A. JAMES CLARK
SCHOOL OF ENGINEERING
Center for Risk and Reliability

SHM2Opt

Instruction Manual

By

Amin Aria

Advisors:

Dr. S. Azarm

Dr. M. Modarres

Fall 2018

This is an instruction manual for SHM2Opt package. The output of this package is a two dimensional layout, including configuration of sensor of different types along the exterior surface of a pipeline in addition to a human inspection scheme. To obtain results using this package, please follow the steps of illustrative example described below.

1- Read **SHM2Opt paper.pdf** to gain an insight on the entire package.

NOTE: It is recommended to skim the **RESS Journal Paper, AA.pdf** as well to obtain a deeper insight on the algorithms utilized in SHM2Opt.

2- 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 exceedes 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#####################
```

3- 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.

```
wilks.txt - Notepad
File  Edit  Format  View  Help
"xexist2"  "yexist2"  "xexist2"  "ynode"  "c#1"  "c#2"  "c#3"  "c#4"  "cannot-link"
"1"  1.3680325371773  6.28278287778247  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
```

4- 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####
```

5- 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).

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

#radius of pipeline
r=1;
#number of damages
n=length(x)
#number of classes
ns=4
#number of sensors types
nsensor=2

#####Minimum Spanning Tree library : used for clustering damages of each sample
library(optrees)

##Distance between damages and nodes
dist=matrix(0,nrow=length(x),ncol=length(y))

##Probability of missing a defect
POND=matrix(0,nrow=length(x),ncol=length(y))

#Damage Missing relation: exp(deltax/a +b ), boundry conditions: exp(deltax=0)=0.1, exp(deltax=20 meters)= 1 >>>
#!!!###NOTE: you can change this relation considering the geometry of pipeline and pattern recognition results
  ##of the pipeline.

a1=c(8.3,8,8,7)
b1=c(-2.7,-3,-3.5,-2.7)
LPONDint=matrix(,nrow=0,ncol=length(x))

###parameters of PoD(size) for Acoustic Emission
#(Rabiei and Modarres, 2013) when PoD(depth=1.5mm)=1
aAE=3.07


###parameters of PoD(size) for HI with Ultrasonic tool
```

6- 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	0	
5	2.992525	3.057143	3.493694	3.120563	2.866719	1.478713839	0.217172	0	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	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												

7- 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 damges 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\

8- 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

9- 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****************
*********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 non-sense results!

10- 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 dete
                                                              on (hit-miss model
                                                              )
            VARIABLE aveut.L                  =        0.494  average utility pe
                                                              sed node
            VARIABLE nPOD.L                   =        8.391  avelog(probability
                                                              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 clo
                           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 $e^{-8.391}$. For more information on the outputs, please refer to the **RESS Journal Paper, AA.pdf**.