

## ABSTRACT

Title of Dissertation Proposal: **COST-EFFECTIVE PROGNOSTICS AND HEALTH MONITORING OF LOCALLY DAMAGED PIPELINES WITH HIGH CONFIDENCE LEVEL**

Amin Aria, Doctor of Philosophy, 2020

Dissertation directed by: **Dr. Mohammad Modarres, Center for Risk and Reliability, Department of Mechanical Engineering**

**Dr. Shapour Azarm, Department of Mechanical Engineering**

Localized pipeline damages, caused by degradation processes such as corrosion, are prominent, can result in pipeline failure and are expensive to monitor. To prevent pipeline failure, many Prognostics and Health Monitoring (PHM) approaches have been developed in which sensor network for online, and human inspection for offline data gathering are separately used. In this dissertation, a two-level (segment- and integrated-level) PHM approach for locally damaged pipelines is proposed where both of these degradation data gathering schemes (i.e., detection

methods) are considered simultaneously. The segment-level approach, in which the damage behavior is considered to be uniform, consists of a static and a dynamic phase. In the static phase, a new optimization problem for health monitoring layout design of locally damaged pipelines is formulated. The solution to this problem is an optimal configuration (or layout) of degradation detection methods with a minimized health monitoring cost and a maximized likelihood of damage detection. In the dynamic phase, considering the optimal layout, an online fusion of high-frequency sensors data and low-frequency inspection information is conducted to estimate and then update the pipeline's Remaining Useful Life (RUL) estimate. Subsequently, the segment-level optimization formulation is modified to improve its scalability and facilitate updating layouts considering the online RUL estimates. Finally, at the integrated-level, the modified segment-level approach is used along with Stochastic Dynamic Programming (SDP) to produce an optimal set of layouts for a long pipeline consisting of multiple segments with different damage behavior.

Experimental data and several notional examples are used to demonstrate the performance of the proposed approaches. Synthetically generated damage data are used in two examples to demonstrate that the proposed segment-level layout optimization approach results in a more robust solution compared to single detection approaches and deterministic methods. For the dynamic segment-level phase, acoustic emission sensor signals and microscopic images from a set of fatigue crack experiments are considered to show that combining sensor- and image-based damage size estimates leads to accuracy improvements in RUL estimation. Lastly, using synthetically generated damage data for three hypothetical pipeline segments, it is shown that the constructed integrated-level approach provides an optimal set of layouts for several pipeline segments.

COST-EFFECTIVE PROGNOSTICS AND HEALTH MONITORING  
OF LOCALLY DAMAGED PIPELINES  
WITH HIGH CONFIDENCE LEVEL

by  
Amin Aria

Dissertation submitted to the Faculty of the Graduate School of the  
University of Maryland, College Park, in partial fulfillment  
of the requirements for the degree of  
Doctor of Philosophy  
2020

Advisory Committee:

Dr. Mohammad Modarres, Advisor and Chair  
Dr. Shapour Azarm, Co-advisor  
Dr. Balakumar Balachandran  
Dr. Mark Fuge  
Dr. Katrina Groth  
Dr. Kayo Ide, Dean's Representative  
Dr. Vasiliy Krivtsov

© Copyright by  
Amin Aria  
2020

## Dedication

*To my wonderful, kind, and thoughtful parents,  
Parto & Morteza*

*Without whom  
nothing was achievable for me in this labyrinth of life!*

## Acknowledgements

As an adventurous and ambitious person, I have always been looking for challenges, and now, after concluding these years of my Ph.D. studies, I am grateful that I can consider Ph.D. as the accomplished challenge of academic challenges; an adventure that never let me down even when it got overwhelming from time to time. Similar to every challenge, it would have been impossible to successfully go through the journey of a Ph.D. without receiving help and support from the wise and kind people who were familiar with the obstacles along the way. Thus, I want to hereby thank all of those who helped me through this wonderful journey.

First and foremost, I would like to express my sincere gratitude to my dear advisors Dr. Mohammad Modarres and Dr. Shapour Azarm for their support, patience, and insightful supervision. Most importantly, as a priceless lesson, they taught me how to be relentless and stay strong when facing seemingly unsolvable problems. Moreover, they were great role models when it came to being modest and thirsty for discovering new things and knowing more, all whilst being the best in their fields of expertise. Specifically, Dr. Azarm taught me the great lesson of resilience and staying motivated. On the other hand, Dr. Modarres has always been an intellectual mentor for me whom I also consider a great life mentor. I am sure I will always look up to Professor Modarres whenever I feel like I need to be a better person in my personal or professional life.

I would also like to express boundless gratitude toward Dr. Lopez Droguett who helped me in the completion of my academic goals. It is truly difficult for me to put his high level of wisdom and intelligence into words. I am so thankful of him since he helped me navigate toward my academic goals as efficiently as possible, all while being thousands of miles away. I also want to thank my friends and officemates, especially Dr. Huisung Yun and Mr. Sergio Cofre, at the lab of Center for

Risk and Reliability at the University of Maryland. Their support, recommendations, and thoughtful suggestions helped me immensely while I was handling the many challenges of this journey. Moreover, I want to thank all my dear Iranian friends who have always been there for me during all these years of being away from home.

In addition, I would like to thank all my dissertation committee members Dr. Balakumar Balachandran, Dr. Mark Fuge, Dr. Katrina Groth, Dr. Kayo Ide, and Dr. Vasiliy Krivtsov for reviewing the dissertation and providing valuable feedback to improve this study.

This research was carried out as a part of the Pipeline System Integrity Management Project, which was supported by the Petroleum Institute, Khalifa University of Science and Technology, Abu Dhabi, UAE. I want to hereby gratefully acknowledge this support, which made this journey much more pleasant and fulfilling for me.

Beyond all, I wish there was a language that could encompass my feelings for my loving parents who truly are the epitome of beauty for me, in this mesmerizing universe. My love and gratitude toward my beloved mom and dad for their unconditional love, support, and sacrifice cannot be contained in words. So, as I am not able to express my love for them in words, let me end by saying that I am eternally thankful of them.

# Table of Contents

<b>Chapter 1: Introduction, Problem Statement, and Research Overview</b> .....	1
1.1 Literature Gaps .....	3
1.1.1 Data Gathering Schemes and their Attributes .....	3
1.1.2 Probabilistic Detection Metrics .....	4
1.1.3 Uncertainty Consideration and Probabilistic Sampling of Localized Damages .....	5
1.1.4 Health Monitoring Optimization .....	6
1.1.5 Data Fusion for Damage Size and RUL Estimation.....	6
1.1.6 Damage Size and RUL Estimation Based on Sensors' Time-Series Data .....	7
1.1.7 Damage Size and RUL Estimation Based on Inspection Images.....	9
1.1.8 Integrated and Dynamic Health Monitoring of Pipelines .....	10
1.2 Problem Statement .....	11
1.3 Research Objectives (ROs) .....	13
1.4 Research Tasks (RTs).....	14
1.5 Overview of the Proposed Approach .....	15
<b>Chapter 2: Layout Optimization of Multi-type Sensors and Human Inspection Tools with Probabilistic Detection of Localized Damages for Pipelines</b> .....	19
2.1 Optimal Health Monitoring Layout Problem Statement .....	19
2.2 Probabilistic Detection Metrics .....	20
2.2.1 Probability of Detection (POD).....	21



2.2.2 Inference Probability (IP).....	22
2.2.3 Measurement Error (ME).....	25
2.3 Proposed Approach for Optimal Placement of Sensors and Human Inspection.....	25
2.4 Health Monitoring Layout Optimization Problem Formulation .....	31
2.5 Examples .....	35
2.5.1 Example 1.....	36
2.5.1.1 Step-by-Step Application of the Proposed Approach .....	37
2.5.1.2 Single Realization Results and Discussion .....	41
2.5.1.3 Multiple Realization Results and Final Aggregate Layout .....	43
2.5.2 Example 2.....	47
2.6 Concluding Remarks .....	48
<b>Chapter 3: Estimating Damage Size and Remaining Useful Life in Degraded</b>	
<b>Structures Using Deep Learning-based Multi-source Data Fusion .....</b>	<b>50</b>
3.1 Proposed Approach for Damage size and RUL Estimation .....	50
3.1.1 Module for Time-series Analysis.....	52
3.1.1.1 LSTM-based Regression for Damage Size Estimation.....	52
3.1.1.2 Customized LSTM-Based Regression Model for Fatigue Crack Size Estimation .....	55
3.1.2 Module for Image Processing.....	57
3.1.2.1 FCDN for Damage Size Estimation .....	57
3.1.2.2 Proposed FCDN-based Approach for Fatigue Crack Size Estimation.....	61

3.1.3 Module for Data Fusion and RUL Estimation .....	64
3.2 Experimental Set-Up and Data Pre-Processing.....	64
3.2.1 Overview of the Experiments.....	65
3.2.2 Application of POF Model in the Time-Series and Image Processing Modules .....	66
3.2.3 Data Pre-Processing and Primary Data Analytics .....	66
3.3. Results and Discussion.....	68
3.3.1 Time-Series Analysis Results.....	69
3.3.2 Image Processing Results.....	71
3.3.3 Data Fusion and RUL Estimation Results.....	73
3.3.4 Evaluation of Performance for All Five Experiments.....	76
3.4. Concluding Remarks .....	79
<b>Chapter 4: Dynamic and Integrated Health Monitoring of Pipelines .....</b>	<b>81</b>
4.1 Problem Description.....	82
4.2 Modified Formulation for Improving Scalability.....	83
4.3 Dynamic (Online) Health Monitoring.....	85
4.4 Stochastic Dynamic Programming (SDP) for Integrated Health Monitoring of Pipelines .	89
4.4.1 Dynamic Programming .....	90
4.4.2 Integrated Health Monitoring of Pipelines using SDP.....	93
4.4.2.1 Integrated Pipeline Health Monitoring: SDP Problem.....	94
4.4.2.2 Illustrative Case Study.....	99

4.4.2.3 Results and Discussion.....	102
4.5 Concluding Remarks .....	104
<b>Chapter 5: Conclusions, Contributions, and Potential Future Directions.....</b>	<b>106</b>
5.1 Summary of the Approaches .....	106
5.1.1 Summary of the Approaches for the Static and Dynamic Phases of the Segment-Level Analysis.....	106
5.1.2 Summary of the Integrated-Level Approach.....	108
5.2 Research Contributions .....	110
5.3 Potential Future Research.....	112
Appendix A .....	116
Appendix B.....	123
Appendix C.....	129
References .....	132

## List of Tables

<i>Table 1- Symbols used in the optimization model (Eqs. 6.1 – 6.21).</i> .....	33
<i>Table 2- Specifications of detection methods</i> .....	37
<i>Table 3- POD as a function of damage type-size class</i> .....	39
<i>Table 4- Class intervals and corresponding ME probabilities (<math>\delta_{k,m}</math>)</i> .....	40
<i>Table 5- Clustered damages of realization #1</i> .....	40
<i>Table 6- Utility weights used in utility function</i> .....	41
<i>Table 7- Comparison of performance of the layouts of Figure 11</i> .....	43
<i>Table 8- Statistics of improvement percentage in layout performance for 46 realizations of Example 1 when clustering and ME are considered</i> .....	44
<i>Table 9 -Hyper parameters of FCDN_AA for semantic segmentation of fatigue crack images...</i>	62
<i>Table 10- Sample AE data for five experiments when the crack length reaches 999 <math>\mu\text{m}</math>.</i> .....	67
<i>Table 11- Growth of fatigue crack of Experiment 1 and corresponding manual size measurements</i> .....	69
<i>Table 12-Hyperparameters of bi-LSTM_AA</i> .....	70
<i>Table 13- LSTM structures with best performance metrics for different <math>N_{tw}</math> values</i> .....	70
<i>Table 14- Hyperparameters and performance metrics of FCDN_AA</i> .....	73
<i>Table 15- Segmented Images and corresponding size estimations of Experiment 1 by FCDN_AA</i> .....	73
<i>Table 16- Hyperparameters of bi-LSTM_AAF</i> .....	74
<i>Table 17- Fusion LSTM structures with best performance metrics for different <math>N_{tw}</math> values</i> .....	75
<i>Table 18- Size and error statistics (mean, standard deviation) corresponding to all 5 experiments (in <math>\mu\text{m}</math>)</i> .....	77

<i>Table 19- Specifications of new realizations for evaluation of scalability .....</i>	84
<i>Table 20- Comparison of the performance of layouts of .....</i>	89
<i>Table 21- The nomenclature for SDP problem.....</i>	96
<i>Table 22- Number of damages and budget limits for each stage of the SDP problem.....</i>	99
<i>Table 23: Optimal layout for each segment for different utility weights (0.2, 0.4, 0.5, 0.7 and 0.9) and total budget limit of (13, 27, 37). .....</i>	103
<i>Table 24: LPOND for the case with budget limit =37.....</i>	104

## List of Figures

<i>Figure 1- Schematic for PHM problem for a multi-segment pipeline .....</i>	<i>12</i>
<i>Figure 2 - Key elements of the problem model .....</i>	<i>13</i>
<i>Figure 3 - Schematic of the constructed solution approach .....</i>	<i>17</i>
<i>Figure 4- Conceptual layout of the pipeline health monitoring problem .....</i>	<i>20</i>
<i>Figure 5 - Key elements of the framework for sensors and human inspection placement .....</i>	<i>20</i>
<i>Figure 6- Geometry of damages on the surface of a pipeline segment.....</i>	<i>22</i>
<i>Figure 7- Coverage and inference boundaries along an unrolled pipeline of length L and radius R. Nodes are located along the pipeline's bottom line (centerline of this figure) for easier illustration.....</i>	<i>23</i>
<i>Figure 8-Schematic of the proposed health monitoring approach .....</i>	<i>27</i>
<i>Figure 9- Example of type-size classes for depth (in mm) distribution of pitting corrosion damages .....</i>	<i>28</i>
<i>Figure 10- The 22<sup>nd</sup> realization out of 46 total realizations of the pipeline of Example 1 .....</i>	<i>38</i>
<i>Figure 11- Optimal health monitoring layout corresponding to the 22<sup>nd</sup> realization when (a) damage clustering and ME are not considered, (b) damage clustering and ME are considered.</i>	<i>42</i>
<i>Figure 12- Scatter plot obtained through aggregation of all 46 optimal layouts of Example 1 ..</i>	<i>45</i>
<i>Figure 13- Final health monitoring layout corresponding to the pipeline segment under study.</i>	<i>45</i>
<i>Figure 14- Longitudinal projection of final health monitoring layout and Pareto designs corresponding to most and least expensive layouts of realizations of example 1. Here, w refers to <math>w_{obj}</math> in Eq. 6.1 .....</i>	<i>46</i>
<i>Figure 15- Two newly generated realizations used as test realizations .....</i>	<i>46</i>

<i>Figure 16- Health monitoring layout corresponding to one realization of the longer pipeline segment .....</i>	<i>47</i>
<i>Figure 17- Schematic of the proposed data fusion and RUL estimation approach.....</i>	<i>51</i>
<i>Figure 18- Structure of a) LSTM network; b) LSTM cell .....</i>	<i>53</i>
<i>Figure 19- Aggregated LSTM-regression model for damage size estimation .....</i>	<i>55</i>
<i>Figure 20- Architecture of FCDN model for semantic image segmentation .....</i>	<i>58</i>
<i>Figure 21-Downsampling and upsampling units of the FCDN model for semantic image segmentation .....</i>	<i>60</i>
<i>Figure 22- a) A sampled microscopic crack image, b) corresponding annotated image. The resolution of both images is <math>480 \times 1024</math>. .....</i>	<i>62</i>
<i>Figure 23- Performance of bi-LSTM_AA: a) Predicted vs POF estimated (real) length of crack of Experiment 1 in mm. (The solid red line is the identity line based on POF estimations); b) RMSE distribution for a set of 300 test runs.....</i>	<i>71</i>
<i>Figure 24- Performance of data fusion and RUL estimation module: a) Predicted vs. POF estimated (real) length of crack of Experiment 1 in mm. (The solid red line is the identity line based on POF based real size values) b) RMSE distribution for 300 test samples.....</i>	<i>75</i>
<i>Figure 25- Predicted RUL vs real RUL values for Experiment 1 where real RUL values are calculated based on manual NIH ImageJ measurements and POF model. The red line is the real RUL identity line. ....</i>	<i>76</i>
<i>Figure 26- Scalability of the original and modified formulations for layout optimization .....</i>	<i>85</i>
<i>Figure 27- Schematic of layout update process.....</i>	<i>86</i>
<i>Figure 28- (a) Original optimal layout corresponding to 22nd realization, (b) Updated layout with AE sensors fixed at nodes [4, 5,8,10] .....</i>	<i>88</i>

*Figure 29- a) General schematic of a multi-stage problem, b) a sequential schematic of a multi-stage problem*..... 91

*Figure 30- Schematic of the SDP problem* ..... 97

*Figure 31- States and stages for the SDP problem for pipeline health monitoring* ..... 101

*Figure 32- Possible paths for state  $x_2 = 15$  and the optimal path* ..... 102



## **Chapter 1: Introduction, Problem Statement, and Research Overview**

Localized pipeline damages are caused by degradation processes such as corrosion and fatigue (Lewandowski, 2002; Okeniyi et al., 2014). These damages are usually distributed sparsely on the pipeline's interior or exterior surface and can lead to catastrophic failures in aging pipelines. These failures may involve human, environmental, and economic losses. As an example, localized corrosion is responsible for more than 80% of pipeline corrosion failures (Papavinasam, 2013). On the other hand, in the US alone, the annual cost of corrosion maintenance, repair, and replacement in transmission pipelines is estimated to be about \$125 billion (Koch, 2016). However, the costly health monitoring and repair downtime associated with these localized damages (Iqbal et al., 2017) can be managed using data-driven and effective Prognostics and Health Monitoring (PHM). A survey shows that more than 80% of field experts in rail and oil and gas industries of the US consider data analytics (including data-driven PHM) among the top three priorities of their industries (Kim et al., 2016). However, while data-driven PHM approaches have been applied in condition monitoring of structures such as bridges and railways (Wu and Jahanshahi, 2018; Guo et al., 2017), they have not been widely used for pipelines condition monitoring due to easy implantation of the currently popular approach, i.e., in-line inspection (Iqbal et al., 2017), and prolonged degradation rate of processes such as corrosion (Alaswad and Xiang, 2017). As such, the description of a data-driven and effective PHM approach for locally damaged (and aging) pipelines is the main subject of this dissertation.

One possible approach to the PHM of locally damaged pipelines involves collecting condition data of the interior and/or exterior pipeline surfaces using external data gathering schemes and then fusing the corresponding multi-source data to estimate the Remaining Useful Life (RUL). In such an approach, uncertainties associated with localized damage and

measurements, especially localized corrosion, should be considered (Zhang and Zhou, 2014). Additionally, geometric constraints of pipelines, and set up and implementation limitations from different data gathering schemes, necessitate having a flexible PHM approach so that its application is not limited to any particular pipeline or operational condition (Iqbal et al., 2017). Undoubtedly, minimizing health monitoring costs and maximizing confidence over the damage size and RUL estimations should also be considered in such a PHM approach.

In this dissertation, a PHM approach for locally damaged pipelines is presented. The presented approach determines optimal health monitoring layouts for segments of a locally damaged pipeline where a segment is defined as a relatively short section of the pipeline with a uniform damage behavior. An optimal layout includes different data gathering schemes (i.e., location and type of sensors and human inspection tools), which are used to gather, cost effectively, high confidence data on the state of the entire pipeline segment. Collected data are then fused together to obtain estimates of the online damage size and pipeline's RUL. These estimates can then be used to update the optimal health monitoring layouts corresponding to different pipeline segments. Moreover, while considering RUL estimates and an overall cost constraint, this dissertation presents approaches for finding optimal health monitoring layouts corresponding to a set of connected pipeline segments with different damage behavior and risk of failure.

This dissertation addresses several gaps in the literature on the PHM of pipelines. These gaps are discussed in Section 1.1. Then, in Section 1.2, a brief description of the overall problem considered is presented. Research objectives are defined in Section 1.3, considering the literature gaps. Finally, this chapter is concluded with a discussion of the research tasks in Section 1.4.

## 1.1 Literature Gaps

Several research gaps in the literature on PHM of pipelines, which are addressed in this dissertation, are outlined in this section.

### 1.1.1 Data Gathering Schemes and their Attributes

Online sensors network and offline human inspection are among the most common data gathering schemes (hereafter also called detection methods) related to degradation and damage, as reported in the literature for pipeline health monitoring (Iqbal et al., 2017). Each of these schemes has some strengths and weaknesses (Iqbal et al., 2017; Alaswad and Xiang, 2017). As an example, a network of sensors can provide frequent and inexpensive health monitoring data but have high uncertainty, bias, and noise<sup>1</sup>. On the other hand, human inspection-based data usually includes less uncertainty and less bias since human can control inspection tool imperfections, yet it is far less frequent and more expensive. Nonetheless, these two schemes are often considered separately in earlier optimization-based models (Zulkifli et al., 2018 ; Perumal, 2014; Zhang and Zhou, 2014; Alduraibi et al., 2016; Argyris et al., 2018). Furthermore, for either of the two schemes, only a few of the key attributes, such as data acquisition frequency, cost, location of sensors or human inspection, type of sensors, inspection tools, and measurement error are considered in each of the previous works (Alaswad and Xiang, 2017).

---

<sup>1</sup> The dimension of sensor measurement data in most cases is different from that of damage data. As an example, the sensor output can be a voltage signal for measuring damage size. Also, noises and biases may be involved in the sensor-based damage measurement data.

### 1.1.2 Probabilistic Detection Metrics

Several probabilistic damage detection metrics have been reported in the literature, which can be used for reliable and optimal pipeline PHM. Probability of Detection (POD) and probabilistic Measurement Error (ME) are two of these metrics (Berens, 1989; Liu et al., 2018; Chatterjee and Modarres, 2013; Zhang and Zhou, 2014). POD is defined as the probability of damage detected by a specific means of a detection device with or without input from a human inspector. POD, in general, depends on a number of factors such as the size and type of damage, type of the detection method, and the distance of damage to the detection device (e.g., sensor). On the other hand, ME is an uncertainty metric defined as the difference between the reported and true values of a measured quantity. As a result, ME for a damage size is the difference between the reported and actual size of the damage. Inference Probability (IP) is another detection metric which is defined as the probability of true prediction of the pipeline's state at a particular point, given the spatial correlation of localized damage and corresponding health monitoring evidence at other locations.

Despite several probabilistic detection metrics proposed in the literature, most of the reported works on pipeline health monitoring methods use deterministic detection models. As an example, a considerable number of previous studies only consider a binary damage detection model (Abdollahzadeh and Navimipour, 2016; Alaswad and Xiang, 2017). As a result, health monitoring layouts produced by the existing optimization approaches (Brunton et al., 2016; Guo

et al., 2017) cannot consider risk and severity of damage of different sizes and types. Additionally, while some works have considered spatial variability of localized damage along a pipeline (Abyani and Bahaari, 2020; Zhang and Zhou, 2014), to the best of the author's knowledge, IP has not been considered in the corresponding literature.

### **1.1.3 Uncertainty Consideration and Probabilistic Sampling of Localized Damages**

Most of the existing pipeline PHM approaches are based on deterministic information about location, type, and size of damage (Ostachowicz et al., 2019). However, the performance of the deterministic approaches is unsatisfying at the early stages of a pipeline design (Ostachowicz et al., 2019) when there are plenty of uncertainties. Computationally expensive finite element simulation methods are used to simulate damaged segments of a structure or pipeline for optimal health monitoring purposes (Hou et al., 2019). However, computational and cost limitations constrain gathering or simulation of damages. As a result, researchers have used computationally efficient methods to obtain limited size samples that represent the entire damaged structure (pipeline) with high confidence. Partial coverage inspection (Terpstra, 2009; Benstock and Cegla, 2016) is one of these methods. Nonetheless, the results of such methods might not be reliable for structures subject to stochastic degradation processes such as localized corrosion (Zhang and Zhou, 2014). The corresponding stochasticity, in terms of damage location, type, and size, can be considered using reliable resampling methods (e.g., the method by Wilks, 1946, and its more recent variations). However, computationally efficient and limited

size (re)sampling approaches for simulating stochastic damages with different sizes and types have not been considered in the literature on PHM of pipelines.

#### **1.1.4 Health Monitoring Optimization**

While different features, such as sensor lifetime or energy consumption (Alaswad and Xiang, 2017; Ostachowicz et al., 2019) are considered in the reported optimal sensor placement works, only a handful of the previous works have considered cost-effectiveness while maximizing the probability of detecting localized damages (Shafiee and Sørensen, 2017; Argyris et al., 2018) to maximize RUL of the pipeline, as it is common in PHM practices (Heidary et al., 2017). Additionally, simplifying assumptions, such as having a fixed number of the same-type sensors and a one-dimensional sensor network, are used in the corresponding literature in order to minimize the computational cost of multi-type sensor optimization models (Guo et al., 2017; Zulkifli et al., 2018 ; Hou et al., 2019; Ostachowicz et al., 2019). On the other hand, the literature on human inspection planning is more focused on optimizing inspection time rather than on inspection location and tool (Zhang and Zhou, 2014; Alaswad and Xiang, 2017).

#### **1.1.5 Data Fusion for Damage Size and RUL Estimation**

PHM of structures, including pipelines, requires an estimate of RUL (Khan and Yairi, 2018). Many approaches have been reported in the literature for online and offline estimation of damage size and resulting RUL. As an example, there is extensive literature on the application of homogenous data fusion (e.g., same-type multi-sensor data fusion) in the PHM of machinery

(Cao et al., 2016; Lei et al., 2017; Fang et al., 2017). Nonetheless, the literature on the application of heterogeneous data fusion (e.g., the fusion of different-type sensor data) for damage size and RUL estimation is very limited (Wu and Jahanshahi, 2018). Furthermore, fusion of heterogeneous data, consisting of the fusion of sensor data and inspection information, is not considered in the corresponding literature. In fact, sensor-based data gathering and human inspection information are mostly considered separately in the reported PHM approaches (Wu and Jahanshahi, 2018).

Assuming time-series data obtained from a sensor network and image-based inspection information, the next two sub-sections briefly discuss the literature on sensor-based and image-based damage size and RUL estimation.

#### **1.1.6 Damage Size and RUL Estimation Based on Sensors' Time-Series Data**

Time-series data from sensors has been studied for damage size and RUL estimation of degraded structures (Rabiei et al., 2016; Zheng et al., 2017; Lei et al., 2018). However, because of the volume and diversity of data, complex data analytic models may be needed in data analysis. For instance, Hidden Markov Models (HMMs) (Rabiei et al., 2018; Yu et al., 2019), Convolutional Neural Networks (CNNs) (Tian, 2012; Li et al., 2018; Cofre Martel, 2018), and Recurrent Neural Networks (RNNs) (Khan and Yairi, 2018; Lei et al., 2018) are among the popular analytics used for such estimations based on time-series data. Unfortunately, the excessive computational effort needed in HMMs, the tradeoffs between the time window size and overfitting in CNN-based models, and gradient vanishing or explosion in RNNs (Lipton et al., 2015), make these methods inappropriate for accurate RUL estimation in cases such as

nonmonotonic degradation processes such as in corrosion or fatigue crack considerations. In such processes, the behavior pattern is modulated and changes occur over different periods of time. Thus, short and immediate temporal correlations may not reveal degradation behaviors (Heidary et al., 2018). As a result, the consideration of longer temporal correlations can lead to a better size estimation for such processes compared to cases when only immediate temporal correlations are considered.

To manage the computational effort associated with longer temporal correlations, Long Short-Term Memory (LSTM) networks can be used (Zheng et al., 2017). These networks have shown promising results when applied to time-series data analysis with long, as opposed to immediate, temporal correlations (Zheng et al., 2017; Elsheikh et al., 2019; Zhao et al., 2019). However, LSTM networks have not been used for damage size and RUL estimation based on Acoustic Emission (AE) technology. More specifically, AE sensor data has not been used for PHM of locally damaged pipelines (which may have cracks).

In this context, AE signals are waves generated and transmitted in the medium of a structure if any deformation happens. As a result, AE is considered a passive and inexpensive Non-Destructive Testing (NDT) technology. AE, in contrast to wave-based NDT methods, do not need actuation (Gholizadeh et al., 2015). Moreover, in contrast to vibration signals, AE signals have an inherently high signal-to-noise ratio and are sensitive to microscopic-level events (Filippov et al., 2018; Cerrada et al., 2018). Hence, comparing to wave-based and vibration methods, AE signals can provide more accurate and less expensive early assessment of damage size. Nonetheless, only a few studies in the literature have investigated AE-based damage size and RUL estimation of structures (Loutas et al., 2017; Eleftheroglou et al., 2018). A hybrid method based on Bayesian analysis and Paris-Erdogan relation for crack-growth rate



has been used for the AE-based estimation of crack size (Rabiei and Modarres, 2013). However, only immediate temporal correlations are considered in the reported approach (Rabiei and Modarres, 2013). Moreover, such approaches will not work well for online damage size estimation due to the need for a computationally expensive Bayesian recursive analysis. Furthermore, none of the above-mentioned works have considered LSTMs and pipeline applications of AE sensors.

### **1.1.7 Damage Size and RUL Estimation Based on Inspection Images**

Automated image-based approaches have recently been used for damage detection and RUL estimation in structures. However, damage sizing is not considered in corresponding works. For example, CNN-based models for detection and diagnosis of various damages are introduced in several reported works (Modarres et al., 2018; Atha and Jahanshahi, 2018; Cofre Martel et al., 2018; Yu et al., 2019b). However, automated image processing (Smith et al., 2018) has not been used for online damage sizing and subsequent RUL estimation. This gap in the literature might be due to the structure of the standard CNNs, which can result in significant loss of resolution in the output due to successive pooling layers (Jegou et al., 2017). Hence, although standard CNN-based models are used successfully for damage detection, (i.e., per-image classification tasks), they are not a good fit for damage sizing (i.e., per-pixel classification tasks). Fortunately, this problem can be solved using semantic image segmentation (Jegou et al., 2017), where images are segmented for determination of the boundary of all objects. Fully Convolutional DenseNet (FCDN) (Jegou et al., 2017) is a class of CNN extensions recently introduced for semantic image segmentation. FCDNs are built upon fully convolutional networks (Long et al., 2015), DenseNets (Huang et al., 2016), and ResNets (He et al., 2015) to

achieve better performance in terms of pixel-wise classification accuracy and convergence time (Jegou et al., 2017).

### **1.1.8 Integrated and Dynamic Health Monitoring of Pipelines**

The majority of the reported optimization-based pipeline PHM approaches only consider a single and relatively small pipeline segment (with less than ten localized damages). They do so since consideration of large segments of a pipeline, with a large number of damages and design variables, in an “all-in-one” optimization problem, results in computational intractability (Alaswad & Xiang, 2017; Hou et al., 2019). Moreover, different pipeline segments may follow different degradation behaviors. As a result, considering all segments together makes simulation and statistical sampling quite difficult even if processes for non-uniform sampling are used (Valor et al., 2015). Furthermore, even for PHM of a single segment pipeline, only a few of the reported approaches consider expected damage behavior over a period of time to do static, but not dynamic and time-dependent, placement of sensors (Ostachowicz et al., 2019). For human inspection, on the other hand, while some reported approaches consider damage growth to tune inspection frequency accordingly, they do not consider the inspection area in their analysis for time-variant health monitoring. Thus, to the best of the author’s knowledge, no reported pipeline PHM approach has considered dynamic placement of detection methods (e.g., sensors and inspection tools) based on online health monitoring data for several segments of a pipeline (Alaswad & Xiang, 2017). Considering the benefits of the dynamic programming, once dealing with multi-stage problems (Bellman, 1954), Stochastic Dynamic Programming (SDP) (Bellman, 2003) can be used for integrated (i.e., multi-segment) health monitoring of pipelines. In other words, SDP can be used to find the optimal set of health monitoring layouts based on the

imperfect information on the failure risk associated with a system of pipeline segments any time new data becomes available.

Considering the present literature and the noted research gaps, a statement of the problem at hand, alongside corresponding research objectives, are provided in the next two subsections.

## 1.2 Problem Statement

Consider a pipeline with localized damages. The problem is to determine an optimized health monitoring layout for each pipeline segment (*Figure 1*) considering a health monitoring cost constraint for the entire pipeline. (Pipeline segments are considered separately to make sure that damage behavior is uniform at each segment. In other words, the longitudinal density of damages is almost uniform along a segment.)

The layout corresponding to each segment includes the location and type of all sensors and human inspection tools for obtaining data regarding the damaged state of the pipeline (*Figure 1*). The damage data gathered using each optimal layout is to be fused for online damage size estimation. The RUL curve for each pipeline segment of interest is then obtained based on damage size estimates, and the corresponding health monitoring layout is updated accordingly.

It is assumed here that the sensors are mounted on the exterior surface of the pipeline. Moreover, it is assumed that sensors transmit time-series data while human inspection tools provide images. The objective function of the layout optimization problem is formulated to be a combination of an average probabilistic detection metric and a health monitoring cost metric. Upon completion of the segment-level analysis, it is desirable to obtain an optimal set of layouts for a set of pipeline segments with an optimized average cost metric and an optimized average probabilistic detection metric, while an overall cost constraint is considered.

