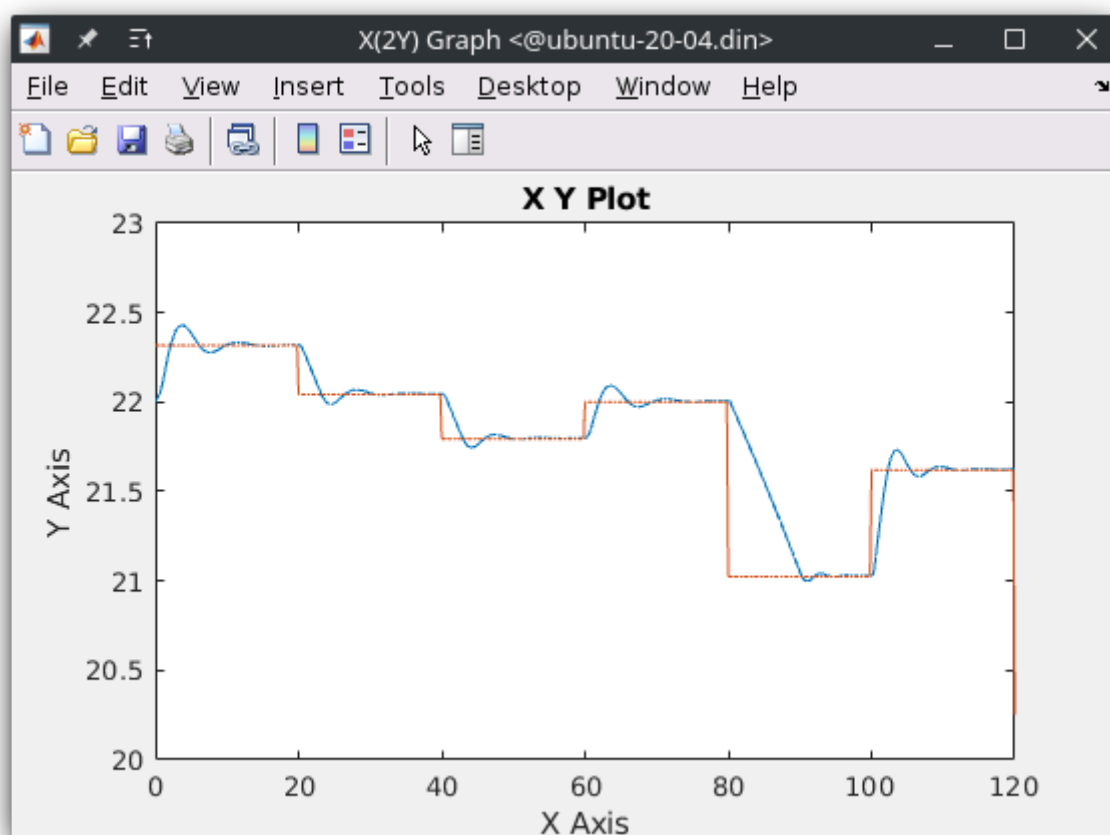


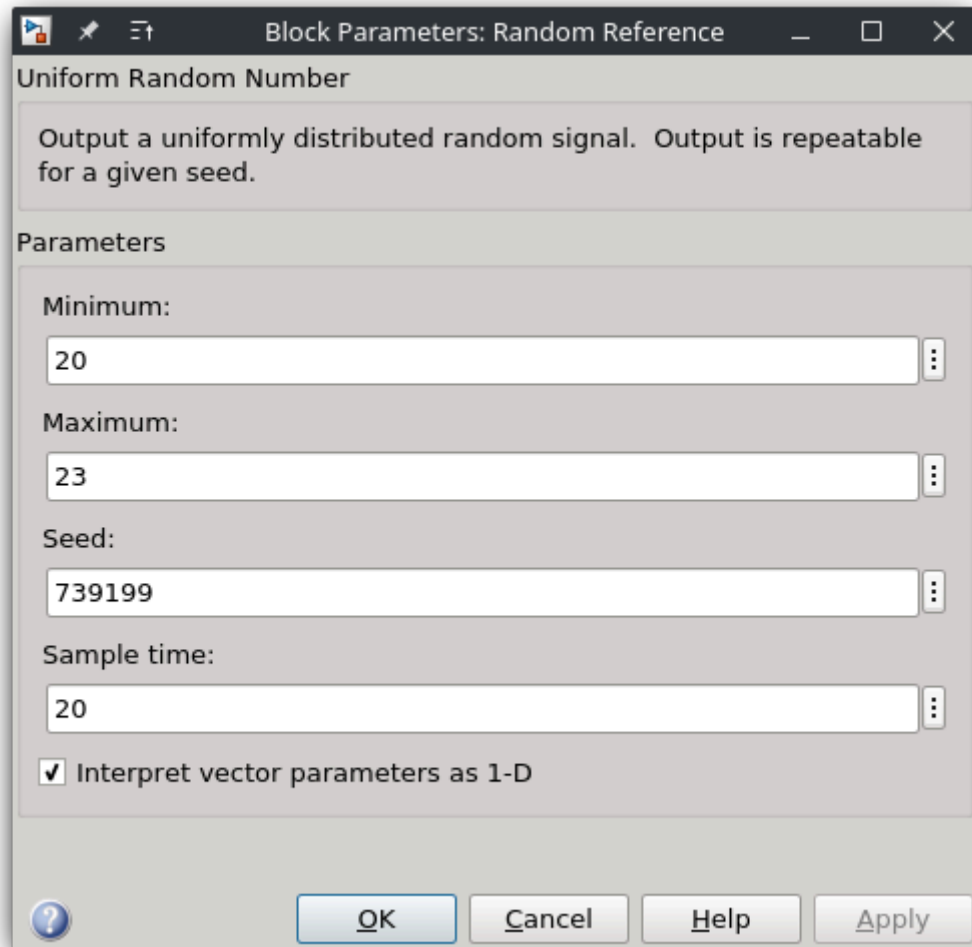
Control predictivo neuronal del reactor

En este caso de estudio aparece un controlador predictivo neuronal que utiliza un modelo de red neuronal de una planta para predecir su comportamiento futuro. El controlador, empleando el modelo, calcula el valor de entrada óptimo para un periodo futuro de tiempo determinado.

