



**universidad  
de león**



# **Escuela de Ingenierías Industrial, Informática y Aeroespacial**

## **GRADO EN INGENIERÍA MECÁNICA**

**Trabajo de Fin de Grado**

**DEGRADACIÓN DE MODELOS LINÉALES DEBIDO A  
ACTUALIZACIONES**

**ON THE DEGRADATION OF LINEAR MODELS DUE TO UPDATES**

**Autor: Narek Stepanyan  
Tutor: Miguel Carriegos Vieira**

**(Diciembre, 2018)**

**UNIVERSIDAD DE LEÓN**  
**Escuela de Ingenierías Industrial, Informática y**  
**Aeroespacial**

**GRADO EN INGENIERÍA MECÁNICA**  
**Trabajo de Fin de Grado**

**ALUMNO:** Narek Stepanyan

**TUTOR:** Miguel Carriegos Vieira

**TÍTULO:** Degradación de modelos lineales debido a actualizaciones

**TITLE:** On the degradation of linear models due to updates

**CONVOCATORIA:** Diciembre, 2018

**RESUMEN:**

En este trabajo se utiliza el software Matlab para estudiar si se produce una degradación del modelo lineal de Penrose (usado como clasificador lineal de patrones) debido a las actualizaciones del propio modelo con cada vector request. Se genera un modelo inicial partiendo de un training Dataset establecido arbitrariamente y después se analiza la degradación del modelo bajo diferentes condiciones y con el número de actualizaciones especificado. De los resultados empíricos obtenidos se deduce que mientras el modelo sea actualizado con los valores calculados por el modelo anterior será estable, es decir no existe variación entre el modelo inicial y el n-ésimo modelo correspondiente a la última actualización, en base a este resultado se procede a desarrollar un teorema matemático que demuestre esta afirmación de forma exacta e irrefutable.

**ABSTRACT:**

This memory deals with the degradation of linear models by means of successive updates. Matlab computer software is used to perform our experiments under different conditions for finding out model changes. Matlab code is provided as well as summaries and reports of some experiments.

**Palabras clave:**

Pseudoinversa de Penrose, Modelo lineal, Matlab, Ataque Flooding.  
Se genera un modelo inicial, partiendo de un data-set

**Firma del alumno:**

**VºBº Tutor/es:**

# ÍNDICE DE CONTENIDOS

<b>Índice de figuras .....</b>	<b>2</b>
<b>Índice de cuadros y tablas .....</b>	<b>3</b>
<b>Glosario de términos.....</b>	<b>4</b>
<b>Introducción .....</b>	<b>5</b>
1.1 Internet .....	5
1.2 Planteamiento del problema .....	6
1.3 Pseudoinversa de Penrose (Clasificador lineal de patrones).....	9
<b>Análisis del deterioro del modelo debido a las actualizaciones.....</b>	<b>11</b>
2.1 Training Dataset y obtención del modelo inicial .....	11
2.1 Modelo sin actualizar .....	15
2.2 Modelo no acumulativo .....	16
2.3 Modelo Acumulativo .....	16
2.4 Modificación del Vector request con funciones .....	18
2.5 Inserción de un vector request y su calificación del Training Dataset .....	21
2.6 Tablas con los resultados .....	23
<b>Conclusión.....</b>	<b>26</b>
<b>Teorema .....</b>	<b>27</b>
<b>Líneas de investigación futuras .....</b>	<b>29</b>
<b>Agradecimientos.....</b>	<b>30</b>
<b>Lista de referencias bibliográficas .....</b>	<b>31</b>
<b>ANEXO-A .....</b>	<b>32</b>

# ÍNDICE DE FIGURAS

Figura 1.1 Esquema de red (Redes de Computadoras, Tanenbaum Wetheral) .....	6
Figura 1.2 Interconexión de redes (Redes de Computadoras, Tanenbaum Wetheral) .....	7
Figura 1.3 Esquema de humano experto asignando valores de riesgo (Fuente propia).....	8
Figura 1.4 Esquema de humano experto y modelo (Fuente propia).....	9
Figura 1.5 Solución real y solución hallada con Pseudoinversa de Penrose (Fuente propia, Anexo A Informe1) .....	11
Figura 2.1 Modelo A (Fuente propia, Anexo-A Informe 2) .....	13
Figura 2.2 Modelo B (Fuente propia) .....	14
Figura 2.3 Modelo B (Fuente propia) .....	15
Figura 2.4 Variación modelo sin actualizar (Fuente propia) .....	16
Figura 2.5 Comparativa de variación acumulativa y no acumulativa (Fuente propia: Anexo-A informe 2) .....	18
Figura 2.6 Comparativa de variación acumulativa y no acumulativa (Fuente propia: Anexo-A informe 2) .....	18
Figura 2.7 representación de la función $g_5y$ para 40 actualizaciones (Fuente: propia Anexo-A informe 2) .....	19
Figura 2.8 representación de la función $g_3y$ para 40 actualizaciones (Fuente: propia Anexo-A informe 2) .....	20
Figura 2.9 representación de la función $g_2y$ para 40 actualizaciones (Fuente: propia Anexo-A informe 2) .....	20
Figura 2.10 representación de la función $g_1y$ para 40 actualizaciones (Fuente: propia Anexo-A informe 2) .....	21
Figura 2.11 representación de la función $g_4y$ para 40 actualizaciones (Fuente: propia Anexo-A informe 2) .....	21
Figura 2.12 representación de la función $g_4y$ para 40 actualizaciones (Fuente: propia Anexo-A informe 2) .....	22
Figura 2.13 variación generada por vector petición del data-set (Fuente propia Anexo-A Informe-2).....	23
Figura 2.14 variación generada por vector petición del data-set (Fuente propia) .....	23

# ÍNDICE DE CUADROS Y TABLAS

Tabla 2.1 Deterioro Modelo A (Fuente Propia) .....24

Tabla 2.2 Deterioro Modelo B (Fuente Propia) .....24

Tabla 2.3 Deterioro Modelo C (Fuente Propia) .....25

# GLOSARIO DE TÉRMINOS

## Training Dataset

Conjunto de datos inicial compuesto por vectores request y sus respectivas calificaciones con los que se obtiene el primer modelo  $\begin{bmatrix} \alpha_0 \\ \beta_0 \end{bmatrix}$ .

## Request

Paquete de información enviado por un cliente a través de internet al servidor con el que desea comunicarse.

## Vector petición o request $\begin{bmatrix} M \\ \underline{f} \end{bmatrix}$

Vector fila  $\underline{f}$  que representa numéricamente la información contenida en el request. El conjunto de vectores request forma la matriz request  $M$ .

## Calificación

Valor numérico asignado por el humano experto a cada vector request  $\vec{v}$ .

## Calificación calculada

Calificación aproximada calculada por el modelo para el vector request  $\underline{V}$

## Deterioro

Es la diferencia en valor absoluto entre el primero modelo  $X_0$  y la última actualización del modelo.

$$d = |X_n| - |X_0| = \left| \begin{bmatrix} \alpha_n \\ \beta_n \end{bmatrix} \right| - \left| \begin{bmatrix} \alpha_0 \\ \beta_0 \end{bmatrix} \right|$$

## Vector representación

Vector fijo  $V_{rep}$  escogido arbitrariamente para graficar la variación de cada modelo  $X_i$  con cada  $i$  actualización.

# Introducción

## 1.1 INTERNET

Aunque el objetivo principal de este trabajo no es el estudio de las redes de comunicación, sí que creo conveniente definir de forma superficial algunos elementos y la estructura básica de internet ya que una de las aplicaciones del modelo estudiado en este trabajo es el de ser un asignador lineal de valores de riesgo a los diferentes clientes que mandan una petición a un determinado servidor.

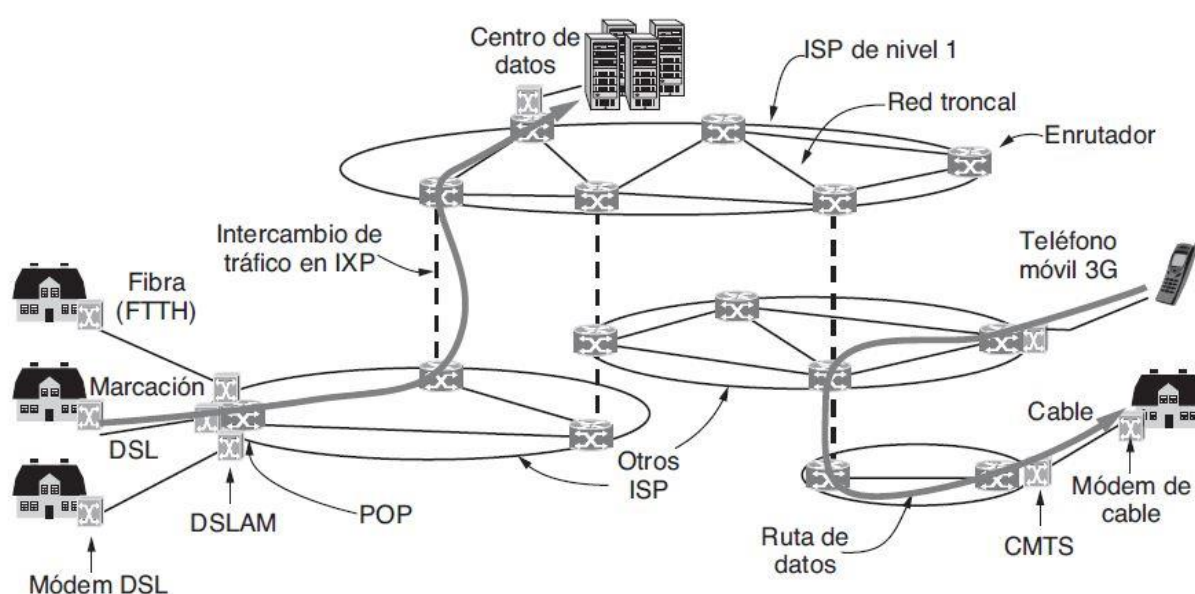


Figura 1.1 Esquema de red (Redes de Computadoras, Tanenbaum Wetheral)

Como se muestra en la figura 1.1 la conexión entre un cliente o usuario particular y el servidor del cual obtiene o manda información se realiza a través de varios niveles. El principal intermediario es el ISP o Proveedor de servicio de internet, es la empresa con la que contratamos el acceso a internet. El medio para transmitir la información puede ser inalámbrico, es decir mediante ondas electromagnéticas o mediante un medio físico como cable de cobre, cable coaxial, fibra óptica o una combinación de ambos. Para solicitar información de un servidor, el usuario particular manda un paquete de información con su IP y más información que es redirigido hasta el servidor que contiene la información deseada, de esta manera se establece un flujo de intercambio de información entre todos los elementos de la red.

En la figura 1.2 se muestra un ejemplo de interconexión entre diferentes niveles de redes de un país y también la conexión entre la red de un país con otro.

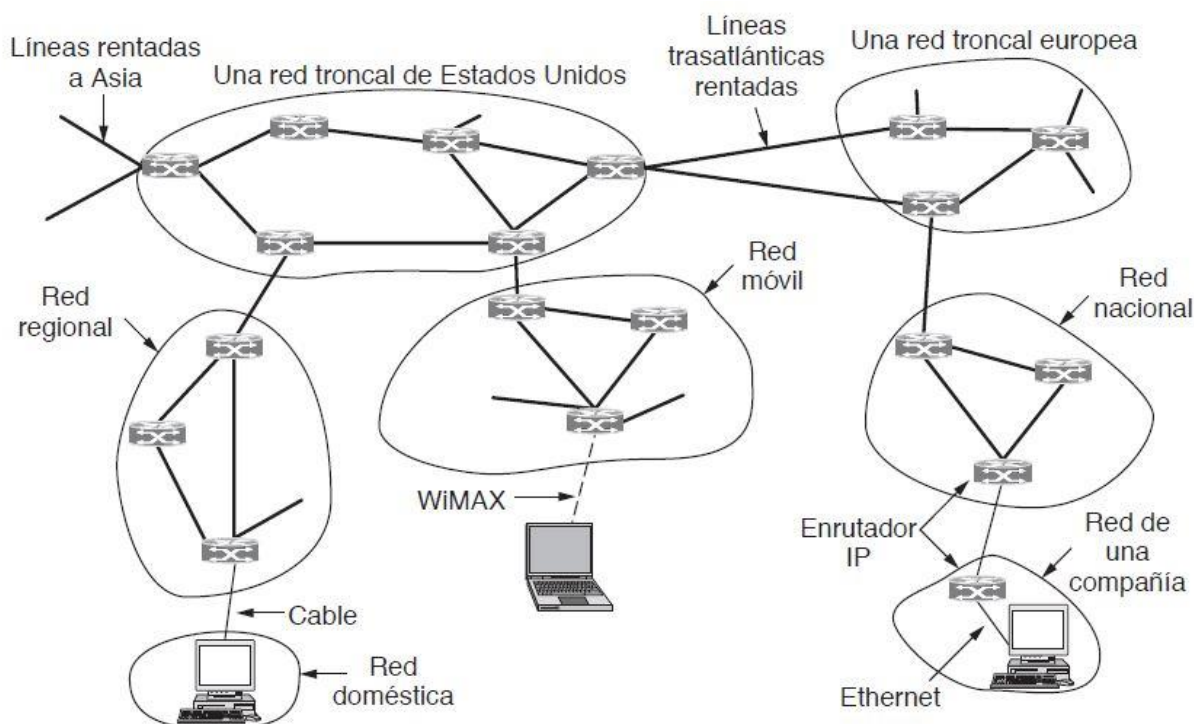


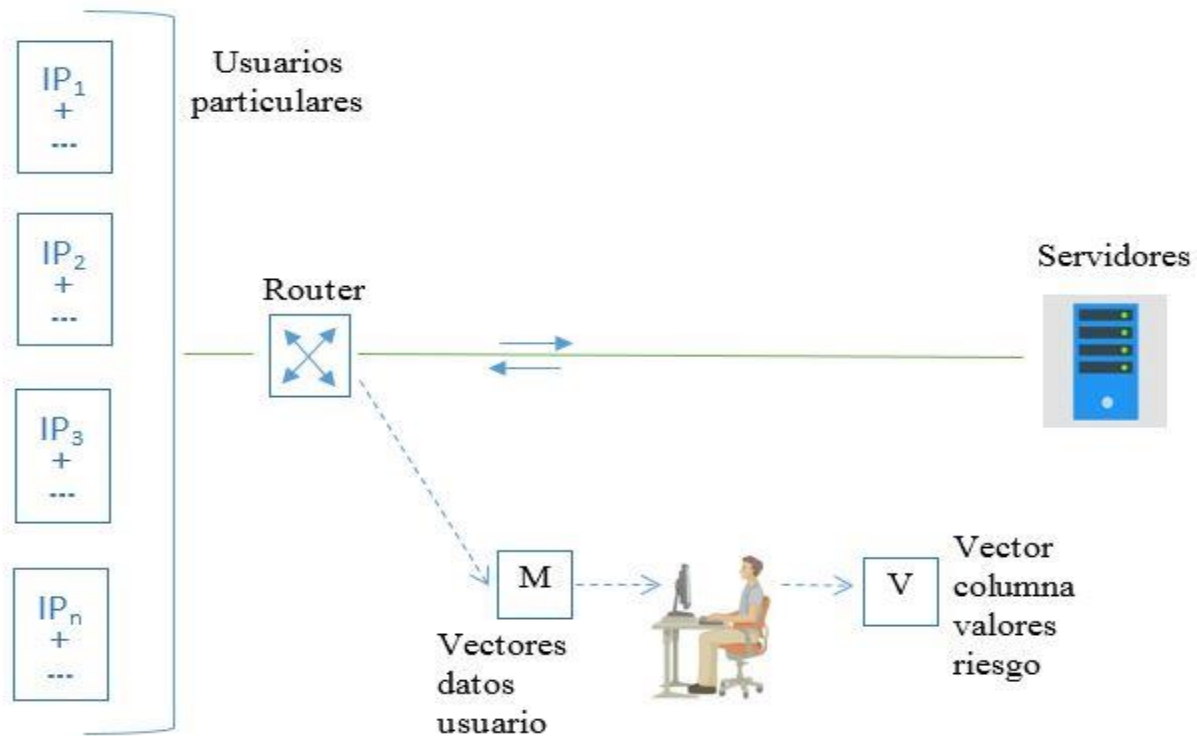
Figura 1.2 Interconexión de redes (Redes de Computadoras, Tanenbaum Wetheral)

## 1.2 PLANTEAMIENTO DEL PROBLEMA

El objetivo de este trabajo es estudiar si es posible sustituir, mediante un clasificador lineal discriminante, la labor de un humano experto que asigna un valor de riesgo o peligrosidad a las solicitudes de conexión o petición de información que un supuesto usuario realiza a un servidor. Como se muestra en la figura 1.3 una cantidad  $n$  de usuarios pueden solicitar información del servidor, lo cual se realiza según alguno de los protocolos de conexión, por ejemplo el TCP/IP. Al realizarse esta petición o request la información llega en forma de paquetes, cada paquete contiene más información a parte de la IP, gracias a parte de esta información los routers direccionan el paquete desde el ordenador del usuario hasta el servidor de destino y después este servidor envía la respuesta al usuario de la misma manera. Estos paquetes de información están constituidos de varias capas y un atacante puede introducir código con fines de dañar o robar información del servidor. En el paso  $M$  a este paquete de información se le asigna un vector de  $i$  componentes que describe el contenido de la solicitud de forma numérica, posteriormente los vectores asociados a cada petición, son examinados



por un humano experto que les asigna un valor de riesgo. Es de destacar que el tipo de protocolo de comunicación no influye en el modelo, ya que el vector request de entrada al modelo puede ser de  $n$  componentes en lugar de 2 como se estudia en este trabajo.



