

# OpenIPDM: A Probabilistic Framework for Estimating the Deterioration and Effect of Interventions on Bridges

ZACHARY HAMIDA\*, BLANCHE LAURENT and JAMES-A. GOULET

Department of Civil, Geologic and Mining Engineering

POLYTECHNIQUE MONTREAL, CANADA

March 31, 2022

## Abstract

This paper describes the OpenIPDM software for modelling the deterioration process of infrastructures using network-scale visual inspection data. In addition to the deterioration state estimates, OpenIPDM provides functions for quantifying the effect of interventions, estimating the service life of an intervention, and generating synthetic data for verification purposes. Each of the aforementioned functions are accessible through an interactive graphical user interface. OpenIPDM is designed based on the research work done on a network of bridges in Quebec, Canada, so that the concepts presented in the software have been validated for applications in a real-world context. In addition, this software provides foundations for future developments in the subject area of modelling the deterioration as well as intervention planning.

**Keywords:** Bridge Deterioration Software, Visual Inspections, Inspector Uncertainty, Structural Health Monitoring.

## 1 Motivation and significance

Monitoring the deterioration state of bridges is a prerequisite for planning maintenance activities and many other decision-making processes [1, 2, 3, 4, 5]. Visual inspection is a common techniques for performing health monitoring tasks and providing data on a network-scale [6, 7, 8]. Nonetheless, interpreting information from visual inspection data is known to be challenging [9, 10, 11]. The challenges arise from technical aspects such as, the subjective nature of the evaluation method [12, 13, 11]; and non-technical aspects such as, navigating and performing analyses on data from thousands of structural elements. Moreover, missing data and outliers can impose inconsistencies or cause instability in any type of analyses performed using the inspection data [14].

In order to address the aforementioned challenges, the software OpenIPDM has been developed to provide access to state-of-the-art deterioration models, all within an interactive graphical user interface (GUI). The GUI facilitates the navigation through the inspection data as well as performing analyses on structural elements, bridges and network of bridges. OpenIPDM also has the capacity to generate synthetic data, perform verification and validation analyses, and automatically manage missing data and outliers. The deterioration model employed in OpenIPDM is based on the SSM-KR framework, which is a hybrid deterioration model that combines state-space models (SSM) with kernel regression (KR) [15]. The SSM-KR enables modelling the deterioration condition and speed using visual inspection data, while taking into account the inspectors uncertainties. In addition, the KR framework in SSM-KR allows exploiting the structural similarities among bridges, which contribute to improving the overall predictive capacity of the deterioration model.

In the context of modelling network-scale deterioration, there exists a handful of softwares with similar functionalities (e.g., Pontis), most of which are: 1) commercial, and 2) rely on frameworks with limited capacity to model the deterioration of infrastructures (e.g., incapacity to estimate and take into

---

\*Corresponding author: zachary.hamida@polymtl.ca

account the inspectors' uncertainty) [16, 17, 11]. OpenIPDM is completely open-source and encompass the most recent developments in the context modelling deterioration based on visual inspections. The software is part of ongoing research, and has already enabled several contributions in the field of structural health monitoring using visual inspections [11, 15, 18, 14].

Aside from performing deterioration analyses on inspection databases, OpenIPDM can serve as a benchmarking tool for developing and enhancing infrastructures deterioration models. In addition, OpenIPDM can facilitate the development and testing of decision-making frameworks, on a small-scale (element-level) or a network-scale. The software provides insights into the development, design and analyses performed on network-scale visual inspections, derived from real case studies.

## 2 Software description

This section provides a general description for the main architecture and functionalities available in the current version of OpenIPDM.

### 2.1 Software architecture

OpenIPDM is built and compiled using the App Designer in MATLAB, with a complete GUI, providing users an easy access to the different functionalities in the software. The workflows in OpenIPDM operate on two types of data: real data and synthetic data. The real data is a reference to the inspection data collected from real infrastructures, while the synthetic data refers to datasets generated to emulates the real data, with known parameters and access to the true deterioration states [11, 15, 18].

