**PRACTICA 5: NN Brazo Robot**



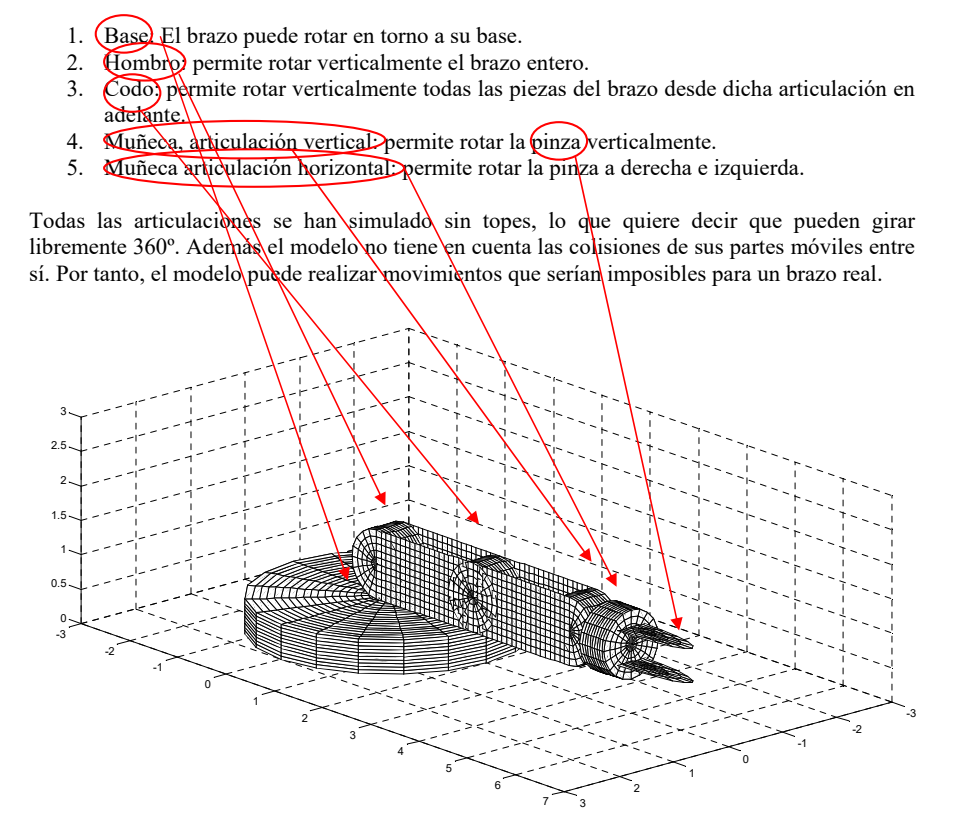
**Nombre:** Gonzalo Sanz Rodríguez.

**Asignatura:** IAAC.

**Introducción:** En esta práctica vamos a hacer uso de las redes neuronales para que un brazo robótico realice un movimiento automático intentando imitar un movimiento establecido por nosotros previamente. Para tener un conocimiento apropiado de la herramienta nntool es recomendable realizar la práctica de XOR mencionada por los profesores.

La primera parte de esta práctica consiste en establecer una secuencia de movimientos distinguiendo de cada una de las 5 partes del robot individualmente. Puesto que si hacemos movimientos muy brucos estos complicarían mucho el aprendizaje a la red, por lo tanto, lo que buscamos es un movimiento sencillo de cada uno de los grados de libertad del robot.

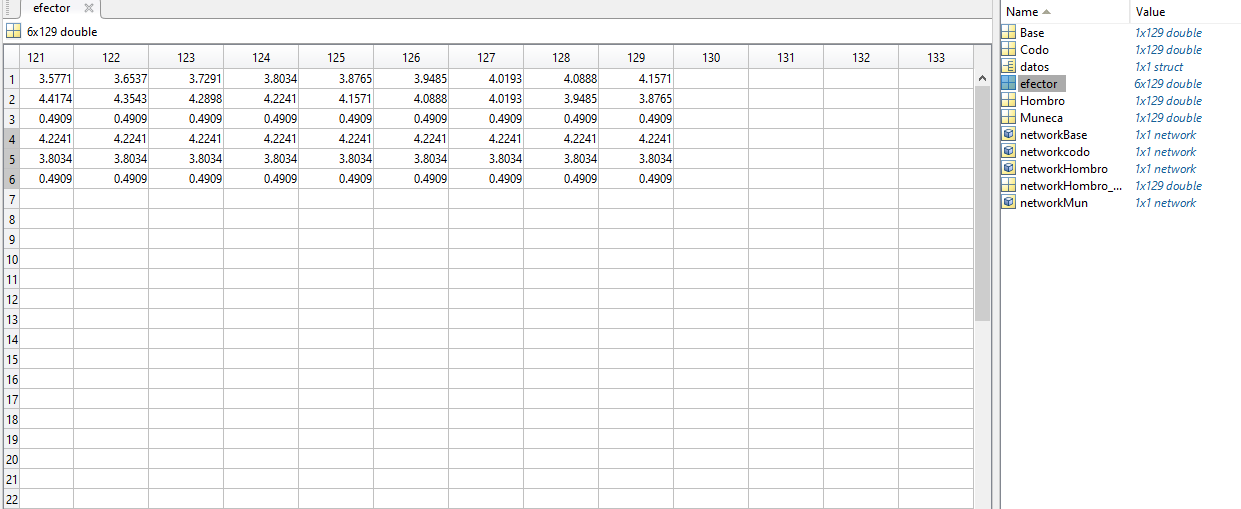
Como es mencionado en el guion de la práctica estos movimientos son controlados con el teclado.