*Figura 1.3 Esquema de humano experto asignando valores de riesgo (Fuente propia)*

En la figura 1.4 se muestra la calificación de riesgo hecha de forma automática por el modelo y la realizada por el humano experto. Lo que se desea averiguar es si actualizando el modelo con cada nuevo request el modelo inicial sufre una variación. Cabe destacar que el modelo se calcula a partir de Dataset inicial, que consiste en un conjunto de vectores request y sus correspondientes calificaciones, asignadas por el humano experto.

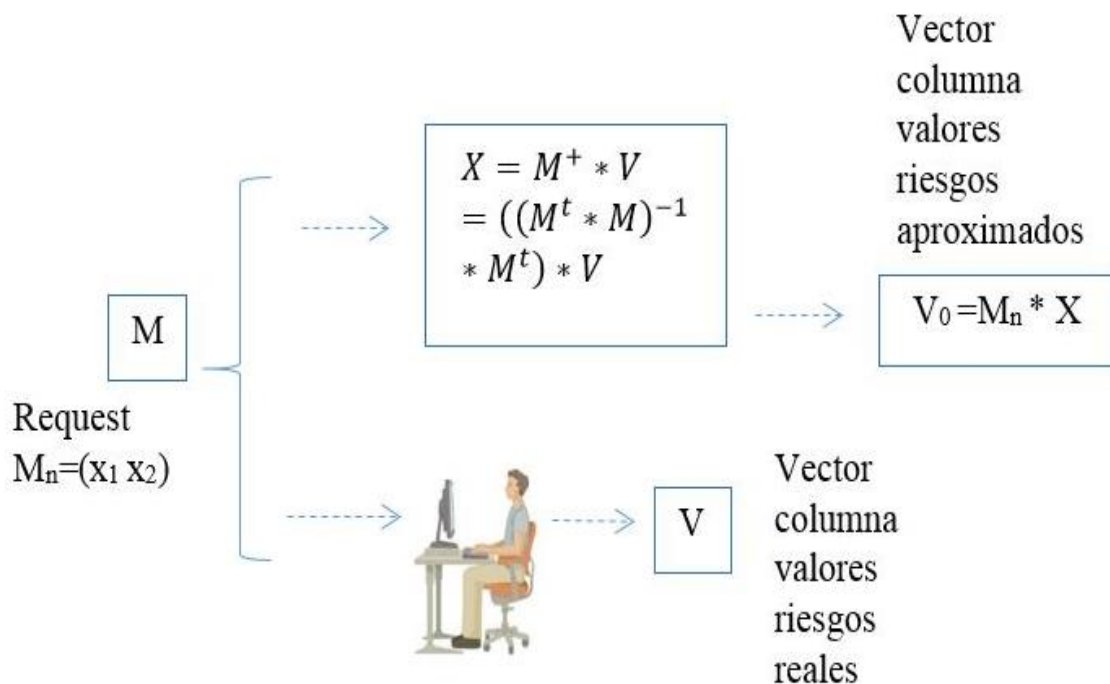


Figura 1.4 Esquema de humano experto y modelo (Fuente propia)

El proceso de asignar un valor de riesgo no sería viable hacerlo en tiempo real, ya que el flujo de interacción entre cliente y servidor sería muy lento si cada petición tuviese que ser examinada por el humano antes de pasar al servidor. Esa asignación de un valor de riesgo se realizaría después, es decir se almacenan los vectores petición y el humano los revisa y asigna a cada uno un valor de riesgo, de esta manera si se detecta una petición no usual se le puede denegar el acceso al servidor a esa dirección IP.

Para sustituir la labor del humano es necesario obtener un modelo  $X$  que de los valores  $V$ . Este modelo se puede obtener de una base de datos generada por calificaciones dadas por el humano experto. El modelo  $X$  no se puede obtener de forma directa según la ecuación (1.4), ya que  $M$  no es una matriz cuadrada y no tiene inversa. Este hecho se debe a que se tienen muchos más usuarios haciendo peticiones al servidor que componentes en el vector petición, esto se aprecia en la ecuación (1.3). Por este motivo se usa el modelo lineal de Penrose o Pseudoinversa de Penrose.

$$M_{fxc} * X_{cx1} = V_{fx1} \quad (1.1)$$

$$\begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \\ \cdot & \cdot \\ \cdot & \cdot \\ M_{f1} & M_{f2} \end{bmatrix} * \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} V_1 \\ V_2 \\ \cdot \\ \cdot \\ V_f \end{bmatrix} \quad (1.2)$$

$$f > c \quad (1.3)$$

$$X = M^{-1} * V ; \quad \nexists M^{-1} \quad (1.4)$$

Se quiere averiguar de forma empírica si el modelo es estable, es decir si no varía, ante un ataque Flooding, que consiste en hacer un gran número de peticiones al servidor por parte de un atacante, de manera que para el servidor esa petición pase de encontrarse fuera del perímetro de seguridad a incluirse en él. Para hacer este experimento usare la herramienta matlab para programar un algoritmo que me actualice el modelo tantas veces como el usuario le indique. Este código se adjunta en el anexo A.

Existen tres formas de calcular con el modelo las calificaciones de cada request, más adelante se detallara el algoritmo de cálculo para cada método y se expondrán los resultados obtenidos con cada método de cálculo de calificaciones.

- Sin actualizar el modelo
- Actualización no acumulativa
- Actualización acumulativa
- Modificación del Vector request con funciones
- Inserción de un vector request y su calificación del training Dataset

### 1.3 PSEUDOINVERSA DE PENROSE (CLASIFICADOR LINEAL DE PATRONES)

Las matrices que no son cuadradas no tienen inversa, para estas se define el concepto de Pseudoinversa, este es su desarrollo cronológico: E. H. Moore 1920, Arne Bjerhammar 1951 y Roger Penrose en 1955. La Pseudoinversa de Penrose de la matriz  $A$  se calcula de acuerdo con la ecuación (1.5) por lo

consiguiente el modelo  $X$  se halla según la ecuación (1.6). La condición para poder hallar  $X$  es que la matriz  $A$  tenga rango máximo.

$$\begin{aligned} A^+ &= (A^t * A)^{-1} * A^t ; & A_{n \times m}; & n > m \\ A^+ &= A^t * (A * A^t)^{-1} ; & A_{n \times m}; & n < m \end{aligned} \quad (1.5)$$

$$M * X = V \quad (1.6)$$

$$X = M^+ * V = ((M^t * M)^{-1} * M^t) * V$$

La solución hallada con la Pseudoinversa de Penrose es la aproximación que genera soluciones cuyo error cuadrático medio es mínimo, es decir la norma al cuadrado del vector que contiene el error de cada punto respecto al valor real es la mínima posible, estos puntos aproximados están confinados en el plano que es solución lineal aproximada del problema.

Gráficamente la solución obtenida es el plano que cuya distancia media ortogonal a la nube de puntos, que es la solución real es mínima, Esto se puede apreciar en el siguiente gráfico. También sería posible utilizar un modelo no lineal, en ese caso la ecuación del modelo, en lugar de ser un plano sería una superficie curva.

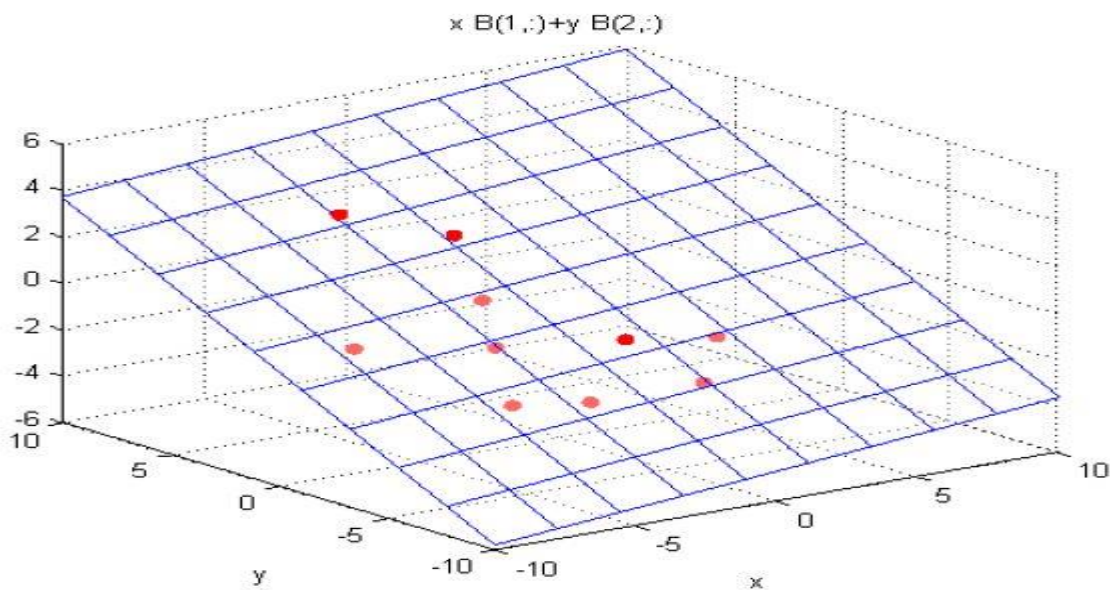


Figura 1.5 Solución real y solución hallada con Pseudoinversa de Penrose (Fuente propia, Anexo A Informe1)

# Análisis del deterioro del modelo debido a las actualizaciones

En este apartado se procede a implementar los códigos adjuntados en el anexo-A para estudiar si el modelo inicial se deteriora o no, esto se hace analizando los resultados y graficas obtenidos al ejecutar el código. Se dedica un apartado a cada algoritmo de actualización, adjuntando los gráficos correspondientes a cada tipo, finalmente se recopilan en una tabla los deterioros obtenidos para cada tipo de modelo inicial: A, B, C y para tres valores diferentes de actualización: 40, 400, 4000. Después en el apartado de conclusiones se analizaran estos resultados. Las representaciones graficas se han hecho con el modelo B y 40 actualizaciones.

A pesar de que en este documento solo se simulen tres modelos y un máximo de 4000 iteraciones, el código está redactado de tal forma que el usuario puede generar modelos de la dimensión deseada de forma aleatoria o bien introduciéndolos manualmente.

## 2.1 TRAINING DATASET Y OBTENCIÓN DEL MODELO INICIAL

Se calculan tres modelos iniciales diferentes, usando la ecuación (1.6) para estudiar si el deterioro del modelo cambia dependiendo del modelo inicial que se elige. Para ello usamos 3 Dataset diferentes.

- **Modelo inicial A: 3 puntos**

$$X = M^+ * V = ((M^t * M)^{-1} * M^t) * V \quad (2.1)$$

$$M = \begin{bmatrix} 1 & 1 \\ 2 & 3 \\ 3 & 3 \end{bmatrix}; \quad \begin{bmatrix} 2 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.30 & -0.99 & 0.89 \\ -0.20 & 0.99 & -0.59 \end{bmatrix} * \begin{bmatrix} 2 \\ 4 \\ 6 \end{bmatrix} \quad (2.2)$$

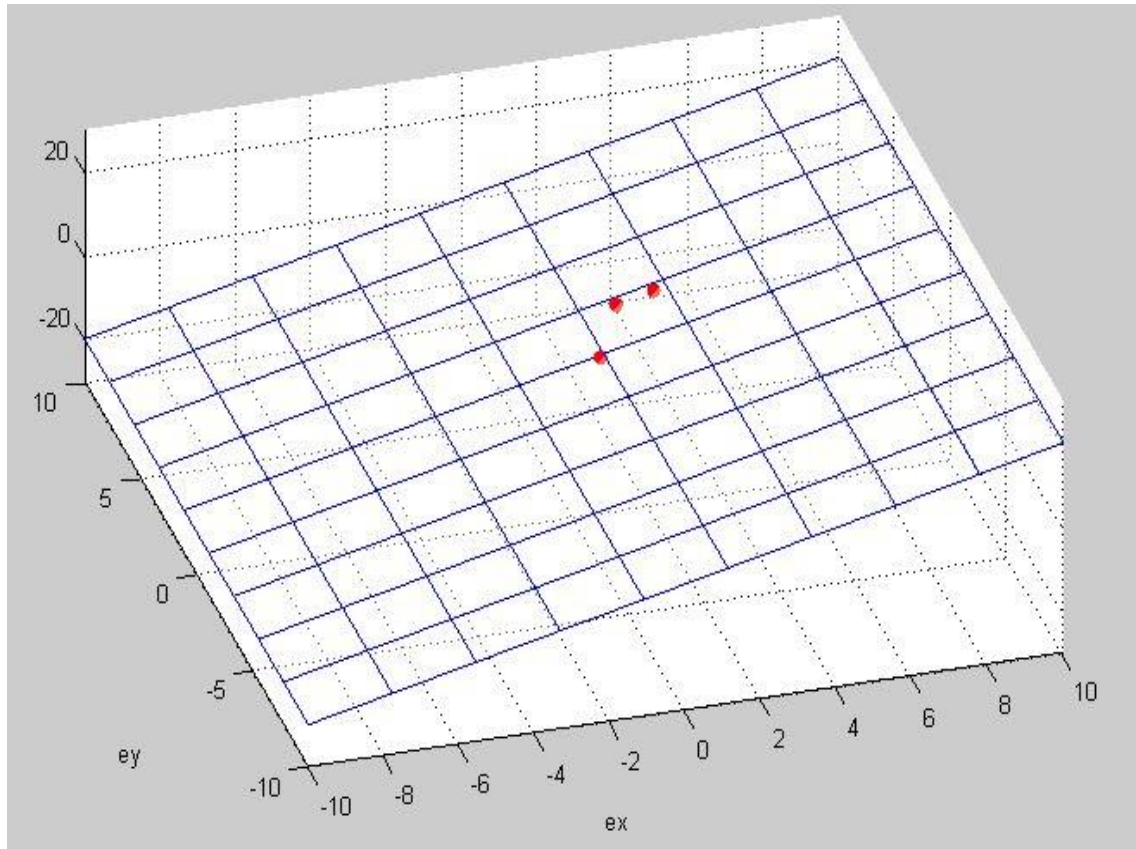


Figura 2.1 Modelo A (Fuente propia, Anexo-A Informe 2)

En este modelo al calcularse a partir de tres puntos siempre será posible encontrar un plano que los contenga a los 3. En los modelos B y C no es posible obtener un plano que cumpla la condición de contener a todos los puntos.