The main interface of OpenIPDM provides navigation features for accessing and obtaining the results of analyses on real data. Figure 1 shows the hierarchical access to information starting from the structure-level to the output, which correspond to the deterioration state estimates for a structural element selected by the user. From Figure 1, the information associated with each bridge  $\mathcal{B}_j$  are

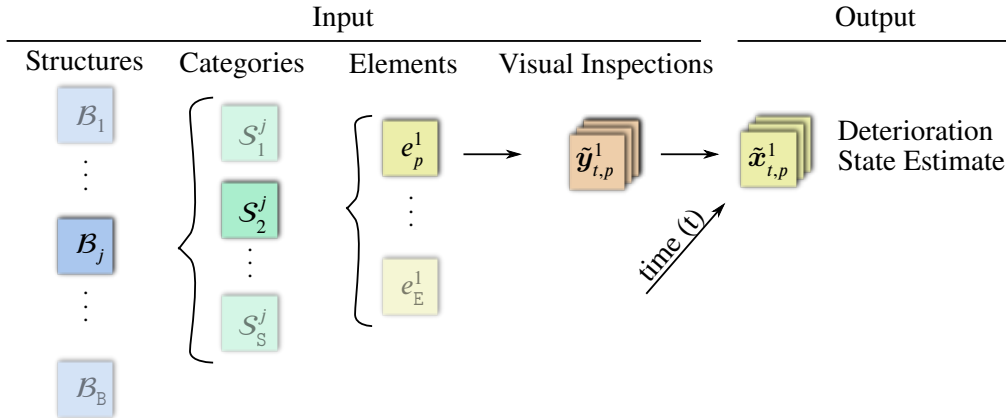


Figure 1: Example for the hierarchy of information, starting from bridge  $\mathcal{B}_j$ , structural category  $\mathcal{S}_2^j$ , structural element  $e_p^j$ , and ending with the deterioration state estimates  $\tilde{\mathbf{x}}_{t,p}^j$  over time  $t$ , based on the inspection data  $\tilde{\mathbf{y}}_{t,p}^j$ .

dynamically stored in a MATLAB array, which offers a better computational efficiency, relative to applying search queries on the entire database to access information. The computational efficiency can be further improved by pre-processing and storing the information of each bridge locally as “.mat” files. This is possible by accessing the function *Pre-process Data* from the *File* menu in OpenIPDM interface. In order to illustrate the input-output data in OpenIPDM, an example for a structural element  $e_1^{130}$  is shown in Figure 2. In this figure, the deterioration model provides the prediction for the deterioration condition of the structural element  $e_1^{130}$ , based on the inspection data  $\tilde{\mathbf{y}}_{t,1}^{130} \in [25, 100]$ . In this context,  $\tilde{y}_{t,1}^{130} = 100$ , refers to a perfect condition for the structural element, while  $\tilde{y}_{t,1}^{130} = 25$ , represent a poor condition for the structural element.

The main interface of OpenIPDM also provides access to additional toolboxes that enable essential

