



PROYECTO FINAL DE CARRERA

*IDENTIFICACIÓN DE UNA MAQUETA DE MOTOR
DE CORRIENTE CONTINUA*

Alumno: Leticia Recarte Mendiburu

Tutor: José Basilio Galván Herrera





INDICE

Capítulo 1: INTRODUCCIÓN	4
<i>OBJETO DE PROYECTO</i>	5
<i>MAQUETA DE MOTOR</i>	8
Capítulo 2: IDENTIFICACIÓN DE SISTEMAS SIMULADOS	11
<i>SISTEMAS DISCRETOS NO RUIDOSOS</i>	12
<i>SISTEMAS DISCRETOS RUIDOSOS</i>	18
<i>AUTOCORRELACIÓN DE ERRORES</i>	27
<i>PERIODO DE MUESTREO</i>	29
<i>SISTEMAS DISCRETIZADOS (I)</i>	30
<i>SISTEMAS DISCRETIZADOS (II)</i>	35
Capítulo 3: IDENTIFICACIÓN DE UN SISTEMA REAL	51
<i>MODELO Output Error</i>	52
<i>MODELO ARX</i>	81
<i>MODELO ARMAX</i>	102
Capítulo 4: CONCLUSIONES	127
<i>BIBLIOGRAFÍA</i>	130





Capítulo 1. INTRODUCCIÓN

1.1 OBJETO DEL PROYECTO

El principal objetivo del proyecto es realizar la identificación de una maqueta de motor de corriente continua. Para llevar a cabo esta identificación vamos a utilizar los modelos dinámicos Output Error, ARX y ARMAX, que contienen diferentes modelos de ruido de forma que podemos estimar el más adecuado para este problema.

Vamos a basar nuestra identificación en técnicas lineales. Debido a que la maqueta presenta una no linealidad, causada por el motor, tipo “zona muerta” tendremos que asegurarnos de trabajar fuera de la zona no lineal. Una vez que se tengan datos de esta zona se tendrá que identificar un punto de trabajo y linealizar alrededor del mismo. Una vez hecho lo anterior se podrá proceder a la identificación del la Función de Transferencia del modelo linealizado.

Para la correcta identificación modelo hay que estudiar el número de polos y ceros y su localización, estimar el ruido resultante y la correlación del mismo para asegurarnos de que no nos estamos dejando sin identificar dinámicas significativas.

Como ya se dijo más arriba, estudiaremos varios modelos dinámicos de sistemas ruidosos: Output Error, ARX y ARMAX. Es sabido que uno de los principales problemas para la identificación viene dado por la elección del periodo de muestreo adecuado (explicaremos este punto más adelante). Para intentar encontrar dicho periodo, para cada modelo estableceremos una serie de periodos de muestreo, con los que se realizarán distintas simulaciones, obteniendo así distintos resultados, para más tarde analizar todos ellos y obtener las conclusiones pertinentes.

Antes de comenzar a trabajar con el motor se va a trabajar con sistemas simulados. Los objetivos que se buscan alcanzar con esta parte del proyecto son:

- Estudiar sistemas discretos, que no entran en el plan de estudios de Ingeniería Técnica Industrial: Como la identificación se hace a través de datos muestreados, es decir del sistema discretizado, se necesita conocimientos de estos sistemas para poder hacer el proyecto.
- Estudiar los problemas de los sistemas discretizados y la influencia del periodo de muestreo.
- Entender bien el proceso de identificación y validación de modelos.



- Estudiar, con una estimación del modelo de la maqueta, los problemas de discretización y ruido que nos podemos encontrar cuando trabajemos con el sistema real.
- Siguiendo el estudio del punto anterior, estimar el mejor rango valores del periodo de muestreo para la identificación de la maqueta.
- Familiarizarse con el toolbox de identificación de Matlab.
- Crear una serie de problemas auxiliares para la identificación de la maqueta: generadores de señal, programas para la validación de modelos, etc.

Los pasos que se van a seguir son progresivos, esto es, se comenzarán con los ejercicios más sencillos en los que el principal objetivo será el propio manejo del programa y más tarde se pasará a ejercicios más complicados, jugando con los sistemas discretos y con los continuos, que estén más enfocados al trabajo que hay que realizar con la maqueta.

Por último se procederá a la identificación de la maqueta. Para ellos se procederá según el siguiente esquema:

1. **Labview. Estudio de sistemas digitales y técnicas de identificación.** Como paso previo se necesita saber utilizar un programa de gestión de datos en tiempo real. En este caso se va a utilizar el Labview, utilizando programas aportados por la profesora Pérez-Ilzarbe. Lo usaremos para obtener los datos con los que proceder a la identificación.
2. **Experimentos.** Se realizarán una serie de experimentos de los cuáles podamos obtener distintas señales de salida útiles para la posterior identificación. Para ello, insertaremos varias señales de entrada a la maqueta y obtendremos diferentes salidas, observando de este modo el comportamiento del motor.
3. **Linealización.** Los datos de entrada y salida se utilizarán para identificar un punto de trabajo fuera de la zona muerta del motor y así proceder a la linealización de los mismos.



4. **Modelización.** Lo que se va a hacer es estudiar qué posible modelo puede adecuarse a los datos obtenidos. Este modelo se traducirá a través de una Función de Transferencia con unos parámetros que habrá que fijar a posteriori, así como con un modelo dinámico (como ya se dijo vamos a usar Output Error, ARX y ARMAX).
5. **Identificación:** Una vez que tenemos los datos (entrada y salida) linealizados y que hemos decidido qué modelo vamos a utilizar, se procede a la identificación de los parámetros del modelo. Con lo que obtenemos una Función de Transferencia del sistema, y en su caso, del ruido.
6. **Validación:** a través de una señal de entrada cualquiera, pero diferente a la usada en la identificación, debemos ser capaces, a través de la Función de Transferencia, de conocer aproximadamente la señal de salida que se va a dar. Si nuestra simulación consigue reproducir, salvo por el ruido, la señal de salida, se considera que la identificación será obtiene.

Siguiendo minuciosamente esta serie de pasos, podremos identificar el modelo linealizado del motor y, a partir de él, podremos estimar la señal de salida para cualquier señal de entrada.

Con estos resultados daremos por concluido nuestro proyecto, pudiendo definir la localización de los polos y ceros el modelo dinámico que se corresponde a la maqueta incluyendo el mejor modelo de estimación del ruido.



1.2 MAQUETA DE MOTOR

La maqueta se compone de tres partes diferenciadas:

- El sistema de control principal
- El programa de visualización y control de Labview
- El sistema electro mecánico con sensores y freno

1. Sistema de control principal

Para la realización del Control Principal se actúa frente a un panel frontal que tiene las siguientes características:

- Entrada al amplificador de potencia (PWM) que acciona al motor de corriente continua.
- Salidas analógica de la velocidad (tacogenerador) y posición del motor.
- Salida de la señal de referencia del motor (controlada por Labview)
- Comparadores (restadores) para obtener la señal del error $R(s)-Y(s)$.
- Salidas de $\pm 10V$ dc. Para alimentar los amplificadores operacionales del circuito eléctrico del compensador.

Gracias a este panel frontal tenemos posibilidad de variar las conexiones según del esquema de control deseado, lazo abierto o lazo cerrado.

En nuestro caso, dicho panel frontal no es utilizado. La única función que tiene es proporcionar a la maqueta la señal para que ésta pueda funcionar con normalidad.

2. Programa de control y visualización Labview

Genera la señal de referencia de la salida del sistema de control principal y la visualiza junto a la señal de velocidad del motor. El visualizador de señales dispone de dos cursores, uno para cada señal, que pueden desplazarse a lo largo de cada señal pinchando sobre ellos y arrastrándolos mediante el ratón.

3. Sistema electro mecánico

La maqueta que se va a analizar (*ver Figura 1*) consta de un motor de precisión de corriente continua, que girará en un sentido u otro dependiendo de la tensión que se le suministre, conectado a un reductor 15:1. Este reductor permite operar con mayor precisión y proporciona mayor par al motor para mover la carga inercial de 2.5 Kg que



lleva acoplada. La carga lleva acoplada un freno de partículas magnéticas que proporciona un par de frenado ajustable.

El control de velocidad del eje motor se hace a través de un tacogenerador de constante de tensión 10V/1000 r.p.m. acoplado al motor. La posición del eje motor y la posición del eje de la carga inercial se obtienen a través de dos encoders.

La posición del eje motor se mide con un codificador óptico incremental (encoder) de 516 pulsos por revolución. La posición de la carga la mide otro codificador óptico incremental, en este caso de 1000 pulsos por revolución.

En las simulaciones que se van a realizar no se va a utilizar el freno de partículas magnéticas. Lo único que interesa es obtener la salida que proporciona el motor para después poder analizarla

El sistema electro mecánico es el que se muestra en la imagen inferior:

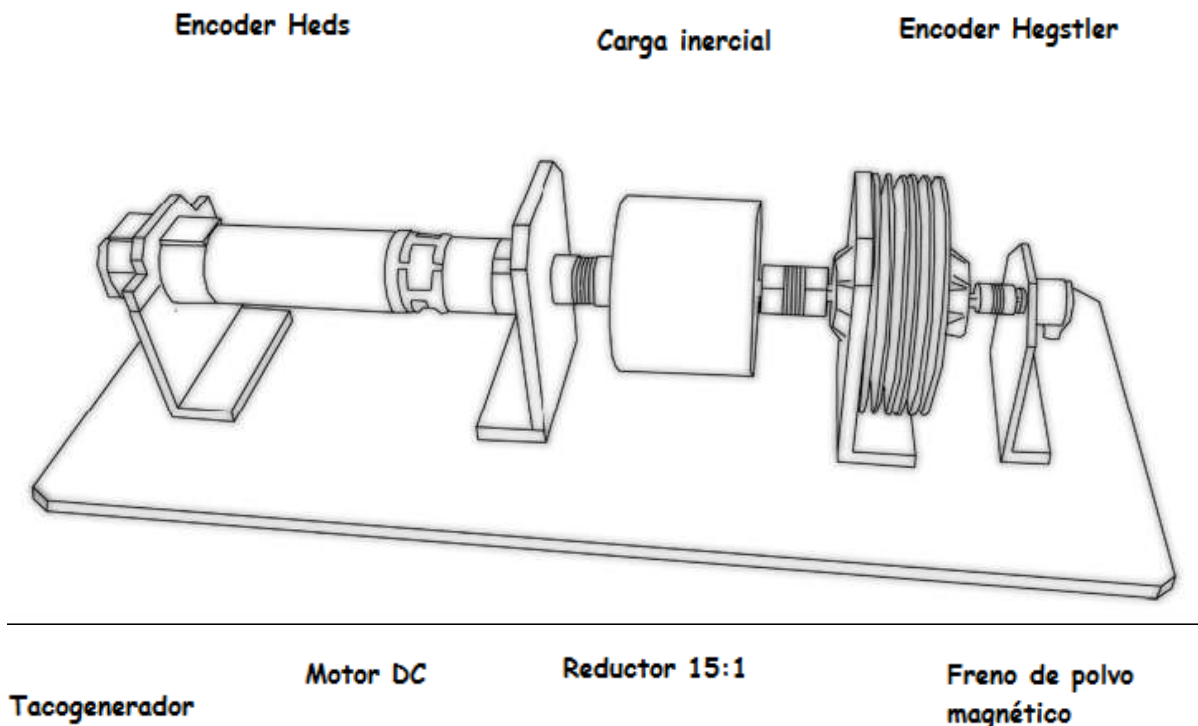


Figura 1: Sistema electro mecánico utilizado.



El esquema de la presente memoria es el siguiente:

- En el capítulo 2 voy a presentar los resultados de simulación: tanto con ruido como sin ruido, para sistemas discretos y discretizados, obteniéndolo con las últimas simulaciones información útil para la identificación del caso real.
- En el capítulo 3 presento los resultados de la identificación para el caso real. Estudiando los tres modelos dinámicos ya indicados y cuál es el más conveniente para esta maqueta.
- En el capítulo 4 presento las conclusiones y propongo futuras líneas de trabajo.



Capítulo 2. IDENTIFICACIÓN DE SISTEMAS SIMULADOS

Antes de empezar a trabajar con la maqueta se va a hacer un estudio detallado en simulación con los siguientes objetivos: aprender las nociones básicas necesarias de sistemas digitales ya que no se estudian estos sistemas en la titulación; por la misma razón, estudiar la base de identificación de sistemas y algunos de los modelos más típicos (Output Error, ARX y ARMAX); familiarizarse con el entorno de Simulink y el toolbox de identificación de Matlab; desarrollar algunos programas auxiliares para la identificación de la maqueta, como por ejemplo el generador de entradas, obtener información sobre los mejores periodos de muestreos para dicha identificación partiendo de estimaciones previas de su modelo. Para conseguir estos objetivos se van a seguir una serie de pasos, con las que poco a poco se irán aprendiendo los conceptos claves para el desarrollo del análisis de la maqueta.

En primer lugar se realizarán identificaciones con señales sin ruido. Posteriormente se identificarán señales con ruido, que asumiremos blanco y de medida durante toda la simulación. Para finalizar habrá que tener en cuenta que los datos del motor, aunque discretos, tienen como fuente un sistema continuo, se dice que es un sistema discretizado. Esto presenta problemas como la elección del periodo de muestreo, que nos marca la frecuencia con la que se obtienen los datos, así como las del paso de sistema discreto identificado (una Función de Transferencia digital) al sistema continuo identificado (una Función de Transferencia continua). Es decir, teniendo los datos en forma discreta los cálculos se facilitan bastante y cuando la identificación se haya realizado se pasan los datos a un sistema continuo, conociendo así cuales son realmente las características del motor. Todo ello se realizará con los toolboxes de identificación y control de Matlab.

Vamos a desarrollar esta parte del trabajo a través de los siguientes pasos:

- Identificación de sistemas discretos sin ruido: este paso servirá para familiarizarse con el entorno de Simulink, conocer el modelo Output Error, empezar a trabajar con el toolbox de identificación de Matlab y entender cómo se valida un sistema.
- Identificación de sistemas discretos con ruido: en este paso se profundiza en el estudio anterior y nos permitirá ver cómo se degrada la identificación al aumentar el nivel del ruido.



- Identificación de un sistema discreto obtenido a partir de uno continuo: en este paso se realiza para estudiar la diferencia entre sistemas discretos y discretizados, la dificultad del paso entre modelos, cómo se realiza dicho paso con el toolbox de control de Matlab y cómo se degrada la identificación del sistema continuo al añadir ruido. También se estudia el problema de la elección de periodo de muestreo.
- Identificación de un sistema continuo: este paso es una profundización del anterior y es el que prepara para la identificación de la maqueta. Se estudia el uso de retenedores y muestreadores para discretizar el sistema, el problema de la elección del periodo de muestreo y la influencia del mismo en la identificación de sistemas ruidosos, y la degradación de dicha identificación al variar la magnitud del ruido. Todo esto usando modelos aproximados de la maqueta para que las conclusiones nos sirvan en el siguiente capítulo.

2.1 IDENTIFICACIÓN DE SISTEMAS DISCRETOS NO RUIDOSOS

Para proceder a la identificación necesitamos un programa que genere entradas adecuadas. El programa que se ha generado nos permite la libertad de fijar el número de muestras de la entrada (cuando trabajamos con sistemas discretos) o bien el periodo de muestreo y el tiempo de funcionamiento del experimento. La señal de entrada generada tiene forma de escalones pudiendo fijar:

- La señal media, es decir, alrededor de la cual se definen los escalones (veremos que esto sea útil cuando tengamos que evitar no linealidades de la maqueta: va en el capítulo 3.1)
- El tiempo máximo de la prueba (o el número de muestras). La altura máxima de los escalones (positiva o negativa) a partir del valor medio.
- El tiempo máximo (o número de muestras máximas) de cada escalón.
- Se utilizan para generar valores aleatorios con ese máximo. De esta forma se consigue todo tipo de escalones (*ver el punto 3.1*) tanto en anchura como en altura. Variando estos valores podemos generar señales que permitan estudiar el permanente como otras que se acerque a un ruido cuasi-blanco.



- Por último, también fijaremos el tiempo en el que se mantiene una señal continua con el valor de la señal media. Esto nos permitirá identificar la ganancia o el punto de trabajo cuando hay que linealizar el sistema.

Los parámetros del programa generador de entradas son las siguientes:

$$\begin{aligned}
 t_{fin} &= 1000 \rightarrow \text{tiempo de finalización de la prueba} \\
 t_c &= 100 \rightarrow \text{tiempo máximo en el que sigue una entrada continua} \\
 y_c &= 2 \rightarrow \text{amplitud máxima} \\
 t_1 &= 800 \rightarrow \text{tiempo en el que acaban los escalones}
 \end{aligned}$$

El sistema que se va a analizar es:

$$G(z) = \frac{z - 0.8}{z^2 + z + 0.25} \cdot 0.01$$

Tenemos una Función de Transferencia sencilla, ya que lo que se pretende es aprender el manejo de los distintos toolboxes de Matlab.

Tras haber obtenido las señales de entrada creamos en Simulink un esquema de las siguientes características:

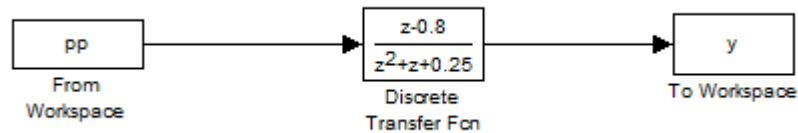


Figura 2: representación de un sistema discreto no ruidoso

El comando “*From Workspace*” indica que proporcionamos al simulador la señal situada en el espacio de trabajo que esté referenciado con el nombre pp. Esta entrada se introduce en la Función de Transferencia discreta y se obtiene una señal de salida que la pasamos al espacio de trabajo gracias a la referencia de “*To Workspace*”.



Por último, se tiene que realizar un programa que realice la identificación del sistema a partir de los datos procedentes del simulador. Vamos a trabajar con el modelo ruido Output Error. Para realizar la identificación se deben seguir una serie de pasos en Matlab:

1. Generar los datos para la identificación a través del comando “iddata”, proporcionando la salida obtenida, la entrada y el periodo de muestreo, que en este caso no es relevante al ser un sistema discreto y se puede tomar como defecto 1.

$$Z = iddata(salida', entrada', T)$$

2. Realización de la identificación (la correspondiente al Output Error): para ello se deben señalar el número de polos y ceros que se estiman pueda tener el sistema y el retraso estimado.

$$M = oe(Z, [c + 1 \ p \ r])$$

Donde “c” es el número de ceros, “p” es el número de polos y “r” corresponde al retardo.

3. Paso del modelo a Función de Transferencia discreta. Se obtienen los parámetros que se utilizan para la creación de la Función de Transferencia a través de la función “polydata”.

$$[A, B, C, D, F] = polydata(M)$$

4. Creación de las funciones de transferencia del sistema y del ruido (en caso de que lo tenga). Para crear la Función de Transferencia de cada modelo de los obtenidos con “polydata” tenemos que usar determinados vectores.

- Output Error: Hay que utilizar los vectores B y F como indica la ecuación.

$$y(t) = \frac{B(q)}{F(q)} u(t - nk) + e(t)$$

- ARX: Tenemos que utilizar los vectores B y A como se indica en la ecuación.

$$y(t) = \frac{B(q)}{A(q)} u(t - nk) + \frac{1}{A(q)} e(t)$$



- ARMAX: Los vectores a utilizar son el B, A y C como se indica en la siguiente ecuación:

$$y(t) = \frac{B(q)}{A(q)}u(t - nk) + \frac{C(q)}{A(q)}e(t)$$

5. Validación del modelo identificado. Para proceder a la validación de un sistema se proporciona la misma señal de entrada, diferente a la utilizada en la identificación, al sistema y a la Función de Transferencia obtenida (modelo obtenido). Para proceder a su comparación.
6. Cálculo del error. Se comparan las dos señales, obtenidas en el punto anterior, la proporcionada por el sistema y la de la Función de Transferencia. Para hacer la comparación usaremos el error cuadrático.
7. Estimación del ruido: en su caso, a partir del error de identificación, asumiendo que todo él es debido al ruido, podemos proceder a una identificación del mismo dependiendo del modelo que hayamos usado. Una estimación de la bondad de la identificación viene dada por lo cerca que esté este ruido identificado de una señal blanca: es decir, asumiendo ruido blanco, que sea autodecorrelacionado y de media nula. En este caso, debido a las características del sistema (sin ruido) no daremos este último paso (*volvemos a este punto en el apartado 2.3*).

Imponemos que el número de polos es 2 y que tiene un único cero. En otras pruebas hechas añadiendo más polos y ceros de los necesario comprobamos que se eliminaban mutuamente o aparecían como muy dominados y se podían eliminar.

En el caso de las señales no ruidosas no se tendrán muchos problemas, ya que los polos y ceros aparecen claramente superpuestos y se eliminarán fácilmente. En otros casos, como cuando al sistema se le añade ruido del exterior no es tan claro si se deben eliminar los polos y ceros, ya que no quedan perfectamente superpuestos. Veremos cómo nos encontramos con este problema cuando estudiemos la maqueta real.

Es de esperar que el error cometido en la identificación sea muy pequeño, ya que no hay ningún tipo de ruido externo al sistema. Por lo que en la identificación no se deberían cometer errores. La salida del sistema y la de la Función de Transferencia identificada será la misma en el proceso de validación. También



podemos decir que la Función de Transferencia identificada debe ser muy parecida a la Función de Transferencia inicial.

Por lo tanto,

$$\frac{0.01z - 0.008}{z^2 + z + 0.25}$$

Como hemos podido comprobar la Función de Transferencia identificada es la misma que la Función de Transferencia inicial.

En este caso, el error en la validación tiene una magnitud de $\text{err}=1.1434\text{e}-033$, prácticamente nulo, como suponíamos. Como se puede observar en la siguiente figura la diferencia entre la salida procedente del sistema (azul) y la procedente de la Función de Transferencia (roja) son prácticamente iguales. En el eje de abscisas se coloca el tiempo y en el de ordenadas se marca la salida.

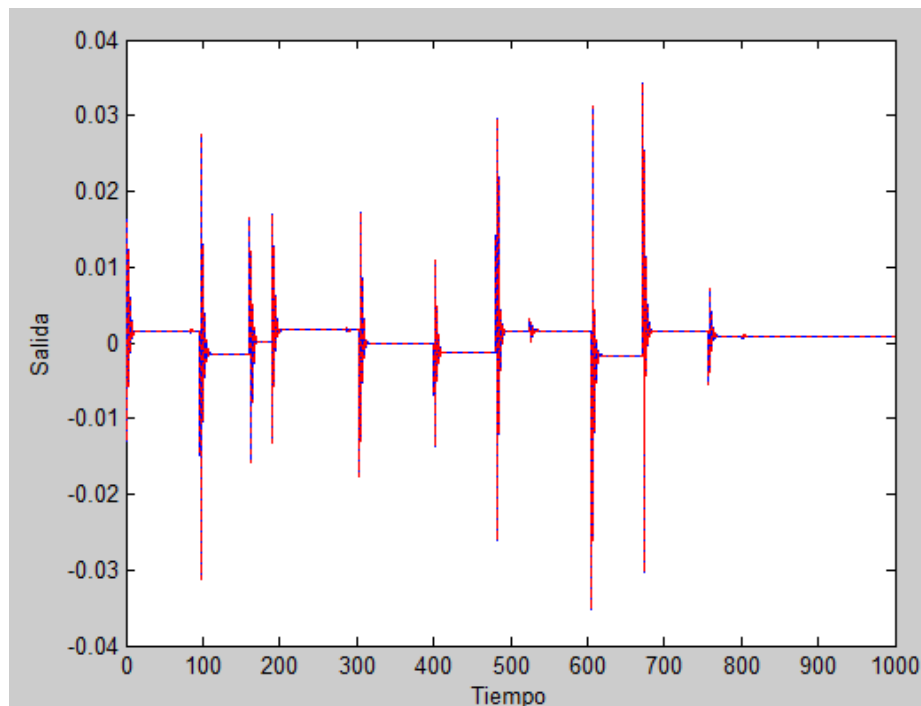


Figura 3: Representación de las salidas

Como no se distingue nada de la *Figura 3*, vamos a realizar un “zoom” de alguna zona para observar la diferencia que existe entre ambas señales, obteniendo la *Figura 4*.

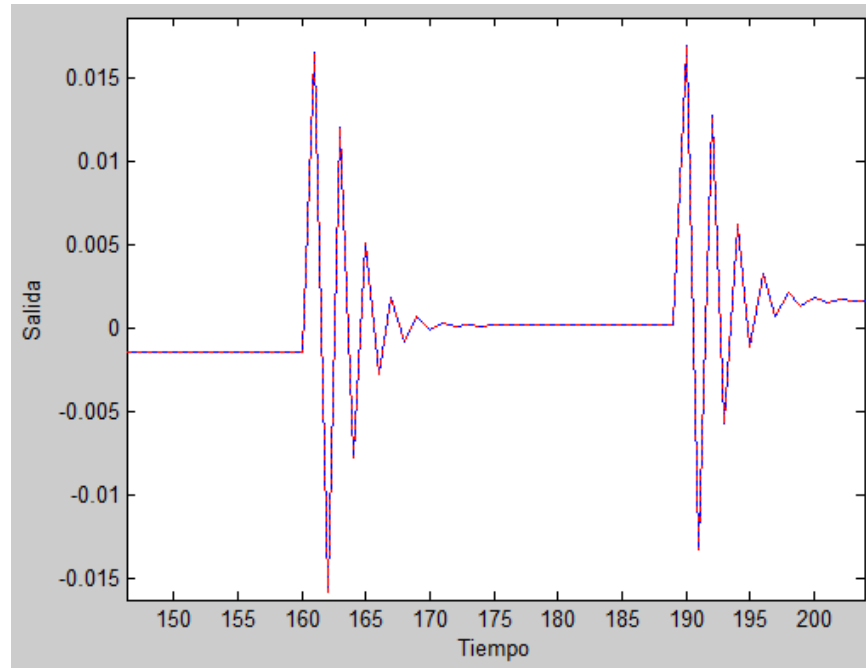


Figura 4: Zoom de la Figura 3

Se puede ver que la diferencia entre ambas señales es tan pequeña que no la podemos apreciar, por eso en la siguiente gráfica vamos a comprobar la diferencia que existe entre ellas.

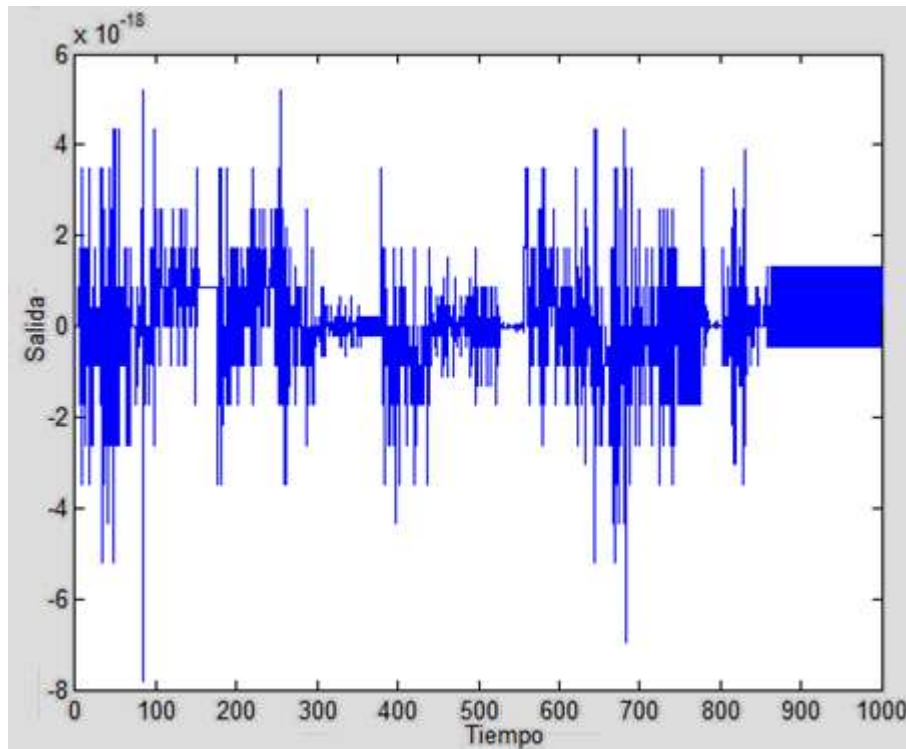


Figura 5: Representación del error



Como ya hemos dicho antes, la diferencia entre las dos señales es muy pequeña y lo podemos comprobar de la *Figura 5*, ya que la diferencia entre las dos ronda un valor de 10^{-18} , que está en el umbral de precisión de Matlab.

En todos los casos que se van a presentar a continuación se va a seguir el mismo procedimiento presentado aquí, la única diferencia será que las ecuaciones cambiarán en algunos términos, dependiendo de las características de cada ejercicio, el tipo de error...

2.2 IDENTIFICACIÓN DE SISTEMAS DISCRETOS RUIDOSOS

En este ejercicio, se da un paso más en la identificación del sistema, ya que, a diferencia del caso anterior, en el que al sistema no se le proporcionaba ningún tipo de ruido externo, en éste sí que se le va proporcionar. El esquema quedará de la siguiente manera:

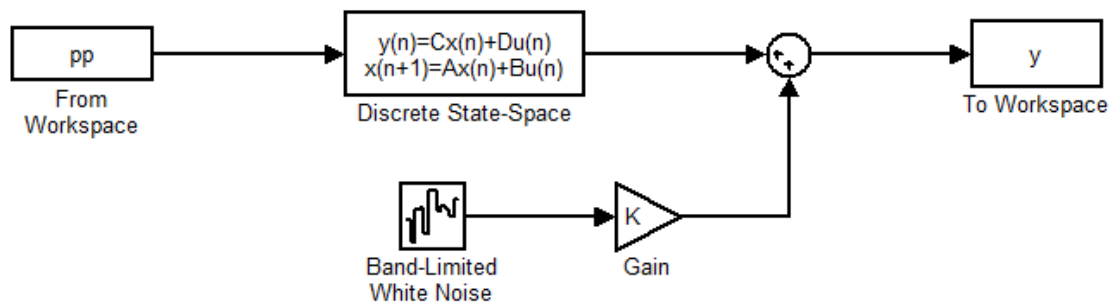


Figura 6: Esquema del sistema discreto ruidoso

La generación de los pulsos de entrada, la identificación y los comandos de conexión entre el simulador y el espacio de trabajo serán los mismos que en el caso anterior. Como ya se ha indicado anteriormente se le suma al sistema un ruido blanco externo usando la subrutina de Simulink “*Band-Limited White Noise*”. A la salida del generador de ruido se coloca ganancia variable que permita amplificar o disminuir la señal del ruido, obteniendo así distintas señales de salida. Si se pone $K=0$ obtenemos el caso anterior, es decir el sistema no ruidoso. Antes de cada simulación se le debe dar un valor diferente al parámetro K para poder definir la amplitud del ruido. Para cambiar la salida del generador de ruido tenemos que aportarle diferentes semillas (en este caso la semilla debe ser un número entero) lo hacemos generando una señal aleatoria que se redondea a un número entero: “*round(rand()*100000)*”



Para que la identificación se acerque más a un problema real se utilizó un sistema aportado por el director del proyecto pero desconocido por mí.

La identificación se hace de una forma más complicada que en el caso del ejercicio anterior, ya que ahora sí que hay que analizar el error. Se ha supuesto que el número de polos es 3 y el número de ceros es 2, así el análisis cambiará con respecto al realizado en el apartado anterior. En ejercicios de este tipo hay que tener cuidado a la hora de jugar con la relación entre el número de polos y ceros y el error que se comete, ya que al proporcionarle al sistema ruido del exterior el error varía considerablemente. Si el ruido es muy elevado no se pueden identificar correctamente los polos y los ceros, por ello no tendríamos posibilidad de eliminar unos con otros, porque el propio Matlab coloca los ceros y los polos en lugares que no son adecuados. Para analizar en mejor medida el error que se comete se creó un bucle que procedía a generar datos para la identificación, a identificar y a validar el sistema identificado en cada paso del mismo: Así de forma automática podíamos obtener gran cantidad de resultados para proceder a su comparación y elegir el mejor de los mismos.

Para la realización del bucle se crea un programa, donde se impone el número de simulaciones que se tienen que dar, en este caso es de 50. Cuando termina la simulación completa en el espacio de trabajo se indica el término del error más pequeño que se ha producido en todas las simulaciones y el sistema que lo obtuvo.

Se realizaron distintas simulaciones variando en cada caso la amplitud del error lo que se conseguía variando la ganancia K del esquema de Simulink (*Figura 6*). Para cada una de las ganancias, se simula cinco veces el bucle obteniendo así distintos valores de error mínimo y distintas Funciones de Transferencia identificadas. La Función de Transferencia de la que partimos fue dada en Espacios de Estado, teoría no estudiada en la titulación, con lo me quedaba ocultar la función de transferencia.

$$F(z^{-1}) = \frac{(1 - 0.6z^{-1})^3}{(1 - 0.9z^{-1})^2 \cdot (1 - 0.8z^{-1})} z^{-1}$$



Las matrices son las siguientes:

$$A = \begin{pmatrix} 2.7 & -1.21125 & 0.722 \\ 2 & 0 & 0 \\ 0 & 0.5 & 0 \end{pmatrix}$$

$$B = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

$$C = \begin{pmatrix} \frac{1}{160} & \frac{-0.6}{160} & \frac{0.36}{160} \end{pmatrix}$$

$$D = 0$$

$$F(z) = \frac{(z - 0.6)^2}{160 \cdot (z - 0.95)^2 \cdot (z - 0.8)}$$

Nótese que la ganancia es 2.

Los datos obtenidos suponiendo tres polos y dos ceros son los siguientes:

2.2.1 K=0,05

Los errores obtenidos y la media de ellos son:

K=0,05	
Simulación	Error cuadrático
1	0,1254
2	0,1234
3	0,1269
4	0,1207
5	0,1191
Media	0,12882

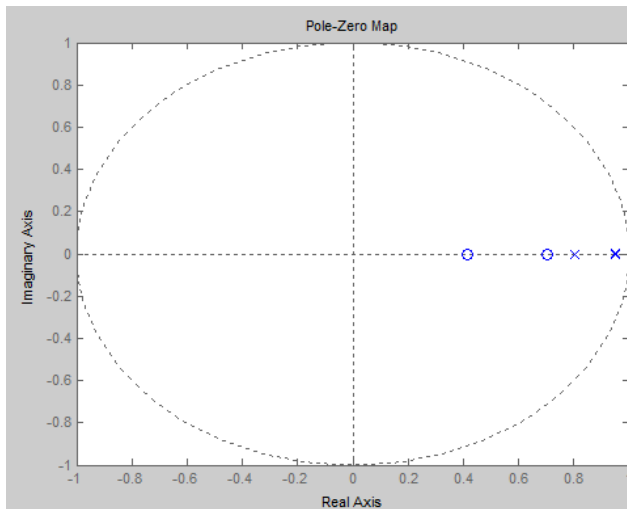
Tabla 1: valor del error cuadrático

El mejor caso que hemos obtenido es cuando el error cuadrático es de 0.1191, ya que es el menor que hemos obtenido. Dicho caso es que vamos a analizar. La Función de Transferencia discreta que le corresponde es:



$$\frac{0.00672 z^2 - 0.008256 z + 0.002882}{z^3 - 2.628 z^2 + 2.285 z - 0.6565}$$

Los polos y ceros obtenidos son:



```
p =
    0.9503 + 0.0036i
    0.9503 - 0.0036i
    0.8023

z =
    0.7045
    0.4123
```

Con ganancia 0.0044.

La gráfica de la salida respecto de la entrada es:

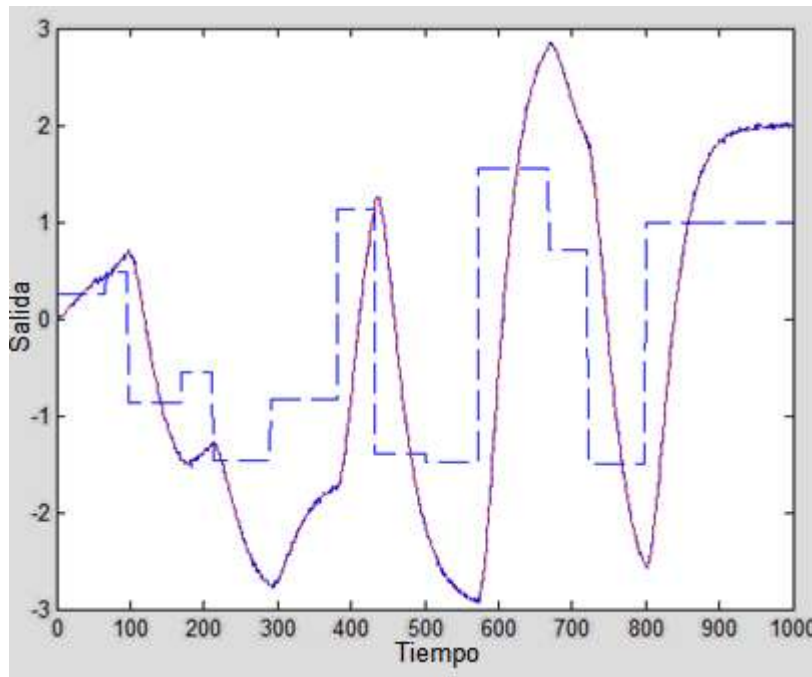


Figura 7: Representación de salidas frente a la entrada



Como se ve, se ha identificado muy bien a los polos pero se falla con los ceros.

