پروژه نهایی دوره متلب مقدماتی
سریه فوریه و کاربرد ان در مهندس

ارایه دهنده : رومینا ریزه بندی

استاد : مهندس غلام پور

شهریور ۱۴۰۳

سریه فوریه چیست؟

مفهوم پایه در پیدایش سری‌های فوریه این است که توابع مختلف را می‌توان به کمک توابع سینوسی و کسینوسی بازنویسی کرد.این تابع به نام ریاضیدان فرانسوی، ژوزف فوریه نامگذاری شده‌است.

در واقع نشانن می دهد که هر تابع را می‌توان به کمک تعداد نامحدودی از توابع سینوسی و یا کسینوسی تولید کرد.

فرم کلی یک سری فوریه برای نمایش یک تابع به شکل زیر است.

$$f(x) = a_0 + \sum_{n=1}^{\infty}\left(a_n \cos\frac{n\pi x}{L} + b_n \sin\frac{n\pi x}{L}\right)$$

ر این رابطه، f(x)، تابعی را نشان می‌دهد که قرار است به کمک توابع سینوسی و کسینوسی بازنویسی شود. L، نصف دوره تناوب تابع را نمایش می‌دهد. a_n، a_0 و b_n نیز ضرایب سری فوریه هستند که شیوه محاسبه آن‌ها در این قسمت مورد مطالعه قرار می‌گیرد.

توجه کنید که نماد سیگما (∑) در عبارت دوم و سوم سمت راست معادله بالا، مجموع مقادیر عبارت مقابل سیگما را در nهای مختلف نمایش می‌دهد.

سریه فوریه چیست؟

ضرایب a_n، a_0 و b_n در رابطه سری فوریه به شکل زیر محاسبه می‌شوند.

T: دوره تناوب است .

$$\omega = 2.\frac{\pi}{T}$$

$$a_0 = \frac{1}{2.l}\int_{-l}^{l} f(x)dx$$

$$a_n = \frac{1}{l}\int_{-l}^{l} f(x).\cos(\omega.n.t)\,dx$$

$$b_n = \frac{1}{l}\int_{-l}^{l} f(x).\sin(\omega.n.t)\,dx$$

کاربرد سریه فوریه در مهندسی

یکی از مهم ترین کاربرد های سریه فوریه در سیستم_جرم _ فنر است(.شکل ۱)

این سیستم ساده ترین سیستم دینامیکی است که از یک جرم، یک فنر و یک میراگر تشکیل شده است .

برای نوشتن معادلهٔ حرکت این سیستم های دینامیکه روش های مختلفی وجود دارد. ما از اصل دالامبراستفاده می کنیم در این روش از قانون دوم نیوتن استفاده می شود که روابط تعادل دینامیکی رو مورد بررسی قرار میدهد.

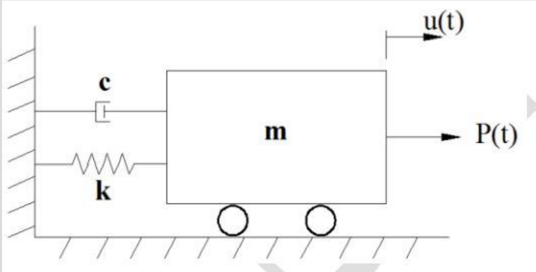
در این روش نمودار جسم آزاد سیستم در یک لحظهٔ دلخواه ترسیم می شود. سپس معادلهٔ تعادل نوشته خواهد شد.(شکل ۲)

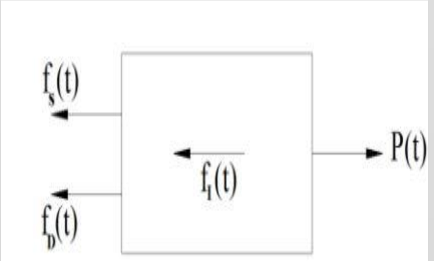
نیرو های مشخص شده :

$F_{I(t)}$=نیروی اینرسه $F_{D(t)}$ =نیروی میرایی $F_{S(t)}$=نیروی فنر

معادله تعادل: $F_{S(t)}+F_{I(t)}+F_{D(t)}$=P(t)

P(t)= نیروی خارجی





برای محاسبه مقادیر در ارتعاشات سیستم یک درجه ازادی اینگونه عمل می کنیم

بخش اول :

نیروی F(t) وارد شده بر جسم را با استفاده از فرمول گفته شده در اسلاید های قبل به صورت تابع سریه فوریه نمایش دهیم.