En la siguiente figura se observa el comportamiento inicial del sistema por defecto, sin alterar nada, durante 120 segundos. El eje X representa el tiempo, el eje Y el valor de la variable de salida, la línea roja es el valor de referencia deseado y la línea azul es el valor real de la salida.



Para obtener resultados comparables en el futuro se ha sustituido el valor de la semilla para la generación aleatoria del valor de referencia a 739199 (equivalente al valor de `round(now)` en el momento de la ejecución), de esta forma se desacopla el valor de la semilla del tiempo actual.



Identificación del sistema

El primer paso en el control predictivo neuronal es identificar el sistema, es decir, determinar el modelo de red neuronal. Para entrenar esta red neuronal se usan entradas y salidas de la planta generadas con anterioridad para predecir los futuros valores de salida de la planta.

Inicialmente, se mantiene la estructura de la red neuronal y se genera un conjunto de 100 datos para entrenar la red neuronal, desechando los pesos anteriormente generados. Se mantendrán los valores máximos y mínimos de entradas y salidas indicados por defecto en todo momento.

Tras entrenar la red neuronal se puede observar que la salida de la red neuronal es muy dispar a la salida real de la planta en el conjunto de datos de validación, por lo que se puede deducir que el comportamiento de esta red neuronal no será adecuado.

Plant Identification <@ubuntu-20-04.din>

File Window Help

Plant Identification

Network Architecture

Size of Hidden Layer: 7

Sampling Interval (sec): 0.2

No. Delayed Plant Inputs: 2

No. Delayed Plant Outputs: 2

☐ Normalize Training Data

Training Data

Training Samples: 100

Maximum Plant Input: 4

Minimum Plant Input: 0

Maximum Interval Value (sec): 20

Minimum Interval Value (sec): 5

☒ Limit Output Data

Maximum Plant Output: 23

Minimum Plant Output: 20

Simulink Plant Model: Browse

cstr

Erase Generated Data Import Data Export Data

Training Parameters

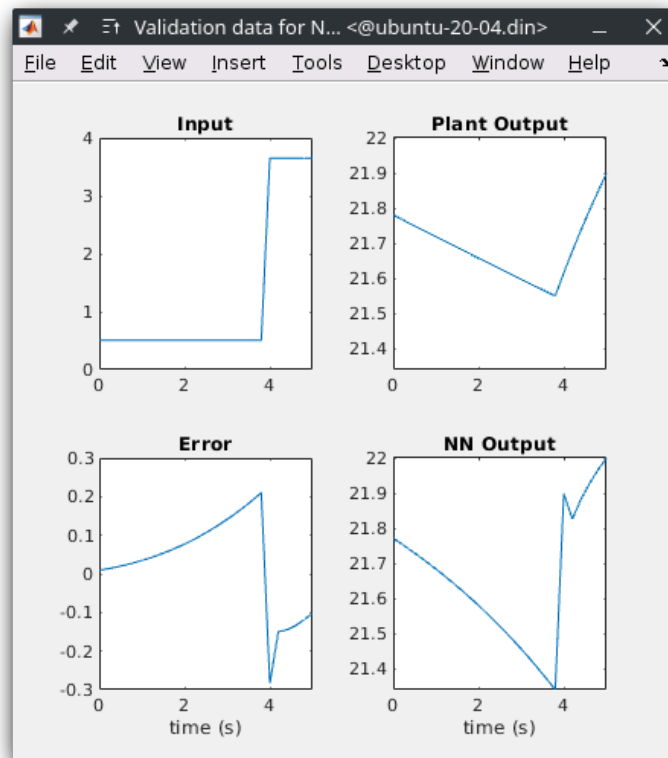
Training Epochs: 200

Training Function: trainlm

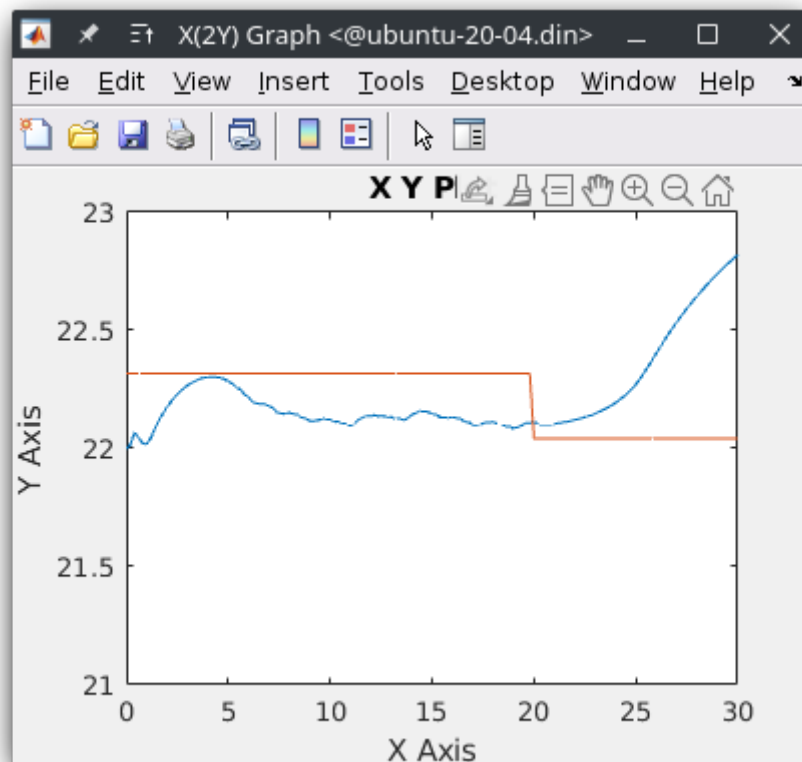
☐ Use Current Weights ☒ Use Validation Data ☐ Use Testing Data

