plotTDKCharDataset

Explore TDK TAS2141 characterization dataset and plot its content.

Syntax

```
plotTDKCharDataset()
```

Description

plotTDKCharDataset() explores the dataset and plot its content in three docked figure windows. Loads dataset location from config.mat.

Examples

```
plotTDKCharDataset();
```

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

- plot
- imagesc
- polarplot

Created on October 24. 2020 by Tobias Wulf. Copyright Tobias Wulf 2020.

```
function plotTDKCharDataset()
    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

    % figure save path for different formats
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    fig1Filename = 'tdk_magnetic_stimulus';
    fig1Path = fullfile(PathVariables.saveFiguresPath, fig1Filename);
    fig1SvgPath = fullfile(PathVariables.saveImagesPath, 'svg', fig1Filename);
    fig1EpsPath = fullfile(PathVariables.saveImagesPath, 'eps', fig1Filename);
    fig1PdfPath = fullfile(PathVariables.saveImagesPath, 'pdf', fig1Filename);
```

```
fig2Filename = 'tdk_cosinus_bridge';
fig2Path = fullfile(PathVariables.saveFiguresPath, fig2Filename);
fig2SvgPath = fullfile(PathVariables.saveImagesPath, 'svg', fig2Filename);
fig2EpsPath = fullfile(PathVariables.saveImagesPath, 'eps', fig2Filename);
fig2PdfPath = fullfile(PathVariables.saveImagesPath, 'pdf', fig2Filename);

fig3Filename = 'tdk_sinus_bridge';
fig3Path = fullfile(PathVariables.saveFiguresPath, fig3Filename);
fig3SvgPath = fullfile(PathVariables.saveImagesPath, 'svg', fig3Filename);
fig3EpsPath = fullfile(PathVariables.saveImagesPath, 'eps', fig3Filename);
fig3PdfPath = fullfile(PathVariables.saveImagesPath, 'pdf', fig3Filename);

% load needed data from dataset in to local variables for better handling
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% check if modulation fits to following reconstructioning
if ~strcmp("triang", Info.MagneticField.Modulation)
    error("Modulation function is not triang.");
end
if ~(strcmp("cos", Info.MagneticField.CarrierHx) && ...
        strcmp("sin", Info.MagneticField.CarrierHy))
    error("Carrier functions are not cos or sin.");
end

% modulation frequency
fm = Info.MagneticField.ModulationFrequency;
% carrier frequency
fc = Info.MagneticField.CarrierFrequency;
% max and min amplitude
Hmax = Info.MagneticField.MaxAmplitude;
Hmin = Info.MagneticField.MinAmplitude;
% step range or window size for output picking
Hsteps = Info.MagneticField.Steps;
% resoulution of H steps
Hres = Info.MagneticField.Resolution;
% get unit strings from
kApm = Info.Units.MagneticFieldStrength;
Hz = Info.Units.Frequency;
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 + "."]);

% load characterization data
Vcos = Data.SensorOutput.CosinusBridge;
Vsin = Data.SensorOutput.SinusBridge;
gain = Info.SensorOutput.BridgeGain;

% clear dataset all loaded
clear Data Info;

% reconstruct magnetic stimulus and reduce the view for example plot by 10
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% number of periods reduced by factor 10
reduced = 10;
nPeriods = fc / fm / reduced;
```

```
% number of samples for good looking 40 times nPeriods
nSamples = nPeriods * 400;
% half number of samples
nHalf = round(nSamples / 2);
% generate angle base
phi = linspace(0, nPeriods * 2 * pi, nSamples);
% calculate modulated amplitude, triang returns a column vector, transpose
Hmag = Hmax * triang(nSamples)';
% calculate Hx and Hy stimulus
Hx = Hmag .* cos(phi);
Hy = Hmag .* sin(phi);
% index for rising and falling stimulus
idxR = 1:nHalf;
idxF = nHalf:nSamples;