بخش دوم :

پس از بدست اوردن سری فوریه تابع F(T)با جایگزاری سریه فوریه در معادله

$$m\, x''(t) + c\, x'(t) + k\, x(t) = F(t)$$

یک معادله دیفرانسیل مرتبه دو ایجاد می شود.حال باید این معادله دیفرانسیل را حل کنیم .

روش حل معادله دیفرانسیل مرتبه دو

بخش اول:

در حالت کلی قالب یک معادله مرتبه دوم غیرهمگن به صورت زیر است:

$$m f''(x) + c f'(x) + k f(x) = d(x)$$

گر سمت راست همین معادله به جای $d(x)$ صفر قرار گیرد، معادله مفروض تبدیل به یک رابطه همگن خواهد شد و داریم:

$$m f''(x) + c f'(x) + k f(x) = 0$$

ابتدا این معادله همگن بدست امده را حل می کنیم. که $y_1(x)$ و $y_2(x)$، جواب های نهایی این معادله اند.

اگر $y_1(x)$ و $y_2(x)$، جواب‌های مستقل خطی از یک معادله همگن باشند، هر ترکیب خطی از این دو تابع نیز جوابی برای معادله دیفرانسیل مذکور خواهد بود.

بنابراین با فرض $y_1(x)$ و $y_2(x)$ هر کدام به تنهایی در معادله صدق کنند، می‌توان گفت تابع رو به رو نیز در معادله اصلی صادق خواهد بود. $Y=c1.y1+c2.y2$

بخش دوم :

بیان می کند که مقدار نهایی معادله دیفرانسل برابر است با:

جواب خاص معادله ناهمگن + جواب عمومی معادله همگن مرتبط با آن = جواب کلی معادله ناهمگن خطی

صورت سوال پروژه

از ما خواسته شده که در یک سیستم جرم - فنر ، شامل یک جرم m، با ثابت فنر k و ضریب میرایی c را مورد بررسی قرار دهیم . فرض کنید نیروی F(t) به جرم وارد شده است و با استفاده از قانون دوم نیوتون می توان عنوان معادله دیفرانسیلی نوشته شده است . m x"(t) + c x′(t) + k x(t) = F(t)

همچنین نیروی اعمالی را به صورت زیر تعریف می نماییم:

$$F(t) = \begin{cases} 0, & -1 < t \le 0 \\ 1, & 0 \le t \le 1 \end{cases}$$

با فرض m=1 , C=2 , K=0 به سوال های زیر پاسخ دهید:

1. دربازه ی زمانی، T=0:10 نمودار جابجایی سیستم را بر حسب زمان رسم کنید .
2. با استفاده از دستور subplot، در همان بازه زمانی، نیروی اعمالی را رسم کنید.
3. پاسخ بدست آمده برای x آیا با آنچه در واقعیت از این سیستم میتوان تصور نمود ، تطابق دارد ؟ چنانچه پاسخ شما به این سوال مثبت می باشد آن را توضیح دهید .

مهم ترین دستورات متلب در سیستم جرم _فنر _میرایی

تابع ناشناس

ویژگی این نوع تابع این هست که تنها یک دستور را اجرا می کنند و نام انها نام متغیری است که در آن ذخیره شده‌است.

ساختار این نوع تابع ایگونه است:
function_name = @(variable_name) matlab_expression

در آن باید به جای function_name، نام مورد نظرمان برای تابع را بنویسیم، به جای variable_name، نام متغیر ورودی و یا متغیرهای ورودی نوشته می شود و به جای عبارت matlab_expression هم باید عبارت تعریف تابع نوشته شود.