- **Modelo inicial B: 5 puntos**

$$M = \begin{bmatrix} -4 & 1 \\ 3 & -3 \\ -2 & 2 \\ 0 & 2 \\ -4 & 3 \end{bmatrix}; \quad V = \begin{bmatrix} -1 \\ -5 \\ -3 \\ 5 \\ -4 \end{bmatrix}; \quad X = \begin{bmatrix} 1.2593 \\ 1.5748 \end{bmatrix} \quad (2.3)$$

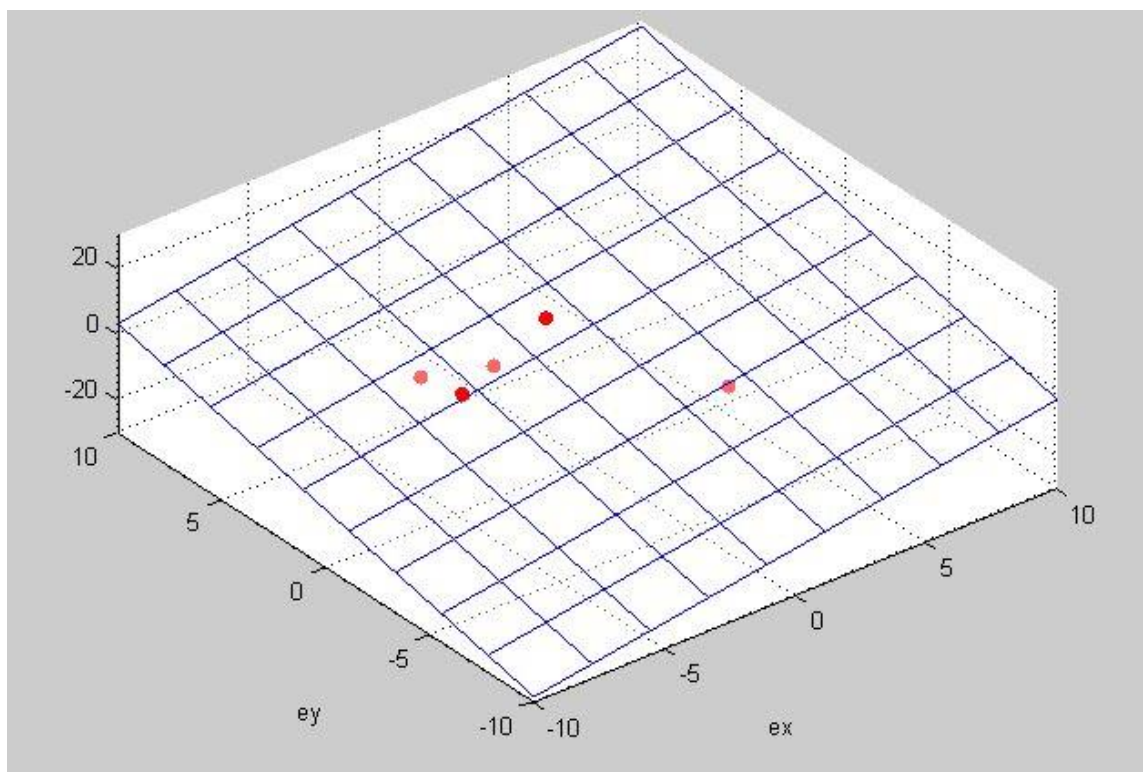


Figura 2.2 Modelo B (Fuente propia)



- **Modelo inicial C: 10 puntos**

$$M = \begin{bmatrix} 20 & -12 \\ 11 & 2 \\ 3 & 5 \\ 18 & -19 \\ 3 & 5 \\ -20 & -6 \\ -16 & -18 \\ 15 & 0 \\ -1 & -13 \\ 14 & -15 \end{bmatrix}; \quad V = \begin{bmatrix} 3 \\ 2 \\ 2 \\ 1 \\ 7 \\ 3 \\ 6 \\ 7 \\ 5 \\ 6 \end{bmatrix}; \quad X = \begin{bmatrix} 0.0457 \\ -0.2074 \end{bmatrix} \quad (2.4)$$

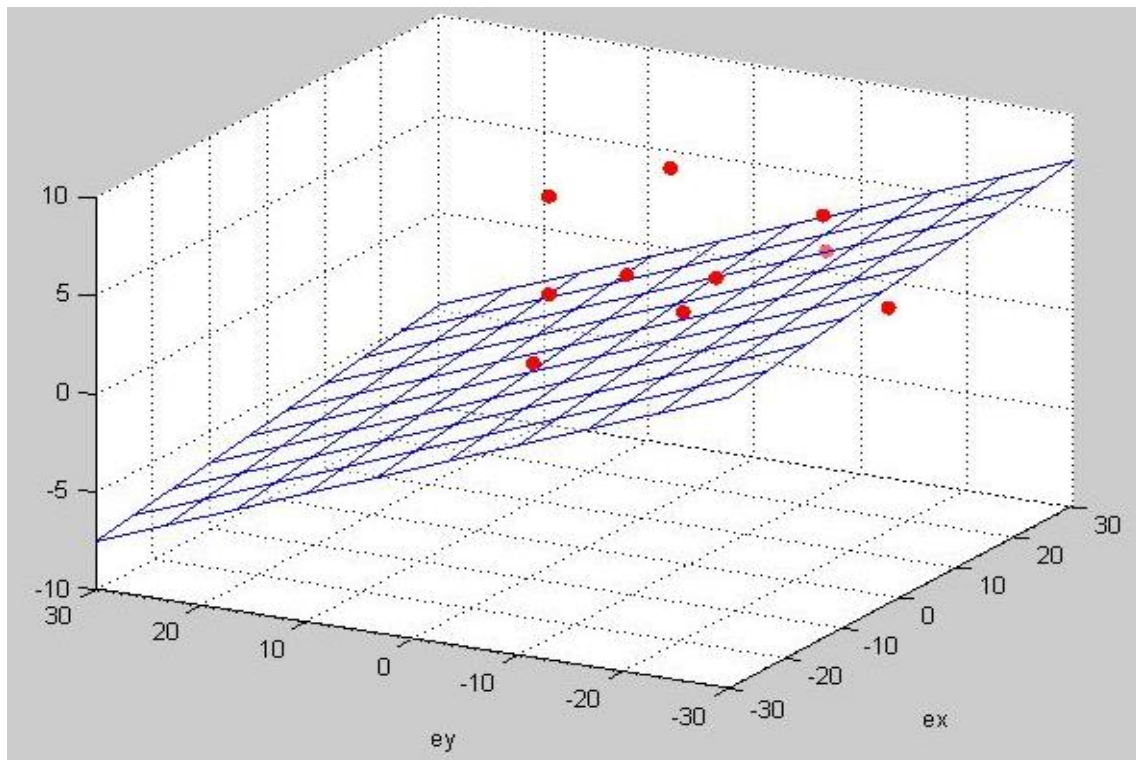


Figura 2.3 Modelo B (Fuente propia)



## 2.1 MODELO SIN ACTUALIZAR

Como es de esperar si el modelo no se actualiza para cada request, no se deteriora como se comprueba en la figura 2.4

$$\begin{aligned} X_0 &= (M)^+ * V \\ X_1 &= \begin{bmatrix} M \\ f_{m+1} \end{bmatrix}^+ * \begin{bmatrix} V \\ f_{m+1} * X_0 \end{bmatrix} \\ X_2 &= \begin{bmatrix} M \\ f_{m+2} \end{bmatrix}^+ * \begin{bmatrix} V \\ f_{m+2} * X_0 \end{bmatrix} \\ X_i &= \begin{bmatrix} M \\ f_{m+i} \end{bmatrix}^+ * \begin{bmatrix} V \\ f_{m+i} * X_0 \end{bmatrix} \end{aligned} \quad (2.4)$$

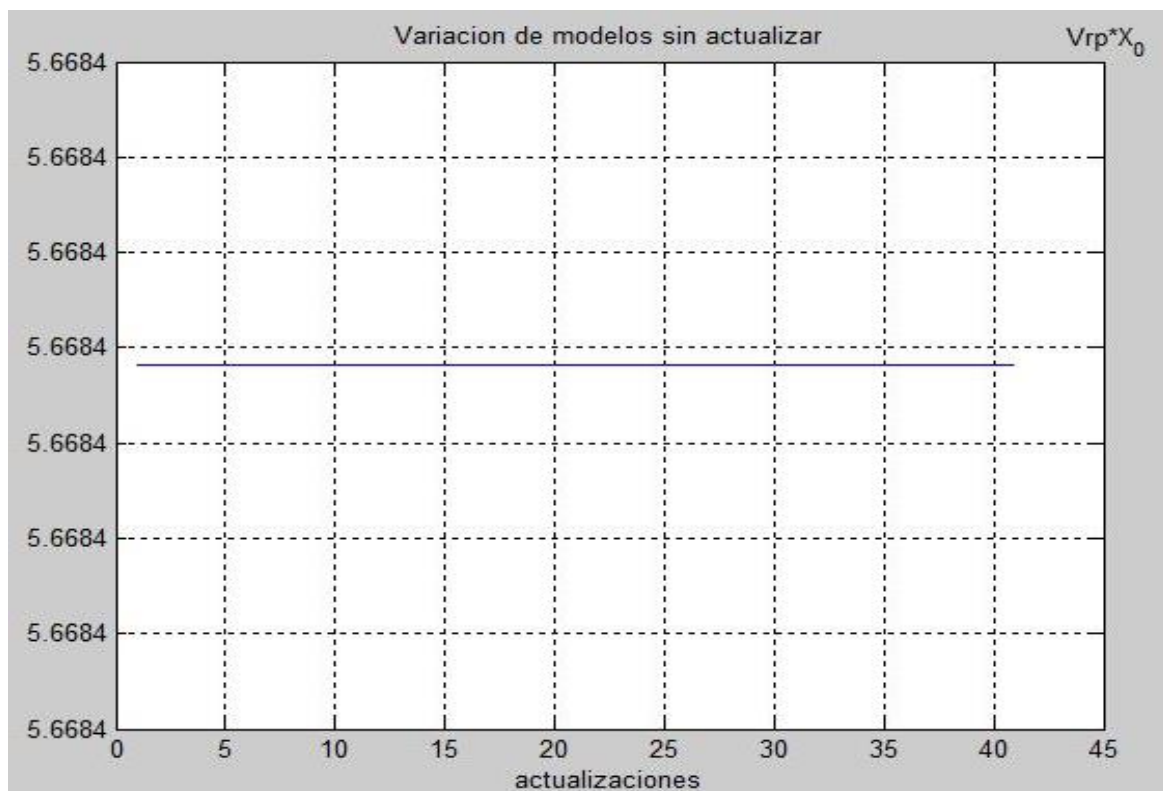


Figura 2.4 Variación modelo sin actualizar (Fuente propia)

## 2.2 MODELO NO ACUMULATIVO

Para actualizar el modelo tantas veces como vectores petición ( $f_{m+i}$ ) lleguen, es posible hacerlo sin acumular los vectores petición anteriores ni los valores de calificación ( $f_{m+i} * X_{i-1}$ ) anteriores.

$$\begin{aligned}
 X_0 &= (M)^+ * V \\
 X_1 &= \begin{bmatrix} M \\ f_{m+1} \end{bmatrix}^+ * \begin{bmatrix} V \\ f_{m+1} * X_0 \end{bmatrix} \\
 X_2 &= \begin{bmatrix} M \\ f_{m+2} \end{bmatrix}^+ * \begin{bmatrix} V \\ f_{m+2} * X_1 \end{bmatrix} \\
 X_i &= \begin{bmatrix} M \\ f_{m+i} \end{bmatrix}^+ * \begin{bmatrix} V \\ f_{m+i} * X_{i-1} \end{bmatrix}
 \end{aligned} \tag{2.5}$$

## 2.3 MODELO ACUMULATIVO

En el caso de actualizar los modelos teniendo en cuenta los vectores petición anteriores y las calificaciones halladas por el modelo anterior el procedimiento se haría usando la ecuación (2.6)

$$\begin{aligned}
 X_0 &= (M)^+ * V \\
 X_1 &= \begin{bmatrix} M \\ f_{m+1} \end{bmatrix}^+ * \begin{bmatrix} V \\ f_{m+1} * X_0 \end{bmatrix} \\
 X_2 &= \begin{bmatrix} M \\ f_{m+1} \\ f_{m+2} \end{bmatrix}^+ * \begin{bmatrix} V \\ f_{m+1} * X_0 \\ f_{m+2} * X_1 \end{bmatrix} \\
 X_i &= \begin{bmatrix} M \\ f_{m+1} \\ f_{m+2} \\ \vdots \\ f_{m+i} \end{bmatrix}^+ * \begin{bmatrix} V \\ f_{m+1} * X_0 \\ f_{m+2} * X_1 \\ \vdots \\ f_{m+i} * X_{i-1} \end{bmatrix}
 \end{aligned} \tag{2.6}$$

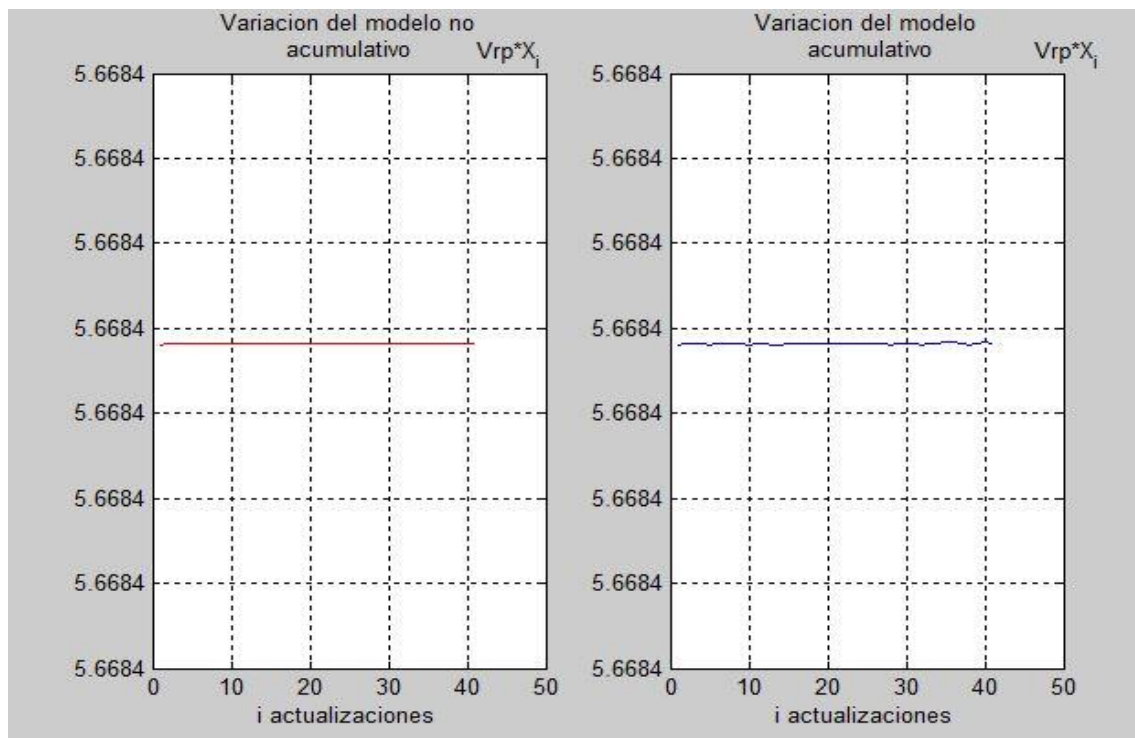


Figura 2.5 Comparativa de variación acumulativa y no acumulativa 40 actualizaciones (Fuente propia: Anexo-A informe 2)

Como se observa en la figura 2.5 y 2.6 ambos deterioros parecen nulos, incluso aumentando el número de actualizaciones a 4000, el modelo acumulativo sigue sin variar de forma significativa, la variación en vista en la gráfica es menor del orden  $10^{-10}$  y se asocia la error de redondeo del ordenador, como se demostrara en el apartado de conclusiones y teorema.

