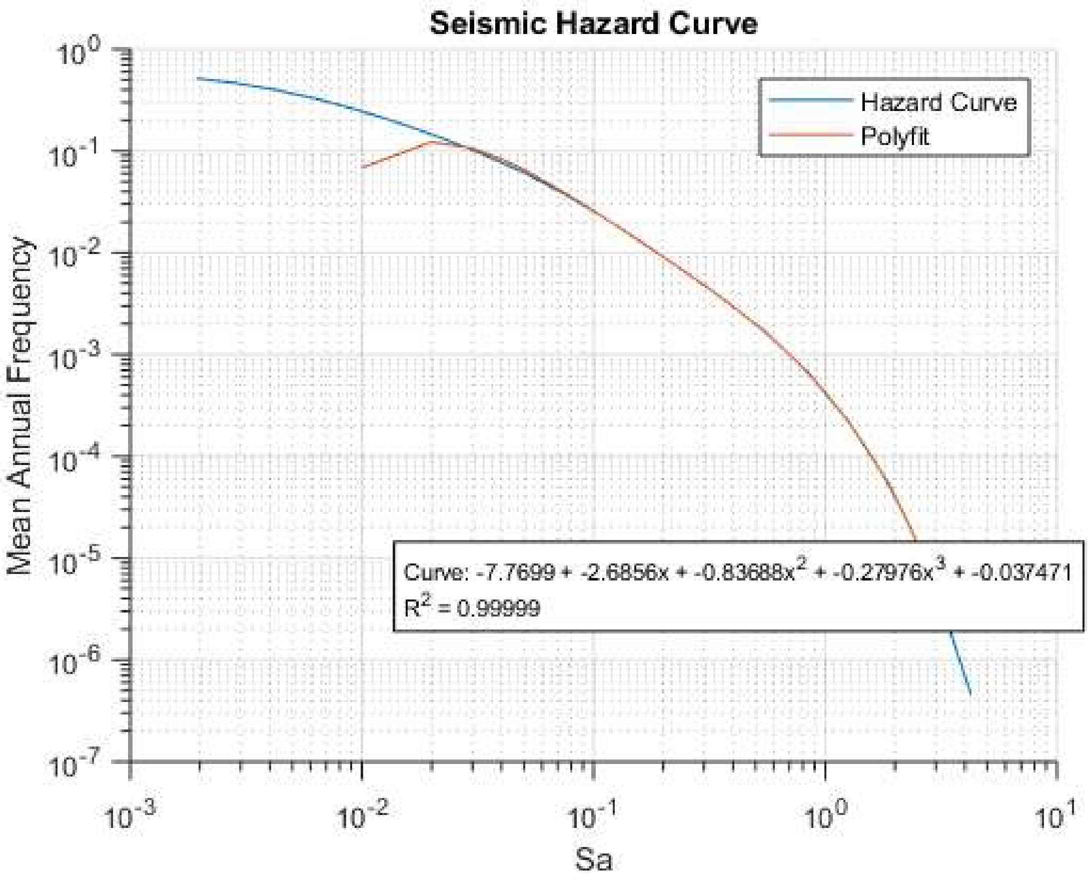
Contents

```
close all; clear all;
```

Load Hazard Curve and Fit a Polynomial in Log-Log Space

```
load("HazardCurve.txt")

order = '4';
curve = HazardCurve;
interval = [0.01,3];
dSa = 0.01;
[handles] = createPolyFit(order, curve, interval, dSa);
```



Initialize Variables for Integration

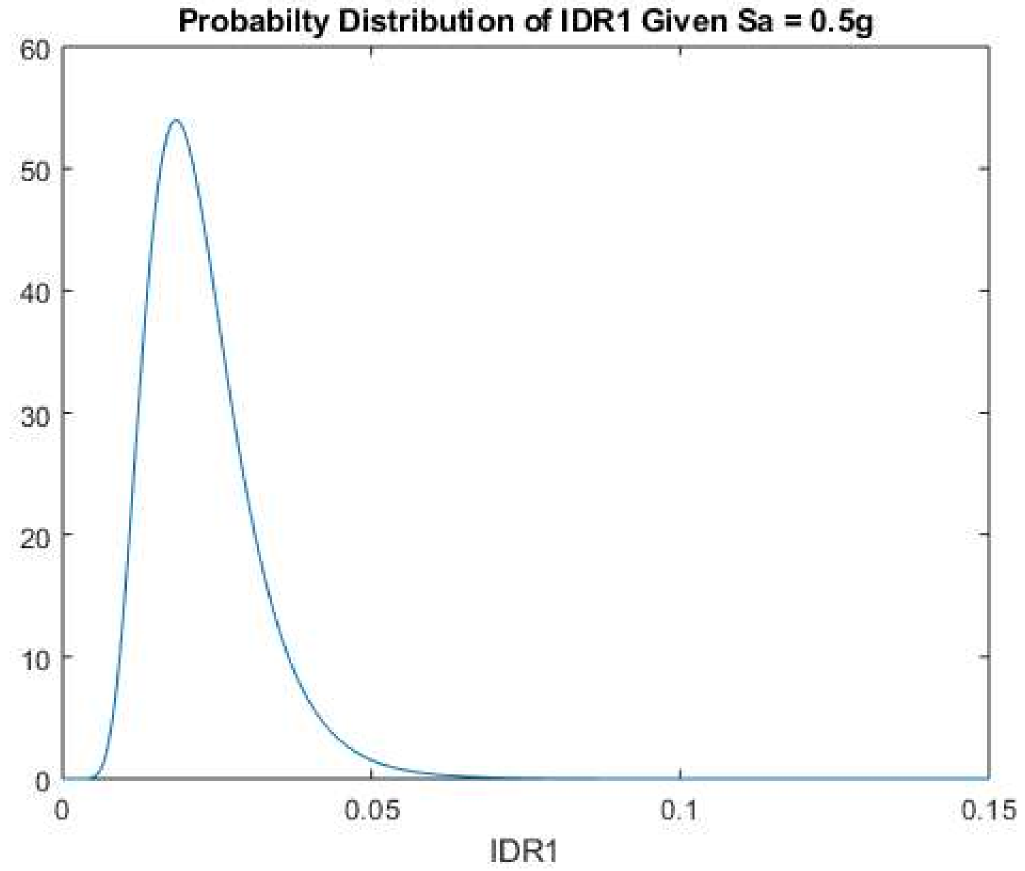
Should we do anything about fact that if the range includes zero we get a Nan?

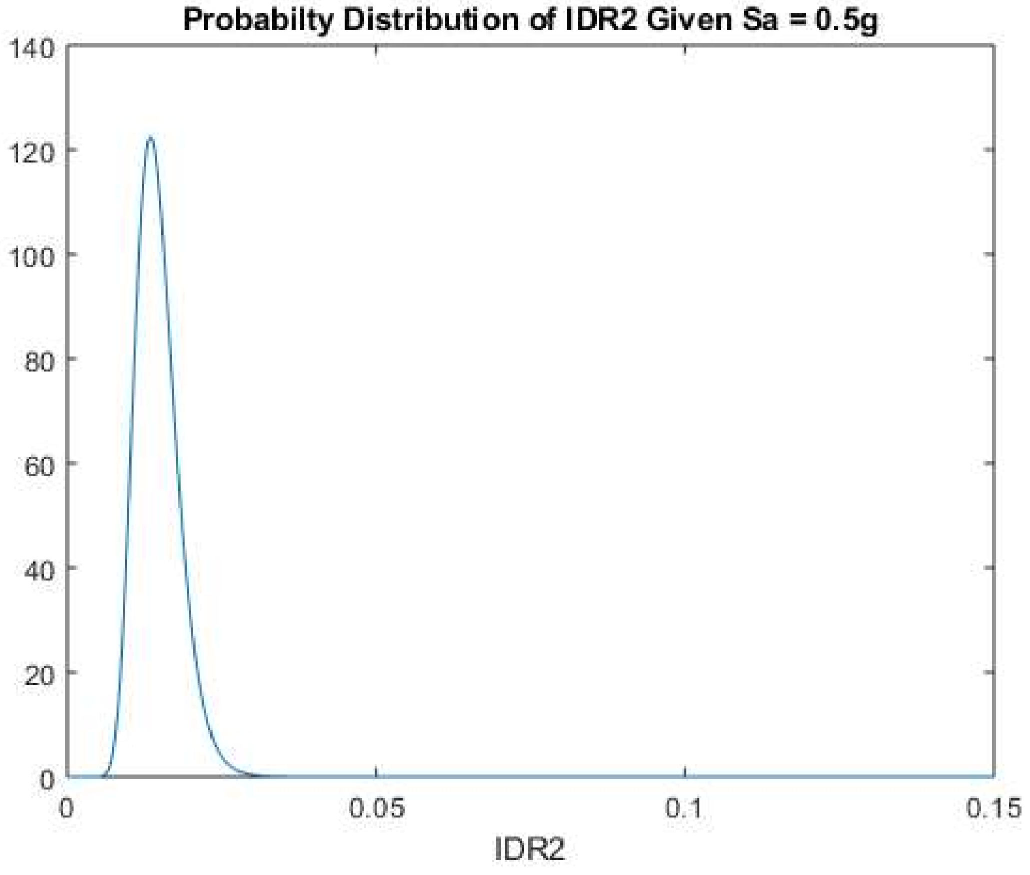
```
handles.EDP.IDR = 0.0001:.0001:.15;
handles.EDP.PFA = 0.01:.01:4.0;
handles.EDP.RIDR = 0.0001:.0001:.1;
```

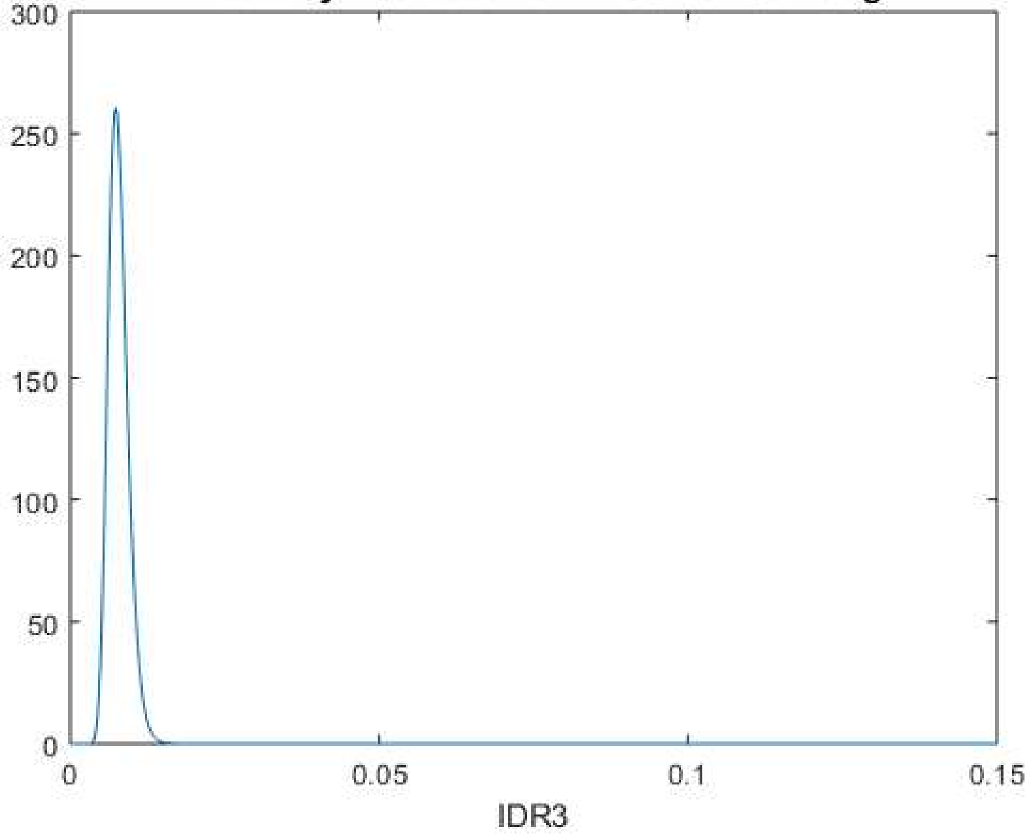
Input and Plot Stripe Analysis Results

```
%GUI Inputs
ns = 4; %Number of Stripes
nedp = 3; %Number of EDPs
n = 30; %Number of Groundmotions

%Load Stripes Values from .csv - offset data by one to only get numeric
%data
%Maybe we could structure this similar to how we structured the fragility
%and loss functions
filenames=["Stripe1_Sa0.10_1col_S.csv","Stripe2_Sa0.35_1col_S.csv",...
    "Stripe3_Sa0.70_1col_S.csv","Stripe4_Sa1.05_1col_S.csv"];

handles.stripes = [0.1,0.35,0.70,1.05];%Not sure how these are input into GUI
handles = LoadStripeData(handles, filenames,ns);
handles = ResponseEstimation(handles);
```

