



به نام خدا



1928

K. N. Toosi University of Technology

دانشگاه صنعتی خواجه نصیرالدین طوسی

دانشکده برق

شبیه سازی و مدلسازی

گزارش تمرین شماره 1

سجاد رجبی باغستان

40005393

استاد : جناب دکتر مهدی علیاری

فهرست مطالب

عنوان	شماره صفحه
بخش ۱: چکیده	3
بخش ۱: سوالات تحلیلی	4
سوال اول	4
بخش ۲: سوالات شبیه سازی	5
سوال اول	5
سوال دوم	7
سوال سوم	15
مراجع	17

چکیده

در این تمرین، به بررسی و مقایسه چهار روش متداول برای تخمین و پیش‌بینی می‌پردازیم، که عبارتند از:

1. کمترین مربعات (LS): یک روش ساده و معمول برای تخمین پارامترها با استفاده از کمینه‌کردن مجموع مربعات خطاها.
 2. کمترین مربعات بازگشتی (RLS) با عامل فراموشی: یک روش پیشرفته‌تر که با در نظر گرفتن عامل فراموشی، توانایی بهبود پاسخ به تغییرات در داده‌های ورودی را دارد.
 3. Sliding Window: یک روش که بر اساس تمرکز بر روندهای اخیر و تشخیص ناهنجاری‌ها استوار است و در پیش‌بینی‌هایی که نیاز به توجه به الگوهای متغیر دارند، مفید است.
 4. کمترین مربعات بازگشتی (RLS) برای Sliding Window: یک ترکیب از RLS با عامل فراموشی با Sliding Window که برای پیش‌بینی در سیستم‌های پویا با روندهای متغیر مؤثر است.
- هر یک از این روش‌ها مزایا و معایب خاص خود را دارند. انتخاب روش مناسب بستگی به عوامل مختلفی دارد، از جمله:

- نوع داده: آیا داده‌ها ایستا هستند یا پویا؟
- میزان پویایی سیستم: آیا سیستم ثابت است یا در حال تغییر؟
- منابع محاسباتی: آیا منابع محدود هستند یا نه؟

به طور خلاصه، LS برای داده‌های ایستا و سیستم‌های با ثبات مناسب است. RLS با عامل فراموشی برای داده‌های پویا و سیستم‌های در حال تغییر مفید است. Sliding Window برای تمرکز بر روندهای اخیر و تشخیص ناهنجاری‌ها مناسب است. و کمترین مربعات بازگشتی (RLS) برای Sliding Window برای پیش‌بینی در سیستم‌های پویا با روندهای متغیر مؤثر است.

انتخاب روش مناسب بستگی به نیازهای خاص شما دارد.

بخش ۱: سوالات تحلیلی

سوال اول

حل سوال:

$$y = \theta_1 + \theta_2 u$$

$$\begin{cases} 4 = \theta_1 + \theta_2 \\ 5 = \theta_1 + 2\theta_2 \\ 7 = \theta_1 + 3\theta_2 \end{cases}$$

$$U = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{bmatrix}$$

$$U^T = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \end{bmatrix}$$

$$y = \begin{bmatrix} 4 \\ 5 \\ 7 \end{bmatrix}$$

$$LS \rightarrow \tilde{\theta} = (U^T U)^{-1} U^T y \rightarrow \left(\begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{bmatrix} \right)^{-1} \cdot \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \end{bmatrix} \cdot \begin{bmatrix} 4 \\ 5 \\ 7 \end{bmatrix} = \begin{bmatrix} 2.333 \\ 1.5 \end{bmatrix}$$

$$\theta_1 = 2.333$$

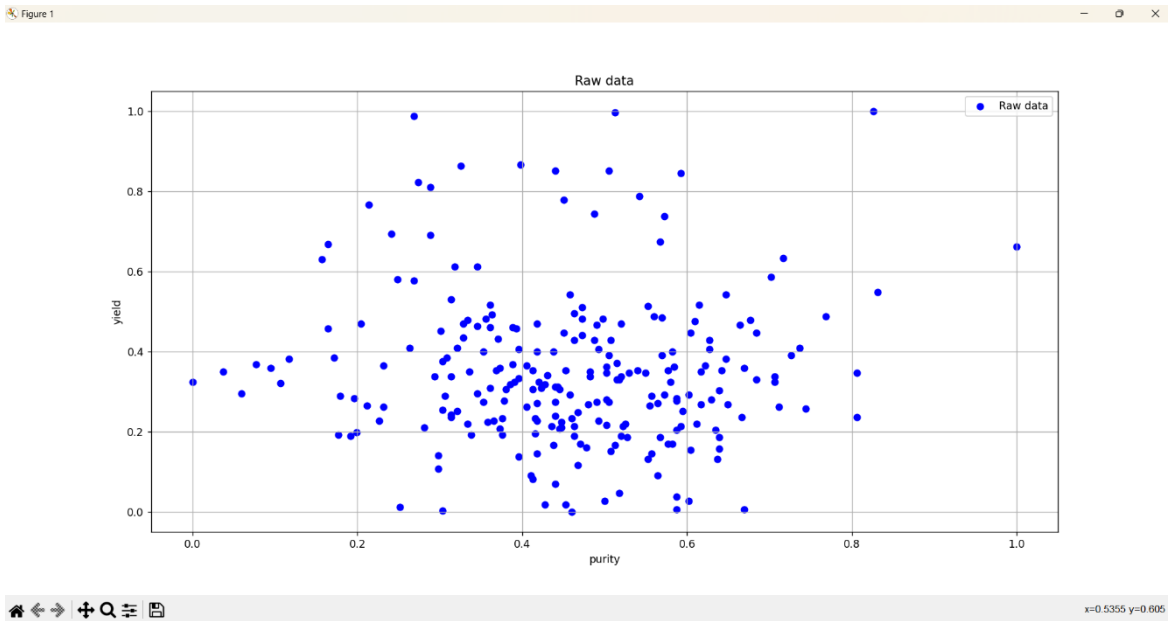
$$\theta_2 = 1.5$$

$$y = 2.333 + 1.5u$$

بخش ۲: سوالات شبیه سازی

سوال اول

نمودار داده ها در این سوال به صورت زیر است:



شکل 1: نمودار ورودی نسبت به خروجی بر اساس داده های خام

با توجه به نمودار میتوان حدس زد که خط پیشنهادی متد کمترین مربع شیب کمی داشته در ناحیه بین 0.4 تا 0.6 قرار میگیرد.

به صورت دستی با استفاده از فرمول least square مقدار بدست آمده به صورت زیر است:

$$\begin{aligned} \text{least square} \rightarrow \tilde{\theta} &= (U^T U)^{-1} U^T y \rightarrow \\ \tilde{\theta} &= -0.0223 \text{ \& Bias value} = 62.9266 \\ y &= 62.9266 - 0.0223u \end{aligned}$$

مقدار شیب خط بدست آمده از least square برابر -0.0223 است و مقدار ثابت معادله (بایاس) برابر با 62.9266 است.

```
# Calculate the least square With equations:
X = np.vstack([U, np.ones(len(U))]).T

U_T = X.T

U_RESULT = np.dot(U_T, X)

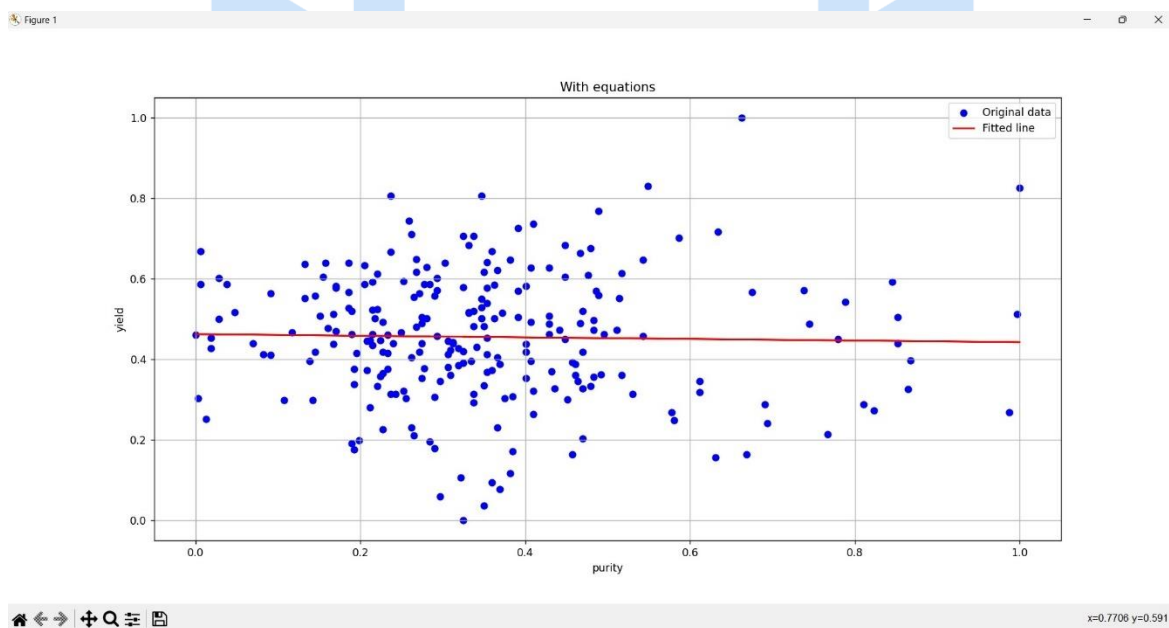
U_RESULT_INV = np.linalg.inv(U_RESULT)

theta = np.dot(np.dot(U_RESULT_INV, U_T), Y)

y = np.dot(theta[0], U) + theta[1]
```