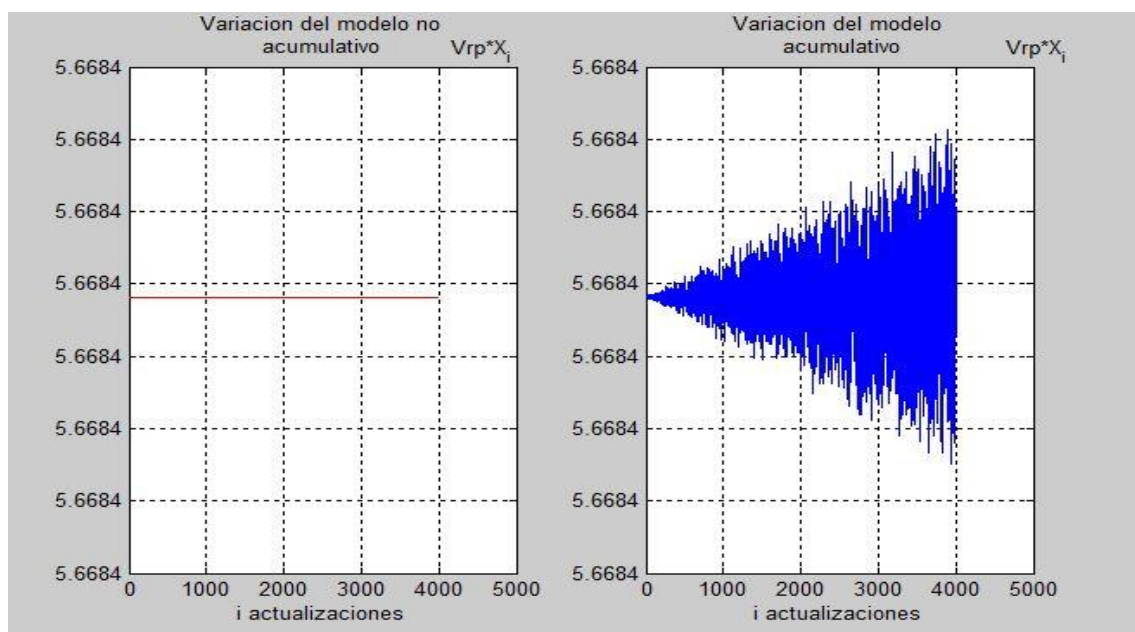


Figura 2.6 Comparativa de variación acumulativa y no acumulativa 4000 actualizaciones (Fuente propia: Anexo-A informe 2)

## 2.4 MODIFICACIÓN DEL VECTOR REQUEST CON FUNCIONES

Definimos cuatro funciones que van a multiplicar a cada vector petición para ver cómo afecta al comportamiento del modelo.

$$f_{m+i} = (f_{m+i}) * g(y) \quad (2.7)$$

$$X_0 = (M)^+ * V$$

$$X_i = \begin{bmatrix} M \\ (f_{m+1}) * g(y_1) \\ \vdots \\ (f_{m+i}) * g(y_i) \end{bmatrix}^+ * \begin{bmatrix} V \\ (f_{m+1}) * g(y_1) * X_0 \\ \vdots \\ (f_{m+i}) * g(y_i) * X_{i-1} \end{bmatrix}$$

$$g_1(y) = \sin y; \quad g_2(y) = 5y; \quad g_3(y) = \ln y; \quad g_4(y) \quad (2.8)$$

La función  $g_4(y)$  ha sido creada como una función nueva, su código está en el informe 3, anexo A.



Figura 2.7 representación de la función  $g_5(y)$  para 40 actualizaciones (Fuente: propia Anexo-A informe 2)

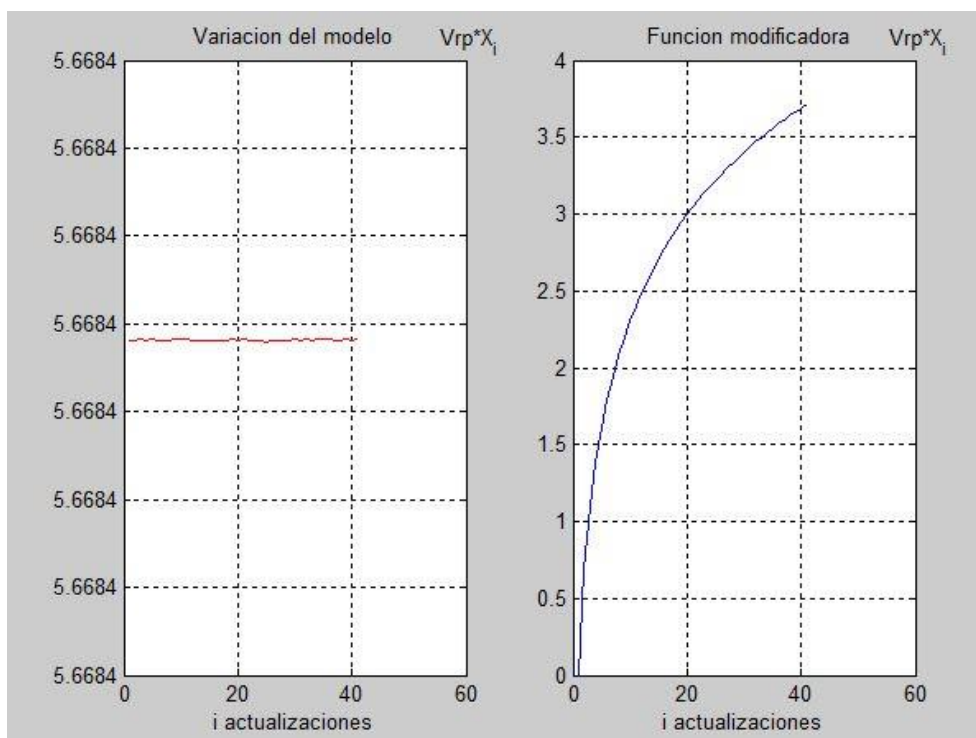


Figura 2.8 representación de la función  $g_3(y)$  para 40 actualizaciones (Fuente: propia Anexo-A informe 2)

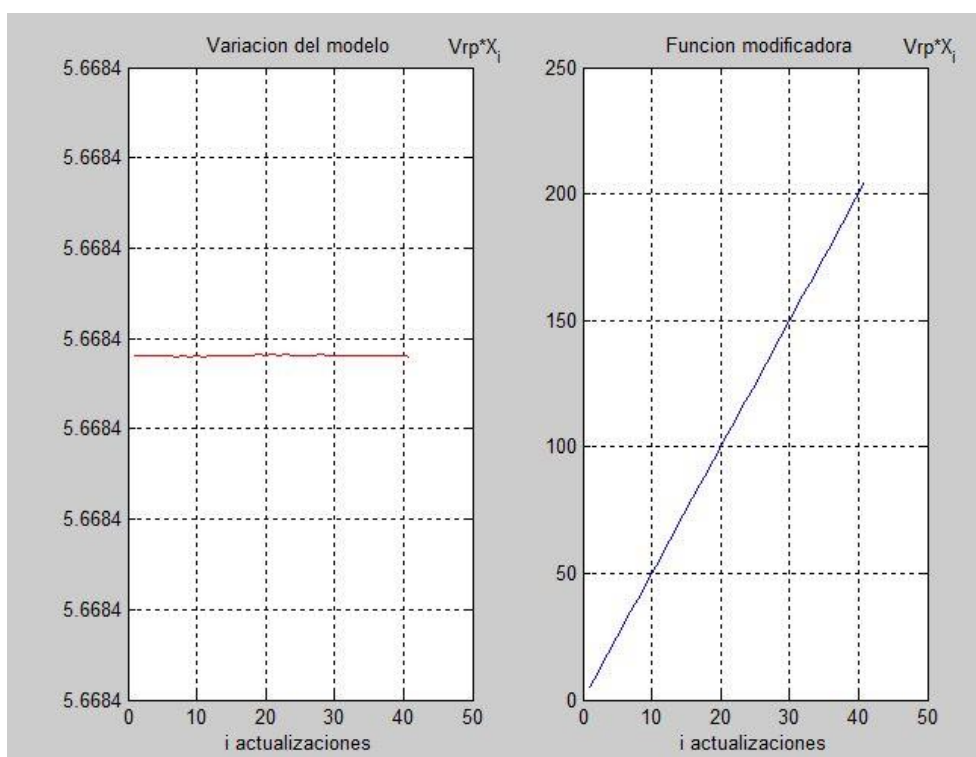


Figura 2.9 representación de la función  $g_2(y)$  para 40 actualizaciones (Fuente: propia Anexo-A informe 2)

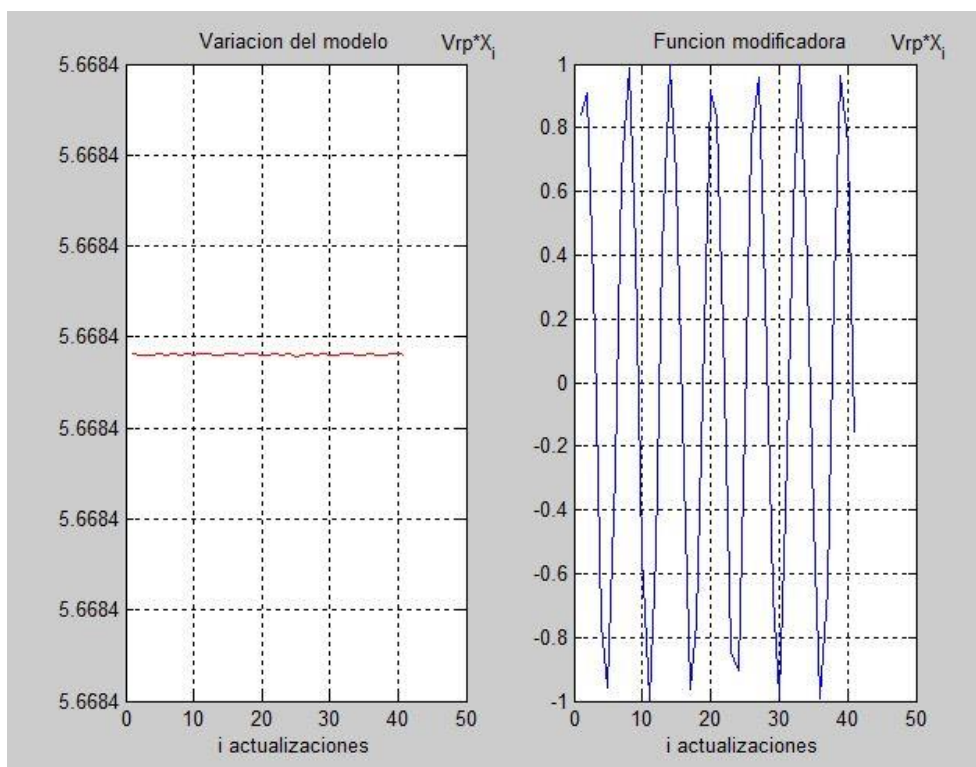


Figura 2.10 representación de la función  $g_1(y)$  para 40 actualizaciones (Fuente: propia Anexo-A informe 2)

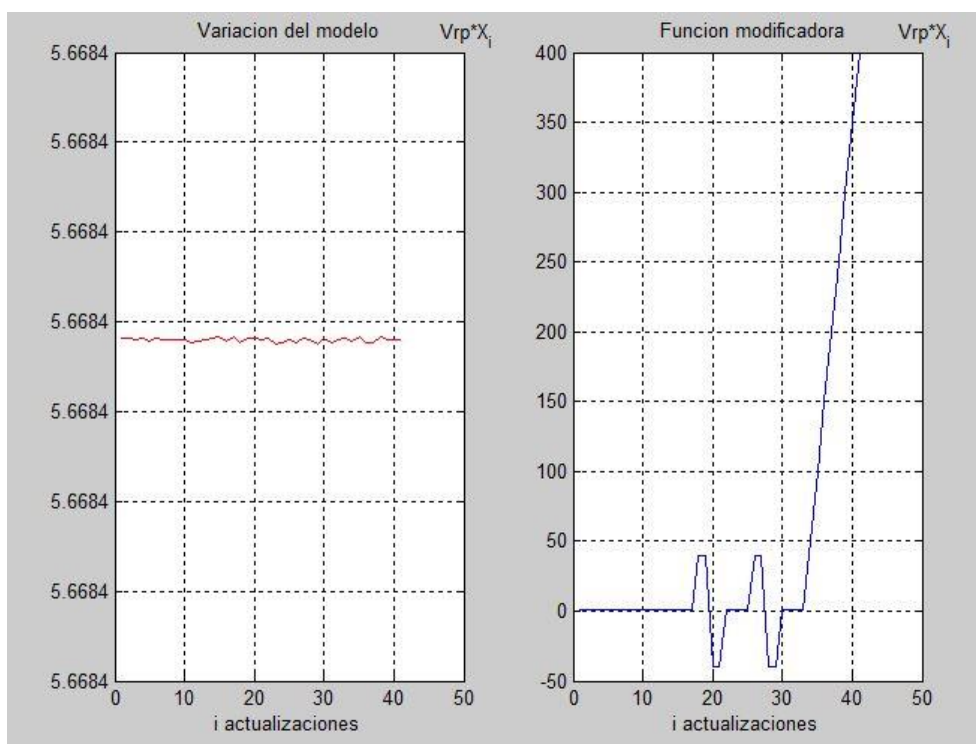


Figura 2.11 representación de la función  $g_4(y)$  para 40 actualizaciones (Fuente: propia Anexo-A informe 2)



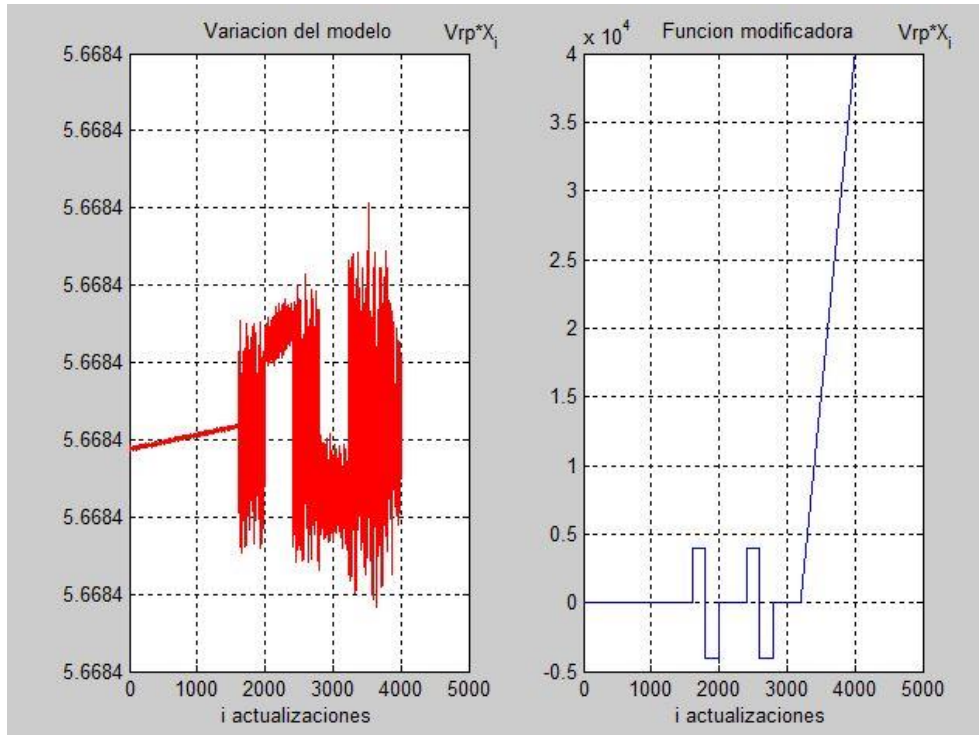


Figura 2.12 representación de la función  $g_4(y)$  para 40 actualizaciones (Fuente: propia Anexo-A informe 2)

En la figura 2.11 ocurre el mismo fenómeno que en la 2.6, no hay variación.

