

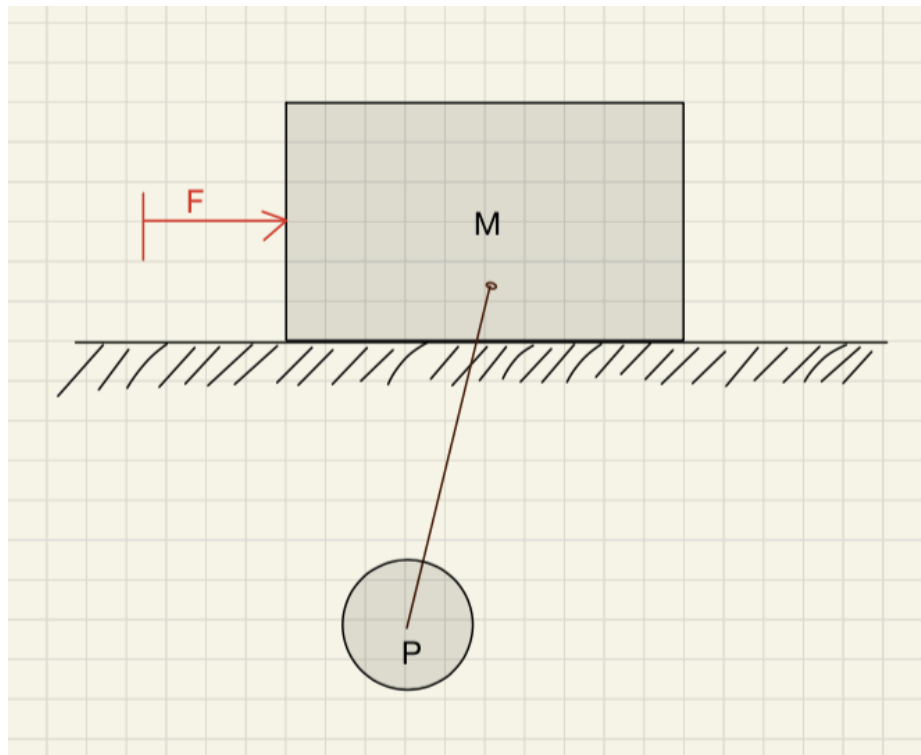
SANTIAGO GARCÍA ARANGO

QUIZ 2

SISTEMAS INTELIGENTES

NOTA: En el quiz se permitirá tener programas desarrollados previamente y utilizarlos como plantilla, pero no se permitirán notas ni diapositivas, como tampoco el uso de internet para fines diferentes al de acceder al moodle.

1. Se tiene un sistema Masa-Resorte-Péndulo, el cual se puede observar a continuación.



El comportamiento del sistema que se observa es el de la masa, donde esta no está acoplada a nada, solo al extremo de actuación que la estará moviendo y al péndulo.

NOTA: Se busca controlar la masa M , el péndulo actuará como una perturbación en el sistema.

SE PIDE LO SIGUIENTE:

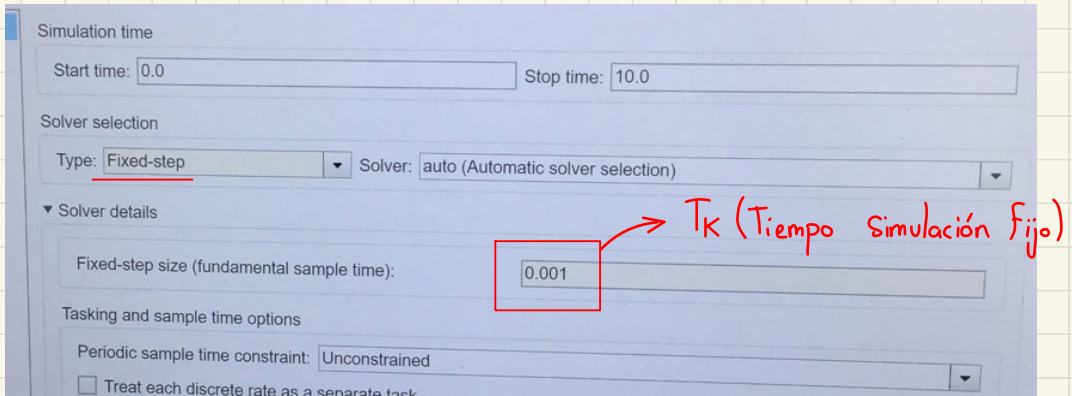
- ✓ A. Abra el archivo Simulink donde se encuentra el sistema masa péndulo ya desarrollado.
- ✓ B. Ajuste el tiempo de simulación estático según se necesite.
- ✓ C. Realice una toma de datos con un solo step e identifique un tiempo de establecimiento adecuado.
- ✓ D. Realice una toma de datos con múltiples SetPoints de manera aleatoria (entre 0 y 0.5), con un tiempo de simulación de 200 segundos.
- ✓ E. Realice un diagrama de identificación inversa, ya sea a mano o en un programa digital y explique en detalle el flujo de información.
- ✓ F. Realice un diagrama de control inverso adaptativo, ya sea a mano o en un programa digital y explique en detalle el flujo de información.
- ✓ G. En simulink, preentrene unas matrices de pesos con una identificación inversa y explique el procedimiento desarrollado.
- ✓ H. En simulink aplique un controlador inverso adaptativo preentrenado y explique en detalle el procedimiento.
- ✓ I. Muestre resultados y gráficas del controlador y explique.