Train Network OK Cancel Apply

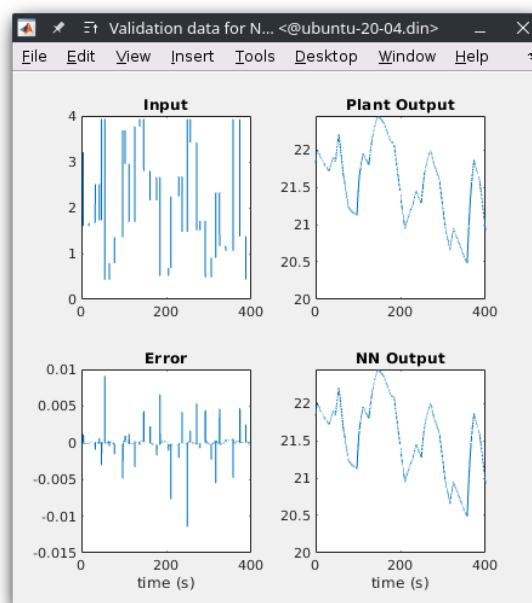
**Your training data set has 100 samples.
You can now train the network.**

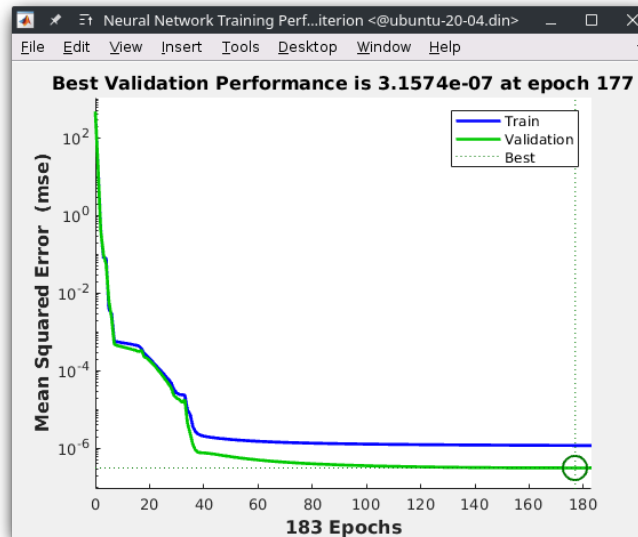


Como era de esperar, el resultado del controlador después de 30 segundos empleando esta red neuronal es totalmente erróneo.



Viendo estos resultados se aumenta el número del conjunto de datos de entrenamiento a 8000, y se vuelve a entrenar la red neuronal desechando los pesos anteriores. En este caso, la salida de la red neuronal en el conjunto de validación se ajusta mucho más a la salida de la planta, y el error cuadrático medio (mse) en el conjunto de validación es de tan solo 3.1574×10^{-7} tras 187 iteraciones en el conjunto de entrenamiento.