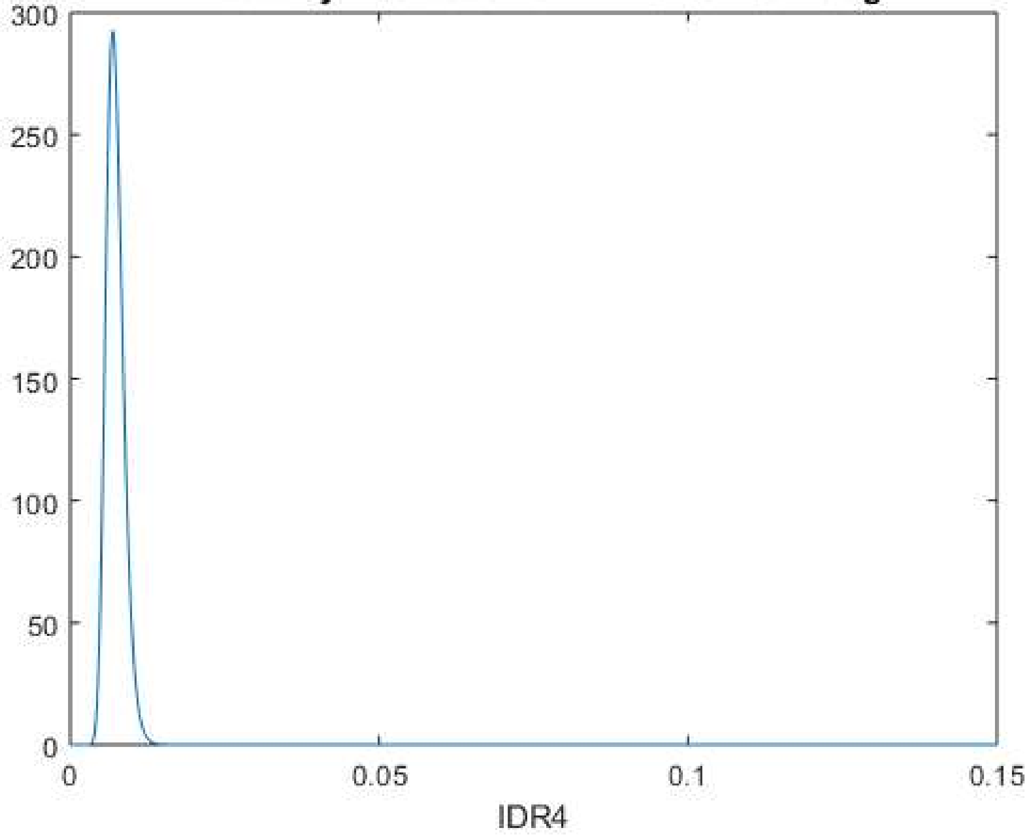
```
Stripe1_Sa0.10_1col_S.csv
Warning: Variable names were modified to make them valid MATLAB identifiers. The
original names are saved in the VariableDescriptions property.
Stripe2_Sa0.35_1col_S.csv
Warning: Variable names were modified to make them valid MATLAB identifiers. The
original names are saved in the VariableDescriptions property.
Stripe3_Sa0.70_1col_S.csv
Warning: Variable names were modified to make them valid MATLAB identifiers. The
original names are saved in the VariableDescriptions property.
Stripe4_Sa1.05_1col_S.csv
Warning: Variable names were modified to make them valid MATLAB identifiers. The
original names are saved in the VariableDescriptions property.
```


Probabilty Distribution of IDR1 Given Sa = 0.5g
60
50
40
30
20
10
0
0
0.05
0.1
0.15
IDR1

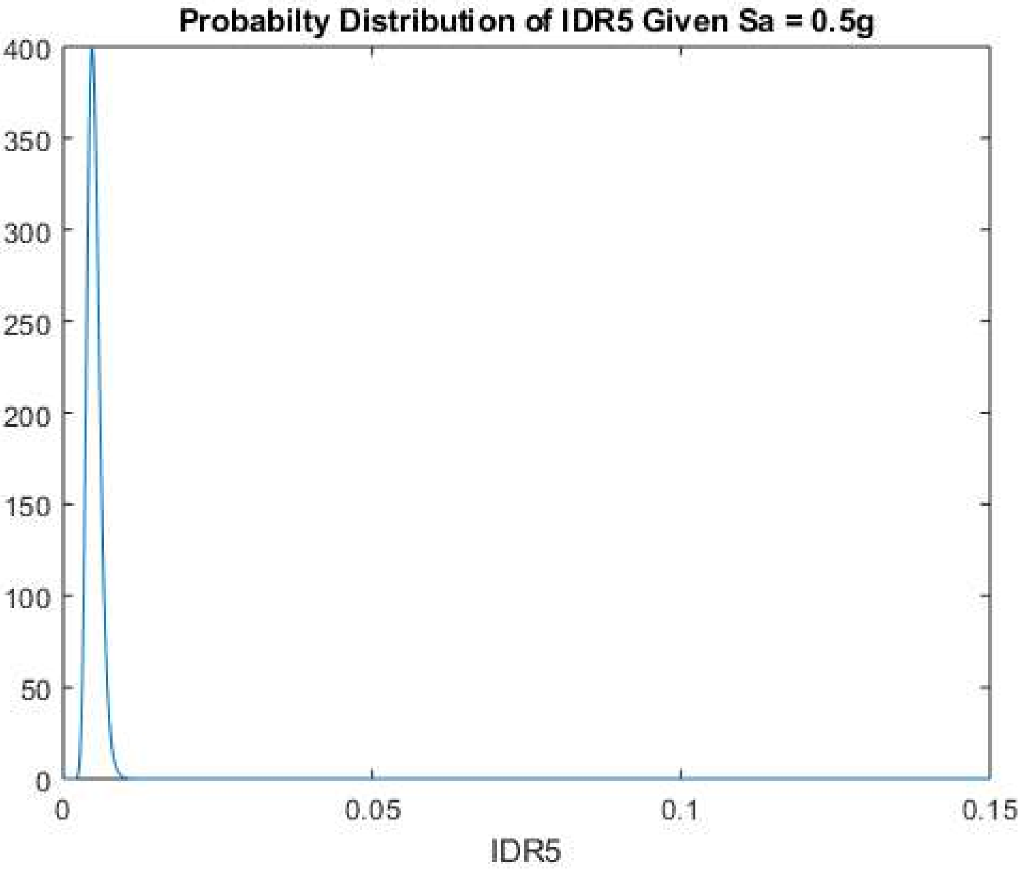

Probabilty Distribution of IDR2 Given Sa = 0.5g
140
120
100
80
60
40
20
0
0
0.05
0.1
0.15
IDR2

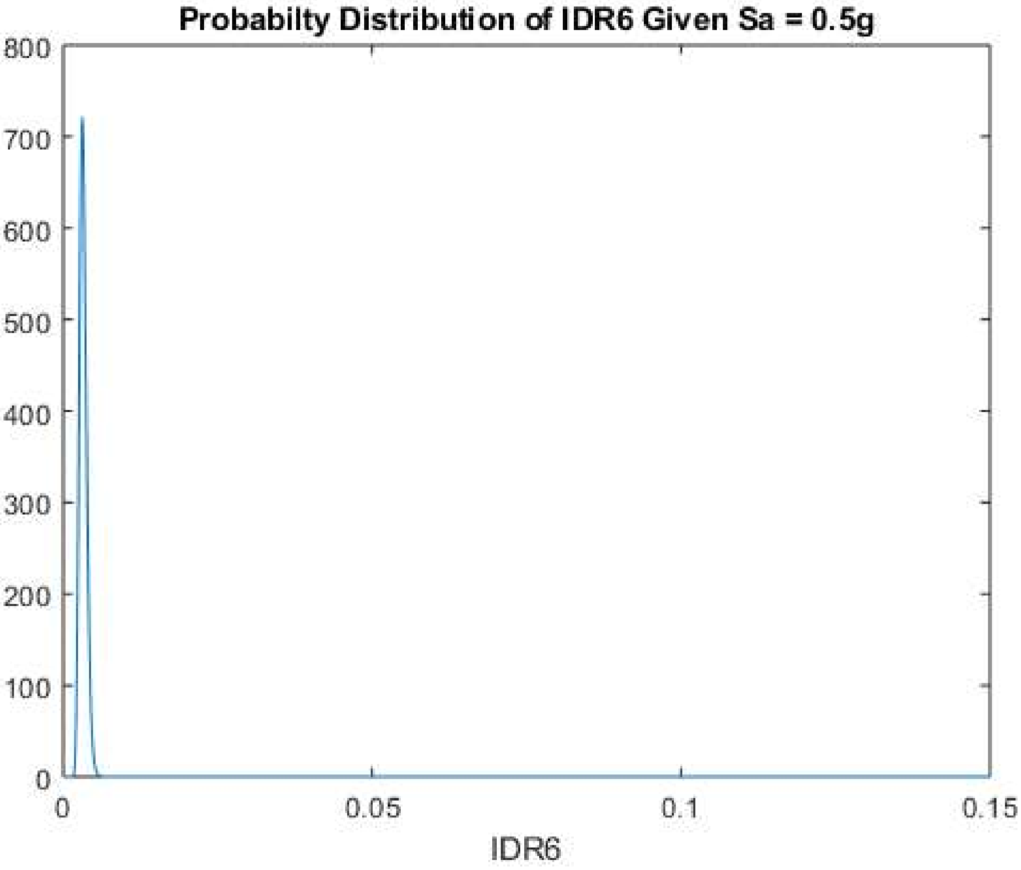


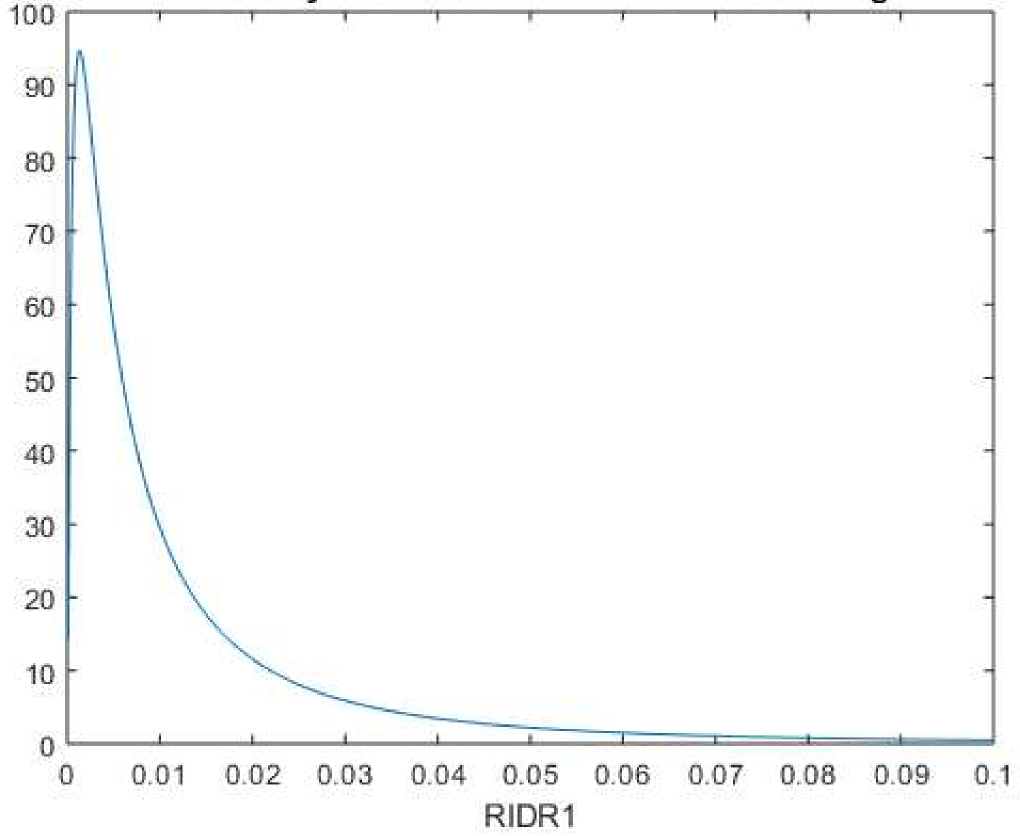
Probabilty Distribution of IDR3 Given Sa = 0.5g



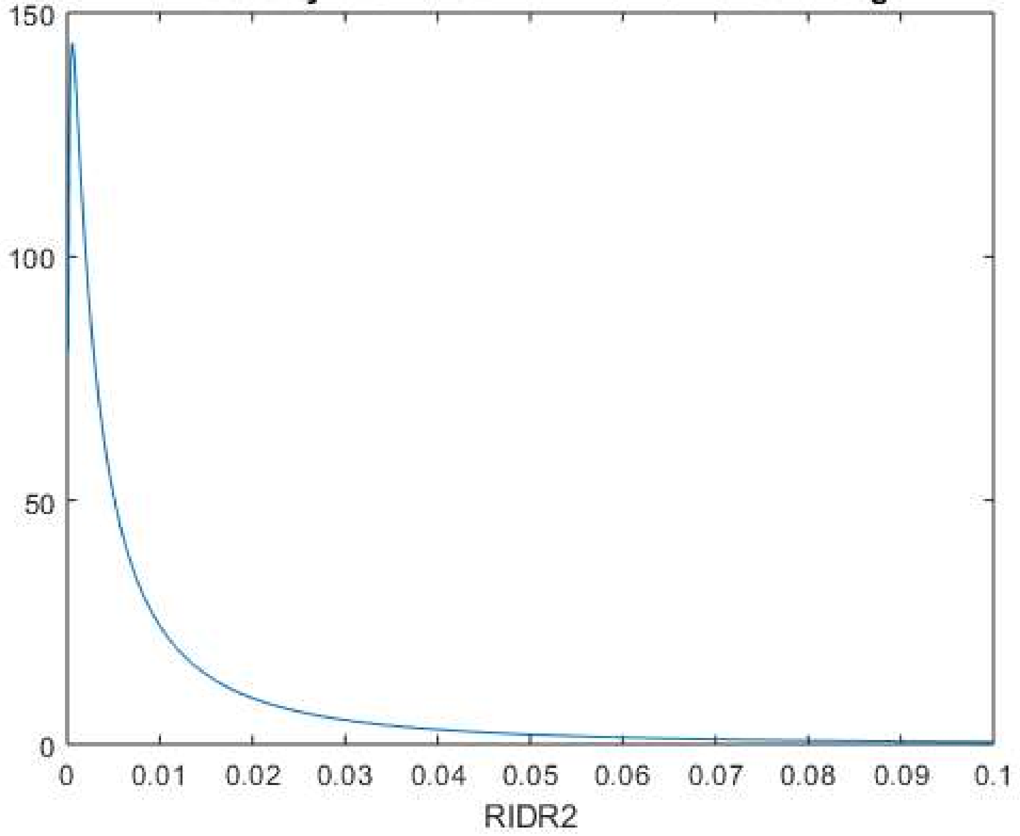
Probabilty Distribution of IDR4 Given Sa = 0.5g


