Politecnico di Bari

DIPARTIMENTO DI INGEGNERIA ELETTRICA E DELL'INFORMAZIONE

Calcolatore automatico di incertezze per multimetri digitali

- Corso di Fondamenti delle Misurazione -

- Ingegneria Informatica e dell'Automazione -

A cura di:
Laura Loperfido
Andrea Lops
Teresa Pantone
Paolo Rotolo

Prof. Ing. Filippo Attivissimo

```
%% Input parameter (part1)
clear
clearvars
clc

disp('*************************************************************************************')
disp('**** Welcome to the automatic uncertainty calculator for digital multimeters ****')
disp('*************************************************************************************')
disp(' ')
disp(' ')
disp(' ')

choice = 0;
disp('What kind of multimeter did you use?')

while(choice~=5) % set multimeter name
    choice = 0;
    while(choice<1||choice>4)
        disp('1 - HP 974A')
        disp('2 - Keysight U1253B')
        disp('3 - Fluke 189')
        disp('4 - Agilent 34401A')
        choice = input('Insert choice: ');
    end

    if(choice==1)
        multimeter = "HP 974A";
        break
    elseif(choice==2)
        multimeter = "Keysight U1253B";
        break
    elseif(choice==3)
        multimeter = "Fluke 189";
        break
```

```
    elseif(choice==4)
        multimeter = "Agilent 34401A";
        break
    end

end

disp(' ')
disp(' ')

choice_meas = 0;
while(choice_meas~=4) % set meas type
    choice_meas = 0;
    while(choice_meas<1||choice_meas>3)
        disp('1 - Vdc')
        disp('2 - R')
        disp('3 - Idc')
        choice_meas = input('Insert choice: ');
    end

    if(choice_meas==1)
        meas_type = "Vdc";
        break
    elseif(choice_meas==2)
        meas_type = "R";
        break
    elseif(choice_meas==3)
        meas_type = "Idc";
        break
    end

end

disp(' ')
disp(' ')

%% Read file
filename = 'Spec_mul.xlsx';
sheet = multimeter;

T = readtable(filename,'Sheet',sheet); % ALL spec data

%% Specs table
rows = find(strcmp(T.meas_type, meas_type)); % selected rows
Specs = T(rows,:);

%% Specs arrays and data
FS = Specs.range; % read fs value
U_G = Specs.U_G; % read uncertainty coeffcient
meas_recommended = num2str(length(FS));

%% Input parameter (part2)

txt_input = ['How many measurements did you make? (',meas_recommended,' Recommended): '];
length_x = input(txt_input); % set number of meas
x_=[]; % x_ = column vector of meas
for i=1:length_x % fill the vector
    x_i = input('Enter the measure: ');
    last_FS = FS(length(FS));
    if  x_i>last_FS % meas must be less than the maxium value of FS
        disp('The inserted meas is bigger than the value of the last FS, insert a coherent m
```

```
    else
        x_(end+1)=x_i;   %#ok<SAGROW>
    end
end

%% FS uncertainty calculation
if Specs.U_FS(1)<1
     U_FS = FS.*Specs.U_FS/100; % full scale uncertainty
else
     U_FS = Specs.Q.*Specs.U_FS;
end

%% Uncertainty computation
N = length(x_); % number of meas
U_G_ = NaN(size(x_)); % preallocation
U_FS_ = NaN(size(x_)); % preallocation
U_ = NaN(size(x_)); % preallocation
u_ = NaN(size(x_)); % preallocation

for k = 1:N % k-th meas

    i = find(x_(k)<FS, 1 ); % row in range evaluation
    range(k) = FS(i); %#ok<SAGROW>
    U_G_(k) = U_G(i)/100*abs(x_(k)); % gain unc of k-th meas
    U_FS_(k) = U_FS(i); % FS unc of k-th meas
    U_(k) = U_G_(k)+U_FS_(k); % absolute unc of k-th meas
    u_(k) = U_(k)/x_(k); % relative unc of k-th meas
end

%% Write file
% some magics with matrixs
U_MX = [x_; U_G_; U_FS_; U_; u_]; % crate matrix of results
U_T = U_MX'; % transposed matrix
Table = array2table(U_T,'VariableNames',{'x_', 'U_G_', 'U_FS_', 'U_', 'u_'});% create table
disp(' ')
disp(' ')
disp('***************')
disp('**** Table ****')
disp('***************')
disp(' ');
disp(Table); % print to disp the table
writetable(Table,filename,'Sheet',5); % write!

%% Plot Uncertainty bounds
x = repelem([0;FS],2); % fill the vector with double FS values
x = x(2:end-1)'; % cut the x vector of one to the right and left

Y_U_G_ = repelem(U_G,2)'; % fill the vector with double U_G values

Y_U_FS = repelem(U_FS,2)'; % fill the vector with double U_FS values

y = Y_U_G_.*x/100+Y_U_FS; % calc uncertainty

% create the figure
fig_title = 'Plot Uncertainty bounds of ' + multimeter;
fig = figure;
plot(x,y) % positive part
hold on
plot(x,-y) % negative part
title(fig_title)
hold off
```

```
fig;
% stamp the figure in a file
print(fig,fig_title,'-dpng')
```