OSPtk: Cost-aware Optimal Sensor Placement Toolkit Enabling Design-for-PHM in Critical Industrial Systems

Liang Tang, Abhinav Saxena, Scott Evans, Naresh Iyer and Helena Goldfarb

GE Global Research, Niskayuna, NY, 12309, USA

liang.tang@ge.com
asaxena@ge.com
evans@ge.com
iyerna@ge.com
goldfarb@ge.com

ABSTRACT

Performance of a Prognostics and Health Management (PHM) system in a fielded application depends on observability from existing monitoring equipment and sensing, which get determined at the design phase. Although various technologies have been proposed in the literature, there is currently a lack of known generic tools specifically designed for performing design stage sensor placement analyses from a PHM perspective. This leads to PHM observability being an afterthought and resulting PHM designs being sub-optimal. This paper describes a new Optimal Sensor Placement (OSP) framework, its implementation as a toolkit and the experience with applying it to a new product design in the context of a Small Modular Reactor (SMR). The formulation adds multiple important features that are critical to PHM applications. Firstly, it establishes a direct link to PHM performance requirements with intent to reduce operational and maintenance costs. Moreover, it acknowledges and accounts for the costs and risks of errors that PHM system will incur, and simultaneously considers operational requirements on sensing for performance, control and/or regulatory requirements. The toolkit described here implements formulations of a large number of requirements scenarios applicable in a generic industrial product development setting.

1. INTRODUCTION

The design and performance of PHM systems heavily depend on the observability provided by monitoring equipment and sensing capabilities. Typically, these considerations are addressed during the design phase of a product or system. However, product design teams often face time and cost limitations that hinder their ability to thoroughly analyze PHM requirements and incorporate them into the design. As a result, PHM considerations are often deferred to a later stage, or in many cases, become an afterthought.

Selecting the right sensor suite is a critical task in PHM design. However, it can be challenging due to several reasons:

1. System Complexity: Complex systems with interconnected components require in-depth understanding of design, operation, and failure modes to assess their impact on health and performance.
2. Lack of Standardization: There is no one-size-fits-all solution when it comes to sensor selection for PHM. Different assets and industries have unique requirements and specifications.
3. Multitude of Sensor Options: Evaluating diverse sensor technologies in terms of performance, accuracy, reliability, and cost-effectiveness is time-consuming.
4. Data Requirements and Processing: Selecting sensors providing accurate, high-quality data at the desired frequency, resolution, and reliability can be complex, considering compatibility with data acquisition and processing infrastructure.
5. Cost Considerations: Balancing upfront costs, installation, and maintenance expenses with potential benefits requires careful consideration, especially when cost constraints are significant.
6. Trade-offs and Prioritization: Prioritizing critical parameters impacting asset health and performance due to infeasibility or cost-effectiveness of monitoring all parameters is often necessary. Therefore, trade-offs with careful consideration of specific goals and constraints are often needed.
7. Limited Data and Experience: Insufficient historical data, absence of established best practices, and lack of

PHM expertise pose challenges in making informed decisions during sensor selection.

Overcoming these challenges requires a multidisciplinary approach involving domain experts, sensor manufacturers, data analysts, and PHM specialists. Collaboration, continuous learning, and iterative improvement are key to navigating the complexity of sensor selection and optimizing the effectiveness of PHM systems, which are often difficult to overcome, especially in large organizations. Lack of a standardized toolset for sensor selection and difficulty in putting together an effective multi-disciplinary team are key challenges to incorporating PHM into a product from its design phase.

Optimal sensor placement has been studied extensively over the past two decades, with researchers addressing various aspects of this topic. Starting with system analysis and specific operator requirements, Jones et al. (2018) considered performance, environmental constraints, and economic factors in selecting candidate sensors. Schmidt and Laerhoven (2001) identified context-discriminating variables and sensors selection based on accuracy and cost considerations. Tjen et al. (2020) employed an entropy-based sensor selection algorithm for structural damage detection. Shieh et al. (2001) provided an overview of sensor performance, showcasing sensors best suited for specific tasks. Kertiou et al. (2018) proposed a method applicable to different IoT middleware for designing accurate solutions with minimized search and selection time. Zhang and Vachtsevanos (2007) quantified fault detectability and optimized cost using particle swarm optimization. Riedel et al. (2015) presented a concept based on plant description and semantic models as well as integration into plant workflow. Kulkarni et al. (2021) provided a literature review and proposed an ordered fuzzy clustering approach to sensor selection. The Design-for-PHM concept has also been promoted in several published works (Kurtoglu et al., 2008; Bodden et al., 2005).

This paper introduces a new OSP process and formulation that integrates multiple important features to tackle some neglected aspects in PHM designs. Firstly, it establishes a direct link to PHM requirements, such as fault detection and isolation coverages, within the optimization framework. Secondly, unlike conventional methods that evaluate sensors separately, our approach incorporates the concept of Virtual Sensor, which combines sensor measurements using sensor fusion to improve overall performance. Thirdly, with the introduction of "sensor grouping" and associated constraints, a comprehensive set of "what-if" scenarios can be easily analyzed. Lastly, a vital connection is established in the OSP process to interact with cost-benefit models derived from cost analysis of maintenance burden data. Cost-benefit analysis is a crucial aspect which legacy approaches often fail to address adequately. The outcome of this work is a generic OSP toolkit that can be utilized in early system design phase to guide the selection and placement of sensors effectively.

The rest of the paper is organized as follows. First, a cost-aware, value-driven OSP process is presented in the next section. Major functions and supporting algorithms that enable the aforementioned key features are explained in detail. The development of a generic toolkit named OSPtk and its application to an SMR design are presented. The paper concludes with some further design considerations and remarks on future development directions.

2. A VALUE-DRIVEN OSP PROCESS

The success of a PHM application hinges on effectively justifying and communicating the costs versus benefits throughout the life cycle of the product or system. The sensor selection and placement process therefore need to be cost-aware and value-driven. In support of the Design-for-PHM concept, we propose an Optimal Sensor Placement (OSP) process as shown in Figure 1.