Probabilty Distribution of IDR5 Given Sa = 0.5g
400
350
300
250
200
150
100
50
0
0
0.05
0.1
0.15
IDR5

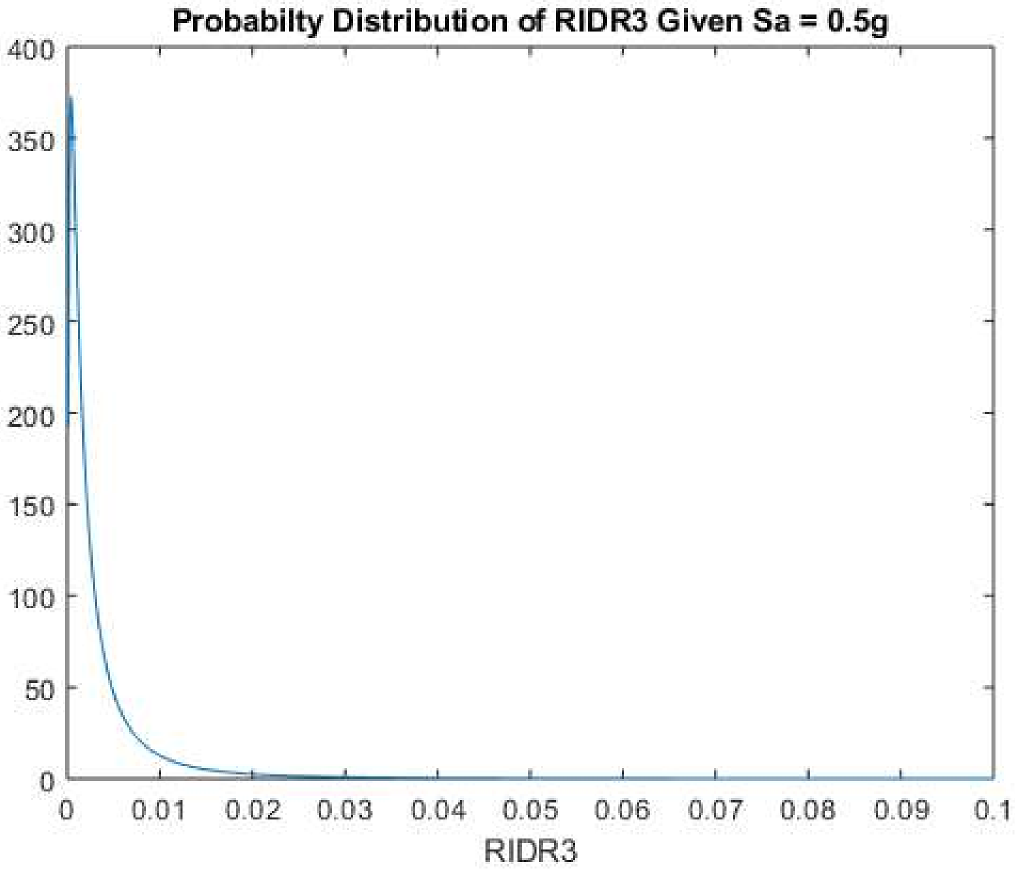

Probabilty Distribution of IDR6 Given Sa = 0.5g
800
700
600
500
400
300
200
100
0
0
0.05
0.1
0.15
IDR6



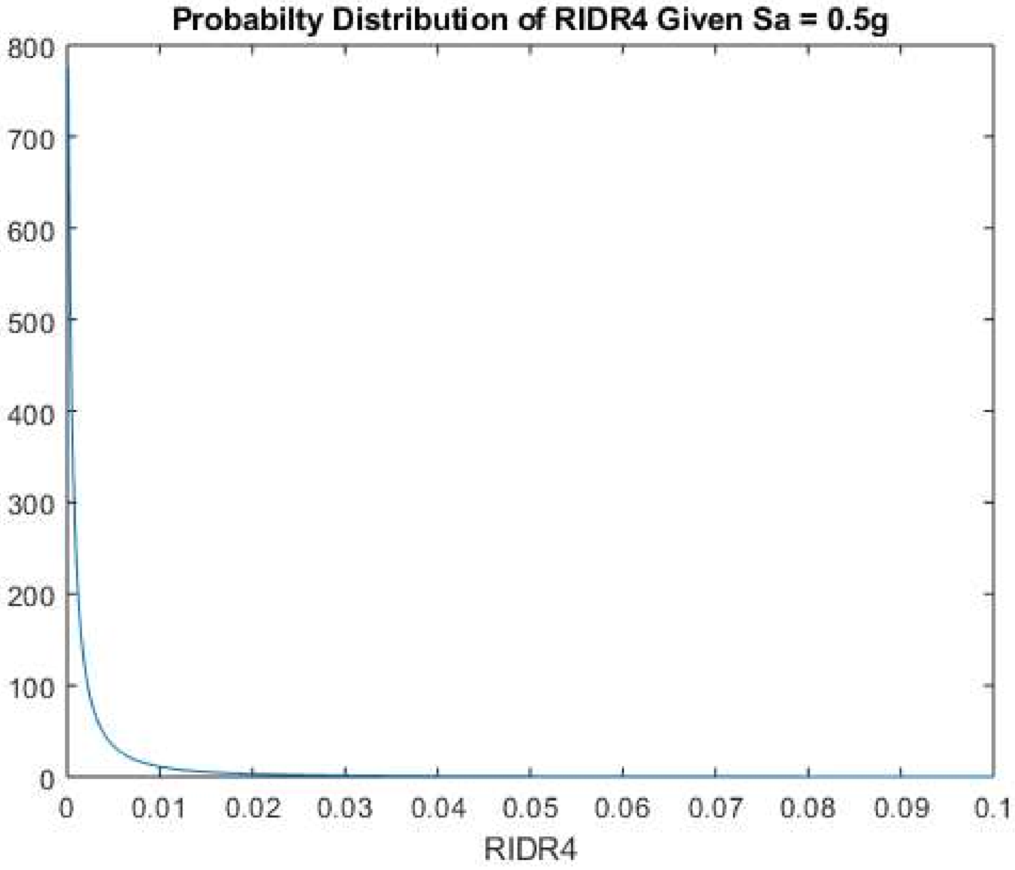
Probabilty Distribution of RIDR1 Given Sa = 0.5g



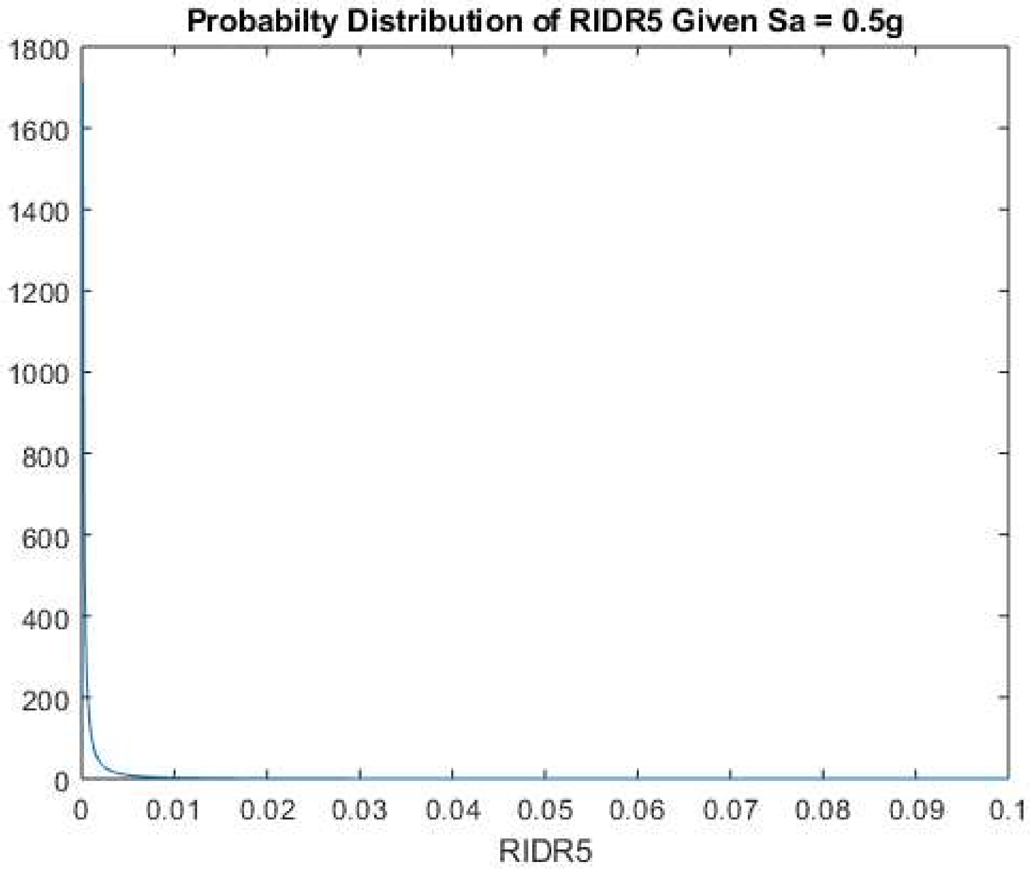
Probabilty Distribution of RIDR2 Given Sa = 0.5g