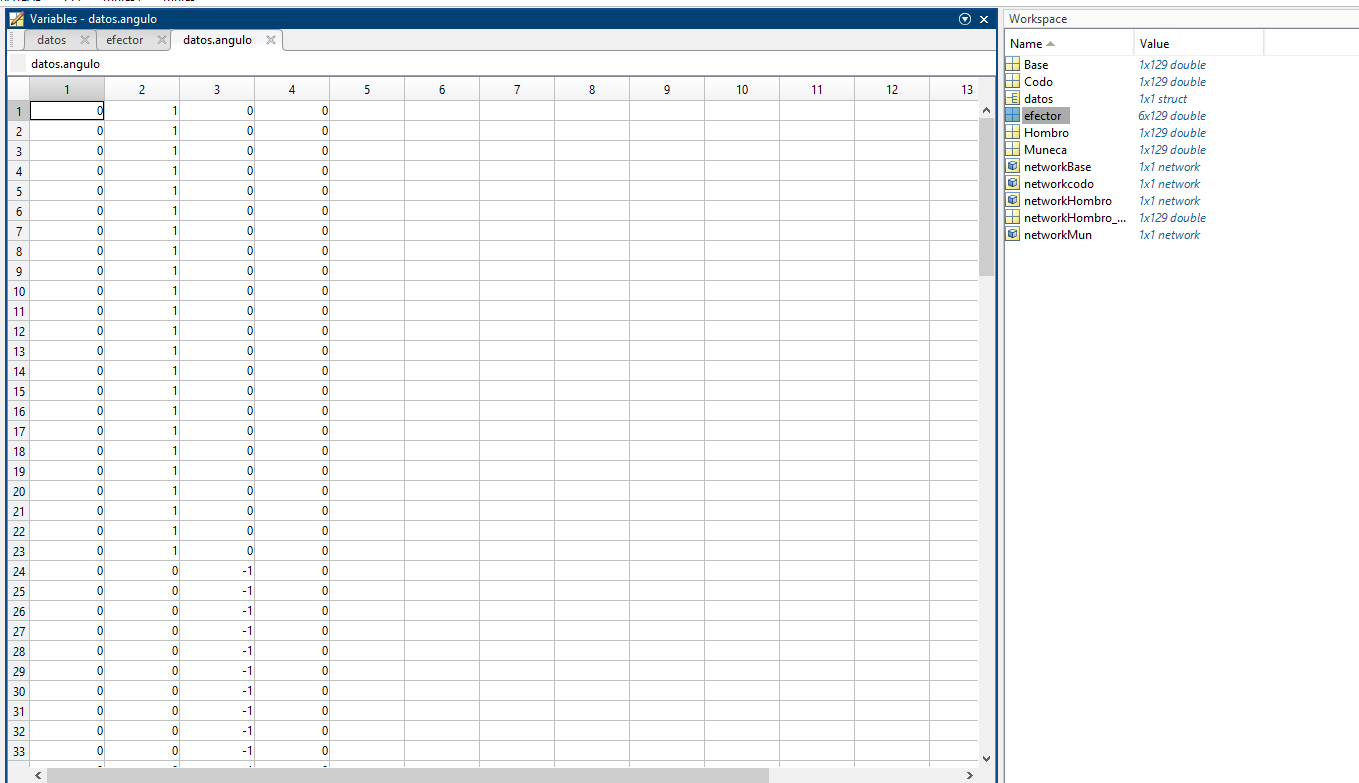
En segundo lugar, deberemos de guardar estos datos y exportarlos al workspace para poder trabajar con ellos en un futuro, ya que son los que nos permitirán configurar y enseñar a las redes de cada una de las articulaciones.

Los datos son almacenados en la variable “datos” que se trata de un struct en la que hay que distinguir dos componentes: el de **efector** en el que se almacenan todas las posiciones por las que pasa el brazo robot y la de **ángulo** que nos muestra los grados que toma cada una de las articulaciones base, hombro, codo y muñeca respectivamente.

Para tratar los datos para que las redes puedan utilizarlos, modificamos la variable efector en la que primero deberemos realizar una transposición de la variable para que puedan tratar mejor los datos las redes y seguidamente añadiremos repetidamente hasta la posición final del array, la última posición del brazo, de esa manera **estamos configurando a la red para que pueda aprenda a llegar a esta posición desde cualquier posición previa.**