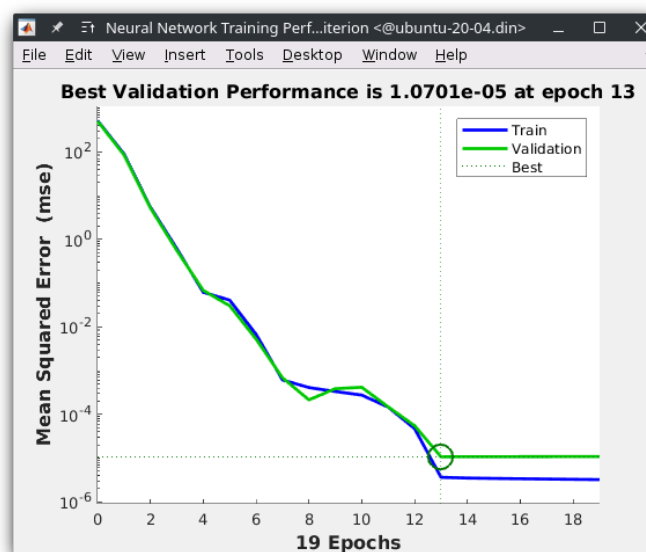


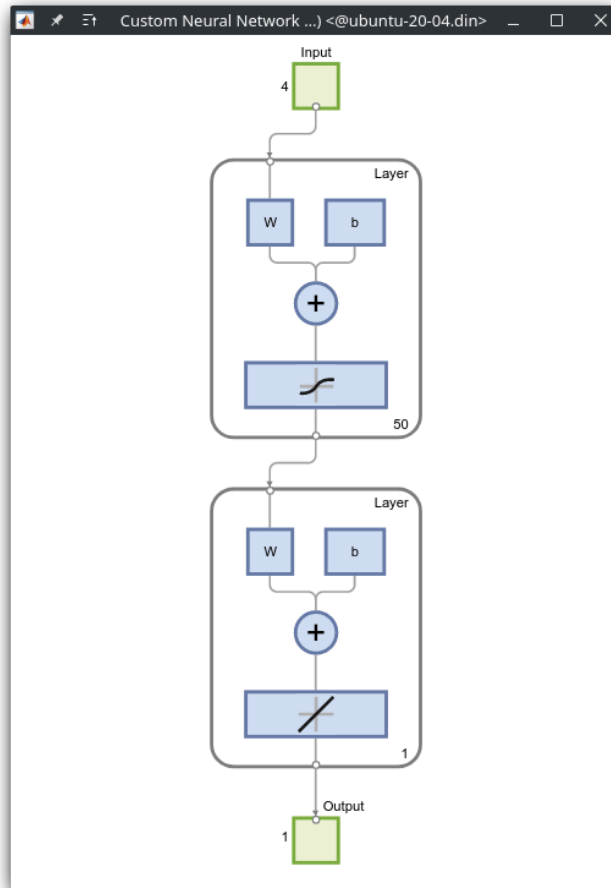
Con esta información se puede determinar que el conjunto de datos de entrenamiento es lo suficientemente grande como para obtener resultados adecuados empleando una red neuronal entrenada sobre los mismos.

Diseño del sistema de control

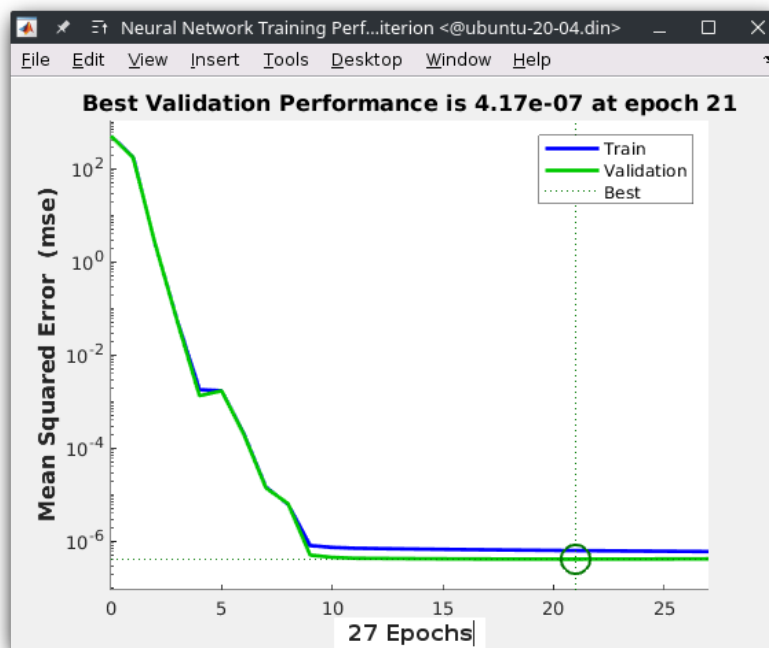
Ahora se procederá a modificar la estructura de la red neuronal para intentar mejorar el error en el conjunto de validación, para ello se aumentará el número de neuronas de la capa oculta a 50 y se reentrenará la red desechando los pesos anteriores.

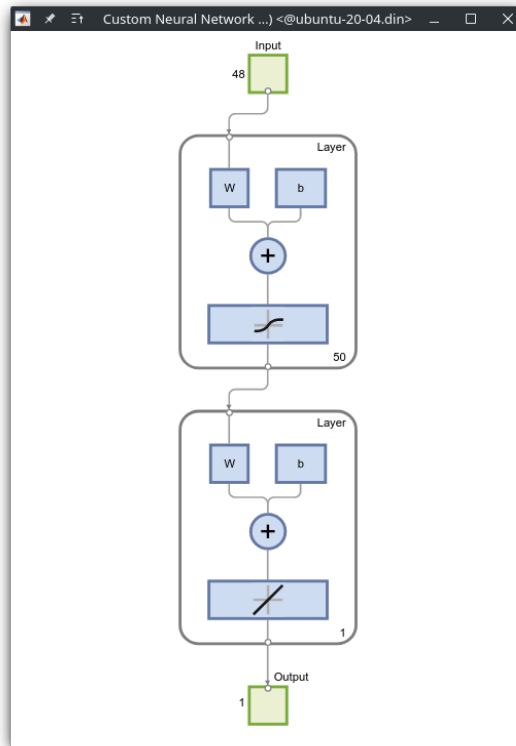
La red neuronal obtenida tiene un error (mse) de 1.0701e-05 tras 13 iteraciones, esto es 2 órdenes de magnitud superior al obtenido anteriormente, lo cual tiene sentido, ya que la red neuronal tan solo tiene 4 entradas y 1 salida por lo que introducir un número muy elevado de neuronas en la capa oculta, como 50, suele empeorar el rendimiento de la red neuronal.



