



به نام خدا



دانشگاه تهران
دانشکده مهندسی برق و کامپیوتر

ابزار دقیق

استاد نیری

تمرین ۱

نام و نام خانوادگی	محمد مشرقی
شماره دانشجویی	810199492
تاریخ ارسال گزارش	

Contents

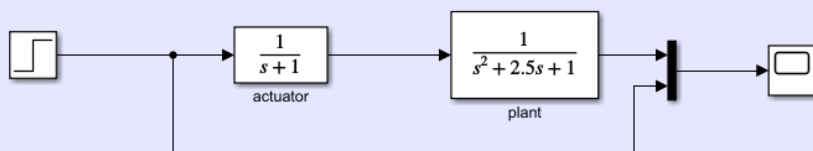
٢.....	١-
٣.....	الف
٤.....	ب
٥.....	ج
٧.....	٢
١٠.....	٣
١٠.....	الف
١١.....	ب
١٣.....	د
٢٠.....	٤
٢٠.....	الف
٢٣.....	قسمت ب
٢٤.....	ج
٢٦.....	٥-
٢٩.....	٦
٣٠.....	ب

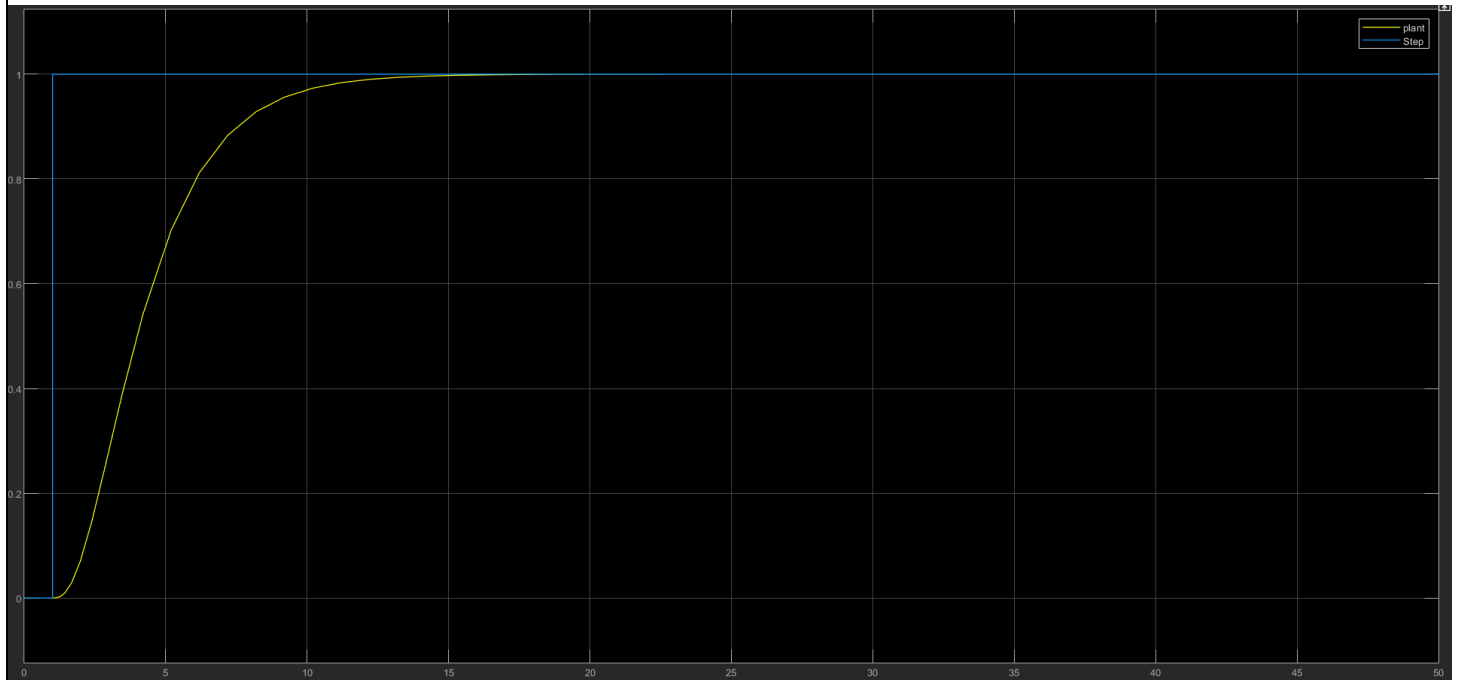
١-

الف

?

1.a





Block Parameters: chaus

Step

Output a step.

Main Signal Attributes

Step time:

1

Initial value:

0

Final value:

0.1

Sample time:

12

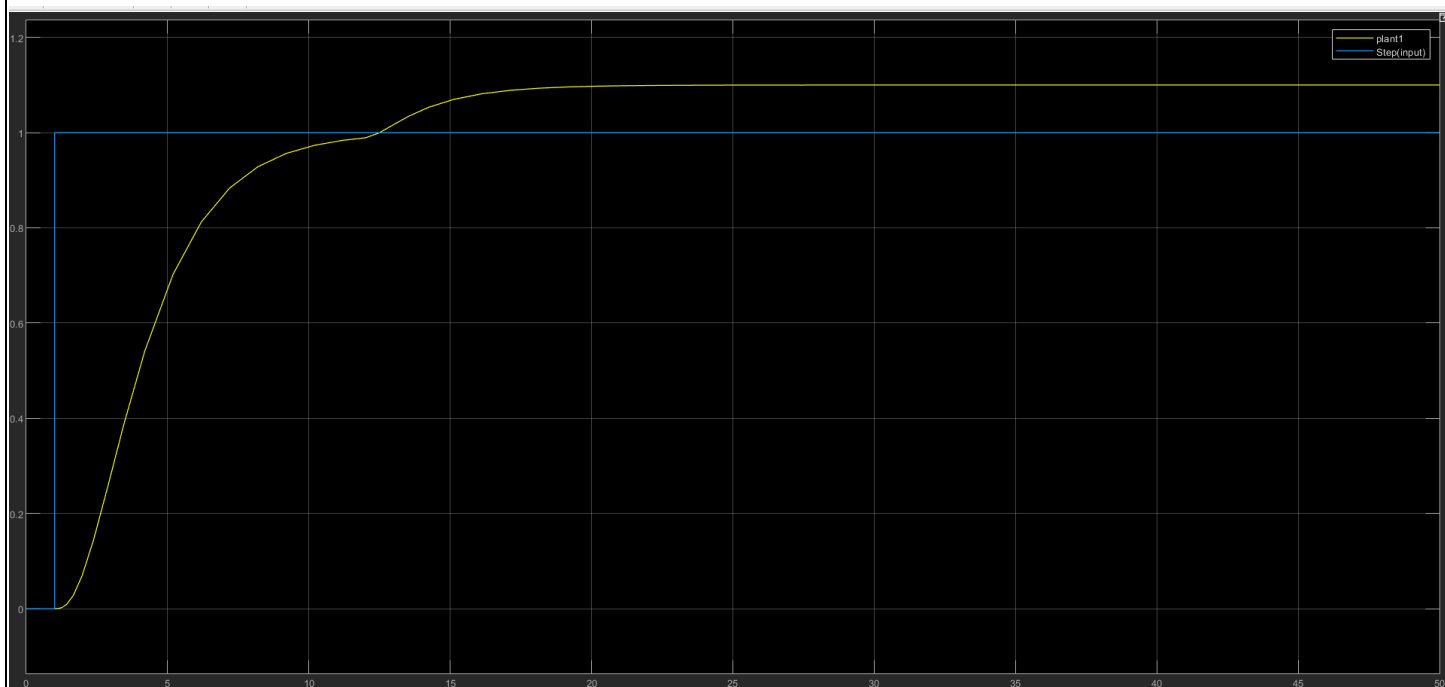
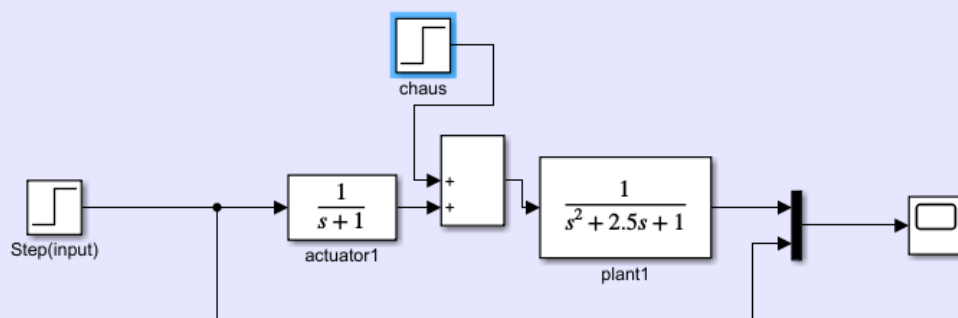
☒ Interpret vector parameters as 1-D

☒ Enable zero-crossing detection

OK Cancel Help Apply

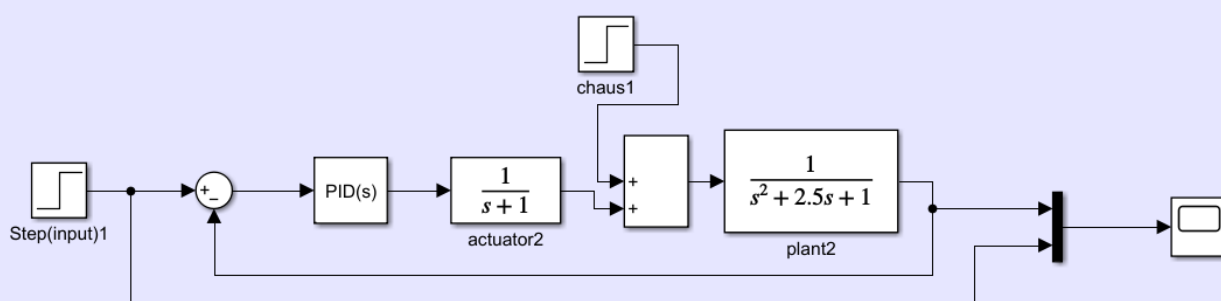
ب
مشخصات اغتشاش گر:

1.b



2

1.c



در این قسمت با اضافه کردن کنترل PID و همینطور فیدبک دار کردن سیستم به مدار شکل بالا تبدیل شد.

