

FACULTAD DE INGENIERÍA INDUSTRIAL, SISTEMAS E INFORMÁTICA
ESCUELA ACADÉMICO PROFESIONAL INGENIERÍA ELECTRÓNICA



TESIS

**“DISEÑO DE UN IDENTIFICADOR DE PROCESOS USANDO
PLATAFORMA DE 8 BITS MEDIANTE ALGORITMO LMS”
PARA OBTENER EL TÍTULO PROFESIONAL DE INGENIERO
ELECTRÓNICO**

PRESENTADO POR:

Arthur Chanel Palomino Manrique

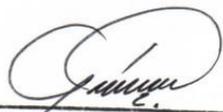
ASESOR:

Ms. Ing. Oscar Miguel De La Cruz Rodríguez

CIP: 85598

HUACHO – PERÚ

2022



Ms. Oscar Miguel De La Cruz Rodríguez
Ingeniero Electrónico
CIP 85598

DISEÑO DE UN IDENTIFICADOR DE PROCESOS USANDO PLATAFORMA DE 8 BITS MEDIANTE ALGORITMO LMS

INFORME DE ORIGINALIDAD

18%

INDICE DE SIMILITUD

16%

FUENTES DE INTERNET

1%

PUBLICACIONES

9%

TRABAJOS DEL ESTUDIANTE

FUENTES PRIMARIAS

1	repositorio.unjfsc.edu.pe Fuente de Internet	2%
2	repositorio.upao.edu.pe Fuente de Internet	2%
3	Submitted to Universidad Nacional Jose Faustino Sanchez Carrion Trabajo del estudiante	2%
4	vsip.info Fuente de Internet	1%
5	hdl.handle.net Fuente de Internet	1%
6	tesis.pucp.edu.pe Fuente de Internet	1%
7	www.coursehero.com Fuente de Internet	1%
8	Submitted to Universidad Tecnologica del Peru Trabajo del estudiante	1%

**DISEÑO DE UN IDENTIFICADOR DE
PROCESOS USANDO PLATAFORMA DE 8
BITS MEDIANTE ALGORITMO LMS**

DEDICATORIA

A mis queridos padres Pedro y María, por su apoyo incondicional que me brindaron poder cumplir mis objetivos en la etapa universitaria y que son todo para mí.

TABLA DE CONTENIDO

CARÁTULA	iii
TÍTULO	iii
DEDICATORIA	iii
TABLA DE CONTENIDO	iv
TABLA DE FIGURAS	vi
TABLA DE TABLAS	ix
RESUMEN.....	x
CAPÍTULO I: PLANEAMIENTO DEL PROBLEMA.....	1
1.1. Descripción de la realidad problemática.....	1
1.2. Formulación del problema	1
1.2.1. Problema general	1
1.2.2. Problemas específicos.....	2
1.3. Objetivos de la investigación	2
1.4. Justificación de la investigación	2
1.5. Delimitación del estudio.....	2
CAPÍTULO II: MARCO TEÓRICO	3
2.1. Antecedentes de la investigación.....	3
2.2. Bases teóricas.....	8
Algoritmo de gradiente estocástico.....	8
2.3. Definición de términos básicos.....	13
2.4. Hipótesis de investigación.....	13
2.5. Operacionalización de las variables.....	13
CAPÍTULO III: METODOLOGÍA	15
3.1. Diseño metodológico.....	15
3.2. Población y muestra	15
3.3. Técnicas de recolección de datos.....	16
3.4. Técnicas para el procesamiento de la información	16

CAPÍTULO IV: RESULTADOS	47
4.1. Análisis de los resultados	47
CAPÍTULO V: DISCUSIÓN	55
5.1. Discusión de resultados	55
CAPÍTULO VI: CONCLUSIONES Y RECOMENDACIONES	56
6.1. Conclusiones	56
6.2. Recomendaciones	56
CAPÍTULO VII: REFERENCIAS.....	57
7.1. Fuentes bibliográficas	57
ANEXOS	58

TABLA DE FIGURAS

Figura 1: Herramienta IDENT del software Matlab	9
Figura 2: Representación gráfica de la respuesta a un escalón de un sistema de primer orden.....	10
Figura 3: Respuesta de sistema de segundo orden ante la entrada escalón unitario	12
Figura 4: Filtro adaptativo como identificador de procesos	16
Figura 5: Planta de primer orden con polo ubicado en $s=-4$ con excitación del sistema	18
Figura 6: Parámetros de la entrada del sistema	19
Figura 7: Parámetros del filtro LMS.....	20
Figura 8 Parámetros del bloque LMS para 6 coeficientes y convergencia 0.001- Régimen estacionario	21
Figura 9: Simulación de la Respuesta Estacionaria con 6 coeficientes y convergencia 0.001	22
Figura 10: Simulación de la Respuesta Estacionaria con 6 coeficientes y convergencia 0.001.....	24
Figura 11 Parámetros LMS para 10 coeficientes y convergencia 0.001- Régimen estacionario	25
Figura 12: Simulación de la Respuesta Estacionaria con 10 coeficientes y convergencia 0.001	26
Figura 13: Simulación del bloque LMS para una convergencia 0.001 y coeficiente 10.....	27
Figura 14: Parámetros LMS para 20 coeficientes y convergencia 0.001- Régimen estacionario	28
Figura 15: Simulación de la Respuesta Estacionaria con 20 coeficientes y convergencia 0.001.....	28
Figura 16: Simulación del bloque LMS para una convergencia 0.001 y coeficiente 20.....	29
Figura 17: Configuración de parámetros de la función escalón	31
Figura 18: Parámetros del bloque LMS para 6 coeficientes y convergencia 0.001- Régimen transitorio.....	32

Figura 19: Simulación de la Respuesta Transitoria con 6 coeficientes y convergencia 0.001	33
Figura 20: Respuesta transitoria para una convergencia 0.001 y coeficiente 6	33
Figura 21: Parámetros del bloque LMS para 10 coeficientes y convergencia 0.001- Régimen transitorio	34
Figura 22: Simulación de la Respuesta Transitoria con 10 coeficientes y convergencia 0.001	35
Figura 23: Respuesta transitoria para una convergencia 0.001 y coeficiente 10	36
Figura 24: Parámetros del bloque LMS para 20 coeficientes y convergencia 0.001- Régimen transitorio	37
Figura 25: Simulación de la Respuesta Transitoria con 20 coeficientes y convergencia 0.001	38
Figura 26: Respuesta transitoria para una convergencia 0.001 y coeficiente 20	38
Figura 27: Circuito utilizando microcontrolador PIC16F877	40
Figura 28: Circuito para ingreso y salida de datos	40
Figura 29 Simulación en Proteus para 6 coeficientes y convergencia 0.001	41
Figura 30: Simulación en PROSPICE para 6 coeficientes y convergencia 0.001	41
Figura 31: Simulación en Proteus para 10 coeficientes y convergencia 0.001	42
Figura 32: Simulación en PROSPICE para 10 coeficientes y convergencia 0.001	43
Figura 33: Simulación en Proteus para 20 coeficientes y convergencia 0.001	44
Figura 34: Simulación en PROPICE para 20 coeficientes y convergencia 0.001	45
Figura 35: Simulación en PROSPICE para 6 coeficientes y convergencia 0.001	45
Figura 36: Simulación en PROSPICE para 10 coeficientes y convergencia 0.001	46
Figura 37: Simulación en PROSPICE para 20 coeficientes y convergencia 0.001	46
Figura 38: Evolución de la respuesta estacionaria para 6 coeficientes	52
Figura 39: Evolución de la respuesta estacionaria para 10 coeficientes	52
Figura 40: Evolución de la respuesta estacionaria para 20 coeficientes	53

Figura 41: Evolución del periodo de establecimiento para 6,10 y 20 coeficientes 54

TABLA DE TABLAS

Tabla 1: Variable independiente.....	13
Tabla 2: Variable Dependiente	14
Tabla 3: Valores de los 6 coeficientes para la respuesta en estado estacionario.....	24
Tabla 4: Valores de los 10 coeficientes para la respuesta en estado estacionario.....	27
Tabla 5: Valores de los 20 coeficientes para la respuesta en estado estacionario.....	30
Tabla 6: Valores de los 6 coeficientes para la respuesta en estado transitorio	34
Tabla 7: Valores de los 10 coeficientes para la respuesta en estado transitorio	36
Tabla 8: Valores de los 20 coeficientes para la respuesta en estado transitorio	39
Tabla 9 Respuesta en régimen estacionario con 6 coeficientes.....	47
Tabla 10 Respuesta en régimen estacionario con 10 coeficientes.....	47
Tabla 11 Respuesta en régimen estacionario con 20 coeficientes.....	48
Tabla 12 Respuesta en régimen estacionario con 20 coeficientes.....	49
Tabla 13 Respuesta en régimen estacionario en Matlab para planta de primer orden con 6, 10 y 20 coeficientes	50
Tabla 14 Respuesta en régimen transitorio en Matlab para planta de primer orden con 6,10 y 20 coeficientes	50
Tabla 15 Respuesta estacionaria del Microcontrolador para planta de primer orden con 6,10 y 20 coeficientes	51
Tabla 16 Respuesta transitoria del Microcontrolador para planta de primer orden con 6, 10 y 20 coeficientes	51

RESUMEN

En la presente investigación “**DISEÑO DE UN IDENTIFICADOR DE PROCESOS USANDO PLATAFORMA DE 8 BITS MEDIANTE ALGORITMO LMS**”.

En la investigación se desarrolla un algoritmo implementado en un microcontrolador 16F877 para la identificación de procesos, la realidad del problema y los objetivos de la investigación se analizan en el primer capítulo, y en el segundo capítulo se realizó el marco teórico. Para el desarrollo de la investigación, en el tercer capítulo se realiza la adquisición de datos, haciendo uso del software de simulación Matlab, y a través del microcontrolador 16f877, la prueba es en régimen permanente y régimen transitorio. En el cuarto capítulo, mediante los datos obtenidos se procesan en tablas y gráficos, y los resultados se comentan en el quinto capítulo, que finaliza con las conclusiones y recomendaciones.

El diseño es pre experimental y la investigación es de tipo aplicada.

Por último, el algoritmo para la identificación de procesos se realizó en Proton Basic para 6, 10 y 20 coeficientes, obteniéndose un tiempo de 2.8ms de proceso en el algoritmo.

Palabras Claves: Algoritmo, microcontrolador, régimen.

ABSTRACT

In the present investigation "DESIGN OF A PROCESS IDENTIFIER USING AN 8-BIT PLATFORM THROUGH LMS ALGORITHM".

In the investigation, an algorithm implemented in a 16F877 microcontroller is developed for the identification of processes, the reality of the problem and the objectives of the investigation are analyzed in Chapter 1, and in Chapter 2 through the theoretical framework. To investigate, in chapter 3 the data acquisition is carried out, using the Matlab simulation software, because in the next part the data acquisition is carried out using a 16f877 microcontroller, the test is in permanent regime and transient regime, in the fourth chapter, the data obtained are processed in the form of tables and graphs, and the results are discussed in chapter 5, which ends with the conclusions and recommendations.

The design is pre-experimental and the research is applied.

Finally, the algorithm for the identification of processes was carried out in Proton Basic for coefficients of 6, 10 and 20, obtaining an algorithm process time of 3ms.

Keywords: Algorithm, microcontroller, steady state.

INTRODUCCIÓN

CAPÍTULO I: PLANEAMIENTO DEL PROBLEMA

1.1. Descripción de la realidad problemática

En la actualidad, los procesos industriales tienen una relación de entrada y salida llamada función de transferencia que determina la estabilidad del proceso y permite calcular el tiempo de estabilización, parámetro clave en los procesos industriales, esta función de transferencia puede ser calculada teóricamente usando ecuaciones diferenciales. Modelando el dispositivo y luego, la transferencia de este sistema dinámico al plano laplaciano, permite al diseñador incorporar un bloque en serie con la función de transferencia para mejora del tiempo de respuesta; este bloque a menudo se denomina controlador PID.

El modelo dinámico que describe el modelo de planta en la práctica cambia con el tiempo, lo que significa que la función de transferencia de la planta cambia con el tiempo, generalmente porque los componentes que componen la planta se degradan con el tiempo, lo que modifica la dinámica de los sistemas también cuando se realiza el mantenimiento correctivo. se realiza en el equipo y se le hacen modificaciones, todo lo cual hace que la función de transferencia del equipo cambie, de modo que los controladores diseñados para mejorar los tiempos de respuesta ya no funcionan como se esperaba.

El problema radica en determinar el modelo de la planta cuando este cambia en forma temporal.

1.2. Formulación del problema

1.2.1. Problema general

¿Cómo el diseño del algoritmo LMS en los procesos industriales permitirá la identificación del proceso?

1.2.2. Problemas específicos

- ¿Cómo el uso del lenguaje de alto nivel permitirá la implementación del algoritmo LMS?
- ¿Cómo la identificación del proceso de primer orden permitirá determinar el modelo del algoritmo LMS?
- ¿Cómo la función de transferencia permitirá comprobar el identificador de procesos en sistemas de primer orden?

1.3. Objetivos de la investigación

1.3.1. Objetivo general

Diseñar un identificador de procesos usando algoritmo LMS.

1.3.2. Objetivos específicos

- Elaborar el algoritmo LMS en un microcontrolador, haciendo uso de la programación de alto nivel.
- Determinar el modelo del algoritmo LMS para la identificación del proceso de una planta de primer orden.
- Comprobar el identificador de procesos a través de la función de transferencia del sistema.

1.4. Justificación de la investigación

El presente trabajo permitirá elaborar una propuesta de solución a la implementación de un algoritmo LMS para la identificación del proceso.

1.5. Delimitación del estudio

El estudio está limitado solo hasta el diseño y la simulación del identificador de proceso usando Matlab y Proteus.

CAPÍTULO II: MARCO TEÓRICO

2.1. Antecedentes de la investigación

2.1.1. Investigaciones internacionales

(Zapata, 2020) en su tesis “Estudio comparativo de los algoritmos de filtrado adaptativo (LMS, NLMS, RLS) para la eliminación del ruido audible externo en tiempo real”. Investigación realizada en la Escuela Politécnica Nacional de la Facultad de Ingeniería Eléctrica y Electrónica de la ciudad de Quito. El presente trabajo presenta un estudio de algoritmos de mínimos cuadrados medios adaptativos (LMS), mínimos cuadrados medios normalizados (NLMS) y mínimos cuadrados recursivos (RLS) como propósito, la aplicación de estos algoritmos en tiempo real. Con base en este objetivo, se analizan ciertos conceptos como señales estacionarias y no estacionarias, filtros Wiener, filtrado adaptativo, además se estudian formulaciones matemáticas de los algoritmos LMS, NLMS y RLS para permitir sus simulaciones en tiempo real. Luego de este análisis e implementación, se determinó que el mejor algoritmo en términos de carga computacional y error total de muestra fue NLMS. Por lo tanto, se implementa en Simulink en tiempo real.

(Hérrandez, Chacón, & Navarro, 2019) en su artículo titulado “Implementación de un sistema generador y eliminador de ECO con filtro adaptativo usando el algoritmo LMS en un FPGA ARTIX-7”, investigación realizada por el Instituto Tecnológico de Chihuahua, México. En este trabajo se presenta un método para realizar procesamiento de señales vía filtros digitales en sistemas basados en FPGA en dos situaciones diferentes: cuando se conoce el modelo a implementar y cuando se desconoce. La principal contribución de esta investigación es la metodología de

implementación, diseñada con pocos recursos utilizando la versión gratuita del entorno de desarrollo Vivado de Xilinx.

(Del Angel, Maya, Avalos, Trejo, & Sánchez, 2022) en su artículo “Estructura alternante para sistemas de beamforming adaptativo basada en los algoritmos APL/SR-LMS” investigación realizada por la sección de estudios de Posgrado e Investigación del Instituto Politécnico Nacional de la Ciudad de México. Los resultados demuestran que la estructura propuesta tiene las mejores características de los algoritmos combinados, obteniendo un algoritmo con una alta velocidad de convergencia y menor costo computacional en comparación con otros algoritmos basados en combinaciones, convexas convencionales.

(Jiménez & Mauricio, 2017) en su artículo “Algoritmos LMS de filtrado adaptativo para cancelación de Eco Acústico en sistemas de Telecomunicaciones” investigación realizada por la Universidad Pedagógica y Tecnológica de Colombia. El objetivo general es comparar las características de modelado, simulación y rendimiento de los filtros LMS adaptativos para aplicaciones de cancelación de eco. En la parte metodológica, se revisan los tres algoritmos tradicionales de filtrado adaptativo, a saber, el algoritmo tradicional (LMS), el algoritmo LMS normalizado (NLMS) y el algoritmo LMS firmado (SLMS). Finalmente, la comparación entre los algoritmos se evalúa utilizando métricas de rendimiento como el tiempo de respuesta y la velocidad de convergencia.

(Cuevas, 2020) en su tesis “Algoritmos de Filtrado Kernel Adaptativo Multi-Output” Investigación realizada en la Universidad de Cantabria, España. Este proyecto se centrará en desarrollar nuevos modelos de aprendizaje multitarea, y más concretamente, algoritmos de filtrado adaptativo kernel de tipo multisalida que consigan explotar las correlaciones existentes entre distintas series temporales para

mejorar los resultados obtenidos. Además, su complejidad está diseñada para ser mucho menor que otros modelos desarrollados hasta la fecha, como los Procesos Gaussianos (GP). Para evaluar el método propuesto, se empleó una base de datos generada por simulación con cierto control sobre sus parámetros para realizar un análisis completo del desempeño de todos los algoritmos desarrollados, luego, los dos seleccionados su aplicación sobre la base de datos real exhiben mejor actuación.

