Damage Detection of Miter Gates Through Nonlinear Model Updating

TRENT E. SCHREIBER and YANG WANG

ABSTRACT

Miter gates are a critical component of many navigation locks in the inland waterway infrastructure. Due to deterioration and impacts from debris or vessels, gaps can develop between the bearing surfaces supporting the miter gates. This loss of bearing contact creates a redistribution of stress that is typically not accounted for in design, which can lead to accelerated deterioration and possibly unexpected failure of the gate. We present a novel approach to detecting the loss of bearing contact by finite element (FE) model updating. The contact interface is modeled using nonlinear springs with unknown parameters. These parameters are estimated by minimizing the difference between in-situ measured strain and the strain predicted by the FE model. This is naturally formulated as an optimization problem, where the unknown spring parameters are the optimization variables. The ensuing optimization problem is nonconvex, therefore, multiple local searches are performed to increase the chance of finding the global optimum. Two types of springs are considered for modeling the nonlinearity of the boundary conditions, namely, a gap element and a cubic spring. This paper presents both the formulation and validation of the proposed approach.

INTRODUCTION

Navigation locks facilitate the passage of cargo along waterways that are impeded by dams. The lock chamber is comprised of two concrete walls and two lock gates. The lock gates are steel framed structures that open and close. When the lock gates are closed, the chamber can be filled or emptied to match the upstream or downstream water level. Miter gates, which are named after the geometry of the gate when closed, are the most common type of lock gate. For miter gates, gaps can develop between the bearing surfaces that support the structure. This loss of bearing contact can have severe implications on the structural behavior. Traditional maintenance activities involve sending divers or closing and dewatering the lock to allow for visual inspection. The high costs and downtime associated with these techniques lead to infrequent

School of Civil and Environmental Engineering, Georgia Institute of Technology, 790 Atlantic Dr. NW Atlanta, GA 30332, USA

inspections, typically once every 5-10 years. Closure of a navigation lock, as a result of repairs or maintenance, presents substantial economic losses due to the volume of freight that travels along waterways. Proper maintenance and early detection of damage are crucial in avoiding prolonged outages of locks.

Some researchers have studied statistical methods for autonomous damage detection [1, 2]. The use of machine learning algorithms is also a popular strategy for structural health monitoring [3, 4]. Other researchers have explored Bayesian model updating to estimate unknown model parameters related to the boundary conditions of miter gates [5-7]. Most studies have utilized strain data, but some have applied computer vision to perform model updating with vision-based displacement measurements [8].

This paper proposes a novel approach to detecting damage of the bearing surfaces through structural model updating using static strain measurements. The bearing contact is modeled with nonlinear springs and the spring parameters are estimated (or updated) through a nonconvex optimization problem that minimizes the difference between experimental and analytical strain. The estimated model parameters can then be used as damage indicators and the updated model can be used to evaluate the severity of possible damage. The proposed approach is validated through numerical simulations of the downstream lock gate at The Dalles Lock and Dam on the Columbia River.

FORMULATION

In the proposed approach, model parameters related to gaps between bearing surfaces are adjusted to minimize the difference between experimentally measured strains and the strains predicted by the finite element (FE) model. Consider a structure with $n_{\boldsymbol{\beta}}$ boundary conditions that have unknown gaps between bearing surfaces. To help illustrate this gap, consider Figure 1, where the one-dimensional element under load P must elongate a distance Δ before Surface A makes contact with Surface B. This boundary condition provides restraint when the two surfaces (A and B) are in contact and pressing against each other (i.e., bearing).

Each unknown boundary condition is represented by a nonlinear spring that is parameterized by β_j, which is the j-th entry of the updating vector variable $\boldsymbol{\beta} \in \mathbb{R}^{n_{\boldsymbol{\beta}}}$. Two types of nonlinear springs are considered, namely, a gap element and a cubic spring. Shown in Figure 2(a), the spring force, f_{s}, is a function of both the spring deformation, u_j, and the parameter β_j. Note that the deformation of the spring, u_j, is positive in the compressive direction. For the gap element in Figure 2(b), the value k_{p} is the penalty spring stiffness once the gap is closed. In this work, k_{p} is fixed and not updated (i.e., only the β_j parameters are estimated). Shown in the figure and in the gap element function in Table I, the parameter β_j explicitly represents the gap depth. For the cubic spring in Figure 2(c), the effect of the gap is only approximated based on the value of β_j, but the function, also shown in Table I, is smooth unlike the gap element.