شکل 2: نوشتن معادلات least square به صورت دستی در پایتون

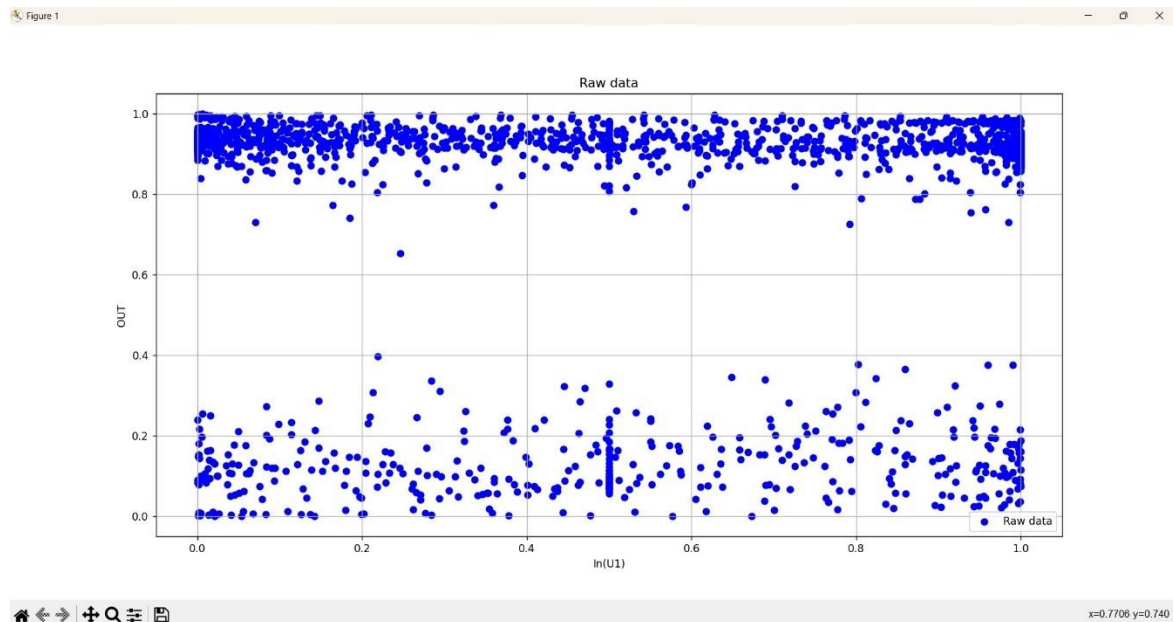
خط پیشنهادی و فیت شده با توجه به روش least square در پایتون :



شکل 3: خط بدست آمده با توجه به روش least square

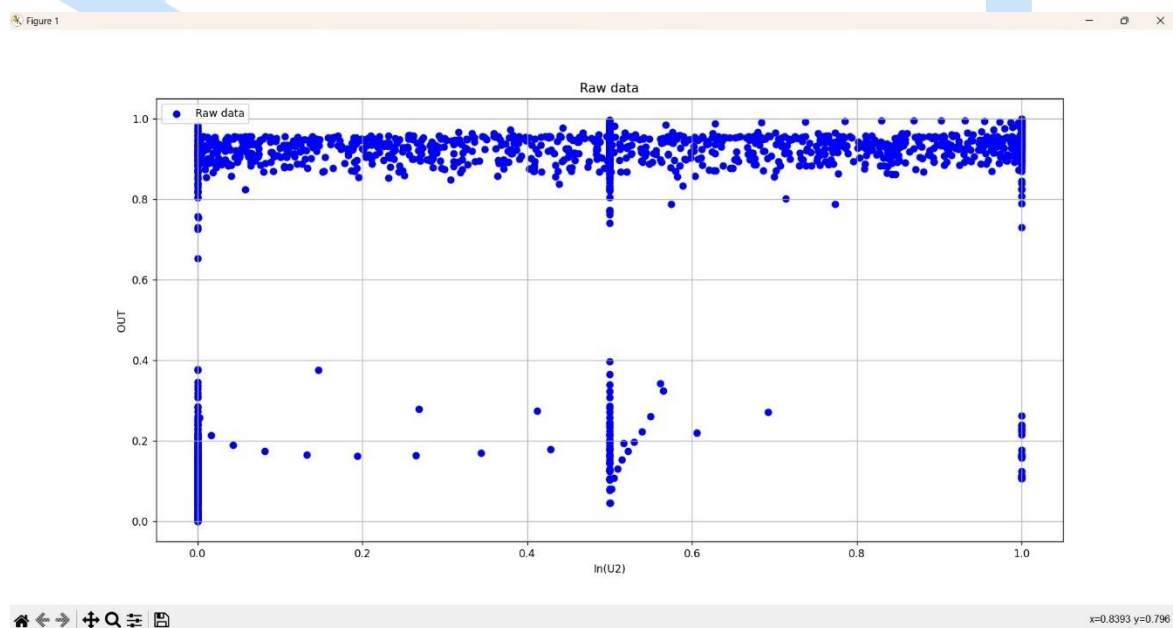
سوال دوم

نمودار خروجی بر حسب ورودی اول:



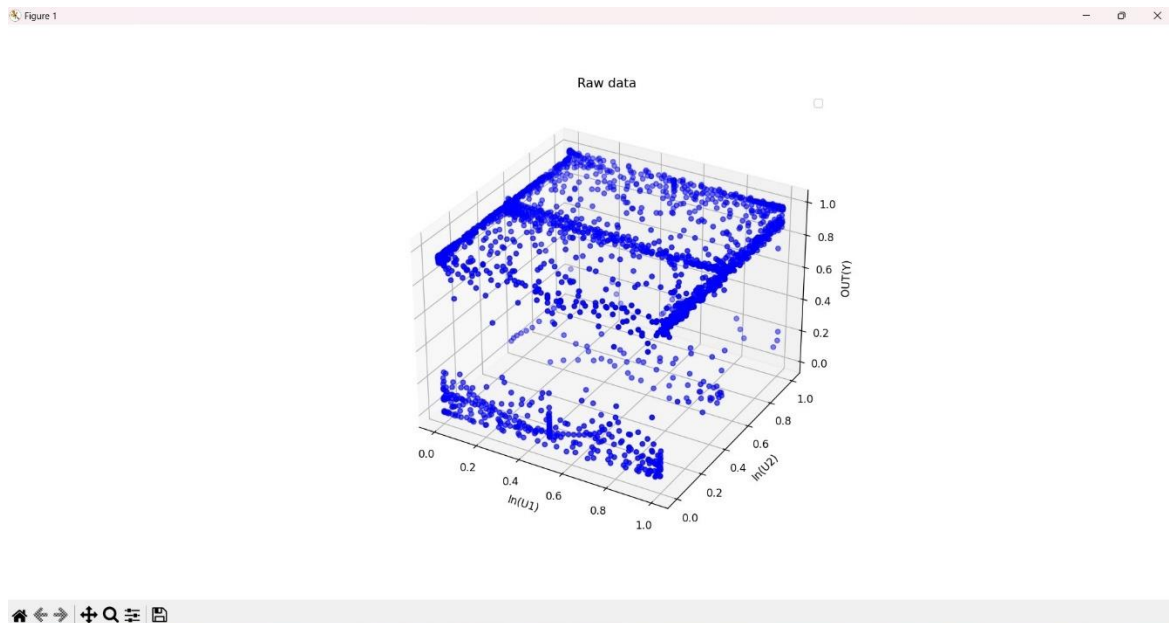
شکل 4: نمودار خروجی نسبت به ورودی اول

نمودار خروجی نسبت به ورودی دوم:



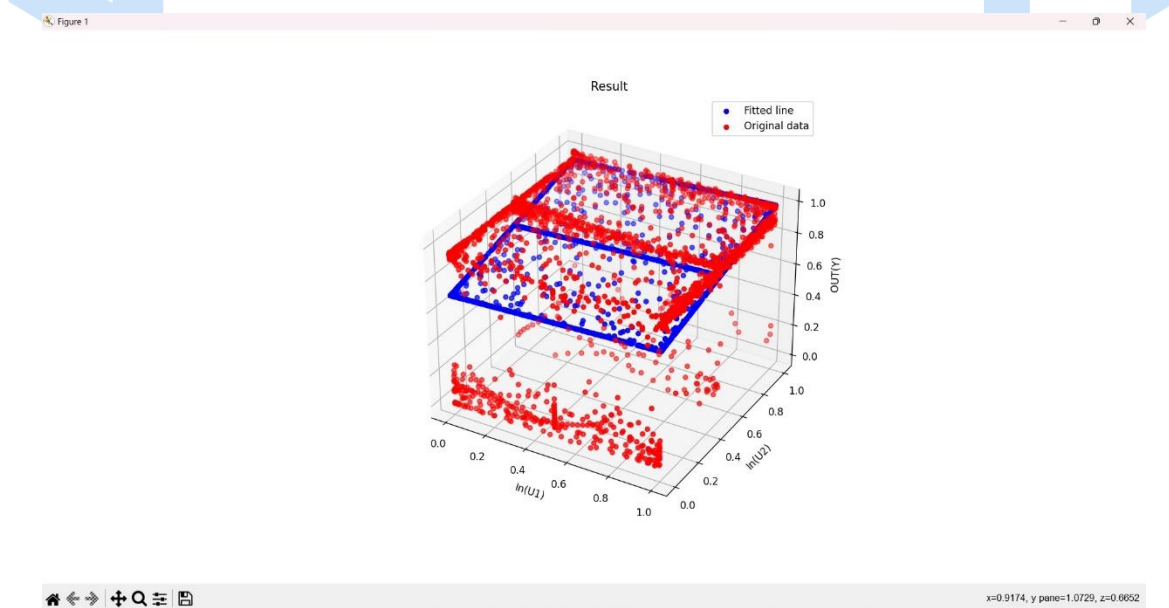
شکل 5: نمودار خروجی نسبت به ورودی دوم

نمودار سه بعدی داده ها (دو ورودی و خروجی) هم به صورت زیر است:



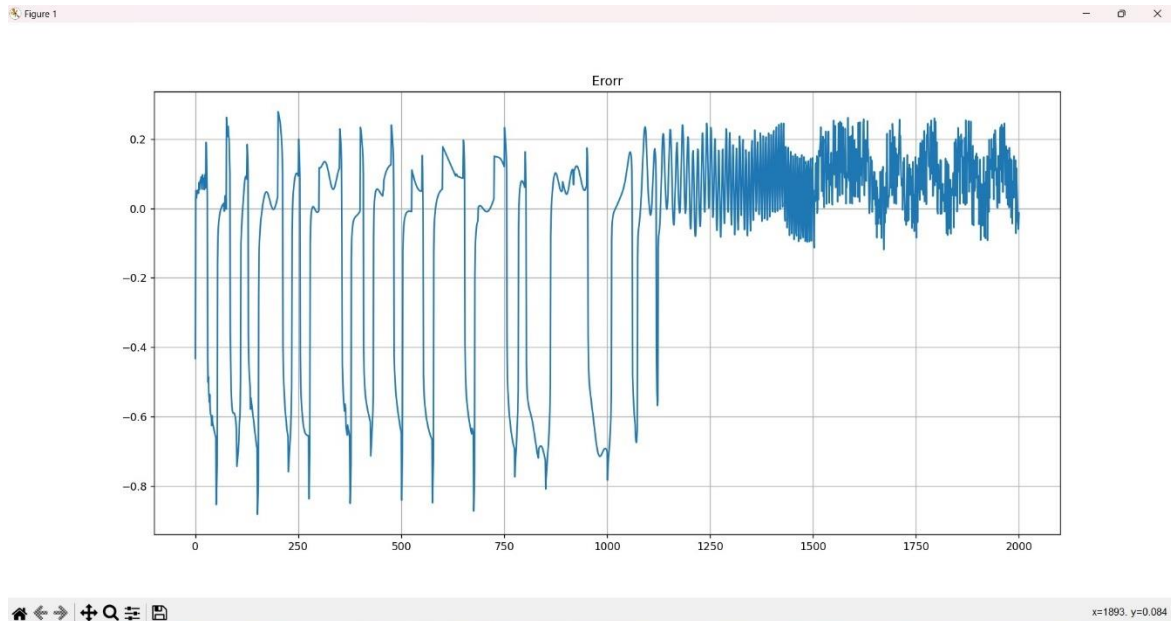
شکل 6: نمودار خروجی نسبت به ورودی اول و ورودی دوم

