

JHS

# CAPITULO 3

## REDES NEURONALES Y CONTROL AUTOMATICO

---

### 3.1.- Introducción

En el control de procesos a través de redes neuronales no existe una forma y estructura única de control para el sistema, todo dependerá del tipo y de la información que se posea de dicho proceso. Lo clásico ha sido trabajar con la identificación del sistema o su inversa, partiendo de la base que una red neuronal es capaz de representar cualquier conjunto de pares entrada y salida (lineal o no linealmente separables). Esta identificación no es hecha en base de determinar los parámetros de la planta (aunque en algunos casos sí), sino que se basa en la imitación del comportamiento de la planta o su inversa.

Según esto, el problema del control estaría resuelto. Sin embargo, el lograr que una red aprenda a identificar un sistema dependerá de dos factores primordiales: uno es el generar un conjunto de entrenamiento adecuado que permita a la red emular en forma precisa el comportamiento del sistema, y el segundo es utilizar un algoritmo de aprendizaje que permita a la red generar un conjunto de pesos o conexiones que sirva para tal fin.

En el presente capítulo se expondrá una serie de técnicas para el control de procesos a través de la identificación de sistemas por medio de redes neuronales, basados en el esquema y aplicaciones introducidos por Paul J. Werbos [6]. Hay que tener en cuenta que existen numerosos otros métodos para el control por medio de redes neuronales, pero se ha querido presentar de esta forma debido que a los conocedores del control clásico les resultará más natural su comprensión.



### **3.2.- Redes neuronales para el control**

En el control automático, numerosos tipos de redes pueden ser usadas como controladores, pero hay algunas que por su misma estructura resultan más adecuadas en una gran cantidad de aplicaciones. Entre las más populares se encuentra la red de retropropagación cuya característica de propagar el error a través de la red resulta fundamental para la obtención de las medidas de error en dichos controladores. Tal característica se comprenderá en forma más clara a medida que se expliquen las distintas aplicaciones a lo largo del desarrollo de este capítulo.

Como se mencionó anteriormente, las técnicas que se mostrarán a continuación están basadas en la identificación de sistemas, donde la red de retropropagación se ajusta plenamente a sus requerimientos, por lo que ahora en adelante se supondrá que cuando se hable de una red se referirá a ésta.

### **3.3.- Control supervisado**

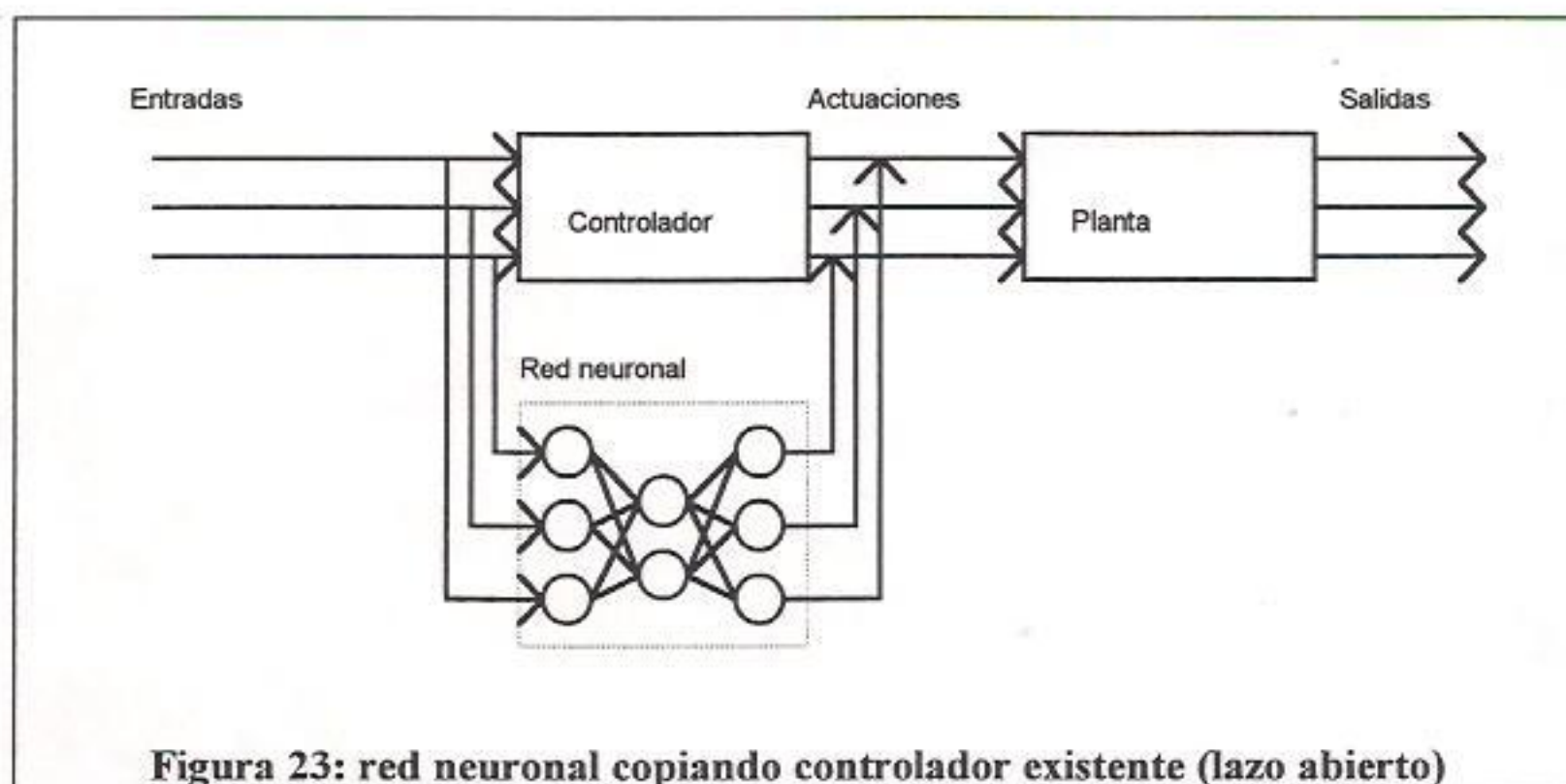
En el control supervisado, una red neuronal artificial aprende a imitar un controlador existente, el cual ya sabe como realizar la tarea de control. Si ya existe un controlador para el proceso ¿para qué tener este controlador neuronal?. Según Widrow y Smith (1964) existen dos respuestas para esto: el controlador al cual se copia su funcionamiento puede ser un controlador impracticable, como por ejemplo una persona; por otro lado, la red neuronal puede ser capaz de lograr una efectiva regla de control basado en la representación del estado del sistema que es más fácil de medir que la representación requerida por el controlador existente.

Este tipo de controlador puede ser clasificado de dos formas: control en lazo abierto y control en lazo cerrado. En ambos casos, la idea principal se centra en que si existe un controlador capaz de controlar la planta, entonces la información requerida para entrenar una red neuronal puede ser obtenida a partir de éste.

- **Control supervisado en lazo abierto**

En el caso de un controlador neuronal en lazo abierto (figura 23), al igual que un controlador clásico, se actuará sobre la planta sin ver lo que está sucediendo a la salida, o sea, sin una señal de error que le diga como está llevando a cabo su labor. Por estos motivos, sus aplicaciones estarán limitadas a aquellos procesos en los cuales no existan perturbaciones o que estas sean insignificantes.

