5 TANDA 3: CAMBIO DE REFERENCIAS

En este capítulo vamos a realizar pruebas con distintas referencias.

Se trata de cambiar las amplitudes máximas y mínimas de los escalones para observar cómo influye esto en el comportamiento del sistema.

CON REFERENCIA DOBLE

Se trata de comprobar qué efectos tiene sobre los resultados que se obtienen con el esquema de control propuesto, utilizar una referencia con escalones cuya amplitud es exactamente el doble que la que hemos venido utilizando (figura 40)

Lo que haremos será aplicar el método descrito en apartado 4.2.2 pero utilizando esta nueva referencia.

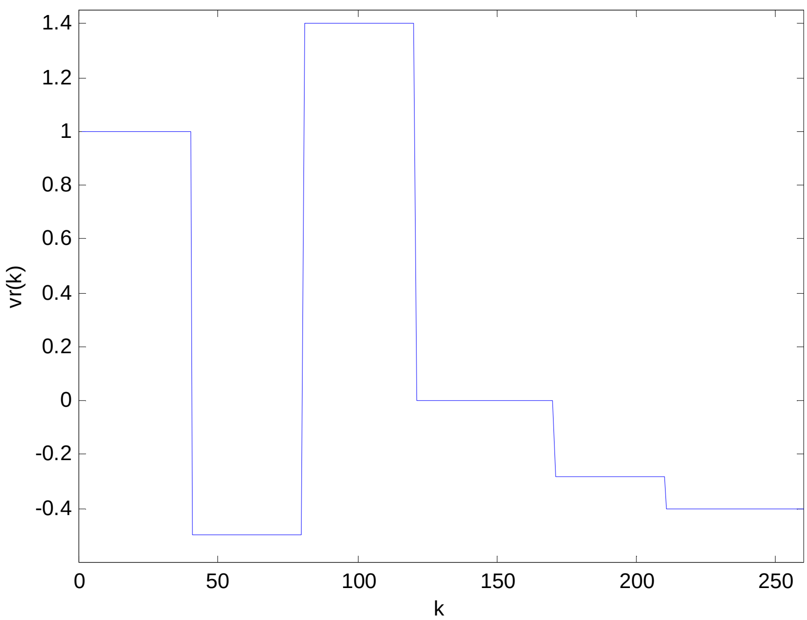


figura 40

Partimos de npas=100 alfa =0.001 e intentamos optimizar el valor de kp aplicando el método al igual que hicimos antes.

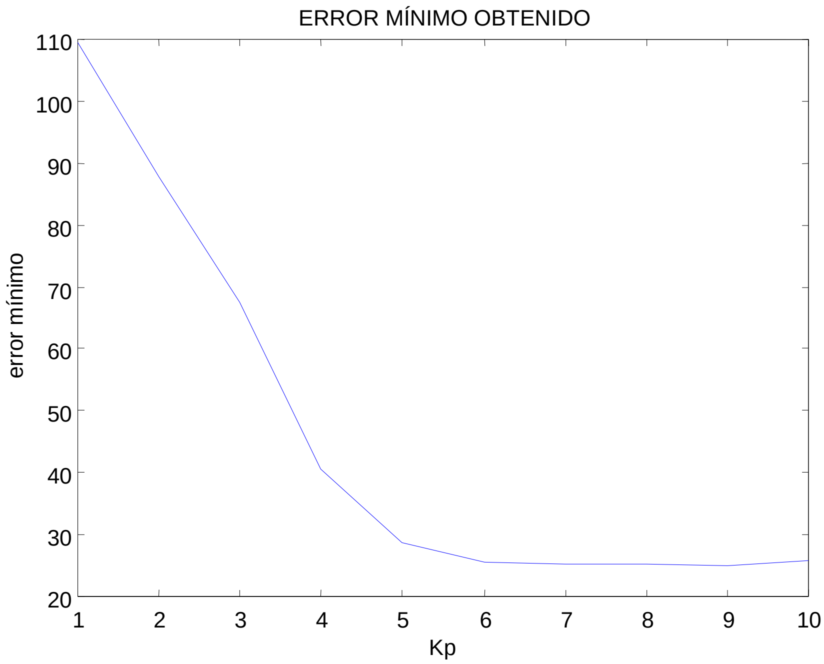


figura 41

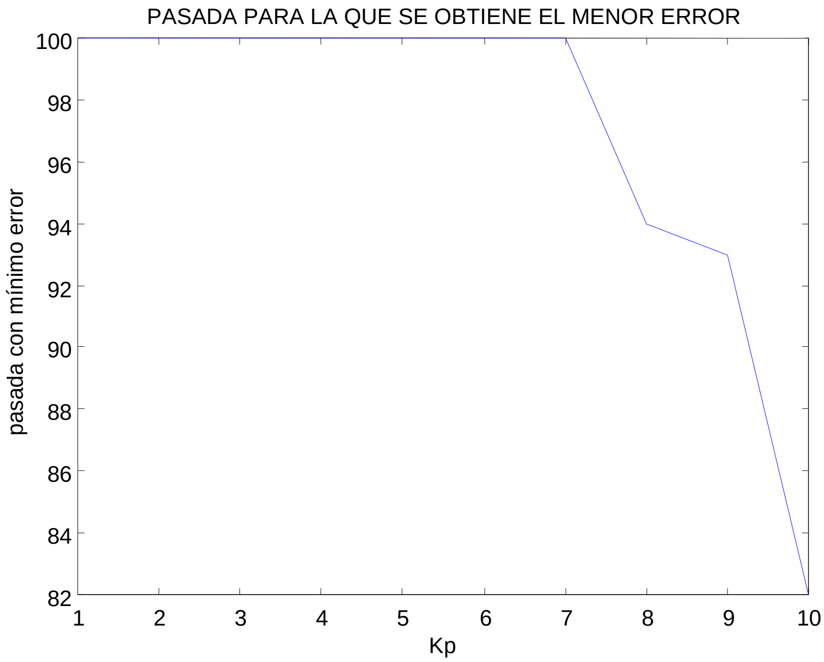


figura 42

Variamos alfa en escalones de 0.0005 .

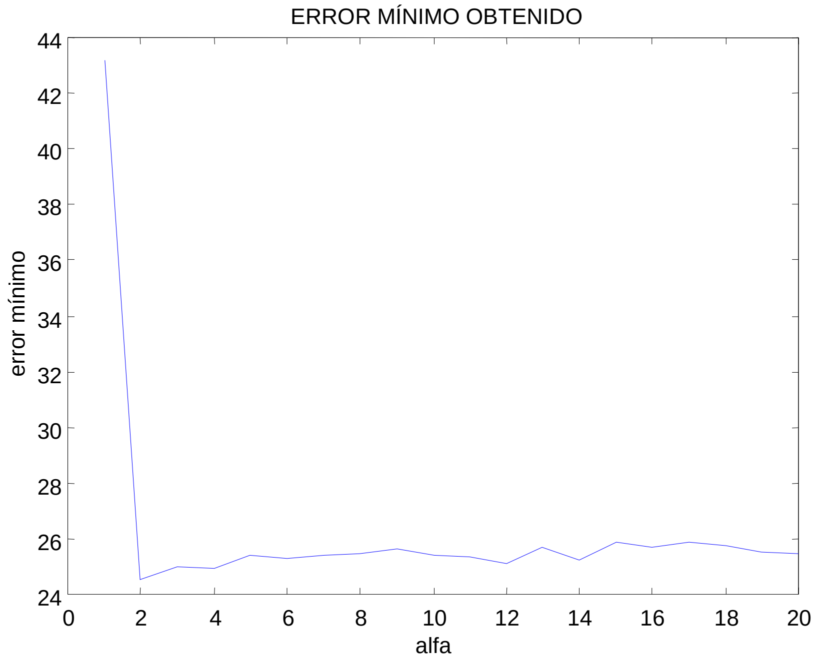


figura 43

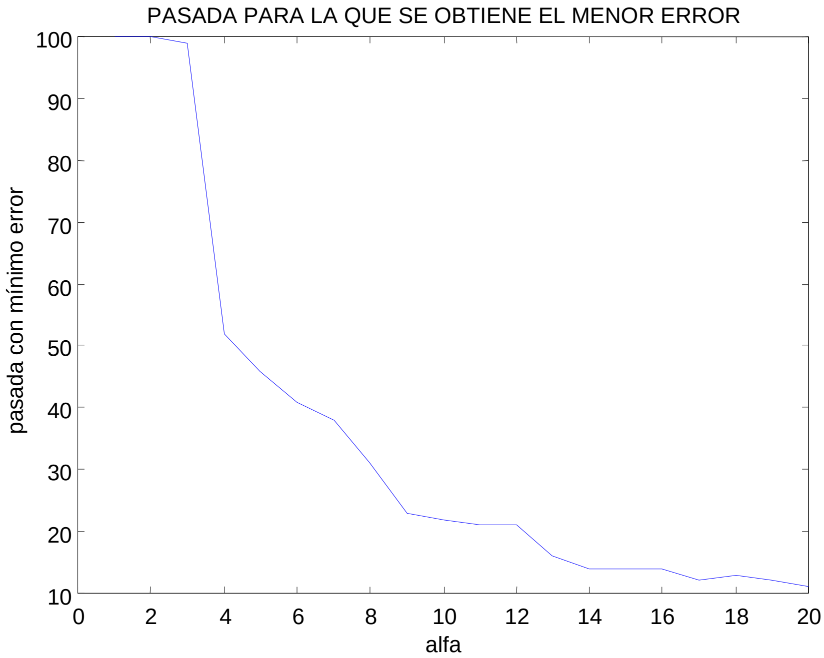


figura 44

CON REFERENCIA MITAD

Hacemos exactamente lo mismo pero utilizando ahora un referencia con escalones cuya amplitud es la mitad que la de los escalones de la referencia que veníamos utilizando desde el principio de este proyecto.