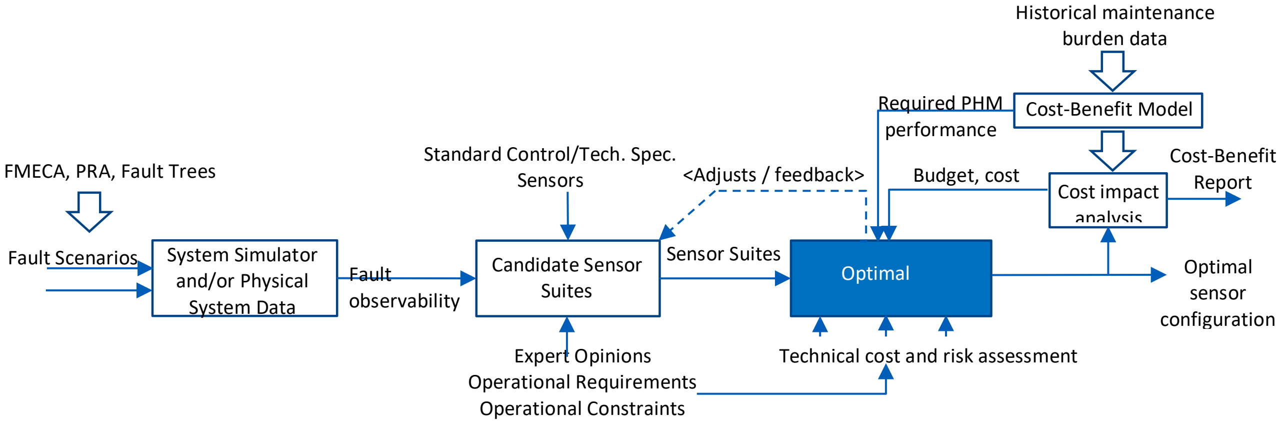


Figure 1. A value-driven OSP process

The process starts with analyzing system failure modes, effect, and criticality analysis (FMECA) information, and Probabilistic Risk Assessment (PRA) output. This analysis generates a list of fault scenarios that the PHM system will be designed to “cover” by observing fault symptoms with sensors. Note that in this paper, we will use the terms “failure” and “fault” interchangeably in the context of OSP. The authors acknowledge the differences in these terms when it comes to a broader PHM context.

The next step in this process is to analyze the symptoms of the faults and start considering candidate sensors to monitor them. This can be done through simulations (if available) or by analyzing actual data from the system (or similar systems). This analysis will generate fault observability measures, which quantify the level of confidence in detecting the faults using the candidate sensors. It is anticipated a large set of candidate sensors will be included in the initial candidate sensor suites.

The initial candidate sensor suites will undergo a thorough review by subject matter experts to assess their technical soundness. Additionally, important "meta" data, such as fault criticality, failure rate, and other relevant factors, will be incorporated. Mandatory sensors, necessary for controls, certification, or regulatory requirements, will also be included to determine if they contribute to the overall effectiveness of PHM.

The initial candidate sensors, fault modes, and cost considerations outlined in the Cost Impact Analysis are provided to the generic OSP tool as a configuration file for optimization. The resulting optimal sensor configuration, along with key PHM performance metrics (such as fault coverage, false alert rate, true detection rate, etc.), are then utilized in the Cost Impact Analysis process to ensure that cost impact goals are met. Finally, a comprehensive cost-benefit report will be generated for management review, summarizing the outcomes and benefits of the sensor placement optimization.

To support this process, a generic toolkit called OSPtk has been developed. OSPtk allows designers to explore and test multiple what-if scenarios at a fast speed to iterate between options. Contrary to traditional processes where sensing is designed primarily for process control and operations, OSPtk facilitates advocacy of PHM centric needs and value proposition early into the design process. Trade-offs arising from quality, cost, and number of sensors to provide fault coverage prioritized by respective maintenance cost burdens can be analyzed quickly and outcomes can be incorporated in design decisions.

OSPtk also provides flexibility to express constraints or design preferences, incorporates cost and budget information, and eliminates the need for designers to possess expertise in formulating and solving optimization problems or even an in-depth PHM experience. Most importantly, it allows to establish a connection between performance requirements from a PHM system (presumably established from maintenance burden analysis from historical data) to capital costs of optimizing sensing for maximum fault coverage.

3. Technical Approach

Optimal sensor placement can be formulated as a constrained combinatorial optimization problem as follows. Given a sensor suite consisting of m candidate sensors, let S be a binary decision vector

$$S = [s_1, \ldots, s_m],\ s_i \in \{0,1\} \qquad (1)$$

where 1 indicates a sensor being selected, and 0 being not selected. The goal of OSP is to select a subset of sensors to optimize PHM design objectives, subject to requirements and budget constraints, i.e.,

$$\begin{aligned} &min \quad f(P(s), C(s)) \\ &s.t. \quad g(P(s), C(s)) \leq 0 \\ &\qquad\ \ U(s)=0 \\ &\qquad\ \ V(s) \leq 0 \end{aligned} \qquad (2)$$

where *f*, *g*, *U* and *V* are the objective, inequality constraint and equality constraint functions respectively. *P(s)* represents overall PHM performance (such as detection or isolation coverages) as a function of selected sensors. *C(s)* represents the overall costs associated with selected sensors including not only sensor hardware, installation, and maintenance costs, but also the costs for implementing the associated PHM functionalities. Function *g* is an inequality constraint based on PHM performance and costs. *U(s)* and *V(s)* represent the inequality and equality *sensor grouping* constraints that can be used to conduct various trade studies (as detailed in Section 3.3).

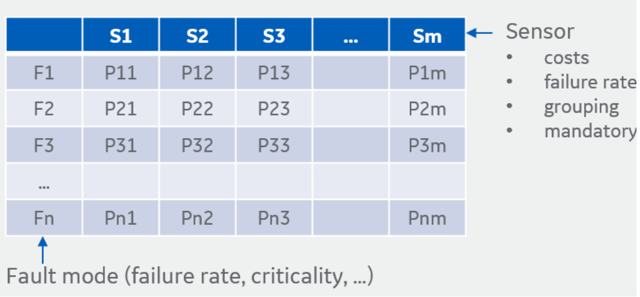
Note that all the functions (*f*, *g*, *U* and *V)* can be defined as vector-valued functions. Specifically, when *f* is defined as a vector-valued objective function, OSP takes the form of a multi-objective (or Pareto) optimization problem. However, linearly combining multiple objectives with weighting factors appears to be a practical alternative for balancing performance and cost considerations. Examples of *f* and *g* functions are shown in the use cases in Section 3.4.

The PHM performance metrics *P(s)* can be defined in various ways. We will introduce herein a method to calculate fault detection and fault isolation coverages using the Dependency Matrix (D-Matrix) as the underlying sensor/fault correlation model. This formulation allows a direct connection to PHM requirements in the OSP process.

3.1. The Dependency Matrix (D-Matrix)

The Dependency Matrix (D-Matrix) is a tool used in Testability Analysis to identify dependencies between system components and determine the best locations to place sensors for test purposes (MIL-STD-2165, 1985; Singh et al., 2010). The D-Matrix can be created in a number of different ways, for example, by subject matter experts, through simulation studies, direct inference from actual system data where available, or a combination of these. Once the D-Matrix is created, it can be used to analyze critical components and their fault modes in the system and determine the optimal locations to place sensors to detect and isolate them (Thombare and Dole, 2015). In earlier applications, the elements in D-Matrix are utilized to denote the correlation/dependency between a fault state (mode) and a test and are binary by definition. The interpretation of D-Matrix elements was later extended to represent a probability of detection, or a confidence value between 0 and 1 (Kulkarni et al., 2021).