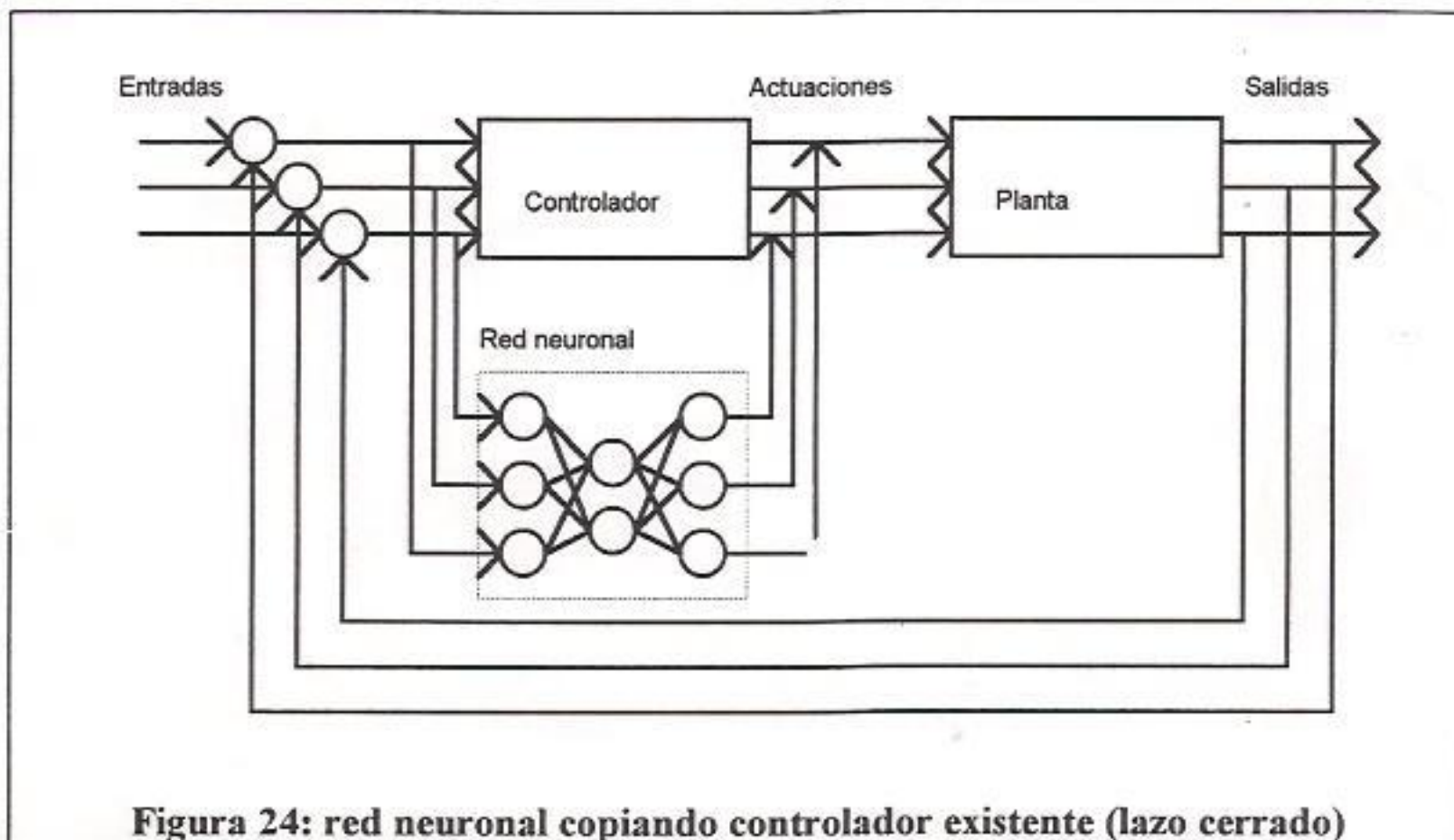
Tal como es de suponerse, en este caso las entradas de la red corresponderán a la salida deseada y su salida será equivalente a la actuación del sistema, por lo que su entrenamiento consistirá en la presentación de estos pares entrada/salida con lo cuales la red generará una representación del controlador y será capaz de emular su comportamiento.





- **Control supervisado en lazo cerrado**

Para un control supervisado en lazo cerrado (figura 24) las entradas de la red corresponderán a la señal de error entre las salidas deseadas y las reales. Asimismo, las salidas de la red serán equivalentes a la actuación o salida del controlador. En comparación con el caso anterior este tipo de controlador puede ser mucho más efectivo en el caso que en el entrenamiento se considere todas las combinaciones posibles de entrada/salida que puedan existir, incluyendo señales de error con todas las perturbaciones que pueda sufrir la planta o proceso.



**Figura 24: red neuronal copiando controlador existente (lazo cerrado)**

### 3.4.- Control inverso directo

Otra forma simple de controlar procesos mediante redes neuronales consiste en generar el controlador con el modelo inverso de la planta. Este método fue propuesto por Widrow, McCool y Medoff en 1979 [6] como control adaptivo para sistemas lineales. También, según lo presentado por Srinivasan, Barto y Ydstie en 1988 [6], éste puede ser aplicado a sistemas no lineales usando una red de retropropagación (backpropagation) para generar la inversa. Por otro lado, en robótica ha sido sumamente utilizado este método para determinar la cinemática inversa (Grossberg y Kuperstein en 1986) [6].

En este caso, dicha estimación de parámetros no consiste en encontrar la ecuación matemática que modele el comportamiento del sistema. Más bien, consiste en imitar el comportamiento del sistema; como poseemos los valores de entrada y salida del sistema podemos utilizarlos para entrenar una red neuronal cuyos parámetros de entrenamiento serían equivalentes a éstos.

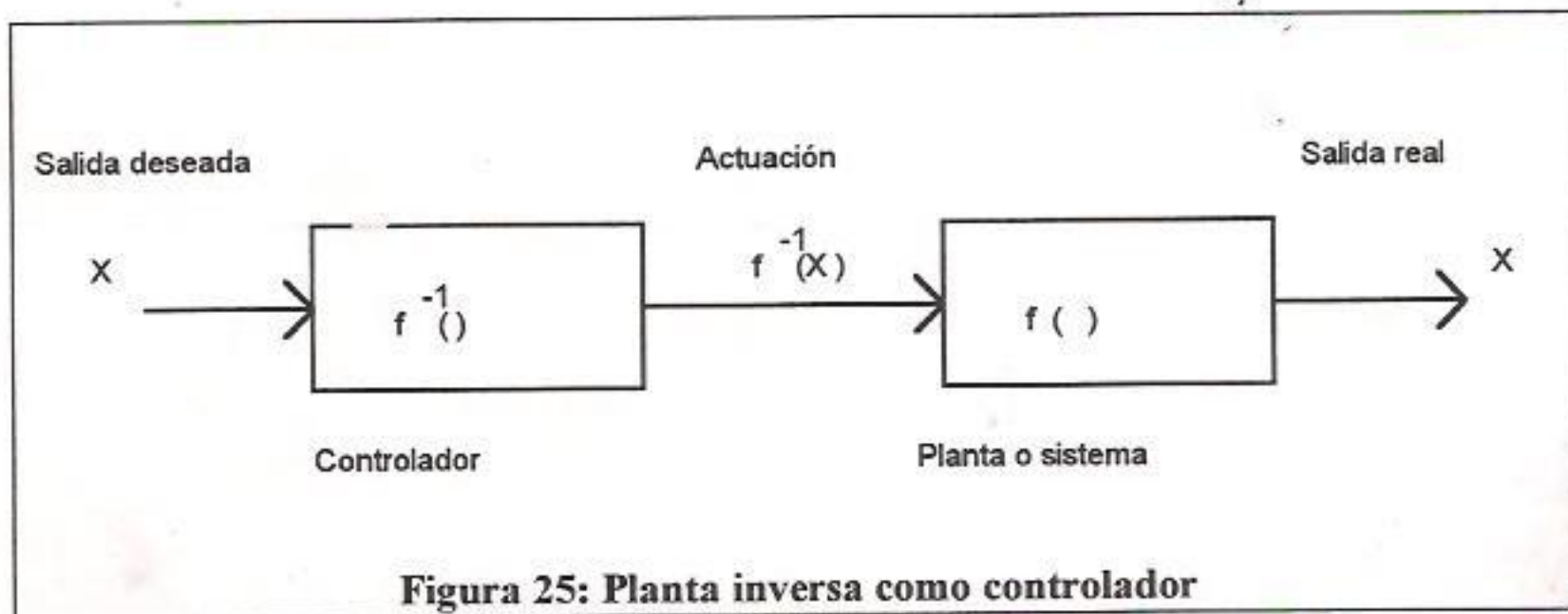
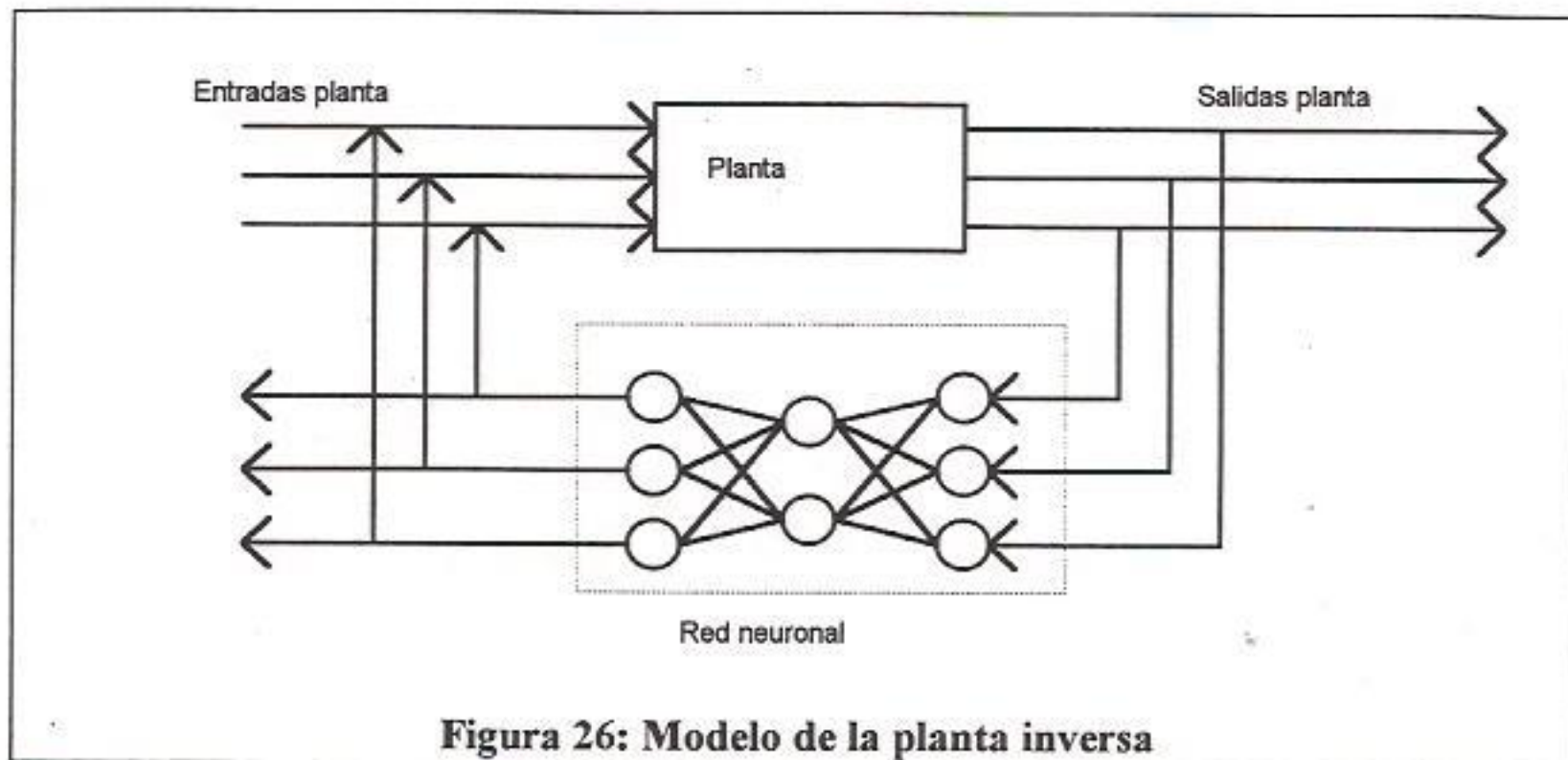


