



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

**CARRERA DE INGENIERÍA EN ELECTRÓNICA E
INSTRUMENTACIÓN**

**PROYECTO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL
TÍTULO DE INGENIERO EN ELECTRÓNICA E INSTRUMENTACIÓN**

AUTORES: López Caisaguano, Verónica Gabriela

Paredes Anchatipán, Alex Darwin

**TEMA: DISEÑO E IMPLEMENTACIÓN DE UN ALGORITMO DE
CONTROL PREDICTIVO PARA UNA PLANTA DE FLUJO UTILIZANDO
UN CONTROLADOR DE AUTOMATIZACIÓN PROGRAMABLE PARA
EL LABORATORIO DE REDES INDUSTRIALES Y CONTROL DE
PROCESOS DE LA UNIVERSIDAD DE LAS FUERZAS ARMADAS –
ESPE EXTENSIÓN LATACUNGA**

DIRECTORA: Ing. Llanos, Jacqueline

CODIRECTOR: Ing. Ortiz, Diego

Latacunga, Febrero 2015

UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE
INGENIERIA EN ELECTRÓNICA E INSTRUMENTACIÓN

CERTIFICADO

Ing. Jacqueline, Llanos (DIRECTORA DE TESIS)

Ing. Diego, Ortiz (CODIRECTOR DE TESIS)

CERTIFICAN

Que el trabajo titulado “DISEÑO E IMPLEMENTACIÓN DE UN ALGORITMO DE CONTROL PREDICTIVO PARA UNA PLANTA DE FLUJO UTILIZANDO UN CONTROLADOR DE AUTOMATIZACIÓN PROGRAMABLE PARA EL LABORATORIO DE REDES INDUSTRIALES Y CONTROL DE PROCESOS DE LA UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE EXTENSIÓN LATACUNGA”, realizado por la Señorita López Caisaguano Verónica Gabriela y el Señor Paredes Anchatipán Alex Darwin, ha sido guiado y revisado periódicamente y cumple normas estatutarias establecidas por la Universidad de las Fuerzas Armadas ESPE.

Debido a que constituye un trabajo de excelente contenido científico y aplicable para el desarrollo profesional, SI recomiendan su aplicación.

El mencionado trabajo consta de un documento empastado y un disco compacto el cual contiene los archivos en formato portátil de Acrobat (pdf).

Autorizan a la Señorita López Caisaguano Verónica Gabriela y al Señor Paredes Anchatipán Alex Darwin que lo entregue al Ing. José Bucheli Andrade, en su calidad de Director de Carrera.

Latacunga, Febrero del 2015

Ing. Llanos, Jacqueline
DIRECTOR

Ing. Ortiz, Diego
CODIRECTOR

UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE
INGENIERIA EN ELECTRÓNICA E INSTRUMENTACIÓN

DECLARACIÓN DE RESPONSABILIDAD

Nosotros: López Caisaguano , Verónica Gabriela
 Paredes Anchatipán, Alex Darwin

DECLARAMOS QUE:

El proyecto de grado denominado “DISEÑO E IMPLEMENTACIÓN DE UN ALGORITMO DE CONTROL PREDICTIVO PARA UNA PLANTA DE FLUJO UTILIZANDO UN CONTROLADOR DE AUTOMATIZACIÓN PROGRAMABLE PARA EL LABORATORIO DE REDES INDUSTRIALES Y CONTROL DE PROCESOS DE LA UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE EXTENSIÓN LATACUNGA”, ha sido desarrollado en base a una investigación exhaustiva, respetando derechos intelectuales de terceros, conforme las citas que constan al pie de las páginas correspondientes, cuyas fuentes se incorporan en la bibliografía.

Consecuentemente este trabajo es de nuestra autoría.

En virtud de esta declaración, nos responsabilizamos del contenido, veracidad y alcance científico del proyecto de grado en mención.

Latacunga, Febrero del 2015.

López Caisaguano, Verónica Gabriela Paredes Anchatipán, Alex Darwin

C.I.: 150084984-7

C.I.:050361493-5

UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE
INGENIERIA EN ELECTRÓNICA E INSTRUMENTACIÓN

AUTORIZACIÓN

Nosotros: López Caisaguano , Verónica Gabriela
 Paredes Anchatipán , Alex Darwin

Autorizamos a la UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE, la publicación, en la biblioteca virtual de la Institución del Trabajo “DISEÑO E IMPLEMENTACIÓN DE UN ALGORITMO DE CONTROL PREDICTIVO PARA UNA PLANTA DE FLUJO UTILIZANDO UN CONTROLADOR DE AUTOMATIZACIÓN PROGRAMABLE PARA EL LABORATORIO DE REDES INDUSTRIALES Y CONTROL DE PROCESOS DE LA UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE EXTENSIÓN LATACUNGA”, cuyo contenido, ideas y criterios son de nuestra exclusiva responsabilidad y autoría.

Latacunga, Febrero del 2015

López Caisaguano, Verónica Gabriela Paredes Anchatipán, Alex Darwin

C.I.: 150084984-7

C.I.:050361493-5

DEDICATORIA

Verónica Gabriela López Caisaguano

En primer lugar a Dios, A mi madre Cecilia, por enseñarme a ser fuerte y perseverante y con palabras de aliento jamás me dejó decaer, a mi padre Domingo que siempre me apoyó a culminar esta etapa de la vida, a mis hermanos Lizeth, Alex y Paúl que estuvieron incondicionalmente, a mi sobrina Valentina que con su sonrisa me alegra el día. A Darío por haberme acompañado los últimos años de carrera universitaria y formar parte de mi vida.

Y a todas las personas que me apoyaron desinteresadamente a culminar mi carrera y seguir adelante en el camino de la vida.

Alex Darwin Paredes Anchatipán

Este proyecto va dedicado a mi familia, quienes me apoyaron en todo momento, contribuyendo incondicionalmente a lograr mis metas y objetivos. Espero que la dedicación y esfuerzo empleado en este proyecto y a lo largo de mi vida correspondan de alguna manera a lo que ellos me han brindado.

A todas las personas que me acompañaron en el transcurso de este camino con palabras de aliento y motivación, que de una u otra manera hicieron posible este logro.

AGRADECIMIENTO

Verónica Gabriela López Caisaguano

Agradezco a Dios por la fortaleza dada durante toda mi vida, a mi familia porque siempre me apoyaron y fueron el pilar fundamental para poder concluir esta carrera universitaria.

A mi compañero de tesis Alex por su gran apoyo, y confianza para poder concluir este proyecto.

A la Ing. Jacqueline Llanos por su guía, y haber sido una excelente profesora durante mis años de estudio.

Finalmente a todos mis amigos con los que compartí momentos inolvidables.

Alex Darwin Paredes Anchatipán

Mi más sincero agradecimiento a toda mi familia, quienes a lo largo de este período me han apoyado y motivado en esta formación académica y en todos los aspectos de mi vida.

A mis amigos incondicionales gracias por su apoyo, que formaron parte para conseguir esta meta y constituyen una gran parte de mi vida.

Finalmente un eterno agradecimiento a la Ing. Jacqueline Llanos y el Ing. Diego Ortiz por el tiempo, la guía, el apoyo y la confianza en el desarrollo de este proyecto.

ÍNDICE DE CONTENIDOS

CONTENIDO

PORTADA.....	i
CERTIFICADO.....	ii
DECLARACIÓN DE RESPONSABILIDAD.....	iii
AUTORIZACIÓN.....	iv
DEDICATORIA.....	v
AGRADECIMIENTO.....	vi
ÍNDICE DE CONTENIDOS.....	vii
ÍNDICE DE TABLAS.....	xiv
ABSTRACT.....	xvi
CAPÍTULO 1.....	1
GENERALIDADES.....	1
1.1. ANTECEDENTES.....	1
1.2. IDENTIFICACIÓN DEL SISTEMA.....	3
1.3. IDENTIFICACIÓN DEL SISTEMA USANDO MATLAB.....	8
1.4. CONTROL PREDICTIVO.....	18
1.5. DISPOSITIVOS DE CONTROL PARA ALGORITMOS DE CONTROL AVANZADO.....	28
CAPÍTULO 2.....	32
METODOLOGÍA DE IDENTIFICACIÓN DEL SISTEMA.....	32
2.1. ADQUISICIÓN DE DATOS.....	33
2.2. PREPROCESAMIENTO DE LOS DATOS.....	49
2.3. MODELADO DEL SISTEMA EN MATLAB.....	51
CAPÍTULO 3.....	60
DISEÑO Y SIMULACIÓN DEL CONTROLADOR PREDICTIVO BASADO EN MODELOS.....	60
3.1. DISEÑO DEL CONTROLADOR PREDICTIVO EN LABVIEW.....	60

3.2. SIMULACIÓN DEL CONTROLADOR MPC PARA LA PLANTA DE FLUJO	68
3.3. COMPARACIÓN ENTRE MPC Y PID SIMULADOS	77
CAPÍTULO 4.....	81
IMPLEMENTACIÓN DEL CONTROLADOR PREDICTIVO BASADO EN MODELOS EN LA PLANTA DE FLUJO	81
4.1. HARDWARE	81
4.2. SOFTWARE.....	87
CAPÍTULO 5.....	109
ANÁLISIS DE RESULTADOS.....	109
CAPÍTULO 6.....	126
CONCLUSIONES Y RECOMENDACIONES	126
6.1. CONCLUSIONES	126
6.2. RECOMENDACIONES	128

ANEXOS

ANEXO 1. EXPERIMENTOS DE TIEMPO MUERTO.....	¡ERROR! MARCADOR NO DEFINIDO.
ANEXO 2. DATOS DE ENTRADA – SALIDA .	¡ERROR! MARCADOR NO DEFINIDO.
ANEXO 3. MODELOS UTILIZADOS PARA LA VALIDACIÓN....	¡ERROR! MARCADOR NO DEFINIDO.
ANEXO 4. SIMULACIÓN DEL CONTROL PID.....	¡ERROR! MARCADOR NO DEFINIDO.
ANEXO 5. RESULTADOS DE LOS CONTROLES MPC SIMULADO Y MPC IMPLEMENTADO	¡ERROR! MARCADOR NO DEFINIDO.
BIBLIOGRAFÍA.....	129

LINKOGRAFÍA.....	131
------------------	-----

ÍNDICE DE FIGURAS

FIGURA 1.1: ETAPAS DE UN PROCESO DE IDENTIFICACIÓN.....	5
FIGURA 1.2: A) ENTRADA BINARIA ALEATORIA PARA SISTEMAS LINEALES, B) ENTRADA ESCALONADA ALEATORIA PARA SISTEMAS NO LINEALES. [2].....	8
FIGURA 1.3: ASPECTO DE LA INTERFAZ GRÁFICA DE USUARIO (GUI).....	10
FIGURA 1.4: MODELO ARX EN <i>IDENT</i>	13
FIGURA 1.5: MODELO ARMAX EN <i>IDENT</i>	14
FIGURA 1.6: MODELO OUTPUT – ERROR EN <i>IDENT</i>	15
FIGURA 1.7 : MODELO BOX – JEKINS EN <i>IDENT</i>	16
FIGURA 1.8: MODELO DE ESPACIO DE ESTADOS EN <i>IDENT</i>	17
FIGURA 1.9: HORIZONTES DE PREDICCIÓN Y DE CONTROL.	20
FIGURA 1.10: DISPOSITIVO NI MYRIO.....	29
FIGURA 1.11: CONVERTOR DE VOLTAJE/CORRIENTE.	30
FIGURA 1.12: RESISTENCIAS DE PRECISIÓN DE 249 OHMS.	31
FIGURA 2.1: FLUJOGRAMA DE LA METODOLOGÍA PARA LA IDENTIFICACIÓN DE UN SISTEMA.....	32
FIGURA 2.2: REPRESENTACIÓN DE ENTRADAS Y SALIDAS DE LA PLANTA DE FLUJO	34
FIGURA 2.3: EJEMPLO DE TIEMPO MUERTO EN PROCESO DE FLUJO.....	35
FIGURA 2.4: ANÁLISIS EN FRECUENCIA DE LA SALIDA DEL PROCESO DE FLUJO.....	37
FIGURA 2.5: ANÁLISIS EN FRECUENCIA, SEÑAL DE SALIDA ORIGINAL VS FILTRADA.	37

FIGURA 2.6: SEÑAL DE SALIDA EN EL TIEMPO, ORIGINAL Y FILTRADA.....	38
FIGURA 2.7: COMPARACIÓN ENTRE LA SEÑAL ORIGINAL, SEÑAL FILTRADA Y PROMEDIADA.....	39
FIGURA 2.8: VALIDACIÓN DEL RANGO SUPERIOR DE ENTRADA	42
FIGURA 2.9: VALIDACIÓN DEL RANGO INFERIOR DE ENTRADA.....	42
FIGURA 2.10: SEÑAL ESCALÓN BINARIA	43
FIGURA 2.11: SEÑAL ESCALÓN DE CINCO NIVELES.....	44
FIGURA 2.12: RESPUESTA DEL SISTEMA A LA ENTRADA ESCALÓN	45
FIGURA 2.13: DIAGRAMA DE BLOQUES DEL HARDWARE PARA ADQUISICIÓN.....	46
FIGURA 2.14: ESTRUCTURA DEL PROYECTO DE ADQUISICIÓN DE DATOS EN LABVIEW	47
FIGURA 2.15: PANEL FRONTAL Y DIAGRAMA DE BLOQUES DEL VI EN LA FPGA.....	48
FIGURA 2.16: PANEL FRONTAL Y DIAGRAMA DE BLOQUES DEL VI EN LA PC.....	49
FIGURA 2.18: CORRECCIÓN DEL TIEMPO DE MUESTRO EN <i>IDENT</i>	50
FIGURA 2.19: IMPORTACIÓN DE DATOS A <i>IDENT</i>	51
FIGURA 2.20: GRÁFICA EN EL TIEMPO DE VALORES ENTRADA-SALIDA.....	52
FIGURA 2.21: INTERFAZ DE MODELOS EN ECUACIONES DE ESTADO	53
FIGURA 2.22: SALIDA REAL Y SIMULADA EN <i>IDENT</i>	54
FIGURA 2.23: COMPARACIÓN ENTRE MODELOS POR NÚMERO DE ESTADOS.....	55
FIGURA 2.24: VENTANA DE LA HERRAMIENTA <i>IDENT</i> PARA VALIDACIÓN	56
FIGURA 2.25: COMPARACIÓN DE MODELOS PARA LOS DATOS “FLUJO”.....	57
FIGURA 2.26: COMPARACIÓN DE MODELOS PARA LOS DATOS “FLUJO1”.....	57
FIGURA 2.27: COMPARACIÓN DE MODELOS PARA LOS DATOS “FLUJO2”.....	58
FIGURA 3.1: PALETA DE CONTROL PREDICTIVO EN LABVIEW	60
FIGURA 3.2: FUNCIÓN <i>CREATE MPC CONTROLLER</i>	61

FIGURA 3.3: PARÁMETROS PARA EL CONTROLADOR MPC.....	62
FIGURA 3.4: ECUACIONES EN ESPACIO DE ESTADOS PARA EL MPC	63
FIGURA 3.5: MATRICES DE COSTO PARA EL CONTROLADOR PREDICTIVO.....	65
FIGURA 3.6: RESTRICCIONES PARA EL CONTROLADOR PREDICTIVO.....	66
FIGURA 3.7: FUNCIÓN <i>IMPLEMENT MPC CONTROLLER</i>	67
FIGURA 3.8: FUNCIÓN <i>GENERATE TIME PROFILES</i>	68
FIGURA 3.9: PERFILES DE <i>SETPOINT</i> EN EL PANEL FRONTAL	69
FIGURA 3.10: FUNCIÓN <i>STEP FORWARD MPC WINDOW</i>	69
FIGURA 3.11: <i>DISCRETE STATE SPACE</i> EN LABVIEW.....	70
FIGURA 3.12: DIAGRAMA DE BLOQUES DE LA SIMULACIÓN MPC	72
FIGURA 3.13: PESTAÑA MODELO EN EL PANEL FRONTAL DE LA SIMULACIÓN	74
FIGURA 3.14: PESTAÑA MPC EN EL PANEL FRONTAL DE LA SIMULACIÓN	75
FIGURA 3.15: PESTAÑA GRÁFICAS EN EL PANEL FRONTAL DE LA SIMULACIÓN	76
FIGURA 3.16: <i>SETPOINT VS PROCESS VALUE</i> , (A) PREDICTIVO, (B) PID.....	78
FIGURA 3.17: SEÑALES DE CONTROL, (A) PREDICTIVO, (B) PID.....	79
FIGURA 4.1: DIAGRAMA DE BLOQUES DEL LAZO DE CONTROL PREDICTIVO	81
FIGURA 4.2: FOTOGRAFÍA DE LA IMPLEMENTACIÓN EN LA PLANTA	82
FIGURA 4.3: CREACIÓN DE UNA RED MEDIANTE SÍMBOLO DEL SISTEMA.....	83
FIGURA 4.4: DISPOSITIVO MYRIO EN EL SOFTWARE DE RECONOCIMIENTO NI MAX	83
FIGURA 4.5: CONFIGURACIÓN DE CONEXIÓN WI-FI PARA MYRIO	84
FIGURA 4.6: UBICACIÓN DE LOS <i>JUMPERS</i> DE CONFIGURACIÓN DEL CONVERSOR V/I	85
FIGURA 4.7: CONFIGURACIÓN DE LOS <i>JAMPERS</i> DE ENTRADA Y SALIDA.....	86
FIGURA 4.8: CONEXIÓN FÍSICA ENTRE MYRIO, CONVERSOR Y RESISTENCIA.....	87

FIGURA 4.9: LISTA DE <i>TOOLKITS</i> NECESARIOS EN EL DISPOSITIVO MYRIO.....	88
FIGURA 4.10: ESQUEMA DEL PROYECTO “PREDICTIVO” EN LABVIEW	89
FIGURA 4.11: DIAGRAMA DE BLOQUES DEL VI FPGA.VI.....	90
FIGURA 4.12: PANEL FRONTAL DEL VI FPGA.VI.....	91
FIGURA 4.13: DIAGRAMA DEL BLOQUES DEL VI RT.VI	92
FIGURA 4.14: INICIALIZACIÓN DE PARÁMETROS MPC Y TIEMPO DE MUESTREO.....	93
FIGURA 4.15: CONVERSIÓN DEL MODELO EN TIEMPO CONTINUO A TIEMPO DISCRETO	93
FIGURA 4.16: CREACIÓN DEL CONTROLADOR PREDICTIVO BASADO EN MODELOS .	94
FIGURA 4.17: PROCESO PARA ABRIR UNA CONEXIÓN A LA FPGA.....	94
FIGURA 4.18: LAZO TEMPORIZADO PARA AJUSTAR LOS PARÁMETROS MPC.....	96
FIGURA 4.19: CONFIGURACIÓN DEL CONTROLADOR MPC DENTRO DEL LAZO DE CONTROL	97
FIGURA 4.20: ACONDICIONAMIENTO EN LABVIEW DEL <i>PROCESS VALUE</i>	98
FIGURA 4.21: ACONDICIONAMIENTO EN LABVIEW DEL <i>CONTROL VALUE</i>	99
FIGURA 4.22: FUNCIÓN <i>UPDATE MPC WINDOWS</i> PARA EL <i>SETPOINT</i>	100
FIGURA 4.23: IMPLEMENTACIÓN DEL CONTROLADOR MPC.....	101
FIGURA 4.24: PESTAÑA MODELO EN EL PANEL FRONTAL RT.VI.....	102
FIGURA 4.25: PESTAÑA CONTROLADOR MPC EN EL PANEL FRONTAL RT.VI	102
FIGURA 4.26: PESTAÑA VARIABLES EN EL PANEL FRONTAL RT.VI	103
FIGURA 4.27: DIAGRAMA DEL BLOQUES DEL VI PC.VI.....	104
FIGURA 4.28: CONVERSIÓN DE PRESIÓN DIFERENCIAL A CAUDAL EN LABVIEW..	105
FIGURA 4.29: CONVERSIÓN DEL <i>SETPOINT</i> INH ₂ O – MA EN PC.VI.....	106
FIGURA 4.30: CONVERSIÓN DEL <i>PROCESS VALUE</i> MA – INH ₂ O EN PC.VI.....	106
FIGURA 4.31: CONVERSIÓN DEL <i>CONTROL VALUE</i> MA - % EN PC.VI	107

FIGURA 4.32: PANEL FRONTAL DEL VI PC.VI.....	108
FIGURA 4.33: SETPOINT 10 – 25 INH ₂ O CONTROL MPC, (A) SIMULADO, (B) REAL.....	110
FIGURA 4.34: ACCIÓN DE CONTROL DE 10 – 25 INH ₂ O EN EL CONTROLADOR MPC, (A) SIMULADO, (B) REAL.....	111
FIGURA 4.35: SETPOINT 10 – 25 INH ₂ O, (A) MPC, (B) PID	113
FIGURA 4.36: ACCIÓN DE CONTROL 10 – 25 INH ₂ O, (A) MPC, (B) PID	114
FIGURA 4.37: SETPOINT 25 – 40 INH ₂ O, (A) MPC, (B) PID	115
FIGURA 4.38: ACCIÓN DE CONTROL 25 – 40 INH ₂ O, (A) MPC, (B) PID	116
FIGURA 4.39: SETPOINT 40 – 25 INH ₂ O, (A) MPC, (B) PID	117
FIGURA 4.40: ACCIÓN DE CONTROL 40 – 25 INH ₂ O, (A) MPC, (B) PID	118
FIGURA 4.41: SETPOINT 25 – 10 INH ₂ O, (A) MPC, (B) PID	119
FIGURA 4.42: ACCIÓN DE CONTROL 25 – 10 INH ₂ O, (A) MPC, (B) PID	120
FIGURA 4.43: SETPOINT 10 – 40 INH ₂ O, (A) MPC, (B) PID	121
FIGURA 4.44: ACCIÓN DE CONTROL 10 – 40 INH ₂ O, (A) MPC, (B) PID	122
FIGURA 4.45: SETPOINT 40 – 10 INH ₂ O, (A) MPC, (B) PID	123
FIGURA 4.46: ACCIÓN DE CONTROL 40 – 10 INH ₂ O, (A) MPC, (B) PID	124

ÍNDICE DE TABLAS

TABLA 2.1. TIEMPOS MUERTOS PARA DIFERENTES CAMBIOS DE ENTRADA.....	36
TABLA 3.1. COMPARACIÓN ENTRE MPC Y PID SIMULADOS	80
TABLA 4.1. DATOS PARA EL ACONDICIONAMIENTO DEL <i>PROCESS VALUE</i>	98
TABLA 4.2. DATOS PARA EL ACONDICIONAMIENTO DEL <i>CONTROL VALUE</i>	99
TABLA 4.3. DATOS PARA EL AJUSTE DEL <i>SETPOINT</i> INH ₂ O - MA	105
TABLA 4.4. DATOS PARA EL AJUSTE DEL <i>PROCESS VALUE</i> MA – INH ₂ O.....	106
TABLA 4.5: COMPARACIÓN ENTRE MPC SIMULADO Y MPC IMPLEMENTADO	112
TABLA 4.6. COMPARACIÓN ENTRE CONTROLES MPC Y PID IMPLEMENTADOS.....	125

RESUMEN

En este proyecto se realiza el diseño, simulación e implementación de un controlador predictivo basado en modelos, conocido como MPC (del inglés Model Predictive Control), aplicado a una planta de control de flujo en el laboratorio de Redes Industriales y Control de Procesos de la Universidad de las Fuerzas Armadas ESPE Extensión Latacunga. La simulación se realiza en el software LabVIEW, la implementación es sobre un dispositivo NI myRIO, gracias a la flexibilidad que ofrecen en software y hardware para este tipo de control en particular.