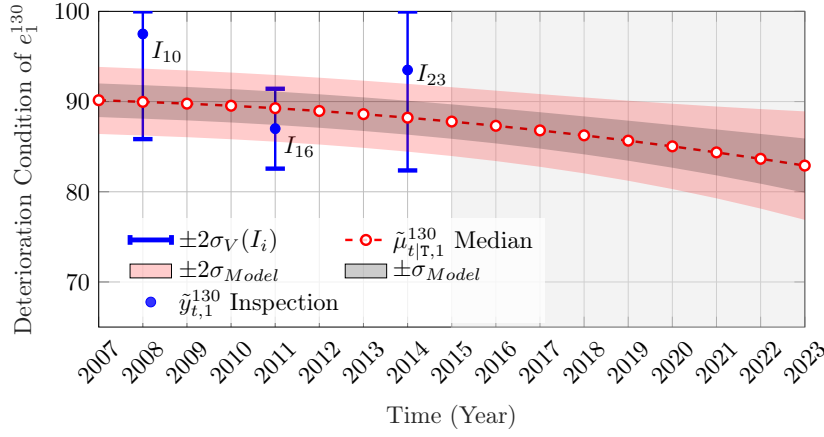


Figure 2: Condition deterioration analysis based on observations  $\tilde{\mathbf{y}}_{t,1}^{130} \in [25, 100]$  of the structural element  $e_1^{130}$  with error bars representing the inspectors estimated uncertainties, and the shaded area representing the prediction period.

functionalities in the software. Figure 3 shows a summary for the architecture of the software.

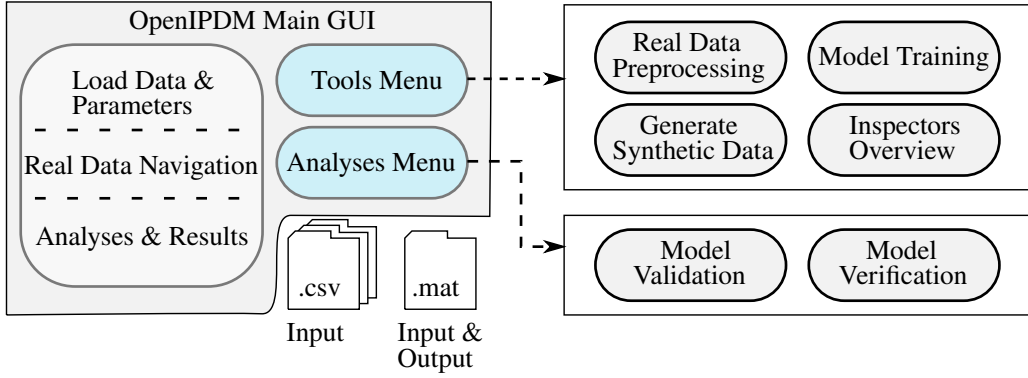


Figure 3: OpenIPDM main architecture and functionalities along with the acceptable input-output file formats.

In Figure 3, the software OpenIPDM takes as input two types of files format, which are, “.csv” and “.mat”. The “.csv” format mainly represent the input files that contain the real database information (e.g., inspections data); while the “.mat” format represents the input files associated with model parameters, pre-processed data and synthetic datasets. Moreover, the main interface of OpenIPDM has two important menus, namely the *Tools* menu and the *Analyses* menu, which are detailed in the next section.

## 2.2 Software functionalities

This section provides an overview for the main functionalities in OpenIPDM, which are accessible through either the *Tools* menus or the *Analyses* menu.

### 2.2.1 Software tools

The functionalities in the *Tools* menu are related to pre-processing real data, parameter estimation for the deterioration framework and generating synthetic data. The toolboxes available in the *Tools* menu are briefly described in the following list:

- **Read Real Database:** the purpose of this toolbox is to pre-process the real visual inspections database, where it takes “.csv” files as an input and produces “.mat” files ready for analyses in MATLAB. The extracted “.mat” files mainly contain information about the inspections and

structural attributes, which are arranged in a MATLAB cell-array variable following the template shown in Table 1.

Table 1: Example for data structure of the MATLAB cell-array variable containing information about the elements and structures.

	$e_1$	$\dots$	$e_E$
structure #1	$[T \times V]$	$\dots$	$[T \times V]$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
structure #B	$[T \times V]$	$\dots$	$[T \times V]$

The  $B$  rows in the cell-array correspond to the total number of structures that are visually inspected, while each column correspond to an element in the structure, with an array of input data  $T \times V$ . The  $T$  rows represent the total number of inspections performed on the structural element, while the  $V$  columns represent the element's information imported by the user. The information for each element include (from left-to-right): observations  $\tilde{y}$ , inspector ID  $I_i$ , inspection year  $t$ , structure ID, element's material identifier, structure's age and a subset of structural attributes selected by the user.