Figura 25: Planta inversa como controlador

Como se presenta en la figura 25, si conocemos completamente los parámetros de la planta,  $f()$  podemos generar un controlador en lazo abierto con su inversa ( si es posible),  $f^{-1}()$ , y así obtener la salida deseada que se manifiesta según la identidad  $f(f^{-1}(X))=X$ .



Como queremos generar la inversa de la planta, las entradas de la planta serán las salidas deseadas para la red y las salidas de la planta con dichas entradas corresponderán a la entrada de la red. Según este esquema, presentado en la figura 25, el entrenamiento de la red tendría que ser fuera de línea, es decir, la red tendría que ser entrenada en forma previa a ser usada como controlador como se presenta en la figura 26.



**Figura 26: Modelo de la planta inversa**

Como es de suponerse, este método en la realidad no sucede en forma tan simple ni tan exacta. El modelo aquí presentado no contempla que el sistema pueda sufrir perturbaciones, ni se pueda realimentar para corregir el error. El mayor problema con la identificación de la inversa se produce cuando varias entradas a la planta producen una misma salida, es decir, cuando la inversa de la planta no está bien definida.

### 3.5.- Controladores adaptivos

El control adaptivo paramétrico es el problema de controlar la salida de un sistema con estructura conocida, pero con parámetros desconocidos. Para hacer el problema analíticamente tratable, en la teoría de sistemas adaptivos la planta a ser controlada se asume lineal e invariante en el tiempo y con parámetros desconocidos. Dichos parámetros pueden ser considerados como elementos de un vector 'p'. Si 'p' es conocido, el vector de parámetros 'c' de un controlador puede ser elegido como 'c\*' tal que la planta junto con el controlador se comporte como un modelo de referencia descrito por una ecuación de diferencias (o diferencial) lineal con coeficientes constantes. Si 'p' es desconocido, el vector 'c(t)' tiene que ser ajustado en línea usando toda la información posible concerniente al sistema.

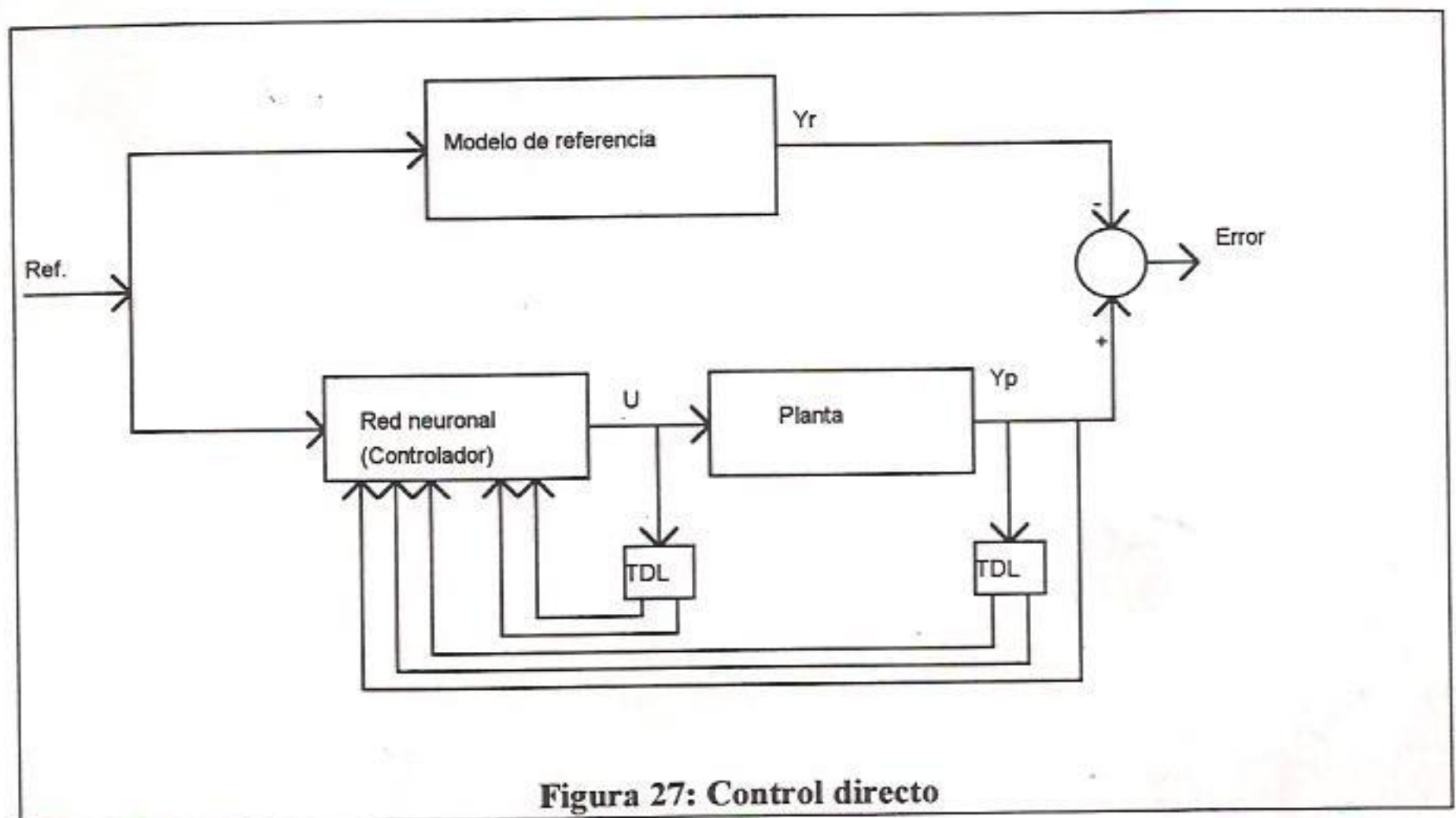
Existen dos tipos de control adaptivo neuronal para una planta desconocida: control directo y el control indirecto.

- **Control directo**

Este tipo de controlador es aplicado cuando la planta es no lineal y dinámica, es decir, cuando los valores presentes de las salidas dependen de los valores pasados de las entradas y salidas. Por esto, la idea principal del funcionamiento de este controlador es que modifique sus pesos según algún tipo de error o desempeño del controlador a partir de la información que se pueda obtener del sistema, para lo cual utiliza los valores pasados de las actuaciones y salidas de la planta.

La figura 27 presenta el esquema básico de este tipo de controlador, donde los valores pasados se manifiestan a través de retardos (TDL) de las señales en cuestión.