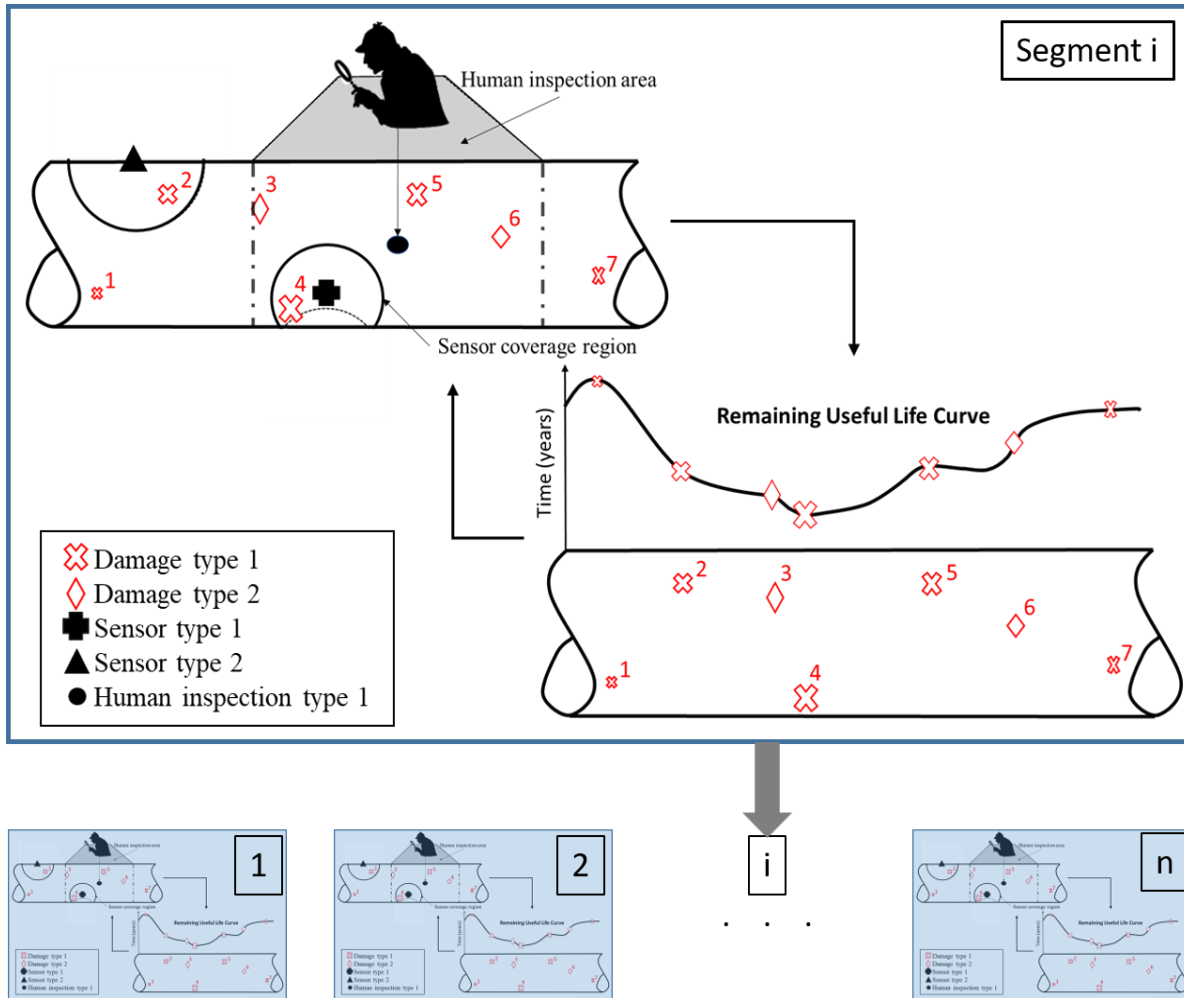


Figure 1- Schematic for PHM Problem for a Multi-segment Pipeline

The key elements of the framework for this problem are illustrated in *Figure 2*. For each pipeline segment, size distribution is assumed to be available for each damage type based on historical, or *a priori* gathered, health monitoring data. (A damage type refers to an underlying failure mechanism that results in damage. Damage refers to localized degradation throughout this dissertation.) It is assumed that a segment-wise pipeline vulnerability model is also known based on pipeline health monitoring history or up to date monitoring data/information. The vulnerability model consists of distributions that reveal a pointwise probability of having

damage of a certain type and size at an arbitrary location on the inner or outer surface of the pipeline.

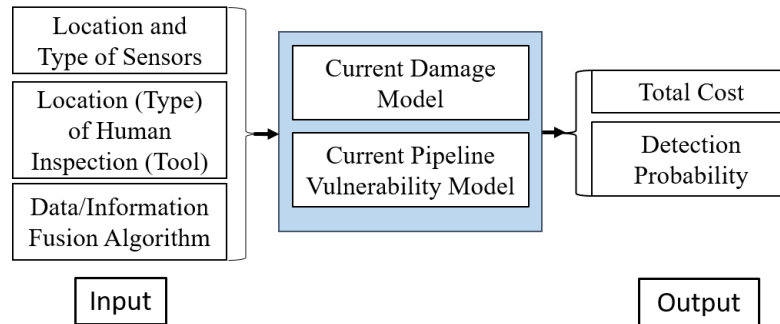


Figure 2 - Key Elements of the Problem Model

### 1.3 Research Objectives (ROs)

Considering the previously stated problem to develop a cost-effective and reliable PHM approach for locally damaged pipelines, the research objectives RO1 to RO4 listed below are determined:

**RO1-** Formulate an optimization problem for obtaining a two-dimensional layout of location and type of sensors and human-based inspection in order to maximize confidence over the assessment of pipeline health while a health monitoring cost metric is minimized.

**RO2-** Probabilistically assess and infer pipeline conditions under limited detection data in a cost-effective way.

**RO3-** Consider uncertain data and information from various sources and determine ways to fuse them together in order to obtain online damage size and RUL estimates for each pipeline segment of interest. Update health monitoring layouts corresponding to different pipeline segments considering RUL curves.

**RO4-** Find an optimal set of layouts for a set of pipeline segments considering different RUL prediction scenarios and a total health monitoring budget.

## 1.4 Research Tasks (RTs)

To achieve the above mentioned ROs, several research tasks are considered as in the following:

**RT1-** Formulate an optimization problem where a combination of detection methods (e.g., sensors network and human inspection) are modeled while multiple detection and measurement metrics are considered to model probabilistic detection of all damages with different sizes and types along a pipeline segment (RO1).

**RT2-** Consider a fewer number of assumptions and a larger number of detection methods' key attributes in the health monitoring layout optimization problem while attempting to maintain the computationally effort tractable (RO1).

**RT3-** Modify and/or redefine extant detection metrics to facilitate their use in the context of optimization. Additionally, define new probabilistic metrics to consider inference in the health monitoring optimization problem (RO2).

**RT4-** Use available statistical and limited size sampling methods to account for stochastic behavior of localized damages so that the final health monitoring layout of each pipeline segment at each assessment period is not dependent on any particular sample of localized damages (RO2).

**RT5-** Fuse data gathered through different detection methods to attain the RUL curve for each pipeline segment and update the health monitoring layout accordingly (RO3).

**RT6-** Use stochastic optimization to develop a model for finding the optimal set of health monitoring layouts for a set of pipeline segments considering different scenarios for RUL prediction and a total budget limit (RO4).

In short, in this dissertation, an overall approach is constructed for dynamic and integrated PHM of locally damaged pipelines. This approach is developed considering the noted research gaps (Section 1.1) and defined research tasks (Section 1.4). An overview of the approach is discussed in the next section. Following that, different levels and phases of the approach are detailed in Chapters 2 through 4.

### **1.5 Overview of the Proposed Approach**

The proposed approach for the dynamic PHM of locally damaged pipelines includes two main levels of analysis and computations. The first level is for a segment analysis (dashed blue box in *Figure 3*, called segment-level), while the second level is for integrated planning of health monitoring of pipelines (solid black box in *Figure 3*, called integrated-level). These two levels are briefly explained next.

The segment-level approach consists of two phases: a static phase and a dynamic phase (*Figure 3*). In the static phase (ROs 1 and 2, RTs 1 to 4), the intent is to obtain an optimal health-monitoring layout for a pipeline segment. Sensors network and human inspection, as popular detection methods per literature for health monitoring of locally damaged pipelines, are considered in the approach of the static segment-level phase.

Using the static segment-level approach, an optimization model is formulated that can be used to choose the type and location of sensors and human inspection tools along the pipeline. In the proposed formulation, several key attributes of these detection methods, such as a health monitoring cost metric, and three probabilistic detection metrics (namely, POD, IP, and ME to detect, infer, and size different damages), are considered. Statistical sampling methods are used

to simulate data based on limited data history and obtain the final health monitoring layout while stochasticity of localized damages is considered. This approach is detailed in Chapter 2. There, applications of the approach and corresponding results are given and discussed using two notional examples using synthetically generated damage data.

The second phase of the segment-level approach is called the “Dynamic Phase”. This phase includes data fusion for damage size and RUL estimation (RO3, RT5). The corresponding approach consists of three modules. In the first module, an LSTM-regression model is used to develop a sensor-based estimation of the damage size. In the second module, a CNN semantic image segmentation approach is used to build automated damage size estimations in which a pixel-wise classification is carried out on images of the damaged areas. Finally, in the third module, damage size estimates obtained by the first two modules are fused together for an online RUL estimation of the structure. This approach is detailed in Chapter 3. AE sensors data and microscopic images of a set of fatigue experiments are used there to demonstrate the performance of the proposed approach.

The second level of the constructed PHM approach (integrated-level) is for dynamic and integrated pipeline health monitoring. Once an updated RUL curve for each pipeline segment is obtained, the primary optimal health monitoring layouts will be updated, while fixing a preferred number of sensors and exploring different per segment cost constraints. As such, for each pipeline segment, a set of optimal health monitoring layouts corresponding to different values of cost constraints are obtained and fed into the integrated module of the proposed PHM approach (solid black box with white background in *Figure 3*). Next, an optimal set of layouts for all pipeline segments of interest are obtained using SDP. In the formulated SDP problem, health monitoring cost is considered as the state variable. Moreover, the average POD of all



damages over all segments of interest is considered in the objective function. In doing, an optimal set of health monitoring layouts for pipeline segments of interest is obtained as a function of time (RO 4, RT 6). Layout update process and integrated pipeline health monitoring are detailed in Chapter 4. Corresponding results are discussed there using synthetically generated data and considering three hypothetical pipeline segments.

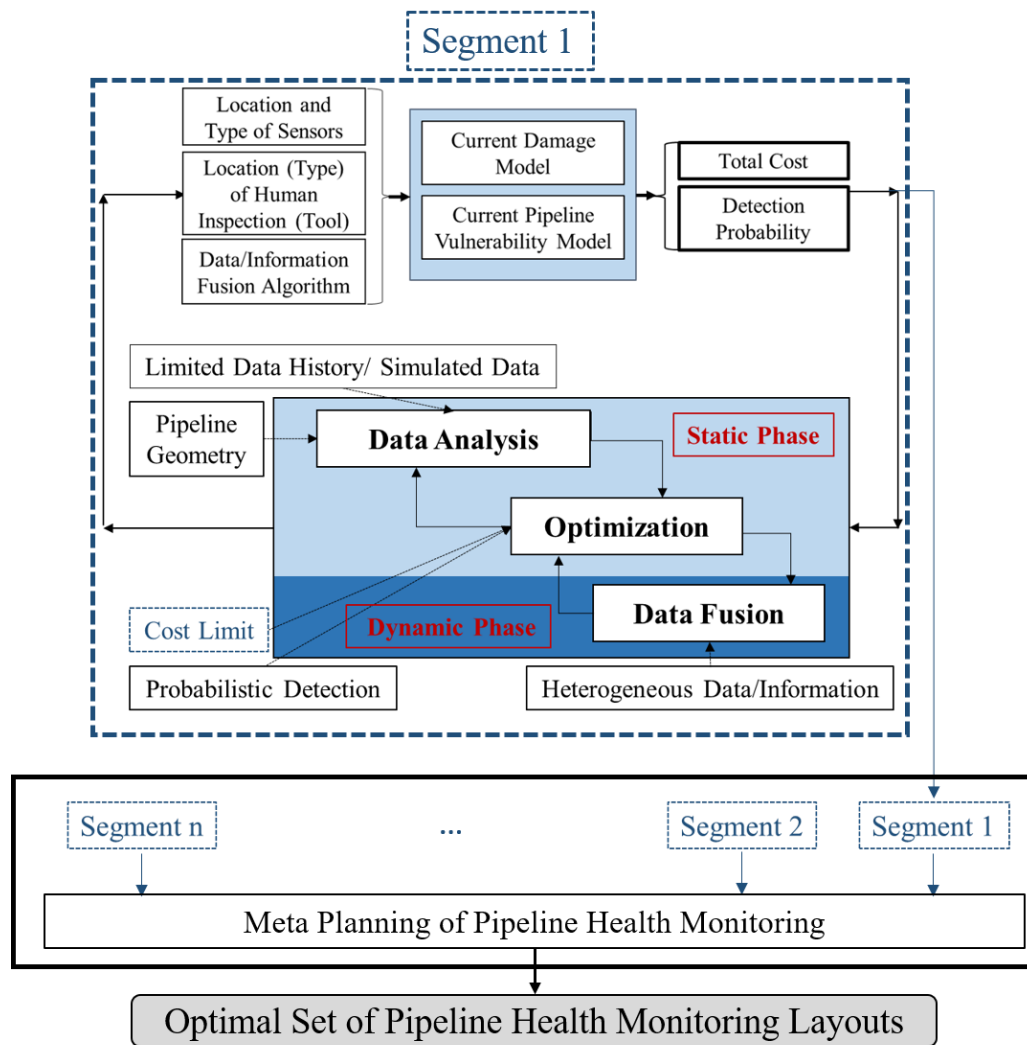


Figure 3 - Schematic of the Constructed Solution Approach

The remainder of this dissertation is organized as follows. Different level of details and phases of the constructed approach (*Figure 3*) are provided in Chapters 2 through 4. In those chapters where their application and performance are evaluated using demonstrative examples. Lastly, a summary of the dissertation, concluding remarks, delivered contributions, and some future research directions are provided in Chapter 5.

## **Chapter 2: Layout Optimization of Multi-type Sensors and Human Inspection Tools with Probabilistic Detection of Localized Damages for Pipelines<sup>2</sup>**

A new health monitoring approach for a pipeline segment is introduced in this chapter. This approach determines an optimized layout for multi-type sensors and human inspection areas/tools for a pipeline for which a probabilistic detection of damages is considered.

The remainder of this chapter is organized as follows. A statement for the problem is described in section 2.1. Three probabilistic detection metrics and their aggregation are detailed in section 2.2. The steps of the proposed approach for optimal placement of sensors and human inspection are presented in section 2.3 with the optimization problem formulated in section 2.4. Two examples with corresponding results are discussed in section 2.5. Finally, the chapter ends with some concluding remarks in section 2.6.

### **2.1 Optimal Health Monitoring Layout Problem Statement**

Consider a pipeline segment that is subjected to localized damages (*Figure 4*). The problem is to determine an optimized health monitoring layout for the pipeline segment. The layout determines the location and type of all sensors and human inspection tools for obtaining data regarding damage state of a pipeline (*Figure 4*). Here, it is assumed that sensors are mounted on the exterior surface of the pipeline. The objective function of the layout optimization problem is formulated to be a combination of an average probability of detection and a cost metric.

---

<sup>2</sup> This chapter is based on the following paper: Aria, A., Azarm, S., and Modarres, M., 2020. Layout Optimization of Multi-type Sensors and Human Inspection Tools with Probabilistic Detection of Localized Damages for Pipelines. *IEEE Access*, DOI: [10.1109/ACCESS.2020.2992671](https://doi.org/10.1109/ACCESS.2020.2992671)

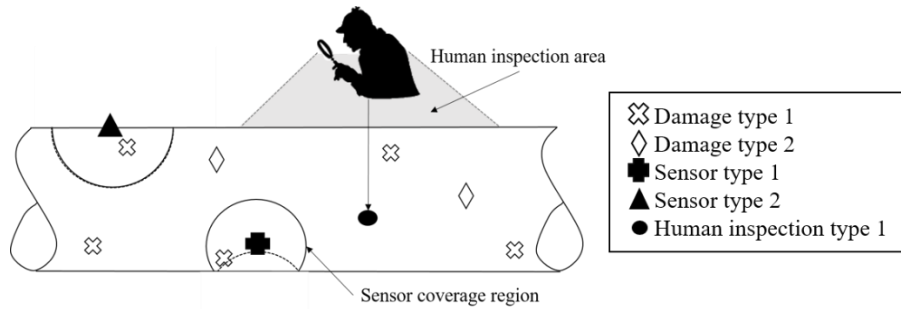


Figure 4- Conceptual layout of the pipeline health monitoring problem

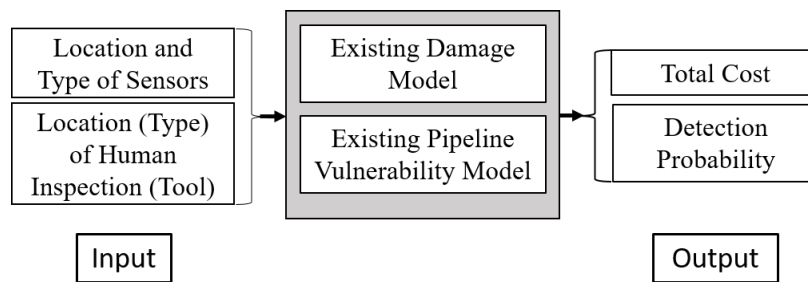


Figure 5 - Key elements of the framework for sensors and human inspection placement

The key elements of the framework for this problem are illustrated in *Figure 5*. (Refer to *Figure 2* for an explanation of the elements.)

A description of the probabilistic detection metrics, which are used in the optimization-based approach of this chapter, is followed.

## 2.2 Probabilistic Detection Metrics

Three detection metrics, i.e., POD, IP, and ME, are formulated and employed in the proposed approach of this chapter. These metrics are described in Sections 2.2.1 to 2.2.3. Note that sensors and human inspections are referred to as detection methods throughout this chapter where human inspection is modeled as a high coverage and low frequency data generating sensor. Detection methods are assigned to candidate locations for detecting damages which are

defined as nodes of a graph in this chapter. A graph is defined considering a particular “sample” of localized damages. A sample is a particular realization of the pipeline with damages. It is assumed that there is a single node in the vicinity of each damage of a pipeline realization (sample).

### 2.2.1 Probability of Detection (POD)

For a damage of particular type and size to be detected by a detection tool (e.g., acoustic emission (AE) sensor), POD depends on factors such as size and type of a damage and distance between the damage and a corresponding detection method. These dependencies are typically investigated separately in the literature (Zhang and Zhou, 2014; Sause et al., 2018). In this chapter, a new tabularized and aggregate model of POD as a function of damage size and damage-to-sensor distance is developed (see details in section 2.3, Step 6).

The use of this new aggregate POD model is facilitated through a binary (hit or miss) detection model, namely  $P_{\text{exist}}$ . A damage is considered detectable in the proposed  $P_{\text{exist}}$  model regardless of its type and size, if it is located in the coverage range of the detection method. The coverage range of a detection method is defined as the distance beyond which the marginal POD of a damage by that particular detection method is zero.

As shown in *Figure 6* for an unrolled pipeline segment, the longitudinal and (unrolled) circumferential coordinates are used in the  $P_{\text{exist}}$  model (and throughout this chapter) as coordinates of the location of damages on the surface of the pipeline. To account for the circularity of the circumferential coordinate (y-axis in *Figure 6*), a replica of each damage is considered with the same longitudinal component as that of the original damage. The replica’s circumferential component is:  $(y-2\pi R)$ , where  $y$  denotes the circumferential component of the

original damage location and  $R$  is the radius of the pipeline. The origin of the  $x$ - $y$  coordinate system is located at the leftmost point of the bottom line of the pipeline. The location vector of both the damage and its replica are inserted into the  $P_{\text{exist}}$  model (Eq. 1) to determine whether a damage ( $d_i$ ) is in the coverage range ( $cr_j$ ) of the detection method at a node  $j$  ( $dm_j$ ). To do this, consider a damage  $k$  denoted by  $d_k$  (located at:  $\mathbf{a}_k = (x_k, y_k)$ ) and damage  $i$ ,  $d_i$  (located at:  $\mathbf{a}_i = (x_i, y_i)$ ), as shown in *Figure 6*. A replica of  $d_i$  (located at:  $\mathbf{a}_i' = (x_i, y_i - 2\pi R)$ ) and  $d_k$  itself are in the coverage range of the detection method at node  $j$  (located at:  $\mathbf{b}_j = (x_j, y_j)$ ). Hence, both damages are detectable by the detection method at node  $j$ ,  $dm_j$ .

$$P_{\text{exist}}(d_i, dm_j) = \eta_{i,j} = \begin{cases} 1 & \text{if } \|\mathbf{b}_j - \mathbf{a}_i\| \leq cr_j \\ & \text{or } \|\mathbf{b}_j - \mathbf{a}_i'\| \leq cr_j \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

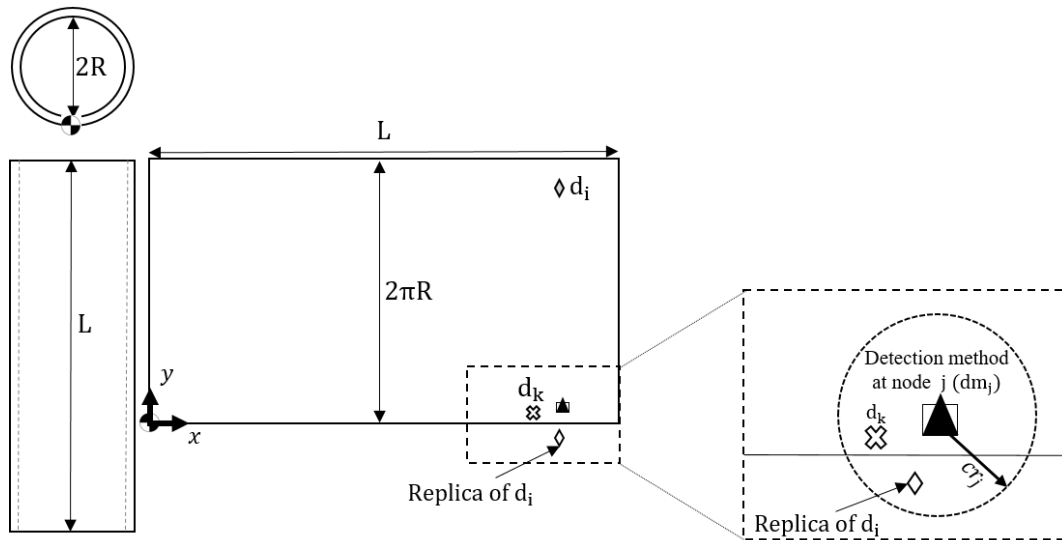


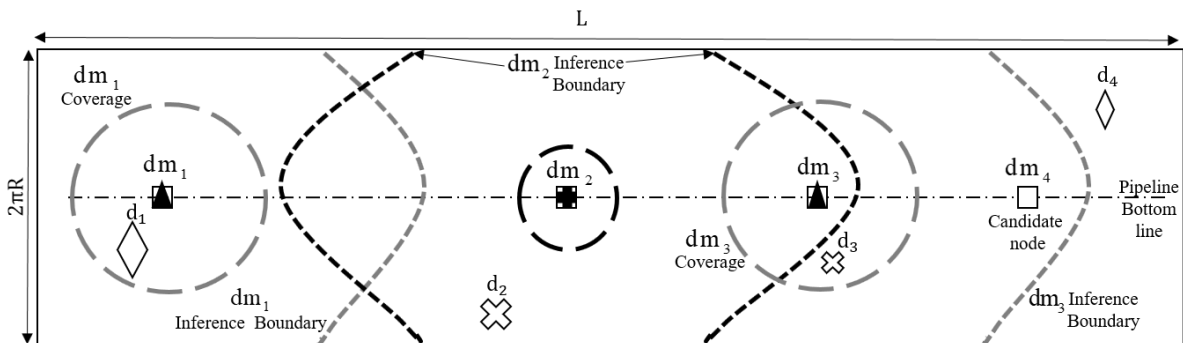
Figure 6- Geometry of damages on the surface of a pipeline segment

### 2.2.2 Inference Probability (IP)

IP is a metric that accounts for the detectability of a damage of a particular size and type at a particular point of the pipeline given the health monitoring data at some other point of the

pipeline. To better illustrate IP, the differences between IP and POD in the proposed approach are illustrated in *Figure 7*. As shown, a pipeline segment with two damage types (hollow diamond and hollow multiplication symbols) and detection methods (solid triangle and solid plus symbols) are considered. A hollow square node is located in the vicinity of each of the damages and detection methods are assigned to these nodes. For example, a detection method of type triangle is assigned to node 3 while a detection method type zero is assigned to node 4 (The assignment of no detection method to a node is considered as the detection method type zero.)

Each detection method has a specific coverage range, cost, precision (measurement error), data acquisition frequency, and inference capability. In *Figure 7*, coverage boundaries are shown with dashed circles while inference boundaries are shown with dashed parabolas. Note that the second inference boundary of the detection method at node 1 ( $dm_1$ ) and at node 3 ( $dm_3$ ) are not shown in *Figure 7* to avoid clutter. Moreover, for human inspection coverage range is equal to half of the inspection length along the pipeline since inspection tool coverage range is not of importance as long as it is assumed that the tool is taken to the exact location of all damages in the inspection area.



*Figure 7- Coverage and inference boundaries along an unrolled pipeline segment of length  $L$  and radius  $R$ . Nodes are located along the pipeline's bottom line (centerline of this figure) for easier illustration.*

For each node  $j$ , inference boundaries are the locus of zero IP and an inference distance limit is defined as the minimum distance of zero IP locus from the corresponding node. IP is a function of the damage-to-node distance, detection method type, damage size and type, and similarity of the degradation behavior at the damage and node locations. Hence,  $IP(d_i, dm_j)$  is zero at inference boundaries of node  $j$  while  $IP(d_i, dm_j)$  is unity if  $d_i$  is located at node  $j$ . A damage will be missed if POD and IP values corresponding to all neighboring detection methods are zero at the damage location. To minimize probability of missing a true damage, detection methods should be placed in a way that minimize Probability of Not Detection (POND), accordingly:

$$\Omega_i = (1 - \Theta_i) \times (1 - \Phi_i) = \left[ \prod_{j=1}^N (\bar{\theta}_{i,j}) \right] \times (1 - \Phi_i) \quad (2)$$

where  $\Omega_i$  is POND of damage  $i$ ,  $\Theta_i$  is the aggregate IP of damage  $i$  (resultant IP of all detection methods assigned to the nodes in the vicinity of damage  $i$ ), and  $\Phi_i$  is the aggregate POD of damage  $i$  (resultant POD of all detection methods with damage  $i$  in their coverage range).  $N$  is the number of nodes over the pipeline surface of interest. Moreover,  $\bar{\theta}_{i,j}$  is an element of  $\bar{\Theta}$  matrix, which represents the probability of missing  $d_i$  through inference based on health monitoring data gathered by  $dm_j$ . Note that in Eq. 2 (and elsewhere in the proposed approach), it is assumed that the probability of a false positive detection is zero.

To simplify the model, a log-linear version of POND, LPOND, is used (Eq. 3) where  $\Omega'_i$  is LPOND of damage  $i$ .

$$\Omega'_i = \left[ \sum_{j=1}^N \log(\bar{\theta}_{i,j}) \right] + \log(1 - \Phi_i) \quad (3)$$



Application of a tabularized LPOND based on Eq. 3 is discussed in section 2.3 (Step 6).

### **2.2.3 Measurement Error (ME)**

ME is an uncertainty metric defined as the difference between reported and true values of a quantity. In this chapter, the probabilistic ME, simply referred to as ME, is defined as the probability of the reported size value of a damage of known type not being within the interval  $[(\text{True size} - \epsilon_l), (\text{True size} + \epsilon_u)]$ . The quantities  $\epsilon_l$  and  $\epsilon_u$  are for pre-specified acceptable lower and upper error margins, specified by an expert, and “True size” refers to the actual size of the damage. Note that only tool measurement errors but not the human error are considered in the calculation of ME values. Nonetheless, human error can be easily included (see e.g., Knop et al., 2019).

The LPOND model, as an aggregate of POD and IP models, and probabilistic ME are considered in a step-by-step approach for pipeline health monitoring, as discussed next.

## **2.3 Proposed Approach for Optimal Placement of Sensors and Human Inspection**

The proposed approach simulates and randomly places localized damages on the pipeline surface and repeats the process over many realizations, each with the consideration of degradation and maintenance history of similar pipelines. Using the approach, a pipeline is divided into multiple segments, each having a uniform longitudinal and non-uniform circumferential density of damages. For each realization, the location and size of the localized damages are randomly simulated and placed over each pipeline segment (see e.g., Chatterjee and Modarres, 2013; Barton et al., 2017). However, for non-uniform longitudinal damage density, one can use a method like the one by Valor et al. (2015). Moreover, clustering is

conducted on all realizations (or samples) of spatially distributed damages to smooth the optimization work and improve the choice and placement of detection methods. Additionally, to account for stochasticity of localized damages with a limited computational expense, the proposed approach uses a modified version of a well-known probabilistic sampling method, i.e., Wilks method (Wilks, 1941). As such, the proposed approach is also a good fit to be used at early stages of a pipeline design.

*Figure 8* shows an overview of the various steps of the approach, as shown with the number to the right side of each block. These steps are detailed next.

**Step 1 - Obtain Modified Vulnerability Distributions (MVDs):** Collect prior maintenance and degradation data of a similar pipeline. Map the available data over the pipeline surface and perform data analysis, including data simulation (augmentation) for model selection and fitting. Subsequently, obtain three distributions for each localized damage type: longitudinal, circumferential spatial, and size distributions. Assume the three distributions are independent. Modify the distributions to have a higher intensity for larger size damages in the pipeline areas with a higher risk of failure.

**Step 2 - Determine Segments and Mesh Cell Size:** Based on longitudinal MVD for each damage type, divide the pipeline into segments with an (approximately) uniform longitudinal density of damages. Mesh each pipeline segment into circumferential strips (as mesh cells) with their width laid out along the longitudinal direction (x axis) of the pipeline (recall *Figure 6*). Considering constant longitudinal density of damages and using Poisson distribution, determine the width of each strip so that the probability of having more than a damage of each type in each strip is negligible.

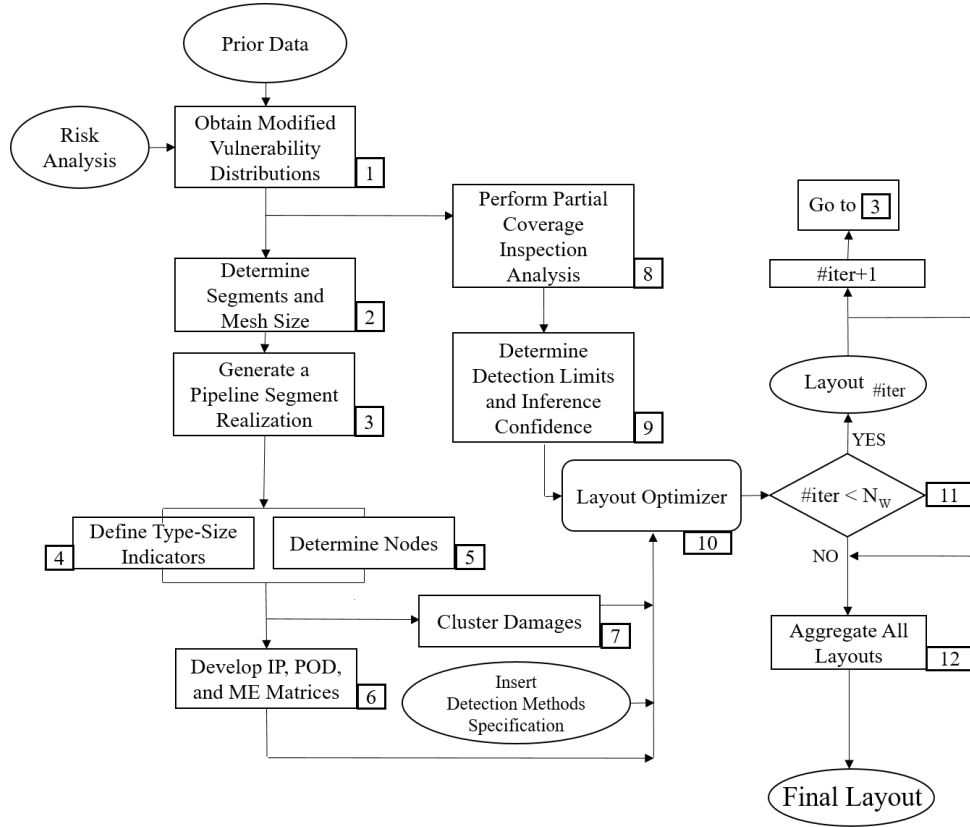
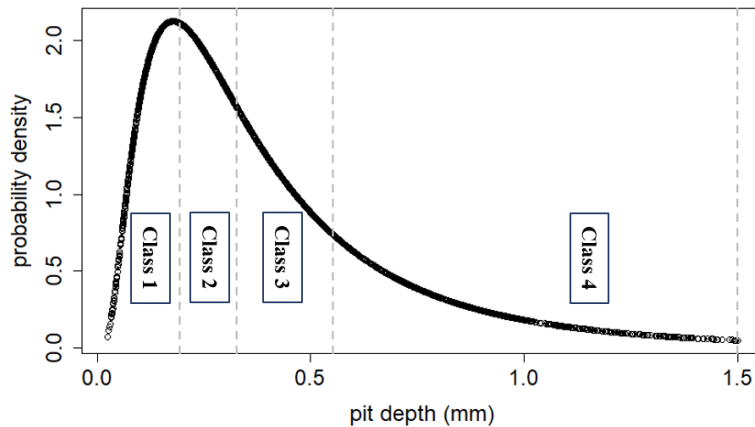


Figure 8-Schematic of the proposed health monitoring approach

**Step 3 – Generate a Pipeline Segment Realization:** Assuming constant longitudinal density of damages along a pipeline segment, generate the longitudinal location of randomly generated localized damages of each type using the Poisson and binomial distributions. Determine segment realizations involving placement of damages of different types on the strips. Use the Poisson distribution to calculate the probability of having a single damage of an arbitrary type in each strip. Use this value in a Bernoulli test as a success probability. If the simulated outcome of a Bernoulli test is one, place a damage in the strip and determine its circumferential location based on the circumferential MVD.