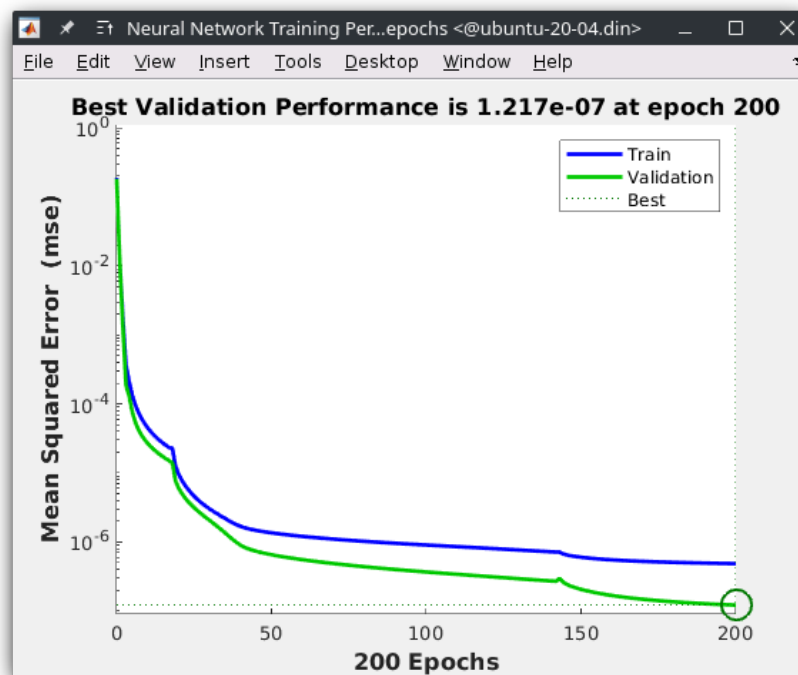


Si se aumenta el número de entradas y salidas de la planta a 24 cada una, manteniendo el número de neuronas en la capa oculta en 50, se comprueba que el error (mse) vuelve a bajar a $4.17\text{e-}07$.

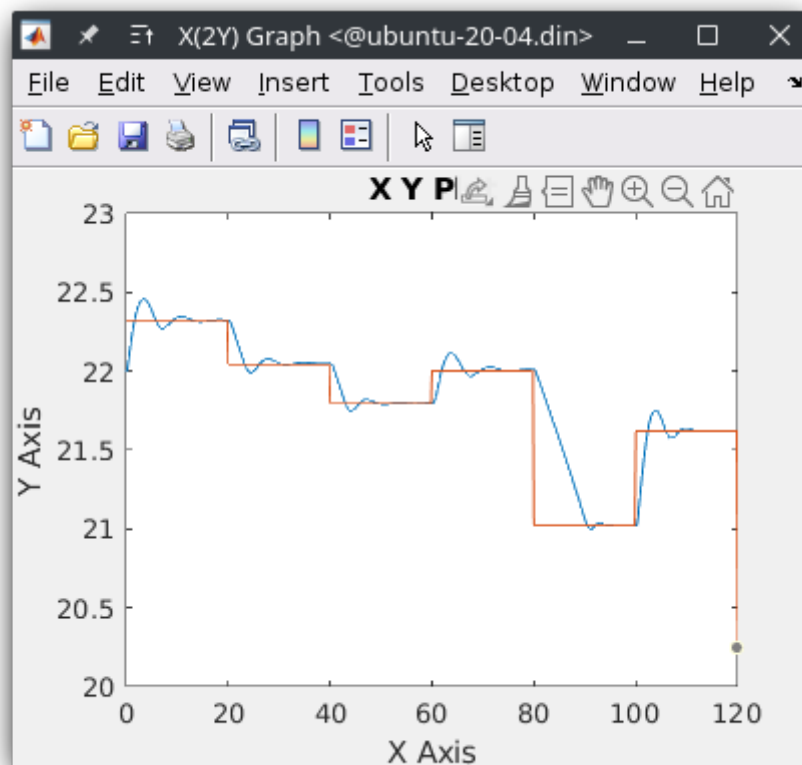




El aumento del número de neuronas de la red lleva a un aumento considerable del tiempo de entrenamiento de la red neuronal y del tiempo de simulación, sin llevar a un mejor rendimiento. Por lo tanto, se ha decidido bajar el número de neuronas de la capa oculta a 2 y el número de entradas y salidas de la planta a 1. Este modelo tiene un error (mse) de $1.217e-07$.

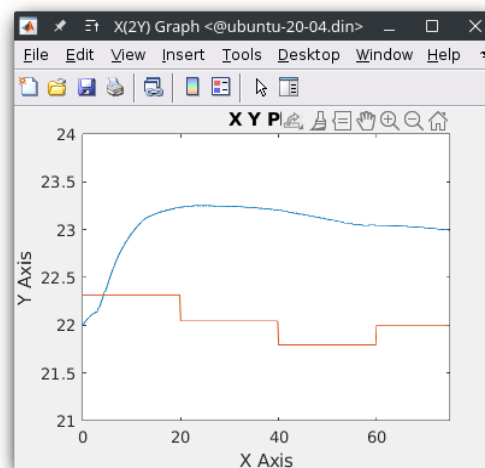


Con la red neuronal resultante se obtiene la simulación de a continuación, es muy similar a la simulación original, por lo tanto, se considera exitosa la identificación del sistema.

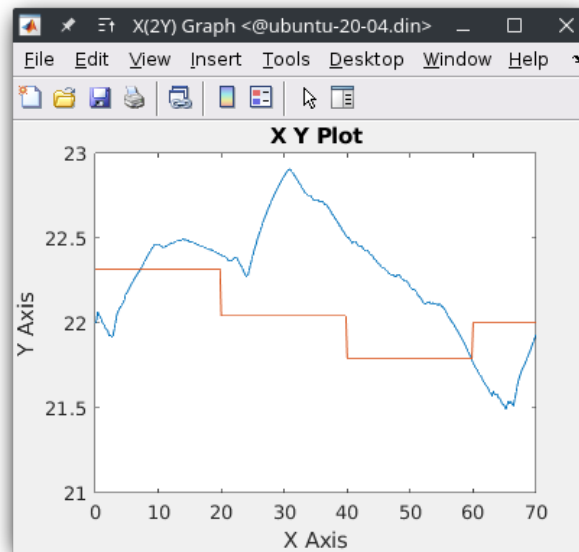


Influencia del modelo en la calidad del control

Tener un modelo neuronal de calidad es necesario para obtener un control adecuado, ya se ha visto como se comporta el control con un modelo adecuado, por lo que se mostrará una simulación con un modelo inadecuado, por ejemplo, con un modelo entrenado con un conjunto de 100 datos. Se puede ver que el control es totalmente erróneo.



