



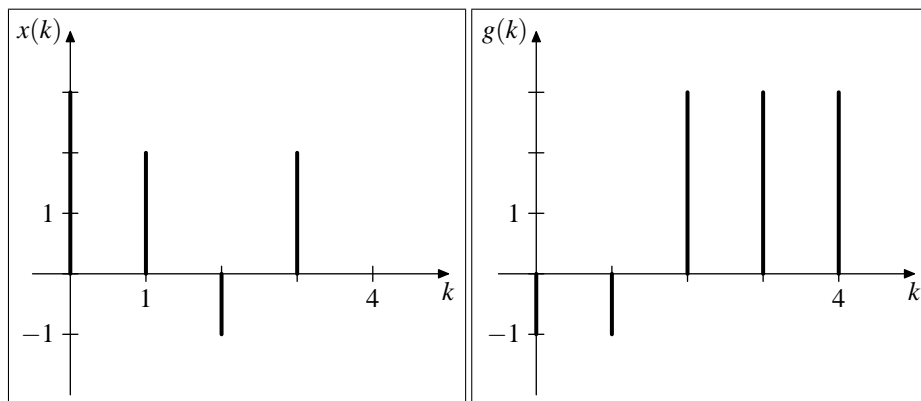
Aufgabe

Skizzieren Sie die gegebenen, zeitdiskreten Energiesignale $x(k)$ und $g(k)$ und berechnen und skizzieren Sie

1. Autokorrelation $s_{XX}(\tau)$ und $s_{GG}(\tau)$,
2. Kreuzkorrelation $s_{GX}(\tau)$,
3. Faltungssignal $y(k) = g(k) * x(k)$!

k	< 0	0	1	2	3	4	≥ 5
x	0	3	2	-1	2	0	0
g	0	-1	-1	3	3	3	0

Energiesignale



1 Autokorrelation

Ich habe für dieses Problem ein C-Programm geschrieben, dessen Quelltext in Abbildung 5 zu sehen ist. Dieses Programm liest zuerst die Anzahl der Funktionswerte und die Funktionswerte selbst ein. Daraufhin wird die Korrelation der eingegebenen Werte nach folgender Gleichung bestimmt.

$$s_{XX}(\tau) = \frac{1}{N} \sum_{k=0}^N x(k) \cdot x(k - \tau) \quad (1)$$

Danach werden die errechneten Werte im TeX-Format in die Datei `sxx.txt` geschrieben. Nach zwei Durchläufen mit s_{XX} und s_{GG} entsteht Tabelle 1. Die grafische Veran-

τ	-4	-3	-2	-1	0	1	2	3	4
$s_{XX}(\tau)$	0	6/5	1/5	2/5	18/5	2/5	1/5	6/5	0
$s_{GG}(\tau)$	-3/5	-6/5	3/5	16/5	29/5	16/5	3/5	-6/5	-3/5

Tabelle 1: Autokorrelation von s_{XX} und s_{GG}

schauung in den Abbildungen 1 und 2 zeigt die Axialsymmetrie der Autokorrelationsfunktion. Außerdem erkennt man, daß bei $\tau = 0$ ein Extremwert erreicht wird. Die Signale sind ja bei Überdeckung am „selbstähnlichsten“, d.h. identisch.