In this work, the D-Matrix represents fault modes as rows and sensors as columns. Each element of the D-Matrix signifies a confidence measure (or *score*) that reflects how effectively the sensor can detect the fault symptoms. There are also meta data, such as fault criticality, failure rates, and costs, associated with faults and sensors defined as a part of the D-Matrix as shown in Figure 2. The design of a PHM system may necessitate distinct requirements for the coverage of critical and non-critical faults, which should be evaluated separately. Moreover, mandatory sensors also need to be included in the OSP process (with the option of zero or discounted costs) if they contribute to PHM.



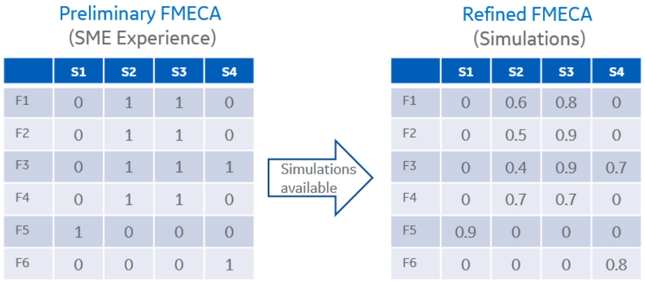
	S1	S2	S3	...	Sm
F1	P11	P12	P13		P1m
F2	P21	P22	P23		P2m
F3	P31	P32	P33		P3m
...					
Fn	Pn1	Pn2	Pn3		Pnm

Sensor
- costs
- failure rate
- grouping
- mandatory

Fault mode (failure rate, criticality, ...)

Figure 2. D-Matrix and associated meta data

It is noteworthy that the form of the D-Matrix can evolve during different stages of development as additional information becomes available. The OSPtk allows to iterate through these scenarios quickly as they progressively evolve. As illustrated in Figure 3, binary values were initially assigned by subject matter experts at an early stage. These values were later refined to represent fault detection confidence when simulation studies could be performed. The OSP framework's ability to accommodate diverse forms, preferably a mixture of them coexisting in the same D-Matrix, is crucial because knowledge of different subsystems may not be acquired at the same time. Multiple iterations of the OSP process are expected throughout the development cycle.



Preliminary FMECA
(SME Experience)

	S1	S2	S3	S4
F1	0	1	1	0
F2	0	1	1	0
F3	0	1	1	1
F4	0	1	1	0
F5	1	0	0	0
F6	0	0	0	1

Simulations available

Refined FMECA
(Simulations)

	S1	S2	S3	S4
F1	0	0.6	0.8	0
F2	0	0.5	0.9	0
F3	0	0.4	0.9	0.7
F4	0	0.7	0.7	0
F5	0.9	0	0	0
F6	0	0	0	0.8

Figure 3. The evolution of a D-Matrix

3.1.1. Gaussian Naïve Bayes Method for Constructing D-Matrix

While there are multiple ways to construct a D-Matrix and interpret it accordingly, this paper illustrates a simplistic yet systematic method that utilizes simulation data to provide a D-matrix *score* reflecting individual sensor's ability to detect faults using Gaussian Discriminant Analysis. This scoring methodology is then extended using Gaussian Naïve Bayes and multi-feature classification algorithms to provide scores for multiple sensors. This approach allows a rapid, preliminary assessment of the sensor's ability to observe a fault without conducting an extensive study.

The concept is shown in Figure 4, where two sensors, s1 and s2 are considered. The marginal distributions are shown with respect to their ability to detect a given fault (indicated by orange points and label = 1). It should be noted that the sensor data used in the analysis may not necessarily be raw data but can also include normalized or calculated features derived from the raw sensor data.

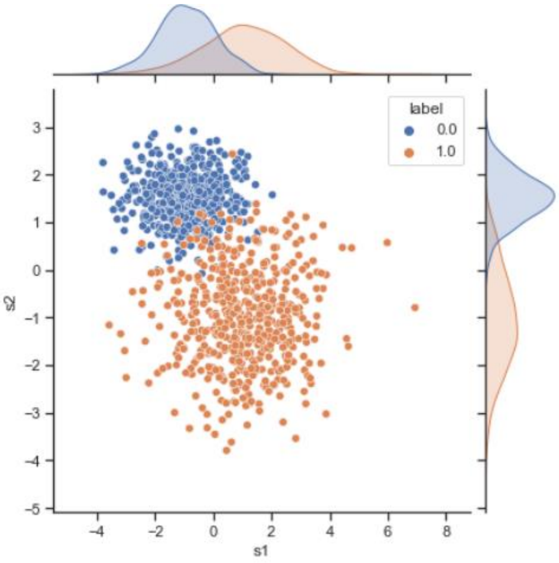


Figure 4. Sensors 1 and 2 scatterplot for health (blue) and faulty (orange) data

In this example, neither s1 nor s2 alone could provide satisfactory fault detection performance due to the overlap of distributions evident in the plot. However, if s1 and s2 are considered collectively, a Naïve Bayes classifier can be trained using labelled training data. Figure 5 below shows the Receiver Operating Characteristic (ROC) curves for fault detection methods based on s1, s2 and "s1 & s2". The fusion has clearly improved fault detection performance. This example also illustrates a scenario where individual sensors may not provide sufficient detection capability while the fused "Virtual Sensor", in this case the Naïve Bayes classifier, can offer required fault detection performance.

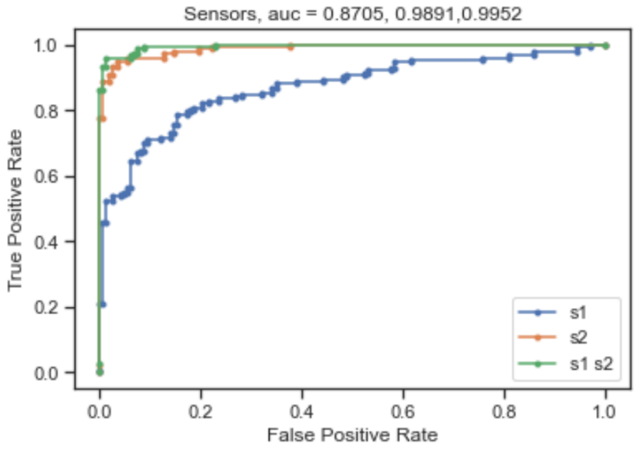


Figure 5. ROC curves for individual and group of sensors

When considering the ROC curve, the Area Under the Curve (AUC) provides a comprehensive evaluation of a sensor's capability to differentiate between fault and normal conditions. Since an AUC of 0.5 indicates a random guess, a reasonable way to define D-Matrix score can be,

$$Score = 2 \times (AUC - 0.5) \quad (3)$$

Alternatively, it can also be defined as the True Positive Rate at a particular False Positive Rate (e.g., 0.05) on the ROC curve.

