**COVER SHEET**

***NOTE:***

- ***Please attach the signed copyright release form at the end of your paper and upload as a single 'pdf' file***
- ***This coversheet is intended for you to list your article title and author(s) name only***
- ***This page will not appear in the book or on the CD-ROM***

Title: *A NoSQL-Based Data Management Infrastructure for Bridge Monitoring Database*

Authors (names are for example only): Seongwoon Jeong
Yilan Zhang
Jerome P. Lynch
Hoon Sohn
Kincho H. Law

PAPER DEADLINE: ****May 31, 2015****

PAPER LENGTH: ****8 PAGES MAXIMUM ****

**Please submit your paper in PDF format. We encourage you to read attached Guidelines prior to preparing your paper—this will ensure your paper is consistent with the format of the articles in the CD-ROM.**

**NOTE:** Sample guidelines are shown with the correct margins. Follow the style from these guidelines for your page format.

Hardcopy submission: Pages can be output on a high-grade white bond paper with adherence to the specified margins (8.5 x 11 inch paper. Adjust outside margins if using A4 paper). Please number your pages in light pencil or non-photo blue pencil at the bottom.

Electronic file submission: When making your final PDF for submission make sure the box at "Printed Optimized PDF" is checked. Also—in Distiller—make certain all fonts are embedded in the document before making the final PDF.

(FIRST PAGE OF ARTICLE)

## ABSTRACT

There have been many significant advancements in the development of sensor and sensor network technologies for structural health monitoring applications. As structural monitoring technologies mature and their instrumentation on civil infrastructures continues to grow, data management has become an important issue for long term monitoring. On one hand, the data management system needs to be scalable to handle increasing amount of data over time as well as increasing number of sensors instrumented. On the other hand, the system needs to be sufficiently flexible to support tools for data analysis for diagnostic and prognostic purpose. This paper describes a data management infrastructure for bridge monitoring using state-of-the art NoSQL database technologies. Specifically, the discussion focuses on the data flow issue and the interaction between the monitoring database system and data analysis modules. The sensing data collected from the Telegraph Road Bridge in Monroe, Michigan is employed to illustrate the functionalities implemented so far and supported by the system.

## 1. INTRODUCTION

There have been many significant advancements in the development of sensors and sensor technologies for structural monitoring applications. Many bridges are now permanently instrumented with sensors (such as accelerometers, displacement transducers, strain gauge, thermal/temperature devices) [1, 2]. As sensor technologies mature and become economically affordable, their deployment for large scale infrastructure monitoring will continue to grow [3]. Sensor technologies have brought many advantages on the maintenance of infrastructure and facilitate routine inspection

Seongwoon Jeong, Dept. of Civil & Environ. Eng., Stanford University, Stanford, CA, USA 94305
Yilan Zhang, Dept. of Civil & Environ. Eng., University of Michigan, Ann Arbor, MI, USA 48109-2125
Jerome P. Lynch, Dept. of Civil & Environ. Eng., University of Michigan, Ann Arbor, MI, USA 48109-2125
Hoon Sohn, Dept. of Civil & Environ. Eng., KAIST, Daejeon 305-701, Republic of Korea
Kincho H. Law, Dept. of Civil & Environ. Eng., Stanford University, Stanford, CA, USA 94305

using advanced nondestructive evaluation technology (such as ultrasonic devices, acoustic emission, laser scanner) [4, 5]. In addition to acquiring valuable information about the state of the structures and the potential for identifying structural damage, sensing technologies can help facilitate and prioritize maintenance work. The trend for bridge monitoring system involves permanent installation of hundreds and thousands of sensors. The collection of long-term sensor data allows extraction of statistically meaningful information and affords data-driven predictive analysis [6, 7].

While long-term monitoring can potentially enable better assessment of the state of infrastructures, the enormous amount of sensor information poses significant challenges in data management. The data needs to be stored, processed, interpreted and, desirably, integrated with lifecycle bridge management. While structural health monitoring research continues to develop and explore new sensor technologies, very few efforts have been devoted to investigate proper data management tools to efficiently store, manage and retrieve sensor data [8, 9, 10].

