plotTDKCharField

Explore TDK TAS2141 characterization field.

Syntax

```
plotTDKCharField()
```

Description

plotTDKCharField() explore characterization field of TDK sensor.

Examples

```
plotTDKCharField();
```

Input Arguments

None

Output Arguments

None

Requirements

- Other m-files: none
- Subfunctions: none
- MAT-files required: data/TDK_TAS2141_Characterization_2020-10-22_18-12-16-827.mat, data/config.mat

See Also

- plotTDKCharDataset

Created on October 28. 2020 by Tobias Wulf. Copyright Tobias Wulf 2020.

```
function plotTDKCharField()
    try
        % load dataset path and dataset content into function workspace
        load('config.mat', 'PathVariables');
        load(PathVariables.tdkDatasetPath, 'Data', 'Info');
%          close all;
    catch ME
        rethrow(ME)
    end

    % load needed data from dataset in to local variables for better handling %%
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % get from user which field to investigate and limits for plateau
    fields = Info.SensorOutput.CosinusBridge.Determination;
    nFields = length(fields);
    fprintf('Choose 1 of %d fields ...\n', nFields);
    for i = 1:nFields
        fprintf('%s\t:\t(%d)\n', fields{i}, i);
    end

    iField = 1; % input('Choice: ');
    field = fields{iField};
```

```
pl = 5; % input('Plateu limit in kA/m: ');

Vcos = Data.SensorOutput.CosinusBridge.(field);
Vsin = Data.SensorOutput.SinusBridge.(field);
gain = Info.SensorOutput.BridgeGain;
HxScale = Data.MagneticField.hx;
HyScale = Data.MagneticField.hy;
Hmin = Info.MagneticField.MinAmplitude;
Hmax = Info.MagneticField.MaxAmplitude;

% get unit strings from
kApm = Info.Units.MagneticFieldStrength;
mV = Info.Units.SensorOutputVoltage;

% get dataset infos and format strings to place in figures
% subtitle string for all figures
infoStr = join([Info.SensorManufacturer, ...
    Info.Sensor, Info.SensorTechnology, ...
    Info.SensorType, "Sensor Characterization Dataset."]);
dateStr = join(["Created on", Info.Created, "by", 'Thorben Sch\"uthe', ...
    "and updated on", Info.Edited, "by", Info.Editor + "."]);

% clear dataset all loaded
clear Data Info;

% figure save path for different formats %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fName = sprintf("tdk_char_field_%s", field);
fPath = fullfile(PathVariables.saveImagesPath, fName);

% define slices and limits to plot %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Hslice = [128 154 180 205]; % hit ca. 0, 5, 10, 15 kA/m
Hlims = [-pl pl];
mVpVlims = [-175 175];

% create figure for plots %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fig = figure('Name', 'Char Field', 'OuterPosition', [0 0 35 30]);
tiledlayout(fig, 2, 2);

% title and description
disp('Info:');
disp([infoStr; dateStr]);
fprintf('Title: TDK Characterization Field - %s\n', field);
disp('Description:');
disp(["a) Cosine Bridge Characteristic"; ...
      "b) Transfer slices for different const. H_y of Vcos"; ...
      "c) Sine Bridge Characteristic"; ...
      "d) Transfer slices for different const. H_x of Vsin"]);

% set colormap
colormap('jet');

% cosinus bridge %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
nexttile(1);
im = imagesc(HxScale, HyScale, Vcos);
set(gca, 'YDir', 'normal');
set(im, 'AlphaData', ~isnan(Vcos));
xticks(-20:10:20);
yticks(-20:10:20);
```

```
    axis square xy;

    % plot lines for slice to investigate
    hold on;
    for i = Hslice
        yline(HyScale(i), 'k:', 'LineWidth', 3.5);
    end
    hold off;

    xlabel(sprintf('$H_x$ in %s', kApm));
    ylabel(sprintf('$H_y$ in %s', kApm));
    title(sprintf('a) $V_{cos}(H_x,H_y)$, Gain $ = %.1f$', gain));

    cb = colorbar;
    cb.Label.String = sprintf('$V_{cos}$ in %s', mV);
    cb.Label.Interpreter = 'latex';
    cb.TickLabelInterpreter = 'latex';
    cb.Label.FontSize = 20;

    % cosinus bridge sclices %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    nexttile(2);
    % slices
    p = plot(HxScale, Vcos(Hslice,:));

    % plateau limits
    if pl > 0
        hold on;
        xline(Hlims(1), 'k-.', 'LineWidth', 2.5);
        xline(Hlims(2), 'k-.', 'LineWidth', 2.5);
        hold off;
    end

    legend(p, {'$H_y \approx 0$ kA/m', ...
               '$H_y \approx 5$ kA/m', ...
               '$H_y \approx 10$ kA/m', ...
               '$H_y \approx 15$ kA/m'},...
            'Location', 'SouthEast');
    xlabel(sprintf('$H_x$ in %s', kApm));
    title('b) $V_{cos}(H_x,H_y)$, $H_y = $ const.');
    ylim(mVpVlims);
    xlim([Hmin Hmax])

    % sinus bridge %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    nexttile(3);
    im = imagesc(HxScale, HyScale, Vsin);
    set(gca, 'YDir', 'normal');
    set(im, 'AlphaData', ~isnan(Vsin));
    xticks(-20:10:20);
    yticks(-20:10:20);
    axis square xy;

    % plot lines for slice to investigate
    hold on;
    for i = Hslice
        xline(HxScale(i), 'k:', 'LineWidth', 3.5);
    end
    hold off;

    xlabel(sprintf('$H_x$ in %s', kApm));
    ylabel(sprintf('$H_y$ in %s', kApm));
    title(sprintf('c) $V_{sin}(H_x,H_y)$, Gain $ = %.1f$', gain));

    cb = colorbar;
    cb.Label.String = sprintf('$V_{sin}$ in %s', mV);
    cb.Label.Interpreter = 'latex';
    cb.TickLabelInterpreter = 'latex';
    cb.Label.FontSize = 20;

    % sinus bridge sclices %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    nexttile(4);
    % slices
    p = plot(HxScale, Vsin(:,Hslice));

    % plateau limits
    if pl > 0
        hold on;
        xline(Hlims(1), 'k-.', 'LineWidth', 2.5);
        xline(Hlims(2), 'k-.', 'LineWidth', 2.5);
        hold off;
    end

    legend(p, {'$H_x \approx 0$ kA/m', ...
               '$H_x \approx 5$ kA/m', ...
               '$H_x \approx 10$ kA/m', ...
               '$H_x \approx 15$ kA/m'},...
            'Location', 'SouthEast');
    xlabel(sprintf('$H_y$ in %s', kApm));
    title('d) $V_{sin}(H_x,H_y)$, $H_x = $ const.');
    ylim(mVpVlims);
    xlim([Hmin Hmax])

    % save results of figure %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%     yesno = input('Save? [y/n]: ', 's');
%     if strcmp(yesno, 'y')
%         savefig(fig, fPath);
%         print(fig, fPath, '-dsvg');
%         print(fig, fPath, '-depsc', '-tiff', '-loose');
%         print(fig, fPath, '-dpdf', '-loose', '-fillpage');
%     end
%     close(fig)
end
```

Published with MATLAB® R2020b