% find absolute min and max values in bridge outputs for uniform colormap
A = cat(3, Vcos.Rise, Vcos.Fall, Vcos.All, Vcos.Diff, Vsin.Rise, ...
    Vsin.Fall, Vsin.All, Vsin.Diff);
Vmax = max(A, [], 'all');
Vmin = min(A, [], 'all');
clear A;

% figure 1 magnetic stimulus
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fig1 = figure('Name', 'Magnetic Stimulus', ...
    'NumberTitle' , 'off', ...
    'WindowStyle', 'normal', ...
    'MenuBar', 'none', ...
    'ToolBar', 'none', ...
    'Units', 'centimeters', ...
    'OuterPosition', [0 0 30 30], ...
    'PaperType', 'a4', ...
    'PaperUnits', 'centimeters', ...
    'PaperOrientation', 'landscape', ...
    'PaperPositionMode', 'auto', ...
    'DoubleBuffer', 'on', ...
    'RendererMode', 'manual', ...
    'Renderer', 'painters');

tdl = tiledlayout(fig1, 2, 2, ...
    'Padding', 'compact', ...
    'TileSpacing' , 'compact');

title(tdl, 'Reconstructed $H_x$-/ $H_y$-Stimulus in Reduced View', ...
    'FontWeight', 'normal', ...
    'FontSize', 18, ...
    'FontName', 'Times', ...
    'Interpreter', 'latex');

subtitle(tdl, [infoStr; dateStr], ...
    'FontWeight', 'normal', ...
    'FontSize', 14, ...
    'FontName', 'Times', ...
    'Interpreter', 'latex');

xlabel(tdl, sprintf('$\\phi$ in rad, %d periods, reduced by factor %d', ...
    nPeriods*reduced, reduced), ...
    'FontWeight', 'normal', ...
    'FontSize', 16, ...
    'FontName', 'Times', ...
    'Interpreter', 'latex');
```

```
    ylabel(tdl, sprintf('$H_x$, $H_y$, $|H|$ in %s', kApm), ...
        'FontWeight', 'normal', ...
        'FontSize', 16, ...
        'FontName', 'Times', ...
        'Interpreter', 'latex');

    % Hx stimulus
    nexttile;
    p = plot(phi, Hmag, phi, -Hmag, phi(idxR), Hx(idxR), phi(idxF), Hx(idxF));
    set(p, {'Color'}, {'k', 'k', 'b', 'r'}');
    legend([p(1) p(3) p(4)], {'mod', 'rise', 'fall'},...
        'FontWeight', 'normal', ...
        'FontSize', 9, ...
        'FontName', 'Times', ...
        'Interpreter', 'latex', ...
        'Location', 'NorthEast');

    xticks((0:0.25*pi:2*pi) * nPeriods);
    xticklabels({'0', '8\pi', '16\pi', '24\pi', '32\pi', '40\pi', '48\pi', ...
        '56\pi', '64\pi'});
    xlim([0 phi(end)]);
    ylim([Hmin Hmax]);

    xlabel('$\phi$', ...
        'FontWeight', 'normal', ...
        'FontSize', 12, ...
        'FontName', 'Times', ...
        'Interpreter', 'latex');

    ylabel('$H_x(\phi)$', ...
        'FontWeight', 'normal', ...
        'FontSize', 12, ...
        'FontName', 'Times', ...
        'Interpreter', 'latex');

    title(sprintf(...
        'Modulation $f_m = %1.2f$ %s, Cos-Carrier $f_c = %1.2f$ %s', ...
        fm, Hz, fc, Hz), ...
        'FontWeight', 'normal', ...
        'FontSize', 12, ...
        'FontName', 'Times', ...
        'Interpreter', 'latex');

    % Hy stimulus
    nexttile;
    p = plot(phi, Hmag, phi, -Hmag, phi(idxR), Hy(idxR), phi(idxF), Hy(idxF));
    set(p, {'Color'}, {'k', 'k', 'b', 'r'}');
    legend([p(1) p(3) p(4)], {'mod', 'rise', 'fall'},...
        'FontWeight', 'normal', ...
        'FontSize', 9, ...
        'FontName', 'Times', ...
        'Interpreter', 'latex', ...
        'Location', 'NorthEast');

    xticks((0:0.25*pi:2*pi) * nPeriods);
    xticklabels({'0', '8\pi', '16\pi', '24\pi', '32\pi', '40\pi', '48\pi',...
        '56\pi', '64\pi'});
    xlim([0 phi(end)]);
    ylim([Hmin Hmax]);

    xlabel('$\phi$', ...
        'FontWeight', 'normal', ...
```

```
    'FontSize', 12, ...
    'FontName', 'Times', ...
    'Interpreter', 'latex');

ylabel('$H_y(\phi)$', ...