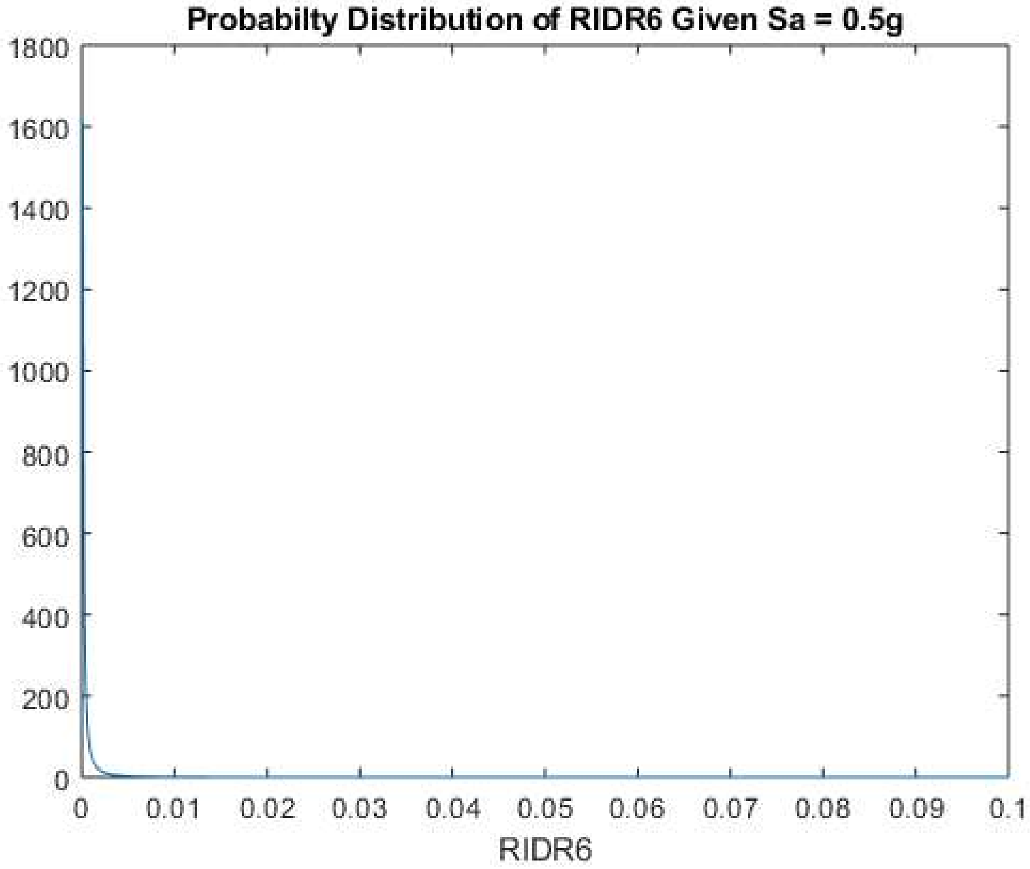


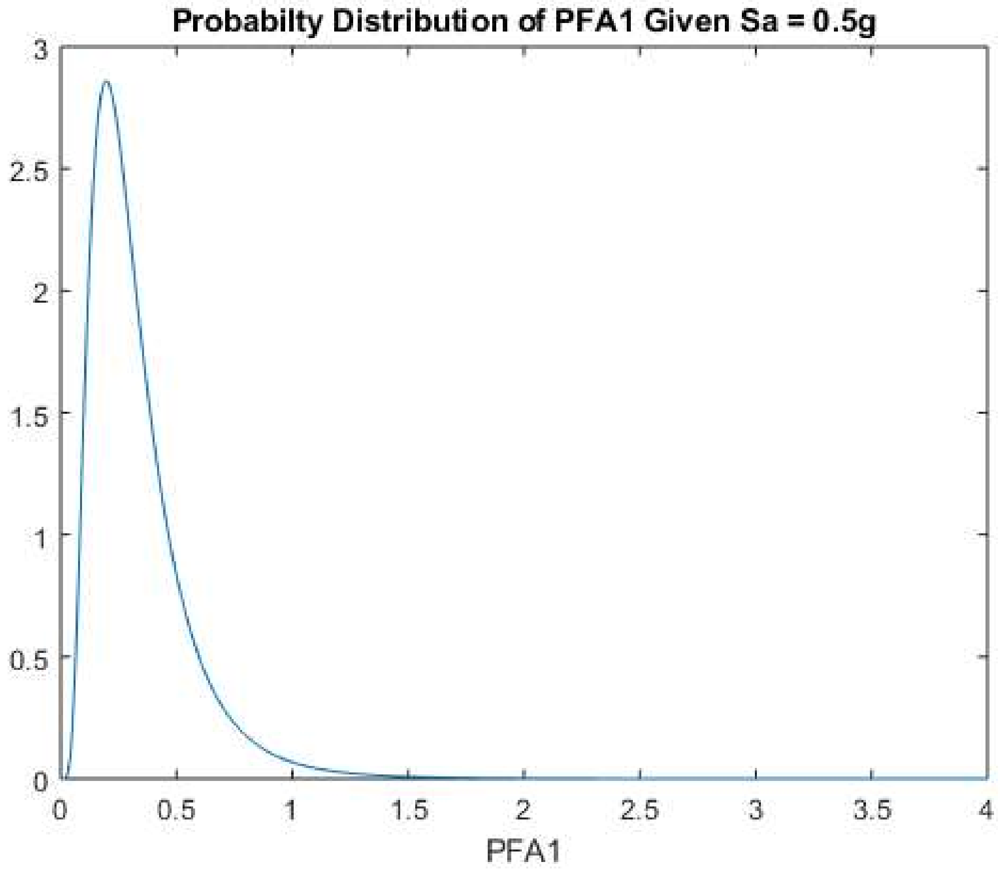
Probabilty Distribution of RIDR3 Given Sa = 0.5g

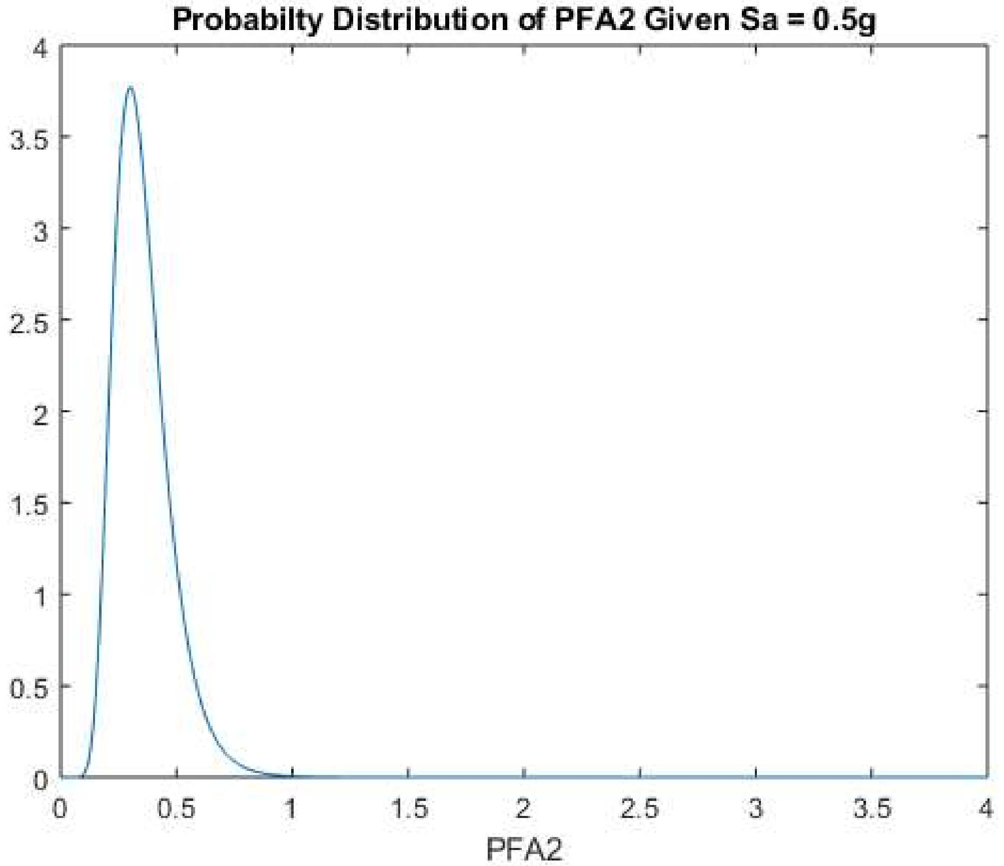


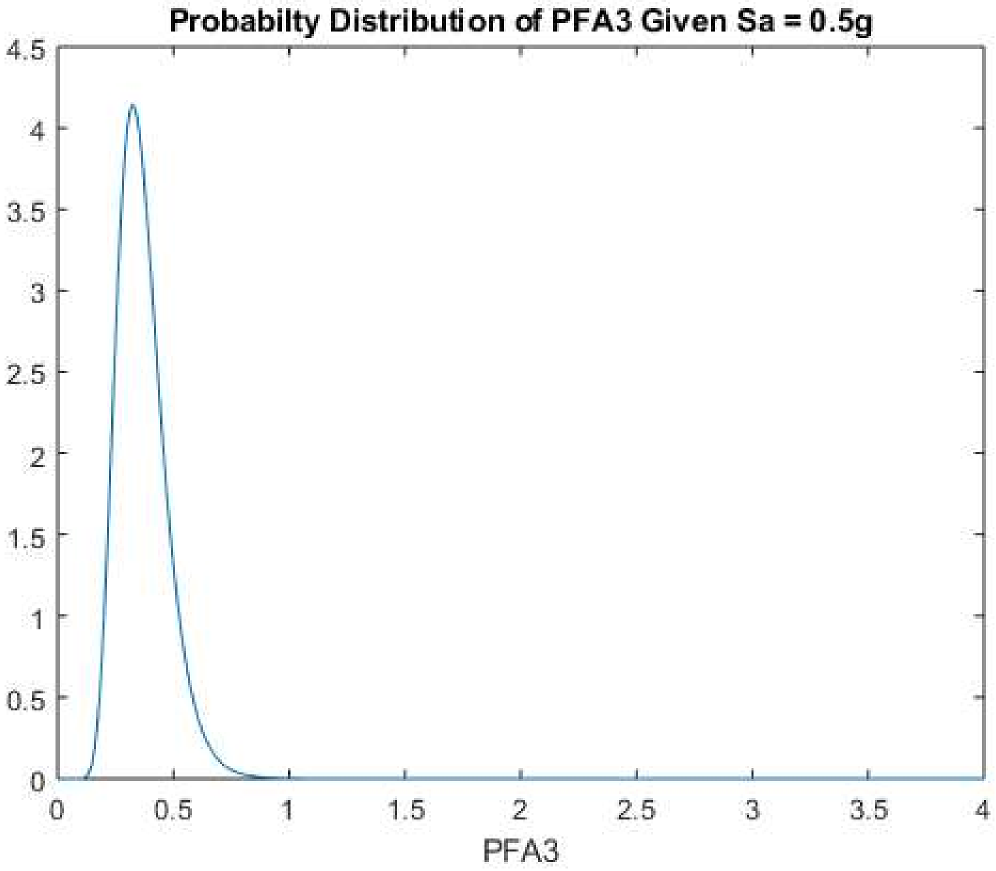
Probabilty Distribution of RIDR4 Given Sa = 0.5g

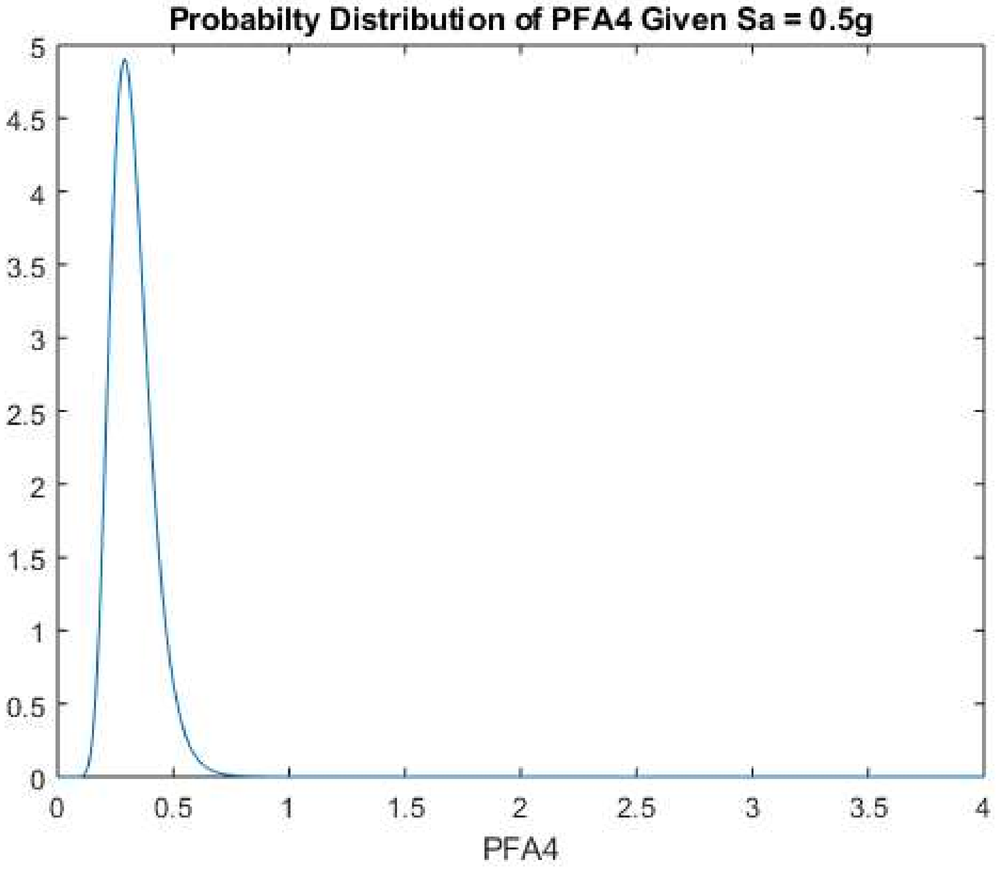

Probabilty Distribution of RIDR5 Given Sa = 0.5g
1800
1600
1400
1200
1000
800
600
400
200
0
0
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0.1
RIDR5

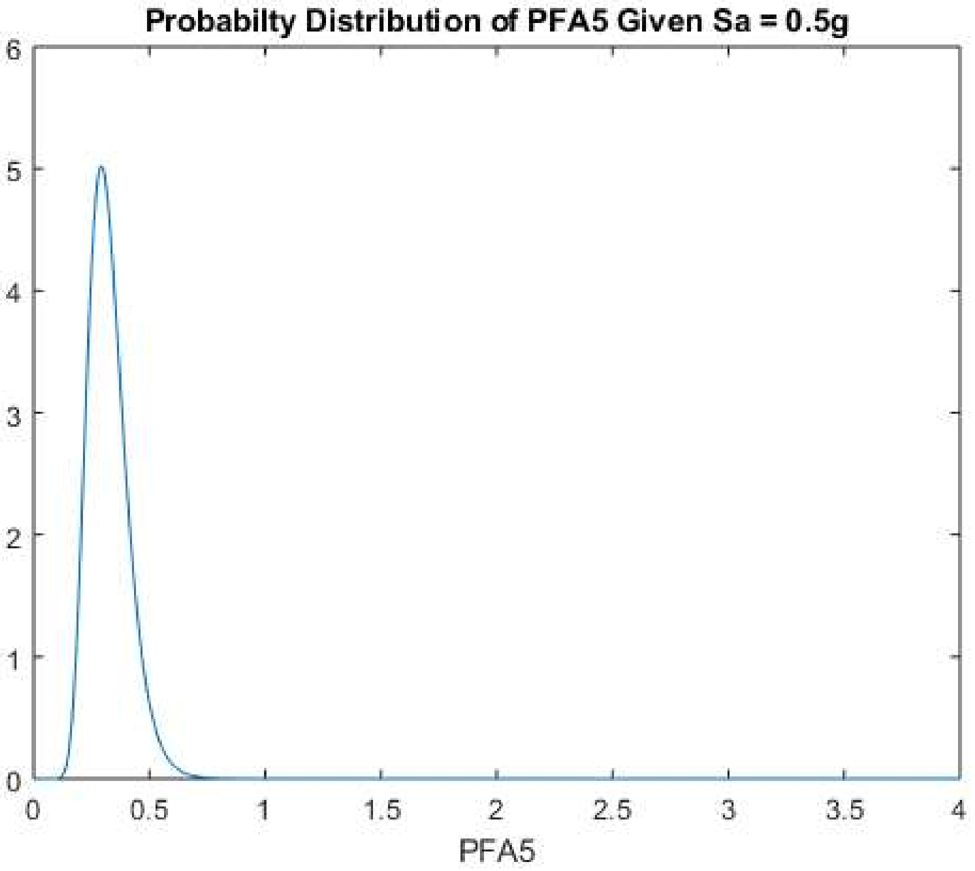

Probabilty Distribution of RIDR6 Given Sa = 0.5g
1800
1600
1400
1200
1000
800
600
400
200
0
0
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0.1
RIDR6