**Figura 27: Control directo**

En el control directo, los parámetros son ajustados directamente para reducir alguna norma del error entre el sistema real y un modelo de referencia. El problema de estos controladores radica en que dicho error no es el error propiamente del controlador, sino que de la combinación entre el controlador y la planta. Por lo tanto no existe ninguna norma directa de error para el ajuste de los pesos en el controlador.

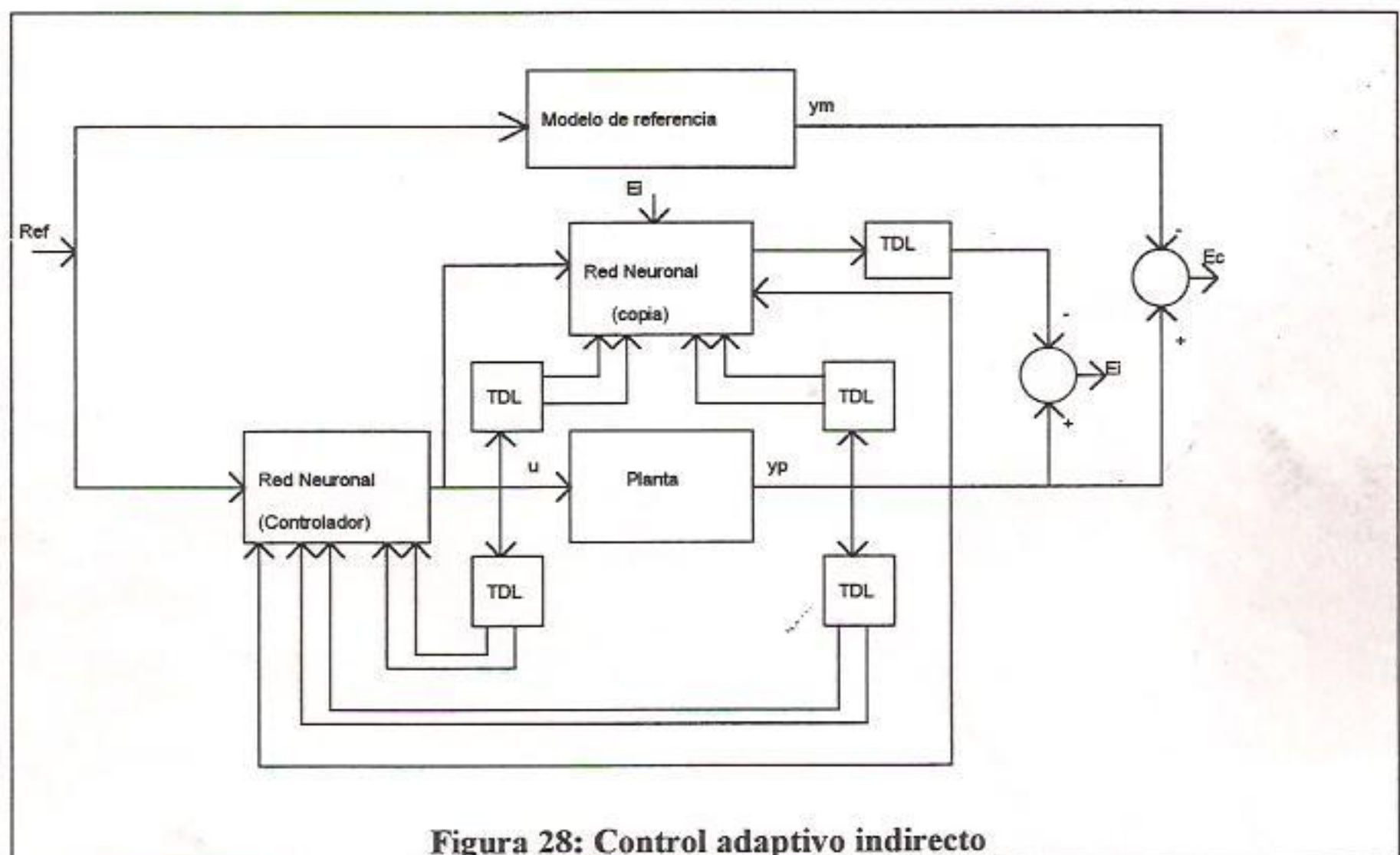
En la actualidad este es un problema en estudio, en donde muchos investigadores buscan encontrar alguna forma para la corrección de los pesos a partir de los datos existentes; entre estos investigadores se encuentra Narendra quien últimamente ha trabajado en algunos tópicos de neurocontrol.



- **Control indirecto**

Este tipo de control (ver figura 28) está formado por dos redes de neuronas dependientes. La primera red actuará como controlador propagando la salida deseada a través de ella generando así la actuación a la planta. Dicha actuación es utilizada en forma paralela como entrada a una segunda red neuronal que copia el comportamiento de la planta. La diferencia entre la salida de la planta real y de esta última red es la medida del error usada de alguna forma adaptando los pesos de ambas redes con el fin de que esta diferencia tienda a cero.

Se llama control indirecto debido a que la señal de error no corresponde a la diferencia con el modelo de referencia sino con respecto a la red que imita la planta. La siguiente figura presenta el esquema general de este tipo de controlador:



**Figura 28: Control adaptivo indirecto**

Las entradas y salidas retardadas de la planta forman las entradas para la red neuronal controladora la cual genera la señal de retroalimentación para el control de la



planta. Los parámetros de la red neuronal de copia son ajustados por la retropropagación del error  $E_i$  a través del modelo de identificación de la planta (red neuronal de copia).

Otro esquema planteado por Widrow y Nguyen [7], basados en el control indirecto, consiste en usar una red de dos capas que aprenda en línea el comportamiento de la planta, para luego ser usada para obtener el error equivalente en su entrada y propagarlo a la red de control que se ubica en cascada con la anterior. La figura 29 presenta este esquema que es conocido con el nombre de aprendizaje general (general learning).