También se puede ver que si se aumenta de forma indiscriminada el número de neuronas en la capa oculta, como por ejemplo a 100, el rendimiento final en el control será subóptimo.



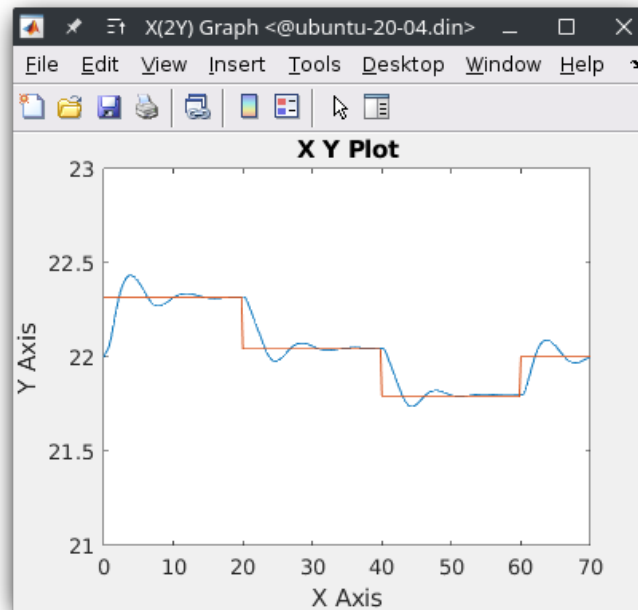
Por lo tanto, se demuestra que es imprescindible tener un modelo neuronal correcto para que el control sea adecuado.

Influencia de los parámetros del controlador

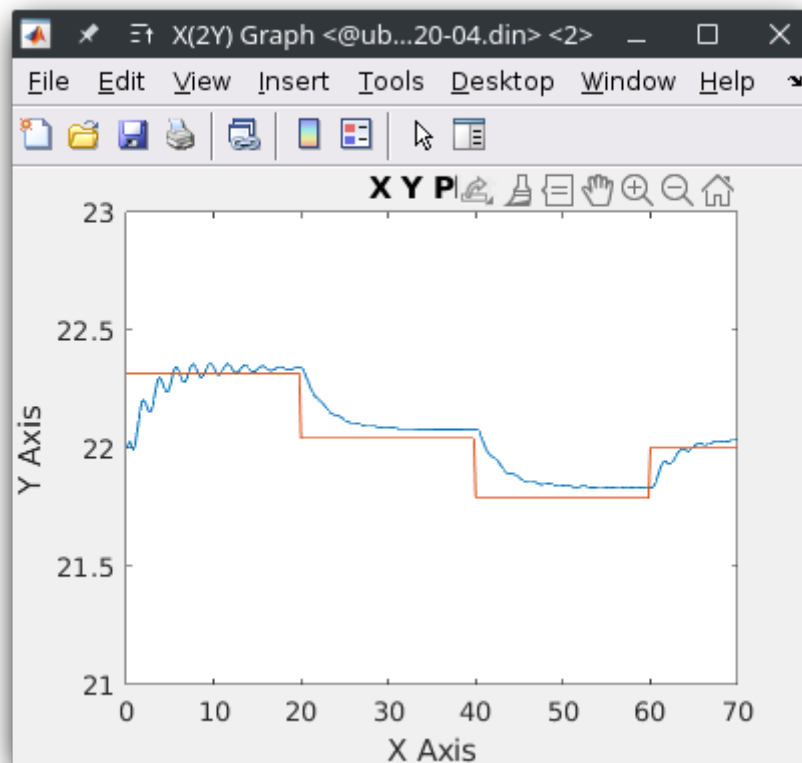
Para analizar la influencia de los parámetros del control sobre la salida de la simulación hay que tener en cuenta el valor inicial de los parámetros. En este caso, los parámetros iniciales son $N2=7$ (horizonte de predicción), $Nu=2$ (horizonte de control), $Wc=0.05$ (factor de peso del control).

The figure shows a configuration window titled 'Neural Network Predictive Control <@ubuntu-20-04.din>'. It contains several input fields and buttons for configuring the controller. The 'Cost Horizon (N2)' is set to 7, and the 'Control Horizon (Nu)' is set to 2. The 'Control Weighting Factor (λ)' is set to 0.05, and the 'Search Parameter (λ)' is set to 0.001. The 'Minimization Routine' is set to 'csrchbac', and the 'Iterations Per Sample Time' is set to 2. There are buttons for 'Plant Identification', 'OK', 'Cancel', and 'Apply'. A blue text box at the bottom states: 'Perform plant identification before controller configuration.'

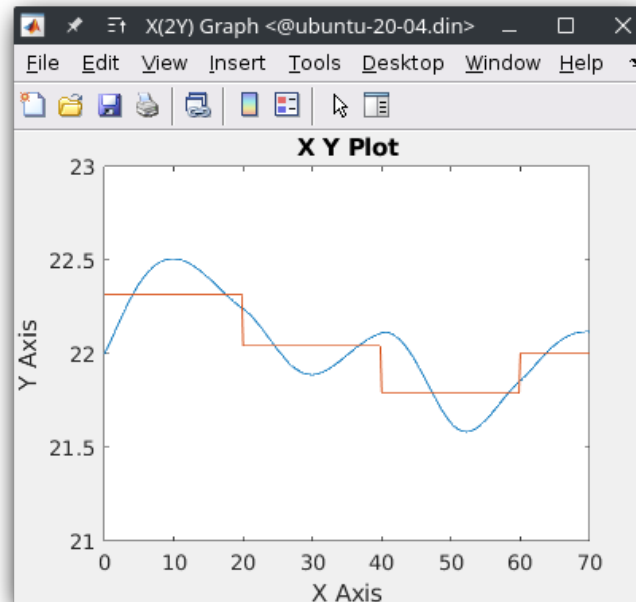
La señal de salida durante 70 segundos con los parámetros por defecto y la red neural entrenada anteriormente es la siguiente.



