# مدل سازی فضای حالت در متلب

# (State Space modeling in Matlab)

*Ali Aghayifar*

این آموزش رو با یک مثال ارائه خواهیم کرد
یک چرخ فلک رو در نظر بگیرد این چرخ و فک رو تحت یک سیستم درجه دو مدل سازی می کنیم و با شرط اینکه ما شرایط اولیه ای غیر صفر داریم :

معادلات حالتی شبیه زیر انتظار می ره :

$\theta' = \mathbf{A}\,\theta + \mathbf{B}\,u$

$y = \mathbf{C}\,\theta + \mathbf{D}\,u$

که در اینجا Y خروجی دلخواه ما هست . خوب زیاد وارد روابط فیزکی و علت این روابط نمی شویم و جلو می رویم فقط در حال حاضر رابطه های زیر را قبول کنید :
میزان اینرسی :

$\frac{1}{2} I\omega^2$

به طوری که  طبق روابط ریاضی $\omega = \theta'$.  است و

$\frac{1}{2} k\theta^2$

در واقع از اینجا کار می شروع می شود این یک رابطه بین متغیر های ما است .

$k\theta + b\omega + I\omega' = 0$

می توان $\omega'$ را از رباطه فوق به صورت زیر ساده کرد :

$\omega' = -b/I\ \omega - k/I\ \theta$

خروجی مطلوب ما دقیقا مقدار $\theta$ است یعنی برای خروجی دلخواه ما داریم :

$y = \theta$

پس روابط فوق را یک بار دیگر برای شروع کار می نویسم :

$\theta' = \omega$

$\omega' = -k/I\ \theta - b/I\ \omega$

$y = \theta$

**بردار حالت ما از دو متغیر $\omega$ و $\theta$ تشکیل شده است این متغیر های این هستند پس ماتریس بردار ما بصورت زیر خواهد بود :**

$$\dot{\vec{\mathbf{x}}} = \mathbf{A}\vec{\mathbf{x}} + \mathbf{B}\mathbf{u}$$
$$\mathbf{y} = \mathbf{C}\vec{\mathbf{x}} + \mathbf{D}\mathbf{u}$$

به طوری که بردار حالت ما است :

$$\vec{\mathbf{x}} = \begin{bmatrix} \theta \\ \omega \end{bmatrix}$$

اولین سطر A و اولین سطر B ضرایب مورد نظر ما برای $\theta'$ و سطر دوم A و سطر دوم B ضرایب **$\omega'$ هستند . C و D نیز ضرایب خروجی ما هستند .**

$$\dot{\vec{\mathbf{x}}} = \begin{bmatrix} 0 & 1 \\ -k/I & -b/I \end{bmatrix}\vec{\mathbf{x}} + \begin{bmatrix} 1 \\ 1/I \end{bmatrix}\mathbf{u}$$
$$\mathbf{y} = \begin{bmatrix} 1 & 0 \end{bmatrix}\vec{\mathbf{x}} + \begin{bmatrix} 0 \end{bmatrix}\mathbf{u}$$

$$\vec{\mathbf{x}}(0) = \begin{bmatrix} 0 \\ \omega(0) \end{bmatrix}$$

**بنابر این داریم :**

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -k/I & -b/I \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} 1 \\ 1/I \end{bmatrix}$$

$$\mathbf{C} = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

$$\mathbf{D} = \begin{bmatrix} 0 \end{bmatrix} \qquad \vec{\mathbf{x}}(0) = \begin{bmatrix} 0 \\ \omega(0) \end{bmatrix}$$

حال بیایید این متغیر ها را وارد Matlab کنیم :

بعد از اعلان آمادگی Matlab می توانیم مقادیر دلخواه را برای ثابت های خود را وارد کنیم این ثابت های با توجه به ساختار سیستم ما تعیین می شوند . فرض ما بر مقادیر زیر خواهد بود :

I = 1,500,000kgm$^2$

b = 200,000Nm/s

k = 4,000,000Nm

پس وارد می کنیم :

```
>> I = 1500000;

>> b = 200000;

>> k = 4000000;

>> A = [ 0  1 ; -k/I  -b/I ];

>> B = [ 0 ; 1/I ];

>> C = [ 1  0 ];

>> D = 0;

>> second_ss = ss(A, B, C, D)
```

چیزی جدید که اینجا می بینید تابع () ss است بگذارید اندکی در مورد آن توضیح دهیم :

Ss برای مشخص کردن فضای حالت یا تبدیل مدل های تغییر ناپذیر با زمان به فضای حالت است . از ss برای تولید مقادیر حقیقی یا موهومی فضای حالت ( که SS objects نامیده می شوند ) یا مبدل تابع تبدیل استفاده می شود .

```
Bioemm = ss(a,b,c,d)
```

دستور فوق مدل فضای حالت یک پیوستگی زمانی است

$$\dot{\bar{x}} = A\bar{x} + Bu$$
$$y = C\bar{x} + Du$$

اگر فرض کنیم که N حا لت و Ni تا ورودی و No تا خروجی داریم آنگاه

A یک ماریس N*N

B یک ماتریس N*Ni

C یک ماتریس No*N

D یک ماتریس No*Ni

ممکن است در بعضی اوقات نیاز به نوشتن

```
bioemm = ss(a,b,c,d,'Property1',Value1,...,'PropertyN',ValueN)
```

باشد در این حالت هر عبارت که در '' قرار میگرد برای حالت بعدش خود یک نام هست و این دو با هم جفت خواهند گشت .