Selecting an appropriate database tool for specific application is key to successful deployment of a data management system. Different database tools have different strength and properties. Given the potentially enormous quantity and diversity of sensing data and complexity of bridge model, it would be desirable that the database tools employed for bridge monitoring and management system are highly scalable and flexible. Traditional relational database management systems (RDBMS) have the strict table-type data structure and explicit relationships defined among the data. Recent studies have shown that RDBMS do not perform well when dealing with large volume of unstructured data [11, 12]. Not Only SQL (NoSQL) database systems, which are highly scalable and support flexible data schema, have been proposed as an alternative to RDBMS [13, 14]. It has been reported that NoSQL database system can achieve better performance than RDBMS in terms of scalability, flexibility, and low latency by relaxing the rigid data consistency and strict data schema definition of RDBMS [11, 14].

This paper discusses a data management infrastructure for SHM system utilizing NoSQL database systems. The data management infrastructure is designed not only for the management of sensor data but also for bridging the database support for bridge information modeling and management. For the prototype development, NoSQL database tools, Apache Cassandra [15] and MongoDB [16], are selected to store and manage sensing data and the relevant bridge information. Software tools are being developed to automatically convert, parse, and store sensor data. Furthermore, with the NoSQL databases as data storage for onsite computer and central server, interfaces are being designed and implemented to link external analysis modules to enhance data interoperability and software integration. In this paper, sensor data collected from Telegraph Road Bridge (TRB) in Monroe, Michigan are employed to illustrate the interactions between the database system and external analysis modules that are commonly used for structural identification and data mining and analysis.

## 2. DATA MANAGEMENT FRAMEWORK

Figure 1 shows the overall system architecture of a data management infrastructure for SHM, which consists of an on-site computer, a main server, a local computer, and a web interface to users and tools. Once sensors acquire and, if

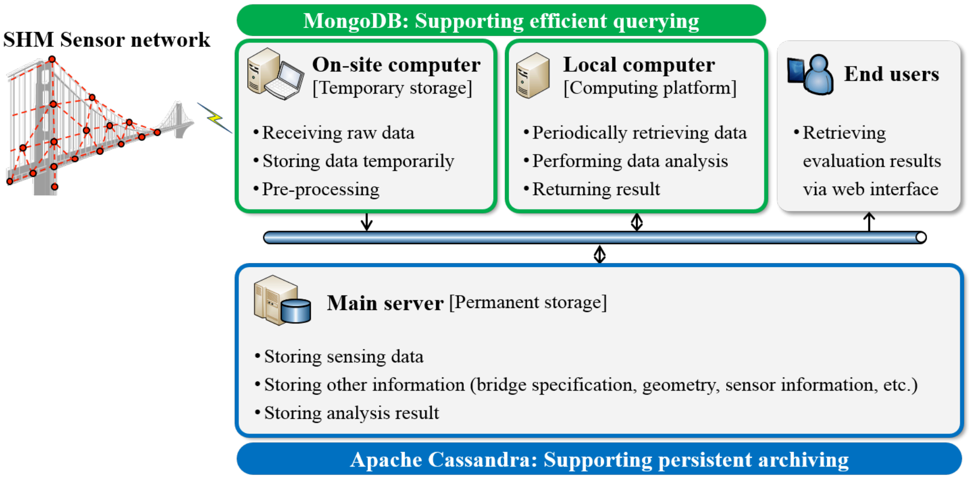

SHM Sensor network
MongoDB: Supporting efficient querying
On-site computer
[Temporary storage]
• Receiving raw data
• Storing data temporarily
• Pre-processing
Local computer
[Computing platform]
• Periodically retrieving data
• Performing data analysis
• Returning result
End users
• Retrieving evaluation results via web interface
Main server [Permanent storage]
• Storing sensing data
• Storing other information (bridge specification, geometry, sensor information, etc.)
• Storing analysis result
Apache Cassandra: Supporting persistent archiving

Figure 1. Data management framework for bridge monitoring

necessary, preprocess the response data from the bridge, the sensing data are then transmitted to the on-site computer. The on-site computer temporarily stores the data and, if appropriate, performs real time analysis. The data are then parsed and sent to the main server. The main server is the central data repository where all sensing data and relevant bridge information (such as bridge geometry and analysis results) are stored and persistently archived. Local computers allow the users and application tools to retrieve data from the central server for data analysis and return the results to the repository. Finally, end users such as bridge managers can retrieve the analysis results about the state of the bridge via a web-based interface. To facilitate bridge management functions, data schemas for the bridge information are currently designed according to XML-based bridge information modeling (BrIM) structure [17, 18].