El MPC diseñado cuenta con las siguientes características: una función de costo que busca minimizar las acciones de control necesarias para corregir el error futuro, calculado en base al modelo de predicción, sujeto a restricciones de desigualdad que vienen definidas por el proceso y sus condiciones de operación, además se requiere de una etapa de optimización que envía las acciones de control óptimas. Para un buen diseño del MPC se requiere de un buen modelo de la planta y del modelo de predicción.

El modelamiento de la planta de flujo se realiza empleando ecuaciones en espacio de estados, para lo cual se propone y valida una metodología que incluye las etapas de: obtención de datos, adquisición de datos, preprocesamiento de datos, modelamiento y validación. Se muestran la simulación del comportamiento del controlador en lazo cerrado, esto principalmente para obtener valores de diseño ajustables como: tamaño del horizonte de predicción y control, matrices de ponderación y restricciones del sistema; que son usados de referencia para la implementación.

Finalmente se implementa el algoritmo de control en el proceso de flujo real para analizar su desempeño. Este controlador fue comparado con un PID evaluando: tiempo de estabilización, porcentaje de sobreimpulso y acciones de control, demostrando su superioridad.

Palabras Clave: MODELO DE CONTROL PREDICTIVO/SOFTWARE LABVIEW/CONTROL PID

ABSTRACT

This project is about the design, simulation and implementation of a model based predictive controller, known as MPC (Model Predictive Control), applied to a flow process in the Industrial Networks and Process Control laboratory of Army University ESPE – Latacunga. The simulation is performed in the LabVIEW software, and the implementation on a NI myRIO device, due to the flexibility offered by the software and equipment respectively for this particular type of predictive control.

The MPC designed in this study has the following characteristics: a cost function that seeks to minimize the control actions necessary to correct the future error calculated based on the prediction model, under inequality constraints that are defined by the process and operating conditions, after a stage of optimization, it gives optimal actions control. An important component is the plant model, used in turn as the prediction model.

The flow plant modeling is made by state-space equations, for this is proposed and validated a methodology for obtaining a good model including the next steps: obtaining data, acquisition data, preprocessing data, modeling and validation. The controller behavior is simulated in a closed loop, this mainly to obtain adjustable design values like: size of prediction horizon and control, weighting matrices and system constraints, which are used as a reference for implementation.

Finally, the designed control algorithm is implemented in the real flow process to analyze its performance. This controller is compared with a PID control and appreciates principally the stabilization time, overshoot percent and control actions.

Keywords:MODEL PREDICTIVE CONTROL/SOFTWARE LABVIEW/
CONTROLPID

CAPÍTULO 1

GENERALIDADES

1.1. ANTECEDENTES

Dentro de la ingeniería del control y principalmente en el diseño de controladores avanzados como parte muy importante es el modelamiento de plantas, y más usado o defendido la identificación de sistemas que permitan experimentar algoritmos de control y evaluar sus resultados.

En esta sección se analizará identificación de sistemas para generar modelos de plantas, además algoritmos de control avanzados.

La identificación de sistemas comprende el diseño óptimo de los experimentos para generar eficientemente información útil para aproximar dichos modelos. Para la identificación de sistemas se usa métodos estadísticos para crear modelos matemáticos de sistemas dinámicos a partir de valores medidos

Este modelo permite al diseñador realizar y validar mediante simulación el ajuste de los parámetros de controladores eficientes, que permiten obtener una respuesta que satisfaga las especificaciones de diseño. En este tema se estudian diferentes alternativas para obtener el modelo de un sistema como paso previo al diseño de un controlador.

1.1.1. CONCEPTOS DEL SISTEMA

Sistema es todo ambiente en la que interactúan variables de diferentes tipos para originar señales observables. Con la posibilidad de estimar los estados del sistema desde el conocimiento de sus entradas y salidas.

Las señales observables que son de beneficio para el espectador se denominan salidas del sistema, mientras que las señales que pueden ser manipuladas por dicho espectador son las entradas del mismo.

El resto de señales que influyen en salidas pero no son manipuladas por el observador se denominan perturbaciones.

Los sistemas pueden clasificarse en sistemas lineales y no lineales.

Sistemas lineales: son aquellos que pueden describirse mediante ecuaciones diferenciales lineales. La propiedad más importante es que admiten el principio de superposición, que es una herramienta matemática que permite descomponer un problema lineal en dos problemas sencillos.

Sistemas no lineales: se caracteriza por ecuaciones diferenciales no lineales, pueden presentar un comportamiento al principio perfectamente explicable mediante la matemática lineal, para luego comportarse, conforme avanza su desarrollo, de un modo completamente caótico cuando éste alcanza un valor crítico.

1.1.2. MODELO DE UN SISTEMA

Cuando se necesita conocer el comportamiento de un sistema en unas determinadas condiciones y ante unas determinadas entradas, se puede recurrir a la experimentación sobre dicho sistema y a la observación de sus salidas.

Sin embargo, en muchos casos la experimentación puede resultar compleja o incluso imposible de llevar a cabo, lo que hace necesario trabajar con algún tipo de representación que se aproxime a la realidad, y a la que se conoce como modelo.

1.1.3. TIPOS DE MODELOS

a. En función del grado de formalismo matemático

Modelos mentales, intuitivos o verbales: Estos modelos carecen de formalismo matemático.

Modelos no paramétricos: Muchos sistemas quedan perfectamente determinados mediante un gráfico o tabla que describa sus propiedades dinámicas mediante un número no finito de parámetros. Por ejemplo, un

sistema lineal queda definido mediante su respuesta al impulso o al escalón, o bien su respuesta en frecuencia.

Modelos paramétricos o matemáticos: Para aplicaciones avanzadas, puede ser necesario utilizar modelos que describan las relaciones entre las variables del sistema mediante expresiones matemáticas como pueden ser ecuaciones diferenciales (sistemas continuos) o en diferencias (sistemas discretos).

b. En función del tipo de sistema y representación matemática

Determinísticos: Se dice que es un modelo determinístico cuando expresa la relación entre entradas y salidas mediante una ecuación exacta.

Estocásticos: Este tipo de modelo posee un cierto grado de incertidumbre.

Estáticos: Un sistema es estático cuando la salida depende únicamente de la entrada en ese mismo instante. En estos sistemas existe una relación directa entre entrada y salida, independiente del tiempo.

Dinámicos: Un sistema dinámico es aquel en el que las salidas evolucionan con el tiempo tras la aplicación de una determinada entrada.

Continuos: Los sistemas continuos trabajan con señales continuas y se caracterizan mediante ecuaciones diferenciales.

Discretos: Los sistemas discretos trabajan con señales muestreadas, y se describen mediante ecuaciones en diferencia.

1.2. IDENTIFICACIÓN DEL SISTEMA

Identificación del Sistema es la obtención de forma experimental de un modelo que reproduzca con suficiente exactitud, las características dinámicas del proceso objeto de estudio.

1.2.1. PROCESO DE IDENTIFICACIÓN

El proceso de identificación del sistema contiene los siguientes pasos:

Obtención de Entrada-Salida

Se selecciona el tipo de entradas con las cuales se excitará el sistema y se registrar la evolución de las salidas durante un intervalo de tiempo.

Tratamiento previo de datos registrados

Los datos registrados están habitualmente acompañados de ruidos indeseados u otro tipo de imperfecciones que puede ser necesario corregir antes de iniciar la identificación del modelo. Se desea, “preparar” los datos para facilitar y mejorar el proceso de identificación.

Elección de una estructura de modelo

Se selecciona un conjunto de modelos candidatos, dentro del cual se seleccionara un modelo apropiado, basándose en los datos disponibles de entrada y salida. Es decir se selecciona un modelo que mejor se ajuste a los datos.

Obtención de parámetros del modelo

Se procede a la estimación de los parámetros de la estructura que mejor ajustan la respuesta del modelo a los datos de entrada-salida obtenidos experimentalmente.

Validación del modelo

Consiste en establecer si el modelo logrado satisface el grado de exactitud requerido para la aplicación en cuestión.

Si se llega a la conclusión de que el modelo no es válido, se deben revisar los siguientes aspectos como posibles causas:

a) El conjunto de datos de entrada-salida no proporciona suficiente información sobre la dinámica del sistema.

b) La estructura escogida no es capaz de proporcionar una buena descripción del modelo.

c) El criterio de ajuste de parámetros seleccionado no es el más adecuado.

Según las causas estimadas, deberá repetirse el proceso de identificación desde el punto correspondiente. Por tanto, el proceso de identificación es un proceso iterativo, cuyos pasos pueden observarse en el organigrama de la Figura 0.1.

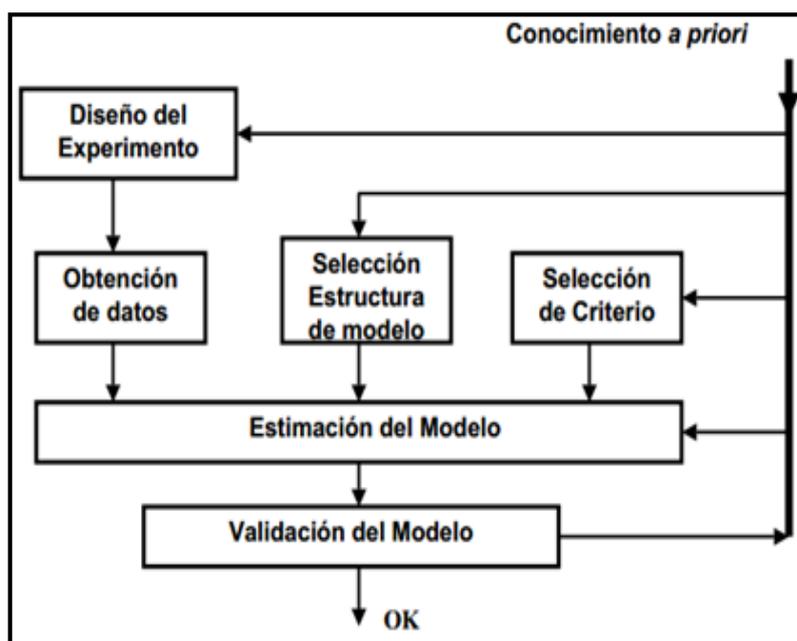


Figura 0.1: Etapas de un proceso de identificación. [1]

1.2.2. MÉTODOS DE IDENTIFICACIÓN

Existen diversos métodos de identificación que pueden clasificar dependiendo de distintos criterios:

c. De acuerdo al tipo de modelo obtenido

Método paramétrico

Estos métodos requieren la elección de una posible estructura del modelo, de un criterio de ajuste de parámetros, y por último de la estimación de los parámetros que mejor ajustan el modelo a los datos experimentales.

Método no paramétrico

Algunos de estos métodos son: análisis de la respuesta transitoria, análisis de la respuesta en frecuencia, análisis de la correlación, análisis espectral, análisis de Fourier, etc.

d. De acuerdo a la aplicación

Métodos de identificación off-line (a posteriori)

Utilizados en aquellas aplicaciones en que no se requiera un ajuste continuado del modelo. En estos casos, se realiza la identificación previa de la planta, considerándose que la validez de los parámetros obtenidos no se verá alterada con el paso del tiempo.

Métodos de identificación on-line (identificación recursiva)

Utilizados en aquellas aplicaciones en que los parámetros se van actualizando continuamente a partir de nuevos datos de entrada-salida obtenidos durante la evolución del proceso. Estos métodos son muy utilizados en sistemas de control adaptivo.

De acuerdo al criterio de ajuste

Existen diversos métodos matemáticos para ajustar los parámetros de una estructura a un conjunto de datos de entrada-salida. El más utilizado en el campo de identificación es el método de mínimos cuadrados.

1.2.3. CONSIDERACIONES AL MANIPULAR EL MÉTODO DE IDENTIFICACIÓN DE SISTEMAS

Todo conocimiento físico previo sobre la planta a identificar facilita en gran medida el proceso de identificación.

e. Obtención de datos

Primero se realiza algún tipo de experimento sobre el sistema bajo estudio para obtener los datos de entrada-salida que servirán de base para la obtención del modelo final.

f. Elección de la Señal a medir

Se registra las señales (mediante algún tipo de sistema de adquisición y el correspondiente sistema de almacenamiento de datos), se debe tener en cuenta que pueden existir señales que aunque afecten a la evolución de la salida, no pueden considerarse como entradas debido a la imposibilidad de actuar sobre ellas.

En el caso de que estas señales puedan ser medidas, pueden considerarse también como entradas al sistema (midiéndose sus valores durante el experimento). Caso contrario se deben considerar como perturbaciones.

Entradas: Señales que se puedan manipular

Salida: Señales que se puedan medir

Perturbaciones: no se pueden manipular ni medir

g. Elección del tipo de entradas

Las entradas al sistema deben ser cuidadosamente elegidas de forma que los datos recogidos proporcionen toda la información posible sobre el sistema. A este aspecto, conviene tener en cuenta los siguientes pasos:

La señal de entrada debe contener el mayor número de frecuencias posibles. Por ejemplo, una señal senoidal pura no es adecuada en un experimento de identificación, puesto que sólo se obtendrá la respuesta del sistema para frecuencia de dicha señal. Las señales escalonadas (con cambios bruscos) son utilizadas ya que contienen un espectro suficientemente amplio de frecuencias.

Para sistemas lineales: dos niveles de entrada, preferiblemente barriendo todos los rangos de variación permitidos. En este tipo de sistemas se suelen utilizar señales binarias de duración aleatoria (conocidas como señales binarias aleatorias o pseudoaleatorias), como la mostrada en la Figura 0.2 (a). Sin embargo, para sistemas no lineales es necesario trabajar con más de dos niveles de entrada, como se muestra en la Figura 0.2 (b).

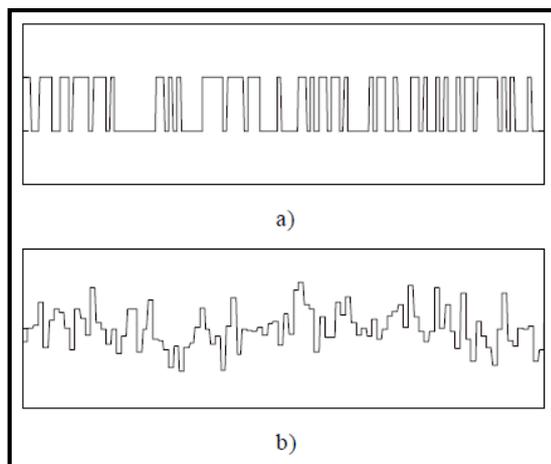


Figura 0.2: a) Entrada binaria aleatoria para sistemas lineales, b) Entrada escalonada aleatoria para sistemas no lineales. [2]

h. Elección del período de muestreo

La elección del periodo de muestreo está directamente relacionada con las constantes de tiempo del sistema, y tiene una influencia decisiva en el experimento de identificación.

Así, un periodo de muestreo se relaciona con las constantes de tiempo del sistema:

- Si es muy pequeño: Obtención de datos redundantes
- Si es muy grande: No es posible reproducir la dinámica del sistema.

Regla comúnmente usada: Frecuencia de muestreo de diez veces el ancho de banda del sistema.

i. Elección del número de muestras a tomar

En principio cuanto más información se tenga sobre el sistema, más exacto será el proceso de identificación.

El número de muestras a recoger durante el experimento viene limitado por la capacidad del dispositivo de memoria utilizado.

1.3. IDENTIFICACIÓN DEL SISTEMA USANDO MATLAB

El software MATLAB, posee una herramienta para la identificación de sistemas que se lo ejecuta con el comando *ident*, el cual nos muestra una interfaz gráfica para dicho propósito.

Permite crear y utilizar modelos de sistemas dinámicos no fácilmente modelados a partir de primeros principios o especificaciones. Puede utilizar los datos de entrada-salida de dominio de tiempo y dominio de la frecuencia para identificar las funciones de transferencia en tiempo continuo y discreto, modelos de procesos en ecuaciones de espacio de estado, entre otros.

El *toolbox* ofrece técnicas de identificación, como máxima probabilidad, la minimización de la predicción de errores (PEM), y sistema de identificación sub-espacial. Para representar la dinámica de sistemas no lineales, se puede estimar modelos Hammerstein - Wiener y modelos no lineales ARX con la red wavelet, árbol - partición, y no linealidades.

El estudio de la metodología de identificación de sistemas puede ser: obtención de modelos por métodos paramétricos y no paramétricos.

1.3.1. INTRODUCCIÓN A LA INTERFAZ GRÁFICA GUI

Las últimas versiones del *toolbox ident* proporcionan un Interfaz Gráfica de Usuario que permite manejar todas las funciones anteriores de forma prácticamente transparente al usuario, sin necesidad de llamarlas desde la línea de comandos.

Para arrancar el interfaz de usuario (GUI), basta con teclear en línea de comandos de Matlab: ***ident***. Apareciendo la ventana principal como se muestra en la Figura 0.3.

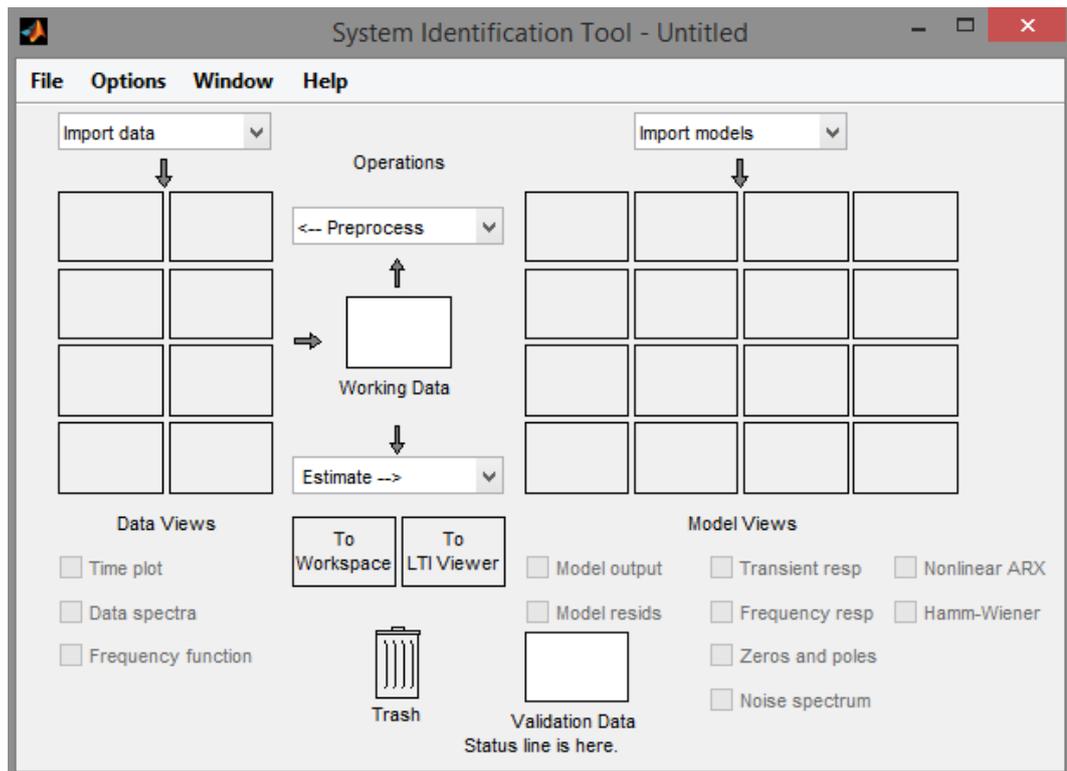


Figura 0.3: Aspecto de la Interfaz Gráfica de Usuario (GUI).

