UNIVERSIDADE DE PERNAMBUCO (UPE)

VINICIUS JOSE FERNANDES RIBEIRO

Atividade Avaliativa

Sistemas de Comunicação

Recife

2024

Resumo

Documento com o intuito de alocar as informações necessárias para realizar as atividades avaliativas propostas pelo Prof. Diego Rátiva, na disciplina de Sistemas de comunicação.

Atividades

Seguirei com a alocação dos exercícios e a transcrição dos respectivos exercícios Propostos

Exercício 1, 2, 3, em range:

3.10 EXERCÍCIOS COM O MATLAB

Cálculo de Transformadas de Fourier

Nesta seção de exercícios baseados em computador, consideremos dois exemplos para ilustrar o uso de DFT no cálculo da transformada de Fourier. Usaremos MATLAB para calcular a DFT com o algoritmo de FFT. No primeiro exemplo, o sinal é $g(t) = e^{-2t} u(t)$, com início em $t = 0$, e no segundo, $g(t) = \Pi(t)$, com início em $t = -½$.

EXEMPLO COMPUTACIONAL C3.1

Empreguemos a DFT (implementada pelo algoritmo de FFT) para calcular a transformada de Fourier de $e^{-2t} u(t)$ e, a seguir, tracemos o gráfico do resultante espectro de Fourier.

> Primeiro, devemos determinar T_s e T_0. A transformada de Fourier de $e^{-2t} u(t)$ é $1/(2\pi f + 2)$. Esse sinal passafaixa não é limitado em frequência. Tomemos sua largura de banda essencial como a frequência em que $|G(f)|$ se torna igual a 1% do valor de pico, que ocorre em $f = 0$. Observemos que
>
> $$|G(f)| = \frac{1}{\sqrt{(2\pi f)^2 + 4}} \approx \frac{1}{2\pi f} \qquad 2\pi f \gg 2$$
>
> O pico de $|G(f)|$ ocorre em $f = 0$, em que $|G(0)| = 0,5$. Portanto, a largura de banda essencial B corresponde a $f = B$, com
>
> $$|G(f)| \approx \frac{1}{2\pi B} = 0,5 \times 0,01 \quad \Rightarrow B = \frac{100}{\pi}\ \text{Hz}$$
>
> e, da Eq. (3.105b),

$$T_s \leq \frac{1}{2B} = 0{,}005\pi = 0{,}0157$$

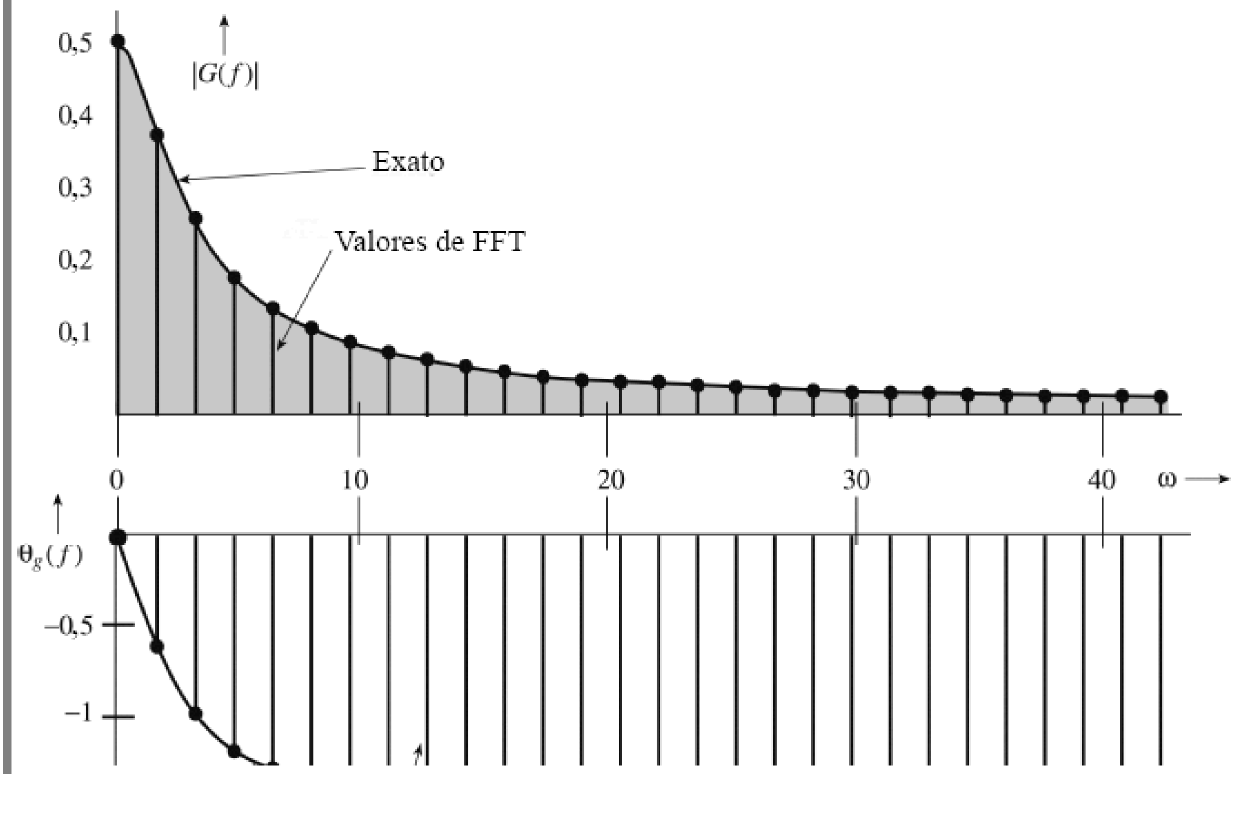
Arredondemos esse valor para $T_s = 0{,}015625$ segundo, de modo que tenhamos 64 amostras por segundo. Agora, devemos determinar T_0. O sinal não é limitado no tempo. Precisamos truncálo em T_0, tal que $g(T_0) \ll 1$. Escolhamos $T_0 = 4$ (oito constantes de tempo do sinal), o que resulta em $N_0 = T_0/T_s = 256$, que é uma potência de 2. Vale ressaltar que há muita flexibilidade na determinação de T_s e T_0, dependendo da precisão desejada e da capacidade computacional disponível. Poderíamos ter escolhido $T_0 = 8$ e $T_s = 1/32$, o que também resultaria em $N_0 = 256$, mas implicaria um erro de mascaramento ligeiramente maior.