INTEGRAL()

در متلب برای انتگرال گیری از دستور integral () و همچنین برای گرفتن انتگرال معین در متلب تنها کافی است که بازه های ورودی و خروجی را اینگونه مشخص کنیم (,a,b) integral .

دستورOde45

Ode45دستوربهترین انتخاب برای حل انواع معادلات دیفرانسیل است و ساختار کلی ان به این صورت هستش. [t,y]=ode45(odefun,tspan ,y0)
که odefun: معادله دیفرانسیل در فرم تابع وtspan : بازه زمانی و y0 : شرایط اولیه

دستور ode45به عنوان ورودی، معادله دیفرانسیل را در فرم تابع می‌گیرد. یعنی بایستی معادله دیفرانسیل را با استفاده از تابع تعریف کنیم. تعریف تابع می‌تواند به صورت فانکشنال یا معمولی باشد.

توجه کنید که علگرهای odeدر متلب، فقط معادله دیفرانسیل مرتبه اول را حل می‌کنند. یعنی به عنوان ورودی باید فقط معادله دیفرانسیل مرتبه اول به آنها بدهیم. برای حل معادله دیفرانسیل با مرتبه بالاتر، بایستی یک معادله مرتبه nرا به n معادله مرتبه اول تبدیل کنیم.

دستورsubplot

دستور subplot(n,m,p)در نرم افزار متلب (Matlab)برای ترسیم چند نمودار در یک پنجره figureاستفاده می شود. این دستور پنجره ترسیمات یا figureرا به nسطر و mستون تقسیم کرده و دستور رسمی که در ادامه آن می آید را در مکان pام قرار می دهد به همین دلیل از این دستور استفاده کردیم

در ابتدای کد از سه دستور

clear all
close all,
Clc

استفاده شده.

که به ترتیب دستور ها برای حذف متغیرها در workspace , بستن تمام پنجره های باز شده با استفاده از برنامه متلب , حذف تمام مقادیر و مطالب موجود در command window است .
سپس یک متغیر با نام T برای ذخیره دوره تناوب تابع تعریف کردیم و با توجه به تابع صورت سوال به این نتیجه می رسیم که دوره تناوب ما برابر عدد ۲ است .

در سطر بعدی ما یک متغیر به نا omegaرا تعریف کردیم که در واقع همان بسامد زاویه ای یا سرعت زاویه ای می باشد که از فرمول omega $=(2.\pi)/T$ بدست می اید .

در سطر بعدی یک متغیر به نام n_term تعریف می کردیم برای انکه انتها بازه سری را مشخص کنیم درواقع تمام سری ها تا مقدار n_term ادامه پیدا می کند.

```
clear all %#ok
close all
clc
T = 2;
omega = 2 * pi / T;
n_term = 10;
```

در ادامه توابع an,bn,a0را به صورت تابع ناشناس تعریف کردیم . ویژگی این نوعتابع این هست که تنها یک دستور را اجرا می کنند و نام انها نام متغیری است که در آن ذخیره شده‌است.
اگر به کد هایی که برای تعریف توابع استفاده شده است نگاه کنیم .این عبارت را مشاهده می کنید double(t > 0 & t <= 1), استفاده از این عبارت برای تعریف تابع F(t) است .
درواقع ما یک شرط تعیین کردیم اگر این شرط برقرار شود به ۱ و در غیر این صورت صفر را نمایش می دهد .
اما اگر تنها عبارت داخل () را بنویسیم کد ما اشتباه است زیرا در نوع داده های تابع integral مشکل ایجاد می شود. در تابع یکپارچه سازی خروجی باید به صورت عددی (single /double) باشد اما در این حالت مشکل منطقی (logical)تعریف می شود. برای رفع این مشکل باید خروجی یک مقدار عدد داشته باشد به همین منظور از دستور double استفاده کردیم تایک قدار عددی داشته باشیم.

```
a0 = 1 / T * integral(@(t) double(t > 0 & t <= 1), 0, T);
an = @(n) 2 / T * integral(@(t) double(t > 0 & t <= 1) .* cos(omega * n * t), 0, T);
bn = @(n) 2 / T * integral(@(t) double(t > 0 & t <= 1) .* sin(omega * n * t), 0, T);
a_n = zeros(n_term, 1);
b_n = zeros(n_term, 1);
```