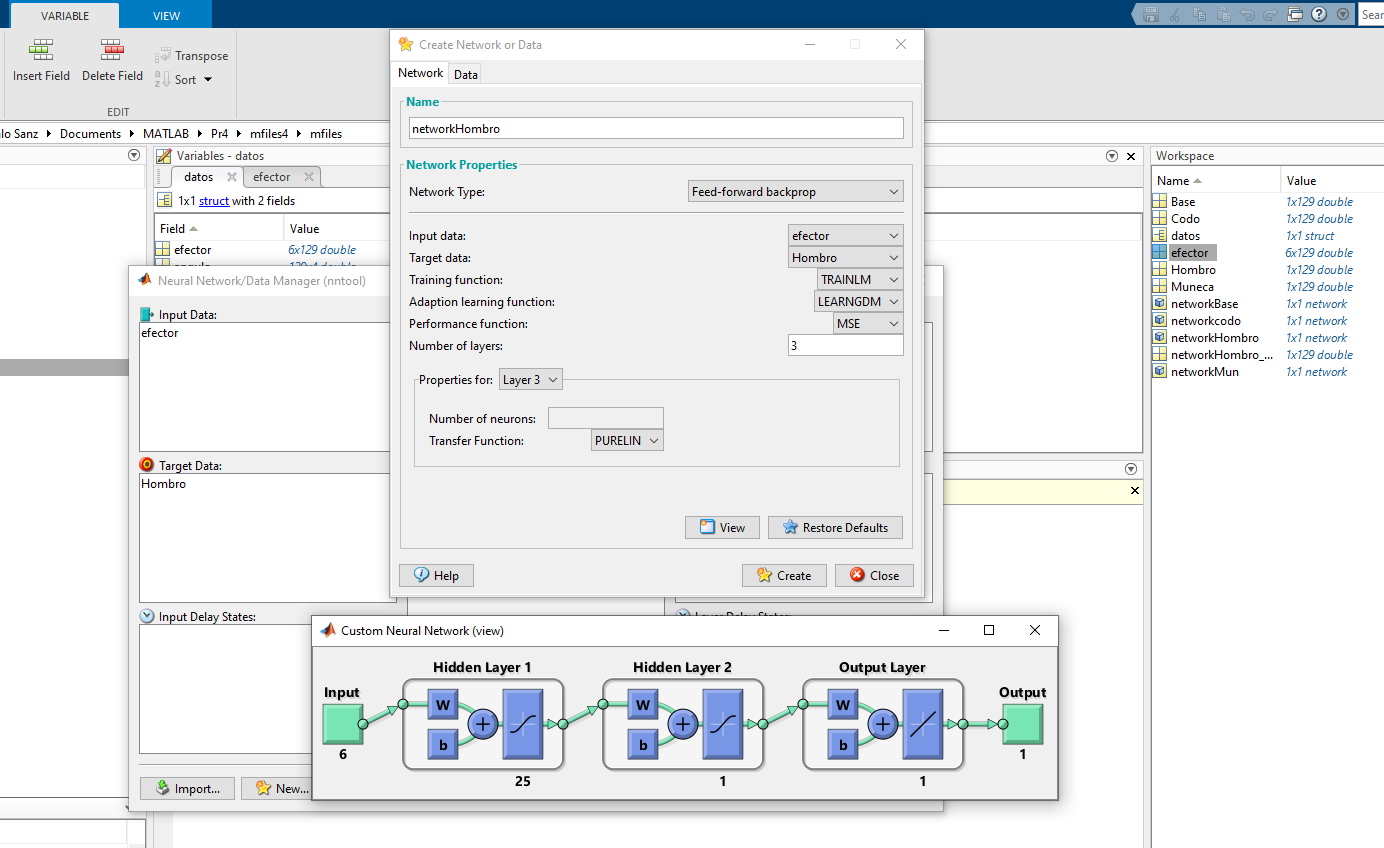


Para los datos de salida es más sencillo simplemente basta con separar cada una de las **columnas en variables independientes,** significando los grados que toman la base, hombro, codo y muñeca respectivamente, realizando también la transposición de cada una de las mismas.

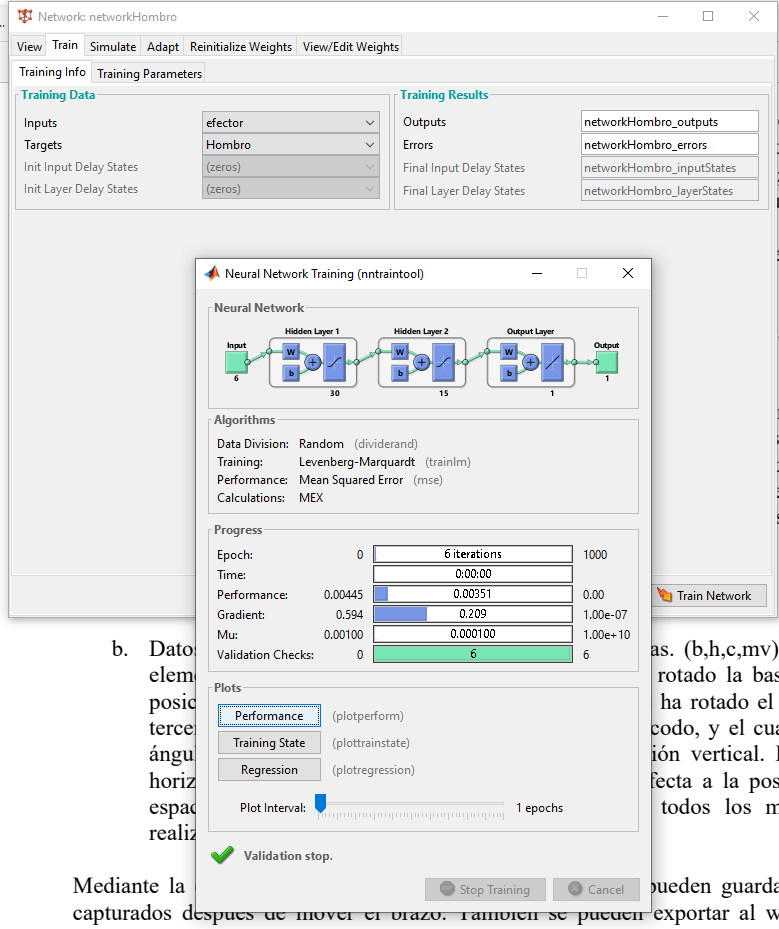


Posteriormente pasaremos a crear, configurar y enseñar a las redes con la herramienta nntool. Procederemos a importar los datos de entrada de salida, y nos disponemos a crear las redes, en las que seguimos la estructura dictada en el guion tratándose de 3 capas con 2 de ellas ocultas y en las que utilizaremos el número de neuronas **distinto en cada articulación en base a los resultados obtenidos con las redes.**