Como o sinal tem uma descontinuidade do tipo degrau em $t = 0$, o valor da primeira amostra (em $t = 0$) é 0,5, média dos valores nos dois lados da descontinuidade. O programa de MATLAB que implementa a DFT com o algoritmo de FFT é o seguinte:

```
Ts=1/64; T0=4; N0=T0/Ts;
t=0:Ts:Ts*(N0-1);t=t';
g=Ts*exp(-2*t);
g(1)=Ts*0.5;
G=fft(g);
$[Gp,Gm]$=cart2pol($real(G),imag(G)$);
k=0:N0-1; k=k';
w=2*pi*k/T0;
subplot(211),stem(w(1:32),Gm(1:32));
subplot(212),stem(w(1:32),Gp(1:32))
```

Como G_q tem período N_0, $G_q = G_{(q+256)}$, de modo que $G_{256} = G_0$. Portanto, basta traçar o gráfico de G_q no intervalo $q = 0$ a $q = 255$ (e não 256). Além disso, devido à periodicidade, $G_{-q} = G_{(-q+256)}$, ou seja, os valores de G_q no intervalo $q = -127$ a $q = -1$ são idênticos aos valores de G_q no intervalo $q = 129$ a $q = 255$. Logo, $G_{-127} = G_{129}$, $G_{-126} = G_{130}$,..., $G_{-1} = G_{255}$. Adicionalmente, devido à propriedade de simetria conjugada da transformada de Fourier, $G_{-q} = G^*_q$; assim, $G_{129} = G^*_{127}$, $G_{130} = G^*_{126}$,..., $G_{255} = G^*_1$. Consequentemente, para sinais de valores reais, não é necessário marcar no gráfico os valores de G_q com q maior que $N_0/2$ (128, neste caso), pois são os complexos conjugados dos valores de G_q com $q = 0$ a 128.

O gráfico do espectro de Fourier na Fig. 3.40 mostra amplitude e fase das amostras de $G(f)$ tomadas em intervalos de $1/T_0 = 1/4$ Hz, ou $\omega_0 = 1{,}5708$ rad/s. Na Fig. 3.40, mostramos apenas os primeiros 28 pontos (em vez dos 128 pontos), para evitar o acúmulo excessivo de dados no gráfico.



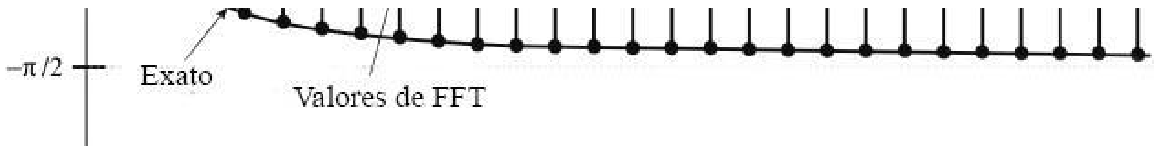


Figura 3.40 Transformada de Fourier discreta de um sinal exponencial $e^{-2t}u(t)$. O eixo horizontal é ω (em radianos por segundo).

Neste exemplo, dispúnhamos da expressão analítica de $G(f)$, o que nos permitiu fazer escolhas INTELIGENTES para B (ou frequência de amostragem f_s). Na prática, em geral, não conhecemos $G(f)$. Na verdade, isso é exatamente o que desejamos calcular. Nesses casos, para determinar B ou f_s , devemos lançar mão de evidências circunstanciais. Devemos, sucessivamente, reduzir o valor de T_s e calcular a transformada até que o resultado satisfaça o desejado número de algarismos significativos.

A seguir, calcularemos a transformada de Fourier de $g(t) = 8\,\Pi(t)$.

EXEMPLO COMPUTACIONAL C3.2

Empreguemos a DFT (implementada pelo algoritmo de FFT) para calcular a transformada de Fourier de $8\,\Pi(t)$ e tracemos o gráfico do resultante espectro de Fourier.

Essa função retangular e sua transformada de Fourier são mostradas na Fig. 3.41a e b. Para determinar o valor do intervalo de amostragem T_s , devemos, primeiro, definir a largura de banda essencial B. Da Fig. 3.41b, vemos que $G(f)$ decai lentamente com f. Consequentemente, a largura de banda essencial B é bastante grande. Por exemplo, em $B = 15{,}5$ Hz (97,39 rad/s), $G(f) = -0{,}1643$, o que corresponde a cerca de 2% do valor de pico, $G(0)$. Poderíamos, então, tomar a largura de banda essencial como 16 Hz. No entanto, deliberadamente, tomaremos $B = 4$ Hz, por dois motivos: (1) mostrar o efeito de mascaramento e (2) o uso de $B > 4$ implicaria enorme número de amostras, que não poderiam ser mostradas de forma adequada em uma página de livro sem perda de detalhes fundamentais. Portanto, aceitaremos a aproximação para que possamos esclarecer conceitos de DFT por meio de gráficos.