NOTA: El controlador no tiene que ser perfecto, puede oscilar un poco y moverse, pero debe llevar el sistema al setpoint deseado(se puede mover el tiempo de setpoint si se necesita).

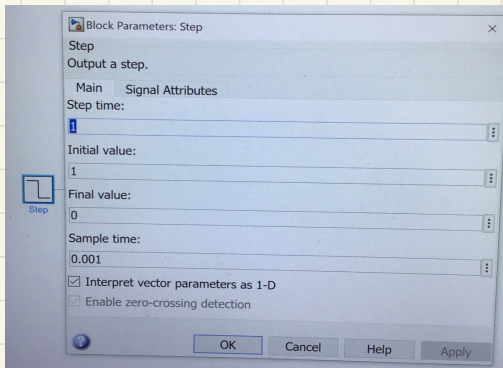
NOTA IMPORTANTE: En un documento de word o pdf explique en detalle el paso a paso de cada numeral y muestre imágenes y resultados, explicando cada uno de ellos.

MUCHOS ÉXITOS!

En primer lugar, se fija el tiempo de simulación estático a un tiempo fijo de "0.001s":

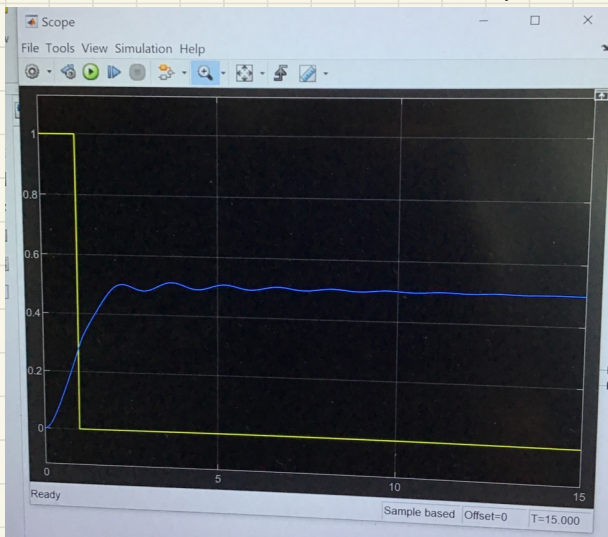


Para realizar la correcta identificación y análisis del sistema, se procede a tener como entrada un step de magnitud unitaria, de la siguiente forma:



Donde su duración será de 1 segundo, con magnitud unitaria y luego el SP se irá a cero (debido a que de lo contrario, el péndulo tiene características dinámicas integradoras y seguiría de manera indeterminada).

Al hacer este cambio, el sistema responde así:



Se puede evidenciar claramente el tiempo de estabilización cercano a los 10 segundos (según el criterio de "settling time de 2%). Esto es un dato relevante para nuestros próximos análisis.