**Step 4 - Define Type-Size Indicators:** Once a segment realization is simulated, determine a type-size class vector for each of the damages of that realization. For each damage type, define type-size classes which refer to intervals of the corresponding size MVD with a same cumulative probability. Use a Monte Carlo simulation to assign one of these classes to each of the simulated damages of the realization at hand. As an example, consider *Figure 9* that presents a size MVD for a pitting corrosion damage (Ossai et al., 2015). For this size MVD, the number of classes is set to four. If a (randomly generated) class of damage  $i$  ( $d_i$ ) is equal to 3,  $d_i$  will be a pit damage with a depth in the interval 0.33-0.55 mm and its binary class vector will be  $C_i = [0,0,1,0]$ .

In the case of multiple damage types (e.g., pitting corrosion and stress corrosion cracking damages), extend the above-explained technique so that classes 1 to 4, for example, represent pitting corrosion size MVD while classes 5 to 8 represent stress corrosion cracking.



*Figure 9- Example of type-size classes for depth (in mm) distribution of pitting corrosion damages*

**Step 5 - Determine Nodes:** Place a single node near each damage of a segment realization. The longitudinal location of the node is the same as that of the corresponding damage. In contrast,

its circumferential location is randomly generated to make it possible for the optimization solver to distinguish detection methods with different coverage and detection capabilities.

**Step 6 - Develop IP, POD, and ME Matrices:** Follow the instructions below and develop matrices to be used for modeling probabilistic detection in an optimization context.

For IP, develop a  $\log(\overline{\Theta}^{k,m})$  matrix for each combination of damage type-size classes (k) and detection methods type (m). These matrices will be used in the optimization formulation to calculate the resulting  $\log(\overline{\Theta})$  matrix of LPOND of each damage (Eq. 3). On the other hand, to calculate the POD term of Eq. 3 in a linear optimization context, develop  $\Lambda_i$  as the POD matrix of damage i with element  $\lambda_i^{m,j}$ . Use Eq. 4 to calculate  $\lambda_i^{m,j}$  that denotes POD for damage i of a *priori* known class k which will be recognized by detection method type m (if used) at node j. Assume uncorrelated POD dependency on damage size ( $\lambda_{s_i^m}$ ) and damage-to-node distance ( $\lambda_{d_i^{m,j}}$ ).

$$\lambda_i^{m,j} = \lambda_{s_i^m} \times \lambda_{d_i^{m,j}} \quad (4)$$

Lastly, calculate tabularized ME values to be used in the optimization problem. At first develop a matrix  $\Delta$  with its element  $\delta_{k,m}$  defined to be the ME value corresponding to the detection of a damage of class k by detection method type m. Next, considering particular placement of damages in the realization at hand, use Eq. 5 to calculate the mean ME ( $\delta_j^m$ ) corresponding to detection of all damages i in the coverage range of detection method m, if it is used at node j. The number of corresponding damages of interest is denoted by  $N(i,j)$  and  $c_{i_k}$  is the element of type-size class vector for damage i.

$$\delta_j^m = \frac{\sum_{i,k} \delta_{k,m} \times c_{i_k}}{N(i,j)} \quad (5)$$

**Step 7 - Cluster Damages:** Cluster damages of the realization at hand to smooth the optimization work. To do that, define a Minimal Spanning Tree (MST) (Kruskal, 1956) with damages at its nodes. Calculate clustering distance limit considering the inference distance limit (section 2.2.2) and length of the longest edges of the corresponding MST. Then, determine the number of clusters accordingly. Define a similarity metric based on the model in Eq. 1 and Eq. 6.10. Use the constrained k-means (Wagstaff et al., 2001) algorithm to cluster damages where the defined similarity metric is considered and damages at a distance greater than clustering distance limit are not assigned to a same cluster.

**Steps 8 and 9 - Perform Partial Coverage Inspection Analysis and Determine Detection**

**Limits and Inference Confidence:** Use partial coverage inspection analysis (Benstock and Cegla, 2016) to estimate the state of the entire pipeline segment under study with a minimal monitoring coverage (i.e., minimum health monitoring cost) and a certain confidence level. In the case of locally damaged pipelines, use partial coverage inspection to determine the damage detection percentage required for having the worst-case damage of the entire segment under investigation being detected with p% confidence. Define an overall detection constraint as well as a lower limit on detection percentage for each damage cluster considering the required detection percentage.

**Step 10 – Perform Health Monitoring Layout Optimization:** Feed the results of pre-processing stage (i.e., Steps 1 through 9) as well as available detection methods and their specifications into the optimization model. (This step and the corresponding formulated problem are detailed in section 2.4)

**Steps 11 – Implement Probabilistic Sampling:** Use a modified version of Wilks method (Pourgol-Mohammad, 2007) to determine the minimum number of realizations ( $N_w$ ) of a stochastic phenomenon (e.g., localized corrosion) needed to capture at least  $p_1\%$  of variations (e.g., spatial and size variations of localized damages) with  $p_2\%$  confidence. Repeat steps 1 to 10 of the proposed approach  $N_w$  times and obtain  $N_w$  optimal health monitoring layouts.

**Steps 12 – Aggregate All Layouts:** Aggregate all  $N_w$  layouts to form an aggregated layout where  $p_1\%$  of variations are considered with  $p_2\%$  confidence. Use K-means clustering to find clusters of detection methods in the aggregated layout. Determine number of clusters considering the value of  $N_w$  and total number of detection methods in the aggregated layout. Then, place the detection methods at the center of corresponding clusters to form the final layout.

## 2.4 Health Monitoring Layout Optimization Problem Formulation

The health monitoring layout optimization problem is formulated as a mixed integer nonlinear programming problem where the objective function (Eq. 6.1) is formed by a weighted sum of an average utility and an average LPOND function. The utility function is formulated based on key features of available detection methods. These key features include data gathering cost, coverage range, and an information metric that is defined as a linear combination of data acquisition frequency, ME, and information value of the gathered data. The utility is maximized in order to maximize the information value and data acquisition frequency while the cost of the health monitoring layout and ME are minimized. Moreover, an average LPOND is maximized

to, equivalently, maximize the geometric mean of POND of all damages in the realization of the pipeline segment under study.

The variables being optimized in the optimization problem are  $dm_j$  (Eq. 6.19), i.e., an integer indicator for the type of detection method used at each node  $j$ . All other variables (i.e., binary detection indicators and detection method variables in Eqs. 6.10-12) are either determined in the pre-processing stage of the approach or are a function of  $dm_j$ . Furthermore, the optimization problem has several constraints, including (1) detection limits (Eq. 6.14-15): these constraints are determined in the pre-processing stage using partial coverage inspection analysis. (2) Overall inference limit (Eq. 6.21): this constraint is also determined in the pre-processing stage. (3) Cost limit (Eq. 6.16): the expected cost of detection methods along a segment cannot be higher than a pre-specified limit.

In the proposed formulation, the detection method type vector (Eqs. 6.12-13) is used to show the type of detection method used at each node. Detection methods can refer to the type of sensors or human inspection tool used. A detection method type zero indicates that no detection method is assigned to a corresponding node. Moreover, the utility of detection methods' attributes is given as an input for each detection method (Eq 6.5).

The formulation of the optimization problem is shown by Eqs. 6.1-6.21, with *Table 1* providing a definition for all symbols in the problem. Note that bold letters in *Table 1* and formulated problem represent vectors or matrices while regular letters represent scalars.



Table 1- Symbols used in the optimization model (Eqs. 6.1 – 6.21).

Symbol	Definition
$b_l$	Lower bound of overall detection percentage
$b_u$	Upper bound of overall detection percentage
$\mathbf{C}_i$	Damage type-size vector with element $c_{i,k}$
$co_u$	Upper bound of expected cost of detection
$\mathbf{CO}$	Detection methods cost vector (its $m+1^{\text{th}}$ element is cost of detection method $m$ )
$\mathbf{CR}$	Detection methods coverage vector (its $m+1^{\text{th}}$ element is coverage radius of detection method $m$ )
$dc_{ic}$	Minimum detection for damages in cluster $ic$
$dm_j$	Integer indicator of detection method used at node $j$
$E(.)$	Expected value of $(.)$
$fr_m$	Data acquisition frequency of detection method type $m$
$G_L$	Lower bound of LPOND
$G_U$	Upper bound of LPOND
$I$	Damage number superscript
$im_j$	Information metric of detection method used at node $j$
$iv_m$	Information value of detection method type $m$
$\mathbf{ID}_j$	Detection method type vector corresponding to node $j$
$J$	Node number superscript
$m$	Type indicator for detection method used at each node
$M$	Number of available detection methods
$N$	Number of damages/nodes in the segment realization at hand
$N_{cl}$	Number of clusters of damages in the segment realization at hand
$Q$	Number of damage type-size classes
$RE_{max}$	Upper limit of detection redundancy for each damage
$RE_u$	Upper limit of expected detection redundancy
$t$	Vector superscript for transpose of a vector
$U(.)$	Input utility for $(.)$
$U_j$	Utility value of the detection method used at node $j$
$U_{tot}$	Total utility value of a layout
$(x,y)$	Location coordinates of a damage or node (see Section 2.2.1)
$y'$	Circumferential coordinate of the replica of a damage (see section 2.2.1)
$w_{(.)}$	Weighting factor of $(.)$
$\delta_j^m$	Mean ME corresponding to detection method $m$ used at node $j$ (see Eq. 5)
$\eta_{ij}$	$P_{exist}$ binary indicator (Eqs. 1 and 6.10)
$\overline{\theta}_{ij}^{k,m}$	Probability of missing damage $i$ of type $k$ through inference by detection method $m$ at node $j$ (Eq. 6.2)
$\Lambda_i^J$	Column $j$ of POD matrix of damage $i$ (Eq. 6.2)
$\pi_i$	Detection indicator of damage $i$ (Eq. 6.11)
$\Omega'_i$	LPOND of damage $i$ (Eq. 6.2)

$$\text{Minimize}_{dm_j} \left( w_{obj} \frac{E(\Omega') - G_L}{G_U - G_L} + (1 - w_{obj}) \frac{-U_{tot}}{b_u \times N} \right) \quad (6.1)$$

where:

$$\Omega'_i = \sum_{m=0}^M \sum_{j=1}^N \sum_{k=1}^Q \{c_{i,k} \times id_{j,m} \times \log(\overline{\theta_{i,j}^{k,m}})\} + \sum_{j=1}^N \log(1 - \eta_{i,j} \times \Lambda^j \times \mathbf{ID}_j^t) \quad (6.2)$$

$$\frac{\sum_{i=1}^N \Omega'_i}{N} = E(\Omega') \quad (6.3)$$

$$U(im_j) = \sum_{m=0}^M [id_{j,m} \times (w_{fr} \times U(fr_m) + w_{iv} \times U(iv_m) + w_{ME} \times U(\delta_j^m))] \quad (6.4)$$

$$U_j = w_{co} \times U(\mathbf{ID}_j \cdot \mathbf{CO}^t) + w_{cr} \times U(\mathbf{ID}_j \cdot \mathbf{CR}^t) + U(im_j) \quad (6.5)$$

$$U_{tot} = \sum_{j=1}^N (U_j \times (1 - id_{j,m=0})) \quad (6.6)$$

$$0 \leq w_{obj} \leq 1 \quad (6.7)$$

$$w_{cost} + w_{cov} + w_{Fr} + w_{IV} + w_{ME} = 1 \quad (6.8)$$

$$w_{cost}, w_{cov}, w_{Fr}, w_{IV}, w_{ME} \geq 0 \quad (6.9)$$

Subject to:

$$\text{if } [(x_i - x_j)^2 + \min(|y_i - y_j|, |y'_i - |2\pi R - y_j||)]^2 \leq \mathbf{ID}_j \cdot \mathbf{CR}] \eta_{i,j} = 1, \text{ otherwise } \eta_{i,j} = 0 \quad (6.10)$$

$$\text{if } [\sum_{j=1}^N \eta_{i,j} \geq 1] \pi_i = 1, \text{ otherwise } \pi_i = 0 \quad (6.11)$$

$$\text{if } [dm_j = m] id_{j,m} = 1, \text{ otherwise } id_{j,m} = 0 \quad (6.12)$$

$$\sum_{m=0}^M id_{j,m} = 1 \quad (6.13)$$

$$b_l \times N \leq \sum_{i=1}^N \pi_i \leq b_u \times N \quad (6.14)$$

$$\sum_i \pi_i \geq dc_{ic}, i \in \text{cluster \#ic} \quad (6.15)$$

$$E(\text{cost}_j) \leq co_u \quad (6.16)$$

$$\sum_{j=1}^N \eta_{i,j} \leq Re_{max} \quad (6.17)$$

$$E(\sum_{j=1}^N \eta_{i,j}) \leq Re_U \quad (6.18)$$

$$0 \leq dm_j \leq M \quad (6.19)$$

$$1 \leq ic \leq N_{cl} \quad (6.20)$$

$$G_L \leq \Omega'_i \leq G_U \quad (6.21)$$

In the problem formulation, Eq. 6.2 represents LPOND of damage  $i$  (see Section 2.2). Eq. 6.4 defines information metric corresponding to data on all the damages detected by the detection method used at node  $j$ . Moreover, Eq. 6.5 defines the utility value corresponding to the detection method used at each node. On the other hand, Eq. 6.10-6.21 represent constraints of the formulated problem where overall and per cluster detection limits (Eqs. 6.14-15) and LPOND allowable range (Eq. 6.21) should be determined through partial coverage inspection analysis.

The above-formulated optimization problem is modified, using linear and integer programming techniques (Crowder et al., 1983), so that the corresponding problem can be solved using a mixed integer non-linear programming (Bussieck and Pruessner, 2003) solver.

## 2.5 Examples

Two notional examples are solved in this section using the proposed approach. In both examples, only localized damages resulting from internal pitting corrosion but not any other failure mechanism are considered. Moreover, pit depth is considered as damage size while pit length is neglected following the work in (Shabarchin and Tesfamariam, 2016). For both examples, it is assumed that the pipeline segment under consideration is used for transporting crude oil. Except for the assumed pipeline radius ( $R = 1$  unit of length, i.e., 1 meter), other pipeline specifications, and their effect on the performance of detection methods, are not considered. Furthermore, due to lack of access to real world data synthetic internal pitting corrosion data are generated considering values, models, and relations reported in the literature

for pitting corrosion or similar degradation processes (e.g., Sause et al., 2018; Rabiei and Modarres, 2013; and Heidary et al., 2018). Generated data are then used as the input in both examples. Stochasticity of the localized damages is also considered through probabilistic sampling.

In the first example, section 2.5.1, a problem corresponding to a short pipeline segment is solved to illustrate the proposed approach step-by-step. The corresponding results are used to evaluate the performance of the proposed approach. The final health monitoring layout is obtained for this example through aggregation of layouts corresponding to different realizations of pitting corrosion damages over the pipeline segment. For this example, in the layout optimization formulation, there are 142 continuous variables representing utility of features of the detection method at each node and 198 binary variables representing indicators for probabilistic detection, inference, and the types of detection method at each node. Moreover, there are 345 constraints. For the second example, a longer pipeline segment is considered in section 2.5.2), where only results corresponding to a single realization of damages are presented. This example is presented to show the applicability of the proposed approach to a larger sized problem for which there are 490 continuous variables, 670 binary variables, and 11,065 constraints in the layout optimization problem.

### **2.5.1 Example 1**

In this example, a pipeline segment with 50-meter length ( $L$ ) and 1-meter radius ( $R$ ) is considered. It is assumed that there are only two detection options: an AE sensor (detection method type 1 ( $m=1$ )) and human inspection with an ultrasonic tool (detection method type 2

( $m=2$ ). An assignment of no detection method to the nodes is the other option ( $m=0$ ). *Table 2* shows assumed specifications of the detection methods.

*Table 2- Specifications of detection methods*

Attribute	AE sensor	Human Inspection with ultrasonic tool
Coverage radius (meter)	0.400	20.0
Cost	1.000	10.0
Utility (cost)	1.000	0.10
Utility (coverage)	0.013	1.00
Utility (frequency)	1.000	0.01
Utility (information value)	0.020	1.00

### 2.5.1.1 Step-by-Step Application of the Proposed Approach

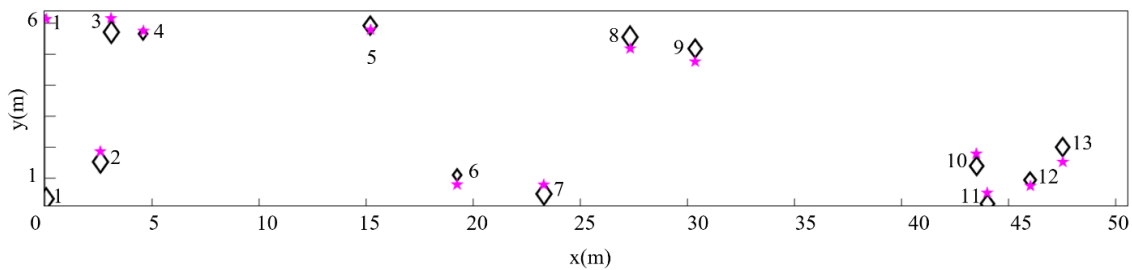
Considering the modified Wilkes method by Pourgol-Mohammad (2007), forty-six realizations ( $N_w$  in Step 11 of *Figure 8*) of the corresponding pipeline segment are generated to guarantee capturing 95% of spatial and size variations of localized damages with 90% two-sided confidence. There are 5 to 15 damages in each of these realizations. The 22<sup>nd</sup> realization has a relatively dense placement of damages (with 13 damages) that is good for illustration purposes. Hence, a brief discussion of the steps in the proposed approach (as shown in *Figure 8*) for the 22<sup>nd</sup> realization of pipeline damages is followed.

**Step 1** – Based on the existing literature (Wang et al., 2016; Lewandowski, 2002; Valor et al., 2015), it is assumed that the longitudinal MVD for internal pitting corrosion over the segment under investigation is 0.2 pits per unit length of pipeline. Also, considering the higher likelihood of corrosion at the lower quadrant of oil pipelines (Barton et al., 2017), it is assumed that the circumferential MVD of localized damages follows a normal distribution with zero mean at the bottom of the pipeline. Moreover, assuming that risk of failure is consistent with damage size

over the pipeline segment of interest, the distribution shown in *Figure 9* is considered as the size MVD.

**Step 2-** Using Poisson distribution with identically, independently and uniformly distributed damages along the pipeline segment, the mesh strip width is found to be 0.5 meters, so that the probability of having more than one localized damage at each mesh strips is less than 0.01.

**Steps 3, 4** –The problem setup corresponding to the 22<sup>nd</sup> realization is provided in *Figure 10* where axes are defined based on Eq.1 and *Figure 6*. Moreover, diamonds of different sizes represent localized damages of classes 1 through 4 (see *Figure 9*) where classes are randomly assigned to each damage.



*Figure 10- The 22<sup>nd</sup> realization out of 46 total realizations of the pipeline of Example 1*

**Step 5-** Thirteen nodes are located along the pipeline segment (stars represent nodes in *Figure 10*). The longitudinal locations of the nodes are set to coincide with those of corresponding pits while the circumferential location of the nodes is determined by adding a random value from the interval  $(-0.5, 0.5)$  to that of the corresponding damage. This interval is chosen considering AE sensor coverage radius (See *Table 2*).

**Step 6-** In this step, to develop the aggregate POD model (Eq. 4) for AE sensors, a decaying power relation is defined for dependency of POD on damage-node distance. This relation is defined based on distance metric in Eq. 1, assumed coverage radius of AE sensors (0.4 m), and the information provided in (Sause et al., 2018). However, this dependency is neglected for

human inspection with ultrasonic tool since the inspector takes the inspection tool to the exact location of damages in the inspection area. On the other hand, dependency of POD on damage size is calculated for AE sensors and human inspection with ultrasonic tool using probabilistic relations provided in (Zhang and Zhou, 2014) and (Chatterjee and Modarres, 2013), respectively. As such, POD as a function of type-size class of 22<sup>nd</sup> realization damages is presented in *Table 3*.

*Table 3- POD as a function of damage type-size class*

Class (k)	Acoustic Emission (m=1)	Human Inspection (m=2)
1	0.32	0.02
2	0.54	0.07
3	0.72	0.25
4	0.90	0.77

IP matrices are developed next. Due to a lack of access to real world data and in the absence of a pattern recognition practice on historical data, it is assumed here that damage class and the inference distance (between a node and the point to be inferred) are important inference factors to be considered. As such,  $\log(\overline{\Theta^{k,m=1}})$  matrices (see section 2.3, Step 6) for AE sensors (m=1) are developed where a positive correlation of IP and damage size and a negative correlation of IP and inference distance are considered. Moreover, it is assumed that inference distance limit (section 2.3, Step 7) increases with damage size and is approximately 20 meters for all classes of damages. On the other hand, for human inspection with ultrasonic tool (m=2), it is assumed that IP values are greater than and proportional to those of an AE sensor. This assumption is made since human inspection provides information on the state of all points in the greater inspection area as opposed to limited coverage range of an AE sensor. Subsequently,  $\log(\overline{\Theta^{k,m=2}})$  is assumed to be  $1.5 \times \log(\overline{\Theta^{k,m=1}})$  for any damage of any class.

Lastly, to obtain ME values ( $\delta_{k,m}$  in Eq. 5), at fist relations provided in (Rabiei and Modarres, 2013) and (Chatterjee and Modarres, 2013) are used to simulate size values sensed by AE sensors and ultrasonic tool (i.e. inspection tool), respectively. Then, probabilistic ME values are calculated as probability of sensed size values not being laid in the interval of the original class of a damage of interest. Resulted ME values and class intervals are reported in Table 4.

Table 4- Class intervals and corresponding ME probabilities ( $\delta_{k,m}$ )

Class (k)	AE sensors (m=1)	Human Inspection (m=2)	Class Interval (mm)
1	0.130	0.147	[0 0.19]
2	0.060	0.151	[0.19 0.32]
3	0.045	0.142	[0.32 0.55]
4	0.015	0.145	[0.55 1.50]

**Step 7-** Following the proposed clustering approach (section 2.3, Step 7), damages of the 22<sup>nd</sup> realization are clustered to 3 groups where the clustering distance limit is determined to be 15 meters. In addition, the minimum number of detected damages of each cluster (Table 5) is determined considering the overall detection lower limit (Eq. 6.14) obtained using partial coverage inspection analysis.

Table 5- Clustered damages of realization #1

Cluster #	Damage #	Minimum Detected Damages
1	1,2,3,4	2
2	5,6,7,8,9	3
3	10,11,12,13	2

**Steps 8, 9-** In parallel, it is assumed that the overall detection lower limit is 50% for an estimation of the entire segment, with 90% confidence. Considering the overall detection lower limit, minimum detection of each of clusters is determined in step 7.



**Step 10-** Subjective utility weights (*Table 6*) are used to form the utility function that is defined to be maximized (see *Table 1* for abbreviations). Moreover, the cost limit is set to be 25 (see *Table 2* for cost metric values). Also, LPOND limits (Eq. 6.21) are set to be -1.5 and -12.

*Table 6- Utility weights used in utility function*

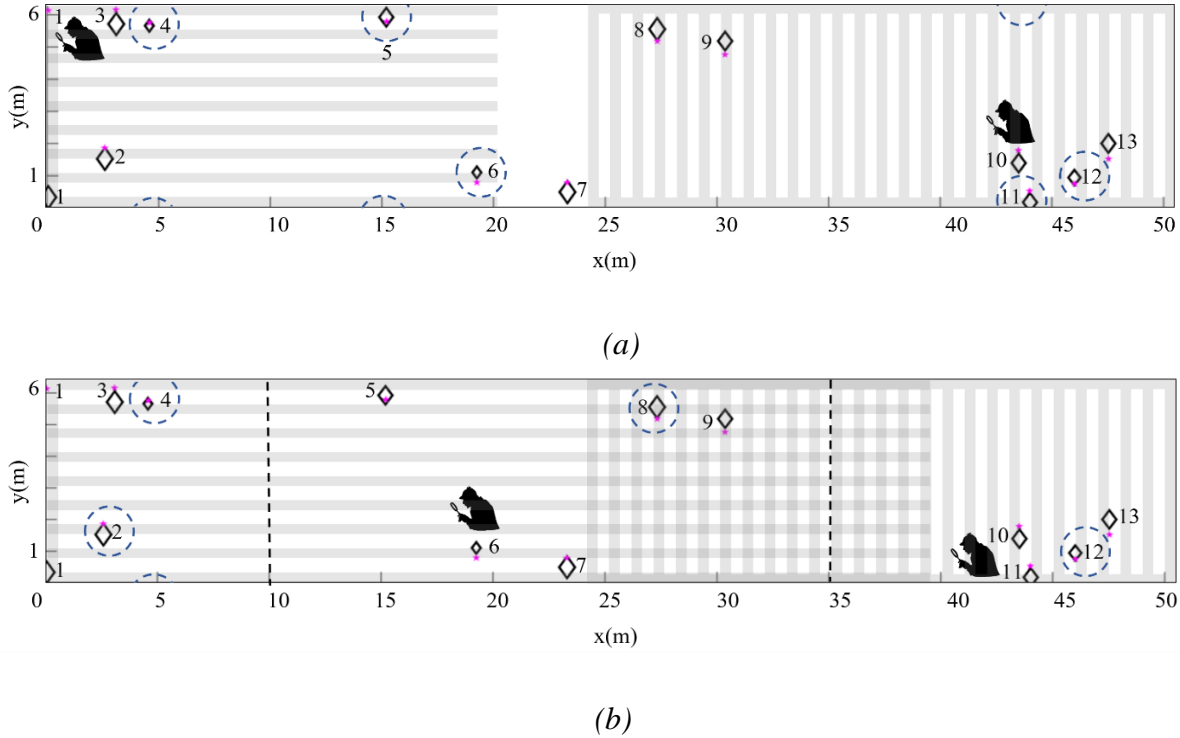
$W_{co}$	$W_{cr}$	$W_{fr}$	$W_{iv}$	$W_{ME}$
0.3	0.3	0.1	0.15	0.15

### 2.5.1.2 Single Realization Results and Discussion

Two health monitoring layouts and results corresponding to the 22<sup>nd</sup> realization of the pipeline segment of this example are discussed in this section. All the results presented in this section, including layouts illustrated in *Figure 11*, are obtained using the GAMS® optimization software program and BARON® solver (Tawarmalani and Sahinidis, 2005) using a desktop computer with a 64-bit Windows® 10 operating system, an Intel® Core™ i7-2760QM CPU at 2.40 GHz, and 16.0 GB of RAM. Also, the damage simulation is conducted using the R programming language. The corresponding code package is described in Appendix A. For all the results of this chapter, the weighting factor of the objective function ( $w_{obj}$  in Eq. 6.1) is equal to 0.5 unless mentioned otherwise.

Dashed circles in *Figure 11* are the coverage boundary of the AE sensors and hatched areas represent the inspection areas. Moreover, double hatched areas are inspected twice (Note that inspection time is not discussed in this dissertation and inspections can be done at different time instances). Average LPOND, which is considered in the objective function of the formulated optimization problem (Eq. 6.1) and summarizes POD and IP (Eq. 3), is considered in both layouts of *Figure 11*. However, clustering and ME are not considered in layout (a) to

illustrate improvements achieved upon their consideration in layout (b). Hence, dashed lines in *Figure 11* (b) represent cluster boundaries.



*Figure 11- Optimal health monitoring layout corresponding to the 22<sup>nd</sup> realization when (a) damage clustering and ME are not considered, (b) damage clustering and ME are considered.*

Layouts of *Figure 11* indicate that consideration of clustering and ME in layout (b) has led to a better average utility (*Table 7*). This improvement can be resulted from lower health monitoring cost in layout (b) with four AE sensors, instead of five as in layout (a). Moreover, efficient assignment of AE sensors to nodes 2 and 8, with larger damages and smaller ME values, in layout (b) instead of nodes 4 and 6, as is done in ME-insensitive layout (a), can be the other reason of this improvement.

LPOND (Eq. 3) is a log-linear relation that is not sensitive to decimal probabilistic changes. Hence, insufficient monitoring (e.g.  $LPOND = \log(0.11) = -0.95$ ) of some areas/damages of a segment is likely while other areas are excessively monitored (e.g., LPOND

=  $\log(0.11/10) = -1.95$ ). Thus, damage clustering is utilized here to provide guidelines for a placement of detection methods consistent with the distribution of damages over all parts of a pipeline segment. As such, layout (b) of *Figure 11* has a slightly worst average LPOND, a better overall coverage and a better average redundancy (Table 7) in comparison to layout (a).

Considering the discussion above, it can be concluded that utilizing clustering and ME has resulted in a better performance in the case of 22<sup>nd</sup> realization. Moreover, *Table 7* reveals that usage of clustering not only has not led to sub-optimality, but also has led to an improved optimization runtime. As such, impacts of utilization of clustering and ME on the optimal layouts of all 46 realizations of this example are explored next to check if these improvements are seen for other realizations as well.

*Table 7- Comparison of performance of the layouts of Figure 11*

	Ave (Utility)	Ave (Redun.)	Ave (LPOND)	Run Time (s)
Without clustering and ME	0.29	1.30	-9.10	106
With clustering and ME	0.45	1.46	-8.73	68
Improvement %	55%	12%	-4%	55%

### 2.5.1.3 Multiple Realization Results and Final Aggregate Layout

Statistics of improvements achieved upon consideration of clustering and ME corresponding to all 46 realizations of this notional example (recall the beginning of section 2.5.1) are reported in *Table 8*. Reported statistics indicate that the average utility is improved upon consideration of clustering and ME for majority of the realizations. Nonetheless, the average LPOND gets slightly worse for considerable number of the realizations while average detection redundancy is improved. These observations can be due to a more efficient placement of detection methods following utilization of clustering, as was discussed earlier. On the other

hand, reported statistics for run time reveal a significant improvement for almost all the realizations when damage clustering is considered. The shrinkage of the combinatorial feasible region of the optimization problem following removal of cases that violate detection limits of any damages cluster can be the reason of this observation.

*Table 8- Statistics of improvement percentage in layout performance for 46 realizations of Example 1 when clustering and ME are considered*

	Ave(Utility)	Ave(Redun.)	Ave(LPOND)	Run Time (s)
10% quantile	36.4%	-5.0%	-7.5%	0.0%
90% quantile	62.1%	25.0%	3.0%	210.0%
<b>Average</b>	<b>49.2%</b>	<b>8.8%</b>	<b>-1.0%</b>	<b>93.4%</b>
Standard Deviation	11.0%	13.8%	5.2%	109.7%

Considering *Table 8*, utilizing damage clustering in more computationally complex cases of layouts with ME reduces the run time considerably, while leading to improvements in terms of health monitoring performance (e.g., average utility and average detection redundancy). Thus, we may claim that the proposed approach provides a computationally tractable solution for consideration of probabilistic detection metrics in optimal health monitoring of pipelines and the final layout of each realization should be the one that considers clustering and ME.

An aggregation of all 46 optimal health monitoring layouts of this example is illustrated in *Figure 12* where there are 176 AE sensors (triangles) and 54 inspection nodes (plus signs). As a result, it is expected to have 4 AE sensors and 1 human inspection with ultrasonic tool in each health monitoring layout. Consequently, the final health monitoring layout is obtained (*Figure 13*) where sensors and human inspection are located at the center of 4 sensor clusters and 1 inspection node cluster corresponding to *Figure 12*.