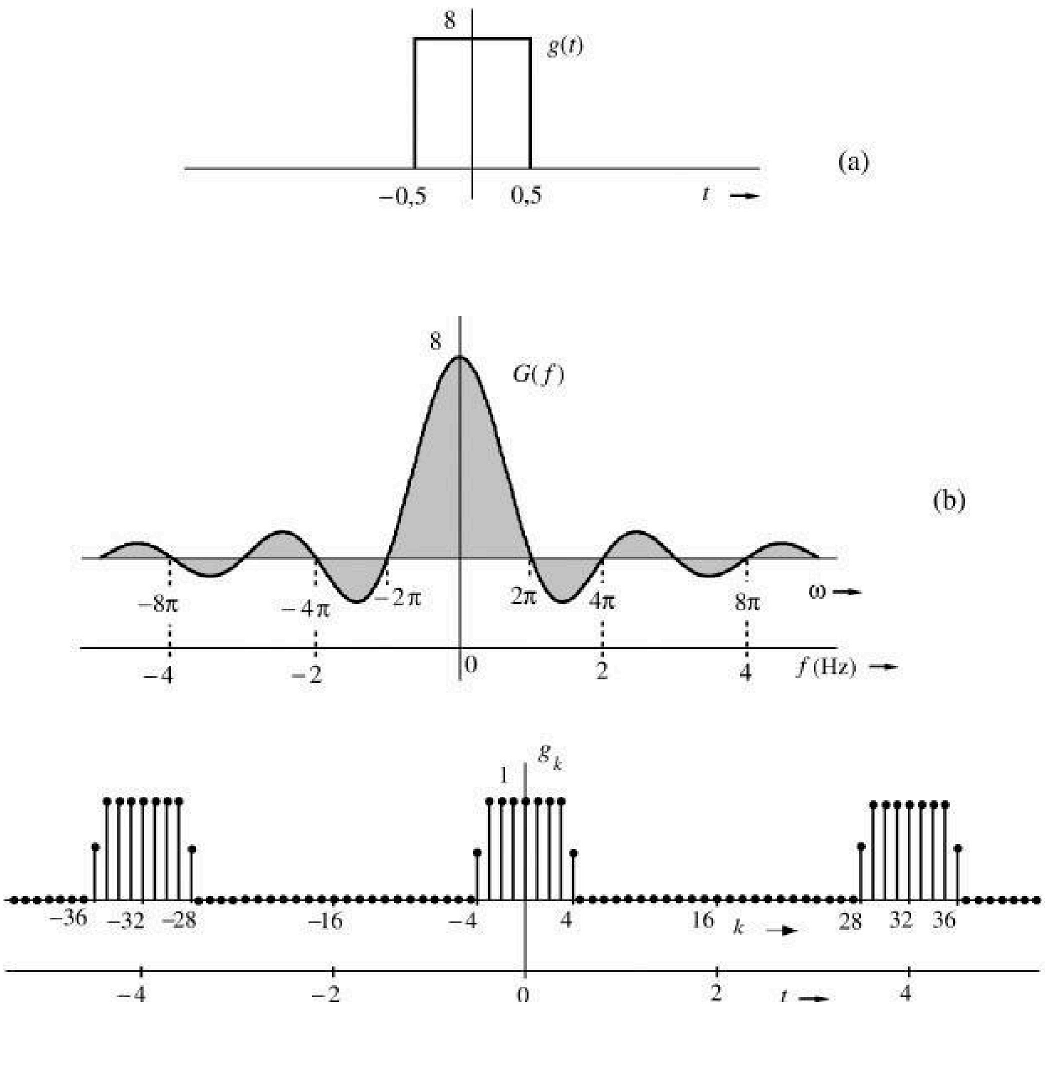
A escolha $B = 4$ resulta em um intervalo de amostragem $T_s = 1/2B = 1/8$ segundos. Examinando novamente o espectro na Fig. 3.41b, vemos que a escolha da resolução de frequência $f_0 = 1/4$ Hz é razoável, e corresponde a quatro amostras em cada lóbulo de $G(f)$. Neste caso, $T_0 = 1/f_0 = 4$ segundos, e $N_0 = T_0/T_s = 32$. A duração de $g(t)$ é de apenas 1 segundo. Devemos repetir $g(t)$ a cada 4 segundos, como indicado na Fig. 3.41c, e tomar amostras a cada 0,125 segundo. Isso nos dará 32 amostras ($N_0 = 32$). Também temos

$$\begin{aligned} g_k &= T_s g(kT) \\ &= \frac{1}{8} g(kT) \end{aligned}$$

Como $g(t) = 8\,\Pi(t)$, os valores de g_k são 1, 0 ou 0,5 (nos pontos de descontinuidade), como mostrado na Fig. 3.41c; nessa figura, por conveniência, g_k é mostrado como função de t e de k.

Na dedução da DFT, supomos que $g(t)$ tem início em $t = 0$ (Fig. 3.39a) e tomamos N_0 amostras no intervalo $(0, T_0)$. No caso em consideração, contudo, $g(t)$ tem início em $t = -½$. Essa dificuldade é facilmente resolvida quando observamos que a DFT obtida por este procedimento é, na verdade, a DFT de g_k repetido a cada T_0 segundos. Da Fig. 3.41c, fica claro que a repetição periódica do segmento de g_k no intervalo de −2 a 2 segundos é equivalente à repetição do segmento de g_k no intervalo de 0 a 4 segundos. Portanto, a DFT das amostras colhidas entre −2 e 2 segundos é igual à DFT das amostras colhidas entre 0 e 4 segundos. Assim, independentemente do instante em que $g(t)$ tem início, sempre podemos tomar as amostras de $g(t)$ e repetilas periodicamente no intervalo de 0 a T_0 . No presente exemplo, os valores das 32 amostras são

$$g_k = \begin{cases} 1 & 0 \le k \le 3 \quad \text{e} \quad 29 \le k \le 31 \\ 0 & 5 \le k \le 27 \\ 0{,}5 & k = 4.28 \end{cases}$$



8
g(t)
−0,5
0,5
t
8
G(f)
−8π
−4π
−2π
2π
4π
8π
ω
−4
−2
0
2
4
f (Hz)
g_k
1
−36
−32
−28
−16
−4
4
16
k
28
32
36
−4
−2
0
2
t
4

(a)

(b)

(c)

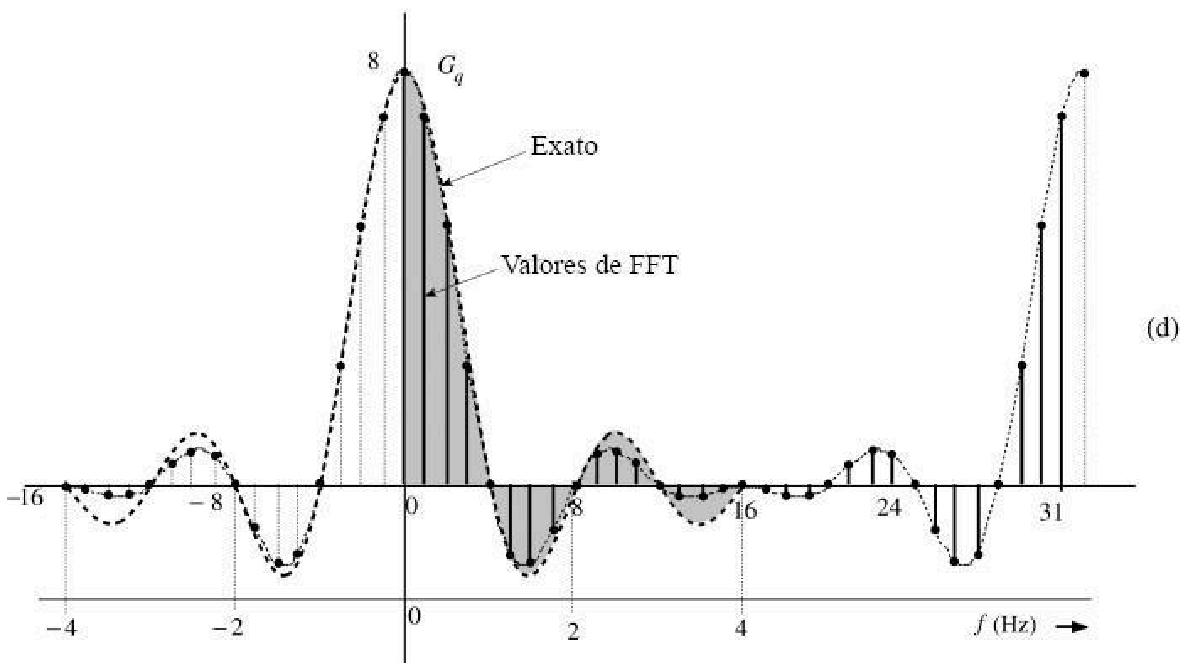


Figura 3.41 Transformada de Fourier discreta de um pulso retangular.