Probabilty Distribution of PFA1 Given Sa = 0.5g
3
2.5
2
1.5
1
0.5
0
0
0.5
1
1.5
2
2.5
3
3.5
4
PFA1

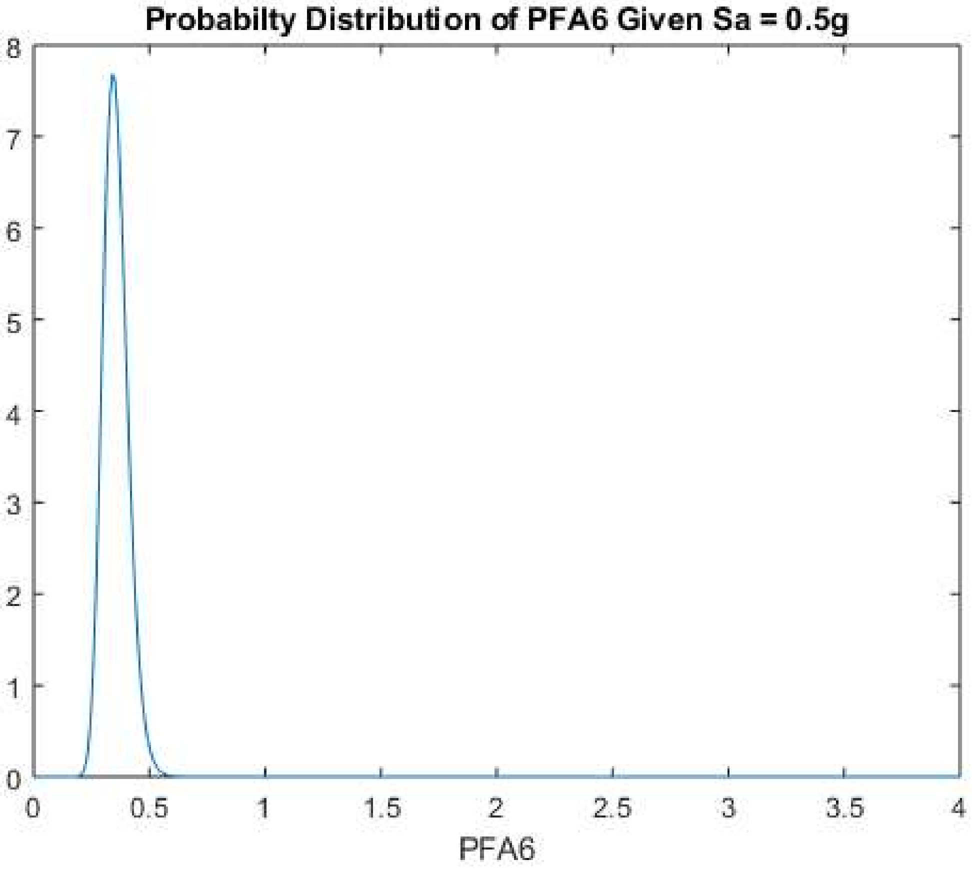

Probabilty Distribution of PFA2 Given Sa = 0.5g
4
3.5
3
2.5
2
1.5
1
0.5
0
0
0.5
1
1.5
2
2.5
3
3.5
4
PFA2


Probabilty Distribution of PFA3 Given Sa = 0.5g
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
0
0.5
1
1.5
2
2.5
3
3.5
4
PFA3

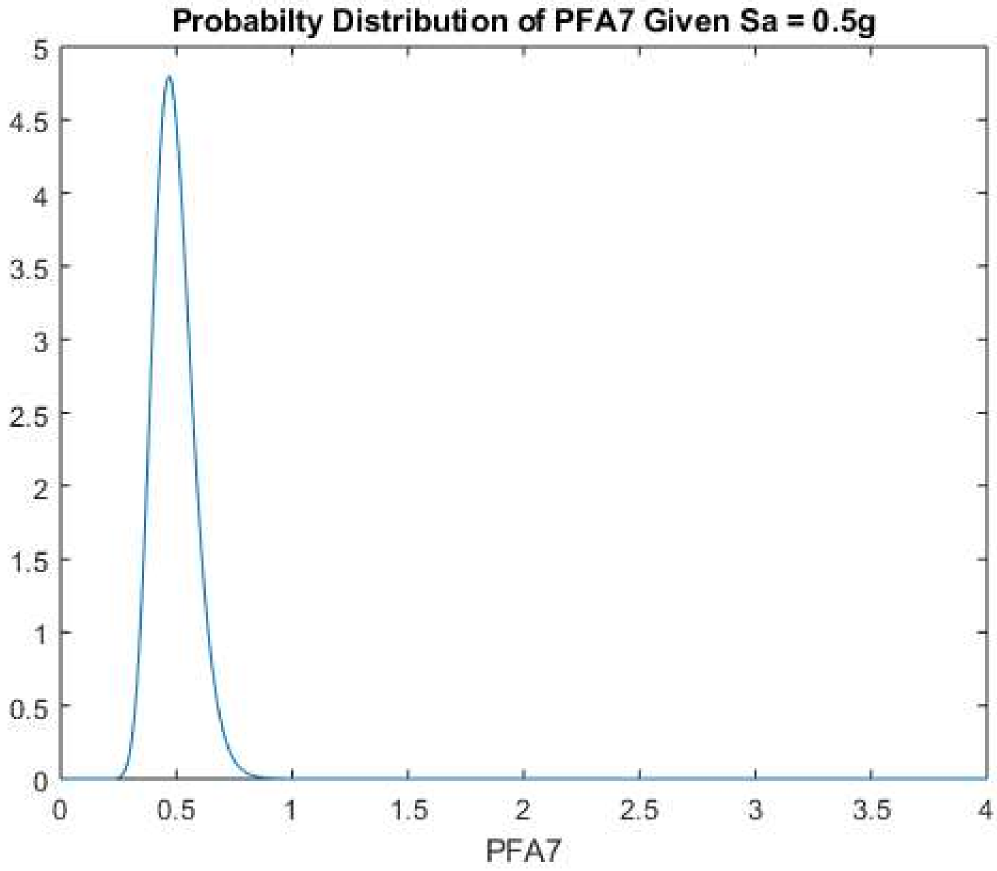

Probabilty Distribution of PFA4 Given Sa = 0.5g
5
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
0
0.5
1
1.5
2
2.5
3
3.5
4
PFA4



Probabilty Distribution of PFA5 Given Sa = 0.5g

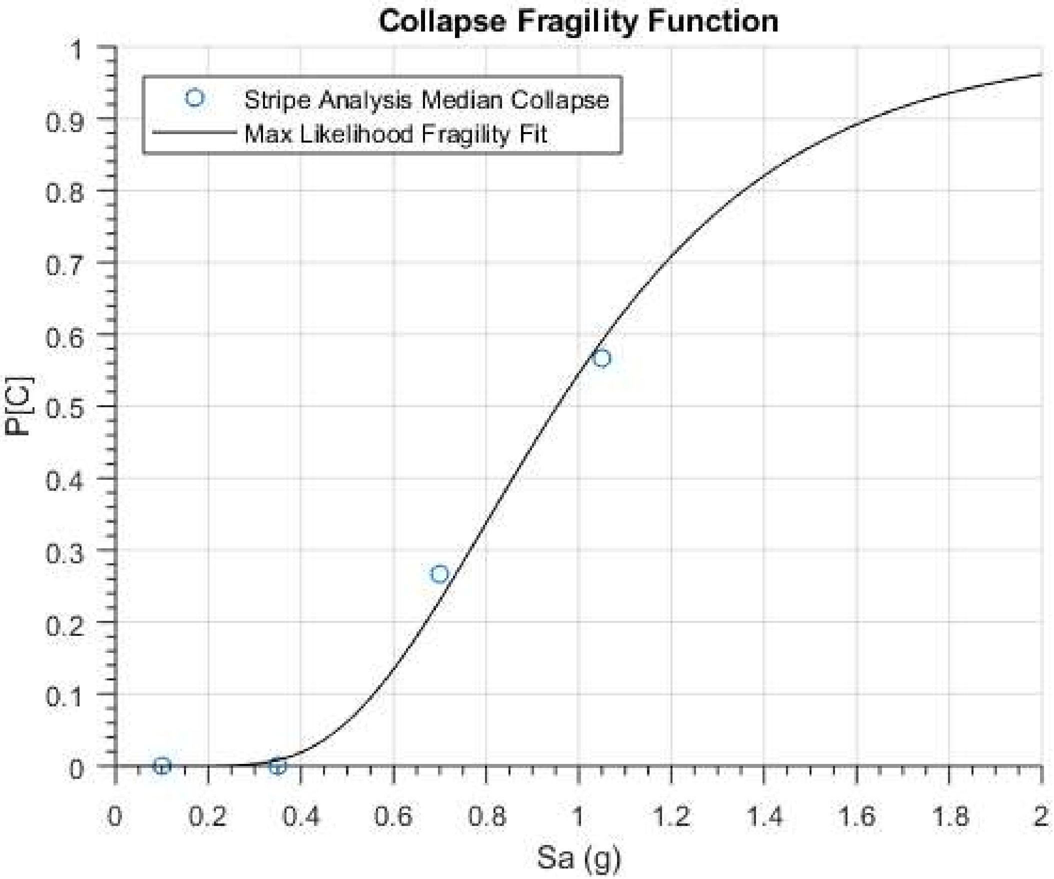


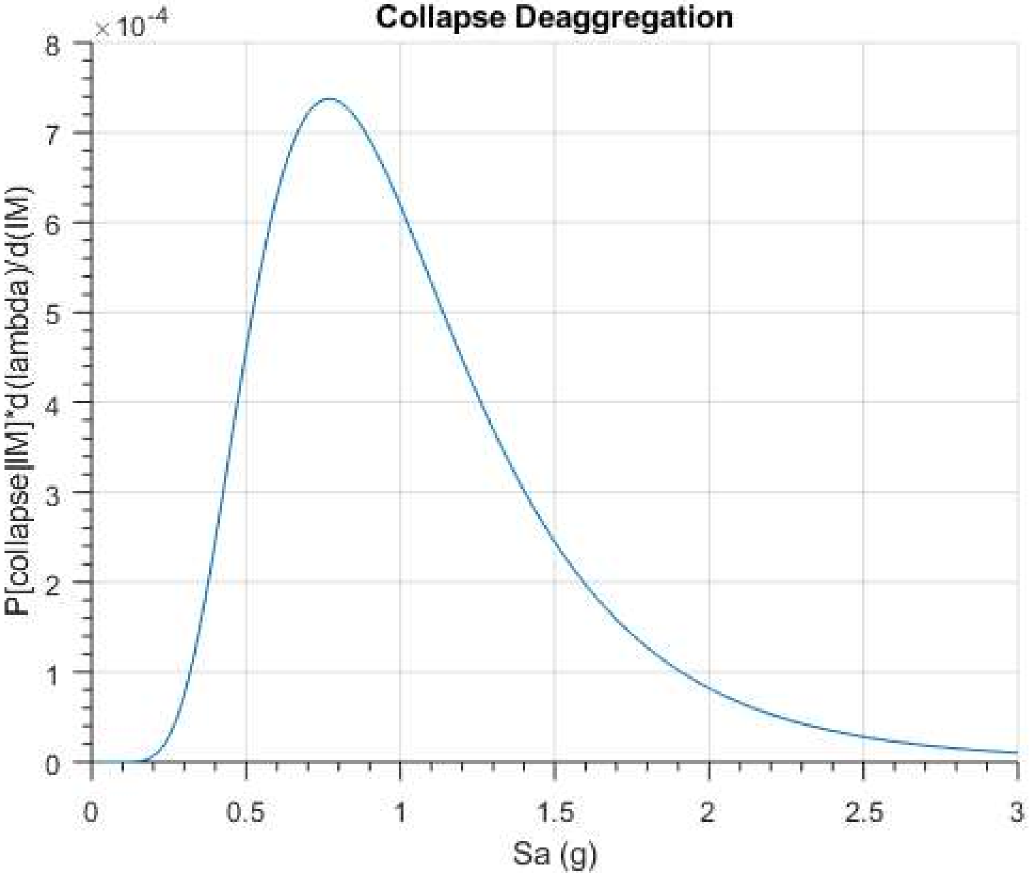
Probabilty Distribution of PFA6 Given Sa = 0.5g