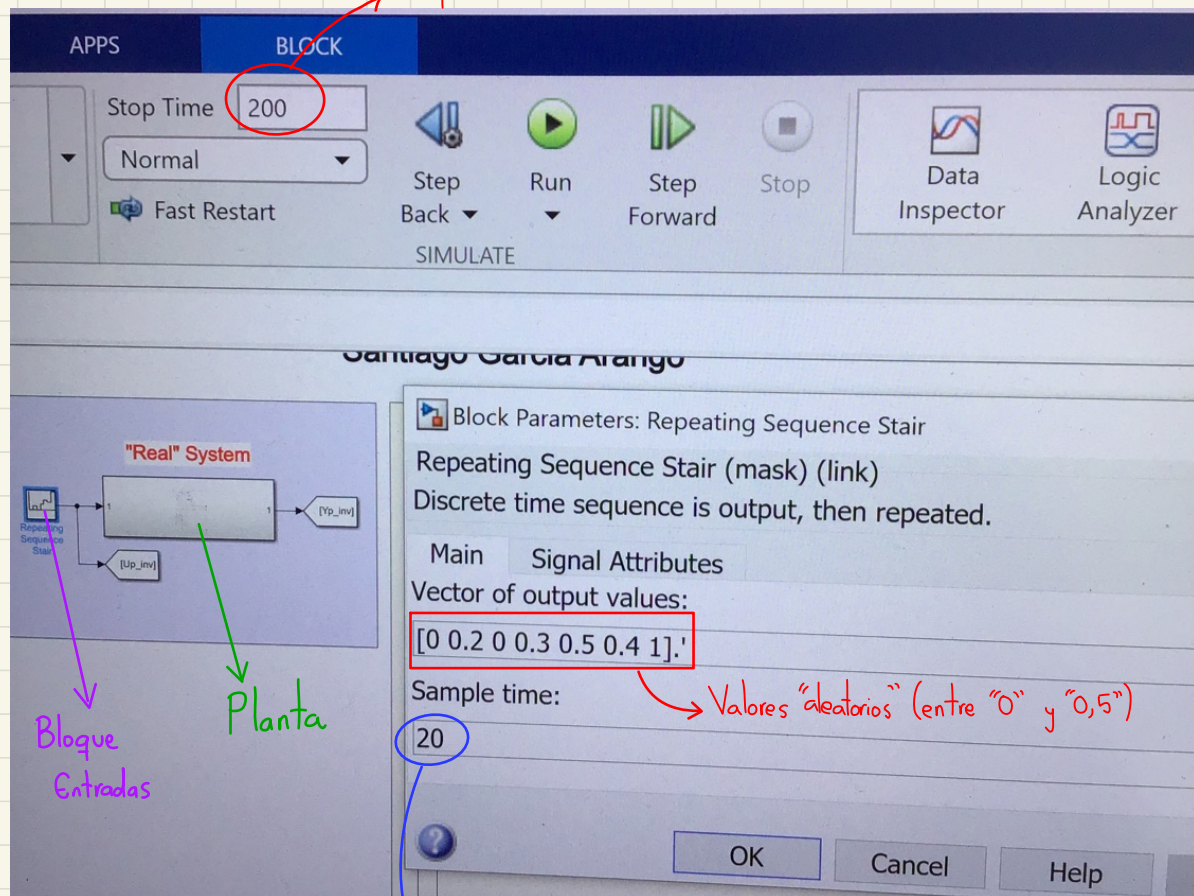
De igual forma, vemos que el sistema demuestra su comportamiento integrador de forma evidente a lo largo del step de prueba.

$t_{ss} \approx 10 \text{ segs}$

Luego de realizar este proceso, el siguiente paso importante es realizar una identificación del sistema. En este caso realizaré una identificación NO paramétrica, pues el objetivo es que una red neuronal se aprenda el comportamiento del sistema de forma inversa, es decir, que a partir de la salida del mismo, sea capaz de entrenar la red para obtener su información "inversa". El control inverso luego tomará esto como insumo para efectuar las acciones de control necesarias en el péndulo-masa.

La identificación debe abarcar varios valores del margen de trabajo dentro de los rangos de funcionamiento del sistema. Esto se realizará con el bloque "Repeating Sequence Stair" en simulink, de la siguiente forma:

Tiempo total simulación (toma de datos)



Tiempo suficiente para "estabilización".

Luego de la toma de datos(del sistema masa-péndulo), se tiene un resultado interesante de tipo integrador (como se dijo anteriormente)...

Este resultado nos permite identificar la planta y el control inverso debe estar en condiciones de adaptarse a este comportamiento integrado, pues lo único que necesitamos ahora es la planta (y aunque fuese más directo si NO fuese integradora, esto NO es un stopper para el control):



Aclaración importante: en esta etapa tenemos dos opciones... la primera es cambiar al sistema para que no tenga comportamiento integrador, y la segunda es seguir con dicho comportamiento, pero teniendo presente que el control inverso debe ser capaz de enviar señales positivas y negativas para garantizar las leyes de control...