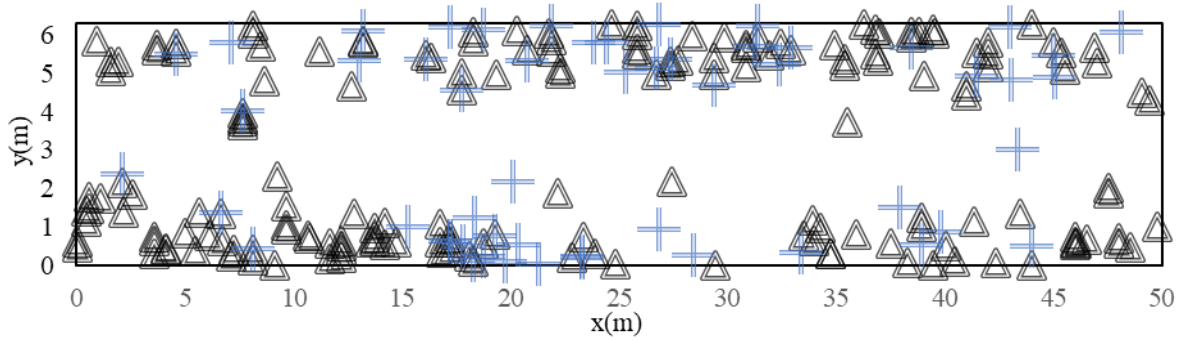


Figure 12- Scatter plot obtained through aggregation of all 46 optimal layouts of Example 1

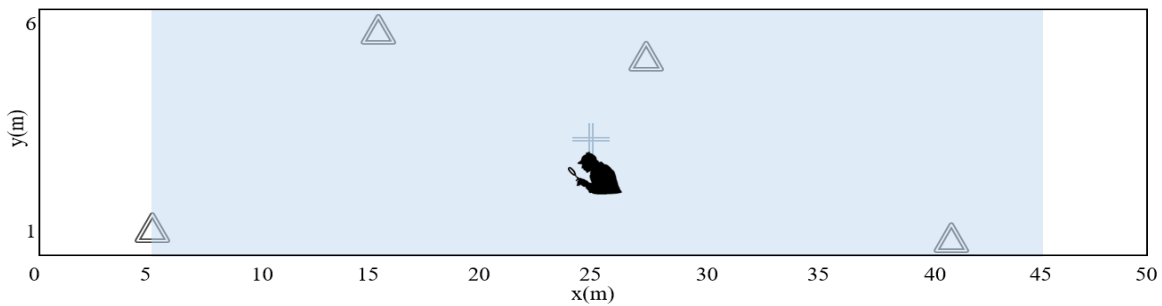
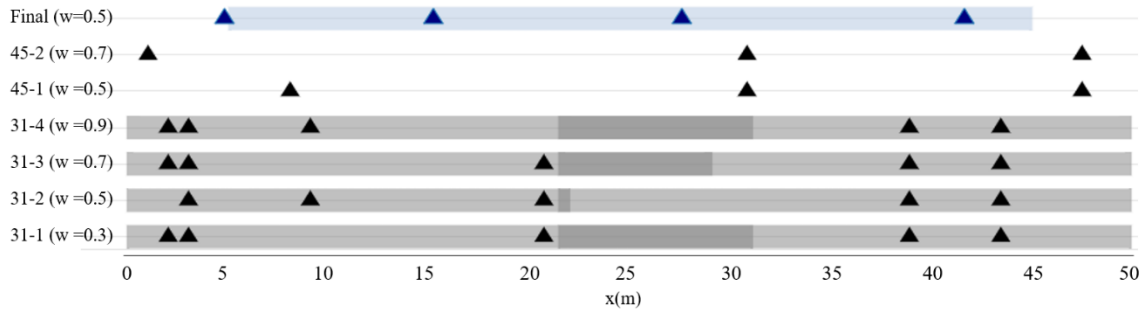


Figure 13- Final health monitoring layout corresponding to the pipeline segment under study

All 46 realizations of this example, i.e., a spectrum of localized damages with different sizes and at different locations, are considered in obtaining the final layout of *Figure 13*. Moreover, placement of sensors and human inspection, is consistent with the uniform longitudinal and non-uniform circumferential spatial distribution of localized damages (see section 2.5.1.1). Nonetheless, to verify the choice of this layout as the final layout, the performance of this layout is compared with that of (Pareto) optimal designs corresponding to the most and least expensive (in terms of health monitoring total cost) layouts of all 46 realizations. Pareto optimal designs are different layout designs corresponding to the various values of weight factor  $w_{obj}$  in the objective function (Eq. 6.1) of formulated optimization

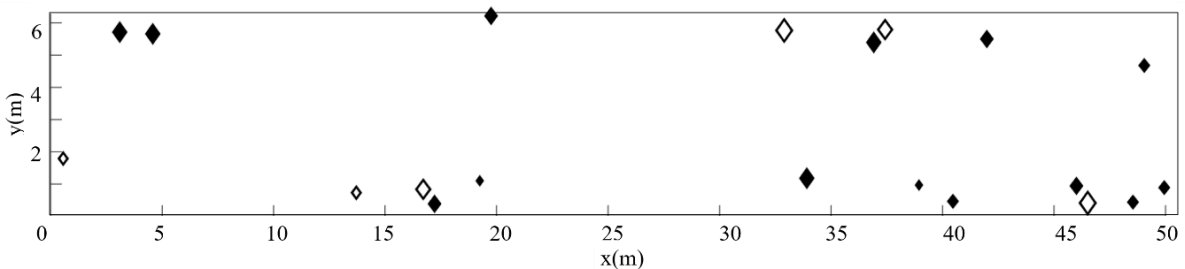
problem. In *Figure 14*, longitudinal projection of Pareto designs corresponding to realizations 31 (most expensive layouts with redundant inspection) and 45 (least expensive layouts with no



*Figure 14- Longitudinal projection of final health monitoring layout and Pareto designs corresponding to most and least expensive layouts of realizations of example 1. Here,  $w$  refers to  $w_{obj}$  in Eq. 6.1*

human inspection) are presented. Also, design “final” represents the longitudinal projection of layout of *Figure 13*. In *Figure 14*, the triangles denote AE sensors while shaded strips denote inspection area. In addition, double shaded strips represent regions that are inspected twice.

Using the Analytic Hierarchy Process (Saaty, 2008), performance of the projected layouts of *Figure 14* was compared for two newly generated realizations of the segment under investigation (filled and hollow diamonds in *Figure 15*). It was concluded that the final health monitoring layout (*Figure 13*) outperforms all layouts of *Figure 14* for both test realizations in terms of average utility and average LPOND.



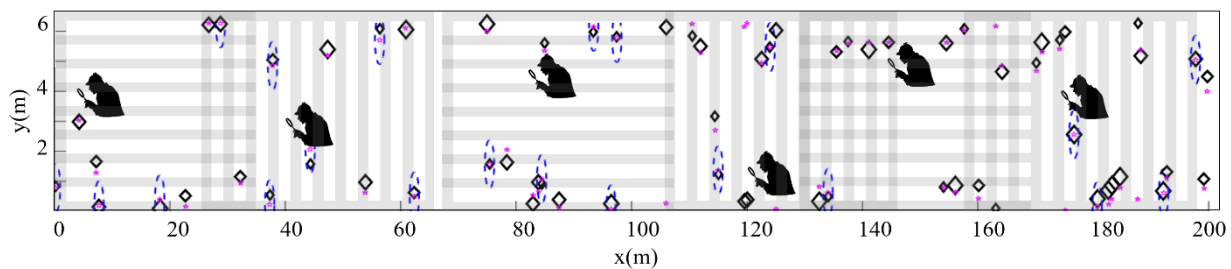
*Figure 15- Two newly generated realizations used as test realizations*

### 2.5.2 Example 2

To show the applicability of the proposed approach for larger and more densely corroded pipelines, a pipeline segment of 200-meter length and 1-meter radius is considered. It is assumed that the longitudinal intensity of localized damages is 0.4 pit per meter. Data generation process and all other assumptions, conditions, and relations used in this example are the same as those in the previous example.

The health monitoring layout corresponding to one realization of the segment for this example is shown in *Figure 16* where diamonds of different sizes denote localized damages which belong to different classes. Stars represent nodes and dashed ovals represent AE sensor coverage boundary. Also, hatched areas represent inspection areas.

There are 22 AE sensors and 6 inspection nodes in the optimal layout of *Figure 16* that is reasonable considering the long pipeline segment and greater corrosion density. Similar to the previous example, the final health monitoring layout of this example (with 95% coverage of localized corrosion variations with 90% confidence) can be attained through aggregation of 46 optimal layouts corresponding to 46 realizations of the longer pipeline segment of this example.



*Figure 16- Health monitoring layout corresponding to one realization of the longer pipeline segment*

## 2.6 Concluding Remarks

This chapter presents a new approach for an early stage health monitoring layout optimization for pipelines with a focus on probabilistic detection of localized damages. This approach considers different detection methods (e.g. different sensors and human inspection tools) simultaneously. The optimal layout includes location and type of sensors and human inspection (tools).

The proposed approach has several important features. (i) It uses three metrics for probabilistic detection of damages with different levels of severity. As shown in the results and discussion section, the use of these metrics can reduce the cost of health monitoring. (ii) It is based on an optimization objective function which is formed based on a weighted sum of two functions: a health monitoring utility function and a function for probabilistic detection of damages. By changing the weights for these functions, the user can explore different optimized layout solutions and also manage tradeoffs between the two functions. (iii) It considers a significant number of key attributes of the detection methods, such as detection cost, coverage capability, data acquisition frequency, and measurement error. Compared to the existing literature, this will provide a more detailed account of the detection methods considered. (iv) It uses probabilistic sampling methods for simulating and placing damages on a pipeline surface considering relevant available data. In this way, not only the expected value of damage specifications (e.g., damage location and size) can be considered, but also their variations can be accounted for.

The proposed approach is demonstrated with two notional examples. For the first example, a step-by-step demonstration of the proposed approach is provided. In addition, considering the severity level of different damages, it is shown that the proposed layout



optimization approach can obtain a better solution than a single detection method approach. For the second example, a longer pipeline segment with more densely localized damages is considered to show applicability of the proposed approach to larger sized problems.

In the next chapter, a detailed explanation of the approach for the dynamic segment-level phase of health monitoring of locally damaged pipelines is presented.

## **Chapter 3: Estimating Damage Size and Remaining Useful Life in Degraded Structures Using Deep Learning-based Multi-source Data Fusion<sup>3</sup>**

One of the key contributions of the PHM approach in this dissertation is that it considers multi-source data fusion in a hybrid context at the dynamic segment-level phase. In this approach for multi-source data fusion, online, frequent, and indirect sensor data along with periodic, less frequent yet accurate inspection images are considered together with a Physics of Failure (POF) model. The final outcome of the developed fusion approach (i.e., dynamic segment-level phase in *Figure 3*) is a more accurate, frequent and online RUL estimation.

The rest of this chapter is organized as follows. Section 3.1 provides a detailed explanation of the developed approach for damage size and RUL estimation where corresponding deep learning methods are discussed. Section 3.2 summarizes the experimental set-up as explained and conducted by Yun (2018), general data analytics and data pre-processing. Section 3.3 discusses the results attained. Lastly, section 3.4 concludes the chapter where possible extensions and future work are briefly discussed.

### **3.1 Proposed Approach for Damage size and RUL Estimation**

The approach for damage size and RUL estimation consists of three key modules, as shown in *Figure 17*: (In *Figure 17*, the number shown in the top right corner of each block refers to the module number, as described next.)

---

<sup>3</sup> This chapter is based on the following paper: Aria, A., Lopez Droguett, E., Azarm, S. and Modarres, M., 2019. Estimating damage size and remaining useful life in degraded structures using deep learning-based multi-source data fusion. Structural Health Monitoring, DOI: [10.1177/1475921719890616](https://doi.org/10.1177/1475921719890616)

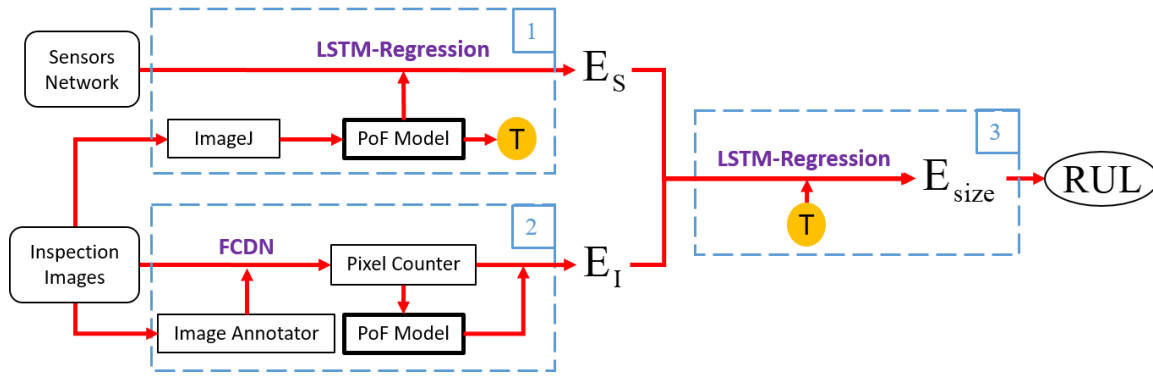


Figure 17- Schematic of the proposed data fusion and RUL estimation approach

**1- Module for Time-series Analysis:** In this first module, a particular class of LSTM networks along with a Multi-Layer Perceptron regression (Rosenblatt, 1961) is used to produce frequent damage size estimates. Multiple-sensor data are considered to denoise data from the sensor of interest. Then, corresponding LSTM-regression model is trained considering denoised sensor data. Size estimations produced by a POF model are considered as ground truth (denoted by T in Box 1) in the training process. The POF model is parametrized based on manual image-based size measurements produced using a NIH ImageJ tool (Schneider et al., 2012).

**2- Module for Image Processing:** In this module, a customized FCDN model is used to estimate damage size through a pixel-wise classification of microscopic images of a damaged area. The corresponding FCDN is trained using annotated images and the output of FCDN, i.e., accurate but low frequency images which are segmented to damage and background parts, is fed into a pixel counter unit to estimate the damage size. Resulting size estimates, which have a frequency similar to that of the first module, are then used to parametrize a corresponding POF model. Next, the parametrized POF model is used to produce image-based size estimate at each instance of time with a sensor-based estimate available.

**3- Module for Data Fusion and RUL Estimation:** In this module, sensor- and image-based damage size estimates obtained from the first two modules (i.e.,  $E_S$  and  $E_I$  in *Figure 17*) are fused together to obtain a final estimation of the damage size and RUL. Using the POF model for data augmentation in the first two modules, the frequency mismatches between sensor- and image-based estimates are resolved. Moreover, the ground truth values produced in the first module are again used here to parametrize the POF model. As such, size estimates of the same frequency are fused in this module to obtain high frequency and accurate damage size and RUL estimates.

The above-mentioned modules and corresponding deep learning models are further detailed in Sections 3.1.1 through 3.1.3.

### **3.1.1 Module for Time-series Analysis**

This section starts off with a general overview of the LSTM networks and their application for damage size estimation. Next, since the performance of the developed approach is investigated in this chapter using data associated with fatigue crack experiments, the proposed LSTM regression model and its dependency on POF model for fatigue crack size estimation is discussed.

#### **3.1.1.1 LSTM-based Regression for Damage Size Estimation**

Consider a structure subjected to a degradation process such as fatigue which results in crack damages. Let us assume that there is time-series data from a sensor known to correlate with degradation behavior of the structure. Also, assume that the degradation behavior at each instance of time  $t$  is only dependent on a limited number (i.e.,  $N_{tw}$ ) of sensor signals transmitted

just before time  $t$ . If the time-series data includes  $N_{ft}$  features extracted from denoised sensor signals, then  $\mathbf{X}^{t\alpha}$  can be defined as a vector consisting of  $N_{ft}$  features corresponding to the signal transmitted at time  $t_\alpha < t$  where  $\alpha$  is integer and  $1 \leq \alpha \leq N_{tw}$ . Now assume a LSTM network with  $N_{tw}$  layers (Figure 18). This network is set up to estimate damage size at time  $t$  considering the last  $N_{tw}$  transmitted signals. For this LSTM network,  $\mathbf{X}^{t\alpha}$  is the input of the layer at time  $t_\alpha$  (Figure 18a) and  $\mathbf{h}^{t\alpha}$  represents the corresponding output. Moreover, all the LSTM layers of this network include the same LSTM cell (Figure 18b). Consequently, a particular LSTM cell structure is considered in obtaining the final output of this network. Moreover, all  $N_{tw}$  signals of interest and corresponding temporal correlations are considered in obtaining the output, e.g., a vector of neural node values corresponding to estimated damage size for time  $t$ .

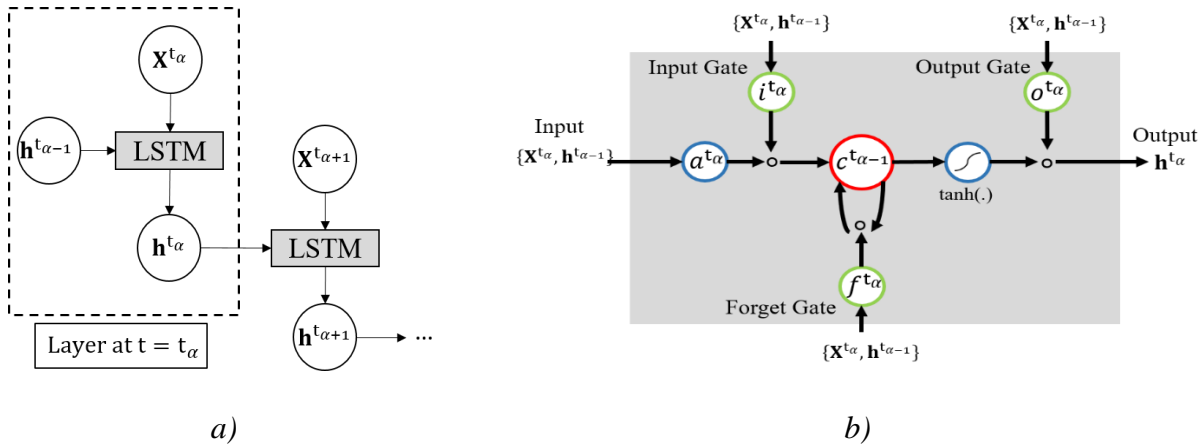


Figure 18- Structure of a) LSTM network; b) LSTM cell

Consider the LSTM cell at time  $t_\alpha$  (Figure 18b). At first, the input vector is fed into the input port ( $a^{t\alpha}$ ) which considers immediate temporal correlations using a hyperbolic tangent activation function. Next, the Hadamard product (denoted by  $\circ$ ) and three different types of gates (Figure 18b) are utilized. All these gates include a sigmoid activation function. Meanwhile, a memory cell ( $c^{t\alpha-1}$ ) is employed to consider longer temporal correlations. The

input port, the gates, and the memory cell are referred to as LSTM operation units in this chapter. For the sake of computational consistency, each operation unit has  $N_n$  neural nodes. A concurrent usage of all of these operation units facilitates a computationally manageable flow of information in LSTM models. This way, the LSTM models can handle long temporal correlations through a manageable computational effort (Zheng et al., 2017).

The LSTM operation units and their functionality are detailed in the literature. It is discussed there (Zheng et al., 2017; Elsheikh et al., 2019) that dropout ( $P_d$  as dropout probability) and batching ( $N_b$  as batch size) are used to avoid overfitting. While different variations of a LSTM cell are reported (Zheng et al., 2017; Elsheikh et al., 2019), the above explained structure is the cell used in the bi-directional LSTM model of this chapter. In a bi-directional LSTM model, each of the  $N_{tw}$  layers consists of two cells so that both backward and forward temporal correlations are considered (Elsheikh et al., 2019). As such, a bi-directional LSTM is used in the LSTM-regression model of this chapter for damage size estimation (*Figure 19*).

In the aggregated model of *Figure 19*, the output of the bi-directional LSTM at the end of each training iteration is a set of weights corresponding to  $N_n$  neural nodes for each of the operation units of a LSTM cell. These weights are then fed into a Multi-Layer Perceptron regression model, with a rectified linear unit (ReLU) activation function (Elsheikh et al., 2019), to estimate the damage size at time  $t$ . In the regression model, the true damage size (i.e., ground truth) values, which are obtained by Eq. 1 and are denoted by “T” in *Figure 19* as well as Module 1 of *Figure 17*, are used as reference points of a loss function, which is then minimized in the regression process. In parallel, the structure of LSTM cell, i.e., weights of neural nodes, is updated through a back-propagation scheme. Consequently, following  $N_{tr}$  training iterations and

using samples of sensor time-series data, the optimal structure of LSTM cell, which leads to a minimized regression loss, is determined. Hence, the damage size at any time  $t$  can be estimated considering the last  $N_{tw}$  sensor signals transmitted prior to time  $t$ .

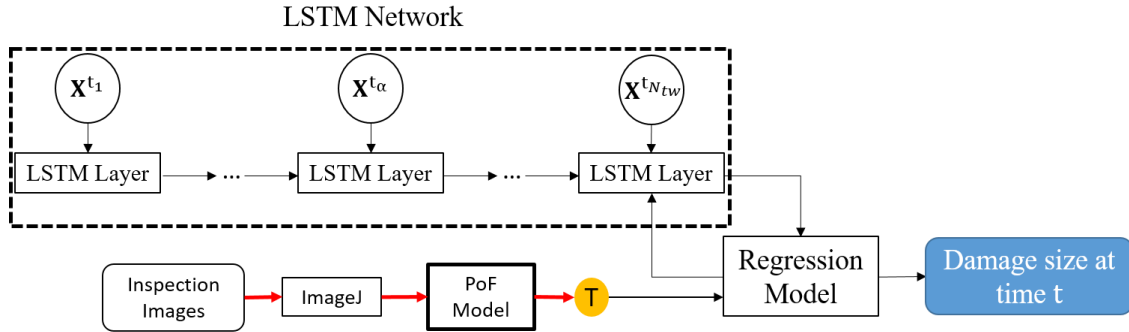


Figure 19- Aggregated LSTM-regression model for damage size estimation

### 3.1.1.2 Customized LSTM-Based Regression Model for Fatigue Crack Size Estimation

In the results and discussion section of this chapter (Section 3.3), the aggregated LSTM-regression model of *Figure 19* is employed to estimate the fatigue crack size based on the corresponding AE signals. In order to guarantee obtaining the best accuracy of crack size estimates using the proposed LSTM-regression model, some issues should be addressed. Firstly, optimal values of hyperparameters should be determined. Hyperparameters are external to the model structure and their values are specified before the start of training process of the model. Hyperparameters of the proposed LSTM-regression model, i.e.,  $[N_{tw}, N_{ft}, N_n, P_d, N_b]$ , and a corresponding grid search are discussed in section 3.3.1. Secondly, accurate and dependable size measurements are required as regression reference points, if high confidence in size estimates is desired. To address this issue for the case of fatigue crack size estimation, an integral form of the Paris-Erdogan relation for fatigue crack growth (Bannantine et al., 1990) is used (Eq. 7) as the POF model of the hybrid approach of the time-series analysis module. This POF

model is parametrized based on manual image-based crack size measurements produced by the NIH ImageJ tool. The resulting crack size estimates are then used as the reference points of the regression procedure.

$$a_i = \int_{i_0}^{i-1} C(1.12\Delta\sigma\sqrt{\pi a})^m dN \quad (7)$$

In Eq. 7,  $a_i$  represents crack size (length) at loading cycle  $i$ .  $N$  denotes the loading cycle and  $i_0$  is the loading cycle at which crack is initiated.  $\Delta\sigma$  is the cyclic loading stress amplitude. Also,  $C$  and  $m$  are model parameters to be determined. This choice of POF model is based on existing literature on promising results from AE-based crack growth rate estimation using a Paris-Erdogan model (Bassim et al., 1994; Rabiei and Modarres, 2013; Keshtgar et al., 2018).

The last issue about the LSTM-regression model of *Figure 19* is the choice of the loss function. A loss function can be defined and manipulated to favor size overestimation as it is desired in PHM of structures where an underestimation of damage size and subsequent overestimation of RUL is not acceptable (Lei et al., 2018). Hence, a modified RUL-based loss function (Zheng et al., 2017) is employed in this chapter for fatigue crack size estimation. Eq. 8 formulates this loss function where model parameters  $a$  and  $b$  are determined considering manual crack length measurements. Here,  $L_{pre}$  denote LSTM predicted size values while  $L_{POF}$  is POF generated crack size value considered as true damage size. Note that negative values of LSTM predicted crack size values are penalized in Eq. 8 using  $P_{neg}$  term.



$$\text{RUL loss} = \left\{ \begin{array}{ll} \exp\left(\frac{L_{\text{pre}} - L_{\text{POF}}}{a}\right) & 0 \leq L_{\text{POF}} \leq L_{\text{pre}} \\ \exp\left(\frac{L_{\text{pre}} - L_{\text{POF}}}{-b}\right) & 0 \leq L_{\text{pre}} < L_{\text{POF}} \\ \exp\left(\frac{L_{\text{pre}} - L_{\text{POF}}}{-b}\right) + P_{\text{neg}} & L_{\text{pre}} < 0 \end{array} \right\} \quad (8)$$

### 3.1.2 Module for Image Processing

In this section, first, an overview of the main building blocks of FCDNs with a minor focus on an application in damage size estimation is presented. Then, a customized FCDN structure and its application for fatigue crack size estimation in relation with segmented and annotated images is discussed.

#### 3.1.2.1 FCDN for Damage Size Estimation

A FCDN model for semantic image segmentation, with a minor focus on segmentation of images of damaged area with two classes of objects (damage and background), is discussed in this Section. *Figure 20* illustrates general architecture of a FCDN model where downsampling and upsampling paths are considered together to resolve the issue of resolution loss in standard CNNs. In a downsampling process (Jegou et al., 2017), successive convolutions are carried out to generate low resolution feature maps which are used to find coarse patterns in the original input (image). On the other hand, successive transposed convolutions are utilized in upsampling path to produce high resolution feature maps and find fine patterns in the original image. In transposed convolution (Dumoulin and Visin, 2016), higher resolution feature maps are produced while maintaining spatial consistency with respect to the original and lower resolution

feature maps. As such, downsampling and upsampling counterpart units, which are at the same resolution level (*Figure 20*), are connected in a FCDN model to maintain the spatial consistency. Coarse pattern features maps produced by a downsampling unit are passed on to the counterpart upsampling unit through skip connection (*Figure 20*). Next, all the coarse and fine pattern feature maps at the same resolution level are concatenated and fed into the next upsampling unit. Doing so, the new fine pattern feature maps with a higher resolution are produced considering both fine and coarse patterns of the original image. This process is referred to as multi-scale supervision (Jegou et al., 2017) and leads to computationally efficient recovery of the resolution of the original input image. The heart of efficient multi-scale supervision in FCDN models is the reverse correlation between number and resolution of feature maps (*Figure 20*).

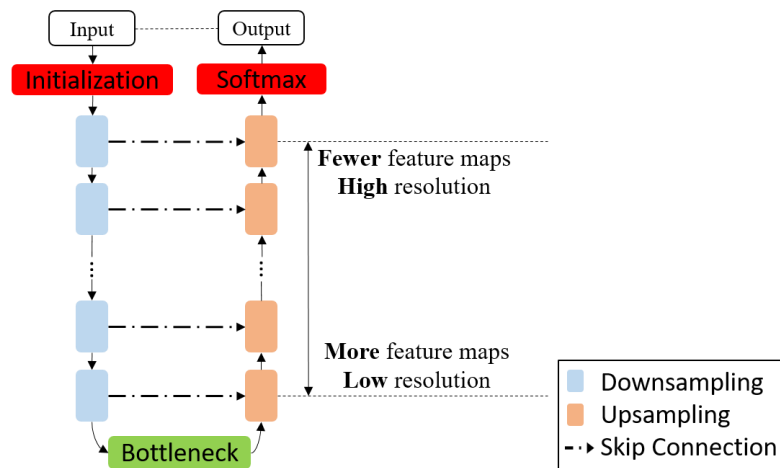


Figure 20- Architecture of FCDN model for semantic image segmentation

A brief discussion of the mechanism behind computational efficiency in FCDN models is followed where the units of FCDN architecture, from input to output, are shortly explained based on the original work in (Jegou et al., 2017). To have a more focused discussion, images of damaged area are considered as the model input in the following description.

- 1- Initialization unit:** The input of this unit is three layers of an RGB image of a damaged area. A standard CNN with a set of  $k_{in}$  convolution kernels of shape  $[S_{in1} \times S_{in2}]$  form this block. These convolution kernels essentially help with locating the damage. Hence, their shape is a function of damage geometry. The output of this unit is  $k_{in}$  feature maps with no loss of resolution with respect to the input.
  
- 2- Downsampling unit:** This unit consists of a dense block (Huang et al., 2016), followed by a concatenation and transition down sub-unit.  $N_s$  number of this unit are consecutively used to produce a set of feature maps at different resolution levels. The proposed structure of this unit is shown in *Figure 21* and its sub-units are briefly discussed below:

**Dense Block (DB):** A DB (Huang et al., 2016) is consisted of  $N_l$  consecutive layers. The input of the first layer is the output of the preceding unit (i.e., initialization unit or an earlier downsampling unit). The input of other layers, however, is the concatenation of the original input of the DB and output of preceding layer in the DB. In each layer, a set of  $k$  convolutions with kernels of shape  $[S_{db} \times S_{db}]$  are carried out on the input. Hence, output of each layer is a set of  $k$  feature maps. As a result, the final output of the block is the concatenation of all  $N_l \times k$  feature maps with the same resolution as the input of DB.

**Concatenation:** To facilitate multi-scale supervision (Jegou et al., 2017), output of the DB is concatenated with the identity mapping of the DB input and the result is fed into the transition down unit as well as the upsampling unit at the same resolution level.

**Transition Down (TD):** Resolution (spatial dimensionality) of feature maps is reduced in this sub-unit through a  $[S_t \times S_t]$  convolution and a  $[2 \times 2]$  max pooling. With the concatenated feature maps as the input, the output of a TD sub-unit is a set of reduced resolution feature maps.

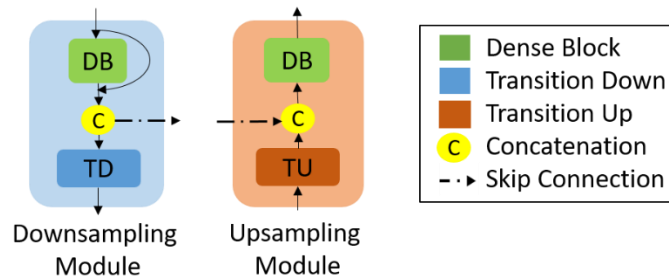


Figure 21-Downsampling and upsampling units of the FCDN model for semantic image segmentation

- 3- **Bottleneck:** It is a DB at the end of downsampling path and before upsampling path. It has  $N_{bl}$  layers and its output is  $N_{bl} \times k$  feature maps with the same resolution of bottleneck input.
- 4- **Upsampling unit:** This unit is consisted of a transition up sub-unit followed by concatenation and a DB (Figure 21). Similar to downsampling path,  $N_s$  number of this unit are consecutively used in upsampling path. However, they are used here to recover original resolution in a step-by-step manner. Therefore, each upsampling unit is connected to its downsampling counterpart at the same resolution level through a skip connection (Figure 20). The main difference of these counterparts is that no identity mapping is done in upsampling units. In other words, only feature maps generated by the preceding unit are upsampled in the transition up (TU) sub-unit of each unit to avoid

feature explosion. To do so, transposed convolution is utilized. As such, a  $[S_t \times S_t]$  convolution kernel is used in transposed convolution of each TU sub-unit to partially recover the resolution of feature maps produced at the preceding unit. Next, upsampled feature maps are concatenated with those of the same resolution produced at the counterpart downsampling unit. Lastly, concatenated coarse and fine pattern feature maps are fed into a DB to produce new fine pattern feature maps at the same resolution level.

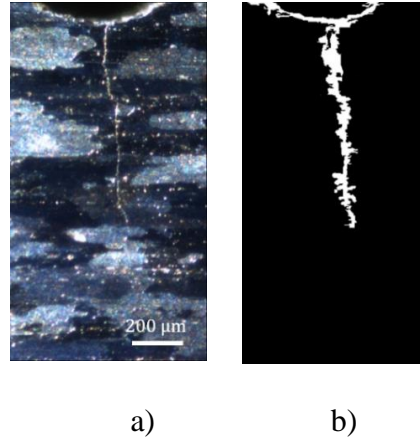
- 5- **Softmax unit:** a  $1 \times 1$  same convolution is carried out on high resolution feature maps attained at the end of upsampling path. The result is a set of  $n$  feature maps with  $n$  representing the number of classes of objects in the images (e.g.,  $n=2$  for images with damage and background as the classes). Lastly, a softmax function determines per pixel class of objects.

Considering the above-explained FCDN architecture, a FCDN model is customized in the next section for fatigue crack size estimation. The customized FCDN model will be used in the image processing module of the developed approach (recall Module 2 in *Figure 17*).

### 3.1.2.2 Proposed FCDN-based Approach for Fatigue Crack Size Estimation

A customized FCDN model, an image annotator unit and a pixel counter unit are used in the image processing module of the approach in this chapter (recall Module 2 of *Figure 17*). To customize and train a FCDN model for fatigue crack size estimation, hundreds of (RGB) images

of cracked area should be annotated. Thus, microscopic images of five fatigue crack experiments (see section 3.2) were annotated using MATLAB® and Photoshop®. A sample of the original and annotated images are presented in *Figure 22*.



*Figure 22- a) A sampled microscopic crack image, b) corresponding annotated image. The resolution of both images is  $480 \times 1024$ .*

*Table 9 -Hyper parameters of FCDN\_AA for semantic segmentation of fatigue crack images*

Hyperparameter	(Optimal) Value
Batch size	10
Dropout probability	0.2
Number of Initialization kernels ( $k_{in}$ )	48
Shape of Initialization kernels ( $[S_{in1} \times S_{in2}]$ )	$[32 \times 3]$
Number of sampling modules ( $N_s$ )	Grid Search: [1,2]
Number of layers per DB ( $N_l$ )	Grid Search: [3,4,6,8,10]
Number of feature maps per DB layer ( $k$ )	Grid Search: [1,2,3,4,5,6]
Shape of DB kernels ( $[S_{db} \times S_{db}]$ )	$[5 \times 5]$
Shape of transition kernel ( $[S_t \times S_t]$ )	$[3 \times 3]$
Number of layers in bottleneck DB ( $N_{bl}$ )	$N_{bl} = N_l$

For the case of fatigue crack size estimation, the output of the FCDN model will be a two color (e.g., black and white) segmented image where each color represents one class of objects (e.g., black for background and white for crack pixels). Thus, annotated ground truth images are also in black and white as illustrated in *Figure 22b*.

Considering the microscopic fatigue crack images (*Figure 22a*) and corresponding annotated images (*Figure 22b*), a FCDN model is customized. A majority of the hyperparameters of this model, which is referred to as the FCDN\_AA model, are determined during the primary analysis of crack and annotated images (*Table 9*). However, the optimal values of  $[N_l, k, N_s]$  are determined through a grid search process over the intervals presented in *Table 9*. An optimal architecture of FCDN\_AA and its performance are discussed in *section 3.3.2*.