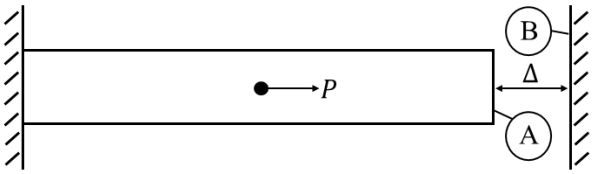
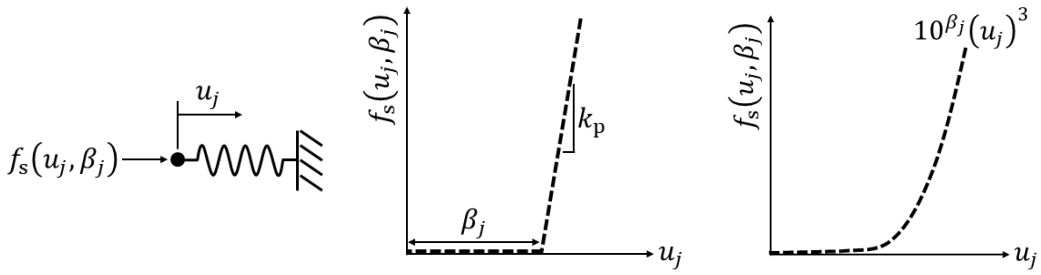


Figure 1. Illustration of a gap between bearing surfaces.



(a) Nonlinear spring (b) Gap element (c) Cubic spring

Figure 2. Nonlinear spring and force-displacement diagrams.

TABLE I. NONLINEAR SPRING FUNCTIONS

Spring type	Spring force
Gap element	$f_s(u_j, \beta_j) = \begin{cases} k_p(u_j - \beta_j) & \text{if } u_j \geq \beta_j \\ 0 & \text{else} \end{cases}$
Cubic spring	$f_s(u_j, \beta_j) = \begin{cases} 10^{\beta_j} u_j^3 & \text{if } u_j \geq 0 \\ 0 & \text{else} \end{cases}$

For a structure with N-degrees of freedom (DOFs), the nonlinear spring forces are arranged in the nonlinear force vector, $\mathbf{f}_{\text{nl}} \in \mathbb{R}^N$, corresponding to the appropriate DOFs. Since the nonlinear springs have been grouped in the nonlinear force vector, the stiffness matrix, $\mathbf{K} \in \mathbb{R}^{N \times N}$, contains only the linear elements, which are assumed to be accurate. The applied loads must be known a priori and assembled in the global load vector, $\mathbf{f} \in \mathbb{R}^N$. The global displacement vector, $\mathbf{d} \in \mathbb{R}^N$, must satisfy the nonlinear, static equilibrium equation,

$$\mathbf{f} + \mathbf{f}_{\text{nl}}(\mathbf{d}, \boldsymbol{\beta}) = \mathbf{K}\mathbf{d}, \tag{1}$$

where the nonlinear force vector, $\mathbf{f}_{\text{nl}} \in \mathbb{R}^N$, is a function of both the displacements and spring parameters.

Consider that experimental strain measurements were taken at n_s locations and are collected in the vector $\boldsymbol{\varepsilon}^{\text{EXP}} \in \mathbb{R}^{n_s}$. To utilize strain measurements, we need to be able to relate the nodal displacements of the FE model to strain within an element. This is accomplished with the element strain-displacement matrix found in most textbooks on the finite element method, such as [9] and [10]. The strain-displacement mapping is achieved by vertically stacking the mapping vectors for each element corresponding to a measurement location and horizontally aligning system DOFs to construct the system strain-displacement matrix, $\mathbf{B} \in \mathbb{R}^{n_s \times N}$. Consequently, the global displacement vector, $\mathbf{d} \in \mathbb{R}^N$, is mapped to the analytical strain vector,

$$\boldsymbol{\varepsilon} = \mathbf{B}\mathbf{d}, \tag{2}$$

where each entry of $\boldsymbol{\varepsilon} \in \mathbb{R}^{n_s}$ is the elemental strain that corresponds to the experimentally measured value in $\boldsymbol{\varepsilon}^{\text{EXP}} \in \mathbb{R}^{n_s}$.