2.2.2 K=0,1

Los errores obtenidos y la media de ellos son:

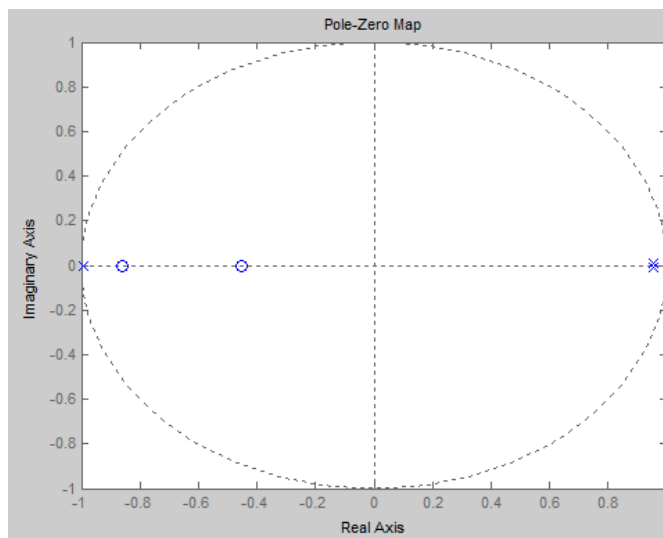
K=0,1	
Simulación	Error cuadrático
1	0,4755
2	0,5217
3	0,49
4	0,4692
5	0,4892
Media	0,12882

Tabla 2: valor de error cuadrático

El mejor caso que hemos obtenido es cuando el error cuadrático es de 0.4692, ya que es el menor que hemos obtenido. Dicho caso es que vamos a analizar. La Función de Transferencia discreta que le corresponde es:

$$\frac{0.003432 z^2 + 0.004508 z + 0.001337}{z^3 - 0.9114 z^2 - 0.9862 z + 0.9022}$$

Los polos y ceros obtenidos son:



$$p =$$

$$\begin{aligned} & -0.9940 \\ & 0.9527 + 0.0096i \\ & 0.9527 - 0.0096i \end{aligned}$$

$$z =$$

$$\begin{aligned} & -0.8613 \\ & -0.4522 \end{aligned}$$



En este caso se vuelve a identificar bien los polos dominantes, pero aparece uno negativo lo que significa que debido a ruido, identifica al sistema como pulsante. Vuelve a fallar en los ceros.

La gráfica de la salida respecto de la entrada es:

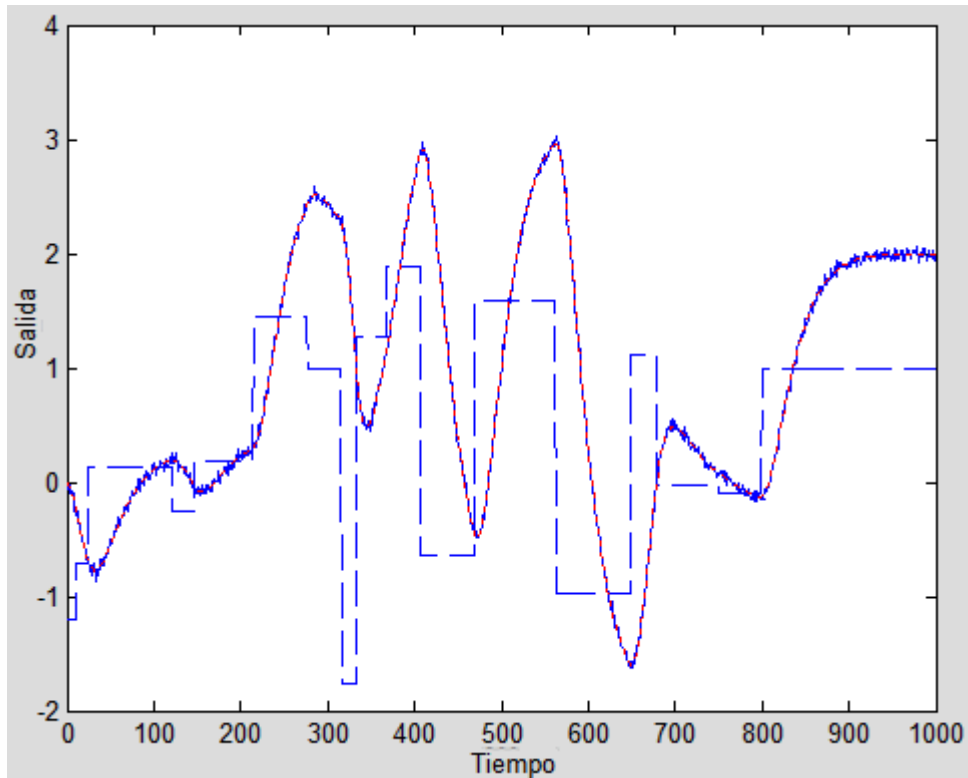


Figura 8: Representación de salidas frente a la entrada

2.2.3 $K=0,2$

Los errores obtenidos y la media de ellos son:

K=0,2	
Simulación	Error cuadrático
1	1,9743
2	1,991
3	2,1905
4	1,933
5	1,9217
Media	2,0021

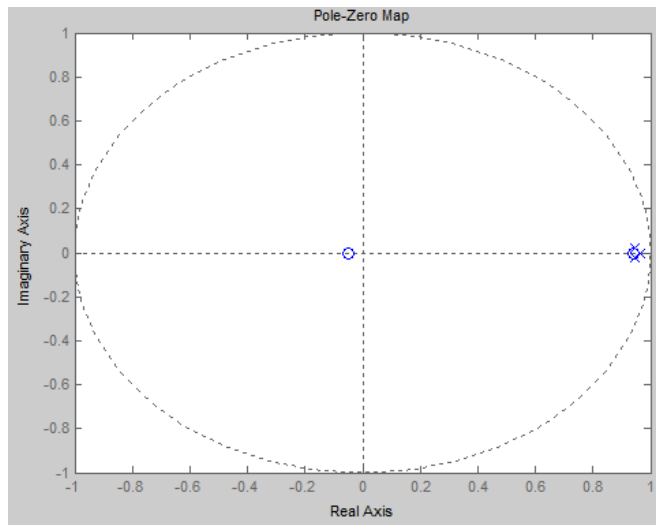
Tabla 3: valor de error cuadrático



El mejor caso que hemos obtenido es cuando el error cuadrático es de 1,9217, ya que es el menor que hemos obtenido. Dicho caso es que vamos a analizar. La Función de Transferencia discreta que le corresponde es:

$$\frac{0.004246 z^2 - 0.003793 z - 0.0001961}{z^3 - 2.855 z^2 + 2.718 z - 0.8624}$$

Los polos y ceros obtenidos son:



p =
 0.9643
 0.9454 + 0.0245i
 0.9454 - 0.0245i

 z =
 0.9424
 -0.0490

La gráfica de la salida respecto de la entrada es:

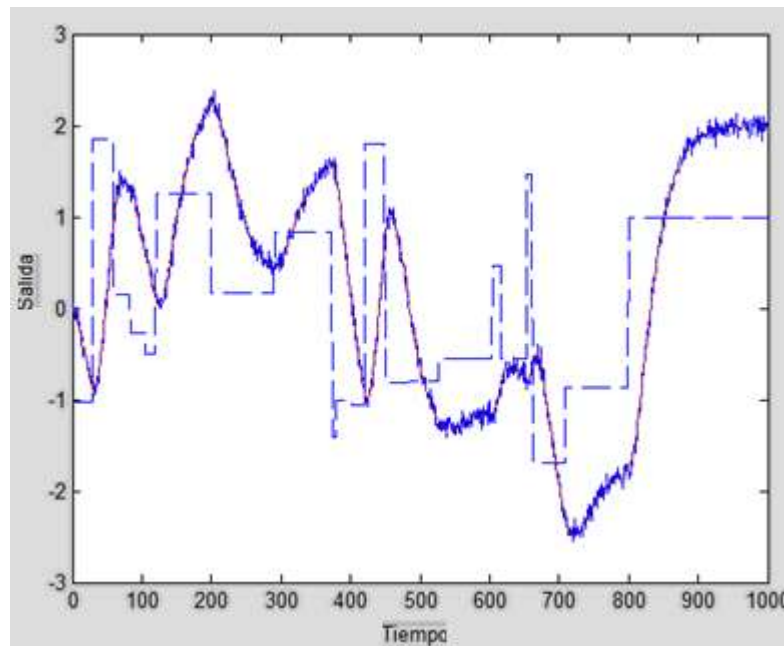


Figura 9: Representación de las salidas frente a la entrada



En este caso identifica aproximadamente los dos polos dominantes, pero prácticamente elimina un polo y un cero. Es como si no fuera capaz de identificar el polo dominante.

2.2.4 K=0,5

Los errores obtenidos y la media de ellos son:

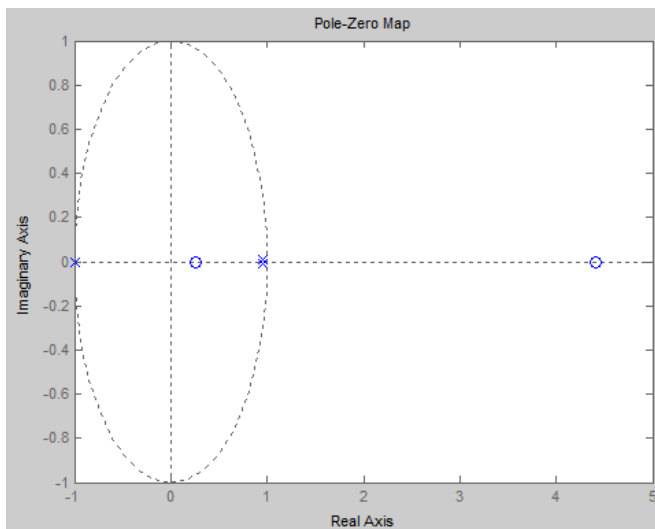
K=0,5	
Simulación	Error cuadrático
1	13,1248
2	12,007
3	12,2616
4	16,9252
5	12,3831
Media	13,3403

Tabla 4: valor del error cuadrático

El mejor caso que hemos obtenido es cuando el error cuadrático es de 1,9217, ya que es el menor que hemos obtenido. Dicho caso es que vamos a analizar. La Función de Transferencia discreta que le corresponde es:

$$\frac{-0.003764 z^2 + 0.0176 z - 0.004318}{z^3 - 0.9168 z^2 - 0.9708 z + 0.8924}$$

Los polos y ceros obtenidos son:



p =

$$\begin{matrix} -0.9859 \\ 0.9514 + 0.0065i \\ 0.9514 - 0.0065i \end{matrix}$$

z =

$$\begin{matrix} 4.4157 \\ 0.2598 \end{matrix}$$



De nuevo identifica bien los polos dominantes, pero vuelve a ver un sistema pulsante y falla en los ceros.

La gráfica de la salida respecto de la entrada es:

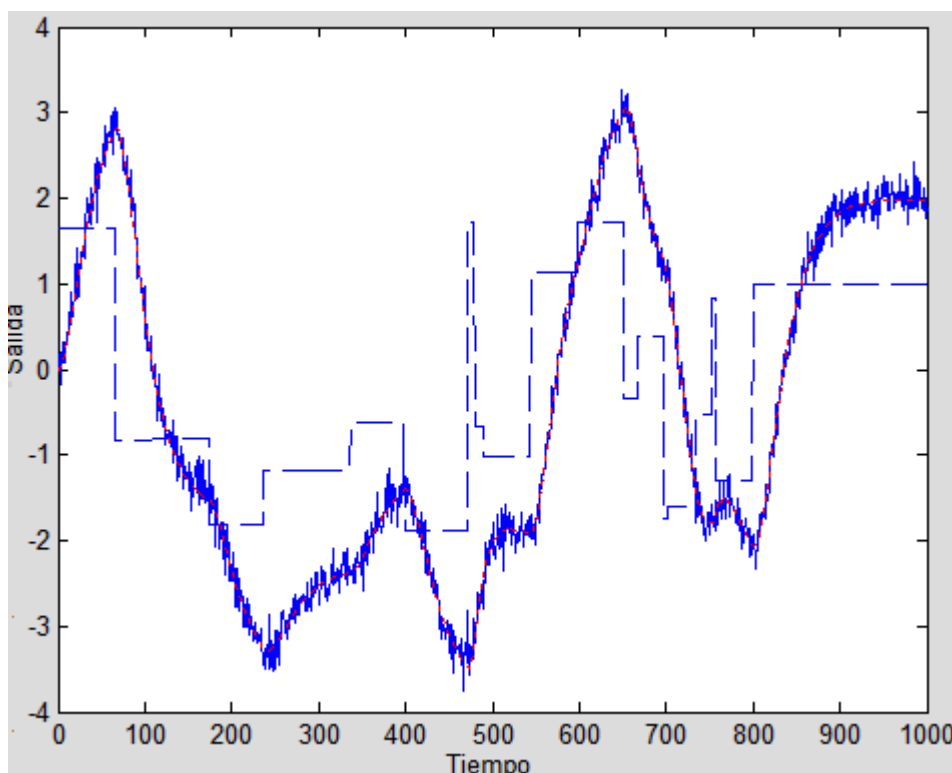


Figura 10: Representación de salidas frente a la entrada

Como se puede observar cuanto mayor es la ganancia del ruido, el error que se obtiene en la validación del sistema es mayor. En todas las simulaciones tenemos dos polos complejos conjugados a las altura de -0.95 , aunque el resto de los polos y ceros varíen considerablemente su localización. Podemos, en todo caso ver varios problemas que aparecen con el ruido:

- Fallan en la identificación de ceros
- Posibilidad de que en la identificación aparezca un sistema pulsante (lo que en continuo significaría que está mal muestreado).
- Fallan en la identificación de los polos dominados.

Todavía podemos ampliar el análisis hecho hasta ahora intentando identificar el ruido del sistema. Hemos asumido que el modelo del sistema es el Output Error.



Esto significa que el ruido se suma directamente a la salida del sistema. Por lo tanto, si aceptamos que el sistema está perfectamente identificado, las muestras de error que se producen en la validación coincidirían con las del ruido. Como lo asumimos blando, en particular debería estar decorrelacionado. En el siguiente apartado vamos a explicar cómo hemos estudiado la correlación de esta señal y qué resultados hemos obtenido.

2.3 AUTOCORRELACIÓN DE ERRORES

Para estudiar la corrección de la identificación la adecuación del modelo dinámico usado se puede estudiar el ruido estimado por la validación.

A partir, del modelo y el error cometido se puede estimar el ruido: en el caso del Output Error coincide (ya veremos más adelante los casos ARX y ARMAX) y si aceptamos la hipótesis de que éste debe ser blanco podemos estimar la bondad de la identificación. Por ejemplo, su autocorrelación debería ser nula.

La autocorrelación nos informa de la relación entre distintas muestras de una señal. En el caso de que la señal proceda de un sistema dinámico, el valor de una muestra depende del valor de las anteriores, se dice que están correlacionadas. El ruido de, como hemos dicho, lo suponemos blanco: sus muestras no deberían estar autocorrelacionadas.

Resumiendo: para calcular el error en cada punto, se restan las salidas del sistema con la del sistema identificado coordenada a coordenada. Como ya se dijo, debido al modelo que estamos usando, suponemos que este error coincide con el ruido. Entonces, se puede calcular la autocorrelación del mismo. Como hemos asumido que es un ruido blanco, la señal debe ser decorrelacionada. Es decir, cuando la autocorrelación sea muy cercana a cero es que el sistema identificado es el correcto.

Vamos a estudiar la autocorrelación de dos muestras sucesivas:

$$|correlacion| = \sum_{i=1}^{n-1} e(i) \cdot e(i + 1)$$

Siendo n el número de muestras.



Se va a indicar la autocorrelación de los errores de algunas de las simulaciones del ejercicio anterior, cuando $K=0,05$ y para $K=0,5$. Para ello se tuvo que modificar el programa de identificación añadiéndole una serie de ecuaciones para hallar la correlación.

2.3.1 $K=0,05$

K=0,05	
Simulación	Autocorrelación
1	0,0014
2	0,0105
3	0,0023
4	0,0059
5	0,0017

Tabla 5: valor de la autocorrelación

Para comprobar cuál ha sido la mejor simulación vamos observando la tabla de la autocorrelación con la del error cuadrático. Observando ambas tablas llegamos a la conclusión que la mejor simulación es la número 5 que coincide con la muestreada anteriormente ya que también es la que aporta un menor error.

2.3.2 $K=0,5$

K=0,5	
Simulación	Autocorrelación
1	0,6115
2	0,2309
3	0,7597
4	0,5204
5	0,4364

Tabla 6: valor de la autocorrelación

Para comprobar cuál ha sido la mejor simulación vamos observando la tabla de la autocorrelación con la del error cuadrático. Observando ambas tablas llegamos a la conclusión que la mejor simulación es la número 2 que coincide con la muestreada anteriormente, ya que también es la que apunta un menor error.



2.4 PERIODO DE MUESTREO

El periodo de muestreo sirve para obtener información o extraer datos de un sistema de acuerdo con una frecuencia dada. Esta forma de toma de datos hace, como ya hemos dicho, que se discretice el sistema continuo.

En principio parece lógico que cuanto menor sea el periodo de muestreo, mejor resultado tendremos en la identificación, ya que obtenemos más información del comportamiento del sistema. Sin embargo, el proceso de discretización hace que un polo del sistema continuo de valor s_p pase a convertirse en un polo del sistema discretizado de valor $e^{-s_p T}$.

En sistemas ruidosos con periodos de muestreo muy pequeños, debidos a que todos los polos y ceros son cercanos a la unidad, se corre el peligro que los pequeños errores de localización de los polos debidos al ruido se magnifiquen cuando se pasa a continuo. Fallando así la identificación.

Para ajustar un periodo de muestreo adecuado, hay que tener en cuenta antes con qué tipo de sistema nos encontramos: Si es un sistema rápido (polos alejados del eje de ordenadas) o si es un sistema lento (polos cercanos al eje de ordenadas). Si el sistema es muy rápido hay que imponerle un tiempo de muestreo muy elevado en cambio cuando el sistema es lento se puede usar un tiempo de muestreo bajo a menos que el sistema sobreoscile con una frecuencia alta. En general se debe cumplir que el periodo de muestreo sea menor que la décima parte del tiempo de estabilización y de la frecuencia de oscilación del sistema.

$$T < \min \left\{ \frac{-\operatorname{Re}(s_p)}{4.6}, \frac{2\pi}{10 w_p} \right\}$$

Siendo s_p el polo dominante del sistema y w_p la frecuencia amortiguada del mismo.



2.5 SISTEMAS DISCRETIZADOS (I)

Con este trabajo se pretende observar el error que se puede producir en la identificación al cambiar la amplitud del ruido y el periodo de muestreo.

Como ya se ha dicho, y comprobaremos más adelante, es muy importante utilizar un periodo de muestreo adecuado ya que tanto si es demasiado grande como demasiado pequeño pueden aparecer errores en la identificación. De hecho, como ya se dijo, usaremos los estudios de simulación que quedan por realizar para estudiar los periodos de muestreos más adecuados para la identificación de la maqueta.

Para realizar el ejercicio vamos a seguir los siguientes pasos:

1. Se parte de la Función de Transferencia de un sistema continuo. Para ir paso a paso en el estudio de discretización de sistemas, en este primer caso, se pasa dicha función a discreta por medio de la orden “c2dm”, especificando el periodo de muestreo.
2. Cuando se tiene la Función de Transferencia Discreta se identifica con los métodos estudiados en los anteriores experimentos.
3. La nueva Función de Transferencia Digital identificada se pasa a un su correspondiente continua por medio de la orden “d2c”.
4. Se comparan las dos Funciones de Transferencia: la original y la Continua identificada para distintos tiempos de estabilización.

Gracias a este experimento, se obtienen las bases para cuando se tenga que trabajar con la maqueta, ya que la maqueta en sí es un sistema continuo.

En el esquema de Simulink (*Figura 11*) hay que cambiar una serie de parámetros, pero otros como las entradas, salidas y las conexiones entre el simulador y el espacio de trabajo no hacen falta cambiarlas. Para la identificación del error no hace falta crear un subprograma, ya que el propio esquema proporciona dicho dato.



La Función de Transferencia del sistema que vamos a identificar tiene dos polos en -90 y una ganancia de $0,8$. Utilizamos este sistema porque, según resultados previos, creemos que puede ser una aproximación al modelo de la maqueta que queremos identificar.

La Función de Transferencia discreta que utilizamos es la siguiente:

$$G(z) = \frac{0.8}{\left(\frac{1}{90}s + 1\right)^2}$$

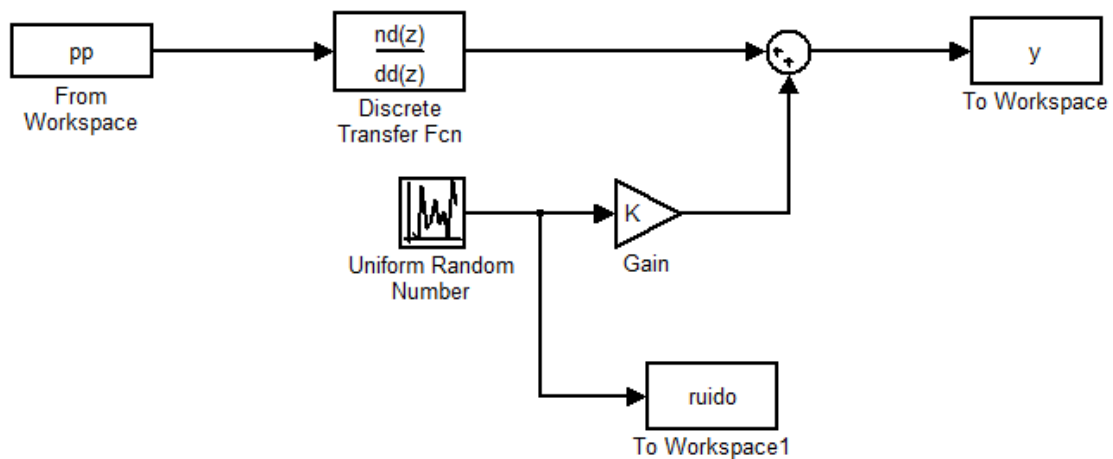


Figura 11: Esquema de un sistema discreto

La salida al “*Workspace*” del ruido lo hacemos porque se daban resultados extraños y queríamos analizar cómo se comportaba la función de Simulink.

Los errores obtenidos en el proceso de validación se muestran en la siguiente tabla, aunque para su representación hemos elegido los mejores casos, esto es cuando el ruido es más bajo:



	K de ruido					
	0	0,1	0,3	0,5	1	2
T=0,0001seg	4,42E-24	1,6694	15,1475	46,6831	227,484	693,9226
	1,18E-23	1,6934	40,746	44,56	184,0104	651,0331
T =0,0005seg	3,22E-26	1,605	14,8021	43,5715	164,0509	671,297
	4,34E-26	1,6611	14,8779	40,3521	175,6442	829,9071
T =0,001seg	1,37E-26	1,725	15,5033	43,5219	166,9913	666,2284
	4,44E-27	1,6687	15,7567	45,072	164,9331	706,6288
T =0,005seg	4,04E-29	1,6101	14,7481	42,597	165,5998	658,5092
	5,08E-29	1,5587	14,7982	40,3588	159,9309	697,1486
T =0,01seg	5,12E-30	1,611	15,9062	39,8344	168,1788	616,4322
	7,51E-30	1,6233	15,4015	41,7514	161,911	645,7262
T =0,05seg	2,60E-31	1,6988	15,2375	41,5039	167,6863	637,3376
	3,35E-30	1,6956	14,8746	43,8632	177,1994	692,1662

Tabla 6: Representación de errores

Se puede observar que al variar la amplitud del ruido el error que se obtiene es mayor cuanto mayor es dicha amplitud, como podíamos esperar.

Analizando los datos del error también podemos ver, como ya habíamos previsto que la identificación se degradada para periodos de muestreo altos, pero también para los muy bajos.

Teniendo en cuenta que los polos del sistema a identificar están en -90 , podemos estimar que el tiempo de estabilización del sistema sea cercano a $0,05$ (aproximadamente $4.6/90$), por lo que la cota máxima para el periodo de muestreo sería $0,005$. Notemos que es alrededor de ese valor, en especial con ruido alto, donde se dan los mejores resultados.

La posición de los polos varía según lo hace la amplitud del ruido. Cuando la ganancia del error es nula, los polos quedan perfectamente situados en -90 . Para $K=0,1$ los polos se mantienen alrededor de su posición real, pero cuando la ganancia del ruido aumenta, los polos se dispersan y no mantienen alrededor de esa posición.

2.5.1 $K=0,1$ y $T=0,005$ segundos \rightarrow ruido=1,6101

La Función de Transferencia identificada es la siguiente:

$$\frac{191.4 s + 4.108e004}{s^2 + 467.7 s + 5.134e004}$$

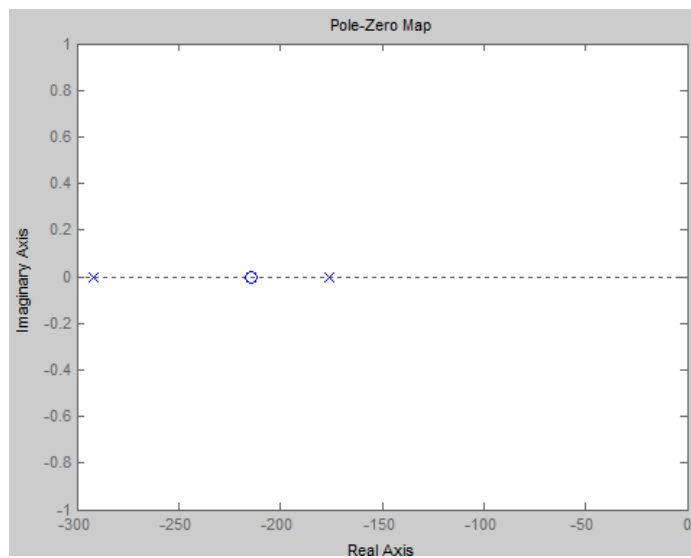
Los polos y los ceros que se han obtenido son:

$p =$

-291.7637
-175.9611

$z =$

-214.5918



La gráfica donde se puede apreciar la salida es:

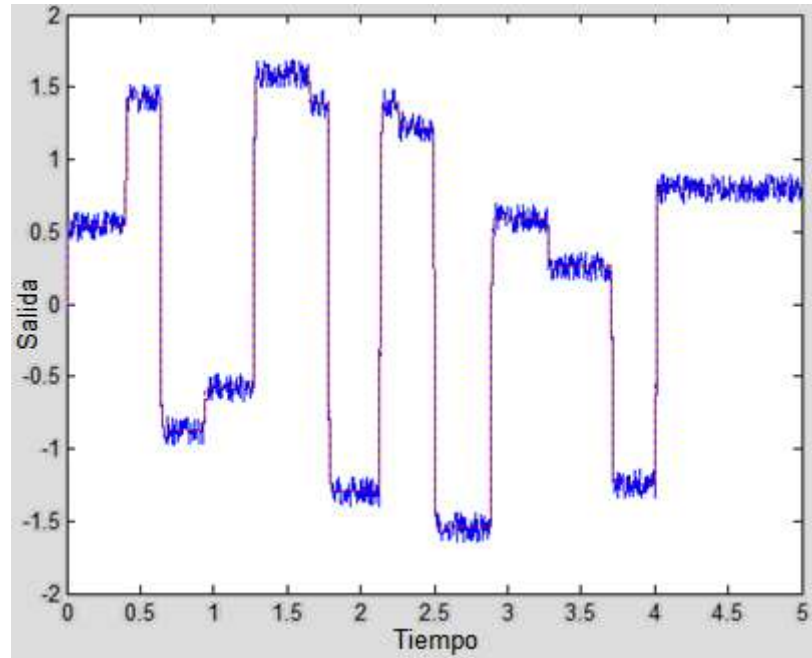


Figura 12: Representación de las salidas

El comportamiento de los polos y ceros, tanto como la magnitud del ruido, es que cuanto mayor es el periodo de muestreo y la K del ruido.

Algunas simulaciones se salen del comportamiento habitual de ellas en determinadas condiciones, como puede ser el error de 518,6163 en un tiempo tan pequeño como $T=0,0005$ segundos, ya que el error más habitual ronda entre un 1,4 un 1,7. Hay casos puntuales en algunos periodos de muestreo en los que la señal de entrada no es la adecuada y se obtienen unos errores muy elevados, por eso las simulaciones hay que repetir las varias veces \rightarrow bucles.

En general, vemos que la identificación, en este caso, se degrada fuertemente en cuanto aparece ruido y que también se degrada al variar el periodo de muestreo. Como esta era la comprobación que queríamos hacer este experimento, pasamos al siguiente donde se reproduce más fielmente el caso real.



2.6 SISTEMAS DISCRETIZADOS (II)

La maqueta con la que se va a trabajar es un sistema continuo, por eso previamente vamos a trabajar con un sistema simulado que tenga características parecidas a la maqueta. En el apartado anterior ya hemos dado pasos El sistema que se va a utilizar para el análisis es el siguiente:

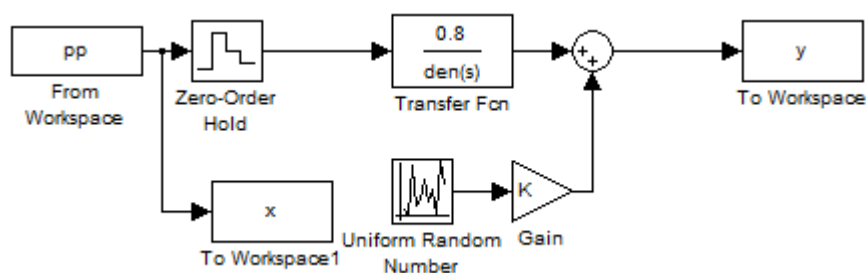


Figura 13: Esquema de un sistema continuo que ha sido discretizado

Para realizar la identificación del sistema discreto se ha colocado un retenedor de orden cero, que mantiene la señal constante entre periodos de muestreo, delante de la Función de Transferencia continua y se obtienen los datos de la salida con un muestreador (nos da los datos cada periodo de muestreo). De esta forma, cada periodo de muestreo tenemos un dato de entrada y uno de salida, de forma similar a un sistema discreto. Un sistema continuo en vez de proporcionar datos cada cierto tiempo daría una señal continua, sin ningún tipo de interrupción en el tiempo.

La función que transferencia que se va a utilizar es la que se supone que puede tener la maqueta, para luego poder comparar los resultados que se obtienen en este ejercicio con los que se obtienen al realizar la simulación con dicha maqueta. En este caso, se supone que el modelo dinámico utilizado es Output Error, pero en la maqueta habrá que analizar qué modelo dinámico es el más adecuado.

En primer lugar se realiza el análisis del sistema sin elasticidad. Posteriormente se pasará a analizar un sistema con elasticidad.



2.6.1 SIN ELASTICIDAD

Para facilitar los cálculos, como en el ejercicio anterior, se va a realizar una aproximación de la Función de Transferencia.

$$G(s) = \frac{0.8}{\left(\frac{1}{90}s + 1\right)^2}$$

Como ya hemos estudiado en casos anteriores, hay que tener cuidado con el periodo de muestreo que se utiliza. Se puede calcular una cota superior del periodo de muestreo a través de la fórmula:

$$T_m = \left(\frac{4.6}{\text{Re}(sp)}\right) / 10 \text{ segundos}$$

Con la Función de Transferencia que indicada en la parte superior, los polos quedan en -90, lo que da un periodo de muestreo de unas 5 milésimas de segundo.

Para poder realizar adecuadamente el análisis se va a ir cambiando la ganancia de ruido y el periodo de muestreo. Tras haber realizado todas las simulaciones, los mejores casos, esto es, en los que el ruido es menor, quedan reflejados en la siguiente tabla:

	K de ruido					
	0	0,01	0,03	0,05	0,1	0,5
T =0,0001seg	8,67E-04	0,0199	0,1458	0,3955	20,4847	73,1314
	3,90E-03	0,0172	0,1436	0,3946	1,5775	112,8536
T =0,0005seg	8,34E-02	0,0715	0,1812	0,5007	1,6356	39,2575
	7,26E-02	0,1078	0,2977	0,5359	1,6247	39,2729
T =0,001seg	0,1745	0,428	0,4227	0,5302	2,5133	39,521
	0,1889	0,2712	0,2586	0,6462	1,7217	39,7066
T =0,005seg	1,433	1,2008	1,326	2,3966	3,3515	41,2214
	1,857	1,3477	1,3053	1,552	2,205	41,2969
T =0,01seg	1,8431	4,0762	3,483	2,5348	4,9398	42,7552
	3,4312	2,0854	2,8273	3,0486	4,8066	40,9555

Tabla 7: Representación de los errores



Como se puede observar gracias a la tabla de datos obtenida, cuanto mayor es la amplitud del ruido y el periodo de muestreo, el error que obtenemos al validar el sistema identificado es mayor. Es interesante notar que los periodos de muestreo pequeños cometen más error que otros mayores para ruidos altos. Es de esperar: con esos ruidos se cometen más errores en la localización del sistema discretizado y es para esos periodos cuando más se amplifican los errores al calcular la Función de Transferencia Continua correspondiente.

Para saber si la identificación está correctamente realizada, además de analizar el error también hay que analizar los polos y ceros que tiene el sistema identificado y su localización.

La Función de Transferencia que hemos impuesto al sistema tiene dos polos en -90 y no tiene ningún cero. Si la simulación estuviese correctamente realizada, únicamente se obtendrían dos polos pero en la realidad se obtienen dos polos y un cero. Dicho cero no es de mayor importancia ya que su valor es muy elevado y queda completamente dominado por los polos, así que lo podemos despreciar.

Vamos a tomar los mejores casos de la tabla para una $T=0,001$ segundos cuando $K=0,01$ y cuando $K=0,1$.

2.6.1.1 $K=0,01; T=0,001$ segundos

La Función de Transferencia identificada es la siguiente:

$$\frac{-3.084 s + 5994}{s^2 + 169.4 s + 7491}$$

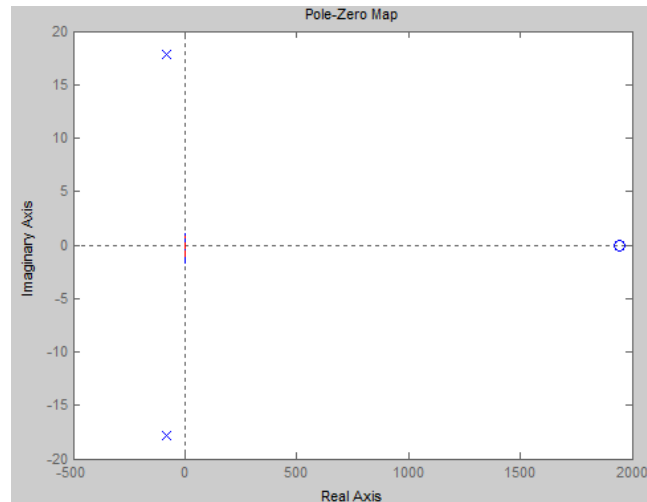
Ganancia de 0.8



Los polos y los ceros que se han obtenido son:

```
p =
-84.6936 +17.8370i
-84.6936 -17.8370i

z =
1.9439e+003
```



Como se ve, la identificación de los polos es bastante buena. De hecho mejor que en el apartado anterior o la parte imaginaria no tiene mayor importancia pues da una sobreoscilación inapreciable.