For optimization loss function, the Intersection over Unions (IoU) score is considered in FCDN\_AA model. IoU is defined in Eq. 9 where  $pre_i$  and  $y_i$  represent FCDN predicted and true class of pixel  $i$ , respectively, while  $c$  is used to designate the crack or background.

$$IoU(c) = \frac{\sum_i (pre_i = c \ \& \ y_i = c)}{\sum_i (pre_i = c \ \text{or} \ y_i = c)} \quad (9)$$

The average mean IoU is formulated in Eq. 10 where  $E_{set}(\cdot)$  represents the expected value with respect to training or testing set of images. The average mean IoU is considered as the optimization loss function of FCDN\_AA model.

$$E(IoU) = E_{set}(E_c(IoU(c))) \quad (10)$$

Upon optimizing the loss function, the output of FCDN\_AA, i.e., a segmented black and white image, is generated. With white representing crack pixels, the crack size is estimated next by a pixel counter unit in which the length of the median line of white pixels, excluding the notch, is calculated. The resulting crack size estimates are then used to parametrize a POF model

(Eq. 7) and generate automated and frequent image-based size estimates. Next, the resulting estimates are fed into the data fusion and RUL estimation module.

### **3.1.3 Module for Data Fusion and RUL Estimation**

Following generation of the damage size estimates by the time-series analysis and image processing modules, a model similar to the LSTM-regression model of time-series analysis module (*Figure 19*) is parameterized and trained for data fusion and final size estimation. Since image-based estimates are generally much less frequent in comparison to those of sensor data, POF model is utilized in the image processing module as well as time-series analysis module to solve the corresponding frequency mismatch problem. Then, the same-frequency size estimates produced by the first two modules are fused together using the LSTM-regression model of the data fusion and RUL estimation module. Subsequently, the final output of the proposed approach, i.e., an online estimation of RUL, is produced.

Performance of the proposed approach and its final output are evaluated using AE data and microscopic images produced in a set of fatigue crack experiments. These experiments are summarized next.

## **3.2 Experimental Set-Up and Data Pre-Processing**

In this section, first an overview of the fatigue crack experiments is provided. These fatigue crack experiments were conducted at the University of Maryland's Center for Risk and Reliability as part of another research effort (Yun and Modarres, 2018). The relevant specifications and features of those experiments are briefly discussed here. An application of a



POF model in time-series analysis and image processing modules of the developed approach is discussed next considering the experimental data. Lastly, the data pre-processing for the two mentioned modules is explained.

### **3.2.1 Overview of the Experiments**

Five fatigue crack experiments were carried out by Yun (Yun, 2019). In those experiments, aluminum alloy 7075-T6 dogbone specimens, with a notch at the center as a stress concentrator, were tested under uniaxial cyclic loading. In the experimental set up, two AE sensors were symmetrically placed at 23 mm distance from the specimen notch so that two AE sensor data streams can be used for signal denoising purposes and increase integrity of extracted AE features as a result. Furthermore, the experimental set up is managed to minimize AE noise corresponding to vibration in mechanical connections (Yun, 2019; Yun et al., 2018). However, only AE features corresponding to one AE sensor are considered here for the final assessment of crack length.

Regarding the loading conditions in Yun's experiments (Yun, 2019), the maximum load for experiments 1 through 5 is reported to be 13 through 17 kN, respectively, with 0.1 stress ratio and 5 Hz frequency. Loading was ceased every 1000 cycles for 2 minutes to capture images with a frequency of 0.2 Hz. Hence, the time stamp (loading cycle) of all images and sensor readings are available and are used in parametrizing POF models to temporally relate corresponding sensor- and image-based size estimations.

### 3.2.2 Application of POF Model in the Time-Series and Image Processing Modules

During the fatigue testing, an optical microscope system took images of crack initiation/growth area. Corresponding images were used for both manual and automated crack length estimation. Limited manual NIH ImageJ measurements are used for POF model parametrization in the time series analysis module. Doing so, POF generated length estimates are used as ground truth at instances of time with an AE signal available to train the LSTM models of time-series analysis and data fusion modules. On the other hand, the above-explained procedure is also carried out for the image processing module. However, the POF model is parametrized there based on corresponding accurate and automated image-based estimates to generate frequent size estimates and facilitate data fusion process.

### 3.2.3 Data Pre-Processing and Primary Data Analytics

AE features files were obtained for five fatigue crack experiments by Yun (2019). A primary exploration of features files revealed that a large portion of the transmitted AE signals corresponds to pre-initiation stage of the crack. Furthermore, the literature review revealed that the tasks involving crack detection (i.e., crack initiation recognition) and crack length estimation are separately investigated in nearly all of the reported works (Bassim et al., 1994; Keshtgar et al., 2018; Rabiei and Modarres, 2013; Shateri et al., 2017). Moreover, the AE cumulative count and energy have shown promising results in the literature on crack detection and growth estimation using AE (Keshtgar et al., 2018; Smith et al., 2018). Consequently, the AE cumulative count and energy are used as the input features in our time-series analysis ( $N_{ft}=2$ , see section 3.3.1 ) where crack detection and length estimation tasks are carried out in a two-fold process: ***1- Crack***

**detection:** considering a previous work (Keshtgart and Modarres, 2013), abrupt change in the AE count per signal is considered for crack initiation detection. It turned out that this value is in the range of [65 150] for all of Yun’s experiments where crack initiation is defined to be the time at which crack reaches 250  $\mu\text{m}$  of length. Thus, 100 units change in AE count per signal was set as the crack initiation threshold. 2- **Crack length estimation:** assuming an existing crack, the two mentioned AE features of the post-initiation set were considered as the input to the time-series analysis module to estimate the crack length.

### 3.1.1.2 Customized LSTM-Based Regression Model for Fatigue Crack Size Estimation

*Table 10- Sample AE data for five experiments when the crack length reaches 999  $\mu\text{m}$*

Experiment	Time (sec)	Cum(Count)	Normalized Cum(Count)	Cum(Energy)	Normalized Cum(Energy)
1	9,371	434,265	1.40	$1.07 \times 10^9$	11.30
2	2,569	31,403	3.62	$1.2 \times 10^7$	2.51
3	6,510	44,611	1.18	$3.1 \times 10^7$	1.10
4	8,944	30,682	1.54	$1.3 \times 10^7$	1.66
5	4,009	44,985	1.41	$2.7 \times 10^7$	1.32

Through a preliminary implementation of the proposed two-fold process, it turned out that crack length estimation results were not promising when different experiments are considered in the training and testing processes. This observation is logical considering the considerable variability in features values corresponding to similar crack length in different experiments as it is shown in *Table 10* (normalized columns represent original value of the corresponding feature divided by the last value of that feature in the pre-initiation stage of the crack).

The significant variability in the AE waveforms and resulting features have also forced most of the researchers to study AE-based damage growth trend as opposed to AE-based damage size estimation (Yun et al., 2018; Shateri et al., 2017). As such, one LSTM regression model is

trained and tested for each of the five experiments of this study in the time-series analysis module as well as data fusion module. Normalized cumulative AE count and normalized cumulative AE energy are considered as the input of the time-series analysis module in this study.


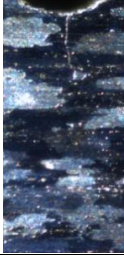
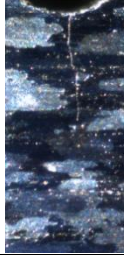


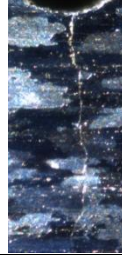
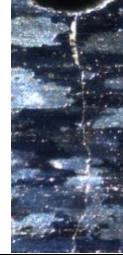
For a detailed illustration of results and evaluation of the performance of the proposed approach, size estimates corresponding to Experiment 1 are presented and comprehensively discussed in sections 3.3.1 through 3.3.3.

### **3.3. Results and Discussion**

Among the five experiments in this study, Experiment 1, with the smallest maximum loading (13 KN), showed the slowest average growth rate where it took more than 5 hours to reach the assumed failure threshold of 1,650  $\mu\text{m}$  of crack length. As a result, the largest AE features file (with more than 150,000 entries) and the largest images set (with more than 1,900 clear images) were generated in Experiment 1 (Yun, 2019). *Table 11* summarizes microscopic images of Experiment 1 where seven different crack lengths are recognized. This is important for illustration and training purposes as long as only three or four different crack lengths are captured in images of other experiments. Hence, results corresponding to Experiment 1, as the data and image wealthy experiment, are detailed in sections 3.3.1 through 3.3.3. However, corresponding performance metrics of Experiment 1 are also compared with those of experiments 2 through 5 in section 3.3.4 to show the accuracy of the developed approach on different experiments and data sets. All the results shown in the following sections are obtained using a desktop computer with a 64-bit Windows 10 operating system, an Intel® Core™ i7-8700k CPU @ 3.70 GHz, 32.0 GBs of RAM, and a Nvidia® Titan V graphical processing unit.

Moreover, the corresponding code files (see Appendix B) were developed in a TensorFlow® environment using Python® programming language.

Table 11- Growth of fatigue crack of Experiment 1 and corresponding manual size measurements

Fatigue Crack Image							
Length	0 $\mu m$	364 $\mu m$	676 $\mu m$	926 $\mu m$	1,234 $\mu m$	1,496 $\mu m$	1,538 $\mu m$

### 3.3.1 Time-Series Analysis Results

AE Features file of the first sensor of Experiment 1 is considered here for analysis. This file consists of 155,650 entries (for AE signals) where about 30,000 signals correspond to pre-initiation stage of the crack and 50,000 entries correspond to crack lengths beyond a failure threshold. Consequently, at first, crack initiation is recognized considering the defined crack initiation threshold (i.e., more than 100 units change in count rate per AE signal) and it is determined that a crack is initiated after 6,920 seconds with an abrupt change of 135 units in AE count per signal (this value is also close to the estimated initiation time (7,320 seconds) based on crack images). Once the crack is detected, post-initiation signals are considered to train the model of time-series analysis module and make crack length estimations. POF model (Eq. 7) is parametrized [ $C=1.160 \times 10^{-5}$ ,  $m=1.282$ ] based on manual and limited NIH ImageJ measurements to generate ground truth values. Moreover, a set of more than 72,200 AE signals with a crack length smaller than failure threshold (1,650  $\mu m$ ) is randomly divided into portions of size [training: 0.6, cross validation: 0.1, testing: 0.3] to be used for LSTM-regression model

training. Through a grid search process on 1,024 possible combinations, optimal structure of LSTM model of time-series analysis module with best performance metrics was obtained. In the grid search process,  $P_{neg}$  of the RUL loss function (Eq. 8) was set to be 100 while parameters  $a$  and  $b$  were set to 1 and 0.5, respectively. Hyperparameters of the optimal LSTM model, namely bi-LSTM\_AA, and grid search intervals are presented in *Table 12*.

*Table 12-Hyperparameters of bi-LSTM\_AA*

LSTM Hyperparameter	Grid Search Values	Value in optimal Structure
Temporal correlation range ( $N_{tw}$ )	[5,10,20,40]	40
Training steps ( $N_{tr}$ )	[100, 200, 300,400]	300
Batch size ( $N_b$ )	[20, 64, 128, 256]	256
Number of nodes ( $N_n$ )	[32,64,128, 256]	128
Dropout probability ( $P_d$ )	[0.3,0.4,0.5,0.6]	0.3

*Table 13- LSTM structures with best performance metrics for different  $N_{tw}$  values*

$N_{tw}$	$N_{tr}$	$N_b$	$N_n$	$P_d$	Cross RMSE	Cross Loss	Test RMSE
5	200	64	256	0.5	0.073	0.550	0.077
5	200	128	128	0.4	0.096	0.235	0.096
10	400	128	128	0.3	0.072	0.358	0.070
10	400	128	32	0.4	0.101	0.874	0.103
20	200	128	128	0.3	0.090	0.419	0.090
20	300	128	128	0.3	0.071	1.274	0.074
<b>40</b>	<b>300</b>	<b>256</b>	<b>128</b>	<b>0.3</b>	<b>0.085</b>	<b>0.088</b>	<b>0.090</b>
40	400	256	128	0.3	0.100	0.099	0.096

To show the effect of consideration of longer temporal correlations on length (size) estimation accuracy, values of hyperparamters of LSTM models with the best performance metrics corresponding to different  $N_{tw}$  values are reported in *Table 13*. Three performance metrics (RMSE of cross validation set, RUL loss of cross validation set, and mean RMSE of 300 random test sets) are presented there. Reported values in *Table 13* (with values of hyperparamters of bi-LSTM\_AA in bold) reveal that, in general, consideration of longer temporal correlations ( $N_{tw}$ ), larger batches ( $N_b$ ), and smaller dropout probability ( $P_d$ ) lead to a

better size estimation accuracy. This observation essentially implies that, in accordance to our expectation, consideration of longer temporal correlations leads to a better sizing accuracy.

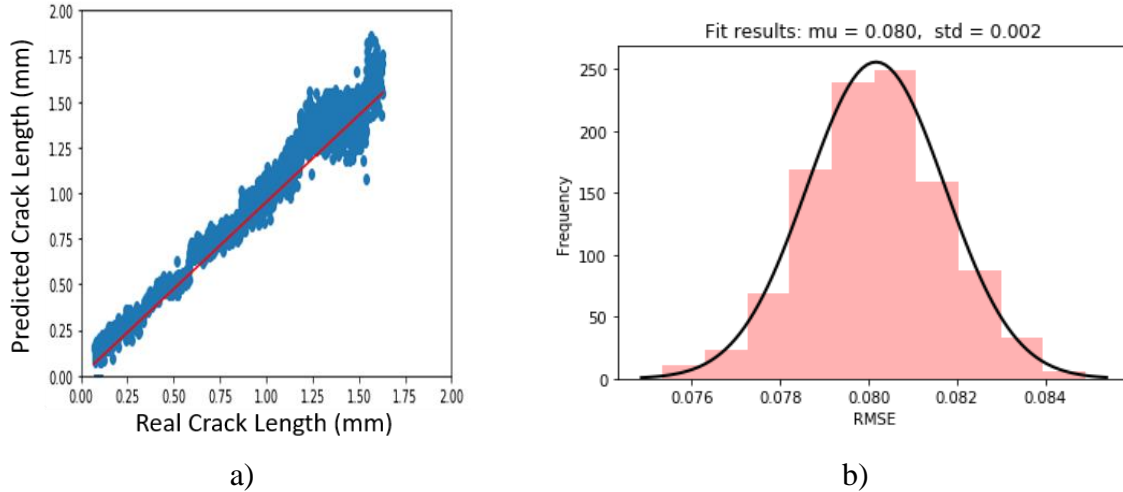


Figure 23- Performance of bi-LSTM\_AA: a) Predicted vs POF estimated (real) length of crack of Experiment 1 in mm. (The solid red line is the identity line based on POF estimations); b) RMSE distribution for a set of 300 test runs

To illustrate the performance of bi-LSTM\_AA model, *Figure 23a* depicts the LSTM vs. POF estimated size values based on the AE features file of Experiment 1. Moreover, *Figure 23b* illustrates the testing RMSE corresponding to predicted crack length values for 300 random samples of Experiment 1, which show acceptable variability in predictions for different sample sets. Finally, it only takes about a second to test a sample of 10,000 AE signals. Hence, online size estimation is feasible using the time-series analysis module of our proposed approach.

### 3.3.2 Image Processing Results

To make image-based automated size estimations on fatigue crack of Experiment 1, a set of 1410 training images, 434 validation images, and 833 testing images corresponding to experiments 2 to 5 were used to train and test different FCDN structures in a grid search process. Considering candidate values of hyperparameters (Table 9), 60 combinations were evaluated in

the grid search to find the optimal values of hyperparameters for the FCDN\_AA model with the best performance metrics and greatest average mean IoU (Eq. 4). Corresponding results are reported in *Table 14*. Reported accuracies represent a true positive probability which is the percentage of correctly classified pixels of each class (crack and background) or the entire image. *Table 14* indicates that the model with the greatest average mean IoU (i.e. FCDN\_AA) has a relatively simple structure ( $N_s = 1$ ) even though more complex structures were also investigated. This observation sounds reasonable considering the bi-class nature of our work while more complex structures have shown promising results for cases with more than five classes of objects (Jegou et al., 2017). Another point about the reported values in *Table 14* is that accuracies are greater than average mean IoU values since misclassified pixels of each class are also considered in the denominator of IoU (Eq. 9).

Segmented images corresponding to sample crack images of Experiment 1 (*Table 11*) are presented in *Table 15*. Reported length values are produced by the pixel-counter unit of image processing module and error values are calculated with respect to corresponding manual NIH ImageJ length estimations (*Table 11*) where a negative error represents underestimation of crack length.







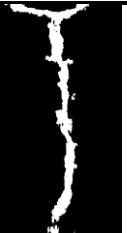
Similar to the time-series analysis module, promising sizing errors and fast testing process (it only takes 7 seconds to produce 100 segmented images) makes the proposed image processing module a good fit for online and accurate damage size estimation. As such, reported length values in *Table 15* are used to parameterize POF model of image processing module ( $[C = 1.198 \times 10^{-5}, m = 1.284]$ ) so that length estimations with a consistent frequency with those of the time-series analysis module are produced to be used for a final and accurate size estimation in data fusion and RUL estimation module.



Table 14- Hyperparameters and performance metrics of FCDN\_AA

Hyperparameter	Optimal Value
Number of epochs	200
Number of sampling modules ( $N_s$ )	1
Number of layers per DB ( $N_l$ )	4
Number of feature maps per DB layer ( $k$ )	8
Training IoU	0.918
Total training accuracy	0.996
Testing IoU	0.810
Total testing accuracy	0.984
Crack testing accuracy	0.946
Test precision	0.982
Test recall	0.983

Table 15- Segmented Images and corresponding size estimations of Experiment 1 by FCDN\_AA

Segmented Images							
Length	0 $\mu m$	380 $\mu m$	647 $\mu m$	924 $\mu m$	1185 $\mu m$	1509 $\mu m$	1547 $\mu m$
Error	0%	+4.3%	-4.2%	-0.2%	-3.9%	+0.9%	+0.6%

### 3.3.3 Data Fusion and RUL Estimation Results

The input of this module includes two crack size estimations of each time instance produced by former modules of the proposed approach. Similar to the time-series analysis module, the input file of this module is also divided into portions of size [training: 0.6, cross validation: 0.1, testing: 0.3] to facilitate training and testing of the proposed bi-directional LSTM model of the current module. In training and testing processes, POF generated ground truth values of time-series analysis module are again used as ground truth (i.e., real size values). Through a grid search process similar to the one in section 3.3.1, the optimal structure of the bi-

LSTM model of this module, namely bi-LSTM\_AAF, is determined and corresponding results are reported in Table 16.

Table 16- Hyperparameters of bi-LSTM\_AAF

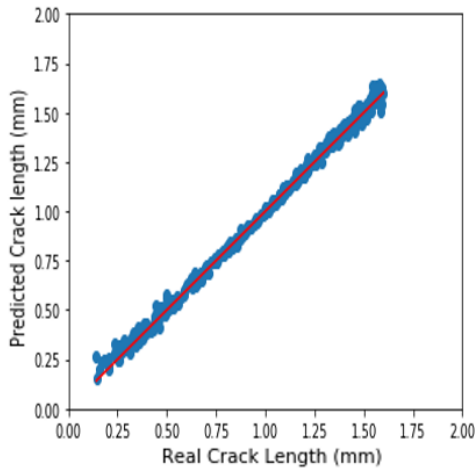
LSTM Hyper Parameter	Grid Search Values	Value in optimal Structure
Time window size ( $N_{tw}$ )	[5,10,20,40]	20
Training Steps ( $N_{tr}$ )	[100, 200, 300,400]	300
Batch Size ( $N_b$ )	[20, 64, 128, 256]	20
Number of Nodes ( $N_n$ )	[32,64,128, 256]	256
Dropout Probability ( $P_d$ )	[0.3,0.4,0.5,0.6]	0.4

LSTM structures with best performance metrics for different  $N_{tw}$  values resulted from the grid search process are reported in Table 17 where values corresponding to the bi-LSTM\_AAF are in bold. Similar to time-series analysis module, Table 17 reveals that in general consideration of longer temporal correlations leads to a better sizing accuracy.

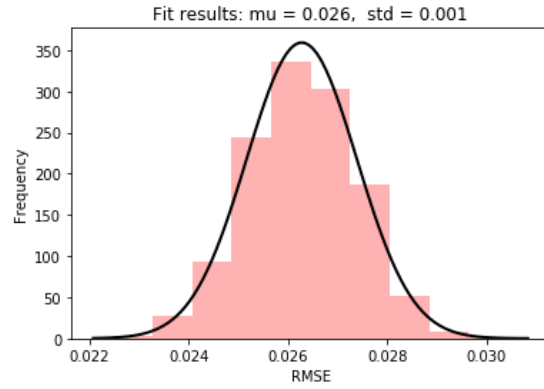
Figure 24 depicts performance of the bi-LSTM\_AAF model on the test portion of the input data of this module. In comparison to the performance of the time-series analysis module, a better prediction accuracy is achieved where the sparsity of estimated size values has significantly reduced in Figure 24a with respect to Figure 23a. Moreover, mean test RMSE for 300 random test samples (Figure 24b) has dropped to 0.026 from 0.080 (mean test RMSE in Figure 23b). The standard deviation is also considerably improved. This significant improvement in performance was expected since data fusion module benefits from frequent sensor data and high accuracy image-based estimates to produce final frequent and accurate size estimates. As such, fused size estimates are favored over the output of both preceding modules. Moreover, fast data fusion analysis (0.87 seconds to produce test size estimates corresponding to a sample of 10000 AE signals) indicates high feasibility of the proposed approach for online damage size estimation.

Table 17- Fusion LSTM structures with best performance metrics for different  $N_{tw}$  values

$N_{tw}$	$N_{tr}$	$N_b$	$N_n$	$P_d$	Cross RMSE	Cross Loss	Test RMSE
5	300	128	256	0.4	0.048	0.035	0.056
5	400	128	128	0.4	0.041	0.032	0.040
10	300	128	128	0.3	0.035	0.032	0.043
10	400	256	256	0.3	0.030	0.028	0.037
20	300	128	128	0.3	0.033	0.033	0.032
<b>20</b>	<b>400</b>	<b>128</b>	<b>256</b>	<b>0.3</b>	<b>0.026</b>	<b>0.025</b>	<b>0.027</b>
40	300	256	256	0.4	0.028	0.035	0.028
40	300	128	128	0.4	0.038	0.039	0.037



a)



b)

Figure 24- Performance of data fusion and RUL estimation module: a) Predicted vs. POF estimated (real) length of crack of Experiment 1 in mm. (The solid red line is the identity line based on POF based real size values) b) RMSE distribution for 300 test samples

Size estimations resulted from data fusion are used to predict RUL considering the assumed failure threshold ( $1650 \mu\text{m}$  of crack length). Figure 25 illustrates predicted RUL values vs. Real RUL for Experiment 1. The discontinuity in graph is because of cooling intervals in Yun's experiments (Yun, 2019) where most of clear crack images were taken and only a few AE signals were transmitted. As Figure 25 depicts, the accuracy of predicted RUL values is acceptable and predicted values are very close to real RUL values at most instances of time.

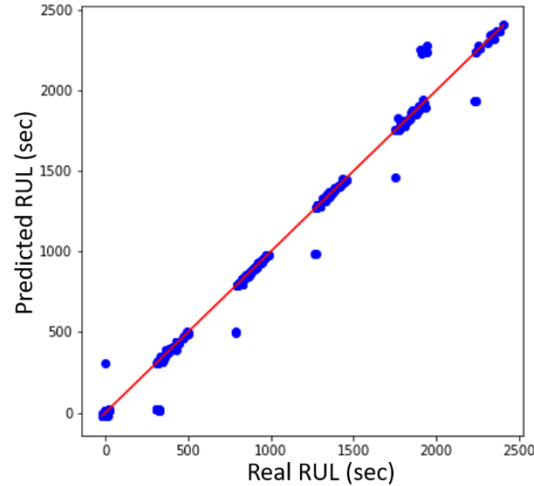


Figure 25- Predicted RUL vs real RUL values for Experiment 1 where real RUL values are calculated based on manual NIH ImageJ measurements and POF model. The red line is the real RUL identity line.

### 3.3.4 Evaluation of Performance for All Five Experiments

As it was discussed in Section 3.2.2 and shown in *Table 10*, variability in AE waveforms and resulted features motivated us to train LSTM models for one single experiment and make final size estimates for the same experiment. As such, results corresponding to Experiment 1 were presented in the previous subsections. Nonetheless, size estimation statistics and errors corresponding to predictions made for all experiments are presented in *Table 18* to demonstrate the accuracy of the proposed approach once applied on different data sets (experiments). Time-series analysis and fusion tuned models of Section 3.3, namely bi-LSTM\_AA and bi-LSTM\_AAF, as well as the tuned image processing model FCDN\_AA are used to produce the statistics of *Table 18*. As such, no new grid search was conducted. Errors are calculated for produced crack length estimates with respect to POF produced ground truth values based on manual NIH ImageJ measurements. Moreover, RMSE is calculated for Modules 1 and 3 using 300 test samples. As *Table 18* indicates, best statistics are not necessarily achieved for

Experiment 1. As an example, automated image-based size estimates (Module 2) have a smaller error for Experiments 2 and 3 which seems reasonable since images from Experiments 2 through 5 were included in the training set of FCDN\_AA. On the other hand, smaller error values corresponding to Modules 1 and 3 are obtained for Experiment 1 as expected. Furthermore, *Table 18* reveals that Module 3 has better error statistics in comparison to those of Module 1 for all experiments in accordance to the discussion in Section 3.3.3 on results of Experiment 1. However, the highest size estimation accuracy for Experiments 2 through 5 is achieved for image processing module, not the data fusion module. This observation can be due to small size features file corresponding to these experiments that provides LSTM models with insufficient data for an accurate training process. As an example, the features file of Experiment 4, the second largest after Experiment 1, has only around 20,000 entries as opposed to that of Experiment 1 with more than 155,000 entries (Section 3.3.1).

*Table 18- Size and error statistics (mean, standard deviation) corresponding to all 5 experiments (in  $\mu m$ )*

Experiment		1	2	3	4	5
Real Size		<b>(929,406)</b>	(714,406)	(903,422)	(862,364)	(757,356)
Module 1 (LSTM)	Size Estimation	<b>(914,406)</b>	(851,469)	(987,403)	(908,378)	(751,346)
	RMSE	<b>(80,2)</b>	(77,2)	(59,3)	(100,3)	(89,2)
Module 2 (FCDN)	Size Estimation	<b>(952,414)</b>	(730,412)	(922,424)	(887,374)	(773,368)
	Absolute Error	<b>(22,9)</b>	(17,7)	(19,9)	(26,10)	(16,5)
Module 3 (Fusion)	Size Estimation	<b>(937,400)</b>	(724,429)	(912,424)	(884,396)	(743,356)
	RMSE	<b>(17,2)</b>	(22,4)	(31,6)	(26,11)	(25,3)

Nonetheless, fusion of automated and less frequent image-based estimates with more frequent sensor-based estimates has still led to improvements for all experiments in terms of producing more frequent size estimates in comparison to automated image-based ones and a better accuracy in comparison to sensor-based estimates. Consequently, generic trends which were discussed in Section 3.3.3 for Experiment 1 are valid for all five experiments and it can be

argued that the proposed approach can be used to predict RUL for different experiments subjected to different loading conditions.

Comparing the obtained performance statistics with those of the existing approaches, it turned out that, to the best of our knowledge, the data fusion approach proposed here outperforms the existing ones. The reported AE-based approaches for damage sizing using data fusion mostly use Bayesian-based techniques and report accuracies less than 80 percent (Eleftheroglou et al., 2018; Rabiei and Modarres, 2013). For damage initiation prediction, fully adaptive particle filtering is used by Rabiei and Modarres (2018), and the reported accuracy is 97.5%. However, considering the RMSE values in *Table 18*, the average accuracy for the proposed approach is 97.6% for damage sizing but not recognition of damage initiation only. Moreover, while the approach proposed here can be considered computationally complex, reported approaches are Bayesian-based and, as a result, are also computationally expensive while being very sensitive to manually-chosen initial values of model parameters (Rabiei and Modarres, 2018). However, less computationally complicated models may obtain results as good as those of the proposed approach. As an example, benefiting from LSTM and FCDN models of the first two modules, a linear fusion model for Module 3 of the proposed approach achieves an average accuracy almost as good as the Module 3 results reported in *Table 18*. Nonetheless, it is expected that the sizing accuracy corresponding to simple models drops when the frequency of high-confidence image-based estimations is decreased or if the data fusion is carried out for degradation processes faster than fatigue crack.

### 3.4. Concluding Remarks

This chapter addresses an important gap in the literature on structural damage size and RUL estimation. It presents a new deep learning based approach for multi-source data fusion for damage size and RUL estimation. The proposed approach considers frequent sensor-based damage size estimates which are produced using a LSTM regression model where long temporal correlations are considered. Moreover, automated but accurate and less frequent size estimates are produced by the proposed approach using a CNN-based semantic image segmentation approach, namely FCDN, where a pixel-wise classification task is carried out on images of a damaged area. Using POF models, the frequency mismatch problem associated with sensor- and image-based size estimates is resolved and corresponding size estimates are fused in another LSTM regression model. Fusion resulted damage size estimates are more accurate with respect to sensor-based estimates and more frequent with respect to image-based estimates. These size estimations are then used to predict the RUL.

Performance of the proposed approach is evaluated using the AE signals and microscopic images corresponding to five fatigue crack experiments under different loading conditions. It is shown that the AE features can be used not only for crack detection and crack growth assessment, as has been reported in the literature, but also for crack size estimation. Moreover, it is shown that computationally reasonable consideration of longer temporal correlations in LSTM models leads to a better accuracy of size estimates needed for RUL estimation. Furthermore, automated accurate crack size estimates were produced using FCDN model. The results show that data fusion provides size estimates with a higher accuracy in comparison to sensor-based estimates. Moreover, it is shown that size estimates resulted from data fusion have

a better accuracy even in comparison to less frequent automated image-based estimates if enough sensor data for an accurate training of LSTM models is available.

The static and dynamic phases of the segment-level of the constructed PHM approach were discussed in the last two chapters. In the next chapter, the approach for the integrated-level is detailed where the results of the segment-level analysis are used as the input.



## Chapter 4: Dynamic and Integrated Health Monitoring of Pipelines

The pipeline's segment-level analysis, as presented in Chapters 2 and 3, estimates RUL for a single pipeline segment based on the damage data gathered using an optimal health monitoring layout for the pipeline segment. However, some modifications to the formulated optimization problem in section 2.4 can improve the scalability of the proposed segment-level layout optimization approach. Doing so, larger pipeline segments, with a larger number of localized damages, can be handled in the segment-level analysis. Results from the segment-level analysis can then be used in the integrated-level analysis to find an optimal set of layouts, including configuration of sensors and human inspection areas, for several pipeline segments. In obtaining the optimal set of layouts, an overall cost constraint and online damage size (or RUL) estimates can be considered. (Note that the frequency of inspection or data sensing is assumed as given: recall inputs of the optimization model discussed in Chapter 2.)

In this chapter, a methodology for solving the problem of integrated and dynamic health monitoring of pipelines is proposed. In the proposed methodology, at first, some modifications are introduced to improve the scalability of the segment-level approach and facilitate layout updates. Then, SDP is utilized to find an optimal set of layouts for pipeline segments of interest. As such, the rest of this chapter is organized as follows: Section 4.1 presents a description of the problem of integrated and dynamic health monitoring of pipelines. Then, Section 4.2 provides an explanation of the modifications made to the segment-level optimization formulation (section 2.4) and discusses the resulting scalability improvement. Next, Section 4.3 discusses the process for updating health monitoring layouts based on online damage size estimates. Following that, Section 4.4 presents a brief overview of SDP and its application for

integrated health monitoring of a pipeline including a case study. Lastly, the chapter is completed with some concluding remarks in Section 4.5.

#### **4.1 Problem Description**

Let us consider a hypothetical pipeline divided into several segments with uniform damage behaviors. In the shadow of online damage size estimates for each individual pipeline segment, layouts corresponding to each of the segments need to be updated accordingly while different budget allocations are explored (see cost constraint in *Figure 3*). Each layout consists of a configuration of multi-type sensors and human inspection along the pipeline segment of interest. Once a layout is updated, the number, location, and type of sensors as well as inspection areas may be changed while some sensors are fixed. An overall cost constraint does not allow the use of individually optimal layouts. Hence, it is desired to find an optimal set of layouts that maximizes average detection probability and average health monitoring utility (including cost) for all segments while allocated segment-wise budgets meet the overall cost constraint. We propose here to utilize SDP for this matter. However, an improved segment-level layout optimization approach, with a better scalability, will be helpful since longer pipeline segments can be handled at the segment level and, as a result, it will be needed to consider fewer pipeline segments in the SDP problem. As such, some modifications are made to the layout optimization approach of Chapter 2. These modifications are discussed next.