Since the displacements are obtained from the nonlinear, static equilibrium equation in Eq. (1) and implicitly depend on the spring parameters, we can write the displacement vector as a function of $\boldsymbol{\beta} \in \mathbb{R}^{n_\beta}$, i.e., $\mathbf{d}(\boldsymbol{\beta}) \in \mathbb{R}^N$. The spring parameters are estimated by minimizing the difference between the experimental and analytical strain vectors in a least-squares sense,

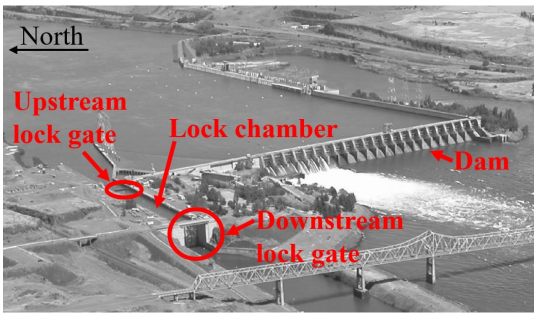
$$\begin{aligned} \underset{\boldsymbol{\beta}}{\text{minimize}} \quad & \|\mathbf{W}\{\boldsymbol{\varepsilon}^{\text{EXP}} - \mathbf{B}\mathbf{d}(\boldsymbol{\beta})\}\|_2^2 \\ \text{subject to} \quad & \mathbf{L}_{\boldsymbol{\beta}} \leq \boldsymbol{\beta} \leq \mathbf{U}_{\boldsymbol{\beta}}, \end{aligned} \tag{3}$$

where $\mathbf{W} \in \mathbb{R}^{n_s \times n_s}$ is a diagonal weighting matrix and $\mathbf{L}_{\boldsymbol{\beta}} \in \mathbb{R}^{n_{\boldsymbol{\beta}}}$ and $\mathbf{U}_{\boldsymbol{\beta}} \in \mathbb{R}^{n_{\boldsymbol{\beta}}}$ are the lower and upper bounds on $\boldsymbol{\beta} \in \mathbb{R}^{n_{\boldsymbol{\beta}}}$, respectively. As a result of the implicit function $\mathbf{d}(\boldsymbol{\beta}) \in \mathbb{R}^N$ that originated from Eq. (1), the optimization problem in Eq. (3) is nonconvex.

THE DALLES MITER GATE

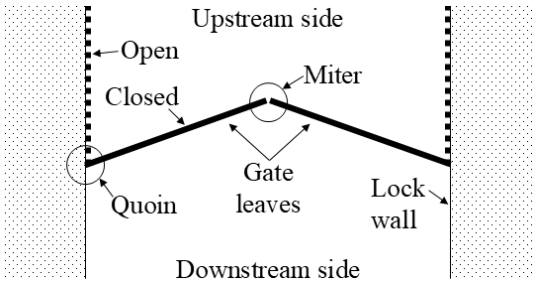
Overview of The Dalles Miter Gate

An overview of The Dalles Lock and Dam is shown in Figure 3(a). The downstream lock gate, which is a miter gate, is more clearly shown in Figure 3(b). A simple schematic of a miter gate is shown in Figure 3(c) to highlight some of the main components. A miter gate consists of two leaves that swing open and close. In Figure 3(c), the dashed, thick black lines show the gate in the open position while the solid, thick black lines show the gate closed. The gate-to-wall interface is referred to as the quoin and allows rotation about a vertical axis so that the gate can be opened and closed. The region where the two leaves meet is referred to as the miter. When closed, the gate holds back water on the upstream side. This difference in water level, shown in Figure 3(d), loads the gate. Under hydrostatic load, the two leaves of the gate bear on each other (at the miter) and the lock walls (at the quoin) continuously along the gate height.