Elaborado por: López, V y Paredes, A

La ventana dispone de dos zonas con varios recuadros cada una. El tablero de datos situado en la zona izquierda de la pantalla, permite incluir en cada uno de los recuadros un conjunto distinto de datos de entrada-salida, representados por un icono. El tablero de modelos está en la zona derecha de la pantalla, y puede contener en cada uno de sus recuadros diferentes modelos obtenidos a partir de la identificación realizada con datos del tablero de datos. Cada modelo quedará representado también por un icono distinto.

Los datos del tablero de datos pueden provenir de las siguientes fuentes:

- i. De otras sesiones anteriores con el GUI.
- ii. Del *workspace* de MATLAB.
- iii. Del tratamiento de otro conjunto de datos contenido en el tablero de datos.

Por su parte, los modelos pueden provenir de las siguientes fuentes:

- i. De sesiones anteriores con el GUI.
- ii. Del *workspace* de MATLAB.
- iii. De la identificación a partir de los datos contenidos en el tablero de datos.

j. Los datos de trabajo

Todos los procesos realizados mediante el GUI actúan sobre los llamados “Datos de Trabajo”, contenidos en el recuadro central de la ventana (*working data*). Para modificar los datos de trabajo basta con arrastrar con el ratón el icono con los nuevos datos de trabajo desde el tablero de datos hasta el recuadro *working data*.

k. Los datos de validación

Del mismo modo, todos los procesos del GUI que necesiten datos para validación los tomarán del recuadro (*validation data*) situado debajo del tablero de modelos.

l. Las representaciones (Views)

Tanto los datos de entrada-salida como los modelos pueden representarse en pantalla de diversas formas.

Para representar en pantalla un conjunto de datos del tablero de datos, en primer lugar hay que hacer clic con el ratón sobre su icono, quedando éste resaltado mediante una línea más gruesa. Pueden seleccionarse varios conjuntos de datos simultáneamente. Para desactivar un conjunto de datos, se vuelve a hacer clic con el ratón sobre su icono. A continuación se selecciona en el menú de *Data Views* el tipo de representación que se desea: representación temporal de las señales (*Time plot*) o del espectro de las mismas (*Data spectra*).

Con los modelos se procede de igual manera, seleccionando con el ratón aquéllos que se quieren representar, y escogiendo el tipo de representación entre salida del modelo (*Model output*), residuos del modelo (*Model resids*), respuesta transitoria (*Transient resp*), respuesta frecuencial

(*Frequency resp*), ceros y polos (*Zeros and poles*) y espectro del ruido (*Noise spectrum*).

1.3.2. ESTRUCTURA DE MODELOS

El Toolbox permite trabajar varios modelos como:

m. Modelo ARX

Es el modelo más utilizado, su estructura lineal se basa en una ecuación en diferencias, como la mostrada en la Ecuación (0.1).

$$\begin{aligned} y(t) + a_1y(t - 1) + \dots + a_{na}y(t - na) \\ = b_1u(t - nk) + \dots + b_{nb}u(t - nk - nb + 1) \end{aligned} \quad (0.1)$$

La distribución es definida por los tres enteros na , nb y nk . Definiendo na es igual al número de polos y $nb - 1$ es el número de ceros, mientras que nk es el tiempo de demora (tiempo muerto) en el sistema.

Para un sistema en virtud de la muestra de datos de control, simplemente nk es igual a 1 si no hay un tiempo muerto. Para los sistemas múltiples de entrada nk y nb son vectores fila, donde el i -ésimo elemento de la orden o retraso es asociado con el i -ésimo de entrada. Un ejemplo en la herramienta *ident* se ve en la Figura 0.4.

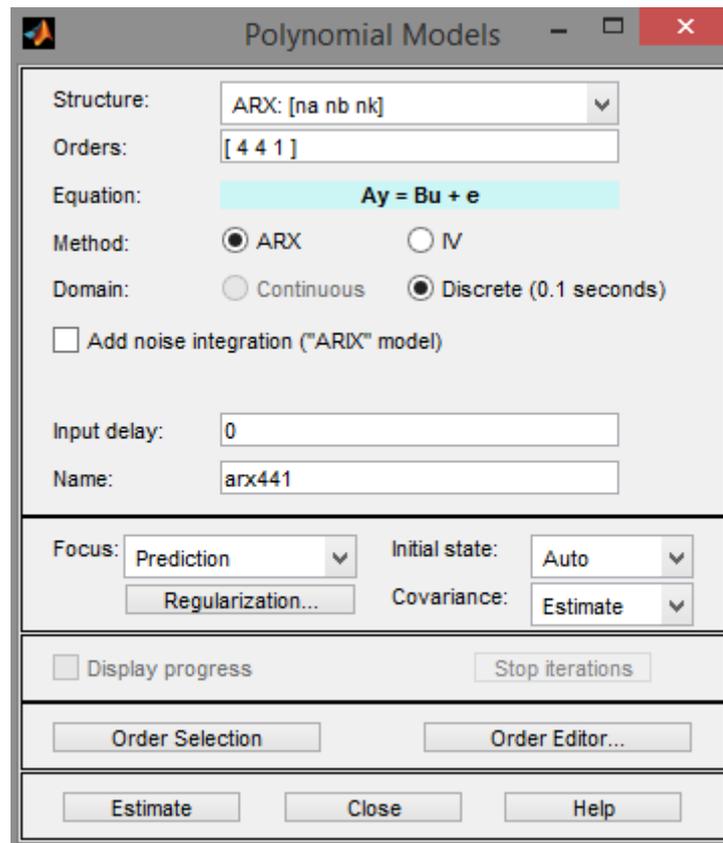


Figura 0.4: Modelo ARX en *ident*

Elaborado por: López, V y Paredes, A

n. Modelo ARMAX

Es un modelo entrada- salida lineal de solo una entrada U y de salida Y , la cual se puede escribir como indica la Ecuación (0.2).

$$\begin{aligned}
 y(t) + a_1y(t - 1) + \dots + a_nay(t - na) \\
 = b_1u(t - nk) + \dots + b_{nb}u(t - nk - nb + 1) \\
 + e(t) + c_1e(t - 1) + \dots + c_{nc}e(t - nc)
 \end{aligned} \quad (0.2)$$

A grandes rasgos el modelo ARMAX utiliza los mismos coeficientes en cuanto a polos y ceros del modelo ARX, con la diferencia de que no, representa los desplazamientos temporales del ruido parametrizado. La diferencia se puede apreciar en la Figura 0.5.

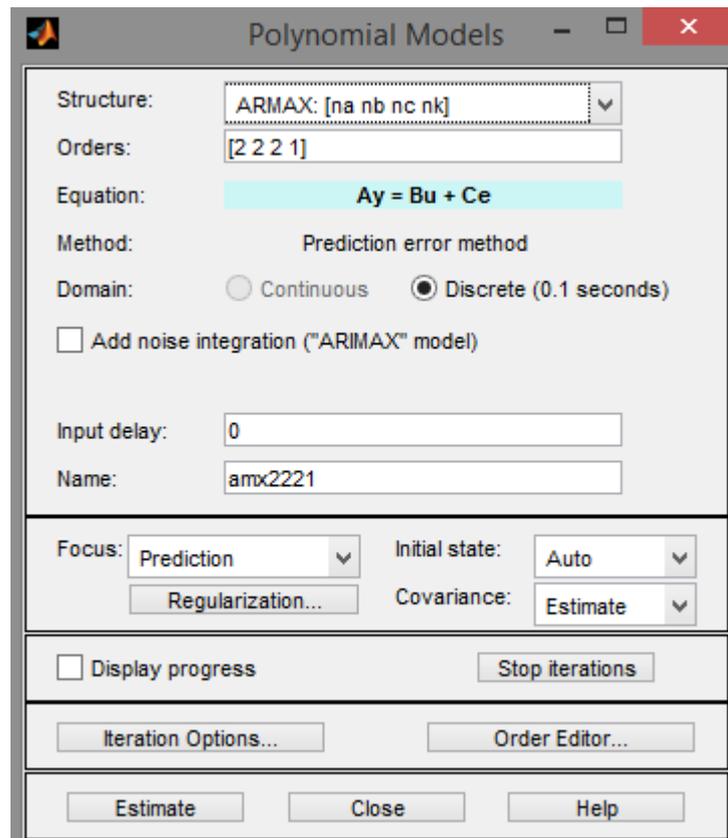


Figura 0.5: Modelo ARMAX en *ident*.

Elaborado por: López, V y Paredes, A

Modelo OUTPUT – ERROR

Es un caso especial del modelo ARX, su diferencia está en el modelado del ruido, indicado en la Ecuación (0.3).

$$y(t) = \frac{B(q)}{F(q)}u(t - nk) + e(t) \quad (0.3)$$

Donde $B(q)$ y $F(q)$ son polinomios con parámetros nb y nf , que son las órdenes de las secuencias dependientes, como se indica en la Ecuación (0.4) y Ecuación (0.5) respectivamente, y los nk son los retardos de la entrada. El gráfico de la estimación en *ident* se puede observar en la Figura 0.6.

$$nb: B(q) = b_1 + b_2q^{-1} + \dots + b_{nb}q^{-nb+1} \quad (0.4)$$

$$nf: F(q) = 1 + f_1q^{-1} + \dots + f_{nf}q^{-nf} \quad (0.5)$$

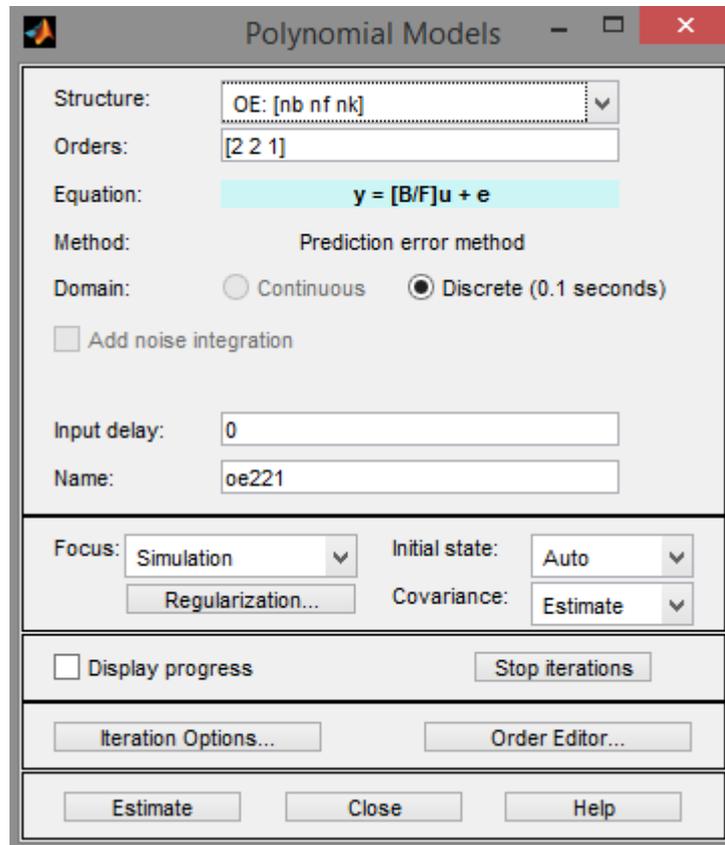


Figura 0.6: Modelo OUTPUT – ERROR en *ident*

Elaborado por: López, V y Paredes, A

o. Modelo BOX-JEKINS

Es otra variación del modelo ARX en cuanto al modelado del ruido. Las ecuaciones que lo describen se indican a partir de la Ecuación (0.6) hasta la Ecuación (0.10).

$$y(t) = \left[\frac{B(q)}{F(q)} \right] u(t - nk) + \left[\frac{C(q)}{D(q)} \right] e(t) \quad (Box - Jenkins) \quad (0.6)$$

$$nf: F(q) = 1 + f_1q^{-1} + \dots + f_{nf}q^{-nf} \quad (0.7)$$

$$nb: B(q) = b_1 + b_2q^{-1} + \dots + b_nq^{-nb+1} \quad (0.8)$$

$$\text{nc: } C(q) = 1 + c_1q^{-1} + \dots + c_{\text{nc}}q^{-\text{nc}} \quad (0.9)$$

$$\text{nd: } D(q) = 1 + d_1q^{-1} + \dots + c_{\text{nd}}q^{-\text{nd}} \quad (0.10)$$

Un ejemplo de la modelación con el modelo BOX – JEKINS mediante el uso de MATLAB, se puede observar en la Figura 0.7.

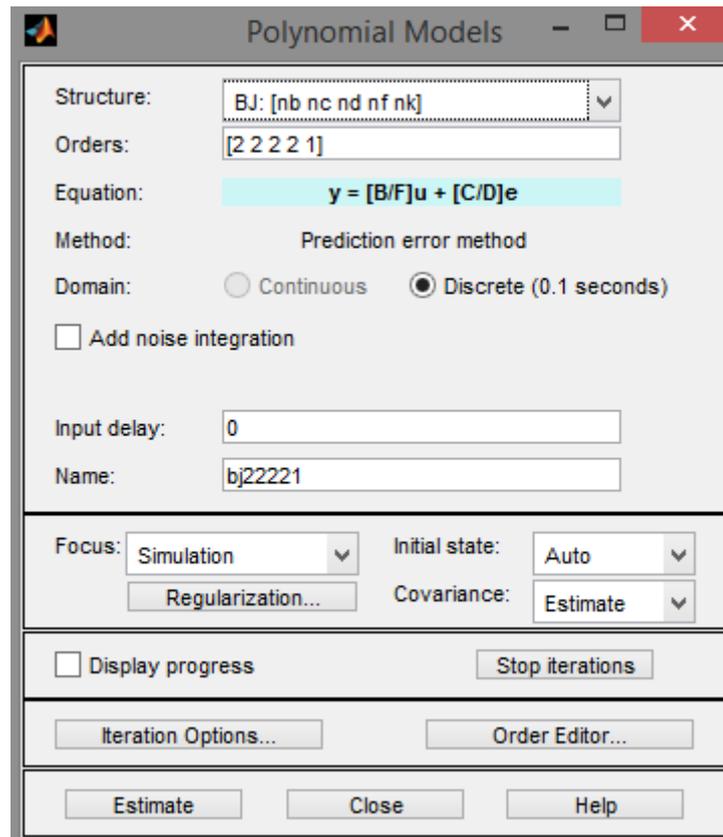


Figura 0.7 : Modelo BOX – JEKINS en *ident*

Elaborado por: López, V y Paredes, A

p. Modelo en ESPACIO DE ESTADOS

Este modelo se detalla matemáticamente en dos ecuaciones, la de estados de la forma indicada en la Ecuación (0.11), y la de salidas como la indicada en la Ecuación (0.12).

$$x(t + 1) = Ax(t) + Bu(t) + Ke(t) \quad (0.11)$$

$$y(t) = Cx(t) + Du(t) + e(t) \quad (0.12)$$

El índice más importante de la estructura del modelo es su orden, es decir; la dimensión del vector de estado x . Si el coeficiente k se lleva a cero, el método se vuelve de tipo Output – Error.

La opción PEM estima con el método de error de predicción (Prediction Error Method).

La opción N4SID estima los parámetros utilizando el método de sub espacios.

La ventana del *toolbox ident* para la estimación por el método de espacio de estados puede apreciarse en la Figura 0.8.

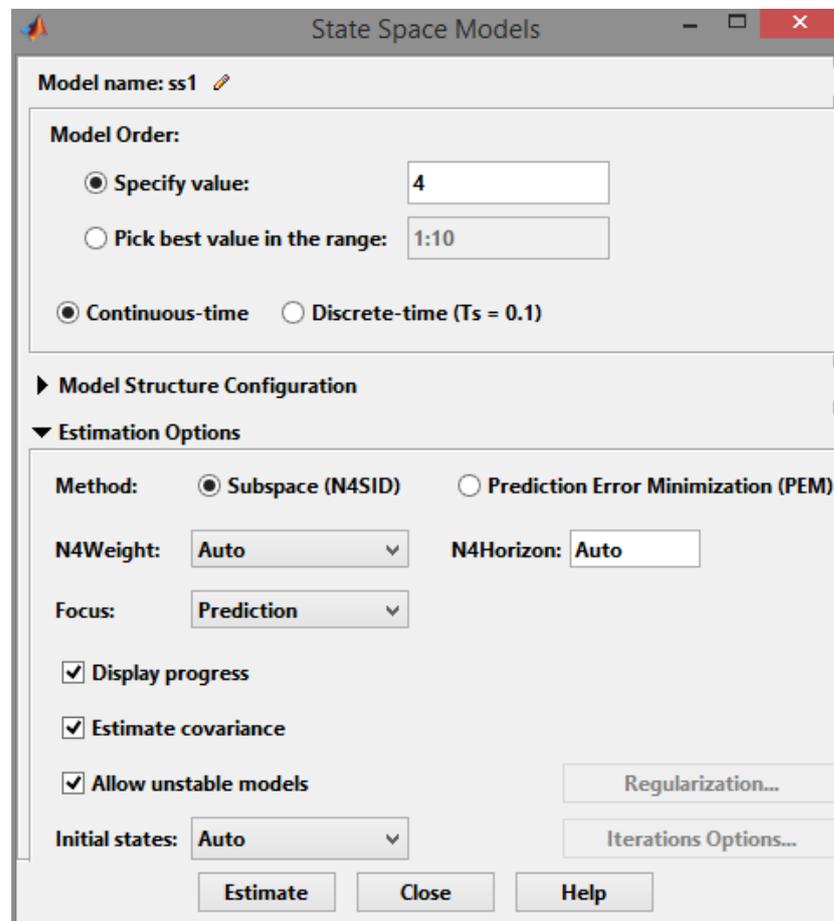


Figura 0.8: Modelo de Espacio de Estados en *ident*

Elaborado por: López, V y Paredes, A

1.4. CONTROL PREDICTIVO

El control predictivo, compone un campo muy amplio de métodos de control desarrollados en torno a ciertas ideas comunes e integra diversas disciplinas como control óptimo, control estocástico, control de procesos con tiempos muertos, control multivariable o control con restricciones.

El control predictivo no es una estrategia de control específica, sino que se trata más bien de un campo muy amplio de métodos de control desarrollados en torno a ciertas ideas comunes. Estos métodos de diseño conducen a controladores lineales, que poseen prácticamente la misma estructura.

Las ideas que aparecen en mayor o menor medida en toda la familia de controladores predictivos son básicamente:

- Uso explícito de un modelo para predecir la salida del proceso en futuros instantes de tiempo (horizonte).
- Cálculo de las señales de control minimizando una cierta función objetivo.
- Estrategia deslizante, de forma que en cada instante el horizonte se va desplazando hacia el futuro, lo que implica aplicar la primera señal de control en cada instante y desechar el resto, repitiendo el cálculo en cada instante de muestreo

1.4.1. INTRODUCCIÓN AL CONTROL PREDICTIVO BASADO EN MODELOS

El término control predictivo basado en modelos (MPC en inglés *Model Predictive Control*) se aplica a un conjunto de estrategias de control por computador, organizadas en torno a algunas ideas comunes.

El MPC está basado en un modelo del proceso a controlar, el cual es empleado para predecir la evolución futura del estado o de las salidas.

El modelo del proceso es conocido como *modelo de predicción*, siendo cualquier tipo de modelo matemático que exprese al sistema.

El intervalo de tiempo (expresado en instantes de muestreo) sobre el que se predice la evolución de las salidas es conocido como *horizonte de predicción*. Por tanto, si el horizonte de predicción es N , empleando el modelo de predicción se calculará la secuencia de salidas como indica la Ecuación (0.13).

$$\hat{y}(t + 1|t), \hat{y}(t + 2|t), \dots, \hat{y}(t + N|t) \quad (0.13)$$

Para el cálculo de esta secuencia se emplea la información de la evolución del proceso hasta el instante t , es decir:

- Valores pasados de las entradas o actuaciones del proceso.
- Valores presentes y pasados de las salidas del proceso.