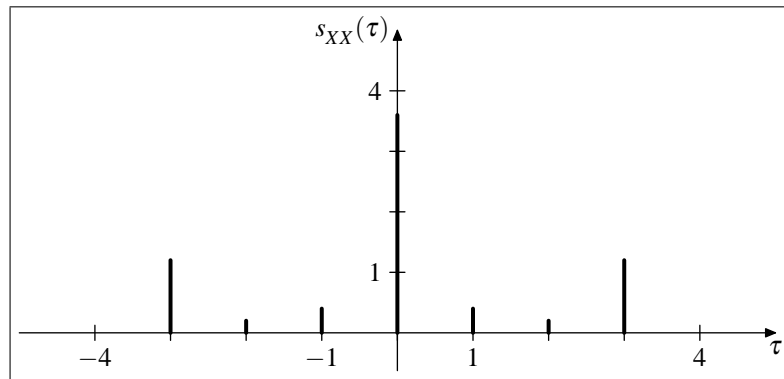


Abbildung 1: Diagramm zur Autokorrelation von s_{XX}

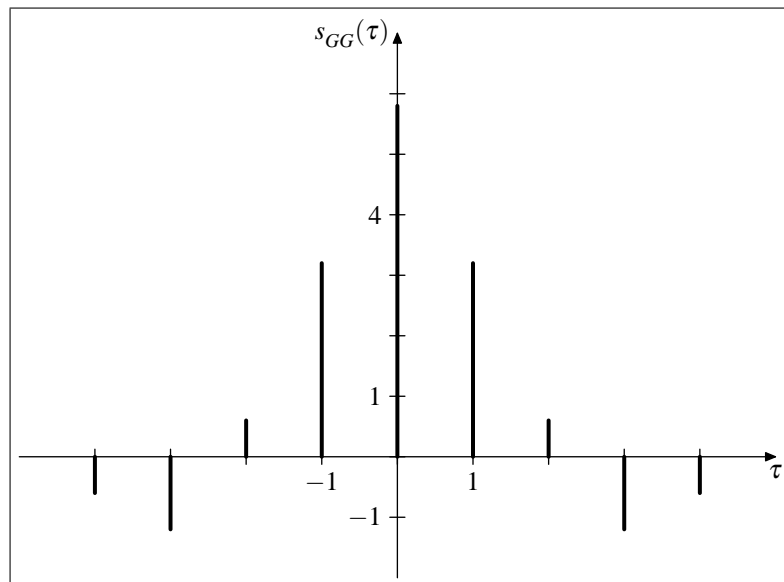
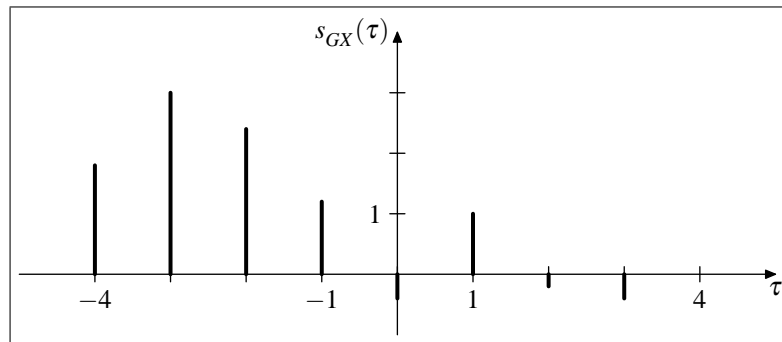


Abbildung 2: Diagramm zur Autokorrelation von s_{GG}

τ	-4	-3	-2	-1	0	1	2	3	4
$s_{GX}(\tau)$	0	$-2/5$	$-1/5$	$5/5$	$-2/5$	$6/5$	$12/5$	$15/5$	$9/5$

Tabelle 2: Kreuzkorrelation s_{GX}

Abbildung 3: Diagramm zur Kreuzkorrelation von s_{GX}

2 Kreuzkorrelation

Auch in diesem Fall bietet sich ein kleines Computerprogramm an. Der Quelltext hierfür ist in Abbildung 6 zu sehen. Es werden die Funktionswerte zweier Funktionen eingelesen und danach durch die folgende Gleichung die Kreuzkorrelation bestimmt.

$$s_{GX}(\tau) = \frac{1}{N} \sum_{k=0}^N g(k) \cdot x(k - \tau) \quad (2)$$

Danach werden die errechneten Werte im \TeX -Format in die Datei `sgx.txt` geschrieben. Dadurch entsteht Tabelle 2. In Abbildung 3 ist die Kreuzkorrelation der beiden Signale dargestellt.

3 Faltung

Das dritte Programm (siehe Abbildung 7) berechnet das Faltungssignal zweier Eingangssignale. Die Werte ergeben sich wie folgt

$$y(k) = \sum_{i=-\infty}^{\infty} g(i) \cdot x(k - i) =: g(k) * x(k) \quad (3)$$

Danach werden die errechneten Werte im \TeX -Format in die Datei `y.txt` geschrieben. Dadurch entsteht die Tabelle 3. Das Ergebnis der Faltung ist in Abbildung 4 dargestellt.