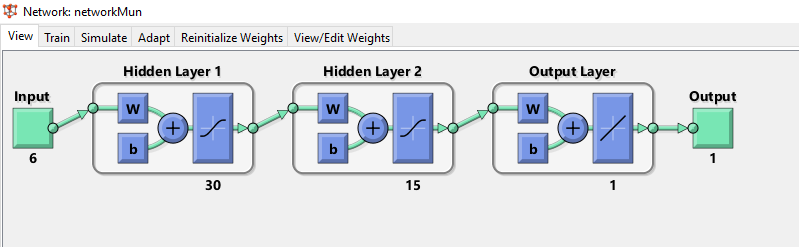
(las neuronas aplicadas en esta red no son las definitivas).

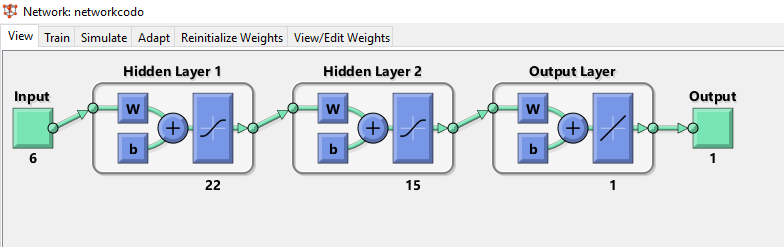


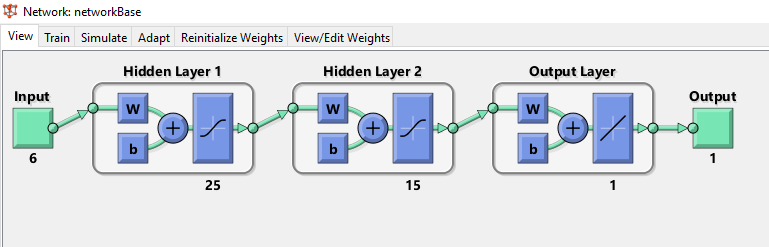
Seguidamente procederemos a entrenar estas redes buscando que encuentren los mínimos globales, serán entrenadas con los datos E/ S que hemos suministrado previamente.

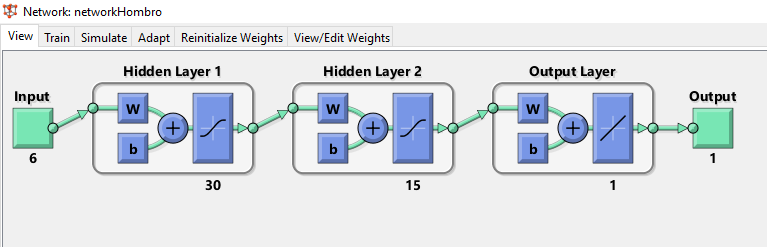


La estructura de las redes que escogí es la siguiente 2 capas ocultas, la capa de salida con función de transferencia Purelin:









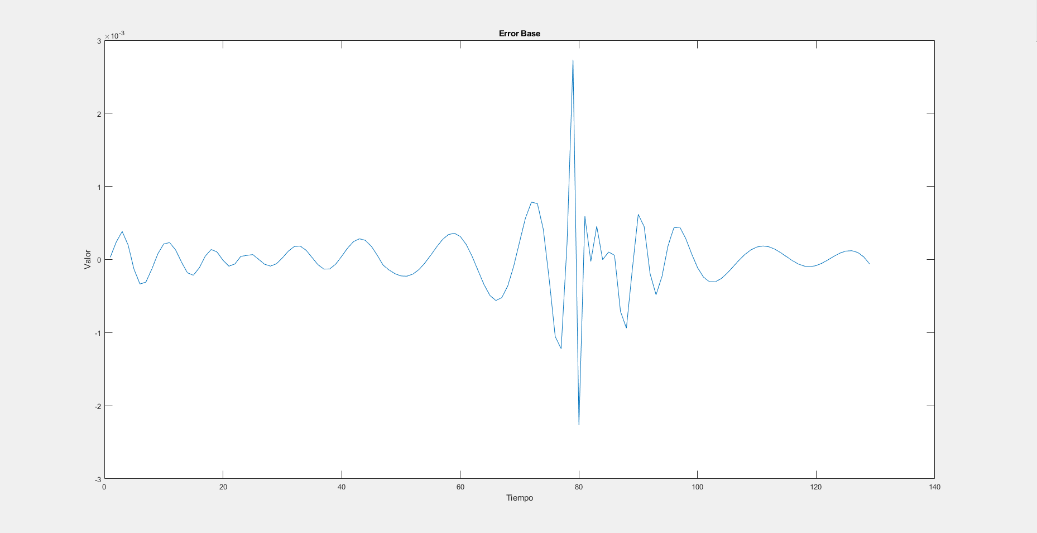
Es muy importante fijarse en el error obtenido por cada una de las redes, puesto que buscamos un error menor a 10 ^ -3 en valor absoluto, para cada una de las salidas del conjunto de entrenamiento, esto se puede observar con la variable de error perteneciente a cada red.