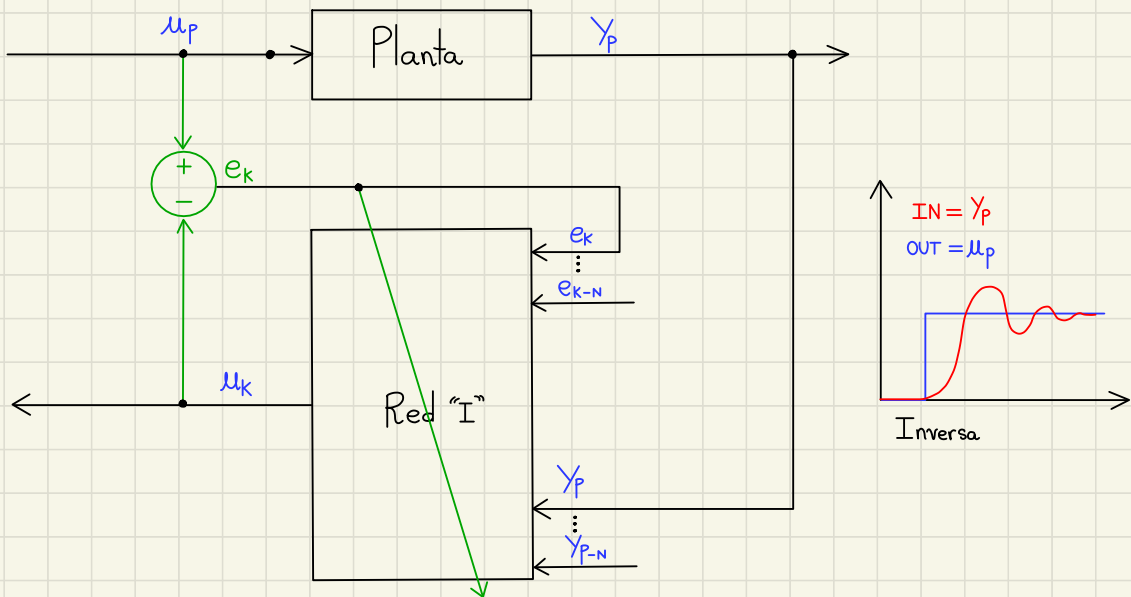
Debido a que ya se comenzó por el primer proceso, continuaré con la misma dinámica de la planta y lo controlare con este sistema integrador ya analizado.

Lo que sigue ahora es comenzar el proceso de identificación NO paramétrica, el cual lo realizaremos a través de una red inversa que logre estimar la entrada de la planta, a través de la salida (y así poder hallar posibles "Uk", según el setpoint deseado)...

A continuación haremos el proceso de identificación NO paramétrica inverso.

IDENTIFICACIÓN NO PARAMÉTRICA INVERSA

Se llama “Inversa”, porque lo que se está aprendiendo la red, no es la salida del sistema, sino la entrada.



Este diagrama se eligió de la siguiente forma, para poder en un futuro próximo, implementar un esquema de control inverso...

En la próxima página, se explicará el proceso de forma organizada y detallada...

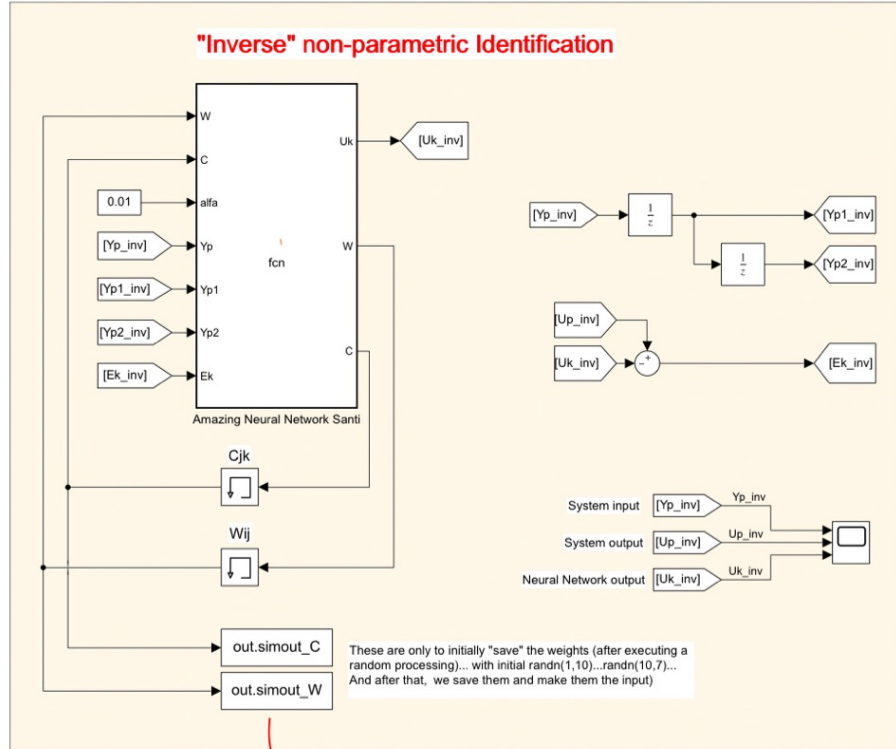
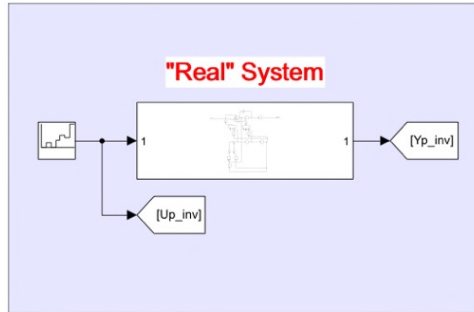
Nota: para este ejemplo, se tomará como entradas a la red inversa las siguientes señales:

y_p : salida planta real
 y_{p1} : salida planta anterior
 y_{p2} : salida planta anterior anterior
 e_k : error actual red ($U_p - U_k$)

Las ecuaciones y el entrenamiento se explicarán luego de mostrar los diagramas.

INVERSE IDENTIFICATION

Santiago Garcia Arango



Nos sirven para "pre-entrenar" la red que se utilizará en el control inverso.

El bloque de simulink para el entrenamiento se realiza de la siguiente forma:

```

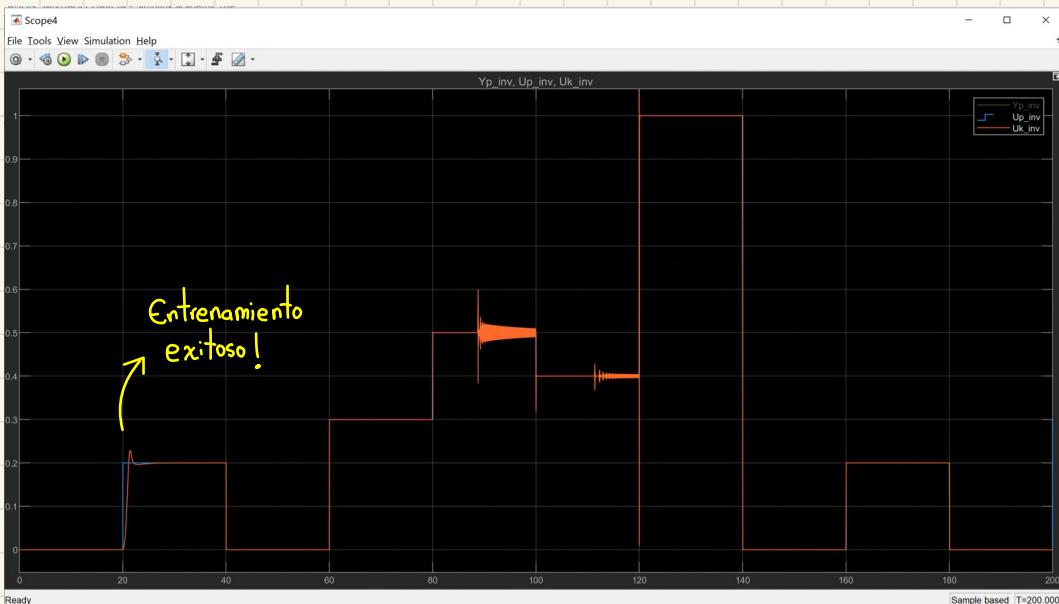
8  % REMARK: This processing is based on an "Inverse Identification"
9  % -----
10
11 function [Uk, W, C] = fcn(W, C, alfa, Yp, Yp1, Yp2, Ek)
12
13     activation_function_1 = "tanh";
14     activation_function_2 = "none";
15
16     X = [Yp Yp1 Yp2]; % Create "entries" vector
17     Aj = W*X'; % Add hidden layer
18
19     if (activation_function_1 == "tanh")
20         H = tanh(Aj);
21     end
22
23     Ak = C'*H; % Add output layer
24
25     if (activation_function_2 == "none")
26         Uk = Ak;
27     end
28
29     C = C+alfa*Ek*H; % Update weights of C
30
31     if (activation_function_1 == "tanh")
32         H2 = sech(Aj).^2;
33     end
34
35     W = W+alfa*Ek*(C.*H2)*X; % Update weights of W
36
37 end