There have been a number of NoSQL database systems with different features and properties. In general, NoSQL database systems can be categorized into key-value stores, document-oriented stores, and column family stores according to the data model employed [11]. In this study, we select MongoDB [19], document-oriented data storage that features powerful query capability, and Apache Cassandra [20], a column-oriented storage that features scalable distributed data storage. Specifically, because of its scalability and flexible data schema structure, Apache Cassandra is employed for persistent data archiving for the central data repository. On the other hand, MongoDB is selected for the on-site and local computers because of its schema-free features, rich aggregation, and efficient query for supporting real time analysis (such as Internet of Things (IoT) applications [21-24]). Details on the mapping and implementation of BrIM schemas on Cassandra and MongoDB database system have been discussed elsewhere [10]. This paper describes the basic data flows for the archiving of sensor data and the integration of the database system with external application modules.

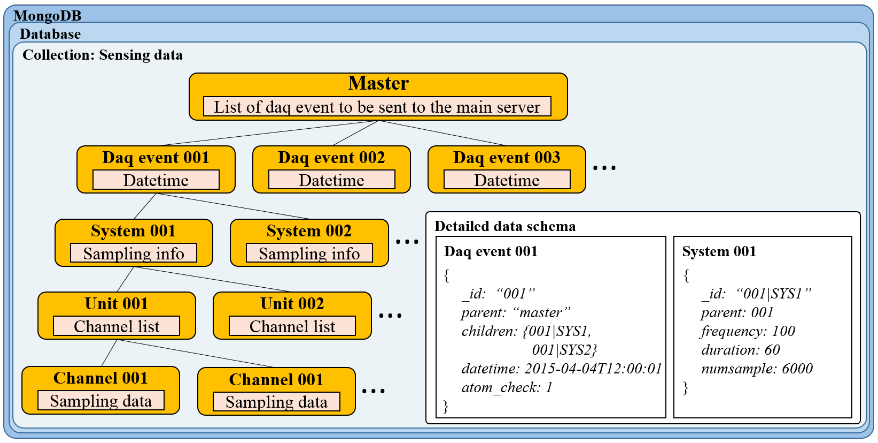
## 3. AUTOMATED DATA FLOW

This section describes in details the basic data flow for parsing, transmitting and archiving sensor data from the sensor units to the local, on-site database and the central data repository at the main server. To automate the process, schemas for sensor data are designed for MongoDB and Apache Cassandra database systems. Specifically, for illustration purpose, we use NARADA wireless sensor units [25, 26] on our prototype development.

Figure 2 shows the basic hierarchical schema for the MongoDB database on the onsite computer. The uppermost namespace is called database, which contains a set of collections. The root node of a collection consists of a single “master” document (a fundamental data units in the MongoDB database). For the tree-hierarchy, the non-leaf nodes (such as “daqevent”, “system”, and “unit”) provide the basic information about the DAQ and sensors while the leaf nodes contains the acquired sensing data. The “master” document contains information about the data flow regarding how the data are to be stored. For each DAQ event, a single document named “daqevent” as well as other subordinate documents such as “system” and “unit” are created to store the metadata (such as acquisition time, sampling rate, and sampling period) about the sensing data. The (time series) sensing data are stored in the leaf node “channel”. The “master” document retains a list of the “daqevent” documents that are to be sent to the main server. The organization supports efficient search and transmission of the sampling data from the on-site computer to the main server without duplicating the data. The system ensures that all sampling data are properly transmitted to the main server by marking every “daqevent” and “channel” document if and only if the sampling data has been successfully sent and received by main server.