Vale ressaltar que a última amostra é tomada em t = 31/8 e não em t = 4, pois a repetição do sinal reinicia em t = 4, de modo que a amostra em t = 4 é igual à amostra em t = 0. Com $N_0 = 32$, $\Omega_0 = 2\pi/32 = \pi/16$. Logo, [ver a Eq. (3.103a)],

$$G_q = \sum_{k=0}^{31} g_k e^{-jq\frac{\pi}{16}k}$$

O programa MATLAB que usa o algoritmo de FFT para calcular a DFT é dado a seguir. Primeiro, escrevemos um programa MATLAB para gerar 32 amostras de g_k e, então, calculamos a DFT.

```
% (c32.m)
B=4;    f0=1/4;
Ts=1/(2*B); T0=1/f0;
N0=T0/Ts;
k=0:N0; k=k';
for m=1:length(k)
$  $ if k(m)$>$=0 & k(m)$<$=3, gk(m)=1; end
$  $ if k(m)==4 & k(m)==28 gk(m)=0.5; end
$  $ if k(m)$>$=5 & k(m)$<$=27, gk(m)=0; end
$  $ if k(m)$>$=29 & k(m)$<$=31, gk(m)=1; end
end
gk=gk';
Gr=fft(gk);
subplot(211),stem(k,gk)
subplot(212),stem(k,Gr)
```

A Fig. 3.41d mostra o gráfico de G_q .

As amostras G_q são espaçadas de $f_0 = 1/T_0$ Hz. Neste exemplo, T_0 = 4 segundos, de modo que a resolução de frequência f_0 é ¼ Hz, como desejado. A frequência de dobramento $f_s/2 = B$ = 4 Hz corresponde a $q = N_0/2 = 16$. Como G_a tem período N_0 (N_0 = 32), os valores de G_q para q entre −16 e −1 são iguais àqueles para q entre 16 e 31. A DFT nos fornece amostras do espectro $G(f)$.

Para facilitar a comparação, a Fig. 3.41d também mostra a curva hachurada 8 sinc(πf), que é a transformada de Fourier de 8 Π(t). Os valores de G_q calculados pela DFT exibem erro de mascaramento, o que fica claro quando comparamos os

dois gráficos. O erro em G_2 é da ordem de apenas 1,3%. No entanto, o erro de mascaramento aumenta rapidamente com *r*. Por exemplo, o erro em G_6 é de cerca de 12%, e o erro em G_{10}, 33%. O erro em G_{14} é de assustadores 72%. O erro percentual aumenta de forma muito rápida nas proximidades da frequência de dobramento (r = 16), pois $g(t)$ tem uma descontinuidade degrau, o que faz com que $G(f)$ decaia muito lentamente, como $1/f$. Assim, nas proximidades da frequência de dobramento, a cauda invertida (devido ao mascaramento) é quase igual a $G(f)$. Além disso, os valores extremos são a diferença entre os valores exato e da parte que sofreu dobra (quase iguais aos exatos). Consequentemente, o erro percentual nas proximidades da frequência de dobramento (r = 16, neste exemplo) é muito alto, embora o erro absoluto seja muito pequeno. Fica claro que, para sinais com descontinuidades do tipo degrau, o erro de mascaramento nas proximidades da frequência de dobramento sempre será grande (em termos percentuais), qualquer que seja o valor escolhido para N_0. Para garantir erro de mascaramento desprezível para qualquer valor de q, devemos assegurar que N_0
$\gg q$. Essa observação se aplica a todos os sinais com descontinuidade do tipo degrau.

Filtragem

Quando pensamos em filtragem, em geral, o fazemos em termos de uma solução orientada a hardware (ou seja, montagem de um circuito com componentes *RLC* e amplificadores operacionais). Contudo, a filtragem também admite uma solução orientada a software [algoritmo computacional que fornece a saída filtrada $y(t)$, para uma dada entrada $g(t)$]. Isso pode ser implementado de modo conveniente via DFT. Seja $g(t)$ o sinal a ser filtrado; então, os valores G_q, DFT de g_k, são calculados. O espectro G_q é formatado (filtrado) como desejado através da multiplicação de G_q por H_q, em que H_q são as amostras da função de transferência do filtro, $H(f)$ [$H_q = H(qf_0)$]. Por fim, calculamos a DFT inversa (ou IDFT) de $G_q H_q$ e obtemos a saída filtrada y_k [$y_k = T_s\, y(kT)$]. O próximo exemplo ilustra este procedimento.

EXEMPLO COMPUTACIONAL C3.3

O sinal $g(t)$ na Fig. 3.42a é aplicado a um filtro passabaixos ideal, cuja função de transferência $H(f)$ é mostrada na Fig. 3.42b. Usemos a DFT para calcular a saída do filtro.