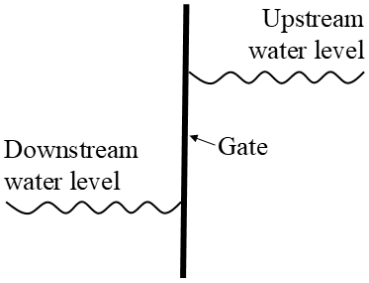


(a) Lock and dam


(b) Downstream miter gate (closed)



(c) Plan view of miter gate



(d) Side view of miter gate

Figure 3. The Dalles Lock and Dam.

Since the leaves of the gate are mirror images of each other, only one leaf needs to be modeled. A 3D finite element model of the south leaf was created in Abaqus and is shown in Figure 4. The model has approximately 940,000 elements, consisting primarily of reduced integration shell elements. Each leaf of the gate rests on a ball and socket joint at the base, known as the pintle, which provides horizontal and vertical restraint. There is also a horizontal restraint at the top of the leaf, referred to as the gudgeon. When loaded with hydrostatic pressure, the gate is supported by the pintle, gudgeon, and bearing surfaces along the miter and quoin. To validate the proposed approach of damage detection, we consider a loss of bearing contact in a section of the quoin near the bottom of the gate. Conventional boundary conditions are applied to the rest of the quoin, which restrain the quoin from translating horizontally in the XZ plane. Contact with the other leaf is modeled by restraining the miter from translating perpendicular to its contact surface, while allowing movement parallel to the surface.

To reduce the number of nonlinear elements, multi-point constraints are used to constrain nodes along a segment of the quoin down to a so-called virtual node, v_j. The nonlinear spring is then attached between this virtual node and a so-called support node, s_j, that is free to translate parallel to the lock chamber wall (in the YZ plane) but is restrained perpendicular to the wall (along the X axis). As the quoin pushes into the lock chamber wall, the nonlinear spring is compressed and the resulting spring force is applied perpendicular to the wall (in the X direction). Shown on the right side of Figure 4, ten pairs of virtual-support nodes, each consisting of a 16-inch segment of the quoin, are used in this numerical study. With one parameter being updated for each nonlinear spring, the model updating problem attempts to estimate the parameters of these ten springs ($n_{\boldsymbol{\beta}} = 10$).