2.1.2. Investigaciones nacionales

(Gastañadui & Sánchez, 2020) en su tesis “Implementación de un algoritmo LMS en PLC para la atenuación de ruido eléctrico en señales analógicas”, Investigación realizada en la Universidad Privada Antenor Orrego de la ciudad de Trujillo. El presente trabajo de investigación tiene como finalidad implementar un algoritmo LMS en un PLC para atenuar el ruido eléctrico en señales analógicas. Dado que las señales de los sensores en la industria son susceptibles al ruido, especialmente CA a 60 Hz, y muchos PLC no tienen filtros durante la lectura, esto puede causar serios problemas. Estas señales son críticas para controlar el proceso que interviene, por lo tanto, es necesario implementar un algoritmo LMS para atenuar este ruido presente en la señal analógica. El algoritmo LMS se implementa en un PLC con una frecuencia de muestreo de 12 ms, utilizando el lenguaje SCL. La distorsión armónica total (THD%) representa el grado de atenuación del ruido y es inversamente proporcional al factor de convergencia. El tiempo de adaptación del filtro es inversamente proporcional al factor de convergencia

(Colchado & Solano, 2019) en su tesis “Diseño de un controlador difuso basado en Microcontrolador usando el método del centroide”, Investigación realizada en la Universidad Privada Antenor Orrego de la ciudad de Trujillo. El propósito de

este trabajo de investigación es diseñar e implementar un controlador difuso utilizando el método del centroide en una plataforma de 8 dígitos. Dado que el método del centroide utiliza matemáticas complejas, se desarrolló un método alternativo en el que se reducen las matemáticas, lo que facilita la implementación de un controlador difuso dentro de una plataforma digital de 8 bits. En el presente trabajo de investigación se calculó el método simplificado anterior y de la misma manera se realizaron las simulaciones y pruebas necesarias para validar el método.

(Martell, 2018) en su tesis “Diseño e implementación de sistema electrónico para el monitoreo de la temperatura en neonatos” Investigación realizada en la Universidad Tecnológica del Perú de la ciudad de Lima. El presente trabajo de investigación consiste en medir la temperatura del recién nacido mediante un termistor NTC de alta precisión y luego digitalizarla mediante el convertidor analógico digital interna del microcontrolador PIC16F877A, que también procesará la señal adquirida. Después de completar el proceso En la imagen de arriba, el valor de la temperatura se muestra en la pantalla LCD de 16x2 y finalmente se envía a través del módulo Bluetooth HC-05, para que también se pueda mostrar en tiempo real en dispositivos Android profesionales. Health, desarrollado en App Inventor a través de una app. Este microcontrolador, a su vez, cuenta con diodos LED y un zumbador que actúan como alerta cuando la temperatura del recién nacido está dentro o fuera de rango. establecido. Cabe mencionar que la parte de programación se ha explicado utilizando el programa PIC C Compiler en lenguaje C.

(Torres, 2019) en su tesis “Diseño e implementación de un sistema de control electrónico para filtros hepa en una cabina de bioseguridad clase II para el laboratorio Bermanlab de Trujillo” Investigación realizada en la Universidad Tecnológica del Perú de la ciudad de Lima. El trabajo de investigación actual

incluye la implementación de un prototipo de sistema de control electrónico para filtros Hepa en un gabinete de bioseguridad con lógica de control de presión para garantizar un ambiente estéril, limpiar contaminantes y brindar al usuario la seguridad de las muestras y áreas de trabajo. En la primera etapa se verificó el desempeño del filtro Hepa, mediante la aplicación de software libre se logró desarrollar el diseño propuesto en el microcontrolador pic 16f877 de la familia Microchip.

Al igual que Proteus, contiene la etapa de simulación de nuestro sistema electrónico. Utilizamos componentes como transmisores de presión diferencial, microcontroladores PIC y componentes electrónicos. Nos ayudaron a diseñar e implementar un prototipo electrónico del control de alerta temprana de fallas del filtro Hepa. Las pruebas de rendimiento correspondientes se llevaron a cabo durante el desarrollo de un prototipo de control con un transmisor de presión diferencial, se obtuvieron valores precisos de presión diferencial de 0,5 inH₂O a 1,5 inH₂O, verificando dos niveles de presión diferentes a través de la manguera en las posiciones positiva y negativa del punto de estabilización del flujo de aire.

(Ramirez, 2022) en su tesis “Análisis del algoritmo Fista orientado a mejorar la velocidad de convergencia”, Investigación realizada en la Pontificia Universidad Católica del Perú de la ciudad de Lima. El presente trabajo de investigación consiste en estudiar el algoritmo de optimización de primer orden FISTA (Fast Iterative Thersholding Algortihm), el cual es utilizado en problemas inversos cuya solución se resume a la minimización de funciones convexas empleando información de la gradiente y de iteraciones previas. En este contexto, se analizan métodos que buscan la optimización del algoritmo por medio de tamaños de paso adaptativos para delimitar el paso de la gradiente y una mejor solución inicial mediante la

reducción de dimensiones a través de las técnicas conocidas como Screening y Warm Start, produciendo así datos más sparse. Además, se comprueba la eficacia de los métodos desarrollados por medio de un algoritmo generalizado, en el cual son evaluados los datos aleatorios generados sintéticamente en imágenes, con la finalidad de obtener la mejor tasa de convergencia.

2.2. Bases teóricas

Algoritmo de gradiente estocástico

El algoritmo LMS es un algoritmo de gradiente estocástico en el que cada valor de coeficiente del filtro transversal se itera en la dirección del gradiente y la señal de error es relativa a la magnitud al cuadrado del valor del coeficiente. Así, el algoritmo LMS está íntimamente relacionado con el concepto de aproximación estocástica descubierto en estadística por Robbins y Monro (1951) para resolver ciertos problemas de estimación de parámetros de secuencia.

a) Identificación De Procesos

Mediante la identificación del proceso se pretende:

Captar y almacenar datos de entrada y salida de la planta.

Obtener la función matemática que describe al proceso.

Para el modelamiento de la planta, se desarrolló en hardware y software una tarjeta de adquisición que facilite la toma de datos.

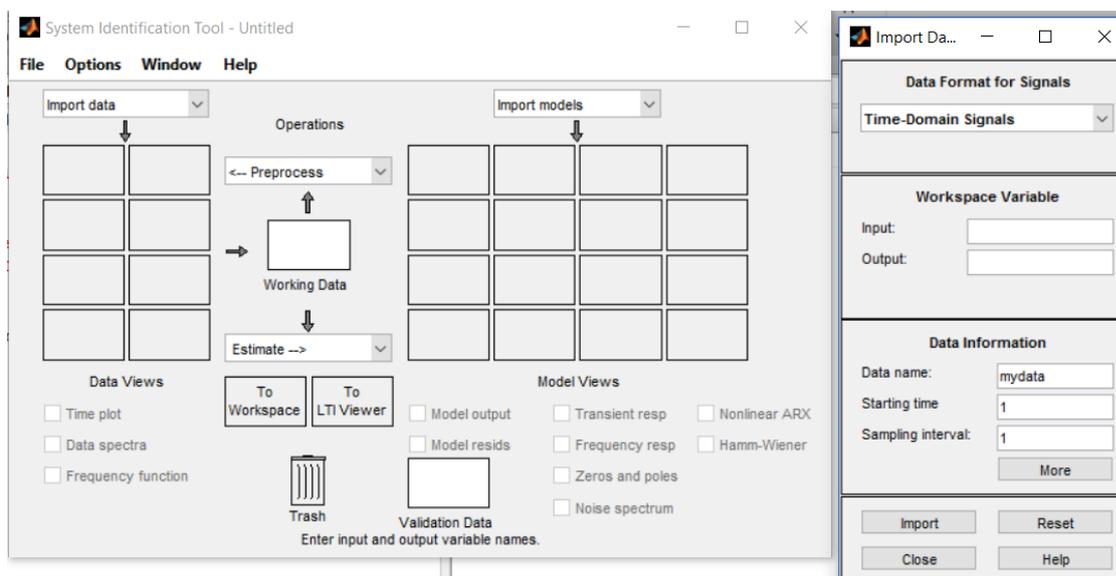
Inicialmente, para la tarjeta de adquisición de datos, se tiene que desarrollar el hardware, el cual es mostrado en el desarrollo del presente trabajo, con el cual se obtendrá las señales para luego ser transmitidas a la Pc. El software Labview se

encargará de recepcionar los datos para luego almacenarlos en un archivo con extensión XLS.

Una vez obtenidos los datos con la ayuda del software Labview, éstos se introducirán en la herramienta IDENT del software Matlab para la posterior obtención de la función matemática que describe la planta que controlaremos.

La figura 1 muestra la venta principal de la herramienta IDENT.

Figura 1
Herramienta IDENT del software Matlab



Fuente: (Cuevas, 2020)

Para describir el comportamiento de los procesos en los sistemas de control, se suelen utilizar modelos matemáticos. Estos modelos son esenciales para la simulación y el

diseño de control. Para el presente trabajo de investigación los sistemas que se modelan, se asemejan a sistemas de primer y segundo orden, es por esto que se detallan a continuación.

b) Sistemas de primer orden

Una planta de primer orden tiene un modelo matemático representado por la ecuación [1]:

$$G(s) = \frac{K}{1+Ts} \quad ec. (1)$$

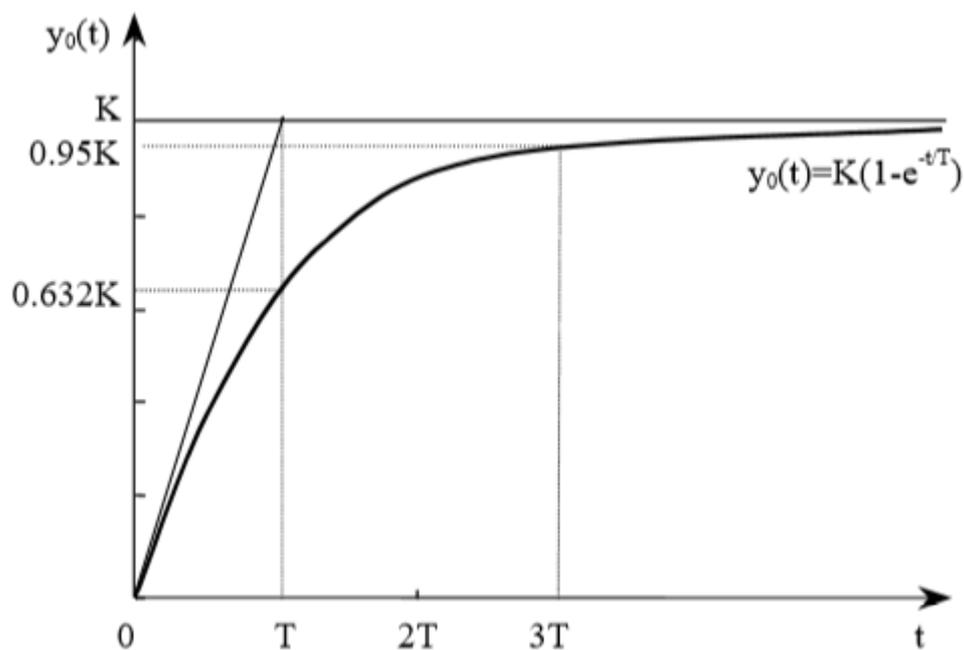
La respuesta de este sistema ante una entrada escalón unitario tiene por expresión:

$$y(t) = K\left(1 - e^{-\frac{t}{T}}\right) \quad ec. (2)$$

Por lo tanto, su representación gráfica es de la siguiente manera:

Figura 2

Representación gráfica de la respuesta a un escalón de un sistema de primer orden



Fuente: (Ogata, 1996)

Los parámetros característicos de esta grafica son:

- K: la ganancia estática se define como el valor final ante la entrada escalón unitario.
- T: constante de tiempo (es el tiempo en que se alcanza el 63% del valor final).
- $T_s = 3T$: tiempo de establecimiento (es el tiempo que tarda la respuesta en entrar y permanece en la zona del $\pm 5\%$ entorno a su valor de equilibrio).

c) Sistemas de segundo orden

$$G(s) = \frac{K\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \quad \text{ec. (3)}$$

Cuando se modela un sistema de segundo orden la función de transferencia se expresa de la siguiente forma:

Donde:

- ✓ K: ganancia estática
- ✓ ω_n : Frecuencia natural no amortiguada
- ✓ ξ : coeficiente de amortiguamiento

Los dos polos de este sistema pueden ser reales o complejos conjugados, dependiendo del valor que tome el coeficiente de amortiguamiento ξ . [4]

Para el caso de tener polos complejos conjugados estos serán de la forma:

$$s = -\sigma \pm \omega_d j \quad \text{ec. (5)}$$

Donde:

- ✓ $0 < \xi < 1$
- ✓ $\sigma = \xi\omega_n$: constante de amortiguamiento
- ✓ $\omega_d = \omega_n(1 - \xi^2)^{1/2}$: frecuencia amortiguada

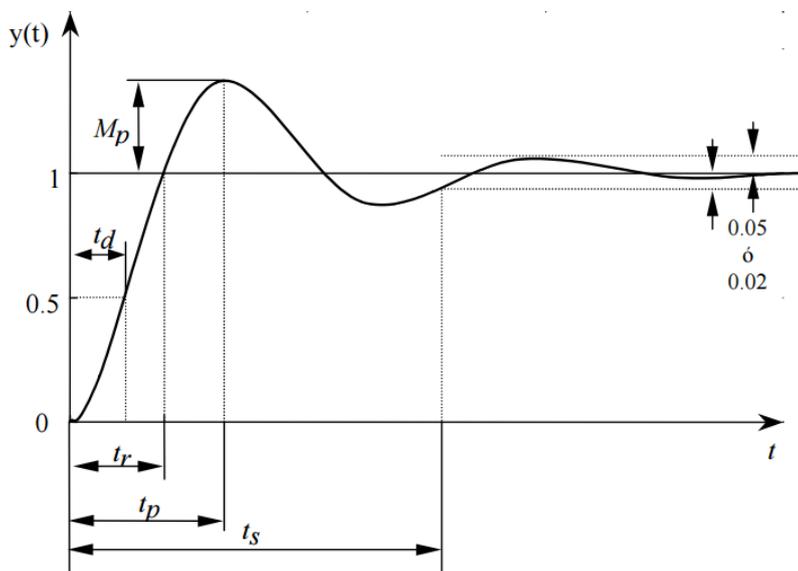
Si σ es positivo el sistema será estable. Si ξ es mayor que la unidad, los polos serán reales y el sistema no presentará oscilaciones. Por el contrario si ξ es menor que la unidad, los polos serán complejos y el sistema oscilará. [5]

$\xi < 0$	INESTABLE
$\xi > 1$	SOBREAMORTIGUADO
$\xi = 1$	CRITICAMENTE AMORTIGUADO
$0 < \xi < 1$	SUBAMORTIGUADO

Ante la función escalón unitario como señal de entrada, la respuesta de un sistema de segundo orden subamortiguado, queda representada en la figura 3.

Figura 3

Respuesta de sistema de segundo orden ante la entrada escalón unitario



Fuente: (Ogata, 1996)

Donde:

- Pendiente en el origen.

$$\dot{y}(0) = 0$$

- Tiempo de establecimiento.

$$t_s \approx \frac{\pi}{\sigma}$$

- Tiempo de subida.

$$t_r \approx \frac{\pi - \vartheta}{\omega_d} \quad \text{con} \quad \theta = \arctg \frac{\omega_d}{\sigma}$$

- Tiempo de pico.

$$t_p = \frac{\pi}{\omega_d}$$

- Sobreoscilación.

$$M_p = e^{-\frac{\zeta\pi}{\sqrt{1-\zeta^2}}} = e^{-\frac{\pi}{\operatorname{tg} \theta}}$$

2.3. Definición de términos básicos

Filtro adaptivo LMS: El filtro adaptativo LMS (Least Mean Squares) es un canal FIR TREE (Retroalimentación de motivación limitada) que se utiliza para reducir los sonidos.

Filtrado: Su función es extraer instantáneamente datos de una señal contaminada y usarla al mismo tiempo para actualizar el rendimiento del sistema sin causar error.

2.4. Hipótesis de investigación

2.4.1. Hipótesis general

El diseño del algoritmo LMS en los procesos industriales permitirá la identificación del proceso.

2.4.2. Hipótesis específicas

- El uso del lenguaje de alto nivel permite la implementación del algoritmo LMS.
- La identificación del proceso de primer orden permite determinar el modelo del algoritmo LMS.
- La función de función de transferencia permite verificar el identificador de procesos.

2.5. Operacionalización de las variables

Tabla 1

Variable independiente

VARIABLE	DEFINICION	INDICADORES	TÉCNICA	INSTRUMENTO
	CONCEPTUAL			

Algoritmo	Definida como la diferencia entre la señal deseada y la señal producida a la salida del filtro	Numero de coeficientes	Observación	Informe
LMS		Tiempo de convergencia	Observación	Informe

Tabla 2
Variable Dependiente

VARIABLE	DEFINICION CONCEPTUAL	INDICADORES	TÉCNICA	INSTRUMENTO
Identificación de proceso	Obtener la función matemática que describe al proceso.	Respuesta estacionaria	Observación	Reporte simulación
		Respuesta transitoria	Observación	Reporte simulación

CAPÍTULO III: METODOLOGÍA

3.1. Diseño metodológico

Tipo de investigación

Aplicada: A partir de conocimientos establecidos se dará solución a un problema planteado

Nivel de investigación

Aplicativo: Tiene como finalidad solucionar un problema.