## 2.5 INSERCIÓN DE UN VECTOR REQUEST Y SU CALIFICACIÓN DEL TRAINING DATASET

Esta modificación consiste en introducir como vector petición un de los vectores del data-set inicial, de manera que su calificación no necesita ser calculada ya que se conoce, porque fue asignada por el humano experto como valor real. Representación en la ecuación (2.8)

Este vector perturbación se introduce cada un  $1/4$  del número total de actualizaciones, como se observa en la figura 2.4 este valor produce una variación del modelo, el número de actualizaciones es 400, por lo tanto como se aprecia en la figura las variaciones ocurren en 100, 200, 300 y 400.

$$X_i = \begin{bmatrix} M \\ f_{m+1} \\ f_{m+2} \\ f_{m+i} \\ f_{m_{set}} \end{bmatrix}^+ * \begin{bmatrix} V \\ f_{m+1} * X_0 \\ f_{m+2} * X_1 \\ f_{m+2} * X_{i-1} \\ V_{set} \end{bmatrix} \quad (2.9)$$

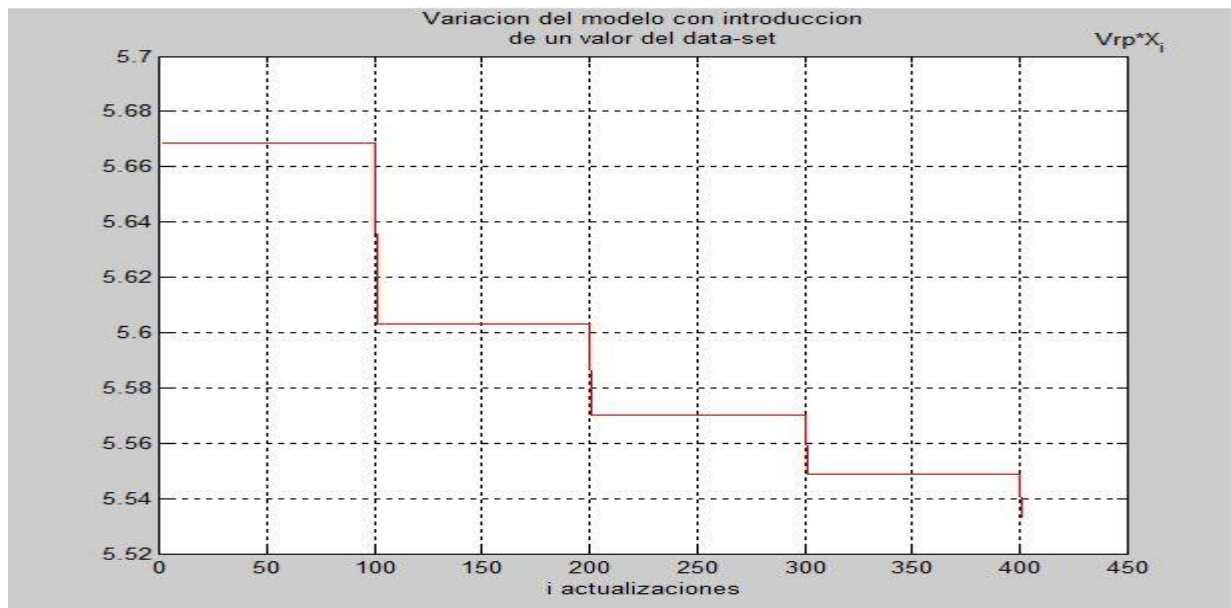


Figura 2.13 variación generada por vector petición del Dataset (Fuente propia Anexo-A Informe-2)

En la Figura 2.6 se muestra una variación generada por la introducción de un vector petición perteneciente al Dataset inicial y su respectiva calificación, se observa que el deterioro es mayor aunque el número de actualizaciones es 40. Esto es debido a que la perturbación se hace más significativa ya que el número de actualizaciones no es tan grande como en la Figura 2.5.

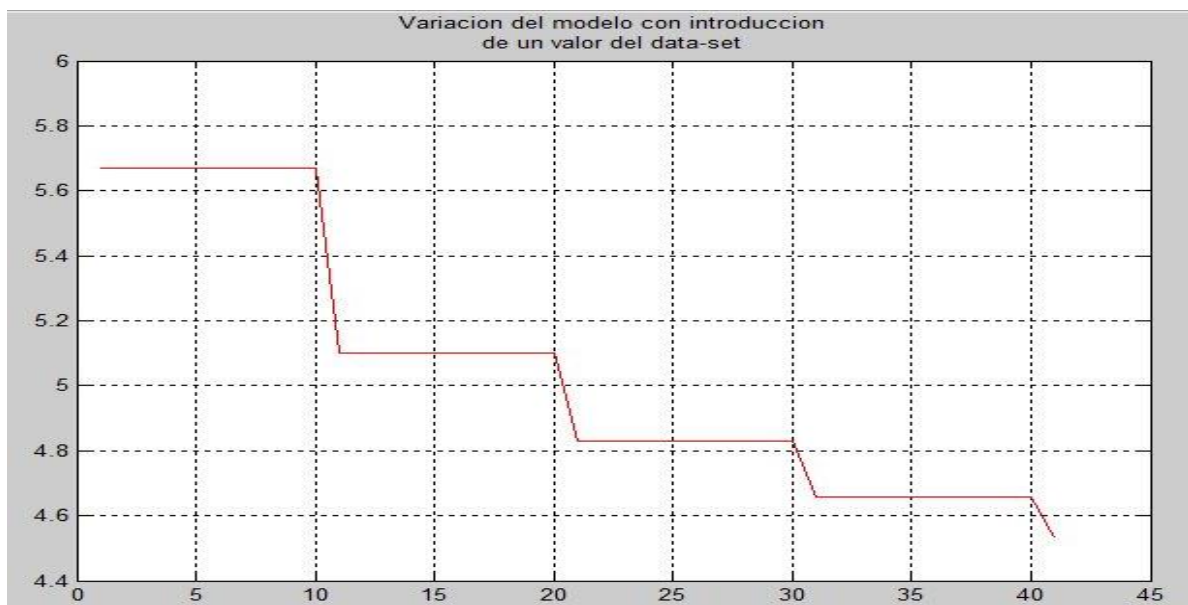


Figura 2.14 variación generada por vector petición del Dataset (Fuente propia Anexo-A Informe-2)



La diferencia entre los deterioros de las figuras 2.5 y 2.6 se puede justificar de manera cuantitativa teniendo en cuenta el porcentaje que representan los vectores request del Dataset respecto al número de actualizaciones y sabiendo que en ambos casos el vector request del data set se introduce cada  $\frac{1}{4}$ .

$$\begin{aligned} d_{40} &\approx 5.6 - 4.6 = 1 & P_{40} &= \frac{4}{40} * 100 = 10\% \\ d_{400} &\approx 5.6 - 5.4 = 0.2 & P_{400} &= \frac{4}{400} * 100 = 1\% \\ d_{40} &> d_{400} & P_{40} &> P_{400} \end{aligned} \quad (2.9)$$

## 2.6 TABLAS CON LOS RESULTADOS

Definición del deterioro o variación del modelo final  $X_f$  respecto al inicial,  $X_0$  y su respectivo criterio de signos.

$$|X_f| - |X_0| = \begin{bmatrix} |\alpha_f| & - & |\alpha_0| \\ |\beta_f| & - & |\beta_0| \end{bmatrix} = \begin{bmatrix} d_\alpha \\ d_\beta \end{bmatrix} \quad (2.10)$$

$$\begin{aligned} d_\alpha \text{ ó } d_\beta &> 0 \rightarrow \alpha_f > \alpha_0 \text{ ó } \beta_f > \beta_0 \\ d_\alpha \text{ ó } d_\beta &< 0 \rightarrow \alpha_f < \alpha_0 \text{ ó } \beta_f < \beta_0 \end{aligned} \quad (2.11)$$

Para representar gráficamente la variación de los modelos respecto a cada número de actualización se utiliza un vector representación  $V_{rep} = (a, b)$ , si se da el caso de que los dos términos del vector representación son iguales y  $d_\alpha = -d_\beta$  aunque se produjese un deterioro del modelo, gráficamente no se podría apreciar, cómo se comprueba en la ecuación (2.12) y (2.13).

$$V_{rep} = (a, b); \quad a = b; \quad d_\alpha = -d_\beta \quad (2.12)$$

$$[a \quad b] * \begin{bmatrix} \alpha_0 \\ \beta_0 \end{bmatrix} = a * \alpha_0 + b * \beta_0 = a * (\alpha_0 + \beta_0) \quad (2.12)$$

$$[a \quad b] * \begin{bmatrix} \alpha_0 + d_\alpha \\ \beta_0 + d_\beta \end{bmatrix} = a * (\alpha_0 + d_\alpha) + a * (\alpha_0 - d_\alpha) = a * (\alpha_0 + \beta_0) \quad (2.13)$$

Como se aprecia en la tabla 2.1, 2.2 y 2.3 los deterioros son todos prácticamente 0, el aumento del valor del deterioro al aumentar las actualizaciones de 40 a 4000 para el modelo B y con la función modificadora  $g_3(y)$  va desde  $10^{-14}$  a  $10^{-9}$ , esto es debido a que el ordenador tiene una precisión finita y el error producido al hacer el redondeo en cada iteración se acumula y como resultado se tiene ese aumento del deterioro.

$$MA = \begin{bmatrix} 1 & 1 \\ 2 & 3 \\ 3 & 3 \end{bmatrix}; \quad VA = \begin{bmatrix} 2 \\ 4 \\ 6 \end{bmatrix}; \quad XA = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$$

Tabla 2.1 Deterioro Modelo A (Fuente Propia)

	nact	No acum.	Acum.	$g_1(y)$	$g_2(y)$	$g_3(y)$	$g_4(y)$	Vector
MA	40	$1.0e-14^*$ [0.13] [0.13]	$1.0e-14^*$ [0.39] [0.21]	$1.0e-14^*$ - [0.35] [0.15]	$1.0e-15^*$ [-0.44] [0.99]	$1.0e-15^*$ [0.00] [0.58]	$1.0e-13^*$ [-0.33] [-0.14]	[0.46] [0.31]
	400	$1.0e-14^*$ [0.13] [0.13]	$1.0e-13^*$ [0.45] [0.24]	$1.0e-13^*$ [0.26] [0.24]	$1.0e-13^*$ [0.27] [0.23]	$1.0e-14^*$ [0.11] [0.11]	$1.0e-12^*$ [-0.14] [-0.11]	[0.06] [0.04]
	4000	$1.0e-14^*$ [0.13] [0.13]	$1.0e-12^*$ [-0.17] [-0.05]	$1.0e-12^*$ [-0.40] [-0.27]	$1.0e-12^*$ [-0.58] [-0.26]	$1.0e-12^*$ [0.07] [-0.39]	$1.0e-12^*$ [-0.26] [-0.31]	$1.0e-4^*$ [62] [-41]

$$MB = \begin{bmatrix} -4 & 1 \\ 3 & -3 \\ -2 & 2 \\ 0 & 2 \\ -4 & 3 \end{bmatrix}; \quad VB = \begin{bmatrix} -1 \\ -5 \\ -3 \\ 5 \\ -4 \end{bmatrix}; \quad XB = \begin{bmatrix} 1.2593 \\ 1.5748 \end{bmatrix}$$

Tabla 2.1 Deterioro Modelo B (Fuente Propia)

	nact	No acum.	Acum.	$g_1(y)$	$g_2(y)$	$g_3(y)$	$g_4(y)$	Vector
MB	40	$1.0e-15^*$ [0.44] [0.44]	$1.0e-15^*$ [0.88] [0.66]	$1.0e-13^*$ [0.66] [0.65]	$1.0e-13^*$ [0.35] [0.34]	$1.0e-14^*$ [0.15] [0.00]	$1.0e-13^*$ [-0.25] [0.24]	[0.24] [0.32]
	400	$1.0e-15^*$ [0.44] [0.44]	$1.0e-13^*$ [0.19] [0.21]	$1.0e-12^*$ [0.50] [0.50]	$1.0e-11^*$ [0.12] [0.12]	$1.0e-13^*$ [0.10] [0.10]	$1.0e-13^*$ [-0.50] [0.46]	[0.02] [0.03]
	4000	$1.0e-15^*$ [-0.44] [-0.44]	$1.0e-13^*$ [-0.90] [0.16]	$1.0e-9^*$ [-0.43] [0.43]	$1.0e-9^*$ [0.33] [-0.33]	$1.0e-9^*$ [0.14] [-0.14]	$1.0e-11^*$ [-0.78] [0.78]	[0.02] [0.03]

$$MC = \begin{bmatrix} 20 & -12 \\ 11 & 2 \\ 3 & 5 \\ 18 & -19 \\ 3 & 5 \\ -20 & -6 \\ -16 & -18 \\ 15 & 0 \\ -1 & -13 \\ 14 & -15 \end{bmatrix}; \quad VC = \begin{bmatrix} 3 \\ 2 \\ 2 \\ 1 \\ 7 \\ 3 \\ 6 \\ 7 \\ 5 \\ 6 \end{bmatrix}; \quad XC = \begin{bmatrix} 0.0457 \\ -0.2074 \end{bmatrix}$$

