سیگنال‌ها و سیستم‌ها - دکتر سلیمی‌بدر

امیرحسین منصوری - ۹۹۲۴۳۰۶۹ - تمرین کامپیوتری سری ۲

سوال ۱

ابتدا دو تابع زیر را تعریف می‌کنیم.

```
function out = my_square_wave(T, A, D)
    scaledDutyCycle = D * 100;
    newOmega_0 = (2*pi)/T;
    dutyCycleShift = (T*D)/2;
    out = @(t) square(newOmega_0*(t+dutyCycleShift), scaledDutyCycle) * (A/2) + (A/2);
end

function out = my_sawtooth(T, A)
    newOmega_0 = (2*pi)/T;
    shift = T/2;
    out = @(t) sawtooth(newOmega_0*(t+shift), 1) * A;
end
```

شرح توابع به صورت زیر است:

- تابع my_sqaure_wave، با گرفتن پارامترهای T که برابر دوره تناوب است، A که نشان‌دهنده دامنه سیگنال موج مربعی است، و D که برابر duty cycle سیگنال و عددی بین صفر و ۱ است و قسمتی در یک تناوب که مقدار سیگنال صفر می‌شود را نشان می‌دهد را می‌گیرد، و تابعی دیگر برمیگرداند که همان تابع موج مربعی با پارامترهای داده شده است (درواقع خروجی این تابع، یک تابع است).
  برای ساختن این تابع، از تابع square متلب استفاده شده است. این تابع، سیگنال موج‌مربعی را با دامنه ۱ و دوره تناوب 2π پیاده می‌کند، که حول محور t نوسان می‌کند. برای تبدیل این دوره تناوب به دوره تناوب دلخواه T، کافیست ω_0 جدید سیگنال مطلوب را با توجه به T داده شده محاسبه کنیم.
  چون ω_0 تابع square برابر ۱ است، برای رسیدن به ω_0 جدید کافیست مقدار داده شده به تابع square را ضرب در $\frac{2\pi}{T}$ کنیم.
  همچنین برای این که سیگنال تعریف شده نسبت به $t = 0$ متقارن باشد، لازم است به مقدار $\frac{T \cdot D}{2}$ سیگنال را به چپ شیفت دهیم.
  در نهایت برای اعمال دامنه خواسته شده، با توجه به این که دامنه‌ی سیگنال ساخته شده توسط sqaure برابر ۲ است (بین ۱ و ۱- نوسان می‌کند)، کافیست مقادیر برگردانده شده از square را در مقدار $\frac{A}{2}$ ضرب کنیم، و سپس به اندازه $\frac{A}{2}$ سیگنال را به بالا شیفت دهیم.
  همچنین چون مقدار duty cycle ای که تابع square می‌گیرد بین صفر تا ۱۰۰ است، مقدار D داده شده را در ۱۰۰ ضرب می‌کنیم و به تابع square می‌دهیم.

- تابع my_sawtooth، با گرفتن پارامترهای T که دوره تناوب و A که دامنه است، تابعی برمی‌گرداند که سیگنال دندان اره‌ای با پارامترهای داده شده است (این تابع نیز یک تابع برمی‌گرداند).
  این تابع از تابع sawtooth متلب استفاده می‌کند که تابع موج دندان اره‌ای با دامنه ۱ و دوره تناوب 2π را می‌سازد. تبدیل دوره تناوب و تبدیل دامنه، مشابه تابع my_square_wave انجام شده است؛

با این تفاوت که نیازی به شیفت به بالا نبوده است. همچنین برای تقارن سیگنال خروجی نسبت به $t = 0$، تابع به اندازه $\frac{T}{2}$ به چپ شیفت داده شده است.

