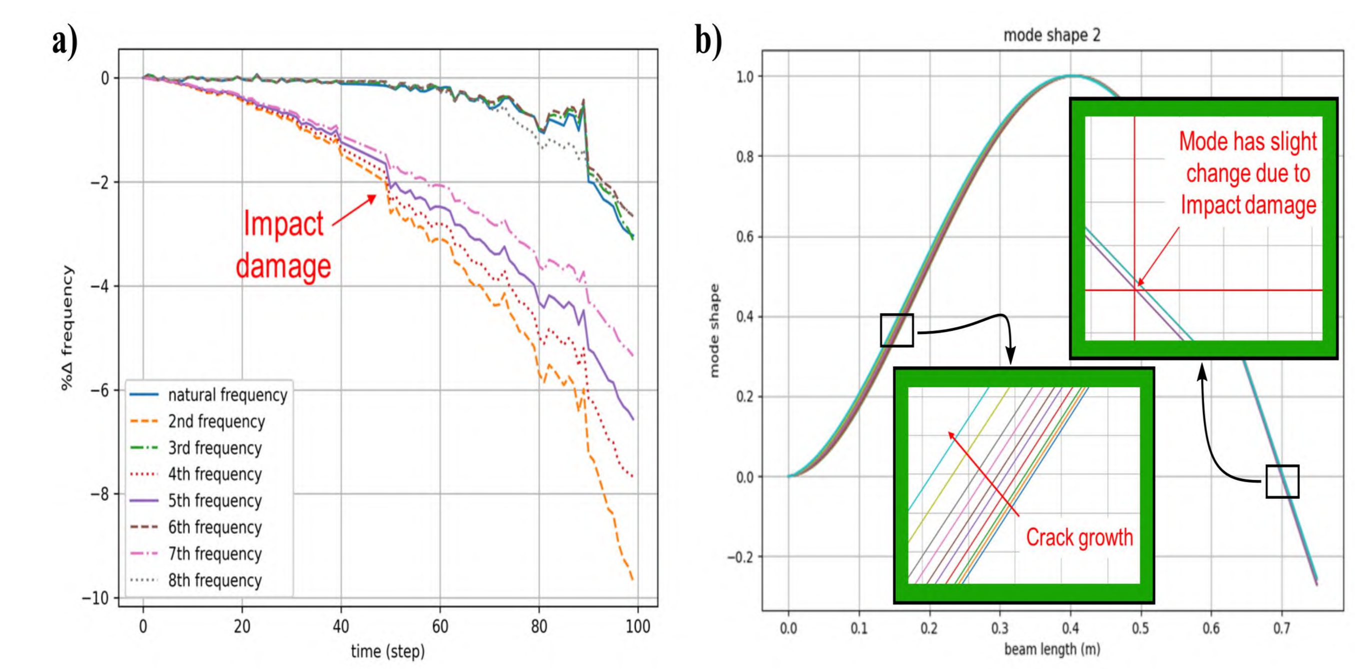
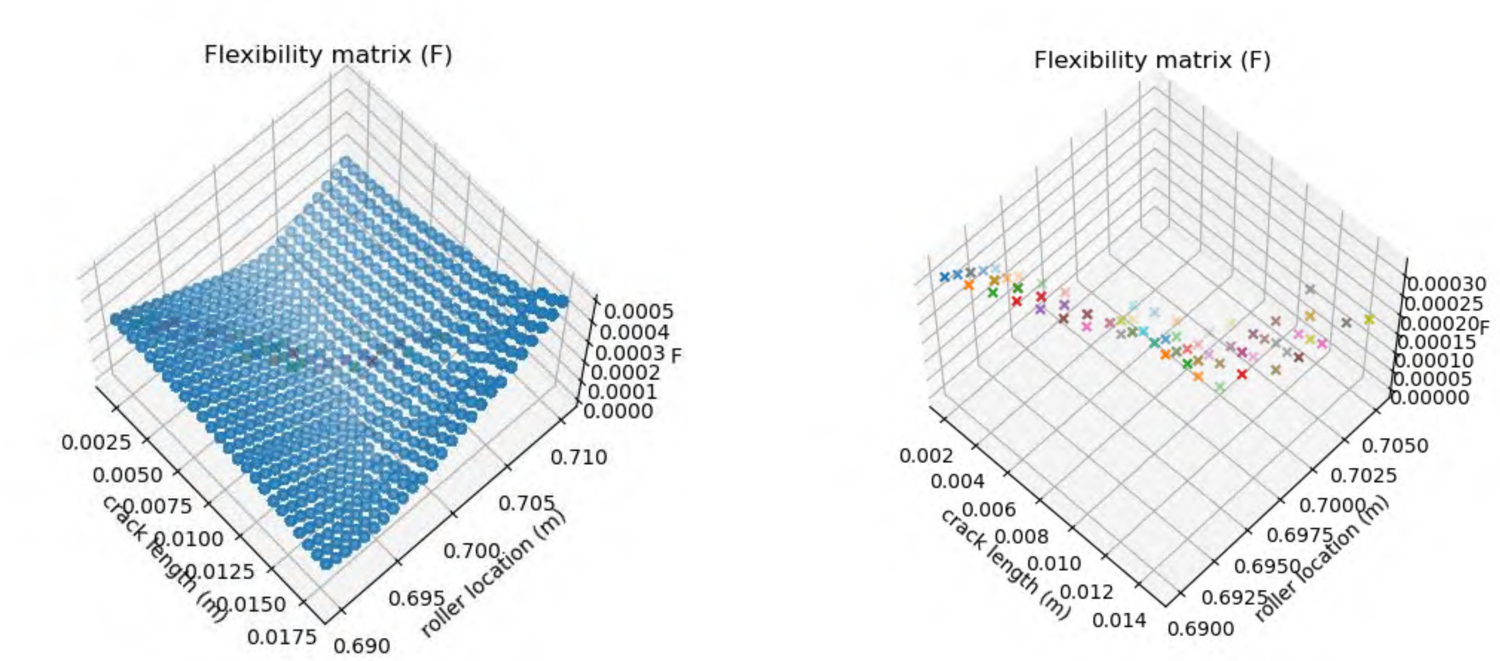


Figure 3: The growth of the fatigue and impact damage

Continuing, this thrust also implemented a particle swarm that runs for a finite number of steps. When this is run on any set of data, each particle travels to the next data point and determines if it is lower than the previous. This is looped until the number of input steps is reached. Once reached, the lowest particle coordinate is saved as the optimal location for the stepper motor boundary conditions for the FEA model. This method was used on the Flexibility Matrix and is shown below.



Future Improvements

The continuation of this project is detailed below:

1. Finalize and test FEA Model Updating Script (Thrust 3)
2. Integrate the solved optimal FEA boundary conditions to the physical test stand using LabView
3. Integrate various sensors onto the physical test beam
4. Integrate limit switches as an emergency stop

Funding

This work is partially supported by the National Science Foundation, Grant number 1937535. This work is also partially funded by the University Of South Carolina through the Magellan Scholar Award, award number 1155400-20-52512. The support of these agencies is gratefully acknowledged. 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 University of South Carolina.

NSF

Acknowledgements

Thank you, Dr. Downey, for sponsoring the project, his continuous engagement, mentorship, and for providing us with this valuable opportunity. Also a special thank you to the Naval Surface Warfare Center, Carderock Division for the opportunity to work on such a challenging and rewarding project.