La gráfica donde se puede apreciar la salida es:

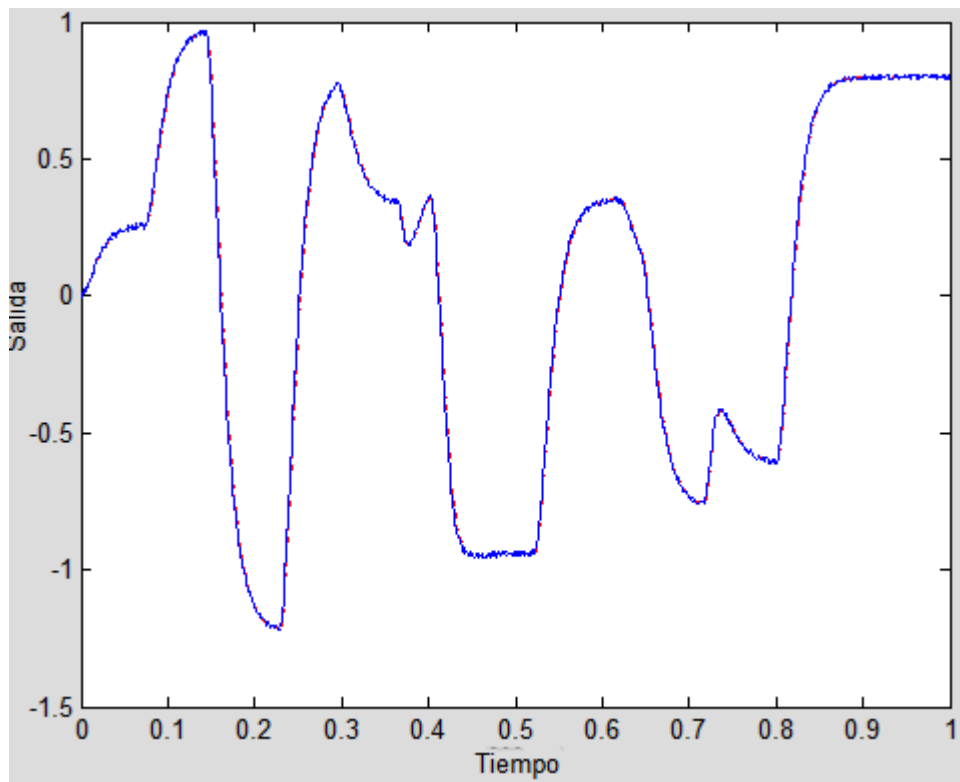


Figura 14: Representación de las salidas



Vamos a comprobar mediante la siguiente gráfica si hay alguna diferencia entre las dos ondas de salida:

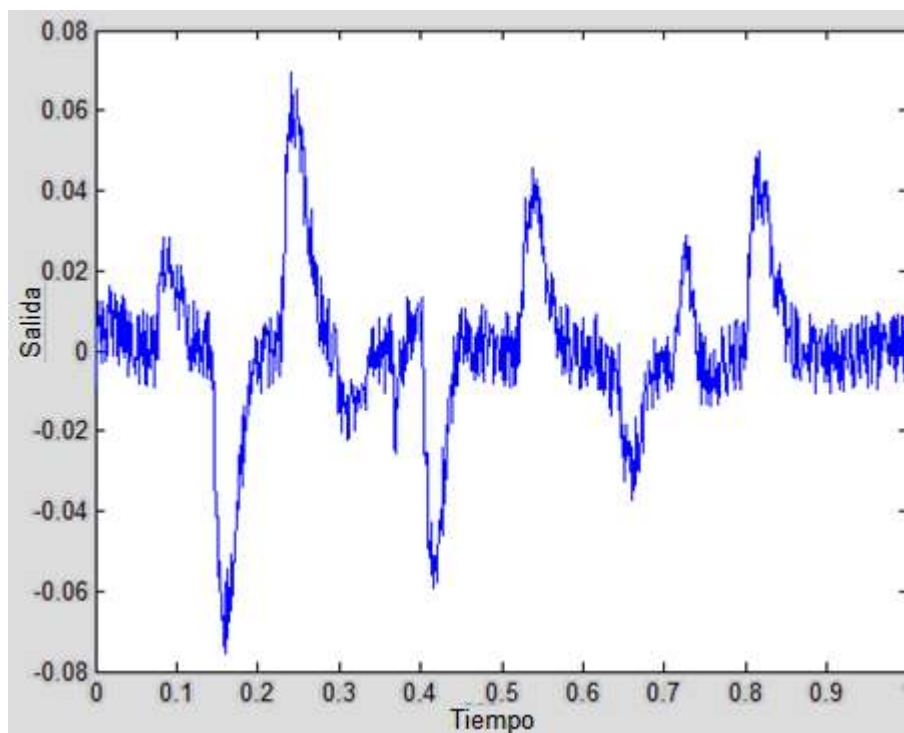


Figura 15: Representación del error

2.6.1.2 $K=0,5; T=0,001$ segundos

La Función de Transferencia identificada es la siguiente:

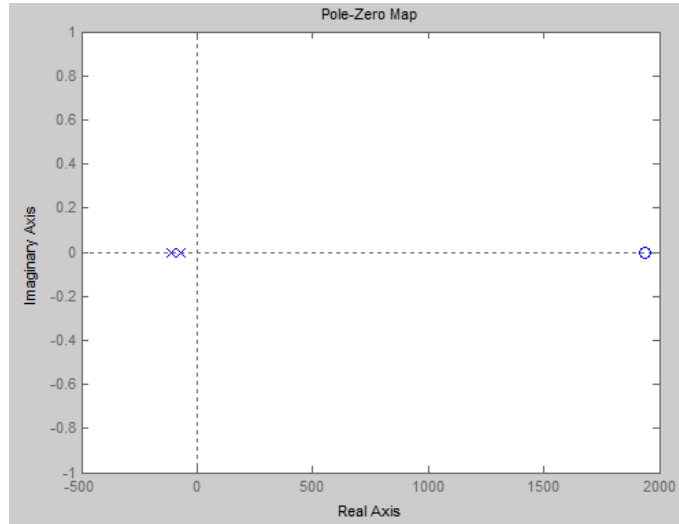
$$\frac{-3.228 s + 6261}{s^2 + 182 s + 7767}$$

La ganancia de la Función de Transferencia es de 0.8.



Los polos y los ceros que se han obtenido son:

```
p =  
-113.7085  
-68.3021  
  
z =  
1.9399e+003
```



Como vemos, con un ruido muy alto la identificación se degrada, aunque los polos siguen estando alrededor de su valor real.

La gráfica donde se puede apreciar la salida es:

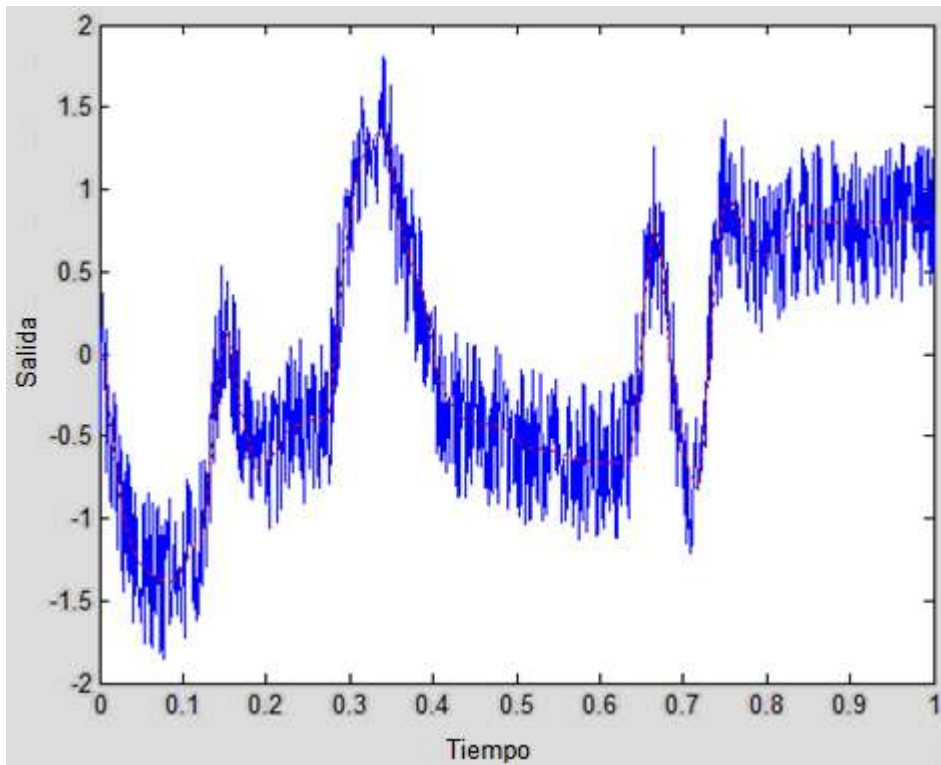


Figura 16: Representación de las salidas



Vamos a comprobar mediante la siguiente gráfica si hay alguna diferencia entre las dos ondas de salida:

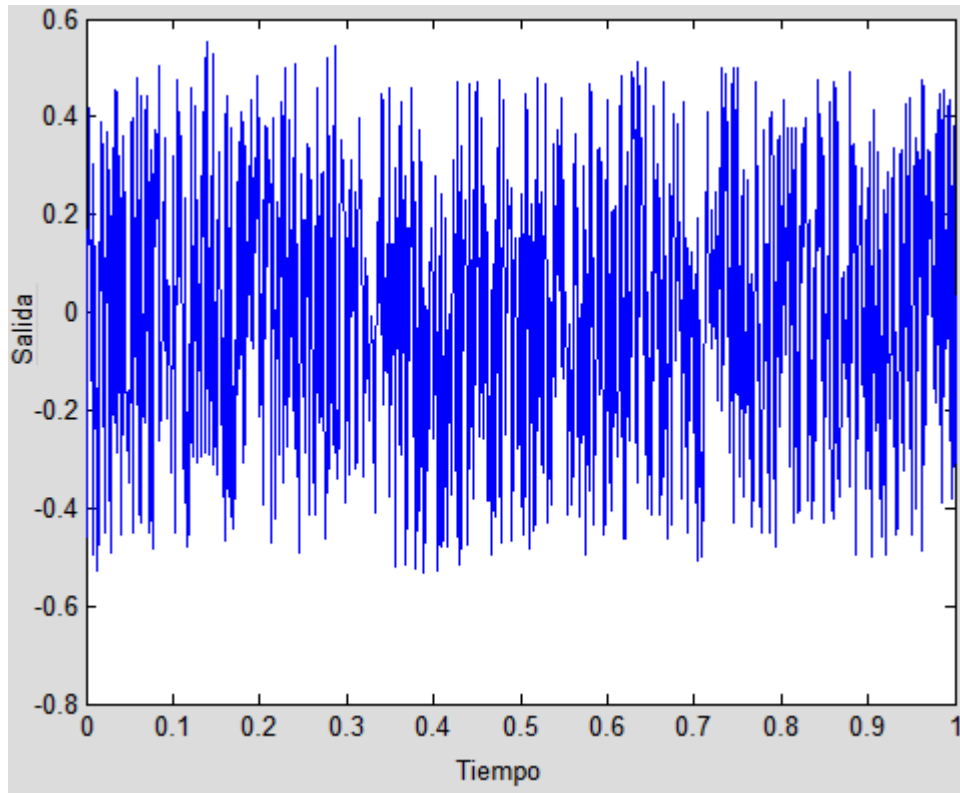


Figura 17: Representación del error

Como se puede observar gracias a las gráficas, que cuando aumentamos la ganancia del ruido, vemos como la diferencia entre la señal de salida de la simulación y la señal de salida de la Función de Transferencia identificada varían en mayor medida.

Para saber que la identificación se ha realizado correctamente además de analizar los polos y los ceros y el error también se debe analizar la ganancia. En varias simulaciones realizadas a la hora de efectuar la comprobación no se obtiene la ganancia correcta, lo que indica que la identificación no está bien realizada y se tendría que volver a identificar, analizando previamente el problema.



2.6.2 CON ELASTICIDAD

En el caso que se va a analizar a continuación se va a utilizar una Función de Transferencia similar a la que se esperaría que de la maqueta si esta presenta elasticidad. Se supone que el sistema tiene dos ceros en $-120i$, siendo i la unidad imaginaria, un polo en -90 y otros dos polos en $\pm 125i$, por lo que la Función de Transferencia quedaría de la siguiente manera:

$$G(s) = \frac{125^2}{120^2} \cdot 0,8 \cdot \frac{s^2 + 120^2}{\left(\frac{1}{90}s + 1\right) \cdot (s^2 + 125^2)}$$

Para realizar la identificación hay que tener en cuenta que nos encontramos con dos ceros y tres polos. El término de $\frac{125^2}{120^2}$ ayuda a mantener la ganancia de la Función de Transferencia en 0,8.

El esquema que queda en el Simulink es de las siguientes características:

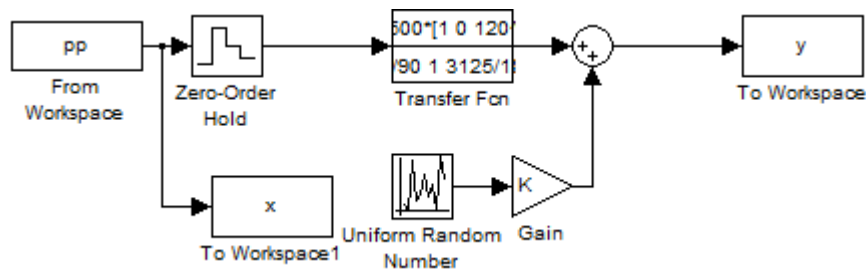


Figura 18: Representación de un sistema con elasticidad

Para poder realizar adecuadamente el análisis se va a ir cambiando la ganancia de ruido y el periodo de muestreo. Tras haber realizado todas las simulaciones, los mejores casos, esto es, en los que el ruido es menor, quedan reflejados en la siguiente tabla:



	K de ruido				
	0	0,00005	0,0001	0,01	0,1
T =0,00005seg	4,22E-04	0,0211	0,0505	0,1446	1,6657
	8,54E-02	0,0589	0,1583	0,0433	1,7211
T =0,00008seg	3,60E-18	1,0245	0,1182	0,7866	2,0807
	8,41E-02	0,4653	0,337	0,5193	1,7683
T =0,0001seg	1,55E-18	1,3212	431,0668	1,5799	2,5946
	0,0663	0,7539	0,0829	1,3799	4,0926
T =0,001seg	1,78E-18	0,0049	0,0117	0,0156	1,5694
	3,13E-23	0,0124	0,0197	5,5201	1,5717

Tabla 8: Representación de errores

A través de la tabla vemos que cuando menor error se produce y cuando mejor se identifican los polos y los ceros es cuando $T=0,001$ segundos. Los polos y ceros que no dan con ese periodo de muestreo es el más ajustado a su situación real. Por eso, concluimos que el mejor periodo de muestreo es éste el último.

Vamos a realizar el análisis con distintas ganancias de ruido para un periodo de muestreo de $T=0,001$ segundos.

2.6.2.1 $K=0,00005$

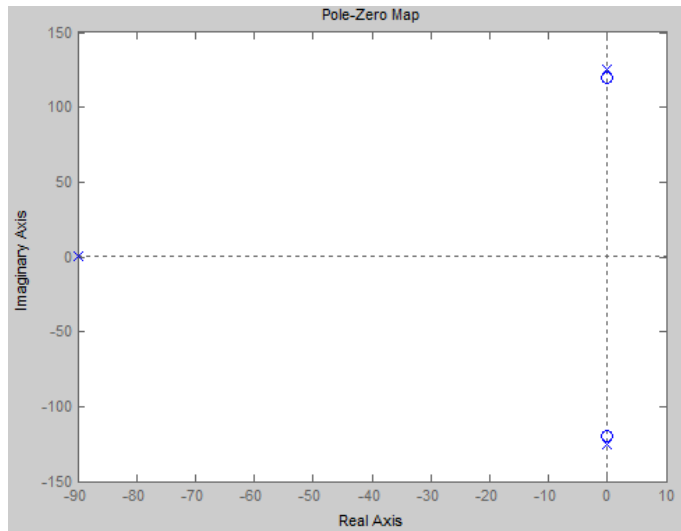
La Función de Transferencia identificada es la siguiente:

$$\frac{78.12 s^2 + 1.868 s + 1.125e006}{s^3 + 89.94 s^2 + 1.563e004 s + 1.406e006}$$



Los polos y los ceros que se han obtenido son:

```
p =  
  
1.0e+002 *  
  
0.0000 + 1.2504i  
0.0000 - 1.2504i  
-0.8994  
  
z =  
  
1.0e+002 *  
  
-0.0001 + 1.2002i  
-0.0001 - 1.2002i
```



Vemos que la identificación con ruido muy bajo es muy buena.

La gráfica donde se puede apreciar la salida es:

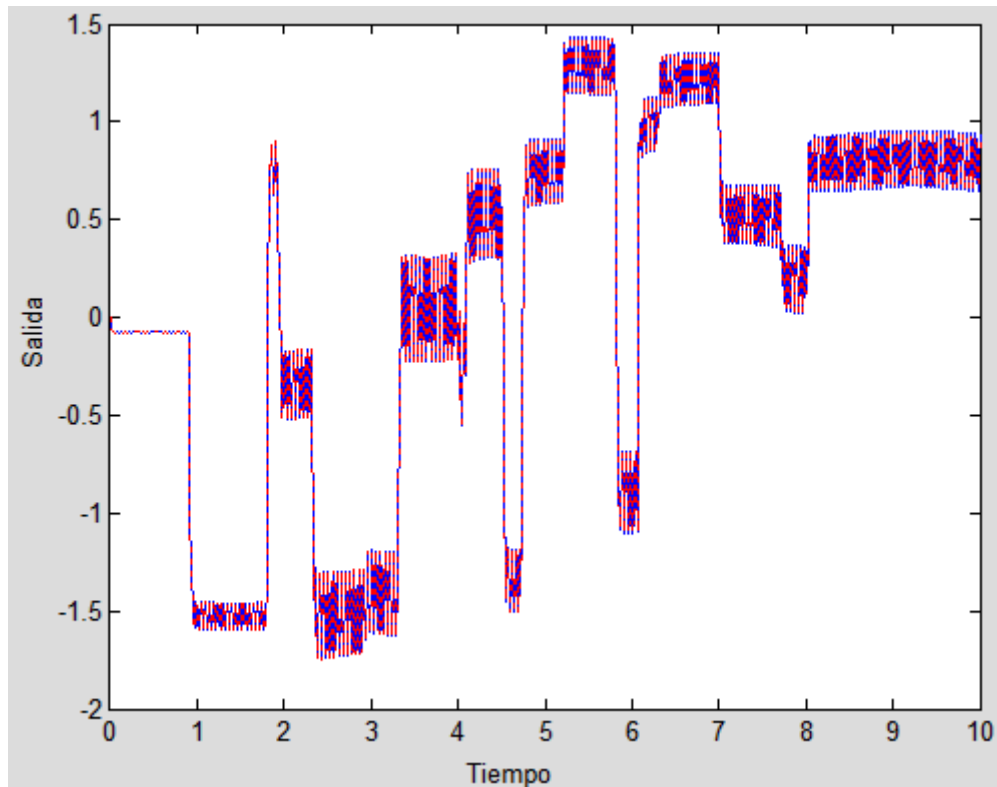


Figura 19: Representación de las salidas



2.6.2.2K=0,1

La Función de Transferencia identificada es la siguiente:

$$\frac{79.98 s^2 + 14.14 s + 1.151e006}{s^3 + 91.96 s^2 + 1.563e004 s + 1.437e006}$$

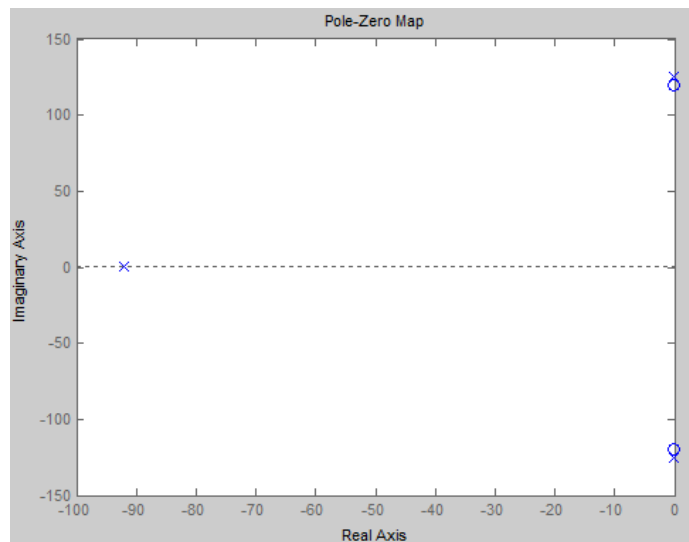
Los polos y los ceros que se han obtenido son:

p =

```
1.0e+002 *
-0.0001 + 1.2503i
-0.0001 - 1.2503i
-0.9195
```

z =

```
1.0e+002 *
-0.0009 + 1.1996i
-0.0009 - 1.1996i
```



De nuevo tenemos una identificación muy buena, tanto del polo como de la elasticidad.



La gráfica donde se puede apreciar la salida es:

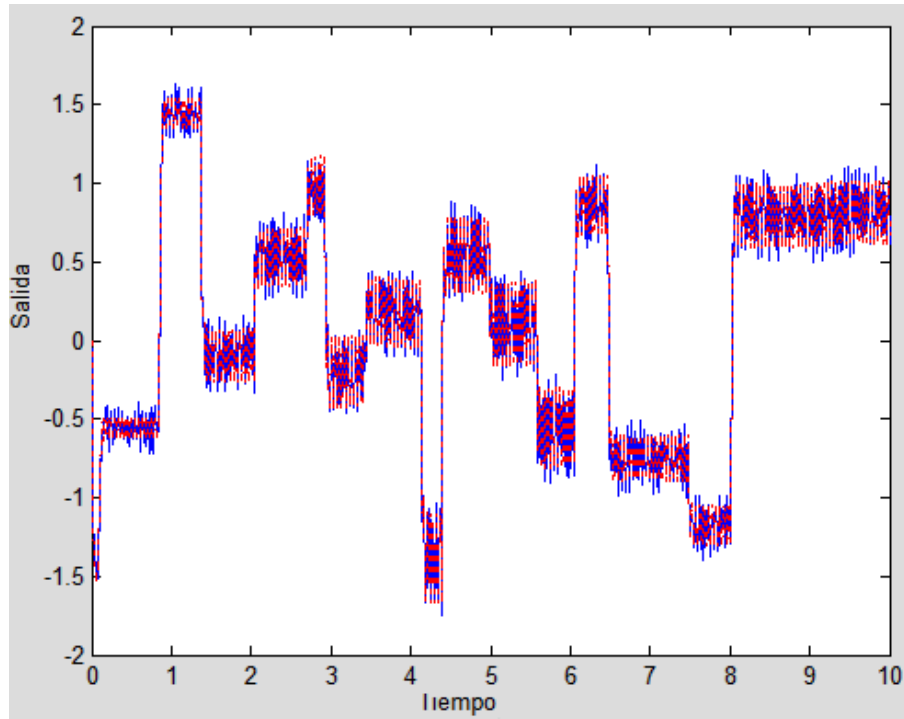


Figura 20: Representación de las salidas

2.6.3 Con elasticidad (II)

Como hemos hallado que los mejores resultado en la identificación se consiguen para el periodo de muestreo $T=0.001$ segundos, se va a efectuar un barrido más concentrado alrededor de ese valor ajustar mejor dicho parámetro.

	K de ruido					
	0	0,0005	0,001	0,005	0,01	0,1
T =0,0005seg	2,64E-21	3,93E-05	2,65E-04	0,9715	16,708	16,3435
	9,55E-23	0,3353	1,87E-04	10,8118	0,0157	12,4876
T =0,001seg	1,78E-18	3,92E-05	1,57E-04	0,0039	0,0157	1,5641
	3,31E-23	3,91E-05	1,56E-04	1,0969	0,0157	17,2041
T =0,003seg	5,96E-13	0,0046	0,0041	1,1748	0,0158	16,6787
	3,72E-13	0,0024	0,0146	0,0039	0,0168	13,8915
T =0,005seg	4,85E-10	8,01E-04	0,0014	0,0929	0,9554	1,604
	2,79E-26	1,80E-04	0,0019	0,2236	0,2848	1,564
T =0,01seg	1,30E-03	6,13E-04	0,0011	0,0106	0,0473	1,5766
	1,11E-02	5,33E-04	0,0012	0,0109	0,0348	2,1881

Tabla 9: Representación de errores



Si se analiza la tabla se advierte que la simulación que muestra mejores resultados es cuando $T=0,005$ segundos, ya que aunque suba mucho la ganancia del error la identificación de los polos es muy buena.

Vamos a realizar el análisis con distintas ganancias de ruido para un periodo de muestreo de $T=0,005$ segundos.

2.6.3.1 $K=0,0005$

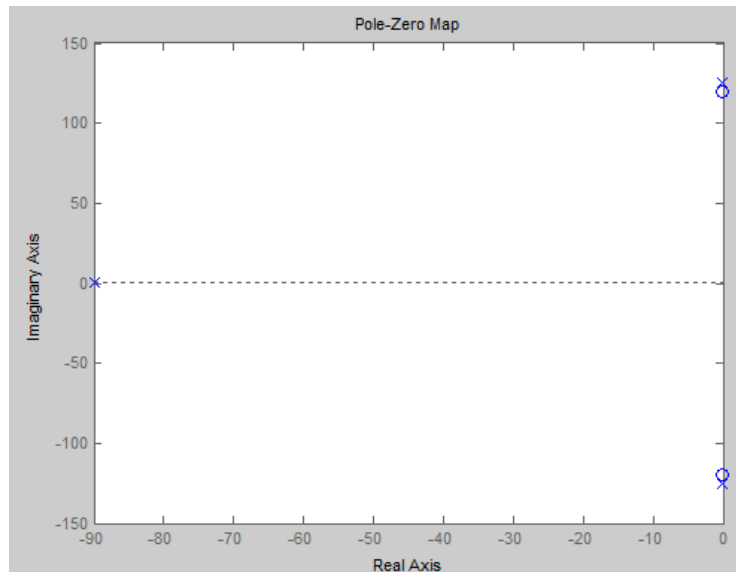
La Función de Transferencia identificada es la siguiente:

$$\frac{78.11 s^2 + 5.801 s + 1.125e006}{s^3 + 89.97 s^2 + 1.563e004 s + 1.406e006}$$

Los polos y los ceros que se han obtenido son:

```
p =
1.0e+002 *
-0.0000 + 1.2501i
-0.0000 - 1.2501i
-0.89996

z =
1.0e+002 *
-0.0004 + 1.1999i
-0.0004 - 1.1999i
```





La gráfica donde se puede apreciar la salida es:

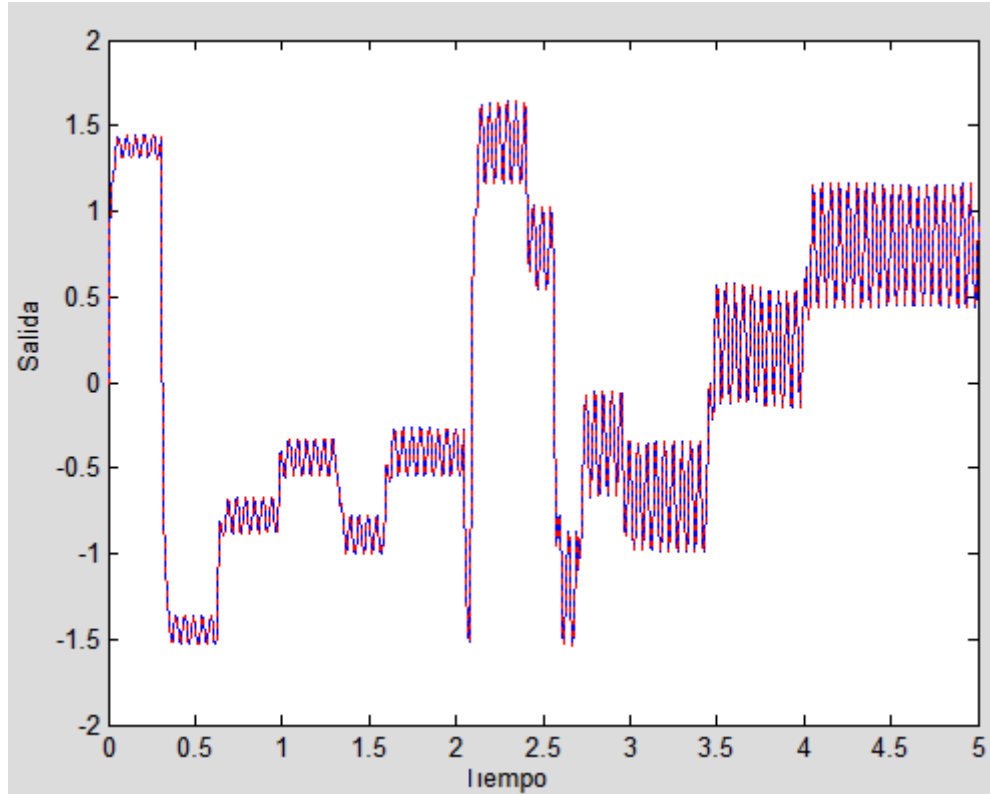


Figura 21: Representación de las salidas

2.6.3.2K=0,01

La Función de Transferencia identificada es la siguiente:

$$\frac{78.35 s^2 + 0.62 s + 1.129e006}{s^3 + 90.29 s^2 + 1.563e004 s + 1.411e006}$$

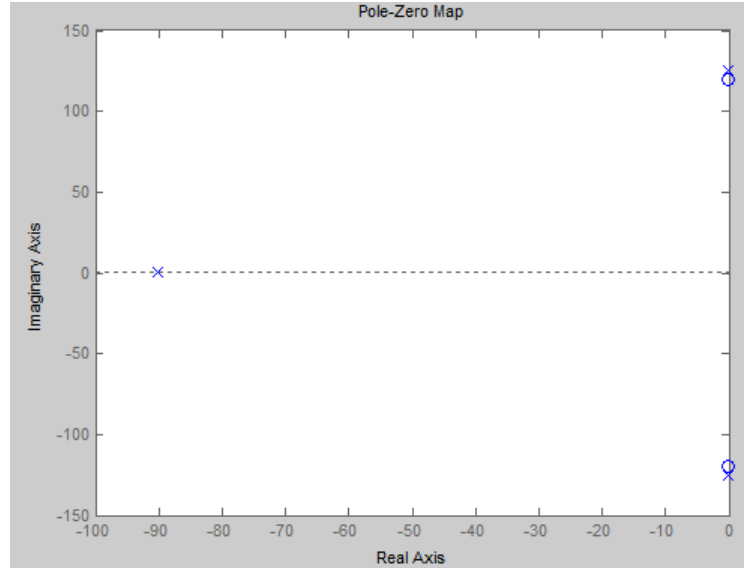


Los polos y los ceros que se han obtenido son:

```

p =
    1.0e+002 *
    -0.0000 + 1.2500i
    -0.0000 - 1.2500i
    -0.9028

z =
    1.0e+002 *
    -0.0000 + 1.2002i
    -0.0000 - 1.2002i
  
```



La gráfica donde se puede apreciar la salida es:

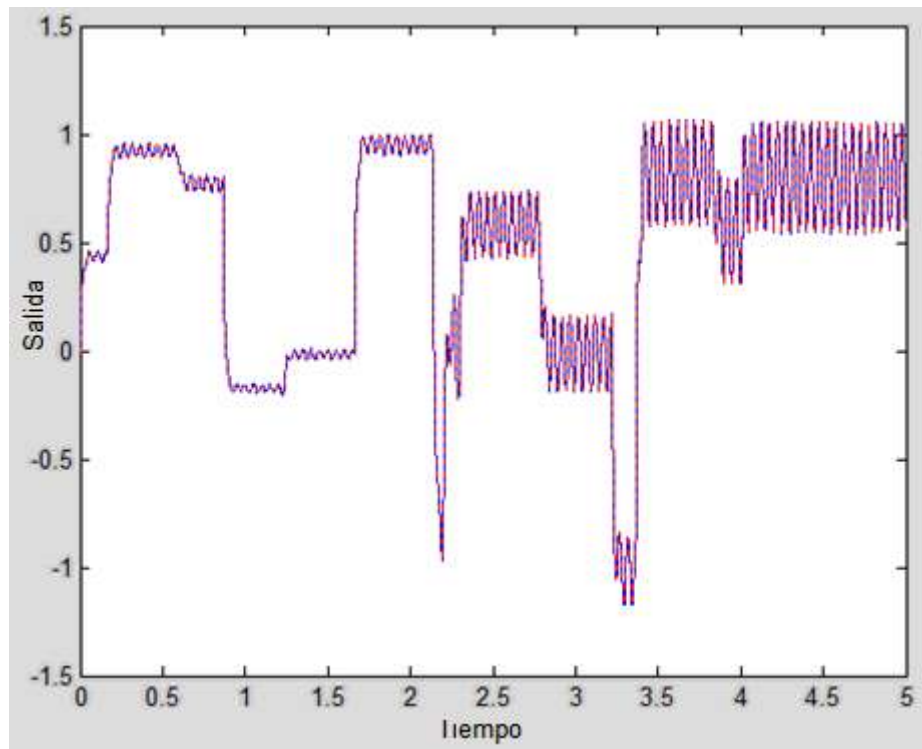


Figura 22: Representación de las salidas



Como conclusiones tenemos:

- Parece que los periodos de muestreo más prometedores para la maqueta están alrededor de 0.005 segundos.
- Al aumentar el ruido se degrada la identificación de los polos dobles.
- Parece posible hacer una identificación de la elasticidad en el caso de que esta exista.



Capítulo 3. IDENTIFICACIÓN DE UN SISTEMA REAL

Tras adquirir una serie de conocimientos previos sobre el manejo del programa principal, Matlab y de los sistemas continuos discretizados, somos capaces de proceder la identificación de la maqueta.

Lo que se pretende es que mediante distintos tipos de análisis se averigüe el número de polos y ceros de la maqueta, la situación de estos, el modelo dinámico que le corresponde y su ganancia.

Lo primero que hay que hacer es conectar la maqueta al ordenador. Hay que encender el motor, conectándolo a la red y también hay que encender una torreta que permite la conexión entre el ordenador y la maqueta, haciendo posible el intercambio de datos entre ellos. Para poder tomar y proporcionar datos se usan dos programas de Labview, uno funcionando en modo “run”, para poder obtener datos en cualquier momento. El otro subprograma es el que permite introducir la señal de entrada creada a la maqueta y obtener la de salida correspondiente.

Con Matlab se genera una señal de entrada discreta, que se pasará a continua a través de un retenedor de orden cero que está integrado en la tarjeta digitalizadora y tras obtener los datos por medio de Labview, el programa permite realizar la identificación. Estos datos son obtenidos por la tarjeta digitalizadora funcionando en este caso como un muestreador. De esta forma obtenemos datos de entrada y salida discretos: hemos discretizado el sistema continuo y podemos pasar a la identificación del dicho sistema. La identificación se puede realizar de distintas formas, dependiendo del modelo dinámico que se supone que pueda tener la maqueta. Los tres modelos dinámicos que se van a analizar son el Output Error, el ARX y el ARMAX. Tras haber realizado el análisis con los tres modelos de error posibles se comprobará la correlación, el error y la localización de los polos y los ceros para certificar qué caso es el mejor de todos.

Para realizar el trabajo tenemos que tener en cuenta que el sistema es no lineal debido a que tiene una zona muerta alrededor de 0 (se estima que está entre -1 y +1). Por lo tanto vamos a trabajar sólo con valores positivos que eviten dicha zona linealizando alrededor de un punto de trabajo. Hemos escogido el punto estable que se obtiene para la entrada de valor 3Volt. Las señales de entrada van a variar alrededor de este punto, sin alcanzar la zona muerta, y de la respuesta del motor a ellas podremos obtener el valor de salida del punto estable para proceder a la linealización alrededor del mismo.



Lo que pretendemos es realizar la linealización de sistemas no lineales alrededor del punto de trabajo. Si realizamos la comparación entre la entrada y la salida no nos dará el valor correspondiente a un sistema lineal. En cambio, si lo hacemos con incrementos muy pequeños de dichas señales sí que obtenemos parámetros de un sistema lineal, por lo que sí se puede modelar como un sistema de ese tipo. La forma de hacerlo es tomar los datos de entrada y a los de salida usadas para identificar y restarles el punto de trabajo. Estos datos se usan para el cálculo de la Función de Transferencia y para cualquier otro tipo de cálculo de dinámica lineal.

El desarrollo general del trabajo comienza con una previa identificación de la maqueta y se realizan tres comparaciones para validar la Función de Transferencia. Cada identificación da lugar a una Función de Transferencia característica, gracias a la cual se podrán comparar las salidas procedentes de Matlab con las salidas de la maqueta, conociendo la diferencia entre ellas, esto es, conociendo el ruido y por tanto también pudiendo calcular la correlación. Para certificar que la identificación se ha realizado correctamente, debemos de realizar otras dos identificaciones, esto es, se deben realizar tres identificaciones por cada periodo de muestreo, y observar que los resultados de todas las simulaciones son similares.

El análisis se va a realizar dependiendo de los tres modelos dinámicos y el orden que se va a seguir es el establecido en la parte de arriba: Output Error, ARX y ARMAX.

3.1 MODELO Output Error

Todos los casos que se van a realizar son con un tiempo de 5 segundos, pero la toma de muestras se realizará con distintos periodos de muestreo. En principio se tomarán medidas con periodos de muestreo de 0,7 milisegundos y se irá aumentando este parámetro hasta llegar a un tiempo final de 7 milisegundos. De esta forma cubrimos los mejores valores obtenidos con la simulación.

Para la generación de entradas realizamos un programa que genera pulsos de tipo escalón alrededor del valor 3 con alturas y números de muestras en cada escalón generados de forma aleatoria. Tanto la altura como la anchura máxima del escalón se pueden fijar en cada experimento. Al final de la señal se genera una entrada constante de valor tres que nos permite identificar el punto de trabajo (*ver Figura 23*). Como por ejemplo:

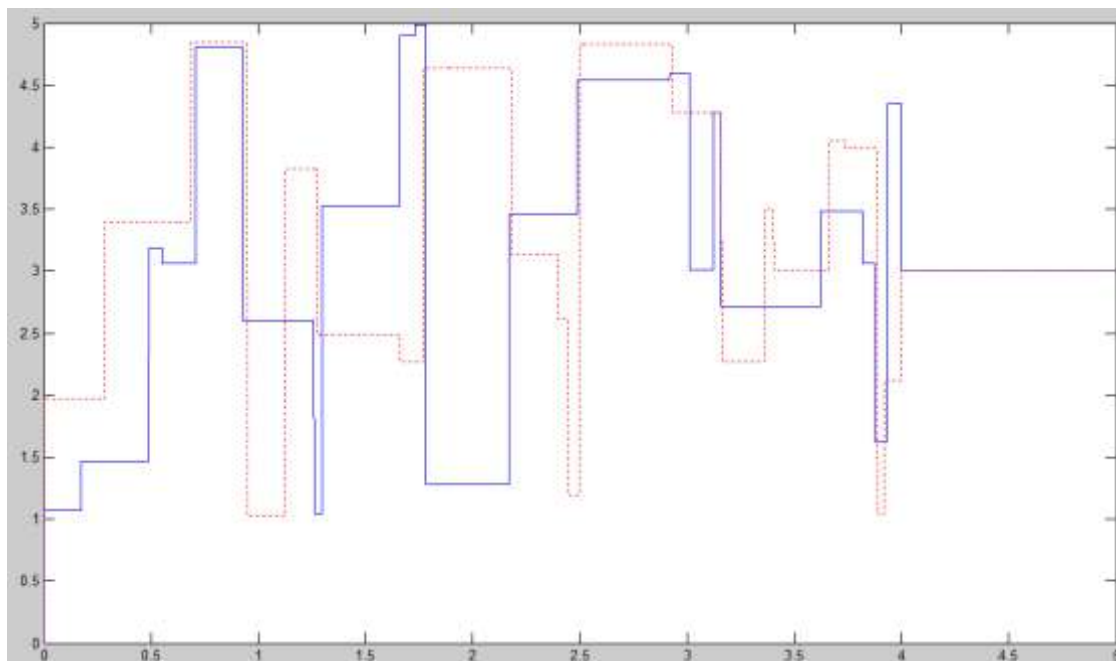


Figura 23: Distintas entradas

Tras haber generado la entrada, en uno de los programas realizados en Labview se indica la entrada que se le quiere proporcionar a la maqueta y el nombre que se quiere que tenga el archivo con los datos de salida.

Al obtener la salida hay que identificarla. Se debe tener en cuenta que los datos que proporciona Labview son en forma de fila, por lo que para utilizar dichos datos en Matlab habrá que transponerlos, ya que en Matlab los datos hay que presentarlos en forma de columnas.