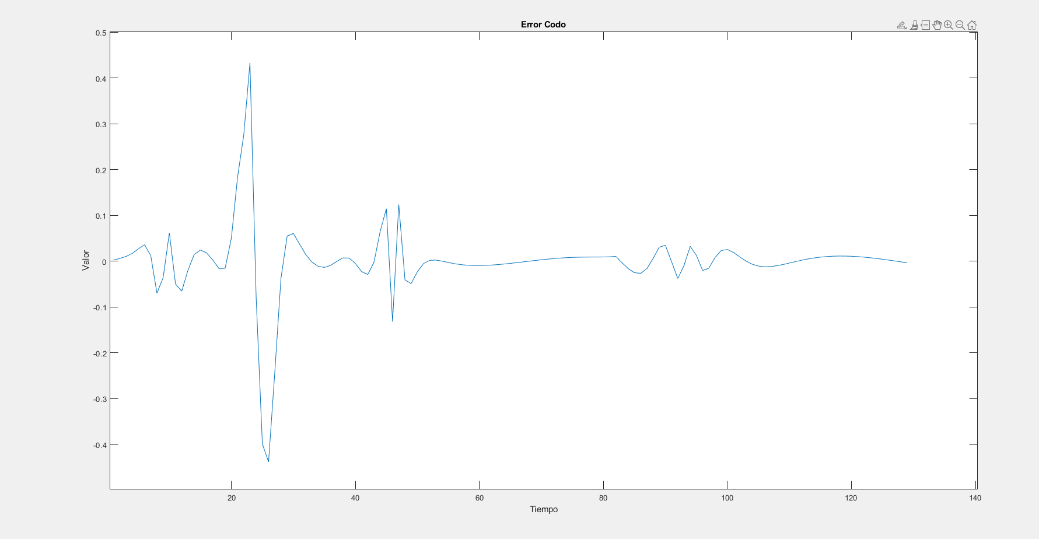
**0.000885989952442046 0.000653054321386515 -0.000599332736013647**

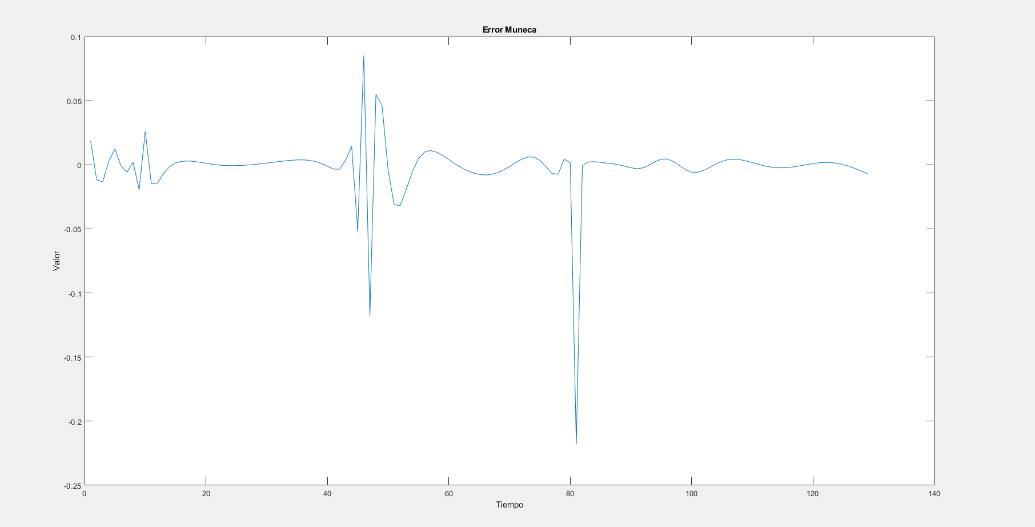
Estos son alguno de los valores obtenidos en las variables, y como podemos observar no sobrepasan el valor establecido, en el caso de que si lo sobrepasaran sería necesario reentrenarlas o restablecer el número de neuronas dispuestas en cada capa.

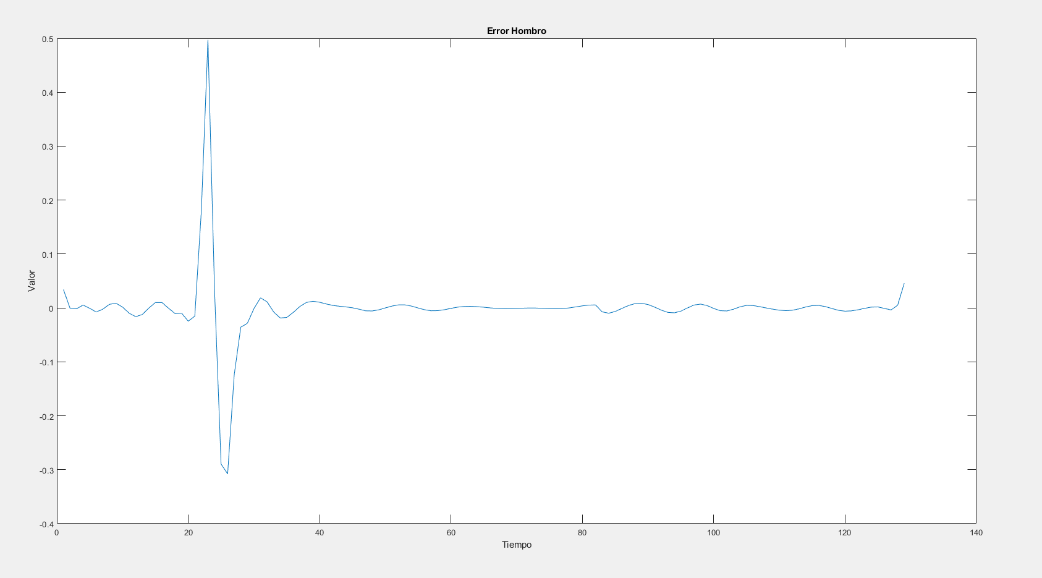
Si decidimos realizar un plot de cada una de las variables acumuladas del error por cada una de las articulaciones, obtenemos lo siguiente:

(Los valores de eje Y están expresados en 10 ^ -3)

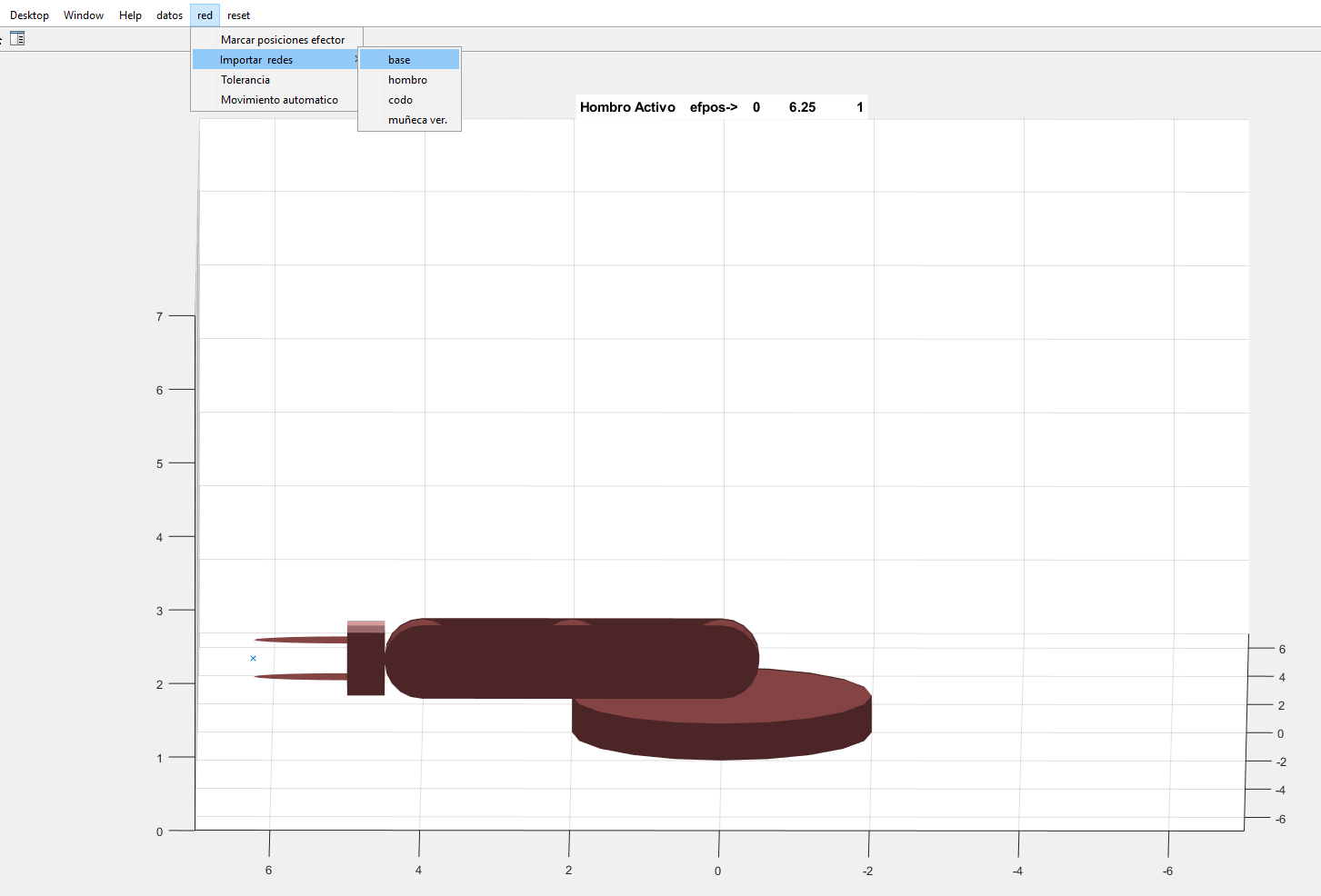




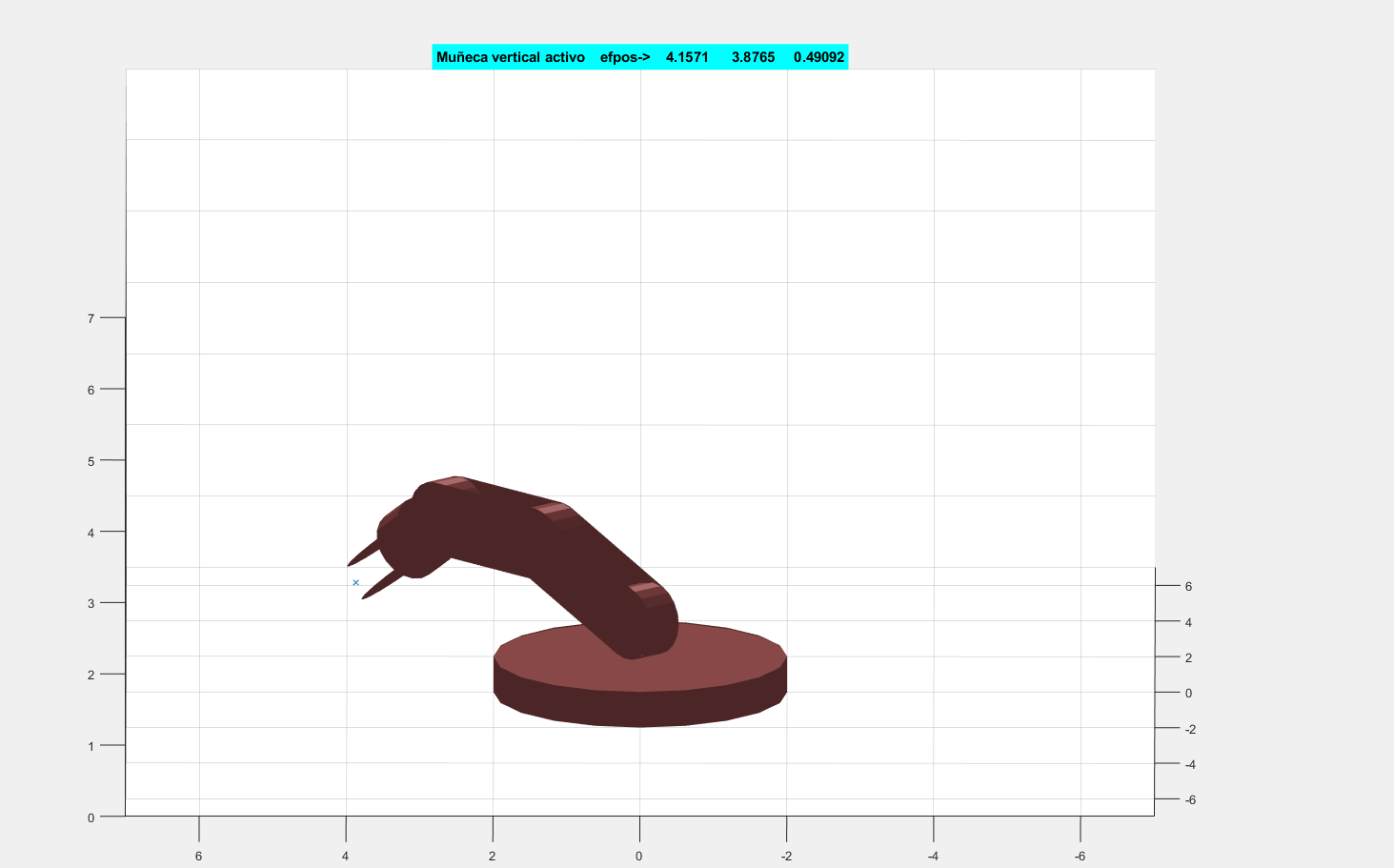




Por último, una vez comprobado el error para a cada una de las redes neuronales, solo nos quedaría importar estas redes en el brazo robot, y probar que el movimiento automático lo realice correctamente y de forma continua.