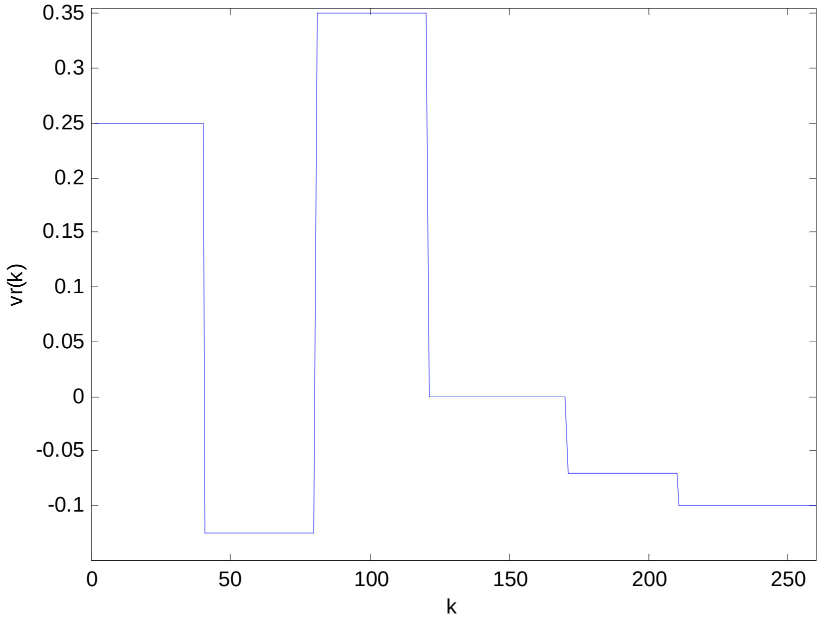


figura 45

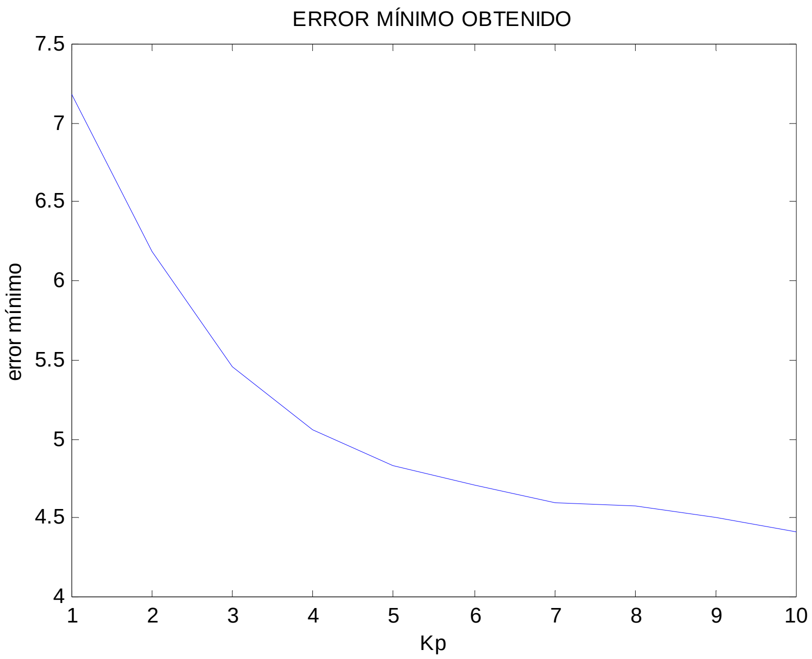


figura 46

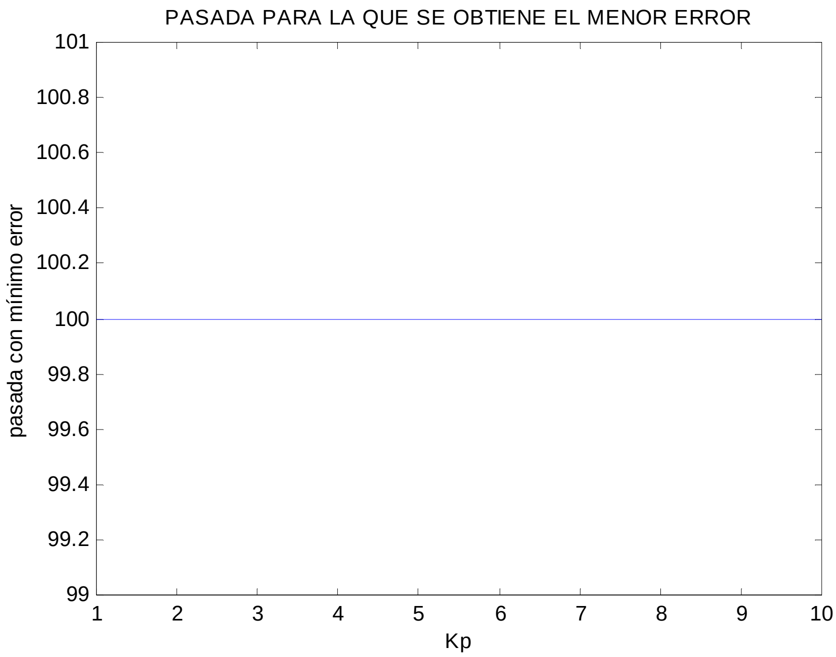

PASADA PARA LA QUE SE OBTIENE EL MENOR ERROR
pasada con mínimo error
Kp
101
100.8
100.6
100.4
100.2
100
99.8
99.6
99.4
99.2
99
1
2
3
4
5
6
7
8
9
10

figura 47

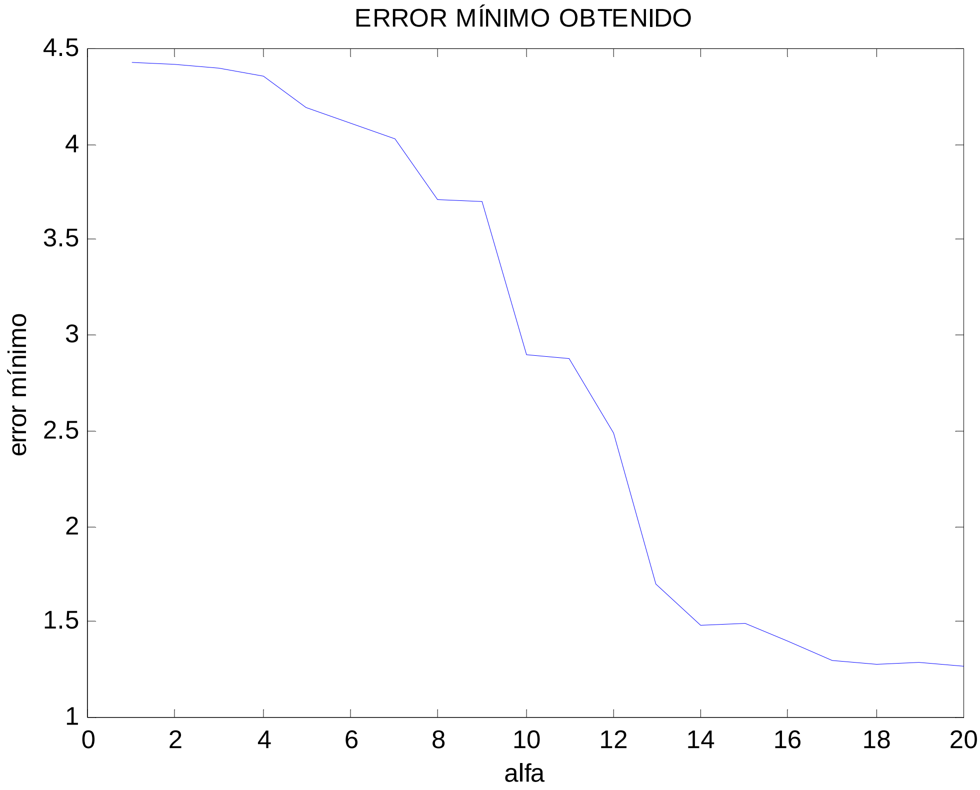

ERROR MÍNIMO OBTENIDO
error mínimo
alfa
4.5
4
3.5
3
2.5
2
1.5
1
0
2
4
6
8
10
12
14
16
18
20

figura 48

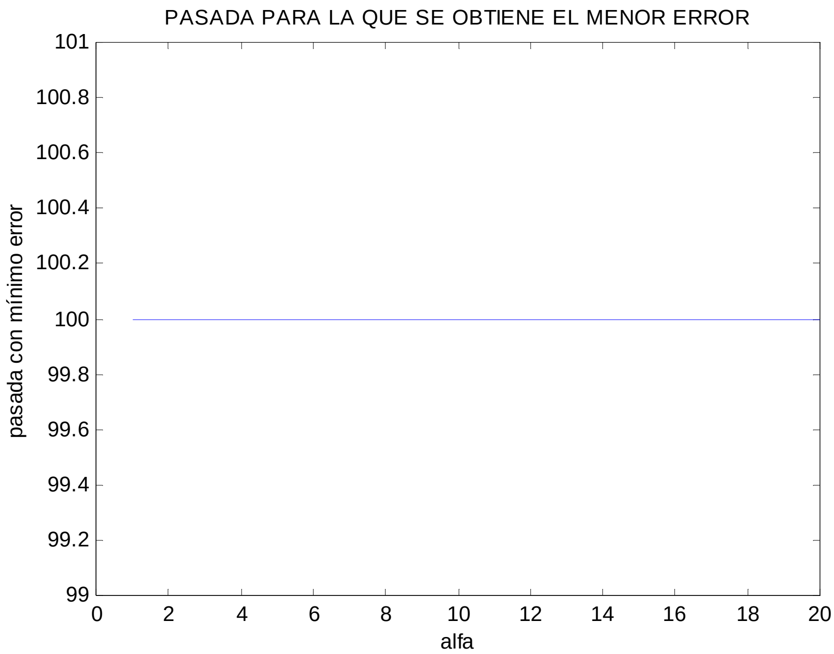


figura 49

COMPARACIÓN CON PID PARA REFERENCIA DOBLE:

Al igual que hicimos con la referencia original, vamos a compararla con el PID óptimo para esa regencia

kp =0.9000

kd =0.4000

ki =0.1000

suma: errores =50.2709

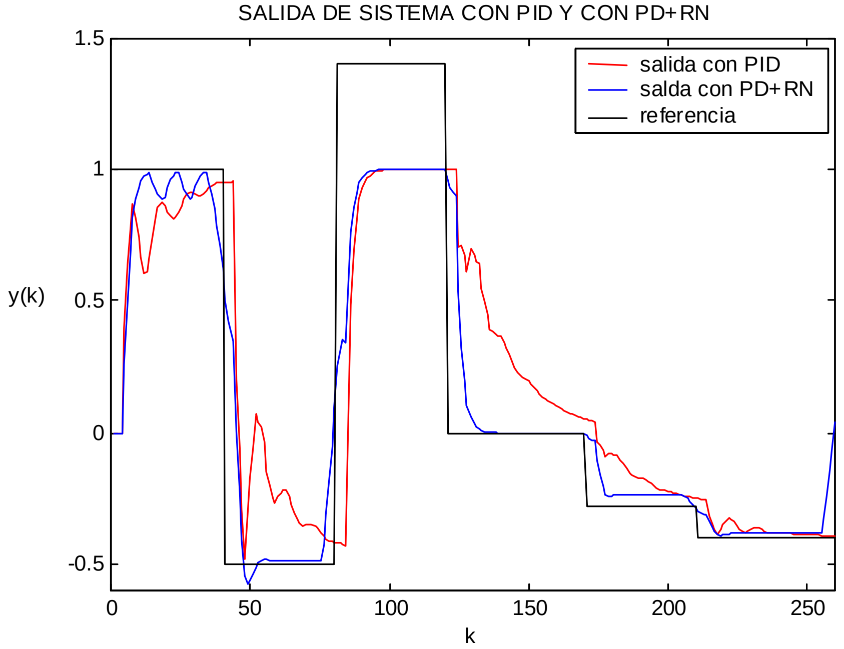


figura 50

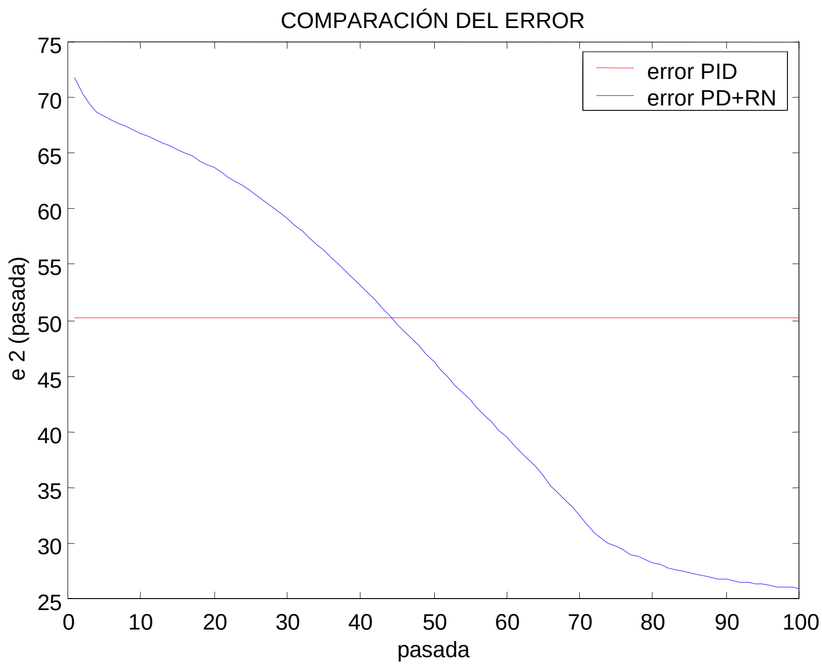


figura 51

COMPARACIÓN CON PID PARA REFERENCIA MITAD:

Hacemos exactamente lo mismo para la regencia mitad.(figuras 52 y 53)

5.1 CONCLUSIONES:

Usando regencia s con con escalones de amplitudes diferente, el esquema de control también funciona, y sigue consiguiendo mejores resultados que el PID optimizado para esas referencias.

Los parámetros que consiguen el mejor comportamiento cambian si utilizamos otra referencia.