Figure 3 shows the data schema for storing the sensing data in the Apache Cassandra database system. Specifically, we employ Oliot-EPCIS platform for the Cassandra database [27]. Apache Cassandra’s data partitioning strategy represented by “consistent hashing” has shown performance advantages on distributed database, but the consistent hashing strategy could weaken the range query performance since the time series data can be distributed to different nodes. The Oliot-EPCIS guarantees that the consecutive data can be placed in the same node by allocating partitioning key according to the sensor’s ID and acquisition date. For the current schema design, each row stores up to one second of time series data acquired by a single sensor channel.

To automate the data flow from the sensor node to MongoDB database on the onsite computer, and then to the Cassandra database in the main server, two interface programs, written using Python, are developed. The first program, which serves as the interface between NARADA sensor units and MongoDB in the on-site computer,



```
{
    _id:  “001”
    parent: “master”
    children: {001|SYS1,
                   001|SYS2}
    datetime: 2015-04-04T12:00:01
    atom_check: 1
}
```

```
{
    _id:  “001|SYS1”
    parent: 001
    frequency: 100
    duration: 60
    numsample: 6000
}
```

Figure 2. Data schema for MongoDB

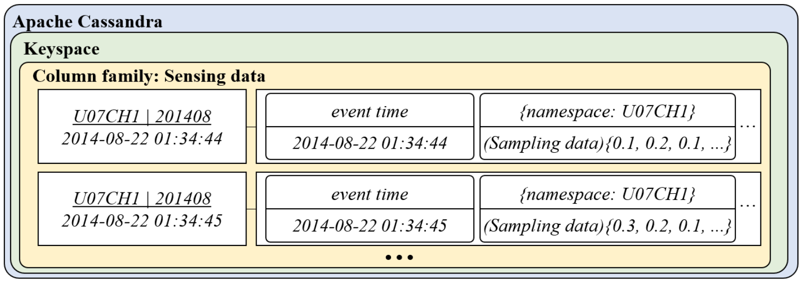


Figure 3. Data schema for Apache Cassandra

keeps monitoring the designated directory in the on-site computer where the NARADA sensor units send the data. Once the sensors send a new set of data, the program parses the raw data into an input format according to the defined data schema and uploads the data to the MongoDB database. The second program connects the MongoDB in the on-site computer and Apache Cassandra database in the main server. Once the first program pushes a new DAQ event to the waiting list in the "master" document of the MongoDB, the second program accesses the acquired data stored in the MongoDB and parses the data into the Oliot-EPCIS input format. The parsed data is then sent to the Apache Cassandra database in the main server via the web interface of the Oliot-EPCIS platform.

## 4. DATABASE SUPPORTS FOR MONITORING APPLICATION

In addition to support systematic persistent store and archive of data, one of the key functions of the database system is to provide easy access to server by application tools. In this study, we illustrate the seamless integration and data interoperability among different tools using the database systems.