حال با بازی کردن با پارامتر های PID به مقادیر زیر می رسیم:

Controller parameters

Source: internal

Proportional (P): 3

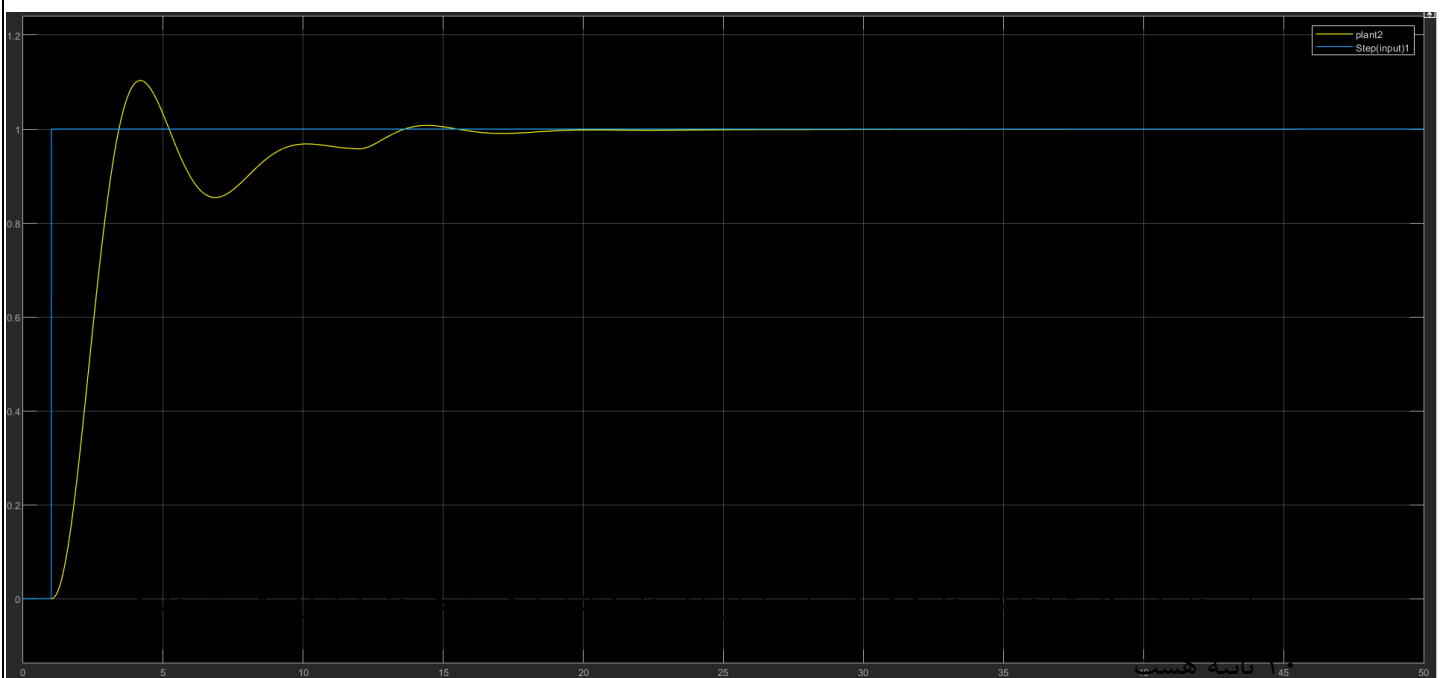
Integral (I): 0.5

Derivative (D): 0.5

☒ Use filtered derivative

Filter coefficient (N): 100

نمودار نهایی:



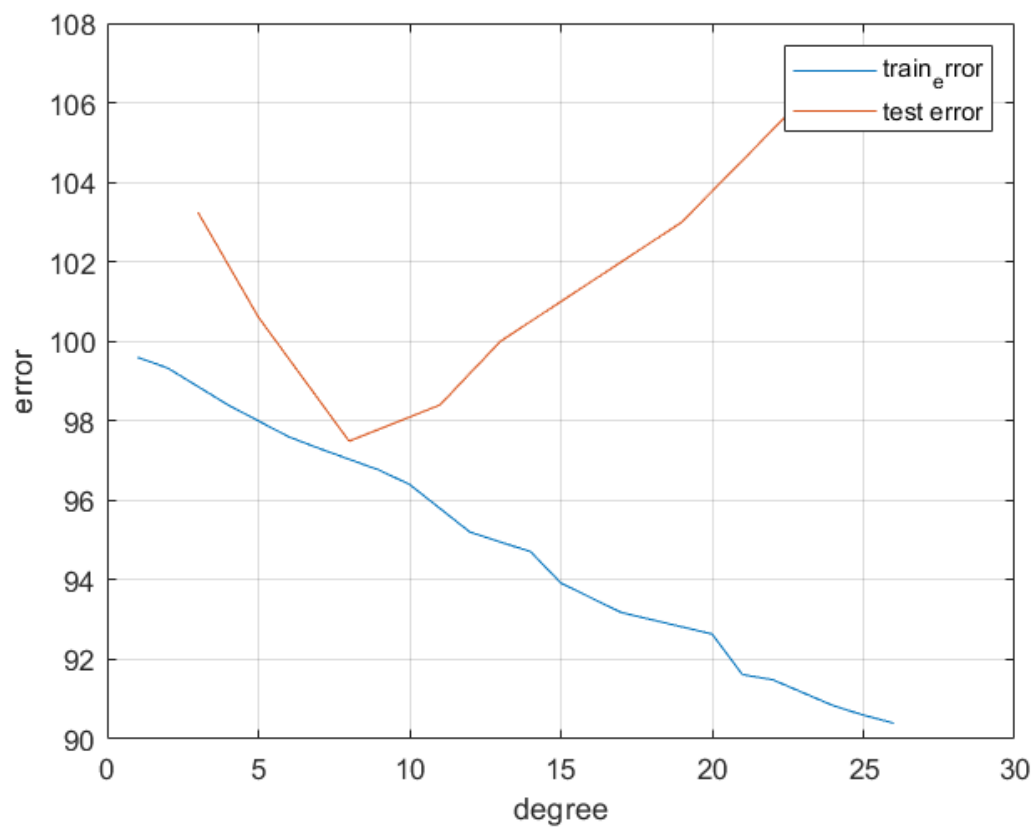
همچنین اگر بخواهیم با توجه به نیاز های سوال طراحی کنیم می توانیم تغییرات زیر رو اعمال کنیم:

زمان تثبیت: ضریب کسری تناسبی (K_p) به زمان تثبیت تأثیر می گذارد زیرا تعیین می کند که کنترل کننده چه میزان به سیگنال خطای کنونی پاسخ می دهد. افزایش مقدار K_p منجر به کاهش زمان

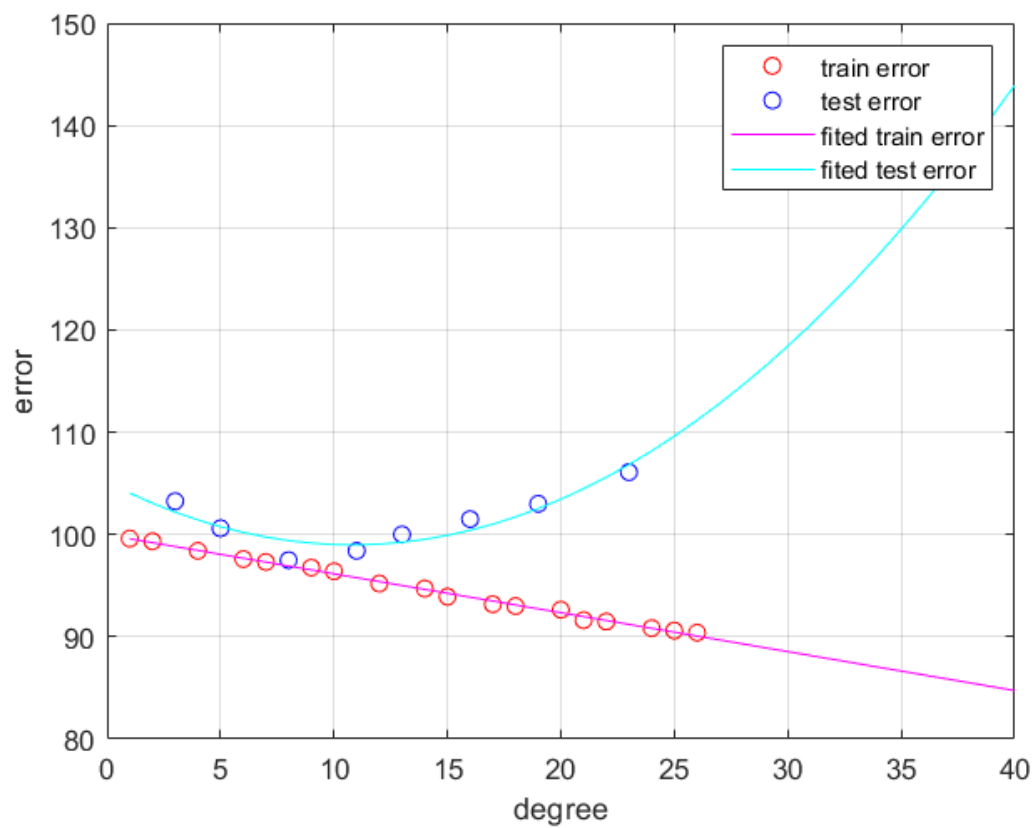
تثبیت می‌شود، اما K_p بالایی باعث بروز بیش‌انحرافی و عدم پایداری می‌شود که منجر به افزایش زمان تثبیت می‌شود. ضریب کسری تکمیلی (K_i) به زمان تثبیت تأثیر می‌گذارد زیرا تعیین می‌کند که چه مقدار پاسخ کنترل‌کننده به تجمع خطاهای گذشته داده شود. افزایش مقدار K_i می‌تواند به کنترل‌کننده در حذف خطاهای وضعیت پایدار کمک کند که باعث کاهش زمان تثبیت می‌شود. در نهایت، ضریب کسری مشتق (K_d) به زمان تثبیت تأثیر می‌گذارد زیرا به سرعت به تغییرات ناگهانی در سیگنال خطای پاسخ می‌دهد و به کنترل‌کننده در دستیابی به وضعیت پایدار کمک می‌کند.

بیشانه‌یابی (overshoot): بیشانه‌یابی میزانی است که کمیت سیستم بیش از مقدار پایدار نهایی خود قبل از خاموش شدن دارد. ضریب تکثیری (K_p) بیشانه‌یابی را به شدت تحت تأثیر قرار می‌دهد، به طوری که افزایش ارزش K_p منجر به پاسخ بیش از حد خشن و بیشانه‌یابی می‌شود. ضریب انباشته‌کننده (K_i) نیز می‌تواند به کاهش بیشانه‌یابی اثر بگذارد، با کاهش خطاهای حالت پایدار. در نهایت، ضریب تفاضلی (K_d) با کاهش پاسخ سیستم به نویزهای فرکانس بالا، می‌تواند بیشانه‌یابی را کاهش دهد.

زمان بالا رفتن: زمان بالا رفتن، زمانی است که سیستم برای رسیدن به حالت پایدار از حالت اولیه به حالت نهایی، نیاز دارد. ضریب پروپورشنال گین (K_p) از بیشترین تأثیر بر زمان بالا رفتن دارد و با تعیین واکنش کنترلر به تغییرات در سیگنال خطا، مشخص می‌شود. افزایش K_p می‌تواند باعث شود سیستم بهتر و پاسخگوتر شود و زمان بالا رفتن کمتر شود. با این حال، ضریب K_p بیش از حد بالا می‌تواند باعث افتال و ناپایداری و در نتیجه زمان بالاتری برای رسیدن به حالت نهایی شود. اثر انتگرال گین (K_i) همچنین می‌تواند بر زمان بالا رفتن تأثیر بگذارد، با اصلاح خطاهای حالت پایدار، که ممکن است بر زمان پاسخ کلی سیستم تأثیر بگذارد. ضریب دیفرانسیل گین (K_d) تأثیر کمتری بر زمان بالا رفتن دارد و عمدتاً به تأثیر سیستم در پاسخ به تغییرات ناگهانی در سیگنال خطا می‌پردازد.