- **Generate Synthetic Database:** generating synthetic data is important for verifying any hypothesis in the deterioration model, performing controlled experiments and assessing the performance of the state estimation as well as parameter estimation framework. The toolbox ensures that the generated dataset is quantitatively and qualitatively representative of the real inspections database, through several criteria defined in the work of Hamida and Goulet [11, 15, 18].
- **Model Training:** there are three model training toolboxes, which are: *model training - generic*, *model training - intervention* and *model training - network*. The *model training - generic* toolbox is designated for estimating the deterioration model parameters of a single structural category (e.g., all the elements of type *beam*  $\mathcal{S}_1^j$ ), while the *model training - intervention* toolbox enables estimating the model parameters associated with the interventions framework [18]. On the other hand, the toolbox *model training - network* enables estimating the model parameters (deterioration and intervention) in batch-mode, for multiple structural categories (e.g., beams  $\mathcal{S}_1^j$ , slabs  $\mathcal{S}_2^j, \dots$ , etc). Prior to the training step, the user can determine the size of the training-validation-testing sets, which are employed to avoid overfitting. The parameter estimation functions rely mainly on the maximum likelihood estimate (MLE) method, and updates the parameters using a gradient-based optimization approach, and in some instances, the parameter estimation is done using Bayesian inference (e.g., estimating the inspectors parameters) [11, 19]. Due to the high number of structural elements in a normal case scenario (e.g.,  $E \gg 10^3$ ), the majority of operations throughout the parameter estimation are vectorized and rely mainly on using (GPU) as well as parallel computing using (CPU).

### 2.2.2 Software analyses

The *Analyses* menu in OpenIPDM provides access to model validation and verification analyses using real and synthetic data, respectively. The toolboxes under this menu are:

- **Model Verification:** the verification analyses on synthetic data involves assessing the predictive capacity of the deterioration model by comparing the estimated deterioration state with the true state. There are different verification assessments, such as, examining the prediction errors over time, and investigating any biases in the estimation. Furthermore, it is also possible to verify the estimation of the model parameters with the true parameters, especially, the parameters associated directly with the inspectors.
- **Model Validation:** this toolbox validates the deterioration model performance by relying on inspection data that is never employed in training the deterioration model. In order to perform the validation using this toolbox, it is required to have two complete real databases with one

database containing more inspection data than the other. The additional inspection data is considered as a test set, which can be utilized by the toolbox to determine if any bias exist in the deterioration model prediction or to report the log-likelihood of the additional inspection data.

### 2.2.3 Unit tests

In order to verify and maintain the different functionalities in the software, a number of unit tests are created and embedded in the interface. The main purpose of the unit tests is to verify the deterioration model performance using different sets of input. The unit tests are available and accessible from the *Developer* menu, for toolboxes with analyses and/or parameter estimation functionalities.

## 3 Illustrative Example

In this section, an example for generating and using synthetic data is presented. The example case include three steps : 1) generating synthetic data, 2) estimating model parameters, and 3) performing verification analyses. Each of the aforementioned steps are detailed in the following sections.

### 3.1 Generate Synthetic Dataset

In order to generate synthetic dataset, the toolbox *Generate Synthetic Data* is employed, which is accessible from the menu *Tools* in the main menu bar of OpenIPDM. The interface for this toolbox is simple, and requires mainly the model parameters as an input. For this example case, we generate inspection data for  $E = 20000$  structural elements, with the average service-life for each structural element is  $T = 60$  years. The total number of inspectors  $\mathcal{I} = \{I_1, I_2, \dots, I_I\}$  involved is  $I = 223$  synthetic inspectors. Each synthetic inspector is assumed to have an observation error model,  $v_t : V \sim \mathcal{N}(\mu_V(I_i), \sigma_V^2(I_i))$ . The standard deviation  $\sigma_V(I_i)$  and bias  $\mu_V(I_i)$  are generated for each synthetic inspector using a uniform distribution and according to user defined values. The main purpose of generating and assigning  $\sigma_V(I_i)$  and  $\mu_V(I_i)$  for each inspector is to be able to verify and compare the generated (i.e., true) bias with the estimated bias from the parameter estimation framework. In this example, we rely on the default input in the GUI, with minor adjustments shown in Table 2.

