PAM-signaler i MATLAB

Dette dokument beskriver brugen af foldningsfunktionen i MATLAB som en metode til at generere et PAM-signal, også kaldet et "pulstog". Et PAM-signal i kontinuert tid kan helt generelt defineres via formel[1] (1)

$$v(t) = \sum_{k=0}^{\infty} a_k \cdot g(t - k \cdot T) \tag{1}$$

hvor

- a_k er informationssymboler ($k = 0, 1, 2, \ldots$)
- $g(t)$ er pulsformen
- T er symbolperioden, dvs. tidsintervallet mellem to a_k-symboler skal sendes.

I digitale kommunikationssystemer er vi dog kun interesseret i signalerne til diskrete tidspunkter, dvs. de analoge signaler er blevet digitaliseret (dvs. samplet og kvantiseret) og vi er derfor kun interesseret i værdierne til tidspunkterne $t = n \cdot T_s$, hvor T_s er samplingsintervallet og $n = 0, 1, 2, \ldots$. En tidsdiskret udgave af ligning (1) fås derfor ved at erstatte t med $n \cdot T_s$ i ligning (1):

$$v(n \cdot T_s) = \sum_{k=0}^{\infty} a_k \cdot g(n \cdot T_s - k \cdot T) \tag{2}$$

I praksis vil samplingsintervallet, T_s, være valgt så $T = m \cdot T_s$ ($m \in \mathbb{N}$) for en passende værdi af m. Herved kan ligning (2) omskrives

$$\begin{aligned} v(n \cdot T_s) &= \sum_{k=0}^{\infty} a_k \cdot g(n \cdot T_s - k \cdot m \cdot T_s) \quad &(3)\\ &= \sum_{k=0}^{\infty} a_k \cdot g((n - k \cdot m) \cdot T_s) \quad &(4) \end{aligned}$$

Indføres derefter notationen x_n som en kortere måde at skrive $x(n \cdot T_s)$ på fås:

$$v_n = \sum_{k=0}^{\infty} a_k \cdot g_{n-k\cdot m} \tag{5}$$

Hvis man opskriver summens led fås

$$v_n = a_0 \cdot g_n + a_1 \cdot g_{n-m} + a_2 \cdot g_{n-2m} + \ldots \tag{6}$$

[1] Jf. også formel (5.1) i Knud J. Larsen, "Introduktion til digital kommunikation", 2017, hvor der dog er tale om, at informationssymbolerne, a_k er defineret for alle k-værdier. I denne note vil a_k dog kun være relevante for $k \geq 0$.

Inden for matematikken defineres *foldningen* af to tidsdiskrete sekvenser som vist i ligning (7) (hvoraf den ene sekvens blot kan være den samme som g_k-sekvensen ovenfor)

$$y_n = \sum_{l=0}^{\infty} b_l \cdot g_{n-l} \tag{7}$$

Det ses, at ligningerne (5) og (7) har næsten samme form, og da MATLAB har en indbygget funktion, kaldet `conv()`, der implementerer[2] ligning (7), er det nærliggende at forsøge at omskrive ligning (5), så den har samme form som ligning (7), så et PAM-signal i MATLAB kan genereres via MATLAB's `conv()`-funktion.

Hvis man tilføjer nogle led med værdien 0 til ligning (6) fås

$$\begin{aligned} v_n = a_0 \cdot g_n + 0 \cdot g_{n-1} + 0 \cdot g_{n-2} + \ldots + 0 \cdot g_{n-m+1} + \\ a_1 \cdot g_{n-m} + 0 \cdot g_{n-m-1} + 0 \cdot g_{n-m-2} + \ldots + 0 \cdot g_{n-2m+1} + \\ a_2 \cdot g_{n-2m} + 0 \cdot g_{n-2m-1} + 0 \cdot g_{n-2m-2} + \ldots + 0 \cdot g_{n-3m+1} + \\ a_3 \cdot g_{n-3m} + 0 \cdot g_{n-3m-1} + 0 \cdot g_{n-3m-2} + \ldots + 0 \cdot g_{n-4m+1} + \ldots \end{aligned} \tag{8}$$

Hvis man *definerer* en anden diskret sekvens, b_l på følgende måde:

$$b_l = \begin{cases} a_{l/m} & \text{for } l = k \cdot m \text{ hvor } (k \in \mathbb{N}_0) \\ 0 & \text{ellers} \end{cases} \tag{9}$$

kan ligning (8) omskrives som

$$v_n = \sum_{l=0}^{\infty} b_l \cdot g_{n-l} \tag{10}$$

som netop er identisk ved udtrykket for foldningen i ligning (7). Dvs. et PAM-signal kan dannes i MATLAB ved hjælp af en foldning, dvs. ved hjælp af MATLAB's `conv()`-funktion, når bare man har bestemt b_l sekvensen på baggrund af a_k sekvensen.

Ud fra ligning (9) vil de første elementer i b_l sekvensen være:

$$\begin{aligned} b_0 &= a_0, \\ b_1 &= b_2 = \ldots = b_{m-1} = 0, \\ b_m &= a_1, \\ b_{m+1} &= b_{m+2} = \ldots = b_{2m-1} = 0, \\ b_{2m} &= a_2 \\ &\vdots \end{aligned}$$

dvs. b_l-sekvensen dannes ved at der indsættes $m-1$ '0'-værdier efter hver a_k-værdi, dvs. som

$$a_0, 0, 0, \ldots, 0, a_1, 0, 0, \ldots, 0, a_2, 0, 0, \ldots$$

Dvs. det er muligt at generere et PAM-signal i MATLAB på flg. måde:

1. Lav en *opsamplet* version, kaldet au_l (som svarer til b_l-sekvensen ovenfor), af a_k-sekvensen ved at indsætte $m-1$ 0-værdier efter hver værdi fra a_k-sekvensen.

[2] I MATLAB vil der naturligvis være tale om, at både b- og g-sekvenserne har en endelig længde, men det er faktisk ikke nødvendigt i formel (7).

2. Udregn foldningen af au_l og g_l via MATLAB's `conv()`-funktion

Dvs. flg. MATLAB kode kan benyttes til at generere et pulstog:

```
a = ...;  % Informationssekvens
m = ...;  % Antallet af samples pr. symbolperiode
g = ...;  % Udtryk for pulsformen
au = reshape([a; zeros(m-1,length(a))],1,[]);  % Upsamplet version af a-sekvens
v = conv(au,g);
```

Efter denne MATLAB-kode vil rækkevektoren, v, indeholde den samplede udgave af PAM-signalet.