También se emplea una secuencia de actuaciones o señales de control futuras (*horizonte de control*), la serie será como la mostrada en la Ecuación (0.14).

$$u(t|t), u(t + 1|t), \dots, u(t + N_u - 1|t) \quad (0.14)$$

Usualmente se suele considerar un *horizonte de control* $N_u < N$, de manera que para los instantes futuros más allá del horizonte de control se supone que la señal de control es constante. Estos conceptos se ilustran en la Figura 0.9.

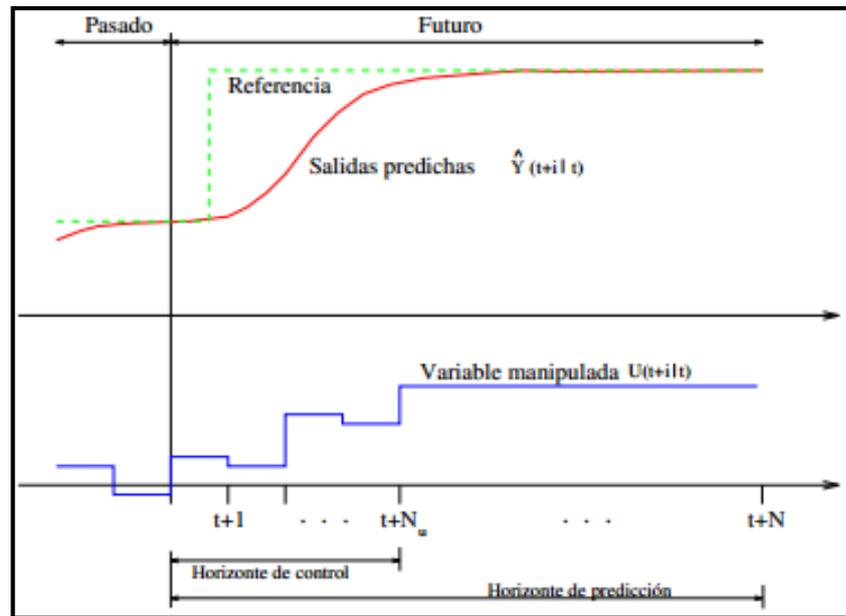


Figura 0.9: Horizontes de predicción y de control. [3]

La secuencia de actuaciones futuras condiciona en gran medida la evolución futura del proceso. Para medir la bondad del control obtenido se puede formular un criterio o índice de comportamiento, de manera que se puede obtener, mediante técnicas de optimización, la secuencia de actuaciones óptima.

Esta sería la que, de acuerdo con el criterio elegido, proporcione el mejor control. El cálculo de la señal de control mediante la minimización de un índice de comportamiento o función objetivo, es otra de las ideas comunes a las técnicas de control predictivo.

La forma de la función objetivo varía de un tipo de control a otro. Una de las formas más populares es la de un criterio cuadrático en el que se calcula la suma de los errores predichos (diferencia entre las salidas predichas y la trayectoria de referencia) al cuadrado a lo largo del horizonte de predicción. Es habitual también considerar algún tipo de ponderación del esfuerzo de control, por lo que la forma del criterio será como la mostrada en la Ecuación (0.15).

$$J(u) = \sum_{k=1}^N (\hat{y}(t+k|t) - w(t+k))^2 + \lambda \sum_{k=1}^N u(t+k-1|t)^2 \quad (0.15)$$

Dónde:

u es la secuencia de actuaciones futuras,

$w(t+k)$ es el valor de la trayectoria de referencia

λ es un parámetro que pondera la importancia que tiene en la Optimización el término del esfuerzo de control.

En vista de lo anterior el objetivo de un controlador predictivo será obtener en cada instante de muestreo la secuencia de actuaciones u^* que hace mínimo el índice de funcionamiento, es decir:

$$u^* = \arg \min_{u \in U} J(u) \quad (0.16)$$

Donde U es el conjunto de secuencias de actuaciones admisibles. Este conjunto vendrá definido por restricciones sobre las entradas y salidas de la planta (las cuales a su vez determinan restricciones sobre las entradas).

Con la posibilidad de limitar la búsqueda de la secuencia de actuaciones mediante restricciones es una de las características más apreciables y genuinas del control predictivo. En caso de no considerarse restricciones $U \in \mathbb{R}^N$. [3]

1.4.2. ELEMENTOS BÁSICOS DEL MPC

Todos los controladores predictivos poseen elementos comunes y para cada uno de estos elementos se pueden elegir diversas opciones, dando lugar a distintos algoritmos.

q. Modelo de Predicción

Es el modelo matemático que describe el comportamiento de la planta a controlarse, la representación y las características del modelo pueden ser lineales, no lineales, en tiempo continuo, o en tiempo discreto (más empleado), en variables de estado o en funciones de entrada y salida. El

diseño del modelo debe capturar la máxima dinámica del sistema, para permitir el cálculo de las predicciones y el análisis teórico.

El espacio temporal en el que se considera la evolución futura de la salida del proceso para el cálculo de la señal de control se denomina *horizonte de predicción*, y se denota por N . Las predicciones de la variable controlada en N instantes de muestreo en adelante con datos hasta el instante t será como la indicada en la Ecuación (0.13).

Se calcula con información de los valores pasados de las variables manipuladas y los valores presentes y pasados de las variables controladas. Si se conoce el tiempo muerto del proceso t_0 (estimado por un múltiplo entero del periodo de muestreo), se elegirá el comienzo del horizonte N_1 coincidente con t_0 .

r. Función Coste

Esta función suele tener la forma de un criterio cuadrático en la que aparecen los errores de seguimiento predichos y el esfuerzo de control. De manera general la función objetivo toma la forma:

$$J(u) = \sum_{k=N_1}^{N_2} \delta(k) [\hat{y}(t+k|t - w(t+k))]^2 + \lambda \sum_{k=1}^{N_u} \lambda(k) \Delta u(t+k-1|t)^2 \quad (0.17)$$

En esta función N_1 y N_2 determinan el comienzo y el final del horizonte de predicción, mientras que N_u es el horizonte de control. Los dos primeros parámetros marcan los límites de los instantes en los que es deseable que la salida siga a la trayectoria de referencia.

Si el retardo del proceso es d el valor más razonable para N_1 es $d + 1$. Por otra parte, el horizonte de control N_u no tiene por qué coincidir con N_2 , pudiendo ser un valor menor.

Esto trae como consecuencia la reducción del número de variables de decisión por lo que la optimización conllevará menos cálculo.

En este caso se suele suponer que la entrada permanece constante a partir del instante $t + N_u - 1$, es decir:

$$\Delta u(t + k - 1) = 0 \quad k > N_u \quad (0.18)$$

Por otra parte, los coeficientes $\delta(k)$ y $\lambda(k)$ son secuencias de ponderación que usualmente toman la forma:

$$\delta(k) = \alpha^{N_2 - k} \quad (0.19)$$

Si $0 < \alpha < 1$ los errores futuros más lejanos de t tienen más peso, dando lugar a un control más suave.

Si $\alpha > 1$ el error en los instantes más cercanos a t tienen más peso, por lo que el control será más rápido.

s. Estrategia de horizonte deslizante

Consiste en utilizar sólo el primer valor de la secuencia calculada $u^*, u^*(t|t)$, descartando el resto de la secuencia.

En el instante siguiente $t + 1$ se repite el cálculo y se aplica sólo el primer valor de la secuencia óptima, y así sucesivamente.

Debido a las incertidumbres de modelado, en general, $u^*(t + 1|t) \neq u^*(t + 1|t + 1)$. Esta forma de aplicar la ley de control supone la solución de un problema de control óptimo en bucle abierto en cada instante de muestreo que se aplica en bucle cerrado, con el fin de predecir en el siguiente instante de muestreo con datos más precisos que en el instante anterior, y por tanto calcular una solución más acertada. [4]

1.4.3. CONTROL PREDICTIVO EN LABVIEW

La herramienta computacional LabVIEW en el toolkit Control Design and Simulation, presenta un módulo denominado Control Predictive. Un VI

utilizado es el CD CREATE MPC CONTROLLER VI, para crear un controlador MPC.

Este VI calcula el controlador MPC basándose en un modelo en espacio de estados, incluye acción integral para mejorar la diferencia entre la planta real y el modelo de la planta.

El CD CREATE MPC CONTROLLER VI, posee una entrada que pertenece a los Parámetros de Estimador de Estados, que es el modelo interno del controlador, MPC, además se puede estimar los estados del modelo usando la función de observador discreto fuera del controlador MPC.

t. Horizonte de control y horizonte de predicción

Para construir el controlador MPC, es necesario proveer la siguiente información:

Horizonte de predicción (N_p), es el número de muestras en el futuro en la cual el controlador MPC predice la salida de la planta.

Horizonte de control (N_c), es el número de muestras dentro del horizonte de predicción durante el cual el controlador actuará sobre la planta.

Los dos conceptos ya se ilustraron en la Figura 0.9.

Es necesario tomar en cuenta:

- El horizonte de control es menor que el horizonte de predicción.
- Si el horizonte de predicción es corto, se reduce la longitud de tiempo en la que el controlador MPC puede predecir la señal de salida de la planta. Un horizonte de predicción muy pequeño causa que el controlador MPC opere algo parecido al controlador con realimentación tradicional.
- Un horizonte de predicción grande incrementa la capacidad de predecir al controlador MPC y observar una restricción potencial y evitarlo. Un horizonte de predicción muy grande decrementa el rendimiento del controlador MPC por el aumento de cálculos extras que realiza el algoritmo de control.

- Si el horizonte de control es corto, y la acción de control es grande podría pasarse del punto de referencia luego que el horizonte de control finalice.

Sin embargo como el controlador sigue ejecutándose la salida de la planta va a seguir colocando cerca del punto de referencia. Un horizonte de control grande produce cambios más agresivos en la acción de control. Esos cambios agresivos pueden resultar en oscilaciones o en energía innecesaria.

Cuando se tienen sistemas de control con retrasos de tiempo, para este tipo de modelos incluye un tiempo de retraso equivalente a D instantes de muestreo, es decir el controlador tienen un efecto de retraso y_{K+1} .

En esta situación es necesario aumentar este tiempo al horizonte de predicción de tal forma que el horizonte de predicción esté definido por $N_p > D$, mientras que el horizonte control sea $N_c \ll N_p - D$.

u. Función de costo

La función de costo a minimizar que utiliza LabVIEW está representada en la Ecuación (0.20), que es un algoritmo de optimización cuadrática:

$$\begin{aligned}
 j(k) = & \sum_{i=N_w}^{N_p} [\hat{y}(k+i|k) - r(k+i|k)]^T \cdot \mathbf{Q} \cdot [\hat{y}(k+i|k) \\
 & - r(k+i|k)] + \sum_{i=0}^{N_c-1} [\Delta u^T(k+i|k) \cdot \mathbf{R} \cdot \Delta u(k+i|k)] \\
 & + \sum_{i=N_w}^{N_p} [u(k+i|k) - s(k+i|k)]^T \cdot \mathbf{N} \cdot [u(k+i|k) \\
 & - s(k+i|k)]
 \end{aligned} \quad (0.20)$$

Dónde:

N_p Número de muestras del horizonte de predicción

N_w Comienzo del horizonte de predicción

N_c Matriz de muestras del horizonte de control

Q Matriz de ponderación de error de salida

R Matriz de ponderación de la tasa de cambio en la acción de control.

N Matriz de ponderación de la acción de control

$\hat{y}(k + i|k)$ Salida de la planta predicha al tiempo $k + i$, y entrega todos los valores superiores e incluido el instante k .

$r(k + i|k)$ Salida de la planta al tiempo $k + i$, y entrega todos los valores superiores e incluido el instante k .

$\Delta u(k + i|k)$ Tasa de cambio de la acción de control predicha, y entrega todos los valores superiores e incluidos el instante k

$u(k + i|k)$ Acción de control óptima predicha al tiempo $k + i$, y entrega todos los valores superiores e incluido el instante k .

$s(k + i|k)$ Señal de referencia de entrada al tiempo $k + i$, entregando todos los valores superiores e incluidos el instante k .

Donde Q es una matriz de $n_y \times n_y$, R y N son matrices de $n_u \times n_u$ todas estas matrices son positivas.

v. Restricciones

En adición al peso de ponderación de las matrices en la función de costo, se puede detallar las restricciones de los parámetros de los controladores MPC.

El peso de ponderación de las matrices ajusta las prioridades de la acción de control, la tasa de cambio en la acción de control y la salida de las plantas. Tomando en cuenta las restricciones y los pesos de ponderación se puede seleccionar dos instancias de optimización de los controladores.

Método de optimización dual

Se usa la optimización dual para explicar un valor de restricción inicial, final, un valor mínimo y máximo para la acción de control, para la tasa de

cambio y la salida de la planta. Se aprovechan estas limitaciones para representar las restricciones en el mundo real.

Al usar la optimización dual, las restricciones tienen un peso de ponderación igual y no pondera una restricción específica. Cuando se usa el método de optimización dual, el algoritmo MPC ajusta el controlador para que las restricciones especificadas nunca se sobrepasen.

Método de función barrera

La función barrera describe un valor de restricción inicial y final mínimo y máximo para la acción de control, para la tasa de cambio y la salida de la planta.

Sin embargo el método de la función barrera además asocia una penalidad y una tolerancia con cada una de las restricciones.

La penalidad es una restricción y específica cual es la tendencia que evita alcanzar al valor de restricción.

La tolerancia determina la distancia desde el valor de la restricción a que la penalidad empieza activarse.

1.4.4. RESTRICCIONES Y PESOS DE PONDERACIÓN

Con el método de función barrera, se puede priorizar las restricciones y especificar los pesos de ponderaciones. Cuando el algoritmo MPC reconoce que la penalidad sobre una restricción se activa, el algoritmo incorpora la penalidad en la función de costo y se ajusta la acción de control.

Para cada variable restringida el algoritmo MPC debe balancear la penalidad con algún peso de ponderación. El método de función barrera además balancea las restricciones.

Si se considera una situación donde se especifica una restricción máxima junto a la salida de la planta y la acción de control del controlador. La penalidad especificada por y_{\max} es relativa a la penalidad que se especifica para u_{\max} .

Si se especifica una penalidad grande para y_{\max} entonces el algoritmo MPC para u_{\max} , prioriza la salida de restricción de la planta por encima de la restricción de la acción de control.

Por lo tanto en una situación donde las penalidades juntas están activas, el algoritmo MPC intenta minimizar la penalidad en y_{\max} antes de minimizar la penalidad en u_{\max} .

El método de la función barrera es usado cuando se necesita priorizar las restricciones en diferentes parámetros en orden para reflejar sistemas más reales.

1.5. DISPOSITIVOS DE CONTROL PARA ALGORITMOS DE CONTROL AVANZADO

En la industria ecuatoriana los algoritmos de control predictivo no son altamente implementados, por falta de conocimiento, por falta de dispositivos industriales donde implementarlos, este tema de tesis genera las soluciones para la aplicación en sistemas reales y validar la superioridad que tiene este tipo de controladores.

Los PAC's de National Instruments son diseñados con la tecnología de procesamiento más reciente, para resolver problemas avanzados de control en entornos industriales y robustos.

Un PAC de NI es completamente programable con NI LabVIEW, un entorno integrado de desarrollo gráfico. NI CompactRIO es un PAC avanzado de National Instruments con más de 60 módulos de E/S, más de 15 opciones de chasis y múltiples protocolos de comunicación.

VENTAJAS DE LOS PACS DE NI:

- **E/S de Gran Calidad a su Sistema de Control:** Los módulos analógicos están disponibles desde 12 hasta 24 bits y tienen velocidades de muestreo de DC hasta 1 MS/s/canal. Los NI PACs son excelentes en sistemas de control analógico intensivos.
- **NI PAC para Técnicas Avanzadas y Personalizadas de Control:** LabVIEW, el entorno de desarrollo gráfico para PACs de NI, maneja de

una manera natural técnicas de programación más avanzadas, basadas en computación multinúcleo y de hilos múltiples. Esto le ayuda a programar sistemas de control que no serían posibles con una sintaxis más restringida como 61131.

- **Operación en Entornos Robustos** (-40 °C a 70 °C, 5gRMS, 50g de impacto): El hardware de NI está diseñado para integrarse con sistemas existentes de PLCs, así usted puede usar CompactRIO para cubrir sus necesidades más complejas de control y medidas, y continuar usando PLCs estándares implementados actualmente.

1.5.1. NI MYRIO

NI myRIO es un instrumento que tiene un procesador ARM® Cortex™-A9 dual-core en tiempo real y E/S personalizadas de FPGA Xilinx. En la Figura 0.10, se muestra el dispositivo.



Figura 0.10: Dispositivo NI myRIO. [5]

Propiedades:

- Herramienta rentable para enseñar e implementar múltiples conceptos de diseño con un dispositivo
- 10 entradas analógicas, 6 salidas analógicas; 40 líneas E/S digital
- Inalámbricos, LEDs, push-button, acelerómetro interno
- Xilinx FPGA y procesador dual-core ARM® Cortex™-A9

- Programable con NI LabVIEW o C; adaptable para diferentes niveles de programación. [5]

1.5.2. CONVERSION DE VOLTAJE/CORRIENTE

Los acondicionadores de señal de tensión de DRF-VDC y DRF-VAC admiten tensiones CC y CA respectivamente y proporcionan una salida aislada de 0 a 10 Vcc o de 4 a 20 mA. El dispositivo se muestra en la Figura 0.11.

Los modelos están disponibles con tres opciones de alimentación diferentes: 24 Vcc, 120 Vca y 240 Vca.

Los modelos DRF-VDC y DRF-VAC son idóneos para aplicaciones industriales. Todos los modelos se montan en un riel DIN estándar de 35 mm y ofrecen un aislamiento galvánico entre la entrada, la salida y la alimentación.



Figura 0.11: Conversor de Voltaje/Corriente. [6]

CARACTERÍSTICAS:

- Precisión : <0,2% escala completa
- Linealidad : <0,1% escala completa
- Tiempo de respuesta (modelos de salida de señal Vdc): < 70 mseg. (90% de la señal) a 20Hz -3dB

- Tiempo de respuesta (modelos de entrada de señal Vac) : <250 mseg. (90% de la señal).
- Impedancia de entrada: 1 M Ω para rangos de < 1 V, 10 M Ω para rangos de >1v
- Protección por encima del rango: 1.000v para rangos superiores a 100v, 500v para rangos inferiores o iguales a 100v [5].

1.5.3. RESISTENCIAS DE PRECISIÓN

El juego de resistores de corriente SCXI es un paquete de cuatro resistores de precisión para medidas de entradas de corriente de 0-20 mA y 4-20 mA. El juego incluye cuatro resistores de alta precisión de 249 Ohm, 0.1%, 5 ppm, 0.25 W. Se las muestra en la Figura 0.12.



Figura 0.12: Resistencias de precisión de 249 ohms. [7]

CAPÍTULO 2

METODOLOGÍA DE IDENTIFICACIÓN DEL SISTEMA

En este capítulo se describe la metodología propuesta e implementada para obtener el modelo por ecuaciones en espacio de estados de la planta de flujo. Este modelo es usado como elemento importante del control predictivo que predecirá respuestas de salida de la planta frente a ciertas entradas. Al final del capítulo se presenta el modelo de la planta al aplicar la metodología.

La propuesta metodológica para identificación del sistema se muestra en la Figura 0.1, que parte desde la obtención de los datos necesarios, hasta la validación del modelo obtenido.

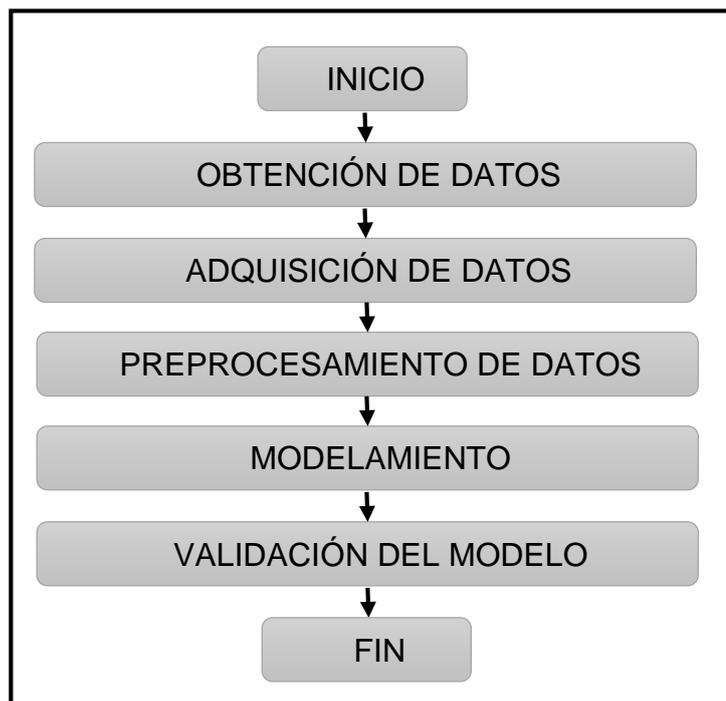


Figura 0.1: Flujograma de la metodología para la identificación de un sistema

Elaborado por: López, V y Paredes, A

La identificación del sistema de la planta de flujo se realiza con la ayuda del software MATLAB y su toolbox *ident* para obtener el modelado de la planta en ecuaciones de espacio de estado, ya que el tipo de expresión matemática

es utilizada en control predictivo con el software LABVIEW, donde se implementará el control.