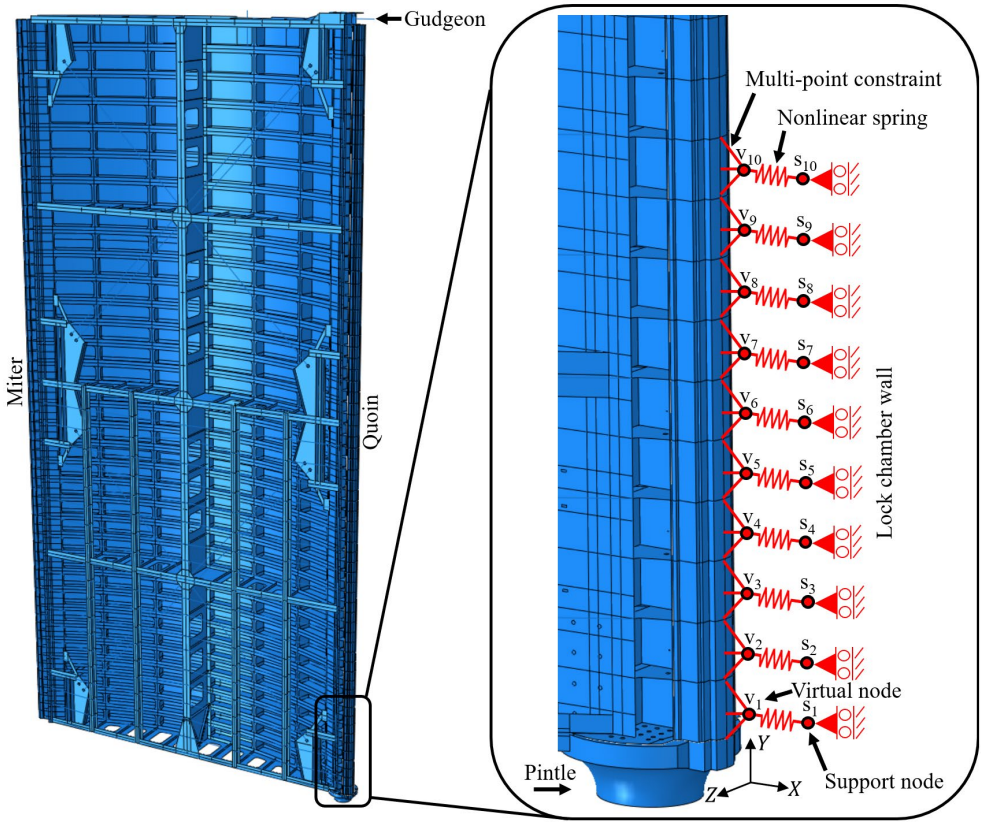
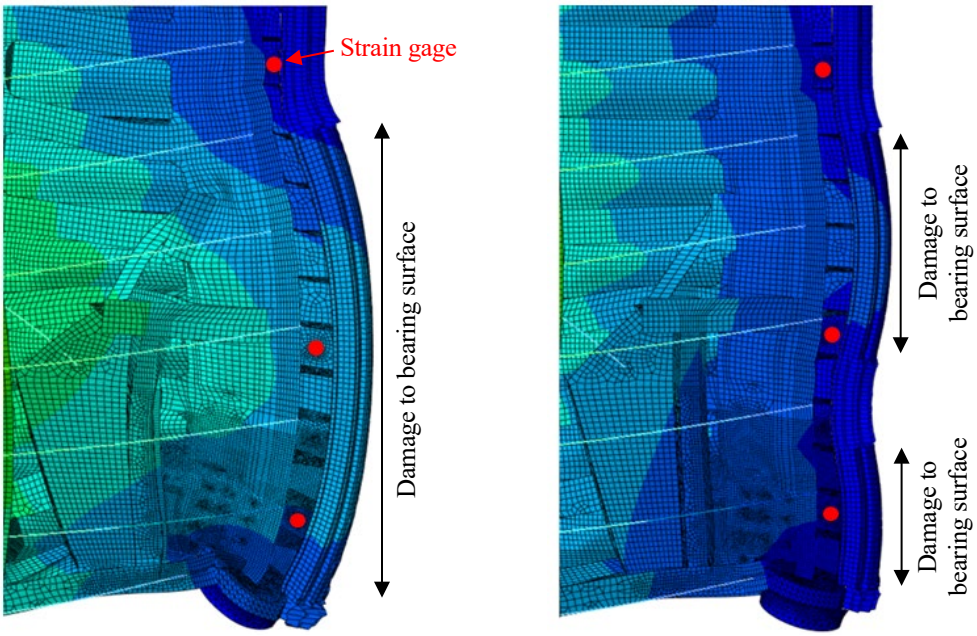


Figure 4. Finite element model of The Dalles miter gate (south leaf).

To simplify the problem, the virtual nodes ($v_1, \ldots, v_{10}$) are constrained to move only in the XY plane (i.e., a vertical plane perpendicular to the lock chamber wall). In reality, the quoin can also move in the Z direction until making contact with the wall, in which case friction restricts any further movement. However, this makes the problem significantly more difficult and is outside the scope of this paper. Furthermore, it is common practice to allow the quoin to freely move vertically (in the Y direction) despite friction forces since the pintle provides a relatively stiff vertical constraint. In this approximate approach, the nonlinearity is a function of the virtual node translation in only the X direction (perpendicular to the wall). Therefore, using static condensation [11], the original model which has millions of DOFs is reduced to a model with just ten DOFs (one X direction DOF for each virtual node). After using the retained DOFs of the virtual nodes to solve the nonlinear equilibrium equation, the nodal displacements of the elements defining the strain gages are calculated to determine the analytical strain.