Tras haber generado la entrada, haberla pasado por el motor y haber obtenido su salida correspondiente, para proceder con la identificación debemos restarle a la señal de entrada el valor del punto de trabajo. A la señal de salida, por otra parte, hay que restarle el valor de salida del punto de trabajo, pero en este caso este valor no queda perfectamente definido como en el caso anterior. Para conocerlo, vamos a realizar la media de los puntos en los que la onda se ha estabilizado ante entrada 3 y dicho valor será el que habrá que restarle a la salida. Este procedimiento hay que seguirlo con todos los casos, para poder obtener un sistema perfectamente identificado. Aunque hagamos la media de la salida en cada simulación, todos los valores obtenidos son prácticamente iguales, ya que todos ellos rondan en una ganancia del 2.64.



3.1.1 Sistemas de 3º orden

Se va a suponer que la maqueta se caracteriza por tener **dos ceros y tres polos**, como se ha supuesto en los experimentos de simulación realizados anteriormente en el caso de elasticidad. La identificación completa se efectúa de la siguiente manera:

$$Z = \text{iddata}(\text{salida}', \text{entrada}', T)$$

$$M = \text{oe}(Z, [\text{cero} + 1 \text{ polos } \text{retardo}])$$

Donde *salida* y *entrada* son los datos obtenidos de la maqueta, *T* es periodo de muestreo utilizado, *cero* el número de ceros que pedimos en la Función de Transferencia Discreta, *polos* el número de polos que pedimos en la Función de Transferencia Discreta y *retardo* el retraso presentado por dicha Función (este último se estima 1 en todos los experimentos).

De la variable *M* se pueden obtener los vectores que contienen los coeficientes de la Función de Transferencia.

$$[A, B, C, D, F] = \text{polydata}(M)$$

Para el caso de Output Error la Función de Transferencia Digital se obtiene de la expresión:

$$y(t) = \frac{B(q)}{F(q)} u(t - nk) + e(t)$$

Para obtener la Función de Transferencia en base a la cual se van a realizar las distintas validaciones hay que realizar todo lo que se ha explicado anteriormente. Todas las funciones de transferencia que se generan, se deben guardar tal y como las expresa Matlab ya que se comprobó que si tomamos los datos a mano truncando en la tercera o cuarta cifra decimal, el resultado que se obtiene posteriormente no es correcto. Esto es debido a los tiempos de muestreo tan pequeños que tenemos que utilizar.

Tras haber identificado la Función de Transferencia que se va a utilizar como modelo hay que validarla: por cada una de ellas se generan tres nuevas entradas y se obtienen sus salidas correspondientes tanto en la maqueta como en el modelo identificado. Esto último se hace con la orden de Matlab.



$$y = \text{lsim}(G, \text{entrada}, T', \text{zoh}')$$

Donde G es la Función de Transferencia del modelo identificado.

Tras haber calculado la salida simulada en Matlab (y), ésta se va a comparar con la salida obtenida de la maqueta (salida), siendo las dos salidas obtenidas de la misma entrada, logrando de esta forma calcular el error que se comete.

$$\text{error} = 0,5 \cdot (\text{salida}' - y)' \cdot (\text{salida}' - y)$$

El error que se obtiene es el sumatorio de los errores cuadráticos que se dan en cada una de las muestras. Por ello, cuando el número de muestras es mayor, el error que se obtiene es superior. Para poder comparar los resultados obtenidos con diferentes tiempos de muestreo, se hace una media, dividiendo el ruido que se ha obtenido, entre el número de muestras totales de la simulación.

Como ya se ha dicho, otra medida de la bondad de la identificación es el estudio de la autocorrelación del ruido estimado, explicado en el capítulo anterior. En el caso del Output Error esta estimación coincide con el error medido.

El estudio de la autocorrelación del ruido estimado se realiza de la forma citada en el apartado de conocimientos previos y de la misma manera que en el caso del error es necesario calcular la media, para posteriormente poder comparar distintas simulaciones. Tanto el error como la autocorrelación, no se comienzan a cuantificar desde el inicio de la simulación, ya que las señales están afectadas por las condiciones iniciales, por lo que se comienza a cuantificar el ruido a partir de los 0,1 segundos.

Para cada uno de los periodos de muestreo se han identificado tres funciones de transferencia y utilizando cada una de ellas se han realizado tres validaciones, obteniendo sus correspondientes errores y autocorrelaciones del ruido estimado. En la siguiente tabla se indican los mejores resultados obtenidos (marcamos en negrita el mejor de cada caso), esto es, indicamos las identificaciones que nos han dado un menor ruido y menor correlación.



Periodo de muestreo (seg)	0,0007	0,0009	0,001	0,003	0,005	0,007
error 1	5,1176	3,2626	2,9036	0,9568	0,5028	0,3719
error medio 1	7,16E-04	5,48E-04	5,81E-04	5,74E-04	5,00E-04	5,20E-04
autocorrelación 1	6,2658	3,0437	2,679	0,6057	0,1883	0,169
autocorrelación media 1	8,77E-04	5,48E-04	5,35E-04	3,63E-04	1,88E-04	2,36E-04
error 2	4,563	3,7464	3,3505	0,9792	0,5659	0,3919
error medio 2	6,39E-04	6,74E-04	6,70E-04	5,87E-04	5,65E-04	5,48E-04
autocorrelación 2	3,871	4,2365	4,6341	0,6024	0,2095	0,1257
autocorrelación media 2	6,82E-04	7,62E-04	9,26E-04	3,60E-04	2,10E-04	1,76E-04

Tabla 10: Representación de errores y autocorrelaciones

Las autocorrelaciones que se han obtenido tienen un valor cercano a cero, lo que quiere decir que el ruido que puede tener la maqueta puede ser modelado como del tipo estimado por modelo Output Error (error del sensor). Los errores tampoco son muy elevados por lo que podríamos aceptar este método como bueno, pero antes se van a analizar el número de polos y ceros y su localización.

Los errores que se obtienen son mayores cuanto más se aleja el tiempo de muestreo del valor que ya habíamos estimado como mejor en simulación: 5 milésimas de segundo

De cada uno de los periodos de muestreo vamos a elegir el mejor caso, el que ésta marcado en negrita, en el que el ruido y la correlación son menores y vamos a analizar tanto el número de polos y ceros y su localización, como la comparación de las salidas.

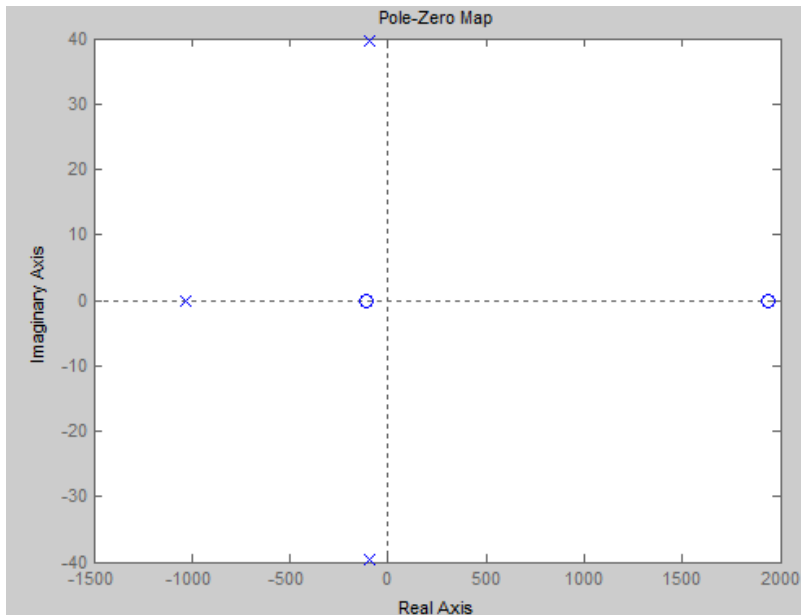
3.1.1.1 T=0,0007 segundos

La Función de Transferencia correspondiente al mejor caso con este periodo de muestreo es:

$$\frac{-46.16 s^2 + 8.431e004 s + 9.611e006}{s^3 + 1216 s^2 + 1.983e005 s + 1.021e007}$$



Tras haber obtenido la Función de Transferencia se analizan los polos y los ceros utilizando el comando “*pzmap*”:



```
p =
    1.0e+003 *
    -1.0339
    -0.0911 + 0.0397i
    -0.0911 - 0.0397i

z =
    1.0e+003 *
    1.9339
    -0.1077
```

En cuanto a los polos y ceros, vemos que hay un polo y un cero muy dominados y que el sistema es prácticamente de segundo orden con dos polos en -91.1 con una pequeña parte imaginaria que no aporta una sobreoscilación estimable (menor del 1%) y un cero en -107.7.

La ganancia del sistema es 0.9413

La última comprobación a realizar es comprobar las salidas, tanto la del motor como la procedente del Matlab, con respecto a la entrada.

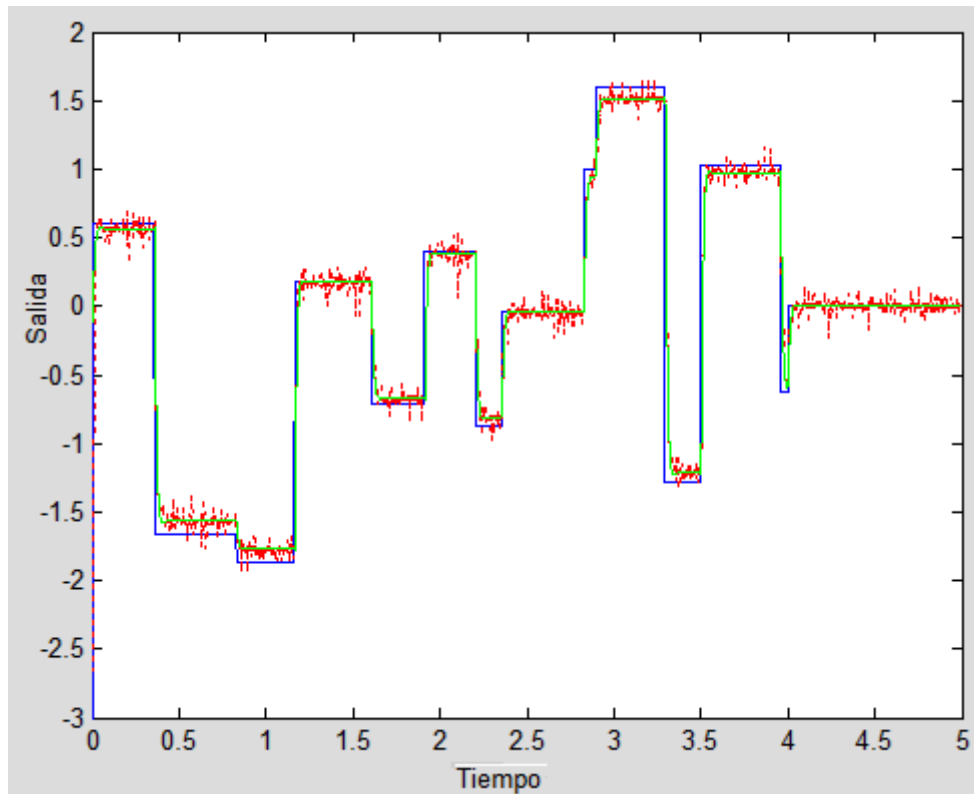


Figura 24: Representación de salidas y la entrada

La diferencia entre la señal de salida del motor y la salida procedente de la Función de Transferencia identificada no se puede apreciar en la gráfica de la parte superior. En la gráfica que se muestra a continuación, se va a representar la diferencia entre ambas señales. Dicha gráfica no va a ser representada desde el inicio de la simulación, ya que queremos evitar el efecto de las condiciones iniciales, por lo que comenzamos a partir de los 0,2 segundos.

Ya sabemos que en este caso este error es la estimación del ruido que además sabemos que está prácticamente desconexionado.

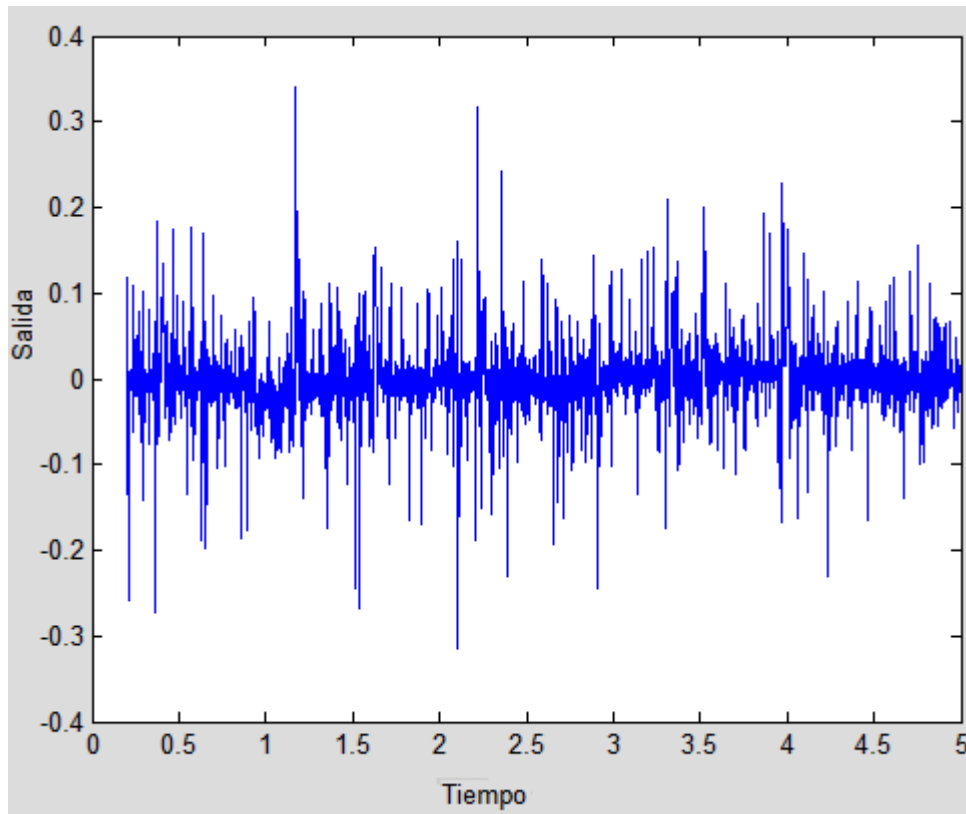


Figura 25: Representación del error

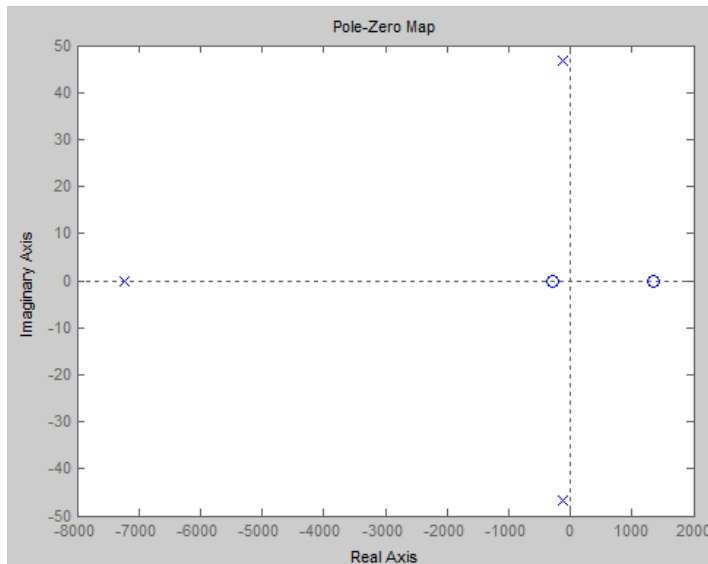
3.1.1.2 $T=0,0009$ segundos

La Función de Transferencia correspondiente al mejor caso con este periodo de muestreo es:

$$\frac{-345.6 \text{ s}^2 + 3.64\text{e}005 \text{ s} + 1.322\text{e}008}{\text{s}^3 + 7482 \text{ s}^2 + 1.914\text{e}006 \text{ s} + 1.4\text{e}008}$$



Tras haber obtenido la Función de Transferencia se analizan los polos y los ceros:



```
p =
    1.0e+003 *
    -7.2195
    -0.1312 + 0.0466i
    -0.1312 - 0.0466i

z =
    1.0e+003 *
    1.3390
   -0.2856
```

En cuanto a los polos y ceros, vemos que hay un polo y un cero muy dominados y que el sistema es prácticamente de segundo orden con dos polos en -131.1 con una pequeña parte imaginaria que no aporta una sobreoscilación estimable (menor del 0.015%) y un cero en -285.7.

La ganancia del sistema es 0.9443, similar a la anterior

La última comprobación a realizar es comprobar las salidas, tanto la del motor como la procedente del Matlab, con respecto a la entrada.

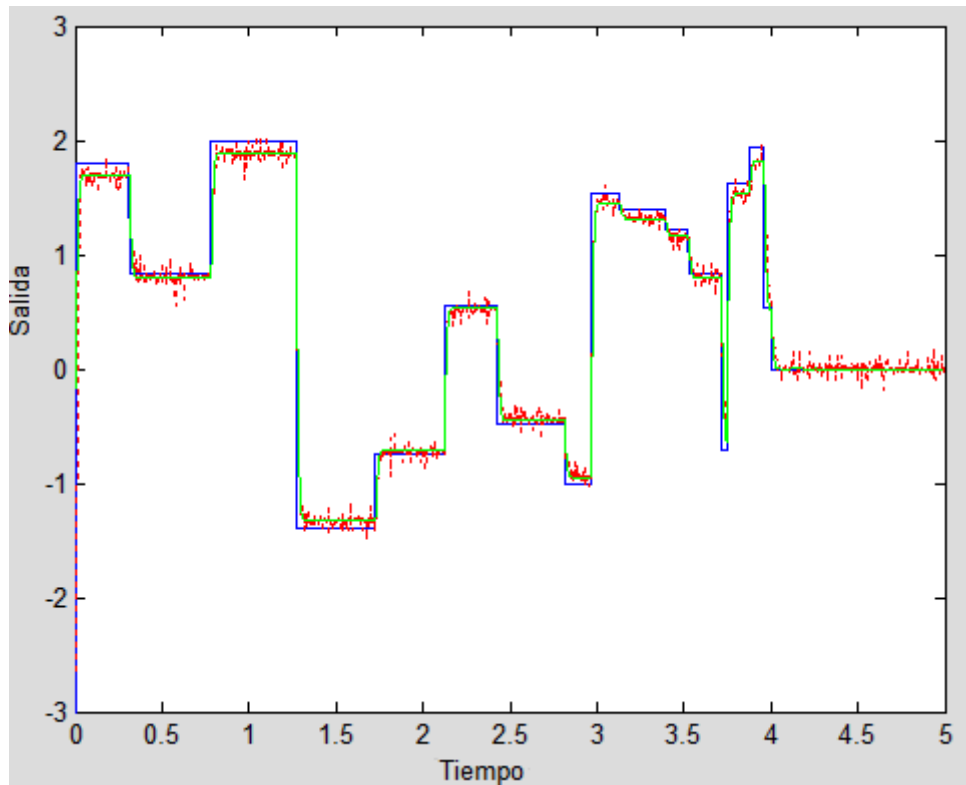


Figura 26: Representación de las salidas con respecto a la entrada

De nuevo la salida queda bien identificada.

La diferencia entre la señal de salida del motor y la salida procedente de la Función de Transferencia identificada no se puede apreciar en la gráfica de la parte superior. En la gráfica que se muestra a continuación, se va a representar la diferencia entre ambas señales. Dicha gráfica no va a ser representada desde el inicio de la simulación, ya que queremos evitar el efecto de las condiciones iniciales, por lo que comenzamos a partir de los 0,2 segundos.

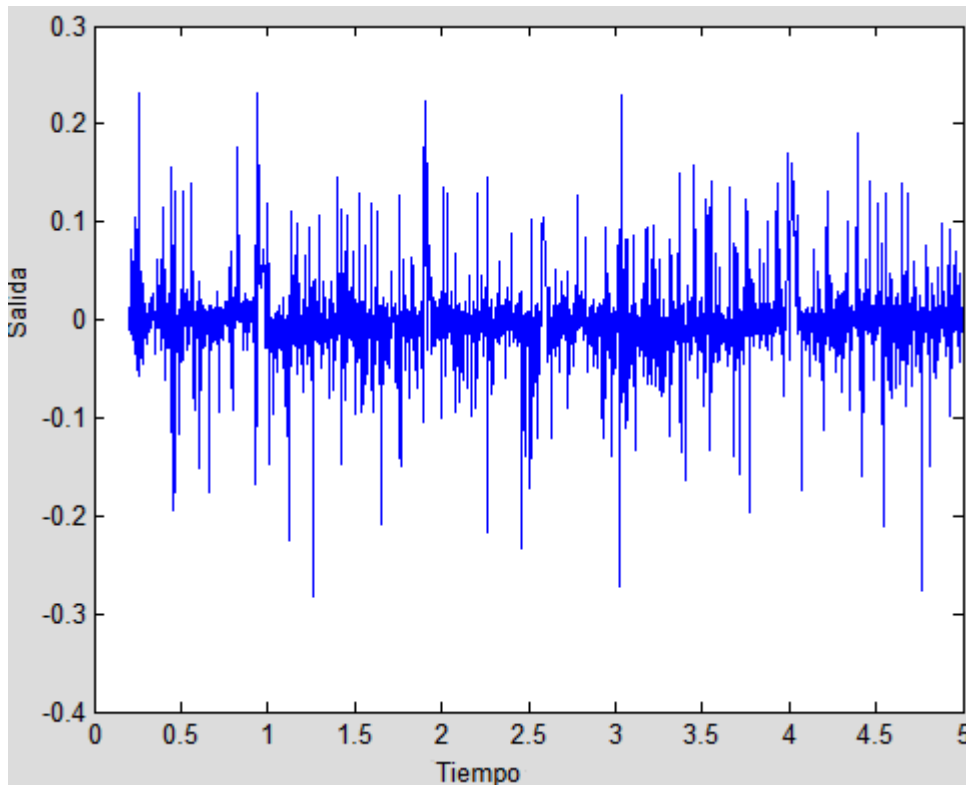


Figura 27: Representación del error

Como vemos se confirma la estimación de órdenes y ganancia del sistema aunque hay imprecisión en el valor de los polos y, sobre todo, del cero. Esto último es de esperar ya que su efecto es muy pequeño al estar claramente a la izquierda de los polos. Además, ya hemos comprobado la dificultad de identificar los ceros en sistemas ruidosos.

3.1.1.3 $T=0,001$ segundos

La Función de Transferencia correspondiente al mejor caso con este periodo de muestreo es:

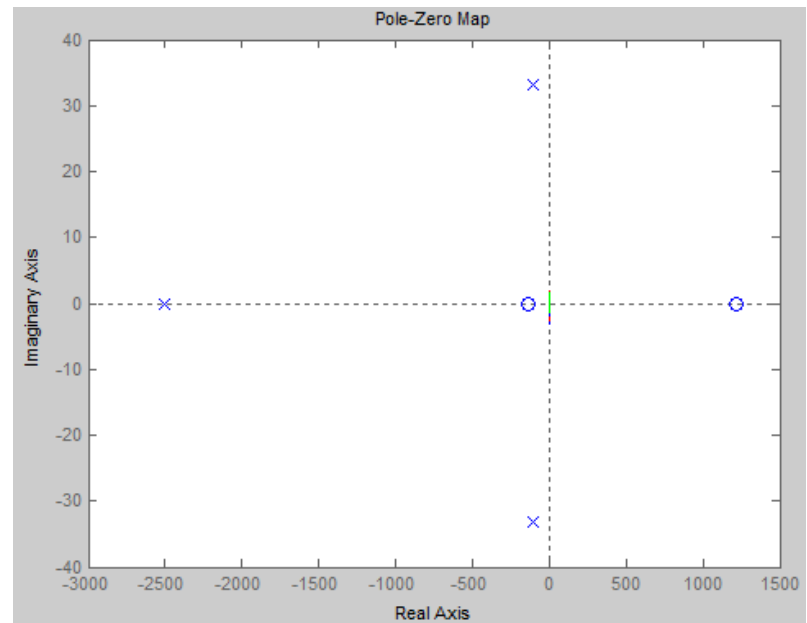
$$\frac{-165.5 s^2 + 1.784e005 s + 2.755e007}{s^3 + 2708 s^2 + 5.243e005 s + 2.9e007}$$



Tras haber obtenido la Función de Transferencia se analizan los polos y ceros:

```
p =
1.0e+003 *
-2.5028
-0.1024 + 0.0330i
-0.1024 - 0.0330i

z =
1.0e+003 *
1.2154
-0.1370
```



De nuevo, estudiando los polos y ceros, vemos que hay un polo y un cero muy dominados y que el sistema es prácticamente de segundo orden con dos polos en -102.4 con una pequeña parte imaginaria que no aporta una sobreoscilación estimable (menor del 0.06%) y un cero en -137.

La ganancia del sistema es 0.95, similar a la anterior

La última comprobación a realizar es comprobar las salidas, tanto la del motor como la procedente del Matlab, con respecto a la entrada.

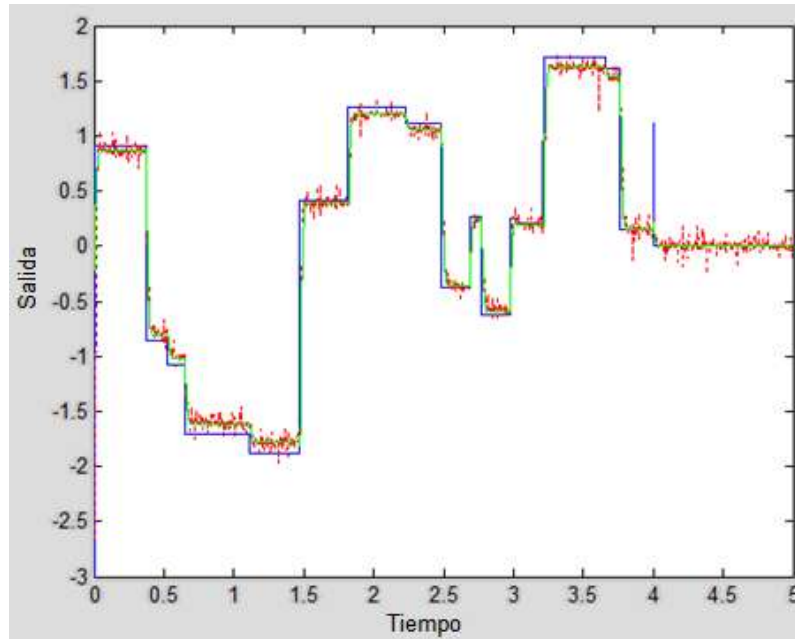


Figura 28: Representación de las salidas respecto de la entrada

La diferencia entre la señal de salida del motor y la salida procedente de la Función de Transferencia identificada no se puede apreciar en la gráfica de la parte superior. En la gráfica que se muestra a continuación, se va a representar la diferencia entre ambas señales. Dicha gráfica no va a ser representada desde el inicio de la simulación, ya que queremos evitar el efecto de las condiciones iniciales, por lo que comenzamos a partir de los 0,2 segundos.

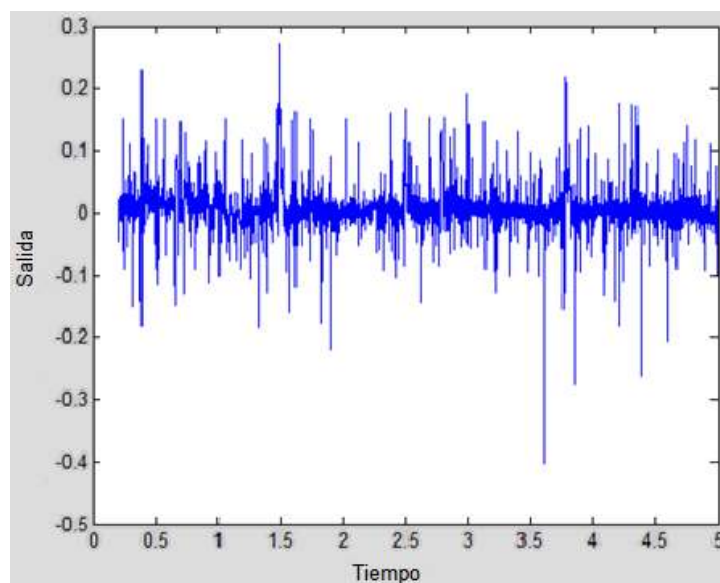


Figura 29: Representación del error



Se sigue confirmando la estimación de órdenes y ganancia del sistema aunque hay imprecisión en el valor de los polos y, sobre todo, del cero.

3.1.1.4 T=0,003 segundos

La Función de Transferencia correspondiente al mejor caso con este periodo de muestreo es:

$$\frac{53.27 s^3 - 3.872e004 s^2 + 1.035e008 s + 1.248e010}{s^4 + 1028 s^3 + 1.442e006 s^2 + 2.52e008 s + 1.317e010}$$

Lo que da una ganancia de 0.9476.

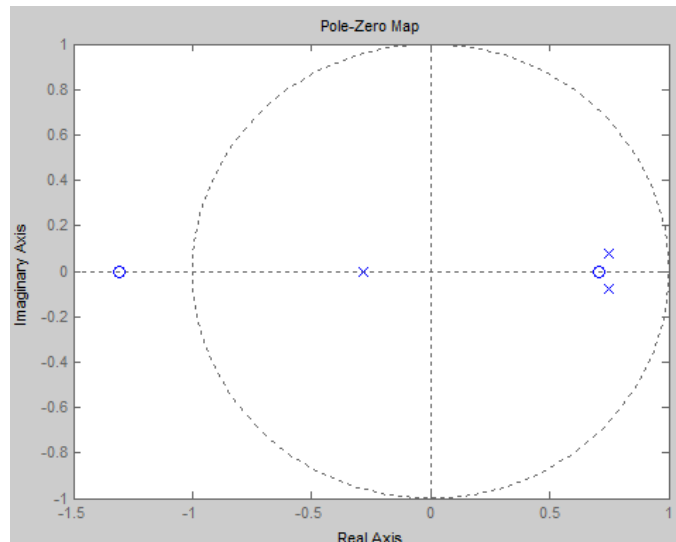
Sin embargo, vemos que ha aumentado el número de polos y ceros con respecto a lo pedido. Esto es consecuencia de que la Función de Transferencia Digital tiene un polo negativo.

$$\frac{0.1275 z^2 + 0.07608 z - 0.1178}{z^3 - 1.208 z^2 + 0.1376 z + 0.1606}$$

Si identificamos los polos y los ceros de la Función de Transferencia Digital los resultados obtenidos son los siguientes:

```
p =
    0.7464 + 0.0777i
    0.7464 - 0.0777i
   -0.2851

z =
   -1.3051
    0.7083
```





Se ha comprobado que la Función de Transferencia Digital tiene un polo negativo. Por lo tanto este experimento no es válido ya que se considera que está mal muestreado para este modelo (Output Error) ya que lo ve como pulsante.

3.1.1.5T=0,005 segundos

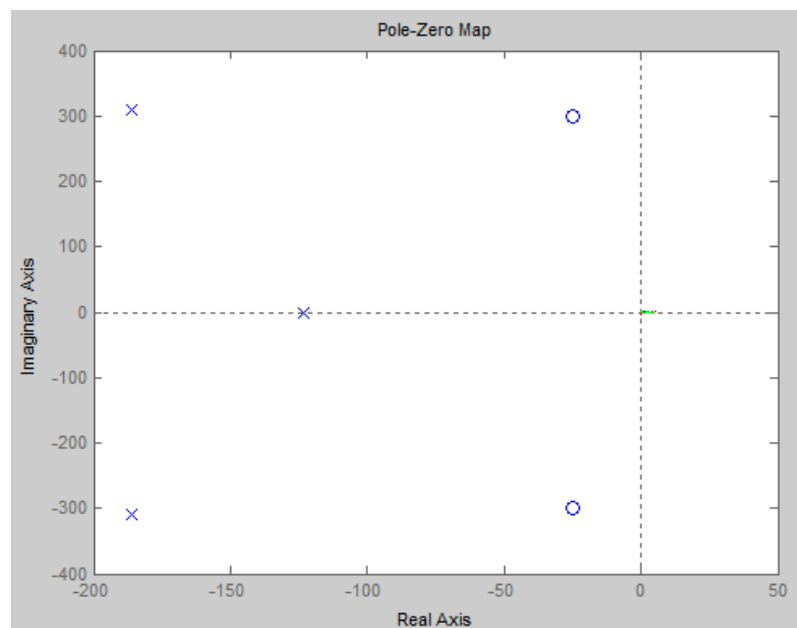
La Función de Transferencia correspondiente al mejor caso con este periodo de muestreo es:

$$\frac{167.9 s^2 + 8285 s + 1.524e007}{s^3 + 495.1 s^2 + 1.766e005 s + 1.609e007}$$

Tras haber obtenido la Función de Transferencia se analizan los polos y los ceros:

```
p =
1.0e+002 *
-1.8604 + 3.1014i
-1.8604 - 3.1014i
-1.2299

z =
1.0e+002 *
-0.2467 + 3.0024i
-0.2467 - 3.0024i
```



La última comprobación a realizar es comprobar las salidas, tanto la del motor como la procedente del Matlab, con respecto a la entrada.

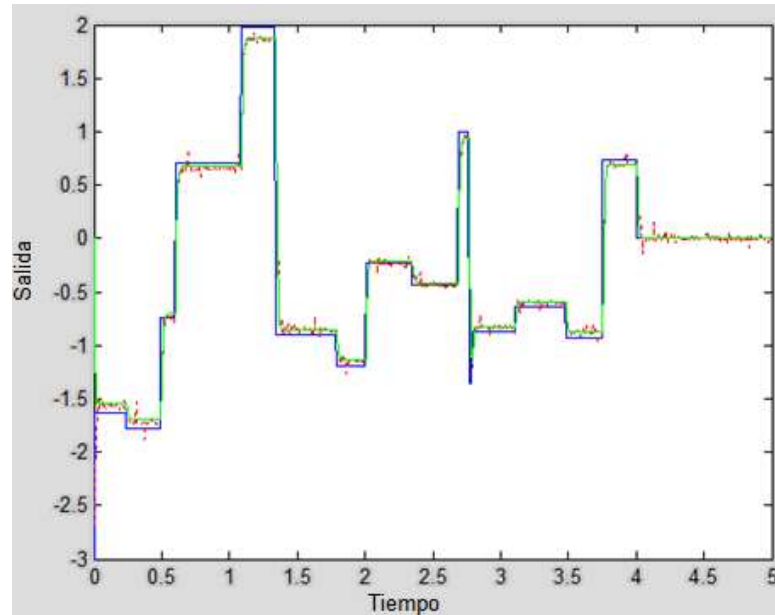


Figura 30: Representación de las salidas respecto de la entrada

La diferencia entre la señal de salida del motor y la salida procedente de la Función de Transferencia identificada no se puede apreciar en la gráfica de la parte superior. En la gráfica que se muestra a continuación, se va a representar la diferencia entre ambas señales. Dicha gráfica no va a ser representada desde el inicio de la simulación, ya que queremos evitar el efecto de las condiciones iniciales, por lo que comenzamos a partir de los 0,2 segundos.

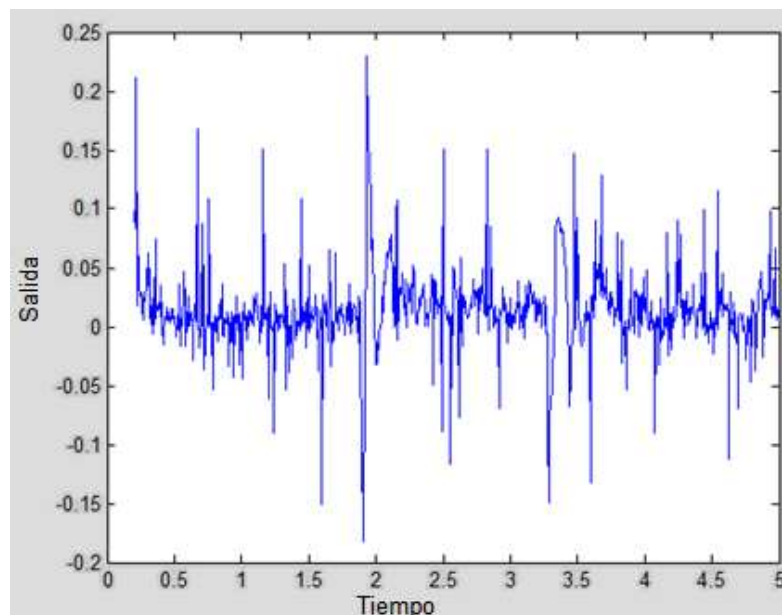


Figura 31: Representación del error



Como se puede observar en el mapa de polos y ceros, la localización de éstos se da en lugares poco comunes comparando con las anteriores simulaciones. Por eso, vamos a realizar una nueva simulación y poder así certificar si la identificación está correctamente realizada o no lo está.

Segunda simulación

La Función de Transferencia correspondiente al mejor caso con este periodo de muestreo es:

$$\frac{35.93 s^3 + 8.001e004 s^2 + 2.901e007 s + 1.143e010}{s^4 + 974.6 s^3 + 7.319e005 s^2 + 1.544e008 s + 1.206e010}$$

Lo que da una ganancia de 0.9478.