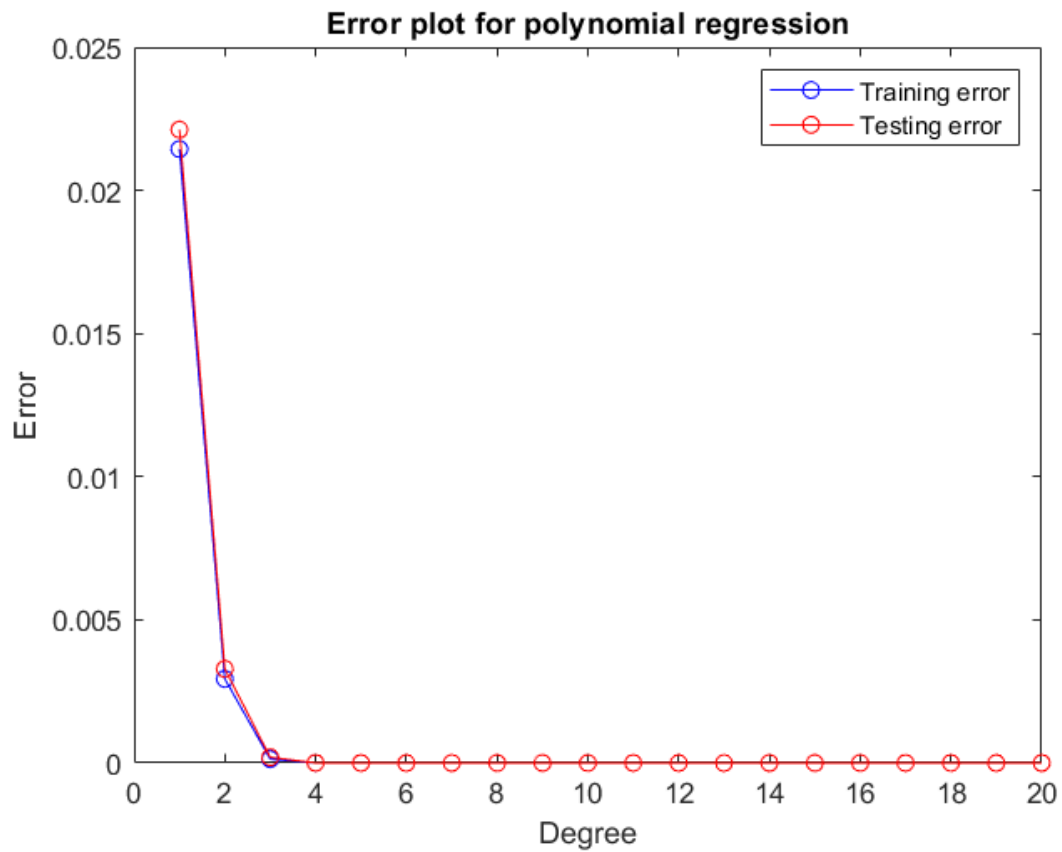
اما اگر داده رو ببینیم این دو داده کامل نیستن و به ازای بعضی از درجه ها داده نداریم پس باید برا درجه هایی که داده نیست پیش بینی کنیم برای اینکار هم داده های ترین و هم داده های تست رو فیت می کنیم با توجه به شکل برا ترین درجه یک و برا تست درجه دو نیاز است داریم:



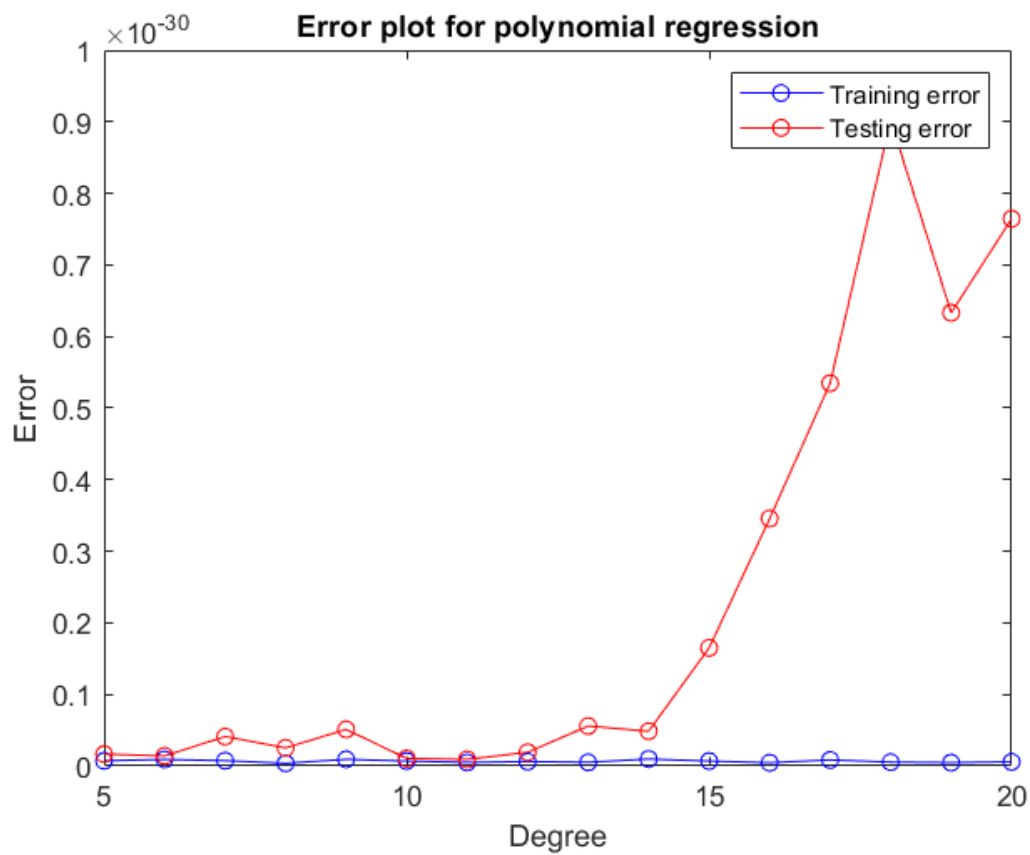
در درجه برابر با ۸ داده های تست کمترین خطا رو داشته اند و به خط ارور ترین هم نزدیک ترین فاصله رو دارند از این رو درجه بهترین درجه هست.

الف

ابتدا داده ها رو تولید و سپس دسته بندی می کنیم و از درجه یک تا بیست نمودار شونو رسم می کنیم.
داریم:



با توجه به اینکه درجه یک و دو خطای زیادی دارند آن دو رو حذف و دوباره رسم می کنیم.



حال طبق تصویر درجه ۶ بهترین هست چون کمترین خطا و اینکه ترین و تست به هم نزدیک شدن

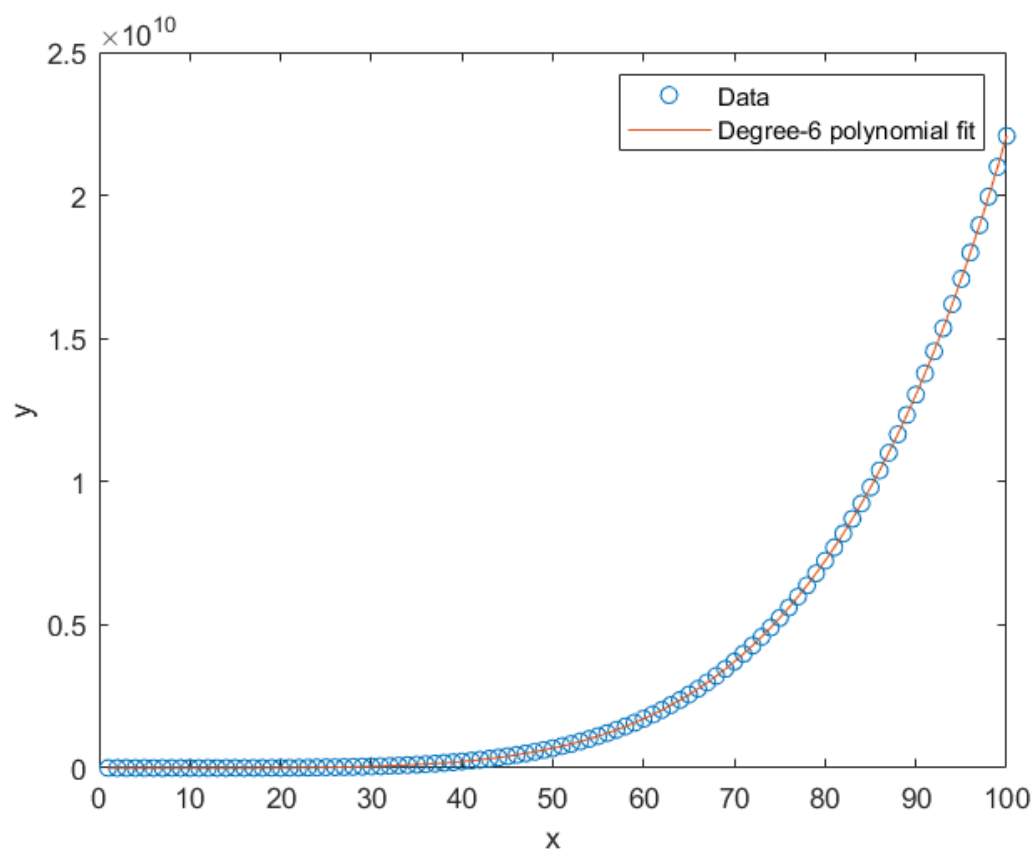
ب

درجه مناسب از بخش قبل که ۶ هست رو داریم

حال اگر طبق فرمول جلو ببریم داریم: روش LS

ضرایب به شرح روبه رو هست :

```
coeffs = 7x1
3.0000
7.2500
9.3333
0.5000
0.7500
2.2000
-0.0000
```

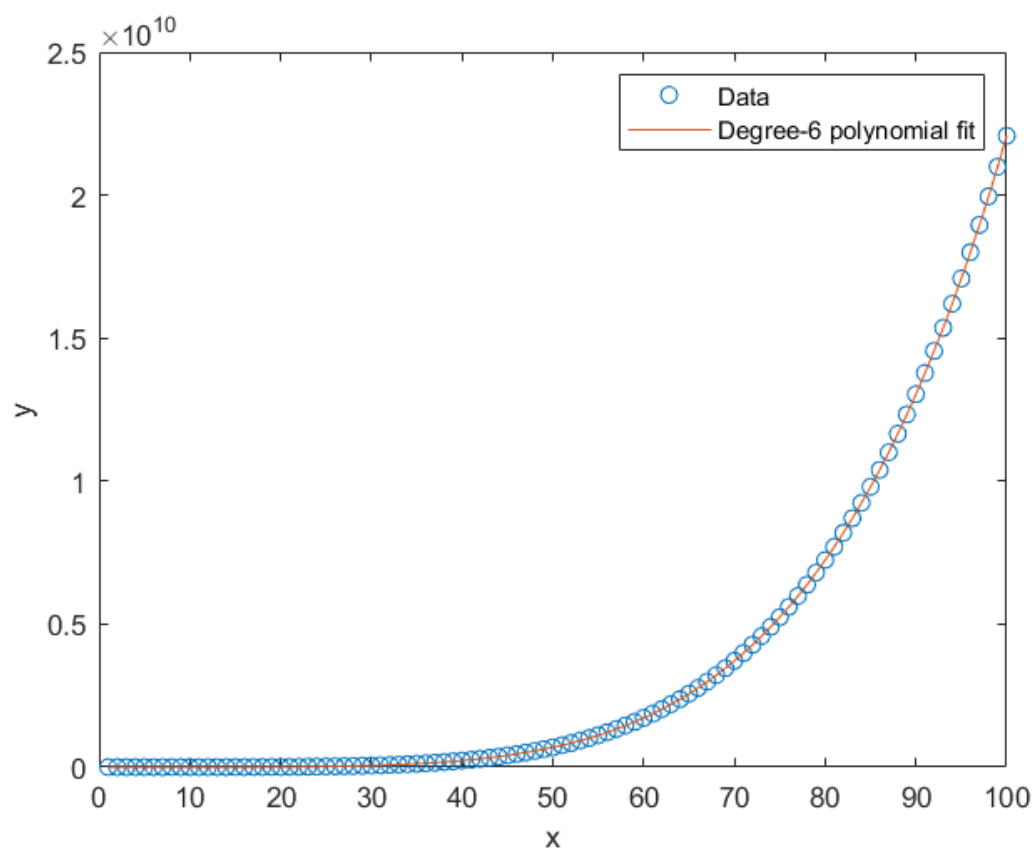


و این هم نمودار فیت شده .