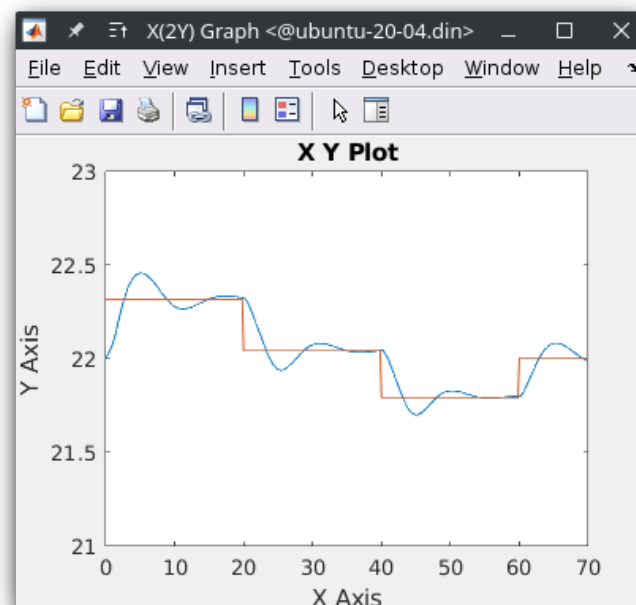
Al subir el horizonte de predicción a 28 el actuador al inicio varía mucho más frecuentemente, con múltiples subidas y bajadas, sin embargo, en el periodo transitorio a penas excede el valor de referencia, al contrario de cuando estaba en 10, pero en el periodo estacionario nunca llega a alcanzar el valor de referencia después de una bajada.



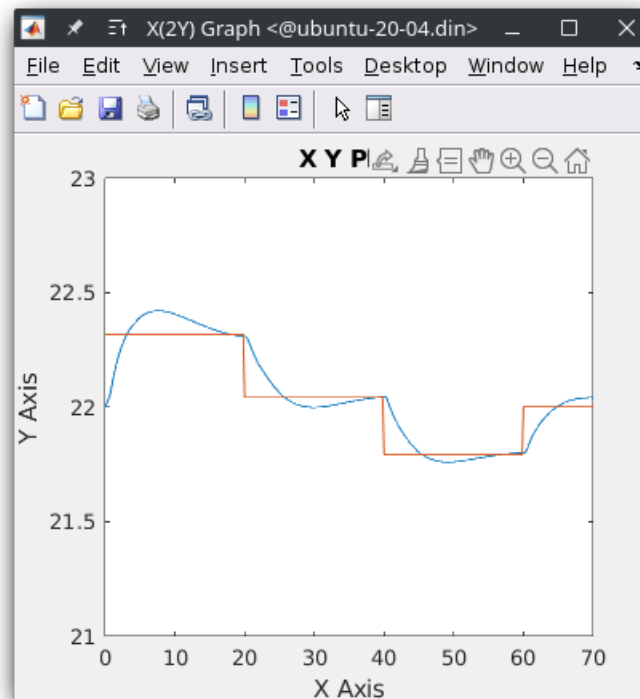
Si se disminuye el horizonte de predicción a 2, se ve que el actuador varía menos veces, pero los cambios que realiza son mucho más abruptos, y en el periodo estacionario excede notablemente el valor de referencia tanto en la subida como en la bajada. Además, no da tiempo a que llegue al estacionario antes de que cambie la señal, por lo que apenas está relacionada con la señal de referencia.



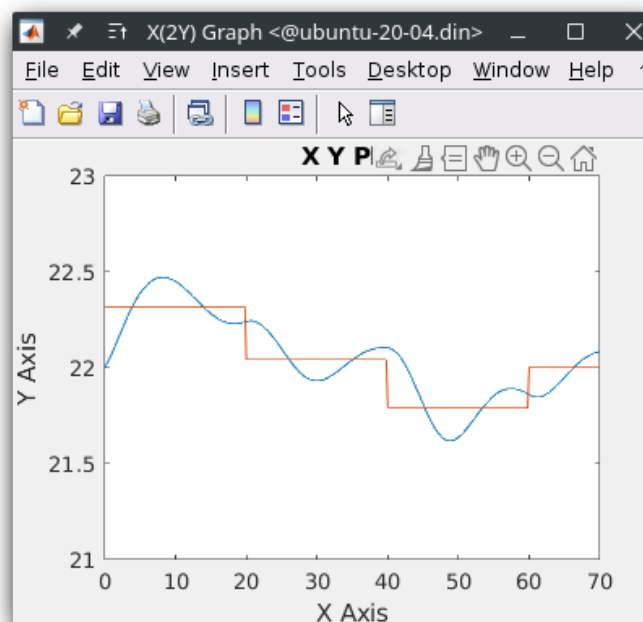
El valor del horizonte de control por defecto es 2, lo cual ya es el mínimo para este sistema, por lo que se comprueba el comportamiento al incrementarlo a 5. En este caso se ve que al aumentar esta variable el comportamiento es similar a cuando se disminuía el horizonte de predicción, la frecuencia de cambio del actuador disminuye, por lo que aumenta el tiempo en el que la señal tarda en llegar al valor de referencia. Aunque en este caso el estacionario es más similar al original.