```

Función Activación
Interna es
Tangente Hiperbólica

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

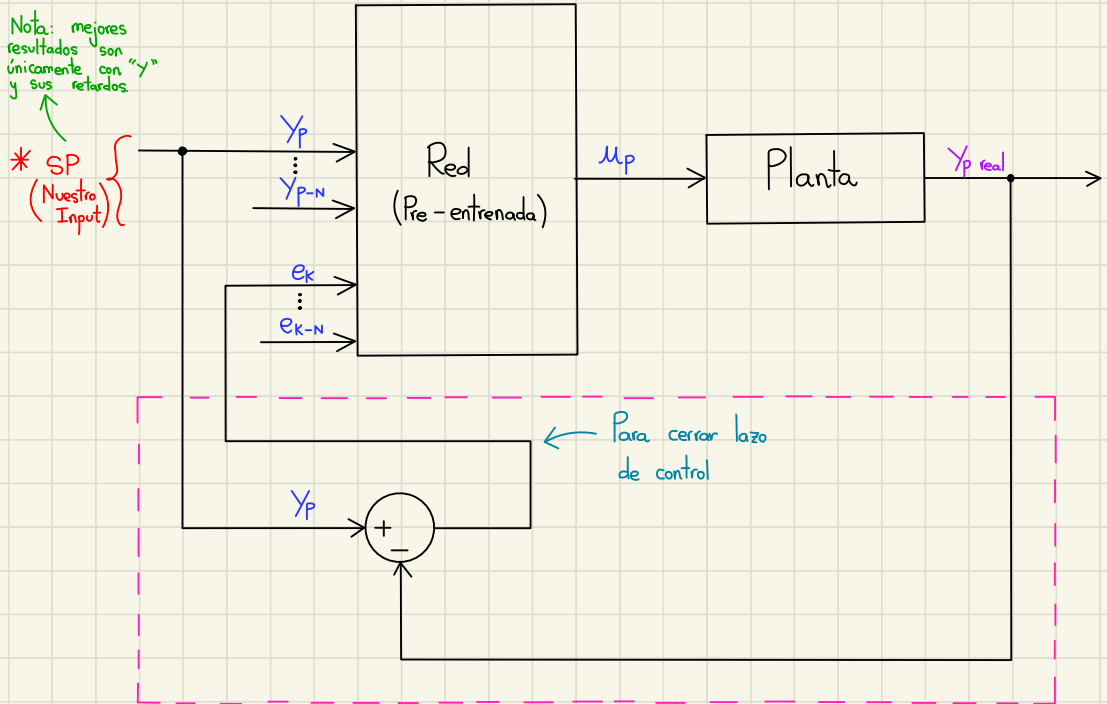
derivada es
"sech(x)²"



CONTROL INVERSO

Ahora que ya tenemos forma de hacer identificaciones no paramétricas inversas, el siguiente proceso a realizar es efectuar el control de un sistema con ayuda de esta identificación inversa.

El objetivo es entrenar una red inversa en primera instancia (que a partir de señal de salida planta [Y_p], logre reconocer la entrada del sistema [U_p]), y teniendo esto ya tuneado, se procede a ingresar señal cuadrada como “setpoint” de [$Y_{deseada}$], y el sistema debe ser capaz de encontrar la señal [U_p] requerida, para ingresarla al sistema real (esto con ayuda de la red ya entrenada y los pesos de la misma fijos (o variables)).



Si se agrega este feedback, es “Control Inverso Adaptativo”

Nota: en los resultados experimentales en Simulink, los mejores obtenidos son los que NO involucran el error