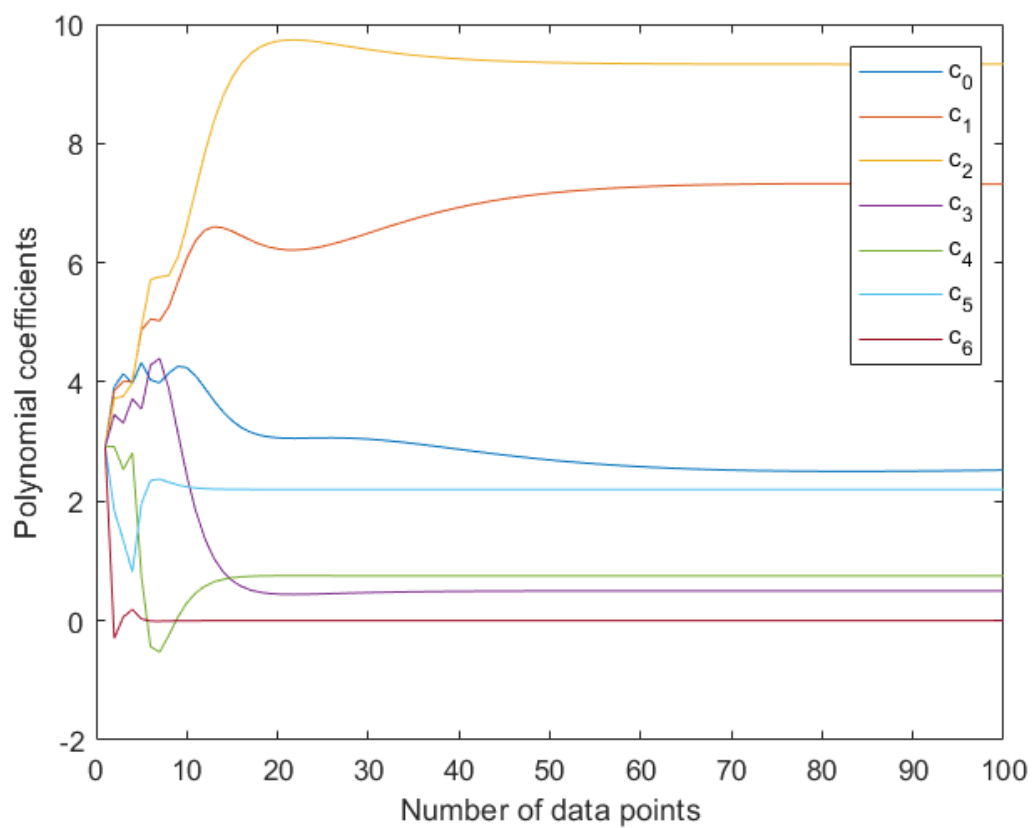
د

در اینجا از روش RLS استفاده می کنیم:

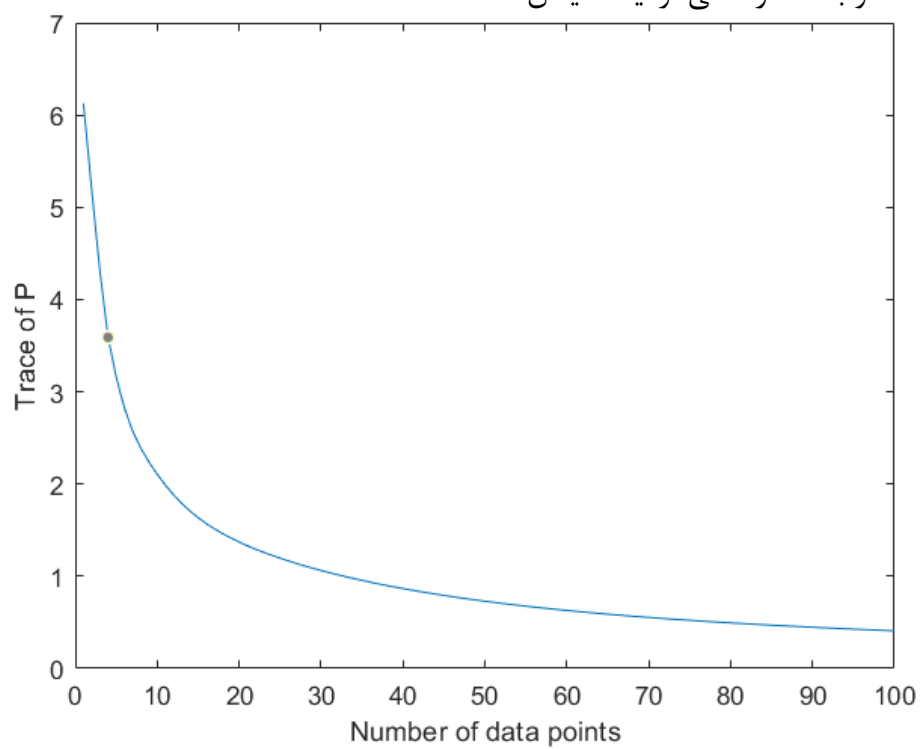
```
theta = 7×1
2.4511
7.3224
9.3300
0.5001
0.7500
2.2000
-0.0000
```



نمودار ضرایب داده بر حسب تعداد داده:



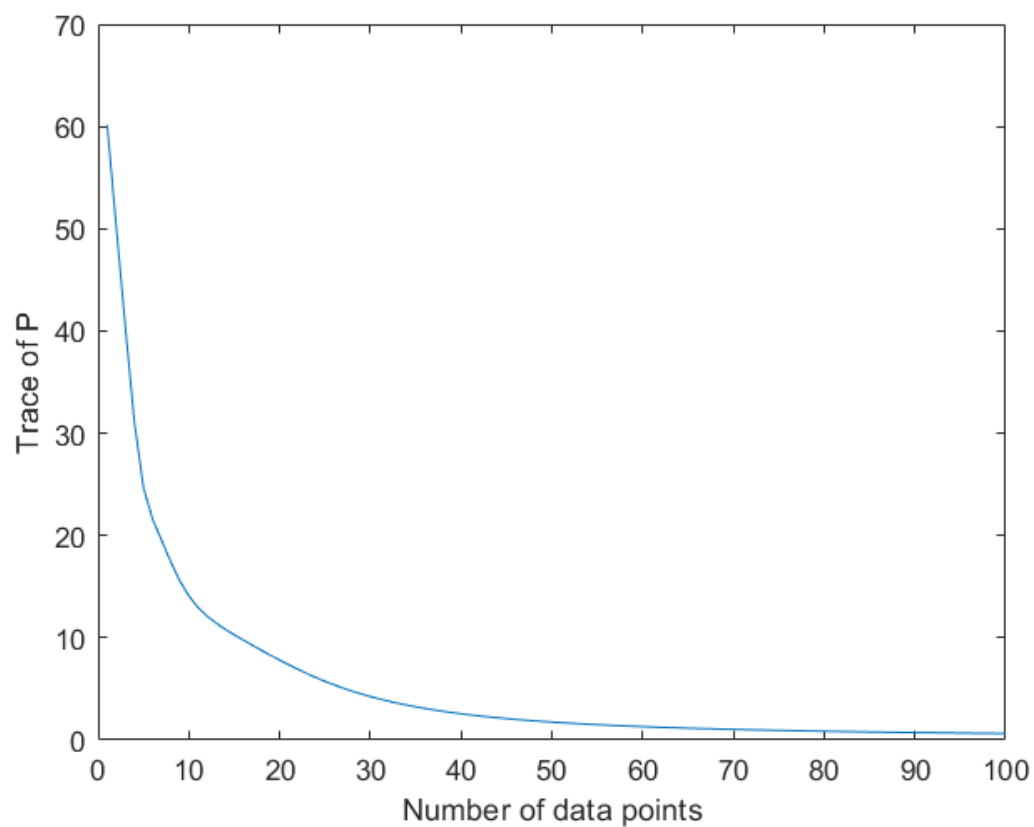
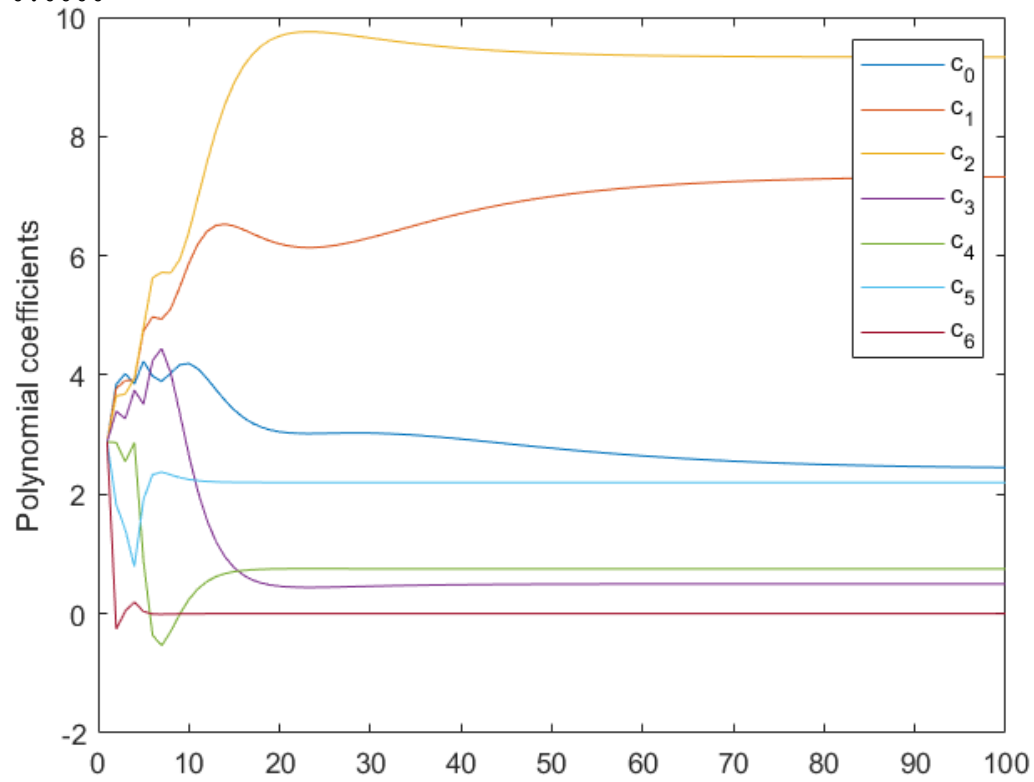
همانطور که در نمودار می بینیم در ابتدا مقدار تغییرات زیاد بوده اما با بیشتر شدن داده ها سرعت تغییرات کم شده و به مقدار اصلی نزدیک میشن.



وقتی تعداد داده زیاد میشه تریس ماتریس p کم و کمتر می شود که متوجه می شویم که مقادیر روی قطر اصلی کوچک تر شدن بنابراین طبق فرمول $k(m+1)$ کوچکتر شده و بعد از آن $tetha(m+1)$ کوچک می شود.