In this study, two simulations were performed with different boundary conditions applied to the virtual nodes, mimicking possible damage to the bearing surface along the lock chamber wall. Gravity and hydrostatic loads were applied to the gate. Abaqus's connector element was used to model the nonlinearity of the boundary conditions. For each simulation, the deformed shape near the pintle is shown in Figure 5. Note that this is the same section of the quoin that is shown in the blown-up window of Figure 4. The simulation in Figure 5(a) represents a situation where a large, 160-inch section of the quoin has lost bearing contact with the lock chamber wall. Whereas in Figure 5(b), some sections of the quoin contact the wall while other, smaller sections are free. In each simulation, "experimental" strain was extracted from Abaqus at three locations ($n_s = 3$) near the base of the quoin, which are marked with red circles in the figure.



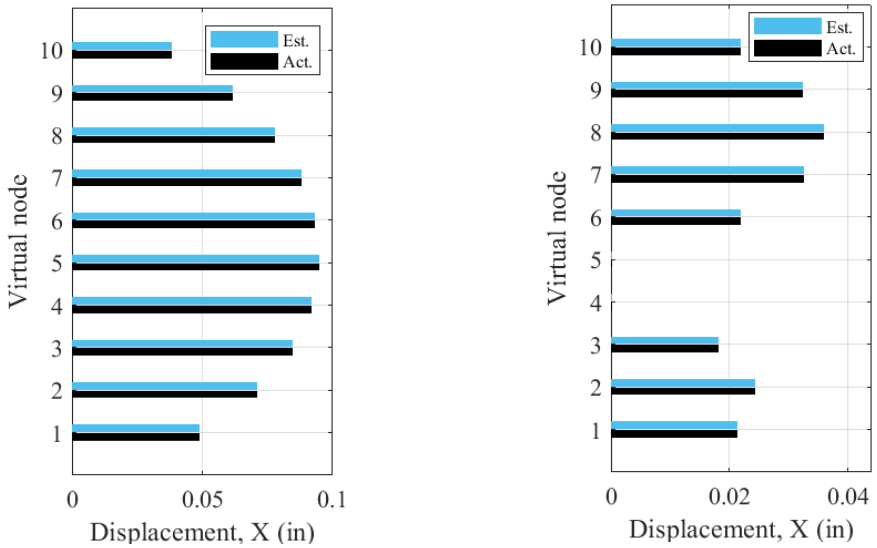
(a) Damage simulation 1 (b) Damage simulation 2

Figure 5. Deformed shape near pintle for numerical simulations.

Model Updating Results

For each simulation in Figure 5, the optimization problem in Eq. (3) was solved using the gap element first and then the cubic spring. With inches as the unit for displacement, the bounds on the optimization variables were set to $\beta_j \in [0, 0.2]$ for the gap element. When using the cubic spring, the bounds were set as $\beta_j \in [-30, 30]$. For the gap element, the (constant) penalty spring stiffness once the gap is closed was set to $k_{\mathrm{p}} = 10^6 \cdot K_{\max}$, where $K_{\max}$ is the maximum, absolute value in the stiffness matrix. For normalization, each strain measurement is denoted as $\varepsilon_i^{\mathrm{EXP}}$. The corresponding weighting, i.e., the i-th diagonal entry of matrix $\mathbf{W}$, was set to $1/\varepsilon_i^{\mathrm{EXP}}$. The gradient of the objective function was estimated numerically using the finite difference method. All runs were performed on a machine with an Intel Core i9-12950HX CPU and 64 GB of RAM. The Levenberg-Marquardt algorithm was employed through MATLAB's `lsqnonlin` toolbox.