Diseño:

Pre experimental: Se analiza una sola variable y el grado de control es mínimo.



O₁: Algoritmo LMS

X: Microcontrolador

O₂: Identificador de procesos

3.2. Población y muestra

3.2.1. Población

La población está representada por los bloques de transferencia de la planta.

3.2.2. Muestra

Funciones de transferencia de primer orden

3.3. Técnicas de recolección de datos

Mediante un informe, obtendremos los siguientes indicadores:

- Respuesta estacionaria de la planta de primer orden
- Respuesta transitoria de la planta de primer orden
- Tiempo de establecimiento del sistema
- Tiempo de desfase del sistema

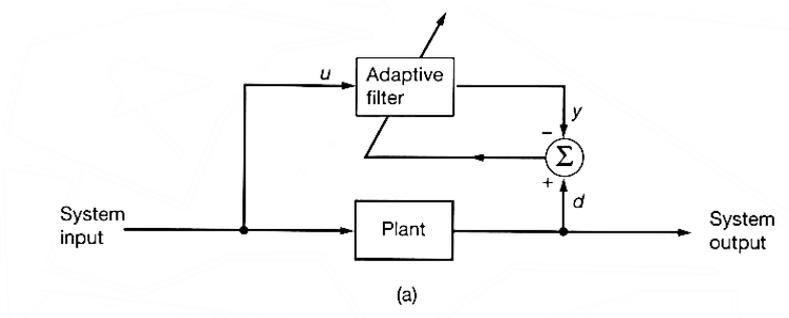
3.4. Técnicas para el procesamiento de la información

Diseño del Algoritmo LMS utilizado como identificador de Procesos

La figura 4, muestra el algoritmo LMS implementado, mediante un filtro adaptativo usado como identificador de procesos. La siguiente ecuación pertenece a un filtro de respuesta finita (FIR) $y(k) = \sum_{i=1}^N a_i u(k - n + 1)$

Figura 4

Filtro adaptativo como identificador de procesos



Fuente: (Creus, 2010)

Los coeficientes, a través de la función de minimización son los siguientes.

$$a_0(k) = a_0(k - 1) - \gamma 2e(k)r_2(k - 0)$$

$$a_1(k) = a_1(k - 1) - \gamma 2e(k)r_2(k - 1)$$

$$a_2(k) = a_2(k - 1) - \gamma 2e(k)r_2(k - 2)$$

$$\therefore a_{N-1}(k) = a_{N-1}(k-1) - \gamma 2e(k)r_2(k+1-N)$$

Donde:

γ : Factor de convergencia

$e(k)$: Error del sistema

$r_2(k) = u(k)$: Excitación del sistema

El algoritmo esta dado por la siguiente ecuación:

$$d(k) = Ch0(k)$$

$$u(k) = r_2(k) = Ch1(k)$$

$$e(k) = Ch0(k) - y(k)$$

Los pasos para la implementación del algoritmo, son los siguientes:

1. Considerar una interrupción cada 1ms
2. Definir y desplazar los datos de los coeficientes

$$a_{0i}, a_{1i}, a_{2i}, a_{3i}, a_{4i}, a_{5i} \dots a_n = 0$$

$$a_{0f} = a_{0i}; a_{1f} = a_{1i}; \dots a_{5f} = a_{5i} \dots a_{nf} = a_{ni}$$

3. Mover los valores del primer canal.

$$r_2(n) = r_2(n-1)$$

$$r_2(n-1) = r_2(n-2)$$

\vdots

$$r_2(2) = r_2(1)$$

$$r_2(1) = r_2(0)$$

4. Identificar $r_2(0) = ch1$.
5. Determinar $y(k) = a_{0f}r_2(0) + a_{1f}r_2(1) + a_{2f}r_2(2) + a_{3f}r_2(3) + a_{4f}r_2(4) + a_{5f}r_2(5) \dots a_{nf}r_2(n)$
6. Calcular el error $e(k) = ch0 - y(k)$
7. Determinar los coeficientes.

$$a_{0i} = a_{0f} + \gamma e(k)r_2(0)$$

$$a_{1i} = a_{1f} + \gamma e(k)r_2(1)$$

$$a_{2i} = a_{2f} + \gamma e(k)r_2(2)$$

$$a_{3i} = a_{3f} + \gamma e(k)r_2(3)$$

$$\vdots$$

$$a_{ni} = a_{nf} + \gamma e(k)r_2(n)$$

Respuesta Estacionaria

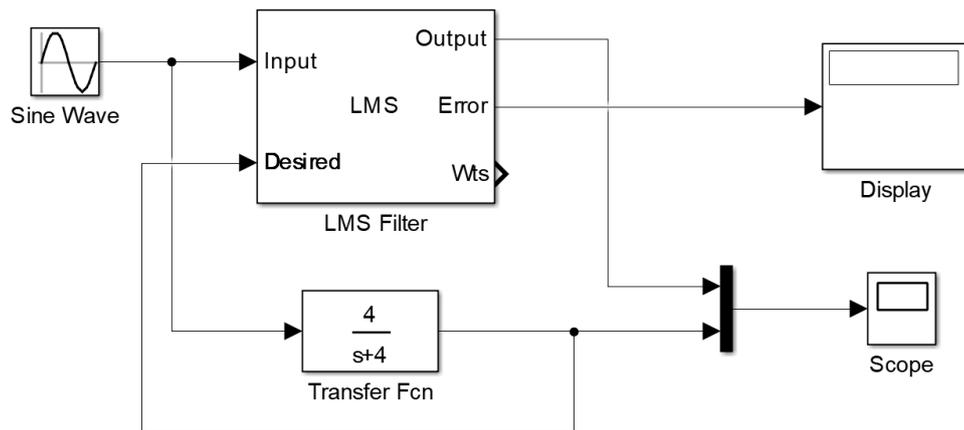
Para una planta de primer orden con polo en $s = -4$ y una excitación del sistema $v(t) = [2.5 + \sin(2\pi t)]v$.

$$F(s) = \frac{4}{s + 4}$$

En la figura 5 se puede apreciar la implementación de la planta de primer orden, utilizando el filtro adaptativo LMS para diferentes coeficientes y convergencia de 0.001, haciendo uso del software de simulación MATLAB.

Figura 5

Planta de primer orden con polo ubicado en $s=-4$ con excitación del sistema



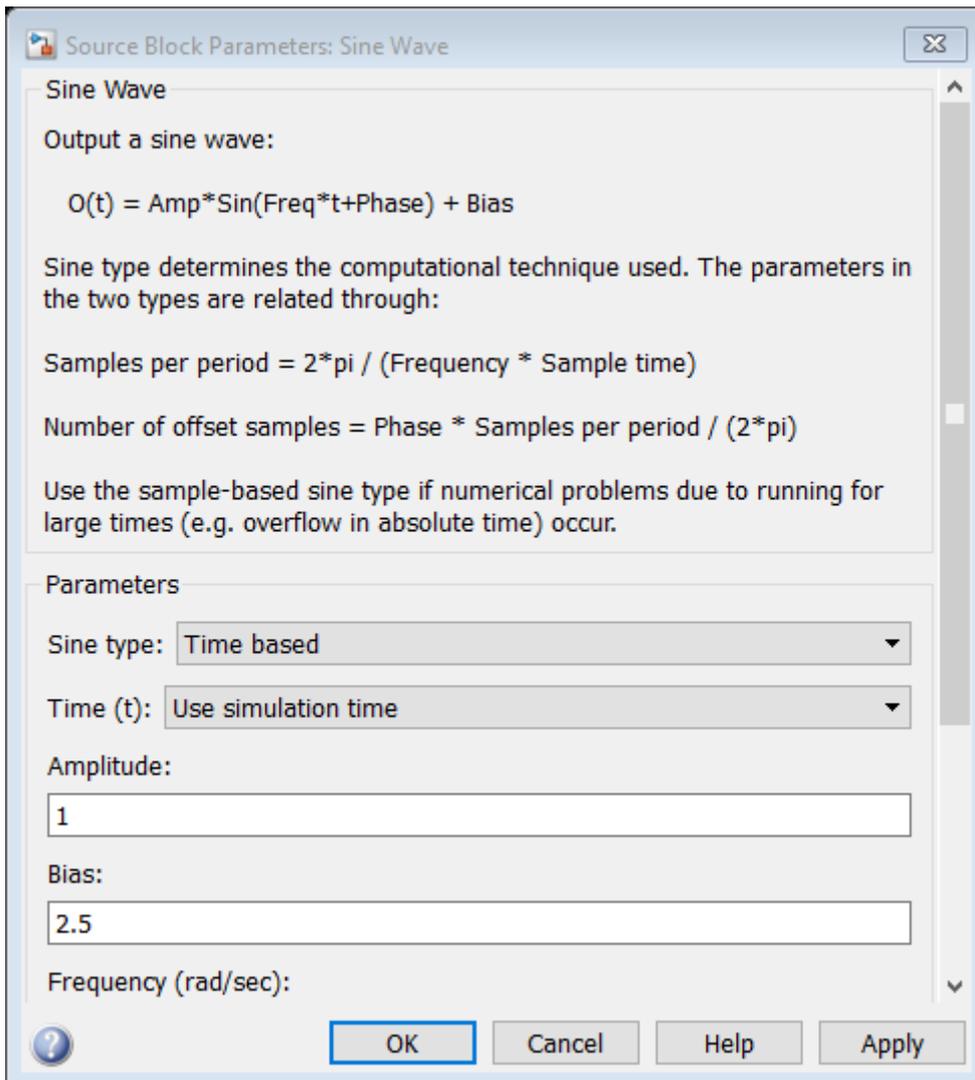
Fuente: Software Matlab - Simulink

La figura 6, indica los parámetros de la entrada del sistema, mediante la siguiente expresión

$$v(t) = [2.5 + \sin(2\pi t)]v.$$

Figura 6

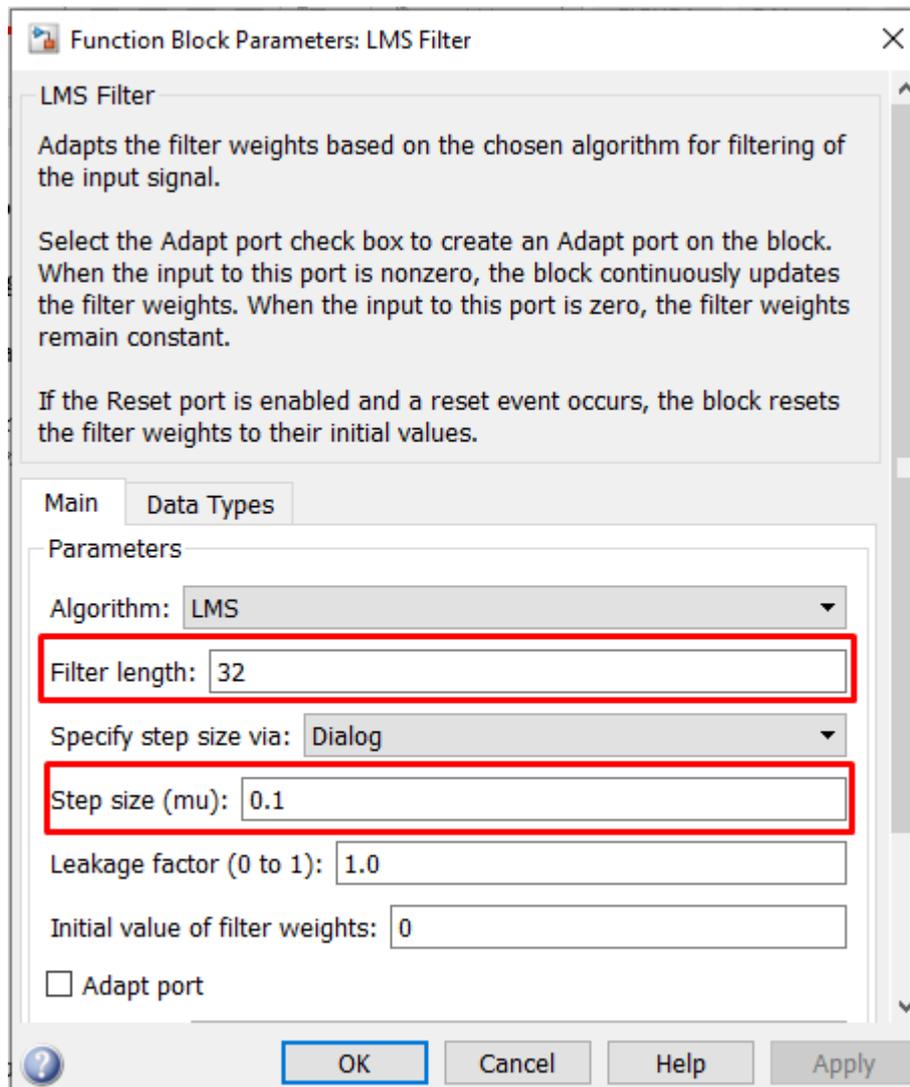
Parámetros de la entrada del sistema



Fuente: Software Matlab - Simulink

La figura 7, indica los parámetros del filtro LMS, para diferentes valores de coeficientes y convergencia.

Figura 7
Parámetros del filtro LMS



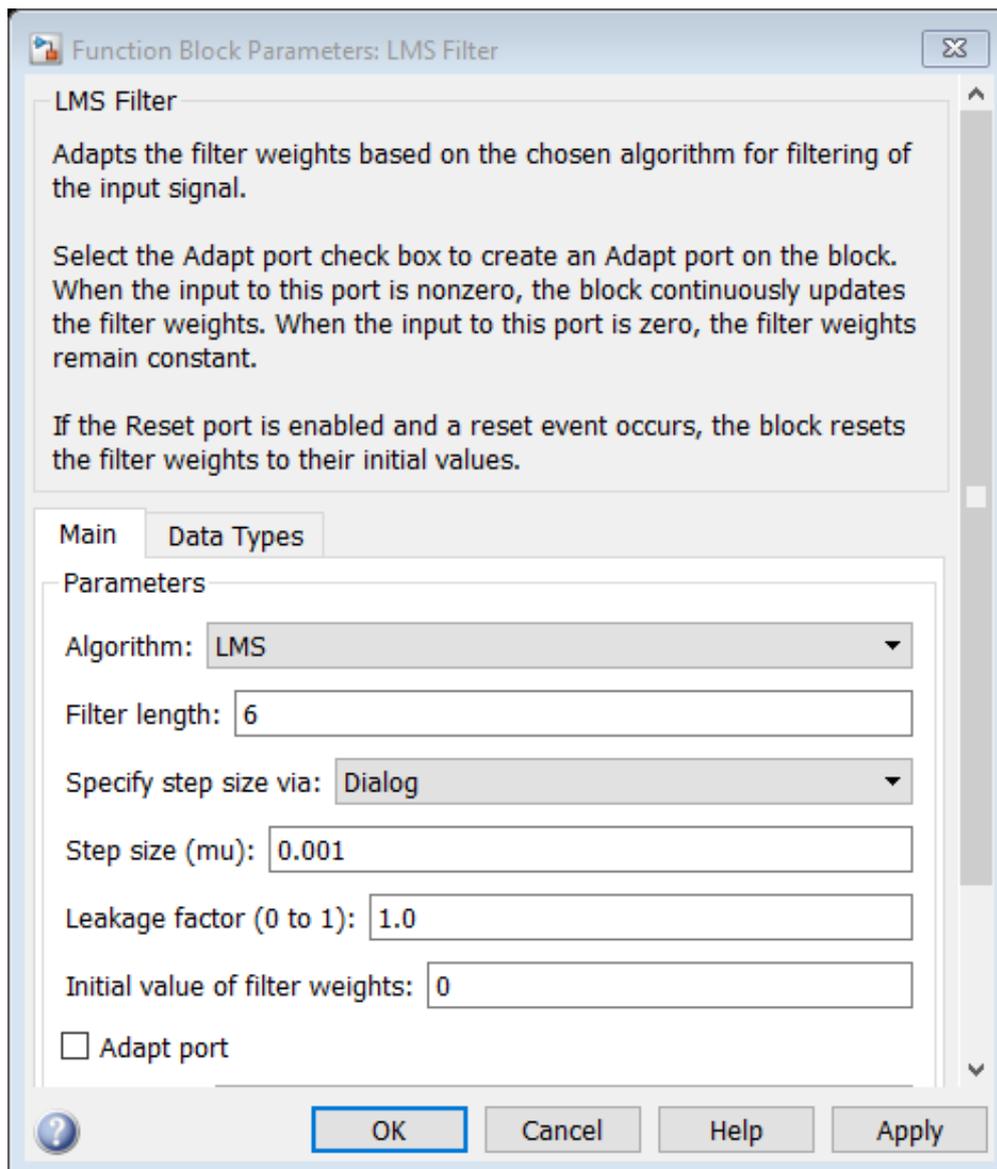
Fuente: Software Matlab - Simulink

Respuesta Estacionaria con 6 coeficientes y convergencia 0.001

La figura 8, indica la configuración de los parámetros del filtro LMS para 6 coeficientes y una convergencia de 0.001 en régimen estacionario.

