Contents


```
%%This program imports data from a DemoMotor .csv file and provides
%clasification and RUL if a fault is detected.  This system is meant as an
%example and should not be used for existing applications.  Use of this
%example is at your own risk and no warranty is implied.  Contact MotorDoc
%LLC for additional information at info@motordoc.com.
```

Step 1: Setup and open matlab.mat then get the file using Windows.

```
%load(matlab.mat);
[WorkData,path]=uigetfile('*.csv');
Z=(fullfile(path,WorkData));
X=1;
```

Step 2: Import the data from the text file. Copy 'importfile.m'.

```
%IMPORTFILE Import data from a text file
%  MTR1 = IMPORTFILE(FILENAME) reads data from text file FILENAME for
%  the default selection.  Returns the data as a table.
%
%  MTR1 = IMPORTFILE(FILE, DATALINES) reads data for the specified row
%  interval(s) of text file FILENAME. Specify DATALINES as a positive
%  scalar integer or a N-by-2 array of positive scalar integers for
%  dis-contiguous row intervals.
%
%  Example:
%  mtr1 = importfile("C:\Users\howar\OneDrive\Documents\MATLAB Projects\Motor for article\Data\mtr1.csv", [2, Inf]);
%
%  See also READTABLE.
%
% Auto-generated by MATLAB on 27-Aug-2021 10:31:33
```

Set up the Import Options and import the data

```
opts = delimitedTextImportOptions("NumVariables", 15);

% Specify range and delimiter
%opts.DataLines = dataLines;
opts.Delimiter = ",";

% Specify column names and types
opts.VariableNames = ["ID", "Va", "Vb", "Vc", "Aa", "Ab", "Ac", "watts",...
    "rise", "speed", "PF", "Vu", "Au", "vibe"];
opts.VariableTypes = ["double", "double", "double", "double", "double",...
    "double", "double", "double", "double", "double", "double", "double", ...
    "double", "double"];

% Specify file level properties
opts.ImportErrorRule = "error";
```

```
opts.ExtraColumnsRule = "ignore";
opts.EmptyLineRule = "read";

% Specify variable properties
opts = setvaropts(opts, ["ID", "Va", "Vb", "Vc", "Aa", "Ab", "Ac",...
    "watts", "rise", "speed", "PF", "Vu", "Au", "vibe"], "FillValue", 0);

% Import the data
TestDataAnalysis1 = readtable(Z);
%
clear opts;
%
```

Step 3: Smooth the Data

```
TestDataAnalysis = smoothdata(TestDataAnalysis1,"movmedian",5,...
    'DataVariables',["Va", "Vb", "Vc", "Aa", "Ab", "Ac", "watts",...
    "rise", "speed", "PF", "Vu", "Au", "vibe"]);
```

Step 4: Parse data to obtain speed

```
tt=tail(TestDataAnalysis,1);
X=tt.speed;
if X>0
```

```
    %%Step 5: Pre-set classification and remaining useful life (estRUL);
    yfit=0;
    estRUL=10000;
    %PerformClassification
    yfit=trainedModel1.predictFcn(tt);
    if yfit==0
        Cond=0;
        disp("Good");
    elseif yfit==1
        Cond=1;
        disp("Unbalance");
    else
        Cond=2;
        disp("Severe Unbalance");
    end
```

```
Good
```

Step 5: Remaining Useful Life Evaluations

```
    %Determine how many lines in data for evaluation and select last 800
    th=height(TestDataAnalysis);
    if th>800
        th=800;
    end
    ttd=tail(TestDataAnalysis,th);
    TAu=predictRUL(mdlUnbal, ttd(th,:),thresholdAu);
    TVu=predictRUL(mdlUnbala,ttd(th,:),thresholdVu);
   % TPF=predictRUL(mdlUnbalb,ttd(th,:),thresholdPF);

    if TAu < TVu
        estRUL=TAu;
    else
```

```
        estRUL=TVu;
    end
    if TPF<estRUL
        estRUL=TPF;
    else
        estRUL=TPF;
    end
    if estRUL<800
        disp(estRUL);
    else
        disp("Monitor");
    end
```

```
    0 hr
```

```
else
        disp("Motor Off")
end
```

Published with MATLAB® R2021b