k	0	1	2	3	4	5	6	7
$y(k)$	-3	-5	8	14	10	9	3	6

Tabelle 3: Faltung $g(k) * x(k)$

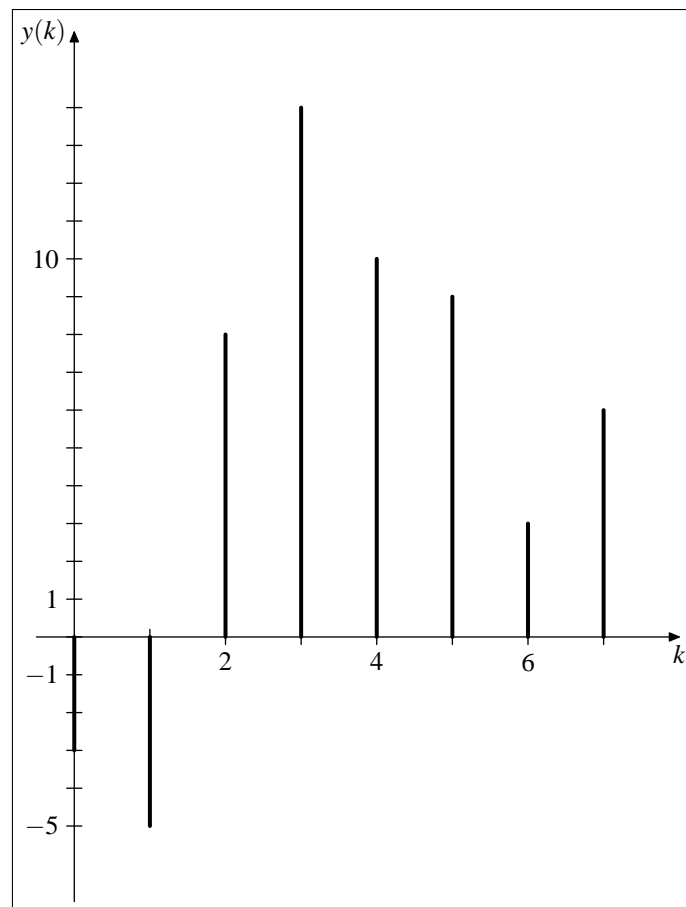


Abbildung 4: Diagramm zur Faltung von $g(k) * x(k)$

```
#include <stdio.h>

int
main (int argc, char **argv)
{
    int k, N, tau, x[20], g[20], sgx[39];
    FILE *fsgx = fopen("sgx.txt", "wt");

    printf("Anzahl der Funktionswerte: ");
    scanf("%d", &N);
    for (k = 0; k < N; k++)
    {
        printf("g(%d) = ", k);
        scanf("%d", &g[k]);
    }
    for (k = 0; k < N; k++)
    {
        printf("x(%d) = ", k);
        scanf("%d", &x[k]);
    }
    for (tau = -N + 1; tau < N; tau++)
    {
        sgx[tau + N - 1] = 0;
        for (k = 0; k < N; k++)
        {
            if ((k - tau < 0) || (k - tau > N - 1))
            {
                continue;
            }
            sgx[tau + N - 1] += g[k] * x[k - tau];
        }
        fprintf(fsgx, "$\\tau$");
        for (tau = -N + 1; tau < N; tau++)
        {
            fprintf(fsgx, "&$%d$", tau);
        }
        fprintf(fsgx, "\\ \\hline\\n$_{GX}$ (\\tau)$");
        for (tau = -N + 1; tau < N; tau++)
        {
            fprintf(fsgx, "&$%d/%d$", sgx[tau + N - 1], N);
        }
        return 0;
    }
}
```

Abbildung 6: C-Quelltext zur Kreuzkorrelation

```
#include <stdio.h>

int
main (int argc, char **argv)
{
    int i, N, k, g[20], x[20], y[38];
    FILE *fy = fopen("y.txt", "wt");

    printf("Anzahl der Funktionswerte: ");
    scanf("%d", &N);
    for (i = 0; i < N; i++)
    {
        printf("g(%d) = ", i);
        scanf("%d", &g[i]);
    }
    for (i = 0; i < N; i++)
    {
        printf("x(%d) = ", i);
        scanf("%d", &x[i]);
    }
    for (k = 0; k <= 2 * (N - 1); k++)
    {
        y[k] = 0;
        for (i = 0; i <= k; i++)
        {
            if (k - i >= N) continue;
            y[k] += g[i] * x[k - i];
        }
    }
    fprintf(fy, "%d\n", y[k]);
    for (k = 0; k <= 2 * (N - 1); k++)
    {
        fprintf(fy, "%d\n", y[k]);
    }
    printf("y(k) = ");
    for (k=0; k<=2*(N-1); k++)
    {
        printf("%d\n", y[k]);
    }
    return 0;
}
```

Abbildung 7: C-Quelltext zur Faltung