

SV0: Rechnerpraktikum

Abgabe: bis zum 04.01.2017

Abzugeben: harm.m, aufgabe2.m, aufgabe3.m, swal.m, swalsys.m, aufgabe6.m, aufgabe7.m in einem Archiv

Orthogonale Funktionssysteme

Berechnung der Walshfunktion (k-Ordnung) in Sequenzordnung nach dem Algorithmus aus SWICK:

1. Darstellung der Zahl als Binärzahl, z.B. $k = 11_D = 1011_B$ mit der Stellennummerierung $k_3 k_2 k_1 k_0$

2. Start mit dem Grundelement: $[1]$

3. Mit dem höchsten Index von k beginnend, wird der Vektor wie folgt erweitert:

falls $k_i = 0$: symmetrische Erweiterung

falls $k_i = 1$: zentralsymmetrische Erweiterung

Beispiel: $k = 11_D = 1011_B$

Start: $[1]$

$k_3 = 1$: $[1 \quad -1]$

$k_2 = 0$: $[1 \quad -1 \quad -1 \quad 1]$

$k_1 = 1$: $[1 \quad -1 \quad -1 \quad 1 \quad -1 \quad 1 \quad 1 \quad -1]$

$k_0 = 1$: $[1 \quad -1 \quad -1 \quad 1 \quad -1 \quad 1 \quad 1 \quad -1 \quad 1 \quad -1 \quad -1 \quad 1 \quad -1 \quad 1 \quad 1 \quad -1]$

1. Programmieren Sie die MATLAB-Funktion: function y = harm(k,t).

Die Funktion soll die zeitdiskreten Exponentialfunktionen $e^{(j \cdot k \cdot 2\pi \cdot t)}$ berechnen. k und t dürfen Skalare oder Zeilenvektoren sein. Eine Überprüfung auf korrekte Parameterübergabe muss nicht erfolgen. Das Ergebnis ist eine (k, t) -Matrix.

Eingabebeispiele:

harm(0,0) $\rightarrow 1$

harm(0,[0:0.1:0.9]) $\rightarrow [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]$

harm([0 1], 0) $\rightarrow [1; 1]$

Überprüfen Sie die Funktion mit den gegebenen und selbst gewählten Beispielen.

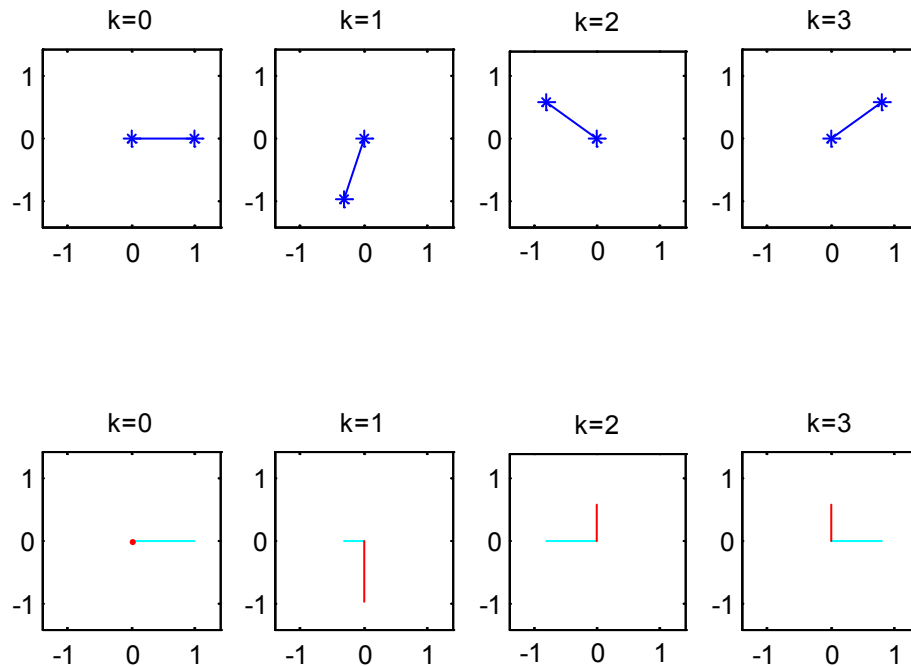
>> help pi

1 Punkt

2. Programmieren Sie das MATLAB-Skript: aufgabe2.m

Das Ergebnis der in Aufgabe 1 programmierten Funktion harm soll als Animation dargestellt werden. Hierfür soll die Ausgabe für $k=[0:3]$ in vier horizontal nebeneinander liegenden Subplots in Zeigerdarstellung (kartesisches Koordinatensystem) erfolgen. Darunter soll das Ergebnis getrennt nach Realteil (x-Achse) und Imaginärteil (y-Achse) in 4 weiteren Subplots dargestellt werden. Nutzen sie für den Real- und Imaginärteil verschiedene Farben (siehe Abbildung ↓). Für t ist der Vektor $t=[0:0.01:0.99]$ zu wählen. Nach der Darstellung der jeweiligen Teilergebnisse für jedes t_n soll eine kurze Pause erfolgen.