El experimento para realizar la identificación del sistema consiste en medir la respuesta del proceso, sometido a las entradas, es decir, se lo hace a lazo abierto, sin ningún tipo de realimentación o control.

1.6. ADQUISICIÓN DE DATOS

El modelado mediante identificación se realiza de manera experimental, conociendo la salida de respuesta del sistema a una excitación en la entrada. Por lo que el primer paso para realizar el modelado con esta técnica, es la obtención de datos de entrada y salida del proceso.

De estos datos depende todo el modelo y las etapas posteriores para determinarlo, por lo que es necesario considerar: las señales a medir, tiempo muerto del sistema, dinámica del proceso, frecuencia de muestreo, rangos útiles, tipo de entrada y el número de muestras a tomar. El control predictivo al tener como principal característica usar un modelo matemático para predecir las señales futuras, de no usar un modelo adecuado, en vez de tener un controlador óptimo, se tendrá un controlador inestable o incluso menos eficiente que otros controles más sencillos.

1.6.1. SEÑALES A MEDIR

Como primera consideración, se determina las señales que influyen en el sistema y que se pueda cuantificar, en caso de no poder cuantificarlo, se considera como perturbación externa.

El controlador predictivo al tener la capacidad de controlar sistemas de multiple entrada y multiple salida (MIMO), puede manejar varias entradas que influyan en el comportamiento de la o las salidas del sistema, maximizando el rendimiento del controlador.

Para la planta de flujo en este estudio, se utiliza un sistema de entrada simple y salida simple (SISO), ya que son las únicas variables que se puede medir en el proceso; como entrada se tiene la señal de corriente del actuador de 4 – 20 mA, que ingresa al variador de velocidad para controlar

la bomba y ésta a su vez la variable a controlar (flujo); y como salida se tiene la señal del transmisor de presión diferencial de 4 – 20 mA (configurado entre 7 – 45 inH₂O), que representa el caudal de agua del proceso. Cualquier otra entrada al sistema se considera perturbación externa para este experimento. En la Figura 0.2, se muestra una representación de las entradas y salidas con sus límites de magnitud.

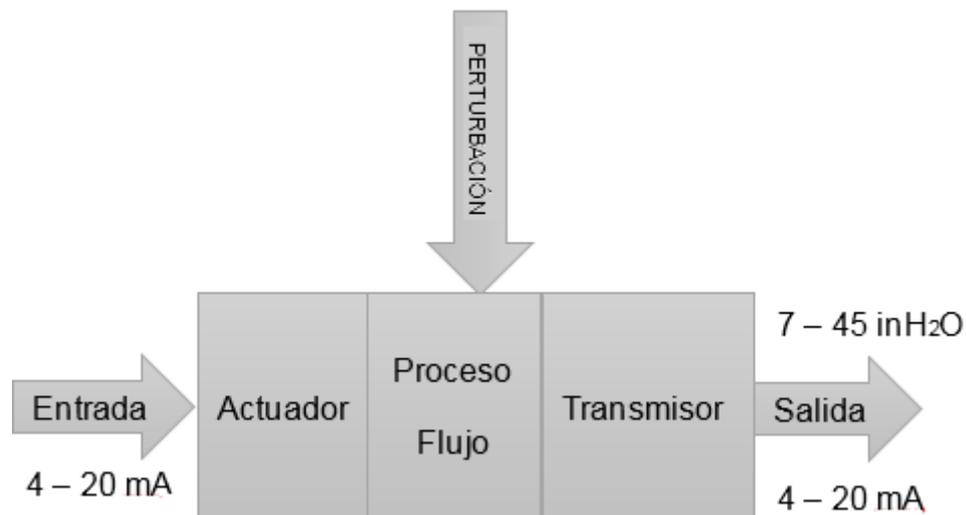


Figura 0.2: Representación de entradas y salidas de la planta de flujo

Elaborado por: López, V y Paredes, A

1.6.2. TIEMPO MUERTO DE RESPUESTA DEL SISTEMA

El tiempo muerto de respuesta del sistema para este experimento se considera el tiempo que la salida del sistema permanece sin variación desde que se aplicó un cambio en la entrada del mismo, es decir, el tiempo que le toma reaccionar al proceso desde que hubo un cambio en el actuador.

Este tiempo es un parámetro muy importante a considerar, pues de este valor dependerá el resto del experimento así como el diseño del controlador mismo, ya que nos da el limitante en cuanto a velocidad de control, pues cualquier cambio en el actuador con tiempos menores al tiempo muerto no afectará la salida, por lo tanto la acción de control sería innecesaria y nada práctica.

Se realiza cuatro cambios en tres puntos de operación, los cuales son: punto inicial (4 mA), punto medio (12 mA) y punto final (20 mA); el tiempo muerto se considera el mayor del conjunto (menor frecuencia) de la Tabla 0.1.

Los gráficos de cada uno de los experimentos se indican en el **¡Error! No se encuentra el origen de la referencia.**

Tabla 0.1. Tiempos muertos para diferentes cambios de entrada

Valor inicial (mA)	Valor final (mA)	Tiempo muerto (s)
4	12	1,8
12	20	0,56
20	12	0,75
12	4	0,5
Máximo		1,8

1.6.3. DINÁMICA DEL SISTEMA

La dinámica del sistema se considera como el cambio de la salida del sistema a través del tiempo, sin que haya existido un cambio en la señal de entrada, es decir, las variaciones de la salida en estado estable.

De la Figura 0.3, se puede deducir que la salida o respuesta del sistema es altamente dinámica en el tiempo, y por las limitaciones del tiempo muerto determinado anteriormente, se concluye que no es posible controlar esas variaciones, pues son cambios más rápidos que la respuesta del sistema a una entrada, por tanto, se considera ruido en el proceso, que se lo va a corregir mediante filtros en el software.

El filtro a diseñar, es un filtro pasa-bajo, y la frecuencia de corte se toma de acuerdo a la frecuencia mínima de la señal de salida del proceso de flujo. Para determinarla, usaremos como limitación el tiempo muerto máximo del proceso (Tabla 0.1), y la frecuencia será el inverso de dicho período de tiempo.

De la Ecuación (0.3), se determina la frecuencia de corte en aproximadamente 0,5 Hz.

$$f_{min} = \frac{1}{T_{max}} = \frac{1}{1.8 \text{ s}} = 0,55 \text{ Hz.} \quad (0.3)$$

Con el uso de la herramienta *ident* del software MATLAB, se realiza un análisis en frecuencia de la señal, para validar que la frecuencia de corte es la adecuada para el proceso; en la Figura 0.4, se aprecia la estabilidad del proceso para frecuencias menores a 0,5 Hz, y la variabilidad que va adquiriendo a medida que ésta aumenta, validando así la frecuencia de corte del filtro.

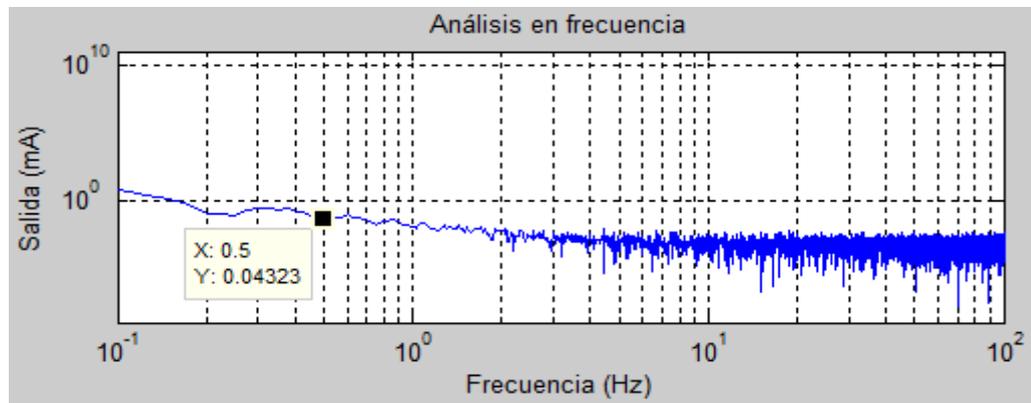


Figura 0.4: Análisis en frecuencia de la salida del proceso de flujo

Elaborado por: López, V y Paredes, A

Al aplicar el filtro en 0,5 Hz y aplicando el mismo análisis en frecuencia, se puede observar que se ha corregido el ruido, el resultado se muestra en la Figura 0.5.

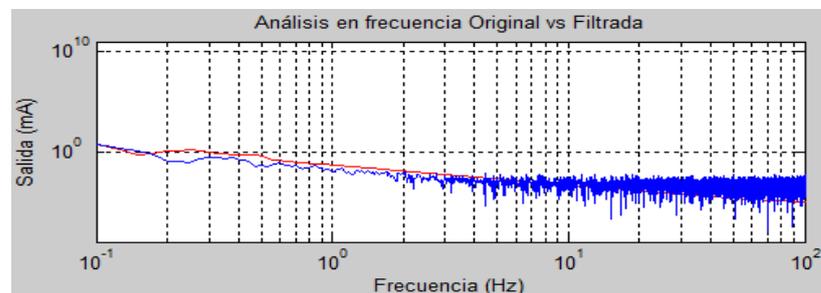


Figura 0.5: Análisis en frecuencia, señal de salida original vs filtrada.

Elaborado por: López, V y Paredes, A

En la Figura 0.6 se muestra las señales en el tiempo; se puede apreciar un pequeño desfase de la señal filtrada respecto a la original, ya que no

afecta al comportamiento final de la señal se desprecia dicho atraso. El hecho de que la señal filtrada empiece en cero se da únicamente al inicio ($t=0$), debido a las condiciones iniciales, del mismo modo no se toma en cuenta ya que solo será al iniciar el proceso.

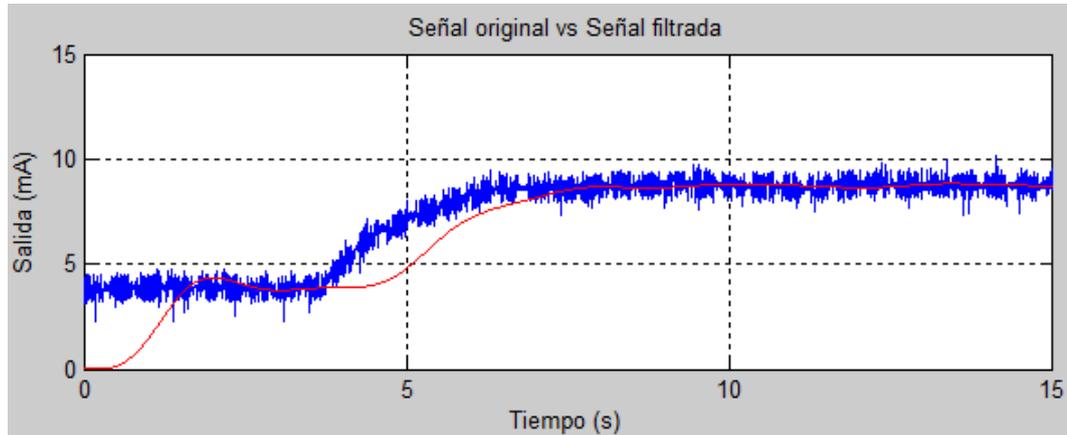


Figura 0.6: Señal de salida en el tiempo, original y filtrada.

Elaborado por: López, V y Paredes, A

Otro método para reducir la variabilidad del proceso es realizar un promedio punto a punto a medida que el tiempo avanza, pero la limitación al utilizar esta técnica es que afecta al tiempo de respuesta del sistema; en la Ecuación (0.4), se puede notar que cada valor (X_i) contribuye al resultado final de manera inversa respecto al número de muestras; a mayor número de muestras (k), menor es la influencia del nuevo valor al promedio total, afectando así la dinámica final del sistema.

$$\bar{X} = \sum_{i=0}^k \frac{X_i}{k} \quad (0.4)$$

Se puede apreciar una comparación entre la señal original y la señal filtrada en la Figura 0.7, notando el retraso que causa el promediar las muestras respecto a utilizar un filtro. El número de muestras ($k=500$), se aproximó hasta tener una estabilidad similar a la del filtro.

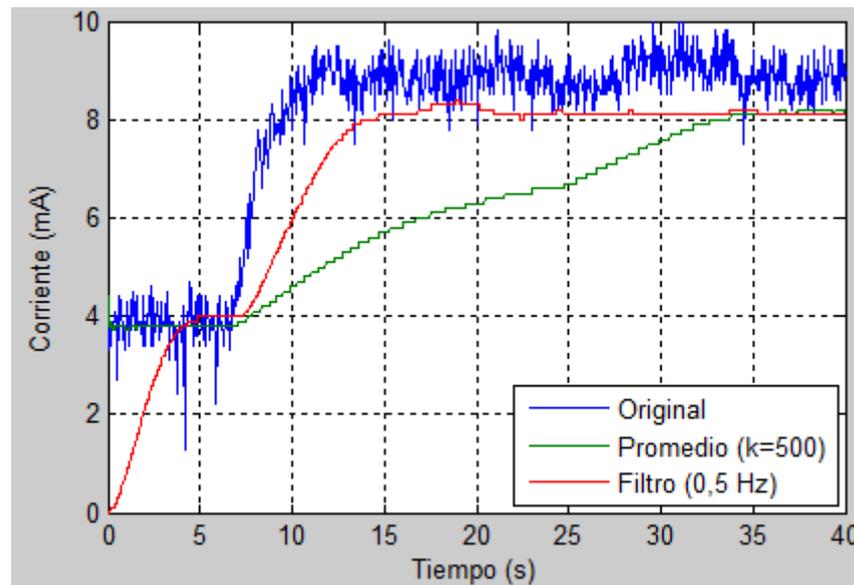


Figura 0.7: Comparación entre la señal original, señal filtrada y promediada.

Elaborado por: López, V y Paredes, A

De aquí en adelante todo el estudio se lo realizará con el filtro ya implementado.

1.6.4. FRECUENCIA DE MUESTREO

La implementación del controlador se lo realizará de manera digital, por tanto; todo el análisis, modelo matemático y adquisición de datos se lo hará de manera discreta, por lo que se determina la frecuencia de muestreo adecuada al proceso de flujo bajo análisis.

Como primera condición se debe cumplir que la frecuencia de muestreo sea el doble de la frecuencia máxima de la señal a discretizar (teorema de muestreo de Nyquist), como se ilustra en la Ecuación (0.5).

$$f_s > 2f_{max} \quad (0.5)$$

En la etapa de filtrado, se diseñó el filtro a 0,5 Hz, por tanto, se toma este valor como frecuencia máxima del sistema, idealmente las frecuencias mayores a ésta deben eliminarse. Se determina en la Ecuación (0.6) que la frecuencia de muestreo debe ser superior a 1 Hz.

$$f_s > 1 \text{ Hz} \quad (0.6)$$

En cuanto a la limitación de la frecuencia máxima de muestro, estrictamente no existe, pues a mayor frecuencia de muestreo, la señal discretizada permanecerá más fiel a la señal original (analógica).

La limitación existe de manera práctica, dado que una frecuencia excesiva de muestreo conlleva a un mayor flujo de datos, velocidad de procesamiento y manejo de memoria, del que es necesario para un proceso en particular, desperdiciando recursos y haciendo poco eficiente al sistema de adquisición de datos.

Como consideración adicional en la identificación de sistemas, si se tiene un tiempo de muestreo demasiado pequeño (frecuencia excesivamente alta), dificulta el proceso de estimación, pues se tienen valores que no aportan significativamente; por lo que es recomendable tomar un valor de frecuencia de muestreo no tan alto respecto a la frecuencia máxima del proceso.

La frecuencia de muestreo usada para el proceso de flujo en este estudio, se establece en cuatro veces la frecuencia de Nyquist, considerando que es un valor adecuado para representar este proceso. La frecuencia de muestreo se determina en la Ecuación (0.7).

$$f_s = 4 * 1 \text{ Hz} = 4 \text{ Hz} \quad (0.7)$$

1.6.5. RANGOS DEL SISTEMA

En este punto se analiza los rangos válidos del sistema, es decir límites inferiores y superiores en las variables de entrada/salida ya sea por restricciones de hardware, software o físicas del sistema.

w. Hardware

La primera limitación se da en el hardware del proceso, como se mencionó en puntos anteriores, tanto la entrada y salida del sistema, está

limitada al rango de 4 – 20 mA. En la entrada los valores corresponden a la limitación dada por el actuador (0 – 100 %) y en la salida es el rango configurado para el transmisor de presión diferencial (7 – 45 inH₂O).

Por tanto todos los rangos a manejar para la identificación del sistema estarán entre los valores de corriente mencionados.

x. Software

Se utiliza el software LabVIEW para la adquisición de datos y el software MATLAB para la identificación del sistema, por tanto, no existe ninguna limitación en cuanto los programas informáticos a utilizar.

y. Restricciones físicas

La salida del sistema no posee restricción, pues el rango de flujo establecido alcanza todo el rango de corriente limitado por hardware.

En cuanto a la entrada, no sucede lo mismo, pues hay un rango inferior que no provoca ningún cambio en la salida del proceso, por tanto, sería innecesario tomar valores dentro de estos límites, pues no aportan nada al sistema y más bien dificultará el proceso de estimación y la identificación del mismo. De igual manera sucede en el rango superior, estos niveles no afectan al sistema pues ya se encuentra saturada la salida (en su máximo valor de 20 mA), por cuanto, son valores irrelevantes para la identificación.

En la Figura 0.8, se puede apreciar el experimento con el cual se determinó el rango superior en 19 mA, el cual consiste en determinar la diferencia entre el rango superior vs el rango inferior.

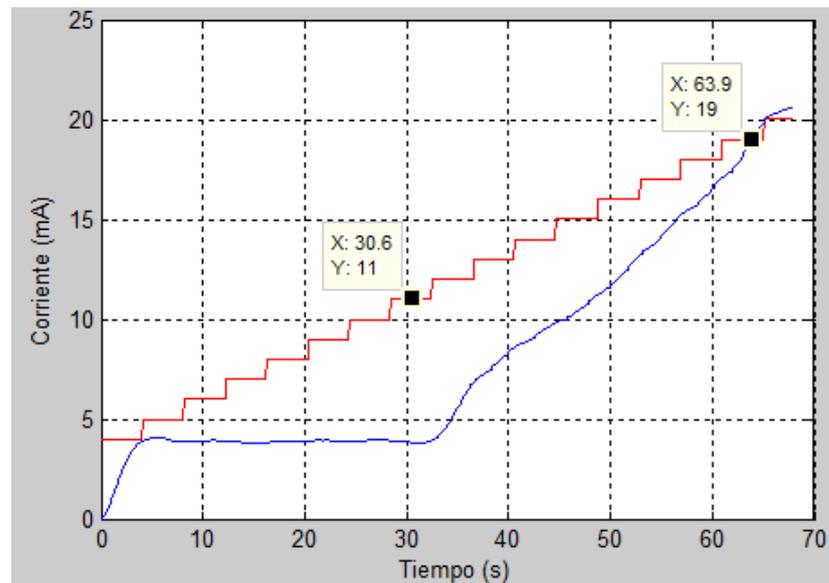


Figura 0.8: Validación del rango superior de entrada

Elaborado por: López, V y Paredes, A

El rango inferior queda definido en 8 mA, se aprecia en la Figura 0.9, que es donde al dar un valor inferior vs a un valor escalón se estabiliza en ese valor.

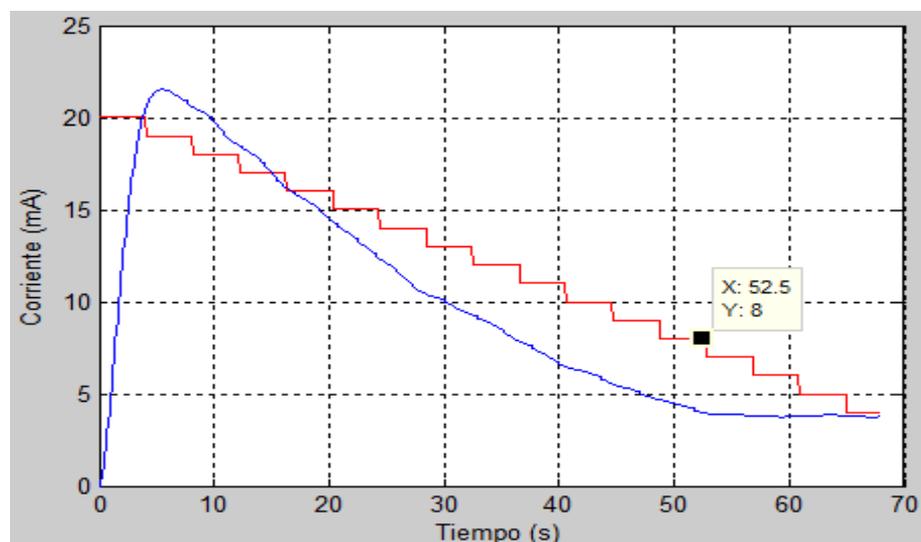


Figura 0.9: Validación del rango inferior de entrada

Elaborado por: López, V y Paredes, A

1.6.6. TIPO DE ENTRADA

El tipo de entrada para excitar al sistema y obtener la respuesta debe contener la mayor cantidad de frecuencias posibles, teniendo así una

respuesta con un ancho de banda amplio, de manera que el modelo estimado sea lo más preciso al real.