Figura 8

Parámetros del bloque LMS para 6 coeficientes y convergencia 0.001- Régimen estacionario

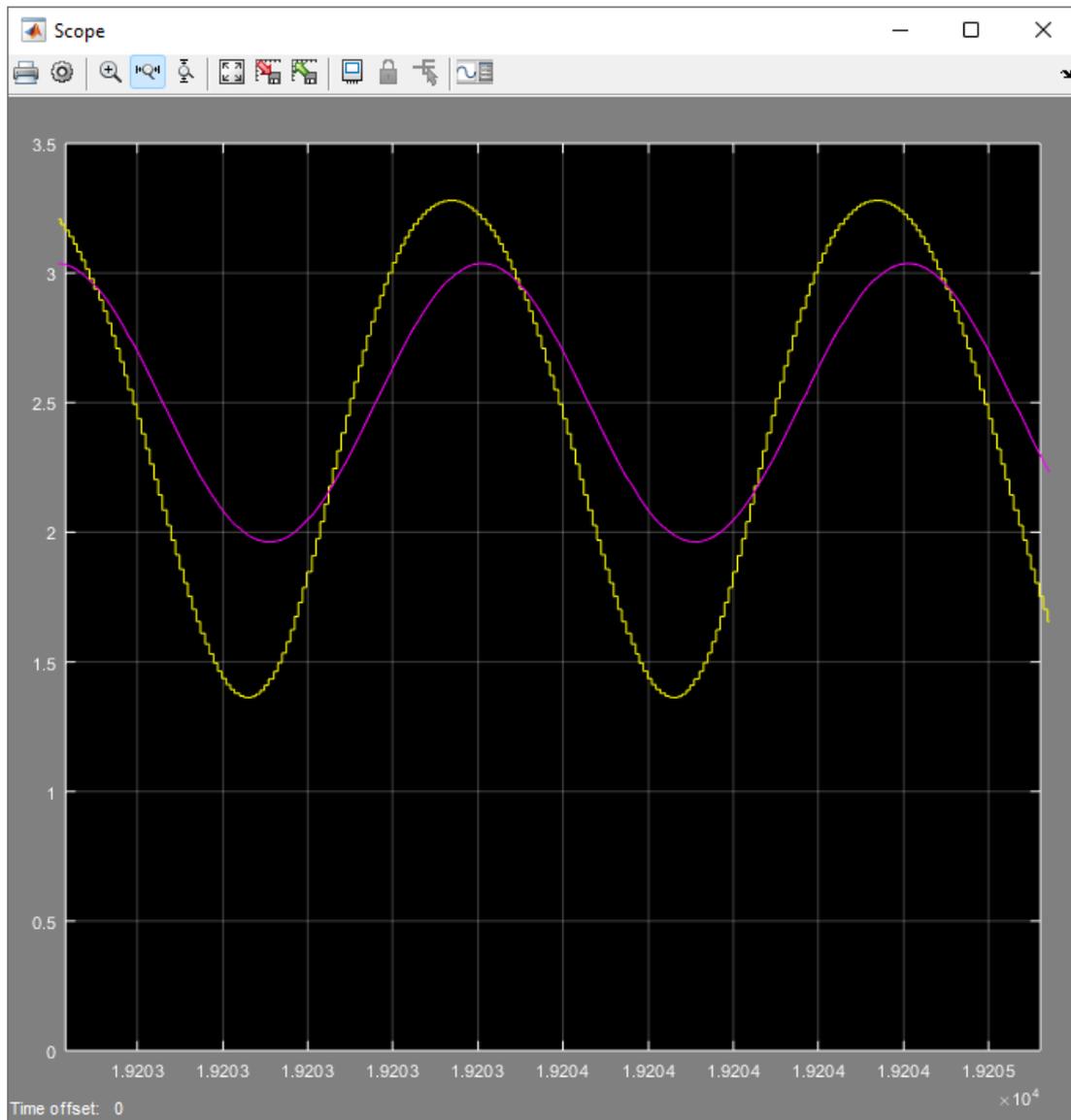


Fuente: Software Matlab - Simulink

La figura 9, indica que la señal tarda en estabilizarse aproximadamente 20000s.

Figura 9

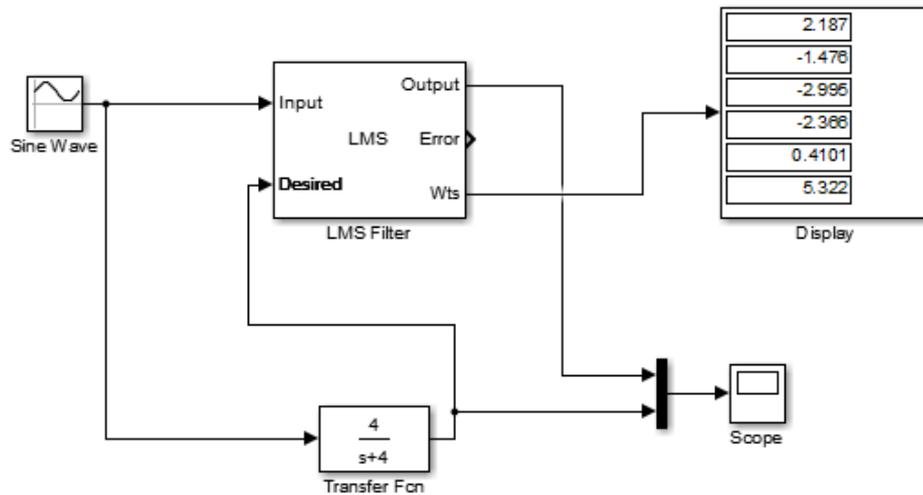
Simulación de la Respuesta Estacionaria con 6 coeficientes y convergencia 0.001



Fuente: Software Matlab - Simulink

La figura 10, indica la respuesta del sistema en simulink, para 6 coeficientes y una convergencia de 0.001.

Figura 10
 Simulación del bloque LMS para una convergencia 0.001 y coeficiente 6



Fuente: Software Matlab – Simulink

La tabla 3, indica los valores para 6 coeficientes de la respuesta en estado estacionario para una planta de primer orden, haciendo uso del software de simulación Matlab -Simulink.

Tabla 3
 Valores de los 6 coeficientes para la respuesta en estado estacionario

Número de coeficiente	Valor de coeficiente
1	2.187
2	-1.476
3	-2.995
4	-2.366
5	0.4101
6	5.322

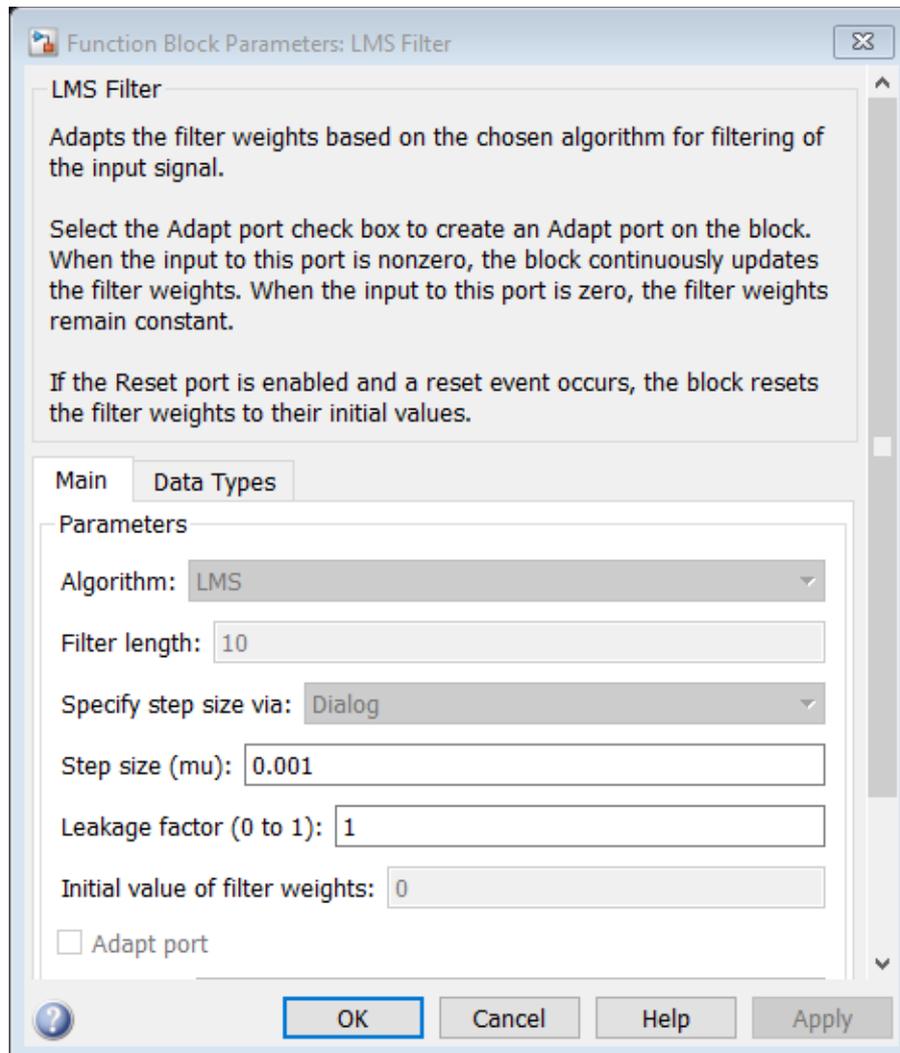
Fuente: Datos de la simulación en Matlab

Respuesta Estacionaria con 10 coeficientes y convergencia 0.001

La figura 11, indica la configuración de los parámetros del filtro LMS para 10 coeficientes y una convergencia de 0.001.

Figura 11

Parámetros LMS para 10 coeficientes y convergencia 0.001- Régimen estacionario

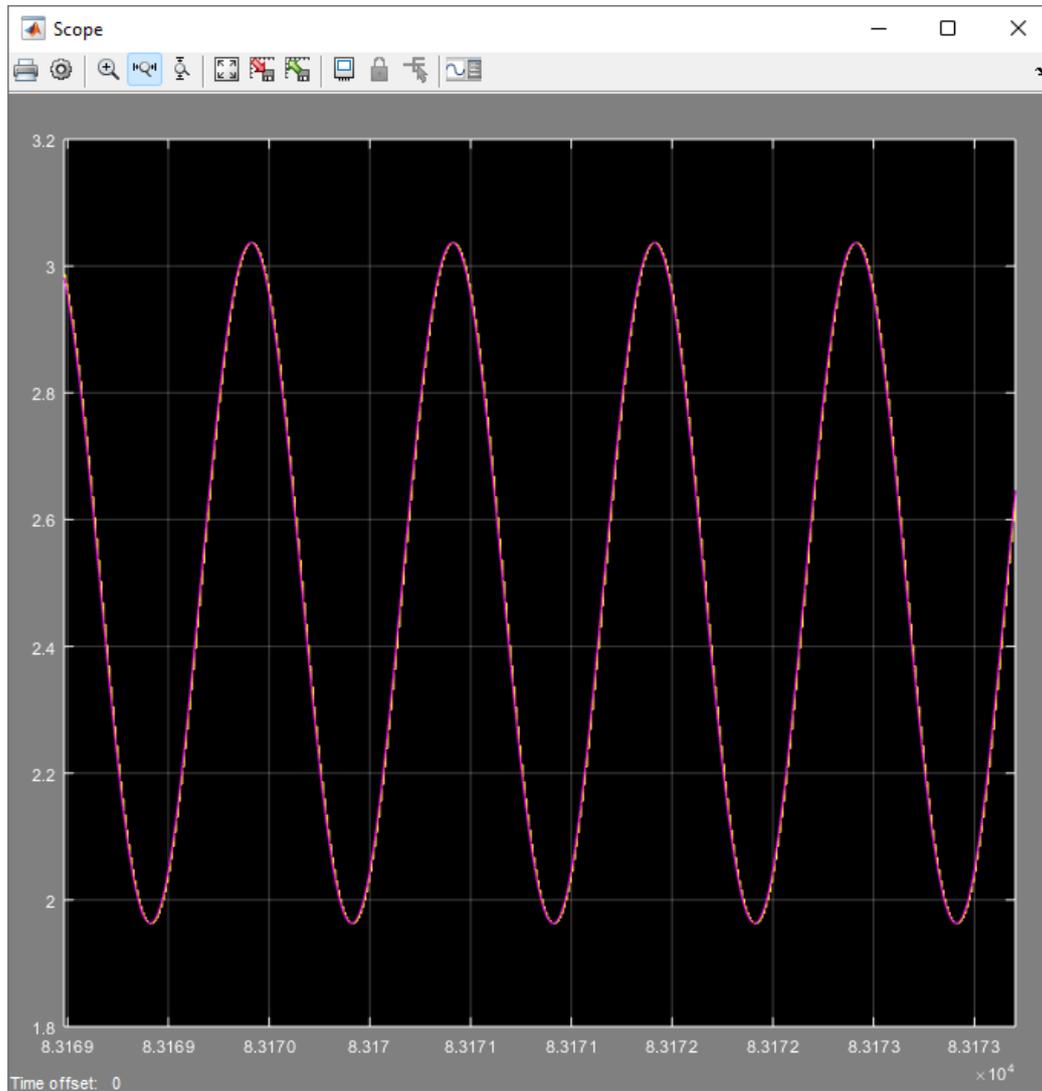


Fuente: Software Matlab - Simulink

En la figura 12, indica que la señal tarda en estabilizarse aproximadamente 60000s.

Figura 12

Simulación de la Respuesta Estacionaria con 10 coeficientes y convergencia 0.001

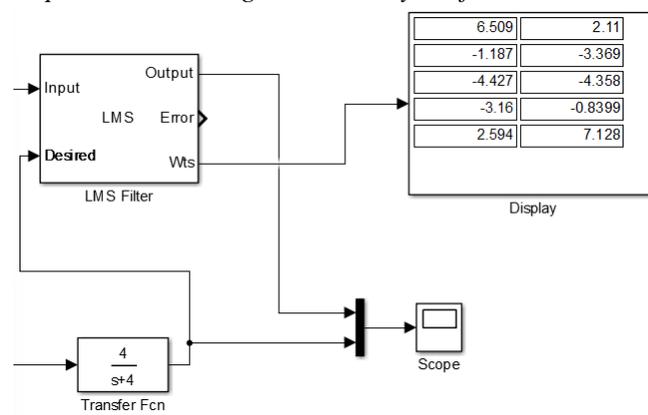


Fuente: Software Matlab - Simulink

La figura 13, indica la respuesta del sistema en simulink, para 10 coeficientes y una convergencia de 0.001.

Figura 13

Simulación del bloque LMS para una convergencia 0.001 y coeficiente 10



Fuente: Software Matlab – Simulink

La tabla 4, indica los valores para 10 coeficientes de la respuesta en estado estacionario para una planta de primer orden, haciendo uso del software de simulación Matlab -Simulink.

Tabla 4

Valores de los 10 coeficientes para la respuesta en estado estacionario

Número de coeficiente	Valor de coeficiente
1	6.509
2	-1.187
3	-4.427
4	-3.16
5	2.594
6	2.11
7	-3.369
8	-4.358
9	-0.8399
10	7.128

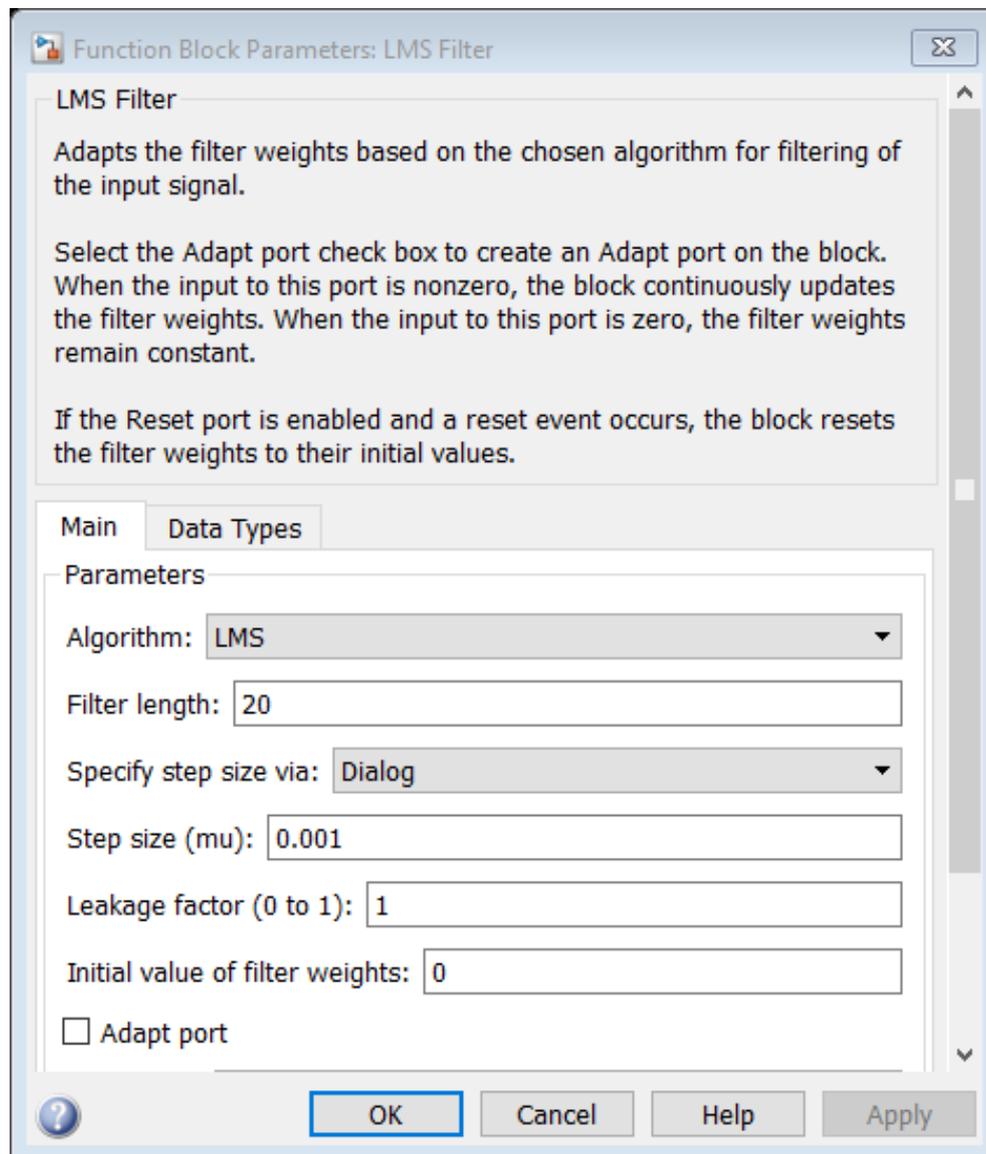
Fuente: Datos de la simulación en Matlab

Respuesta Estacionaria con 20 coeficientes y convergencia 0.001

La figura 14, indica la configuración de los parámetros del filtro LMS para 20 coeficientes y una convergencia de 0.001.

