

Help file to set the batch_Kriging code

Software needed: Matlab, R, FAST (NREL)

Summary:

Different codes are enclosed in the Thesis_Code folder. These produce the active learning with the Kriging model for fatigue analysis.

To run the code, Matlab, R and FAST are demanded. Matlab acts as a glue code, R was set initially due to the interest of working with a noise component in the Kriging model, and FAST solves the coupled response for offshore wind turbines. It is recommended that further implementations of the code should substitute the application of R, and instead use, ooDACE (which can be found for Matlab), without loss of accuracy and stability. And hence, decreasing the complexity of the code implementation.

The code can be analysed as a black-box in its current state. However, only to analyse fatigue on the tower component, and with mean wind speed and turbulence intensity as the main variables in the design of experiments. The code can be easily modified to enclose other loading variables, and it is already prepared to enclose wave parameters and directional parameters. Further changes recommend a detailed analysis of the FAST .dat files that set the turbine model and the environmental models (Thesis document explain the importance of each in the introductory chapters).

As the code was developed modular, the paths need to be corrected. Most of the code uses the “=pwd” command to get the current directory, however, the R script and the script that call R need to have their paths fully updated (these are in the DICE folder).

The following instructions highlight major considerations of the modular code developed.

Batch_Kriging

This folder includes all the codes that enable the application of an adaptive search that uses the Kriging as an interpolator for fatigue design. Figure 1 presents an overview of the different procedures involved in the fatigue design for Offshore Wind Turbines using the Kriging to interpolate the mean.

The algorithm considers three main regions, the adaptive search, the DICE prediction and the FAST cloud runner. The last, and most complex of the three, encloses multiple scripts that analyse the loads on a wind turbine, decomposes the load ranges and calculates the stress-cycle fatigue. This code is analysed within a larger cycle that predicts fatigue damage using a meta-model (the Batch_Kriging). The main file of it is the AKroot.m file that calls R and FAST (same directory). It also responsible for the active learning enrichment. The DICE folder encloses all the information related to R and the scripts to run it. These are called from the root file.

The selected point is then evaluated in the simulate_new_point folder, that is set to evaluate the operational point and enrich the meta-model with the true response.

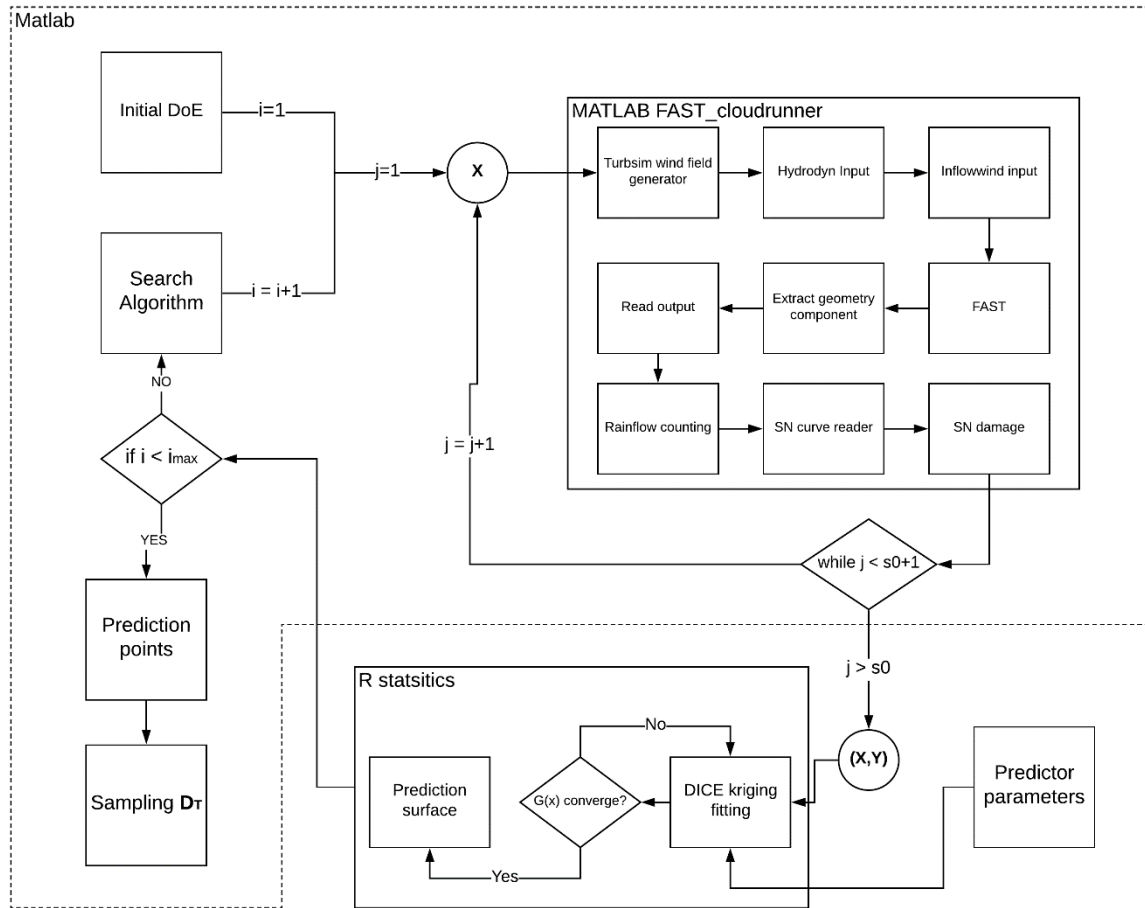


Figure 1 – Batch Kriging analysis procedure.