Tabla 2.3 Deterioro Modelo C (Fuente Propia)

	nact	No acum.	Acum.	$g_1(y)$	$g_2(y)$	$g_3(y)$	$g_4(y)$	Vector
MC	40	$1.0e-16^*$ [-0.13] [ 0.0 ]	$1.0e-15^*$ [0.09] [0.49]	$1.0e-15^*$ [0.99] [0.94]	$1.0e-15^*$ [0.45] [0.66]	$1.0e-16^*$ [0.06] [0.17]	$1.0e-15^*$ [-0.42] [-0.22]	$1.0e-3^*$ [15] [20]
	400	$1.0e-16^*$ [-0.13] [ 0.00 ]	$1.0e-14^*$ [-0.10] [ 0.43 ]	$1.0e-13^*$ [0.62] [0.62]	$1.0e-13^*$ [0.18] [0.18]	$1.0e-15^*$ [0.006] [0.194]	$1.0e-14^*$ [-0.55] [-0.53]	$1.0e-3^*$ [-70] [ 10 ]
	4000	$1.0e-16^*$ [-0.13] [ 0.00 ]	$1.0e-13^*$ [-0.10] [-0.16]	$1.0e-11^*$ [0.64] [0.64]	$1.0e-11^*$ [0.19] [0.19]	$1.0e-10^*$ [0.11] [0.11]	$1.0e-11^*$ [-0.28] [-0.27]	$1.0e-4^*$ [-13] [ 18 ]

Como se refleja en la tabla 2.1 el modelo se puede considerar estable siempre que las calificaciones usadas para actualizar el modelo hayan sido calculadas con el modelo anterior. En este marco, según la tabla 2.1 y la figura 2.9 el menor deterioro se obtiene actualizando el modelo de forma no acumulativa, según la ecuación (2.5)

# Conclusión

Estudiando los resultados obtenidos podemos afirmar que el modelo es estable cuando se actualiza con cada nuevo request, tanto si la actualización es acumulada o no acumulada. Esto se representa en las ecuaciones (2.14).

$$\begin{aligned}
 X_0 &= (M)^+ * V \\
 X_{ina} &= \begin{bmatrix} M \\ f_{m+i} \end{bmatrix} * \begin{bmatrix} V \\ f_{m+i} * X_{i-1} \end{bmatrix} \quad X_{ia} = \begin{bmatrix} M \\ f_{m+1} \\ f_{m+2} \\ \vdots \\ f_{m+i} \end{bmatrix} * \begin{bmatrix} V \\ f_{m+1} * X_0 \\ f_{m+2} * X_1 \\ \vdots \\ f_{m+2} * X_{i-1} \end{bmatrix} \\
 X_{im} &= \begin{bmatrix} M \\ (f_{m+1}) * g(y_1) \\ \vdots \\ (f_{m+i}) * g(y_i) \end{bmatrix} * \begin{bmatrix} V \\ (f_{m+1}) * g(y_1) * X_0 \\ \vdots \\ (f_{m+i}) * g(y_i) * X_{i-1} \end{bmatrix} \\
 |X_i| - |X_0| &= \begin{bmatrix} |\alpha_i| & - & |\alpha_0| \\ |\beta_i| & - & |\beta_0| \end{bmatrix} = \begin{bmatrix} d_\alpha \\ d_\beta \end{bmatrix}; \quad d_\alpha, d_\beta < 10^{-9}
 \end{aligned}$$

$$X_0 \approx X_a \approx X_{ina} \approx X_{im} \approx X_i \quad (2.14)$$

El único caso en el que se da una variación del modelo apreciable, es cuando se introducen request del Dataset inicial y sus correspondientes calificaciones, que aun siendo el valor verdadero varía el modelo.

$$X_{ip} = \begin{bmatrix} M \\ f_{m+1} \\ f_{m+2} \\ f_{m+i} \\ f_{m_{set}} \end{bmatrix} * \begin{bmatrix} V \\ f_{m+1} * X_0 \\ f_{m+2} * X_1 \\ f_{m+2} * X_{i-1} \\ V_{set} \end{bmatrix}$$

$$X_0 \neq X_{ip} \quad (2.15)$$

# Teorema

En el apartado anterior se demuestra empíricamente que la diferencia entre el modelo final y el inicial es prácticamente cero, salvo cuando se actúa de forma externa sobre el algoritmo que calcula los modelos introduciendo valores que no han sido calculados por el modelo anterior. La razón de que el deterioro no sea exactamente cero se debe al error de computación que se produce al redondear el resultado en el proceso iterativo de calcular los modelos.

Guiándonos por los resultados experimentales obtenidos, es posible desarrollar un teorema matemático que demuestra de forma exacta que el deterioro es cero, para ello nos apoyamos en las propiedades de las soluciones obtenidas mediante la Pseudoinversa de Penrose, las cuales satisfacen las condiciones de mínima norma y mínimos cuadrados, es decir que la norma al cuadrado del vector cuyas componentes son la diferencia entre el valor aproximado y el considerado real es la mínima posible, esto se describe en la ecuación (4.1).

$$\vec{V} = \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix}; \|\vec{V}\|^2 = V_1^2 + V_2^2 + V_3^2 \quad (4.1)$$

$$\begin{aligned} V_1 &= V_{aprox\_1} - V_{real} \\ V_2 &= V_{aprox\_2} - V_{real} \\ V_3 &= V_{aprox\_3} - V_{real} \end{aligned}$$

Otro concepto necesario para hacer el desarrollo matemático es el de un argumento que minimiza el valor de una expresión, se explica con un ejemplo en la ecuación (4.2). El valor de  $y$  que hace mínima esa expresión es cero.

$$0 = \arg \min \{(y^2 + 1): y \in \mathbb{R}\} \quad (4.2)$$

$$\begin{array}{ll} y = 0 & 1 \\ y = 1 & 2 \\ y = -3 & 10 \end{array}$$

Obtención del modelo inicial  $\theta$ , a partir de la matriz  $M$  de request y el vector  $\vec{V}$  de calificaciones, que forman el training Dataset.

$$M \in \mathbb{R}^{m \times n} \quad m > n \quad \text{rango } M = n \text{ (maximo)}$$

$$\theta = (M^t * M)^{-1} * M^t * \vec{V} = M^+ * \vec{V} = \arg \min \left\{ \|M * \vec{x} - \vec{V}\|^2 : \vec{x} \in \mathbb{R}^n \right\} \quad (4.3)$$

Introducimos un nuevo vector request  $\underline{f}$  cuya calificación  $\underline{V}$  es calculada con el modelo anterior para calcular la actualización del modelo  $\theta'$ . Cabe destacar que  $\underline{V} = \underline{f} * \theta$  se considera como valor real de la calificación.

$$\begin{bmatrix} M \\ \underline{f} \end{bmatrix} * \theta' = \begin{bmatrix} \vec{V} \\ \underline{f} * (M^t * \underline{V}) \end{bmatrix} \quad (4.4)$$

La actualización automática del modelo  $\theta'$ , bien sea hecha de forma acumulativa o no acumulativa tiene que satisfacer la ecuación (4.5) es decir las condiciones de mínima norma y mínimo error cuadrático.

$$\theta' = \arg \min \left\{ \left\| \begin{bmatrix} M \\ \underline{f} \end{bmatrix} * \vec{x} - \begin{bmatrix} \vec{V} \\ \underline{f} * (M^t * \underline{V}) \end{bmatrix} \right\|^2 : \vec{x} \in \mathbb{R}^n \right\} \quad (4.5)$$

Comprobamos que el argumento  $\vec{x}$  que hace minimiza esa expresión es el correspondiente a la ecuación (4.6). Esto se comprueba en la ecuación (4.7) tras hacer la sustitución.

$$\vec{x} = M^+ * \vec{V} \quad (4.6)$$

$$\left\| \begin{bmatrix} M * (M^+ * \vec{V}) \\ \underline{f} * (M^+ * \vec{V}) \end{bmatrix} - \begin{bmatrix} \vec{V} \\ \underline{f} * (M^t * \underline{V}) \end{bmatrix} \right\|^2 = \left\| \begin{bmatrix} M * \vec{x} - \vec{V} \\ 0 \end{bmatrix} \right\|^2 \quad (4.7)$$

$$\theta' = \theta \quad (4.8)$$

La ecuación (4.7) demuestra que los modelos no cambian al actualizarse, salvo que se introduzcan calificaciones no calculadas por el modelo anterior.

# Líneas de investigación futuras

Estudiar y cuantificar la variación de los modelos debido a la introducción de calificaciones realizadas por el Agente experto. Sería de interés saber si esas variaciones hacen que las calificaciones calculadas por el modelo perturbado disten considerablemente de las reales haciendo no viable el uso del modelo. La obtención de la función de perturbación  $g(y)$  que da como resultado dichas variaciones también podría ser interesante, para así intentar anular dichas perturbaciones para poder usar el modelo automático con calificaciones introducidas por el agente experto cuando el vector request coincida con uno de los usados en le training Dataset. La obtención de la funciona antes mencionada también puede servir para atacar un servidor con seguridad perimetral y así comprobar la robustez de los sistemas de seguridad.

# AGRADECIMIENTOS

A mi familia, porque sin su apoyo y esfuerzo no podría haber llegado hasta aquí.

A Don Miguel Carriegos Vieira, por haberme orientado de forma clara y precisa en la realización de este trabajo y por haber dedicado tiempo a resolver y explicarme las dudas surgidas.

Al profesorado y personal de la Escuela de Ingenierías, Industrial e Informática de la Universidad de Leon, que con esfuerzo y paciencia consiguen satisfacer parte de la curiosidad de los alumnos y dotarles de las competencias profesionales para encarar el desarrollo tecnológico futuro.



# LISTA DE REFERENCIAS BIBLIOGRÁFICAS

- 1 Tanenbaum Wetheral (2012). Redes de Computadoras
- 2 Ross Anderson (2001). Security Engineering
- 3 Thomas N.E. Greville (2003). Generalized Inverses
- 4 M. Hall Jr. (1973). Teoría de los Grupos
- 5 J. C. A. Barata and M. S. Hussein (2011). The Moore-Penrose Pseudoinverse
- 6 Ronald Cramer, Eike Kiltz and Carles Padró (2007). A Note on Secure Computation of the Moore-Penrose Pseudoinverse and Its Application to Secure Linear Algebra
- 7 Documentación de ayuda contenida en el software Matlab
- 8 Alberto Herreros, Enrique Baeyens (2010). Curso de programación en Matlab
- 9 Pedro Corcuera (2010). Creación de interfaces de usuario con MATLAB

# ANEXO-A

Informe 1- Representación gráfica de la solución obtenida usando la Pseudoinversa de Penrose

Informe 2- Análisis deterioro Modelos

Informe 3- Generación de la función  $g_4(y)$

# INFORME\_1

Grafica de Solucion real y Solucion con Pseudoinversa de Penrose..... 1

## Grafica de Solucion real y Solucion con Pseudoinversa de Penrose

```
clear,clc

A=randi([-5,5],10,2)
R=randi([-3,3],10,1)
B=pinv(A)*R

R_Penrose=A*B

x=A(:,1), y=A(:,2), z=R;

f=@(x,y) x.*B(1,:) + y.*B(2,:);
ezmesh(f,[-10,10],10)
colormap([0 0 1])
alpha(.4)
hold on

scatter3(x,y,z,'r','fill')
grid on
```

A =

-5	2
2	0
-5	5
-5	2
0	3
-4	-1
3	-1
3	4
2	-5
-4	-4

R =

-2
-1
2
2
-3
-1
0
-1
1
1

B =

-0.125342948969193  
-0.141177392803951

R\_Penrose =

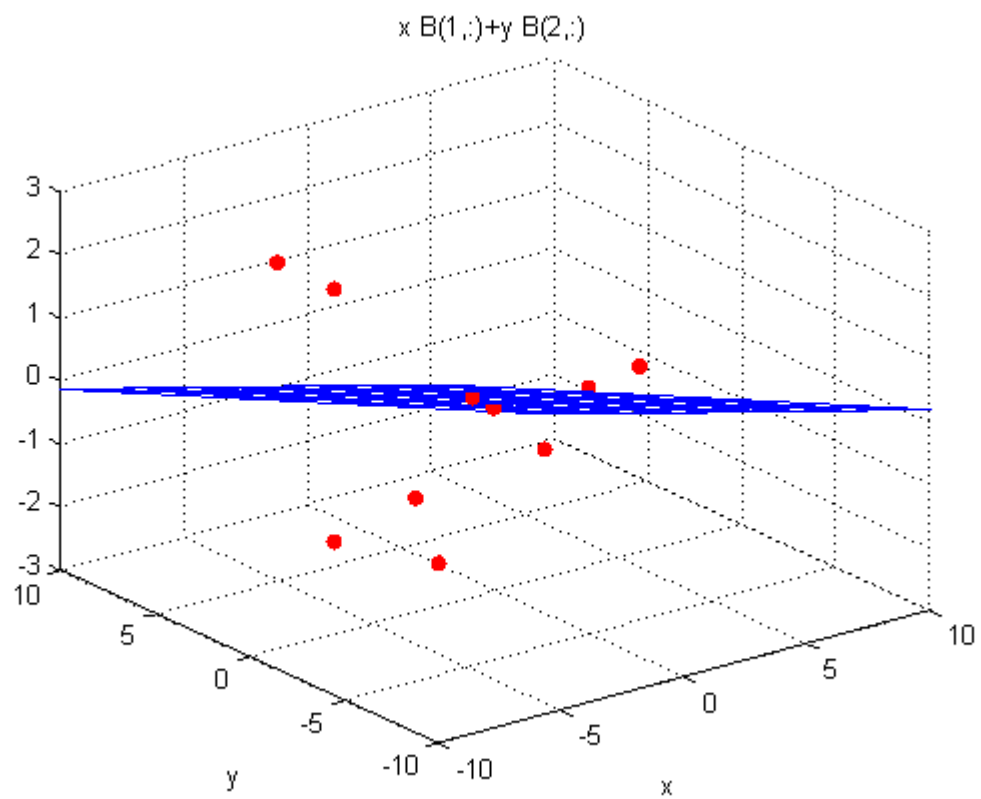
0.344359959238065  
-0.250685897938387  
-0.079172219173787  
0.344359959238065  
-0.423532178411852  
0.642549188680724  
-0.234851454103629  
-0.940738418123383  
0.455201066081367  
1.066081367092576

x =

-5  
2  
-5  
-5  
0  
-4  
3  
3  
2  
-4

y =

2  
0  
5  
2  
3  
-1  
-1  
4  
-5  
-4



*Published with MATLAB® R2013b*

# INFORME\_2

Data-set, calculo del modelo inicial $X_0$ y su grafica 3D .....	1
Sin actualizar Modelo .....	3
Modelo no acumulativo .....	4
Modelo acumulativo .....	4
Representacion grafica .....	5
Perturbacion.....	7
Modificacion del vector request .....	9

## Data-set, calculo del modelo inicial $X_0$ y su grafica 3D

```
clear all
clc
format long

Mrand=randi([-5,5],10,2);
vrand=randi([-3,3],10,1);
%M=input('Introduce la matriz de actualizaciones: ')
%v=input('Introduce el vector de escala de clasificacion: ')

Ma=[1 1; 2 3; 3 3], va=[2; 4; 6];
Mb=[-4 1; 3 -3; -2 2; 0 2; -4 3], vb=[-1; -5; -3; 5; -4];
Mc=[20 -12; 11 2; 3 5; 18 -19; 3 5; -20 -6; -16 -18; 15 0; -1 -13; 14 -15], vc=[3; 2; 2; 1; 7; 3; 6; 7; 5; 6];

M=Mb
v=vb

x_0=pinv(M)*v
v_0=M*x_0;

na=400;

mx=M(:,1); my=M(:,2); mz=v;
f=@(ex,ey) ex.*x_0(1,:) + ey.*x_0(2,:);

%na=input('Ingrese el numero de actualizaciones: ')
y=[1:na+1];

Mact=[];
v_request=[2 2]
%v_request=input('Ingrese el vector request: ')

v_rep=v_request;
%v_rep=input('Ingrese el vector representacion: ')
```

```

for i=1:na
    Mact(i,:)=v_request;
end

figure, ezmesh(f,[-10,10],10)
title('Grafica del modelo inicial')
colormap([0 0 1])
alpha(.4)
hold on
scatter3(mx,my,mz,'r','fill')
grid on