توابع مربوط به سری فوریه به صورت زیر است.

```
function out = fourier_coeff_at(k, func, T)
    omega_0 = (2*pi)/T;

    integrand = @(t) (func(t) .* exp(-1i*k*omega_0*t));
    out = (1/T) * integral(integrand, 0, T);
end
```

این تابع، با گرفتن سیگنال func به صورت یک تابع یک متغیره، و دوره تناوب T، ضریب k ام سری فوریه آن را محاسبه می‌کند. با استفاده از تابع integral متلب، رابطه آنالیز که به صورت زیر است محاسبه می‌شود.

$$a_k = \frac{1}{T} \int_{<T>} x(t) e^{-jk\omega_0 t} dt$$

انتگرال را می‌توان در هر دوره تناوبی محاسبه کرد. در اینجا، انتگرال از ۰ تا T محاسبه شده است. به عنوان مثال، محاسبه این انتگرال در بازه $-\frac{T}{2}$ تا $+\frac{T}{2}$ نیز ممکن بود.

```
function out = fourier_approx(t, m, func, T)
    omega_0 = (2*pi)/T;

    out = zeros(size(t));

    for k = -m:m
        out = out + ((fourier_coeff_at(k, func, T) .* exp(1i*omega_0*k*t)));
    end
end
```

این تابع، حاصل تقریب سیگنال func که به صورت یک تابع یک متغیره داده می‌شود و دارای دوره تناوب t است را در t(ها)ی داده شده برمی‌گرداند. به عبارتی، خروجی این تابع یک عدد یا یک آرایه (بسته به نوع t) خواهد بود. همچنین این تقریب، با استفاده از جملات m- ام تا mام سری فوریه انجام می‌شود.

در حلقه for بالا، به ازای هر k، تک جمله‌ای رابطه سنتز محاسبه می‌شود و با مجموع جملات قبلی که از k های قبلی محاسبه شده، جمع می‌شود؛ درواقع حاصل رابطه زیر محاسبه می‌شود.

$$\hat{x}(t) = \sum_{k=-m}^{m} a_k e^{jk\omega_0 t}$$

در نهایت، کدهای مربوط به رسم نمودار به صورت زیر است.

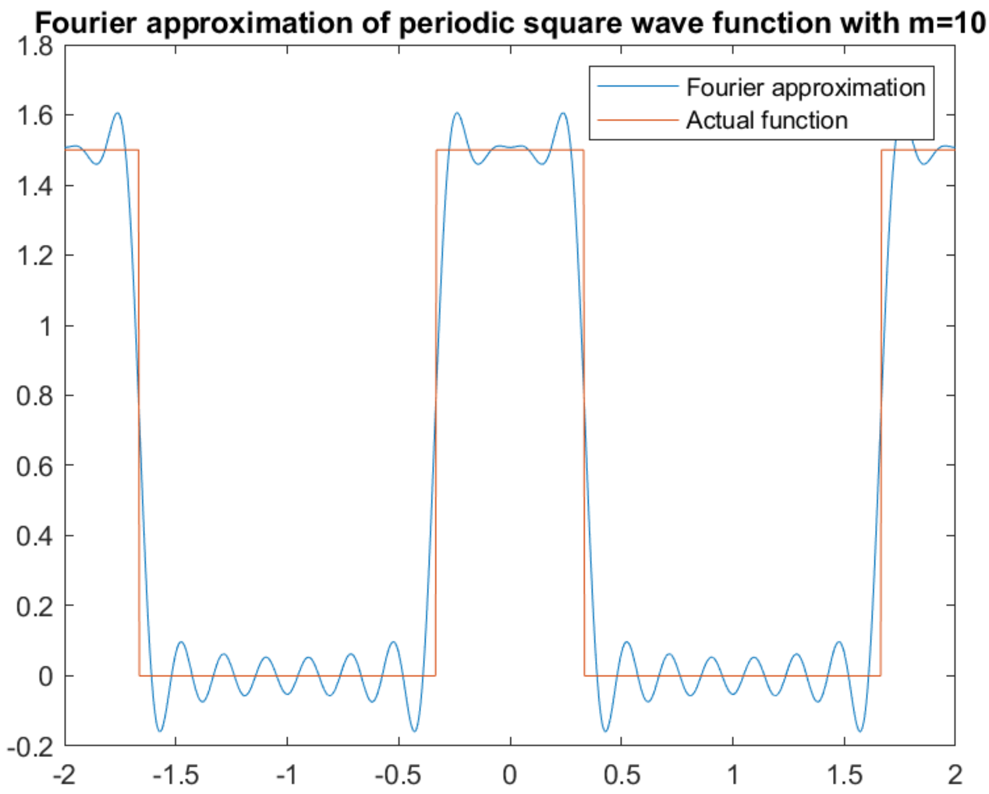
```
T = 2;
A = 1.5;
D = 1/3;

t = linspace(-T, T, 1000);
```

