*Review of Business and Technology Research, Vol. 13, No. 1, 2016, ISSN 1941-9414*

# Software Simulator for Intelligent Health Monitoring System of Windfarm

Aaron R. Rababaah*[1], Joseph Arumala[2], Ibibia K. Dabipi[3], Kenny Fotouhi[4], Gurdeep Hura[5]
[1]Department of Computer Sciences and Information Systems, American University of Kuwait, Salmiya, Kuwait
[2],[4]Department of Technology, University of Maryland Eastern Shore, Maryland, USA
[3]Department of Engineering and Aviation Sciences, Maryland, USA
[5]Department of Mathematics and Computer Sciences, University of Maryland Eastern Shore, Maryland, USA
*[1]arababaah@auk.edu.kw; 2joarumala@umes.edu; 3ikdabipi@umes.edu; 4mfotouhi@umes.edu;
5gshura@umes.edu;

## Abstract

Clean renewable energy is one of the current and future top technologies that the world is seeking. One of the major technologies in this domain is offshore wind energy. One major concern in offshore wind farms is effective and efficient maintenance systems. In an earlier research work the authors of this work have proposed an intelligent method based on neural networks and digital signal processing to develop intelligent mechanical system fault detection. In our previous work, it was shown that the proposed approach was effective and accurate in detecting signal faults as our extensive experimental work demonstrated. The work of this paper will build upon our previous work to extend the application of the system from one turbine into a farm consisting of up to fifty turbines windfarm. To demonstrate how the proposed system would work, we will present the concept using a simulation software.

## 1. Introduction and Related Word

There are number of attempts that tried to use simulation approach to test proposed solutions in the domain of intelligent health monitoring for wind turbines. In this section we will present some of these attempts. Dong-Wan et al. [1] used simulated damaged blades defects to test their structural health monitoring algorithm. Their simulations considered healthy and unhealthy energy harvester. Different causes can be simulated by stress and strain conditions. Crowther [2] studied load measurement and fatigue using blade strain sensors and real time sensing/processing which focused on gears and bearings, continuous updated life prediction and probability of failure oil, grease and filter analysis (lubrication monitoring). Simulation was used to estimate the resonance frequency of gears). Sample data for one week was visualized, processed and presented. Simulated data was also used to demonstrate the significance of pitch-error in gear fatigue. E Di Lorenzo et al. [3] used the aeroelastic code which allows simulating the damages in turbine blades in several ways: by reducing the edgewise/flapwise blades stiffness, by adding lumped masses or considering a progressive mass addiction (i.e. ice on the blades). Korkua [04] used Matlab and Simulink environments to simulate some mechanical components of the turbine system. Schulz and M.J. Sundaresan [05] aimed to model and simulate a network of sensors embedded in a turbine structure to continuously collect data, analyze it and detect failure events.

From our literature survey we did not find any similar work to the work of this paper targeting signal simulation without using the physical system itself. This paper presents a development of a Matlab-based simulation software interface that allows the user to simulate the digital signals collected from up to fifty turbines windfarm with possible damaged and healthy data. The next section will elaborate on the details of the simulator.

## 2. The Simulation software

The simulation software, wt_sim, hereafter, is developed under MatLab development environment. In this section we will present it in details:

**Directory organization:**

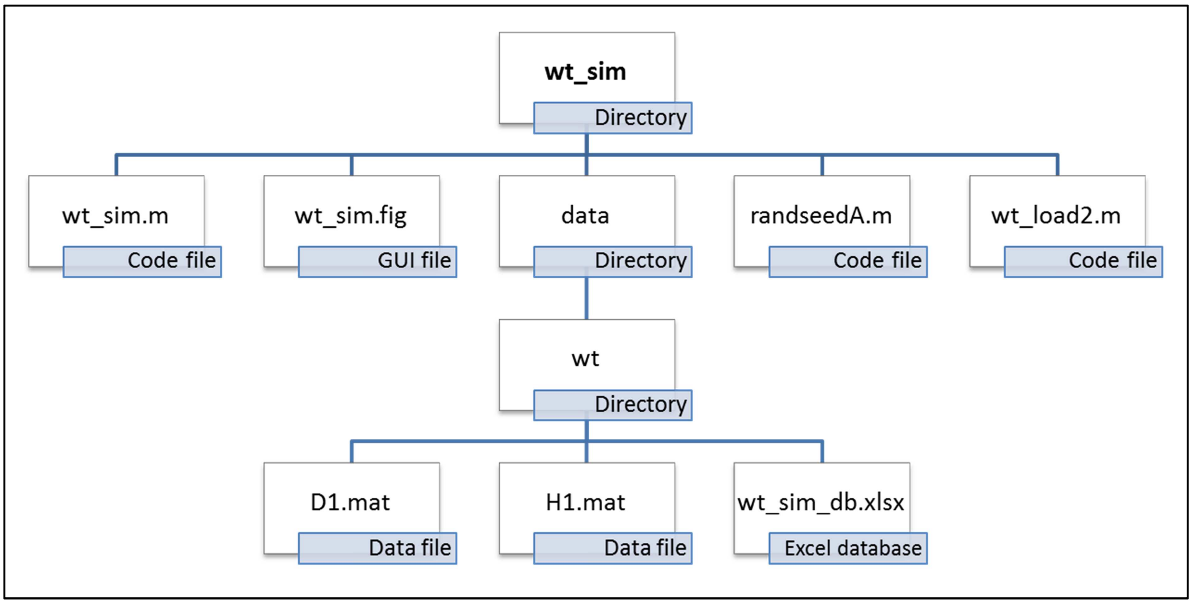


**Figure 1**: Directory organization of the wt_sim software

**Description of the Directory organization**:

- ***wt_sim***: the main directory of the package which includes all code and data.
- ***wt_sim.m***: the source code of the simulator.
- ***wt_sim.fig***: the graphical user interface file that accompanies "wt_sim.m".
- ***data****: a sub* directory that contains the data and database of the simulator.
- ***randseedA.m***: source code for a needed routine used by wt_sim.m to generate random numbers.
- ***wt_load2.m***: source code for a needed routine used by wt_sim.m to load the data files.
- ***wt***: a sub directory contains the input data files and the output database.
- ***D1.mat****:* input data set of damaged sensor data samples.
- ***H1.mat***: input data set of health sensor data samples.
- ***wt_sim_db.xlsx:*** output database excel file.

**Running the simulator**

- Copy the main directory "wt_sim" to the "work" directory of MatLab.
- Change directory to "wt_sim"
- Type the command "wt_sim" in the command prompt window of Matlab
- The GUI interface of the simulator should be active now and start configuration and running the simulator.