Table 2: Input required to configure and generate synthetic dataset using the *Generate Synthetic Data* toolbox.

Configuration of Example Synthetic Dataset			
Time Span	60	$\sigma_V(I_i)$	[1, 6]
Number of Time Series	20000	$\mu_V(I_i)$	[0, 0]
Number of Inspectors	223	Trans. Param.	4

After providing the input, it is possible to start the process of generating the synthetic dataset by pressing the button *Generate*. The time to generate the synthetic data mainly depends on the size of the dataset and the compatibility of configurations with the monotonic deterioration process. The output of this toolbox is a set of “.mat” files, which is obtained after pressing the *Save* button. The list of files is composed of,

- Generated Data: contains all the information involved in generating the synthetic inspection dataset.
- Full and Short: contains the true state and the corresponding sampled observations.
- InspectorID: has the unique identifier (ID) for each synthetic inspector.
- Observed: contains the synthetic inspections data.
- TrueInspector: includes the inspectors true  $\sigma_V(I_i)$ , bias  $\mu_V(I_i)$  as well as the synthetic inspectors IDs, which collectively can be employed to verify the parameter estimation process.

For additional information about the data generation step, the user can refer to the manual available with the software.

### 3.2 Model Parmater Estimation

In order to estimate the deterioration model parameters based on the generated synthetic inspection dataset, we use the toolbox *model training - generic*, accessible from the menu *Tools* in the main menu bar. The input to this toolbox include loading the files: “Observed.mat” and “InspectorID.mat” in the interface, using the load button. Since, the dataset in this example is synthetic, the user has to select the checkbox *Synthetic Data* on the interface. Thereafter, the user has to specify the deterioration model type and the optimization algorithm. In this example, we consider the use of the SSM deterioration model and rely on the *Newton-Raphson* method for the parameter estimation [11]. The next step is to configure the parameter bounds and the test set size, which is done by pressing the button *Model Config*. The *Model Config*. interface allows defining the parameters bounds, as well as other options relating to the parameter estimation process. For this example case, the user can rely on the default configuration for the model parameters, and save the settings by pressing the button *OK*. After this step, the user can processed to estimate the deterioration model parameters according to the configurations provided. It should be noted that the parameter estimation step requires an access to the parallel computing toolbox in MATLAB as well as a dedicated GPU card. At the end of the parameter estimation process, the toolbox provides two “.mat” files containing the model parameters and the inspectors’ parameters.

### 3.3 Verification Analyses & Results

The last step in this case study is to examine the predictive capacity of the deterioration model using the model parameters estimated in the previous step. This is done by using the toolbox *Verify Synthetic Data*, accessible from the *Analyses* menu in the main menu bar of OpenIPDM. There exist several assessments that can be performed in order to verify the deterioration model’s performance. In this example, we consider the average prediction error over a period of time  $T = 10$  years, for a group of  $E = 200$  synthetic structural elements. The inputs of the toolbox are demonstrated on the interface shown in Figure 4. After providing the input files and parameters, it is possible to execute the verification analyses by pressing the button *Run*.

	Sigma_W	Sigma_Conc	Sigma_Speed	Sigma_Acc	mu_Speed	mu_Acc	Min(Speed)
True	0.0050	1.0000	0.1000	0.1000	NaN	NaN	
Estimated	0.0038	1.0001	0.0500	0.0500	0	0	0.150

Figure 4: Input and configuration for the toolbox *Verify Synthetic Data*.