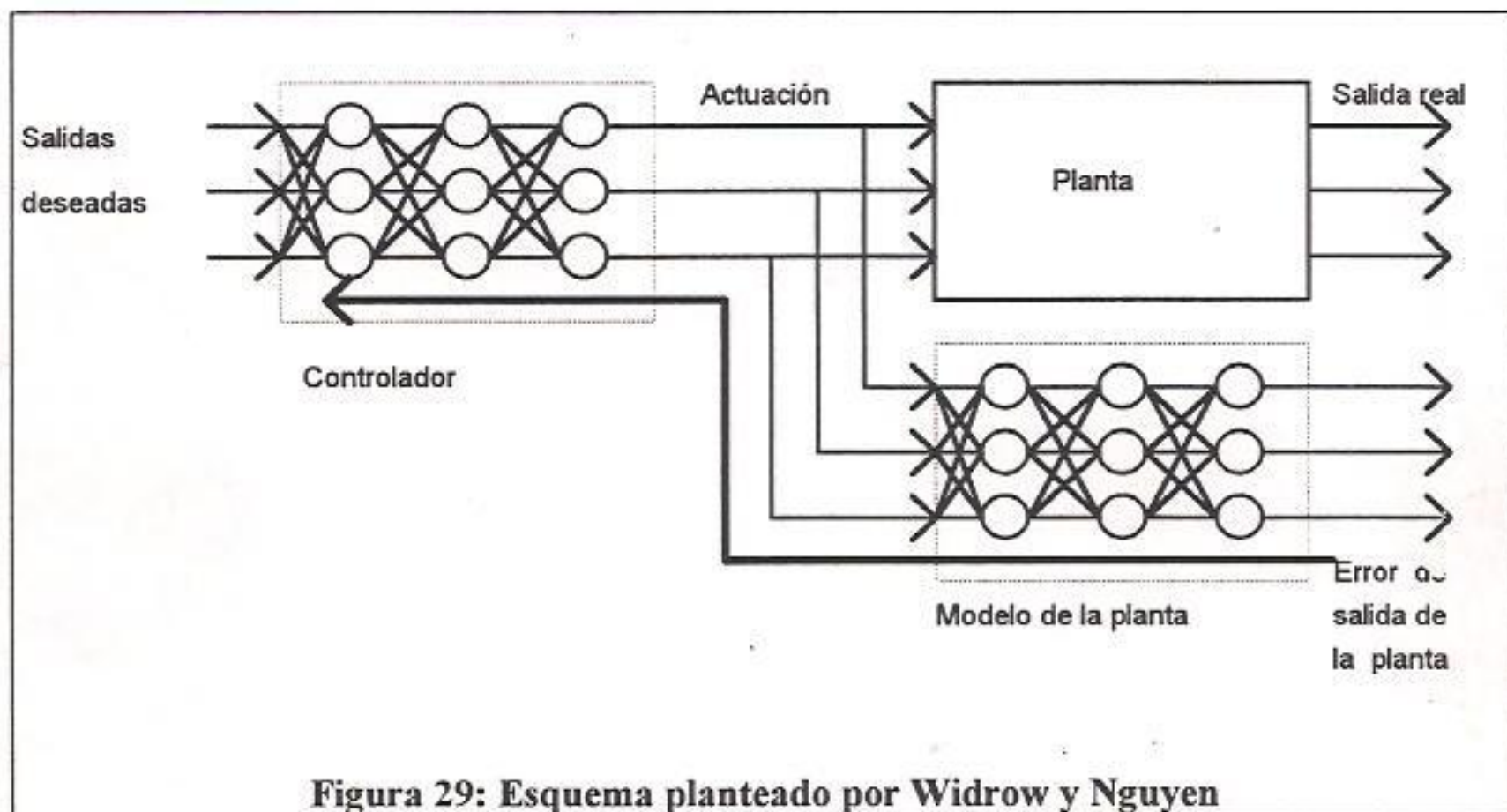
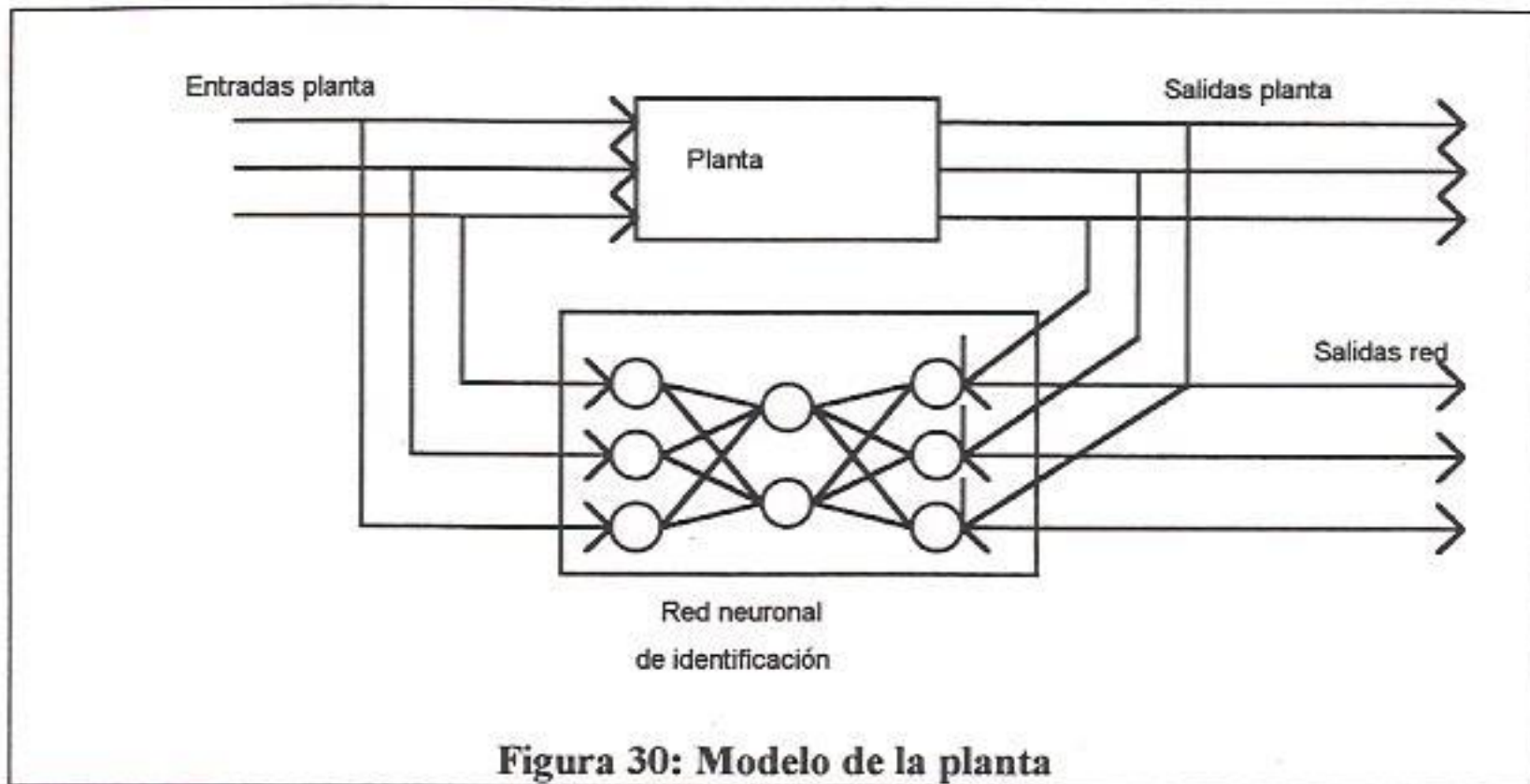


Figura 29: Esquema planteado por Widrow y Nguyen

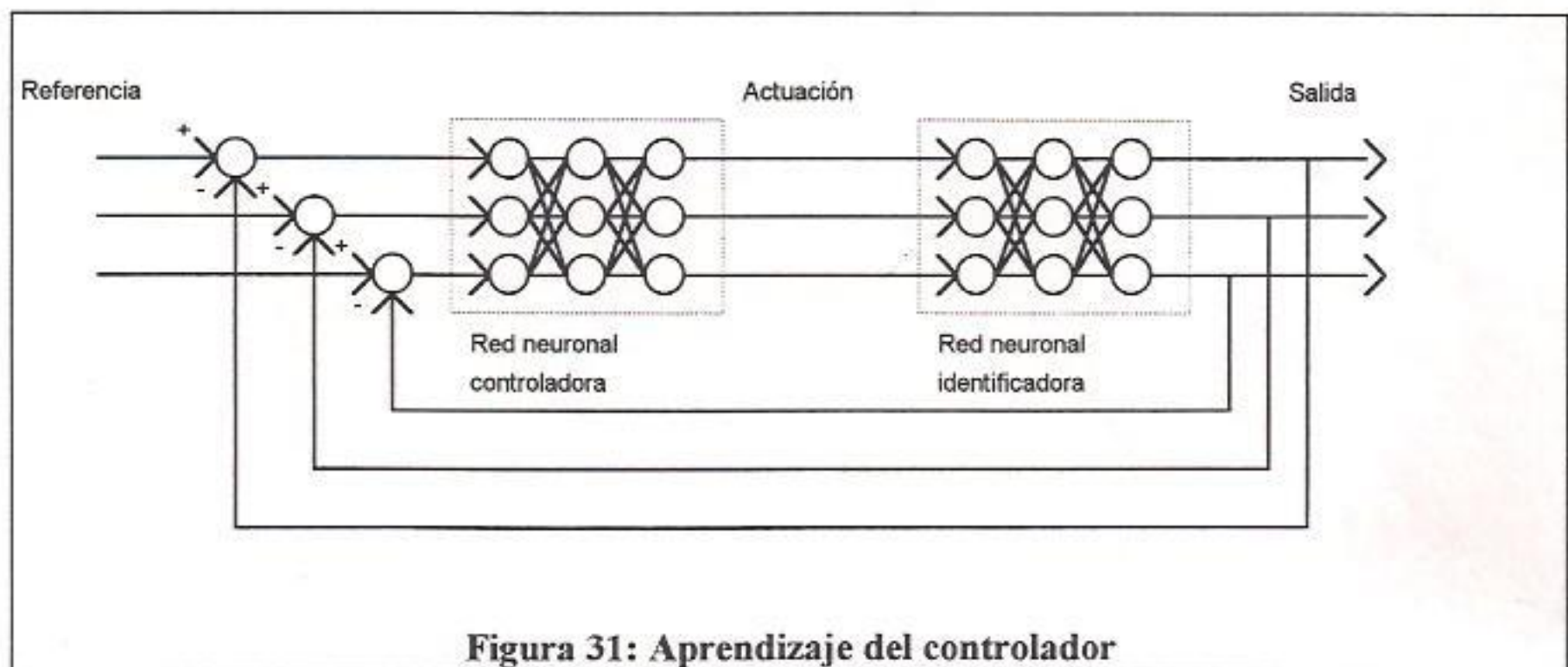
Este esquema, basado en una red de retropropagación, funciona en una primera etapa fuera de línea en donde parte de la red identifica el funcionamiento de la planta. La señal de actuación es usada como patrón de entrada para la red neuronal cuyo valor deseado será la salida de la planta. La diferencia entre ambas salidas (de la red y de la planta), se usa como señal de error para la adaptación de los pesos (ver figura 30). Una vez que este error ha alcanzado niveles aceptables, tendientes a cero, se cancela el aprendizaje de la identificación de la planta.