Figura 14

Parámetros LMS para 20 coeficientes y convergencia 0.001- Régimen estacionario

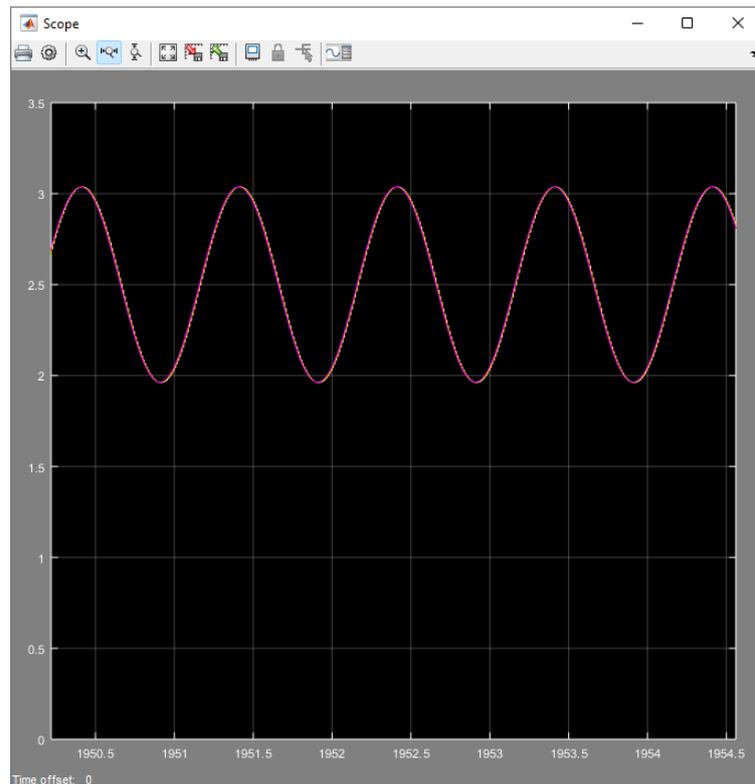


Fuente: Software Matlab - Simulink

La figura 15, indica que la señal tarda en estabilizarse aproximadamente 2000s.

Figura 15

Simulación de la Respuesta Estacionaria con 20 coeficientes y convergencia 0.001

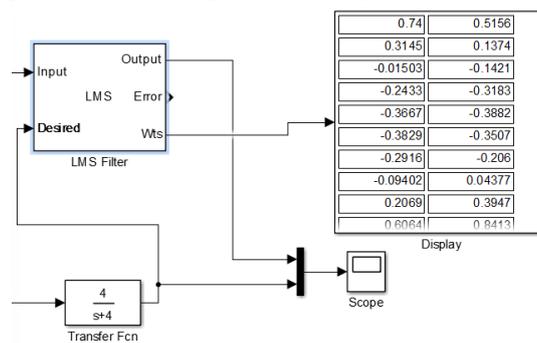


Fuente: Software Matlab - Simulink

La figura 16, indica la respuesta del sistema en simulink, para 20 coeficientes y una convergencia de 0.001.

Figura 16

Simulación del bloque LMS para una convergencia 0.001 y coeficiente 20



Fuente: Software Matlab – Simulink

La tabla 5, indica los valores para 20 coeficientes de la respuesta en estado estacionario para una planta de primer orden, haciendo uso del software de simulación Matlab -Simulink.

Tabla 5*Valores de los 20 coeficientes para la respuesta en estado estacionario*

Número de coeficiente	Valor de coeficiente
1	0.74
2	0.3145
3	-0.01503
4	-0.2433
5	-0.3667
6	-0.3829
7	-0.2916
8	-0.09402
9	0.2069
10	0.6064
11	0.5156
12	0.1374
13	-0.1421
14	-0.3183
15	-0.3882
16	-0.3507
17	-0.206
18	0.04377
19	0.3947
20	0.8413

Fuente: Datos de la simulación en Matlab

Respuesta transitoria.

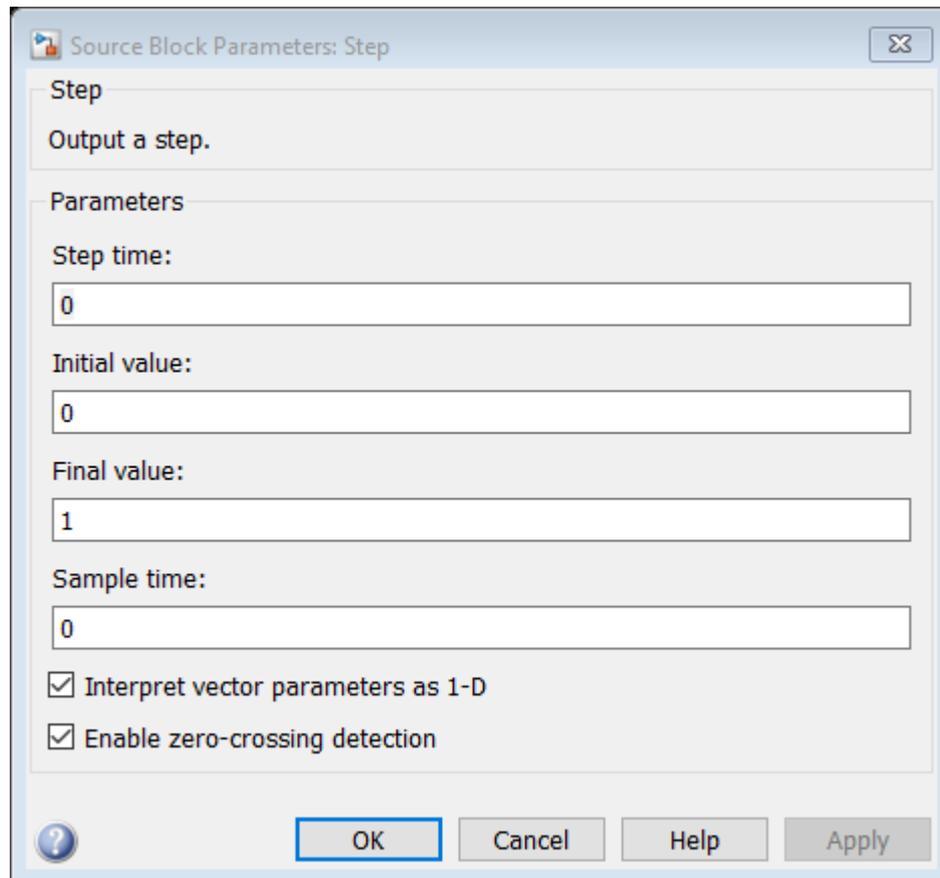
Para una planta de primer orden con polo en $s = -4$ y una excitación del sistema con la función escalón unitario

$$F(s) = \frac{4}{s + 4}$$

La figura 17, indica los parámetros de la señal escalón unitario:

Figura 17

Configuración de parámetros de la función escalón



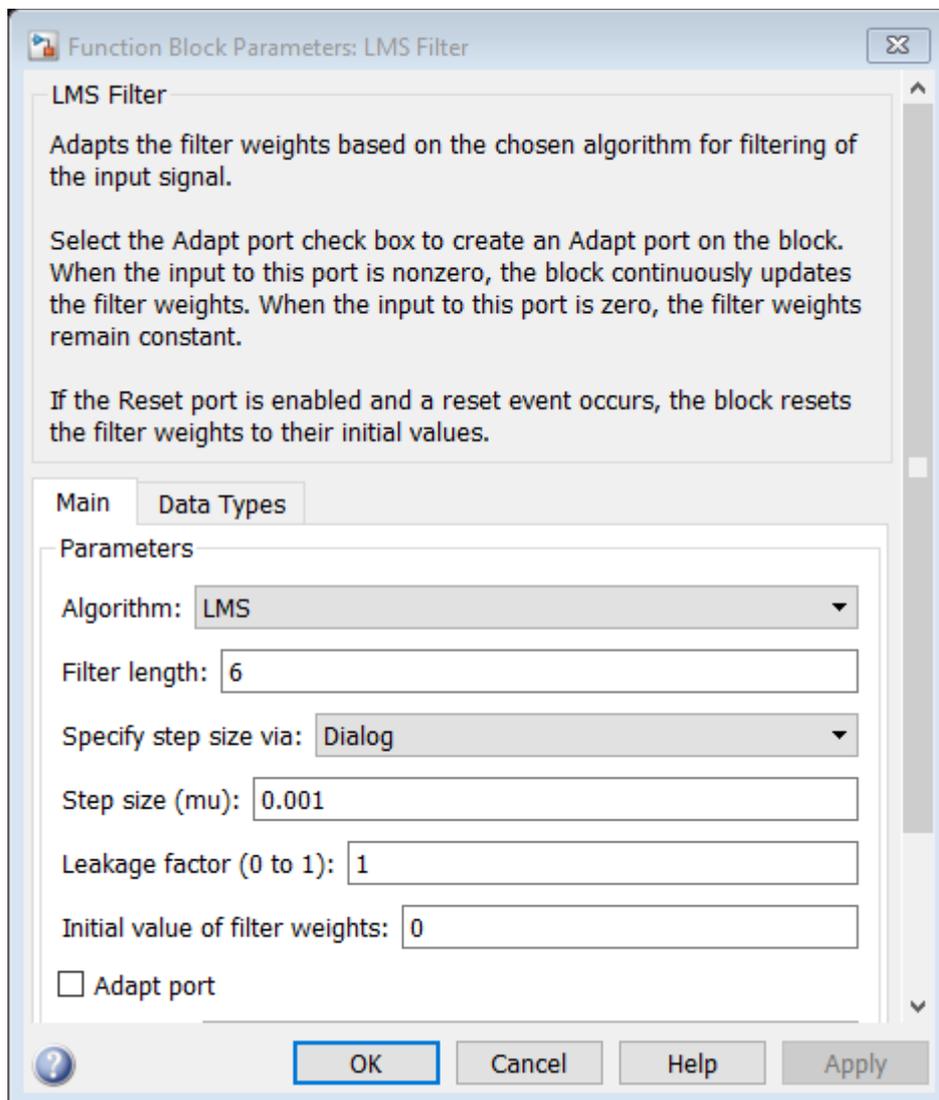
Fuente: Software Matlab - Simulink

Respuesta Transitoria con 6 coeficientes y convergencia 0.001

La figura 18, indica la configuración de los parámetros del filtro LMS para 6 coeficientes y una convergencia de 0.001 en régimen transitorio.

Figura 18

Parámetros del bloque LMS para 6 coeficientes y convergencia 0.001- Régimen transitorio

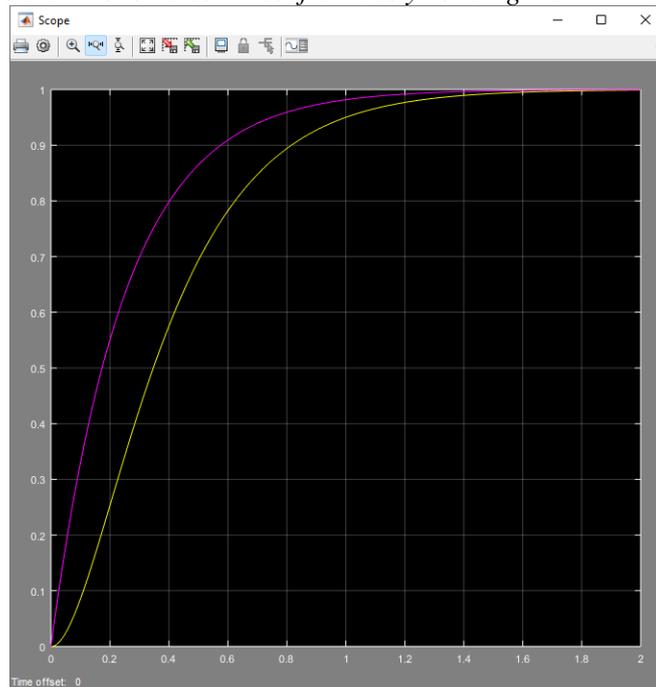


Fuente: Software Matlab - Simulink

En la figura 19, indica que la respuesta transitoria está desfasada 0.15s

Figura 19

Simulación de la Respuesta Transitoria con 6 coeficientes y convergencia 0.001

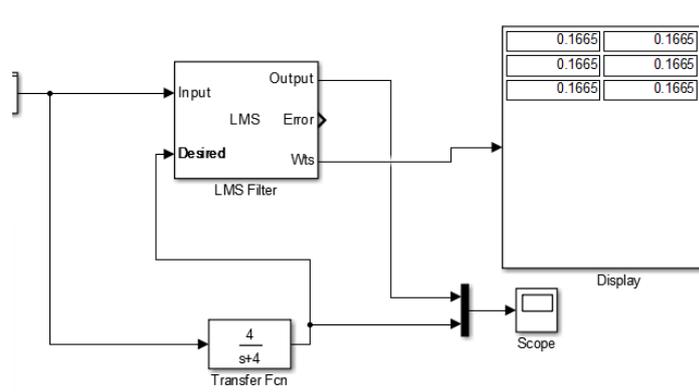


Fuente: Software Matlab - Simulink

La figura 20, indica la respuesta del sistema en simulink de Matlab, para 6 coeficientes y una convergencia de 0.001.

Figura 20

Respuesta transitoria para una convergencia 0.001 y coeficiente 6



Fuente: Software Matlab – Simulink

La tabla 6, indica los valores para 6 coeficientes de la respuesta en estado transitorio para una planta de primer orden, haciendo uso del software de simulación Matlab -Simulink.

Tabla 6

Valores de los 6 coeficientes para la respuesta en estado transitorio

Número de coeficiente	Valor de coeficiente
1	0.1665
2	0.1665
3	0.1665
4	0.1665
5	0.1665
6	0.1665

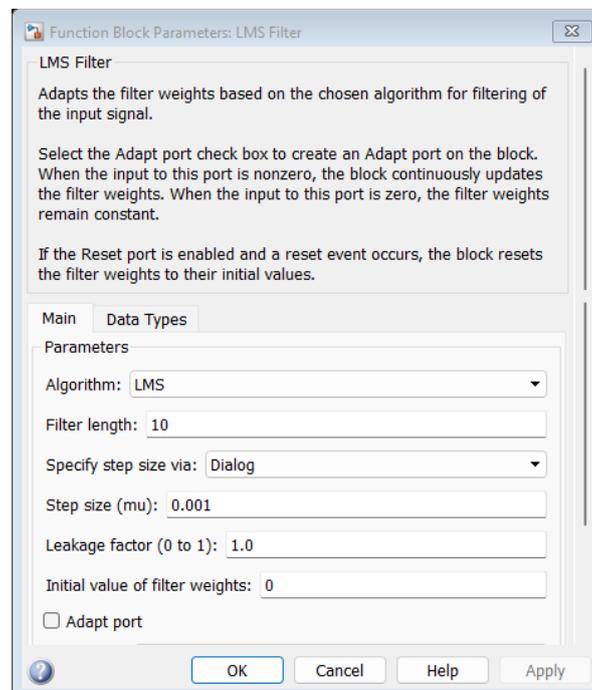
Fuente: Datos de la simulación en Matlab

Respuesta Transitoria con 10 coeficientes y convergencia 0.001

La figura 21, indica la configuración de los parámetros del filtro LMS para 10 coeficientes y una convergencia de 0.001 en régimen transitorio.