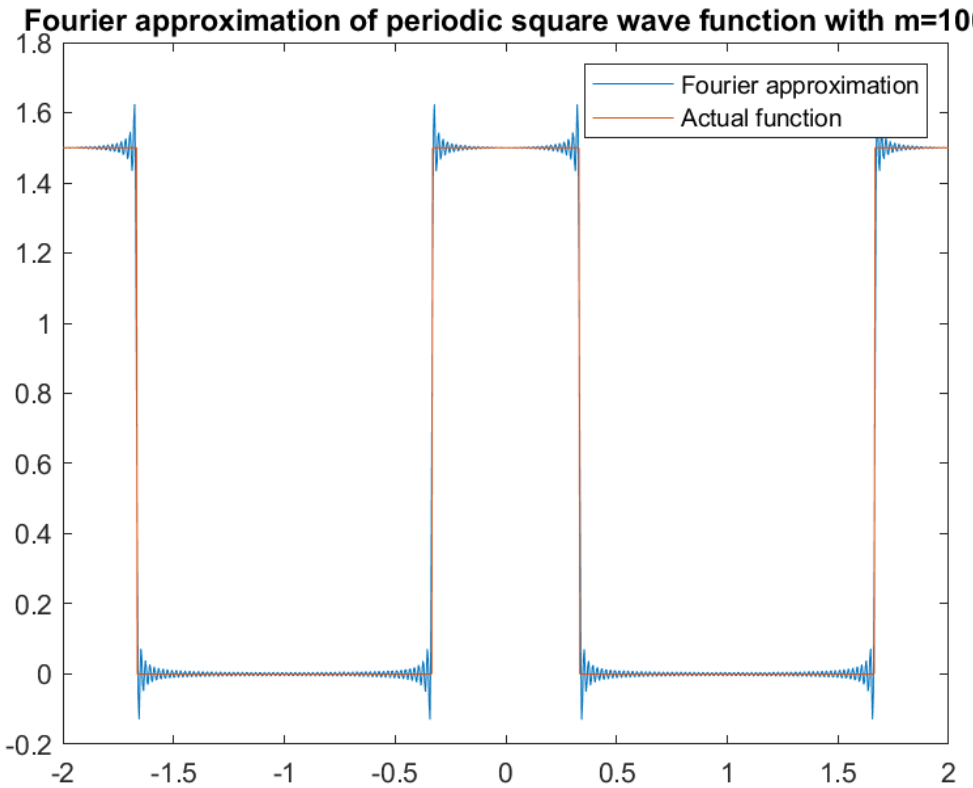
در تکه کد بالا، صرفا مقدار دوره تناوب، دامنه، و duty cycle مقدار دهی می‌شوند. همچنین برای رسم نمودار از $-T$ تا T، با استفاده از linspace، تعداد ۱۰۰۰ نقطه بین این دو مقدار تولید می‌شود.