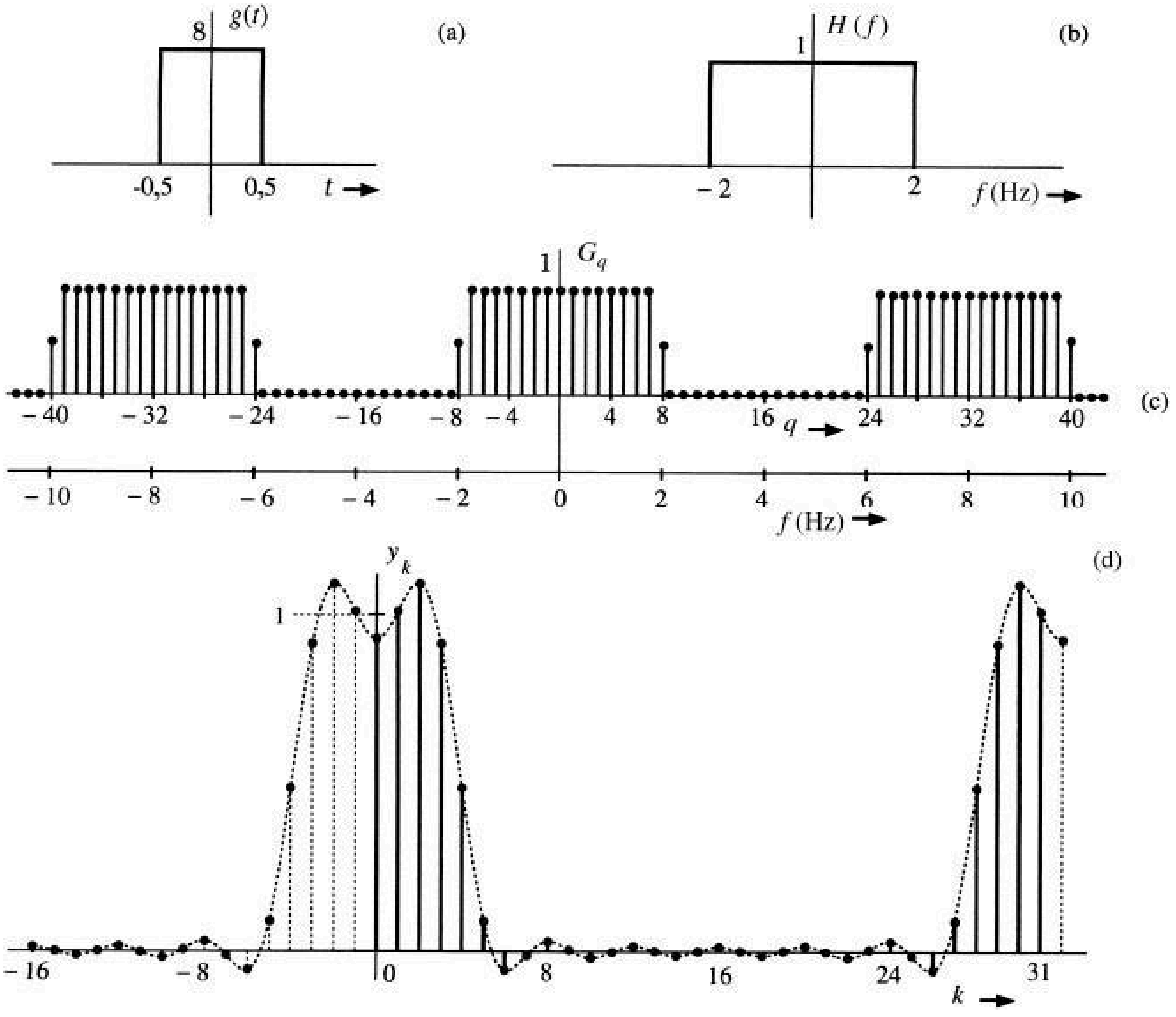


Figura 3.42 Filtragem de *g*(*t*) por *H*(*f*).

Já calculamos a DFT de $g(t)$ com 32 amostras (Fig. 3.41d). Agora, devemos multiplicar G_q por H_q . Para calcular H_q , recordemos que, na determinação da DFT de $g(t)$ com 32 amostras, usamos $f_0 = 0{,}25$ Hz. Como G_q tem período $N_0 = 32$, H_q deve ter o mesmo período e, portanto, amostras espaçadas de 0,25 Hz. Isso significa que H_q deve se repetir a cada 8 Hz ou 16π rad/s (ver Fig. 3.42c). Assim, as 32 amostras de H_q são produzidas, no intervalo $0 \leq f \leq 8$, como

$$H_q = \begin{cases} 1 & 0 \leq q \leq 7 \quad \text{e} \quad 25 \leq q \leq 31 \\ 0 & 9 \leq q \leq 23 \\ 0{,}5 & q = 8.24 \end{cases}$$

Multiplicamos G_q por H_q e calculamos a DFT inversa. O resultante sinal de saída é mostrado na Fig. 3.42d. A Tabela 3.4 lista valores de g_k , G_q , H_q , Y_q e y_k .

No Exemplo C.32, já calculamos a DFT de $g(t)$ com 32 amostras (G_q). O programa MATLAB do Exemplo C3.2 pode ser armazenado como um arquivo.m (por exemplo, "c32.m"). Podemos importar G_q no ambiente MATLAB via comando "c32". A seguir, geramos 32 amostras de H_q , multiplicamos G_q por H_q e, para obter y_k , calculamos a DFT inversa. Também podemos obter y_k calculando a convolução de g_k e h_k .

```
c32;
q=0:32; q=q';
for m=1:length(q)
     if q(m)$>$=0 & q(m)$<$=7, Hq(m)=1; end
     if q(m)$>$=25 & q(m)$<$=31, Hq(m)=1; end
     if q(m)$>$=9 & q(m)$<$=23, Hq(m)=0; end
     if q(m)==8 & q(m)==24, Hq(m)=0.5; end
end
Hq=Hq';
Yq=Gq.*Hq;
yk=ifft(Yq);
clf,stem(k,yk)
```

Tabela 3.4

No.	gk	G_q	H_q	G_qH_q	y_k
0	1	8,000	1	8,000	0,9285
1	1	7,179	1	7,179	1,009
2	1	5,027	1	5,027	1,090
3	1	2,331	1	2,331	0,9123
4	1	0,000	1	0,000	0,4847
5	0,5	−1,323	1	−1,323	0,08884
6	0	−1,497	1	−1,497	−0,05698
7	0	−0,8616	1	−0,8616	−0,01383
8	0	0,000	0,5	0,000	0,02933
9	0	0,5803	0	0,000	0,004837
10	0	0,6682	0	0,000	−0,01966
11	0	0,3778	0	0,000	−0,002156

12	0	0,000	0	0,000	0,01534
13	0	−0,2145	0	0,000	0,0009828
14	0	−0,1989	0	0,000	−0,01338
15	0	−0,06964	0	0,000	−0,0002876
16	0	0,000	0	0,000	0,01280
17	0	−0,06964	0	0,000	−0,0002876
18	0	−0,1989	0	0,000	−0,01338
19	0	−0,2145	0	0,000	0,0009828
20	0	0,000	0	0,000	0,01534
21	0	0,3778	0	0,000	−0,002156
22	0	0,6682	0	0,000	−0,01966
23	0	0,5803	0	0,000	0,004837
24	0	0,000	0,5	0,000	0,03933
25	0	−0,8616	1	−0,8616	−0,01383
26	0	−1,497	1	−1,497	−0,05698
27	0	−1,323	1	−1,323	0,08884
28	0,5	0,000	1	0,000	0,4847
29	1	2,331	1	2,331	0,9123
30	1	5,027	1	5,027	1,090
31	1	7,179	1	7,179	1,009

REFERÊNCIAS

1. R. V. Churchill and J. W. Brown, *Fourier Series and Boundary Value Problems*, 3rd ed., McGrawHill, New York, 1978.
2. R. N. Bracewell, *Fourier Transform and Its Applications*, rev. 2nd ed., McGrawHill, New York, 1986.
3. B. P. Lathi, *Signal Processing and Linear Systems*, Oxford University Press, 2000.
4. E. A. Guillemin, *Theory of Linear Physical Systems*, Wiley, New York, 1963.
5. F. J. Harris, "On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform," *Proc. IEEE*, vol. 66, pp. 51–83, Jan. 1978.
6. J. W. Tukey and J. Cooley, "An Algorithm for the Machine Calculation of Complex Fourier Series," *Mathematics of Computation*, Vol. 19, pp. 297–301, April 1965.