سپس با استفاده از متد least square بهترین خط برای این سیستم مدل می کنیم.
نمودار آن به صورت زیر است



شکل 7: نمودار بهترین خط بدست آمده با استفاده از متد least square

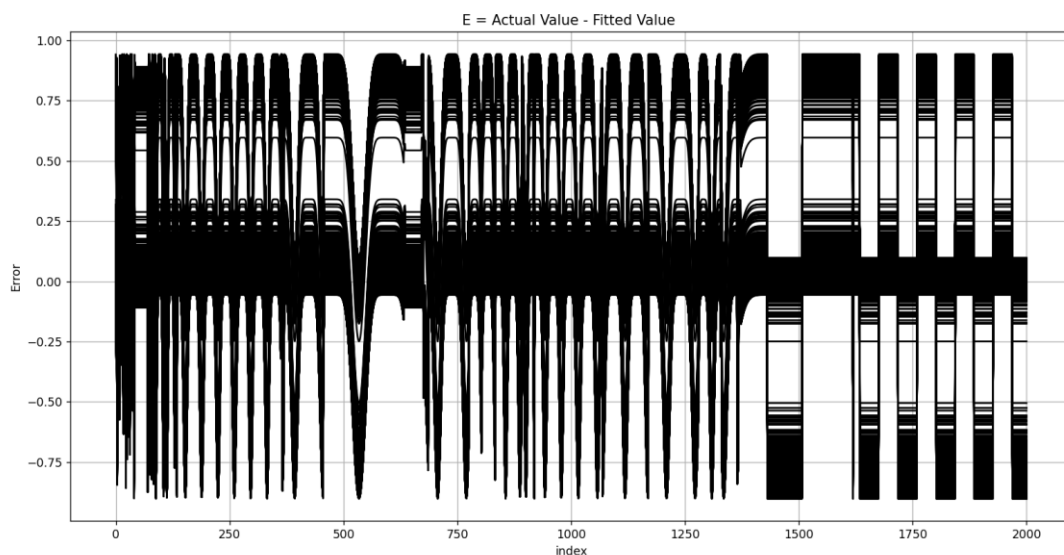
نمودار خطای مدلی که با استفاده از متد least square بدست آورده ایم نیز به صورت زیر است:



شکل 8: نمودار خطا مدل ساخته شده توسط **least square** نسبت به سیستم واقعی

و حالا برای متد forgetting factor باید لامبدا های مختلف را تست کنیم.

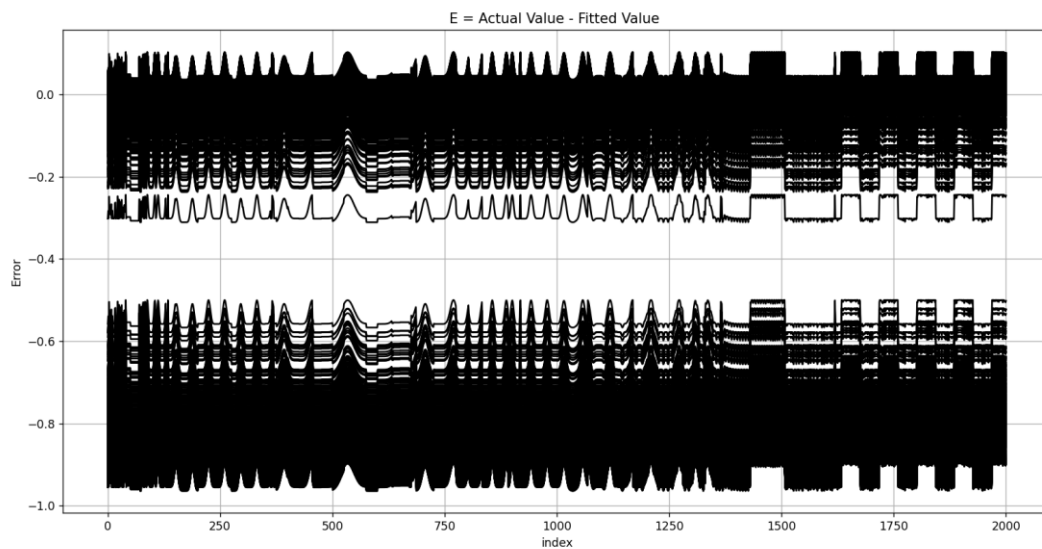
برای لامبدای 0.3 نمودار خطا به صورت زیر بدست می آید:



شکل 10: نمودار خطا مقداری که از مدل **forgetting factor** به دست آمده نسبت با لامبدای 0.3

همانطور که مشاهده می شود مقدار خطا زیاد است و نواسانات زیادی دارد زیرا مقدار لامبدا نزدیک صفر است و به داده های گذشته وزن بیشتری داده شده است.