The processing time for the verification analyses mainly depends on the number of synthetic structural elements  $E$  involved in the analyses. After the analyses are done, the software outputs two graphs, shown in Figures 5 and 6. The first graph shows the absolute average error for the model predictions of the expected deterioration condition, speed and acceleration, as shown in Figure 5. On the other

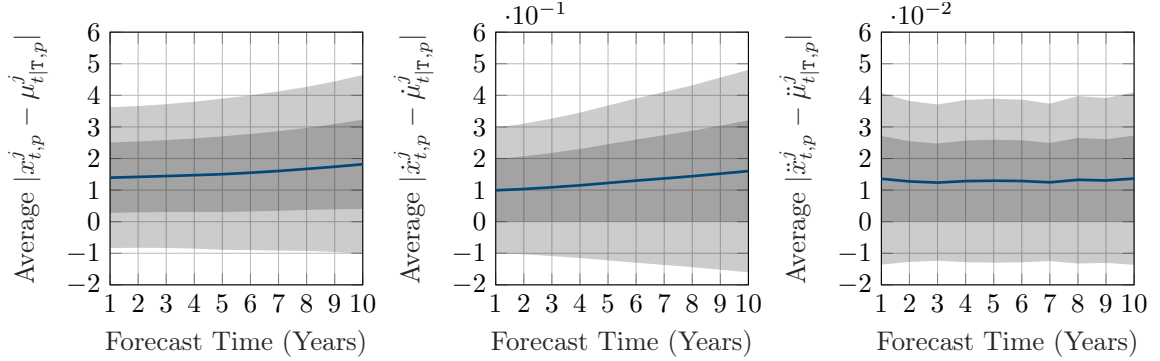


Figure 5: Absolute average error in prediction time for the expected condition, speed and acceleration based on the true condition, speed and acceleration respectively, with the confidence interval represented by the standard deviations  $\pm\sigma$  (dark grey shading) and  $\pm 2\sigma$  (light grey shading).

hand, the second graph shows the average error for the model predictions of the expected deterioration condition, speed and acceleration (Figure 6). Figures 5 and 6 represent the outcome of a single

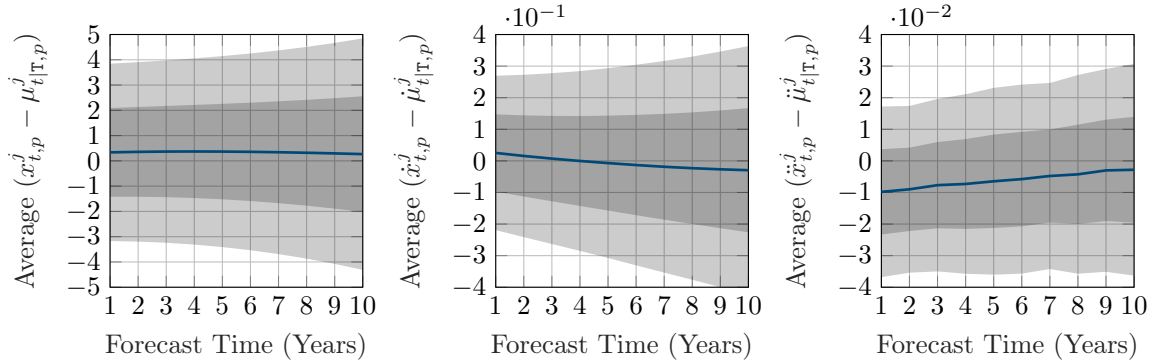


Figure 6: Average error in prediction time for the expected condition, speed and acceleration based on the true condition, speed and acceleration respectively, with the confidence interval represented by the standard deviations  $\pm\sigma$  (dark grey shading) and  $\pm 2\sigma$  (light grey shading).

assessment, however, there are other verification assessments, which are described in details within the software manual.