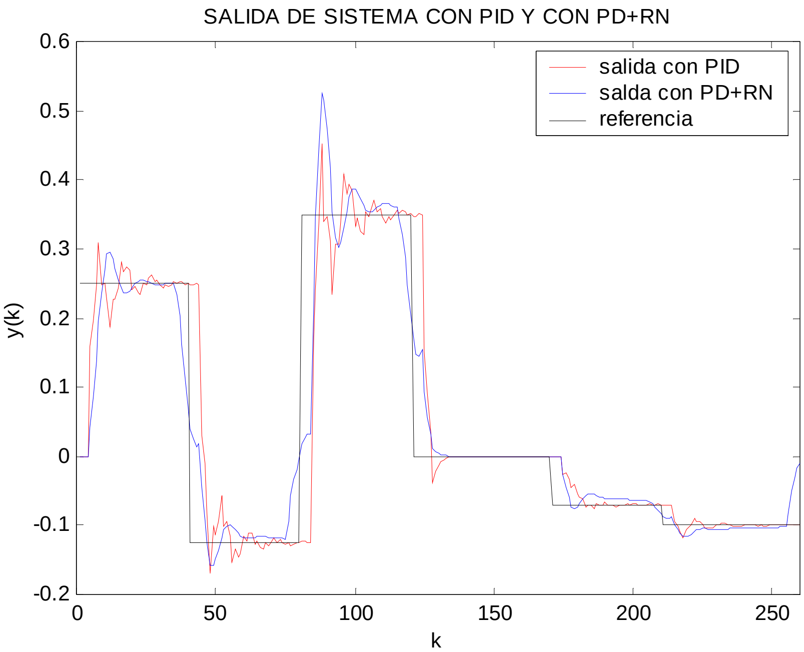


figura 52

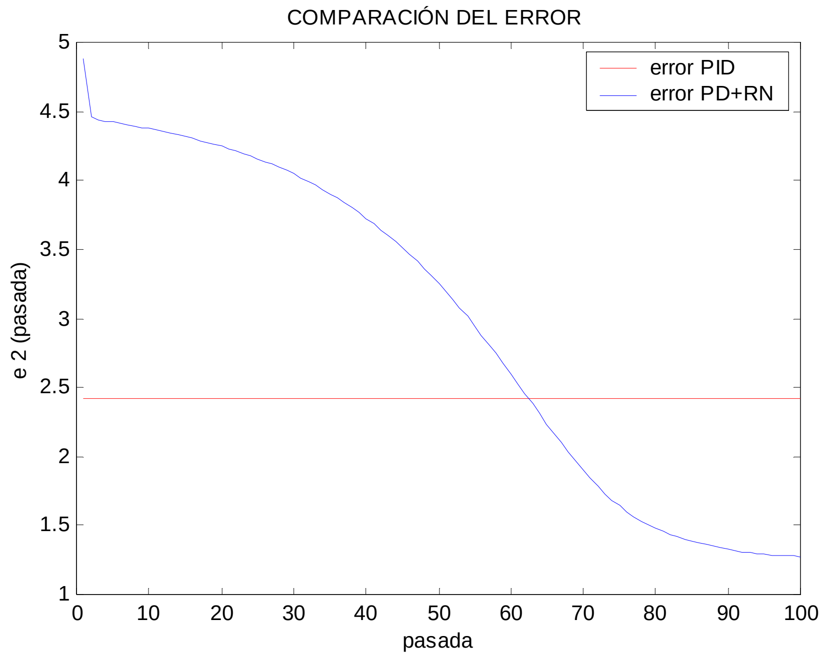

COMPARACIÓN DEL ERROR
error PID
error PD+RN
e 2 (pasada)
pasada
5
4.5
4
3.5
3
2.5
2
1.5
1
0
10
20
30
40
50
60
70
80
90
100

figura 53

6 TANDA 4:MEJORAS EN EL ESQUEMA DE CONTROL

6.1 PARADA DEL ENTRENAMIENTO DE LA RED

Para que este controlador pueda llevarse a aplicaciones reales es necesario que el controlador nunca “desaprenda”, es decir que cuando el comportamiento empiece a empeorar cesen los entrenamientos y los pesos de la red neuronal conserven el valor que tengan en ese momento.

Este mecanismo nos permitirá utilizar el controlador aún cuando no sea

Para implementar esta desactivación del aprendizaje un opción puede ser utilizar la derivada de la función suma de errores que hemos venido representando.

Esta curva va representando la suma de los errores (diferencia entre referencia y salida) al cuadrado para cada pasada de entrenamiento.
Mientras esta curva vaya decreciendo la suma de los errores al cuadrado va disminuyendo, es decir , la salida es mejor que en la pasada anterior, por lo que podemos concluir que la red está aprendiendo. En este caso no nos interesa parar la red, sino dejarla que siga
En caso de que la curva crezca, la suma de los errores al cuadrado será mayor que en la pasada anterior, por lo que detendremos el aprendizaje de la red y los pesos conservarán el valor que tengan en ese momento.

Para ello, lo primero que primero que se nos ocurre es imponer un condición del tipo:

Si Sumaerrores (k-1)-sumaerrores (k)> 0 entonces continúa aprendiendo
Sino detener aprendizaje

Una vez que esa condición se cumpla, ya no nunca más se actualizarán los pesos, sino que conservarán los valores que tenían.

Para evitar que un aumento puntual de la variable que suma los errores detenga el aprendizaje , podría ponerse una condición más compleja.

Si (Sumaerrores (k-1)-sumaerrores (k)< 0 durante t pasadas) o (Sumaerrores (k-1)-sumaerrores (k)>umbra_error) entonces detener aprendizaje

Sino continúa aprendiendo

Vamos a ver un ejmplo:

Partimos de un experimento satisfactorio, en el que la red va mejorando su comportamiento a lo largo de las pasadas:

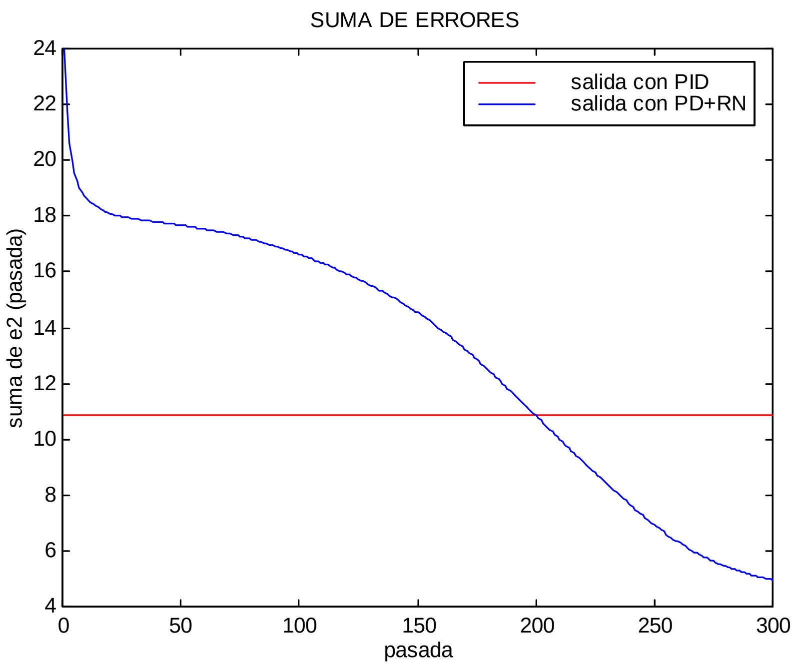


figura 54

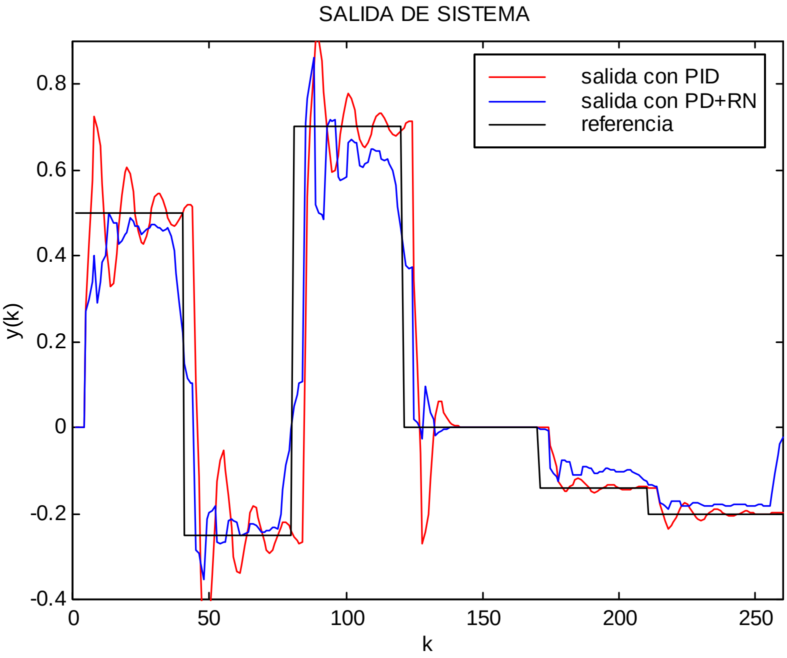


figura 55

Aumentamos el número de pasadas de 300 a 2000 y vemos cómo los resultados empeoran sensiblemente (figuras 56 y 57)

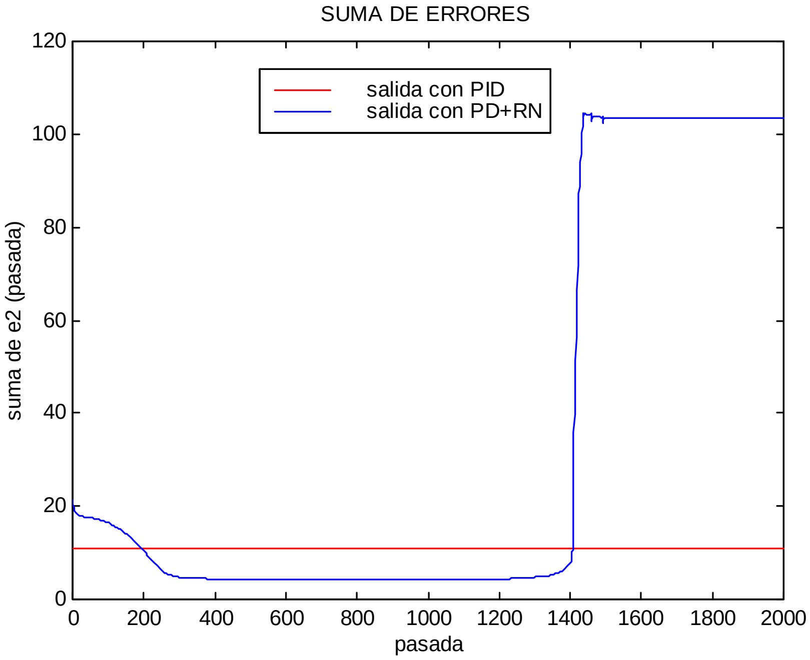


figura 56

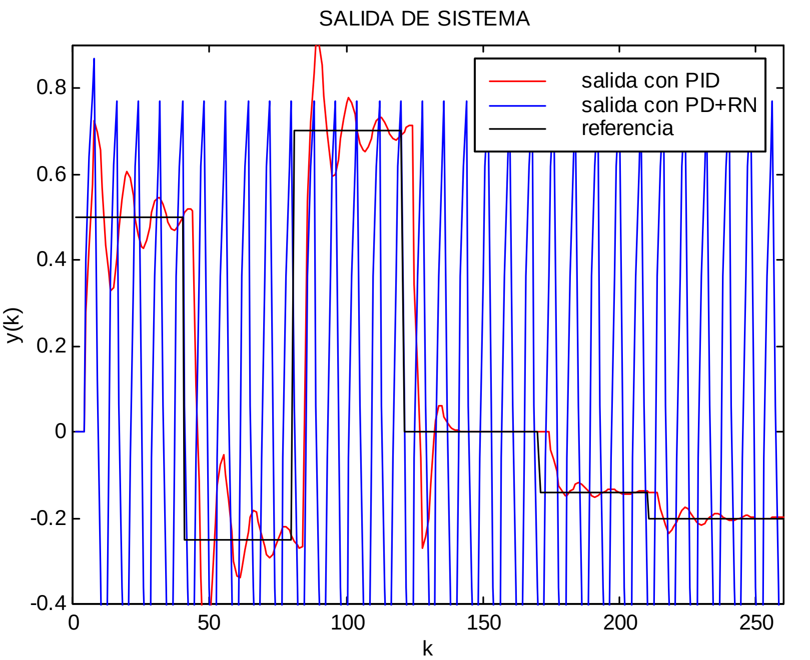


figura 57

A pasar de que los parámetros utilizados eran los adecuados el sistema se ha ido degradando a los largo se las pasadas de aprendizaje, llegando a ser incluso mucho peor que un controlado PID a secas.