Se puede aplicar una entrada con señales de varias frecuencias y obtener la respuesta del sistema para este determinado ancho de banda. Pero esto implicaría emplear cuantiosas señales a distintas frecuencias, limitando también el ancho de banda de respuesta a los rangos de frecuencia aplicados.

De manera práctica entonces, se opta por aplicar señales escalón (cambios súbitos) en la entrada, pues, matemáticamente este tipo de señal contiene un ancho de banda muy amplio, por lo que resulta conveniente utilizarlo como tipo de entrada, implicando una respuesta del sistema a todas las frecuencias necesarias para describirlo completamente.

Suele utilizarse dos tipos de señales escalón: binaria como la mostrada en la Figura 0.10, que cambia únicamente entre dos valores, típicamente entre el límite superior e inferior, utilizada para identificar sistemas que tengan características lineales; y la multinivel como la indicada en la Figura 0.11, aquella que cambia entre niveles de todo el rango de la señal, usada para sistemas no lineales.

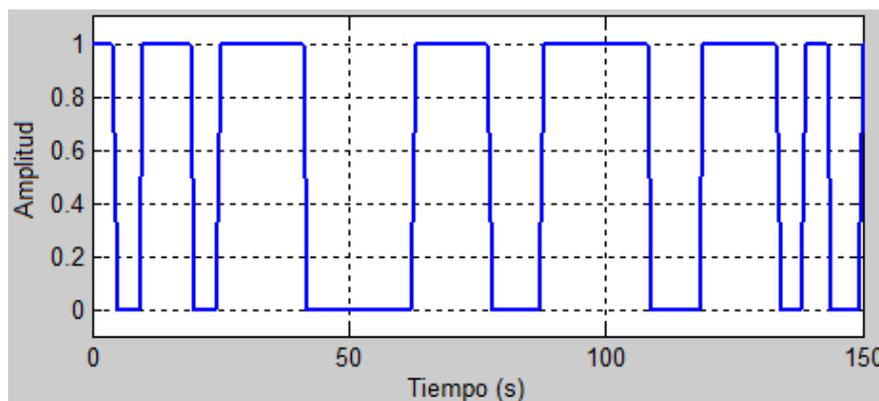


Figura 0.10: Señal escalón binaria

Elaborado por: López, V y Paredes, A

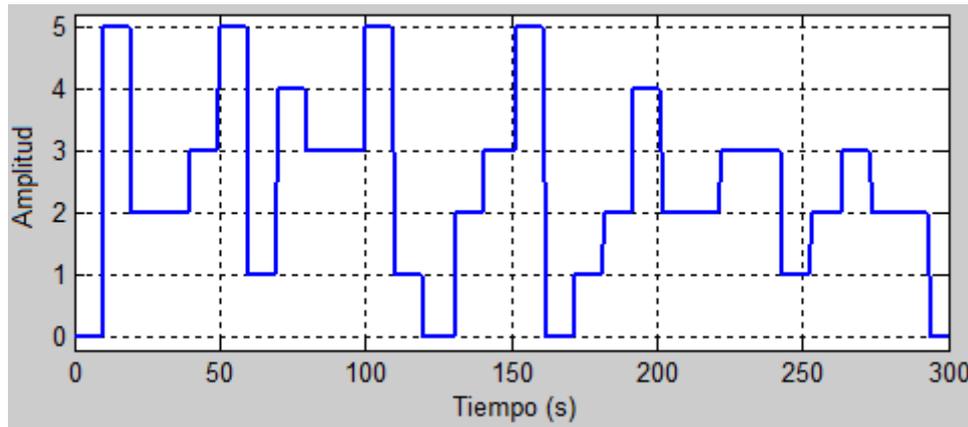


Figura 0.11: Señal escalón de cinco niveles

Elaborado por: López, V y Paredes, A

Para este proceso de flujo en particular, se conoce que es no lineal, por tanto, se realiza la adquisición de datos con la señal escalón de varios niveles.

Para este tipo de señales, se modifica el tiempo de cada pulso de manera aleatoria, para así obtener toda la información necesaria para describir el proceso, de otro modo sería equivalente a tener la respuesta al impulso.

Los tiempos mínimo y máximo de los pulsos depende de la dinámica propia del sistema, pulsos muy amplios causarán que el sistema se estabilice y esto dificulta estimar la respuesta estocástica del sistema, al contrario pulsos muy cortos en los que el sistema sea incapaz de reaccionar solo introducirán ruido y no aportarán en la respuesta del sistema, de igual manera afectando de manera negativa en el proceso de identificación.

El tiempo mínimo del pulso para este sistema se toma como el doble del tiempo de muestreo, como la Ecuación (0.8) indica.

$$T_{min} = 2 * \left(\frac{1}{4 \text{ Hz}} \right) = 0,5 \text{ s} \quad (0.8)$$

Para el período más amplio de los pulsos, se determinará el tiempo máximo de respuesta del sistema a una entrada escalón desde el valor mínimo al máximo en la entrada (8 mA – 19 mA). El experimento se puede

observar en la Figura 0.12, así como el tiempo aproximado de respuesta de siete segundos aproximadamente.

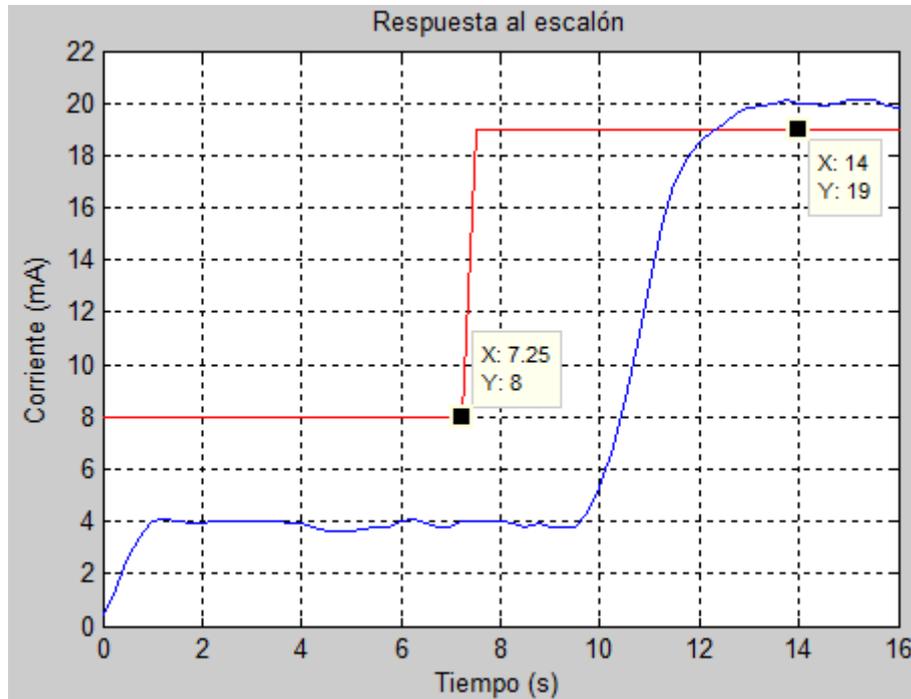


Figura 0.12: Respuesta del sistema a la entrada escalón

Elaborado por: López, V y Paredes, A

Por tanto el rango de tiempo aleatorio para cada pulso estará comprendido entre 0.5 y 7 segundos, determinando así todo el rango de respuesta del sistema, desde su mínima variación hasta su estabilización en 7 segundos.

1.6.7. NÚMERO DE MUESTRAS

De la cantidad de muestras a tomar, depende la cantidad de información para identificar el sistema, si bien un gran número de muestras nos proporciona más información del proceso, se requerirá mayor procesamiento y memoria para analizarlos, siendo incluso información repetida e innecesaria. Por tanto, se debe elegir adecuadamente esta cantidad.

Dado que los tiempos de los pulsos son aleatorios, como se mencionó anteriormente, es imposible saber cuántas variaciones vamos a generar en

un determinado tiempo o una cantidad específica de muestras, por tanto se considera el valor del pulso más prolongado de siete segundos, y se aseguran cien cambios como mínimo, puesto que habrá tiempos inferiores a éste, teniendo más cambios y por tanto más información, pero matemáticamente se asegura que habrá al menos cien variaciones en la entrada.

A partir de estos datos, se calcula el número de muestras como se indica en la Ecuación (0.9)

$$T = (7 \text{ s.}) * 100 * f_s = (700 \text{ s.}) * 4 \text{ Hz} = 2800 \text{ muestras} \quad (0.9)$$

1.6.8. ADQUISICIÓN DE DATOS

Una vez definidos los puntos anteriores, se diseña nuestro sistema de adquisición de datos, tanto en hardware y software para este procedimiento se utiliza productos de National Instruments.

Para el hardware se utiliza un dispositivo MyRIO, un computador, para almacenamiento debido a que la planta de flujo opera con corriente y la MyRIO únicamente con voltaje, adicionalmente se usa un conversor voltaje – corriente, y una resistencia de precisión de 249 ohmios, para convertir de I/V como se aprecia en el diagrama de bloques de la Figura 0.13.

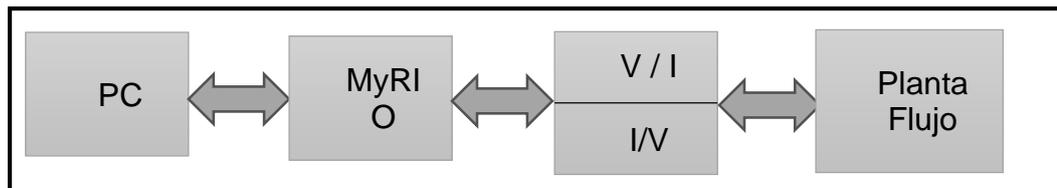


Figura 0.13: Diagrama de bloques del hardware para adquisición

Elaborado por: López, V y Paredes, A

En cuanto a software se realiza un proyecto en el software LABVIEW, con la estructura indicada en la Figura 0.14.

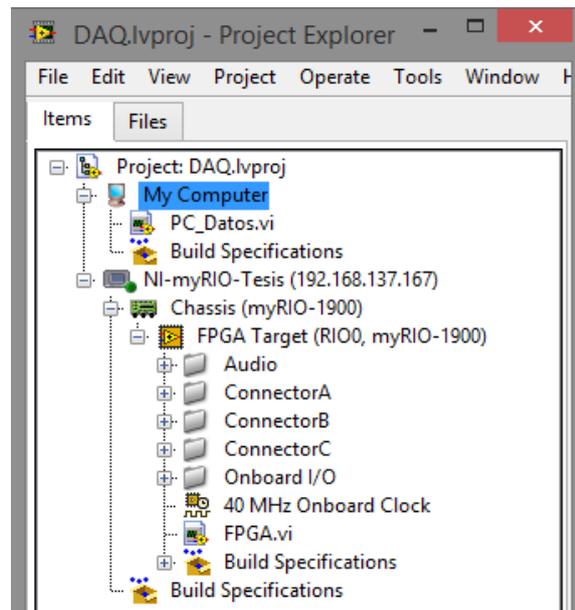


Figura 0.14: Estructura del proyecto de adquisición de datos en LabVIEW

Elaborado por: López, V y Paredes, A

La conexión entre el computador y la myRIO se realiza bajo la arquitectura inalámbrica WiFi. Como se observa en la Figura 0.14, el proyecto consta de dos VI's, la toma de datos por FPGA y el almacenado mediante el computador.

En la Figura 0.15, se muestra el panel frontal y el diagrama de bloques respectivamente usado en la FPGA para la adquisición de datos, se ve que únicamente se toma los datos en la entrada analógica AIO, y se envía voltaje por la salida analógica AOO; ambas ubicadas en el conector C del dispositivo myRIO.

La estructura completa y disposición de las entradas/salidas del dispositivo myRIO se puede consultar en el manual, en la página de National Instruments¹.

¹ www.ni.com/pdf/manuals/376047a.pdf

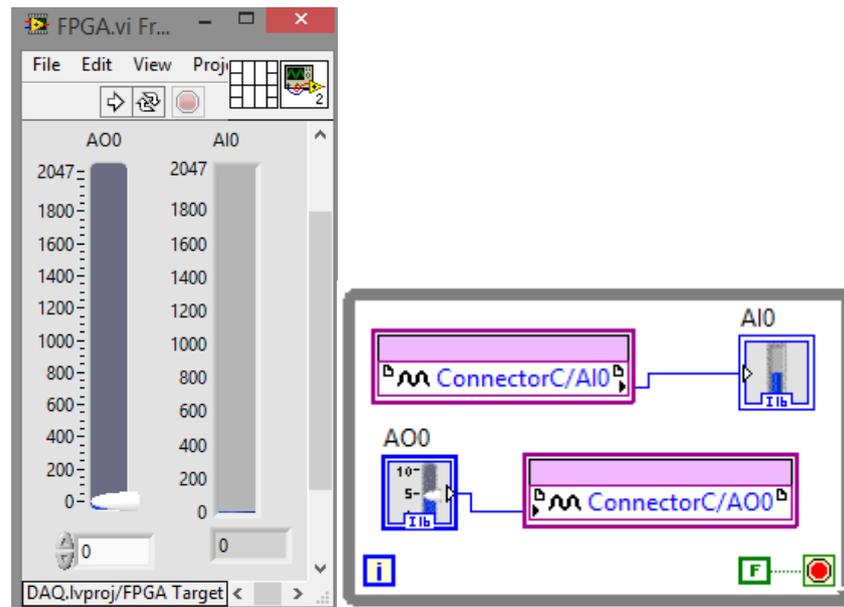
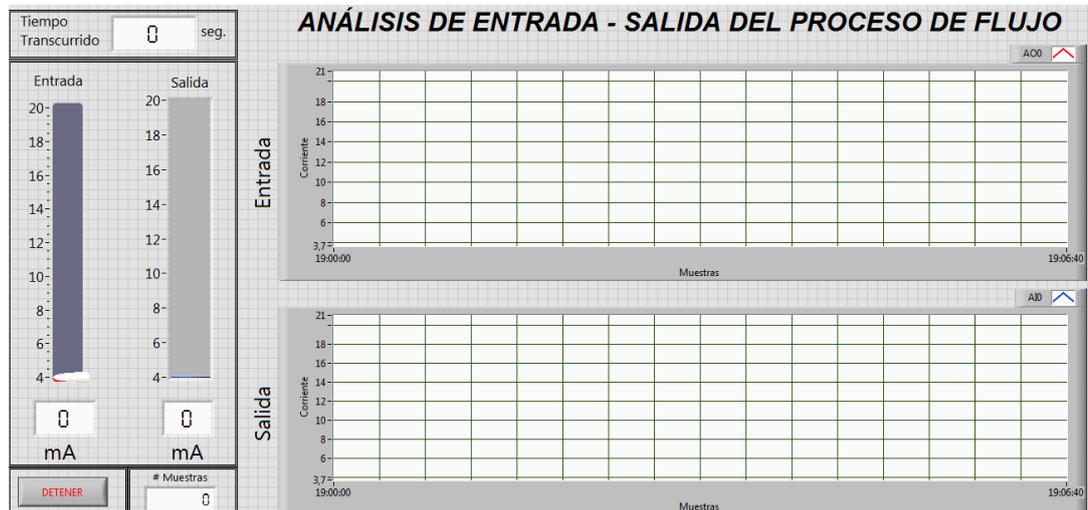


Figura 0.15: Panel frontal y diagrama de bloques del VI en la FPGA

Elaborado por: López, V y Paredes, A

El programa utilizado para almacenar los datos se puede observar en la Figura 0.16, tanto su panel frontal como su diagrama de bloques respectivamente.



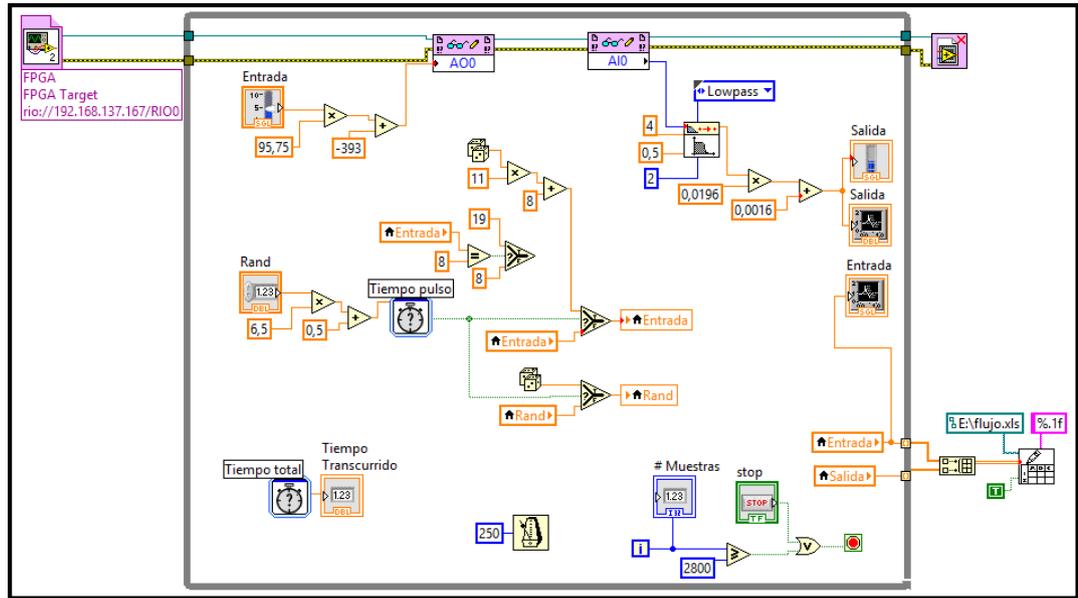


Figura 0.16: Panel Frontal y diagrama de bloques del VI en la PC

Elaborado por: López, V y Paredes, A

1.7. PREPROCESAMIENTO DE LOS DATOS

Después de tomar las consideraciones anteriores, y una vez obtenidos los datos de entrada – salida, se debe verificar que sean adecuados para la identificación del sistema, pues cualquier error en la recolección de datos influirá en el modelo del proceso. Por tanto, se debe corregir antes de utilizarlos para identificar el sistema.

1.7.1. DATOS ERRÓNEOS

Son datos ajenos al proceso, que pueden haber sido causados por errores en el hardware y/o software del sistema de adquisición de datos, o a su vez por agentes externos. Al ser datos que no aportan información sobre el proceso, son considerados *datos basura* y pueden ser fácilmente eliminados del experimento.

Si el dato se encuentra en la mitad de dos, se reemplaza este valor por el promedio de las muestras anterior y siguiente a ésta.

Para este estudio en particular, se utiliza el software LabVIEW para la adquisición de datos, al guardar la matriz de datos (entrada – salida), el

primer dato generalmente suele ser erróneo, pues se almacena el valor antes de medirlo, siendo residuos de un experimento anterior o basura en la memoria, por lo que siempre se descarta la primera muestra de cada experimento.

De igual manera, al importar estos datos al software MATLAB, el último dato suele contener el valor de cero, puesto que primero se crea una variable con la asignación a cero, por lo que después de importar al *workspace* de MATLAB se eliminó el último valor.

Una vez verificados que los valores obtenidos son correctos, se puede proceder con el proceso de identificación.

1.7.2. TIEMPO DE MUESTREO INADECUADO

Como se indicó previamente, el tiempo de muestreo es muy importante para la correcta identificación del sistema, en caso de haberse elegido un valor inadecuado y sea poco práctico o no se logre repetir el experimento, se puede realizar una corrección del tiempo de muestreo de manera matemática, aunque cabe recalcar que no dará los mismo resultados que repetir el experimento con el tiempo de muestreo correcto, puesto que será una aproximación matemática, por lo que se recomienda renovar los datos con un nuevo ensayo. En todo caso *ident* nos ofrece la opción de *Resample* con una sencilla interfaz, la misma que se puede ver en la Figura 0.17.

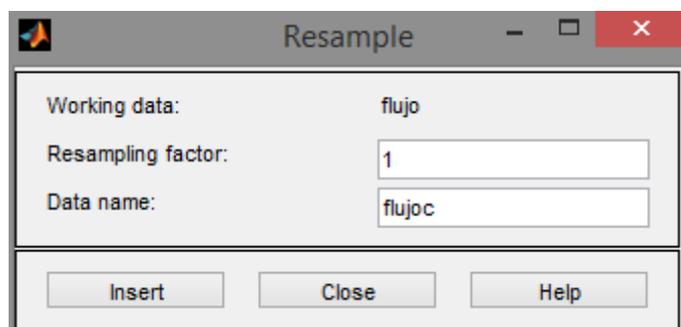


Figura 0.17: Corrección del tiempo de muestro en *ident*

Elaborado por: López, V y Paredes, A

1.8. MODELADO DEL SISTEMA EN MATLAB

1.8.1. IMPORTACIÓN DE LOS DATOS DE ENTRADA-SALIDA

Una vez que se adquirió los datos con las consideraciones anteriores, lo importamos al *workspace* de MATLAB para posteriormente introducirlo en la herramienta *ident* como lo indica la Figura 0.18. Añadiendo nombres, unidades y notas que si bien no son obligatorias, facilitan el entendimiento al trabajar con estos valores.