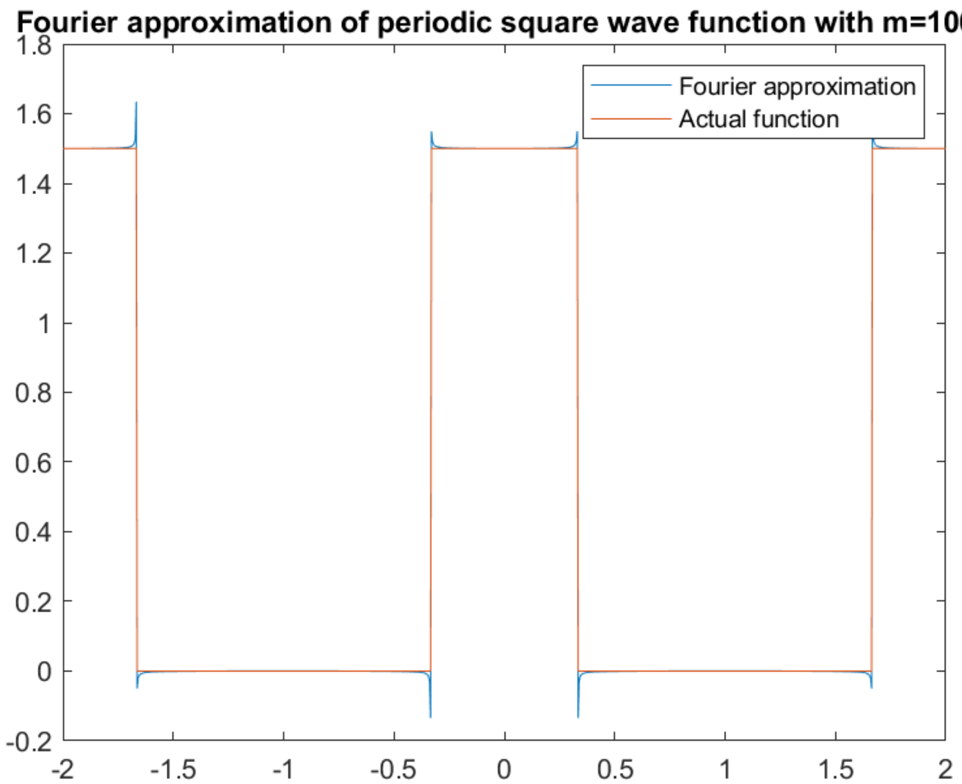
```
m = 100;

y_func = my_square_wave(T, A, D);
y_approx = fourier_approx(t, m, y_func, T);

figure;
plot(t, real(y_approx));

hold on;
plot(t, y_func(t));

title(strcat('Fourier approximation of periodic square wave function with m= ', num2str(m)))
legend('Fourier approximation','Actual function')
```

در بالا، تابع سیگنال برگردانده شده توسط my_square_wave در y_func ذخیره می‌شود. سیگنال حاصل از تقریب این تابع نیز در y_approx ذخیره می‌شود. در نهایت این دو تابع در یک صفحه رسم می‌شوند (کلیدواژه hold on برای رسم دو نمودار در یک صفحه استفاده می‌شود). همچنین از strcat برای چسباندن دو string، و از num2str برای تبدیل عدد به رشته استفاده شده است.

نتیجه رسم نمودار به صورت زیر است (با مقدار m=10,100,1000).