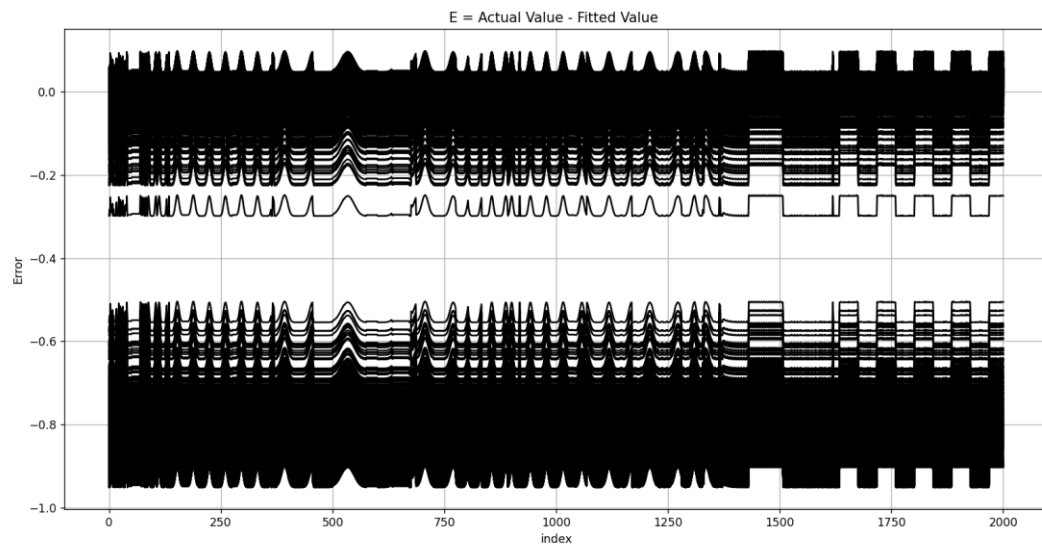
برای لامبدای 0.7 نمودار خطا به صورت زیر بدست می آید:



شکل 11: نمودار خطا مقداری که از مدل **forgetting factor** به دست آمده نسبت با لامبدای 0.7

مقدار خطا کاهش پیدا کرده است ولی همچنان زیاد است. برای کاهش آن باید باز هم مقدار لامبدا را افزایش دهیم و به یک نزدیک کنیم تا به داده های اخیر وزن بیشتری داده شود.

برای لامبدای 0.9 نمودار خطا به صورت زیر بدست می آید:



شکل 12: نمودار خطا مقداری که از مدل **forgetting factor** به دست آمده نسبت با لامبدای 0.9

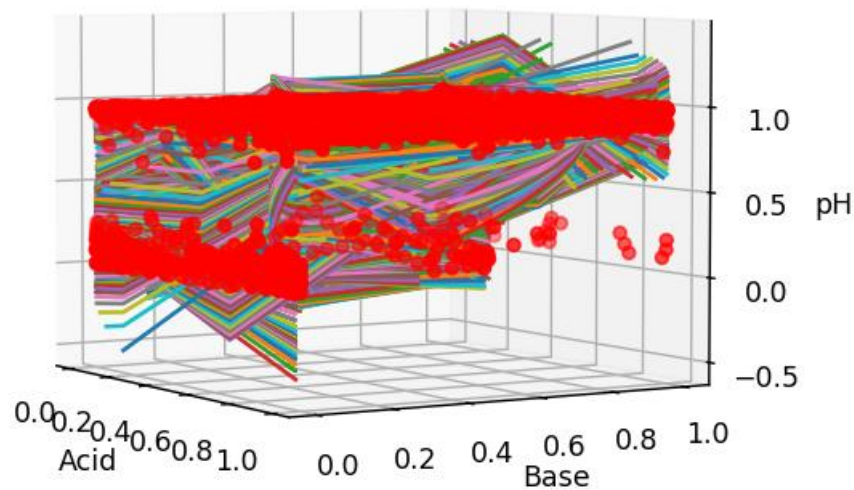
همانطور که مشاهده می شود مقدار خطا نسبت به حالت قبل کاهش پیدا کرده است چون لامبدا نزدیک یک است و داده های اخیر وزن بیشتری دارند و نوسانات خطا هم کاهش پیدا کرده است.

و همچنین نمودار خطای مدلی که با استفاده از متد forgetting factor بدست آمده است به صورت زیر است:

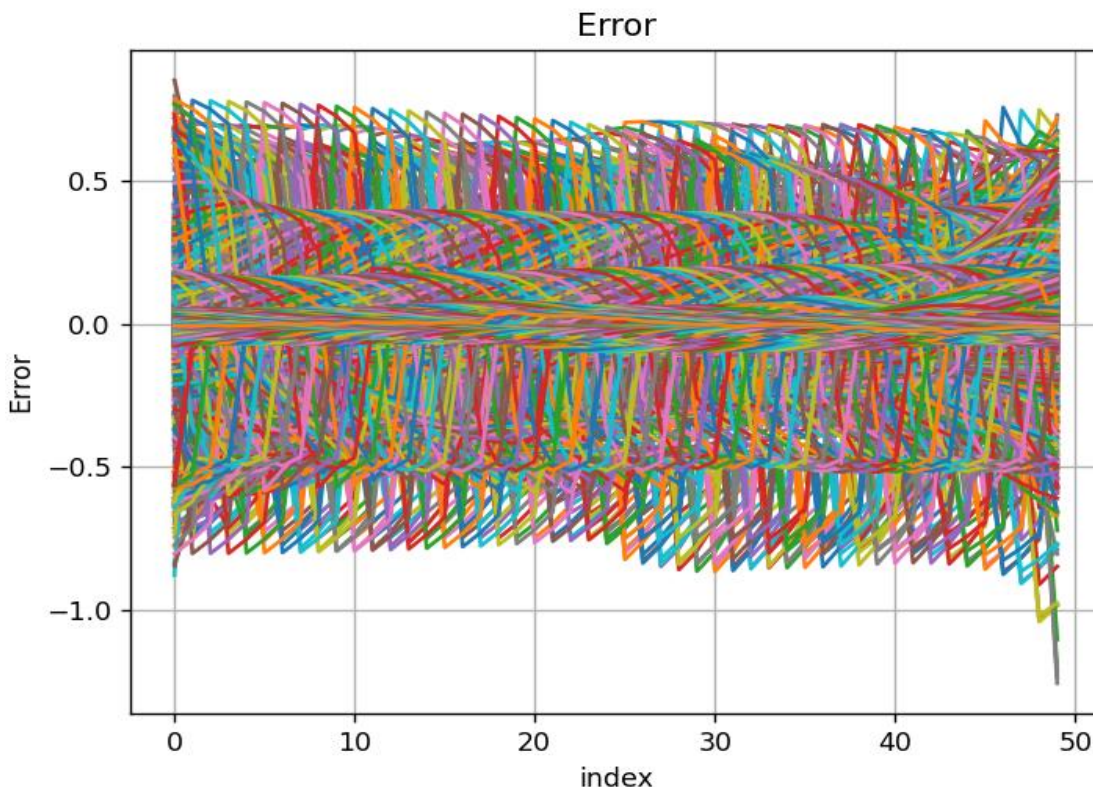
همانطور که از نمودار ها مشخص است مقدار خطا در مدلی که از متد forgetting factor بدست آورده ایم بسیار کمتر است از مدلی که از متد least square بدست آمده است و که علت آن هم این است که در متد forgetting factor به داده های اخیر وزن بیشتری داده شده است و داده های گذشته اهمیت کمتری دارند. پس مقدار نوسانات خطا در داده های اخیر کمتر می شود.

و حالا خطای مدلی که از sliding window بدست آمده است به صورت زیر است:

Fitted Lines Sliding Window



شکل 13: نمودار بهترین خط بدست آمده با استفاده از متد sliding window LS



شکل 14: نمودار خطا مقداری که از مدل **sliding window** به دست آمده نسبت به مقدار واقعی

روند کلی خطای ابتدا به صورت صعودی است، سپس به طور نوسانی در محدوده 0 تا 0.2 باقی می‌ماند و در نهایت در انتهای نمودار به صورت نزولی می‌باشد. نوسانات خطا در تخمین خطوط برازش شده برای جریان محلول اسید بیشتر از نوسانات خطا در تخمین خطوط برازش شده برای جریان محلول پایه است. وجود چند نقطه داده با خطای بسیار بیشتر از میانگین در نمودار خطای جریان محلول اسید نشان می‌دهد که این روش در برخی موارد ممکن است دقت پایینی داشته باشد. با توجه به نمودارهای به دست آمده از روش‌های مختلف، بهترین روش باید خطای کمتری داشته باشد. با توجه به نمودار، روش کمترین مربعات با فراموشی متغیر (least square with forgetting factor) دارای خطای کمتری است، بنابراین بهترین روش برای این نوع داده است. همچنین، خطای آن در داده‌های اخیر بسیار کم و تقریباً در بازه 0.1 تا -0.1 است، بنابراین از دقت کافی برخوردار است.

مقایسه متد **Least square (LS)** و **Recursive Least Squares (RLS)** با عامل **Sliding Window** و **Forgetting factor**

با توجه به جزئیات ارائه شده، می‌توانیم به نتیجه‌گیری‌های زیر برسیم:

Least square (LS)

- نقاط قوت: سادگی و کارآمدی محاسباتی برای سیستم‌های ایستا.
- نقاط ضعف: عدم سازگاری با داده‌های در حال تغییر و تأثیر قابل توجه داده‌های گذشته بر مدل.

Least square with Forgetting Factor

- نقاط قوت: **Sliding window** بهتر در حذف داده‌ها عمل می‌کند و **RLS** با عامل فراموشی، انعطاف بیشتری در سازگاری با تغییرات دارد.
- نقاط ضعف: از نظر محاسباتی پرهزینه‌تر است.

Sliding Window

- نقاط قوت: حذف صریح داده‌های قدیمی، اطمینان از بازتاب روندهای اخیر و پیاده‌سازی آسان.
- نقاط ضعف: احتمال از دست رفتن اطلاعات ارزشمند و نیاز به انتخاب دقیق اندازه پنجره.

تصمیم‌گیری:

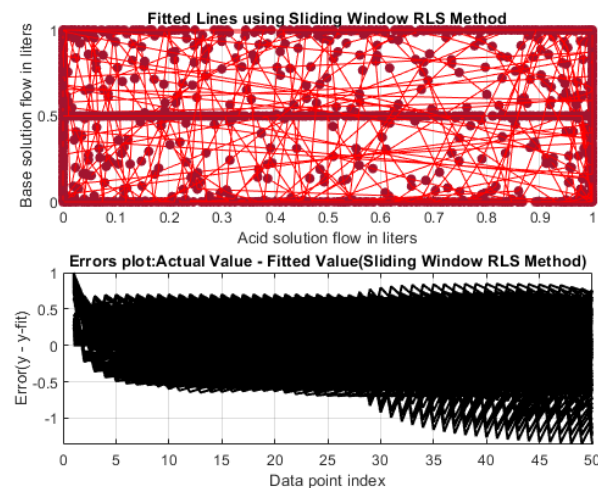
- برای سیستم‌های ایستا با منابع محدود، **LS** ممکن است کافی باشد.
- در سیستم‌های پویا با اهمیت بالای داده‌های اخیر، **RLS** با عامل فراموشی یا **Sliding window** ممکن است بهتر عمل کنند.
- برای تعادل بین هزینه محاسباتی و سازگاری، **RLS** با عامل فراموشی ممکن است مناسب باشد، اما اگر سادگی در اولویت باشد، **Sliding window** بهتر است.