Collapse Fragility, MAF, Probability in 50 years

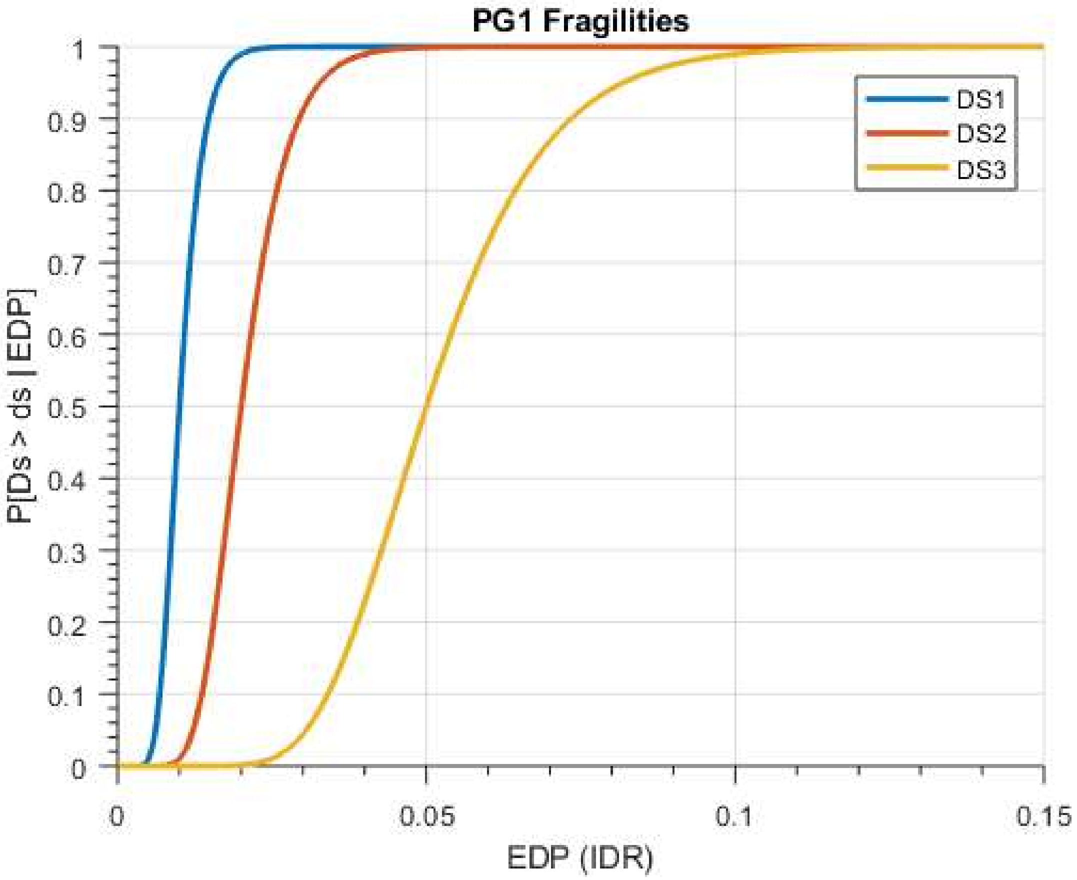
```
handles = CollapseFragility(n,handles); %Maybe split out functions for MAF and Probability in
 50 years
```

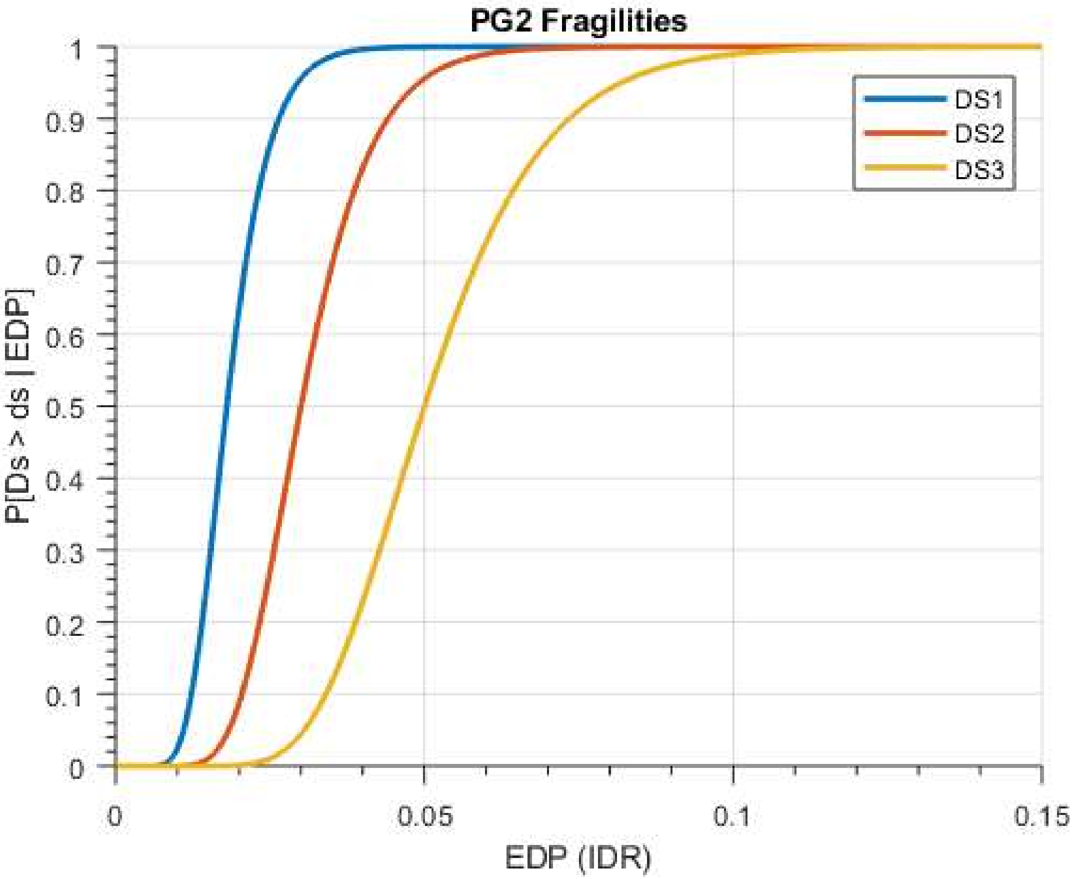


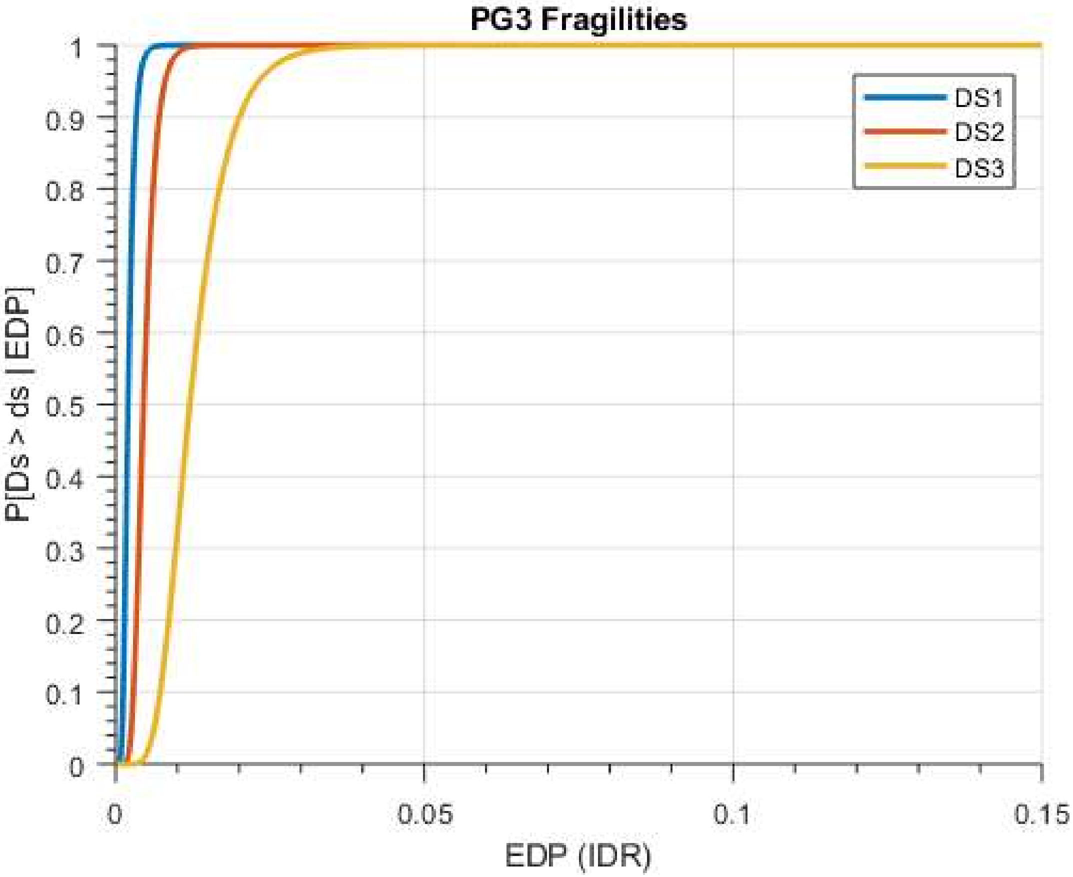


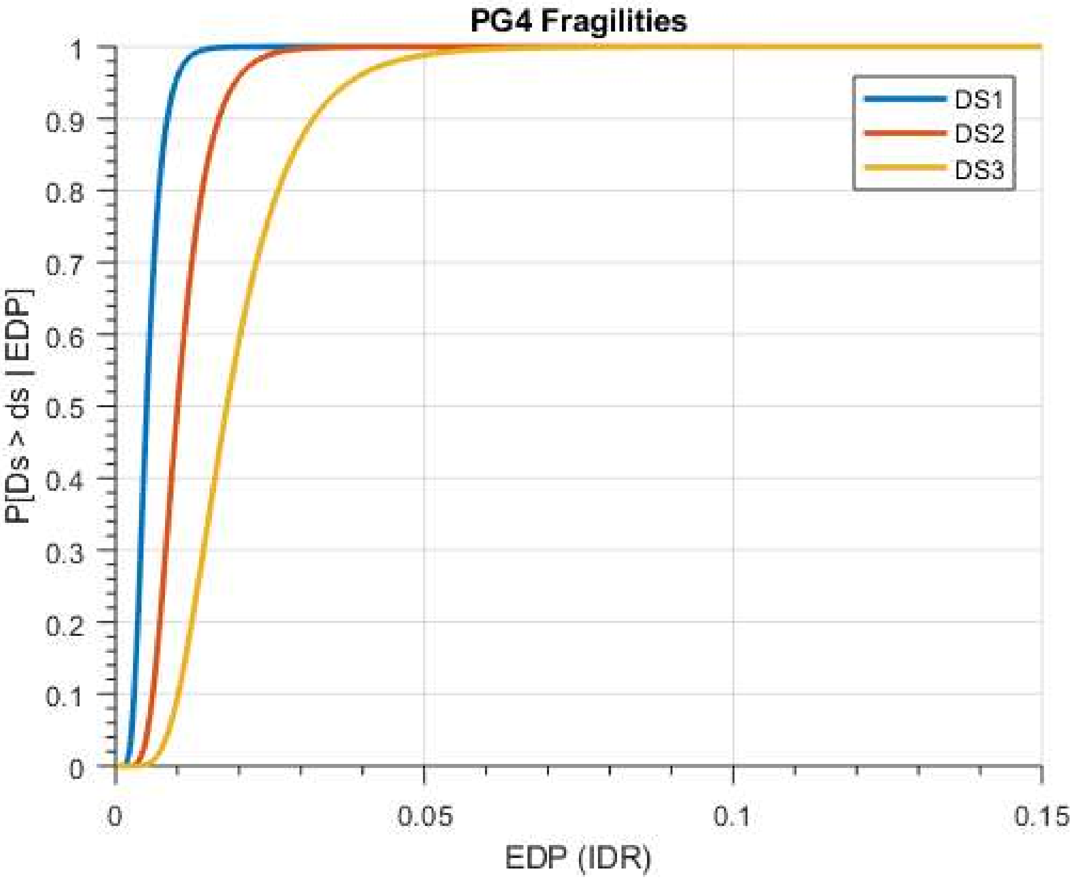
Load Fragility and Loss Functions

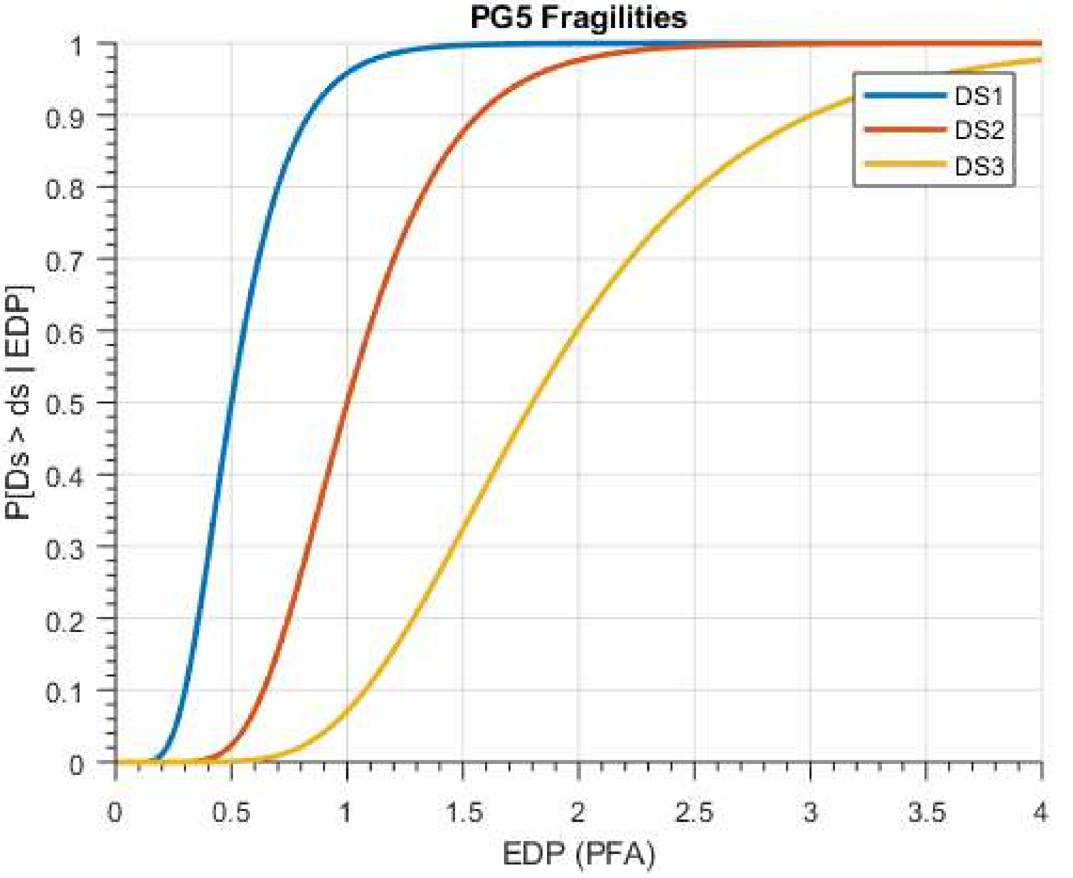
```
handles = loadComputeDamageFragilities(handles, 'SampleFragilityLossFunctionsS.csv');
```