Once the D-Matrix is constructed, PHM performance metrics can be evaluated based on D-Matrix values and relevant faults/sensor attributes. As an example, we will show how fault detection coverage (FDC) can be calculated. It is, however, possible to expand the D-Matrix form to tackle fault isolation coverage (FIC) and other PHM performance metrics.

3.1.2. Fault Detection Coverage

Given a candidate sensor suite and the decision variable S as shown in Eq. (1), the fault detection confidence for fault mode F_i can be defined as,

$$P(F_i) = \max\,(P_{i1} \cdot s_1, \dots, P_{im} \cdot s_m) \quad (4)$$

At the requirement level, a fault mode F_i is considered "covered" when its detection confidence $P(F_i)$ is above a user defined threshold (e.g., 0.95). System overall fault detection coverage (FDC) is defined as the percentage of covered fault modes, i.e.,

$$FDC = \frac{F_C}{F_T} * 100\% \quad (5)$$

where F_C is the number of covered faults, and F_T is the total number of faults. Note that fault criticality can be factored into the above equation to weight more heavily on critical faults.

3.2. Virtual Sensor

Naïve Bayes based virtual sensor construction is already described in Section 3.1.1. Virtual sensors are added to the D-Matrix as additional columns and are treated by the optimization algorithm the same way as physical sensors with the following exceptions:

1. When the binary decision variable s_i for a virtual sensor is set, the decision variables for all physical sensors (that are used by this virtual sensor) are automatically set.
2. Sensor hardware related costs are only counted for physical sensors.

The Naïve Bayes based approach can (theoretically) be applied to all sensor combinations to explore sensor fusion benefits, but a heuristic approach that considers a subset of higher potential combinations are often more practical.

3.3. Sensor Grouping & Constraints

Sensor grouping is a new concept introduced to allow various trade studies such as:

- sensors placed at different locations
- high accuracy/cost vs. low accuracy/cost sensors
- a high-cost sensor vs. multiple low-cost sensors

For example, if a temperature sensor can be placed at three different locations, each with its pros and cons in terms of performance and costs, then a sensor group can be defined for these three locations, e.g. group 9, and an equality constraints can be formulated as,

$$U(s, 9) - 1 = 0 \quad (6)$$

where function *U(s,9)* returns the number of sensors in decision variable *s* that belong to "Sensor Group 9". If a sensor group is defined by a *masking* vector, i.e.,

$$Z_i = [z_{i,1}, \dots, z_{i,m}],\ z_{i,j} \in \{0,1\} \quad (7)$$

where $z_{i,m} = 1$ indicates sensor s_m belongs to sensor group i, then the sensor grouping constraint in Eq. (6) can be rewritten using a dot product formula:

$$S \cdot Z_9 = 1 \quad (8)$$

User can try to define the equality and inequality constraint functions (*U* and *V*) for various "what-if" scenarios, e.g., "pick one sensor from Group 1 and at least 2 sensors from Group 2"[1].

3.4. Use Cases for Optimization Objectives, Constraints and Trade Studies

While most legacy approaches define their own specific objective functions and constraints, this new approach leaves that flexibility to the end user. As a result, PHM system designers can formulate the optimization process based on their own specific program needs. Table 1 below shows several use cases with the corresponding objective and constraints functions.

Table 1. User-defined optimization objective and constraint functions.

Use case	Formulation
Optimize cost efficiency while maintaining fault detection and isolation coverages above 95%	$\min \; C(s)$ $\text{s.t.} \; FDC(s) \geq 95\%$, $FIC(s) \geq 95\%$
Achieve maximum average fault detection and isolation coverages while adhering to a budget limit of $1M, ensuring detectability of all critical faults	$\min \; -FDC(s) - FIC(s)$ $\text{s.t.} \; C(s) \leq \$1M$, $FDC_{critical}(s) = 100\%$
Optimize the weighted average of cost, FDC and FIC	$\min \; C(s) - FDC(s) - FIC(s)$

These mathematical formulae can be conveniently described using a human-readable data object file format, such as JSON (JavaScript Object Notation). For example, the third use case above can be described as shown in Figure 6. In this case, if the unit for cost is $1M, FDC and FIC are in percent, then the formula reflects the trade-off that 1% improvement in FDC or FIC is equivalent to $1M savings in maintenance over a certain defined operation period.

```
"Objective": [ { "ID": "O1",
                 "Metric": "FDC",
                 "Type": "Overall",
                 "Weight": -1   },
               { "ID": "O2",
                 "Metric": "FIC",
                 "Type": "Overall",
                 "Weight": -1   },
               { "ID": "O3",
                 "Metric": "Cost",
                 "Type": "NA",
                 "Weight": 1 }   ]
```

Figure 6. Objective function in JSON format

Similarly, various trade studies can be conducted using sensor group constraints and sensor meta data specifications. A few examples are illustrated in Table 2 in Appendix using JSON format.

3.5. Optimization Algorithm

The problem setting presents several challenges that need to be addressed in the development of the optimization algorithm. Firstly, the algorithm must accommodate a generic objective function that is not pre-established, as objectives are often different across different industry use cases. Moreover, the utilization of undifferentiable functions, such as the *max* operator and thresholding in the computation of certain performance metrics like fault detection coverage, adds further complexity to the optimization process.

Exhaustive search is a viable option for achieving a global optimal solution when dealing with a small problem space. However, as the search space expands, the "curse of dimensionality" makes the brute force method impractical. In such cases, many *metaheuristic* algorithms such as ant colony optimization, genetic and evolutionary algorithms, iterated local search, simulated annealing and tabu search can be applied (Blum et al., 2011). The toolkit aims to offer a variety of algorithmic options. So far, the Genetic Algorithms (GA) (Mitchell, 1996) have been implemented and thoroughly tested. The individuals of the initial population can be randomized uniformly with a value between 0 and $2^m - 1$ and be treated as a binary vector. Mutation and crossover operations are applied to the binary vector thereafter following standard GA method. A select few elite individuals are preserved from one generation to the next. Penalty functions are employed to discourage and penalize combinations that violate the constraints. The exit criteria can be based on the convergence of the objective function, total number of generations, total computing time, or any combinations of these conditions as specified by the designer. Standard GA software libraries can be utilized to implement this algorithm, and our particular implementation, therefore, is not described here any further.