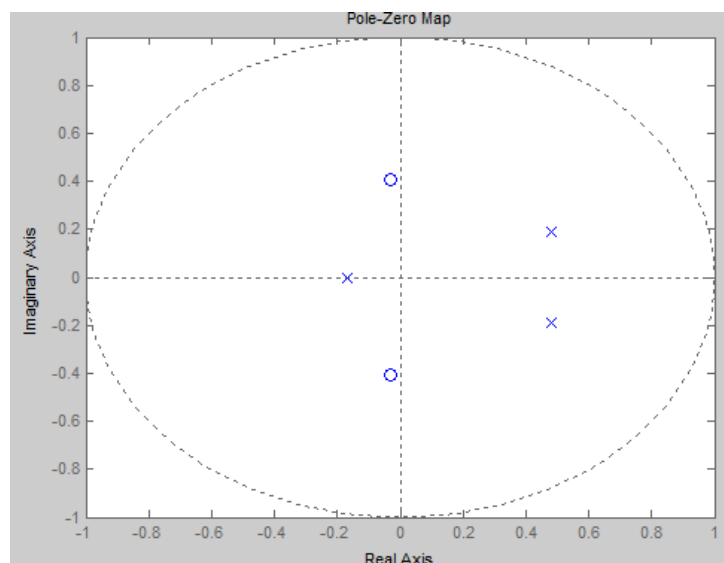
Sin embargo, vemos que ha aumentado el número de polos y ceros con respecto a lo pedido. Esto es consecuencia de que la Función de Transferencia Digital tiene un polo negativo.

$$\frac{0.2758 z^2 + 0.01719 z + 0.04627}{z^3 - 0.7902 z^2 + 0.1031 z + 0.0451}$$

Si identificamos los polos y los ceros de la Función de Transferencia Digital los resultados obtenidos son los siguientes:

```
p =
0.4799 + 0.1887i
0.4799 - 0.1887i
-0.1696

z =
-0.0312 + 0.4084i
-0.0312 - 0.4084i
```





Se ha comprobado que la Función de Transferencia Digital tiene un polo negativo. Por lo tanto este experimento no es válido ya que se considera que está mal muestreado para este modelo (Output Error) ya que lo ve como pulsante.

Las simulaciones con este periodo de muestreo no son válidas, ya que en el primero los polos y los ceros no estaban colocados adecuadamente y en el segundo caso el experimento no es válido, considerando que está mal muestreado. Por lo tanto no se consigue un resultado similar a los anteriores a este caso.

3.1.1.6T=0,007 segundos

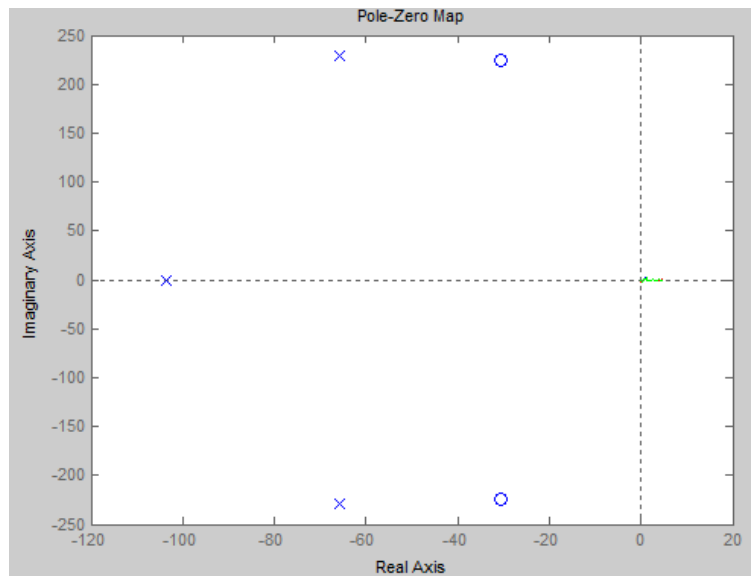
La Función de Transferencia correspondiente al mejor caso con este periodo de muestreo es:

$$\frac{109.1 s^2 + 6706 s + 5.59e006}{s^3 + 235.2 s^2 + 7.032e004 s + 5.869e006}$$

Tras haber obtenido la Función de Transferencia se analizan los polos y los ceros:

```
p =
1.0e+002 *
-0.6582 + 2.2883i
-0.6582 - 2.2883i
-1.0351

z =
1.0e+002 *
-0.3075 + 2.2430i
-0.3075 - 2.2430i
```



La última comprobación a realizar es comprobar las salidas, tanto la del motor como la procedente del Matlab, con respecto a la entrada.

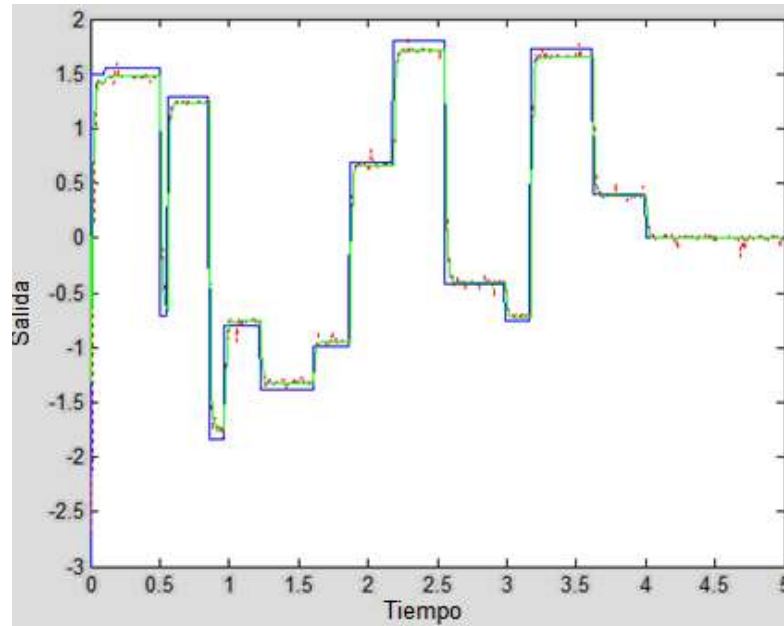


Figura 32: Representación de las salidas respecto de la entrada

La diferencia entre la señal de salida del motor y la salida procedente de la Función de Transferencia identificada no se puede apreciar en la gráfica de la parte superior. En la gráfica que se muestra a continuación, se va a representar la diferencia entre ambas señales. Dicha gráfica no va a ser representada desde el inicio de la simulación, ya que queremos evitar el efecto de las condiciones iniciales, por lo que comenzamos a partir de los 0,2 segundos.

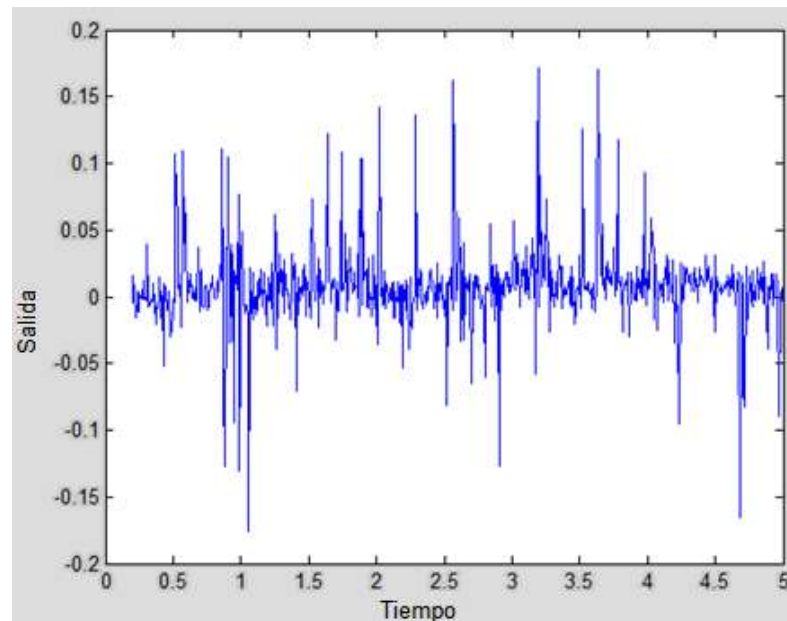


Figura 33: Representación del error



Como se puede observar en el mapa de polos y ceros, la localización de éstos se da en lugares más lentos, polos más a la derecha comparando con las anteriores simulaciones. Por eso, vamos a realizar una nueva simulación y poder así certificar si la identificación está correctamente realizada o no lo está.

Segunda simulación

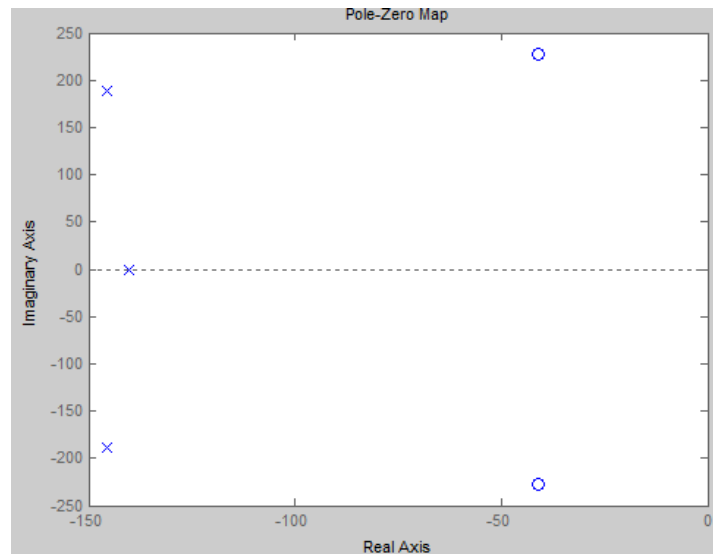
La Función de Transferencia correspondiente al mejor caso con este periodo de muestreo es:

$$\frac{141.1 s^2 + 1.154e004 s + 7.568e006}{s^3 + 431.7 s^2 + 9.769e004 s + 7.971e006}$$

Tras haber obtenido la Función de Transferencia se analizan los polos y los ceros:

```
p =
1.0e+002 *
-1.4570 + 1.8859i
-1.4570 - 1.8859i
-1.4034

z =
1.0e+002 *
-0.4089 + 2.2799i
-0.4089 - 2.2799i
```



La última comprobación a realizar es comprobar las salidas, tanto la del motor como la procedente del Matlab, con respecto a la entrada.

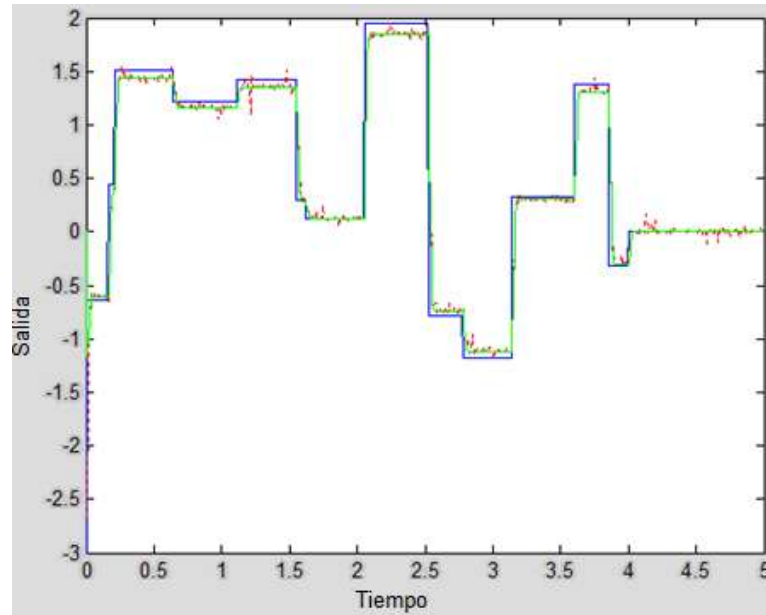


Figura 34: Representación de las salidas respecto de la entrada

La diferencia entre la señal de salida del motor y la salida procedente de la Función de Transferencia identificada no se puede apreciar en la gráfica de la parte superior, aunque es posible ver errores en algunos transitorios, en incluso en permanentes. En la gráfica que se muestra a continuación, se va a representar la diferencia entre ambas señales. Dicha gráfica no va a ser representada desde el inicio de la simulación, ya que queremos evitar el efecto de las condiciones iniciales, por lo que comenzamos a partir de los 0,2 segundos.

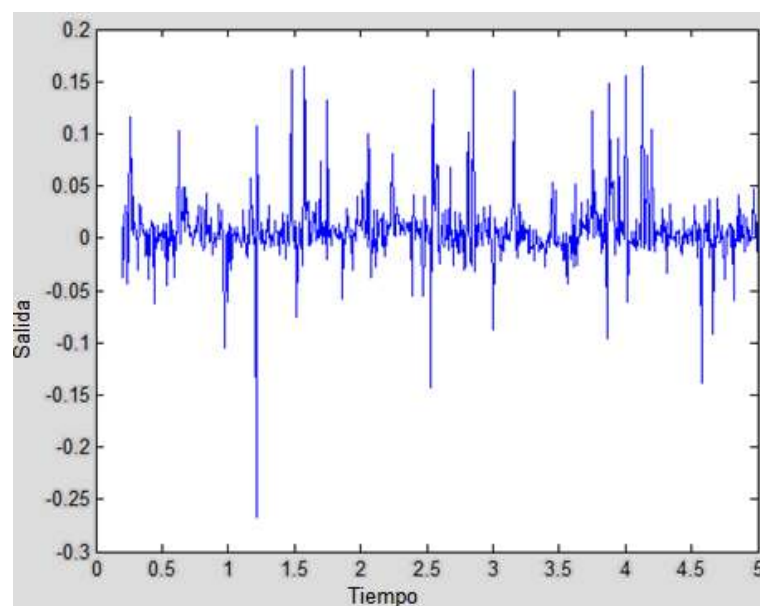


Figura 35: Representación del error



En este caso la situación es más parecida a las anteriores: aunque los polos sean más rápidos es un sistema triple, lo que hace que se identifique el sistema.

3.1.2 Disminución de orden: sistema de 2º orden

Del estudio anterior concluimos que una buena aproximación a los órdenes del sistema es suponer que se tienen **dos polos y un cero**: es lo que se obtiene en 3 de los 5 casos válidos, mientras que uno de los otros dos busca un sistema compatible con este resultado. Vamos a proceder a una nueva serie de experimentos bajo esta hipótesis para intentar fijar el valor de estos y la ganancia del sistema.

Para la realización de la identificación vamos a utilizar el mismo procedimiento que en el caso anterior, pero tendremos que cambiar la ecuación donde se hacen referencia al número de polos y ceros. Vamos a analizar sólo los valores de periodo de muestreo que dieron menores errores en las anteriores identificaciones. Los dos mejores datos (el mejor en negrita), menor ruido y menor correlación, que se han obtenido de cada periodo de muestreo son los siguientes:

Periodo de muestreo (seg)	0,003	0,005	0,007
ruido 1	0,9882	0,493	0,4312
ruido medio 1	5,93E-04	4,90E-04	6,03E-04
autocorrelación 1	0,6413	0,2544	0,1213
autocorrelación media 1	3,85E-04	2,50E-04	1,70E-04
ruido 2	1,0141	0,5424	0,4764
ruido medio 2	6,10E-04	5,42E-04	6,66E-04
autocorrelación 2	0,7046	0,1911	0,1933
autocorrelación media 2	4,23E-04	1,90E-04	2,70E-04

Tabla 11: Representación del ruido de de la correlación

Como se ve, los resultados son similares a los de la tabla10. Por lo que podemos aceptar que la identificación no se degrada al reducir el orden del sistema identificado.

Las correlaciones que se han obtenido tienen un valor cercano a cero, lo que quiere decir que el ruido que puede tener la maqueta puede ser del modelo Output Error. Los errores tampoco son muy elevados por lo que podríamos aceptar este



método como bueno, pero antes se van a analizar el número de polos y ceros y su localización.

De cada uno de los periodos de muestreo vamos a elegir el mejor caso, el que ésta marcado en **negrita**, en el que el ruido y la correlación son menores y vamos a analizar tanto el número de polos y ceros y su localización, como la comparación de las salidas.

3.1.2.1 T=0,003 segundos

La Función de Transferencia correspondiente al mejor caso con este periodo de muestreo es:

$$\frac{-6.871 s^2 - 1.137e005 s + 1.707e008}{s^3 + 1618 s^2 + 1.83e006 s + 1.814e008}$$

Lo que da una ganancia de 0.941.

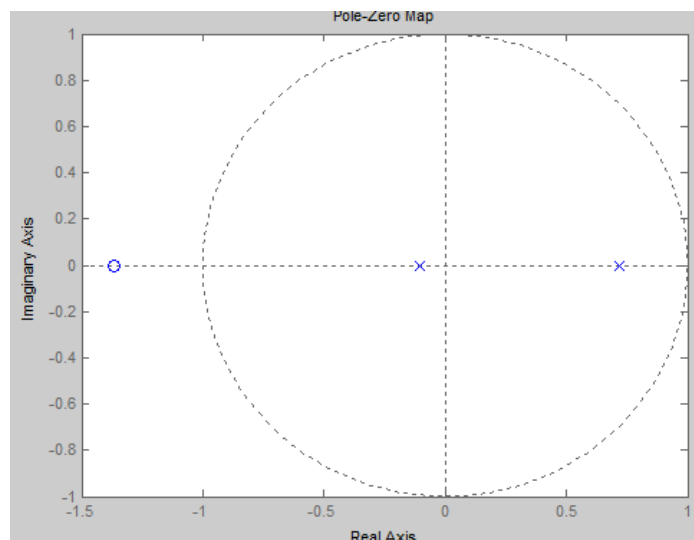
Sin embargo, vemos que ha aumentado el número de polos y ceros con respecto a lo pedido. Esto es consecuencia de que la Función de Transferencia Digital tiene un polo negativo.

$$\frac{0.1224 z + 0.1671}{z^2 - 0.6173 z - 0.07505}$$

Si identificamos los polos y los ceros de la Función de Transferencia Digital los resultados obtenidos son los siguientes:

$$p = \begin{matrix} 0.7213 \\ -0.1040 \end{matrix}$$

$$z = \begin{matrix} -1.3650 \end{matrix}$$





Se ha comprobado que la Función de Transferencia Digital tiene un polo negativo. Por lo tanto este experimento no es válido ya que se considera que está mal muestreado para este modelo (Output Error) ya que lo ve como pulsante.

3.1.2.2 T=0,005 segundos

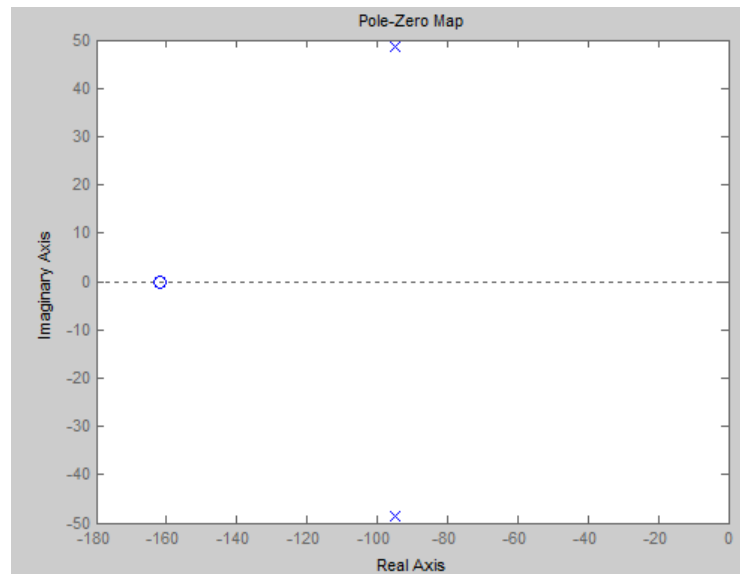
La Función de Transferencia correspondiente al mejor caso con este periodo de muestreo es:

$$\frac{66.35 s + 1.075e004}{s^2 + 189.7 s + 1.137e004}$$

Tras haber obtenido la Función de Transferencia se analizan los polos y los ceros:

```
p =
-94.8652 +48.6625i
-94.8652 -48.6625i

z =
-161.9915
```



Estudiando los polos y ceros vemos que los polos están -94.865 con una pequeña parte imaginaria que no aporta una sobreoscilación estimable (menor del 0.2%) y el cero en -162.

La ganancia del sistema es 0.9455, similar los todos los casos anteriores.

La última comprobación a realizar es comprobar las salidas, tanto la del motor como la procedente del Matlab, con respecto a la entrada.

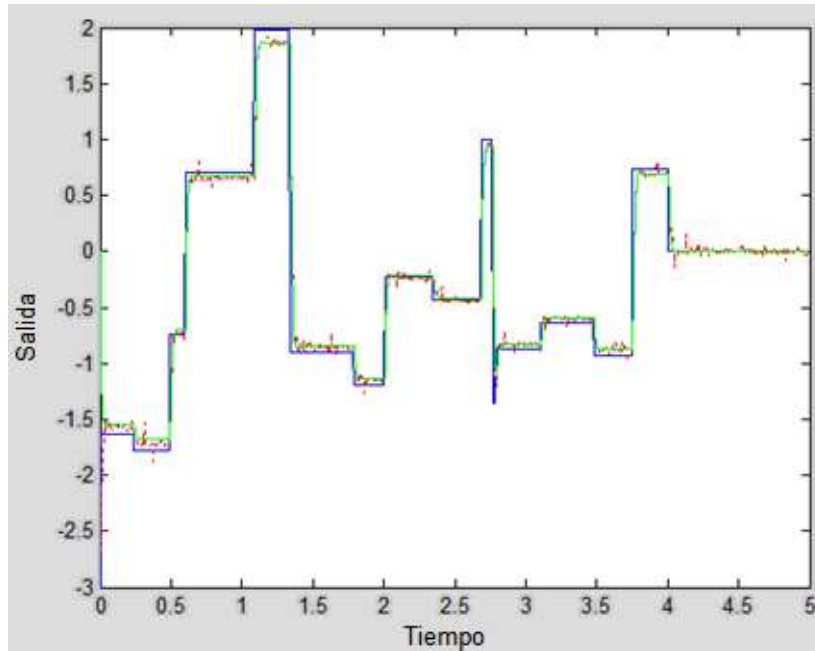


Figura 36: Representación gráfica de las salidas respecto de la entrada

La diferencia entre la señal de salida del motor y la salida procedente de la Función de Transferencia identificada no se puede apreciar en la gráfica de la parte superior. En la gráfica que se muestra a continuación, se va a representar la diferencia entre ambas señales. Dicha gráfica no va a ser representada desde el inicio de la simulación, ya que queremos evitar el efecto de las condiciones iniciales, por lo que comenzamos a partir de los 0,2 segundos.

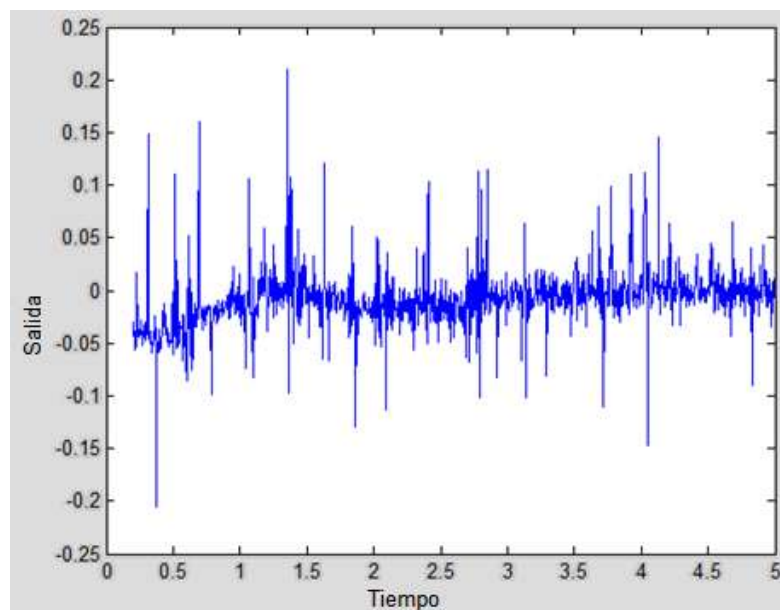


Figura 37: Representación del error

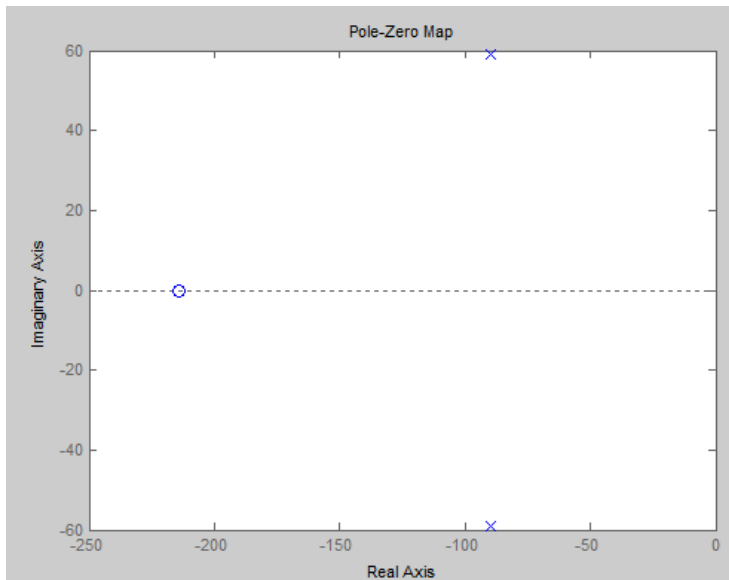


3.1.2.3 T=0,007 segundos

La Función de Transferencia correspondiente al mejor caso con este periodo de muestreo es:

$$\frac{51.01 s + 1.093e004}{s^2 + 179.2 s + 1.153e004}$$

Tras haber obtenido la Función de Transferencia se analizan los polos y los ceros:



```
p =
-89.6243 +59.1026i
-89.6243 -59.1026i

z =
-214.2327
```

Estudiando los polos y ceros vemos que los polos están -89.62 con una pequeña parte imaginaria que no aporta una sobreoscilación estimable (menor del 1%) y el cero en -214.

La ganancia del sistema es 0.9448, similar los todos los casos anteriores.

La última comprobación a realizar es comprobar las salidas, tanto la del motor como la procedente del Matlab, con respecto a la entrada.

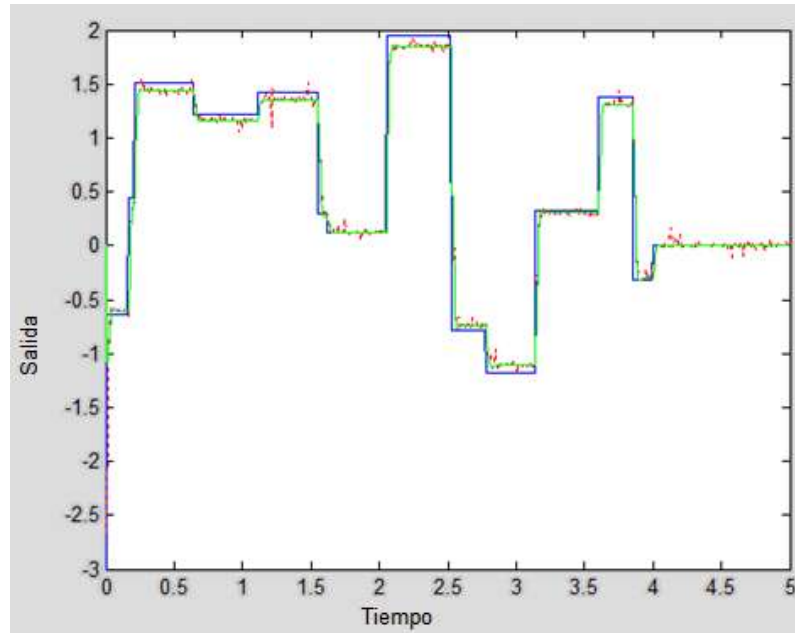


Figura 38: Representación de las salidas respecto de la entrada

La diferencia entre la señal de salida del motor y la salida procedente de la Función de Transferencia identificada no se puede apreciar en la gráfica de la parte superior. En la gráfica que se muestra a continuación, se va a representar la diferencia entre ambas señales. Dicha gráfica no va a ser representada desde el inicio de la simulación, ya que queremos evitar el efecto de las condiciones iniciales, por lo que comenzamos a partir de los 0,2 segundos.

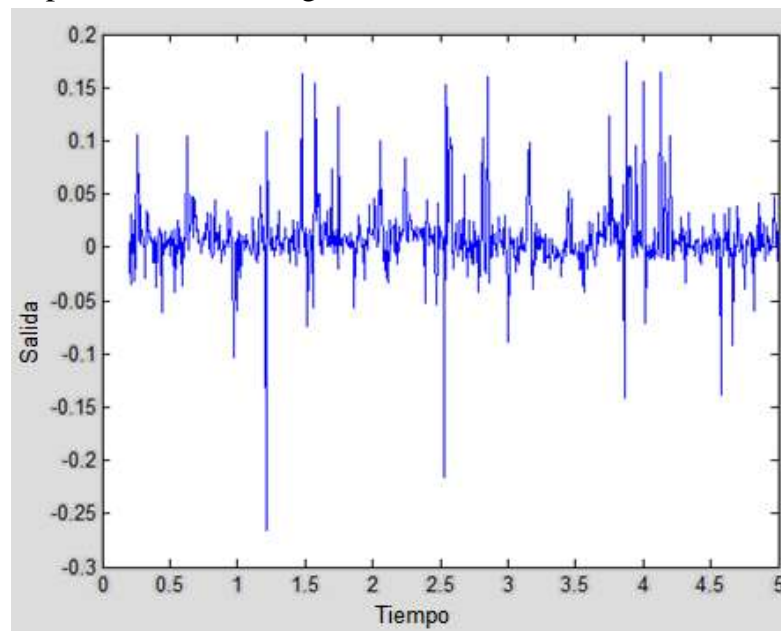


Figura 39: Representación del error



Del análisis anterior podemos concluir que el sistema se puede aproximar por uno de segundo orden con un cero con los polos en $-90 \pm 60i$ y el cero, de forma menos precisa, en -170 con una ganancia de 0.946 :

$$G(s) = 65.2447 \frac{s + 170}{s^2 + 180s + 11700}$$

La ganancia del sistema es 0.946 , similar los todos los casos anteriores.

Todos los datos obtenidos son los que se han utilizado para crear la Función de Transferencia.

La última comprobación a realizar es comprobar las salidas, tanto la del motor como la procedente del Matlab, con respecto a la entrada. Si comprobamos la salida que de este sistema propuesto frente a la respuesta del motor obtener la gráfica de la *Figura 40*.

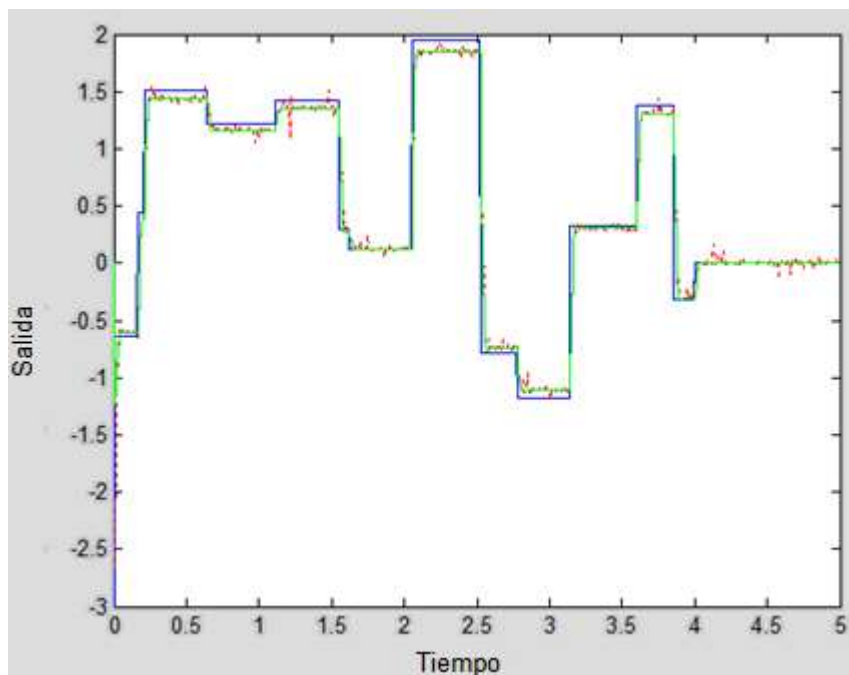


Figura 40: Representación de las salidas respecto de la entrada

La diferencia entre la señal de salida del motor y la salida procedente de la Función de Transferencia identificada no se puede apreciar en la gráfica de la parte superior. En la gráfica que se muestra a continuación, se va a representar la diferencia entre ambas señales. Dicha gráfica no va a ser representada desde el inicio de la



simulación, ya que queremos evitar el efecto de las condiciones iniciales, por lo que comenzamos a partir de los 0,2 segundos.

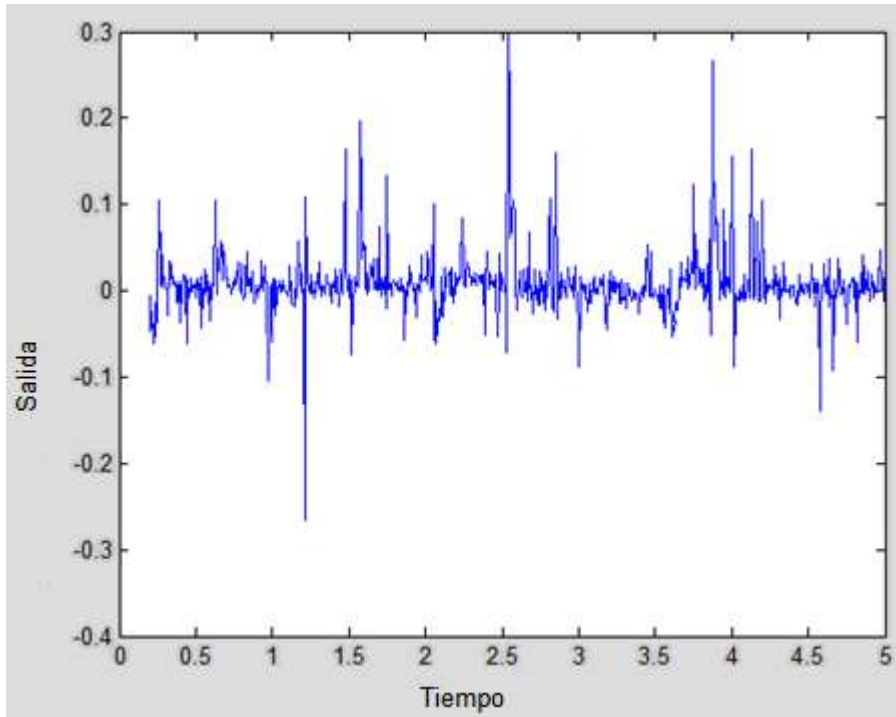


Figura 41: Representación del error

Vamos a comprobar el error y la correlación que se han obtenido:

$$error = 0.5078$$

$$autocorrelacion = 0.2927$$

El error y la correlación que se han obtenido tienen valores similares a los que se han obtenido en algunas simulaciones anteriores. Por ello, concluimos que la Función de Transferencia obtenida se puede corresponder como el modelo de la linealización de la maqueta.

Los resultados así obtenidos se han verificado reproduciendo los experimentos con otros tipos de entrada. En particular se redujo la anchura máxima de los escalones de forma que se excitarán otras dinámicas. Las conclusiones obtenidas de estos experimentos son totalmente compatibles con los mostrados.



3.2 MODELO ARX

El segundo modelo dinámico que se va a analizar es el modelo ARX.

La ecuación que rige este modelo es la siguiente:

$$y(t) = \frac{B(q)}{A(q)}u(t - nk) + \frac{1}{A(q)}e(t)$$

A partir de este modelo podemos estimar el ruido $e(k)$: Lo que hacemos es calcular la salida del sistema identificado (por lo tanto sin ruido), se le añade el ruido provocado en el modelo ARX, para ello hay que estimar la muestra del ruido en este instante de simulación, la mejor estimación asumiendo ruido blanco es cero, y tener en cuenta las muestras de ruido estimadas anteriormente. Una vez que hemos estimado la salida ruidosa se calcula la diferencia entre la salida del sistema real y la del simulado, este valor es la estimación de la muestra de ruido en ese instante.

El esquema que en Simulink es el siguiente:

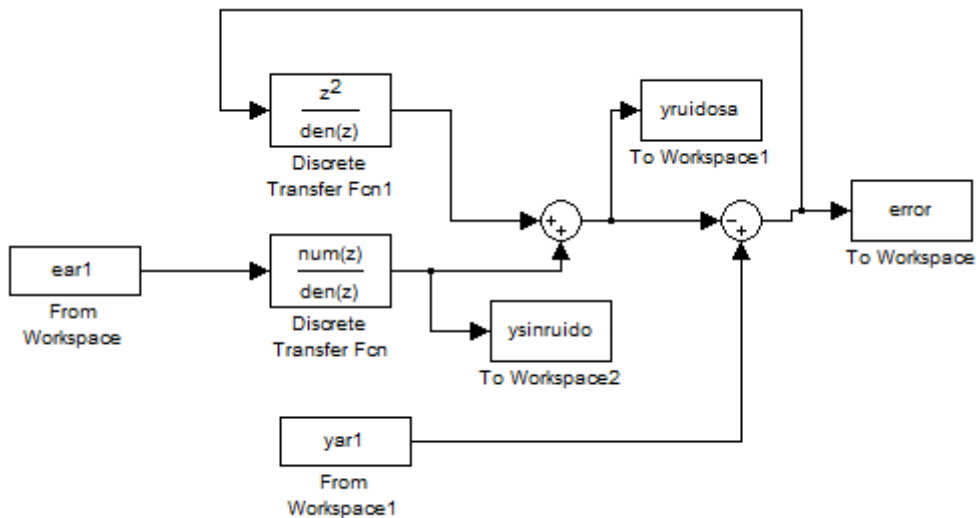


Figura 42: Esquema de un modelo ARX

Donde yar1 es la salida de la maqueta para la entrada ear1.



Las entradas que se crean para los procesos de identificación y verificación se definen de la misma forma que las entradas creadas para el modelo del Output Error. Dichas entradas van a tener periodos de muestreo que comienzan en las 3 milésimas de segundo hasta las 7 milésimas, pasando de milésima en milésima. De esta forma recorreremos de forma densa la zona para los que se alcanzan mejores resultados.