**Figura 30: Modelo de la planta**

Para entrenar el controlador se coloca éste en cascada con la planta simulada, en forma similar a un controlador y una planta real. La señal de referencia a usar es la misma usada en la etapa anterior. El patrón de entrada para la red es el error entre la salida del sistema ( $Y_s$ ) y la señal de referencia, y el patrón de salida es la señal de referencia. Entonces, al llevarse a cabo la adaptación de los pesos el error se propagará a través de la red de identificación del sistema sin modificar sus pesos, con lo que se obtendrá un error equivalente en la actuación con el cual se modificarán los pesos de la red controladora (ver figura 31).



**Figura 31: Aprendizaje del controlador**

Después de un determinado número de iteraciones, en que el error ha disminuido hasta un umbral aceptable, se conecta la red controladora a la planta real en forma análoga al esquema de la figura 31 pero en vez de la red neuronal identificadora se conecta la planta .

Este esquema posee algunas desventajas que deben ser consideradas:

- ( i) El controlador no es operacional mientras se realiza el aprendizaje.
- ( ii) La planta es asumida como un proceso físico estático.

Algunos autores, con el fin de mejorar estas deficiencias plantean que el aprendizaje del controlador sea llevado a cabo en línea con la planta real. Para esto colocan en paralelo la red identificadora (ver figura 26) realizando el aprendizaje con el error proveniente de la diferencia entre la salida de la planta real y la salida de la red identificadora (cuyos pesos permanecen fijos), propagándolo hasta el controlador para que modifique sus pesos. Como es de suponer, la red neuronal controladora no puede poseer cualquier valor en sus pesos; por lo general éstos son fijados según el esquema presentado en la figura 31, u obtenido como controlador inverso (figura 26).



### 3.6.- Retropropagación en el tiempo (BTT)

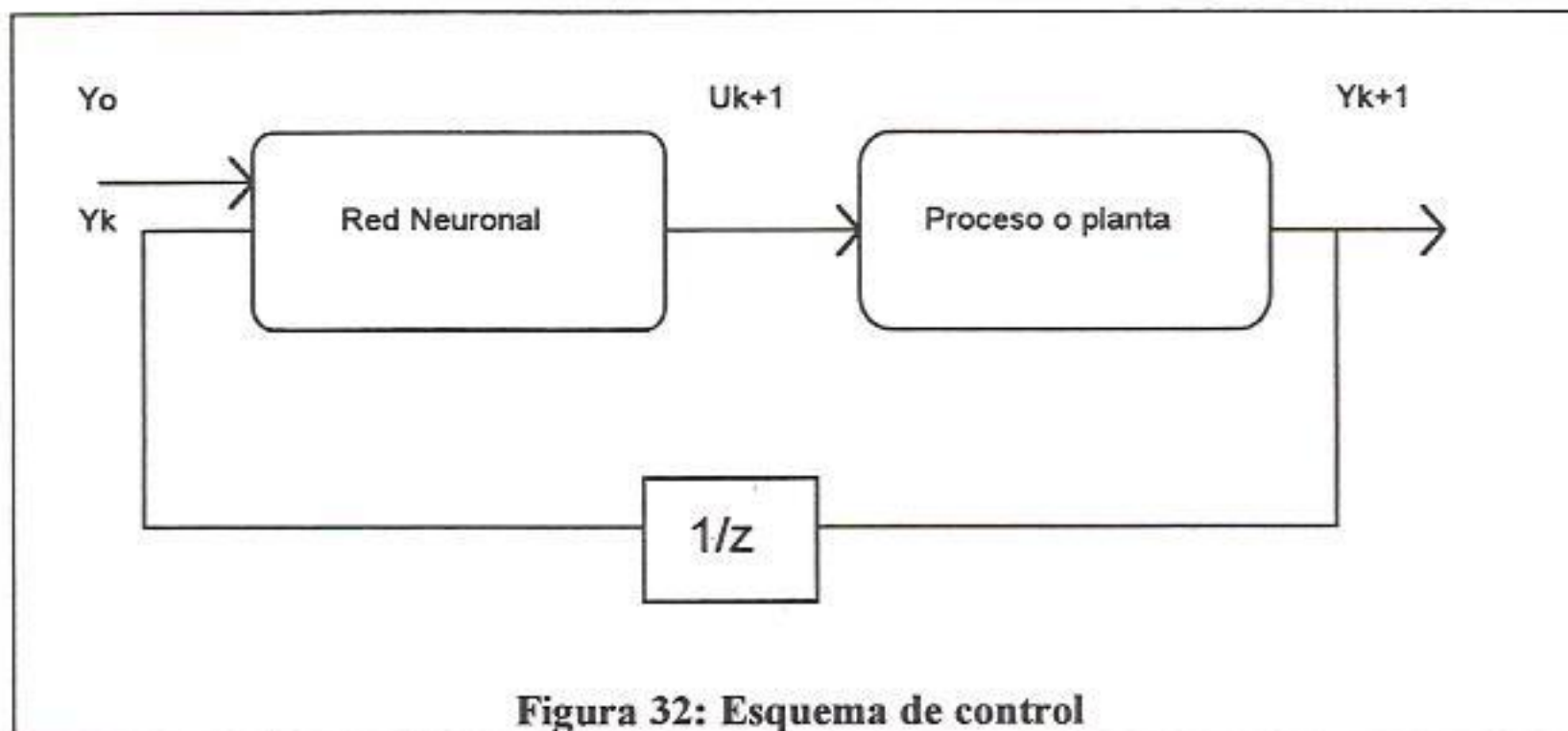
La retropropagación en el tiempo (BTT: Backpropagation through time), tal como lo dice su nombre, es un algoritmo basado en la retropropagación del error bajo un esquema de recurrencia, es decir, un esquema donde las salidas de la red son retroalimentadas a las entradas.

Su principal aplicación es en torno a problemas de optimización en donde el usuario o diseñador puede tomar cualquier función de utilidad, medida de su labor, o función de costo que maximizar o minimizar. Además, el método está enfocado al impacto de las acciones presentes sobre una utilidad futura. Dichas aplicaciones muchas veces no han sido hechas acorde las formulaciones y técnicas de la teoría del control estándar y no queda tan claro el como integrarlas a ellas.

Para la explicación de este algoritmo nos basaremos en la implementación realizada por Widrow y Nguyen [7] del camión con acoplado. Hay que tener claro que dicha forma de implementar este algoritmo no es la única y que existen muchos otros métodos más poderosos que éste. Sin embargo, por su simplicidad sirve de introducción para la implementación de este método.