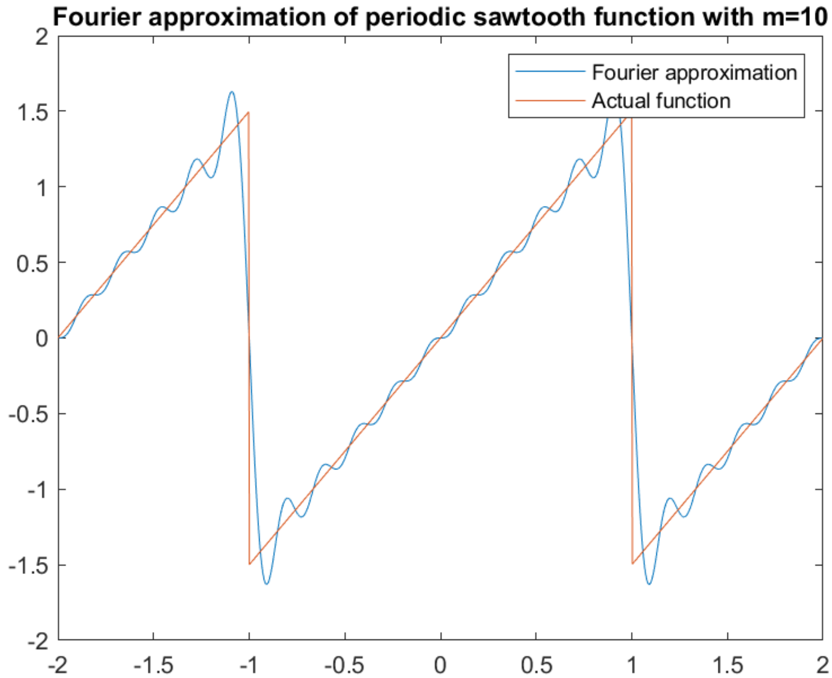


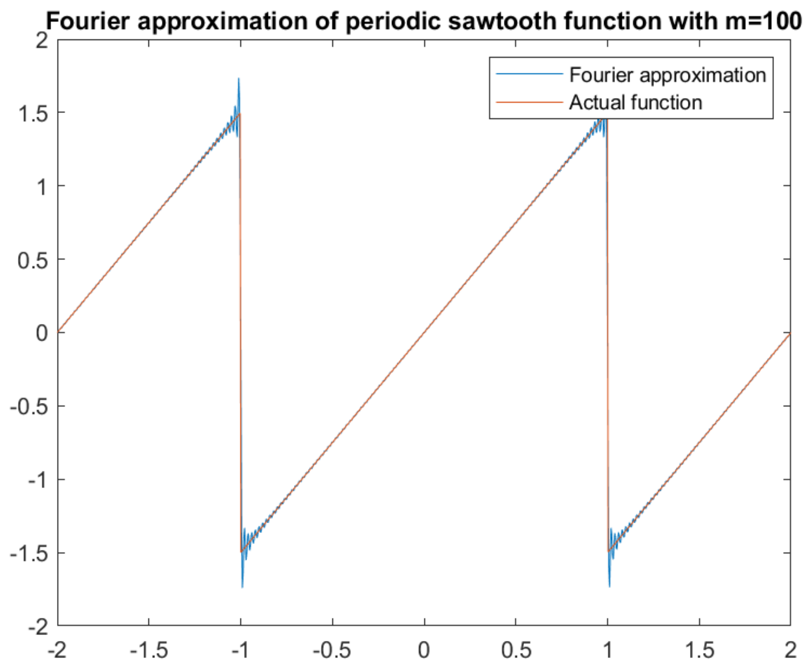
در نمودارهای بالا، سیگنال آبی تقریب فوریه، و سیگنال نارنجی سیگنال اصلی است.

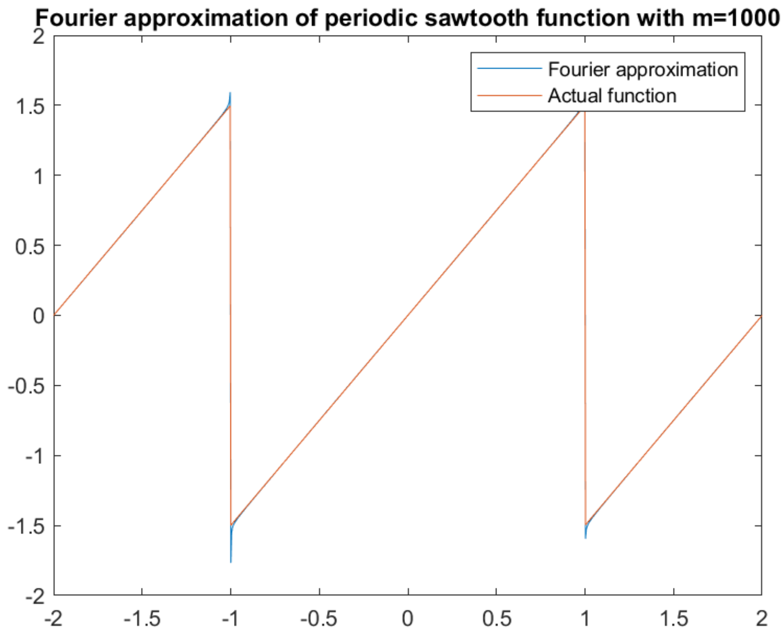
کد رسم سیگنال دندان‌اره‌ای، بسیار مشابه کد رسم سیگنال موج مربعی است.

```
m = 100;

y_func = my_sawtooth(T, A);
y_approx = fourier_approx(t, m, y_func, T);

figure;
plot(t, real(y_approx));

hold on;
plot(t, y_func(t));

title(strcat('Fourier approximation of periodic sawtooth function with m= ', num2str(m)))
legend('Fourier approximation','Actual function')
```

و حاصل رسم نیز به صورت زیر خواهد بود.







سوال ۲

پدیده گیبز، هنگامی رخ می‌دهد که سیگنال دارای ناپیوستگی باشد و بخواهیم این سیگنال را با تعداد محدودی از جملات سری فوریه تقریب بزنیم. در این صورت، در نقاط ناپیوستگی جهش ناگهانی خواهیم داشت. البته در تئوری، اگر تعداد بی‌نهایت جمله داشته باشیم، دیگر پدیده گیبز وجود نخواهد داشت.