Con cada periodo de muestreo se hacen tres identificaciones de funciones de transferencia, con respecto a las cuales se van a realizar las validaciones. Las identificaciones se van a realizar de la misma manera que las que se han realizado en el Output Error, únicamente cambiando la siguiente ecuación:

$$Z = \text{iddata}(\text{salida}', \text{entrada}', T)$$

$$M = \text{arx}(Z, [\text{ceros} + 1 \text{ polos } \text{retardo}])$$

Donde *salida* y *entrada* son los datos obtenidos de la maqueta, *T* es periodo de muestreo utilizado, *ceros* es el número de ceros que pedimos en la Función de Transferencia Discreta, *polos* el número de polos que pedimos en la Función de Transferencia Discreta y *retardo* el retraso presentado por dicha Función (este último se estima 1 en todos los experimentos).

De la variable *M* se pueden obtener los vectores que contienen los coeficientes de la Función de Transferencia.

$$[A, B, C, D, F] = \text{polydata}(M)$$

En este modelo dinámico, para definir la Función de Transferencia de la maqueta, tras haber obtenido los vectores con los coeficientes, se seleccionan dos de los vectores: el A y el B, siendo situados como se indica la ecuación de los coeficientes, el vector A en el denominador y el B en el numerador de la Función de Transferencia del sistema. Además A es el denominador de la Función de Transferencia del error, que no tiene ceros

En el modelo ARX hay que definir la Función de Transferencia que tiene la maqueta y también la Función de Transferencia del error, ya que el error en un determinado momento depende del error que se ha producido anteriormente.



La realización del análisis de este modelo es más complicado que en el caso anterior. En primer lugar, hay que realizar la identificación de la Función de Transferencia del sistema para ello se proporciona a la maqueta la entrada, por medio de Labview, obteniendo así la señal de salida correspondiente. Tras haber obtenido la entrada y la salida procedente de la maqueta, a cada uno de los casos hay que restarles el punto de trabajo, que habrá que identificar de los datos de la maqueta, como se ha indicado anteriormente.

Una vez tenemos los datos preparados se calcula la identificación a través del toolbox de identificación de Matlab tal y como se ha explicado antes. Cuando ya tenemos la identificación se procede la validación del mismo. Para ello se introduce una nueva entrada en la maqueta se obtiene la salida. Después se calcula la salida del sistema identificado.

En este caso se necesita la ayuda del simulink, creando el esquema de la *Figura 42*, siendo las dos funciones de transferencia las que se han definido en la parte superior. Notemos que además del ruido también se va calculando la salida del sistema, tanto sin ruido como con la estimación del ruido. Al simulador se le proporciona la salida obtenida de la maqueta para que el propio sistema pueda calcular la diferencia entre las salidas de la maqueta y la que nos proporciona el modelo identificado.

Para obtener el error y la correlación no debemos a empezar a cuantificarlo desde el principio de la simulación, ya que, las primeras muestras están afectadas por las condiciones iniciales. Para evitar ruido valores de error muy elevados, comenzamos a tomar datos desde un tiempo de 0,1 segundos.

3.2.1 Sistema de 3º orden

El primer ejercicio que se va a realizar es considerando que el número de **polos del sistema es tres y el número de ceros es dos**. Lo que se pretende es calcular el error y la correlación en cada uno de los casos. De todos los datos obtenidos se eligen los dos mejores, el mejor en negrita, esto es, los que tengan menor ruido y menor correlación de cada periodo de muestreo, quedando reflejados en la siguiente tabla:



Periodo de muestreo (seg)	0,003	0,004	0,005	0,006	0,007
error 1	0,198	0,1393	0,1158	0,0932	0,0876
error medio 1	1,19E-04	1,11E-04	1,16E-04	1,12E-04	1,20E-04
autocorrelación 1	1,2374	1,0429	0,9361	1,0269	0,7652
autocorrelación media 1	7,40E-04	8,30E-04	9,35E-04	1,23E-03	1,07E-03
error 2	0,214	0,1565	0,1318	0,0932	0,0904
error medio 2	1,28E-04	1,25E-04	1,30E-04	1,12E-04	1,26E-04
autocorrelación 2	1,3199	0,7882	0,8135	1,116	0,6077
autocorrelación media 2	7,92E-04	6,30E-04	8,13E-04	1,34E-03	9,50E-04

Tabla 12: Representación de errores y autocorrelaciones

Como vemos el error medio es comparable al obtenido en el caso del Output ERROR y la autocorrelación es menor, como se podría esperar ya que se trata el error para conseguir que sea el resultado de un ruido blanco.

Las autocorrelaciones que se han obtenido tienen un valor cercano a cero, lo que quiere decir que el ruido que puede tener la maqueta puede ser el del modelo ARX. Los ruidos tampoco son muy elevados por lo que podríamos aceptar este método como bueno, pero antes se van a analizar el número de polos y ceros y su localización.

De cada uno de los periodos de muestreo vamos a elegir el mejor caso, el que está marcado en negrita, en el que el ruido y la correlación son menores y vamos a analizar tanto el número de polos y ceros y su localización, como la ganancia y la comparación de las salidas. Aunque haya periodos de muestreo en los que el caso de menor ruido que se obtiene no es el que se ha elegido, también hay que tener en cuenta la correlación obtenida que en esos casos será mejor que si hubiésemos elegido los datos de la otra identificación.

3.2.1.1 T=0,003 segundos

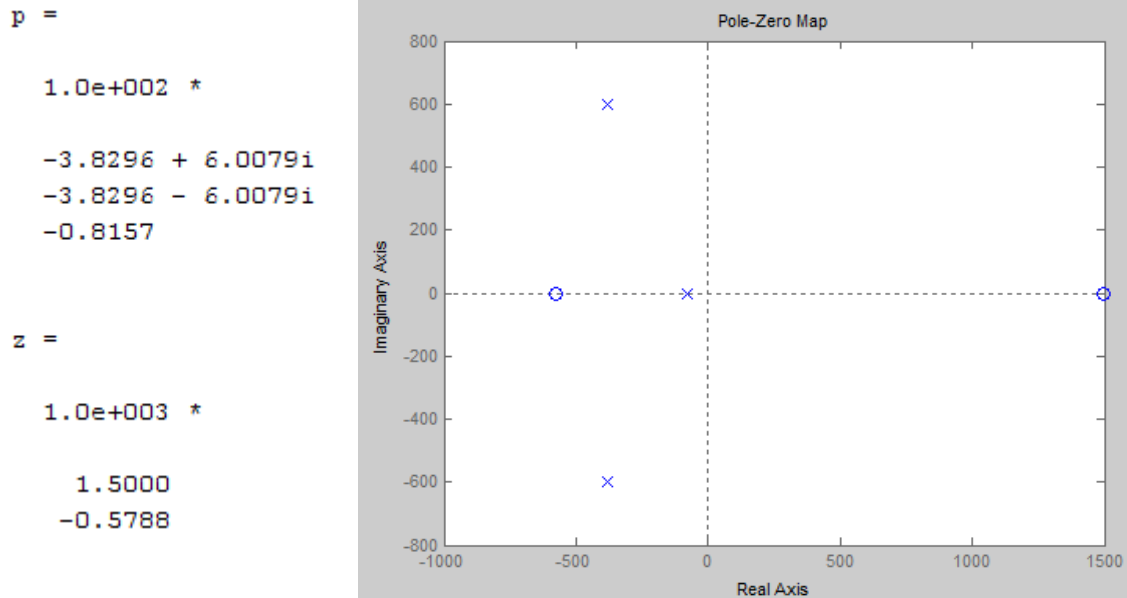
La Función de Transferencia correspondiente al mejor caso con este periodo de muestreo es:

$$\frac{-44.51 s^2 + 4.101e004 s + 3.865e007}{s^3 + 847.5 s^2 + 5.701e005 s + 4.141e007}$$

Lo que da una ganancia de 0.9333, compatible con lo obtenido con Output Error, aunque un poco más baja. Pero de la gráfica de comparación de salida se ve que la salida del sistema identificado queda un poco más bajo.



Tras haber obtenido la Función de Transferencia se analizan los polos y los ceros utilizando el comando “*pzmap*”:



Dos polos y dos ceros están dominados, el sistema se comportaría como un primer orden con un polo en -81.57 .

La última comprobación a realizar es comparar las salidas, tanto la del motor como la procedente del Matlab, con respecto a la entrada.

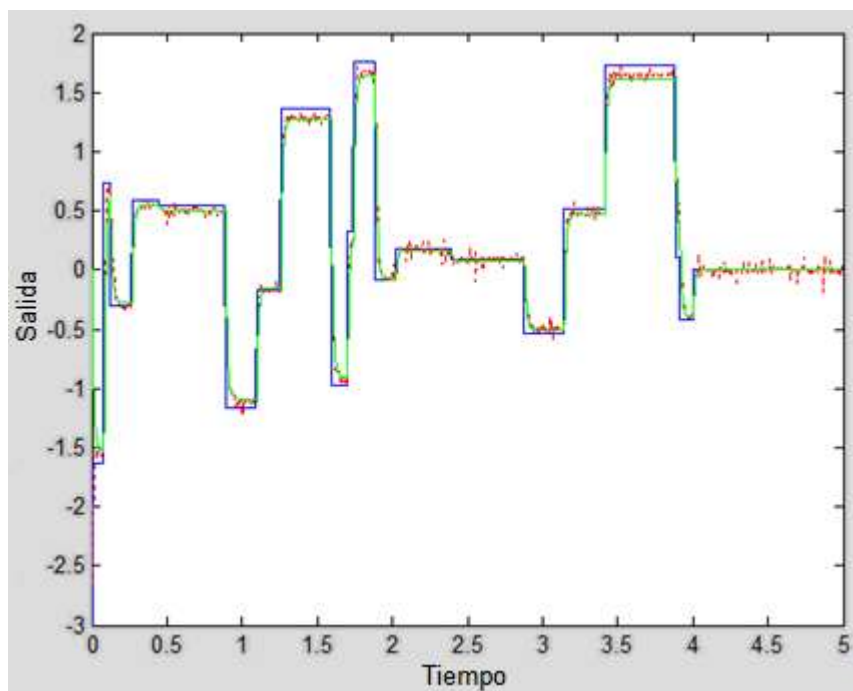


Figura 43: Representación de las salidas con respecto a la entrada



Como vemos queda bien identificado aunque hay un error, como ya se dijo en la identificación de la ganancia.

La diferencia entre la señal de salida del motor y la salida ruidosa procedente de la Función de Transferencia identificada no se puede apreciar en la gráfica de la parte superior, por lo que en la gráfica que se muestra a continuación, se va a representar la diferencia entre ambas señales. Dicha gráfica no va a ser representada desde el inicio de la simulación, ya que queremos evitar el efecto de las condiciones iniciales, por lo que comenzamos a partir de los 0,2 segundos. Se puede ver zonas en que la ganancia está mal identificada.

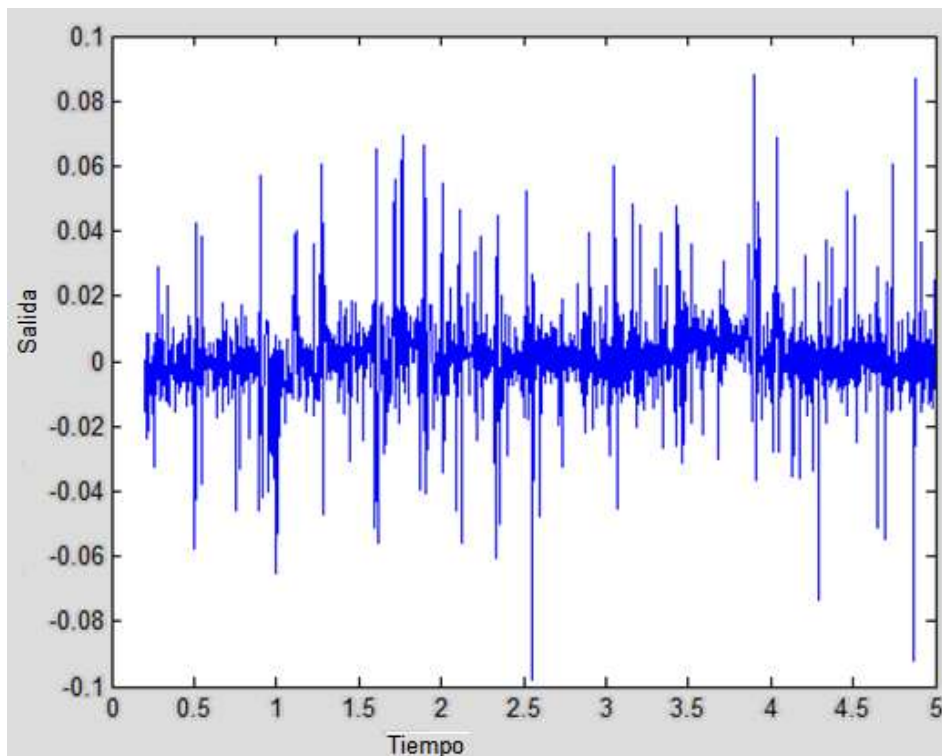


Figura 44: Representación del error

3.2.1.2 $T=0,004$ segundos

La Función de Transferencia correspondiente al mejor caso con este periodo de muestreo es:

$$\frac{40.66 s^3 + 3.781e004 s^2 + 5.383e007 s + 2.343e010}{s^4 + 1202 s^3 + 1.13e006 s^2 + 3.547e008 s + 2.469e010}$$



Lo que da una ganancia de 0.949, compatible con lo obtenido con Output Error.

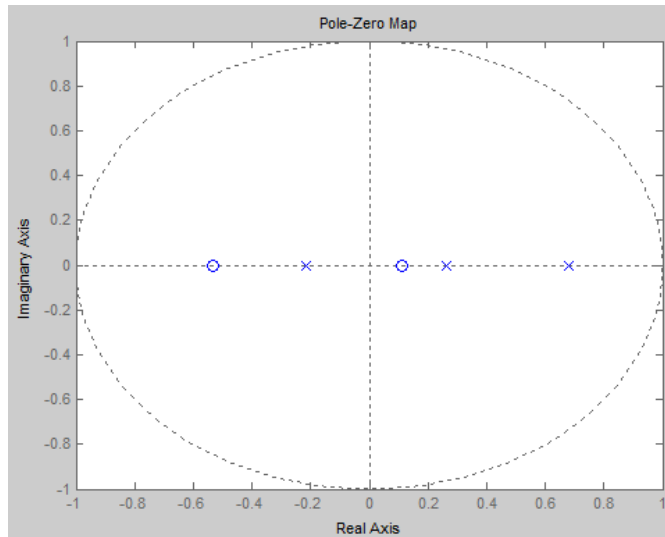
Sin embargo, vemos que ha aumentado el número de polos y ceros con respecto a lo pedido. Esto es consecuencia de que la Función de Transferencia Digital tiene un polo negativo.

$$\frac{0.199 z^2 + 0.08464 z - 0.01173}{z^3 - 0.7285 z^2 - 0.02316 z + 0.03816}$$

Si identificamos los polos y los ceros de la Función de Transferencia Digital los resultados obtenidos son los siguientes:

```
p =
    0.6800
    0.2624
   -0.2139

z =
   -0.5355
    0.1101
```



Se ha comprobado que la Función de Transferencia Digital tiene un polo negativo. Por lo tanto este experimento no es válido ya que se considera que está mal muestreado para este modelo (ARX) ya que lo ve como pulsante.



3.2.1.3 T=0,005 segundos

La Función de Transferencia correspondiente al mejor caso con este periodo de muestreo es:

$$\frac{1.704 s^3 + 7.856e004 s^2 + 3.04e007 s + 1.9e010}{s^4 + 1096 s^3 + 8.298e005 s^2 + 2.5e008 s + 2.003e010}$$

Lo que da una ganancia de 0.9486, compatible con lo obtenido con Output Error

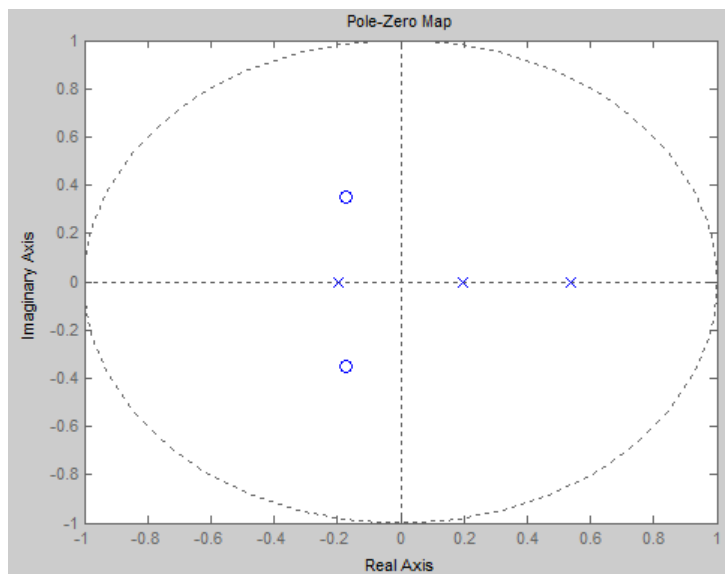
Sin embargo, vemos que ha aumentado el número de polos y ceros con respecto a lo pedido. Esto es consecuencia de que la Función de Transferencia Digital tiene un polo negativo.

$$\frac{0.2803 z^2 + 0.0968 z + 0.04304}{z^3 - 0.5381 z^2 - 0.03993 z + 0.02106}$$

Si identificamos los polos y los ceros de la Función de Transferencia Digital los resultados obtenidos son los siguientes:

```
p =
    0.5398
   -0.1984
    0.1967

z =
   -0.1727 + 0.3517i
   -0.1727 - 0.3517i
```





Por lo tanto este experimento no es válido ya que se considera que está mal muestreado para este modelo (ARX) ya que lo ve como pulsante.

3.2.1.4 $T=0,006$ segundos

La Función de Transferencia identificada para el mejor caso con este periodo de muestreo es:

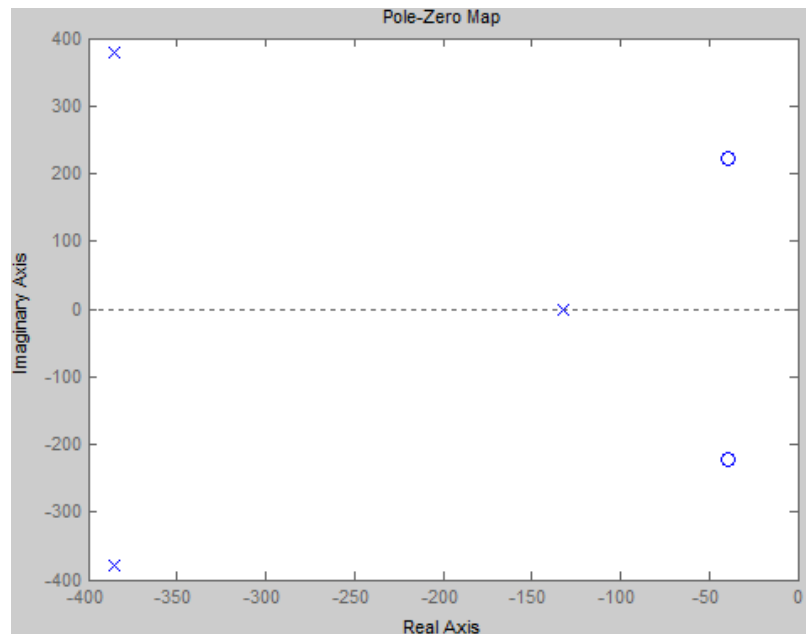
$$\frac{713 s^2 + 5.655e004 s + 3.631e007}{s^3 + 902.5 s^2 + 3.935e005 s + 3.861e007}$$

Lo que da una ganancia de 0.9478, compatible con lo obtenido con Output Error

Tras haber obtenido la Función de Transferencia se analizan los polos y los ceros utilizando el comando “*pzmap*”:

```
p =
1.0e+002 *
-3.8505 + 3.7855i
-3.8505 - 3.7855i
-1.3242

z =
1.0e+002 *
-0.3965 + 2.2214i
-0.3965 - 2.2214i
```



Es decir dos polos prácticamente dominados, un polo dominante en -132.42 y un par de ceros complejos conjugados más cerca del eje y con una gran parte imaginaria.



La última comprobación a realizar es comprobar las salidas, tanto la del motor como la procedente del Matlab, con respecto a la entrada.

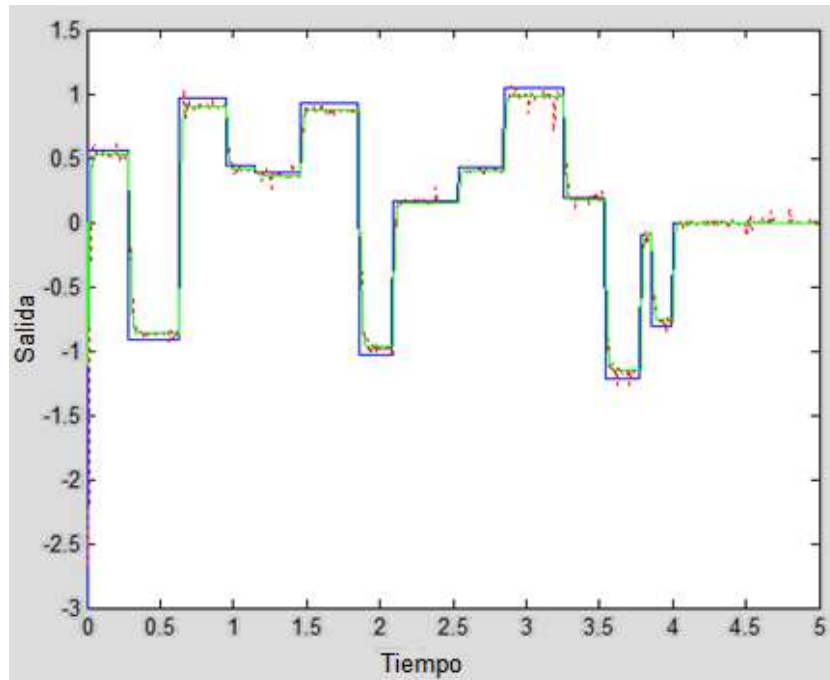


Figura 45: Representación de las salidas respecto de la entrada

La diferencia entre la señal de salida del motor y la salida ruidosa procedente de la Función de Transferencia identificada no se puede apreciar en la gráfica de la parte superior, por lo que en la gráfica que se muestra a continuación, se va a representar la diferencia entre ambas señales. Dicha gráfica no va a ser representada desde el inicio de la simulación, ya que queremos evitar el efecto de las condiciones iniciales, por lo que comenzamos a partir de los 0,2 segundos.

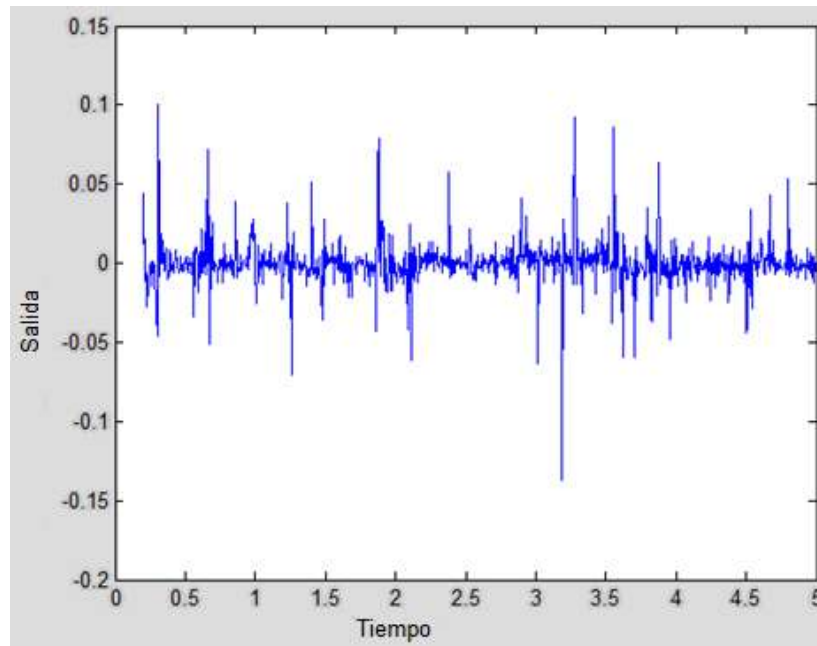


Figura 46: Representación del error

Como se ve, de unos resultados adecuados aunque es difícil entender la naturaleza física de los ceros.

3.1.2.5 T=0,007 segundos

La Función de Transferencia identificada para el mejor caso con este periodo de muestreo es:

$$\frac{-26.76 s^3 + 9.852e004 s^2 + 1.256e007 s + 7.44e009}{s^4 + 786.6 s^3 + 4.365e005 s^2 + 9.57e007 s + 7.959e009}$$

Lo que da una ganancia de 0.9348, compatible con lo obtenido con el Output Error.

Sin embargo, vemos que ha aumentado el número de polos y ceros con respecto a lo pedido. Esto es consecuencia de que la Función de Transferencia Digital tiene un polo negativo.

$$\frac{0.4179 z^2 + 0.05947 z + 0.1283}{z^3 - 0.3739 z^2 + 0.0006195 z + 0.0211}$$

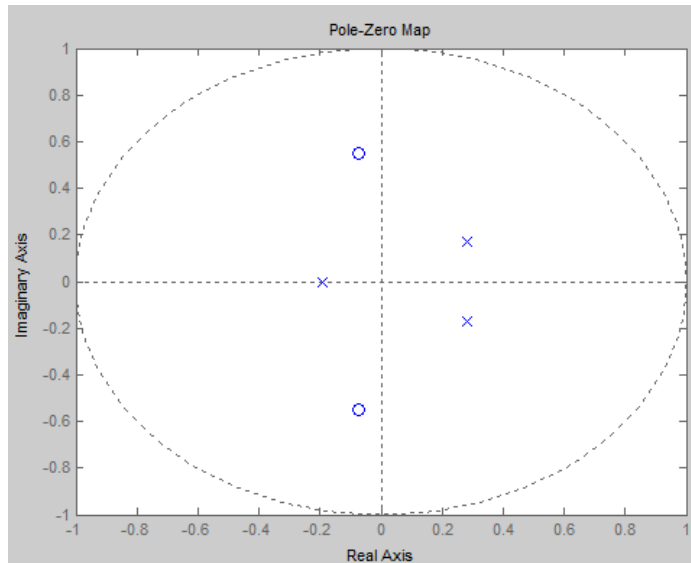
Si identificamos los polos y los ceros de la Función de Transferencia Digital los resultados obtenidos son los siguientes:



```

p =
    0.2832 + 0.1716i
    0.2832 - 0.1716i
   -0.1925

z =
   -0.0712 + 0.5494i
   -0.0712 - 0.5494i
    
```



Por lo tanto este experimento no es válido ya que se considera que está mal muestreado para este modelo (ARX) ya que lo ve como pulsante.

3.2.2 DISMINUCIÓN DE ORDEN: segundo orden

Lo obtenido hasta ahora no parece indicar que este modelo dinámico sea el adecuado para este sistema. Sin embargo, como hicimos en el caso anterior, vamos a disminuir el orden del sistema, ya que, para el Output Error, esto provocó resultados más coherentes.

Vamos a realizar una última prueba, suponiendo que el número de **polos del sistema es dos y el número de ceros es uno**. De cada periodo de muestreo se va a realizar una sola identificación y se van a realizar dos validaciones. Los resultados obtenidos en cuanto a ruido y a correlación son los siguientes:

Periodo de muestreo (seg)	0,003	0,004	0,005	0,006	0,007
error 1	0,2065	0,166	0,1381	0,0899	0,223
error medio 1	1,24E-04	1,33E-04	1,38E-04	1,08E-04	3,12E-04
autocorrelación 1	1,2961	0,8581	0,7107	0,7814	0,6028
autocorrelación media 1	7,78E-04	6,86E-04	7,10E-04	9,37E-04	8,43E-04

Tabla 13: Representación de errores y autocorrelaciones

Las correlaciones que se han obtenido tienen un valor cercano a cero, lo que quiere decir que el ruido que puede tener la maqueta puede ser del modelo ARX. Los ruidos tampoco son muy elevados por lo que podríamos aceptar este método



como bueno, pero antes se van a analizar el número de polos y ceros y su localización.

De los datos que hemos obtenido vamos a realizar la identificación de los polos y ceros y la comparación de las salidas procedentes del motor y la procedente del sistema simulado.

3.2.2.1 T=0,003 segundos

La Función de Transferencia correspondiente al mejor caso con este periodo de muestreo es:

$$\frac{25.15 s^2 - 5.895e004 s + 1.238e008}{s^3 + 1307 s^2 + 1.577e006 s + 1.329e008}$$

Lo que da una ganancia de 0.9315, compatible con lo obtenido con el Output Error.

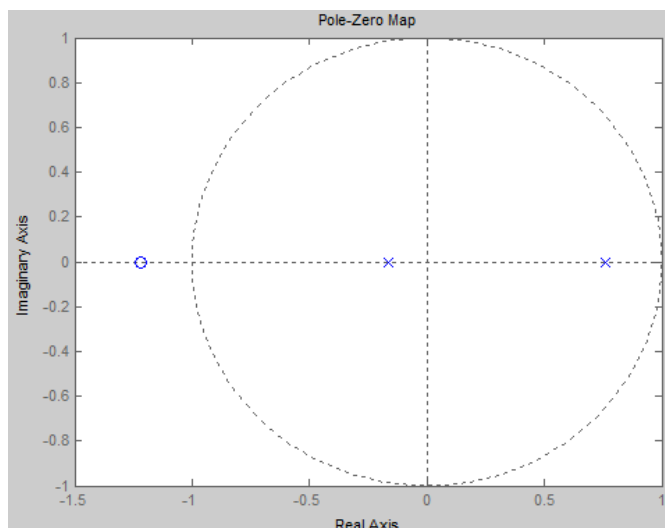
Sin embargo, vemos que ha aumentado el número de polos y ceros con respecto a lo pedido. Esto es consecuencia de que la Función de Transferencia Digital tiene un polo negativo.

$$\frac{0.116 z + 0.1415}{z^2 - 0.6007 z - 0.1229}$$

Si identificamos los polos y los ceros de la Función de Transferencia Digital los resultados obtenidos son los siguientes:

$$p = \begin{matrix} 0.7620 \\ -0.1613 \end{matrix}$$

$$z = \begin{matrix} -1.2197 \end{matrix}$$





Por lo tanto este experimento no es válido ya que se considera que está mal muestreado para este modelo (ARX) ya que lo ve como pulsante.

3.2.2.2T=0,004 segundos

La Función de Transferencia identificada para el mejor caso con este periodo de muestreo es:

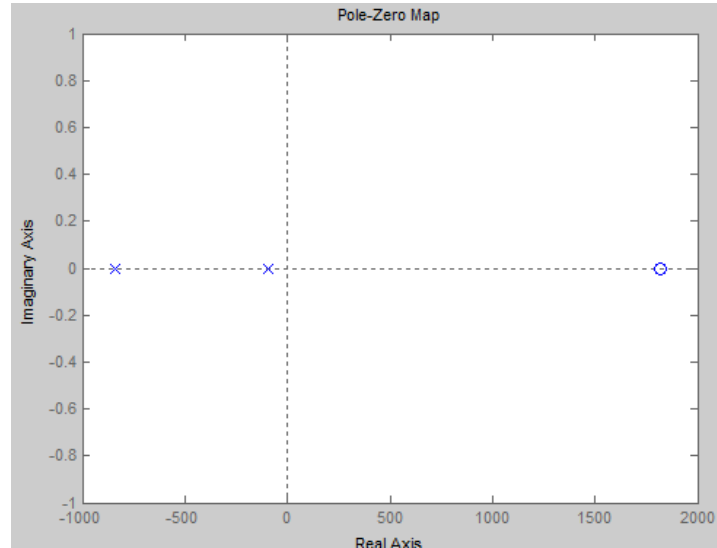
$$\frac{-40.74 s + 7.415e004}{s^2 + 933.4 s + 7.806e004}$$

Lo que da una ganancia de 0.9478, compatible con lo obtenido con Output Error

Tras haber obtenido la Función de Transferencia se analizan los polos y los ceros utilizando el comando “pzmap”:

```
p =
-840.5799
-92.8665

z =
1.8198e+003
```



Es decir, un polo dominado y un cero de fase no mínima muy poco influyente. Por lo que podemos aproximar el sistema por uno de primer orden con polo en -92.8665.

La última comprobación a realizar es comprobar las salidas, tanto la del motor como la procedente del Matlab, con respecto a la entrada.

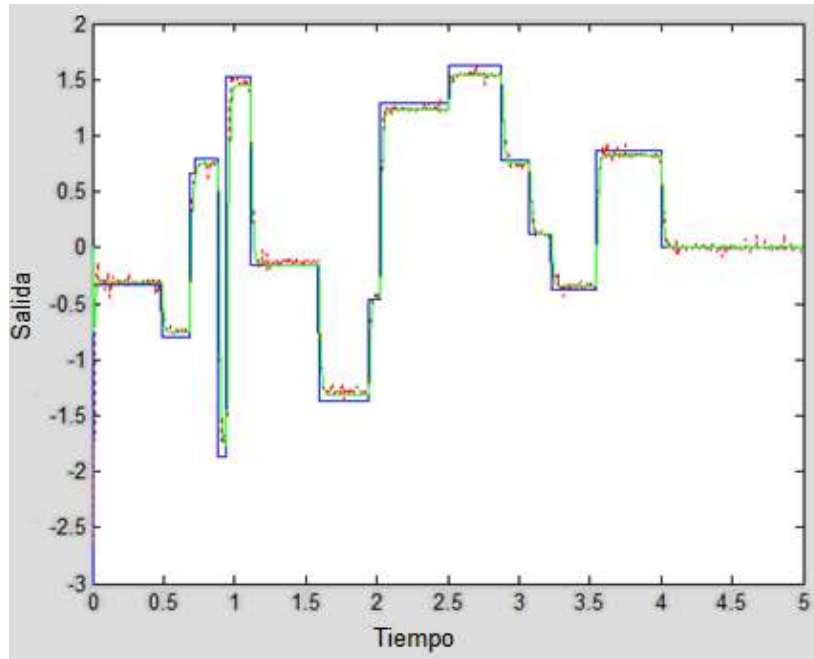


Figura 47: Representación de las salidas respecto de la entrada

La diferencia entre la señal de salida del motor y la salida ruidosa procedente de la Función de Transferencia identificada no se puede apreciar en la gráfica de la parte superior, aunque es posible ver errores en algunos transitorios, que puede venir de que se está intentando reproducir el resultado con un sistema que prácticamente es de primer orden. En la gráfica que se muestra a continuación, se va a representar la diferencia entre ambas señales. Dicha gráfica no va a ser representada desde el inicio de la simulación, ya que queremos evitar el efecto de las condiciones iniciales, por lo que comenzamos a partir de los 0,2 segundos.

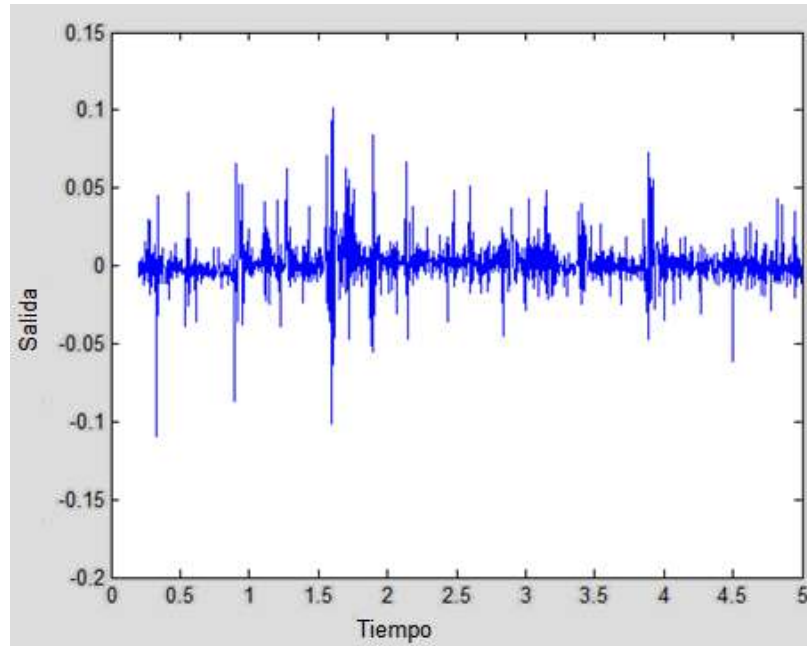


Figura 48: Representación del error

3.2.2.3 $T=0,005$ segundos

La Función de Transferencia identificada para el mejor caso con este periodo de muestreo es:

$$\frac{-17.67 s + 6.427e004}{s^2 + 755.7 s + 6.76e004}$$

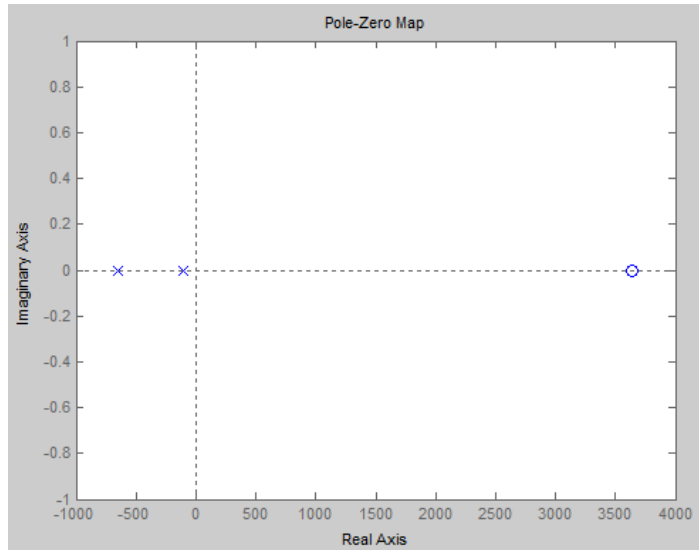
Lo que da una ganancia de 0.9507, compatible con lo obtenido con Output Error, aunque algo baja.