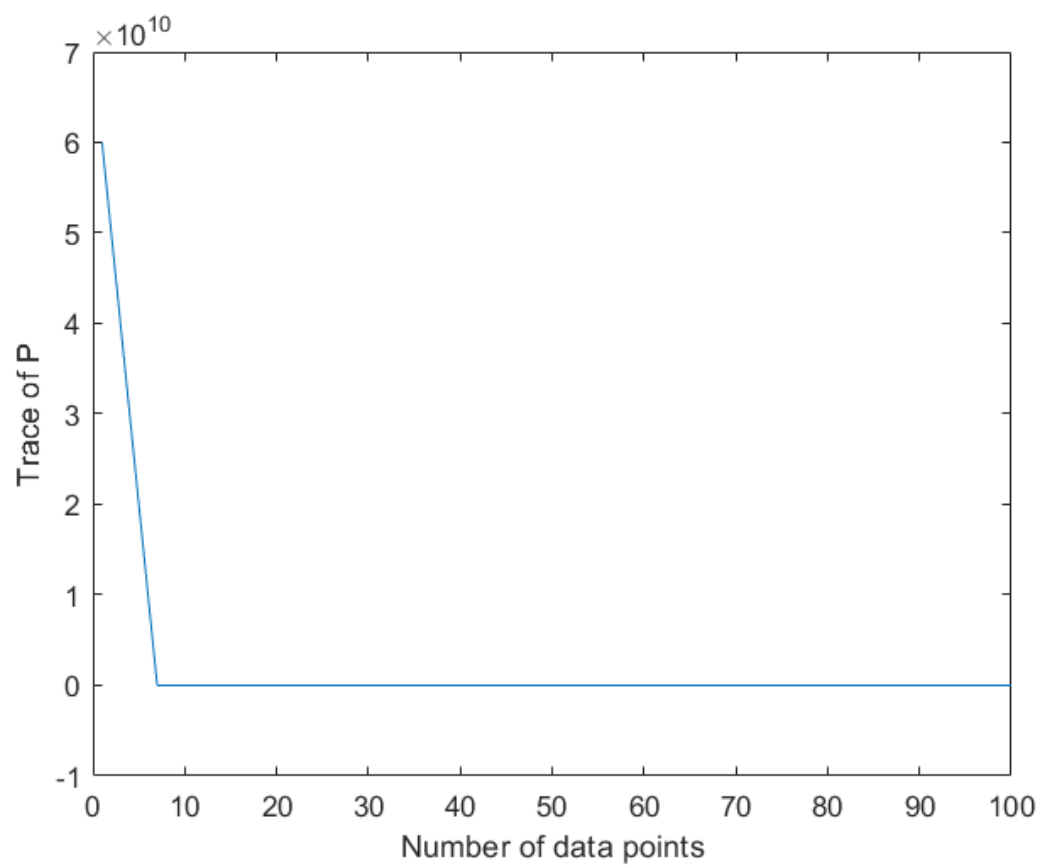
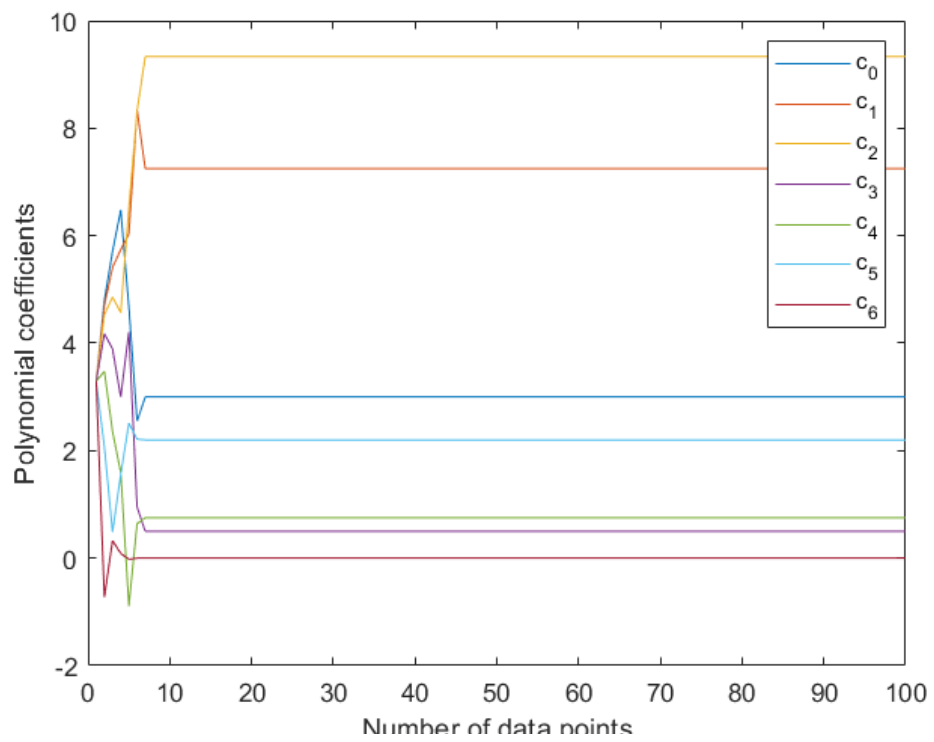
A=1

```
theta = 7x1
2.4511
7.3224
9.3300
0.5001
0.7500
2.2000
-0.0000
```



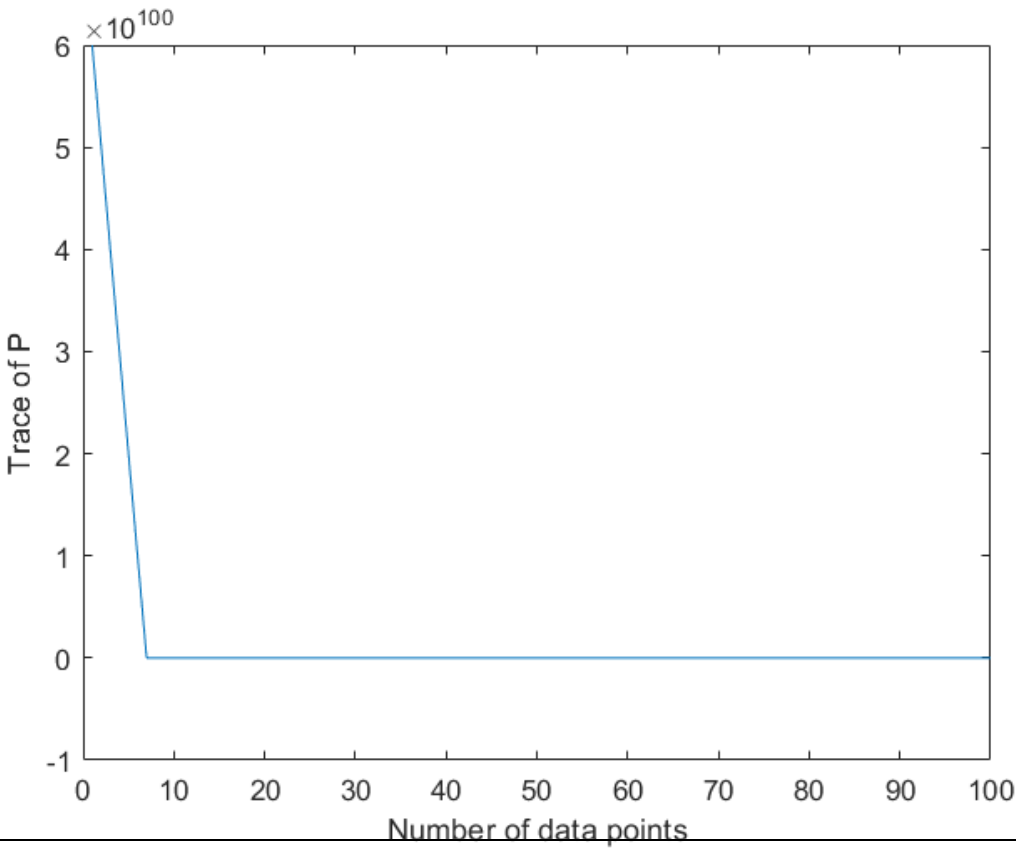
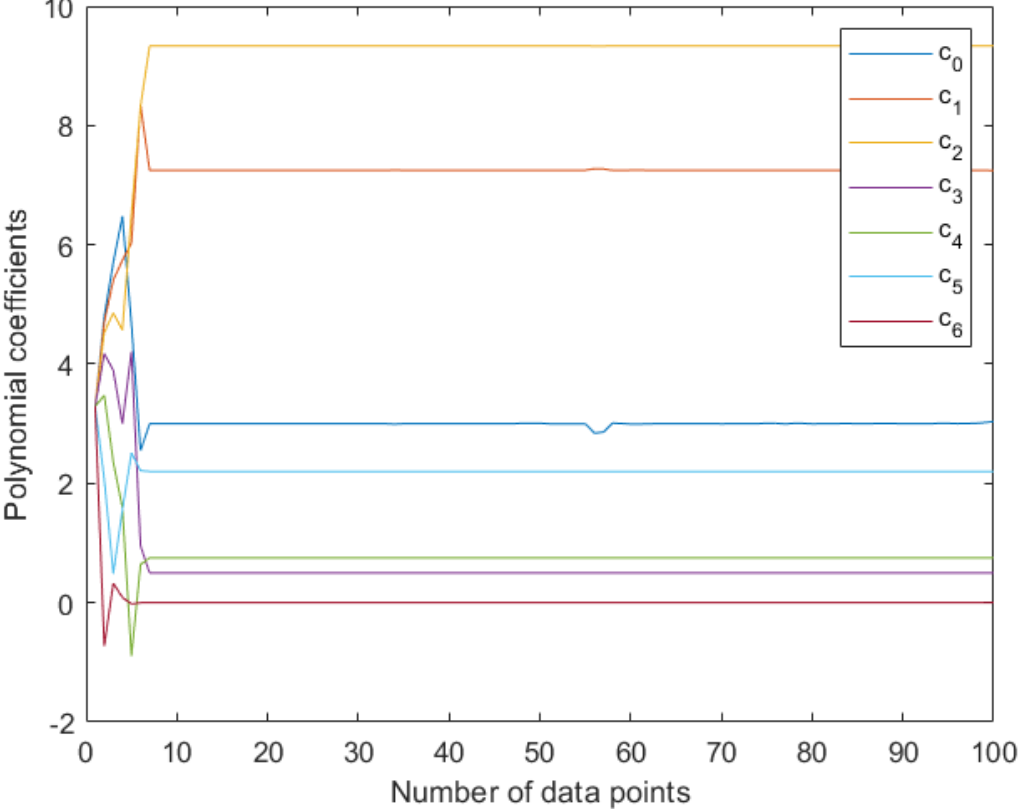
A=10

```
theta = 7×1  
3.0000  
7.2500  
9.3333  
0.5000  
0.7500  
2.2000  
-0.0000
```



A=100

theta = 7×1
3.0364
7.2462
9.3335
0.5000
0.7500
2.2000
0.0000



با توجه به داده های می فهمیم که با بیشتر شدن a سرعت همگرایی بیشتر شده و با تعداد داده کمتری می توان به مقدار نهایی ضریب ها دست یافت.