3.6. Cost-Benefit Model

While the OSP problem can be solved to maximize predictive analytics performance without additional constraints, we strongly advocate for a solution that is guided by the necessary real-world requirements of predictive performance for the specific failure modes of interest. This approach ensures a more accurate and focused solution. Such necessary performance requirements are generated by leveraging a cost-benefit model built to account for the utility of the overall PHM system: costs required to develop, operate, and sustain

[1] A possible formulation is: $S \cdot Z_1 - 1 = 0$ and $S \cdot Z_2 - 2 \geq 0$, where Z_1 and Z_2 are the masking vectors that define sensor groups 1 and 2 respectively.

the system as well as the benefits incurred from the generated predictions. This accounting is expected to result in net positive savings, which would then justify investment in the development of a PHM system. We believe that such a cost-benefit analysis will only permit true insights about the value of the PHM system if it also takes into account the costs incurred due to model errors (i.e., false alarms, false negatives).

Cost-benefit analysis can be done at the level of the overall system/asset or for individual failure modes. This is done by making use of available historical cost data that shows the overall costs incurred by the application of existing paradigms when not utilizing predictive analytics or PHM. When cost-benefit analysis is done at the level of individual failure modes, it will make use of historical data to not only identify which failures to target for OSP, but it will also create minimum requirements on the predictive performance of the PHM models, in order for the expected savings to materialize, even in the presence of model errors. These performance requirements now provide OSP with additional constraints to address in the design of the optimal sensing system. Since the minimum performance requirements are driven by real-world analysis of expected savings, it provides the OSP solution with the option to be no better than those minimum requirements, thus allowing OSP to exercise better cost tradeoffs as dictated by the real application.

In our application of OSPtk to SMR use case, we have been leveraging such a cost-benefit model to derive required PHM performance given operations and maintenance (O&M) cost reduction targets. Historical maintenance burden data from existing nuclear plant fleets was used to derive PHM performance targets towards specific O&M cost reduction goals. This allows for determining whether desired cost reductions can be achieved, and if yes, what would it take to implement such a design.

4. OSPTK SOFTWARE DEVELOPMENT

As shown in Figure 7, the OSPtk software package consists of a graphical user interface (GUI) and an optimizer. The GUI provides a user-friendly interface for various tasks, including creating and managing OSP projects, visualizing system topology, defining sensor and fault mode metadata, accessing and modifying the D-Matrix, managing virtual sensors, setting optimization objectives and constraints, selecting optimization algorithms, configuring optimization parameters, submitting OSP jobs, and reviewing results, among other functionalities. The GUI produces an OSP project file in JSON format and passes it to the optimizer.

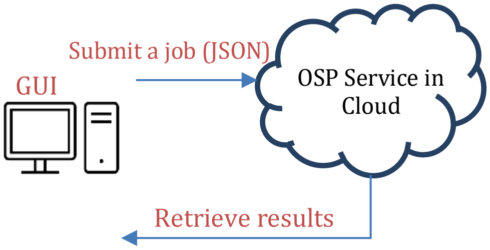


Figure 7. The OSP software package

The optimizer comes in two versions: a cloud-based version that harnesses the power of parallel computing to tackle large-scale problems, and a desktop version designed for small-scale applications. When the cloud-based version is utilized, an OSP job can be submitted with a JSON project file.

5. APPLICATION

OSPtk is being evaluated for sensor placement in the BWRX-300 small modular reactor for PHM design. The BWRX-300, as shown in Figure 8, is a 300 MWe water-cooled, natural circulation SMR with passive safety systems.

These small modular reactors will be generated from a distributed fleet of smaller reactors where units can be brought online or offline as needed. Due to its considerable operational and maintenance costs a distributed fleet will put additional cost burden if remote monitoring is not enabled. This requires prognostics and health management capabilities such as early warning, diagnostics, and prognostics to enable predictive maintenance with high accuracy. Which in turn requires careful selection of sensors and determine their optimal placement.

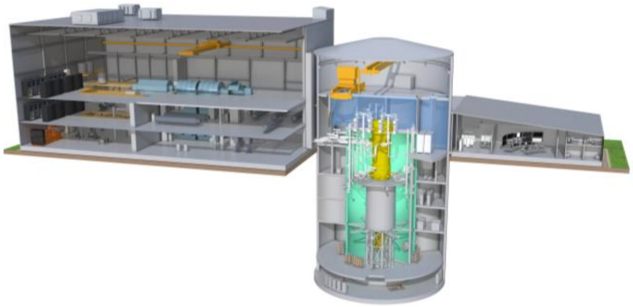


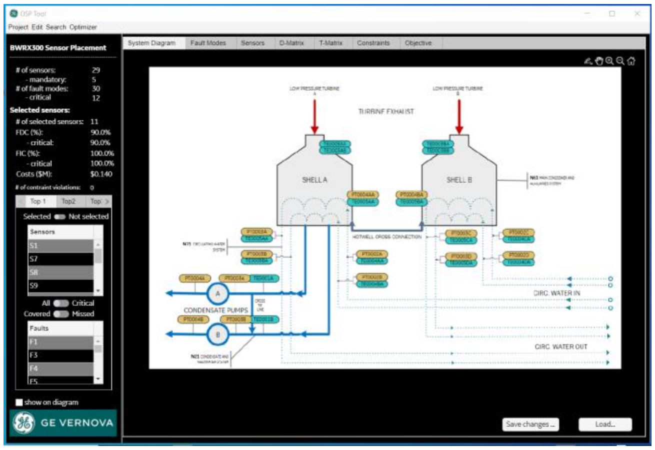
Figure 8. The BWRX-300 Small Modular Reactor[2]

For the OSPtk demonstration, the condenser system was selected as the focal subsystem. In a nuclear reactor such as BWRX-300, the condenser is a large heat exchanger designed to cool exhaust steam from a turbine so that it can be returned to the heat source as water. A circulating water system takes

[2] Image courtesy of GE Hitachi. Source: https://nuclear.gepower.com/build-a-plant/products/nuclear-power-plants-overview/bwrx-300

the heat from the condenser and releases it into the surroundings. Figure 9 shows a top-level diagram of a representative condenser system in the OSPtk GUI.

A list of failure modes was considered for this study including condenser fouling and pipe blockage at various degrees of degradation. Condenser system software simulations (outside OSPtk) were conducted to simulate nominal conditions, as well as different faulty scenarios at varying severity levels and different environmental conditions. Additionally, nominal levels of noise, as determined from past experience, were introduced in the simulation runs. Both healthy and faulty data were collected, and the Naïve Bayes method described in Section 3.1.1 was applied to construct the D-Matrix. To create balanced healthy and faulty data, a Design of Experiments (DoE) method was applied to design the simulation conditions as shown in Table 3 using condenser fouling as an example. Environmental conditions (such as lake water temperatures) were simulated based on historical data from selected reactor sites.