Si introducimos en el código las condiciones descritas, vemos como el sistema detiene su aprendizaje y se mantiene en el punto de entrenamiento en que presentaba mejores resultados (figuras 58 y 59)

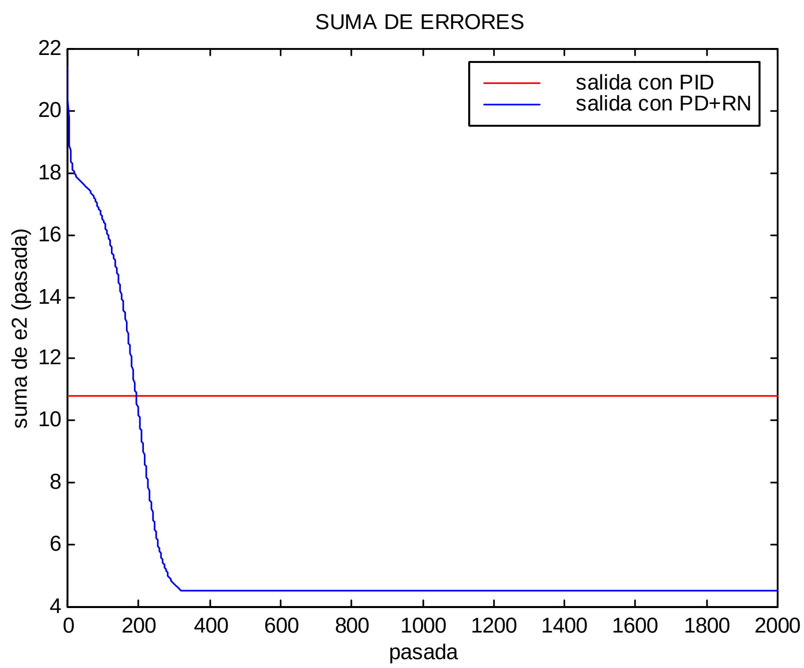


figura 58

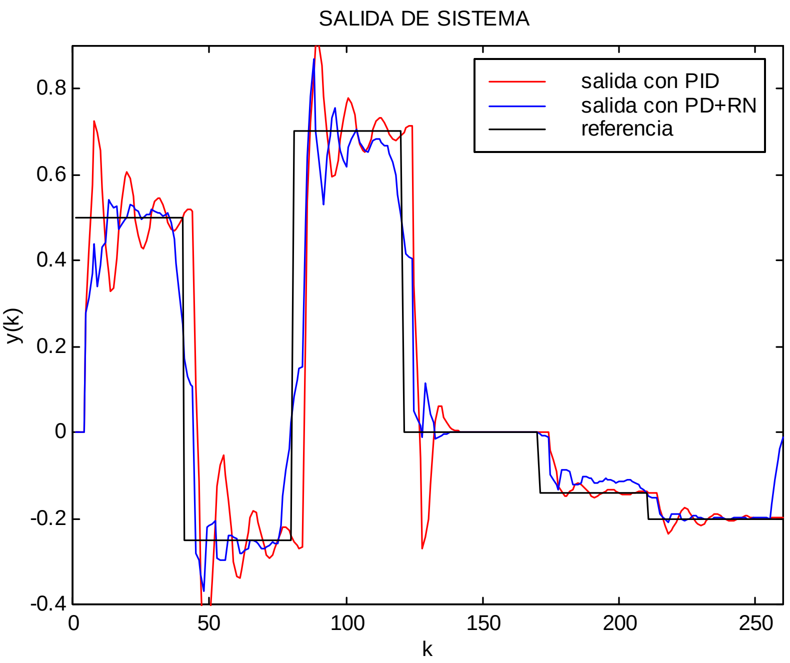


figura 59

6.2 INICIACIÓN DE LOS PESOS

Otro aspecto importante para que este controlador pueda implementarse en la práctica es que sus resultados sean aceptables desde el primer momento

Para ello habría dos posibilidades:

- Utilizar el PID y comenzar con los pesos de la red a 0:

Con este PID ya hemos visto que el efecto integral desestabiliza el sistema.
Además hemos comprobado también que existen valores de kd , kp que mejoran el comportamiento del sistema.

Si calculamos cuál sería el PD optimo (sin considerar la posibilidad de un término integral), el resultado obtenido es .

kp =0.9000

kd =0.3000

suma_errores =17.1292

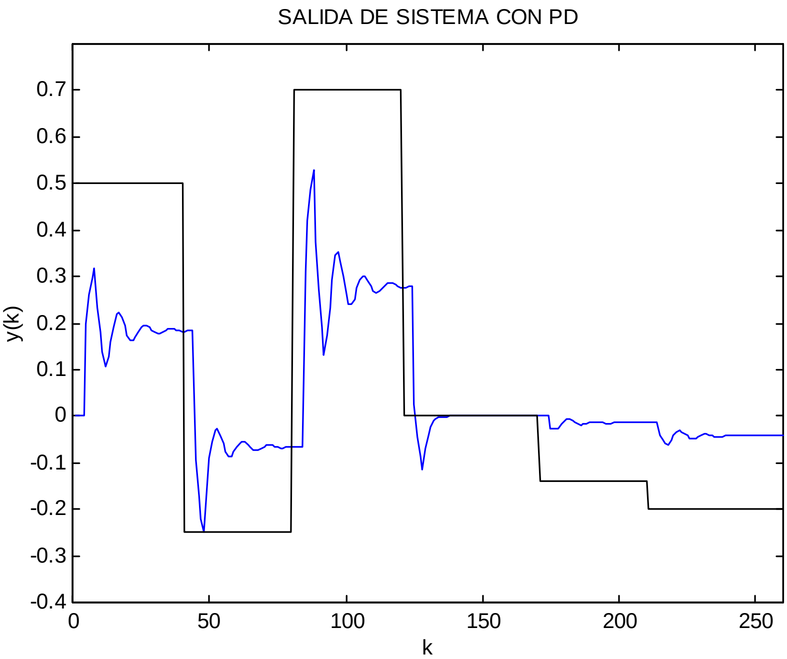


figura 60

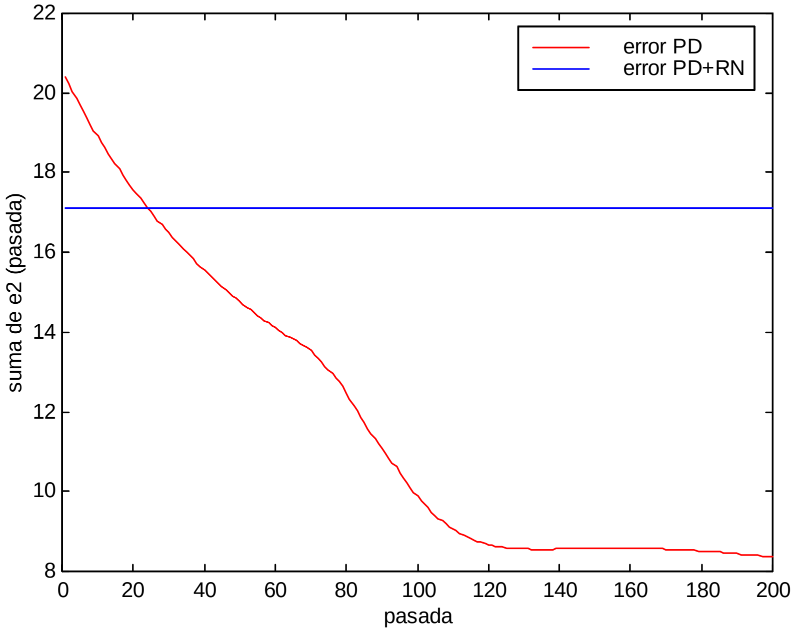

error PD
error PD+RN
suma de e2 (pasada)
pasada
22
20
18
16
14
12
10
8
0
20
40
60
80
100
120
140
160
180
200

figura 61

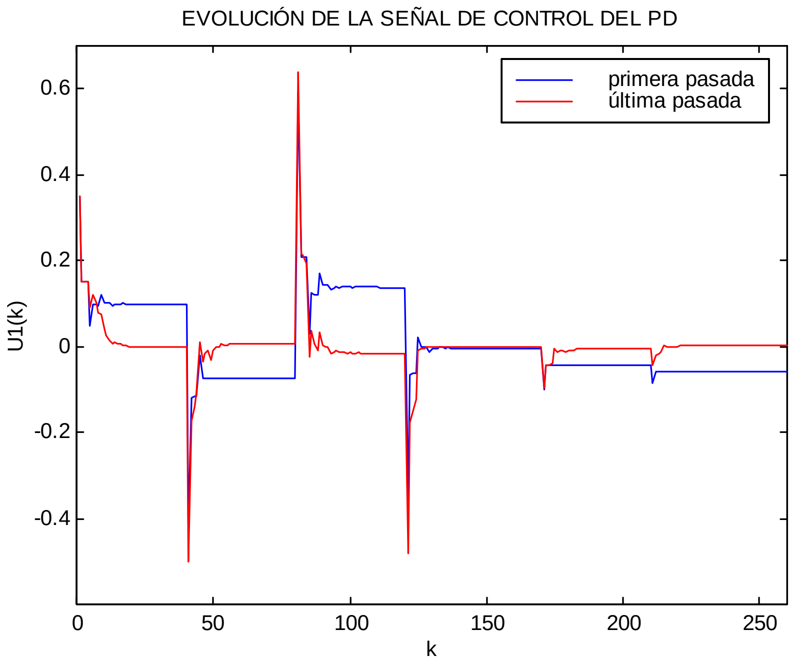


figura 62

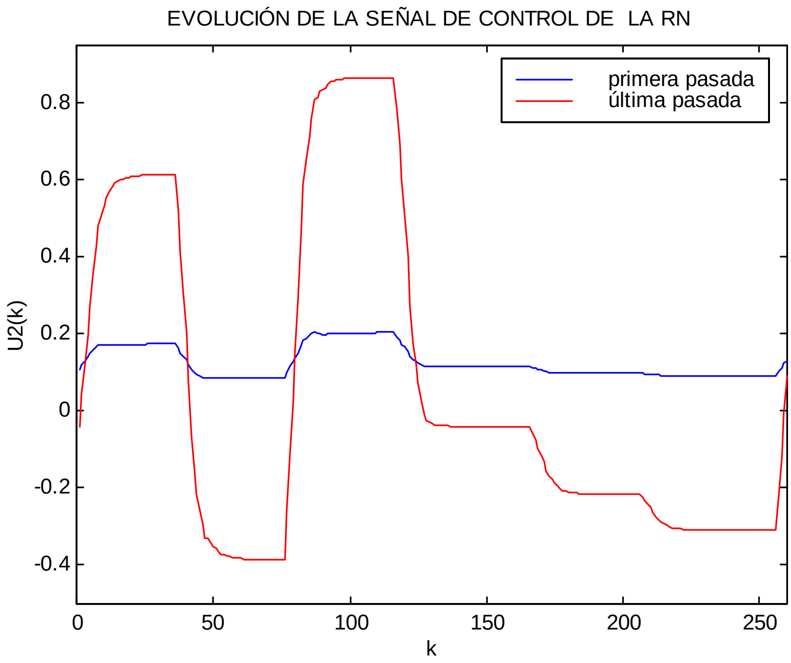


figura 63

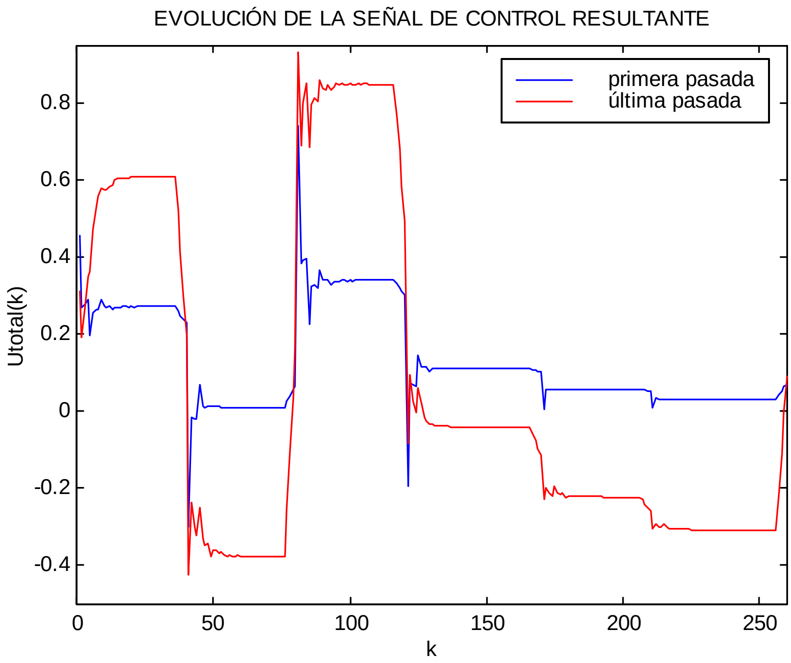


figura 64

Ya que nos interesa el error en el primera pasada vamos a representar en una gráfica las dos componentes de la señal de control durante la primera pasada:

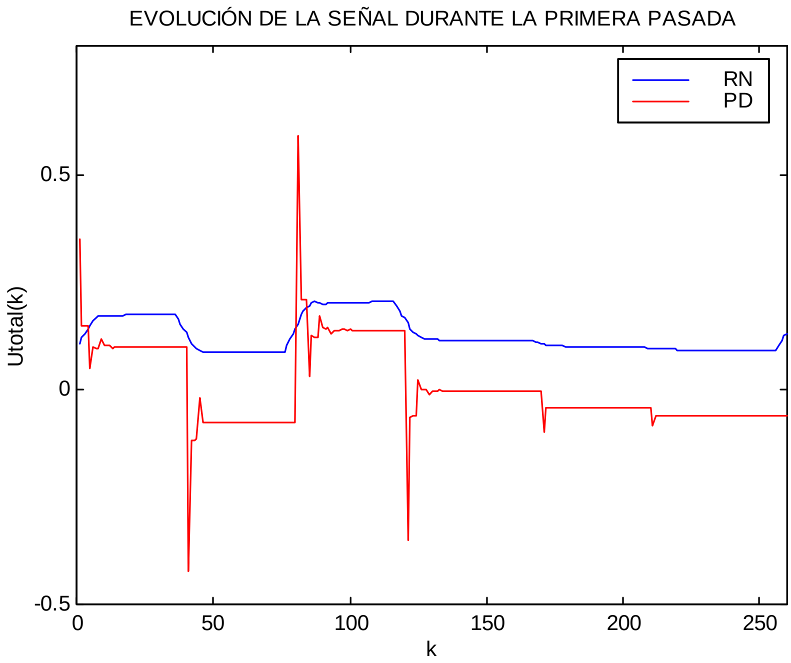


figura 65

Si ahora intentamos

- Podríamos hacer que los pesos iniciales sean 0, de manera que cuando comience a funcionar el sistema sólo esté presente el controlador PD.
- Los pesos que son 0 inicialmente ya no alcazarán otro valor distinto a lo largo de los entrenamientos, a excepción de Wts.
- Los valores del parámetros del PD (kd,kp) que proporcionan un buen rendimiento del sistema a lo largo de los aprendizajes, no tiene por qué coincidir con los del PD óptimo

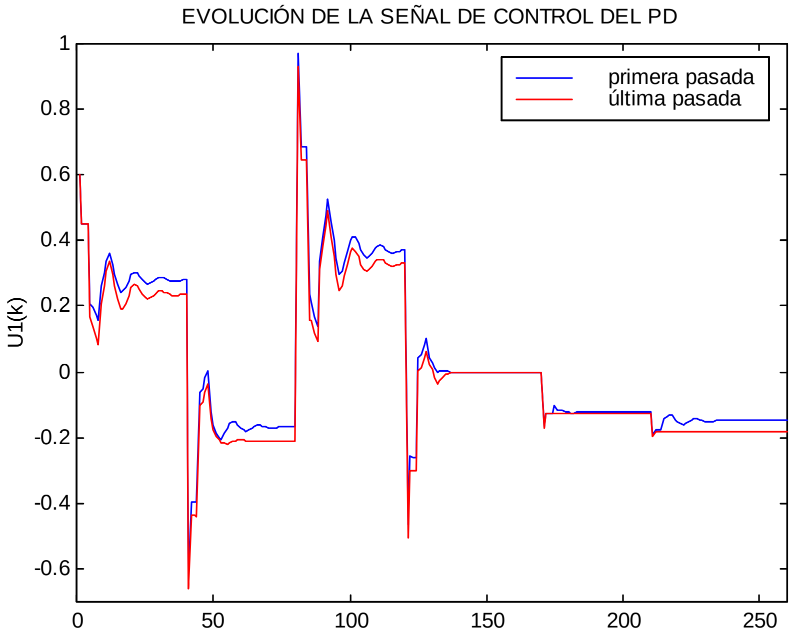


figura 66

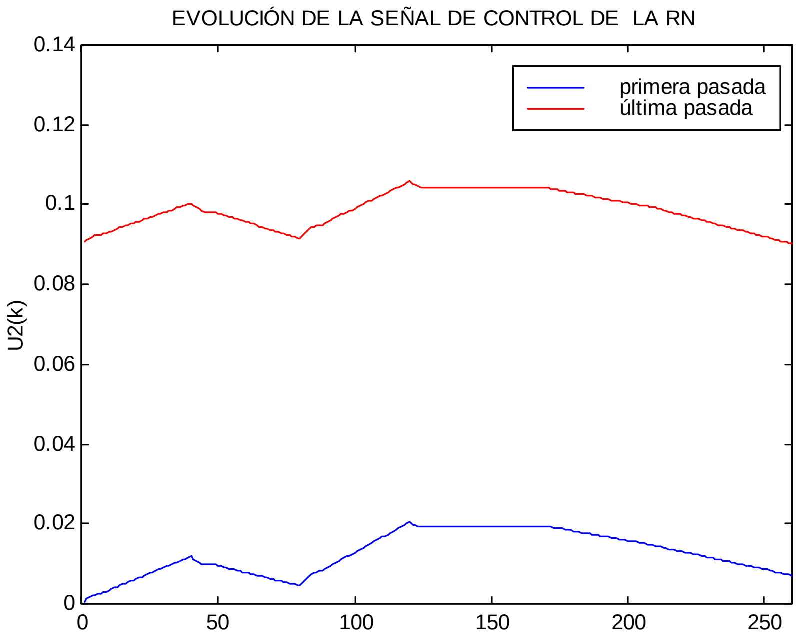


figura 67

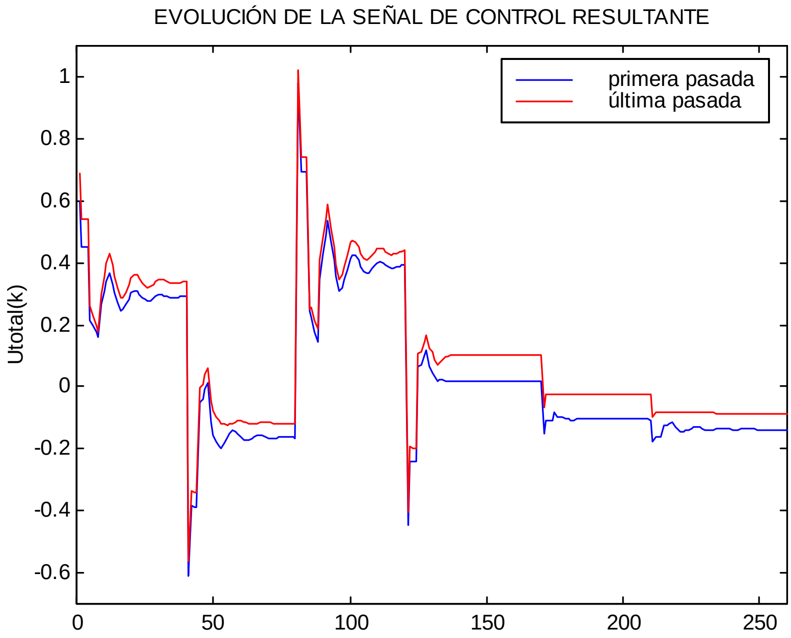


figura 68

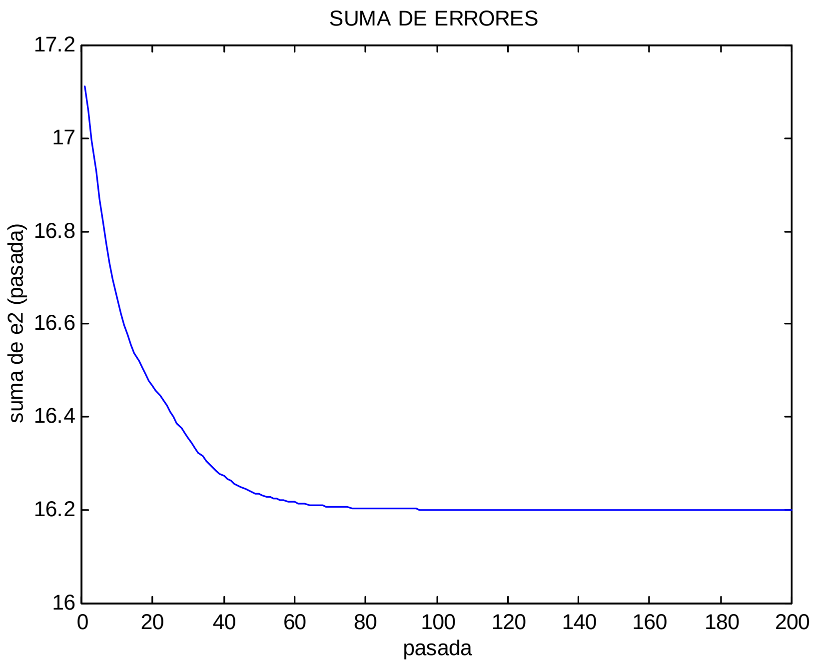


figura 69

Vemos que efectivamente el comportamiento inicial es igual que el de el PD óptimo, pero las mejoría que se obtiene a lo largo de las pasadas es despreciable.

Igualar la salida inicial de la red a la un PD :

De esta forma sería como dos PD en paralelo cuya señal de control se suma a la salida:

- Utilizar el PD e inicializar los pesos de manera que la salida inicial del controlador sea la que proporcionaría el PD óptimo.

Las variables deberán relacionarse de la siguiente manera:

Los parámetros k , que hacen óptimo el PD son la suma de los que se hacen que el sistema con RN tenga un buen comportamiento + los que usaremos para inicializar la red.

$$K_p^{PDoptimo} = K_p^{iniciored} + K_p^{aprendizqajeRN}$$

$$K_d^{PDoptimo} = K_d^{iniciored} + K_d^{aprendizqajeRN}$$

es decir,

$$K_p^{iniciored} = K_p^{PDoptimo} - K_p^{aprendizqajeRN}$$

$$K_d^{iniciored} = K_d^{PDoptimo} - K_d^{aprendizqajeRN}$$

Las variable de inicio de red son las que llamaremos Kp, Kd para simplificar a lo largo de esta tanda :

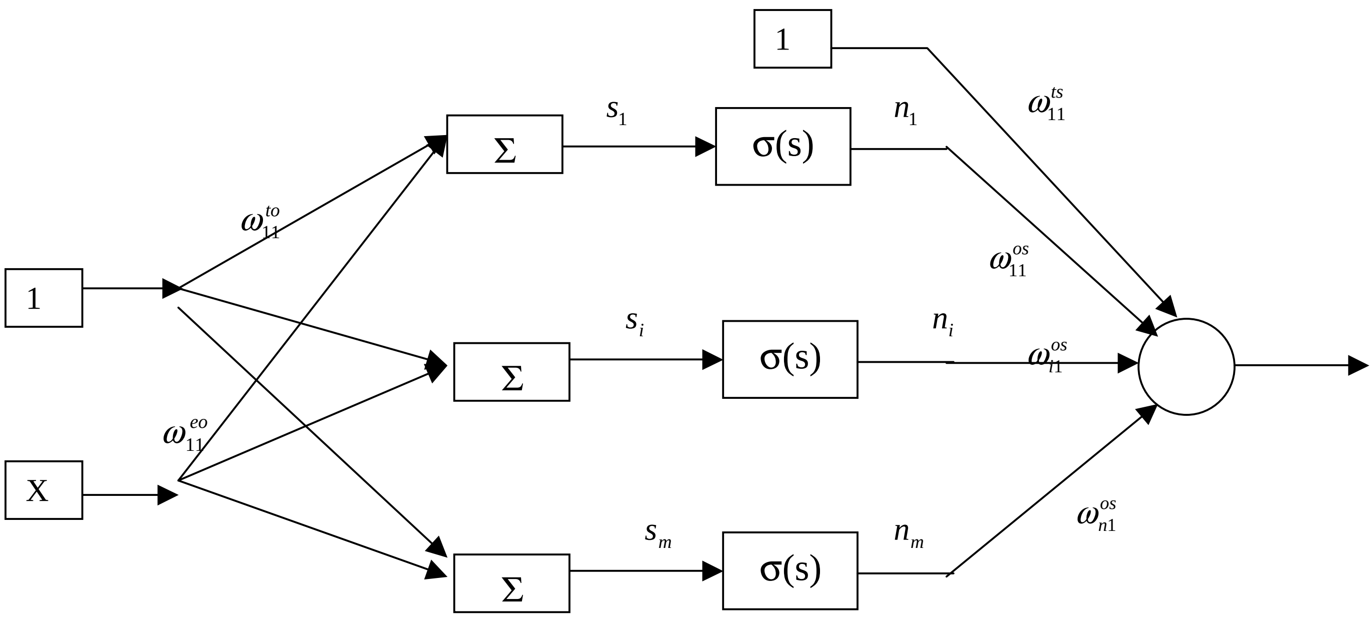


figura 70

Uk =PD(rk, rk-1, yk)