ممکن است گاهی ما بخواهیم که از یک تابع تبدیل معادل فضای حالت بگیریم به فرمان زیر دقت کنید :

```
bioemm_ss = ss(TF)
```

با فرمان فوق تابع تبدیل TF ما از مبدل فضای حالت گذشته و SS object ها در متغیر bioemm_ss ذخیره می شوند .این کار به state-space realization معروف می باشد [1] .

به طور مثال بیایید یک تابع تبدیل تعریف کنیم :

```
Bio=[tf([1 1],[1 3 3 2])]
```

با وارد کردن عبارت فوق در Matlab می بیند :

```
:Transfer function
s + 1
---------------------
s^3 + 3 s^2 + 3 s + 2
```

حال تایپ کنید ss(Bio ) و به جواب های ان دقت کنید .

[1] - این کار توسط تابع tf2ss نیز قابل انجام است

ماتریس های A, B, C, D که ضرایب حات ما هستند نمایش داده می شوند .
وارد ادامه بحث می شویم :
تا اینجا آمده بودیم که

```
second_ss = ss(A, B, C, D)
```

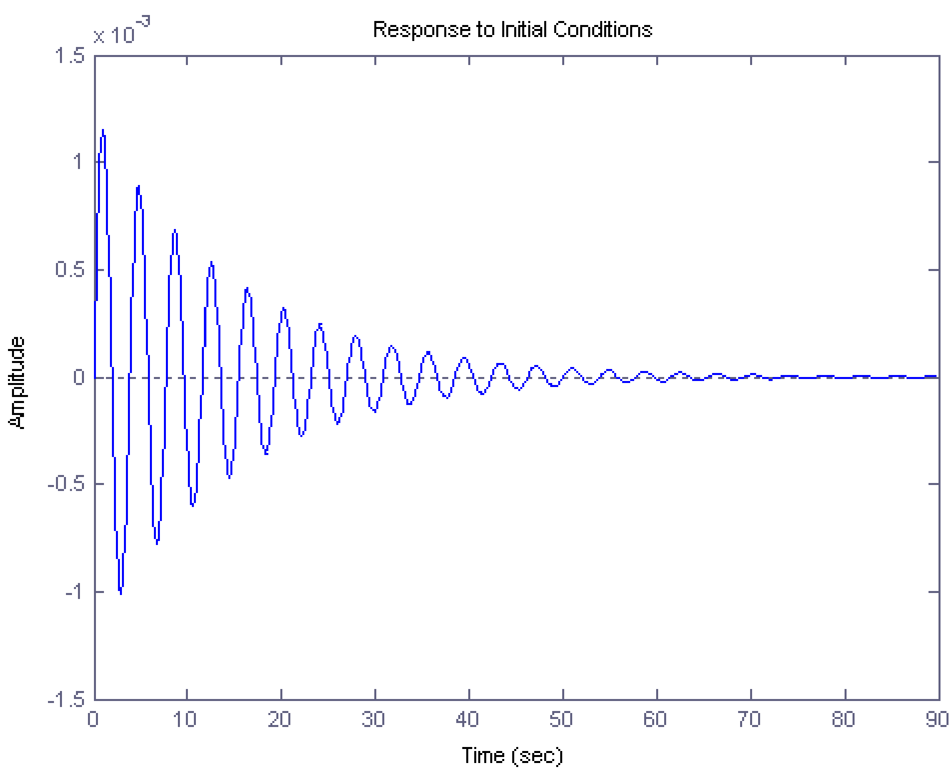
ما می خواهیم پاسخ مدار به شرایط اولیه را که فرض کرده بودیم غیر صفر هست و مشاهده کنیم :

```
x0 = [ 0; 0.002];
```

بردار x0 را به عنوان شرایط اولیه خود تعریف کردیم و باید به گونه ای این شرایط اولیه را به second_ss که در واقع فضای حالت ما را تشکیل می دهد اعمال کنیم از روش زیر استفاده می کنیم :

```
initial(second_ss, x0)
```

با وارد کردن دستور فوق نمایشی از خروجی سیستم ما با گذشت زمان به ازای شرایط اولیه ما نشان داده می شود.




می توان دیاگرام Bode را برای فضای حالت فوق نیز رسم نمود :

تایپ کنید :

```
>> bode(second_ss)
```