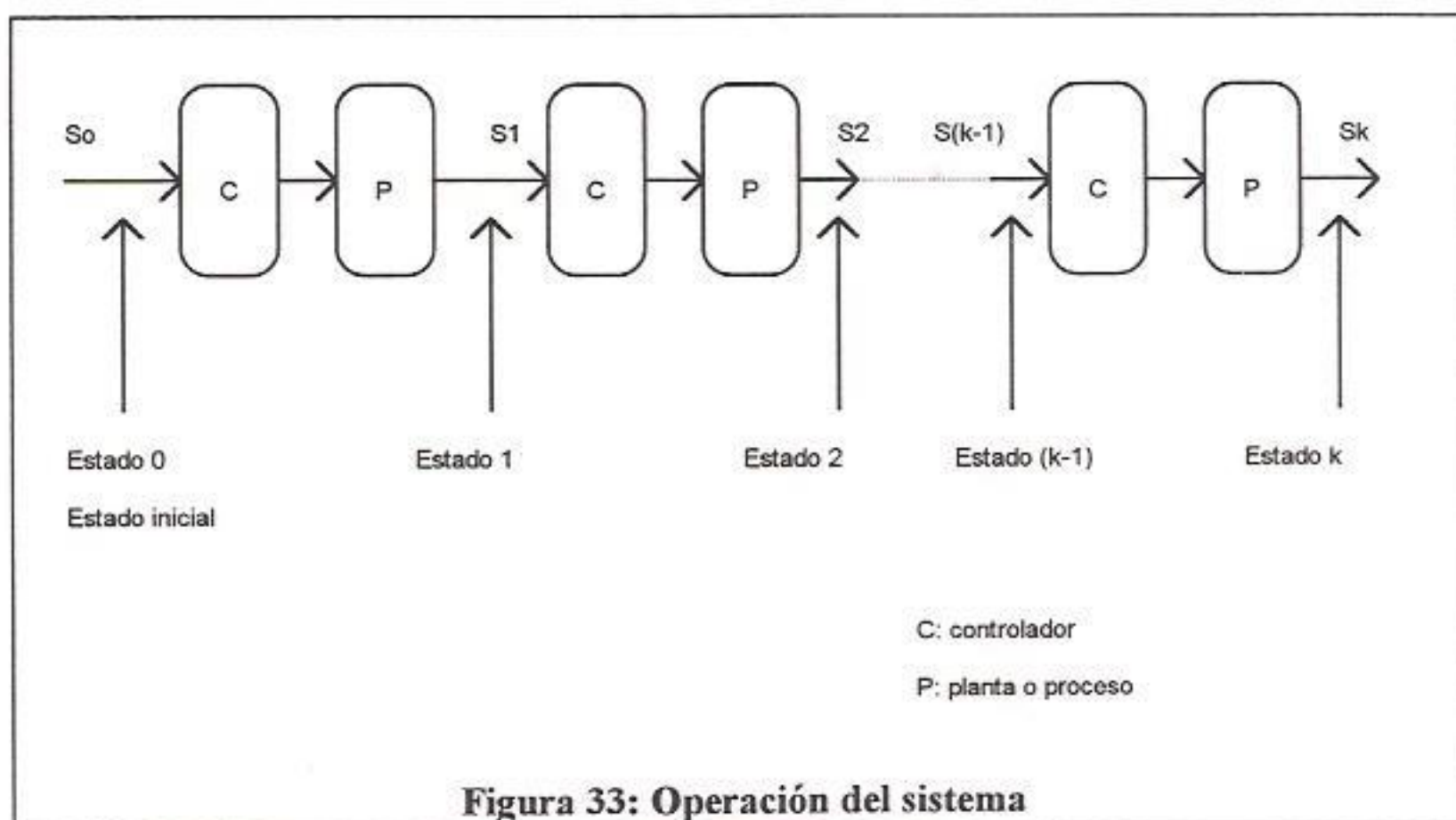
- **Esquema de control**

El esquema de control está basado en una red neuronal cuyas entradas corresponden a las mismas salidas del proceso retardadas en un período. La figura 32 presenta este esquema, en el cual la recurrencia se manifiesta a través de la realimentación de la salida en la entrada de la red y del proceso, lo que es equivalente al retardo de dichas señales en un período.



- **Operación del sistema**

Se explicará la forma de operación del sistema debido a que su funcionamiento debe ser entendido en forma previa al entrenamiento ya que al igual que los otros algoritmos la operación forma parte del entrenamiento, con la diferencia de que en este caso el esquema de operación del sistema es distinto al descrito para los anteriores.

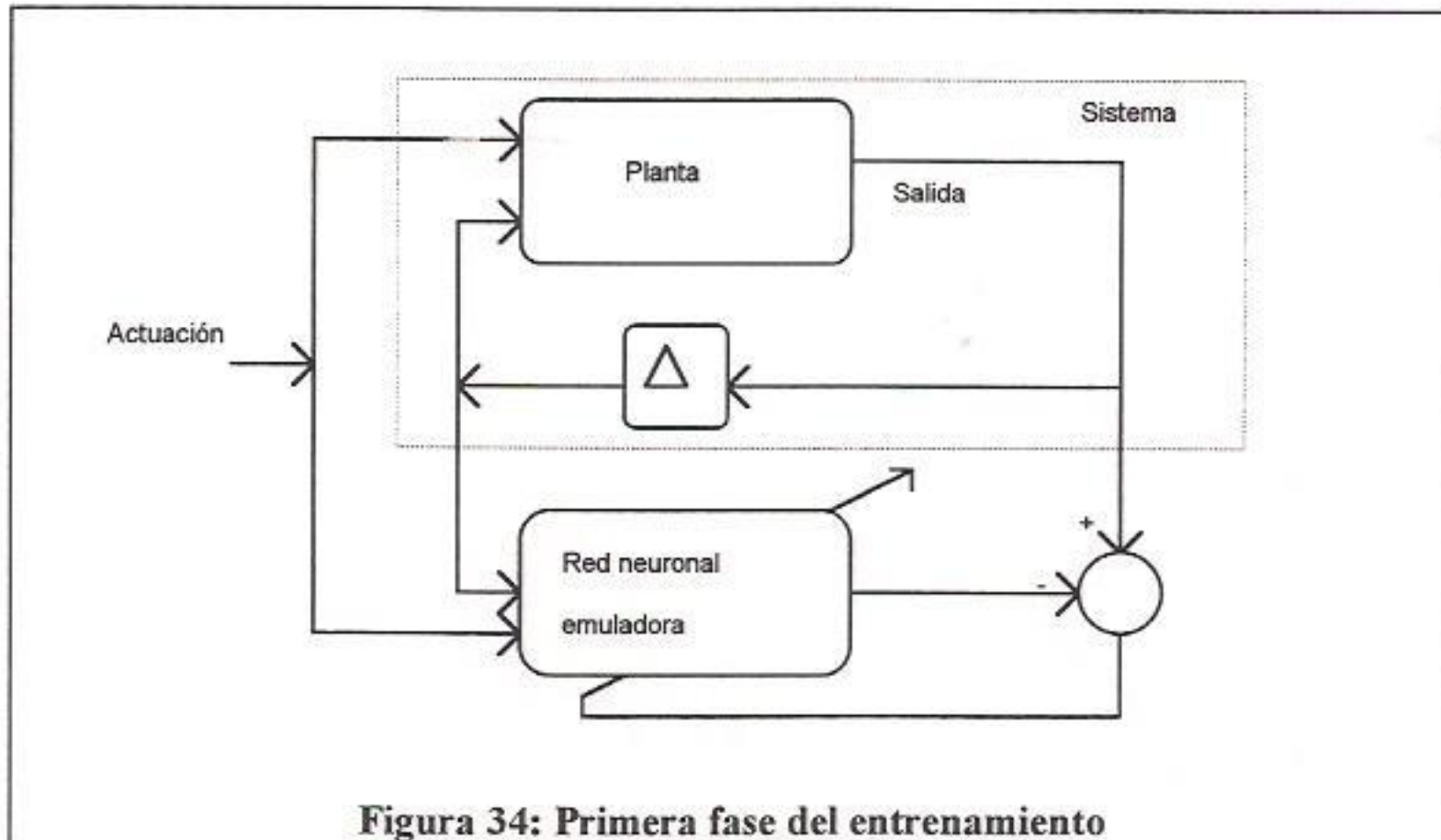




Según el esquema de la figura 33 los bloques etiquetados con la letra C representan al controlador. Los bloques etiquetados con la letra P representan al proceso. Durante la operación del sistema los pesos del controlador permanecen fijos. El vector de estado inicial  $S_0$  produce una señal de actuación sobre el proceso que lo lleva al estado  $S_1$ ; este estado es la próxima entrada al controlador, la que producirá un nuevo ciclo que lleva al sistema al estado  $S_2$ . El proceso continúa hasta que por algún criterio el proceso se detiene o llega a su fin, obteniéndose un estado final  $S_k$  el que es comparado con el estado final deseado generándose un error si existiera diferencia.

- **Entrenamiento**

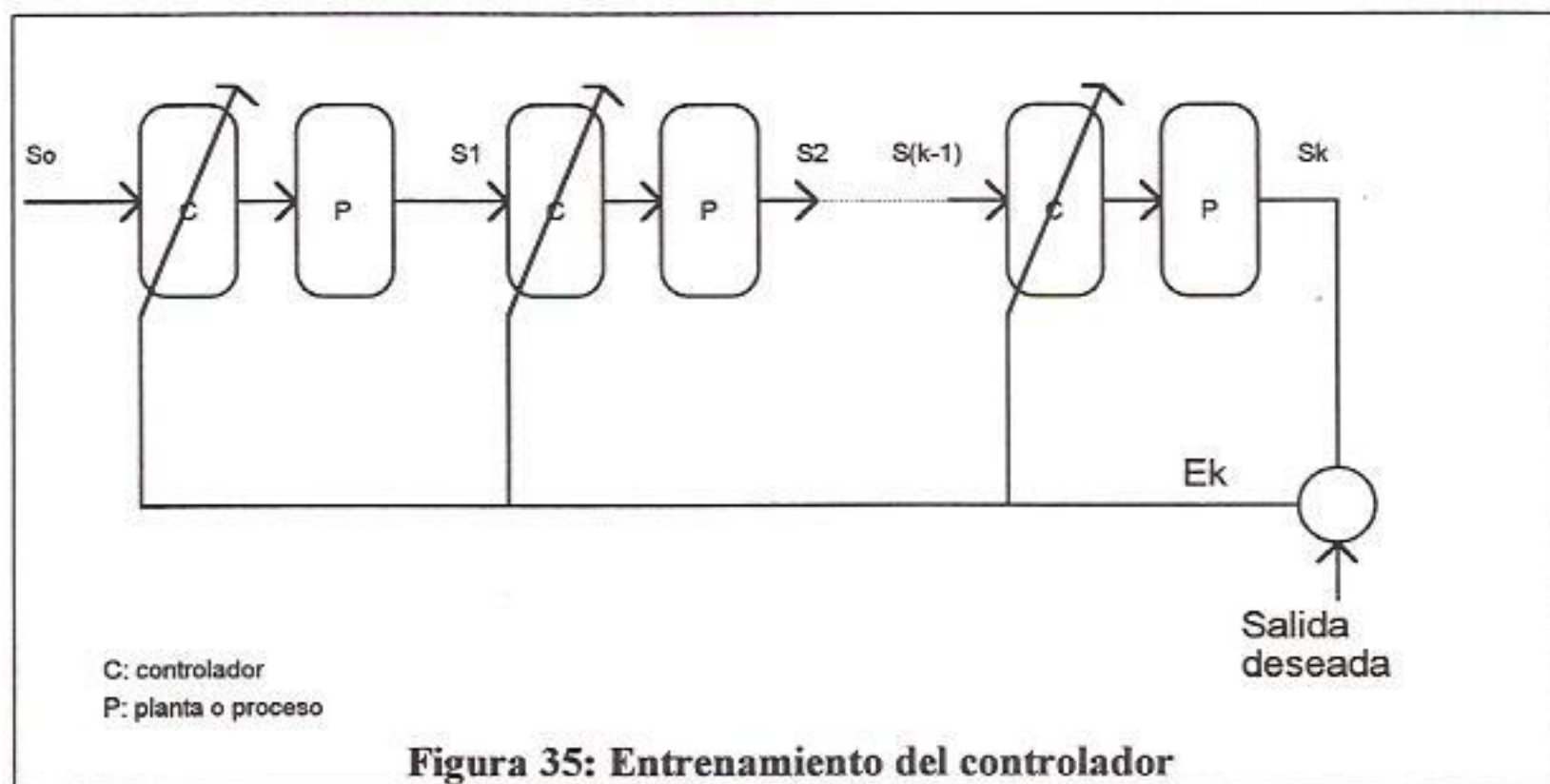
En este algoritmo el entrenamiento ha sido dividido en dos fases: en la primera una red neuronal es entrenada para aprender el comportamiento de la planta o proceso en cuestión, y es llevado a cabo en forma análoga al esquema usado en el control indirecto (ver figura 29).