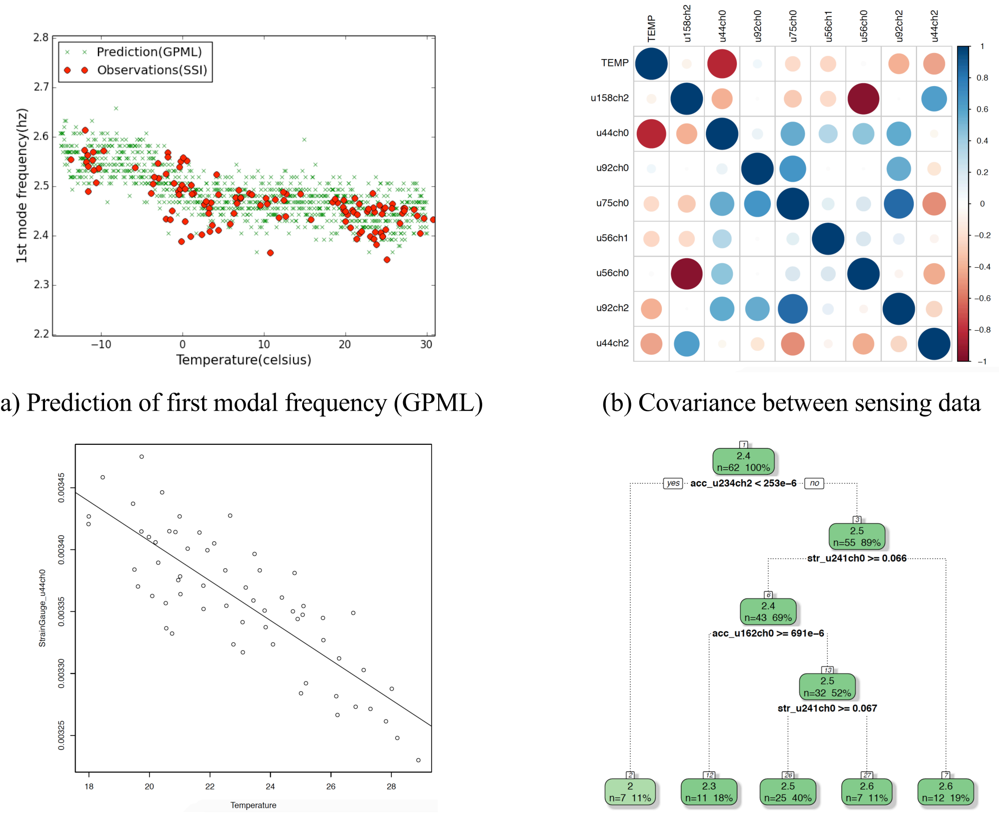
The data sets collected from Telegraph Road Bridge (TRB) in Monroe, Michigan are employed in the application scenario. The data sets contain seven weeks of raw sensor data: one week per a month from August 2014 to February 2015. The data are collected from a sensor network of 14 accelerometers, 40 strain gauges, and 6 thermistors, as described by O'Connor et al [6]. Sensor measurements are acquired approximately every 2 hours for one minute time duration. The sampling rate for the accelerometers is 200Hz, while strain gauges and thermistors have the sampling rate of 100Hz.

To simulate the monitoring scenario, the collected raw sampling data is periodically sent to a (on-site) laptop computer. As discussed in the previous section, the data received are automatically re-structured and stored according to the defined schema on the MongoDB database on the laptop as well as uploaded to the Apache Cassandra database in the server.

For illustrative purpose, the application scenario involves three analysis modules namely a modal analysis module, a Gaussian Process for Machine Learning (GPML) module, and a statistical computing and graphic module. We employ Stochastic Subspace Identification (SSI) algorithm to perform output-only modal analysis [28]. The sensing data are retrieved from the Cassandra database on the main server and processed using the subspace identification package written in MATLAB [29] on the (local) laptop computer. The calculated modal properties are then uploaded to the Apache Cassandra in the main server. The red dots shows in Figure 4(a) depict the first modal frequencies calculated by the modal analysis module (which are plotted here along with temperature measurement).

Once the modal properties are stored in the main server, the GPML module retrieves natural frequencies as well as thermistor data and processed using “scikit-learn” (a Python-based machine learning) package. The GPML module returns a predictive model for natural frequencies based on temperature data. As for illustration, the prediction results for the first modal frequencies for different temperatures are shown as green crosses in Figure 4(a).

The analysis results can be further processed by other application packages. To illustrate, we develop an interface module to allow retrieval of data from R [30]. Figures 4(b), 4(c), and 4(d) show post-processes of analysis results by the R module. Figure 4(b) is a covariance matrix plot between one thermistor and several strain gauges to discover possible interesting correlation pattern between sensors. Figure 4(c) shows the linear model fitting result between temperatures versus strain values acquired from a single strain gauge installed on the bottom of the bridge. Figure 4(d) illustrates the result of the tree analysis to predict the first modal frequency based on the acceleration and strain values measured from each sensor. For example, the result of tree analysis implies that the first modal frequency will become 2.0 when the acceleration measured by the accelerometer “u234ch2”, which measures the acceleration of longitudinal direction at the center of the span, is less than 253×$10^{-6}$ g.



