

```

%%Development of test and training data for future electric motor
%%Motor design data not available, working from nameplate and published
%%curves. For this example, Tamb will be the ambient temperature in degrees
%%C, IcyC is the number of rows we will create in our test data, FaultCodes
%%will be the number equal to the type of fault, or if a fault exists, and
%%Mload will be the motor load. Code can be developed that can make any of
%%these variable.
%%For more information contact MotorDoc LLC at info@motordoc.com
%%This is for demonstration of a process only. Use of this code implies
%%no warranty nor should be used for applications. Formal training in
%%machine learning and languages is recommended before applying in real-
%%world applications.
Tamb = 20;
IcyC = 800;
FaultCode = 0;
Mload = 75;
Q=10;
%%Create data tables which will become part of the workspace, do this
%%complete setup ii number of times.
for ii=1:Q
    Motor1 = table('size',[0 15], 'VariableTypes', {'double','double', ...
        'double','double','double','double','double','double','double', ...
        'double','double','double','double','double','double'});
    Motor1.Properties.VariableNames={'ID','Va','Vb','Vc','Aa','Ab', ...
        'Ac','watts','rise','speed','PF','Vu','Au','vibe','FaultCode'};
    size(Motor1);
    %%Populate a number of tables
    for i=1:IcyC
        %set up the number of rows of data with ID for time first
        ID = i;
        %next we set up speed (fixed) for the appropriate load
        X=(0.0002667*Mload^3)-(0.06176*Mload^2)+(4.237*Mload)+1707;
        %then some operating variation. This can be more creative with
        %variations following expected electrical and thermal conditions
        speed = (((X-1)-(X+1))*rand(1,1)+(X-1));
        %determine base temperature rise from curve
        T=((3.238e-5)*Mload^4)-(0.01*Mload^3)+(1.129*Mload^2)- ...
            (54.45*Mload)+967.9;
        %generate some variation in temperature rise and add ambient
        %temperature
        rise = (((T-1)-(T+1))*rand(1,1)+(T-1))+Tamb;
        %determine watts including some variation
        watts=Mload+rand(1,1);
        %Calculate voltages and random conditions
        Va = ((455-461)*rand(1,1)+455);
        Vb = ((455-461)*rand(1,1)+455);
        Vc = ((455-461)*rand(1,1)+455);
        %calculate currents and random conditions
        Aa = ((80-90)*rand(1,1)+80);
        Ab = ((80-90)*rand(1,1)+80);
        Ac = ((80-90)*rand(1,1)+80);
        %determine average voltage and current
        Vave = (Va + Vb + Vc)/3;
        Aave = (Aa + Ab + Ac)/3;
        %calculate power factor with known information
        PF = 1-((Mload*100)/(1.732*Vave*Aave));
        %determine voltage and current differences from average
        V1a = abs(Vave-Va);
        V1b =abs(Vave-Vb);
    end
end

```

```

V1c = abs(Vave-Vc);
A1a = abs(Aave-Aa);
A1b = abs(Aave-Ab);
A1c = abs(Aave-Ac);
%pick highest voltage or current, divide by average and multiply by
%100% to obtain percentage unbalance
if V1a >= V1b && V1a >= V1c
    V2 = V1a;
elseif V1b >=V1a && V1b>=V1c
    V2=V1b;
else
    V2=V1c;
end
Vu=(V2/Vave)*100;
if A1a >= A1b && A1a >= A1c
    A2 = A1a;
elseif A1b >=A1a && A1b>=A1c
    A2=A1b;
else
    A2=A1c;
end
Au=(A2/Aave)*100;
%calculate vibration with variations
vibe = ((1-2)*rand(1,1)+1);
%generate a fault code. This will be updated when this is used to
%calculate failed conditions for training.
FaultCode = 0;
NewData={ID, Va, Vb, Vc,Aa,Ab,Ac,watts,rise,speed,PF,Vu,Au, ...
    vibe,FaultCode};
Motor1=[Motor1;NewData];
size(Motor1);

end
mtr="mtr"+ii;
writetable(Motor1,mtr+'.csv');

end

```