Tras haber obtenido la Función de Transferencia se analizan los polos y los ceros utilizando el comando “*pzmap*”:



```
p =
-652.0693
-103.6664

z =
3.6371e+003
```



La última comprobación a realizar es comprobar las salidas, tanto la del motor como la procedente del Matlab, con respecto a la entrada.

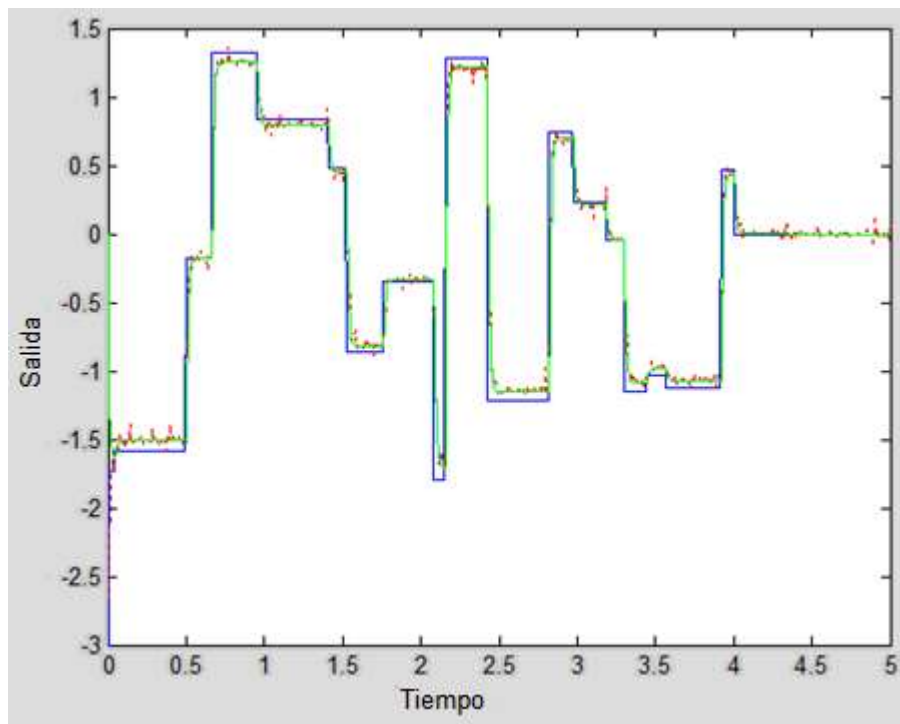


Figura 49: Representación de las salidas respecto de la entrada

La diferencia entre la señal de salida del motor y la salida ruidosa procedente de la Función de Transferencia identificada no se puede apreciar en la gráfica de la parte superior, aunque es posible ver errores en algunos transitorios, como en el caso anterior, posiblemente por las mismas causas. En la gráfica que se muestra a



continuación, se va a representar la diferencia entre ambas señales. Dicha gráfica no va a ser representada desde el inicio de la simulación, ya que queremos evitar el efecto de las condiciones iniciales, por lo que comenzamos a partir de los 0,2 segundos.

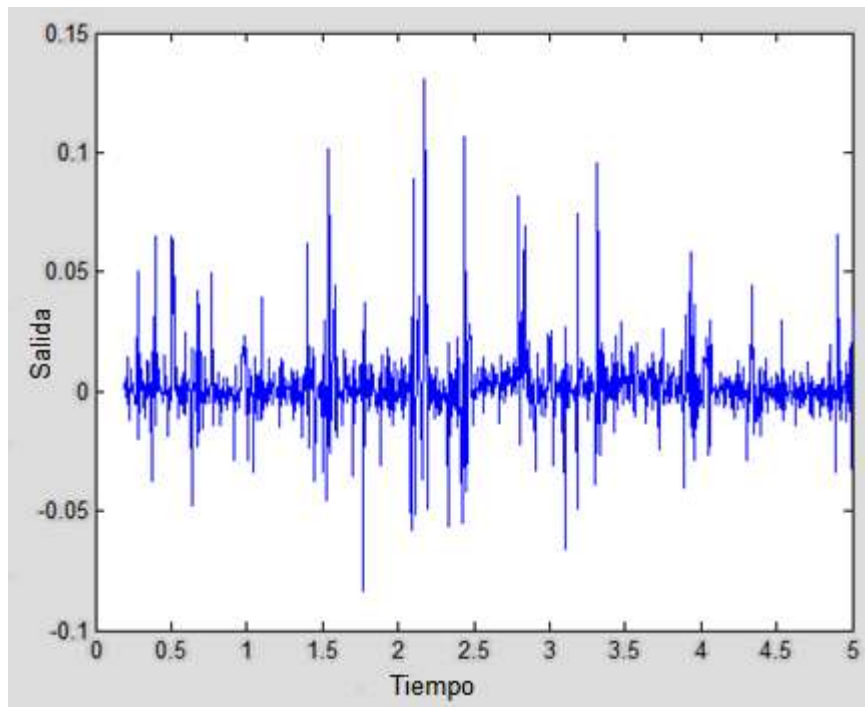


Figura 50: Representación del error

3.2.2.4 $T=0,006$ segundos

La Función de Transferencia identificada para el mejor caso con este periodo de muestreo es:

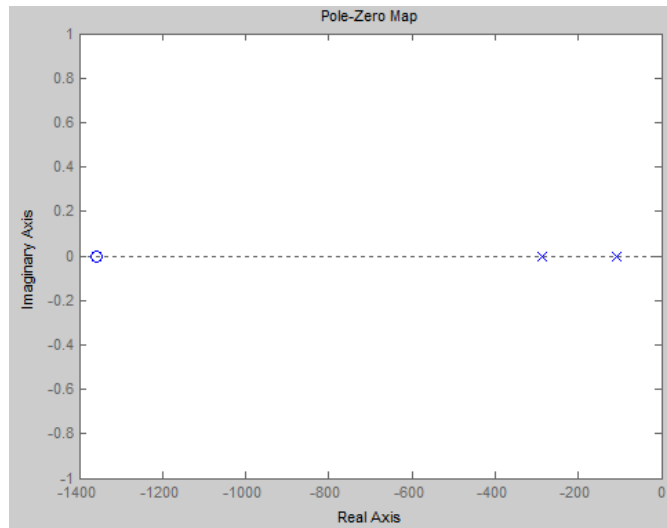
$$\frac{21.81 s + 2.966e004}{s^2 + 395.6 s + 3.128e004}$$

Lo que da una ganancia de 0.9482, compatible con lo obtenido con Output Error, aunque algo baja.

Tras haber obtenido la Función de Transferencia se analizan los polos y los ceros utilizando el comando “*pzmap*”:



```
p =  
-286.3087  
-109.2576  
  
z =  
-1.3597e+003
```



Ahora obtenemos un sistema de segundo orden con polos en -109 y -286 (no completamente dominado). Veremos que el resultado es mejor que los anteriores.

La última comprobación a realizar es comprobar las salidas, tanto la del motor como la procedente del Matlab, con respecto a la entrada.

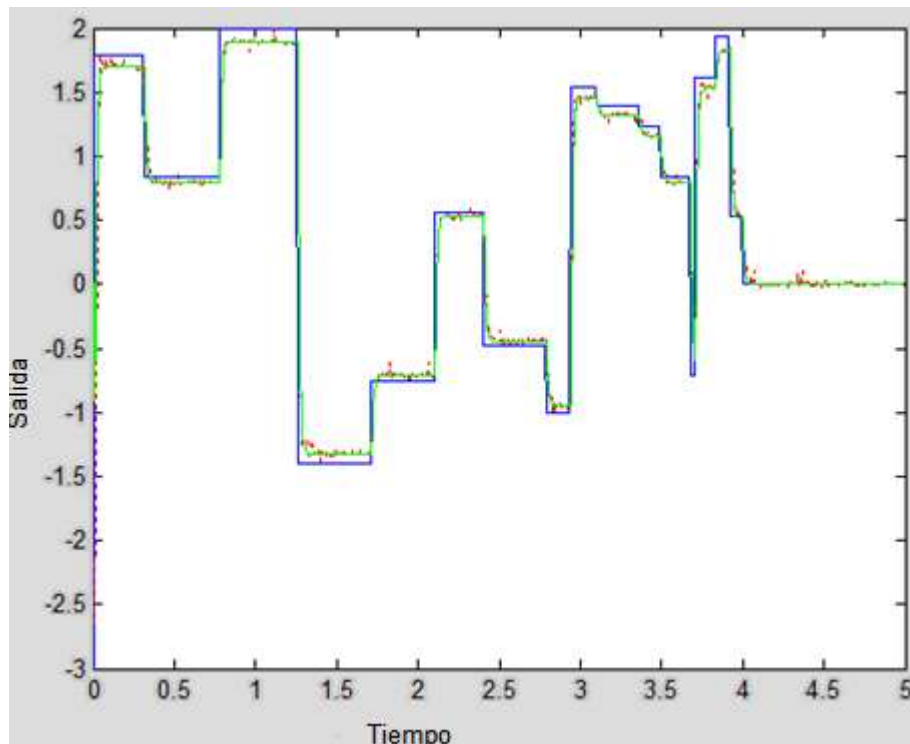


Figura 51: Representación de las salidas respecto de la entrada



La diferencia entre la señal de salida del motor y la salida ruidosa procedente de la Función de Transferencia identificada no se puede apreciar en la gráfica de la parte superior. En la gráfica que se muestra a continuación, se va a representar la diferencia entre ambas señales. Dicha gráfica no va a ser representada desde el inicio de la simulación, ya que queremos evitar el efecto de las condiciones iniciales, por lo que comenzamos a partir de los 0,2 segundos.

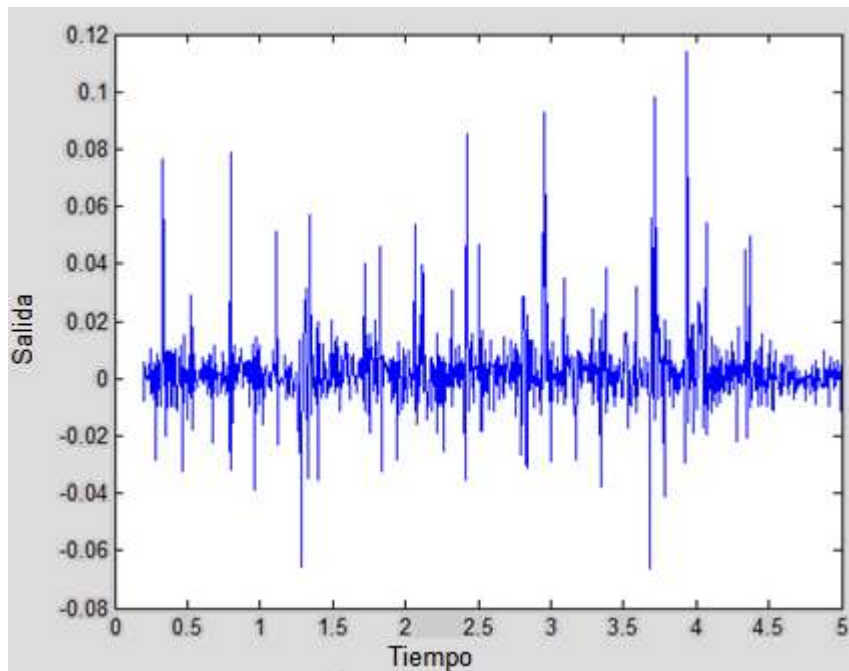


Figura 52: Representación del error

3.2.2.5 $T=0,007$ segundos

La Función de Transferencia identificada para el mejor caso con este periodo de muestreo es:

$$\frac{38.92 s^2 + 3.074e004 s + 4.286e007}{s^3 + 1043 s^2 + 5.208e005 s + 4.507e007}$$

Lo que da una ganancia de 0.9381, compatible con lo obtenido con Output Error, aunque algo baja.

Además, vemos que ha aumentado el número de polos y ceros con respecto a lo pedido. Esto es consecuencia de que la Función de Transferencia Digital tiene un polo negativo.

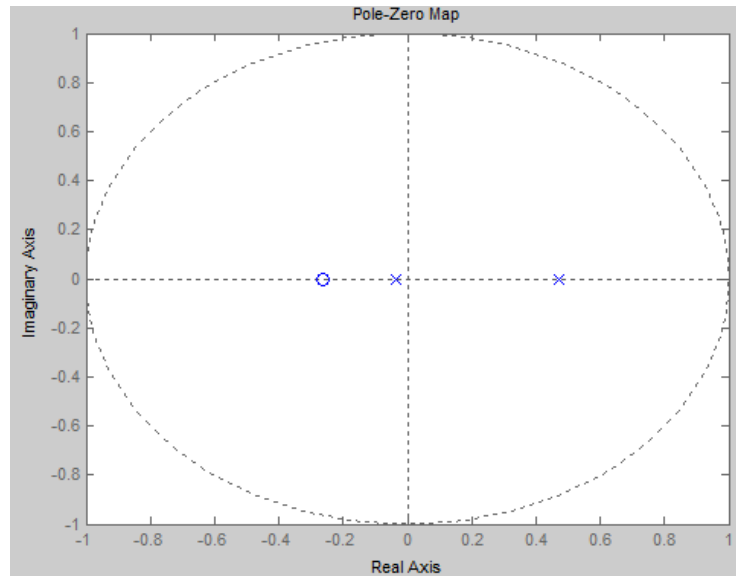


$$\frac{0.4128 z + 0.108}{z^2 - 0.4344 z - 0.01784}$$

Si identificamos los polos y los ceros de la Función de Transferencia Digital los resultados obtenidos son los siguientes:

$$p = \begin{matrix} 0.4722 \\ -0.0378 \end{matrix}$$

$$z = \begin{matrix} -0.2617 \end{matrix}$$



Por lo tanto este experimento no es válido ya que se considera que está mal muestreado para este modelo (ARX) ya que lo ve como pulsante.

• CONCLUSIONES

La reducción de orden no nos ha dado mejores resultados: aunque ahora no fallan tantos experimentos, en general, la dinámica queda peor identificada que en el caso del Output Error. El mejor caso, confirmando las conclusiones del Output Error, es cuando se tiene un sistema de segundo orden.

Podemos concluir que el modelo ARX no es conveniente para la identificación de esta maqueta.



3.3 MODELO ARMAX

El tercer y último modelo dinámico que se va a analizar es el modelo ARMAX.

La ecuación que rige este modelo es la siguiente:

$$y(t) = \frac{B(q)}{A(q)} u(t - nk) + \frac{C(q)}{A(q)} e(t)$$

Es decir, se estima que el ruido tiene los mismos polos que el sistema pero diferentes ceros.

A partir de este modelo es posible hacer un esquema en Simulink que nos permita, conociendo la salida del sistema real y las Funciones de Transferencia identificadas tanto del sistema como del ruido, obtener la salida del sistema identificado y estimar el ruido.

El esquema que hace referencia a este modelo es el siguiente:

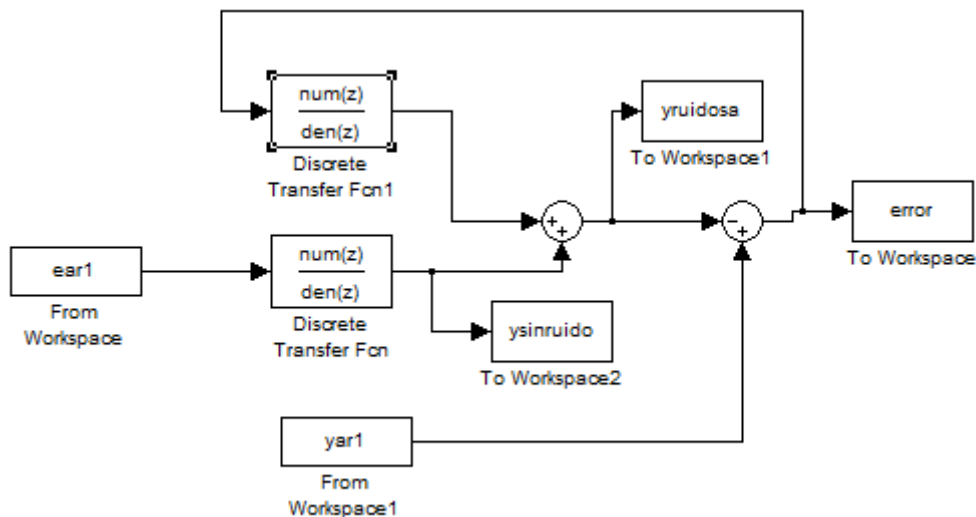


Figura 54: Esquema del modelo ARMAX

Donde “ear1” es la entrada que se ha dado al sistema real, “yar1” la salida del sistema real ante dicha entrada, “Discrete Transfer Fcn” la Función de Transferencia Discreta identificada del sistema, “Discrete Transfer Fcn1” la Función



de Transferencia Discreta identificada del error, “error” el error cometido en la identificación, que, en el caso de identificación perfecta coincide con el ruido del sistema, del primer sumador tenemos la salida ruidosa estimada del sistema.

Para proceder a la identificación y posterior verificación se crean entradas del sistema de la misma forma que las entradas creadas para los dos modelos anteriores.

Se van a utilizar periodos de muestreo que comienzan en las 3 milésimas de segundo y llegan a las 7 milésimas, pasando de milésima en milésima.

Con cada periodo de muestreo se hacen tres identificaciones de funciones de transferencia, con respecto a las cuales se van a realizar las comparaciones. Las identificaciones se van a efectuar de la misma manera que se han realizado en los dos casos anteriores el comando de Matlab que tenemos que utilizar ahora es:

$$Z = \text{iddata}(\text{salida}', \text{entrada}', T)$$

$$M = \text{armax}(Z, [(ceros + 1)_{FT} \text{ polos } ceros_{FTe} \text{ retardo}])$$

Donde *salida* y *entrada* son los datos obtenidos de la maqueta, *T* es periodo de muestreo utilizado, *cero_{FT}* el números de ceros que pedimos en la Función de Transferencia Discreta, *polos* el números de polos que pedimos en la Función de Transferencia Discreta, *cero_{FTe}* el números de ceros que pedimos en la Función de Transferencia del error y *retardo* el retraso presentado por dicha Función (este último se estima 1 en todos los experimentos).

De la variable *M* se pueden obtener los vectores que contienen los coeficientes de la Función de Transferencia.

$$[A, B, C, D, F] = \text{polydata}(M)$$

En simulink, *Figura 54*, se deben definir de nuevo las dos funciones de transferencia, la Función de Transferencia de la propia maqueta y la del error. Para realizar la Función de Transferencia, primero hay que calcular los vectores compuestos por los coeficientes que van a utilizarse. Los vectores hay que situarlos correctamente para obtener la Función de Transferencia adecuada. Para definir la Función de Transferencia de la maqueta hay que utilizar los vectores B y A, como numerador y denominador, respectivamente. En cambio, cuando queremos calcular



la Función de Transferencia del error se usan los vectores C en el numerador y A en el denominador.

El análisis que se va a realizar es del mismo tipo que el realizado con el ARX, ya que lo único que cambia es la Función de Transferencia del error.

3.3.1 Sistema de tercer orden

Como en casos anteriores, se va a comenzar el análisis suponiendo que el número de **polos y ceros es de tres y dos respectivamente**. Para cada uno de los periodos de muestreo se han identificado tres funciones de transferencia y utilizando cada una de ellas se han realizado tres validaciones, obteniendo sus correspondientes errores y autocorrelaciones del ruido estimado. En la siguiente tabla se indican los mejores resultados obtenidos (marcamos en negrita el mejor de cada caso), esto es, indicamos las identificaciones que nos han dado un menor ruido y menor correlación.

Periodo de muestreo (seg)	0,003	0,004	0,005	0,006	0,007
error 1	1,094	0,868	0,7599	0,9766	0,5868
error medio 1	6,56E-04	6,94E-04	7,49E-04	1,17E-03	8,20E-04
autocorrelación 1	1,06708	0,976	1,0143	1,3586	0,8565
autocorrelación media 1	1,00E-03	7,80E-04	1,01E-03	1,63E-03	1,20E-03
error 2	1,18	1,0278	0,8556	0,9958	0,6638
error medio 2	7,08E-04	8,21E-04	8,55E-04	1,19E-03	9,30E-04
autocorrelación 2	1,7675	1,4708	1,1235	1,4909	0,9895
autocorrelación media 2	1,06E-03	1,17E-03	1,12E-03	1,79E-03	1,38E-03

Tabla 14: Representación de errores y correlaciones

Las autocorrelaciones que se han obtenido tienen un valor cercano a cero, lo que quiere decir que el ruido podría estar bien identificado. En este modelo para conseguir la decorrelación tenemos que usar una Función de Transferencia. Los errores tampoco son muy elevados por lo que podríamos aceptar este método como bueno, pero antes se van a analizar el número de polos y ceros y su localización.

Los errores que se obtienen son mayores cuanto más se aleja el tiempo de muestreo del valor que ya habíamos estimado como bueno en simulación: 4 milésimas de segundo.



De cada uno de los periodos de muestreo vamos a elegir el mejor caso, el que está marcado en negrita, en el que el ruido y la correlación son menores y vamos a analizar tanto el número de polos y ceros y su localización, como la comparación de las salidas.

3.3.1.1 T=0,003 segundos

La Función de Transferencia correspondiente al mejor caso con este periodo de muestreo es:

$$\frac{32.66 s^3 - 5.542e004 s^2 + 8.924e007 s + 1.746e010}{s^4 + 1138 s^3 + 1.522e006 s^2 + 2.923e008 s + 1.843e010}$$

Lo que da una ganancia de 0.9473, compatible con lo obtenido con Output Error.

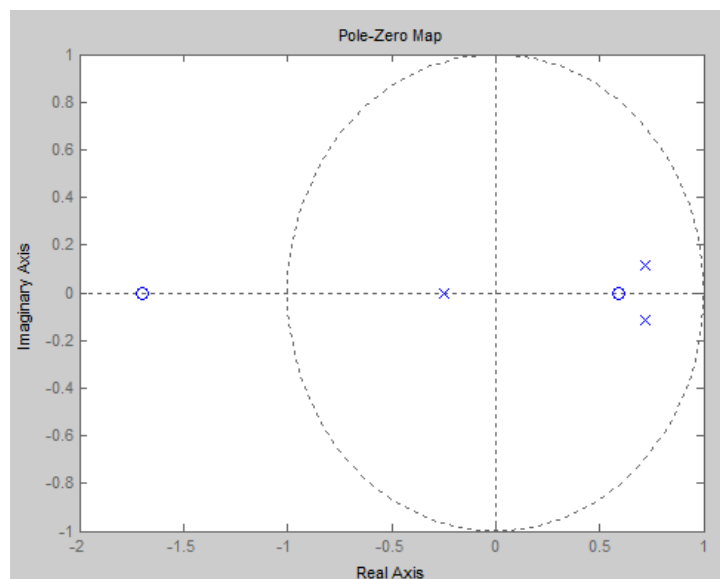
Sin embargo, vemos que ha aumentado el número de polos y ceros con respecto a lo pedido. Esto es consecuencia de que la Función de Transferencia Digital tiene un polo negativo.

$$\frac{0.09946 z^2 + 0.11 z - 0.09997}{z^3 - 1.186 z^2 + 0.1694 z + 0.1317}$$

Si identificamos los polos y los ceros de la Función de Transferencia Digital los resultados obtenidos son los siguientes:

```
p =
0.7176 + 0.1130i
0.7176 - 0.1130i
-0.2497

z =
-1.6983
0.5919
```





Se ha comprobado que la Función de Transferencia Digital tiene un polo negativo. Por lo tanto este experimento no es válido ya que se considera que está mal muestreado para este modelo (ARMAX) ya que lo ve como pulsante.

3.3.1.2 T=0,004 segundos

La Función de Transferencia correspondiente al mejor caso con este periodo de muestreo es:

$$\frac{65.89 s^3 - 267 s^2 + 4.267e007 s + 7.639e009}{s^4 + 453.1 s^3 + 6.962e005 s^2 + 1.294e008 s + 8.011e009}$$

Lo que da una ganancia de 0.9536, compatible con lo obtenido con Output Error.

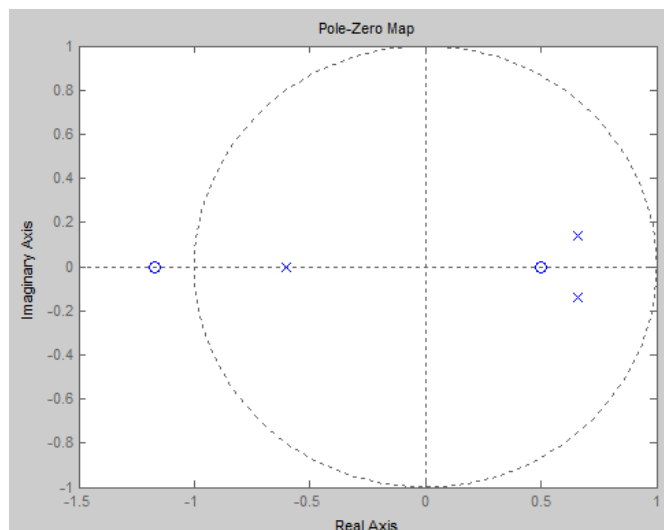
Sin embargo, vemos que ha aumentado el número de polos y ceros con respecto a lo pedido. Esto es consecuencia de que la Función de Transferencia Digital tiene un polo negativo.

$$\frac{0.1935 z^2 + 0.1298 z - 0.1137}{z^3 - 0.7112 z^2 - 0.3402 z + 0.2712}$$

Si identificamos los polos y los ceros de la Función de Transferencia Digital los resultados obtenidos son los siguientes:

```
p =
-0.6020
 0.6566 + 0.1390i
 0.6566 - 0.1390i

z =
-1.1720
 0.5012
```





Se ha comprobado que la Función de Transferencia Digital tiene un polo negativo. Por lo tanto este experimento no es válido ya que se considera que está mal muestreado para este modelo (ARMAX) ya que lo ve como pulsante.

3.3.1.3 T=0,005 segundos

La Función de Transferencia correspondiente al mejor caso con este periodo de muestreo es:

$$\frac{53.58 s^3 + 2.084e004 s^2 + 2.369e007 s + 5.952e009}{s^4 + 389.3 s^3 + 4.553e005 s^2 + 8.971e007 s + 6.274e009}$$

Lo que da una ganancia de 0.9487, compatible con lo obtenido con Output Error.

Sin embargo, vemos que ha aumentado el número de polos y ceros con respecto a lo pedido. Esto es consecuencia de que la Función de Transferencia Digital tiene un polo negativo.

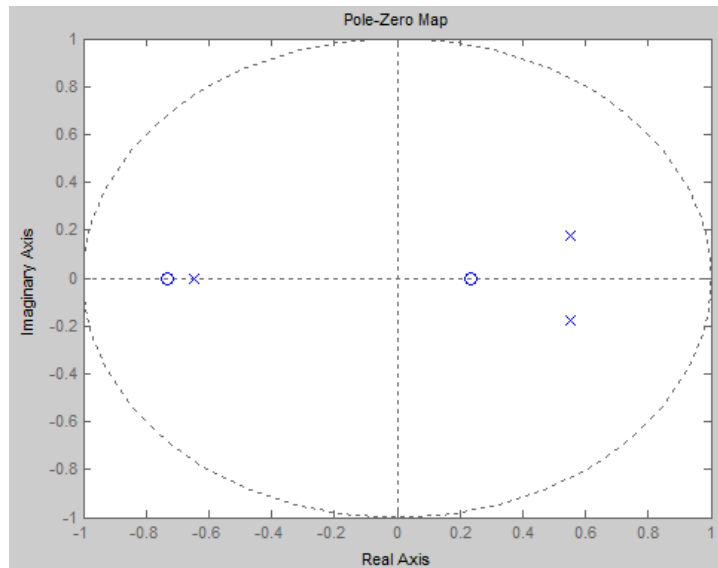
$$\frac{0.2721 z^2 + 0.1356 z - 0.04671}{z^3 - 0.4594 z^2 - 0.38 z + 0.22}$$

Si identificamos los polos y los ceros de la Función de Transferencia Digital los resultados obtenidos son los siguientes:



```
p =
-0.6488
 0.5541 + 0.1788i
 0.5541 - 0.1788i
```

```
z =
-0.7326
 0.2343
```



Se ha comprobado que la Función de Transferencia Digital tiene un polo negativo. Por lo tanto este experimento no es válido ya que se considera que está mal muestreado para este modelo (ARMAX) ya que lo ve como pulsante.

3.3.1.4 T=0,006 segundos

La Función de Transferencia correspondiente al mejor caso con este periodo de muestreo es:

$$\frac{80.08 s^2 + 1.313e004 s + 3.376e006}{s^3 + 286.7 s^2 + 5.151e004 s + 3.562e006}$$

Tras haber obtenido la Función de Transferencia se analizan los polos y los ceros utilizando el comando “pzmap”:

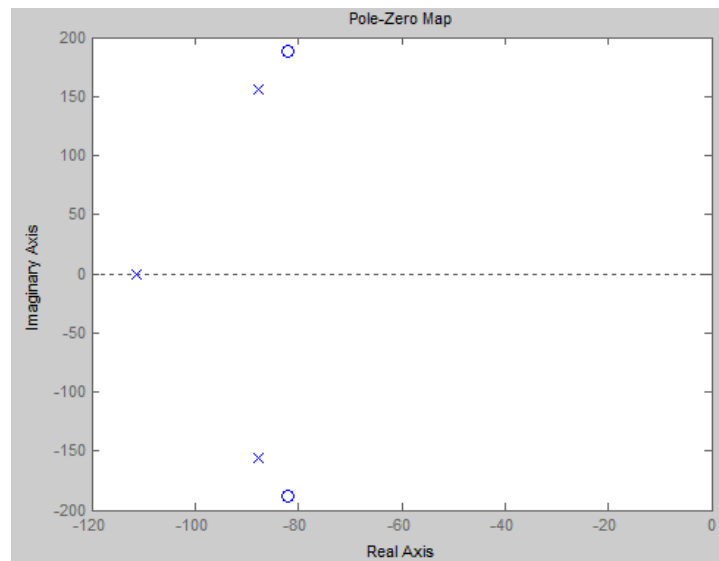


```

p =
1.0e+002 *
-0.8767 + 1.5588i
-0.8767 - 1.5588i
-1.1137

z =
1.0e+002 *
-0.8200 + 1.8823i
-0.8200 - 1.8823i

```



Como vemos estima un sistema de tercer orden, aunque los pares de polos y ceros complejos conjugados prácticamente se están anulando (nótese que si no fuera así la sobreoscilación sería del 17%). Así que podemos concluir que el orden del sistema es menor que el estimado en esta identificación.

Por otra parte, la ganancia del sistema identificado es 0.9478, compatible con los casos ya estudiados.

La última comprobación a realizar es comprobar las salidas, tanto la del motor como la procedente del Matlab, con respecto a la entrada.

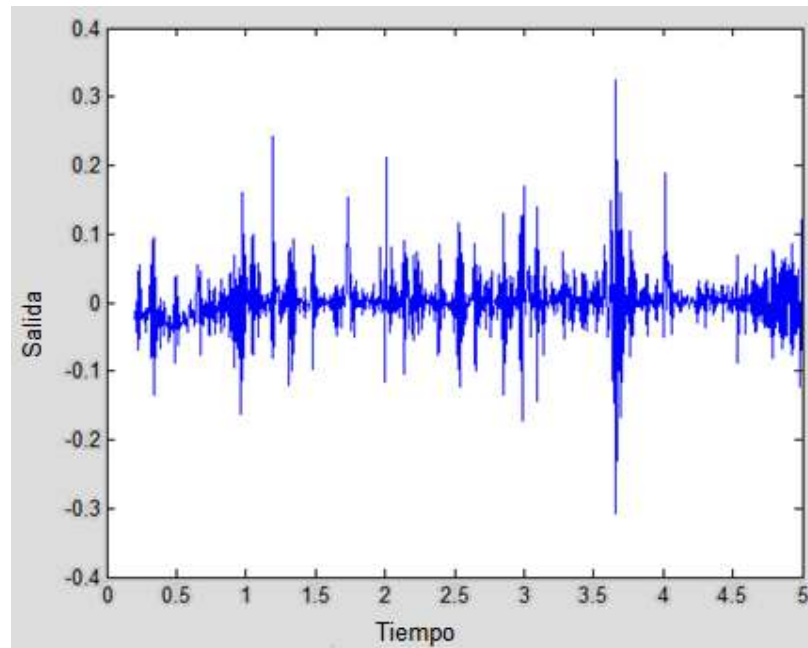


Figura 55: Representación del error

3.3.1.5 T=0,007 segundos

La Función de Transferencia correspondiente al mejor caso con este periodo de muestreo es:

$$\frac{122.5 s^2 + 9621 s + 6.935e006}{s^3 + 334.2 s^2 + 8.474e004 s + 7.337e006}$$

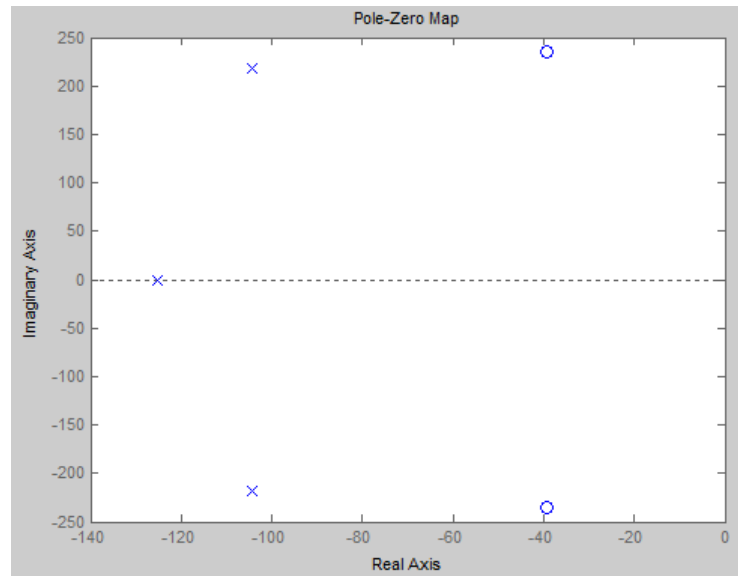
Tras haber obtenido la Función de Transferencia se analizan los polos y los ceros utilizando el comando “pzmap”:



```

p =
    1.0e+002 *
    -1.0446 + 2.1830i
    -1.0446 - 2.1830i
    -1.2527

z =
    1.0e+002 *
    -0.3928 + 2.3470i
    -0.3928 - 2.3470i
  
```



Como vemos estima un sistema de tercer orden, en este caso sin anulaciones claras de polos y ceros. Las distribución de los pares de polos y ceros complejos conjugados sería compatible con una elasticidad pero dan una oscilación muy amortiguada.

Por otra parte, la ganancia del sistema identificado es 0.9452 compatible con los casos ya estudiados.

La última comprobación a realizar es comprobar las salidas, tanto la del motor como la procedente del Matlab, con respecto a la entrada.

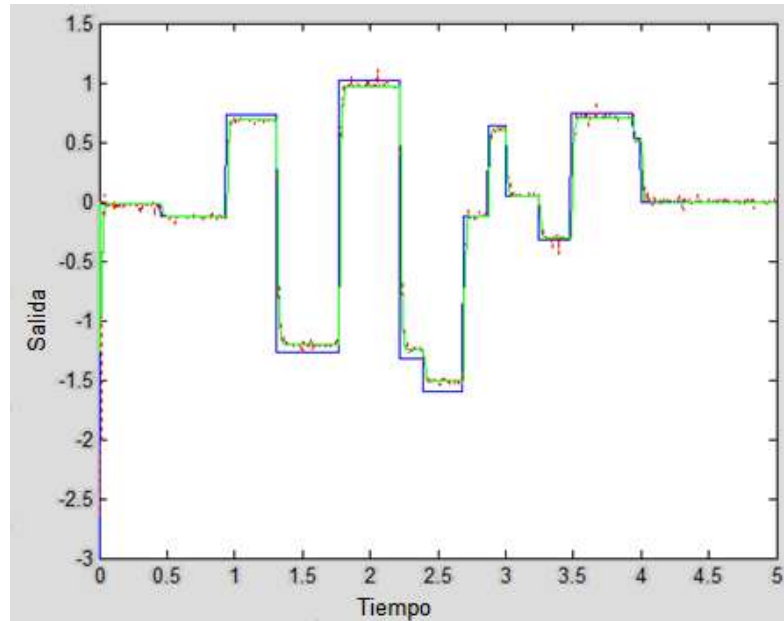


Figura 56: Representación de las salidas respecto de la entrada

La diferencia entre la señal de salida del motor y la salida ruidosa procedente de la Función de Transferencia identificada no se puede apreciar en la gráfica de la parte superior. En la gráfica que se muestra a continuación, se va a representar la diferencia entre ambas señales. Dicha gráfica no va a ser representada desde el inicio de la simulación, ya que queremos evitar el efecto de las condiciones iniciales, por lo que comenzamos a partir de los 0,2 segundos.

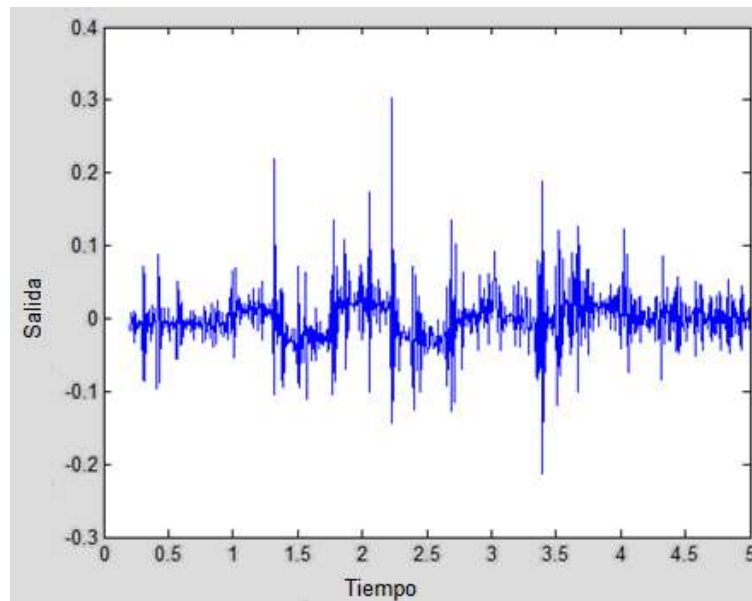


Figura 57: Representación del error



Al igual que en el caso ARX estos experimento han fallado en la mayoría de las ocasiones y en las que no lo ha hecho no dan resultados. Por lo tanto, vamos a comprobar si reduciendo el orden conseguimos, como pasó con el Output Error, mejores resultados.