(a) Prediction of first modal frequency (GPML)

(b) Covariance between sensing data

(c) Linear model fitting

(d) Tree analysis

Figure 4. Data analysis results

## 5. DISCUSSION

In this study, a data management infrastructure for structural health monitoring system is proposed. Two NoSQL database systems, Apache Cassandra and Mongo DB, are deployed to improve the scalability and flexibility of the proposed data management system. On top of the database systems, two software tools are developed to automate data flow from sensor network to the main server. The main server is also connected to the data analysis modules such as Stochastic Subspace Identification module, Gaussian Process regression module, and R. As a result, a user can easily query sensing data and relevant information from the main server and conduct data analysis successfully without the burden on the low-level details of data flow.

## ACKNOWLEDGMENTS

This research is supported by a Grant No. 13SCIPA01 from Smart Civil Infrastructure Research Program funded by Ministry of Land, Infrastructure and Transport (MOLIT) of Korea government and Korea Agency for Infrastructure Technology Advancement (KAIA). The research is also partially supported by the US National Science Foundation (NSF), Grant No. ECCS-1446330 to Stanford University and Grant No. CMMI-1362513 and ECCS-1446521 to the University of Michigan. Additional support was provided by the U.S. Department of Transportation Contract Number OASRTRS-14-H-MICH. The authors thank the Michigan Department of Transportation (MDOT) for access to the Telegraph Road Bridge and for offering support during installation of the wireless monitoring system. Any opinions, findings, conclusions or recommendations expressed in this paper are solely those of the authors and do not necessarily reflect the views of NSF, MOLIT, KAIA or any other organizations and collaborators.

## REFERENCES

1. Zhou, G. D. and Yi, T. H., “Recent developments on wireless sensor networks technology for bridge health monitoring,” *Mathematical Problems in Engineering*, 2013, 947867 (2013).
2. Jang, S., Jo, H., Cho, S., Mechitov, K., Rice, J. A., Sim, S., Jung, H., Yun, C., Spencer, Jr., B. F., and Agha, G., “Structural health monitoring of a cable-stayed bridge using smart sensor technology: deployment and evaluation,” *Smart Structures and Systems*, 6(5-6), pp. 439-459 (2010).
3. Lynch, J. P., and Loh, K. J., “A summary review of wireless sensors and sensor networks for structural health monitoring,” *Shock and Vibration Digest*, 38(2), pp. 91-130 (2006).
4. Lim, H. J., Sohn, H., and Liu, P., “Binding conditions for nonlinear ultrasonic generation unifying wave propagation and vibration,” *Applied Physics Letters*, 104(21), 214103 (2014).
5. Lieske, U., Dietrich, A., Schubert, L., and Frankenstein, B., “Wireless system for structural health monitoring based on Lamb waves,” *Proceedings of SPIE - The International Society for Optical Engineering*, 8343, art. no. 83430B (2012).
6. O'Connor, S. M., Zhang, Y., Lynch, J., Ettouney, M., and Van Der Linden, G., “Automated analysis of long-term bridge behavior and health using a cyber-enabled wireless monitoring system,” *Proceedings of SPIE - The International Society for Optical Engineering*, 9063, art. no. 90630Y (2014).
7. Cross, E. J., Koo, K. Y., Brownjohn, J. M. W., and Worden, K., “Long-term monitoring and data analysis of the Tamar Bridge,” *Mechanical Systems and Signal Processing*, 35(1-2), pp. 16-34 (2013).