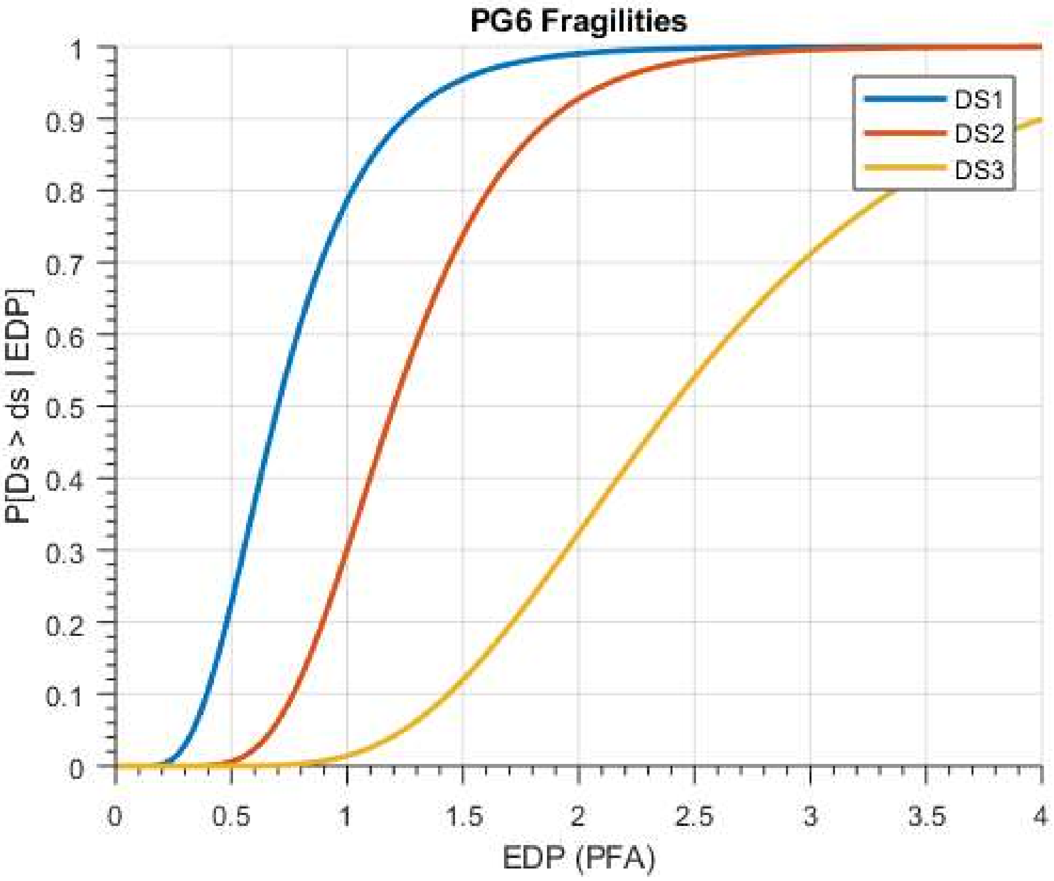


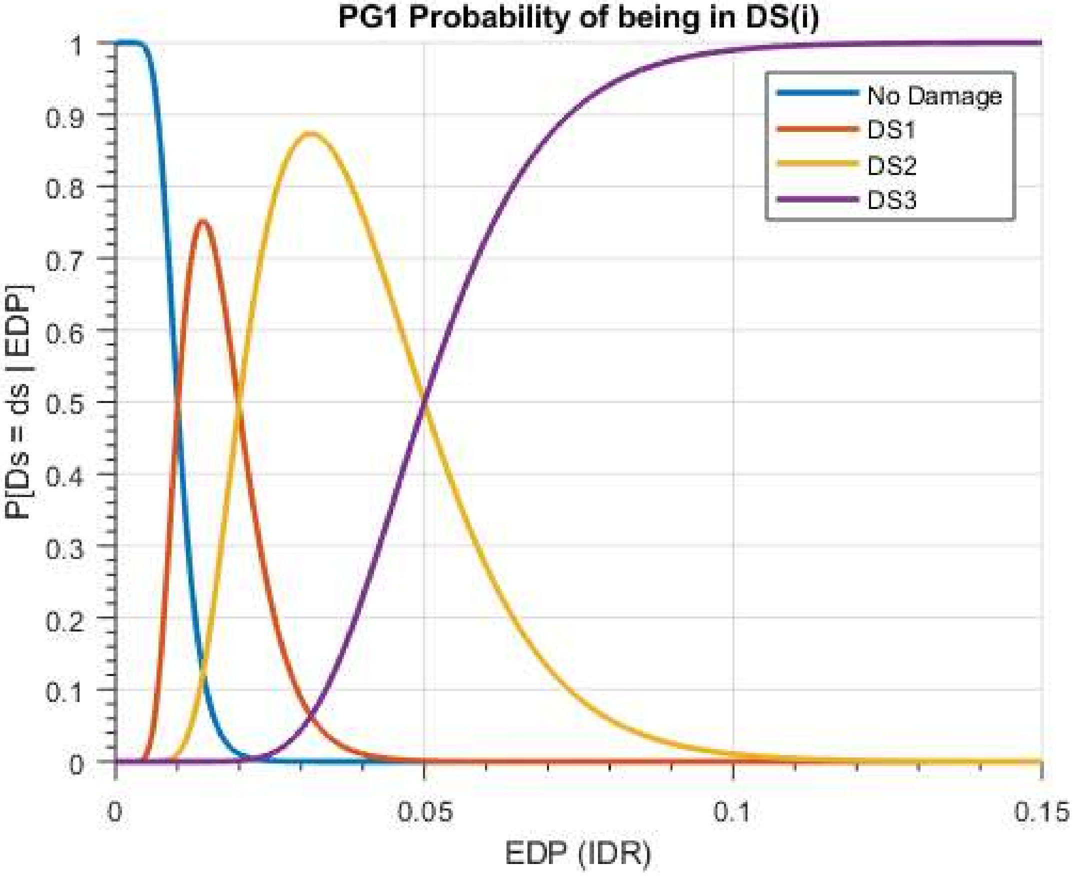

PG2 Fragilities
P[Ds > ds | EDP]
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.05
0.1
0.15
EDP (IDR)
DS1
DS2
DS3

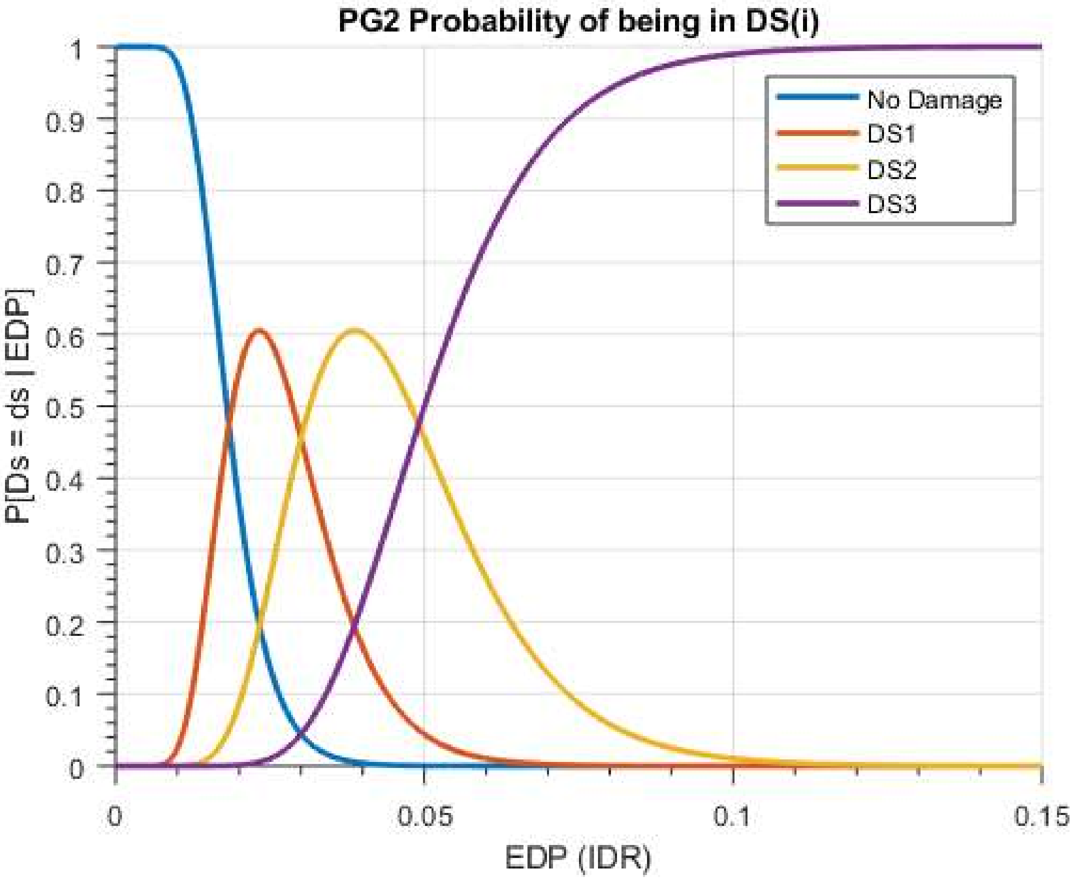

PG3 Fragilities
P[Ds > ds | EDP]
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.05
0.1
0.15
EDP (IDR)
DS1
DS2
DS3


PG4 Fragilities
P[Ds > ds | EDP]
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.05
0.1
0.15
EDP (IDR)
DS1
DS2
DS3

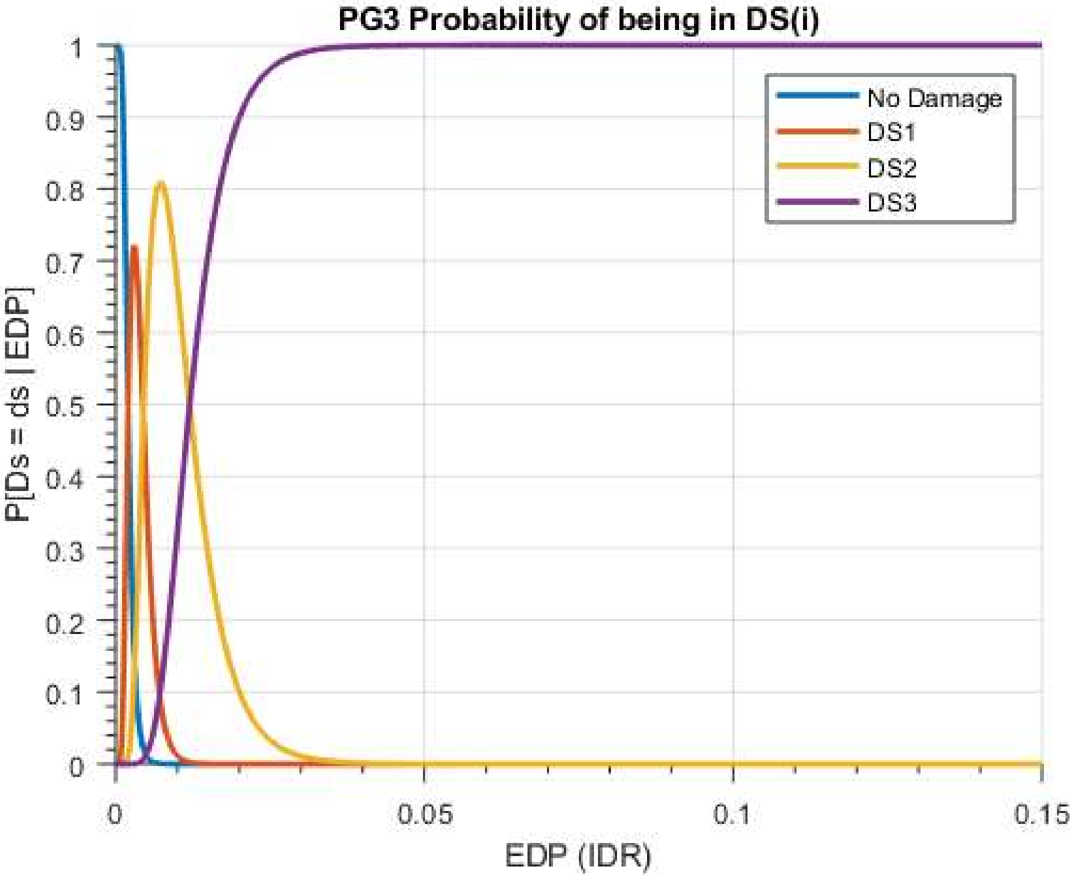

PG5 Fragilities
P[Ds > ds | EDP]
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.5
1
1.5
2
2.5
3
3.5
4
EDP (PFA)
DS1
DS2
DS3


PG6 Fragilities
P[Ds > ds | EDP]
EDP (PFA)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0.5
1.5
2
2.5
3
3.5
4
DS1
DS2
DS3