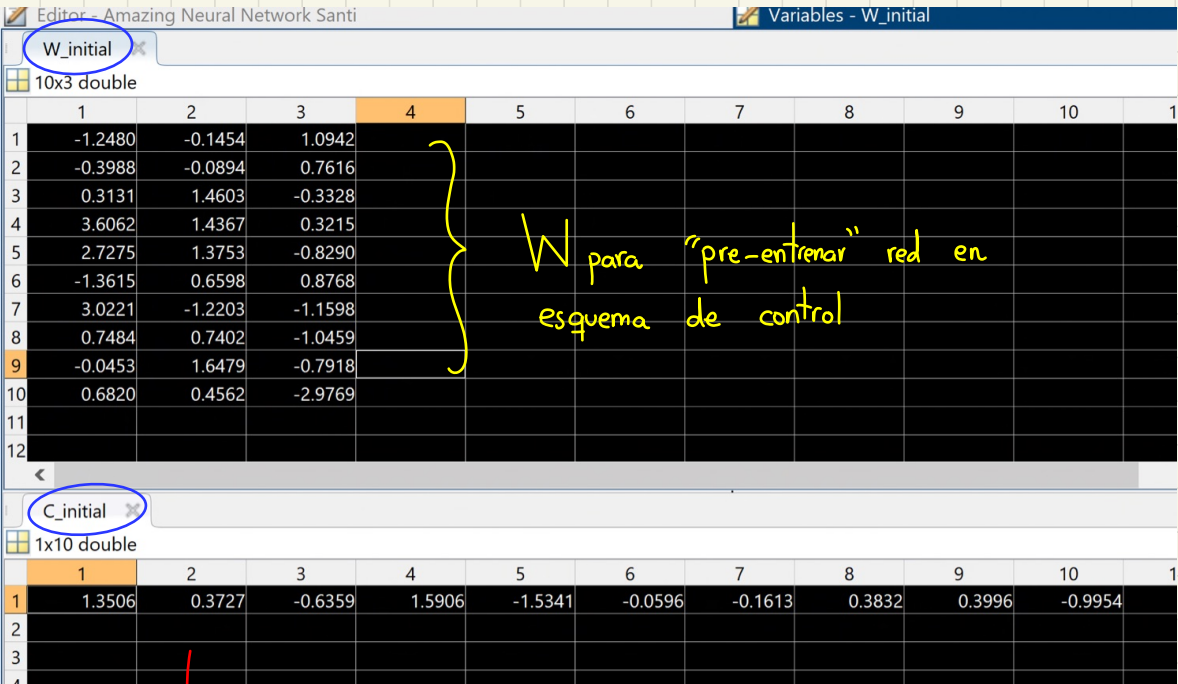
Una aclaración muy importante para todos los sistemas de control (especialmente cuando se planeen implementar en sistemas reales), es que es de suma importancia entender los rangos de acción y saber que, si hay señales que puedan afectar al sistema, se deben mitigar.

Para esto existen dos soluciones:

1. Saturar señales de control.
2. Pre-entrenar el sistema para que comience ya con los pesos estimados de funcionamiento y no vaya a tener riesgo de “entrenarse” y que salgan señales extremas.

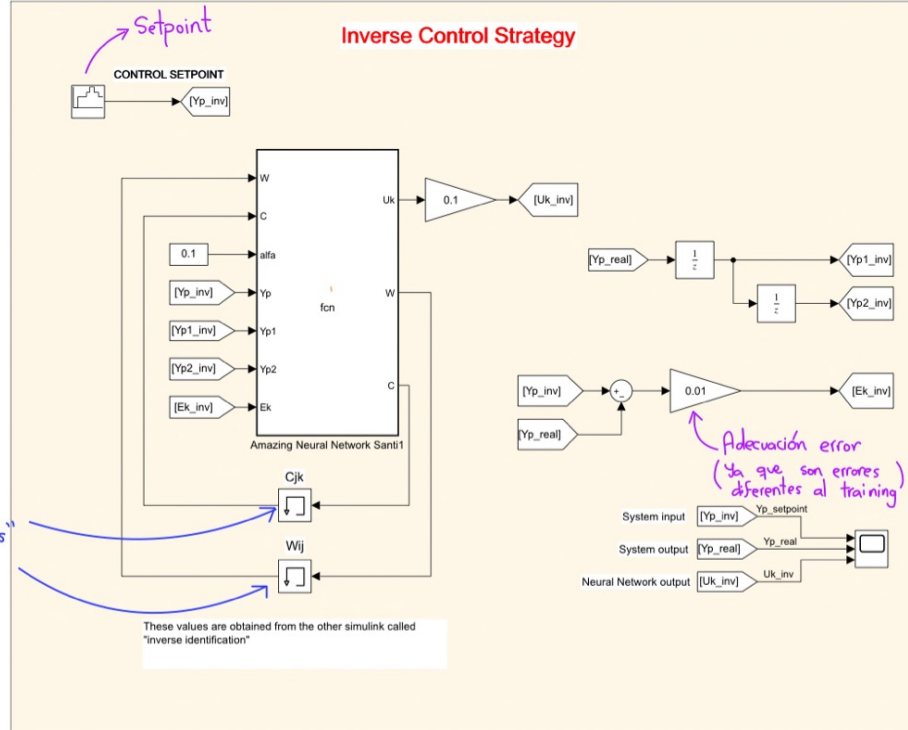
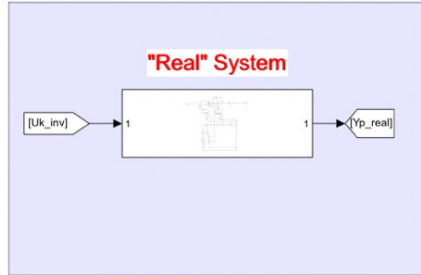
Para este procedimiento, realizaré la segunda opción, pues es más segura y recomendada para la implementación de controladores en la vida real.