Main files:

- Main control file : AK_root.m

This file controls the search procedure.

It loads the initial DoE. And sets the initial conditions to fit the Kriging model (which is fit in R). New implementations should consider changing this to be fit in the ooDACE (in case the noise is not of interest), eliminating the layer of complexity of R. ooDACE was identified later to provide similar levels of accuracy. Moreover, it allows to avoid the usage of xls files which is the implemented currently in DICE (to transfer data between Matlab and R).

To remain applying the DICE, it is also of interest in future implementations to change the input files from XLS to CSV, which are much easier to work with and do not originate a loop of recovery from the Windows (Windows tends to keep memory on the XLS files when these are forced to close).

Note: The paths (user_path, directory_drive) need to be corrected upon utilisation. The first is set for the computer user path which changes with user (to pick the R software). The second is used to set the directory where some modular files are loaded (such as the R file).

- Search : AKinterpolation_search.m

This .m file uses the predictions to search and select the new point to enrich the DoE. In the present example it uses the biased randomization of the design space. In order to remove it, the file needs to be edited (comment the utilisation of the joint PDF loaded from the .mat file).

- Enrichment : expand_DoE_sim.m + Simulate_iterated_point.m

This file performs the runs of the OWT simulation model. The first calls the simulator.

The simulator receives parallel computing. So the number of simulators is an Input. In the case shared 1 simulator is used. To create more simulators just copy **Simulator_1** and change it to Simulator_i (e.g. Simulator_2). The file needs to be reviewed to accommodate other simulator. It used 10 simulators in its previous form (deleted due to the large size of the simulator folders), and these are commented in the current file. To use more simulators just uncomment the code, or add similar correcting for the simulator number.

The codes uses the PARALLEL PROCESSING TOOLBOX, which is demanded to run the “batch” command.

The second part of the .m file is dedicated to treat the simulation runs that are saved in the folder All_simulations_results.

Simulator_1 Folder

This folder encloses the model of OWT turbine and its more important file is the cloud_runner (FAST_cloud_runner in 00_Matlab_Cloud folder), the most extensive and important code in the analysis. This code, as indicated in the figure is responsible for performing the whole analysis of the wind turbine model in terms of fatigue. In case new simulators are created the Number_SIM input needs to be changed to mirror the number of the simulator considered.

This version of the FAST_cloud_runner calculates the loads in the tower in 9 points.

The FAST_cloud_runner is divided in different parts

- Part 01 Turbulence Model – TURBSIM – creates the wind field in Turbsim_Wind_Generator
- Part 02 Hydrodynamic input- HydroDyn – sets the wind field parameters
- Part 03 Inflow_wind_input_generator – Changes the input conditions of the inflow input (mainly direction in the current case)
- Part 04 – FASTx64 – Evaluation of the OWT system
- Part 05 – Postprocessing

All these parts of the code can be edited for further complexity (e.g. evaluation of other components). A description of the input.out file is included. The list of outputs can be consulted in a xls file. New outputs can be added in the .dat file (e.g. ElastoDyn.dat)

The code in its current form is able to get the loads in different tower sections. It is noted that changes in the inputs demands also the analysis of Turbine model. All the control files for the turbine model are easily accessed in the Simulator_1 folder (.dat) files. Analysis of the thesis and NREL’s manuals

allows a better understanding of these. Each .dat file is respective to a modular analysis of the code (e.g. Subdyn is the Substructure dynamics). It is recommended a comprehensive analysis of how wind turbines are simulated in order to understand how to edit these files, and the physical meaning of the different inputs.

Attention to the units of the environmental variables (e.g. it is common to calculate turbulence in decimals but input needs to be in %). Additionally, the original code was set on a double slope DNV SN curve, in order to use a single SN slope, the single m SN slope needs to be uncommented in the runner.

All the saved files are copied to a central directory for post-processing, the post-processing is performed in the simulator and then the results of multiple iteration are copied to an additional directory to be saved.

DICE folder

InitializationRcode.m needs to have the paths updated for the computer in-hand. It demands the location of the .R script and the location of installation of R.

The .R files also need to have the directory updated. While in Matlab it is straightforward to get the self-path in order to correct the paths, in R the same does not occur, and any modifications demands for the paths to be corrected manually.

Note:

The author is not responsible for any incorrect use or misuse of the present codes. Any questions please contact teixeirr@tcd.ie.