```

Ma =

1	1
2	3
3	3

Mb =

-4	1
3	-3
-2	2
0	2
-4	3

Mc =

20	-12
11	2
3	5
18	-19
3	5
-20	-6
-16	-18
15	0
-1	-13
14	-15

M =

-4	1
3	-3
-2	2
0	2
-4	3

v =

```

-1
-5
-3
5
-4

x_0 =

    1.259358288770054
    1.574866310160428

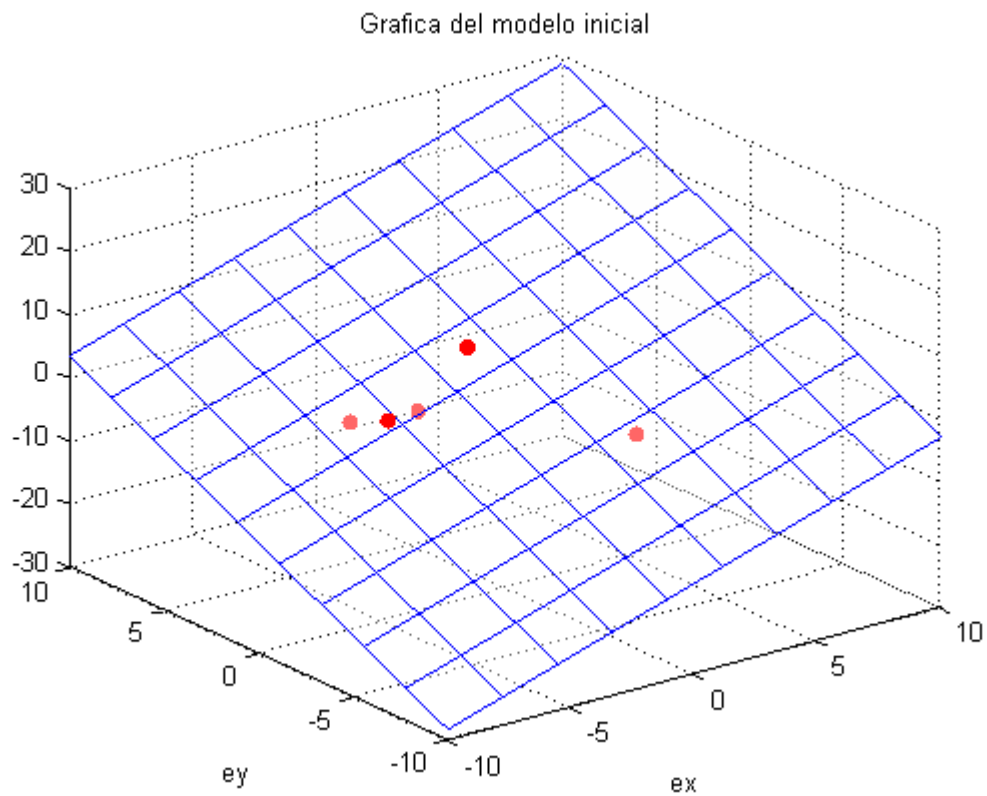
```

```

v_request =

    2    2

```



### Sin actualizar Modelo

```

modelos_sin_actualizar(:,1)=x_0;

for i=1:na
    mm=[M;Mact(i,:)];
    vv=[v;(Mact(i,:)*x_0)];
    modelos_sin_actualizar(:,i+1)=pinv(mm)*vv;
end

L1=length(modelos_sin_actualizar);
deterioro_modelos_sin_actualizaro=abs(modelos_sin_actualizar(:,1))-

```



```
abs(modelos_sin_actualizar(:,L1))

det_sin_act=[modelos_sin_actualizar(:,1)  modelos_sin_actualizar(:,L1)]
```

```
deterioro_modelos_sin_actualizaro =
```

```
1.0e-15 *

-0.222044604925031
0
```

```
det_sin_act =
```

```
1.259358288770054  1.259358288770054
1.574866310160428  1.574866310160428
```

## Modelo no acumulativo

```
modelos_NO_acumulativo(:,1)=x_0;

for i=1:na
mm=[M;Mact(i,:)];
vv=[v;(Mact(i,:)*modelos_NO_acumulativo(:,i))];
modelos_NO_acumulativo(:,i+1)=pinv(mm)*vv;
end

L1=length(modelos_NO_acumulativo);
deterioro_NO_acumulativo=abs(modelos_NO_acumulativo(:,1))-
abs(modelos_NO_acumulativo(:,L1))

det_na=[modelos_NO_acumulativo(:,1)  modelos_NO_acumulativo(:,L1)]
```

```
deterioro_NO_acumulativo =
```

```
1.0e-15 *

-0.444089209850063
-0.444089209850063
```

```
det_na =
```

```
1.259358288770054  1.259358288770054
1.574866310160428  1.574866310160428
```

## Modelo acumulativo

```
mm=[M];
vv=[v];
modelos_acumulativo(:,1)=x_0;
```

```

for i=1:na
mm=[mm;Mact(i,:)];
vv=[vv;(Mact(i,:)*modelos_acumulativo(:,i))];
modelos_acumulativo(:,i+1)=pinv(mm)*vv;
end

L2=length(modelos_acumulativo);
deterioro_acumulativo=abs(modelos_acumulativo(:,1))-abs(modelos_acumulativo(:,L2))
det_a=[modelos_acumulativo(:,1) modelos_acumulativo(:,L1)]

```

deterioro\_acumulativo =

```

1.0e-13 *

0.193178806284777
-0.210942374678780

```

det\_a =

```

1.259358288770054    1.259358288770034
1.574866310160428    1.574866310160449

```

## Representacion grafica

```

za=v_request*modelos_NO_acumulativo;
zna=v_request*modelos_acumulativo;
zsinact=v_request*modelos_sin_actualizar;

figure, subplot(1,2,1), plot(y,za,'r'), subplot(1,2,2), plot(y,zna,'b')
title(subplot(1,2,1), {'Variacion del modelo no'; 'acumulativo'})
xlabel(subplot(1,2,1), 'i actualizaciones')
yy=ylabel(subplot(1,2,1), 'Vrp*X_i', 'rot', 0)
set(yy, 'Units', 'Normalized', 'Position', [0.9, 1, 0]);
grid on

title(subplot(1,2,2), {'Variacion del modelo'; 'acumulativo'})
xlabel(subplot(1,2,2), 'i actualizaciones')
yy=ylabel(subplot(1,2,2), 'Vrp*X_i', 'rot', 0)
set(yy, 'Units', 'Normalized', 'Position', [1, 1, 0]);
grid on

figure, plot(y,zsinact)
title('Variacion de modelos sin actualizar')
xlabel('i actualizaciones')
yy=ylabel('Vrp*X_0', 'rot', 0)
set(yy, 'Units', 'Normalized', 'Position', [1, 1, 0]);
grid on

```

yy =

```

5.380019531250000e+02

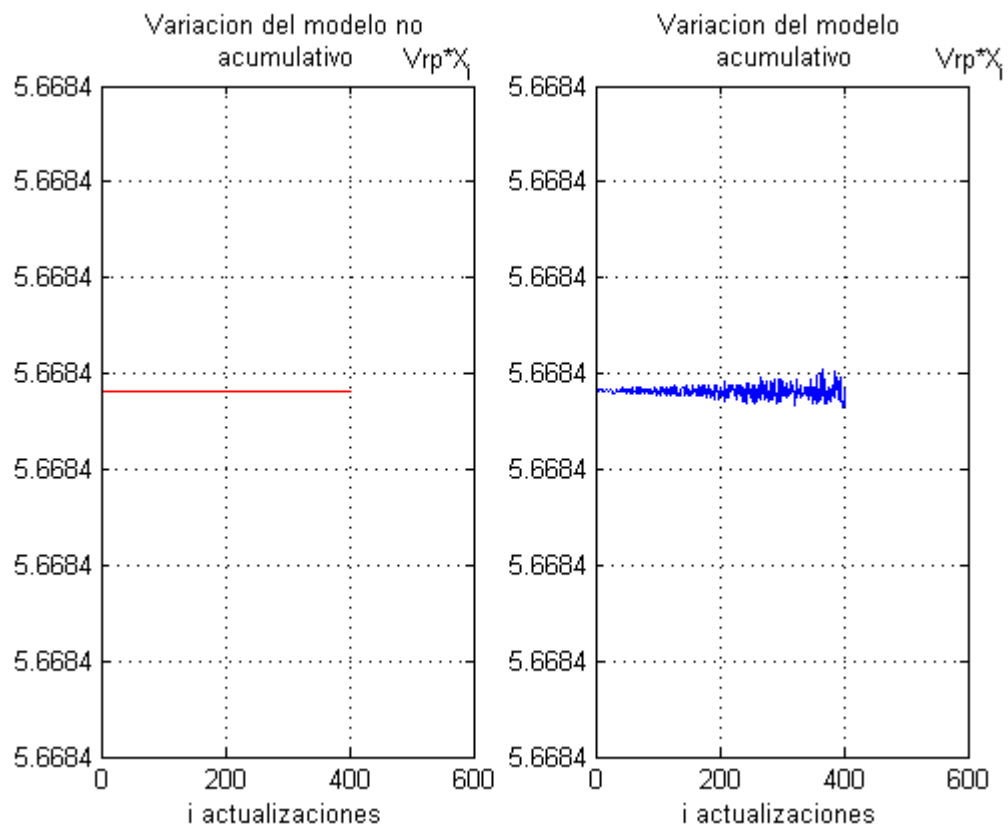
```

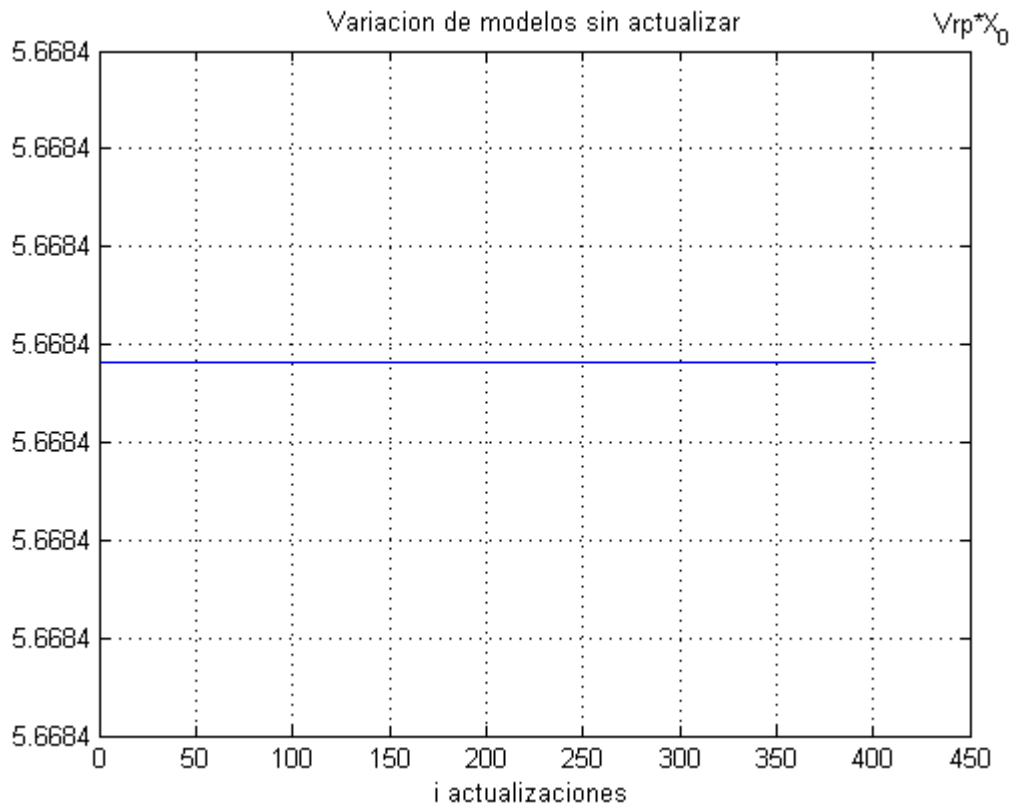
yy =

5.410019531250000e+02

yy =

7.190019531250000e+02





### Perturbacion

```
mm=[M];
vv=[v];
modelos_acumulativo_per(:,1)=x_0;

for i=1:na

    if i==na/4;
        Mact(i,:)=Mact(1,:);
        mm=[mm;(Mact(i,:))];
        vv=[vv; v(1)];
        modelos_acumulativo_per(:,i+1)=pinv(mm)*vv;

    elseif i==na/2;
        Mact(i,:)=Mact(1,:);
        mm=[mm;(Mact(i,:))];
        vv=[vv; v(1)];
        modelos_acumulativo_per(:,i+1)=pinv(mm)*vv;

    elseif i==na*(3/4);
        Mact(i,:)=Mact(1,:);
        mm=[mm;(Mact(i,:))];
        vv=[vv; v(1)];
        modelos_acumulativo_per(:,i+1)=pinv(mm)*vv;

    elseif i==na;
```

```

Mact(i,:)=Mact(1,:);
mm=[mm;(Mact(i,:))];
vv=[vv; v(1)];
modelos_acumulativo_per(:,i+1)=pinv(mm)*vv;

end

mm=[mm;(Mact(i,:))];
vv=[vv;(Mact(i,:)*modelos_acumulativo_per(:,i))];
modelos_acumulativo_per(:,i+1)=pinv(mm)*vv;
end

L2=length(modelos_acumulativo_per);
deterioro_perturbacion=abs(modelos_acumulativo_per(:,1))-
abs(modelos_acumulativo_per(:,L2))

z=v_request*modelos_acumulativo_per;
figure, plot(y,z,'r')
title({'Variacion del modelo con introduccion';'de un valor del data-set'})
xlabel(' i actualizaciones')
yy=ylabel('Vrp*X_i', 'rot', 0)
set(yy, 'Units', 'Normalized', 'Position', [1, 1, 0]);
grid on