Main Initialization Output Saturation Data Types State Attributes

Controller parameters

Source: internal

Proportional (P): 3

Integral (I): 5

Derivative (D): 0

☒ Use filtered derivative

Filter coefficient (N): 100

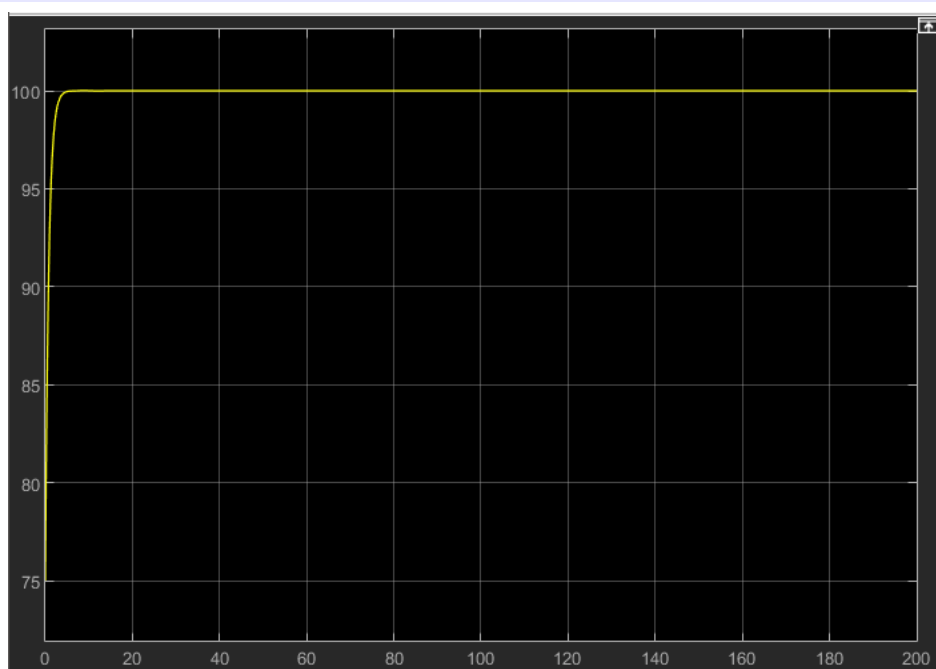
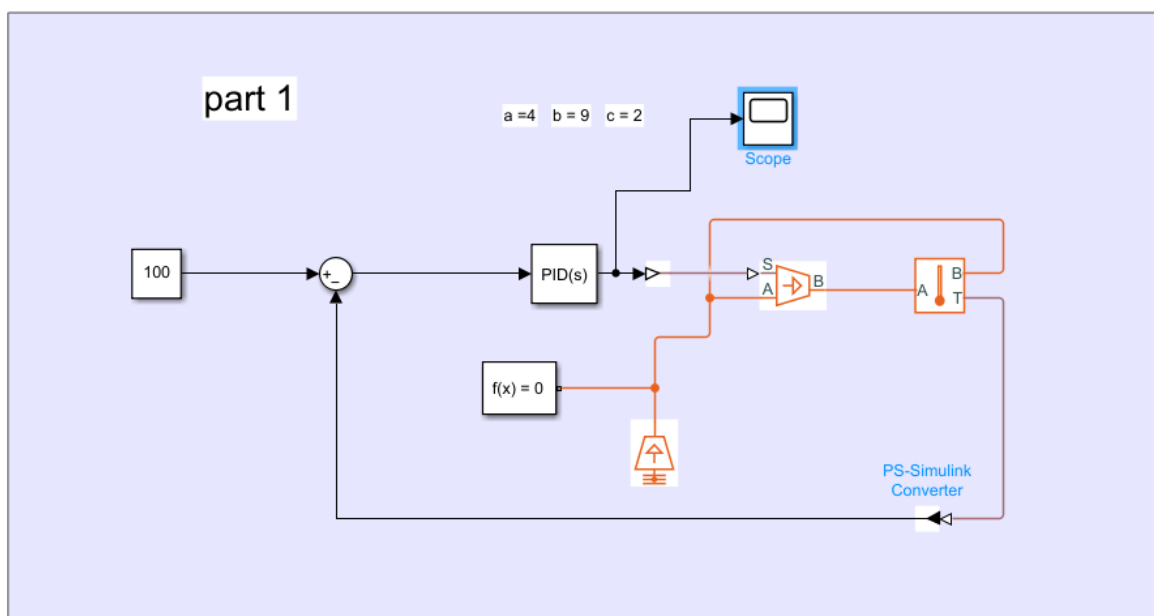
Automated tuning

Select tuning method: Transfer Function Based (PID Tuner App) Tune...

☐ Enable zero-crossing detection

الف

مقادیر دلخواه :



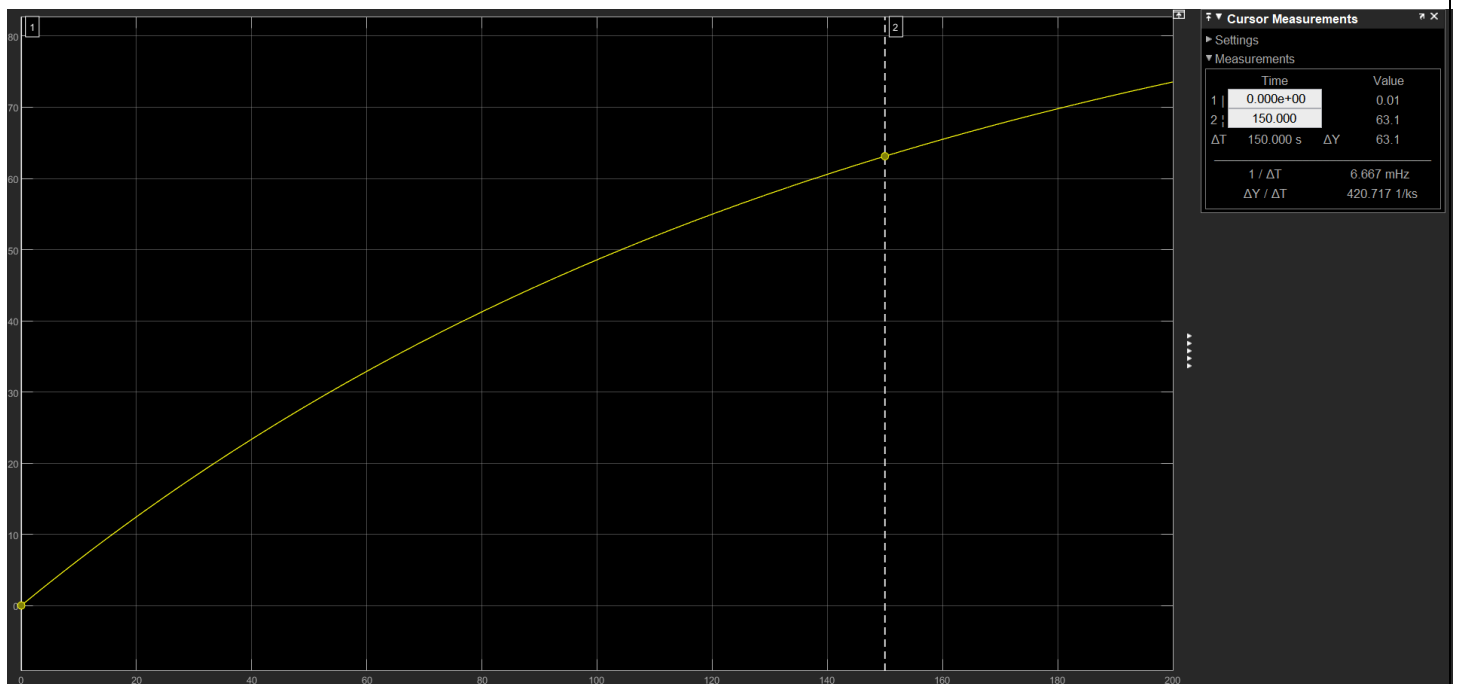
حال به بخش بعدی میریم

با توجه به مقادیر گفته شده $abc = 492$ مقدار ثابت زمانی برابرست با $۱۵۰ = ۳ * ۱۰ * ۵$

که در اون موقع باید به حدود ۶۳ درصد مقدار نهایی خود که ۱۰۰ هست برسد یعنی ۶۳

با بازی کردن با مقادیر داریم:

Main	Initialization	Output Saturation	Data Types	State Attributes
Controller parameters				
Source: internal				
Proportional (P): 0.0001				
Integral (I): 0.00665				
Derivative (D): 0				
<input checked="" type="checkbox"/> Use filtered derivative				
Filter coefficient (N): 100				



3-2

اگر بخوایم اثر بایاس حسگر رو رو ببینیم با اضافه کردن آن مشاهده می کنیم که در مقدار نهایی تاثیر می گذارد و می تواند مقدار نهایی را زیاد یا کم کند



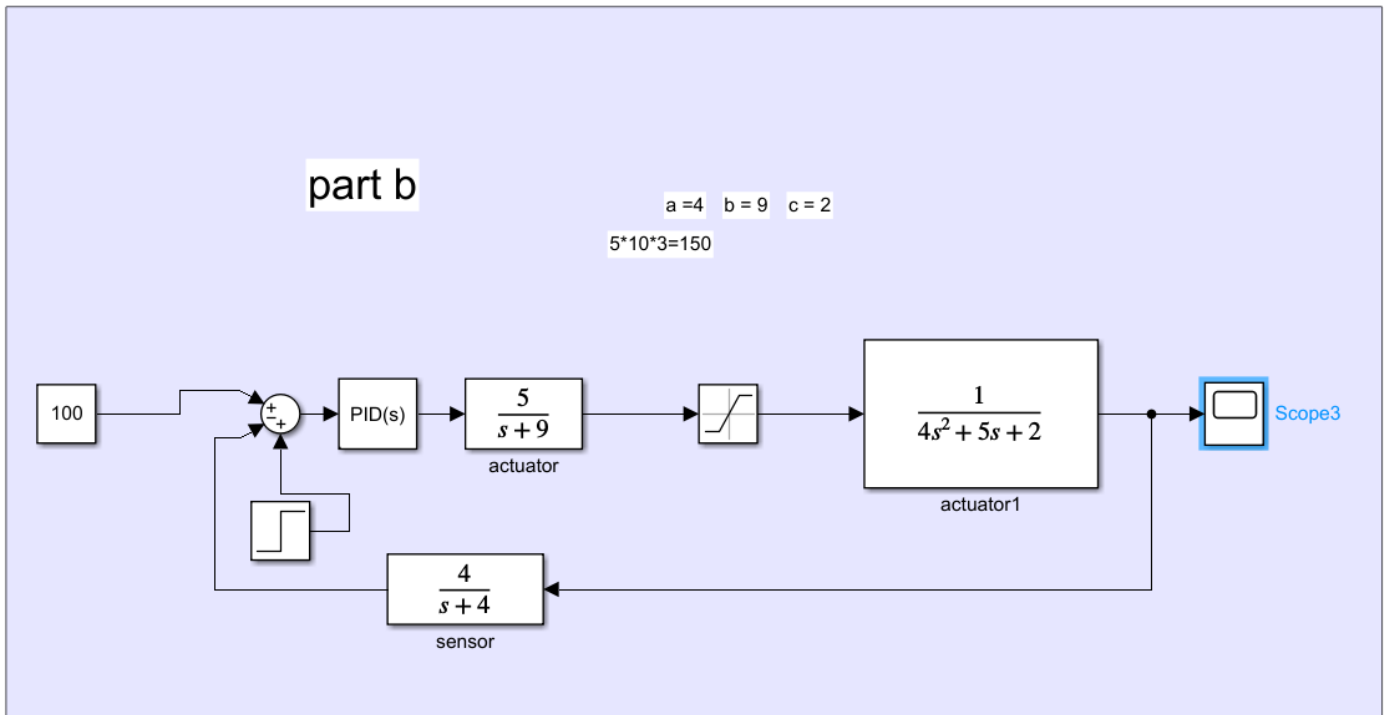
همچنین اگر ثابت زمانی زیاد شود جهش آن را در $t=5$ نمی بینیم و فقط تاثیر آن را در مقدار نهایی