$n_1 = r(k)$

$n_2 = y(k)$

$n_3 = r(k-1)$

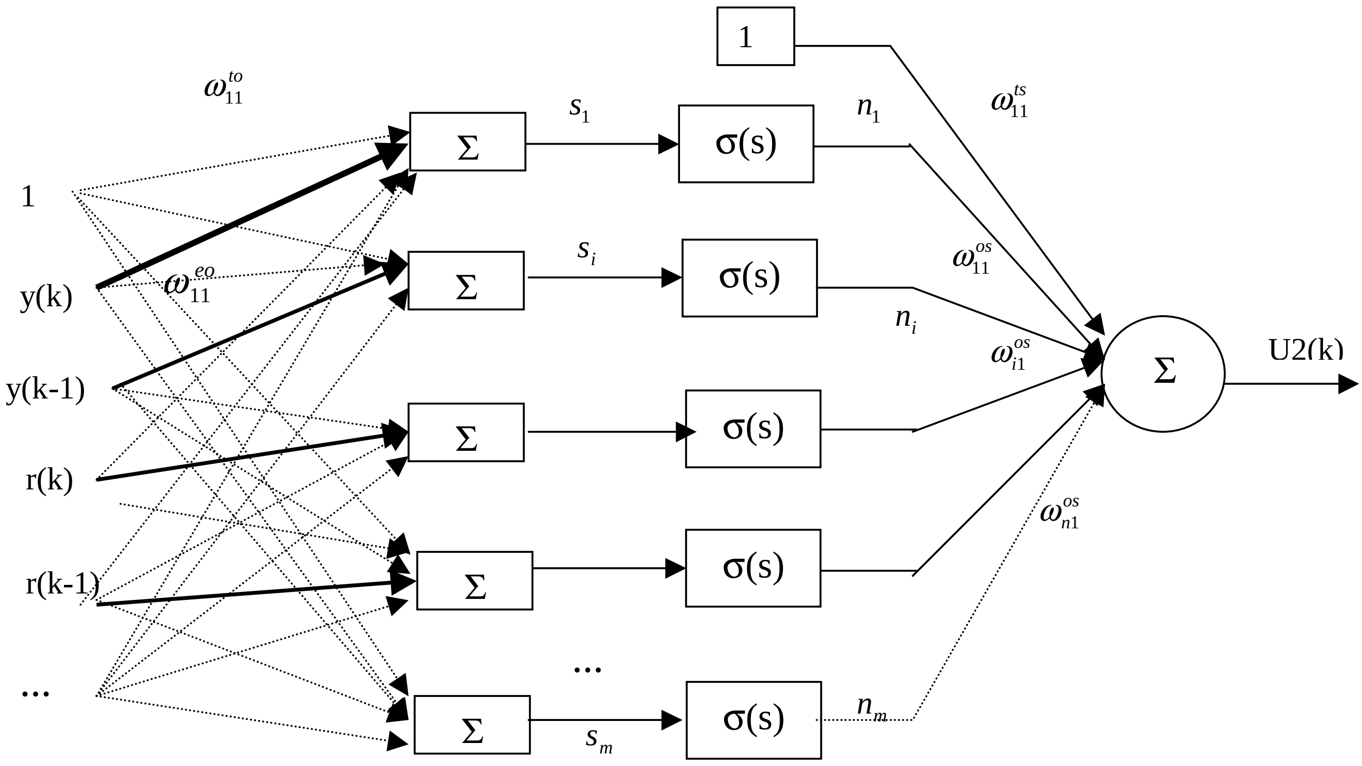
$n_4 = y(k-1)$

$$X^* = \begin{bmatrix} 1 \\ y(k) \\ y(k-1) \\ y(k-2) \\ y(k-3) \\ r(k+ant) \\ r(k+ant-1) \\ \ldots \\ r(k+1) \\ r(k) \\ r(k-1) \\ U(k) \\ U(k-1) \\ U(k-2) \\ \ldots \end{bmatrix}$$

donde la variable *ant* es la anticipación a la referencia que conocemos.

Para que la salida de la red sea igual que la un PID sólo necesitamos utilizar algunos pesos, los correspondientes a r (k) ,r(k-1) ,y(k) , y(k-1).

Esto es e(k-1) = r(k-1) - y(k-1)



1
ω_{11}^{to}
s_1
$\sigma(s)$
n_1
ω_{11}^{ts}
1
y(k)
ω_{11}^{eo}
Σ
s_i
ω_{11}^{os}
n_i
ω_{i1}^{os}
y(k-1)
U2(k)
r(k)
ω_{n1}^{os}
r(k-1)
...
n_m
s_m

figura 71

Por cada variable de entrada contenida en la fórmula de un PD ,sólo un peso de entrada tendrá un valor distinto de cero, y además será un valor pequeño para que su convergencia con los valores a los que tenderá a medida que se sucedan los aprendizajes sea la adecuada.

El resto de variables no implicadas en un PD, tendrán todos los pesos de entrada a 0 en el momento de la inicialización:

Como ya vimos en el apartado 2.3.1, la señal de control de un PID está representada por la ecuación:

$$u_k = \left[Kp \cdot e_k + Ki \cdot \sum_{j=0}^{k} e_j + Kd \cdot (e_k - e_{k-1}) \right]$$

$$u_k = Kp \cdot (r_k - y_k) + Kd \cdot (r_k - y_k - r_{k-1} + y_{k-1})$$

$$u_k = r_k \cdot (Kp + Kd) - r_{k-1} \cdot Kd - y_k \cdot (Kp + Kd) + y_{k-1} \cdot Kd$$

A partir de esta fórmula ya podemos comenzar a asignar los valores iniciales a los pesos de la red.

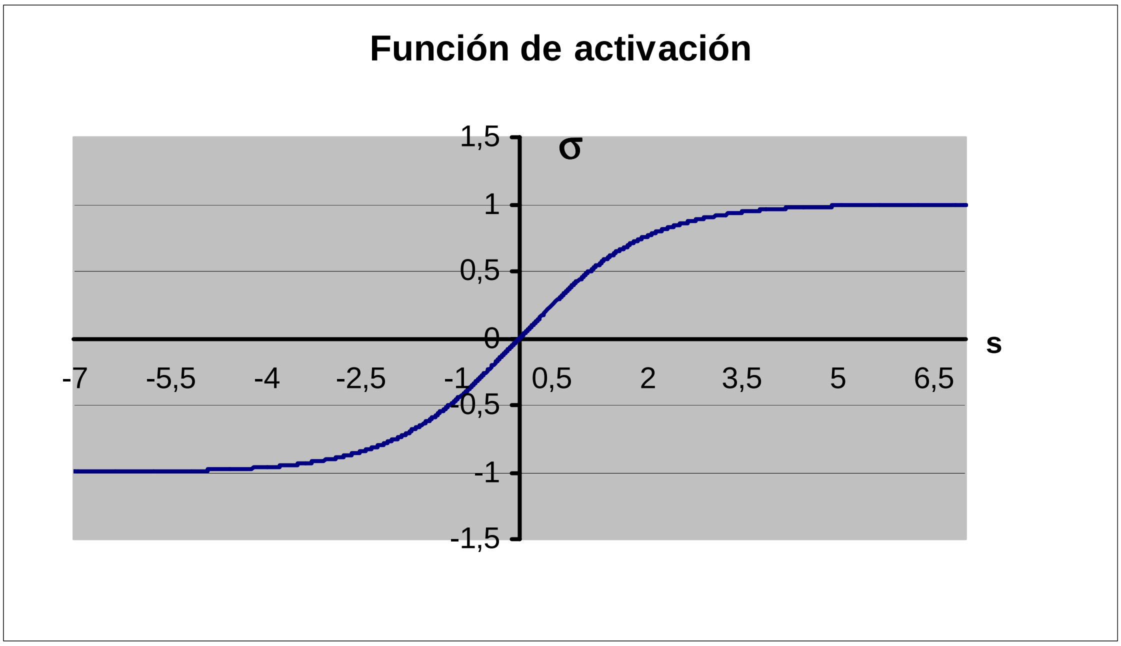


figura 72

Vemos que para valores cercanos a 0 , la función presenta un comportamiento aproximadamente lineal. Esta será la zona que nos interese para poder representa un PID:

Hacemos una interpolación lineal de la función para el rango de valores cercano a cero, que es en el que se aprecia el comportamiento lineal, y representamos el resultado:

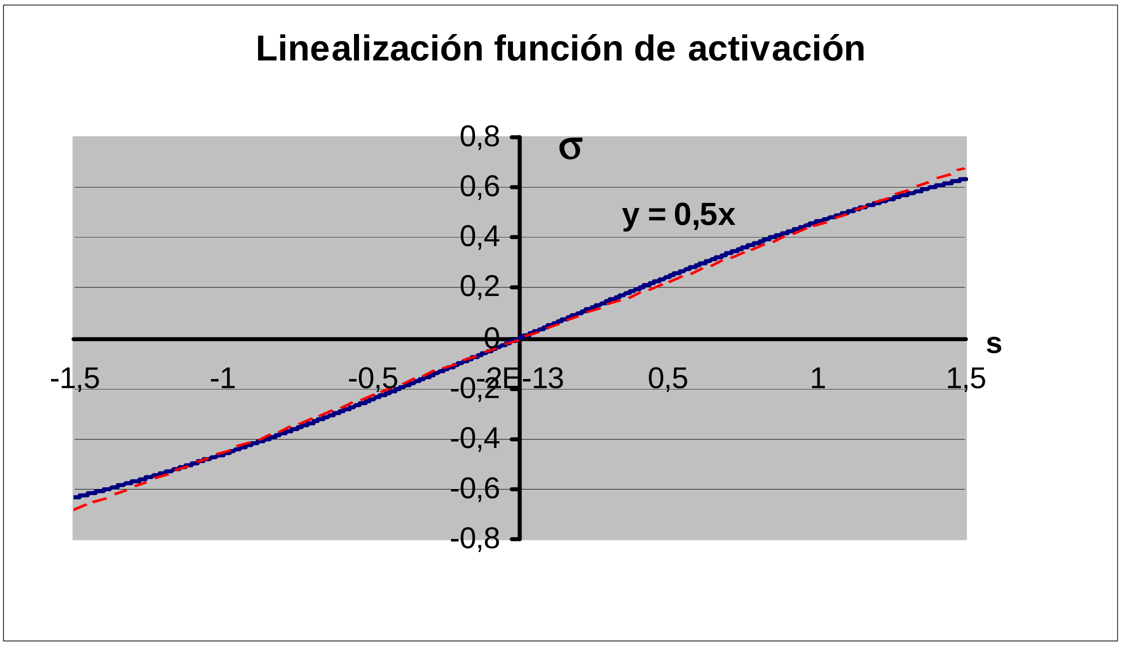


figura 73

Como se desprende de la gráfica , para valores de s comprendidos entre –1,5 y 1,5 podemos aproximar :

$$\sigma(s) \approx 0{,}5 \cdot s \qquad \text{para} \quad s \in (-1.5,+1.5)$$

Teniendo en cuenta que los valores de r(k) e y(k) que venimos utilizando, podemos conseguir que la salida dependa linealmente de r(k), y(k), r(k-1) e y(k-1), al igual que sucede en un controlador PD. Debemos procurar en los valores de s estén contenidos en ese intervalo cuando comience a aprender la red.

Para favorecer valores pequeños de s , podríamos hacer pequeños aquellos valores de W^{eo} que son distintos de 0.

Hacer muy pequeños esos valores tienen el inconveniente de que serían igualados demasiado pronto por los valores de W^{eo} que son 0 de inicio, y por lo tanto la red dejaría de ser un PD antes de lo necesario.