## 4.2 Modified Formulation for Improving Scalability

The layout optimization problem presented in Chapter 2 is formulated as a mixed-integer nonlinear problem. The formulation is nonlinear because of the Euclidian distance relation between the damage and detection device (Eq. 6.10). This relation also introduces further nonlinearity in other equations (e.g., Eq. 6.2) and causes the formulated problem to become NP-hard to solve (Salkin, 1975). To get rid of nonlinearities, improve the scalability of the original optimization problem, and solve it using less computationally expensive mixed-integer programming techniques, it is proposed to do the calculations regarding Eq. 6.10 in step 6 of the approach of *Figure 8*. Specifically,  $\mathbf{H}$ , a three-way array, is introduced here for each pipeline segment realization.  $\mathbf{H}$  is a  $M \times N \times N$  zero-one array where  $M$  is the number of detection options,  $N$  is the number of damages in a pipeline realization and the number of nodes in the corresponding graph (see Chapter 2). The quantity  $\eta_{i,j}^m$  denotes an element of  $\mathbf{H}$  and is equal to one if damage  $i$  is detected by detection method  $m$  (if used) at node  $j$ . Accordingly,  $\mathbf{H}_{i,j}$  is a vector with its  $(m+1)^{\text{th}}$  element equal to  $\eta_{i,j}^m$ . Considering this modification, Eqs. 11 and 12 are introduced as modified versions of Eqs. 6.2 and 6.10, respectively. The symbol “o” in Eq. 11 denotes the Hadamard product (see *Table 1* for the definition of other symbols):

$$\Omega'_i = \sum_{m=0}^M \sum_{j=1}^N \sum_{k=1}^Q \{c_{i,k} \times id_{j,m} \times \log(\overline{\theta_{i,j}^{k,m}})\} + \sum_{j=1}^N \log(1 - \mathbf{\Lambda}^j \times \mathbf{H}_{i,j} \circ \mathbf{ID}_j) \quad (11)$$

$$\eta_{i,j} = \sum_m \eta_{i,j}^m \times id_{j,m} \quad (12)$$

Replacing Eqs. 6.2 and 6.10 of the original formulation (section 2.4) with Eqs. 11 and 12, a modified optimization formulation is achieved, which can be solved using mixed-integer optimization techniques. While, in general, mixed integer optimization problems are also known

to be NP-hard (Salkin, 1975), it is shown in the following section that the modified formulation provides a better scalability performance in comparison to the original formulation.

In next few paragraphs, some results are presented to show improvements that can be achieved in computerized runtimes when “size” of the problem is increased for the modified optimization formulation versus the original one (Chapter 2).

Recall section 2.5.1 where 46 synthetically generated pipeline realizations were used with the modified Wilks method. There were four to fifteen damages in each of those realizations. As shown in *Figure 26*, solid black circles denote log of the computerized runtime (in seconds) vs. number of damages for the optimal layout corresponding to each of those pipeline realizations. The runtime values of those 46 realizations correspond to the amount of time it takes to solve the layout optimization problem using the original optimization formulation from chapter 2. On the other hand, six new realizations are generated, under conditions and assumptions similar to those of section 2.5.1, to show the scalability performance of the proposed modified formulation (*Table 19*). The solid red squares in *Figure 26* denote the log of computerized runtime vs. number of damages for each of the six new realizations using the modified formulation. (Results presented in *Figure 26* are obtained using the desktop computer with specifications described in Section 2.5.1.2)

*Table 19- Specifications of new pipeline realizations for evaluation of scalability*

Realization #	1	2	3	4	5	6
Number of damages	9	15	27	49	65	73
Length of segment (m)	50	50	100	200	300	300

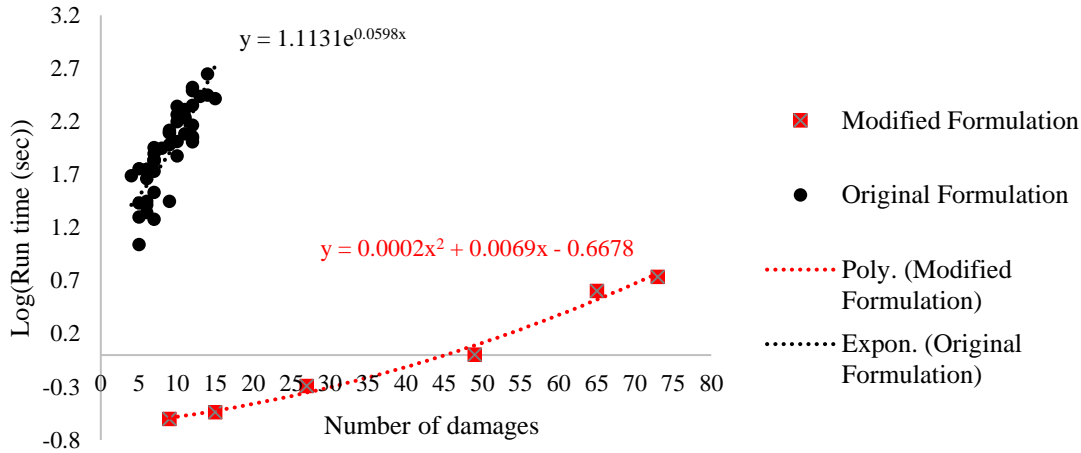


Figure 26- Scalability of the original and modified formulations for layout optimization

As the trend lines in *Figure 26* show, runtime as a function of the number of damages follow an exponential relation for the original optimization formulation (chapter 2) while that of the new formulation follows a second order polynomial. Thus, it can be concluded that, based on these test results, the modified formulation has a better scalability performance and can handle longer and more densely damaged pipeline segments.

Considering the discussion above, from this point on the modified formulation will be used to produce optimal layouts for each pipeline segment of interest.

### 4.3 Dynamic (Online) Health Monitoring

Chapter 2 presented an approach for static (time-invariant) health monitoring layout optimization of pipelines. The work there, in conjunction with the damage size (and RUL) estimates obtained using the approach from Chapter 3, provides support for dynamic (i.e., online) health monitoring of pipelines. In other words, online damage size estimates can be used to

update the configuration (or layout) of damages on the surface of a pipeline and accordingly update the corresponding health monitoring layout (*Figure 27*).

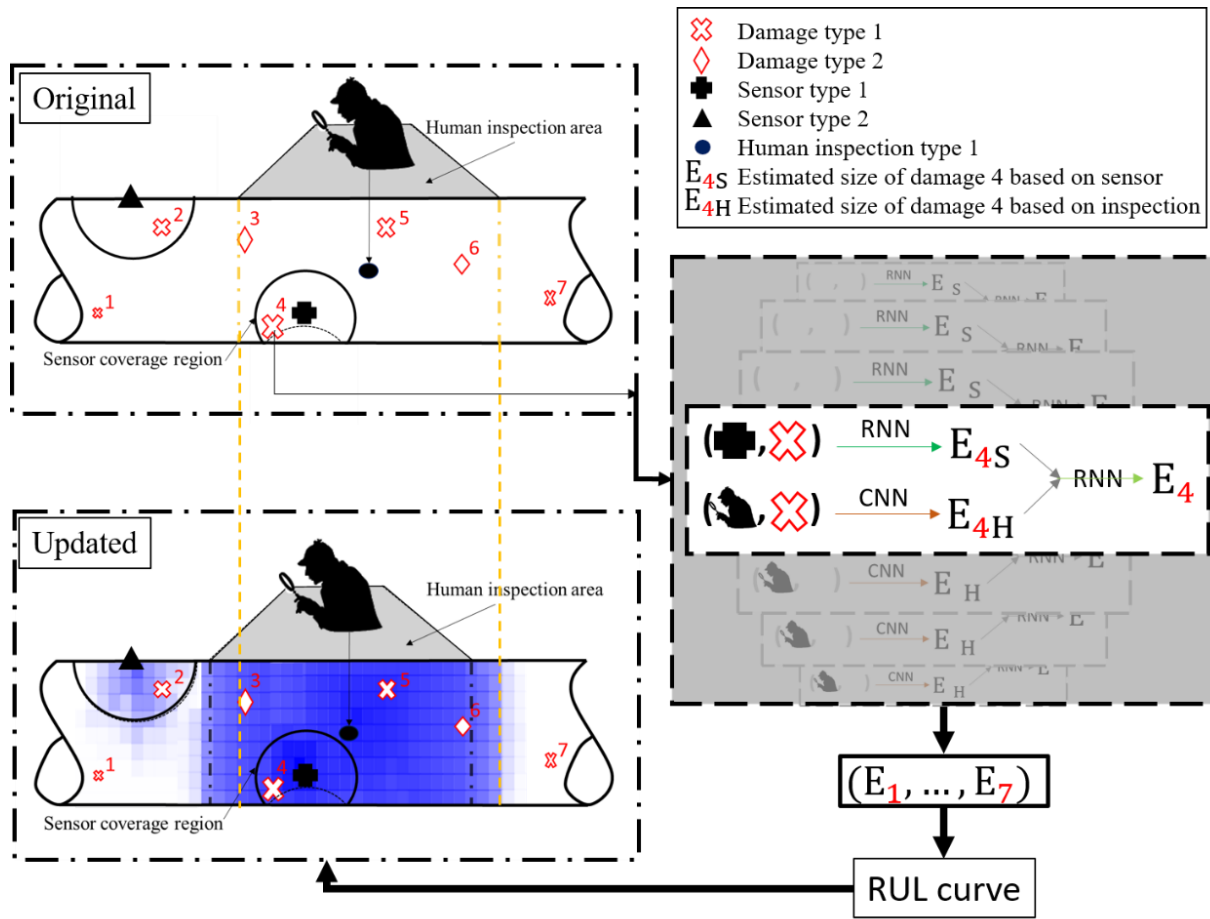


Figure 27- Schematic of layout update process

In *Figure 27*, the top left block, tagged “Original”, schematically show an original layout where sensors and human inspection areas are placed to gather damage data and estimate RUL. It is possible that gathered damage data and RUL estimates for different points of the pipeline surface are not consistent with the original assumptions regarding damage behavior. In that case, it might be needed to change the number and location of sensors, as well as human inspection, in the updated layout (see the “Updated” block in *Figure 27*) if the goal is to maximize confidence in probabilistic detection of damages. (Darker blue areas in the “Updated” block

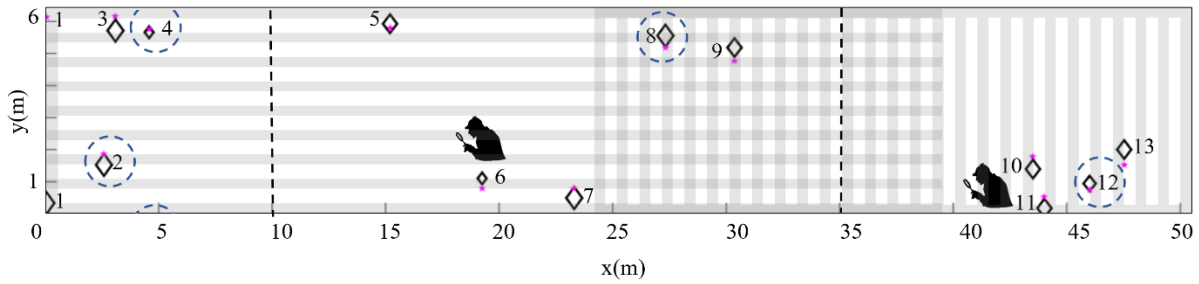
indicate higher confidence in the probabilistic detection of damages.) As an example, while the placement of sensors in the original and updated blocks of *Figure 27* is the same, the human inspection area location is slightly changed in the “Updated” block.

Some additions to the problem formulation of section 2.4 are proposed here to implement this updating process for dynamic health monitoring of pipelines. (Recall the original problem formulation provided in section 2.4 and modified in section 4.2) It is proposed here that for each detection method type to add one constraint to the modified formulation so that some detection methods can be fixed (i.e., during an optimization run), at some graph nodes while the layout is updated in view of the online damage size and RUL estimates. Fixed detection methods are determined according to the original layout and detection considerations. Eq. 13 formulates the constraint for fixing the detection methods where  $id_{j,m}$  is the binary indicator of using detection method type  $m$  at node  $j$  (*Table 1*),  $SS$  is the set of nodes with detection method type  $s$  fixed, and  $N(SS)$  is the number of fixed detection methods (nodes).

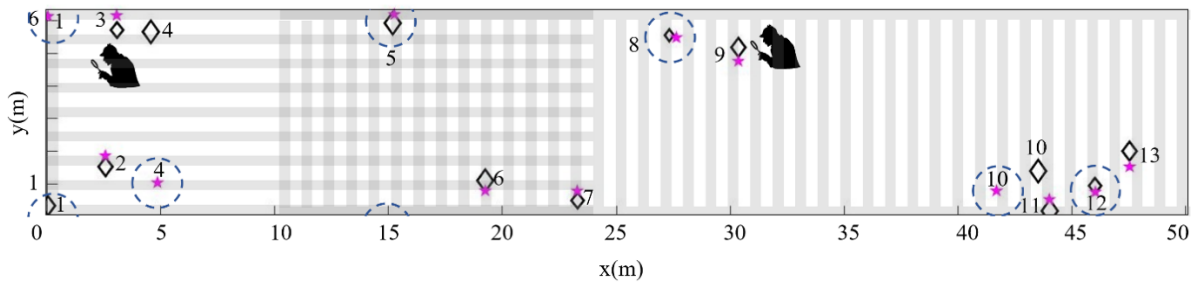
$$\sum_{j \in SS} id_{j,m=s} = N(SS) \quad (13)$$

Adding Eq. 13 to the modified formulation, an optimal health monitoring layout can be obtained considering the online estimated size and location of damages while the detection methods of interest are fixed at corresponding nodes. As an example, suppose that we have used the approach of Chapter 2 during design of a hypothetical pipeline segment of the example in section 2.4. For that pipeline segment, a final static health monitoring layout (*Figure 13*) is obtained considering 46 realizations. Assume that after some elapsed time, damage size estimates, which are obtained using the approach of Chapter 3, show that the current configuration of damages over the internal surface of the pipeline segment is similar to the 22<sup>nd</sup> realization of the example in Section 2.5 (recall *Figure 10*). Hence, it is expected that the

updated layout is similar to the one shown in *Figure 11b* (repeated, for convenience, in *Figure 28a*). However, according to the online damage size estimates, size of diamonds (representing damages with different type-size classes) is bigger for most of the damages in the updated realization of damages (*Figure 28b*). Moreover, the corresponding updated layout should have some detection methods fixed (Eq. 13). AE sensors at nodes [4, 5, 8, 10] are fixed in the updated layout (*Figure 28b*), assuming that the placement of AE sensors in the final static layout (*Figure 13*) cannot be changed. Hence, the layout of *Figure 28b* is slightly different from the layout of *Figure 28a*. The goodness of these two layouts is compared against each other in *Table 20* with results obtained using the desktop computer with specifications described in Section 2.5.1.2.



(a)



(b)

*Figure 28-* (a) *Original optimal layout corresponding to 22nd realization, (b) Updated layout with AE sensors fixed at nodes [4, 5, 8, 10]*



Table 20- Comparison of the performance of layouts of Figure 28

	Ave (Utility)	Ave (Redun.)	Ave (LPOND)	Runtime (s)
Original Layout	0.45	1.46	-8.73	68
Updated Layout	0.37	1.31	-11.48	0.29

Considering the values reported in *Table 20*, adding some and fixing four nodes (AE sensors) in the updated layout (*Figure 28b*) has surprisingly led to a lower value for redundancy of detection in comparison to the original layout (*Figure 28a*). While the original layout is tailored for the exact 22<sup>nd</sup> realization, the final static layout and corresponding fixed AE sensors in the updated layout (*Figure 28b*) are obtained based on a set of 46 realizations. As a result, the above-mentioned performance deficiency of the updated layout could be expected. Nonetheless, a greater density of detection methods in the updated layout has led to a better average LPOND. *Table 20* also confirms that the modified formulation (used to obtain the updated layout of *Figure 28b*) has a much better scalability performance (or optimization run time) in comparison to the original formulation (used to get the original layout of *Figure 28a*).

Note that the frequency (number of times) of inspection is not considered in this research. Hence, damages in more than one inspection area might be inspected at different instances of time. Nonetheless, any changes regarding inspection location in a health monitoring layout can be considered as a guide for further frequency analysis for human inspection planning.

#### 4.4 Stochastic Dynamic Programming (SDP) for Integrated Health Monitoring of Pipelines

SDP (Bellman, 2003) is a technique for decision making under uncertainty. SDP is a stochastic extension of dynamic programming where a set of states (possible values of design variables) is explored at different stages in order to find an optimal policy for a multi-stage

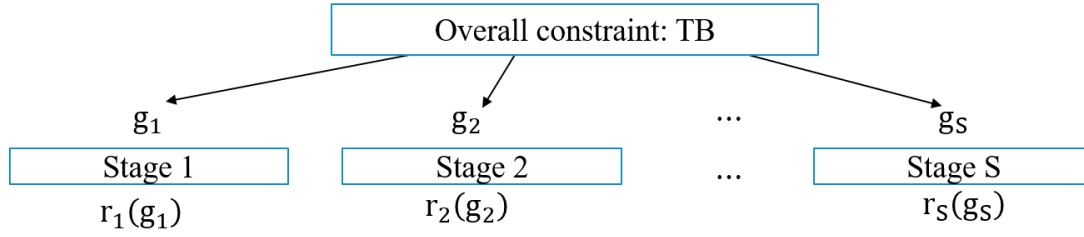
optimization problem. As such, a brief discussion of multi-stage problems and corresponding application of dynamic programming is followed. Next, the difference of dynamic programming and SDP, as its extension, is elaborated. Then, the application of SDP for integrated health monitoring of pipelines is explained using notional examples.

#### 4.4.1 Dynamic Programming

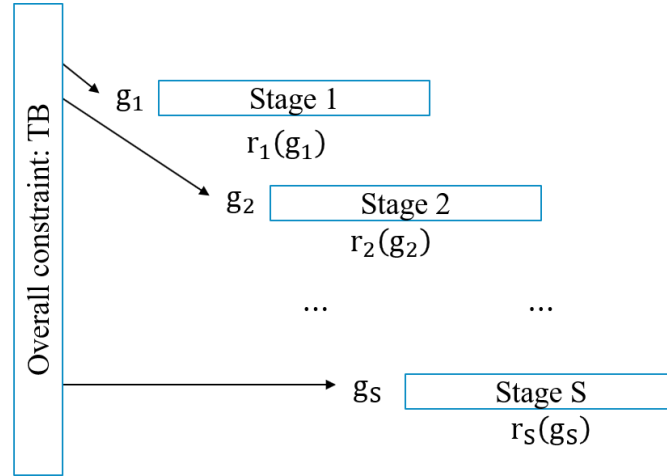
Let's assume we have a system consisting of several sub-units. It is desired to optimize an objective function for the entire system while satisfying a total budget (TB) constraint. As such, one can model the corresponding problem as a distributed optimization problem with  $S$  sub-units (stages), as illustrated in *Figure 29a*. Optimizing the objective function of such a system is equivalent of optimizing sum of objective values for all sub-units (stages) while satisfying the TB constraint (Eq. 14). A formulation for this distributed optimization problem is provided in Eq. 14, where  $TB$  represents the total budget limit,  $(r_i(g_i))$  is the stage-level objective function and  $g_i$  is the stage-level budget. Note that, in Eq. 14, the right hand side value of the stage-level constraints ( $g_i^{\min}$  and  $g_i^{\max}$ ) is different for each stage.

Eq. 14 provides an “all-in-one” solution for the distributed optimization problem at hand and can be computationally expensive to solve, especially in the case of discrete optimization problems, which are quite popular in operations management (e.g., pipeline PHM).

To demonstrate the computational complexity associated with the ‘all-in-one’ formulation for a discrete optimization problem, consider an integer programming optimization problem where the stage-level budget  $g_i$  can only take integer values. As a result, and assuming  $q$  feasible integer budget values for each stage, it may be necessary to explore  $q^S$  combinations for a problem with  $S$  stages to find an optimal solution using the all-in-one formulation.



a)



b)

Figure 29- a) General schematic of a multi-stage optimization problem, b) a sequential solution schematic of a multi-stage optimization problem

$$\begin{aligned}
 & \text{Max}_{g_1, g_2, \dots} r_1(g_1) + r_2(g_2) + \dots + r_s(g_s) \\
 & \text{s. t.} \\
 & \quad g_1, g_2, \dots, g_s \in Z \\
 & \quad g_1 + g_2 + \dots + g_s \leq TB \\
 & \quad g_1 \leq g_1^{\max} \\
 & \quad -g_1 \leq -g_1^{\min} \\
 & \quad g_2 \leq g_2^{\max} \\
 & \quad \vdots \\
 & \quad -g_s \leq -g_s^{\min}
 \end{aligned} \tag{14}$$

On the other hand, due to the existence of an overall constraint (TB in *Figure 29a*) the optimization problems for stages cannot be solved separately to parallelize the computations and reduce the computational cost. Nonetheless, such a discrete optimization problem can be

modeled as a multi-stage one and stage-level optimization problems can be solved consecutively to reduce the computational cost while satisfying the TB constraint (*Figure 29b*). This way, considering the formulation provided in Eq. 14,  $r_i(g_i)$  would be the reward (objective) function of the stage-level optimization problem for stage  $i$  and  $g_i$  is the stage-level budget, which must be within a feasible interval  $[g_i^{\min}, g_i^{\max}]$  for each stage  $i$ . Dynamic programming offers a computationally manageable way to solve this discrete and multi-stage optimization problem.

In dynamic programming (Bellman, 2003), the optimization problem is separately solved for each stage, and different stages are related to each other considering the total budget limit and through a cumulative state value (i.e., cumulative spent budget) which is fed into a recursive reward function (See Section 4.4.1.2 for more details). In this way, the problem is solved backward where the “Principle of Optimality” is assumed to hold. The Principle of Optimality is a form of a Markov property, where it is assumed that the current state of the system (at the current stage) is independent of the preceding states. Following this, the optimization problem of the last stage is solved first in dynamic programming. Each possible reward of the last stage problem is then fed into the second to the last stage problem using a recursive reward function. At each stage, the total budget constraint determines the feasible paths (i.e., a set of reward and state variable values for the last to the current stage) and only feasible paths with a semi-optimal reward value are explored (See section 4.4.2.1 for more details). As such, the number of combinatorial cases needed to be explored is reduced to  $q+(S-1) \times q^2$  in comparison to  $q^S$  combinatorial cases of the all-in-one formulation. The process continues in dynamic programming until the first stage is reached. Following the solution of the first stage problem, the optimal solution is obtained while a computationally tractable number of combinatorial cases has been considered and total budget constraint is satisfied.

As it was mentioned earlier, SDP is a stochastic extension of dynamic programming. The only difference between SDP and dynamic programming is the consideration of uncertainty in calculation of the reward function of each stage ( $r_i(g_i)$  in Eq. 14). In other words, the expected reward, corresponding to different scenarios for the stage-level problem, is considered for each stage in an SDP problem as opposed to dynamic programming, where a deterministic reward is considered.

The next section details an application of SDP to integrated health monitoring of pipelines where three notional examples are considered and demonstrated using SDP.

#### **4.4.2 Integrated Health Monitoring of Pipelines using SDP**

It is desired to find an optimal set of health monitoring layouts for a set of pipeline segments which are subject to different operational conditions and follow different degradation behaviors. As such, the process of finding an optimal set of layouts is called “integrated” health monitoring of pipelines. For a hypothetical pipeline, assume that only low and high-risk failure scenarios are to be considered. Considering two major elements of the risk associated with an event (i.e., the frequency and consequences of an event), it is assumed that the risk of having a failure at each pipeline segment is correlated with the density of localized damages. In other words, it is assumed that having a failure has the same consequences regardless of the localized damage and the frequency of failures is the major affecting factor of the risk of failure. As such, high-risk scenarios have a higher density of localized corrosion and require a greater budget if achieving a particular detection percentage is intended.

As it was stated earlier, the goal here is to find an optimal set of health monitoring layouts for different segments of a pipeline and it is proposed to use SDP for this matter. The key

assumption here is that damage behavior (i.e., density, size, and type of localized damages) in each segment of the pipeline is independent of all other segments considering reported sparsity of localized damages along a pipeline (Lewandowski, 2002; Valor et al., 2015). Hence, the principle of optimality is applicable, and an SDP problem can be formulated to find an optimal set of health monitoring layouts. As such, details of the formulated SDP problem is provided in the following section.

#### **4.4.2.1 Integrated Pipeline Health Monitoring: SDP Problem**

For a set of pipeline segments with a pre-specified total budget and detection percentage limits, the goal is to find an optimal set of health monitoring layouts. Each optimal layout includes a configuration of multi-type sensors and human inspection tools and areas. According to the approach of Chapter 2, given a cost/budget constraint for each pipeline segment, an optimal layout with a weighted sum combination of average LPoND and health monitoring utility (including the utility of cost) can be determined. However, given the online damage size estimates, different damage growth and density scenarios (with different risks of failure) can be considered for each segment. Moreover, different budget allocations are possible for different pipeline segments considering the total budget. Thus, a set of feasible optimal layouts can be obtained for each segment considering different damage scenarios and budget allocations.

For each pipeline segment, feasible optimal layouts are those that meet detection and cost constraints. For instance, consider a damage scenario with ten localized damages of different sizes along a 50-meter pipeline segment. Assume that the cost metric value of a single AE sensor and human inspection is set to be 1 and 10, respectively. With a cost constraint of 12 and a 50% minimum detection, a layout with three sensors and one human inspection (cost metric value of:

$3 \times 1 + 10 = 13 > 12$ ) is not feasible. A layout with two sensors is not feasible either since the required minimum detection cannot be achieved. Feasible optimal are fed into the SDP problem to find an optimal set of layouts. The stages and states of such an SDP problem should be defined first.

In order to use SDP for integrated pipeline health monitoring, each pipeline segment is considered and formulated as a stage of an SDP problem. Additionally, the cumulative discretized cost metric of the last to the current segments is considered as the state variable  $x_i$  at stage  $i$ . The cost metric  $c_i$  represents the cost of an optimal layout for the stage  $i$  which is obtained considering a particular budget allocation  $g_i$  (i.e., cost constraint) and a detection limit. Feasible optimal layouts corresponding to different values of state variable  $x_i$  are determined and generated for each pipeline segment using the approach of Chapter 2. Hence, the reward function for each segment (i.e., stage) is the negated objective function of the segment-level problem (Eq. 6.1) to be maximized. To account for detection uncertainty and size estimation error, damage scenarios of different risk levels are considered for each pipeline segment in the proposed SDP formulation. Once all feasible optimal layouts for all segment-risk scenario pairs are obtained, an overall cost constraint (TB in Eq. 14) is considered in the SDP problem to find an optimal set of layouts with a minimum average LPoND and a maximum average health monitoring utility. Admittedly, this optimal set of layouts is consistent with TB.

$$R_i(x_i) = \max_{c_i, g_i} \{R_{i+1}(x_i - c_i) + r_i(g_i)\} \quad (15.1)$$

$$r_i(g_i) = P_l \times r_l(g_i) + P_h \times r_h(g_i) \quad (15.2)$$

$$\begin{aligned} x_i &= c_i + c_{i+1} + \dots + c_s \\ c_i &\leq g_i \end{aligned} \quad (15.3)$$

Variables used in the formulated SDP problem are described in *Table 21* and the major equations for each stage problem are provided in Eqs. 15.1 through 15.3. However, the equations needed to solve the SDP problem in a step-by-step manner are presented in the following paragraphs, along with their descriptions. The stage recursive reward  $R_i(x_i)$  (Eq. 15.1) is defined considering the recursive relationship given by Eq. 15.2.  $R_i(x_i)$  is used to relate different stages of an SDP problem, as illustrated in *Figure 30*, and calculate the reward value corresponding to each path. Given high- and low-risk damage scenarios for each segment, the expected reward function  $r_i(g_i)$  is defined in Eq. 15.2 for each stage  $i$ . Lastly, state variable  $x_i$ , corresponding to stage  $i$ , is defined in Eq. 15.3.

*Table 21- The nomenclature for SDP problem*

<b>Symbol</b>	<b>Name</b>	<b>Description</b>
$c_i$	Cost metric for segment $i$	Cost metric for the optimal layout of segment $i$ given $g_i$
$g_i$	Budget limit for segment $i$	
$i$	Segment index	
$k$	State value index	
$P_l$	Low-risk likelihood	The associated likelihood of the low-risk damage scenario
$P_h$	High-risk likelihood	The associated likelihood of the high-risk damage scenario
$r_i(g_i)$	Expected reward for segment $i$	Expected objective value for a given budget limit $g_i$ at segment $i$
$r_l(g_i)$	Low-risk reward	Objective value of layout optimization problem (Eq. 6.1) for low-risk damage scenario subject to cost constraint $g_i$
$r_h(g_i)$	High-risk reward	Objective value of layout optimization problem (Eq. 6.1) for high-risk damage scenario subject to cost constraint $g_i$
$R_i(x_i)$	State/reward value for segment $i$	Maximum expected objective value for given state variable $x_i$
$S$	Number of segments	
TB	Total budget limit	Budget limit in the SDP problem
$x_i$	State variable for segment $i$	Sum of the cost metrics from current segment $i$ to the last segment

As mentioned earlier, the calculation process in an SDP problem starts at the last stage (segment  $S$ ). Thus, at first the feasible set of the discrete state variable  $x_s$  is determined for the



last segment S (Eq. 15). Next, the stage reward  $R_s(x_s)$  is determined for feasible values for the state variable  $x_s$ , which is determined considering discrete cost constraint ( $g_s$ ) values, as is indicated in Eq. 16. Note that the sign  $\mathbf{A}$  in Eq. 16 denotes mathematical correspondence.

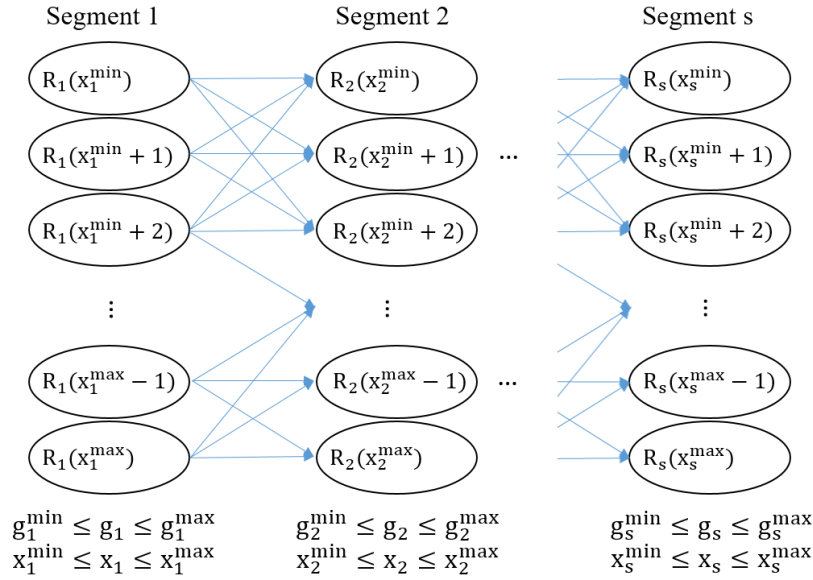


Figure 30- Schematic of the SDP problem

$$\begin{aligned}
 & c_s^{\min} \leq x_s \leq c_s^{\max}, x_s \in \mathbf{N} \\
 & c_s^{\min} \mathbf{A} g_s^{\min}, \dots, c_s^{\max} \mathbf{A} g_s^{\max} \\
 & R_s(x_s) = r_s(g_s)
 \end{aligned} \tag{16}$$

Once feasible set of state values  $x_s$  and corresponding reward values are determined for the last segment, the process determines the feasible state values for the second to the last segment. For this segment and all other segments  $i$ , feasible values of state variable are determined by Eq. 17 where TB denotes the overall cost constraint (i.e., total budget).

$$\begin{aligned}
x_i^{\min} &\leq x_i \leq x_i^{\max}, x_i \in \mathbb{N} \\
x_i^{\min} &= \sum_{j=i}^s c_j^{\min} \\
x_i^{\max} &= \min\left(\sum_{j=i}^s c_j^{\max}, TB - \sum_{j=1}^{i-1} c_j^{\min}\right) \\
c_j^{\max} &A g_j^{\max}, \dots, c_j^{\min} A g_j^{\min}
\end{aligned} \tag{17}$$

Next, for each feasible value of the state variable  $x_i$ , all possible paths from the state  $x_i$  (at stage  $i$ ) to the state  $x_{i+1}$  at next stage  $i+1$  should be listed. Considering detection limits, there is an allowable range for cost constraint  $g_i$  for each segment  $i$  and the following constraint (Eq. 18) must be met in solution of the SDP problem.

$$g_i^{\min} \leq x_i - x_{i+1} < g_i^{\max} \tag{18}$$

Once all possible paths are listed for each feasible state value at stage  $i$ , the recursive reward value, is calculated (Eq. 15.1). The maximum of recursive reward is noted as the state value. As such, the path with the state value (i.e., maximum reward value) and the corresponding path are stored for each value of state variable.

For all paths, i.e., different combinations of state values and stages, the recursive reward value  $R_i(x_i)$  can be calculated through the process discussed above. Once the first segment is reached, all reward values are compared and the optimal value for the SDP will be attained (Eq. 19).

$$\text{Obj}_{\text{SDP}} = \max_{x_1} (R_1(x_1)) \tag{19}$$

Since all paths for different state values at all stages are stored, corresponding segment-wise cost constraints and state values (i.e., cumulative cost metrics) of the optimal path can be

easily found. Some notional examples are discussed in the following section for a better and more descriptive illustration of the application of SDP for integrated pipeline health monitoring.

#### 4.4.2.2 Illustrative Case Study

A short case study is presented in this section. Three 50-meter segments of a locally corroded pipeline with a one-meter radius are considered here. All damage data are synthetically generated following the approach of section 2.5. It is assumed that these pipeline segments are subjected to low and high-risk scenarios, with the likelihood of occurrence of 0.4 and 0.6, respectively. The segments are considered as the three stages of an SDP problem. Similar to section 2.5, it is assumed that health monitoring layouts can include AE sensors (cost metric = 1) as well as human inspection with an ultrasonic tool (cost metric = 10). Feasible layouts are considered to be those with at least 30% detection and a smaller-than-one average detection redundancy for high-risk scenarios. High- and low-risk scenarios for all three segments, and corresponding upper and lower cost constraints are presented in *Table 22*. It is assumed that damage configuration in each low-risk scenario is a subset of high-risk scenario of the same segment. Moreover, cost constraints are determined considering the above-mentioned detection limits and number of damages in low and high-risk scenarios.

