Generering af PAM-signaler i MATLAB

Lars Staalhagen (larst@fotonik.dtu.dk)

12. juni 2017

Dette dokument beskriver, hvordan MATLAB's indbyggede foldningsfunktion, `conv()`, på en nem måde kan benyttes til at generere et tidsdiskret (dvs. samplet) PAM[1]-signal; populært også kaldet et "pulstog".

1 Kontinuerte PAM-signaler

Et (tids-)kontinuert PAM-signal kan helt generelt defineres som i formel (1), hvor det er forudsat, at første informationssymbol, a_0, genereres til $t = 0$, og dermed at der ikke ønskes sendt information før $t = 0$.

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

I formel (1) indgår følgende:

- a_k er informationssymbolerne ($k = 0, 1, 2, \ldots$)
- $g(t)$ er en funktion, der beskriver pulsen som funktion af tiden, t.
- T er symbolperioden, dvs. tidsintervallet mellem to a_k-symboler skal sendes.

Et eksempel på en simpel pulsform er en *firkantformet* puls, $g(t)$ af varighed T, illustreret i figur 1, som kan defineres ligning (2).

$$g(t) = \begin{cases} 1 \quad \text{for} \quad 0 \le t < T \\ \\ 0 \quad \text{ellers} \end{cases} \tag{2}$$

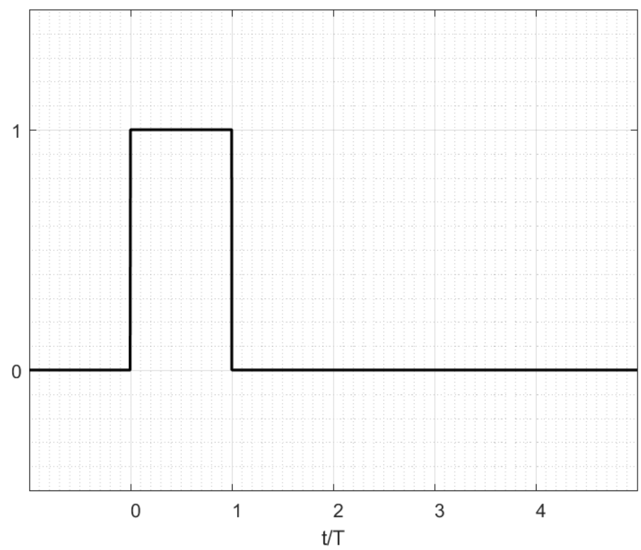
Grunden til, at funktionen for pulsformen i formel (1) har argumentet $t - k \cdot T$ er, at det "forskyder" pulsen, afhængig af nummeret (indeks) på informationssymbolet, k. Figur 2 illustrerer udtrykket $g(t - k \cdot T)$ for $k = 0, 1, 2, 3$, for pulsen i ligning (2) og det kan ses, at grundpulsen "forskydes" $k \cdot T$ til højre – jo større k-værdi, jo mere er pulsen "forskudt".

Figur 3 illustrerer et simpelt PAM-signal, hvor der skal transmitteres informationssekvensen $a_k = +1, +3, -1, +1, -3, +1$, ved at benytte den firkantformede puls fra ligning (2).

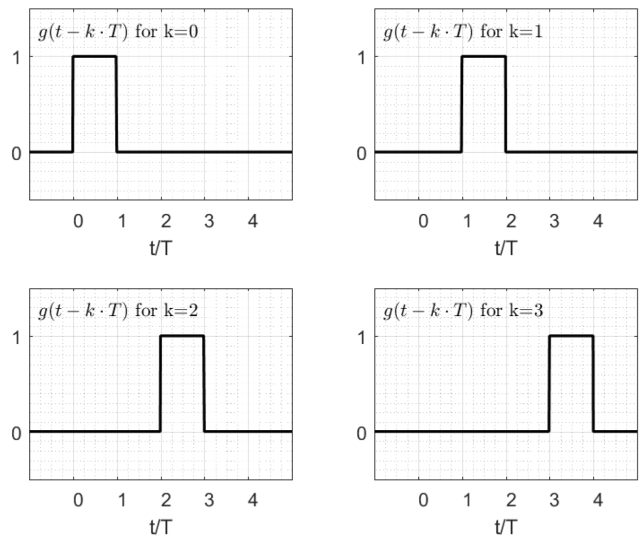
Et PAM-signal kan dog også dannes ud fra andre pulsformer end firkantformede. Et eksempel er en "savtak"-formet puls, der kan defineres som i ligning (3), og figur 4 viser et PAM-signal med samme informationssekvens som ovenfor, men nu med den savtakformede puls i ligning (3).