### Interface Configuration

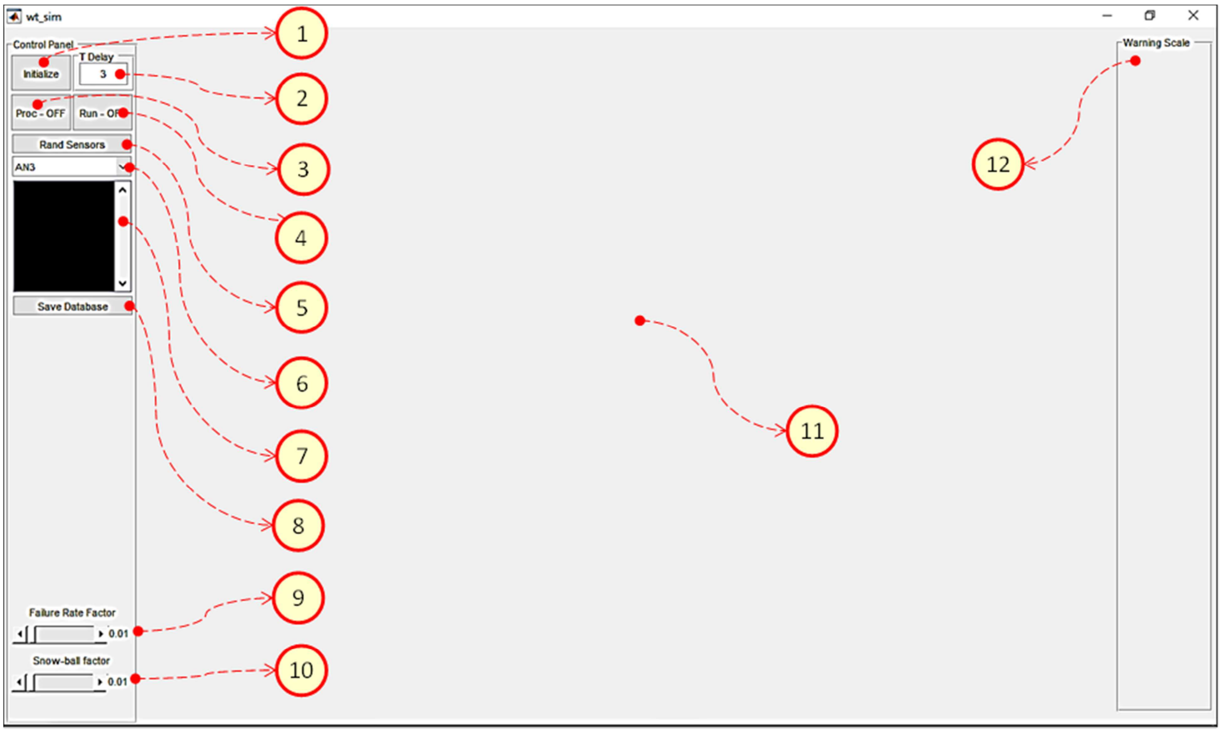


**Figure 2**: GUI Interface of the wt_sim software

1. **Initialize**: this button must be the first control to be activated which initializes the simulator by Loading input data files and by creating axes matrix for wind farm sensory data in the GUI component number (11) in Figure 2.

2. **T Delay**: this is a text (editable) box represents time interval between consecutive sensor readings. The time unit is (seconds) and by default and recommended to be (3 sec) so the simulator will have sufficient time to process data between data sensor sampling.

3. **Proc-OFF/ON:** this button enables/disables data processing to analyze sensor signals and simulate fault detection. It toggles between Proc-OFF and Proc-ON labels.

4. **Run-OFF/ON:** this button enables/disables running the simulator process. It toggles between Run-OFF and Run-ON labels.

5. **Rand Sensors:** this button randomly switches the sensor signal source to be sampled and then processed accordingly. The different signal sources are defined in Figure 3 below.

6. **Drop-Down List:** this control allows the user to switch all sensor signal sources to a one unified source. Sources in Figure 3 are used her as well.

7. **Text port**: this component is to display the reported faults of any wind turbine. A message of turbine number, time of the event and severity of the fault is displayed each time a fault is detected.

8. **Save Database**: this button activates the saving process of the reported fault information to the database file described earlier in section 1 (Directory organization).

9. **Fault Rate Factor**: this slider sets the probability of a fault occurring in a wind turbine mechanical system. The range of values is [0, 1].

| Sensor Label/Signal Name (see Figure 5) | Description | Sensor Model | Units in Data File |
|---|---|---|---|
| AN3 | Ring gear radial 6 o'clock | IMI 626B02 | $m/s^2$ |
| AN4 | Ring gear radial 12 o'clock | IMI 626B02 | $m/s^2$ |
| AN5 | LS-SH radial | IMI 622B01 | $m/s^2$ |
| AN6 | IMS-SH radial | IMI 622B01 | $m/s^2$ |
| AN7 | HS-SH radial | IMI 622B01 | $m/s^2$ |
| AN8 | HS-SH upwind bearing radial | IMI 622B01 | $m/s^2$ |
| AN9 | HS-SH downwind bearing radial | IMI 622B01 | $m/s^2$ |
| AN10 | Carrier downwind radial | IMI 626B02 | $m/s^2$ |
| Speed* | HS-SH | | rpm |

**Figure 3**: Sensor signal sources in the wind turbine system

10. **Snowball Factor**: this slides sets the factor by which next fault rate can be increased, after a fault has been detected. This simulates the snowball effect in machinery failures. When a machine fails, the likelihood of it failing will increase.

11. **Axes Panel**: this area contains a matrix of axes, each axes represents one turbine. Based on the switched source of sensor data, this axes displays the sampled data segment along with its simulated faults and their severity when detected.

12. **Warning Scale**: this is a color-coded scale according to the fault severity of the turbines. The range is [0,100] where each level represents a severity degree of the fault. Colors range from blue, green, yellow, orange, and red with many shades in between to cover all the [0,100] range. Blue represents the lowest whereas dark red represents the highest severity.

13. **Database**: the system maintains a database log file of all events detected. The information of these events is captured in these fields: event# - which is a sequence number indicating the relative order of the event, Time: an index proportional to the time of the sampled signal, Turbine: the turbine id, Sensor: the sensor name that an event was detected in it is data, Location: the location of the sensor on the turbine structure, Severity: the intensity of the accumulated events of the given turbine. Figure 4 illustrates this database file.