Figura 21

Parámetros del bloque LMS para 10 coeficientes y convergencia 0.001- Régimen transitorio

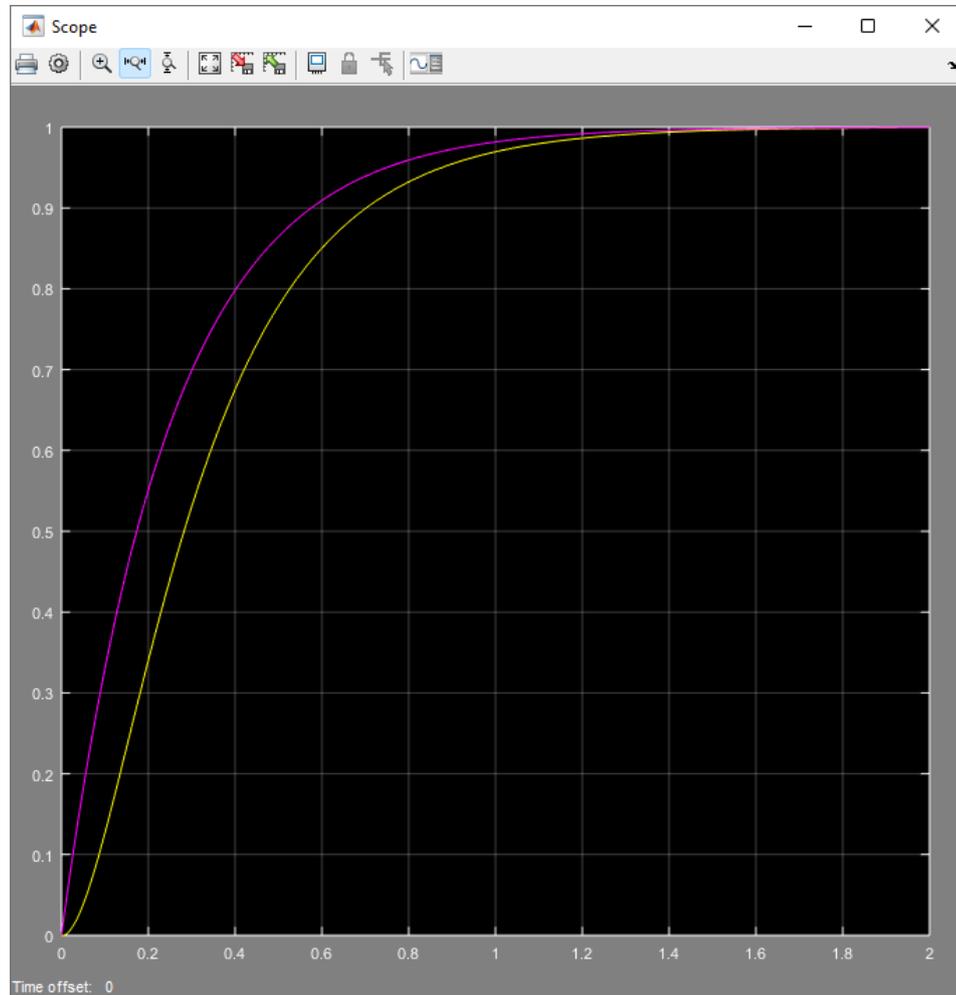


Fuente: Software Matlab - Simulink

En la figura 22, indica que la respuesta transitoria está desfasada 0.11s

Figura 22

Simulación de la Respuesta Transitoria con 10 coeficientes y convergencia 0.001

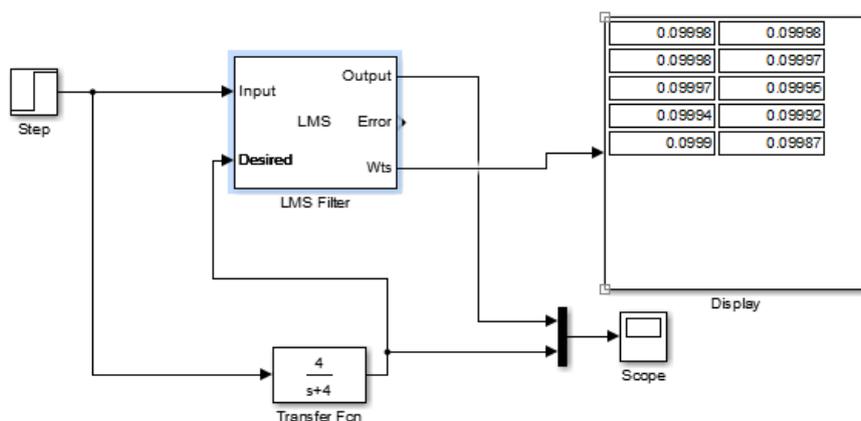


Fuente: Software Matlab - Simulink

La figura 23, indica la respuesta del sistema en simulink de Matlab, para 6 coeficientes y una convergencia de 0.001.

Figura 23

Respuesta transitoria para una convergencia 0.001 y coeficiente 10



Fuente: Software Matlab – Simulink

La tabla 7, indica los valores para 10 coeficientes de la respuesta en estado transitorio para una planta de primer orden, haciendo uso del software de simulación Matlab -Simulink.

Tabla 7

Valores de los 10 coeficientes para la respuesta en estado transitorio

Número de coeficiente	Valor de coeficiente
1	0.09998
2	0.09998
3	0.09997
4	0.09994
5	0.0999
6	0.09998
7	0.09997
8	0.09995
9	0.09992
10	0.09987

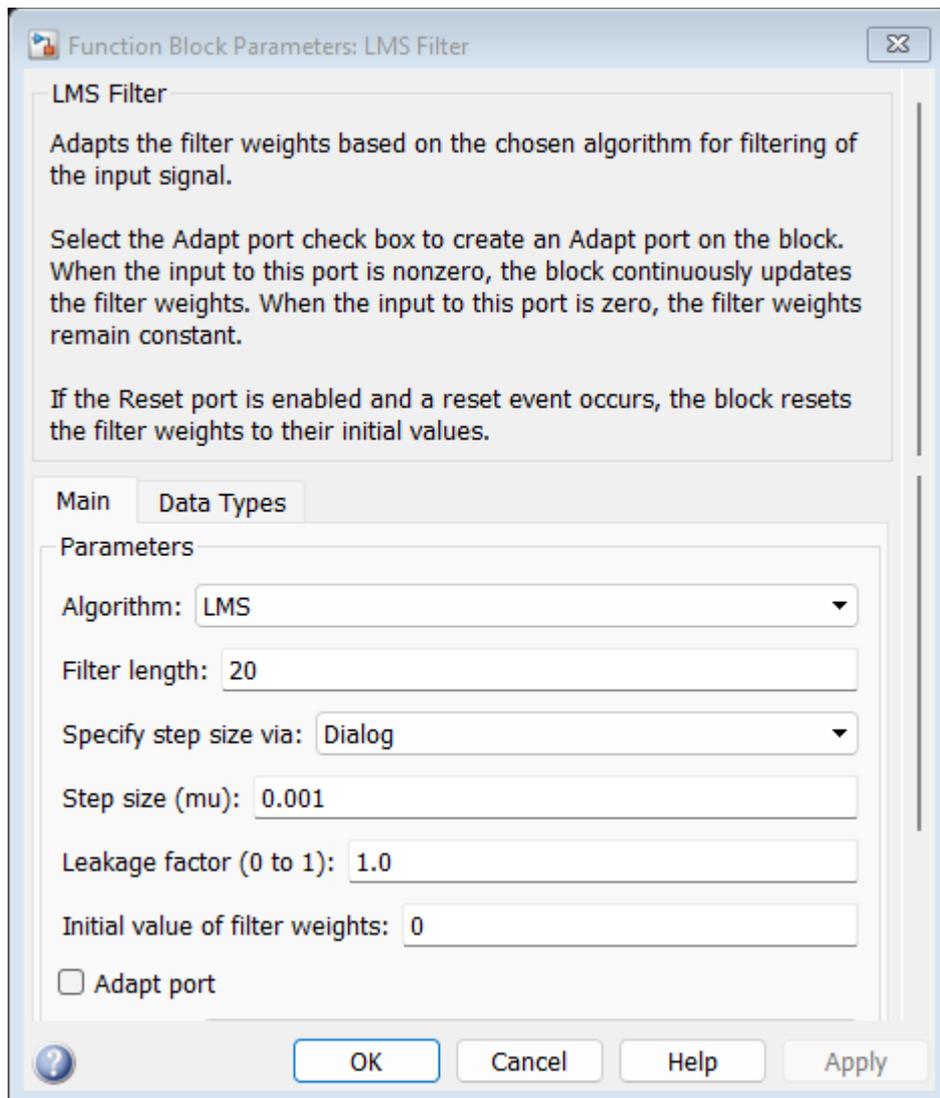
Fuente: Datos de la simulación en Matlab

Respuesta Transitoria con 20 coeficientes y convergencia 0.001

La figura 24, indica la configuración de los parámetros del filtro LMS para 20 coeficientes y una convergencia de 0.001 en régimen transitorio.

Figura 24

Parámetros del bloque LMS para 20 coeficientes y convergencia 0.001- Régimen transitorio

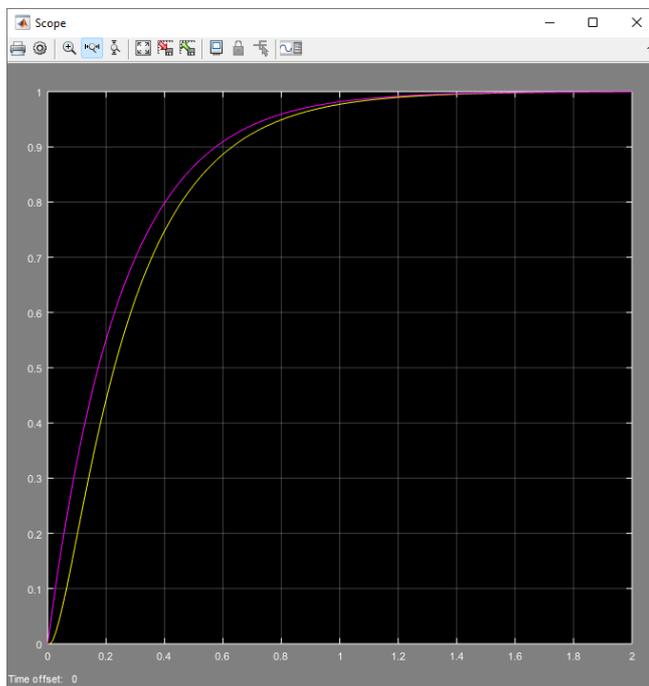


Fuente: Software Matlab - Simulink

En la figura 25, indica que la respuesta transitoria está desfasada 0.056s

Figura 25

Simulación de la Respuesta Transitoria con 20 coeficientes y convergencia 0.001

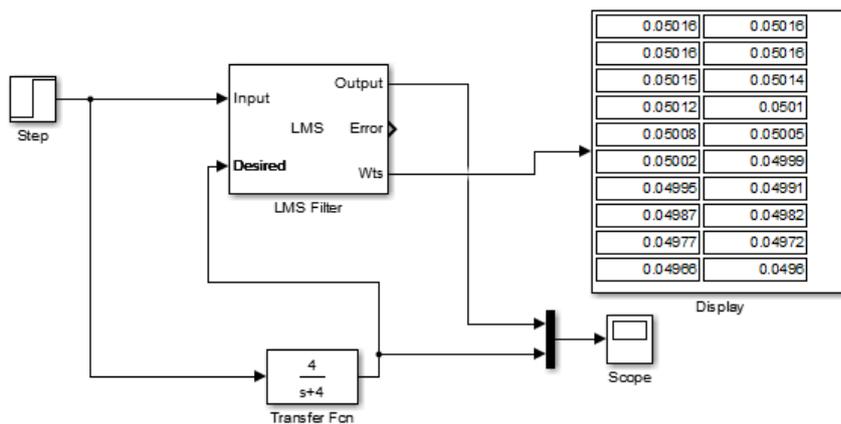


Fuente: Software Matlab - Simulink

La figura 26, indica la respuesta del sistema en simulink de Matlab, para 20 coeficientes y una convergencia de 0.001.

Figura 26

Respuesta transitoria para una convergencia 0.001 y coeficiente 20



Fuente: Software Matlab – Simulink

La tabla 8, indica los valores para 20 coeficientes de la respuesta en estado transitorio para una planta de primer orden, haciendo uso del software de simulación Matlab -Simulink.

Tabla 8

Valores de los 20 coeficientes para la respuesta en estado transitorio

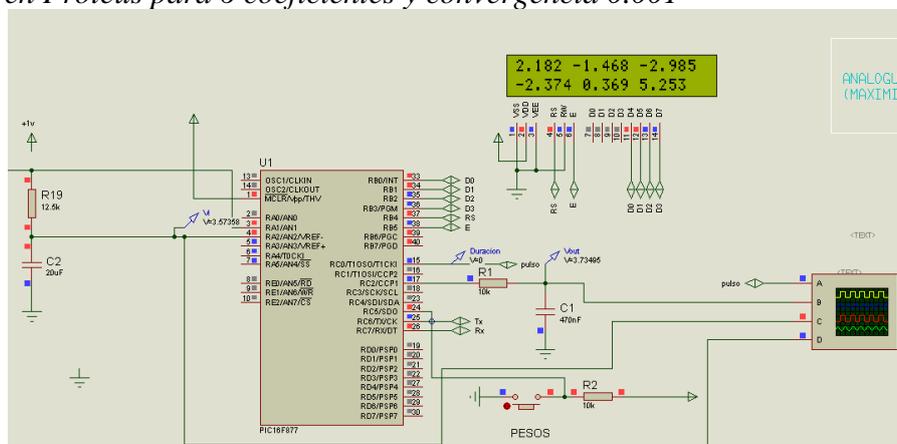
Número de coeficiente	Valor de coeficiente
1	0.05016
2	0.05016
3	0.05015
4	0.05012
5	0.05008
6	0.05002
7	0.04995
8	0.04987
9	0.04977
10	0.04966
11	0.05016
12	0.05016
13	0.05014
14	0.0501
15	0.05005
16	0.04999
17	0.04991
18	0.04982
19	0.04972
20	0.0496

Fuente: Datos de la simulación en Matlab

Respuesta estacionaria usando 6 coeficientes y convergencia a 0.001

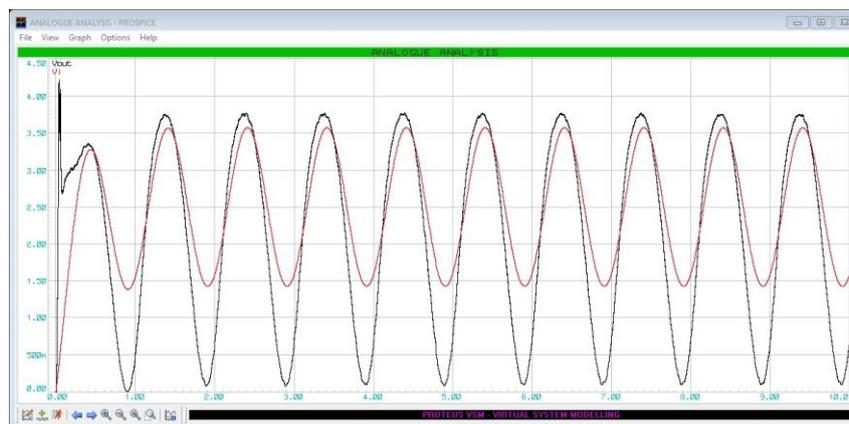
M0	0x0056	2.187
M1	0x005A	-1.476
M2	0x005E	-2.995
M3	0x0062	-2.366
M4	0x0066	0.4101
M5	0x006A	5.322

Figura 29
Simulación en Proteus para 6 coeficientes y convergencia 0.001



Fuente: Software de simulación Proteus

Figura 30
Simulación en PROSPICE para 6 coeficientes y convergencia 0.001



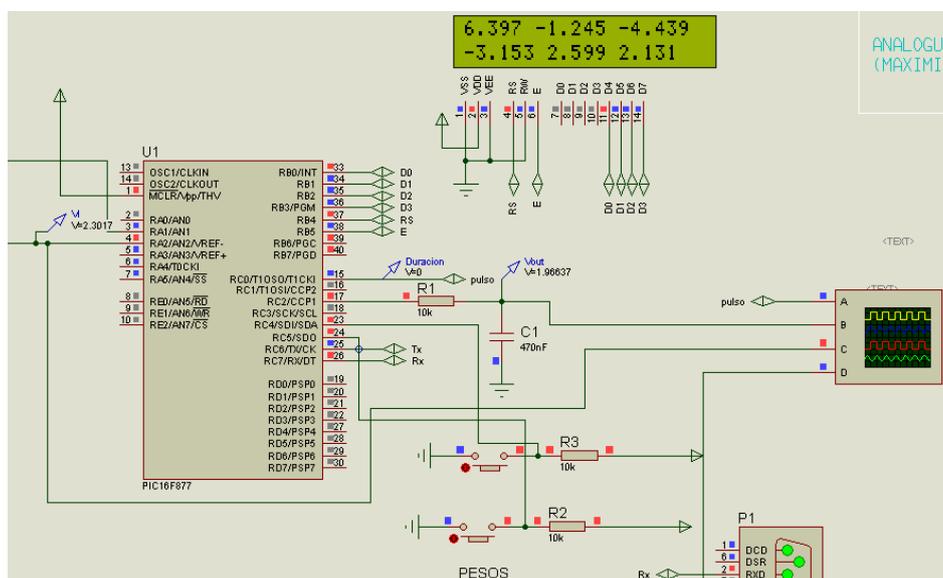
Fuente: Software de simulación Proteus

Respuesta estacionaria usando 10 coeficientes y convergencia a 0.001

M0	0x0056	6.509
M1	0x005A	-1.187
M2	0x005E	-4.427
M3	0x0062	-3.16
M4	0x0066	2.594
M5	0x006A	2.11
M6	0x006E	-3.369
M7	0x0072	-4.358
M8	0x0076	-0.8399
M9	0x007A	7.128