    'FontWeight', 'normal', ...
    'FontSize', 12, ...
    'FontName', 'Times', ...
    'Interpreter', 'latex');

title(sprintf(...
    'Modulation $f_m = %1.2f$ %s, Sin-Carrier $f_c = %1.2f$ %s', ...
    fm, Hz, fc, Hz), ...
    'FontWeight', 'normal', ...
    'FontSize', 12, ...
    'FontName', 'Times', ...
    'Interpreter', 'latex');

% polar for rising modulation
nexttile;
polarplot(phi(idxR), Hmag(idxR), 'b');
title('$|H(\phi)| \cdot e^{-j\phi}$ f. $0 \le \phi \le 32\pi$', ...
    'FontWeight', 'normal', ...
    'FontSize', 12, ...
    'FontName', 'Times', ...
    'Interpreter', 'latex');

% polar for rising modulation
nexttile;
polarplot(phi(idxF), Hmag(idxF), 'r');
title('$|H(\phi)| \cdot e^{-j\phi}$ f. $32\pi \le \phi \le 64\pi$', ...
    'FontWeight', 'normal', ...
    'FontSize', 12, ...
    'FontName', 'Times', ...
    'Interpreter', 'latex');

% figure 2 cosinus bridge outputs
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fig2 = figure('Name', 'Cosinus Bridge', ...
    'NumberTitle' , 'off', ...
    'WindowStyle', 'normal', ...
    'MenuBar', 'none', ...
    'ToolBar', 'none', ...
    'Units', 'centimeters', ...
    'OuterPosition', [0.0 0.0 30.0 30.0], ...
    'PaperType', 'a4', ...
    'PaperUnits', 'centimeters', ...
    'PaperOrientation', 'landscape', ...
    'PaperPositionMode', 'auto', ...
    'DoubleBuffer', 'on', ...
    'RendererMode', 'manual', ...
    'Renderer', 'painters');

tdl = tiledlayout(fig2, 2, 2, ...
    'Padding', 'normal', ...
    'TileSpacing' , 'compact');

title(tdl, ...
    'Measured Cosinus Bridge Outputs of Corresponding $H_x$-/ $H_y$-Amplitudes', ...
    'FontWeight', 'normal', ...
    'FontSize', 18, ...
```

```
        'FontName', 'Times', ...
        'Interpreter', 'latex');

    subtitle(tdl, [infoStr; dateStr], ...
        'FontWeight', 'normal', ...
        'FontSize', 14, ...
        'FontName', 'Times', ...
        'Interpreter', 'latex');

    xlabel(tdl, sprintf('$H_x$, $H_y$ in %s, %d Steps in %.4f %s', ...
        kApm, Hsteps, Hres, kApm), ...
        'FontWeight', 'normal', ...
        'FontSize', 16, ...
        'FontName', 'Times', ...
        'Interpreter', 'latex');

    colormap('jet');

    % cosinus bridge recorded during rising stimulus
    nexttile;
    im = imagesc([Hmin Hmax], [Hmin Hmax], Vcos.Rise);
    set(gca, 'YDir', 'normal');
    set(im, 'AlphaData', ~isnan(Vcos.Rise));
    caxis([Vmin, Vmax]);
    xlim([Hmin Hmax]);
    ylim([Hmin Hmax]);
    yticks(xticks);
    axis square xy;
    grid on;

    xlabel('$H_x$', ...
        'FontWeight', 'normal', ...
        'FontSize', 12, ...
        'FontName', 'Times', ...
        'Interpreter', 'latex');

    ylabel('$H_y$', ...
        'FontWeight', 'normal', ...
        'FontSize', 12, ...
        'FontName', 'Times', ...
        'Interpreter', 'latex');

    title('Rising $H$-Amplitudes', ...
        'FontWeight', 'normal', ...
        'FontSize', 12, ...
        'FontName', 'Times', ...
        'Interpreter', 'latex');

    % cosinus bridge recorded during falling stimulus
    nexttile;
    im = imagesc([Hmin Hmax], [Hmin Hmax], Vcos.Fall);
    set(gca, 'YDir', 'normal');
    set(im, 'AlphaData', ~isnan(Vcos.Fall));
    caxis([Vmin, Vmax]);
    xlim([Hmin Hmax]);
    ylim([Hmin Hmax]);
    yticks(xticks);
    axis square xy;
    grid on;

    xlabel('$H_x$', ...
        'FontWeight', 'normal', ...
```

```
    'FontSize', 12, ...
    'FontName', 'Times', ...
    'Interpreter', 'latex');

ylabel('$H_y$', ...
    'FontWeight', 'normal', ...
    'FontSize', 12, ...