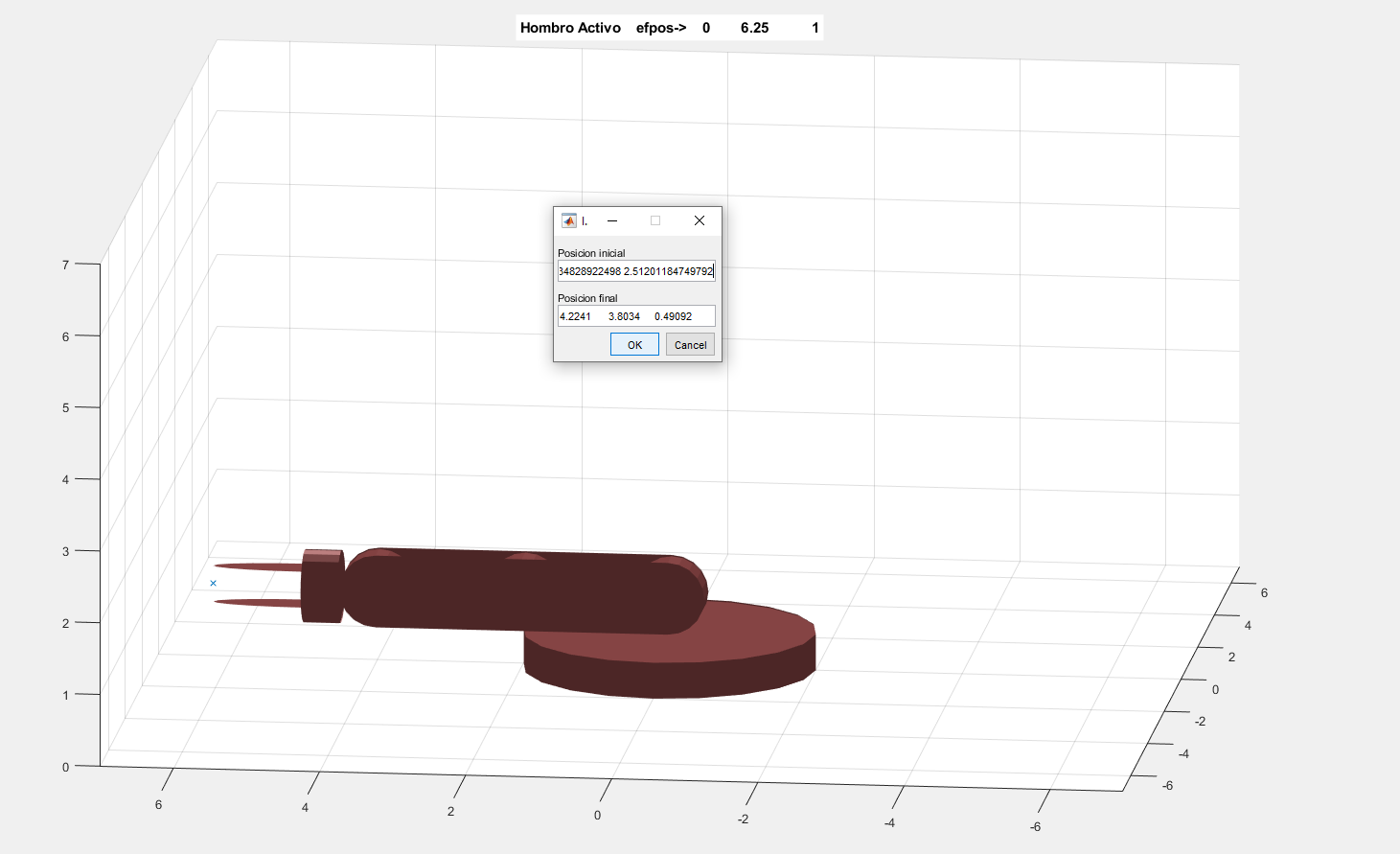


Había establecido como posición final (4.224, 3.8034, 0.49091) y procedemos a mover el brazo. Como observamos en la imagen las redes consiguen llevar al brazo a una posición muy similar a la establecida, esto es debido a que las redes consiguen simular el movimiento, pero no son capaces de imitarlo exactamente igual.



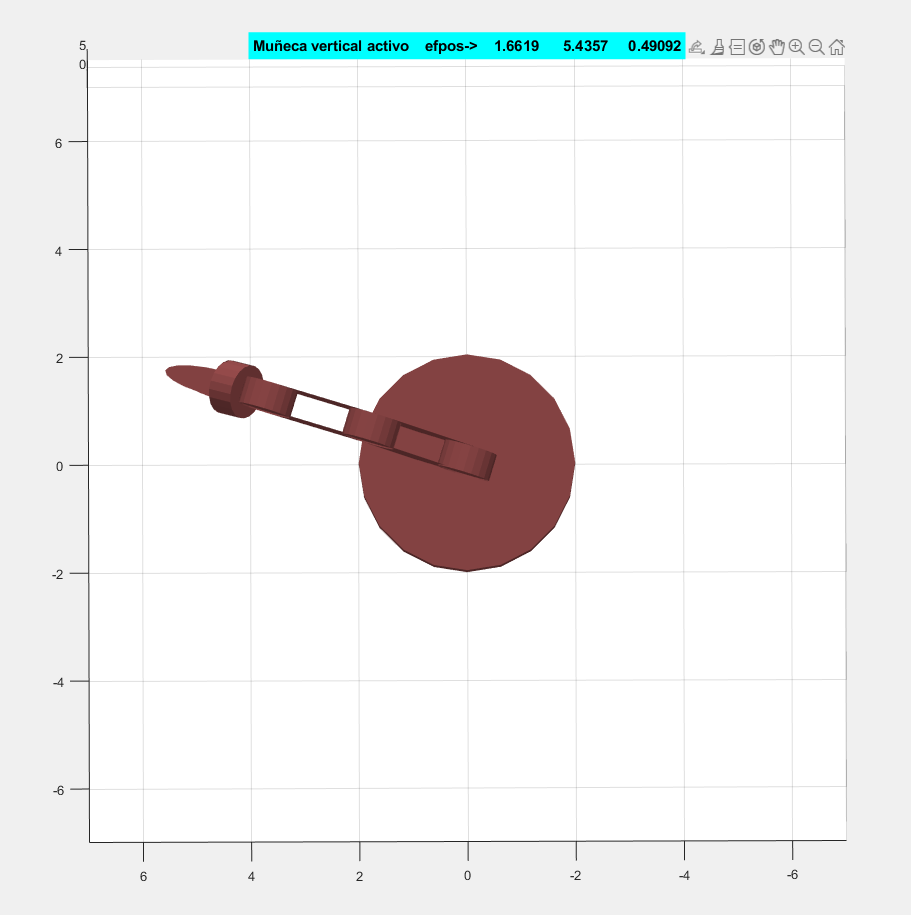
Una vez que hemos conseguido imitar el movimiento completo, probamos a realizar distintas pruebas.

Para empezar, establecemos una posición inicial distinta a la original. De esta manera si el brazo consigue llevarlo a la posición final, esto querrá decir que hemos creado bien el conjunto de entrenamiento y que nuestro brazo es más polivalente



El brazo ha sido capaz de realizar el movimiento por lo que nuestras redes están bien entrenadas.

La segunda prueba a realizar se trata de establecer una posición previa a la final, de esta manera comprobamos que es capaz de parar el movimiento hasta una posición previa a la final y de esta manera ver que, aunque haya sido entrenada para terminar en la final que es capaz de aprender a parar en un movimiento previo. Como vemos en la imagen lo ha realizado satisfactoriamente.

.

Por último pero no menos importante, debemos de realizar una prueba estableciendo una posición que no esta comprendida en el conjunto de entrenamiento, previamente antes de ejecutarlo lo que nos cabe esperar es que el sistema se aproxime bastante a esa posición, pero debido a que no ha sido entrenado con esa posición auxiliar, ya que para llegar a esta posición, debería de haber realizado un conjunto de movimientos previos que la han hecho llegar hasta este punto.

|  |  |
| --- | --- |
|  |  |

Como hemos predicho el brazo no es capaz de simular perfectamente este nuevo movimiento, pero ha sido capaz de acercarse mucho, esto es gracias al conocimiento que ya tienen las redes.