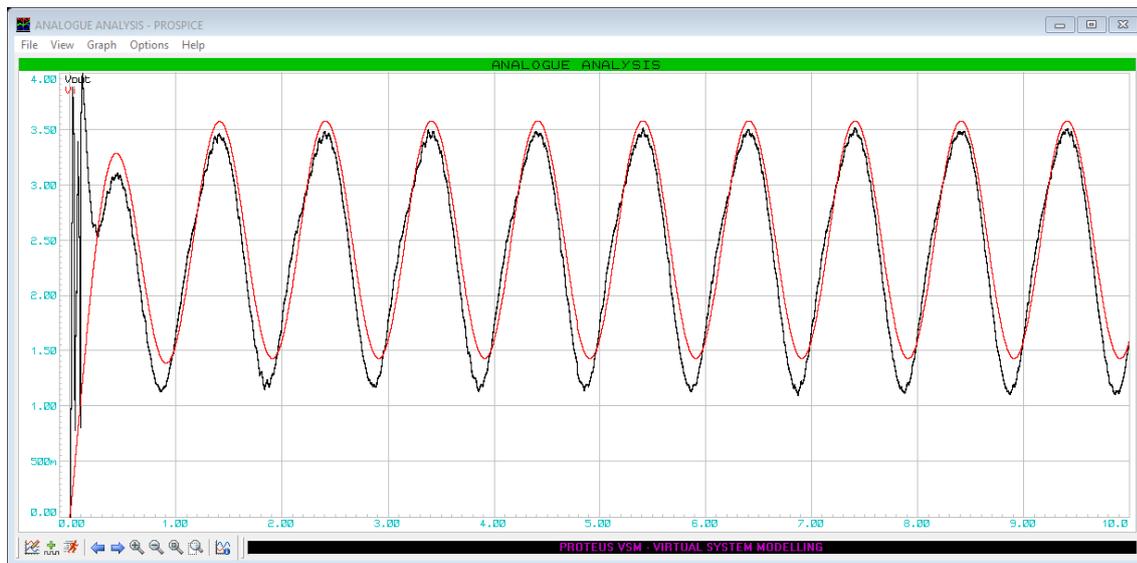
Figura 31

Simulación en Proteus para 10 coeficientes y convergencia 0.001



Fuente: Software de simulación Proteus

Figura 32
Simulación en PROSPICE para 10 coeficientes y convergencia 0.001



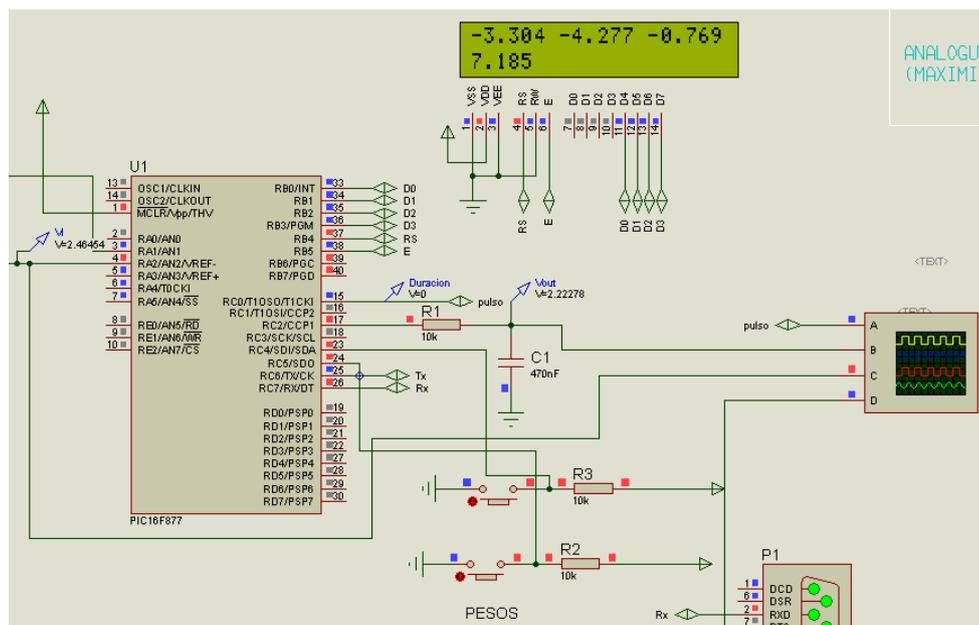
Fuente: Software de simulación Proteus

Respuesta estacionaria usando 20 coeficientes y convergencia a 0.001

M0	0x0056	0.74
M1	0x005A	0.3145
M2	0x005E	-0.01503
M3	0x0062	-0.2433
M4	0x0066	-0.3667
M5	0x006A	-0.3829
M6	0x006E	-0.2916
M7	0x0072	-0.09402
M8	0x0076	0.2069
M9	0x007A	0.6064
M10	0x0056	0.5156
M11	0x005A	0.1374

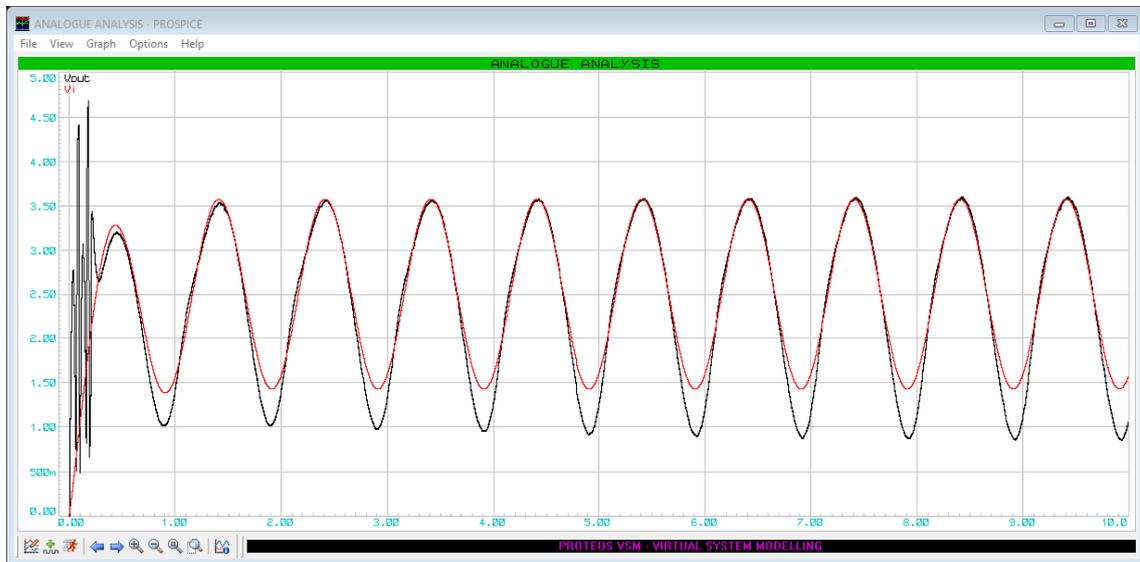
M12	0x005E	-0.1421
M13	0x0062	-0.3183
M14	0x0066	-0.3882
M15	0x006A	-0.3507
M16	0x006E	-0.206
M17	0x0072	0.04377
M18	0x0076	0.3947
M19	0x007A	0.8413

Figura 33
 Simulación en Proteus para 20 coeficientes y convergencia 0.001



Fuente: Software de simulación Proteus

Figura 34
Simulación en PROSPICE para 20 coeficientes y convergencia 0.001

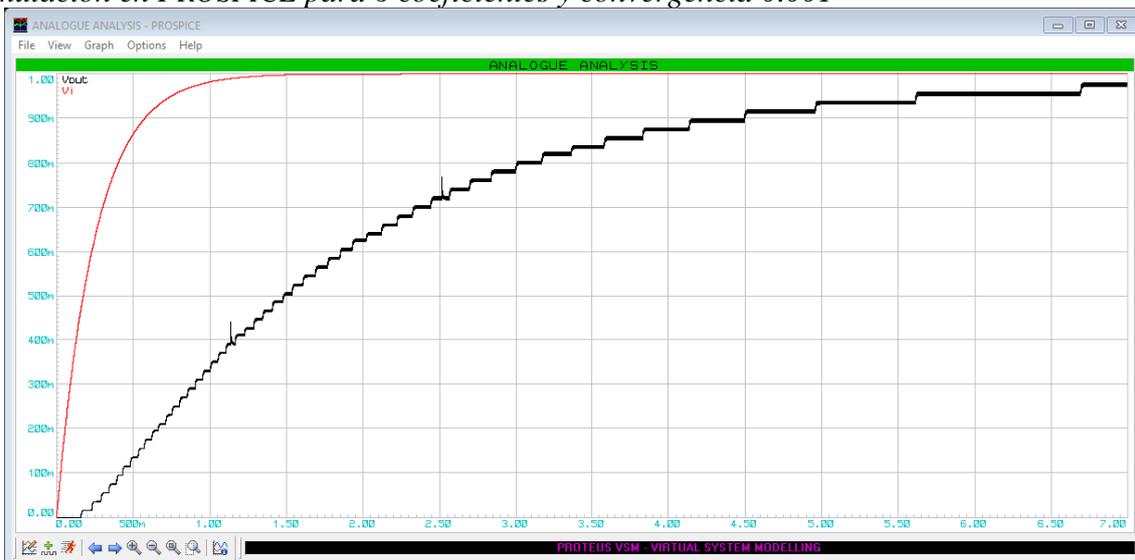


Fuente: Software de simulación Proteus

Respuesta transitoria

Respuesta transitoria usando 6 coeficientes y convergencia a 0.001

Figura 35
Simulación en PROSPICE para 6 coeficientes y convergencia 0.001

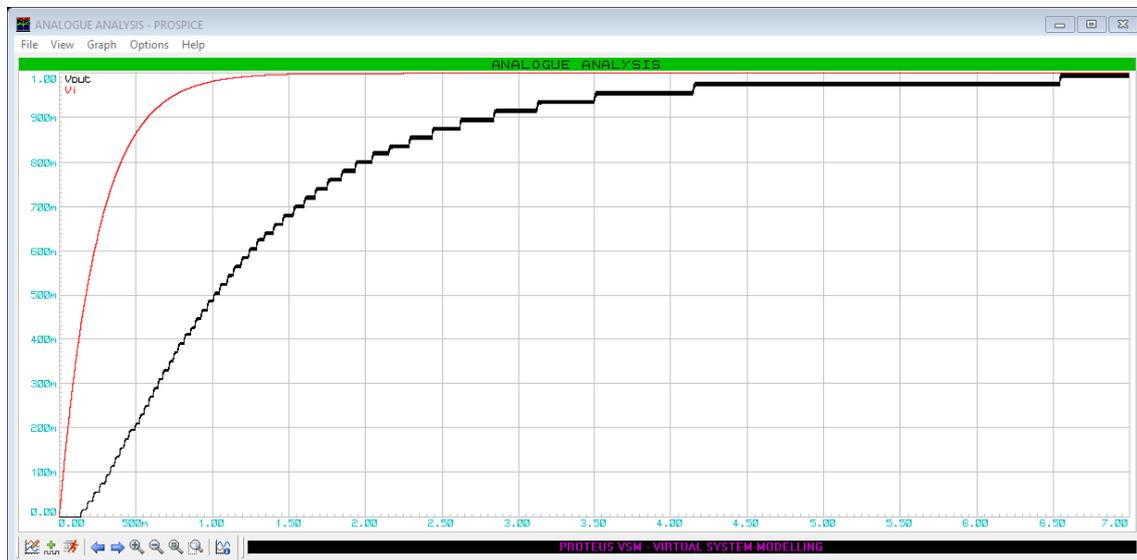


Fuente: Software de simulación Proteus

Respuesta transitoria usando 10 coeficientes y convergencia a 0.001

Figura 36

Simulación en PROSPICE para 10 coeficientes y convergencia 0.001

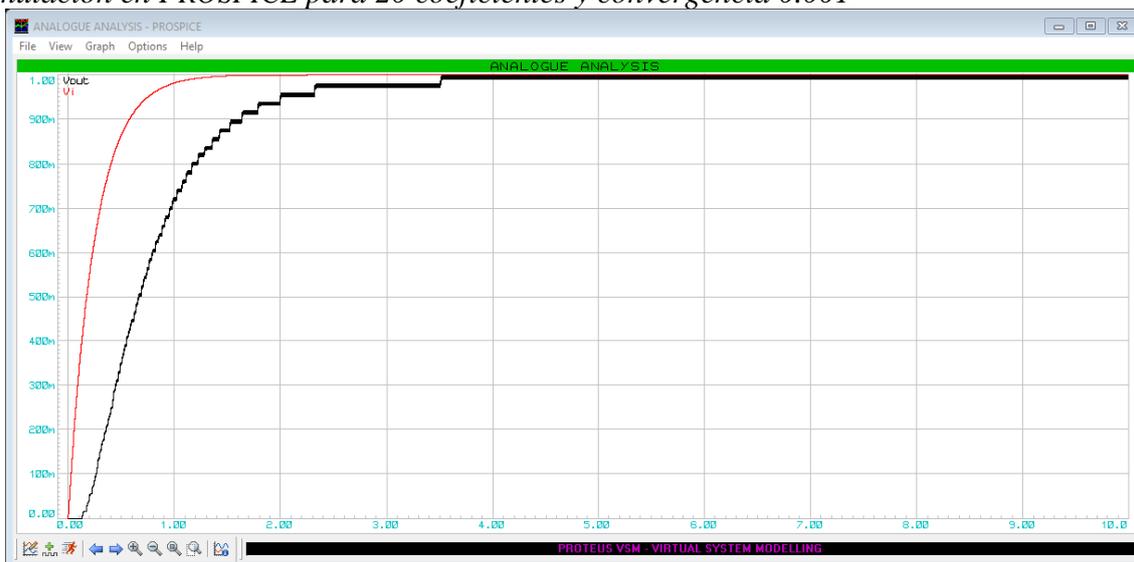


Fuente: Software de simulación Proteus

Respuesta transitoria usando 20 coeficientes y convergencia a 0.001

Figura 37

Simulación en PROSPICE para 20 coeficientes y convergencia 0.001



Fuente: Software de simulación Proteus

CAPÍTULO IV: RESULTADOS

4.1. Análisis de los resultados

La tabla 9, representa la respuesta en régimen estacionario para la planta de primer orden utilizando 6 coeficientes.

Tabla 9

Respuesta en régimen estacionario con 6 coeficientes

U	0.001
M0	2.187
M1	-1.476
M2	-2.995
M3	-2.366
M4	0.4101
M5	5.322

La tabla 10, representa la respuesta en régimen estacionario para la planta de primer orden utilizando 10 coeficientes.

Tabla 10

Respuesta en régimen estacionario con 10 coeficientes

U	0.001
M0	6.509
M1	-1.187
M2	-4.427
M3	-3.16
M4	2.594
M5	2.11
M6	-3.369
M7	-4.358
M8	-0.8399
M9	7.128

La tabla 11, representa la respuesta en régimen estacionario para la planta de primer orden utilizando 20 coeficientes.

Tabla 11

Respuesta en régimen estacionario con 20 coeficientes

U	0.001
M0	0.74
M1	0.3145
M2	-0.01503
M3	-0.2433
M4	-0.3667
M5	-0.3829
M6	-0.2916
M7	-0.09402
M8	0.2069
M9	0.6064
M10	0.5156
M11	0.1374
M12	-0.1421
M13	-0.3183
M14	-0.3882
M15	-0.3507
M16	-0.206
M17	0.04377
M18	0.3947
M19	0.8413

La tabla 12, representa la respuesta en régimen estacionario para la planta de primer orden utilizando 20 coeficientes.

Tabla 12
Respuesta en régimen estacionario con 20 coeficientes

U	0.001
M0	0.74
M1	0.3145
M2	-0.015063
M3	-0.2433
M4	0.3667
M5	-0.3829
M6	-0.2916
M7	-0.09402
M8	0.2069
M9	0.6064
M10	0.5156
M11	0.1374
M12	-0.1421
M13	-0.3183
M14	-0.3882
M15	-0.3507
M16	-0.206
M17	0.04377
M18	0.3947
M19	0.8413

La tabla 13, representa la respuesta en régimen estacionario haciendo uso de software Matlab para la planta de primer orden utilizando 6,10 y 20 coeficientes.

Tabla 13

Respuesta en régimen estacionario en Matlab para planta de primer orden con 6, 10 y 20 coeficientes

Respuesta estacionaria en Matlab		
Coeficientes	Número de convergencia	Tiempo de establecimiento
6	0.001	20000s
10	0.001	60000s
20	0.001	2000s

La tabla 14, representa la respuesta en régimen transitorio haciendo uso de software Matlab para la planta de primer orden utilizando 6,10 y 20 coeficientes.

Tabla 14

Respuesta en régimen transitorio en Matlab para planta de primer orden con 6,10 y 20 coeficientes

Respuesta transitoria en MATLAB		
Coeficientes	Factor de convergencia	Tiempo de Desfasaje
6	0.001	0.156s
10	0.001	0.115s
20	0.001	0.056s

La tabla 15, representa la respuesta en régimen estacionario haciendo uso del microcontrolador PIC16F877 para la planta de primer orden utilizando 6, 10 y 20 coeficientes.