    'FontName', 'Times', ...
    'Interpreter', 'latex');

title('Falling $H$-Amplitudes', ...
    'FontWeight', 'normal', ...
    'FontSize', 12, ...
    'FontName', 'Times', ...
    'Interpreter', 'latex');

% cosinus bridge recorded during superimposed stimulus
nexttile;
im = imagesc([Hmin Hmax], [Hmin Hmax], Vcos.All);
set(gca, 'YDir', 'normal');
set(im, 'AlphaData', ~(~Vcos.All));
caxis([Vmin, Vmax]);
xlim([Hmin Hmax]);
ylim([Hmin Hmax]);
yticks(xticks);
axis square xy;
grid on;

xlabel('$H_x$', ...
    'FontWeight', 'normal', ...
    'FontSize', 12, ...
    'FontName', 'Times', ...
    'Interpreter', 'latex');

ylabel('$H_y$', ...
    'FontWeight', 'normal', ...
    'FontSize', 12, ...
    'FontName', 'Times', ...
    'Interpreter', 'latex');

title('Superimposed $H$-Amplitudes', ...
    'FontWeight', 'normal', ...
    'FontSize', 12, ...
    'FontName', 'Times', ...
    'Interpreter', 'latex');

% cosinus bridge recorded during differentiated stimulus
nexttile;
im = imagesc([Hmin Hmax], [Hmin Hmax], Vcos.Diff);
set(gca, 'YDir', 'normal');
set(im, 'AlphaData', ~isnan(Vcos.Diff));
caxis([Vmin, Vmax]);
xlim([Hmin Hmax]);
ylim([Hmin Hmax]);
yticks(xticks);
axis square xy;
grid on;

xlabel('$H_x$', ...
    'FontWeight', 'normal', ...
    'FontSize', 12, ...
    'FontName', 'Times', ...
```

```
    'Interpreter', 'latex');

ylabel('$H_y$', ...
    'FontWeight', 'normal', ...
    'FontSize', 12, ...
    'FontName', 'Times', ...
    'Interpreter', 'latex');

title('Differentiated $H$-Amplitudes', ...
    'FontWeight', 'normal', ...
    'FontSize', 12, ...
    'FontName', 'Times', ...
    'Interpreter', 'latex');

% add colorbar and place it overall plots
cb = colorbar;
cb.Layout.Tile = 'east';
cb.Label.String = sprintf(...
    '$V_{cos}(H_x, H_y)$ in %s, Gain $ = %.1f$', mV, gain);
cb.Label.Interpreter = 'latex';
cb.Label.FontSize = 16;

% figure 3 sinus bridge outputs
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fig3 = figure('Name', 'Sinus Bridge', ...
    'NumberTitle' , 'off', ...
    'WindowStyle', 'normal', ...
    'MenuBar', 'none', ...
    'ToolBar', 'none', ...
    'Units', 'centimeters', ...
    'OuterPosition', [0.0 0.0 30.0 30.0], ...
    'PaperType', 'a4', ...
    'PaperUnits', 'centimeters', ...
    'PaperOrientation', 'landscape', ...
    'PaperPositionMode', 'auto', ...
    'DoubleBuffer', 'on', ...
    'RendererMode', 'manual', ...
    'Renderer', 'painters');

tdl = tiledlayout(fig3, 2, 2, ...
    'Padding', 'normal', ...
    'TileSpacing' , 'compact');

title(tdl,...
    'Measured Sinus Bridge Outputs of Corresponding $H_x$-/ $H_y$-Amplitudes', ...
    'FontWeight', 'normal', ...
    'FontSize', 18, ...
    'FontName', 'Times', ...
    'Interpreter', 'latex');

subtitle(tdl, [infoStr; dateStr], ...
    'FontWeight', 'normal', ...
    'FontSize', 14, ...
    'FontName', 'Times', ...
    'Interpreter', 'latex');

xlabel(tdl, sprintf('$H_x$, $H_y$ in %s, %d Steps in %.4f %s', ...
    kApm, Hsteps, Hres, kApm), ...
    'FontWeight', 'normal', ...
    'FontSize', 16, ...
    'FontName', 'Times', ...
```

```
    'Interpreter', 'latex');

colormap('jet');

% sinus bridge recorded during rising stimulus
nexttile;
im = imagesc([Hmin Hmax], [Hmin Hmax], Vsin.Rise);
set(gca, 'YDir', 'normal');
set(im, 'AlphaData', ~isnan(Vsin.Rise));
caxis([Vmin, Vmax]);
xlim([Hmin Hmax]);
ylim([Hmin Hmax]);
yticks(xticks);
axis square xy;
grid on;

xlabel('$H_x$', ...
    'FontWeight', 'normal', ...
    'FontSize', 12, ...
    'FontName', 'Times', ...
    'Interpreter', 'latex');