>> help imag, pause, real, xlim, ylim



4 Punkte

3. Programmieren Sie das MATLAB-Skript: `aufgabe3.m`
Überprüfen Sie, ob die beiden Funktionen $e^{(j \cdot k \cdot 2\pi \cdot t)}$ und $e^{(-j \cdot m \cdot 2\pi \cdot t)}$ für $t_n = [0 : 1/512 : 1 - 1/512]$ und $k = m = [0 : 9]$ orthogonal sind. Berechnen Sie hierfür die folgende Matrix

$$O_e(k, m) = \sum \left(e^{(j \cdot k \cdot 2\pi \cdot t_n)} \cdot e^{(-j \cdot m \cdot 2\pi \cdot t_n)} \right)$$
Orthogonalität liegt vor, wenn alle Elemente, die nicht zur Hauptdiagonale gehören, 0 sind. Geben Sie die errechnete Matrix auf der Konsole aus. Außerdem soll in Satzform auf der Konsole mitgeteilt werden, ob die Funktionen orthogonal sind.

2 Punkte

4. Programmieren Sie die MATLAB-Funktion: `function y = swal(k)`.
Die Funktion soll die zeitdiskrete Walshfunktion in Sequenzordnung nach dem Algorithmus aus SWICK (siehe oben) berechnen. k ist eine natürliche Zahl. Eine Überprüfung muss nicht erfolgen. Das Ergebnis ist ein Vektor.
Beispiele: `swal(3) → [1 -1 1 -1]` und `swal(11) → [1, -1, -1, 1, -1, 1, 1, -1, 1, -1, -1, 1, -1, 1, 1, -1]`

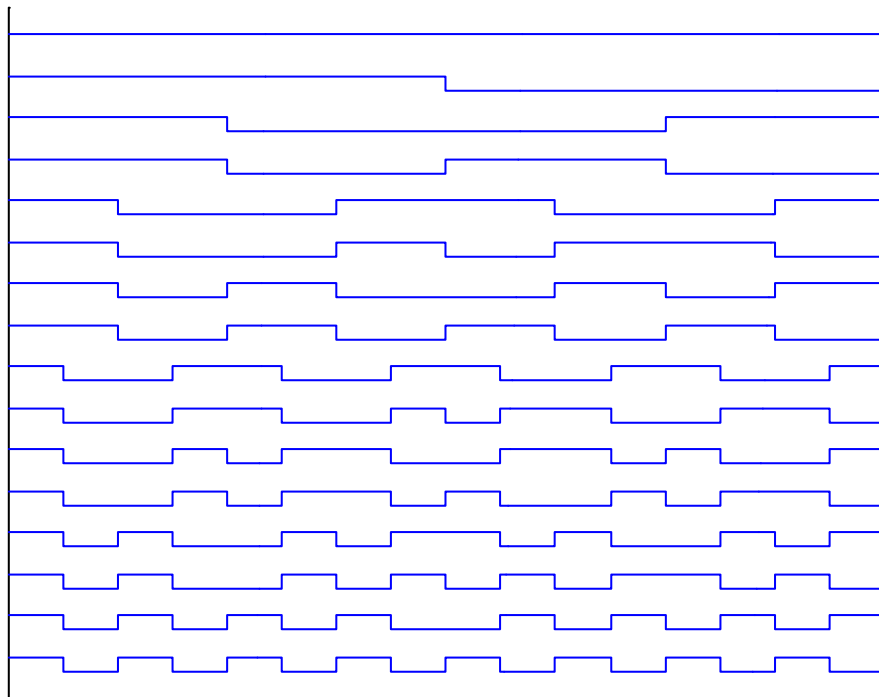
`>> help dec2bin, fliplr`

4 Punkte

5. Programmieren Sie die MATLAB-Funktion: `function y = swalsys(k)`.
Die Funktion soll die zeitdiskreten Walshfunktionen in Sequenzordnung unter Nutzung der in Aufgabe 4 programmierten Funktion berechnen. Dabei gilt $k = 2^n$ mit $n \in \mathbb{N}$. Falls diese Bedingung nicht erfüllt ist, soll eine entsprechende Fehlermeldung ausgegeben werden. Das Ergebnis ist eine quadratische (k, k) -Matrix, welche alle Walshfunktionen `swal(n)` mit $n < k$ enthält. Beachten Sie, dass alle Walshfunktionen k Funktionswerte besitzen sollen. Eine Modifikation der Funktion `swal` ist hierbei erlaubt, sofern die geforderte Funktionalität der vierten Aufgabe erhalten bleibt.

4 Punkte

6. Programmieren Sie das MATLAB-Skript: aufgabe6.m
Berechnen Sie `swalsys(16)` und geben Sie das Ergebnis auf der Konsole aus. Stellen Sie das Ergebnis zusätzlich grafisch dar.



```
>> help stairs
```

3 Punkte

7. Programmieren Sie das MATLAB-Skript: aufgabe7.m
Überprüfen Sie, ob die beiden Funktionen $swal(k)$ und $swal(m)$ für $k = m = [0:15]$ orthogonal sind. Berechnen Sie hierfür die folgende Matrix $O_w(k, m) = \sum (swal(k) \cdot swal(m))$.
Orthogonalität liegt vor, wenn alle Elemente, die nicht zur Hauptdiagonale gehören, 0 sind. Geben Sie die errechnete Matrix auf der Konsole aus. Außerdem soll in Satzform auf der Konsole mitgeteilt werden, ob die Funktionen orthogonal sind. Achten Sie darauf, dass alle `swal`-Funktionen die gleiche Anzahl von Funktionswerten ($2^4 = 16$) besitzen.

2 Punkte