$$g(t) = \begin{cases} t/T \quad \text{for} \quad 0 \le t < T \\ \\ 0 \qquad \text{ellers} \end{cases} \tag{3}$$

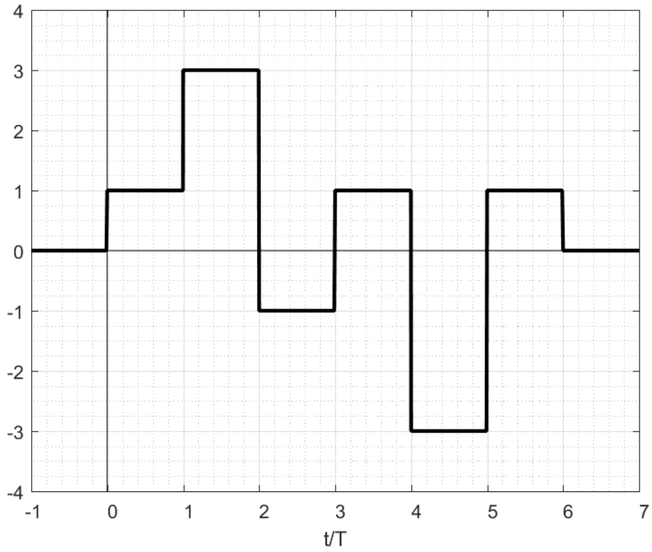
[1] PAM = Puls Amplitude Modulation.



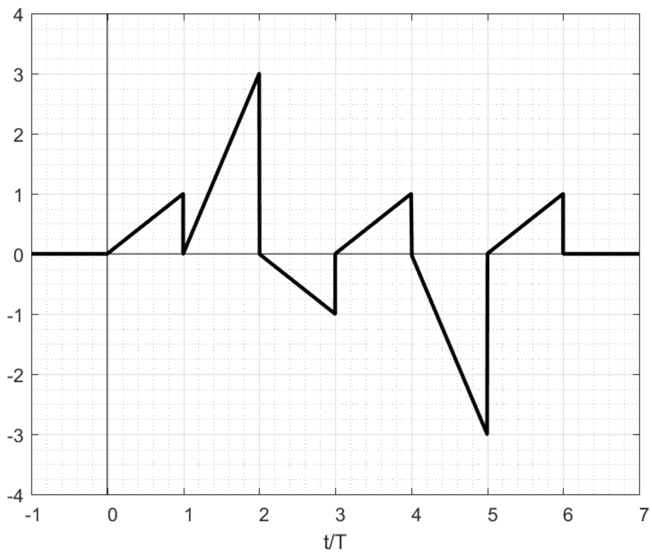
Figur 1: Firkantpuls.



Figur 2: Udtrykket $g(t - k \cdot T)$ for $k = 0, 1, 2$ og 3.



Figur 3: Kontinuert PAM-signal med firkantformet puls.



Figur 4: Kontinuert PAM-signal med savtakformet puls.

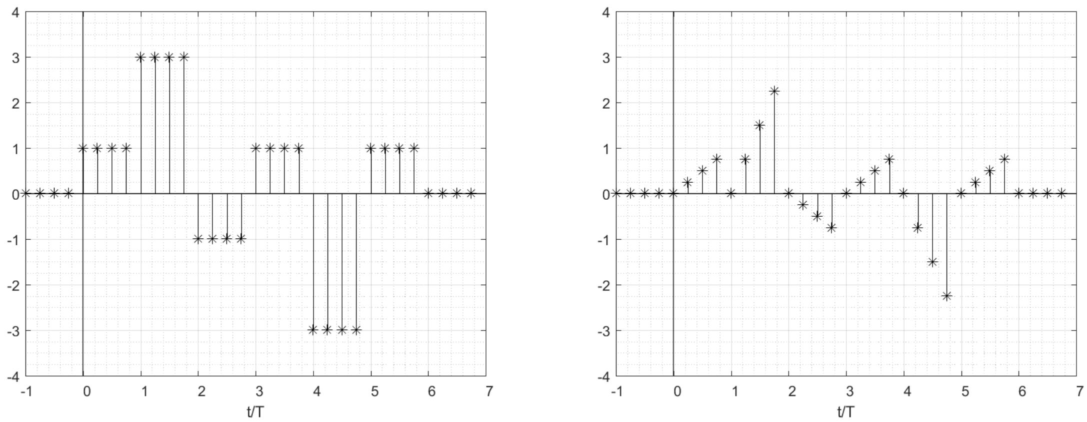
2 Tidsdiskrete PAM-signaler

I digitale kommunikationssystemer er man imidlertid 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):