Códigos (Imagem e transcrição)

Exercício 1

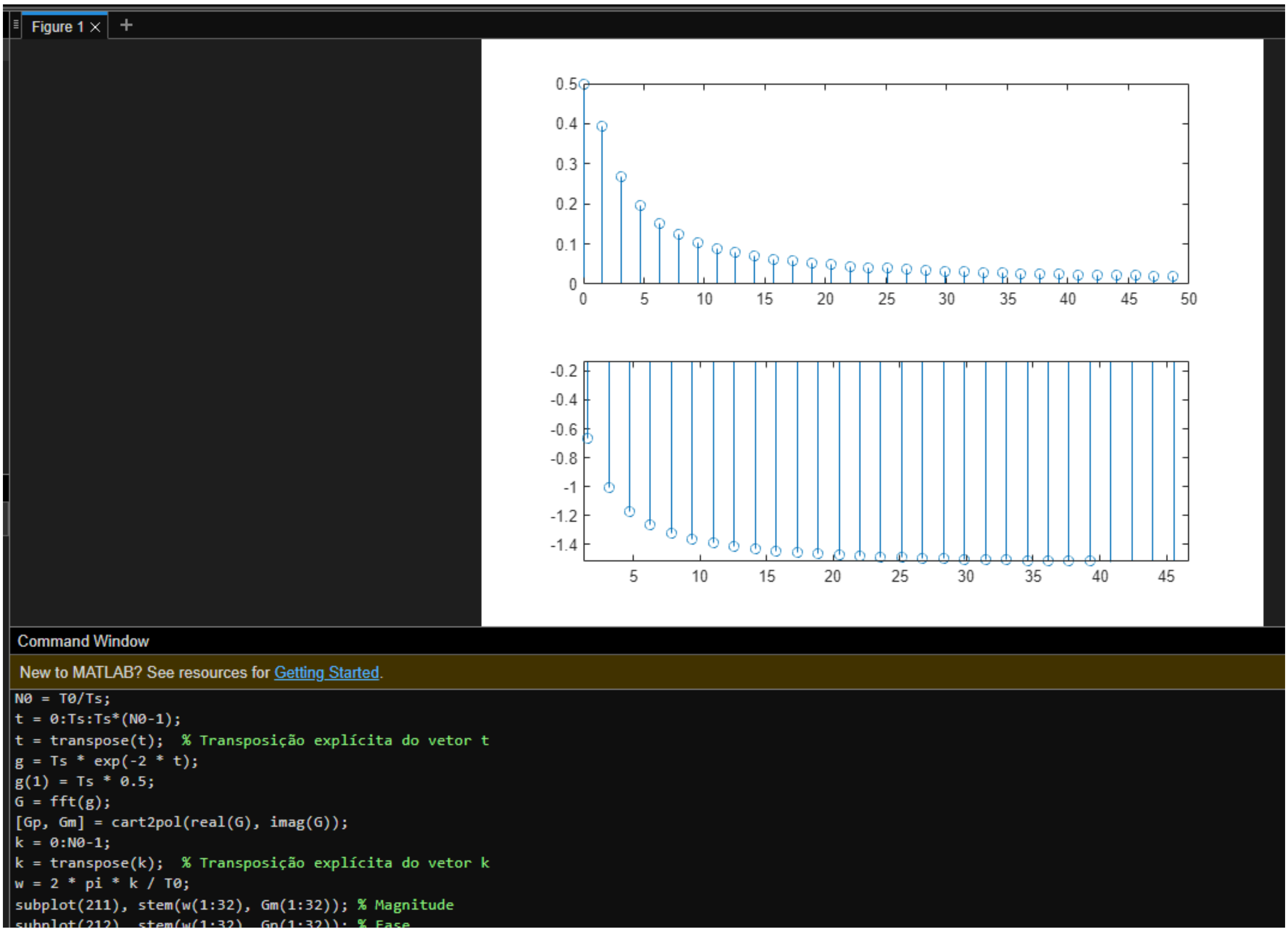
```
Ts=1/64; T0=4; N0=T0/Ts;
t=0:Ts:Ts*(N0-1);t=t';
g=Ts*exp(-2*t);
g(1)=Ts*0.5;
G=fft(g);
$[Gp,Gm]$=cart2pol($real(G),imag(G)$);
k=0:N0-1; k=k';
w=2*pi*k/T0;
subplot(211),stem(w(1:32),Gm(1:32));
subplot(212),stem(w(1:32),Gp(1:32))
```