Debemos buscar por tanto una solución de compromiso para los valores de W^{eo}, de manera que sean lo suficientemente pequeños para que la salida de la función de activación sea lineal con los valores de s, y lo suficientemente grandes como para que destaquen sobre los valores de W^{eo} que eran 0 de inicio, pero que irán aumentado a medida que comience en entrenamiento.
Vamos a ir analizando cada variable de entrada y el valor que deberán tomar sus pesos para asemejarse a un PD:

rk:

$W_{10,nn}$

$W^{eo}_{10,1} = 1$

$W^{eo}_{10,x} = 0$ para x=1...nn

rk-1

$W^{eo}_{11,2} = 1$

$W^{eo}_{11,x} = 0$ para x=1...nn

yk

$W^{eo}_{1,3} = 1$

$W^{eo}_{1,x} = 0$ para x=1...nn

yk-1

$W_{2,4}^{eo} = 1$

$W_{2,x}^{eo} = 0$ para x=1...nn

Los de salida:

$$W_1^{os} = 2 \cdot \frac{\partial U_{pd}}{\partial r_k} = 2(Kp + Kd)$$

$$W_{21}^{os} = 2 \cdot \frac{\partial U_{pd}}{\partial r_{k-1}} = -2Kd$$

$$W_{31}^{os} = 2 \cdot \frac{\partial U_{pd}}{\partial y_k} = -2(Kp + Kd)$$

$$W_{41}^{os} = 2 \cdot \frac{\partial U_{pd}}{\partial y_{k-1}} = 2\,Kd$$

Aplicamos estos valores a los pesos iniciales

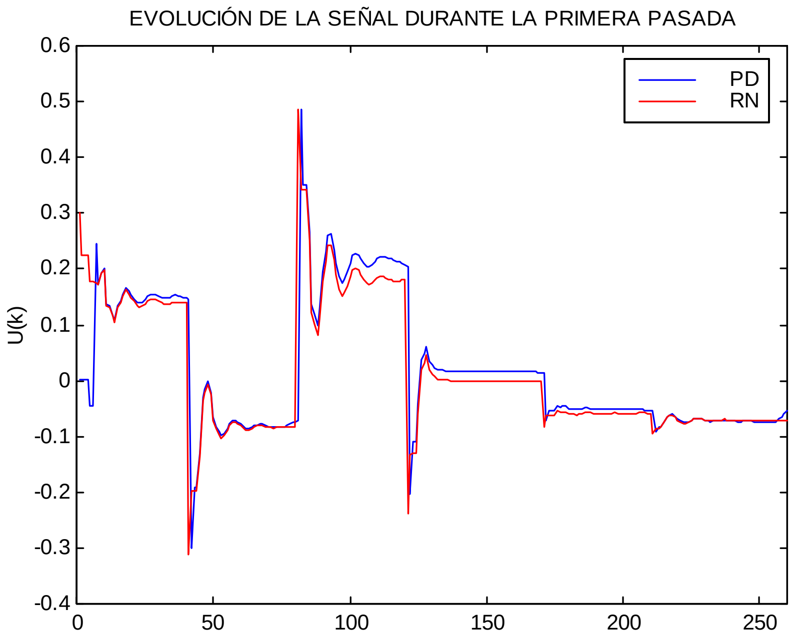


figura 74

Vemos que como buscábamos durante al primera pasada tenemos dos PD con salidas prácticamente iguales, cuya suma será la del PD óptimo.

Sin embargo, vamos a ver que va a ir ocurriendo a lo largo de los entrenamientos:

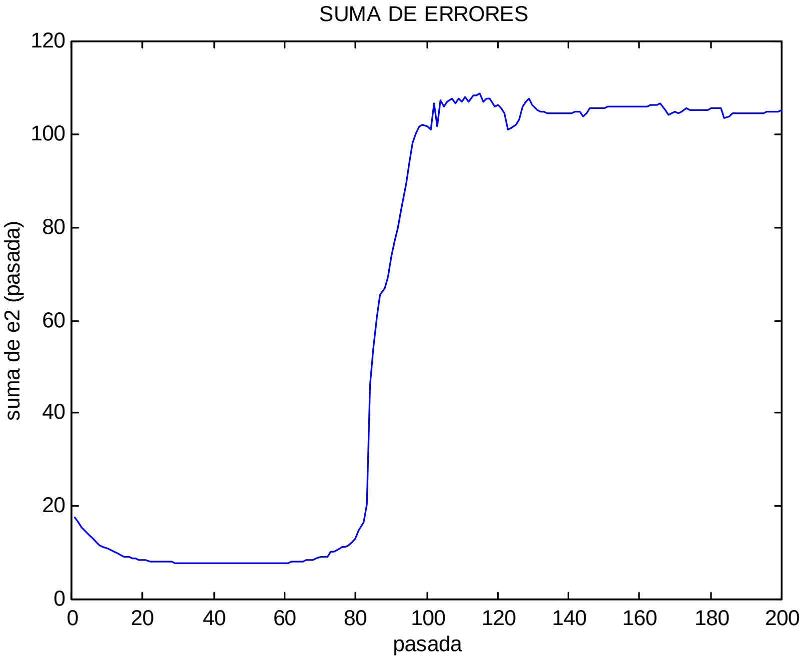


figura 75

El sistema se ha ido desestabilizando y el resultado ha ido empeorando de manera notable. Vamos a analizar por qué:

Si analizamos las fórmulas de actualización de los pesos:

$$\omega_{11}^{ts} = \omega_{11}^{ts} + \alpha \bullet e$$

$$\omega_{i1}^{os} = \omega_{i1}^{os} + \alpha \bullet e \bullet n_i$$

$$\omega_{1i}^{to} = \omega_{1i}^{to} + \alpha \bullet e \bullet \omega_{i1}^{os} \bullet \sigma'(s_i)$$

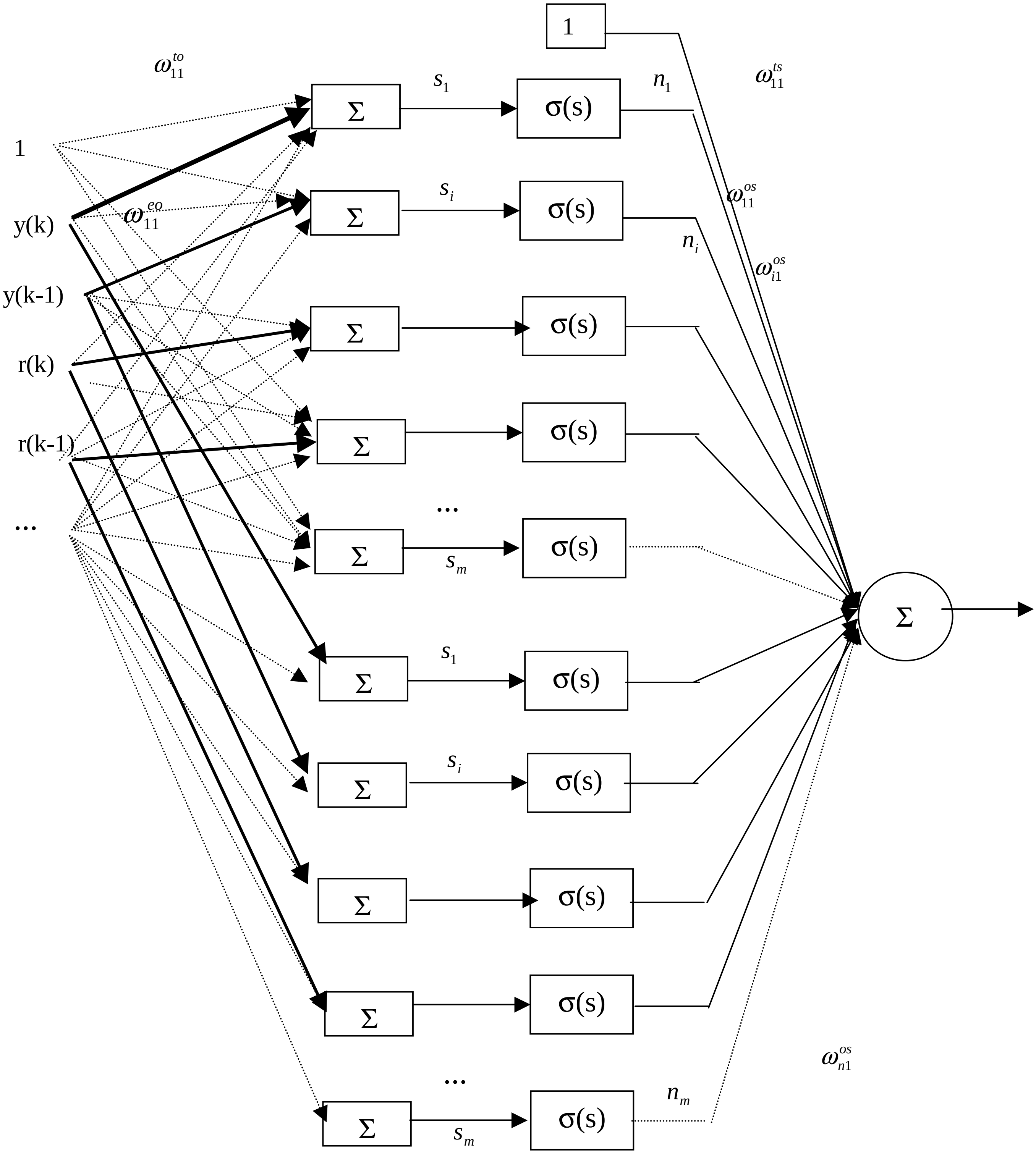
$$\omega_{1i}^{eo} = \omega_{1i}^{eo} + \alpha \bullet e \bullet \omega_{i1}^{os} \bullet \sigma'(s_i) \bullet x$$

Podemos entender que si un peso es cero inicialmente, nunca deja de ser 0, a excepción de Wts.

Por lo tanto , al hacer 0 inicialmente tantos pesos , hemos pasado a tener un sistema con una RN de sólo 4 neuronas:

Además, los pesos que no son cero inicialmente toman valores demasiado grandes. Vamos a intentar solucionar estas situaciones:

Hacemos que haya menos pesos iniciales con valor 0, de forma que se utilicen todas las neuronas. De esta forma conseguimos que el valor de los pesos de entrada que no son 0 no sea tan alto.


1
ω_{11}^{to}
s_1
n_1
ω_{11}^{ts}
1
y(k)
ω_{11}^{eo}
y(k-1)
r(k)
r(k-1)
...
s_i
n_i
ω_{11}^{os}
ω_{i1}^{os}
Σ
σ(s)
...
s_m
s_1
s_i
ω_{n1}^{os}
...
n_m
s_m

rk:

$W_{10,nn}$

$W^{eo}_{10,1} = 0.5$

$W^{eo}_{10,6} = 0.5$

$W^{eo}_{10,x} = 0$ para el resto de x=1...nn

rk-1

$W^{eo}_{11,2} = 0.5$

$W^{eo}_{11,7} = 0.5$

$W^{eo}_{11,x} = 0$ para el resto de x=1...nn

yk

$W^{eo}_{1,3} = 0.5$

$W^{eo}_{1,8} = 0.5$

$W^{eo}_{1,x} = 0$ para el resto de x=1...nn

yk-1

$W^{eo}_{2,4} = 0.5$

$W^{eo}_{2,9} = 0.5$

$W^{eo}_{2,x} = 0$ para el resto de x=1...nn

Los de salida:

$$W_{11}^{os} = 2 \cdot \frac{\partial U_{pd}}{\partial r_k} = 2(Kp + Kd)$$

$$W_{61}^{os} = 2 \cdot \frac{\partial U_{pd}}{\partial r_k} = 2(Kp + Kd)$$

$$W_{21}^{os} = 2 \cdot \frac{\partial U_{pd}}{\partial r_{k-1}} = -2Kd$$

$$W_{71}^{os} = 2 \cdot \frac{\partial U_{pd}}{\partial r_{k-1}} = -2Kd$$

$$W_{31}^{os} = 2 \cdot \frac{\partial U_{pd}}{\partial y_k} = -2(Kp + Kd)$$

$$W_{81}^{os} = 2 \cdot \frac{\partial U_{pd}}{\partial y_k} = -2(Kp + Kd)$$

$$W_{41}^{os} = 2 \cdot \frac{\partial U_{pd}}{\partial y_{k-1}} = 2Kd$$

$$W_{91}^{os} = 2 \cdot \frac{\partial U_{pd}}{\partial y_{k-1}} = 2Kd$$

Si aplicamos estos nuevos pesos iniciales y hacemos alfa (velocidad de aprendizaje) más baja, por ejemplo la mitad, y aumentamos el número de pasadas para que le dé tiempo a prender, obtenemos lo resultados siguientes:

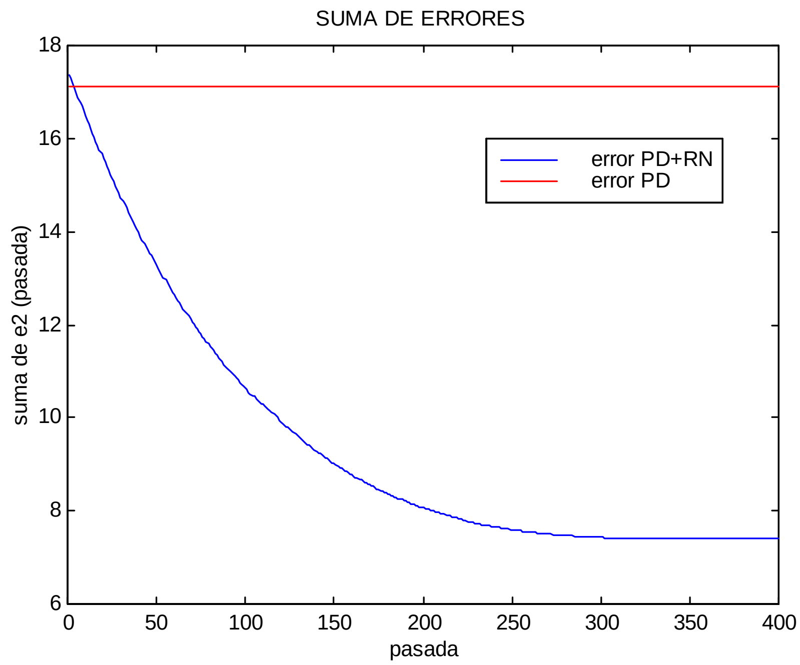


figura 76

Vemos que el comportamiento ha mejorado considerablemente. Partimos de un comportamiento similar al que se obtiene con un PD óptimo y a partir vamos mejorando en cada pasada.