*Table 22- Number of damages and budget limits for each stage of the SDP problem*

Segment # ( <i>i</i> )	Low risk Scenario	High Risk Scenario	Cost Constraint
1	10 damages	15 damages	$5 \leq g_1 \leq 15$
2	7 damages	9 damages	$3 \leq g_2 \leq 9$
3	9 damages	13 damages	$5 \leq g_3 \leq 13$

In this case study, three problems with different overall cost constraint values are solved: TB = 13 (minimum detection); TB = 27 (moderate detection), and TB = 37 (maximum

redundancy). Let's start with the problem with  $TB = 27$  as an example and discuss the results corresponding to all three cases in the next section. It is assumed that the weight of the objective function of layout optimization problem ( $w_{obj}$  in Eq. 6.1) is set to be 0.5. The procedure to determine cost metric of optimal layouts considering cost/budget limits of each segment is described as follows.

First, the expected reward values ( $r_i(g_i)$ ) are obtained for each segment using the approach of Chapter 2 and given the corresponding cost constraints. The process starts with segment 3. As *Table 22* indicates, segment 3 is considered in the last stage of the SDP problem. Optimal layouts corresponding to all integer values of  $g_3$  in the interval  $[5 \ 13]$  are obtained. Since budget limit values are determined considering detection constraints, the constraint on the value of LPOND (Eq. 6.21) is relaxed. As a result, the budget limit constraint will remain active in all the problems ( $g_3 = c_3$ ) and the state variable  $x_3$  can have any integer value in the interval  $[5 \ 13]$ . Considering the feasible values of  $x_3$ , feasible reward values ( $R_3(x_3)$ ) are determined.

$$x_3 = c_3 \in \mathbb{N}, 5 \leq c_3 \leq 13$$

$$R_3(x_3) = r_3(x_3)$$

Following the same procedure and considering Eq. 17, the feasible sets of the state variable at the other two segments are then calculated:

$$x_2 = c_3 + c_2$$

$$x_2 \in \mathbb{N}, 8 \leq x_2 \leq 22$$

$$x_1 = c_3 + c_2 + c_1$$

$$x_1 \in \mathbb{N}, 13 \leq x_1 \leq 27$$

*Figure 31* shows the stages and states of the formulated SDP problem. Once the feasible sets of the state variable are determined for all the stages, corresponding state values (i.e., maximum recursive reward) at each stage are calculated. To do that, the possible paths from current stage to the next stage are listed considering different values of state variable  $x_i$ . For

example, assume that  $x_2 = 15$  (Figure 32). For the stage 2 problem,  $c_2$  can have any value in the range  $[3, 9]$  considering lower and upper cost constraints and Eq. 18. Thus, the possible paths  $(x_2, x_3)$  would be:  $[(15, 6), (15, 7), \dots, (15, 12)]$ . All these paths are illustrated in Figure 32. Among all the feasible paths with  $x_2 = 15$ , the path with the maximum reward ( $R_2(15)$ ) is determined using Eq. 15.1. It turned out that the path  $(x_2, x_3) = (15, 8)$  has the greatest reward value (Figure 32). Thus, the state value  $R_2(15)$  is 1.57.

The above procedure has been used to determine state values for segments 1 and 2 considering feasible values of state variable  $x_i$ . Thus, the optimal solution for the case with  $TB = 27$  is obtained to be: state variables  $(x_1, x_2, x_3) = (27, 19, 12)$ ; cost metrics  $(c_1, c_2, c_3) = (8, 7, 12)$ . This result and those corresponding to other values of  $TB$  are elaborated and discussed in the next section.

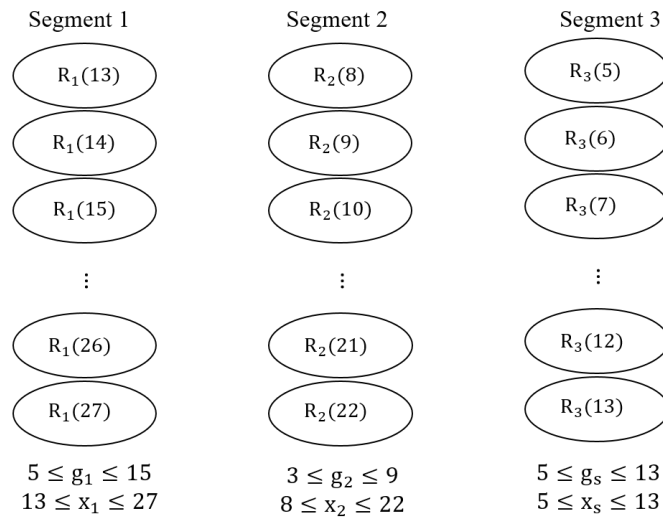


Figure 31- States and stages for the SDP problem for pipeline health monitoring

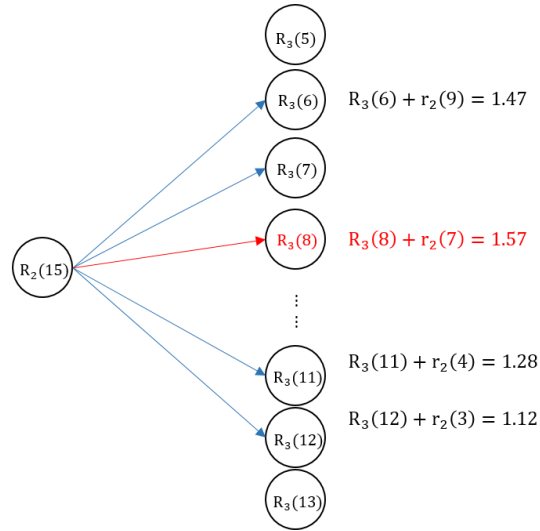


Figure 32- Possible paths for state  $x_2 = 15$  and the optimal path

#### 4.4.2.3 Results and Discussion

The formulated SDP problem is solved using a MATLAB® 2016 code (see Appendix C), where for each stage, different paths and their corresponding expected reward values are calculated given the upper- and lower-cost constraints at each segment. Following the same problem setup as described in the previous section, the SDP is solved for the overall cost constraint (TB) equal to 13 and 37 in addition to 27. The case  $TB = 13$  is chosen considering the sum of the minimum required budget for all three stages to meet the minimum detection limit (30% detection). Additionally,  $TB = 37$  corresponds to the sum of upper cost constraints corresponding to the case with average detection redundancy of one for all three segments. Finally,  $TB = 27$  is considered as an in-between value for overall cost constraint which allows having a human inspection for two segments at most while assigning sensors to the layouts of all three segments.

For the three different values of TB, different objective function weight ( $w_{obj}$ ) values (i.e., 0.2, 0.4, 0.5, 0.7, and 0.9) are considered and corresponding Pareto results are attained

(Table 23). It can be observed in Table 23 that for TB = 13, the minimum required number of the sensors are assigned to the layouts of all three segment and results are not changed with the weight as long as the optimal solution coincides with the only feasible design point.

For two other cases (TB= 27, 37), results change with the  $w_{obj}$ . As an example, let us consider the case with TB =37. For two cases with  $w_{obj}$  equal to 0.2 and 0.7, LPOND and cost metric values corresponding to layouts of the optimal set corresponding to TB =37 are presented in Table 24. One can see that when the weight is increased to 0.7, as a result, LPOND is favored over utility (which considers cost). Thus, the latter optimal set of layouts (corresponding to  $w_{obj}$  = 0.7) is found to be one with better average LPoND value and higher cost. In this set of layouts, the layout for 1<sup>st</sup> segment might be considered as a sub-optimal answer since the health monitoring cost is increased unnecessarily. However, this observation can be easily justified since the optimal set of layouts, with a maximum average utility and a minimum average LPoND, can include individually sub-optimal layouts of different segments.

Table 23: Optimal layout for each segment for different LPOND weights (0.2, 0.4, 0.5, 0.7 and 0.9) and total budget limit of (13, 27, 37).

Total Budget	13					27					37	
	0.2	0.4	0.5	0.7	0.9	0.2	0.4	0.5	0.7	0.9	0.2	0.4
1 <sup>st</sup> segment	5	5	5	5	5	13	9	8	8	10	13	15
2 <sup>nd</sup> segment	3	3	3	3	3	7	7	7	7	8	9	9
3 <sup>rd</sup> segment	5	5	5	5	5	7	11	12	12	9	11	13
Obj. value	1.26	1.08	0.96	0.84	0.81	2.37	2.25	2.37	2.46	2.58	2.61	2.64

Total Budget	37		
w <sub>obj</sub>	0.5	0.7	0.9
1st segment	15	15	[10, 11,12,13,14]
2nd segment	9	9	9
3rd segment	12	13	12
Objective value	2.67	2.82	2.64

Table 24: LPOND for the case with budget limit =37

Total Budget		37		
Wobj	0.2		0.7	
	LPOND	Cost	LPOND	Cost
1 <sup>st</sup> segment	-13.47	13	-13.89	15
2 <sup>nd</sup> segment	-9.95	9	-10.25	9
3 <sup>rd</sup> segment	-12.90	11	-13.93	13
Average	-12.10	11	-12.35	12.33

There is one other observation from *Table 23* which is worth discussing here. Note the unexpected set of results for the case with  $TB = 37$  and  $w_{obj} = 0.9$ . This observation can be justified in the shadow of definition of the LPOND and utility in the problem formulation of section 2.4. There, ME is one of the elements considered in the definition of utility. On the other hand, POD is considered in LPOND. In the particular case at hand, AE sensors have a higher ME and smaller POD values in comparison to human inspection. Thus, even though adding sensors provides us with a slightly greater detection percentage, significantly high measurement error cancels out corresponding improvements in the objective function resulted from a better average LPOND. Thus, the objective function value remains the same for several solutions with different numbers of sensors.

#### 4.5 Concluding Remarks

An extension to the formulation from chapter 2 is considered here. This extension addressed the problem of dynamic and integrated health monitoring of pipelines. The original approach (Chapter 2) is developed to work for a single pipeline segment. However, the extended approach allows us to do integrated health monitoring for several segments of a pipeline using SDP. In the extended approach, the problem formulation of the original approach is first modified so that the problem can be solved using mixed-integer programming. Hence, as it is



shown, the modified approach has better scalability in comparison to the original approach. Additionally, a set of constraints are added to the modified formulation so that the optimal health monitoring layout of each segment can be updated given online damage size estimates and with some detection methods fixed. Following that, the extended approach utilizes SDP to make optimal decisions for dynamic and integrated health monitoring of pipelines.

In order to utilize SDP, two damage scenarios are generated for each pipeline segment and corresponding objective values are obtained to form the feasible region of the SDP problem. In the proposed SDP approach, the layouts of each segment of a pipeline are considered to find an optimal set of layouts considering the detection limits and the limited total budget. In order to evaluate the performance of the extended approach, different budget allocations are considered in a case study, where results are obtained for different objective function weight values to account for the subjective importance of the utility and LPoND as the elements of the objective function of the original approach (Chapter 2). Additionally, it is shown that the optimal set of layouts, with maximum average utility and minimum average LPoND, may include individually sub-optimal layouts.

A summary of the dissertation delivered contributions, and some future directions are presented in the next chapter.

## **Chapter 5: Conclusions, Contributions, and Potential Future Directions**

An approach for dynamic and integrated PHM of locally damaged pipelines is developed in this research. This approach has two main levels: segment and integrated. The segment-level includes two phases, static and dynamic. Moreover, it assumes in the segment-level analysis that a segment has near identical spatial, geometric, and material properties as well as operational conditions. Therefore, damages are expected to be distributed uniformly along a segment. The second level (integrated-level) is for dynamic and integrated health monitoring of the entire pipelines, where a system of pipeline segments is considered. A summary of the approach of each of these levels is provided in the next section. Following that, major contributions of this research are discussed in Section 5.2. Lastly, some potential directions for future research projects are presented in Section 5.3 considering the limitations of the presented work in this dissertation.

### **5.1 Summary of the Approaches**

A summary of each of the approaches developed for the levels and phases of the proposed PHM approach is followed.

#### **5.1.1 Summary of the Approaches for the Static and Dynamic Phases of the Segment-Level Analysis**

The segment-level approach consists of two phases; a static and a dynamic phase. In the static phase, an optimal health monitoring layout for a pipeline segment is obtained to gather damage-related data. The online network of sensors and offline human inspection methods, as popular data gathering schemes (or damage detection methods), are considered there. The

corresponding static phase approach has several important features. (i) It uses three metrics (namely, POD, IP, and ME) for probabilistic detection of damages with different levels of severity. (ii) It is based on an optimization objective function that uses a weighted sum of two functions; a health monitoring utility function and a function for probabilistic detection of damages. By changing the weights for these functions, the analysis explores different optimized layout solutions, including tradeoffs between the two functions. (iii) It considers a significant number of key attributes of the detection methods, such as detection cost, coverage capability, data acquisition frequency, and measurement error. (iv) It uses probabilistic sampling methods for simulating and random placing of localized damages along a pipeline surface considering relevant available data. In this way, not only the expected value of damage specifications (e.g., damage location and size) can be considered, but also their variations can be accounted for. Applications of the approach for optimal and static health monitoring of a pipeline segment are given and discussed in Chapter 2. Considering different levels of severity for different localized damages, it is shown that the developed approach provides a better solution for segment-level health monitoring of pipelines in comparison to available approaches that rely on deterministic damage data and/or only consider one detection method. However, the static phase approach is tailored for uniform localized degradation. Hence, modifications may be needed if PHM of pipeline segments with non-uniform localized degradation is intended.

The second phase of the segment-level approach is the dynamic phase. This phase mostly includes data fusion for damage size and RUL estimation. In the approach of this phase, LSTM-regression models and a CNN semantic image segmentation method are used to construct automated damage size estimates based on sensor data and images of the damaged area. Final damage size estimates are obtained through fusing sensor- and image-based size estimates. An

online RUL estimation of the structure is then obtained accordingly. This approach is detailed in Chapter 3, and its performance is evaluated using the AE signals and microscopic images corresponding to five fatigue crack experiments under different loading conditions. It is shown there that the AE features can be used not only for crack detection and crack growth assessment, as has been reported in the literature, but also for crack size estimation. Moreover, it is shown that computationally reasonable consideration of longer temporal correlations in LSTM models leads to better accuracy of damage sizing needed for RUL estimation. Furthermore, automated and accurate crack size estimates were produced using CNN-based image processing. The results presented in Chapter 3 show that data fusion provides size estimates with higher accuracy in comparison to sensor-based estimates and a better accuracy in comparison to less frequent automated image-based estimates, if enough sensor data for better training of LSTM models are available. It is worth mentioning that it turned out in the training process that, as opposed to multi-object segmentation tasks, FCDNs with a relatively simple structure show promising results once used for damage sizing (where damage and background are the only two classes of objects). Moreover, it was determined that one LSTM model should be trained for each experiment with AE sensor data to achieve higher damage sizing accuracy.

### **5.1.2 Summary of the Integrated-Level Approach**

The second level of the developed PHM approach (integrated-level) is for dynamic and integrated health monitoring of pipelines. The corresponding approach and results were elaborated and discussed in Chapter 4, where the problem formulation of the original approach (Chapter 2) is first modified to improve the scalability. Additionally, a set of constraints are added to the modified formulation so that the optimal health monitoring layout of each segment

can be updated, given online damage size estimates and some detection methods fixed. In the integrated-level approach, at first online damage size and RUL estimates, attained using the approach of Chapter 3, are fed into the modified optimization formulation. As such, the primary layouts, obtained using the static segment-level approach (Chapter 2), are updated while a set of detection methods are fixed, and different cost constraints are explored.

Next, a set of optimal health monitoring layouts is obtained for each pipeline segment. These layouts are then fed into the formulated SDP problem to find an optimal set of layouts for pipeline segments of interest, considering an overall cost constraint. In the formulated SDP problem, it is assumed that high- and low- risk failure scenarios are likely. Moreover, a cumulative health monitoring cost metric is considered as the state variable. The average probability of detection of all damages over all segments of interest and average utility are considered in the state value (i.e., objective function) of the SDP problem, where expected reward corresponding to low- and high- risk scenarios is considered. A case study, including three notional examples, is used there to demonstrate the performance of the developed approach. Pareto solutions corresponding to different trade-offs between two functions of the objective function are explored there. It is shown that the optimal set of layouts may include individually sub-optimal layouts. It is also demonstrated that finding an optimal health monitoring layout for a long pipeline segment, with hundreds of localized damages, can be computationally expensive. However, finding an optimal set of layouts for an equivalent system of pipeline segments is shown to be computationally manageable, once pipeline segmentation is utilized in conjunction with SDP.

## 5.2 Research Contributions

Considering the gaps in the existing literature, as discussed in Chapter 1, and the developed PHM approach, as discussed in Chapters 2 to 4, the followings are cited as the contributions of this dissertation:

- The majority of pipeline health monitoring approaches only consider one detection method (section 1.1.1). However, this work considers a combination of multi-type sensors network, and human. Doing so, a more cost-effective and reliable health monitoring of locally damaged pipelines is achieved. Moreover, in the developed approach, a larger number of key attributes from different detection methods are accounted for. Additionally, this research considers both cost and detection probability and optimizes them concurrently, as opposed to the literature where PHM approaches typically only consider either health monitoring cost or probabilistic detection of damage (section 1.1.4).
- The majority of reported pipeline PHM approaches consider binary detection methods (section 1.1.2). However, this research integrates three probabilistic measures (namely POD, IP, and ME), along with their dependencies on damage type, damage size, and damage-to-sensor distance. In this way, the resulting health monitoring layouts will be consistent with the severity and risk of failure corresponding to different localized damages with different sizes.

- Pipeline PHM analyses used deterministic data regarding location, type, and size of damages (Section 1.1.3). The proposed PHM approach leverages probabilistic and limited size sampling to simulate and place different damage on a pipeline. As such, it is well-suited for applications in the early stages of pipeline design.
- While the literature lacks approaches for the fusion of heterogeneous sensor data and inspection information to estimate damage size and RUL (section 1.1.5), the developed PHM approach considers multi-source data fusion in a hybrid context, where online, frequent, and indirect sensor data along with periodic, less frequent yet accurate inspection images are considered together with a POF model. The outcome of the proposed fusion approach is a more accurate, frequent and online RUL estimation.
- Sensor-based damage size estimation approaches do not consider long temporal correlations due to expensive computations (section 1.1.6). The proposed hybrid fusion approach advances this shortcoming by benefiting from a POF model and utilizing an LSTM-regression model that provides a computationally tractable solution with long temporal correlations, as opposed to immediate correlations, considered for online estimation of damage size based on sensors' time-series data (e.g., AE sensors' signals).

- Past research has considered images for damage localization, but not for damage sizing (section 1.1.7). This research developed a customized FCDN model, which has a structure with as many layers and convolution kernels as needed for damage sizing tasks. Then, the customized FCDN model is used in the proposed fusion approach for automated and online estimation of damage size using images of the damaged area.
- The literature lacks approaches with proper scalability to be used for optimal health monitoring of long and densely damaged pipeline segments (section 1.1.8). However, this research provides a computationally tractable solution for integrated health monitoring of a pipeline. The corresponding solution utilizes SDP, along with the modified segment-level approach with a better scalability, to find an optimal set of health monitoring layouts corresponding to a set of pipeline segments, while detection limits and an overall cost constraint are considered.

### **5.3 Potential Future Research**

Possible extensions of the work reported in this dissertation, considering the limitations of the present work, are listed below.

1- The approach for segment-level health monitoring layout design optimization was originally (Chapter 2) formulated as a mixed-integer nonlinear problem. The corresponding formulation was modified in Chapter 4 to improve the scalability through solving the problem using mixed-integer optimization techniques. Nonetheless, even mixed-integer problems are known to be NP-hard. Thus, new research to modify the



developed approach, so that it can be solved using less computationally expensive techniques, is desirable. As a result, the layout design problem will not be NP-hard anymore.

2- In the approach for online damage size and RUL estimation (Chapter 3), LSTM models for AE-based time-series analysis should be retrained if the approach is used for different experiments or type of damage. This limitation is justified to a reasonable extent considering the significant variability in AE waveforms upon any changes in the operational environment. Nonetheless, an extension of the presented approach would be desirable to address this issue through the usage of deep learning for automated AE feature extraction for damage size and RUL estimation.

3- LSTM and FCDN can be considered as computationally complex models. The trade off between computational complexity and performance of the models can be explored to see if acceptable results can be obtained using less complicated models. This exploration is not considered in this research and can be the topic of another research effort. The results of the trade off exploration can then be compared to those obtained in this research to determine the optimal model with an acceptable performance.

4- Damage growth behavior is not considered in the proposed optimization models for layout design in Chapter 2 and integrated health monitoring in Chapter 4. Thus, while the developed approach for dynamic and integrated PHM of locally damaged pipelines considers online damage size estimates to update health monitoring layouts accordingly,

it doesn't consider inspection time as well as data gathering frequency. As an extension to the current research, future research should consider damage growth behavior in optimization modeling. Doing so, an approach for optimal and online PHM of pipelines can be achieved at where inspection schedule and data acquisition frequency are also optimized.

5- The approach for integrated PHM of pipelines, discussed in Chapter 4, considers two failure risk scenarios for each pipeline segment to account for failure uncertainties. However, there are uncertainties associated with probabilistic detection and sizing of damages which are not considered in the segment-level approach. Moreover, RUL estimation may also involve uncertainties. An extension of the approach of this dissertation can use probabilistic optimization techniques to take those uncertainties into account in order to achieve a more reliable PHM of pipelines.

6- The approach presented in this dissertation considers uncertainties associated with stochastic localized damages and their detection by using probabilistic sampling. Thus, the final layout of each segment is obtained in the presented approach while optimal layouts, corresponding to different realizations of damages, are considered. However, damage and detection uncertainties can be directly considered in the optimization. As such, in future research, the problem of pipeline health monitoring can be formulated as a reliability-based design optimization problem. In doing so, health monitoring layout design will be optimized while detection uncertainties are modeled as probabilistic design constraints.

7- There are numerous sensor types and inspection tools that can be used for PHM of pipelines. However, simultaneous consideration of all of them in an optimization problem may lead to computation intractability. Rating and screening different types of sensors and inspection tools for pipeline health monitoring can be a topic of further research. In such a research, sensors and inspection tools can be ranked considering their key features, e.g., cost, data gathering frequency, and precision, in addition to degradation behavior, pipeline specifications, and degradation likelihood over the pipeline surface. Results of such a screening process can be used to select a set of candidate types of sensors and inspection tools to be considered in the layout design optimization problem.

8- The developed approach for dynamic and integrated PHM of pipelines tries to provide an optimal allocation of detection methods, while all damages with different levels of severity, located at different pipeline segments, are considered. As such, average probabilistic detection and average utility are optimized in finding optimal layouts using the developed approach. However, this problem can also be solved using max-min optimization techniques. Thus, future research may consider different possible problem formulations where max-min optimization techniques are used to solve the same problem for the segment- and integrated-level health monitoring of pipelines. Corresponding results, then, can be compared with those of this dissertation to evaluate the performance of each of the approaches.

## Appendix A

This is an instruction manual for SHM2Opt package. This package is used to obtain optimal health monitoring layouts for locally damaged pipelines. The output of this package is a two-dimensional layout, including configuration of sensor of different types and human inspection along the exterior surface of a pipeline. The code package can be found at <https://github.com/aminaria/SHM2Opt> . To obtain results using this package, please follow the steps of the illustrative example described below.

### 1- Run **Damage Generator\_AA\_UMD.R**

Make any changes in the input section, if necessary.

```
#####INPUTS#####INPUTS#####INPUTS#####INPUTS#####INPUTS#####
#####INPUTS#####INPUTS#####INPUTS#####INPUTS#####INPUTS#####

##damage denisty per meter
rate= 12/50;|

##Length and radius of pipeline
l=50;
r=0.5;

##Number of samples to be produced
#Refer to the journal paper for more info on number of samples required and wilks method)
nwilks=93;

##Number of size-type catagories
#Considering the size distriubtion, size-type classes are produced. Refer to journal paper for more info.
ns=4

###Energy cosntraint
#sensor energy consumption is an exponential function of its distance to closest sensor.
#Refer to the journal paper for more info. (econ = 18 meters)
econ=18

###Inference constraint.
#Considering the similarity matrix that should be defined for the structure, results of inference are not reliable if
#the distance between monitored damage and infered damage exceeds a limit, icon.
icon=20

###Cluster distance constraint
#Cluster distance limit is defined considering energy and inference constraints.
#Refer to the journal paper for more info.
clcon=min(econ,icon)

#####End of INPUTS#####End of INPUTS#####End of INPUTS#####
#####End of INPUTS#####End of INPUTS#####End of INPUTS#####
```

2- The output is the **Wilks.txt** file stored in the folder at where the code file is. Open **Wilks.txt** and choose the sample you want.

```

wils.txt - Notepad
File Edit Format View Help
"xexist2" "yexist2" "xexist2" "ynode" "c#1" "c#2" "c#3" "c#4" "cannot-link"
"1" 1.3680325371773 6.2827828778247 1.3680325371773 5.82560043478023 0 1 0 0 0
"2" 1.8730830422278 0.163966349372375 1.8730830422278 5.99151176359572 1 0 0 0 0
"3" 5.40843657758134 6.22193565878971 5.40843657758134 6.27238147526933 0 1 0 0 0
"4" 8.43873960788437 6.27954669496687 8.43873960788437 0.119455835442889 0 0 0 1 0
"5" 10.4589416280864 0.438015392747837 10.4589416280864 0.3410038202993 0 0 1 0 0
"6" 21.5700527391975 0.500883117414991 21.5700527391975 0.471481504481266 1 0 0 0 0
"7" 31.6710628402076 5.84395284648216 31.6710628402076 6.22115486022248 0 0 0 1 0
"8" 34.7013658705106 5.98941336844656 34.7013658705106 6.17384706966188 0 1 0 0 0
"9" 35.7114668806116 6.14714389448967 35.7114668806116 5.86253776439447 0 0 1 0 0
"10" 45.8124769816217 1.47842494386062 45.8124769816217 1.57594384194166 0 0 1 0 0
"11" 46.3175274866723 5.90746964127485 46.3175274866723 0.122715753117645 1 0 0 0 1
"12" 10.4589416280864 0.438015392747837 10.4589416280864 0.692612394456881 0 1 0 0 0
"13" 11.9740931432379 0.327686646911802 11.9740931432379 0.591096318052637 0 0 0 1 0
"14" 12.9841941533389 5.66544624247412 12.9841941533389 5.34165729411701 0 1 0 0 0
"15" 16.5195476886925 4.70042118543384 16.5195476886925 4.42069603429733 0 1 0 0 0

```

3- Copy the sample under study in the corresponding area in **Optimizer-Feeder\_AA\_UMD.xlsx**

	A	B	C	D	E	F	G	H	I	J	K
1	x	t	y	ty	class#1	class#2	class#3	class#4	Cannot-Link		
2	1	1.368033	6.282783	1.368033	5.8256	0	1	0	0	0	
3	2	1.873083	0.163966	1.873083	5.991512	1	0	0	0	0	
4	3	5.408437	6.221936	5.408437	6.272381	0	1	0	0	0	
5	4	8.43874	6.279547	8.43874	0.119456	0	0	0	1	0	
6	5	10.45894	0.438015	10.45894	0.341004	0	0	1	0	0	
7	6	21.57005	0.500883	21.57005	0.471482	1	0	0	0	0	
8	7	31.67106	5.843953	31.67106	6.221155	0	0	0	1	0	
9	8	34.70137	5.989413	34.70137	6.173847	0	1	0	0	0	
10	9	35.71147	6.147144	35.71147	5.862538	0	0	1	0	0	
11	10	45.81248	1.478425	45.81248	1.575944	0	0	1	0	0	
12	11	46.31753	5.90747	46.31753	0.122716	1	0	0	0	1	
13											
14											
15											

In the meanwhile, open **Matrix Developer\_AA\_UMD.R**, assign values to ‘first’ and ‘last’ variables to choose the sample under study (outloc is the table corresponding to Wilks.txt). Once values are assigned, run only the code that is shown in the figure below.

```
#####
#####Matrix Developer Module##### Copyright 2018, Amin Aria, All rights reserved.
#####

!!!##NOTE: The Minimum Spanning Tree sub-module of the clustering module is included here.

####Sample Assingment#####Sample Assingment#####Sample Assingment#####Sample Assingment#####Sample Assingment
####Sample Assingment#####Sample Assingment#####Sample Assingment#####Sample Assingment#####Sample Assingment

#outloc is the output of Damage Generator module.You can pass the input in a matrix format similar to the format of
#Damage generator output

first=1 #first cell of the sample under study
last=11 #Last cell of the sample under study
x=outloc[first:last,1]
t=outloc[first:last,2]
y=outloc[first:last,3]
ty=outloc[first:last,4]

#####INPUTS#####INPUTS#####INPUTS#####INPUTS#####INPUTS#####INPUTS#####
#####INPUTS#####INPUTS#####INPUTS#####INPUTS#####INPUTS#####INPUTS#####
```

4- Make any changes necessary in the input section of **Matrix Developer\_AA\_UMD.R** and run the remaining part of the **Matrix Developer\_AA\_UMD.R** (starting from input section).

```
24 - #####INPUTS#####INPUTS#####INPUTS#####INPUTS#####INPUTS#####INPUTS#####
25
26 #radius of pipeline
27 r=1;
28 #number of damages
29 n=length(x)
30 #number of classes
31 ns=4
32 #number of sensors types
33 nsensor=2
34
35 #####Minimum Spanning Tree library : used for clustering damages of each sample
36 library(optrees)
37
38 ##Distance between damages and nodes
39 dist=matrix(0,nrow=length(x),ncol=length(y))
40
41 ##Probability of missing a defect
42 POND=matrix(0,nrow=length(x),ncol=length(y))
43
44 #Damage Missing relation: exp(deltax/a +b ), boundry conditions: exp(deltax=0)=0.1, exp(deltax=20 meters)= 1 >>>
45 #!!!##NOTE: you can change this relation considering the geometry of pipeline and pattern recognition results
46 #of the pipeline.
47
48 a1=c(8.3,8,8,7)
49 b1=c(-2.7,-3,-3.5,-2.7)
50 LPONDint=matrix(,nrow=0,ncol=length(x))
51
52 ##parameters of PoD(size) for Acoustic Emission
53 #(Rabiei and Modarres, 2013) when PoD(depth=1.5mm)=1
54 aAE=3.07
55
56
57 ##parameters of PoD(size) for HI with Ultrasonic tool
```

5- From the ‘output’ folder at the code file directory, copy the output matrices (ME1,ME2, LPODT1, LPODT2,DeltaAE,DeltaHI, and LPOND) to the corresponding areas in **Optimizer-Feeder\_AA\_UMD.xlsx**. Note that GAMS is sensitive to rows and columns indices. So make sure that those indices are numbers in ascending order (as you see in the figure below).

LPoND matrix for all damages	1	2	3	4	5	6	7	8	9	10	11	12
1	0	0	0	0	0	0	0.934961	1.300095	1.42215	2.457486	2.640332	
2	2.603806	2.645487	2.273552	1.908939	1.665339	0.32658192	0	0	0	0	0	
3	2.900199	2.943443	2.557561	2.179274	1.92654	0.537578742	0	0	0	0	0	
4	2.634688	2.616711	2.122798	1.689761	1.4004		0	0	0	0	0	
5	2.992525	3.057143	3.493694	3.120563	2.866719	1.478713839	0.217172		0	0	0	
6	0.263302	0.324966	0.751854	1.117247	1.361173	2.696457637	1.481144	1.116224	0.992626		0	0
7	1.687819	1.761147	2.267098	2.68287	2.407306	0.822887003		0	0	0	0	
8	0	0	0	0.09513	0.346706	1.732287904	2.95285	2.618974	2.494944	1.214522		1.16786
9	0	0	0.216735	0.59513	0.846706	2.232287904	3.45285	3.118974	2.994944	1.714522		1.66786
10	0	0	0	0.216772	0.468669	1.855824046	3.120106	3.476946	3.372745	2.09165		2.047061
11	0	0	0	0	0	0	0.934961	1.300095	1.42215	2.457486		2.640332
12												

## 6- Open Clustering-Unit\_AA\_UMD.ipynb

Use Clustering Feeder in **Optimizer-Feeder\_AA\_UMD.xlsx** to form the matrix of damages location.


V	W	X	Y	Z	AA	AB
Clustering Feeder					perimeter	6.28
	x		t			
[	1.368033	,	6.282783	,	-0.002783	],
[	1.873083	,	0.163966	,	0.1639663	],
[	5.408437	,	6.221936	,	0.0580643	],
[	8.43874	,	6.279547	,	0.0004533	],
[	10.45894	,	0.438015	,	0.4380154	],
[	21.57005	,	0.500883	,	0.5008831	],
[	31.67106	,	5.843953	,	0.4360472	],
[	34.70137		5.989413		0.2905866	],
[	35.71147		6.147144		0.1328561	],
[	45.81248		1.478425		1.4784249	],
[	46.31753		5.90747		0.3725304	],
[	0		0			],
[	0	,	0	,		],

Copy this matrix and use it as the input of **Clustering-Unit\_AA\_UMD.ipynb** and run first two boxes. Make sure that the default perimeter value, as well as “x” and “t” values, are consistent with the sample you are using.

```
#####
#####Clustering Unit Module##### Copyright 2018, Amin Aria, ALL rights reserved.
#####

%matplotlib inline
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
# For Interactive Plotting Sliders
from bokeh.plotting import figure
from bokeh.io import push_notebook, output_notebook, show
#from bokeh.models import ColumnDataSource
from ipywidgets import interact
from bokeh.resources import INLINE
# output to the IPython Notebook
output_notebook(resources=INLINE)

x=[[1.368032537, 6.282782878, -0.002782878],
 [1.873083042, 0.163966349, 0.163966349],
 [5.408436578, 6.221935659, 0.058064341],
 [8.438739608, 6.279546695, 0.000453305],
 [10.45894163, 0.438015393, 0.438015393],
 [21.57005274, 0.500883117, 0.500883117],
 [31.67106284, 5.843952846, 0.436047154],
 [34.70136587, 5.989413368, 0.290586632],
 [35.71146688, 6.147143894, 0.132856106],
 [45.81247698, 1.478424944, 1.478424944],
 [46.31752749, 5.907469641, 0.372530359]]
```

 BokehJS 0.12.5 successfully loaded.