3.3.2 Disminución de orden: segundo orden

Como hemos visto, sólo dos de los experimentos han funcionado correctamente y de estos uno indica que el orden real del sistema es menor que el propuesta. Vamos a realizar otra serie de simulaciones considerando que el sistema se caracteriza por tener **dos polos y un cero**.

Para la realización de la identificación vamos a utilizar el mismo procedimiento que en el caso anterior, pero tendremos que cambiar la ecuación donde se hacen referencia al número de polos y ceros. Vamos a analizar sólo los valores de periodo de muestreo que dieron menores errores en las anteriores identificaciones. Los dos mejores datos (el mejor en negrita), menor ruido y menor correlación, que se han obtenido de cada periodo de muestreo son los siguientes:

Periodo de muestreo (seg)	0,003	0,004	0,005	0,006	0,007
error 1	1,1052	0,5175	0,7676	0,6989	0,6521
error medio 1	6,63E-04	4,14E-04	7,67E-04	8,38E-04	9,12E-04
autocorrelación 1	0,6144	0,3362	0,9301	0,7881	0,906
autocorrelación media 1	3,69E-04	2,69E-04	9,29E-04	9,45E-04	1,27E-03
error 2	1,1541	0,6206	0,7707	0,9034	0,6725
error medio 2	6,92E-04	4,96E-04	7,70E-04	9,63E-04	9,41E-04
autocorrelación 2	1,6357	0,6006	0,8528	1,0631	0,9999
autocorrelación media 2	9,81E-04	4,80E-04	8,52E-04	1,27E-03	1,40E-03

Tabla 15: Representación de errores y autocorrelaciones

Las autocorrelaciones que se han obtenido tienen un valor cercano a cero, lo que quiere decir que el ruido que puede tener la maqueta puede ser modelado como del tipo estimado por modelo ARMAX. Los errores tampoco son muy elevados por lo que podríamos aceptar este método como bueno, pero antes se van a analizar el número de polos y ceros y su localización.

De cada uno de los periodos de muestreo vamos a elegir el mejor caso, el que ésta marcado en negrita, en el que el ruido y la correlación son menores y vamos a



analizar tanto el número de polos y ceros y su localización, como la comparación de las salidas.

3.3.2.1 T=0,003 segundos

La Función de Transferencia correspondiente al mejor caso con este periodo de muestreo es:

$$\frac{-6.73 s^2 - 1.241e005 s + 1.456e008}{s^3 + 1479 s^2 + 1.708e006 s + 1.529e008}$$

Lo que da una ganancia de 0.9522, compatible con lo obtenido con Output Error.

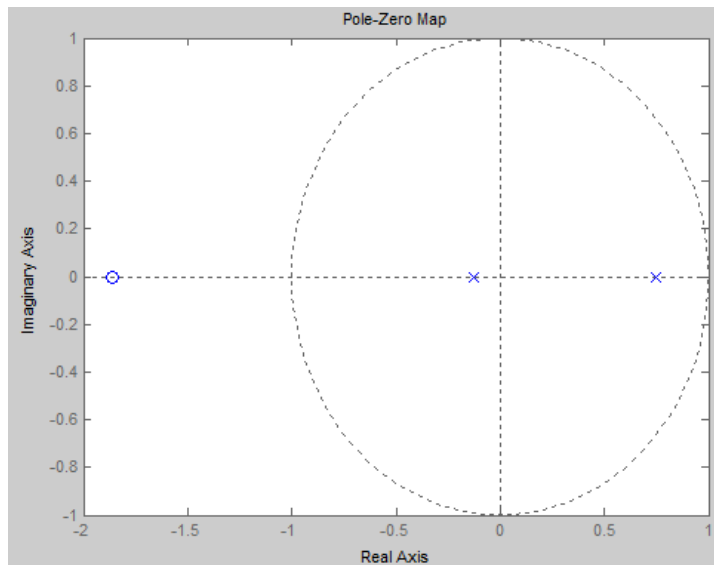
Sin embargo, vemos que ha aumentado el número de polos y ceros con respecto a lo pedido. Esto es consecuencia de que la Función de Transferencia Digital tiene un polo negativo.

$$\frac{0.09476 z + 0.1763}{z^2 - 0.6212 z - 0.09408}$$

Si identificamos los polos y los ceros de la Función de Transferencia Digital los resultados obtenidos son los siguientes:

```
p =
    0.7471
   -0.1259

z =
   -1.8603
```





Se ha comprobado que la Función de Transferencia Digital tiene un polo negativo. Por lo tanto este experimento no es válido ya que se considera que está mal muestreado para este modelo (ARMAX) ya que lo ve como pulsante.

3.3.2.2 T=0,004 segundos

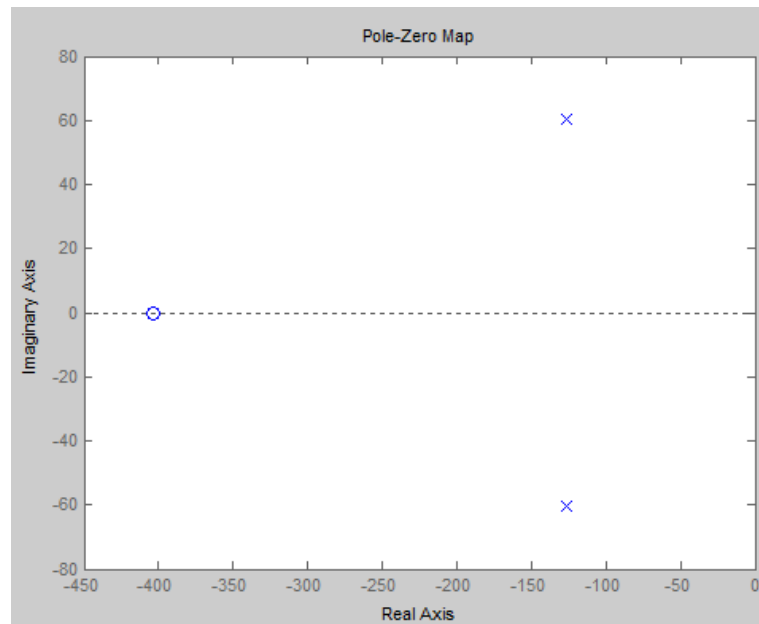
La Función de Transferencia correspondiente al mejor caso con este periodo de muestreo es:

$$\frac{46.46 s + 1.874e004}{s^2 + 253.7 s + 1.976e004}$$

Tras haber obtenido la Función de Transferencia se analizan los polos y los ceros utilizando el comando “pzmap”:

```
p =
    1.0e+002 *
    -1.2686 + 0.6051i
    -1.2686 - 0.6051i

z =
    -403.3961
```



Como vemos estima un sistema de segundo orden con polos en -126.86 y una pequeña parte imaginaria que apenas aporta sobreoscilación (menor del 0.2%). El cero está muy alejado y se puede despreciar. Por otra parte, la ganancia del sistema identificado es 0.9484 compatible con los casos ya estudiados. Como vemos un



resultado comparable a los conseguidos con Output Error pero con los polos un poco más alejados en este caso.

La última comprobación a realizar es comprobar las salidas, tanto la del motor como la procedente del Matlab, con respecto a la entrada.

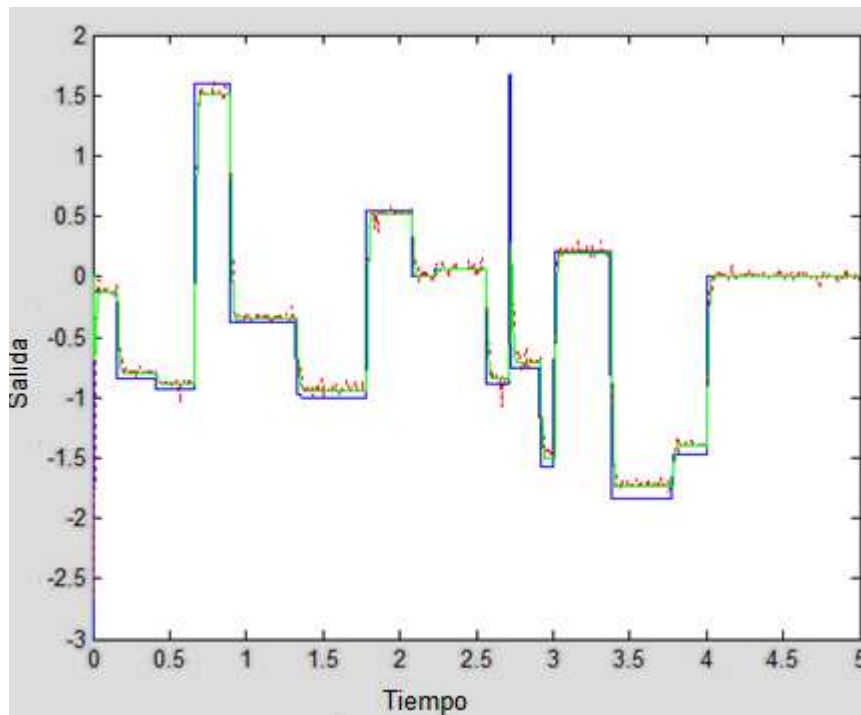


Figura 58: Representación de las salidas respecto de la entrada

La diferencia entre la señal de salida del motor y la salida ruidosa procedente de la Función de Transferencia identificada no se puede apreciar en la gráfica de la parte superior. En la gráfica que se muestra a continuación, se va a representar la diferencia entre ambas señales. Dicha gráfica no va a ser representada desde el inicio de la simulación, ya que queremos evitar el efecto de las condiciones iniciales, por lo que comenzamos a partir de los 0,2 segundos. Esta gráfica parece apreciarse algún error en el permanente.

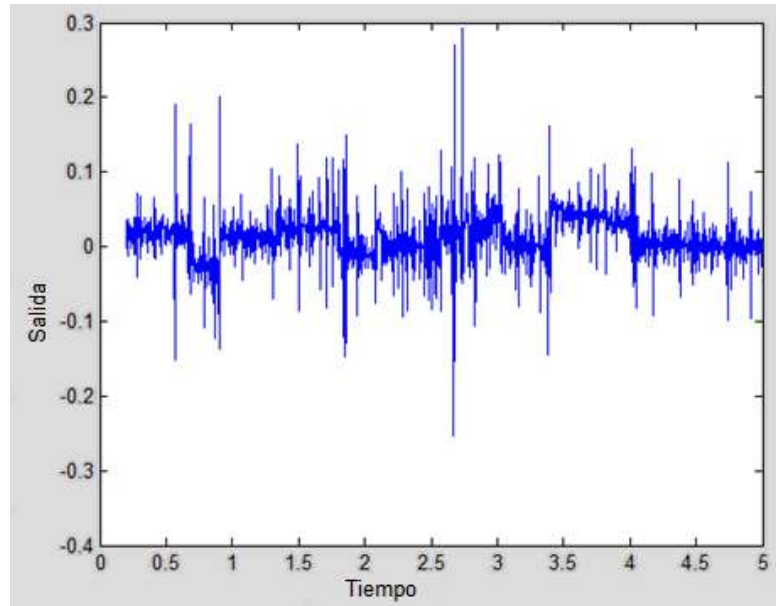


Figura 59: Representación del error

3.3.2.3 T=0,005 segundos

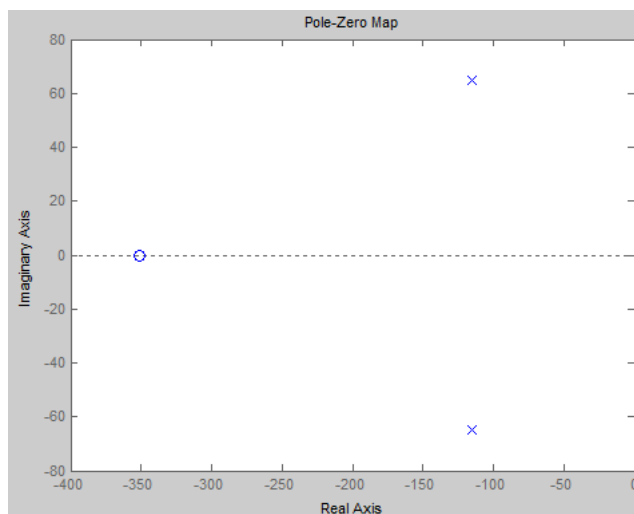
La Función de Transferencia correspondiente al mejor caso con este periodo de muestreo es:

$$\frac{47.07 s + 1.651e004}{s^2 + 229.7 s + 1.741e004}$$

Tras haber obtenido la Función de Transferencia se analizan los polos y los ceros utilizando el comando “pzmap”:

```
p =
1.0e+002 *
-1.1486 + 0.6492i
-1.1486 - 0.6492i

z =
-350.7601
```





En cuanto a los polos y ceros, el sistema es de segundo orden con dos polos en $-114,8$ con una pequeña parte imaginaria que no aporta una sobreoscilación estimable (menor del 0.5%) y un cero en $-350,7$ despreciable. Es un modelo compatible con el anterior y con los obtenidos con Output Error, aunque de nuevo con los polos más alejados.

La ganancia del sistema es 0.9483. Compatible con los casos anteriores

La última comprobación a realizar es comprobar las salidas, tanto la del motor como la procedente del Matlab, con respecto a la entrada.

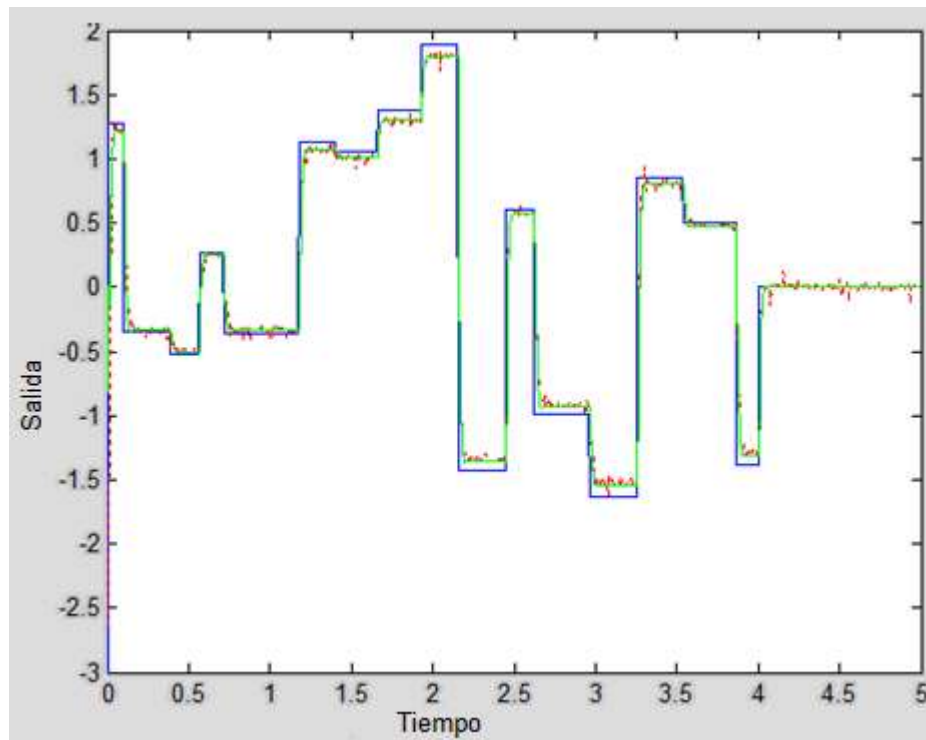


Figura 60: Representación de las salidas respecto de la entrada

La diferencia entre la señal de salida del motor y la salida ruidosa procedente de la Función de Transferencia identificada no se puede apreciar en la gráfica de la parte superior. En la gráfica que se muestra a continuación, se va a representar la diferencia entre ambas señales. Dicha gráfica no va a ser representada desde el inicio de la simulación, ya que queremos evitar el efecto de las condiciones iniciales, por lo que comenzamos a partir de los 0,2 segundos.

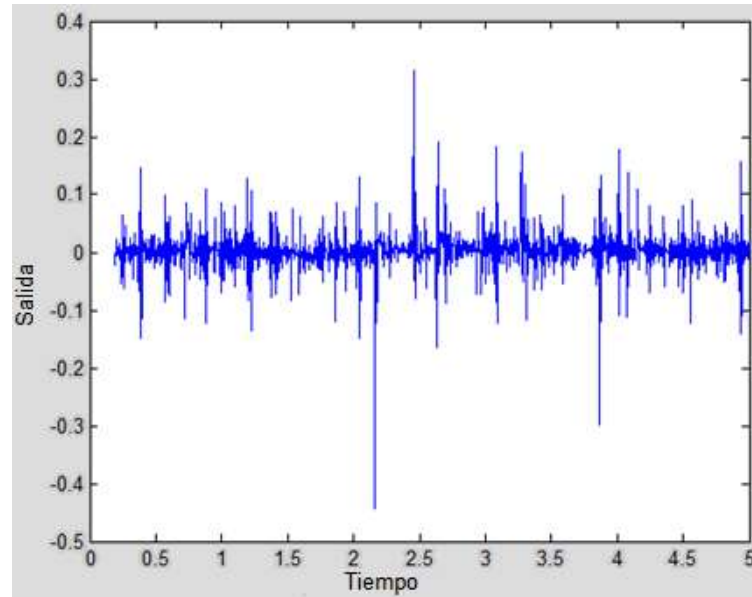


Figura 61: Representación del error

3.3.2.4 T=0,006 segundos

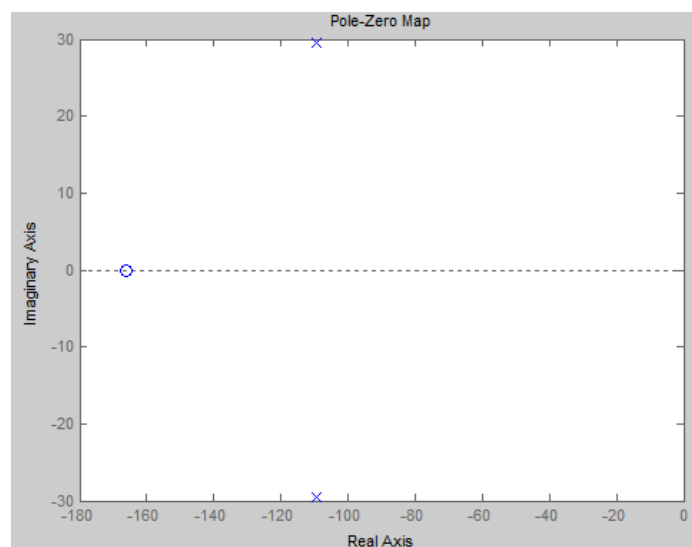
La Función de Transferencia correspondiente al mejor caso con este periodo de muestreo es:

$$\frac{73.31 s + 1.217e004}{s^2 + 218.8 s + 1.285e004}$$

Tras haber obtenido la Función de Transferencia se analizan los polos y los ceros utilizando el comando “pzmap”:

```
p =
1.0e+002 *
-1.0941 + 0.2960i
-1.0941 - 0.2960i

z =
-165.9855
```





En cuanto a los polos y ceros, el sistema es de segundo orden con dos polos en $-109,4$ con una pequeña parte imaginaria que no aporta una sobreoscilación estimable (menor del $0,001\%$) y un cero en -166 . Es un modelo compatible con el anterior pero con un cero más cercano y con los obtenidos con Output Error, aunque de nuevo con los polos más alejados aunque no tanto como en otros casos. Como vemos parece que hay dos opciones: sistema de segundo orden con polos más allá de -110 y cero despreciable u otro con polos más lentos y cero más cerca, similar al obtenido por el Output Error.

La ganancia del sistema es 0.9471 compatible con los casos anteriores.

La última comprobación a realizar es comprobar las salidas, tanto la del motor como la procedente del Matlab, con respecto a la entrada.

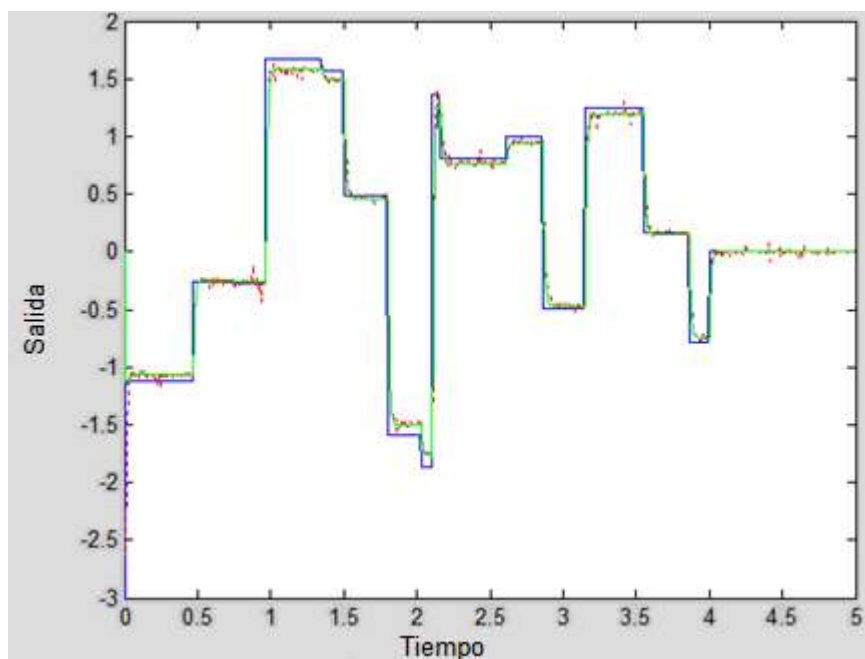


Figura 62: Representación de las salidas respecto de la entrada

La diferencia entre la señal de salida del motor y la salida ruidosa procedente de la Función de Transferencia identificada no se puede apreciar en la gráfica de la parte superior. En la gráfica que se muestra a continuación, se va a representar la diferencia entre ambas señales. Dicha gráfica no va a ser representada desde el inicio de la simulación, ya que queremos evitar el efecto de las condiciones iniciales, por lo que comenzamos a partir de los $0,2$ segundos.

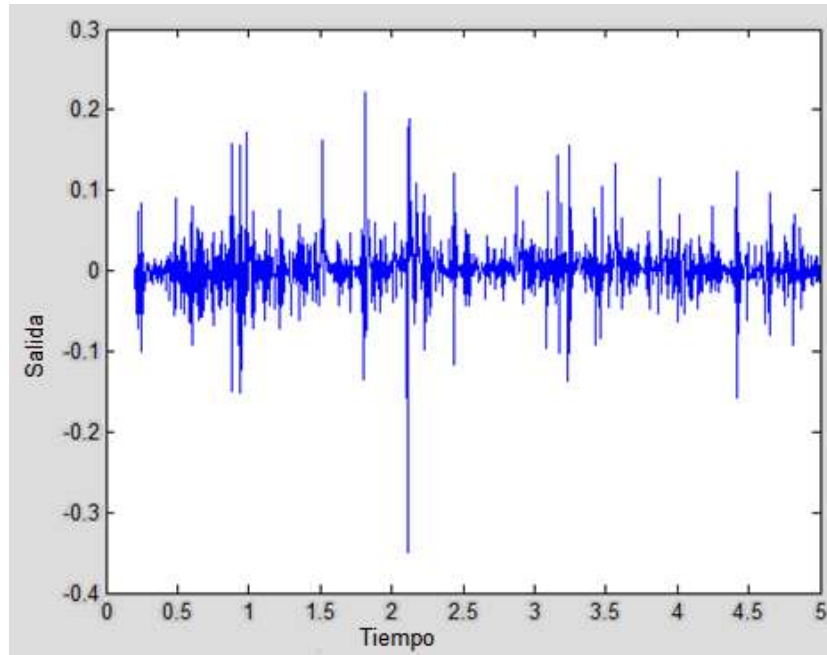


Figura 63: Representación del error

3.3.2.5 T=0,007 segundos

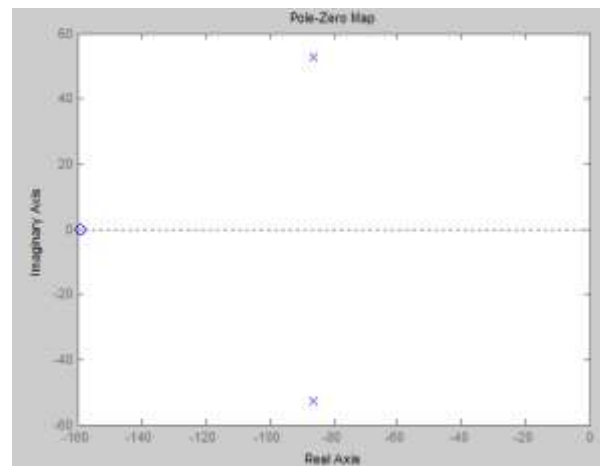
La Función de Transferencia correspondiente al mejor caso con este periodo de muestreo es:

$$\frac{60.86 s + 9678}{s^2 + 172.5 s + 1.02e004}$$

Tras haber obtenido la Función de Transferencia se analizan los polos y los ceros utilizando el comando “pzmap”:

```
p =
-86.2475 +52.5764i
-86.2475 -52.5764i

z =
-159.0168
```





En cuanto a los polos y ceros, el sistema es de segundo orden con dos polos en $-84,25$ con una pequeña parte imaginaria que no aporta una sobreoscilación estimable (menor del 0,6%) y un cero en -159 .

La ganancia del sistema es 0.9488, compatible con casos anteriores.

Como vemos este caso es prácticamente similar a los obtenidos con el Output Error.

La última comprobación a realizar es comprobar las salidas, tanto la del motor como la procedente del Matlab, con respecto a la entrada.

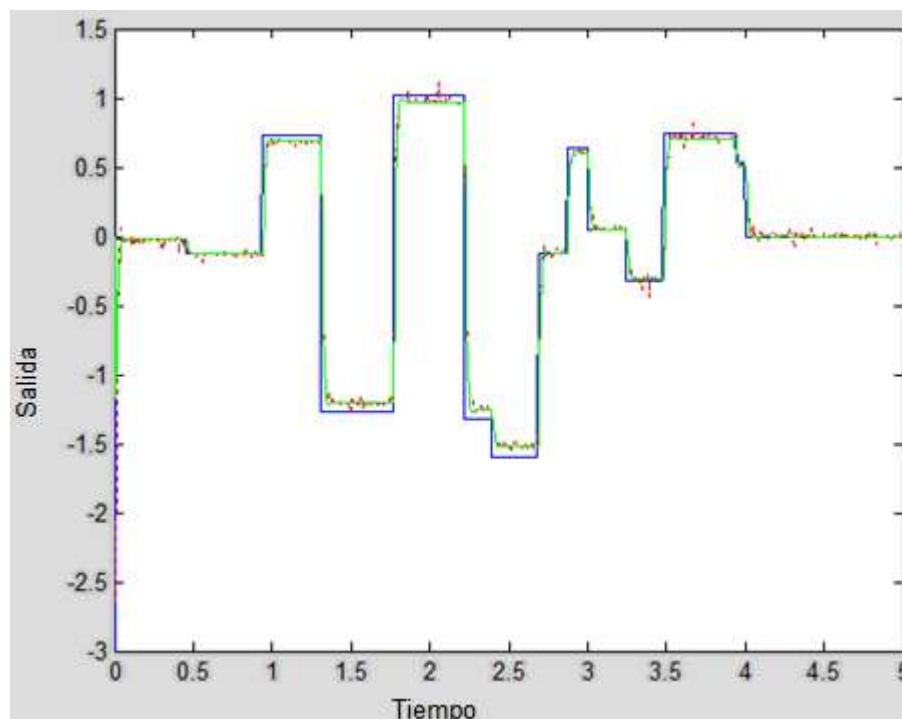


Figura 64: Representación de las salidas respecto de la entrada

La diferencia entre la señal de salida del motor y la salida ruidosa procedente de la Función de Transferencia identificada no se puede apreciar en la gráfica de la parte superior. En la gráfica que se muestra a continuación, se va a representar la diferencia entre ambas señales. Dicha gráfica no va a ser representada desde el inicio de la simulación, ya que queremos evitar el efecto de las condiciones iniciales, por lo que comenzamos a partir de los 0,2 segundos.

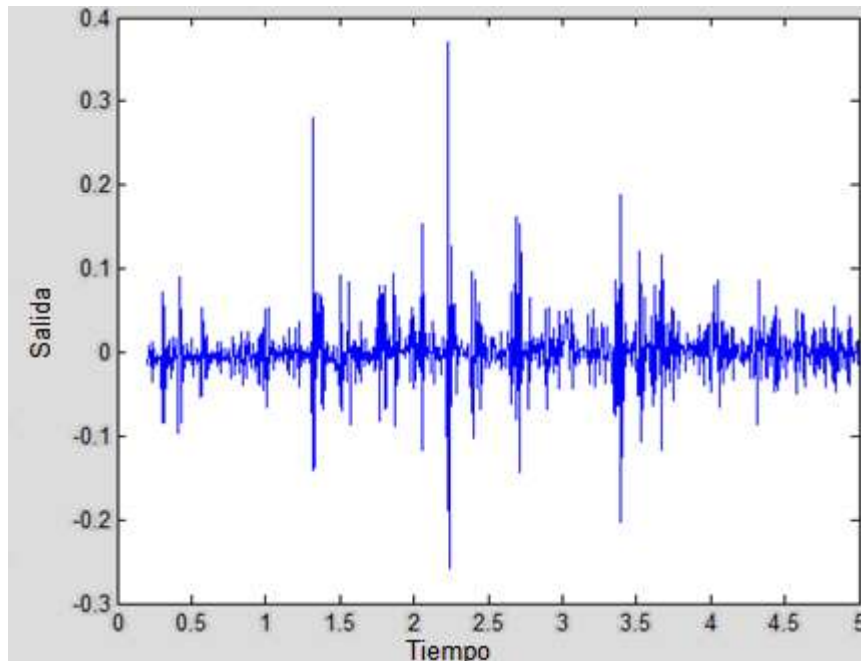


Figura 65: Representación del error

Del análisis anterior podemos concluir que el sistema se puede aproximar por dos modelos diferentes de segundo orden. En el primer caso, aparecerían polos alrededor de -110 , en el segundo estaría alrededor de -90 (en ambos casos con partes imaginarias despreciables) y con un cero alrededor de -300 y -170 , respectivamente. En ambos casos podemos aproximar la ganancia a 0.946 . El segundo caso es similar al del Output Error. Esto lo podemos expresar con las siguientes Funciones de Transferencia (dando como parte imaginaria $+65i$ y $+60i$ respectivamente):

$$F1(s) = \frac{51.48s + 15440}{s^2 + 220s + 16325}$$

$$F2(s) = \frac{65.2447 \cdot (s + 170)}{s^2 + 1800s + 11700}$$

También se puede aceptar que los polos sean reales puros, sin parte imaginaria.



Ahora bien, estos sistemas tienen una respuesta similar porque si bien el segundo tiene polos más lentos su respuesta se hace más rápida gracias al cero. Como se puede ver en la siguiente gráfica donde mostramos la respuesta de los dos sistemas (en azul F1 y en verde F2) ante entrada escalón.

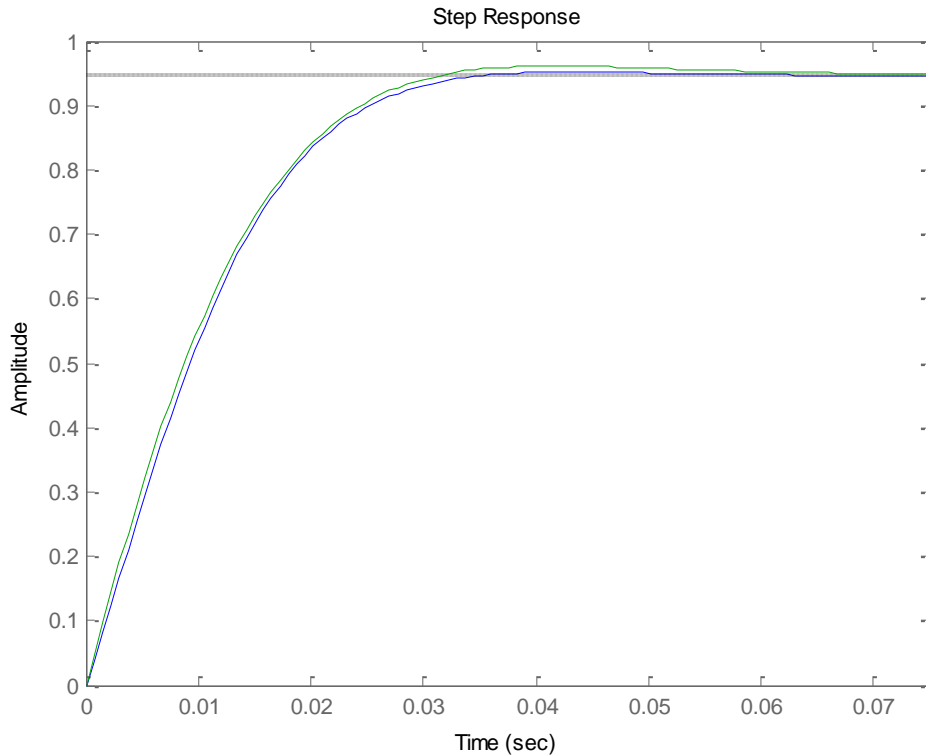


Figura 66: Respuesta ante una entrada escalón

Por lo tanto, ambos casos son igual de aceptables.

Aceptamos como modelo el segundo caso ya que es el también se obtiene del modelo de Output Error. Aunque harían falta más estudios (por ejemplo utilizando técnicas frecuenciales: Diagrama de Bode) para asegurarnos de la bondad de esta propuesta.





Capítulo 4 CONCLUSIONES

Tras la realización del proyecto llegamos a una serie de conclusiones.

Se ha obtenido formación:

- Nociones de control digital: Tras haber realizado el estudio de la maqueta adquirimos una serie de conocimientos que nos permitirán analizar un sistema discreto.
- Nociones de identificación: En las que se incluyen los modelos dinámicos Output Error, ARX y ARMAX, así como el toolbox de identificación de Matlab.
- Uso de Simulink: Esta herramienta apenas se había usado durante la titulación. En partículas se ha usado Simulink para estimar el ruido que tiene la maqueta en los casos de los modelos dinámicos de ARX y ARMAX.
- La relación entre el modelo de un sistema continuo y el modelo discretizado del mismo sistema: Para realizar la identificación de la maqueta debemos tener en cuenta las distintas características que adquiere el motor, siendo el motor un sistema continuo. El análisis se realiza discretizando el sistema.

Para identificar la maqueta se han seguido una serie de pasos, estudiando distintos sistemas que nos ayudarán a que finalmente identifiquemos la maqueta motor.

- Sistemas simulados discretos: Se estudiaron las técnicas de identificación aplicadas a sistemas discretos, analizando también la influencia del ruido en estos procesos.
- Sistemas simulados continuos discretizados: Estudiando la influencia del periodo de muestreo en la identificación en presencia de ruido. También se estudiaron los órdenes de Matlab para el paso de sistema continuo a discretizado y viceversa.
- Sistema continuo simulado: la maqueta es un sistema de este tipo. Con los se estudió el uso de los retenedores y muestreadores.



La maqueta que tenemos es un sistema continuo. Los datos que se toman de ella son discretos gracias al uso de un retenedor de orden cero y un muestreador. Para identificar el sistema tenemos que tener en cuenta el modelo dinámico que puede caracterizar el sistema. Puede ser de tres tipos distintos: Output Error, ARX y ARMAX.

- Output Error.

Tras haber analizado la maqueta con este modelo dinámico, vemos que los resultados obtenidos en la mayoría de las simulaciones de este modelo, son mejores que los resultados obtenidos en los otros dos modelos.

- ARX

El modelo ARX falla en la mayoría de las identificaciones y en los casos que no lo hace comete errores o no da resultados coherentes entre sí. Tras haber analizado la maqueta con este modelo dinámico llegamos a la conclusión de que este modelo no corresponde al de la maqueta.

- ARMAX

Como en el modelo ARX falla en algunas de las identificaciones. En los casos en que se obtienen resultados aceptables, los datos obtenidos son correctos, ya que los polos y ceros se colocan en lugares lógicos y el ruido no es muy elevado. Además lo obtenido es compatible con los resultados, en todo caso más coherentes, dados por el modelo Output Error.

Podíamos concluir que la maqueta se puede representar como un sistema dinámico de Output Error con una Función de Transferencia como la siguiente:

$$G(s) = 65.2447 \frac{s + 170}{s^2 + 180s + 11700}$$

La Función de Transferencia es de segundo orden con los polos en $-90 \pm 60i$ y el cero, no se ha identificado de forma más precisa, pero ronda un valor de -170 con una ganancia de 0.946 .



Tras haber realizado el proyecto, podemos tener distintos caminos para ampliarlo:

- Refinar la identificación, dejando fijos los polos
- Realizar un diagrama de Bode
- Compararlo con otras maquetas
- Identificar la zona muerta



BIBLIOGRAFÍA

- IDENTIFICACIÓN DE SISTEMAS. José Basilio Galván Herrera
- SYSTEM IDENTIFICATION. Torsten Soderström y Petre Stoica
- SISTEMAS DE CONTROL DE TIEMPO DISCRETO. Katsuhiko Ogata