ylabel('$H_y$', ...
    'FontWeight', 'normal', ...
    'FontSize', 12, ...
    'FontName', 'Times', ...
    'Interpreter', 'latex');

title('Rising $H$-Amplitudes', ...
    'FontWeight', 'normal', ...
    'FontSize', 12, ...
    'FontName', 'Times', ...
    'Interpreter', 'latex');

% sinus bridge recorded during falling stimulus
nexttile;
im = imagesc([Hmin Hmax], [Hmin Hmax], Vsin.Fall);
set(gca, 'YDir', 'normal');
set(im, 'AlphaData', ~isnan(Vsin.Fall));
caxis([Vmin, Vmax]);
xlim([Hmin Hmax]);
ylim([Hmin Hmax]);
yticks(xticks);
axis square xy;
grid on;

xlabel('$H_x$', ...
    'FontWeight', 'normal', ...
    'FontSize', 12, ...
    'FontName', 'Times', ...
    'Interpreter', 'latex');

ylabel('$H_y$', ...
    'FontWeight', 'normal', ...
    'FontSize', 12, ...
    'FontName', 'Times', ...
    'Interpreter', 'latex');

title('Falling $H$-Amplitudes', ...
    'FontWeight', 'normal', ...
    'FontSize', 12, ...
    'FontName', 'Times', ...
```

```
    'Interpreter', 'latex');

% sinus bridge recorded during superimposed stimulus
nexttile;
im = imagesc([Hmin Hmax], [Hmin Hmax], Vsin.All);
set(gca, 'YDir', 'normal');
set(im, 'AlphaData', ~(~Vsin.All));
caxis([Vmin, Vmax]);
xlim([Hmin Hmax]);
ylim([Hmin Hmax]);
yticks(xticks);
axis square xy;
grid on;

xlabel('$H_x$', ...
    'FontWeight', 'normal', ...
    'FontSize', 12, ...
    'FontName', 'Times', ...
    'Interpreter', 'latex');

ylabel('$H_y$', ...
    'FontWeight', 'normal', ...
    'FontSize', 12, ...
    'FontName', 'Times', ...
    'Interpreter', 'latex');

title('Superimposed $H$-Amplitudes', ...
    'FontWeight', 'normal', ...
    'FontSize', 12, ...
    'FontName', 'Times', ...
    'Interpreter', 'latex');

% sinus bridge recorded during differentiated stimulus
nexttile;
im = imagesc([Hmin Hmax], [Hmin Hmax], Vsin.Diff);
set(gca, 'YDir', 'normal');
set(im, 'AlphaData', ~isnan(Vsin.Diff));
caxis([Vmin, Vmax]);
xlim([Hmin Hmax]);
ylim([Hmin Hmax]);
yticks(xticks);
axis square xy;
grid on;

xlabel('$H_x$', ...
    'FontWeight', 'normal', ...
    'FontSize', 12, ...
    'FontName', 'Times', ...
    'Interpreter', 'latex');

ylabel('$H_y$', ...
    'FontWeight', 'normal', ...
    'FontSize', 12, ...
    'FontName', 'Times', ...
    'Interpreter', 'latex');

title('Differentiated $H$-Amplitudes', ...
    'FontWeight', 'normal', ...
    'FontSize', 12, ...
    'FontName', 'Times', ...
    'Interpreter', 'latex');
```

```
    % add colorbar and place it overall plots
    cb = colorbar;
    cb.Layout.Tile = 'east';
    cb.Label.String = sprintf(...
        '$V_{sin}(H_x, H_y)$ in %s, Gain $ = %.1f$', mV, gain);
    cb.Label.Interpreter = 'latex';
    cb.Label.FontSize = 16;

    yesno = input('Save? [y/n]: ', 's');
    if strcmp(yesno, 'y')
        % save results of figure 1
        savefig(fig1, fig1Path);
        print(fig1, fig1SvgPath, '-dsvg');
        print(fig1, fig1EpsPath, '-depsc', '-tiff', '-loose');
        print(fig1, fig1PdfPath, '-dpdf', '-loose', '-fillpage');

        % save results of figure 2
        savefig(fig2, fig2Path);
        print(fig2, fig2SvgPath, '-dsvg');
        print(fig2, fig2EpsPath, '-depsc', '-tiff', '-loose');
        print(fig2, fig2PdfPath, '-dpdf', '-loose', '-fillpage');

        % save results of figure 3
        savefig(fig3, fig3Path);
        print(fig3, fig3SvgPath, '-dsvg');
        print(fig3, fig3EpsPath, '-depsc', '-tiff', '-loose');
        print(fig3, fig3PdfPath, '-dpdf', '-loose', '-fillpage');
    end
    close(fig1)
    close(fig2)
    close(fig3)
end
```

Published with MATLAB® R2020b