The optimization process was initiated from 100 random starting points within the bounds of $\boldsymbol{\beta}$. Among the 100 searches, the best solution is chosen as the one corresponding to the minimum objective function value. Using the best solution, the displacements of the virtual nodes are estimated by solving the nonlinear equilibrium equation in Eq. (1). For the cubic spring, these estimated displacements are shown in Figure 6 as the blue bars and are nearly identical to the actual displacements extracted from Abaqus, which are shown as the black bars in the figure. Since the results of the gap element are nearly the same as the cubic spring, only the displacements with the cubic spring are plotted. This demonstrates that the proposed approach can accurately identify gaps between the bearing surfaces that support miter gates. If no gap is present, then the virtual node should have no displacement, like virtual nodes 4 and 5 in Figure 6(b). Any displacement of a virtual node is indication of a gap at that location. When using the gap element, the total computation time for the 100 searches was 132 seconds for the first simulation and 13 seconds for the second simulation. When using the cubic spring, the computation times were 46 seconds and 50 seconds for the first and second simulation, respectively.



(a) Damage simulation 1 (b) Damage simulation 2

Figure 6. Virtual node displacements.

CONCLUSION

This paper presents a novel method of detecting damage to the bearing surfaces that support miter gates. The approach models the unknown boundary conditions as nonlinear springs and then the spring parameters are estimated using static strain measurements. The estimated parameters are used to update the structural model, which can be used to evaluate the extent of possible damage. The proposed approach was able to accurately detect damage in two numerical simulations of The Dalles miter gate.

ACKNOWLEDGEMENTS

The authors would like to thank Brian Eick and Travis Fillmore at the U.S. Army Corps of Engineers - Engineer Research and Development Center for their help with the finite element model of The Dalles miter gate. This research was partially funded by the National Science Foundation (CMMI-2211343). The first author received support from the U.S. Department of Defense SMART Scholarship-for-Service Program. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the sponsors.

REFERENCES

1. Eick, B.A., Z.R. Treece, B.F. Spencer, M.D. Smith, S.C. Sweeney, Q.G. Alexander, and S.D. Foltz. 2018. "Automated damage detection in miter gates of navigation locks," *Structural Control and Health Monitoring*, 25(1):e2053.
2. Smith, M.D., Z.R. Treece, S.L. Bunkley, and V.P. Chiarito. 2015. "Development of the USACE automated SMART gate system for lock gates: Detection of barge impact events using statistical process control," *Proceedings of the 10th International Workshop on Structural Health Monitoring (IWSHM)*, September 1-3, 2015.
3. Vega, M., R. Madarshahian, and M.D. Todd. 2019. "A neural network surrogate model for structural health monitoring of miter gates in navigation locks," *Proceedings of the 37th IMAC, A Conference and Exposition on Structural Dynamics*, January 28-31, 2019.
4. Hoskere, V., B.A. Eick, B.F. Spencer, M.D. Smith, and S.D. Foltz. 2020. "Deep Bayesian neural networks for damage quantification in miter gates of navigation locks," *Structural Health Monitoring*, 19(5):1391-1420.
5. Gomez, F., B.F. Spencer, and M.D. Smith. 2019. "Bayesian modeling updating of miter gates with uncertain boundary conditions," *Proceedings of the 12th International Workshop on Structural Health Monitoring (IWSHM)*, September 10-12, 2019.
6. Yang, Y., R. Madarshahian, and M.D. Todd. 2019. "Bayesian damage identification using strain data from lock gates," *Proceedings of the 37th IMAC, A Conference and Exposition on Structural Dynamics*, January 28-31, 2019.
7. Vega, M.A., M.K. Ramancha, J.P. Conte, and M.D. Todd. 2020. "Efficient Bayesian inference of miter gates using high-fidelity models," *Proceedings of the 38th IMAC, A Conference and Exposition on Structural Dynamics*, February 10-13, 2020.
8. Wang, S., C. Rodgers, T. Fillmore, B. Welsh, T. Golecki, S.A.V. Shajihan, B.A. Eick, and B.F. Spencer. 2023. "Vision-based model updating and evaluation of miter gates on inland waterways," *Engineering Structures*, 280:115674.
9. Chandrupatla, T.R. and A.D. Belegundu. 2012. *Introduction to Finite Elements in Engineering*. Pearson, NJ.
10. Zienkiewicz, O.C. and R.L. Taylor. 2000. *The Finite Element Method*. Butterworth-Heinemann, Oxford, UK.
11. Kassimali, A. 2012. *Matrix Analysis of Structures*. Cengage Learning, Stamford, CT.