| Event No. | Time | Turbine | Sensor | Location | Severity |
|---|---|---|---|---|---|
| 32 | 1361377 | 27 | AN6 | IMS-SH radial | 12 |
| 33 | 1045856 | 42 | AN8 | HS-SH upwind bearing radial | 8 |
| 34 | 1019004 | 22 | AN5 | LS-SH radial | 2 |
| 35 | 2290484 | 27 | AN6 | IMS-SH radial | 13 |
| 36 | 1199395 | 31 | AN3 | Ring gear 6 o'clock | 0 |
| 37 | 1277508 | 42 | AN8 | HS-SH upwind bearing radial | 9 |
| 38 | 1430413 | 22 | AN5 | LS-SH radial | 3 |
| 39 | 544936 | 27 | AN6 | IMS-SH radial | 14 |
| 40 | 1930589 | 42 | AN8 | HS-SH upwind bearing radial | 10 |
| 41 | 963414 | 22 | AN5 | LS-SH radial | 4 |
| 42 | 1115865 | 27 | AN6 | IMS-SH radial | 15 |
| 43 | 1438299 | 42 | AN8 | HS-SH upwind bearing radial | 11 |
| 44 | 1535408 | 49 | AN3 | Ring gear 6 o'clock | 0 |
| 45 | 48245 | 22 | AN5 | LS-SH radial | 5 |
| 46 | 1629737 | 27 | AN6 | IMS-SH radial | 16 |
| 47 | 1927991 | 42 | AN8 | HS-SH upwind bearing radial | 12 |
| 48 | 610399 | 10 | AN9 | HS-SH downwind bearing radial | 0 |

Figure 4: Database file

14. **A final look**: a snap-shot is given in Figure 5 showing a screen shot of the simulator in progress.

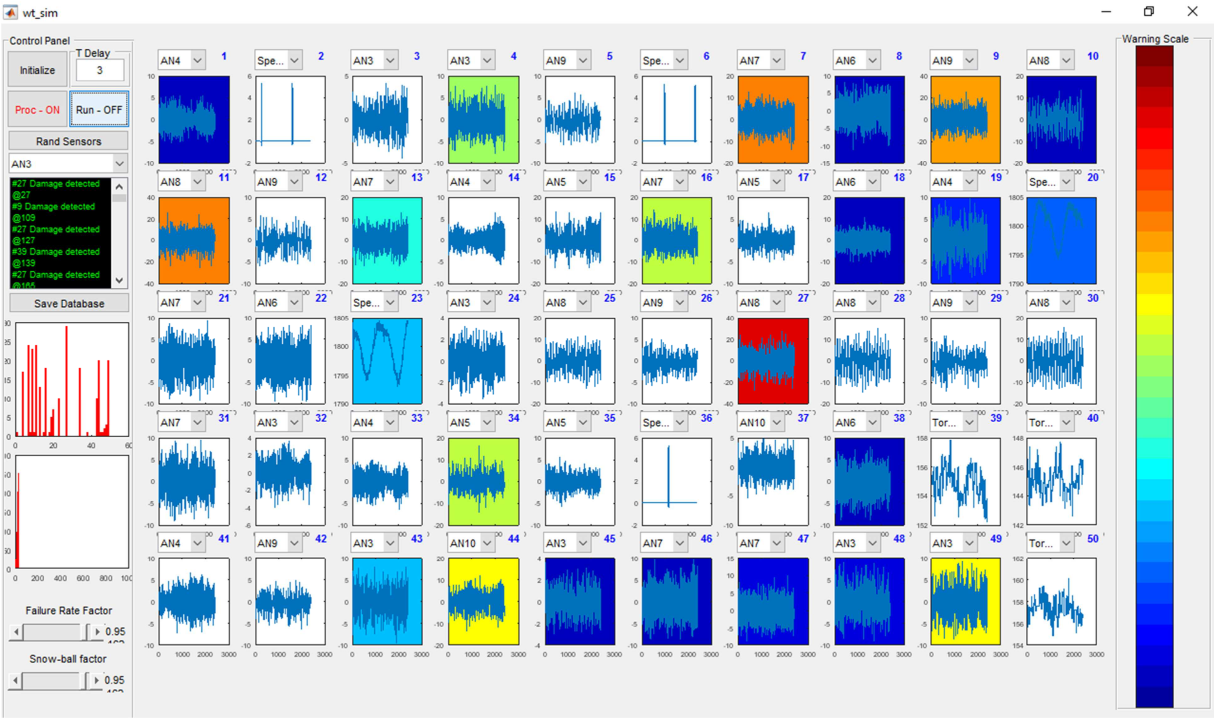


Figure 5: A screen shot of the simulator in progress.

## 3. Capabilities of the software simulator (wt_sim):

The developed intelligent system is built based on the two theories of machine learning and digital signal processing. The main goal of the system is to subject the collected sensory data from a windfarm to a multi-stage process of transformations from raw data into detected classified events upon which informed decisions can be made for more effective and efficient maintenance. The capabilities of wt_sim are presented in this section.