به طور خلاصه، انتخاب رویکرد مناسب باید بر اساس نیازهای خاص و محدودیت‌های سیستم موردنظر صورت گیرد.

مناسب بودن برای	مدیریت داده	پیچیدگی محاسبه	روش
سیستم‌های ایستا	از تمام داده‌های گذشته استفاده می‌کند	پایین	Least square
سیستم‌های پویا با سازگاری نرم	وزن بیشتری به داده‌های اخیر می‌دهد	بالا	Recursive Least Squares with forgetting factor
سیستم‌های پویا با تمرکز بر روند اخیر	داده‌های قدیمی را به‌صراحت حذف می‌کند	متوسط (ناگهانی)	Sliding window

جدول 1: مقایسه متد های طراحی مدل

نمودار خطای متد RLS with Sliding window به صورت زیر است:



شکل 15: نمودار خطا مقداری که از مدل **RLS Sliding Window** به دست آمده نسبت به مقدار واقعی

همانطور که مشاهده می‌شود، شیب نمودار خطا در حال کاهش است، اشاره‌گری به کاهش خطا می‌باشد. میانگین مربعات خطا (**MSE**) در بیشتر نقاط داده کم است که نشان از کارایی قابل توجه مدل **RLS** با استفاده از پنجره کشویی در تخمین خط روند داده‌ها است. این می‌تواند ناشی از ناکافی بودن داده‌ها در بخش‌های ابتدا و انتها برای تخمین دقیق خط روند باشد. همچنین، افزایش ناگهانی خطا در برخی نقاط ممکن است به دلیل ناهنجاری‌های در داده‌ها یا تغییرات ناگهانی در روند اتفاق بیفتد.

• بخش‌های با خطای کم: این بخش‌ها نشان می‌دهند که مدل **RLS** با استفاده از پنجره کشویی به‌طور دقیق خط روند داده‌ها را تخمین زده است، که نشان از مناسب بودن پارامترهای مدل و کافی بودن داده‌ها برای تخمین دقیق می‌باشد.

• بخش‌های با خطای زیاد: در این بخش‌ها، مدل **RLS** با پنجره کشویی مشکلاتی در تخمین خط روند داده‌ها دارد که ممکن است به دلایل متعددی از جمله نبود داده کافی، ناهنجاری در داده‌ها یا تغییرات ناگهانی در روند باشد.

استفاده از الگوریتم **RLS** برای پنجره کشویی تعادلی بین سرعت سازگاری و هزینه محاسباتی فراهم می‌کند، این روش به‌ویژه برای منابع محدود و تمرکز بر روندهای اخیر کارآمد است. به نظر می‌رسد روش **RLS** برای پنجره کشویی بهترین گزینه برای این نوع داده باشد، زیرا مقدار خطا در نمودار بسیار کم است و شیب نمودار منفی و خطا در حال کاهش است.

مقایسه متد Least square (LS) و Recursive Least Squares (RLS) با عامل

RLS for Sliding window, Sliding Window, و Forgetting factor

کاربردها	معایب	مزایا	روش
پیش‌بینی تقاضا در سیستم‌های ایستا - برازش خطی	با داده‌های در حال تغییر سازگار نیست - به تمام داده‌های گذشته به طور مساوی وزن می‌دهد	ساده و کارآمد از نظر محاسباتی	Least square
پیش‌بینی قیمت سهام - ردیابی سیگنال در سیستم‌های پویا	پیچیده‌تر از LS - انتخاب عامل فراموشی مناسب می‌تواند چالش برانگیز باشد	با داده‌های در حال تغییر سازگار است - به داده‌های جدید وزن بیشتری می‌دهد	RLS with Forgetting Factor
- تشخیص ناهنجاری - پیش‌بینی کوتاه‌مدت	می‌تواند در حذف داده‌ها تند و تیز باشد - ممکن است اطلاعات ارزشمندی را از دست بدهد	به طور صریح داده‌های قدیمی را حذف می‌کند - بر روندهای اخیر تمرکز دارد	Sliding window
پیش‌بینی در سیستم‌های پویا با روندهای متغیر - ردیابی سیگنال‌های ضعیف در داده‌های پر سر و صدا	پیچیده‌تر از RLS - نیاز به تنظیم دقیق پارامترها دارد	مزایای RLS و پنجره کشویی را ترکیب می‌کند - به طور پویا با داده‌های جدید سازگار می‌شود - بر روندهای اخیر تمرکز دارد	RLS for Sliding window

جدول 2: مقایسه متد های طراحی مدل

<https://chat.openai.com/> (Chat GPT)

<https://en.wikipedia.org>

<https://matplotlib.org/> (Matplot documentation)

<https://numpy.org/> (Numpy documentation)

<https://pandas.pydata.org/> (Pandas documentation)