در ادامه دو ماتریس صفر برای دریافت مقادیر سری های a_n , b_n تعریف کردیم و مقادیر را با استفاده از یک حلقه for به ازای مقادیر مختلف nاز بازه۱ تا مقدار n-termمحاسبه کردیم و در ماتریس هایa_n , b_n مقادیر را ذخیره می کنیم .

```
a_n = zeros(n_term, 1);
b_n = zeros(n_term, 1);

for n = 1:n_term
    a_n(n) = an(n);
    b_n(n) = bn(n);
end
```

هدف این بخش از کد باسازی تابع F(t) به صورت یک جمع از توابع کسینوسی و سینوسی بر اساس ضرایب محاسبه شده در سری فوریه است .

در این بخش از کد با استفاده از دستور inspace مقادیر متغیر t را مشخص می کنیم .

1. شروع با a0/2

```
F_approx = @(t) a0 / 2 * ones(size(t));
```

این خط کد یک بردار ثابت به نام F_approx ایجاد می کند که اندازه اش برابر با بردار زمانی t است.

استفاده از a0/2 به دلیل این است که در فرمول سریه فوریه تابع مقدار اولیه هنگام جع کردن a0/2 هستش .

2. جمله بندی سریه فوریه:

این حلقه برای جمع کردن چند جمله سریه فوریه تا n_term بر روی F_approx اجرا می شود .

Nنشان دهنده شماره جمله ای است که در سری فوریه در نظر گرفته شده است.

a_n(n) * cos(n * omega * t) و b_n(n) * sin(n * omega * t) به F_approx اضافه می شوند تا تاثیر هارمونیک های کسینوسی و سینوسی بر نیروی تقریب زده شده F(t) ایجاد شود.

a_n ضریب کسینوسی است ونشان میدهد که چقدر از تابع کسینوسی n ام استفاده می شود .

b_n ضریب سینوسی است ونشان میدهد که چقدر از تابع سینوسی n ام استفاده می شود .

Omegaفرکانس زاویه ای پایه است که بر اساس دوره تابع محاسبه می شود و تعریف می کندکه هارمونیک بنیادی در چه فرکانسی است.

```
for n = 1:n_term
    prev_F_approx = F_approx;
    F_approx = @(t) prev_F_approx(t) + a_n(n) * cos(n * omega * t) + b_n(n) * sin(n * omega * t);
end
```

سه متغیر m, c , k را تعریف مکنیم و مقادیر مشخص شده در صورت سوال را برای ان ها در نظر می گیریم .
حال باید معادله دیفرانسیل سیستم جرم و فنر رو با استفاده از نیروی F(t) که با سریه فوریه تقریب زده شده محاسبه کنیم.

1. ابتدا معادله دیفرانسیل را تعریف می کنیم
Y(2)سرعت سیستم را که اولین عنصر در خروجی است نشان می دهد.
F_approx(t) - c * y(2) - k * y(1)) این عبارت نشان دهنده نیروی خالصی است که بر سیستم وارد میشود.این نیروی خالص شامل نیروی خارجی و نیروی میرایی و نیروی فنر است.
معادله نیوتن: F = ma که نیرو را به شتاب تبدیل مکند /m
با توجه به توضیحاتی که در اسلایدهای قبل دادیم علگرهای odeدر متلب، فقط معادله دیفرانسیل مرتبه اول را حل می‌کنند. یعنی به عنوان ورودی باید فقط معادله دیفرانسیل مرتبه اول به آنها بدهیم. برای حل معادله دیفرانسیل با مرتبه بالاتر، بایستی یک معادله مرتبه n را به nمعادله مرتبه اول تبدیل کنیم. .
برای نمونه در این پروژه برای یک معادله مرتبه دو این معادله‌با تغییر متغیر به دو معادله مرتبه اول تبدیل شده‌است. حال این دو معادله را درون یک تابع در متلب نوشته و ذخیره می‌کنیم.

```
d2y_dt2 = @(t, y) [y(2); (F_approx(t) - c * y(2) - k * y(1)) / m];
```