**Figura 34: Primera fase del entrenamiento**

En esta primera fase, la red neuronal es entrenada con las señales de entrada de la planta o proceso, las que son equivalentes a las señales de actuación teniendo como valores deseados la salida de la planta con dicha actuación (ver figura 34).

Una vez concluido este proceso, en el cual la red neuronal es capaz de emular en forma precisa el proceso y conoce su comportamiento, se pasa a la segunda fase donde dicha red es usada como planta para el entrenamiento del controlador neuronal. Como lo muestra la figura 35, el controlador es puesto en cascada con la red emuladora realimentando su salida con la entrada del controlador.



El controlador junto con la red emuladora son puestos en operación de la forma descrita anteriormente. Después de cada ciclo se genera un error  $E_k$ , tal como se muestra en la figura 35. Hay que tener claro que los pesos del controlador durante todo el ciclo permanecen constantes; son modificados sólo una vez terminado el ciclo.

La forma de adaptar los pesos puede ser hecha de variadas formas pero todas son realizadas con el algoritmo de retropropagación. La forma que se presentará a continuación está basada en el control adaptivo indirecto y se ha dividido en los siguientes pasos:

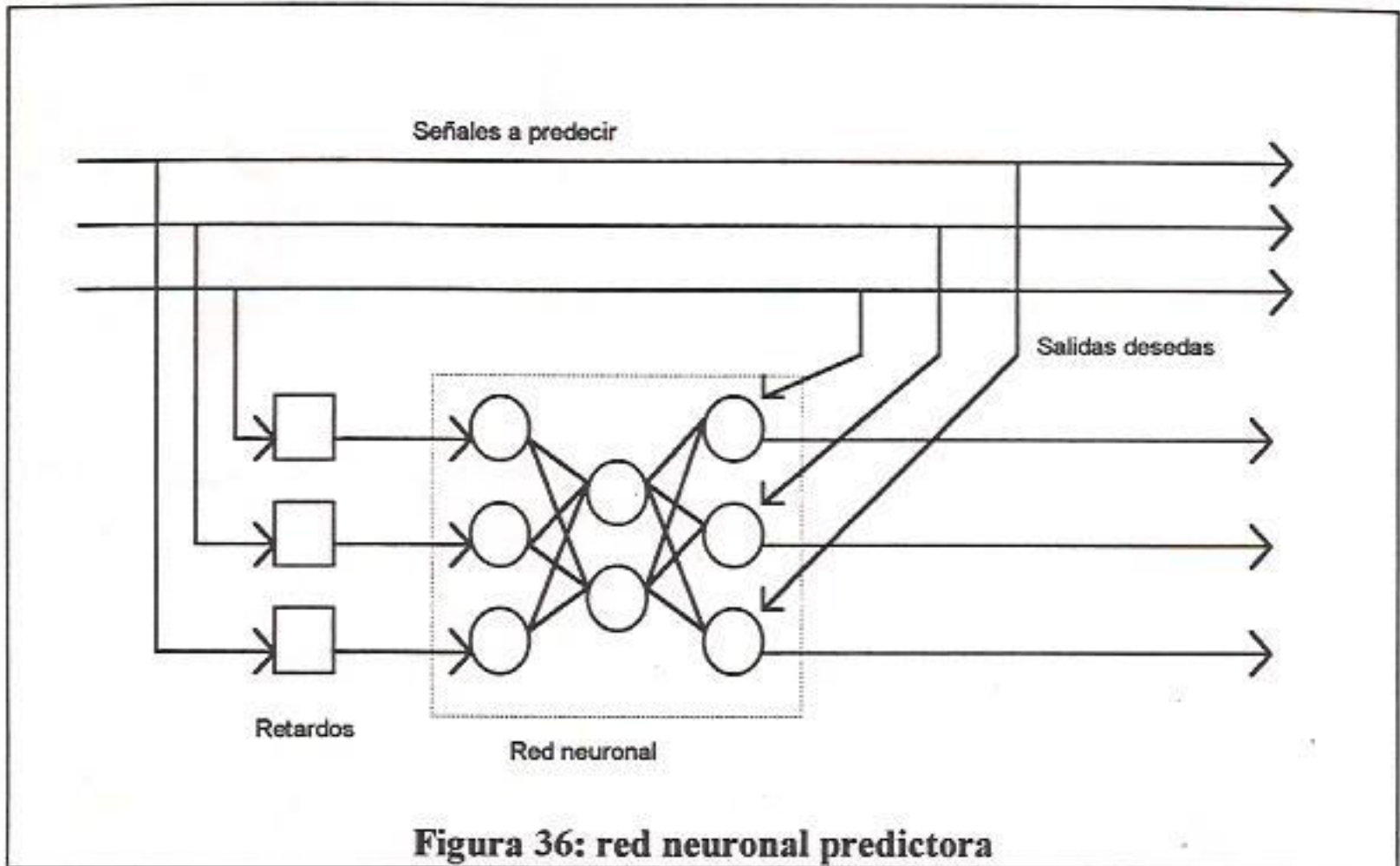


- ( i) Manteniendo los pesos fijos y asignando un estado inicial al controlador realizar un ciclo de operación del sistema y obtener el error  $E_k$  con respecto a la salida deseada.
- ( ii) Modificar los pesos según el esquema de control adaptivo indirecto propagando el error de la salida a través del emulador, obteniendo de esta forma el error del controlador. Lo que se está haciendo aquí corresponderá a la corrección del error del último estado  $S_k$ .
- (iii) Lo que sigue a continuación es la corrección de los pesos de los pasos anteriores. Para esto debemos obtener el error a través de su propagación del estado posterior, es decir propagar el error  $E_k$  a través del emulador y del controlador para obtener  $E(k-1)$ .
- ( iv) Con el valor de  $E(k-1)$  volvemos a ( ii) y repetimos hasta llegar al estado inicial donde volvemos a repetir todos estos pasos hasta que el error  $E_k$  se haga cero o quede dentro de márgenes aceptables.

- **Red predictiva**

La figura 36 ilustra la idea básica de como entrenar una red neuronal como predictor. En este esquema las líneas superiores mostradas son las que llevan las señales cuyos valores se desean predecir. Para propósitos de entrenamiento, las entradas de la red corresponden a dichas señales retardadas en ciertos períodos y las salidas de la red corresponden al valor actual de la red. Entonces, si la red es entrenada utilizando los valores pasados como entradas y dejando como valor deseado el valor actual, la red estará siendo entrenada para predecir valores futuros. Algunos autores que han trabajado en esto han sido Goodwin y Sin en 1984 y Widrow y Stearns en 1985.

Este tipo de identificación es muy usada en los casos que se requiere representar sistemas en los cuales los valores presentes de la salida dependen de los valores pasados. En el caso de la aplicación del camión con acoplado, presentado por Widrow y Nguyen [9], este tipo de red fue utilizada para emular el comportamiento del camión y entrenar así al controlador.



**Figura 36: red neuronal predictora**