8. Law, K. H., Smarsly, K., and Wang, Y., “Sensor data management technologies for infrastructure asset management,” in *Sensor Technologies for Civil Infrastructures: Applications in Structural Health Monitoring*, Wang, M. L., Lynch, J. P., and Sohn, H. (Eds.), Woodhead Publishing, Cambridge, UK, 2(1), pp. 3-32 (2014).
9. Zhang, Y., Kurata, M., Lynch, J. P., Van der Linden, G., Sederat, H., and Prakash, A., “Distributed cyberinfrastructure tools for automated data processing of structural monitoring data,” *Proceedings of SPIE - The International Society for Optical Engineering*, 8347, art. no. 83471Y (2012).
10. Jeong, S., Byun, J., Kim, D., Sohn, H., Bae, I. H., and Law, K. H., "A data management infrastructure for bridge monitoring,” *Proceedings of SPIE - Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2015*, 94350P (2015).
11. Hecht, R., and Jablonski, S., “NoSQL Evaluation: A use case oriented survey,” *Proceedings - 2011 International Conference on Cloud and Service Computing*, CSC 2011, art. no. 6138544, pp. 336-341 (2011).
12. Li, Y., & Manoharan, S. “A performance comparison of SQL and NoSQL databases,” *IEEE Pacific RIM Conference on Communications, Computers, and Signal Processing - Proceedings*, art. no. 6625441, pp. 15-19 (2013).
13. Moniruzzaman, A. B. M., and Hossain S. A., “NoSQL Database: New Era of Databases for Big data Analytics - Classification, Characteristics and Comparison,” *International Journal of Database Theory and Application*, 6(4), pp. 1-14 (2013).
14. Han, J., Haihong, E., Le, G., and Du, J., “Survey on NoSQL database,” Proceedings - *2011 6th International Conference on Pervasive Computing and Applications*, ICPCA 2011, art. no. 6106531, pp. 363-366 (2011).
15. “Cassandra,” [Online]. Available: http://cassandra.apache.org/ [Accessed: May. 2015].
16. “MongoDB,” [Online]. Available: http://www.mongodb.org/ [Accessed: May. 2015].
17. Ali, N., Chen, S. S., Srikonda, R., and Hu, H., “Development of Concrete Bridge Data Schema for Interoperability,” *Transportation Research Record*, 2406(1), 87-97 (2014).
18. “OpenBrIM,” [Online]. Available: https://openbrim.appspot.com/ [Accessed: May. 2015].
19. “Applications Never Before Possible,” [Online]. Available: http://www.mongodb.com/use-cases?_ga=1.233202590.1808870624.1414965022 [Accessed: May. 2015].
20. “Apache Cassandra Use Cases,” [Online]. Available: http://planetcassandra.org/apache-cassandra-use-cases [Accessed: May. 2015].
21. “Application never before possible,” [Online]. Available: http://www.mongodb.com/use-cases [Accessed: May, 2015].
22. “How a database can make your organization faster, better, leaner: examples and guidelines for the enterprise decision maker,” A MongoDB White Paper (2013) [Online]. Available: http://dws.la/wp-content/uploads/2014/12/MongoDB_Better_Faster_Leaner.pdf [Accessed: May, 2015].
23. “Chicago’s WindyGrid: using Mongodb to create a smarter and safer city,” [Online]. Available: http://www.mongodb.com/customers/city-of-chicago [Accessed: May, 2015].
24. “IoT and Big Data: A Joint Whitepaper by Bosch Software Innovations and MongoDB,” A Bosch Software Innovations and MongoDB White Paper (2015) [Online]. Available: http://info.mongodb.com/rs/mongodb/images/MongoDB_BoschSI_IoT_BigData.pdf [Accessed: May, 2015].
25. R. A. Swartz, D. Jung, J. P. Lynch, Y. Wang, D. Shi, and M. Flynn, “Design of a wireless sensor for scalable distributed in-network computation in a structural health monitoring system,” *Proceedings of the 5th International Workshop on Structural Health Monitoring*, pp. 1570–1577, (2005).
26. “NARADA wireless data acquisition and control system,” [Online]. Available: http://www.civionics.com/NaradaTechSheet.pdf [accessed: May, 2015].
27. Le, T. D., Kim, S. H., Nguyen, M. H., Kim, D., Shin, S. Y., Lee, K. E., and da Rosa Righi, R., “EPC information services with No-SQL datastore for the Internet of Things,” *2014 IEEE International Conference on RFID, IEEE RFID 2014*, art. no. 6810711, pp. 47-54 (2014).
28. Peeters, B. and De Roeck, G., “Reference-based stochastic subspace identification for output-only modal analysis,” *Mechanical Systems and Signal Processing*, 13 (6), pp. 855-878 (1999).
29. “Subspace Identification for Linear Systems,” [Online]. Available: http://www.mathworks.com/matlabcentral/fileexchange/2290-subspace-identification-for-linear-systems [accessed: May, 2015].
30. “The R Project for Statistical Computing,” [Online]. Available: http://www.r-project.org/ [accessed: May, 2015].