2.شرایط اولیه و بازه زمانی
y0 شرایط اولیه سیستم که شامل جابجایی اولیه و سرعتاولیه است را ذخیره می کند.
Tspan بازه زمانی را نمایش می دهد.

```
y_0 = [0; 0];
tspan = [0, 10];
```

3. حل معادله دیفرانسیل با ode45

با توجه به توضیحاتی که در اسلاید های قبل داده شد در این قسمت تنها کافی است که از دستور ode45 استفاده کنیم تا معادله دیفرانسیل حل شود.

```
[t, y] = ode45(d2y_dt2, tspan, y_0);
```

d2y_dt2تابعی است که معادله دیفرانسیل را ارایه می دهد.
y0 شرایط اولیه سیستم که شامل جابجایی اولیه و سرعتاولیه است را ذخیره می کند.

Tspan بازه زمانی را نمایش می دهد.

حال که معادله دیفرانیل را با استفاده از کد ها قبل حل کردیم و مقادیر جابه جایی را بدست اوردیم می توانین جابه جایی سیستم بر حسب زمان را رسم کنیم :

```
subplot(2, 1, 1)
plot(t, y(:, 1))
title("جا به جایی سیستم بر حسب زمان")
xlabel("زمان (s)")
ylabel("جا به جایی (m)")
```

در قسمت دوم صورت سوال از ما خواسته شده است نمودار نیروی اعمالی F(t) را رسم کنیم :
ابتدا F_values = F_approx(t) تا به ازای tهای مختلف مقدار نهایی را در متغیری به نام F_values نمایش دهد سپس به هما صورت قبلی با استفاده از دستور subplot plot نمودار خواسته شده را رسم می کنیم.

```
F_values = F_approx(t);

subplot(2, 1, 2)
plot(t, F_values)
title("نیروی اعمالی بر حسب زمان")
xlabel("زمان (s)")
ylabel("نیرو (N)")
```

در قسمت سوم صورت سوال از ما خواسته شده میزان تطابق با سیستم واقعیت را بررسی کنیم :

به همین دلیل باید به نکات زیر توجه کنیم:

1. رفتار دینامیکی: اگر پاسخ سیستم رفتار نوسانی داشته باشد که به مرور زمان کاهش یابد نشان دهنده وجود میرایی مناسب در سیستم است . این موضوع با رفتار واقعی سیستم های فیزیکی تطابق دارد .

2. پاسخ به نیروی اعمالی: اگر پاسخ جابجایی به نیروی اعمالی مشابه بوده و با دامنه و فرکانسی مشابه نوسان داشته باشد، نشان می‌دهد که مدل به درستی نیروهای اعمالی را شبیه‌سازی می‌کند. بررسی کنید که قله‌ها و فرکانس‌های نیروی اعمالی و جابجایی با یکدیگر تطابق دارند یا خیر.

3. پارامترهای فیزیکی: اطمینان حاصل کنید که پارامترهای فیزیکی سیستم (مثل جرم m، ضریب میرایی c c، و ثابت فنر k (در مقادیر صحیح و معتبر تنظیم شده‌اند.اگر داده‌هایز سیستم واقعی داشتیم و این پارامترها را با مقادیر استفاده شده در شبیه‌سازی مقایسه کنید. اگر تمامی شرایط و پارامترها به درستی تنظیم شده باشند و پاسخ شبیه‌سازی با رفتار انتظاری سیستم فیزیکی مطابقت داشته باشد (از نظر دامنه، فرکانس، و شکل پاسخ)، می‌توان گفت که پاسخ شبیه‌سازی با واقعیت تطابق دارد.

منابع:

1. https://www.matlabplus.com
2. Georgi P. (1976). *Fourier Series*. Courier-Dover. ISBN
3. https://matlab1.ir
4. Erwin kreyszing,"Advanced Engineering Mathematics",tenth edition,2011
5. Michael Greenberg,"Advanced Engineering Mathematics",2nd edition