## 4 Impact

OpenIPDM represents a step towards the accessibility to information about network-scale visual inspection data, as well as the deterioration behaviour of infrastructures. In many instances, visual inspection databases are managed by private and/or government agencies, therefore, the access to the inspection data is limited. OpenIPDM overcomes the data availability by enabling users to generate synthetic datasets with similar characteristics to visual inspection data taken from a real case study. Moreover, the software provides insights into managing network-scale visual inspections and provides a template for pre-processing and organizing the information for an efficient computational time.

In addition, OpenIPDM relies on probabilistic methods with a demonstrated capacity to scale and a high potential for improvements in different areas. Promising improvement directions in this context are, enhancing the deterioration model by incorporating additional information about infrastructures, as well

as reducing the computational cost associated with the parameter estimation framework. Furthermore, the software provides solid foundations for the development and testing of decision-making and planning frameworks.

## 5 Conclusions

In this paper, we present OpenIPDM, an open-source software for modelling the deterioration behaviour of infrastructures. The software is composed of several toolboxes which collectively enable the design, validation and verification of network-scale deterioration models. The frameworks currently available in OpenIPDM has state-of-the-art capabilities in terms of estimating the inspectors' uncertainties and modelling the effect of interventions based on visual inspections. In addition, the software provides tools for managing large visual inspection databases as well as handling missing data and outliers. The current version of OpenIPDM mainly focuses on modelling the deterioration process using a data-driven approach based on visual inspections. Future developments could address additional aspects such as, handling failure modes, examining vulnerable zones as well as investigating the mechanisms where the deterioration leads to failure. OpenIPDM is open for contributions and users are encouraged to extend the software capacity in modelling the deterioration behaviour and/or develop additional functionalities such as, the development of decision making and planning modules.

## Acknowledgements

This project is funded by the Transportation Ministry of Quebec Province (MTQ), Canada. The authors would like to acknowledge the support of Simon Pedneault and René Gagnon for facilitating the access to information related to this project.

## References

- [1] S. B. Chase, Y. Adu-Gyamfi, A. Aktan, E. Minaie, Synthesis of national and international methodologies used for bridge health indices, Tech. rep., Turner-Fairbank Highway Research Center (2016).
- [2] W. Zhang, N. Wang, Bridge network maintenance prioritization under budget constraint, *Structural Safety* 67 (2017) 96–104  
URL <https://doi.org/10.1016/j.strusafe.2017.05.001>
- [3] Z. Allah Bukhsh, I. Stipanovic, G. Klanker, A. O'Connor, A. G. Doree, Network level bridges maintenance planning using multi-attribute utility theory, *Structure and Infrastructure Engineering* 15 (7) (2019) 872–885 1573–2479. doi:10.1080/15732479.2017.1414858.
- [4] A. Mandić Ivanković, A. Strauss, H. Sousa, European review of performance indicators towards sustainable road bridge management, in: *Proceedings of the Institution of Civil Engineers-Engineering Sustainability*, Vol. 173, Thomas Telford Ltd, 2019, pp. 109–124 doi:10.1680/jensu.18.00052.
- [5] J. Casas, A. Strauss, S. Fernandes, Cost action tu1406: Quality specifications for roadway bridges, standardization at a european level (bridgespec)—performance indicators, in: *Performance-Based Approaches for Concrete Structures-14th International Federation for Structural Concrete Symposium Proceedings*, Cape Town, South Africa, 2016.  
URL <http://hdl.handle.net/1822/42293>
- [6] M. Moore, B. M. Phares, B. Graybeal, D. Rolander, G. Washer, Reliability of visual inspection for highway bridges, volume i, Tech. rep., Turner-Fairbank Highway Research Center, 6300 Georgetown Pike (2001).