Figura 0.18: Importación de datos a *ident*

Elaborado por: López, V y Paredes, A

Se puede representar los valores de manera gráfica ya sea en el dominio del tiempo o de frecuencia. En la Figura 0.19 se aprecia el experimento representado en el tiempo, se observa la reacción de la salida (flujo), frente a las entradas escalón de corriente que representa la excitación que se le da al sistema.

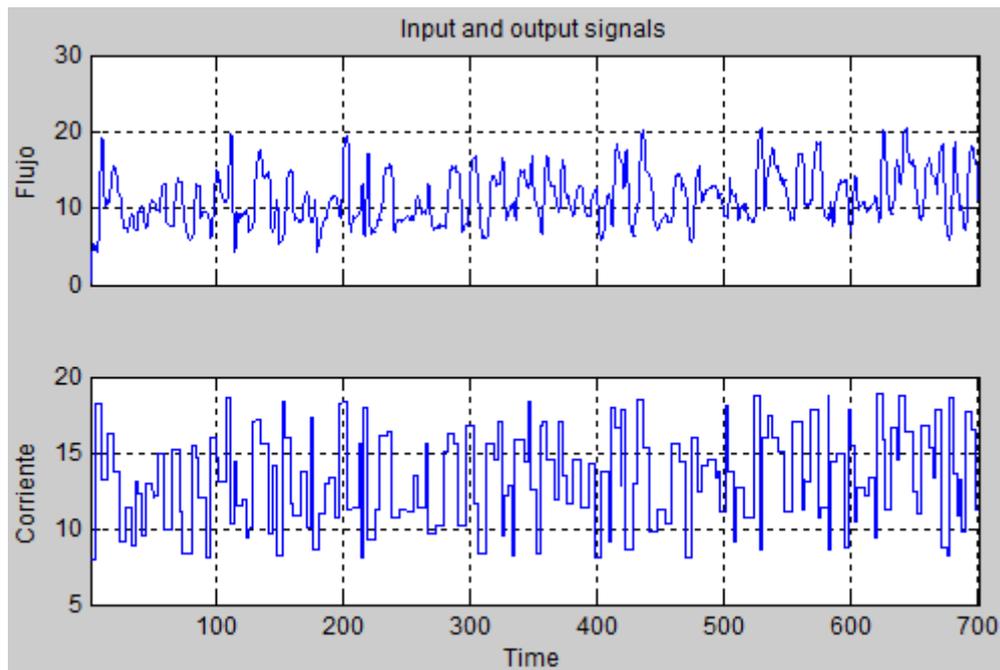


Figura 0.19: Gráfica en el tiempo de entrada-salida

Elaborado por: López, V y Paredes, A

En caso de realizar algún tipo de pre-procesamiento, hacerlo en esta etapa, después de importar los datos y antes de realizar el modelado.

1.8.2. OBTENCIÓN DEL MODELO

La herramienta *ident* ofrece varios tipos de modelos, dependiendo de la aplicación, como ya se dijo, para el control a implementarse se usa el software LabVIEW que utiliza ecuaciones en espacio de estados, por tanto se determinará el modelo directamente bajo esta representación, y a pesar que MATLAB nos ofrece la posibilidad de obtener el modelo de manera discreta tomando en cuenta el tiempo de muestreo, se establece las ecuaciones en tiempo continuo, pues así se tiene un modelo más general y se podrá cambiar el tiempo de muestreo más adelante sin necesidad de volver a determinar el modelo para diferentes tiempos. La ventana para la estimación se puede ver en la Figura 0.20, con las opciones a utilizarse.

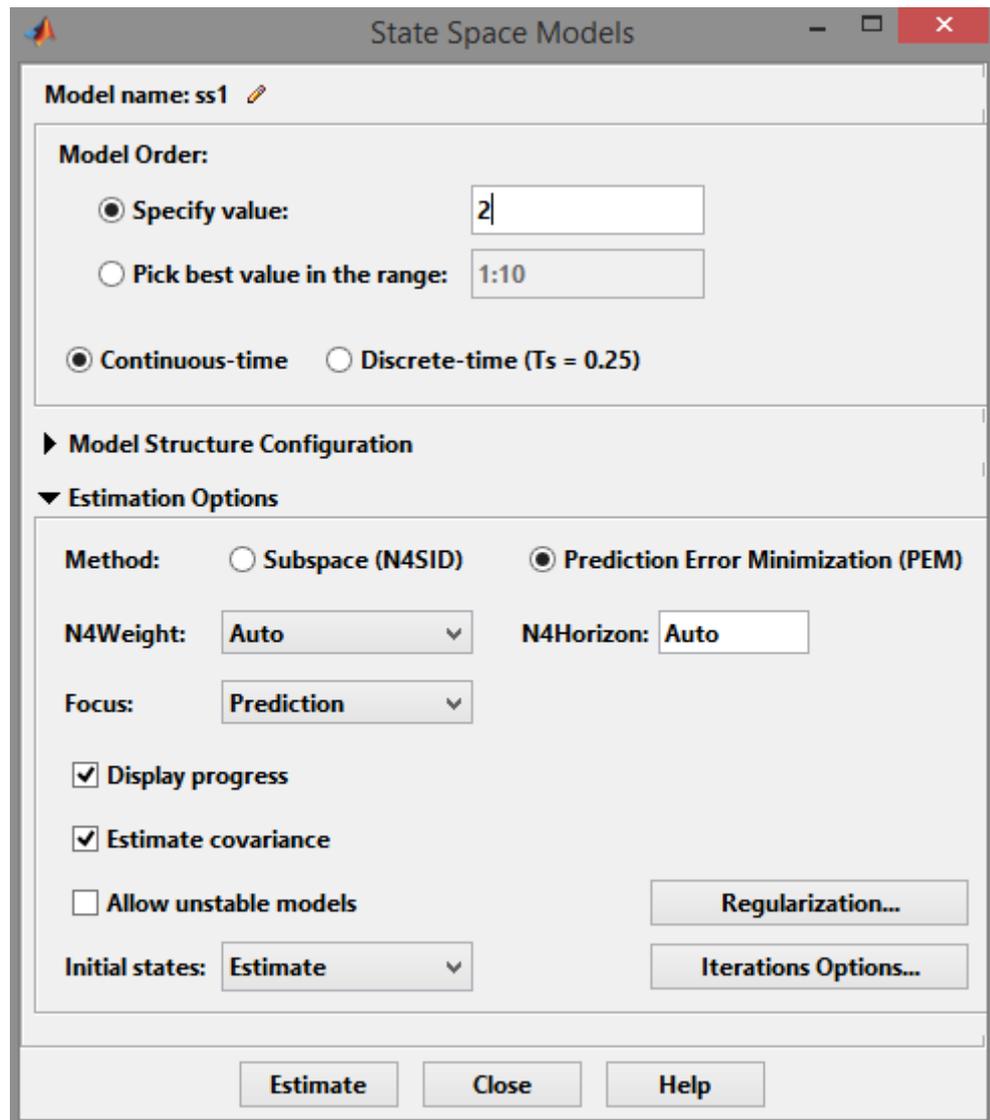


Figura 0.20: Interfaz de modelos en ecuaciones de estado

Elaborado por: López, V y Paredes, A

Una vez obtenido el modelo, puede realizarse varios análisis entre los cuales el principal es *Model Output* que nos permite ver el grado de coincidencia entre la salida real y una simulación con el modelo estimado, indicando además el porcentaje del mismo. Para el ejemplo anterior puede verse el modelo obtenido y su coincidencia en la Figura 0.21, con ampliación en el eje del tiempo.

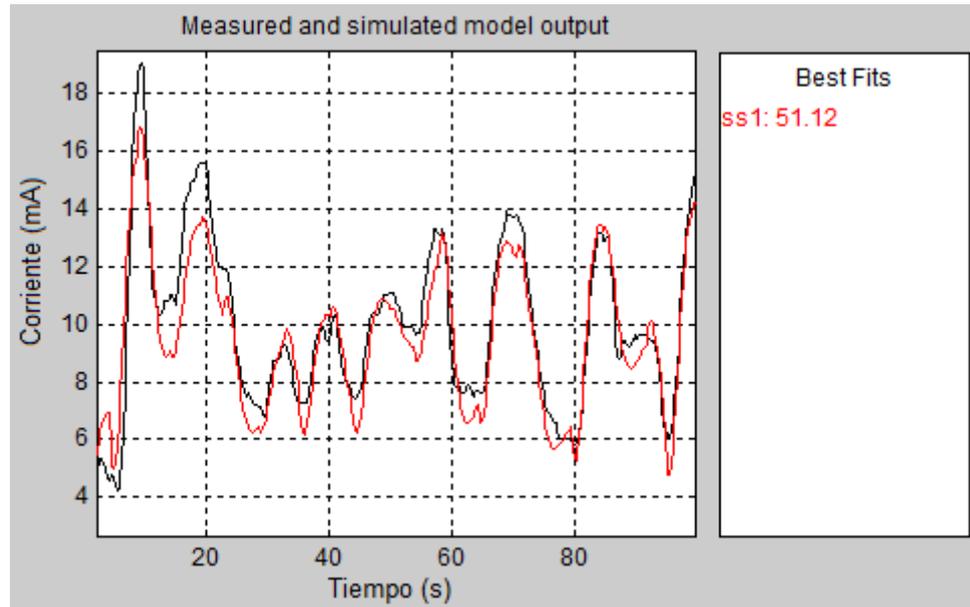


Figura 0.21: Salida real y simulada en *ident*

Elaborado por: López, V y Paredes, A

1.8.3. VALIDACIÓN DEL MODELO

En este punto se verifica la validez del modelo para representar al proceso real, como primer paso se considera el orden del modelo que sea el indicado para el sistema, si bien aumentar el orden suele mejorar el modelo, no siempre es así e incluso un orden muy alto podría reducir la coincidencia, por tanto, se debe elegir un valor adecuado.

Para este proceso en particular, el mejor modelo se obtuvo con un valor de cinco, en la Figura 0.22 se puede observar la comparación entre diferentes modelos que se obtuvo con orden tres a seis, y el mejor grado de coincidencia se da con el valor de cinco.

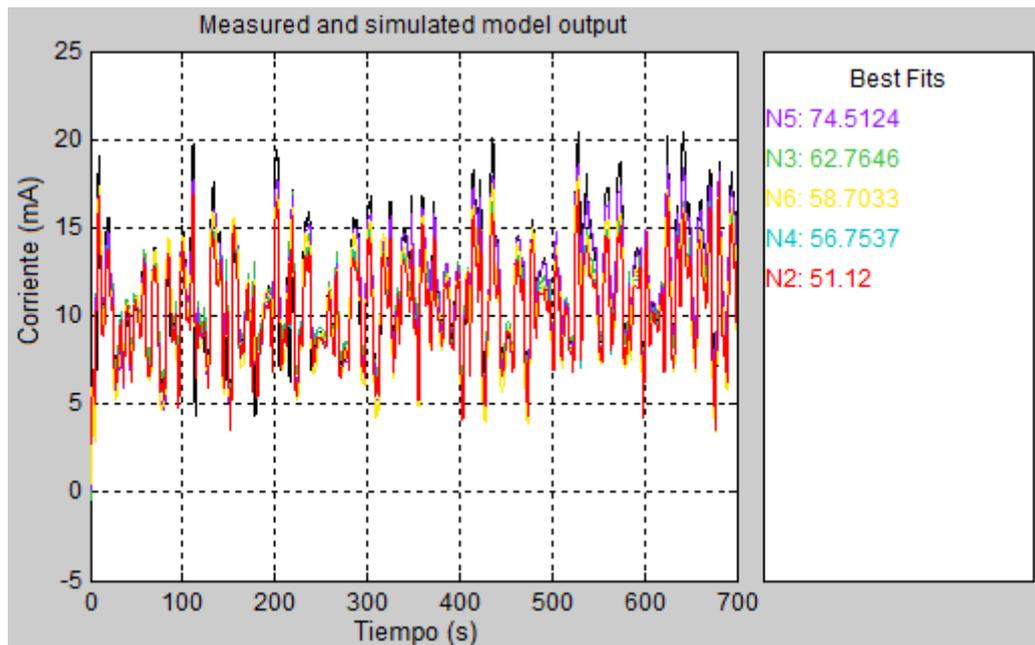


Figura 0.22: Comparación entre modelos por número de estados

Elaborado por: López, V y Paredes, A

Una vez elegido el orden y el respectivo modelo, el siguiente punto es validar su coincidencia para otro tipo de entradas, así nos aseguramos que el modelo es válido para representar el sistema bajo cualquier circunstancia.

Se realiza pruebas con el modelo determinado y si muestra coincidir en porcentajes similares a cualquier tipo de entrada, se acepta como válido. En este estudio como validación adicional se compara varios modelos para diferentes entradas, y el modelo con mayor porcentaje en los experimentos se toma como el más acertado, puesto que demuestra ser el mejor frente a cualquier entrada. La ventana de la herramienta *ident* es como la mostrada en la Figura 0.23, se puede notar que hay tres entradas y cuatro modelos, con los que se validará el modelo elegido.

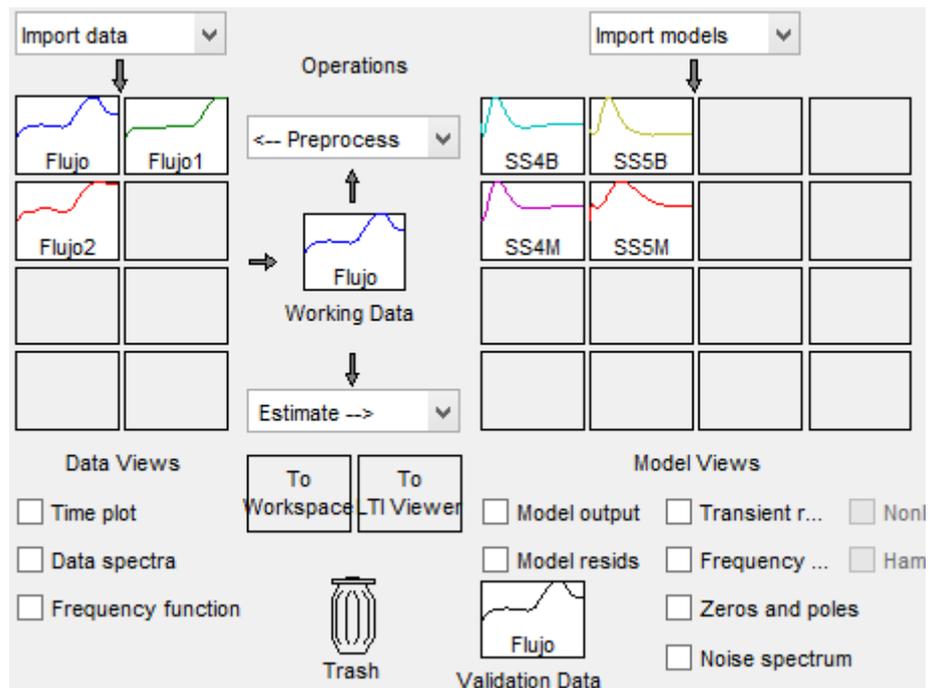


Figura 0.23: Ventana de la herramienta *ident* para validación

Elaborado por: López, V y Paredes, A

Las gráficas independientes de los datos de entrada – salida (Flujo, Flujo1, Flujo2) para la validación se encuentran en el **¡Error! No se encuentra el origen de la referencia..**

Los detalles de cada uno de los modelos (SS4B, SS5B, SS4M, SS5M) se especifican en el **¡Error! No se encuentra el origen de la referencia.**

SS4B es un modelo escalón binarias de orden cuatro.

SS5B resulta similar al anterior pero en orden cinco.

SS4M se obtiene de entradas escalón multinivel de orden cuatro.

SS5M es un modelo con la misma entrada que el anterior en orden cinco.

La primera comparación la realizaremos con los datos denominado “Flujo”, para los cuatro modelos obtenidos. El resultado se muestra en la Figura 0.24.

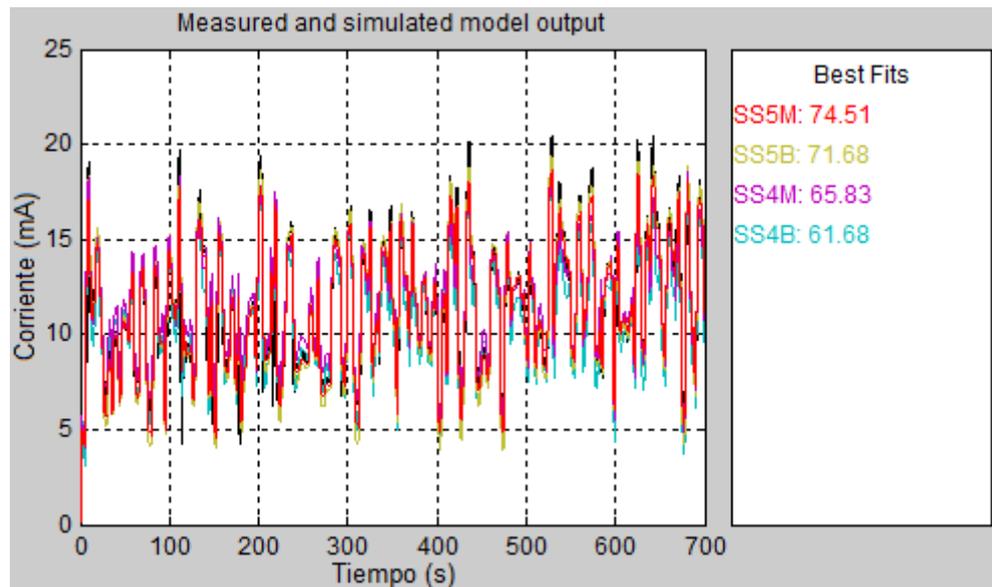


Figura 0.24: Comparación de modelos para los datos “Flujo”

Elaborado por: López, V y Paredes, A

Con el segundo conjunto de datos de entrada – salida llamado “Flujo1” se tienen los resultados expuestos en la Figura 0.25.

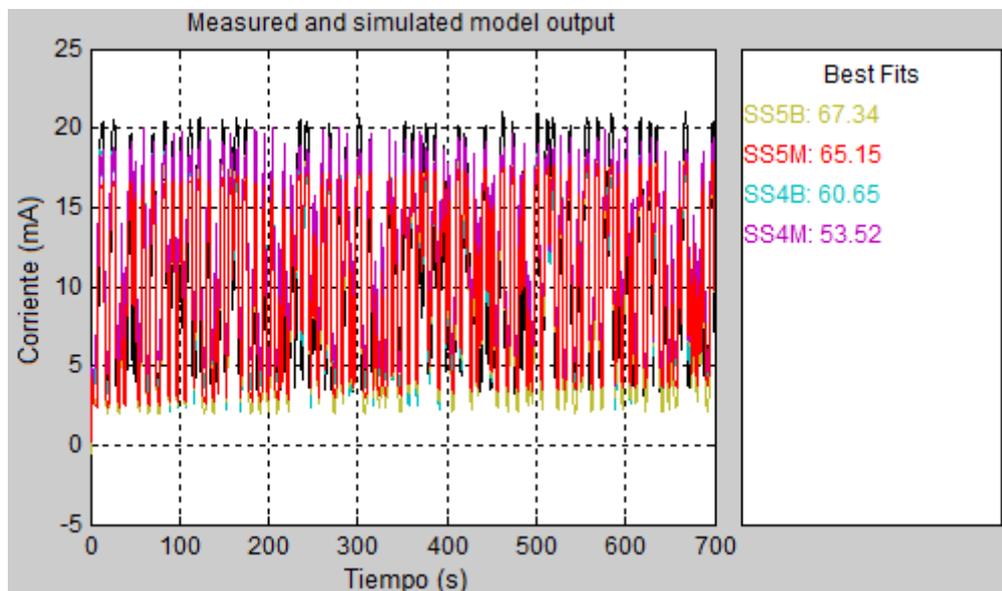


Figura 0.25: Comparación de modelos para los datos “Flujo1”

Elaborado por: López, V y Paredes, A

Finalmente, los resultados para “Flujo2” se indican en la Figura 0.26.