می بینیم



قسمت ب

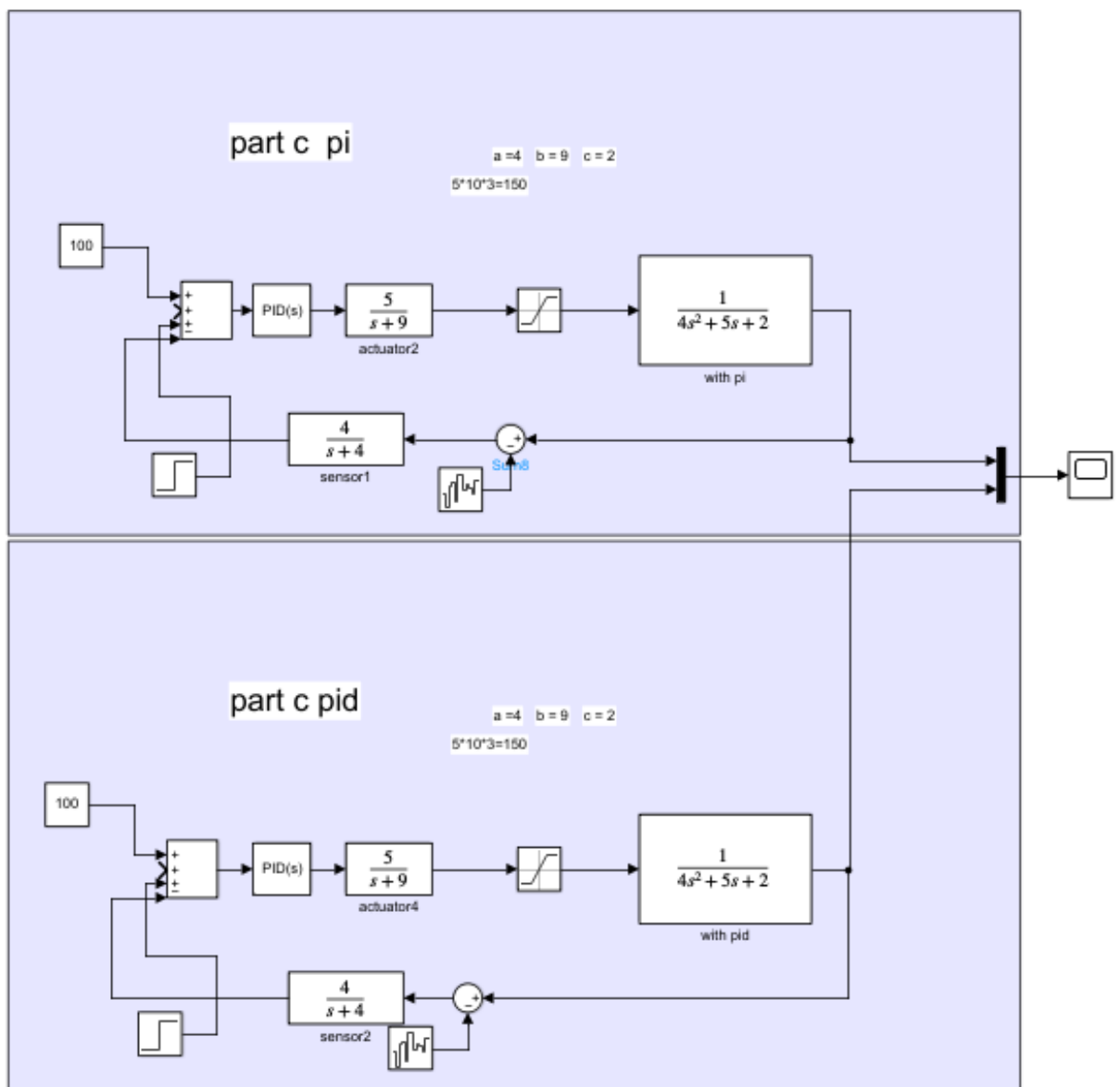
در این قسمت با توجه به saturation گذاشته شده باعث می شود دستگاه محدود شود و کنترلر pi و محرک دستگاه به خوبی نتوانند خروجی خود رو به سیستم اصلی برسوند و به همین علت به نتیجه دلخواه خود در نهایت نمیرسیم.



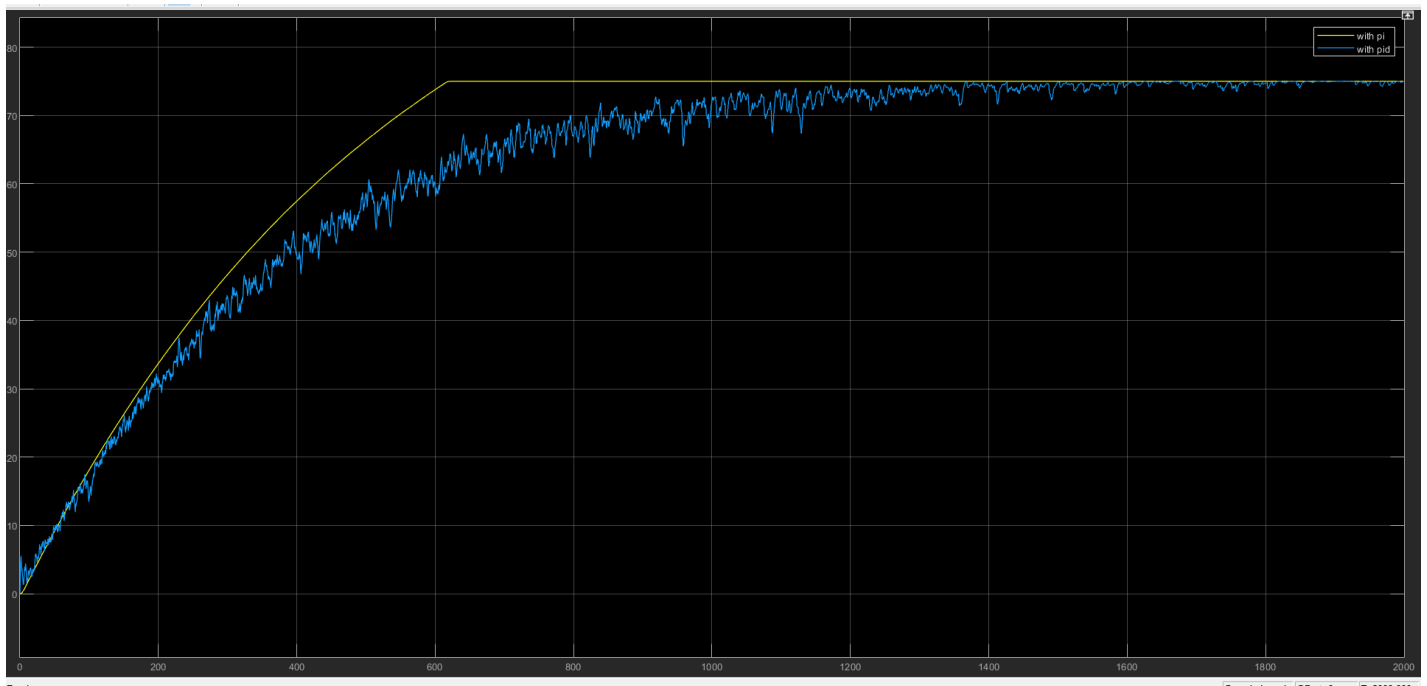
با توجه به عکس بالا قبل از رسیدن به مقدار نهایی خود در ۸۰ می برد و بیشتر نمیشود نتیجه ای که میشه گرفت اینه که باید حتمن محدودیت های دستگاهی رو در محاسبات خود اعمال کنیم.

ج

در این قسمت با اضافه کردن نویز و اینکه بسته به سیستم مشابه با کنترلر pid داریم:



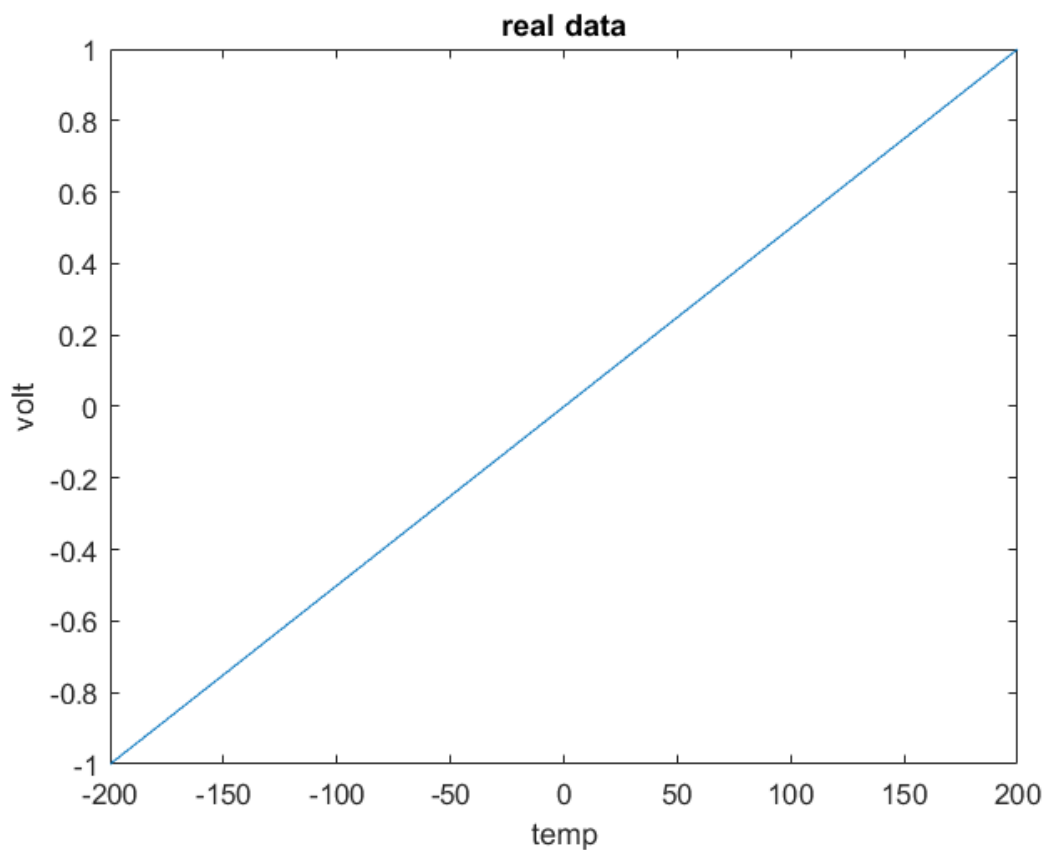
شکل نمودار رو داریم:



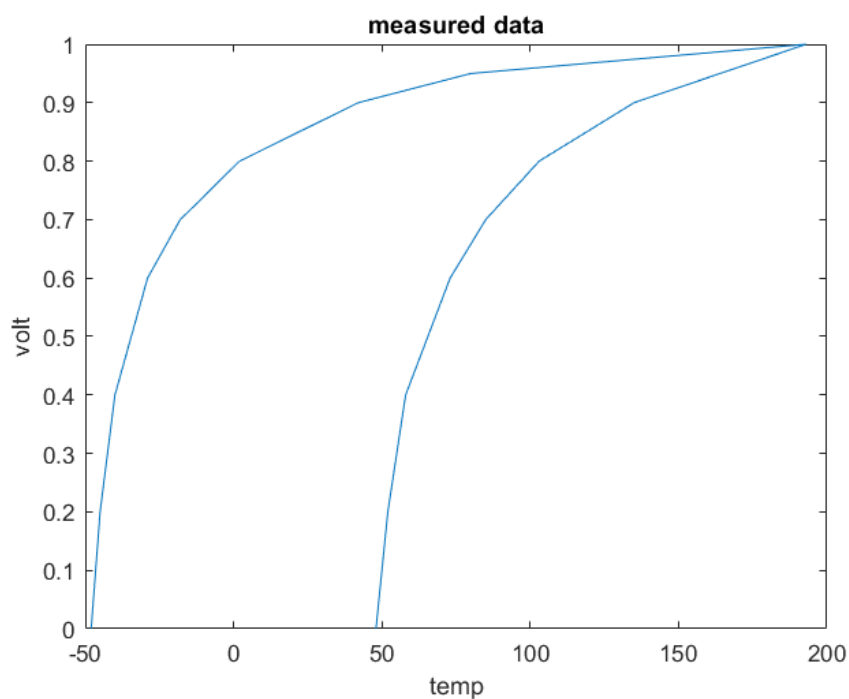
می دانیم دامنه نویز کمه و انتگرالش هم کم هستش (البته بستگی به ضریب داره اما در کل) اما تغییرات لحظه ایش زیاده این باعث میشه با اضافه کردن مشتق تغییرات زیادی در خروجی ببینیم و هم اینکه زمان اینکه به مقدار نهاییش برسه بیشتر میشه

با توجه به ورودی که بین -۲۰۰ تا ۲۰۰ هستش و خروجی که بین -۱ تا ۱ هستش

ابتدا به کمک متلب نمودار مقدار واقعی رو می بینیم که چطور باید کار کنه داریم :