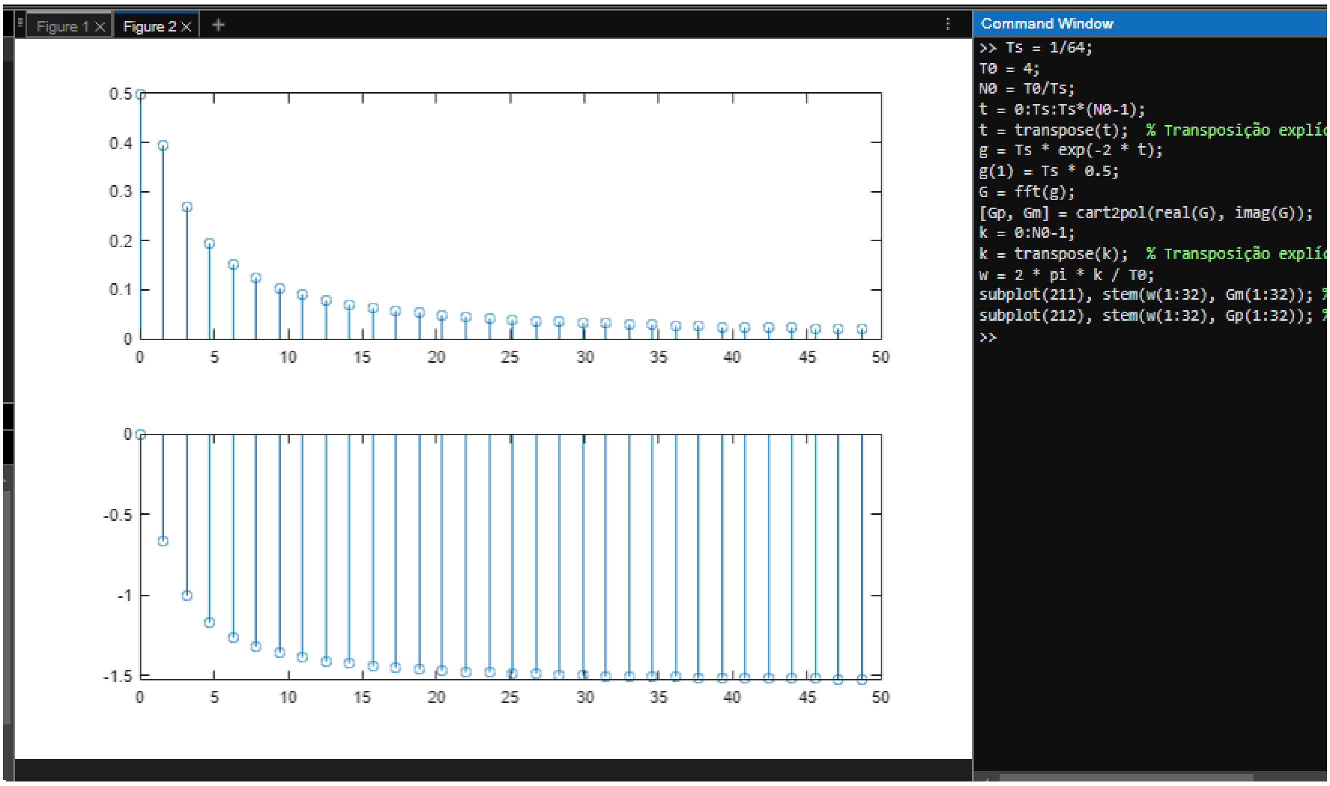
Transcript:

```
Ts=1/64; T0=4; N0=T0/TS;
t=0:Ts:Ts* (N0-1); t=t';
g=Ts*exp(-2*t);
```

```
g(1) Ts*0.5;
G=fft (g);
$ [Gp, Gm] $=cart2pol ($real (G), imag (G) $);
k=0:N0-1; k=k';
w=2*pi*k/TO;
subplot (211), stem (w (1:32), Gm (1:32));
subplot (212), stem (w (1:32), Gp (1:32));
```

Resultado:





corrigido:

```
Ts = 1/64;
T0 = 4;
N0 = T0/Ts;
t = 0:Ts:Ts*(N0-1);
t = transpose(t);  % Transposição explícita do vetor t
g = Ts * exp(-2 * t);
g(1) = Ts * 0.5;
G = fft(g);
[Gp, Gm] = cart2pol(real(G), imag(G));
k = 0:N0-1;
k = transpose(k);  % Transposição explícita do vetor k
w = 2 * pi * k / T0;
subplot(211), stem(w(1:32), Gm(1:32)); % Magnitude
subplot(212), stem(w(1:32), Gp(1:32)); % Fase
```

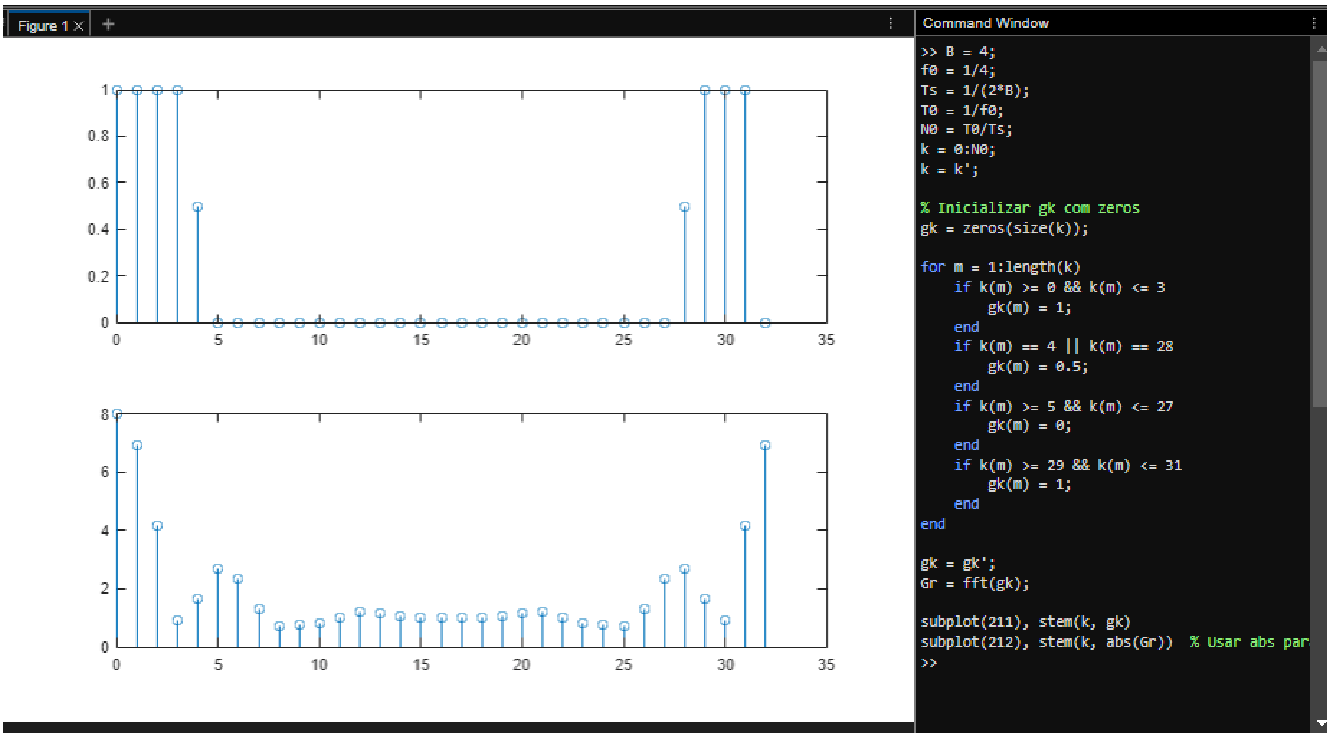
Exercício 2

```
% (c32.m)
B=4;     f0=1/4;
Ts=1/(2*B); T0=1/f0;
N0=T0/Ts;
k=0:N0; k=k';
for m=1:length(k)
$  $ if k(m)$>$=0 & k(m)$<$=3, gk(m)=1; end
$  $ if k(m)==4 & k(m)==28 gk(m)=0.5; end
$  $ if k(m)$>$=5 & k(m)$<$=27, gk(m)=0; end
$  $ if k(m)$>$=29 & k(m)$<$=31, gk(m)=1; end
end
gk=gk';
Gr=fft(gk);
subplot(211),stem(k,gk)
subplot(212),stem(k,Gr)
```