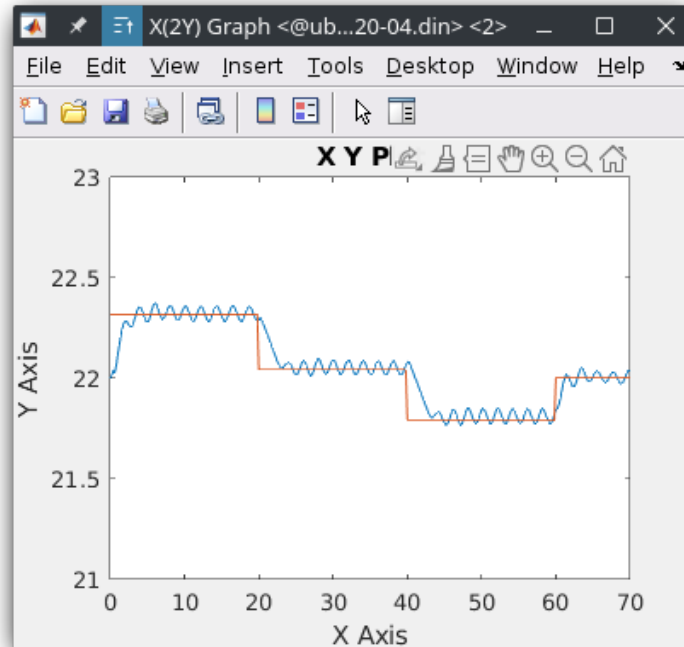
Si se sube el horizonte de control aún más, a 7, se ve como el actuador varía aún menos, reduciendo el número de subidas y bajadas, esto se traduce en una señal aún más suave, pero también está más alejada de la señal de referencia que en el caso original.



A continuación se ha aumentado el factor de peso del control a 0.5, esto vuelve a tener un efecto similar al que se obtiene al disminuir el horizonte de predicción, se logra una señal más suave, con menos subidas y bajadas. Pero está lejos de alcanzar el valor de referencia y no llega a alcanzar un periodo estacionario antes del cambio del valor de referencia.



Si se disminuye el factor de peso del control a 0.005, la señal obtenida tiene un periodo transitorio muy corto, sin embargo, en el estacionario no para de oscilar alrededor del valor de referencia. Se ve que el actuador se está accionando constantemente, causando numerosas subidas y bajadas en la señal.



También se ha intentado ajustar el parámetro de búsqueda, pero la diferencia entre el valor de defecto y los valores 0.01 y 0.0001 era prácticamente imperceptible.

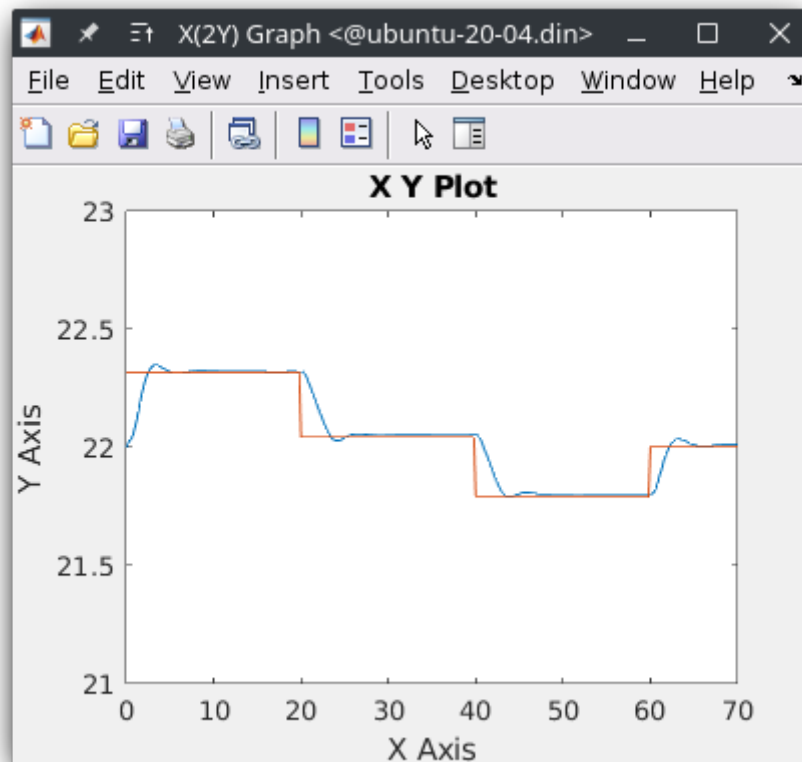
Ahora que se ha descubierto el efecto de los parámetros del controlador sobre la calidad del control, se ha procedido a buscar una combinación de parámetros que mejora de forma notable la señal original. Tras prueba y error se han obtenido los siguientes parámetros: horizonte de predicción = 11, horizonte de control = 2 y factor de peso del control = 0.1.

The figure shows a window titled 'Neural Network Predictive Control <@ubuntu-20-04.din>'. It contains several input fields and buttons for configuring the control system. The configuration parameters are as follows:

Parameter	Value
Cost Horizon (N2)	11
Control Horizon (Nu)	2
Control Weighting Factor (λ)	0.1
Search Parameter (λ)	0.001
Minimization Routine	csrchbac
Iterations Per Sample Time	2

Buttons at the bottom include 'Plant Identification', 'OK', 'Cancel', and 'Apply'. A blue text box at the bottom states: 'Perform plant identification before controller configuration.'

La señal durante 70 segundos resultante es la siguiente.



Si se compara la señal original (derecha) con la actual (izquierda) durante 120 segundos, la diferencia es bastante notable. El sistema final tiene un comportamiento más suave, que no excede el valor de referencia durante el transitorio, ni en la bajada ni en la subida, y que se ajusta al valor de referencia a lo largo del estacionario. Todo esto sin comprometer la velocidad en la que alcanza el estacionario ni su respuesta al cambio.