```

deterioro\_perturbacion =

```

0.029279968304459
0.038691386688071

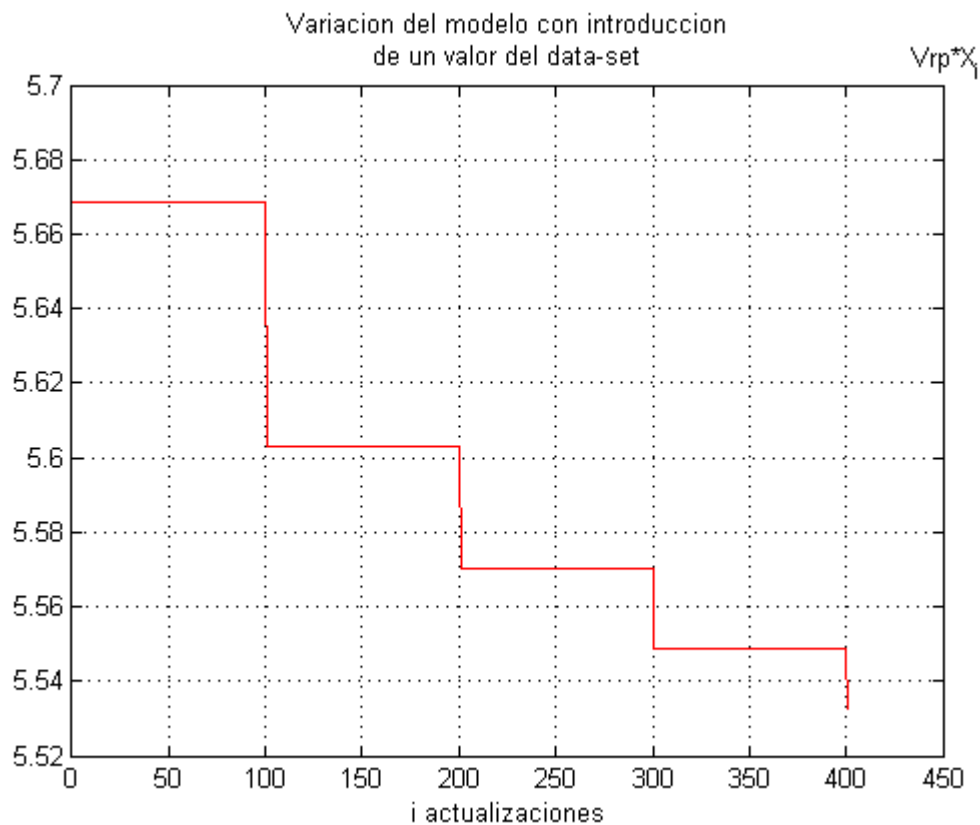
```

yy =

```

9.000019531250000e+02

```



### Modificacion del vector request

```
mm=[M];
vv=[v];

x1=fun(na);
x2=sin (y);
x3=5*y;
x4=log(y);

funciones=[x4; x3; x2; x1];

for j=1:4

    deterioro_modelos_acumulativo_mod=[];
    modelos_acumulativo_mod=[];
    modelos_acumulativo_mod(:,1)=x_0;
    x=[];
    x=funciones(j,:);
    x;

    for i=1:na
        mm=[mm; (Mact(i,:)*x(i))];
        vv=[vv; (Mact(i,:)*(x(i))*modelos_acumulativo_mod(:,i))];
        modelos_acumulativo_mod(:,i+1)=pinv(mm)*vv;
```

```

end

L2=length(modelos_acumulativo_mod);
deterioro_modelos_acumulativo_mod=abs(modelos_acumulativo_mod(:,1))-
abs(modelos_acumulativo_mod(:,L2))
modelos_acumulativo_mod(:,L2)

z_mam=v_request*modelos_acumulativo_mod;

figure, subplot(1,2,1), plot(y,z_mam,'r'), subplot(1,2,2), plot(y,x,'b')
title(subplot(1,2,1), 'Variacion del modelo')
xlabel(subplot(1,2,1),'i actualizaciones')
yy=ylabel(subplot(1,2,1),'Vrp*X_i', 'rot', 0)
set(yy, 'Units', 'Normalized', 'Position', [1, 1, 0]);
grid on

title(subplot(1,2,2), ' Funcion modificadora')
xlabel(subplot(1,2,2),'i actualizaciones')
yy=ylabel(subplot(1,2,2),'Vrp*X_i', 'rot', 0)
set(yy, 'Units', 'Normalized', 'Position', [1, 1, 0]);
grid on

end

```

```
deterioro_modelos_acumulativo_mod =
```

```

1.0e-13 *
-0.102140518265514
0.102140518265514

```

```
ans =
```

```

1.259358288770064
1.574866310160417

```

```
yy =
```

```
1.081001342773438e+03
```

```
yy =
```

```
1.084001342773438e+03
```

```
deterioro_modelos_acumulativo_mod =
```

```

1.0e-11 *
-0.126676447109730
0.126609833728253

```

ans =

1.259358288771320  
1.574866310159162

yy =

1.264001342773438e+03

yy =

1.267001342773438e+03

deterioro\_modelos\_acumulativo\_mod =

1.0e-12 \*

-0.555555601522428  
0.555777646127353

ans =

1.259358288770609  
1.574866310159872

yy =

1.447001342773438e+03

yy =

1.450001342773438e+03

deterioro\_modelos\_acumulativo\_mod =

1.0e-10 \*

-0.598920912864287  
0.598867622159105

ans =

1.259358288829946  
1.574866310100541

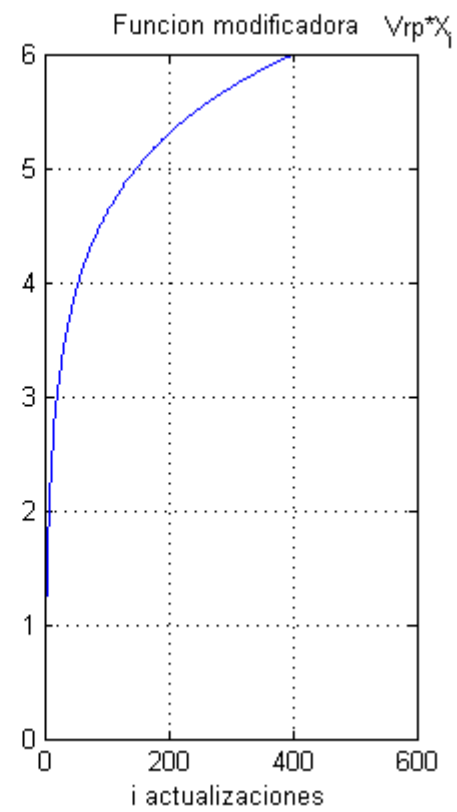
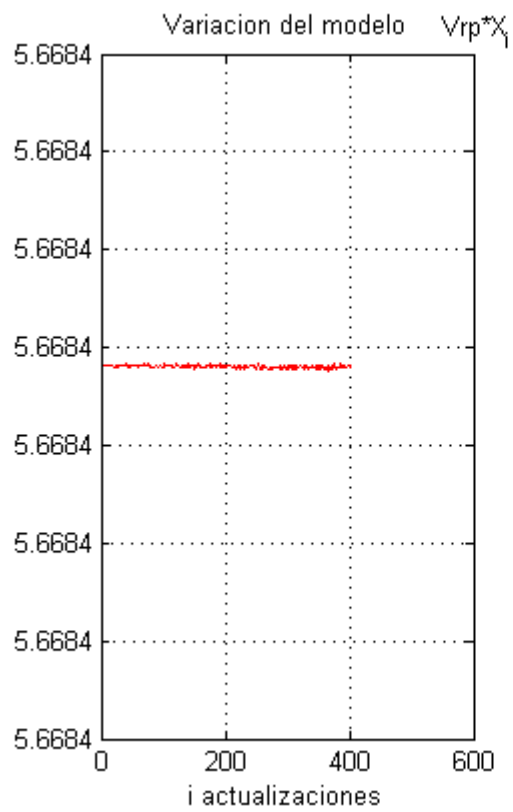
yy =

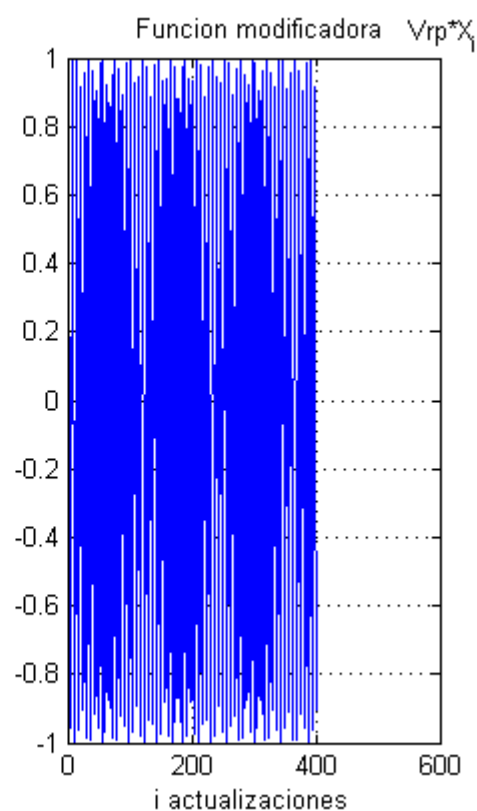
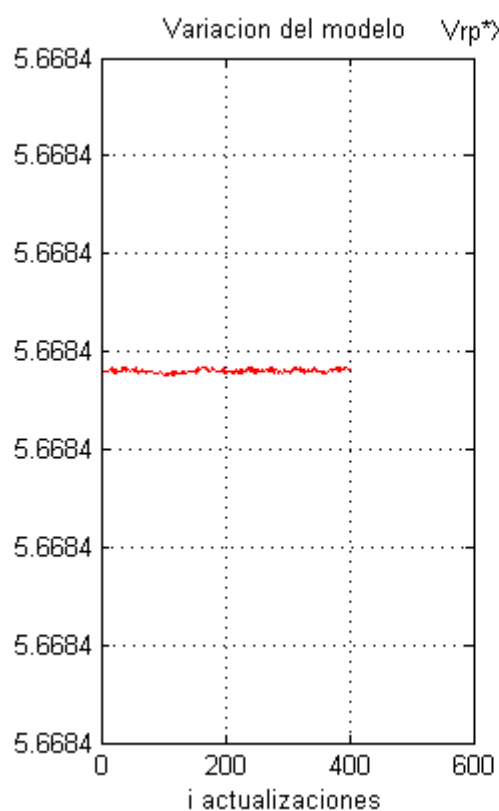
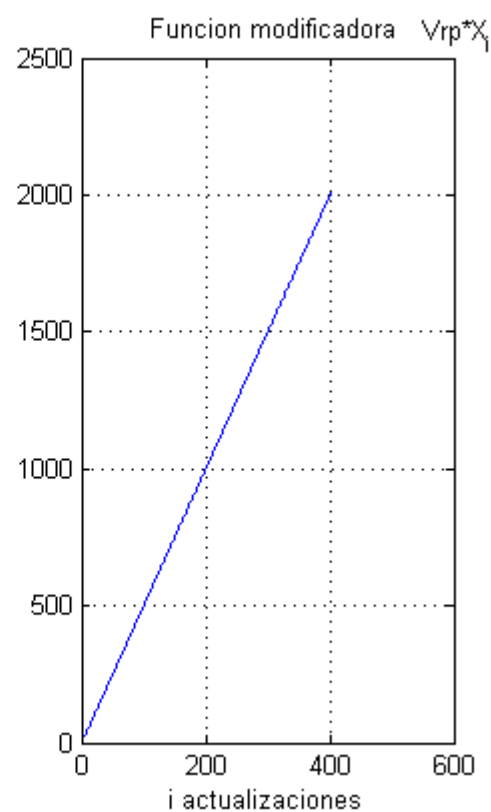
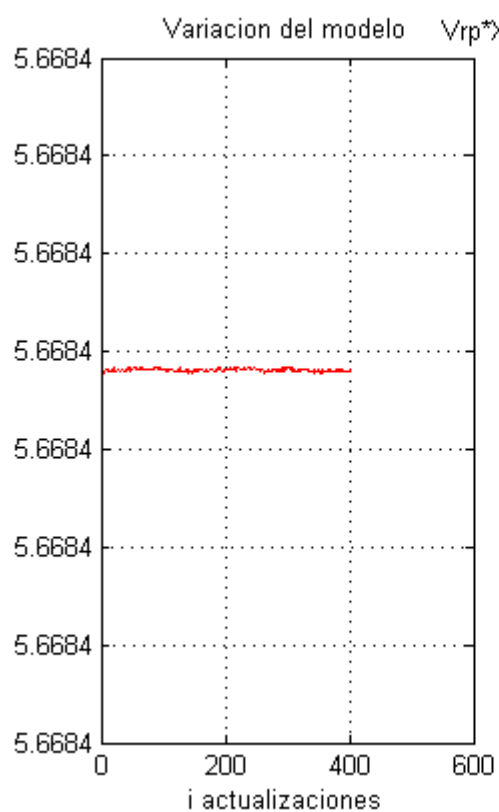


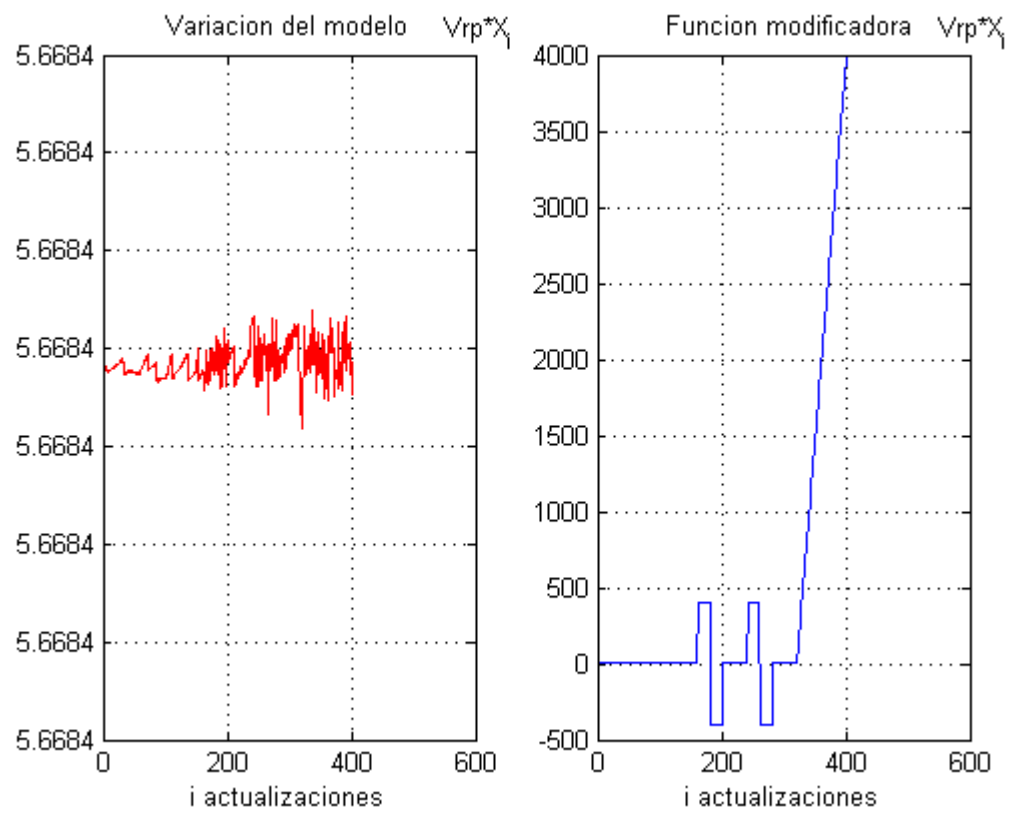
1.630000122070313e+03

yy =

1.633000122070313e+03







*Published with MATLAB® R2013b*

## INFORME 3

```
function y=fun(na)

ancho=0.05*na;

recta1=(ones(1,na*0.4+1));
pico_positivo=na.*(ones(1,ancho));
pico_negativo=-na.*(ones(1,ancho));
recta2=(ones(1,na*0.1));
recta3=recta2;

long_recta4=(1:na*0.2);
recta4=50*long_recta4;

y=[recta1 pico_positivo pico_negativo recta2 pico_positivo pico_negativo recta3 recta4
];

%length(x), length(y)
%plot(x,y);
%grid on
%axis([0 na*1.2 -na*1.2 na*11]);
```