- Input: when modified, wt_sim can receive the input raw data in different formats including: matrix files (.mat), spreadsheet (.xls) and asci (.txt) file formats. Currently wt_sim assumes that the input data is stored as a matrix (.mat) file. wt_sim can be configured to set the size of signal sample to be examined for potential faults.

- Multi-stream data: wt_is capable of processing signal feeds from a wind farm consisting of up to fifty turbines. Each turbine can have multiple sensors as well. The system allows the user to switch from one sensor to another for each turbine. Therefore, turbines are not necessarily are being sampled from the same senor although it is possible to uniformly do that across the windfarm.

- Real time processing: it is crucial for an online processing system to have efficient process stages to digest data in real time. Fast Fourier Transform (FFT) is chosen for its well established reputation for efficiency and suitability for the problem domain of this research.

- Machine Learning: artificial neural networks (ANN) are used to train a model that is capable of detecting the signal faults with noisy data. This solution is generalize its decisions on unseen data samples for future detections.

- Adaptation to new data: once the ANN is trained, it is kept the same as long as the source of the data is pulled from the same system and It is expected to work effectively. In case new training data is available in future, wt_sim is able to be trained to adapt to the new data content.
- Focus of attention: wt_sim keeps track of the failure severity of each turbine and map it to a color-coded map. This allows the users to focus their attention and helps them to manage their maintenance priorities.

## 4. Reducing the Cost of Operational Maintenance:

Traditional machinery maintenance depends on corrective to fix an unexpected failure that has been detected and preventive maintenance in which periodic measures are performed to avoid future failures. On the other hand, the proposed system follows intelligent maintenance approach which depends on frequently collecting data using an array of sensors, analyze this data, evaluate and understand the current situation of the system and take informed decision adaptively to the results. The following are advantages of intelligent maintenance to reduce cost.

- Early detection: The proposed intelligent system prevents catastrophic system failures by real time detection of early signs of these failures and act swiftly to eliminate their major risks and costs.
- System adaptive training: the intelligent ANN failure classifier is flexible. It can be updated by retraining it using most recent collected data and using the new model as the system ages and requires new classification model. This feature is important to make the system more robust by making it up-to-date with the state of the monitored system. This robustness ensures higher detection rate which reduces the risk of major failures which in turns reduces operational and maintenance cost.
- Maintenance schedules: the proposed system helps to make maintenance schedules smarter by basing maintenance jobs upon thresholds of the analyzed data. Meaning, maintenance jobs should be carried out according to the data collected from individual turbines. Each turbine has its health record and state so upon the discovery of a certain risk sign in a turbine, then maintenance can be scheduled. This approach will reduce cost since we carryout maintenance dynamically according to collected data not based on predetermined times which cuts down on unnecessary work.
- The right work at the right time: typical preventive maintenance occurs at a very early time when no maintenance measures are required. This can be counterproductive and costly to the system since every time the machine is maintained, its reliability decreases. Therefore, maintenance should only be done at the right time.

## 5. Conclusions

We have presented a software development for a simulator for wind energy farm that supports up to fifty wind turbines. This effort was a follow-up work on a previous work that investigated intelligent digital signal processing techniques to design health monitoring system for wind turbines. The previous work was a stage in the project that required a follow-up final stage to design and test simulation software that would expand on the techniques studied and apply them on a larger scale. The simulator "wt_sim" was developed and tested and found to be valuable in making it easy for the user to configure and test different scenarios of the windfarm.

## References

[1] Dong-Won Lim et al., Wireless Structural Health Monitoring of Wind Turbine Blades Using an Energy Harvester as a Sensor, American Institute of Aeronautics and Astronautics, 2011.

[2] Crowther, A., "Using Prognostics and Diagnostics for Wind Farm Health Monitoring," US Wind Technical Center, Boulder, CO, 2012.

[03] E Di Lorenzo et al., Structural Health Monitoring challenges on the 10-MW offshore wind turbine model, Journal of Physics: Conference Series, Volume 628, conference 1, 2015.

[04] Korkua, Suratsavadee Koonlaboon, Wireless Health Monitoring And Improvement System For Wind Turbines, University of Texas, Arlington, 2011.

[05] M.J. Schulz and M.J. Sundaresan, Smart Sensor System for Structural Condition Monitoring of Wind Turbines, NREL/SR-500-40089, August 2006.

[06] Aaron R. Rababaah et al., Mechanical System Fault Detection using Intelligent Digital Signal Processing, Journal of Machinery Manufacturing and Automation 2016, Vol. 5 Iss. 1, PP. 27-39.