BWRX300 Sensor Placement

# of sensors: 29
- mandatory: 5
# of fault modes: 30
- critical 12

Selected sensors:

# of selected sensors: 11
FDC (%): 90.0%
- critical: 90.0%
FIC (%): 100.0%
- critical 100.0%
Costs ($M): $0.140
# of contraint violations: 0

Figure 9. The BWRX condenser system in OSPtk GUI

Table 3. Condenser fouling experiment design example

Run	Combination	Factors	
		Health	Lake water Temperatures
1	Normal-1C	Normal	1 °C
2	Fouling5%-1C	Fouling 5%	1 °C
…	…	…	…
39	Normal-24C	Normal	24 °C
40	Fouling40%-24C	Fouling 40%	24 °C

Additional sensors and sensor locations were identified through the condenser system simulation program, expanding the candidate sensor suite to encompass 111 sensors. This represents an order of magnitude increase compared to the original number of sensors in the initial design. By implementing an automated process, an extensive exploration of sensor combinations was conducted, resulting in the discovery and integration of 11 additional virtual sensors into the candidate sensor suite. Figure 10 in Appendix displays the D-Matrix values, represented with a color code where darker shades indicate higher values, for four levels of condenser fouling faults (5%, 10%, 20%, and 40%). It can be observed that several different combinations of sensors (such as VS_S53_15_51) provided improved fault detection coverage for 40% fouling (FL40). These candidates are provided to the design team for considerations and iterated as needed. The sensor combinations (virtual sensors) identified in this study also provided valuable insights for the development of advanced PHM analytics. While we are continuing to iterate on existing and new scenarios as part of development and testing of this tool, the main intent is to mature this tool and hand over to the design teams where they can exercise analyses throughout their design and development cycle as consistent with their respective schedules.

In term of computation speed, with a limited number of failure modes, the GA optimization found optimal solutions within 300 generations in less than 1 minute using an x64-based laptop equipped with an Intel Xeon 3.2 GHz CPU with 6 cores. The evaluation of the toolkit for high-dimensional problems is currently underway.

6. CONCLUSION & FUTURE WORK

The concept of Design-for-PHM represents a significant advancement in system design, emphasizing the need for the early integration of PHM considerations during the design phase. In order to foster this shift in mindset, a generic, cost-aware OSP toolkit has been developed to support the selection and placement of sensors, which is a critical step in PHM design. The toolkit provides unparalleled flexibility, allowing designers to define the optimization problem according to their specific program requirements. Additionally, it offers a platform for conducting diverse trade studies, enabling designers to explore different possibilities and make informed decisions. The toolkit is currently undergoing evaluation for sensor placement and PHM system design in the context of a small modular reactor.

Planned future work includes evaluating the toolkit on a larger scale, incorporating a broader range of failure modes and subsystems. Additionally, the investigation of more computationally efficient combinatorial optimization algorithms is on the agenda. More sophisticated multi-variate machine learning methods will be explored as additional options to enhance over the Naïve Bayes classification method. The consideration of customized plug-ins for seamless integration with specialized simulation software (e.g., sensor meta data retrieval, simulation execution) is also part of the plan.

ACKNOWLEDGEMENT

The information, data, or work presented herein was funded in part by the Advanced Research Projects Agency-Energy (ARPA-E), U.S. Department of Energy, under Award Number DE-AR0001290. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

REFERENCES

Blum, C., Puchinger, J., Raidl, G. R., Roli, A., Hybrid metaheuristics in combinatorial optimization: A survey, *Applied Soft Computing*, Volume 11, Issue 6, 2011, Pages 4135-4151.

Bodden, D. S., Hadden, W., Grube, B. E. and Clements, N. S., 2005. PHM as a Design Variable in Air Vehicle Conceptual Design, *2005 IEEE Aerospace Conference*, Big Sky, MT, USA, 2005, pp. 1-11, doi: 10.1109/AERO.2005.1559640.

Jones, P.M.; Lonne, Q.; Talaia, P.; Leighton, G.J.T.; Botte, G.G.; Mutnuri, S.; Williams, L. A Straightforward Route to Sensor Selection for IoT Systems. *Res. Technol. Manag.* 2018, 61, 41–50.

Kertiou, I.; Benharzallah, S.; Kahloul, L.; Beggas, M.; Euler, R.; Laouid, A.; Bounceur, A. A dynamic skyline technique for a context-aware selection of the best sensors in an IoT architecture. *Ad Hoc Netw.* 2018, 81, 183–196.

Kulkarni, A., Terpenny, J. and Prabhu, V., 2021. Sensor selection framework for designing fault diagnostics system. *Sensors*, 21(19), p.6470.

Kurtoglu, T., Johnson, S., Barszcz, E., Johnson, J., and Robinson, P., 2008. Integrating system health management into the early design of aerospace systems using Functional Fault Analysis. 1-11. 10.1109/*PHM.2008*.4711425.

MIL-STD-2165, *Testability Program for Electronic Systems and Equipments*, 26 January 1985.

Mitchell, M., 1996. *An Introduction to Genetic Algorithms*. Cambridge, MA: MIT Press. ISBN 9780585030944.

Riedel, M.; Arroyo, E.; Fay, A. Knowledge-based selection of principle solutions for sensors and actuators based on standardized plant description and semantic concepts. In *Proceedings of the 2015 IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA)*, Luxembourg, 8–11 September 2015; pp. 1–8.

Schmidt, A.; van Laerhoven, K. How to build smart appliances? *IEEE Pers. Commun.* 2001, 8, 66–71.

Shieh, J.; Huber, J.E.; Fleck, N.A.; Ashby, M.F. The selection of sensors. *Prog. Mater. Sci.* 2001, 46, 461–504.

Singh, S., Holland, S.W., Bandyopadhyay, P.: Trends in the development of system-level fault dependency matrices. In *Proceedings of IEEE Aerospace Conference*, pp. 1–9 (2010)

Thombare, T.R.; Dole, L. D-Matrix: Fault Diagnosis Framework. *Int. J. Innov. Res. Comput. Commun. Eng.* 2015, 3, 1740–1745.

Tjen, J.; Smarra, F.; D'Innocenzo, A. An entropy-based sensor selection algorithm for structural damage detection. In *Proceedings of the 2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*, Virtual. 20–21 August 2020; pp. 1566–1571.

Zhang, G.; Vachtsevanos, G. A Methodology for Optimum Sensor Localization/Selection in Fault Diagnosis. In *Proceedings of the 2007 IEEE Aerospace Conference*, Big Sky, MT, USA, 3–10 March 2007.

BIOGRAPHIES

Liang Tang is a Principal Scientist in AI/Computer Vision at GE Global Research with over twenty years of experience in gas turbine fault diagnostics, prognostics, data analytics, and AI/ML applications for PHM. Liang has published over sixty papers and book chapters, and holds multiple patents related to engine health management applications. Liang is actively engaged in technical committees and standards organizations such as the SAE, ASME and PHM society. He also serves as Associate Editor for multiple journals including the ASME Open Journal and the International Journal of PHM.