6.3 EFECTO INTEGRAL

Sin embargo, ya vimos que con un PID podemos obtener resultados mejores que los que hemos obtenido con el PD óptimo:

Si con los valores del PD óptimo vamos variando ki, y representamos la suma de los errores al cuadrado que vamos obteniendo en cada caso:

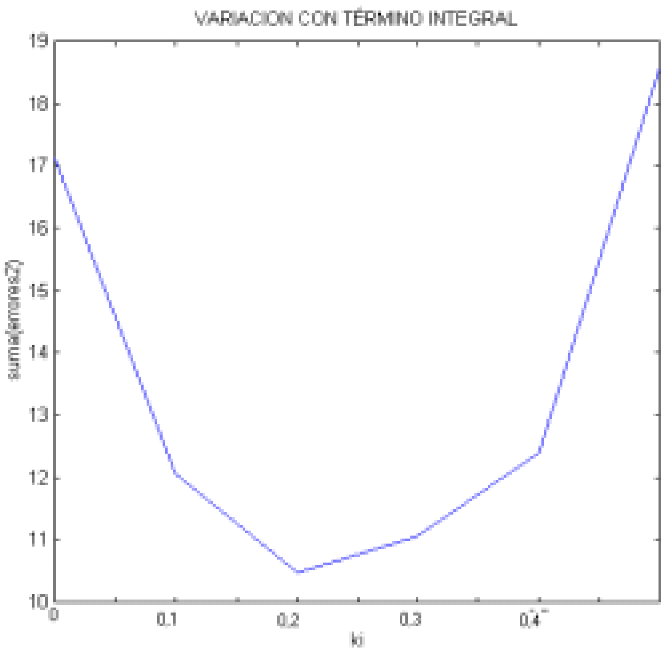
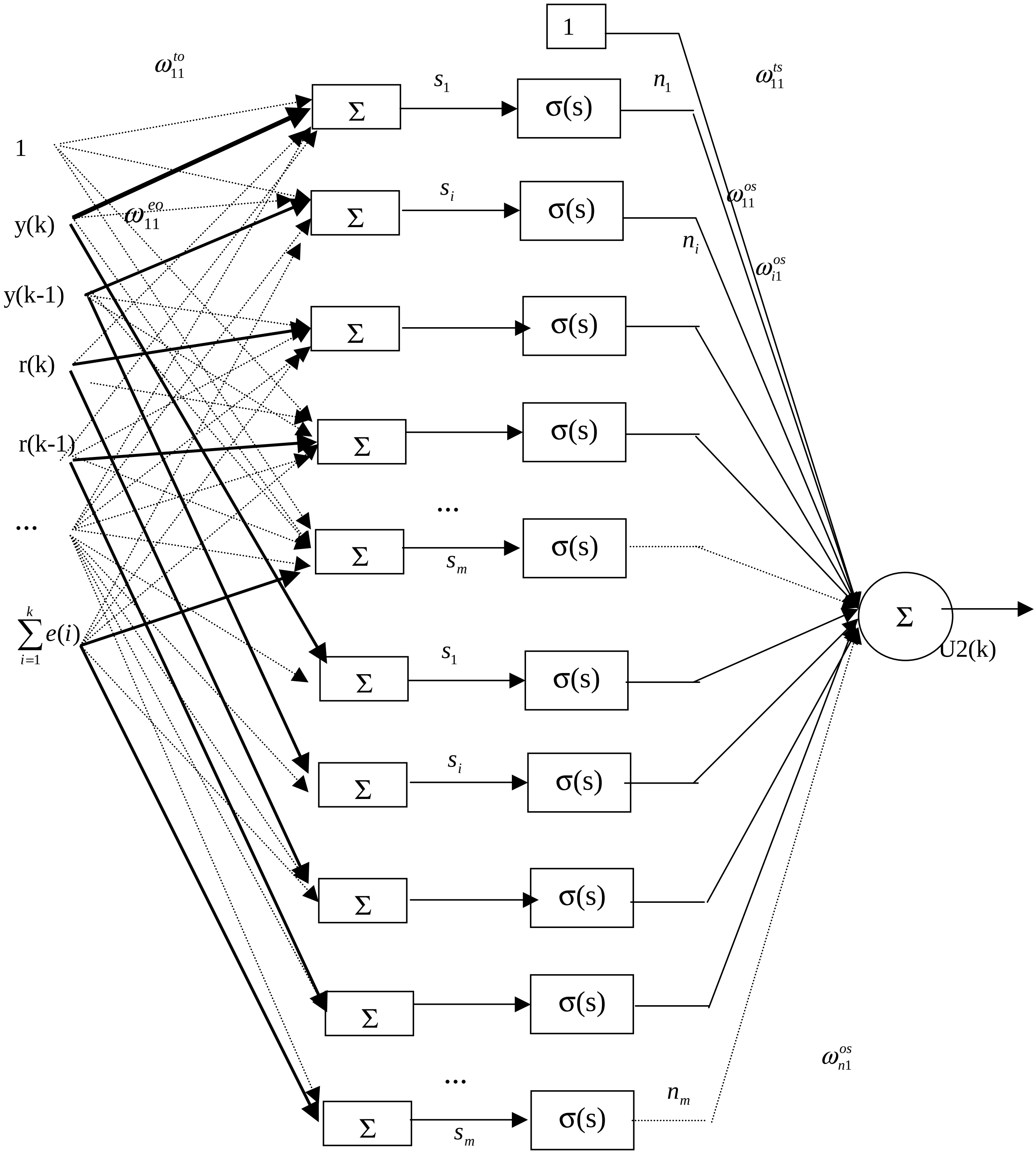


figura 77

Vemos que la suma de errores que se comente con un PID, puede mejorar sensiblemente a la que se comete con un PD añadiendo un término integral. Añadir el término integral al PD ya vimos en la primera tanda de experimentos que desestabilizaba el sistema. Un posible opción sería que la RN añadiera ese término integral mediante la inicialización de los pesos.

Para eso entre las entradas de la RN , que forman el vector X, añadimos un término igual a la suma de los errores que se llevan comentados en esa pasada :

$$X^* = \begin{bmatrix} 1 \\ y(k) \\ y(k-1) \\ y(k-2) \\ y(k-3) \\ r(k+ant) \\ r(k+ant-1) \\ \dots \\ r(k+1) \\ r(k) \\ r(k-1) \\ \sum_{i=1}^{k} e(i) \\ U(k) \\ U(k-1) \\ U(k-2) \\ \dots \end{bmatrix}$$


1
ω_{11}^{to}
ω_{11}^{ts}
s_1
n_1
1
y(k)
ω_{11}^{eo}
s_i
ω_{11}^{os}
n_i
ω_{i1}^{os}
y(k-1)
r(k)
r(k-1)
...
s_m
$\sum_{i=1}^{k} e(i)$
U2(k)
ω_{n1}^{os}
n_m
σ(s)
Σ

Vemos como quedarán lo valores iniciales.

rk:

$W_{10,nn}$

$W^{eo}_{10,1} = 0.5$

$W^{eo}_{10,6} = 0.5$

$W^{eo}_{10,x} = 0$ para el resto de x=1...nn

rk-1

$W^{eo}_{11,2} = 0.5$

$W^{eo}_{11,7} = 0.5$

$W^{eo}_{11,x} = 0$ para el resto de x=1...nn

yk

$W^{eo}_{1,3} = 0.5$

$W^{eo}_{1,8} = 0.5$

$W^{eo}_{1,x} = 0$ para el resto de x=1...nn

yk-1

$W^{eo}_{2,4} = 0.5$

$W^{eo}_{2,9} = 0.5$

$W^{eo}_{2,x} = 0$ para el resto de x=1...nn

suma(e):

$W_{15,5}^{eo} = 0.5$

$W_{15,10}^{eo} = 0.5$

$W_{15,x}^{eo} = 0$ para el resto de x=1...nn

Los de salida:

$$W_{11}^{os} = 2 \cdot \frac{\partial U_{pd}}{\partial r_k} = 2(Kp + Kd)$$

$$W_{61}^{os} = 2 \cdot \frac{\partial U_{pd}}{\partial r_k} = 2(Kp + Kd)$$

$$W_{21}^{os} = 2 \cdot \frac{\partial U_{pd}}{\partial r_{k-1}} = -2Kd$$

$$W_{71}^{os} = 2 \cdot \frac{\partial U_{pd}}{\partial r_{k-1}} = -2Kd$$

$$W_{31}^{os} = 2 \cdot \frac{\partial U_{pd}}{\partial y_k} = -2(Kp + Kd)$$

$$W_{81}^{os} = 2 \cdot \frac{\partial U_{pd}}{\partial y_k} = -2(Kp + Kd)$$

$$W_{41}^{os} = 2 \cdot \frac{\partial U_{pd}}{\partial y_{k-1}} = 2Kd$$

$$W_{91}^{os} = 2\cdot\frac{\partial U_{pd}}{\partial y_{k-1}} = 2\,Kd$$

$$W_{51}^{os} = \frac{\partial U_{pd}}{\partial \sum e} = 2\,Ki$$

$$W_{101}^{os} = \frac{\partial U_{pd}}{\partial \sum e} = 2\,Ki$$

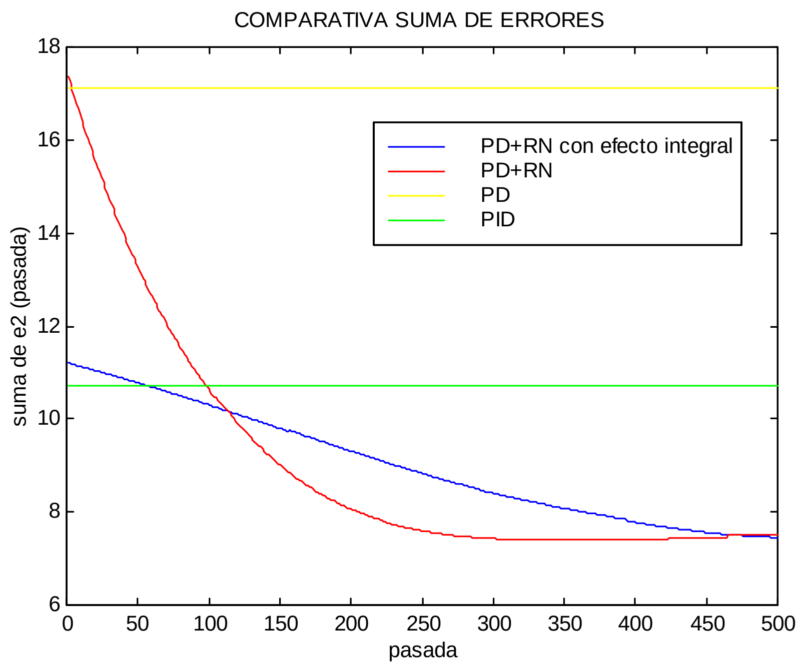


figura 78

En la figura 78 están representados los resultados obtenidos para las cuatro opciones :

- Esquema con PD óptimo .
- Esquema con RN inicializando los pesos según PD óptimo.
- Esquema con PID óptimo.

- Esquema con una RN con una entrada para efecto integral, inicializada según PID óptimo.

6.3.1 CONCLUSIONES:

Podemos conseguir que el sistema sea igual o mejor que un controlador PD o PID desde el momento en que comienza el aprendizaje.

Cuando intentamos mejorar el comportamiento inicial del sistema, sacrificamos velocidad y capacidad de aprendizaje.

Para ello, si deseo un aprendizaje rápido óptimo, renunciando al comportamiento inicial, se deberá seguir el método del apartado 4.2.2 , mejorándolo añadiendo el sistema de detención del aprendizaje en caso de empeoramiento del punto .