- [7] J. W. Soetjipto, T. J. W. Adi, N. Anwar, Bridge deterioration prediction model based on hybrid markov-system dynamic, in: The 6th International Conference of Euro Asia Civil Engineering Forum (EACEF 2017), Vol. 138, Édition Diffusion Presse Sciences, 2017, pp. 05001 doi:10.1051/mateconf/201713805001.
- [8] MTQ, Manuel d'Inspection des Structures, Ministère des Transports, de la Mobilité Durable et de l'Électrification des Transports (Jan 2014).
- [9] E. OBrien, C. Carey, J. Keenahan, Bridge damage detection using ambient traffic and moving force identification, Structural Control and Health Monitoring 22 (2015) 1396–1407. doi:10.1002/stc.1749.
- [10] L. E. Campbell, R. J. Connor, J. M. Whitehead, G. A. Washer, Benchmark for evaluating performance in visual inspection of fatigue cracking in steel bridges, Journal of Bridge Engineering 25 (1) (2019) 04019128 doi:10.1061/(ASCE)BE.1943-5592.0001507.
- [11] Z. Hamida, J.-A. Goulet, Modeling infrastructure degradation from visual inspections using network-scale state-space models, Structural Control and Health Monitoring (2020) 1545–2255doi:10.1002/stc.2582.
- [12] D. Agdas, J. A. Rice, J. R. Martinez, I. R. Lasa, Comparison of visual inspection and structural-health monitoring as bridge condition assessment methods, Journal of Performance of Constructed Facilities 30 (3) (2015) 04015049. doi:10.1061/(ASCE)CF.1943-5509.0000802.
- [13] J. Bennetts, G. Webb, S. Denton, P. Vardanega, N. Loudon, Quantifying uncertainty in visual inspection data, Taylor and Francis, London, UK, 2018, pp. 2252–2259.
- [14] Z. Hamida, J. Goulet, A stochastic model for estimating the network,Àêscale deterioration and effect of interventions on bridges, Structural Control and Health Monitoring (2021) 1545–2255doi:10.1002/stc.2916.
- [15] Z. Hamida, J.-A. Goulet, Network-scale deterioration modelling based on visual inspections and structural attributes, Structural Safety 88 (2020) 102024. doi:10.1016/j.strusafe.2020.102024.
- [16] P. Thompson, R. Shepard, Pontis, in: 7th Conference on Bridge Management, no. 7, Transportation Research Board, 1993.  
URL <https://trid.trb.org/view/390340>
- [17] R. M. Ellis, P. D. Thompson, R. Gagnon, G. Richard, Design and implementation of a new bridge management system for the ministry of transport of québec, in: The 10th International Conference on Bridge and Structure Management, Transportation Research Circular, 2008, pp. 77–86.  
URL <https://trid.trb.org/view/873719>
- [18] Z. Hamida, J.-A. Goulet, Quantifying the effects of interventions based on visual inspections of bridges network, Structure and Infrastructure Engineering (2021) 1–12doi:10.1080/15732479.2021.1919149.
- [19] B. Deka, J. Goulet, Online bayesian inference of process noise in state-space models., Manuscript submitted for publication (2021).

## Current code version

Nr.	Code metadata description	Please fill in this column
C1	Current code version	v1.0.0
C2	Permanent link to code/repository used for this code version	<a href="https://github.com/civml-polymtl/OpenIPDM/">https://github.com/civml-polymtl/OpenIPDM/</a>
C3	Permanent link to Reproducible Capsule	N.A.
C4	Legal Code License	MIT Licenses
C5	Code versioning system used	git
C6	Software code languages, tools, and services used	MATLAB
C7	Compilation requirements, operating environments & dependencies	MATLAB (version 2020b or higher), the MATLAB statistics and machine learning toolbox, and access to GPU and parallel computing (required only for Model Training toolbox).
C8	If available Link to developer documentation/manual	<a href="https://github.com/CivML-PolyMtl/OpenIPDM/tree/main/Help">https://github.com/CivML-PolyMtl/OpenIPDM/tree/main/Help</a>
C9	Support email for questions	<a href="mailto:zac.hamida@gmail.com">zac.hamida@gmail.com</a>

Table 3: Code metadata