Abhinav Saxena is a Principal Scientist in AI/Computer Vision at GE Global Research. Abhinav has been developing ML/AI-based PHM solutions for various industrial systems (aviation, nuclear, power, and healthcare) at GE and has been driving integration of AI-based PHM analytics in GE's industrial systems. He is the PI for ARPA-E GEMINA program led by GE Research on AI-Enabled Predictive Maintenance Digital twins for Advanced Nuclear Reactors. Abhinav is also an adjunct professor in the Division of Operation and Maintenance Engineering at Lulea ˚University of Technology, Sweden. Prior to GE, Abhinav was a Research Scientist with SGT Inc. at NASA Ames Research Center for over seven years. Abhinav's interests lie in developing PHM methods and algorithms with special emphasis on deep learning and data- driven methods in general for practical prognostics. Abhinav has published over 100 peer reviewed technical papers and has co-authored a seminal book on prognostics. He actively participates in several SAE standards committees, IEEE prognostics standards committee, and various PHM Society educational activities, and is a Fellow of the PHM Society. He also served as chief editor of International Journal of Prognostics and Health Management between 2011-2020. Abhinav actively participates in organization of PHM Society conferences and various AI workshops on topics of Digital Twins and AI in Industrial applications.

Scott Evans is a Principal Scientist AI-Machine Learning in the AI-ML-Robotics lab of the GE Global Research. Scott has 54 issued patents and has published over 50 papers in the areas of Machine Learning and AI. Scott's current research areas of interest include Causal AI, Green AI and Energy Aware Machine Learning. Before joining GE, Scott served as a Nuclear Trained Submarine Officer in the US Navy.

Naresh S. Iyer is a Principal Scientist in the AI-ML-Robotics lab of the GE Global Research. He has 22 years of experience in the research and application of machine learning to a variety of industry problems, including asset life prognostics, multi-objective optimization and decision making under uncertainty. He has developed solutions for a diverse range of industrial PHM applications using methods in supervised, unsupervised, semi-supervised learning and evolutionary soft computing. He has over 30 publications and 45 patent filings.

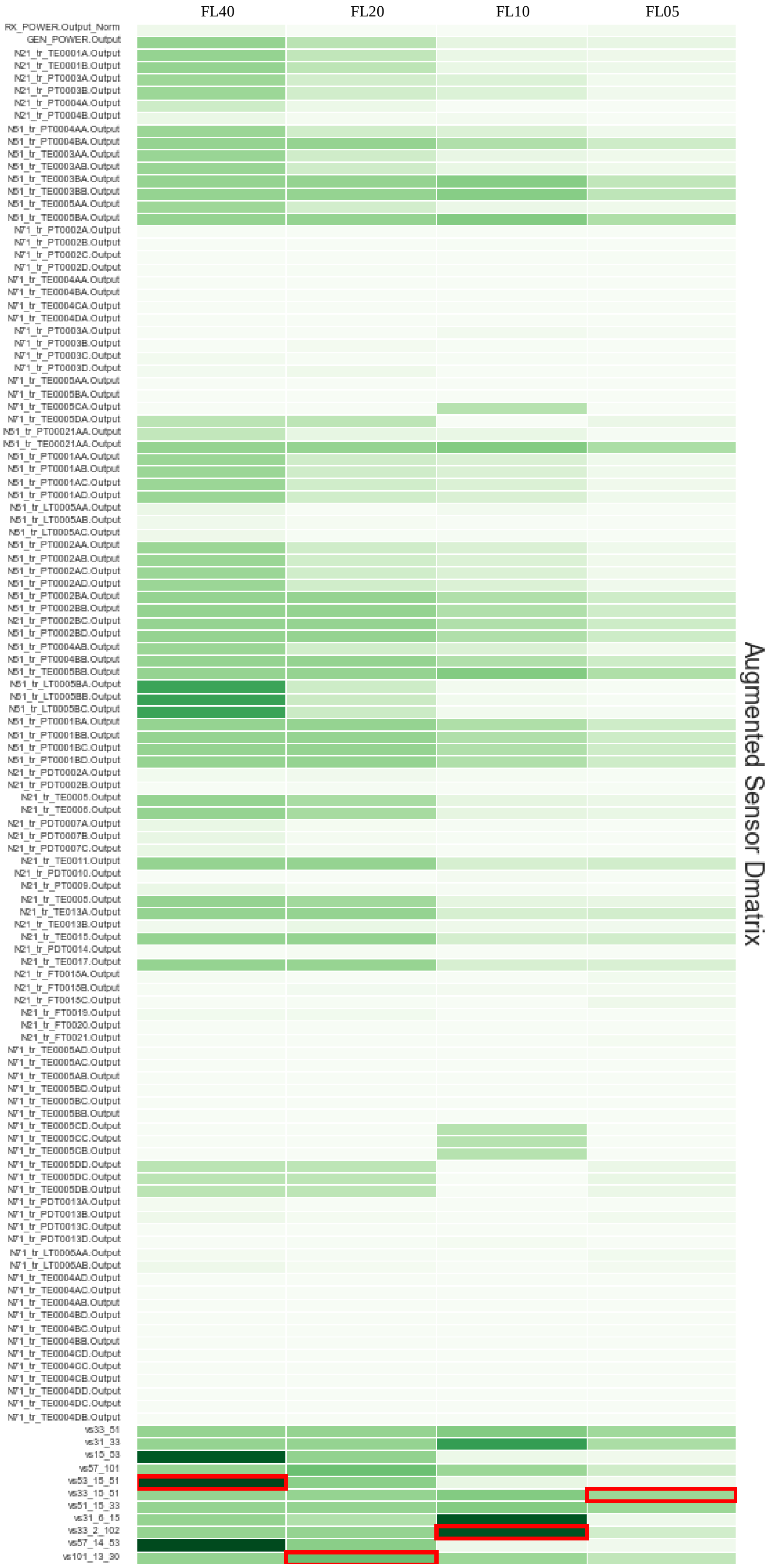
Helena Goldfarb is a Senior Scientist in the AI-ML-Robotics lab of the GE Global Research. Helena has been developing ML/AI-based solutions for various industrial systems with GE businesses and Lockheed Martin. She also has developing solutions sponsored by the government agencies, such as Joint Strike Fighter Propulsion program and DARPA's Measuring Biological Aptitude program. Helena has published over twenty papers and holds fifteen patents.