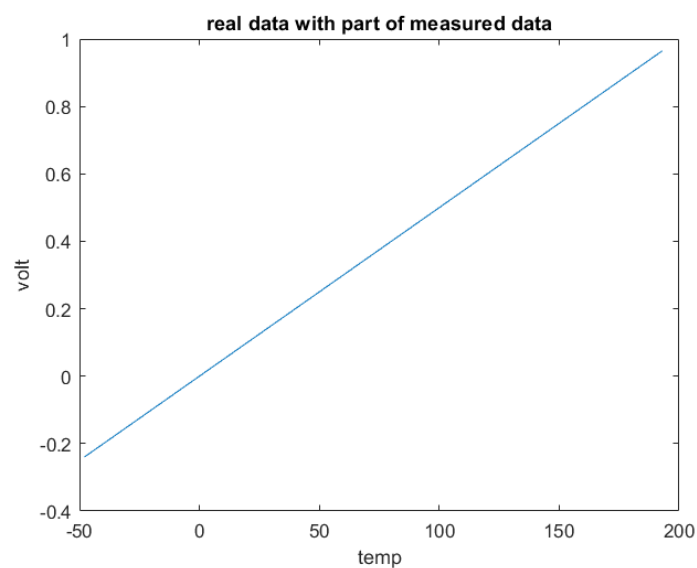
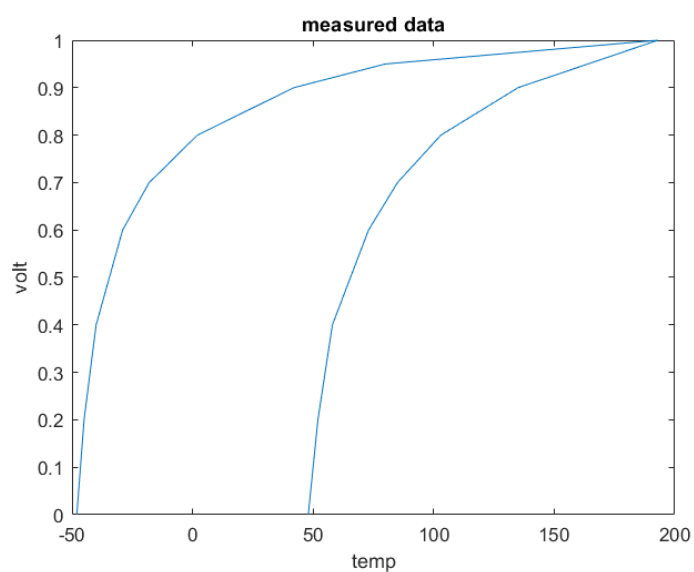


حال نمودار بخش از داده های اندازه گیری شده رو می بینیم:



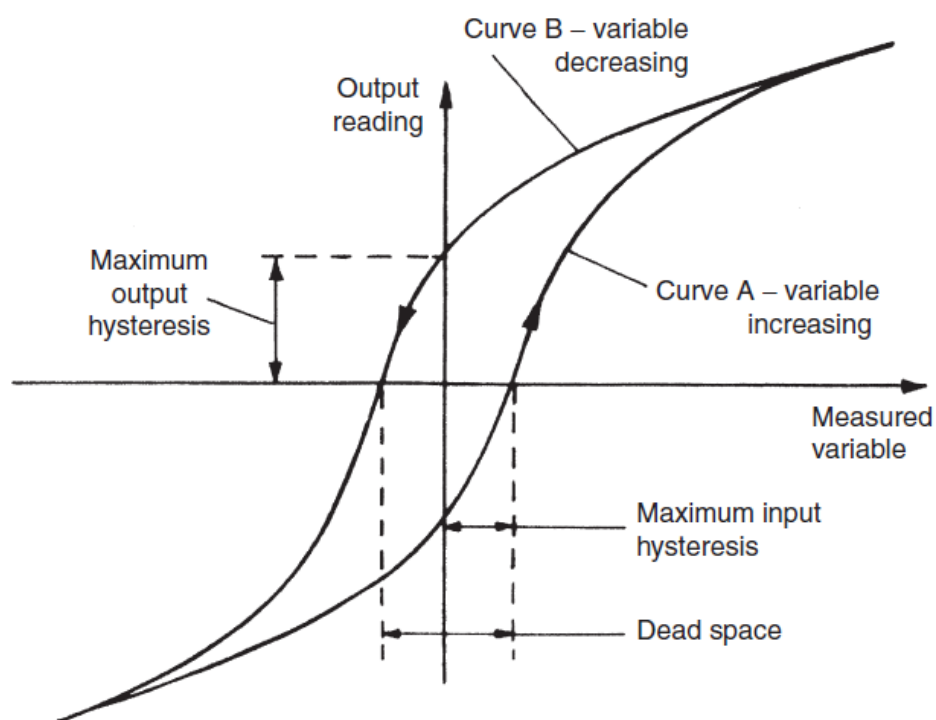
حال می‌تونیم بیایم و دماهای داده شده در داده‌ها را به مقدار واقعی آن را بیایم و مقایسه کنیم

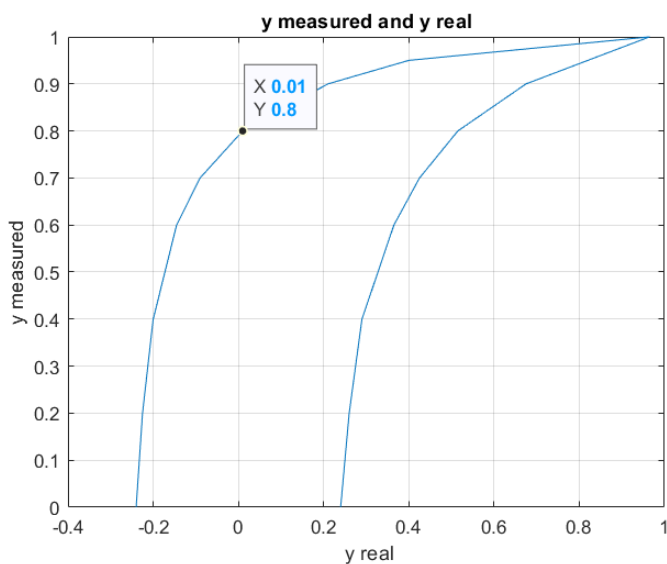
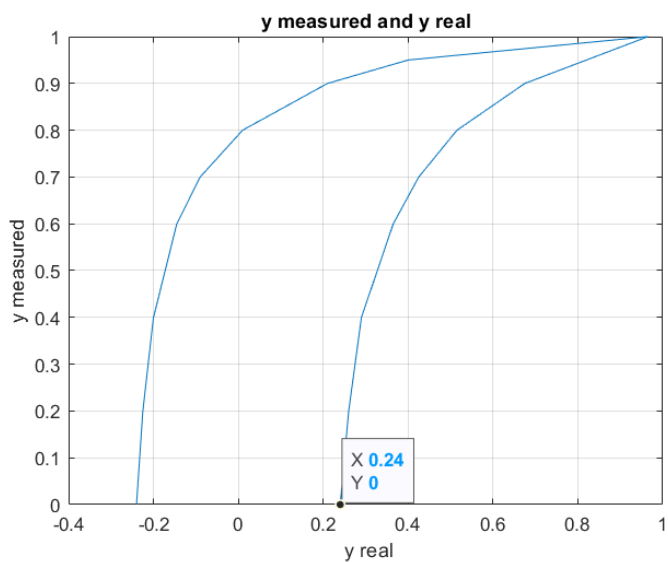
داریم:



حال می‌تونیم نموداری برحسب مقدار واقعی و مقدار اندازه‌گیری شده رسم کنیم

و طبق شکل کتاب روش محاسبه رو هم داریم :





خوب حال با توجه به مقادیر داریم:

$$\max output Hys = 0.8$$

$$\max input Hys = 0.24$$

حال بر حسب FS% می دانیم بازه خروجی دو هستش داریم :

$$\max output Hys = \frac{0.8}{2} * 100 = 40\%fs$$

$$\max input Hys = \frac{0.24}{2} * 100 = 12\%fs$$

با توجه به داده ها:

برای ۲۰ درجه سانتی گراد

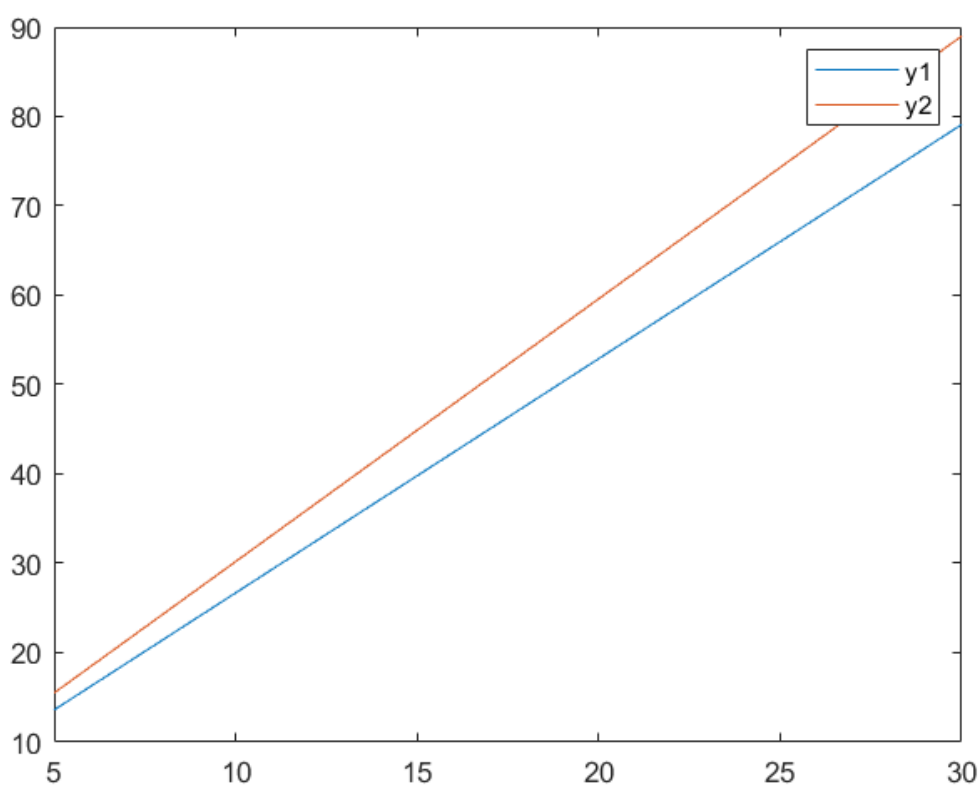
79.1	66	52.9	39.8	26.7	13.6	y
30	25	20	15	10	5	x

برای ۵۰ درجه سانتی گراد

89	74.3	59.6	44.9	30.2	15.5	y
30	25	20	15	10	5	x

ابتدا سعی می کنیم رابطه خطی رو بنویسیم داریم:

$$y_1 = \frac{13.1}{5} x + 0.5 \quad , \quad y_2 = \frac{14.7}{5} x + 0.8$$



حال مقادیر بدست می آوریم:

$$\text{Zero drift (bias)} = |0.8 - 0.5| = 0.3v/v$$

$$\text{Zero drift (bias)} / c^0 = \frac{|0.8 - 0.5|}{50 - 20} = 0.01v/(vc^0)$$

$$\text{sensitivity drift} = |2.94 - 2.62| = 0.32v/v$$

$$\text{sensitivity drift} / c^0 = \frac{|2.94 - 2.62|}{50 - 20} = 0.010667v/(vc^0)$$

ب

برای این قسمت برای اینکه حل کنیم ابتدا دما ی ۵۰ درجه رو روحسگر ۲۰ و ۵۰ در میاوریم

داریم:

For 20c -> x=50 ->y=131.5

For 50c -> x=50 ->y=147.8

حال برای این که در حسگر در ۴۰ را بدست بیاوریم می دانیم که دو سوم اختلاف بین ۲۰ و ۵۰

هستش داریم

$$\text{For } 40c \rightarrow x = 50 \rightarrow y = \frac{40 - 20}{50 - 20} * (147.8 - 131.5) + y_{20c}(50) = 142.36666$$

