

Einführung in die Modellierung

Übung 5

Function Handle

1. Schreiben Sie eine Funktion *mult*, die die Multiplikation zweier als Function Handle übergebene Funktionen im gewählten Definitionsbereich berechnet und die beiden Funktionen mit deren Multiplikation in einem gemeinsamen Diagramm grafisch darstellt.

```
function mult( f1,f2,x1,x2 )
```

Eingabeparameter: $f1, f2$ – die zu multiplizierenden Funktionen
 $x1, x2$ – Anfang und Ende des Definitionsbereichs

Testen Sie Ihre Funktion für einige anonyme und vordefinierte Funktionen.

LÖSUNG:

```
function mult( f1,f2,x1,x2 )
%UNTITLED2 Summary of this function goes here
% Detailed explanation goes here
x = linspace(x1,x2,1000);
mult = f1(x).*f2(x);

plot(x,f1(x))
grid on
hold on
plot(x,f2(x))
plot(x,mult)
legend('f1','f2','f1*f2')
title('Multiplikation zweier Funktionen')
end

% Aufruf
% mult(@(x) (3*x+2), @(x) (x.^2-x-10),-3.5,3.5)
```

2. Gegeben ist eine Funktion *interpolation_messreihe*, die mit Hilfe der Interpolation zwei an sie übergebenen Messreihen durch einen analytischen Ausdruck näherungsweise möglichst gut beschreibt.

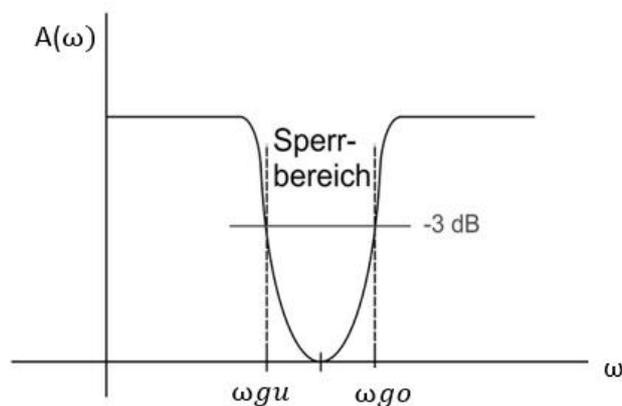
```
function [ p ] = interpolation_messreihe( x,y,n )
```

x, y – die gemessenen Messreihen

n – Ordnung des Interpolationspolynoms

p – Koeffizienten des Interpolationspolynoms

- Laden Sie die Datei *bandsperre.xlsx*. Trennen und speichern Sie die beiden Messreihen: Kreisfrequenz ω (erste Spalte) und Amplitudengang der Bandsperre A (zweite Spalte).
- Interpolieren Sie die aufgenommenen Messwerte mit Hilfe der Funktion *interpolation_messreihe* für unterschiedliche Werte der Polynomordnung.
- Ermitteln Sie die beiden -3 dB Grenzfrequenzen ω_{gu} und ω_{go} , für die die Amplitude A auf 70% des maximalen Wertes abgeklungen ist.



Der Befehl **[a,b] = math.ausdr. (vektor)** gibt den gesuchten Wert und Position in einem Vektor.

- Finden Sie mit Hilfe einer Schleife die erforderliche Ordnung des Interpolationspolynoms damit die Abweichung der Grenzfrequenzen ω_{gu} und ω_{go} der aufgenommenen Messreihen und der Näherung weniger als 1% beträgt.
- Stellen Sie die Messreihe (punktweise, nicht interpoliert) und die Näherungsfunktion grafisch dar.

LÖSUNG:

```
clear all
close all
clc
```

```
messwerte = xlsread('bandsperre.xlsx'); % laden der Messreihen
% Trennen der x und y Messreihen
x = messwerte(:,1);
y = messwerte(:,2);
% Anfangsbedingungen
n = 1; % Grad des Polynoms
```

```

delta = 1; % Abweichung

% fgu und fgo ermitteln

grenze_3db = 0.707; % 3 dB Grenze - 70% Amplitudenabfall
[ymin,xmin] = min(y);% der tiefster Punkt des Amplitudengangs

delta_3db = abs(y - grenze_3db); % Abweichung der y-Werte vom 3 dB Grenze
[delta_min,fgu] = min(delta_3db(1:xmin));% fgu
[delta_min,fgo] = min(delta_3db((xmin+1):length(x))); % fgo

% Berechnung der Polynomwerte
while delta > 0.01 % solange delta > als 1%
    p = interpolation_messreihe(x,y,n);% Aufruf der Interpolationsfunktion
    yp = polyval(p,x);% Polynomwerte für x berechnen
    delta = max([y(fgu)-yp(fgu), y(fgo)-yp(fgo)]); % Abweichung der
    Grenzfrequenzen berechnen
    n = n+1; % Polynomgrad erhöhen
end

% grafische Darstellung der Messreihen und Interpolationspolynoms
xp = linspace(x(1),x(end),1000);
polynom = polyval(p,xp);

figure
semilogx(xp,polynom)
hold on
semilogx(x,y,'*');
xlabel('Kreisfrequenz w, 1/s');
ylabel('Amplitudengang A(w)');
legend('Näherungsfunktion','Messwerte')
grid on

```

```

function [ p ] = interpolation_messreihe( x,y,n )
%interpoliert zwei aufgenommene Messreihen durch ein Polynom

if nargin == 3 | nargin == 2 % die Anzahl der Eingabeparameter prüfen
    if nargin == 2
        if length(x)==length(y)
            n = length(x)-1; % Ordnung des Polynoms bei 2 Eingabeparameter
        else error('Länge der Messreihen stimmen nicht überein'); end
    end

    p = polyfit(x,y,n); % Polynomkoeffiziente ermitteln

else error('Falsche Anzahl der Eingabeparameter. Eingabeparameter:
Messreihen: x,y und evtl. Ordnung n');
end
end

```