APPENDIX

Table 2. Examples for various trade studies

Trade Study	Sensor meta data	Constraints
Same type of sensors placed at different locations • Define two sensors (e.g., S26 and S27) • Create a sensor group (e.g., group 8) • Assign both sensors to this group • Create a constraint (e.g., C5) for group 8	“Sensor": [{ "ID": "S26", "Description": “pipe inlet pressure", "Type": "Pressure", "Mandatory": 0, "VirtualSensor": "NA", "SensorGroup": **8**, "FailureRate": 1e-6, "TotalCost": 0.02 }, { "ID": "S27", "Description": “pipe exit pressure", "Type": “Pressure", "Mandatory": 0, "VirtualSensor": "NA", "SensorGroup": **8**, "FailureRate": 1e-6, "TotalCost": 0.02] }]	"Constraint": [{ "ID": "C5", "Metric": "**SensorGroup**", "Type": **8**, **"Logic": “<=",** "Value": **1** }]
Mark a sensor as mandatory (required for controls /Certification) • Set sensor “Mandatory” flag	“Sensor": [{ "ID": "S1", "Description": “lake water temperature", "Type": "temperature", "Mandatory": **1**, "VirtualSensor": "NA", "SensorGroup": “NA”, "FailureRate": 1e-6, "TotalCost": 0.01 }]	NA
Selecting sensors from various groups involves choosing any desired quantity (e.g., including at least one pressure sensor from one group and a maximum of two temperature sensors from another group) • Define two sensor groups • Assign sensors to corresponding groups • Create sensor group constraints	(not listed for brevity)	"Constraint": [{ "ID": "C7", "Metric": "**SensorGroup**", "Type": **5**, "Logic": “>=", "Value": **1** }, { "ID": "C8", "Metric": "**SensorGroup**", "Type": **6**, **"Logic": “<=",** "Value": **2** }]
Introduce a virtual sensor • Define a sensor as virtual sensor • Specify the physical sensors involved • Only include additional costs (beyond each individual sensor costs)	“Sensor": [{ "ID": "S5", "Description": “VS based on S2, S3, S4", "Type": "**Virtual**", "Mandatory": 0, "VirtualSensor": **"[S2, S3, S4]",** "SensorGroup": “NA”, "FailureRate": “NA”, "TotalCost": 0.01 }]	NA

Figure 10. Color-coded D-Matrix with Virtual Sensors


FL40
FL20
FL10
FL05
RX_POWER.Output_Norm
GEN_POWER.Output
N21_tr_TE0001A.Output
N21_tr_TE0001B.Output
N21_tr_PT0003A.Output
N21_tr_PT0003B.Output
N21_tr_PT0004A.Output
N21_tr_PT0004B.Output
N51_tr_PT0004AA.Output
N51_tr_PT0004BA.Output
N51_tr_TE0003AA.Output
N51_tr_TE0003AB.Output
N51_tr_TE0003BA.Output
N51_tr_TE0003BB.Output
N51_tr_TE0005AA.Output
N51_tr_TE0005BA.Output
N71_tr_PT0002A.Output
N71_tr_PT0002B.Output
N71_tr_PT0002C.Output
N71_tr_PT0002D.Output
N71_tr_TE0004AA.Output
N71_tr_TE0004BA.Output
N71_tr_TE0004CA.Output
N71_tr_TE0004DA.Output
N71_tr_PT0003A.Output
N71_tr_PT0003B.Output
N71_tr_PT0003C.Output
N71_tr_PT0003D.Output
N71_tr_TE0005AA.Output
N71_tr_TE0005BA.Output
N71_tr_TE0005CA.Output
N71_tr_TE0005DA.Output
N51_tr_PT00021AA.Output
N51_tr_TE00021AA.Output
N51_tr_PT0001AA.Output
N51_tr_PT0001AB.Output
N51_tr_PT0001AC.Output
N51_tr_PT0001AD.Output
N51_tr_LT0005AA.Output
N51_tr_LT0005AB.Output
N51_tr_LT0005AC.Output
N51_tr_PT0002AA.Output
N51_tr_PT0002AB.Output
N51_tr_PT0002AC.Output
N51_tr_PT0002AD.Output
N51_tr_PT0002BA.Output
N51_tr_PT0002BB.Output
N21_tr_PT0002BC.Output
N51_tr_PT0002BD.Output
N51_tr_PT0004AB.Output
N51_tr_PT0004BB.Output
N51_tr_TE0005BB.Output
N51_tr_LT0005BA.Output
N51_tr_LT0005BB.Output
N51_tr_LT0005BC.Output
N51_tr_PT0001BA.Output
N51_tr_PT0001BB.Output
N51_tr_PT0001BC.Output
N51_tr_PT0001BD.Output
N21_tr_PDT0002A.Output
N21_tr_PDT0002B.Output
N21_tr_TE0005.Output
N21_tr_TE0006.Output
N21_tr_PDT0007A.Output
N21_tr_PDT0007B.Output
N21_tr_PDT0007C.Output
N21_tr_TE0011.Output
N21_tr_PDT0010.Output
N21_tr_PT0009.Output
N21_tr_TE0005.Output
N21_tr_TE013A.Output
N21_tr_TE0013B.Output
N21_tr_TE0015.Output
N21_tr_PDT0014.Output
N21_tr_TE0017.Output
N21_tr_FT0015A.Output
N21_tr_FT0015B.Output
N21_tr_FT0015C.Output
N21_tr_FT0019.Output
N21_tr_FT0020.Output
N21_tr_FT0021.Output
N71_tr_TE0005AD.Output
N71_tr_TE0005AC.Output
N71_tr_TE0005AB.Output
N71_tr_TE0005BD.Output
N71_tr_TE0005BC.Output
N71_tr_TE0005BB.Output
N71_tr_TE0005CD.Output
N71_tr_TE0005CC.Output
N71_tr_TE0005CB.Output
N71_tr_TE0005DD.Output
N71_tr_TE0005DC.Output
N71_tr_TE0005DB.Output
N71_tr_PDT0013A.Output
N71_tr_PDT0013B.Output
N71_tr_PDT0013C.Output
N71_tr_PDT0013D.Output
N71_tr_LT0006AA.Output
N71_tr_LT0006AB.Output
N71_tr_TE0004AD.Output
N71_tr_TE0004AC.Output
N71_tr_TE0004AB.Output
N71_tr_TE0004BD.Output
N71_tr_TE0004BC.Output
N71_tr_TE0004BB.Output
N71_tr_TE0004CD.Output
N71_tr_TE0004CC.Output
N71_tr_TE0004CB.Output
N71_tr_TE0004DD.Output
N71_tr_TE0004DC.Output
N71_tr_TE0004DB.Output
vs33_51
vs31_33
vs15_53
vs57_101
vs53_15_51
vs33_15_51
vs51_15_33
vs31_6_15
vs33_2_102
vs57_14_53
vs101_13_30
Augmented Sensor Dmatrix