Esto se hace obteniendo las últimas configuraciones de las matrices “C” y “W” de la identificación y cargándolas como los valores iniciales de la red que se implementará en el control inverso. Es decir, estas:

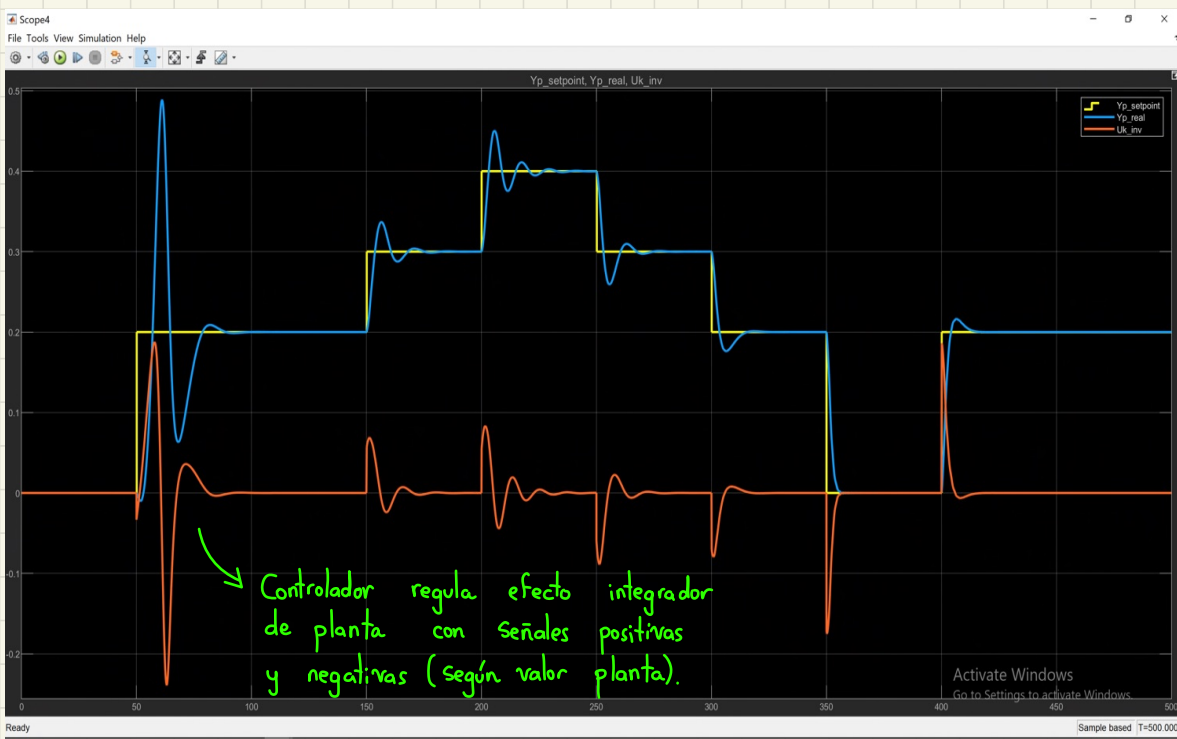


CONTROL

Santiago Garcia Arango



Una vez se implementó el control inverso adaptativo (con un error modificado para garantizar las magnitudes correctas), los resultados son excelentes:



Se puede evidenciar como se desea un setpoint objetivo ($Y_p\text{_setpoint}$ = amarillo), u se ven las señales de acción del controlador inverso ($U_k\text{_inv}$ = naranja) de tal forma que oscilen entre positivo y negativo y logren llevar la señal real de salida ($Y_p\text{_real}$ = azul) al setpoint exacto (incluso con error en estado estable nulo).

Los resultados de este controlador son increíbles, pues con teorías de control clásica, NO hubiese sido posible controlarlo sin antes haber agregado un elemento para contrarrestar el efecto integrador.

Fue un reto genial profe. Muchas gracias.