Next, define cannot-link damages (if any), run the output box, and obtain the output.

```
#####Output Box#####
#####Output Box#####
xdata=x

#number of clusters:3
#ml=must link damages, for example: ml=[[2,3]] which means damages 2 and 3 should be in a cluster
#cl=cannot link damages, for example:
#cl=[[2,3],[5,6]] means damages 2 and 3 cannot be in a cluster. Same for damages 5 and 6.
#NOTE: Python numbering system: first damages is damage zero
a1=cop_kmeans(xdata, 3, ml=[], cl=[])
a1[0]
```

[1, 1, 1, 1, 1, 0, 0, 0, 0, 2, 2]

Considering the output, clusters are like this: \1,2,3,4,5\6,7,8,9\10,11\

### 7- Define each cluster in the cluster box of **Optimizer-Feeder\_AA\_UMD.xlsx**

Clusters	
set node11(j) ;	1*5
set node22(j)	6*9
set node33(j)	10*11



## 8- Open 2D Optimizer\_AA\_UMD.gms

Follow all the hints and instructions in the section:

\*\*\*\*\*INPUTs that their change, or relaxation, changes the final data\*\*\*\*\*

```
*****INPUTs that their change, or relaxation, changes the final data*****  
*****INPUTs that their change, or relaxation, changes the final data*****  
*****INPUTs that their change, or relaxation, changes the final data*****  
*****INPUTs that their change, or relaxation, changes the final data*****  
*****INPUTs that their change, or relaxation, changes the final data*****
```

Input the corresponding values from **Optimizer-Feeder\_AA\_UMD.xlsx** and to the appropriate positions in the input section of the GAMS file. Moreover, copy the excel file in GAMS directory (GAMS makes a directory in your (user) documents folder in Windows®). Make sure that the structure of the feeder excel file is kept as is since GAMS will read tables from that file and location of cells matter.

For example, for LPODT matrices:

```
$CALL GDXXRW.EXE Optimizer-Feeder_AA_UMD.xlsx trace=3 Squeeze=N par=LPODT1  
rng=Sheet1!AJ59 maxDupeErrors=1000000
```

If the yellow cell of LPODT table in the excel file is not located at AJ59, you will get an error or illogical results!

## 9- Run 2D Optimizer\_AA\_UMD.gms

In the display sub-section of output file (2D Optimizer\_AA\_UMD.lst), which will be opened once compilation is finished, you see what we have in the screenshot followed.

```

---- 408 VARIABLE avgdet.L          = 0.727 average detection
      VARIABLE Redun.L            = 0.909 Redundancy of detection (hit-miss model)
      VARIABLE aveut.L            = 0.494 average utility per node
      VARIABLE nPOD.L             = 8.391 avelog(probability of missing defects)

---- 408 VARIABLE s11.L AE 1 at node j indicator
2 1.000, 4 1.000, 5 1.000, 9 1.000, 10 1.000

---- 408 VARIABLE s2.L HI with USonic at node j indicator
8 1.000

---- 408 VARIABLE ME.L measurement error for sensor at node j and the
      t damage
2 0.041, 4 0.014, 5 0.038, 8 0.273, 9 0.038, 10 0.038

---- 408 VARIABLE misslog.L logarithm of prognosis failure for defect i
1 5.180, 2 6.220, 3 7.049, 4 5.707, 5 9.044, 6 6.4
7 8.351, 8 8.270, 9 10.770, 10 11.127, 11 6.680

---- 408 VARIABLE detyuse.L jth sensor usage
2 1.000, 4 1.000, 5 1.000, 8 1.000, 9 1.000, 10 1.000

---- 408 Display 0.000 0.000 SECS 3 MB

```

To summarize it, for the sample under study with 11 damages of different size classes, the optimal solution is:

5 AE sensors at nodes 2,4,5,9, and 10 while human inspection is assigned to node 8. Using this layout, all the damages are detected with average redundancy of 0.909. Additionally, geometric mean of probability of missing a damage is calculated to be  $e^{-8.391}$ .

## Appendix B

This is an instruction manual for the code packages used for sensor- and image-based damage size estimation in Chapter 5. In this appendix, at first the instruction manual for the package FuseLSTM\_AA (for damage size estimation based on sensor data) is described. Next, the instruction manual for biSS\_AA package (for damage size estimation based on images) is elaborated.

### Package “FuseLSTM\_AA”

The package FuseLSTM\_AA, available at [https://github.com/aminaria/FuseLSTM\\_AA](https://github.com/aminaria/FuseLSTM_AA), uses an LSTM regression model to do time-series analysis and estimate crack length (or variable of interest) based on features of data corresponding to a particular time window prior to the time of interest. This package can also be used to fuse different damage size estimates to obtain a more accurate final estimate.

To use this FuseLSTM\_AA package, follow the steps listed below:

1- To do a grid search at first, run "BiLSTM\_AA\_GS.py". Determine:

- a) Grid search intervals
- b) The input file (e.g., Exp 18.csv at the link above)
- c) Features (columns) to be considered, and
- d) The label column (e.g., size column in the case of crack size estimation)

in the section “Grid search intervals” and “Input files”, as shown in the next figure.

```
#####-- Grid search intervals--#####

training = [200,300,400] #Training Steps
Batch_Size = [5,10,20, 64, 128] # Batch sizes to be considered
Numb_Hidden = [16,32, 64, 128] # Number of nodes at each gate or unit
Keep_Prob = [0.4,0.5, 0.6, 0.7] # Keep probability in Training
N_tw = 20 #number of time steps (Temporal Correlation Range)
N_ft = 2 #Nubmer of features to be considered

#####-----Input files-----#####

#Features of the orginial dataset to be considered plus the label column (here 'Size' is label column)
cols= ['NCC','NCE','Size']

##Data reduction step, used for large datasets, is the last argument. In the case that it is 1,
#all entries of the data set are considered in the anlaysis

x2,y2=inputfile('Exp 18.csv',cols,10)
np.save('LSTMx',x2)
np.save('LSTMx',x2)

###
#####-----if used for data fusion
###

...

cols= ['LSTM','FCDN','Real']
xfuse,yfuse=inputfile('LSTM_predict.csv',cols,2)
np.save('LSTMx',xfuse)
np.save('LSTMx',xfuse)

...
```

2- Considering the text output file of the grid search (e.g., see the figure below), find the best model based on reported performance metric and use "BiLSTM\_AA.py" to train that model.

	A	B	C	D	E	F	G	H	I
1	training	batch size	LSTM strcuture	Keep_Prob	acc_cross	loss_cross	acctest	time (sec)	
2	200	5	16	0.4	0.175608948	0.190076128	0.167896912	51.8309114	
3	200	5	16	0.5	0.168014169	0.166245326	0.145380989	50.89574456	
4	200	5	16	0.6	0.213694081	0.237807006	0.217101783	51.12207508	
5	200	5	16	0.7	0.140583709	0.21371083	0.173870757	50.9485569	
6	200	5	32	0.4	0.145599633	0.144543231	0.159248859	50.98169899	
7	200	5	32	0.5	0.132605836	0.165460095	0.153106987	50.99291873	
8	200	5	32	0.6	0.200477064	0.231696263	0.206285387	51.21797609	
9	200	5	32	0.7	0.148394063	0.196893528	0.155777588	51.02469826	
10	200	5	64	0.4	0.165097207	0.172705725	0.15328382	51.86280465	
11	200	5	64	0.5	0.14807266	0.131723925	0.123206072	51.39498377	
12									

For the model with the best performance metrics, determine input variables and input file in the "Input variables/paramters" section of the "BiLSTM\_AA.py" code and run the file.

```
#####-----Input variables/paramters-----#####

#-----Hyperparameters
N_tw = 20 #number of time steps (Temporal Correlation Range)
N_ft = 2 #number of features
cols= ['NCC','NCE','Size'] #Features of the orginial dataset to be considered plus the label column (here 'Size' is label column)

training_steps = 400
batch_size = 10 # number of time windows to be considered at each step of training
n_hidden = 128 #number of LSTM cells repeated
k_p =0.7 # dropout probability for LSTM cells

#####-----Address and Columns of the input data (Y value at the last column)-----#####

##Data reduction step, used for large datasets, is the last argument. In the case that it is 1,
#all entries of the data set are considered in the anlalysis

x2,y2=inputfile('Exp 18.csv',cols,3)
np.save('LSTMMy',y2)
np.save('LSTMx',x2)

###
#####-----if used for data fusion
###

...
cols= ['LSTM','FCDN','Real']
xfuse,yfuse=inputfile('LSTM_predict.csv',cols,2)
np.save('LSTMMy',yfuse)
np.save('LSTMx',xfuse)

...
```

3- Use "BiLSTM\_AA\_predict.py" to predict the value of interest (e.g., size). The output will be a csv file including input features and the predicted value (e.g., final damage size estimate)

```
#####-----Input variables, files, and hyperparameters-----#####
N_tw = 20 #number of time steps
N_ft = 2 #number of features
cols= ['NCC','NCE','Size','FCDN Size','time']

n_hidden = 128 #number of nodes in each gate or unit of a LSTM cell
k_p = 0.7 # Keep Probability probability for LSTM cells
num_classes = 1
#-----
x = tf.placeholder(tf.float32, shape=[None, N_tw,N_ft],name="x")

###-----Address and Columns of the input files (Y value at the last column)-----#####
x2,y2,yFCDN=inputfile('Exp 18.csv',cols,1)

#####-----Data to be analyzed-----#####
LSTMx= x2
LSTMy= y2

###-----if used for fusion
...
cols= ['LSTM','FCDN','Real','FCDN','Time']
xfuse,yfuse,yFCDN=inputfile('LSTM_predict.csv',cols,3)
LSTMx= xfuse #np.load(file = 'LSTMx.npy')
LSTMy= yfuse
...

pre_set= np.sort(np.random.randint(0,len(LSTMx),int(len(LSTMx)*0.7)))
X_predict =LSTMx[pre_set,:,0:(N_ft)]
Y_real = LSTMy[pre_set,:]
Y_FCDN = yFCDN[pre_set,:]
```

Example of final output:

	A	B	C	D
1	predict	FCDN	Time	Real
2	0.218326	0.216	7980.516	0.203
3	0.23303	0.226	7988.067	0.213
4	0.215668	0.232	7992.467	0.219
5	0.246895	0.233	7993.268	0.22
6	0.24942	0.233	7993.268	0.22
7	0.247767	0.233	7993.268	0.22
8	0.231451	0.239	7997.667	0.226
9	0.289406	0.244	8001.268	0.231
10	0.231801	0.244	8001.268	0.231
11	0.281097	0.253	8007.713	0.24
12	0.26086	0.273	8021.318	0.259
13	0.220085	0.279	8026.268	0.266
14	0.267179	0.283	8028.467	0.269
15	0.290626	0.285	8031.268	0.273

## Package “biSS\_AA”

The package biSS\_AA is a modification of George Seif semantic segmentation package (<https://github.com/GeorgeSeif/Semantic-Segmentation-Suite>).

The package biSS\_AA can be found at <https://github.com/aminaria/bi-class-Semantic-Segmentation-in-Python-Tensor-Flow->. To use this modified package, the user may start with ReadME file of the original package and then continue with current instructions manual. This modified package is different from the original package is the following ways:

1- The current package is customized for bi-class segmentation with an example on images of a cracked area (two classes: crack and background).

2- The current package is customized to handle images with significant class imbalance in the pixels.

3- A new model (FC\_Dense\_Net\_AA) is added in ‘FC\_DenseNet\_Tiramisu.py’ code in the models folder. The newly introduced model has a simpler structure in comparison to the original 100 layer Tiramisu model. The new model is specifically fabricated for the relatively simple case of bi-class semantic segmentation. Hence, it achieves accurate results for the cases with two classes of objects. Nonetheless, it is possible that it doesn’t show promising results when used for cases with more than 5 classes of objects.

4- Grid search files (with “\_GS\_AA” suffix in their name) are added to do a grid search regarding the newly introduced model (FC\_DenseNet\_AA) or any other model.

5- In the case of grid search, the user has to determine the search intervals (discrete values) corresponding to each of the hyperparameters of the interest.

6- In the case of grid search, checkpoints corresponding to each of the explored models are saved in the checkpoints folder.



## Appendix C

As it is mentioned in Chapter 6, the developed Meta planning approach for health monitoring of pipelines uses the package code of Appendix A, i.e., SHM2Opt, to produce per segment optimal layouts. For each optimal layout, the objective function value, cost metric, and budget limit are stored to be passed to the SDP problem. The following MATLAB® code is used to solve the SDP problem. The output of this code is the optimal set of layouts corresponding to a set of pipeline segments subject to low and high-risk damage scenarios.

### MATLAB Code

```
function [y,obj]=SDP_3segments_v3_SCM(G,file)
%function takes as input the number of available sensors and the name of the file containing the object fnc for
%each segment and for each possible number of sensors

%%
folder = pwd; %obtain current directory path

% Import the data
[~,~,raw] = xlsread(strcat(folder,file),'Utility'); %read data file
raw(cellfun(@(x) ~isempty(x) && isnumeric(x) && isnan(x),raw)) = {};

% Replace non-numeric cells with NaN
R = cellfun(@(x) ~isnumeric(x) && ~islogical(x),raw); % Find non-numeric cells
raw(R) = {NaN}; % Replace non-numeric cells
Utility = reshape([raw{:}],size(raw)); % Create output variable
clearvars raw R; % Clear temporary variables

%%
S = 3; %Number of segments
C_min = [5 3 5];
C_max = [15 9 13]; %Max sensor capacity
x_sensor = C_max-C_min+1; % possible number of sensors assigned at each segment.

% Utility_2 replaces Utility
Utility_2 = zeros(S,max(C_max));
for j=1:S
    for i=1:x_sensor(j)
        Utility_2(j,C_min(j)+i-1)=Utility(j,i);
    end
end
```

```

% create the f table and put the last stage value into it. f is the obj func.
f = zeros(S,min([max(C_max) G]));
for j=1:min([max(C_max) G])
    f(S,j)=Utility_2(S,j);
end

% create stage2 to stage3 paths
comb = [];
cont =1;
path23 = [];
for i=C_min(2)+C_min(3):min([(G-C_min(3)) C_max(2)+C_max(3)])
    a=0;
    for j=C_min(3):min([(G-C_min(2)-C_min(3)) C_max(3)])
        if (i-j)<=C_max(2) && i-j>=C_min(2)
            a=a+1;
            path23(cont,1) = i;
            path23(cont,2) = j;
            cont = cont+1;
        end
        comb(i+1)=a;
    end
end
paths23 = size(path23);
% calculate the utility at each path
for i=1:paths23(1)
    path23(i,3) = path23(i,1)-path23(i,2);
    path23(i,4) = path23(i,2);
    path23(i,5)=path23(i,3)+path23(i,4);
    path23(i,6)= Utility_2(2,path23(i,3))+f(3,path23(i,4));
end
% choose the best path for each state
for i=C_min(2)+C_min(3):min([(G-C_min(3)) C_max(2)+C_max(3)])
    f(2,i)=max(path23((path23(:,5)==i),6));
end

path12=[];
cont=1;%reset cont
for i=C_min(1)+C_min(2)+C_min(3):min([G (C_max(1)+C_max(2)+C_max(3))])
    a=0;
    for j=C_min(2)+C_min(3):min([(G-C_min(3)) C_max(2)+C_max(3)])
        if (i-j)<=C_max(1) && i-j>=C_min(1)
            a=a+1;
            path12(cont,1) = i;
            path12(cont,2) = j;
            cont = cont+1;
        end
        comb(i+1)=a;
    end
end
paths12=size(path12);

```

```

for i=1:paths12(1)
    path12(i,3) = path12(i,1)-path12(i,2);
    path12(i,4) = path12(i,2);
    path12(i,5)= path12(i,3)+path12(i,4);
    path12(i,6)= Utility_2(1,path12(i,3))+f(2,path12(i,4));
    % 1,2 are the stages here
end

for i=C_min(1)+C_min(2)+C_min(3):min([G (C_max(1)+C_max(2)+C_max(3))])
    f(1,i)=max(path12((path12(:,5)==i),6));
end

obj=max(f(1,:)); % get the objective value

% get each stage variable, only consider one path
result_s(1,:)=path12(path12(:,6)==obj,1);

result_g(1,:)=path12(path12(:,6)==obj,3);
result_s(2,:)=path12(path12(:,6)==obj,2);

result_g(2,:)=path23(path23(:,6)==f(2,result_s(2)),3);
result_s(3,:)=path23(path23(:,6)==f(2,result_s(2)),2);

result_g(3,:)=result_s(3,:);

y=result_g;
end

```

## References

- Abdollahzadeh S. and Navimipour N.J., 2016 Oct. Deployment strategies in the wireless sensor network: A comprehensive review. *Computer Communications*, 91(1), pp. 1-6.
- Abyani, M. and Bahaari, M.R., 2020. A comparative reliability study of corroded pipelines based on Monte Carlo Simulation and Latin Hypercube Sampling methods. *International Journal of Pressure Vessels and Piping*, p. 104079.
- Alaswad, S. and Xiang, Y., 2017. A review on condition-based maintenance optimization models for stochastically deteriorating system. *Reliability Engineering & System Safety*, 157, pp. 54-63.
- Alduraibi, F., Lasla, N. and Younis, M., 2016, May. Coverage-based node placement optimization in wireless sensor network with linear topology. In *2016 IEEE International Conference on Communications (ICC)* (pp. 1-6). IEEE.
- Argyris, C., Chowdhury, S., Zabel, V. and Papadimitriou, C., 2018. Bayesian optimal sensor placement for crack identification in structures using strain measurements. *Structural Control and Health Monitoring*, 25(5), p. e2137.
- Atha, D.J. and Jahanshahi, M.R., 2018. Evaluation of deep learning approaches based on convolutional neural networks for corrosion detection. *Structural Health Monitoring*, 17(5), pp. 1110-1128.
- Bannantine, J., Comer, J. and Handrock, J., 1990. Fundamentals of metal fatigue analysis((Book)). *Research supported by the University of Illinois. Englewood Cliffs, NJ, Prentice-Hall, 1990, 286.*
- Barton, L., Laing, I., Pinto, A. and Ladwa, R., 2017. Offshore Oil and Gas Pipeline: Flow Assurance and Corrosion Modelling for Inspection Prioritization. In ASME 2017 India Oil and Gas Pipeline Conference. American Society of Mechanical Engineers Digital Collection.
- Bassim, M.N., Lawrence, S.S. and Liu, C.D., 1994. Detection of the onset of fatigue crack growth in rail steels using acoustic emission. *Engineering Fracture Mechanics*, 47(2), pp.207-214.
- Benstock, D. and Cegla, F., 2016. Sample selection for extreme value analysis of inspection data collected from corroded surfaces. *Corrosion Science*, 103, pp. 206-214.
- Berens AP. 1989. NDE Reliability Data Analysis. In ASM Handbook, Volume 17: Nondestructive and Quality Control, pp. 689-701. ASM International, Materials Park, Ohio.
- Brunton, B.W., Brunton, S.L., Proctor, J.L. and Kutz, J.N., 2016. Sparse sensor placement optimization for classification. *SIAM Journal on Applied Mathematics*, 76(5), pp. 2099-2122.

Bussieck, M.R. and Pruessner, A., 2003. Mixed-integer nonlinear programming. *SIAG/OPT Newsletter: Views & News*, 14(1), pp. 19-22.

Cao, Y., Xie, Y. and Gebraeel, N., 2018. Multi-sensor slope change detection. *Annals of Operations Research*, 263(1-2), pp. 163-189.

Cerrada, M., Sánchez, R.V., Li, C., Pacheco, F., Cabrera, D., de Oliveira, J.V. and Vasquez, R.E., 2018. A review on data-driven fault severity assessment in rolling bearings. *Mechanical Systems and Signal Processing*, 99(1), pp. 169-196.

Chatterjee, K. and Modarres, M., 2013. A probabilistic approach for estimating defect size and density considering detection uncertainties and measurement errors. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, 227(1), pp. 28-40.

Chen, F.C. and Jahanshahi, M.R., 2018. NB-CNN: Deep learning-based crack detection using convolutional neural network and Naïve Bayes data fusion. *IEEE Transactions on Industrial Electronics*, 65(5), pp. 4392-4400.

Cofre Martel, S., 2018. A deep learning based framework for physical assets' health prognostics under uncertainty for big machinery data. Master's thesis, UNIVERSIDAD DE CHILE.

Cofre Martel, S., Kobrich, P., Lopez Droguett, E., and Meruane, V., 2018, October. Transmissibility based structural assessment using deep convolutional neural network, ISMA 2018, Belgium.

Crowder, H., Johnson, E.L. and Padberg, M., 1983. Solving large-scale zero-one linear programming problems. *Operations Research*, 31(5), pp. 803-834.

Dumoulin, V. and Visin, F., 2016. A guide to convolution arithmetic for deep learning. *arXiv preprint arXiv:1603.07285*.

Eleftheroglou, N., Zarouchas, D., Loutas, T., Alderliesten, R. and Benedictus, R., 2018. Structural health monitoring data fusion for in-situ life prognosis of composite structures. *Reliability Engineering & System Safety*, 178, pp.40-54.

Elsheikh, A., Yacout, S. and Ouali, M.S., 2019. Bidirectional handshaking LSTM for remaining useful life prediction. *Neurocomputing*, 323, pp.148-156.

Fang, X., Paynabar, K. and Gebraeel, N., 2017. Multistream sensor fusion-based prognostics model for systems with single failure modes. *Reliability Engineering & System Safety*, 159, pp. 322-331.

Filippov, A.V., Rubtsov, V.E., Tarasov, S.Y., Podgornykh, O.A. and Shamarin, N.N., 2018. Detecting transition to chatter mode in peakless tool turning by monitoring vibration and

acoustic emission signals. *The International Journal of Advanced Manufacturing Technology*, 95(1-4), pp. 157-169.

Gholizadeh, S., Leman, Z. and Baharudin, B.T.H.T., 2015. A review of the application of acoustic emission technique in engineering. *Struct. Eng. Mech*, 54(6), pp. 1075-1095.

Guo, Y.L., Ni, Y.Q. and Chen, S.K., 2017. Optimal sensor placement for damage detection of bridges subject to ship collision. *Structural Control and Health Monitoring*, 24(9), p. e1963.

He, K., Zhang, X., Ren, S., and Sun., J., 2015, Deep residual learning for image recognition. *CoRR*, abs/1512.03385.

Heidary, R., Gabriel, S.A., Modarres, M., Groth, K.M. and Vahdati, N., 2018. A Review of Data-Driven Oil and Gas Pipeline Pitting Corrosion Growth Models Applicable for Prognostic and Health Management. *Int. J. Progn. Health Manag*, 9(1), pp. 1-13.

Hou, R., Xia, Y., Xia, Q. and Zhou, X., 2019. Genetic algorithm based optimal sensor placement for L1-regularized damage detection. *Structural Control and Health Monitoring*, 26(1), p. e2274.

Huang, G., Liu, Z., and Weinberger, k.Q., 2016. Densely connected convolutional networks. *CoRR*, abs/1608.06993.

Iqbal, H., Tesfamariam, S., Haider, H. and Sadiq, R., 2017. Inspection and maintenance of oil & gas pipelines: a review of policies. *Structure and Infrastructure Engineering*, 13(6), pp.794-815.

Jégou, S., Drozdal, M., Vazquez, D., Romero, A. and Bengio, Y., 2017. The one hundred layers tiramisu: Fully convolutional denseNets for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (pp. 11-19).

Keshtgar, A. and Modarres, M., 2013, January. Acoustic emission-based fatigue crack growth prediction. In *2013 Proceedings Annual Reliability and Maintainability Symposium (RAMS)* (pp. 1-5). IEEE.

Keshtgar, A. and Modarres, M., 2013. Detecting crack initiation based on acoustic emission. *Chemical Engineering*, 33, p. 103-111.

Keshtgar, A., Sauerbrunn, C. and Modarres, M., 2018. Structural Reliability Prediction Using Acoustic Emission-Based Modeling of Fatigue Crack Growth. *Applied Sciences*, 8(8), p. 1225.

Khan, S. and Yairi, T., 2018. A review on the application of deep learning in system health management. *Mechanical Systems and Signal Processing*, 107, pp. 241-265.

Kim, N.H., An, D. and Choi, J.H., 2016. *Prognostics and health management of engineering systems: An introduction*. springer.

- Knop, K., Olejarz, E. and Ulewicz, R., 2019, May. Evaluating and Improving the Effectiveness of Visual Inspection of Products from the Automotive Industry. In International Scientific-Technical Conference MANUFACTURING (pp. 231-243). Springer, Cham.
- Koch, G., Varney, J., Thompson, N., Moghissi, O., Gould, M. and Payer, J., 2016. International measures of prevention, application, and economics of corrosion technologies study. NACE International IMPACT Report.
- Kruskal, J.B., 1956. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society*, 7(1), pp. 48-50.
- Lei, Y., Li, N., Guo, L., Li, N., Yan, T. and Lin, J., 2018. Machinery health prognostics: A systematic review from data acquisition to RUL prediction. *Mechanical Systems and Signal Processing*, 104, pp. 799-834.
- Lewandowski, D., 2002. Gas pipelines corrosion data analysis and related topics, M.S. thesis, Delft University of Technology, Delft, Netherlands.
- Li, X., Ding, Q. and Sun, J.Q., 2018. Remaining useful life estimation in prognostics using deep convolution neural networks. *Reliability Engineering & System Safety*, 172, pp. 1-11.
- Lipton, Z.C., Berkowitz, J. and Elkan, C., 2015. A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv:1506.00019*.
- Liu, X., Zheng, J., Fu, J., Nie, Z. and Chen, G., 2018. Optimal inspection planning of corroded pipelines using BN and GA. *Journal Of Petroleum Science And Engineering*, 163, pp. 546-555.
- Long, J., Shelhamer, E., and Darrell, T., 2015. Fully convolutional networks for semantic segmentation. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR).
- Loutas, T., Eleftheroglou, N. and Zarouchas, D., 2017. A data-driven probabilistic framework towards the in-situ prognostics of fatigue life of composites based on acoustic emission data. *Composite Structures*, 161, pp. 522-529.
- Modarres, C., Astorga, N., Droguett, E.L. and Meruane, V., 2018. Convolutional neural networks for automated damage recognition and damage type identification. *Structural Control and Health Monitoring*, 25(10), p. e2230.
- Okeniyi, J.O., Ambrose, I.J., Okpala, S.O., Omoniyi, O.M., Oladele, I.O., Loto, C.A. and Popoola, P.A.I., 2014. Probability density fittings of corrosion test-data: Implications on C6H15NO3 effectiveness on concrete steel-rebar corrosion. *Sadhana*, 39(3), pp. 731-764.
- Ossai, C.I., Boswell, B. and Davies, I.J., 2015. Predictive modelling of internal pitting corrosion of aged non-piggable pipelines. *Journal of The Electrochemical Society*, 162(6), pp. C251-C259.

Ostachowicz, W., Soman, R. and Malinowski, P., 2019. Optimization of sensor placement for structural health monitoring: a review. *Structural Health Monitoring*, 26(1), DOI:10.1177/1475921719825601

Ostfeld, A., Uber, J.G., Salomons, E., Berry, J.W., Hart, W.E., Phillips, C.A., Watson, J.P., Dorini, G., Jonkergouw, P., Kapelan, Z. and di Pierro, F., 2008. The battle of the water sensor networks (BWSN): A design challenge for engineers and algorithms. *Journal of Water Resources Planning and Management*, 134(6), pp.556-568.

Papavinasam, S., 2013. Corrosion control in the oil and gas industry. Elsevier.

Perumal, K.E., 2014. Corrosion risk analysis, risk based inspection and a case study concerning a condensate pipeline. *Procedia Engineering*, 86, pp. 597-605.

Pourgol-Mohammad, M., 2007. Integrated methodology for thermal-hydraulics uncertainty analysis (IMTHUA) Doctoral dissertation, University of Maryland, College Park, MD.

Rabiei, E., Droguett, E.L. and Modarres, M., 2018. Fully adaptive particle filtering algorithm for damage diagnosis and prognosis. *Entropy*, 20(2), p. 100-110.

Rabiei, E., Droguett, E.L. and Modarres, M., 2016. A prognostics approach based on the evolution of damage precursors using dynamic Bayesian networks. *Advances in Mechanical Engineering*, 8(9), p. 230-243.

Rabiei, E., Droguett, E.L., Modarres, M. and Amiri, M., 2015. Damage precursor based structural health monitoring and damage prognosis framework. *Safety and Reliability of Complex Engineered Systems*, 23, pp. 2441-2449.

Rabiei, M. and Modarres, M., 2013. Quantitative methods for structural health management using in situ acoustic emission monitoring. *International Journal of Fatigue*, 49, pp. 81-89.

Rosenblatt, F., 1961. *Principles of neurodynamics perceptrons and the theory of brain mechanisms* (No. VG-1196-G-8). Cornell Aeronautical Lab Inc, Buffalo NY.

Saaty, T.L., 2008. Decision making with the analytic hierarchy process. *International journal of services sciences*, 1(1), pp. 83-98.

Salkin, H.M., 1975. *Integer programming* (Vol. 4). Reading, MA: Addison-Wesley.

Sause, M.G., Linscheid, F.F. and Wiehler, M., 2018. An experimentally accessible probability of detection model for acoustic emission measurements. *Journal of Nondestructive Evaluation*, 37(2), p. 17.

Schneider, C.A., Rasband, W.S., Eliceiri, K.W. 2012. "NIH Image to ImageJ: 25 years of image analysis". *Nature Methods* 9, pp. 671-675.



Shabarchin, O. and Tesfamariam, S., 2016. Internal corrosion hazard assessment of oil & gas pipelines using Bayesian belief network model. *Journal of loss prevention in the process industries*, 40, pp. 479-495.

Shafiee, M. and Sørensen, J.D., 2017. Maintenance Optimization and Inspection Planning of Wind Energy Assets: Models, Methods and Strategies. *Reliability Engineering & System Safety*, 1(1), pp. 1-19.

Shateri, M., Ghaib, M., Svecova, D. and Thomson, D., 2017. On acoustic emission for damage detection and failure prediction in fiber reinforced polymer rods using pattern recognition analysis. *Smart Materials and Structures*, 26(6), p. 17.

Smith, R., Modarres, M. and Droguett, E.L., 2018. A recursive Bayesian approach to small fatigue crack propagation and detection modeling. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, 232(6), pp. 738-753.

Tawarmalani, M. and Sahinidis, N.V., 2005. A polyhedral branch-and-cut approach to global optimization. *Mathematical Programming*, 103(2), pp. 225-249.

Terpstra, S., 2009, June. Use of statistical techniques for sampling inspection in the oil and gas industry. In 4th European-American Workshop on Reliability of NDE, Berlin.

Tian, Z., 2012. An artificial neural network method for remaining useful life prediction of equipment subject to condition monitoring. *Journal of Intelligent Manufacturing*, 23(2), pp. 227-237.

Valor, A., Alfonso, L., Caleyó, F., Vidal, J., Perez-Baruch, E. and Hallen, J.M., 2015. The negative binomial distribution as a model for external corrosion defect counts in buried pipelines. *Corrosion Science*, 101, pp. 114-131.

Wagstaff, K., Cardie, C., Rogers, S. and Schrödl, S., 2001, June. Constrained k-means clustering with background knowledge. In International Conference of Machine Learning (ICML), 1, pp. 577-584.

Wang, H., Yajima, A., Liang, R.Y. and Castaneda, H., 2016. Reliability-based temporal and spatial maintenance strategy for integrity management of corroded underground pipelines. *Structure and Infrastructure Engineering*, 12(10), pp. 1281-1294.

Wilks, S.S., 1941. Determination of sample sizes for setting tolerance limits. *The Annals of Mathematical Statistics*, 12(1), pp. 91-96.

Wu, R.T. and Jahanshahi, M.R., 2018. Data fusion approaches for structural health monitoring and system identification: Past, present, and future. *Structural Health Monitoring*, DOI: [10.1177/1475921718798769](https://doi.org/10.1177/1475921718798769).

Yu, Y., Wang, C., Gu, X. and Li, J., 2019. A novel deep learning-based method for damage identification of smart building structures. *Structural Health Monitoring*, 18(1), pp. 143-163.

Yun, H., 2019. Entropic approaches for assessment of metal fatigue damage. PhD dissertation, University of Maryland, College Park.

Yun, H., Kahirdeh, A., Christine, M. and Modarres, M., 2018, January. Entropic Approach to Measure Damage with Applications to Fatigue Failure and Structural Reliability. In *2018 Annual Reliability and Maintainability Symposium (RAMS)* (pp. 1-6). IEEE.

Zhang, S. and Zhou, W., 2014. Cost-based optimal maintenance decisions for corroding natural gas pipelines based on stochastic degradation models. *Engineering Structures*, 74, pp. 74-85.

Zhao, R., Yan, R., Chen, Z., Mao, K., Wang, P. and Gao, R.X., 2019. Deep learning and its applications to machine health monitoring. *Mechanical Systems and Signal Processing*, 115, pp. 213-237.

Zheng, S., Ristovski, K., Farahat, A. and Gupta, C., 2017, June. Long short-term memory network for remaining useful life estimation. In *2017 IEEE International Conference on Prognostics and Health Management (ICPHM)* (pp. 88-95). IEEE.

Zulkifli S. N., Rahim H. A., Lau W. J., 2018. Detection of contaminants in water supply: A review on state-of-the-art monitoring technologies and their applications. *Sensors and Actuators B: Chemical*, 255, pp. 2657-2689.