Tabla 15

Respuesta estacionaria del Microcontrolador para planta de primer orden con 6,10 y 20 coeficientes

Respuesta estacionaria en el Microcontrolador		
Coeficientes	Factor de convergencia	Tiempo de Establecimiento
6	0.001	20000s
10	0.001	60000s
20	0.001	2000s

La tabla 16, representa la respuesta en régimen transitorio haciendo uso del microcontrolador PIC16F877 para la planta de primer orden utilizando 6, 10 y 20 coeficientes.

Tabla 16

Respuesta transitoria del Microcontrolador para planta de primer orden con 6, 10 y 20 coeficientes

Respuesta transitoria en el Microcontrolador		
Coeficientes	Factor de convergencia	Tiempo de Establecimiento
6	0.001	0.156s
10	0.001	0.115s
20	0.001	0.056s

La figura 38, representa la evolución de la respuesta en régimen estacionario haciendo uso del microcontrolador PIC16F877 para la planta de primer orden utilizando 6 coeficientes.

Figura 38

Evolución de la respuesta estacionaria para 6 coeficientes



La figura 39, representa la evolución de la respuesta en régimen estacionario haciendo uso del microcontrolador PIC16F877 para la planta de primer orden utilizando 10 coeficientes.

Figura 39

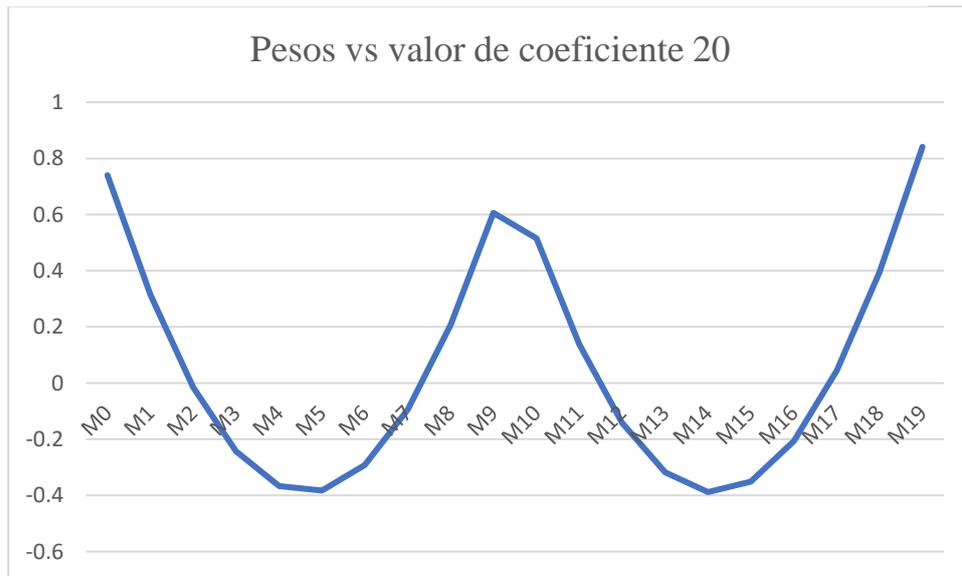
Evolución de la respuesta estacionaria para 10 coeficientes



La figura 40, representa la evolución de la respuesta en régimen estacionario haciendo uso del microcontrolador PIC16F877 para la planta de primer orden utilizando 20 coeficientes.

Figura 40

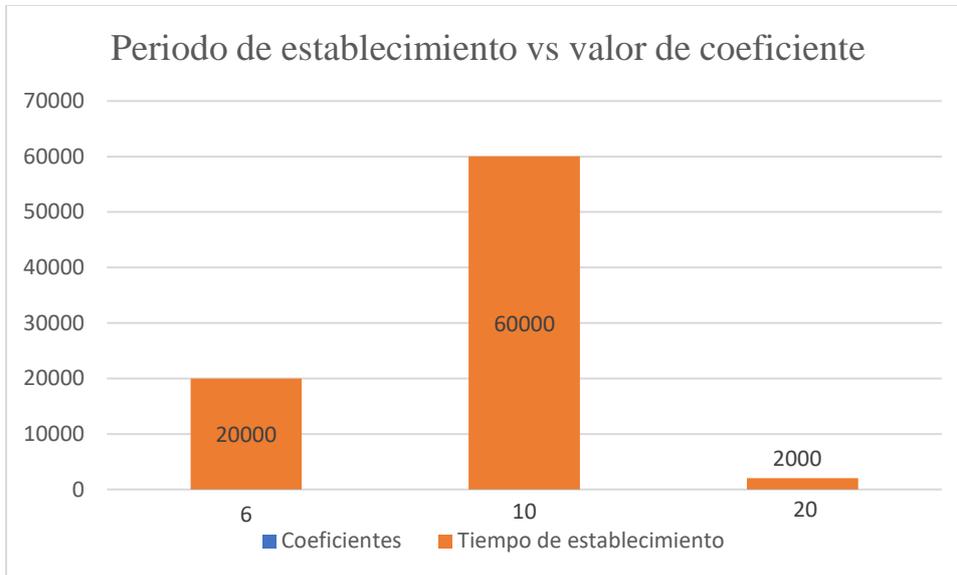
Evolución de la respuesta estacionaria para 20 coeficientes



La figura 41, representa la evolución del tiempo de establecimiento haciendo uso del software de simulación MATLAB para la planta de primer orden utilizando 6,10 y 20 coeficientes

Figura 41

Evolución del periodo de establecimiento para 6,10 y 20 coeficientes



CAPÍTULO V: DISCUSIÓN

5.1. Discusión de resultados

Para (Jiménez & Mauricio, 2017), se realiza la comparación entre los algoritmos y se evalúa utilizando métricas de rendimiento como el tiempo de respuesta y la velocidad de convergencia. En la investigación el peso asignado al identificador de proceso mantiene el factor de convergencia " $U=0.001$ ". Esto se obtuvo procesando la Tabla 12 con un porcentaje máximo de error de 1,3%.

Según (Gastañadui & Sánchez, 2020), el algoritmo LMS se implementa en un PLC con una frecuencia de muestreo de 12 ms, utilizando el lenguaje SCL, el tiempo de adaptación del filtro es inversamente proporcional al factor de convergencia. En la investigación el tiempo de construcción del modelo para una respuesta estacionaria es una función inversa del factor de convergencia " U " para 6, 10 y 20 coeficientes, con los mejores resultados para 20 coeficientes. La Tabla 12 lo demuestra. La suave respuesta del microcontrolador no es adecuada para el cálculo en Matlab para 6 coeficientes.

En la respuesta transitoria, el sistema se comporta igual en Matlab que con un microcontrolador PIC16F877, con un factor de 10 para mejores resultados. Se obtiene un desfase de 11ms que es el tiempo de muestreo del sistema.

CAPÍTULO VI: CONCLUSIONES Y RECOMENDACIONES

6.1. Conclusiones

Se diseñó el identificador de procesos, haciendo uso del algoritmo LMS aplicado a una planta de primer orden.

La elaboración del algoritmo de identificación de procesos, se muestra en el microcontrolador de 8 bits 16f877, se implementó a través del software Proton Basic para 6, 10 y 20 coeficientes, lo que da como resultado un tiempo de procesamiento del algoritmo de 2,21 ms.

Se determinó el modelo del algoritmo LMS a través de la planta de primer orden para 6, 10 y 20 coeficientes en régimen estacionario como en régimen transitorio.

Se verificó que el sistema de 8 bits reconoció el proceso al comparar los datos obtenidos mediante el microcontrolador con los datos obtenidos en Matlab.

6.2. Recomendaciones

Realizar pruebas de tiempo de respuesta para diferentes valores de convergencia a través del software de simulación MATLAB y PROTEUS.

Realizar pruebas de tiempo de respuesta para una planta de segundo orden y diferente valor de convergencia a través del software de simulación MATLAB y PROTEUS.

CAPÍTULO VII: REFERENCIAS

7.1. Fuentes bibliográficas

- Colchado, D., & Solano, R. (2019). *Diseño de un controlador difuso basado en Microcontrolador usando el método del centroide*. Trujillo.
- Creus, A. (2010). *Instrumentación Industrial*. Mexico: Alfaomega.
- Cuevas, D. (2020). *Algoritmos de Filtrado Kernel Adaptativo Multi-Output*. Cantabria.
- Del Angel, J., Maya, X., Avalos, J., Trejo, C., & Sánchez, G. (2022). Estructura alternante para sistemas de beamforming adaptativo basada en los algoritmos APL/SR-LMS. *Elektron*, 20-26.
- Gastañadui, J., & Sánchez, Y. (2020). *Implementación de un algoritmo LMS en PLC para la atenuación de ruido eléctrico en señales analógicas*. Trujillo.
- Hernández, E., Chacón, J., & Navarro, J. (2019). Implementación de un sistema generador y eliminador de ECO con filtro adaptativo usando el algoritmo LMS en un FPGA ARTIX-7. *Pistas Educativas*, 253-270.
- Jiménez, F., & Mauricio, J. (2017). Algoritmos LMS de filtrado adaptativo para cancelación de Eco Acústico en sistemas de Telecomunicaciones. *Revista Colombiana de Tecnologías de Avanzada*.
- Martell, N. (2018). *Diseño e implementación de sistema electrónico para el monitoreo de la temperatura en neonatos*. Lima.
- Ogata, K. (1996). *Sistemas de control en tiempo discreto*. Mexico: Prentice Hall.
- Ramirez, G. (2022). *Análisis del algoritmo Fista orientado a mejorar la velocidad de convergencia*. Lima.
- Torres, J. (2019). *Diseño e implementación de un sistema de control electrónico para filtros hepa en una cabina de bioseguridad clase II para el laboratorio Bermanlab de Trujillo*. Lima.
- Zapata, P. (2020). *Estudio comparativo de los algoritmos de filtrado adaptativo (LMS, NLMS, RLS) para la eliminación del ruido audible externo en tiempo real*. Quito.

ANEXOS

Anexo 1: Características del PIC 16F87X



PIC16F87X

28/40-Pin 8-Bit CMOS FLASH Microcontrollers

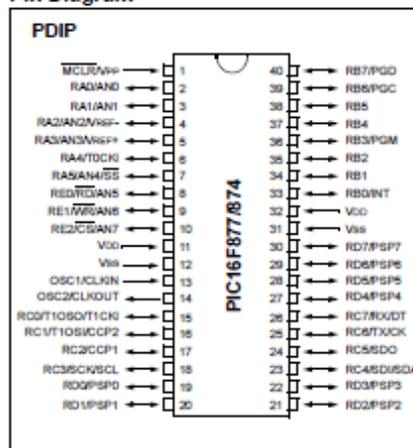
Devices Included in this Data Sheet:

- PIC16F873
- PIC16F876
- PIC16F874
- PIC16F877

Microcontroller Core Features:

- High performance RISC CPU
- Only 35 single word instructions to learn
- All single cycle instructions except for program branches which are two cycle
- Operating speed: DC - 20 MHz clock input
DC - 200 ns instruction cycle
- Up to 8K x 14 words of FLASH Program Memory,
Up to 368 x 8 bytes of Data Memory (RAM)
Up to 256 x 8 bytes of EEPROM Data Memory
- Pinout compatible to the PIC16C73B/74B/76/77
- Interrupt capability (up to 14 sources)
- Eight level deep hardware stack
- Direct, indirect and relative addressing modes
- Power-on Reset (POR)
- Power-up Timer (PWRT) and
Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC
oscillator for reliable operation
- Programmable code protection
- Power saving SLEEP mode
- Selectable oscillator options
- Low power, high speed CMOS FLASH/EEPROM
technology
- Fully static design
- In-Circuit Serial Programming™ (ICSP) via two
pins
- Single 5V In-Circuit Serial Programming capability
- In-Circuit Debugging via two pins
- Processor read/write access to program memory
- Wide operating voltage range: 2.0V to 5.5V
- High Sink/Source Current: 25 mA
- Commercial, Industrial and Extended temperature
ranges
- Low-power consumption:
 - < 0.6 mA typical @ 3V, 4 MHz
 - 20 μ A typical @ 3V, 32 kHz
 - < 1 μ A typical standby current

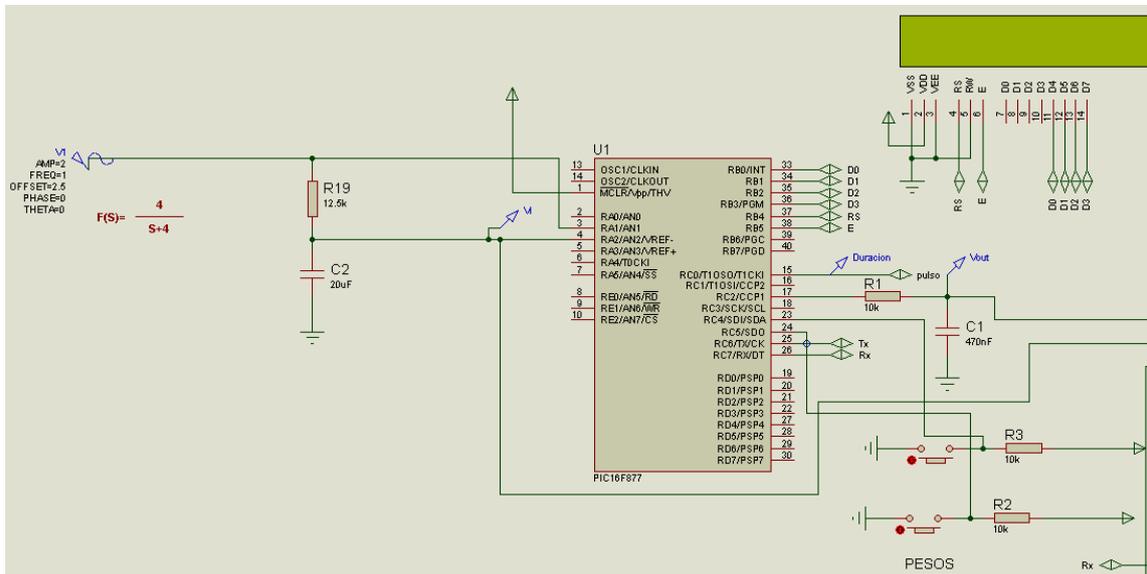
Pin Diagram



Peripheral Features:

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler,
can be incremented during SLEEP via external
crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period
register, prescaler and postscale
- Two Capture, Compare, PWM modules
 - Capture is 16-bit, max. resolution is 12.5 ns
 - Compare is 16-bit, max. resolution is 200 ns
 - PWM max. resolution is 10-bit
- 10-bit multi-channel Analog-to-Digital converter
- Synchronous Serial Port (SSP) with SPI (Master
mode) and I²C™ (Master/Slave)
- Universal Synchronous Asynchronous Receiver
Transmitter (USART/SCI) with 9-bit address
detection
- Parallel Slave Port (PSP) 8-bits wide, with
external RD, WR and CS controls (40/44-pin only)
- Brown-out detection circuitry for
Brown-out Reset (BOR)

Anexo 2: Diseño del circuito en Proteus.



Anexo 3: Matriz de Consistencia

DISEÑO DE UN IDENTIFICADOR DE PROCESOS USANDO PLATAFORMA DE 8 BITS MEDIANTE ALGORITMO LMS	PROBLEMA	OBJETIVO	HIPÓTESIS	VARIABLES Y DIMENSIONES	METODOLOGÍA
	<p>PROBLEMA GENERAL ¿Cómo el diseño del algoritmo LMS en los procesos industriales permitirá la identificación del proceso?</p> <p>PROBLEMAS ESPECÍFICOS</p> <p>¿Cómo el uso del lenguaje de alto nivel permitirá la implementación del algoritmo LMS?</p> <p>¿Cómo la identificación del proceso de primer orden permitirá determinar el modelo del algoritmo LMS?</p> <p>¿Cómo la función de transferencia permitirá comprobar el identificador de procesos en sistemas de primer orden?</p>	<p>OBJETIVO GENERAL Diseñar un identificador de procesos usando algoritmo LMS.</p> <p>OBJETIVOS ESPECÍFICOS</p> <p>Elaborar el algoritmo LMS en un microcontrolador, haciendo uso de la programación de alto nivel.</p> <p>Determinar el modelo del algoritmo LMS para la identificación del proceso de primer orden.</p> <p>Comprobar el identificador de procesos a través de la función de transferencia del sistema.</p>	<p>HIPÓTESIS GENERAL El diseño del algoritmo LMS en los procesos industriales permitirá la identificación del proceso.</p> <p>HIPOTESIS ESPECÍFICAS</p> <p>El uso del lenguaje de alto nivel permite la implementación del algoritmo LMS.</p> <p>La identificación del proceso de primer orden permite determinar el modelo del algoritmo LMS.</p> <p>La función de transferencia permite verificar el identificador de procesos.</p>	<p>VARIABLE 1 Algoritmo LMS</p> <p>Número de coeficientes.</p> <p>Tiempo de convergencia.</p> <p>VARIABLE 2 Identificación proceso</p> <p>Respuesta estacionaria.</p> <p>Respuesta transitoria</p>	<p>Población: Bloque de transferencia de primer y segundo orden.</p> <p>Muestra: Función de transferencia de primer orden</p> <p>Instrumento:</p> <ul style="list-style-type: none"> • Reporte de simulación. • Informe