$$v(n \cdot T_s) = \sum_{k=0}^{\infty} a_k \cdot g(n \cdot T_s - k \cdot T) \tag{4}$$

Figur 5 viser de tidsdiskrete udgave af PAM-signalerne i hhv. figur 3 og figur 4, hvor der i begge tilfælde er samplet 4 gange inden for hver symbolperiode, T, dvs. $T_s = T/4$.



(a) Firkantformet puls (b) Savtakformet puls

Figur 5: Tidsdiskrete PAM-signaler.

I praksis vil samplingsintervallet, T_s, være valgt så $T_s = T/m \Leftrightarrow T = m \cdot T_s$ ($m \in \mathbb{N}$) for en passende værdi af m. Herved kan ligning (4) omskrives

$$v(n \cdot T_s) = \sum_{k=0}^{\infty} a_k \cdot g(n \cdot T_s - k \cdot m \cdot T_s) \tag{5}$$

$$= \sum_{k=0}^{\infty} a_k \cdot g((n - k \cdot m) \cdot T_s) \tag{6}$$

Indføres notationen v_n som en kortere måde at skrive $v(n \cdot T_s)$ på (og tilsvarende mht. pulsen, dvs. $g_{n-k\cdot m} = g((n - k \cdot m) \cdot T_s)$) fås:

$$v_n = \sum_{k=0}^{\infty} a_k \cdot g_{n-k\cdot m} \tag{7}$$

Hvis man herefter opskriver leddene i summen i formel (7) ud fås

$$v_n = a_0 \cdot g_n + a_1 \cdot g_{n-m} + a_2 \cdot g_{n-2m} + \ldots \tag{8}$$

3 Foldningsfunktionen

Matematisk defineres *foldningen* af to tidsdiskrete sekvenser som vist i ligning (9) (hvoraf den ene sekvens blot er g_k-sekvensen som ovenfor)

$$y_n = \sum_{l=0}^{\infty} b_l \cdot g_{n-l} \tag{9}$$

Det ses, at ligningerne (7) og (9) har næsten samme form, og da MATLAB har en indbygget funktion, kaldet `conv()`, der implementerer[2] ligning (9), er det nærliggende at forsøge at omskrive ligning (7), så den har samme form som ligning (9), så et PAM-signal i MATLAB kan genereres via MATLAB's `conv()`-funktion.

Hvis man tilføjer nogle led med værdien 0 til ligning (8) fås

$$\begin{aligned} v_n = a_0 \cdot g_n + 0 \cdot g_{n-1} + 0 \cdot g_{n-2} + \ldots + 0 \cdot g_{n-(m-1)} + \\ a_1 \cdot g_{n-m} + 0 \cdot g_{n-m-1} + 0 \cdot g_{n-m-2} + \ldots + 0 \cdot g_{n-(2m-1)} + \\ a_2 \cdot g_{n-2m} + 0 \cdot g_{n-2m-1} + 0 \cdot g_{n-2m-2} + \ldots + 0 \cdot g_{n-(3m-1)} + \\ a_3 \cdot g_{n-3m} + 0 \cdot g_{n-3m-1} + 0 \cdot g_{n-3m-2} + \ldots + 0 \cdot g_{n-(4m-1)} + \ldots \end{aligned} \tag{10}$$

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

$$b_l = \begin{cases} a_{l/m} & \text{for } l = k \cdot m \text{ hvor } (k \in \mathbb{N}_0) \\ \\ 0 & \text{ellers} \end{cases} \tag{11}$$

kan ligning (10) omskrives som

$$v_n = \sum_{l=0}^{\infty} b_l \cdot g_{n-l} \tag{12}$$

som netop er identisk ved udtrykket for foldningen i ligning (9). 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 (11) 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:

[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 (9).

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. 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.