Transcript:

```
B=4;
f0=1/4;
Ts=1/(2*B); T0=1/f0;
N0=T0/Ts;
k=0:N0; k=k';
for m=1:length(k)
$ $ if k (m) $>$=0 & k (m) $<$=3, gk (m)=1; end
$ $ if k (m)==4 & k (m) ==28 gk (m)=0.5; end
$ $ if k (m) $>$=5 & k (m) $<$=27, gk (m) =0; end
$ $ if k (m) $>$=29 & k (m) $<$=31, gk (m) =1; end
end
gk=gk';
Gr=fft (gk);
subplot (211), stem (k,gk)
subplot (212), stem (k, Gr)
```

Resultado:



Corrigido:

```
B = 4;
f0 = 1/4;
Ts = 1/(2*B);
T0 = 1/f0;
N0 = T0/Ts;
k = 0:N0;
k = k';

% Inicializar gk com zeros
gk = zeros(size(k));

for m = 1:length(k)
    if k(m) >= 0 && k(m) <= 3
        gk(m) = 1;
    end
    if k(m) == 4 || k(m) == 28
        gk(m) = 0.5;
    end
    if k(m) >= 5 && k(m) <= 27
        gk(m) = 0;
    end
    if k(m) >= 29 && k(m) <= 31
        gk(m) = 1;
    end
end

gk = gk';
Gr = fft(gk);

subplot(211), stem(k, gk)
subplot(212), stem(k, abs(Gr))  % Usar abs para a magnitude da FFT
```

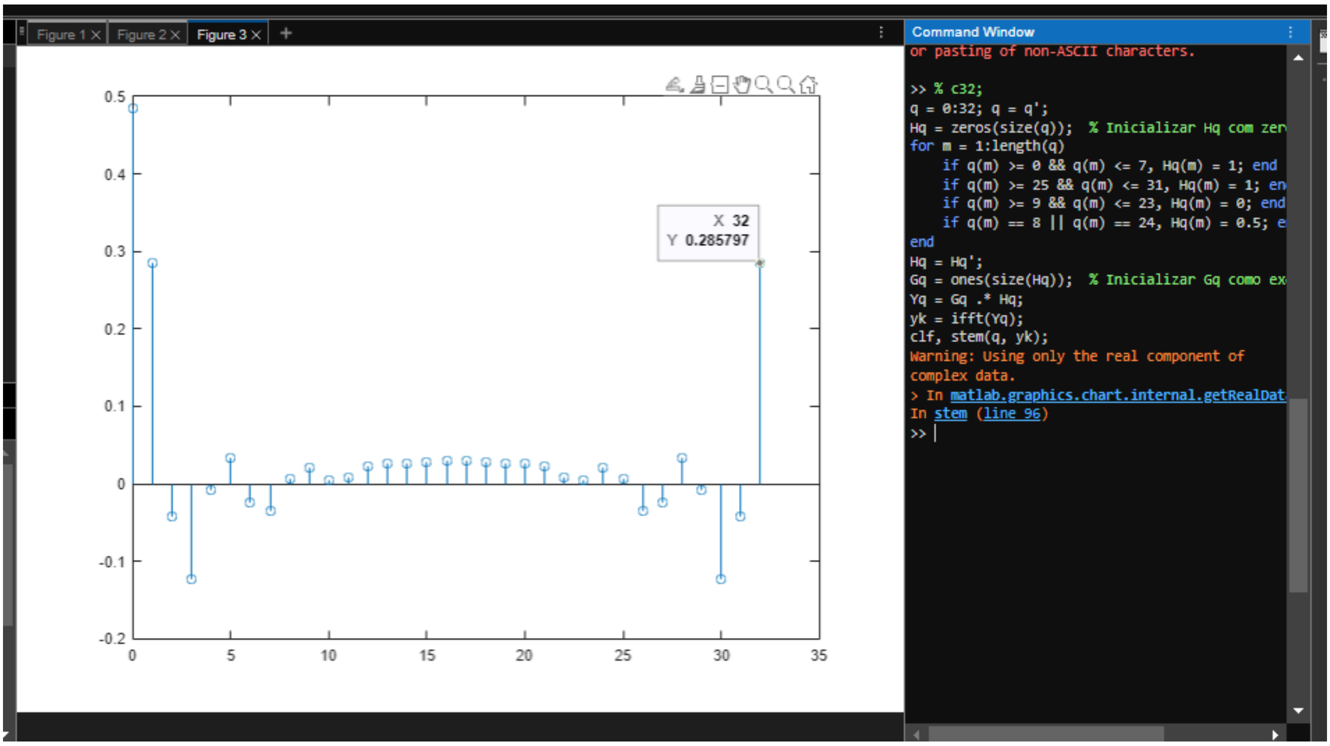
Exercício 3

```
c32;
q=0:32; q=q';
for m=1:length(q)
      if q(m)$>$=0 & q(m)$<$=7, Hq(m)=1; end
      if q(m)$>$=25 & q(m)$<$=31, Hq(m)=1; end
      if q(m)$>$=9 & q(m)$<$=23, Hq(m)=0; end
      if q(m)==8 & q(m)==24, Hq(m)=0.5; end
end
Hq=Hq';
Yq=Gq.*Hq;
yk=ifft(Yq);
clf,stem(k,yk)
```

Transcript:

```
% c32;
q=0:32; q=q';
for m=1:length(q)
      if q(m) $>$=0 & q(m) $<$=7, Hq (m)=1; end
      if q(m) $>$=25 & q(m) $<$=31, Hq (m)=1; end
      if q(m) $>$=9 & q(m) $<$=23, Hq (m)=0; end
      if q(m) ==8 & q(m) ==24, Hq (m)=0.5; end
Hq=Hq';
Yq=Gq. *Hq;
yk=ifft (Yq);
clf, stem (k, yk);
```

Resultado:



Corrigido:

```
% c32;
q = 0:32; q = q';
Hq = zeros(size(q));  % Inicializar Hq com zeros
for m = 1:length(q)
   if q(m) >= 0 && q(m) <= 7, Hq(m) = 1; end
   if q(m) >= 25 && q(m) <= 31, Hq(m) = 1; end
   if q(m) >= 9 && q(m) <= 23, Hq(m) = 0; end
   if q(m) == 8 || q(m) == 24, Hq(m) = 0.5; end
end
Hq = Hq';
Gq = ones(size(Hq));  % Inicializar Gq como exemplo
Yq = Gq .* Hq;
yk = ifft(Yq);
clf, stem(q, yk);
```

—