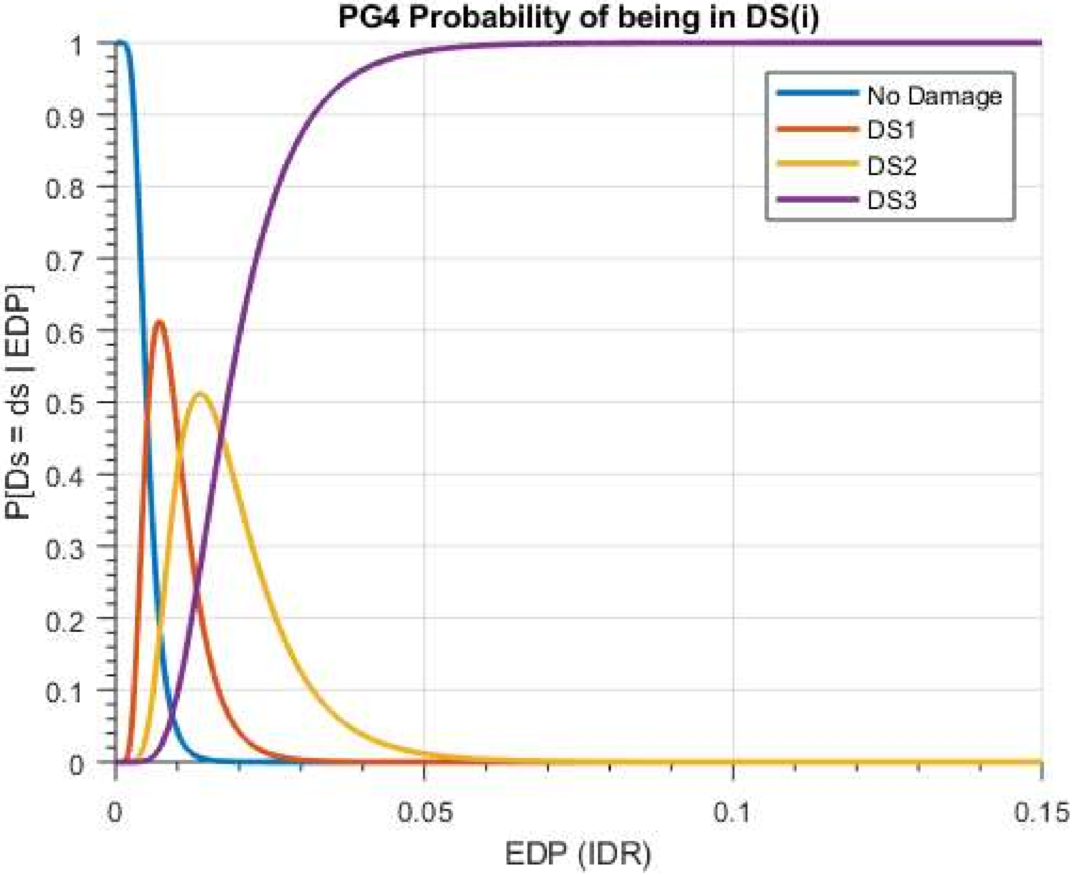

PG1 Probability of being in DS(i)
P[Ds = ds | EDP]
EDP (IDR)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0.05
0.1
0.15
No Damage
DS1
DS2
DS3

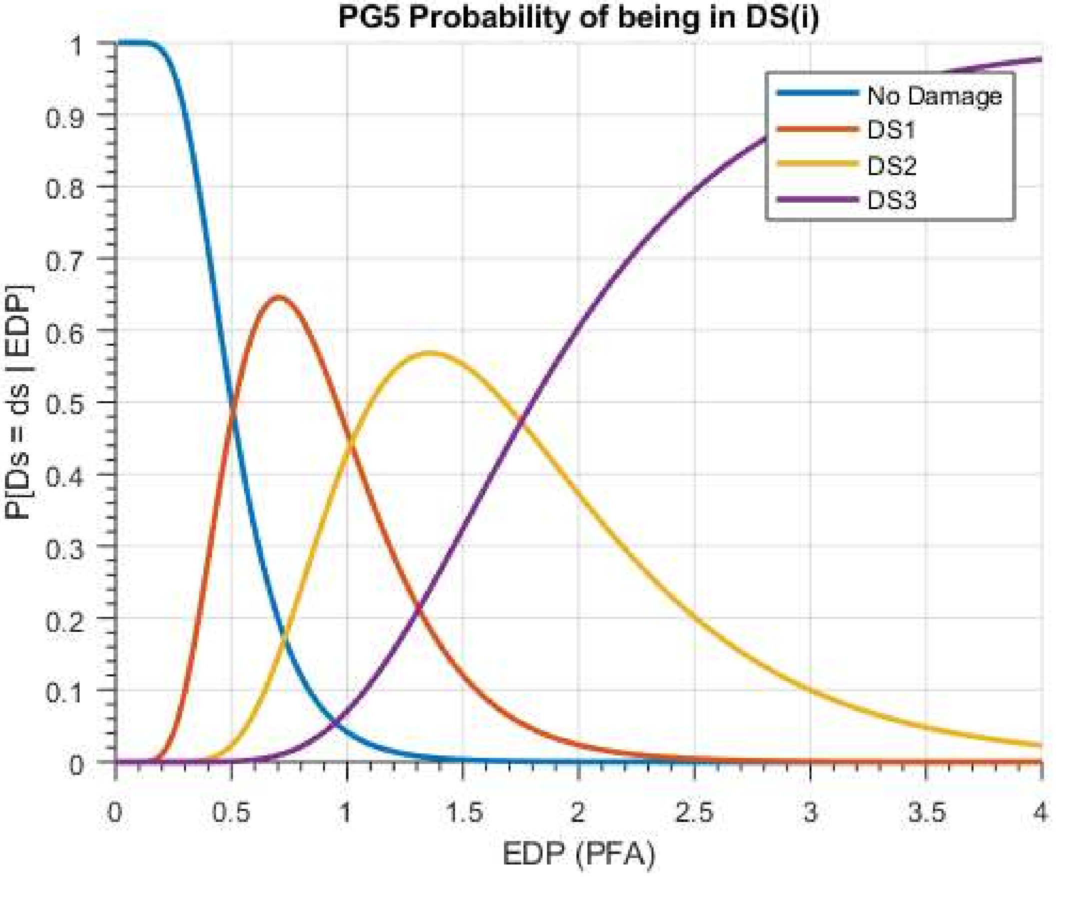


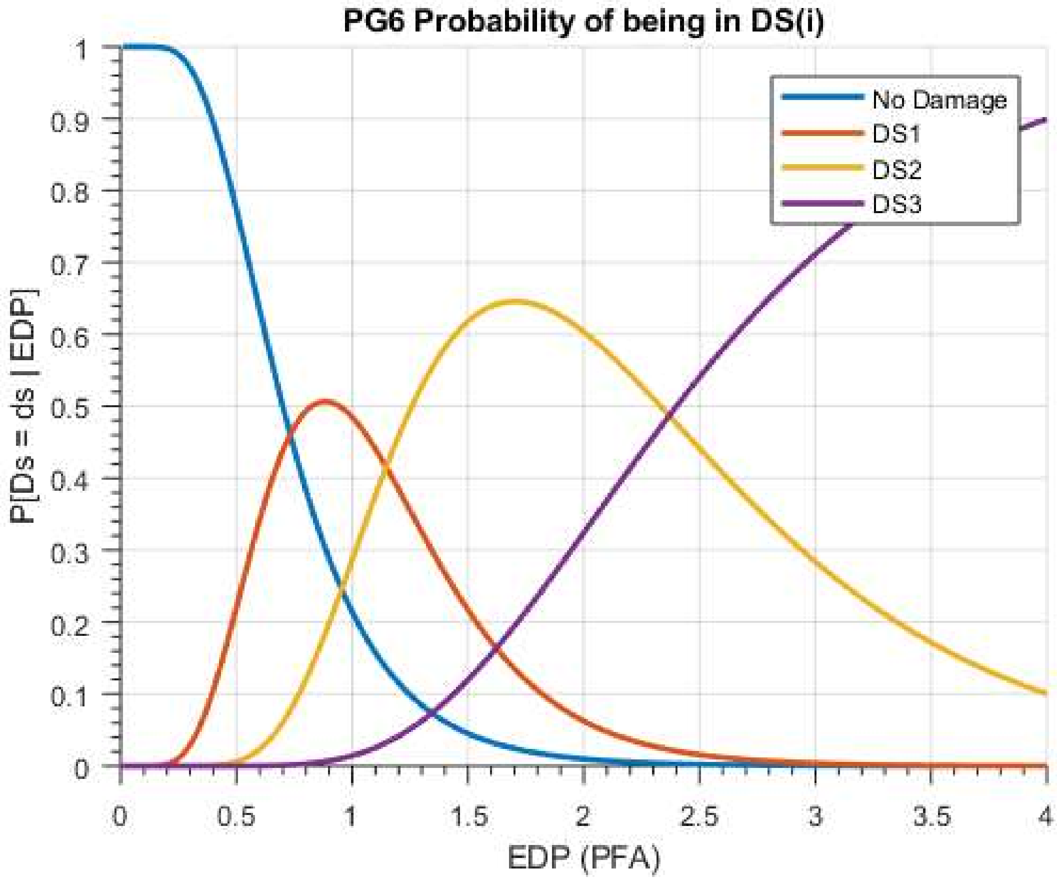
PG2 Probability of being in DS(i)



PG3 Probability of being in DS(i)


PG4 Probability of being in DS(i)
P[Ds = ds | EDP]
EDP (IDR)
0
0.05
0.1
0.15
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
No Damage
DS1
DS2
DS3


PG5 Probability of being in DS(i)
P[Ds = ds | EDP]
EDP (PFA)
0
0.5
1
1.5
2
2.5
3
3.5
4
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
No Damage
DS1
DS2
DS3



Load Structure Data

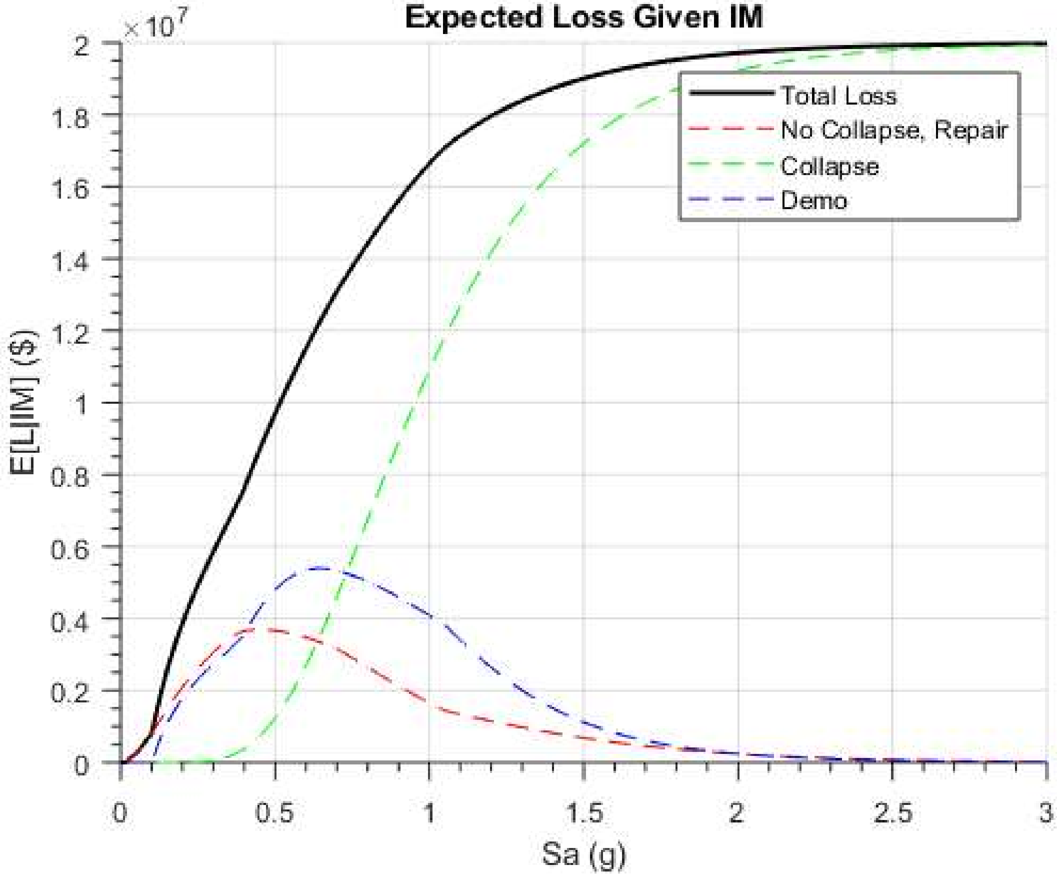
```
% Name of the file to upload
filename = 'SampleBuildingDataS.csv';
% Load in the structure data in correct format
handles = loadStructure(filename, handles);
```

Compute Loss Given IM for structure

Return expected loss given EDP conditioned on no collapse and no repair

```
handles = expectedLoss_EDP(handles);

% Return expected loss given IM conditioned on no collapse and no repair
handles = expectedLoss_IM(handles);

% Probability of Demolition Given No Collapse
handles = ExpectedLoss(handles);

% Find probabilities of all 3 ingredients (repair, demo, collapse) and plot
% all loss functions given IM
handles = repairCollapseDemoProb(handles);

% Calculate the AAL and deaggregation ratios (could be used for fun things)
handles = averageAnnualLoss(handles);
```

```
Average Annual Loss:
   1.2881e+05
```



Published with MATLAB® R2017b