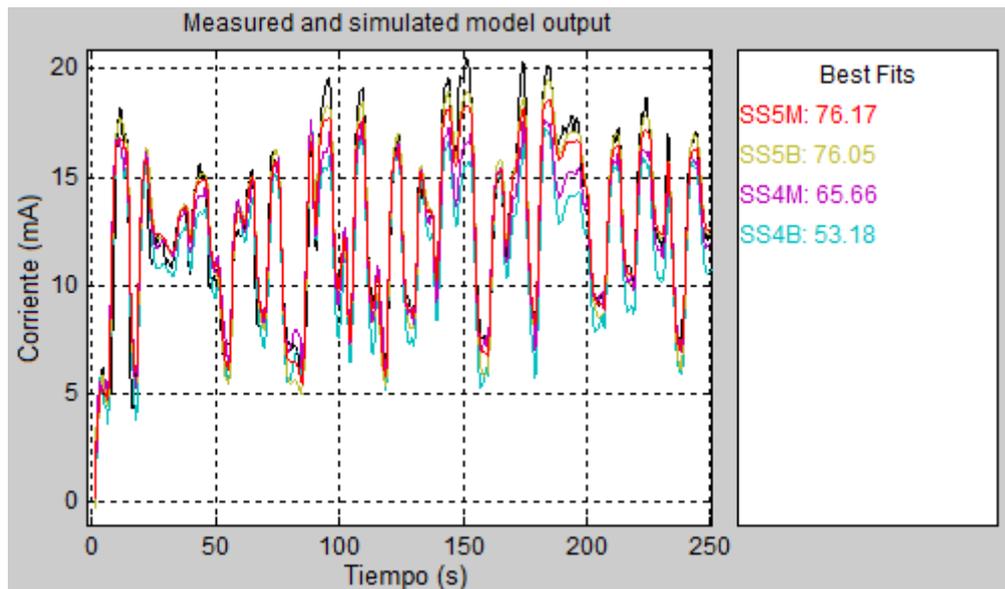


Figura 0.26: Comparación de modelos para los datos “Flujo2”

Elaborado por: López, V y Paredes, A

De las gráficas anteriores se puede determinar que el mejor modelo tanto en porcentaje de asimilación como en repetitividad para diferentes entradas es el modelo estimado con entrada escalón multinivel de orden cinco (número de estados).

El modelo queda determinado en el dominio del tiempo como indica la Ecuación (0.10) (ecuación de estados) y en la Ecuación (0.11) (ecuación de salidas).

$$\frac{dx}{dt} = \begin{bmatrix} -0,7748 & -0,3723 & 0,3336 & 0,3238 & -0,4838 \\ 0,4864 & -1,241 & 0,996 & 0,7281 & -0,8714 \\ 0,2939 & -0,6272 & -1,913 & -3,336 & 2,435 \\ -0,09748 & 0,7894 & 0,3326 & -3,759 & 2,95 \\ -0,2415 & -0,6084 & 0,8691 & 2,112 & -1,815 \end{bmatrix} x(t) \quad (0.10)$$

$$+ \begin{bmatrix} -0,02065 \\ -0,05712 \\ 0,03013 \\ 0,1419 \\ -0,06962 \end{bmatrix} u(t)$$

$$y(t) = [173,9 \quad -26,39 \quad 4,452 \quad 17,45 \quad 12,11] x(t) + [0] u(t) \quad (0.11)$$

En la primera ecuación (2.10) en la expresión $x(t)$, se denomina matriz A y es la matriz de estados, siempre es una matriz cuadrada y dependiendo del orden del modelo varía la matriz, en este caso es de orden cinco.

La expresión $u(t)$ es llamada vector de entrada, se denomina matriz B y representa la relación que tiene las entradas en el sistema. Es un vector de orden cinco por uno ya que tiene salida.

En la segunda ecuación (2.11) la expresión $x(t)$, se denomina vector C, su dimensión es uno por cinco, es la matriz de salida y es la relación de los estados con la salida del sistema.

La expresión $u(t)$ se denomina D es el vector de transmisión directa, generalmente se asume que el sistema no tiene relación directa entre la entrada y salida por eso es cero.

CAPÍTULO 3

DISEÑO Y SIMULACIÓN DEL CONTROLADOR PREDICTIVO BASADO EN MODELOS

En esta etapa se diseña el control predictivo, para ser simulado como primera instancia y a continuación ser implementado. Ambas etapas se basan en el modelo de la planta determinado en el capítulo 2.

El tener un modelo de un sistema real permite que no se haga necesario tener el proceso de manera física para hacer pruebas sobre el mismo, a partir de esto se opta por realizar una simulación del control sobre el proceso utilizando el software LabVIEW, para posteriormente implementarlo sobre el proceso de flujo, programando el algoritmo de control en el dispositivo NI MyRIO.

1.9. DISEÑO DEL CONTROLADOR PREDICTIVO EN LABVIEW

Basado en el diseño expuesto anteriormente programado en Labview, mediante la programación gráfica que ofrece el programa, se empieza por revisar cada uno de los VI's necesarios para la simulación, tanto sus funciones como sus requerimientos. Las cuales se pueden hallar en el diagrama de bloques (no aparecen en el panel frontal), en el *toolkit Control Design & Simulation*, en la Figura 0.1 se ilustra la paleta de funciones.

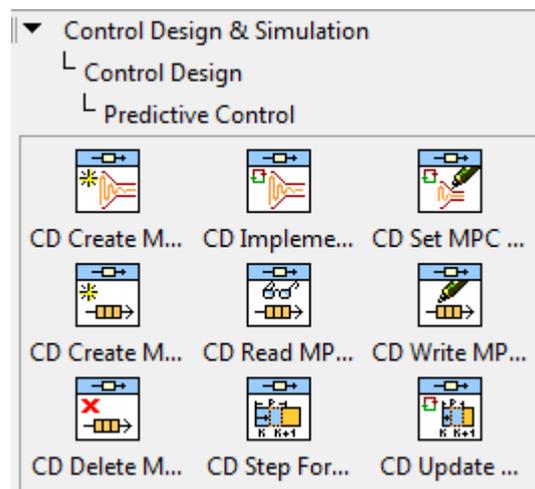


Figura 0.1: Paleta de Control Predictivo en LabVIEW

Elaborado por: López, V y Paredes, A

1.9.1. CREATE MPC CONTROLLER

Esta función permite crear un controlador predictivo basado en modelos para su posterior implementación. Este tipo de controlador requiere de una inicialización con parámetros específicos. En la Figura 0.2 se muestra la función.

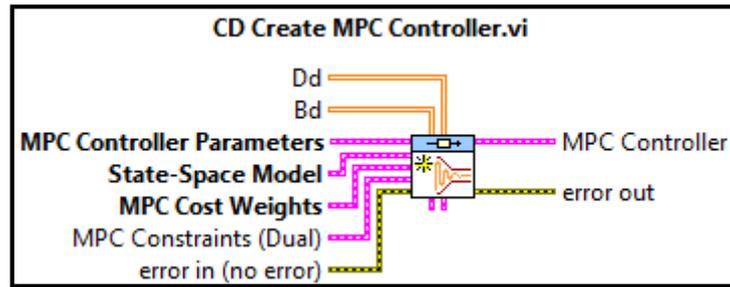


Figura 0.2: Función *Create MPC Controller*

Elaborado por: López, V y Paredes, A

a. MPC Controller Parámetros

Para crear el controlador se requiere de ciertos parámetros básicos como los mostrados en la Figura 0.3, entre ellos tenemos:

Horizonte de predicción, el cual está expresado en número de muestras (el tiempo varía de acuerdo a la frecuencia de muestreo), es decir indica cuántas muestras en adelante se van a predecir la salida del proceso.

Un número pequeño de muestras requiere de menor coste computacional, pero lo acerca a un controlador realimentado tradicional, minimizando la característica de predictivo. Mientras que un horizonte demasiado amplio requiere de mayor cantidad de cálculos, y dependiendo del porcentaje de estimación del modelo con el proceso real la respuesta estimada se aleja de la real, por tanto, las acciones de control calculadas pueden tener un efecto negativo en la salida futura del proceso, puesto que ningún modelo es cien por ciento igual al proceso real.

Un valor adecuado generalmente suele ser un horizonte de predicción cercano al tiempo de estabilización del proceso ante un cambio en la

entrada, es decir, habrá que calcular el número de muestras de acuerdo al tiempo de muestreo, para que coincida con el tiempo de estabilización.

Horizonte de control, al igual que el horizonte de predicción, se encuentra expresado en número de muestras y además indica las acciones de control a ser aplicadas al proceso, es decir, la cantidad de acciones que se calculan y se aplican al sistema en cada cálculo de control. Es evidente que este valor debe ser menor o igual al horizonte de predicción.

Tomando como ejemplo un valor de uno, significa que en cada muestra se calcula la acción de control siguiente y la cual se aplicará al proceso. En cambio, tomando como ejemplo el valor de dos, esto es, calcular dos acciones de control para las siguientes dos muestras, realizando el cálculo cada dos muestras del horizonte de predicción.

El horizonte de control, generalmente es un valor pequeño, pues así se mejora la respuesta instantánea del sistema, con un horizonte demasiado amplio se está controlando a futuro los cambios que probablemente no ocurran como se estima mediante el modelo.

Acción integral, es un valor *booleano* para indicar la presencia o ausencia de acción integral en el controlador, de manera general siempre tener activado esta característica para que el controlador funcione y se pueda corregir la diferencia entre el modelo matemático y el proceso real.

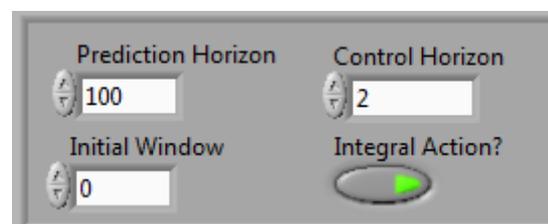


Figura 0.3: Parámetros para el controlador MPC

Elaborado por: López, V y Paredes, A

b. State-Space Model

El modelo en espacio de estados debe ser discreto, especificando el tiempo de muestreo, de acuerdo al modelo indicado el control se crea con las entradas y salidas correspondientes, es decir, aquí se determina si el sistema es SISO o MIMO y cuántas entradas y salidas tiene el proceso.

La entrada necesariamente debe ser en ecuaciones en espacio de estados y en tiempo discreto, de tener otra expresión como función de transferencia, o expresiones en tiempo continuo, LabVIEW nos ofrece herramientas para convertir diferentes modelos a espacio de estados así como también para discretizar modelos que se encuentren definidos de manera continua.

En la Figura 0.4, se muestra como se ingresa las ecuaciones en espacio de estados para crear el controlador.

Model name: Flujo

Sampling Time: 100m

Matrix A:

0	0,93	-0,03	0,03	0,02	-0,03
0	0,05	0,88	0,08	0,04	-0,06
0	0,02	-0,07	0,83	-0,24	0,17
0	-0,01	0,05	0,04	0,71	0,23
0	-0,02	-0,05	0,07	0,15	0,87

Matrix B:

0	-0
0	-0
0	0
0	0,01
0	-0

Matrix C:

0	173,9	-26,39	4,45	17,45	12,11
0					

Matrix D:

0	0
0	

Figura 0.4: Ecuaciones en espacio de estados para el MPC

Elaborado por: López, V y Paredes, A

c. MPC Cost Weights

Las matrices de costo son los valores que permiten optimizar la función objetivo, cada uno de estos valores corresponde a Q, R y N mostrados en la Ecuación (0.20) vista en la sección Función de costo, del primero capítulo. Como se indicó, las dimensiones de cada una de las matrices dependen del número de entradas y salidas del sistema. En la Figura 0.5, se muestran estos parámetros en el panel frontal de LabVIEW.

Output Error Weightings, corresponde a la matriz Q de la función de costo, haciendo una analogía con el controlador PID, sería la constante K_p , es decir, el valor optimiza la señal de control de acuerdo al error entre el valor consigna y el valor medido, cuanto mayor es éste valor, más significativo es el error en la función objetivo. Valores demasiado altos podrían hacer inestable al proceso, pues pequeños errores dan como resultado acciones de control exageradas.

Control Action Change Weightings, es la matriz R de la función objetivo, representa la optimización para la señal de control, es decir, la variabilidad del controlador, a valores altos se tiene una señal de control estable, sin muchas variaciones, lo que ayuda a mantener la vida útil del controlador, además de ahorrar energía en el mismo, con valores inferiores la señal de control es altamente dinámica, lo que brinda una acción de control y por tanto una respuesta más rápida del sistema. De ahí, que se deba elegir el valor adecuado para esta matriz, buscando rapidez en la acción de control a la vez que no se exija mucho esfuerzo al controlador.

Control Action Error Weightings, la matriz N de la optimización, es el valor de ponderación para la diferencia entre el valor consigna y el valor de la acción de control, de ser necesario optimizar este error indicar un valor, caso contrario no implementarlo para evitar que el valor consigna se mezcle con la señal de control, que en la mayoría de casos no es relevante para el control.

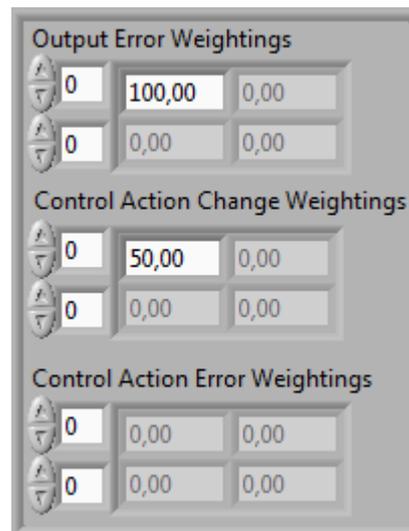


Figura 0.5: Matrices de costo para el controlador predictivo

Elaborado por: López, V y Paredes, A

d.MPC Constraints, consiste en vectores, cuyo número de elementos depende de las entradas y salidas del sistema que representa las restricciones de las mismas, estos parámetros son opcionales, aunque muy necesarios pues es una de las ventajas del controlador predictivo. En la Figura 0.6 se puede observar cada una de las restricciones.

u min – max, son los límites máximo y mínimo de la señal de control, por tanto son las restricciones del actuador, las señales de control calculadas nunca excederán de los valores aquí especificados.

y min – max, son los rangos admisibles para la salida del proceso, el controlador prioriza estos valores frente a otros parámetros incluido el *setpoint*, es decir, las acciones de control se toman para nunca superar estos límites en la salida, a pesar de que el valor consigna indique otro valor. Este parámetro es una de las características del controlador predictivo que otros métodos no poseen, tomar en cuenta los límites reales del proceso.

du min – max, es la pendiente de la señal de control, el valor mínimo corresponde a la pendiente de bajada de control (el control disminuye su valor), por tanto el valor será negativo para que tenga pendiente de bajada, y el valor máximo es cuando el control aumenta su valor, de igual forma será

positiva para ser pendiente de subida. Con estos valores se puede controlar la velocidad de respuesta y el sobre impulso de la señal a controlar.

The image shows a configuration window for an MPC controller. It is divided into two main sections: constraints and stopping criteria.

Constraints:

- u min init:** 0,00 / 0,00
- u min final:** 0,00 / 0,00
- u max init:** 0,00 / 0,00
- u max final:** 20,00 / 0,00
- y min init:** 0,00 / 0,00
- y min final:** 0,00 / 0,00
- y max init:** 0,00 / 0,00
- y max final:** 20,00 / 0,00
- du min init:** 0,00 / 0,00
- du min final:** -0,10 / 0,00
- du max init:** 0,00 / 0,00
- du max final:** 0,10 / 0,00

Stopping Criteria:

- function tolerance: 1E-8
- parameter tolerance: 1E+6
- gradient tolerance: 1E+6
- max iterations: 10000
- max function calls: 10000
- max time (sec): -1

Figura 0.6: Restricciones para el controlador predictivo

Elaborado por: López, V y Paredes, A

1.9.2. IMPLEMENT MPC CONTROLLER

Con el controlador creado, la función que nos permite ejecutar el control es *Implement MPC Controller*, la cual toma el controlador creado, es decir, todas las especificaciones del mismo para realizar el cálculo de la señal de control a ser aplicado a la planta. En la Figura 0.7, se muestra como luce la función en el diagrama de bloques de LabVIEW.

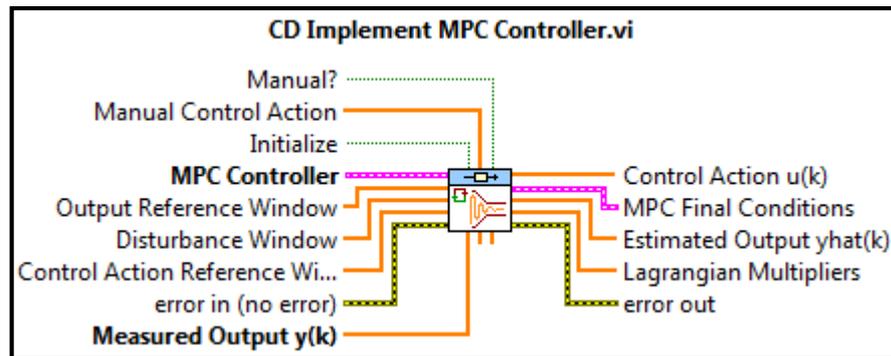


Figura 0.7: Función *Implement MPC Controller*

Elaborado por: López, V y Paredes, A

a. MPC Controller

Es la entrada del controlador creado previamente, obviamente es un requisito obligatorio para la implementación del controlador. No requiere ninguna especificación extra pues simplemente se debe conectar la salida del *Create MPC Controller* a esta entrada.

b. Output Reference Window

Es la ventana que contiene los valores de *setpoint*, un vector que contiene igual número de elementos que el horizonte de predicción ya que las acciones de control se calculan tomando en cuenta la referencia y la salida predicha. Para la simulación e implementación se utilizan métodos diferentes para determinar esta ventana de referencia, que se verá más adelante en cada una de las etapas.

c. Measured Output $y(k)$

Es el *process value* del sistema, la medición de la salida del proceso acondicionada y escalada a los valores con los cuales se haya determinado el modelo matemático (rangos y unidades iguales), no tiene características adicionales, únicamente requiere el valor medido de la salida del sistema.

d. Control Action $u(k)$

Es la acción de control calculada por el MPC, esta salida estará en unidades del modelo matemático y sujeto a las restricciones del controlador, es el valor que va al actuador para corregir la variable a controlar.

1.10. SIMULACIÓN DEL CONTROLADOR MPC PARA LA PLANTA DE FLUJO

Además de las funciones mostradas previamente, para la simulación se requiere de otras adicionales y únicas para esta etapa, que se muestran en esta sección.

1.10.1. GENERATE TIME PROFILES FOR MPC SIMULATION

Esta función facilita la creación de todos los cambios para el *setpoint* durante todo el tiempo de simulación, debido a que no se puede variar en tiempo real estos valores ya que es una simulación la cual mostrará los resultados una vez finalizado la misma, por tanto hay que predefinir los valores consigna. En la Figura 0.8 se muestra la función.

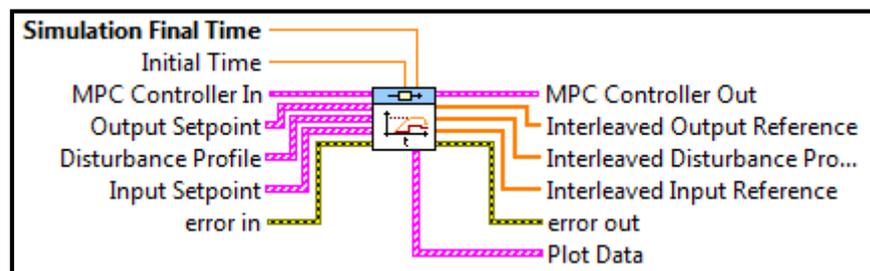


Figura 0.8: Función *Generate Time Profiles*

Elaborado por: López, V y Paredes, A

a. Simulation Final Time

Se indica el valor total de la simulación, en caso de no dar un tiempo inicial (asumiendo como tiempo inicial cero), es un valor de tipo real.

b. Output Setpoint

Se ingresa los valores de *setpoint* junto a los tiempos en los cuales inicia cada uno, en caso de tener más de una entrada se ingresarán dos vectores con las mismas características para cada entrada, en la Figura 0.9 se ilustra como luce los perfiles de los valores consigna con sus respectivos tiempos en el panel frontal.

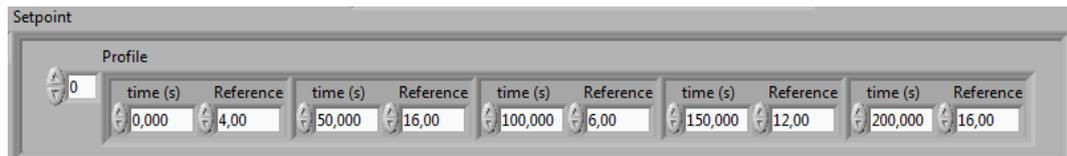


Figura 0.9: Perfiles de *setpoint* en el panel frontal

Elaborado por: López, V y Paredes, A

c. Interleaved Output Reference

Es el resultado total de los perfiles ingresados, con el número de entradas y de cambios en los valores de consigna, es decir se encuentra toda la información del *setpoint* respecto al tiempo total de la simulación.

1.10.2. STEP FORWARD MPC WINDOW

Debido a que se realiza simulación, no se puede aplicar el concepto de horizonte deslizante que maneja el controlador predictivo, de ahí que esta función permite aplicar este concepto utilizando el perfil total de *setpoint* creado anteriormente y de manera progresiva ir actualizando la ventana de referencia que debe ser igual al horizonte de predicción como se mencionó previamente.

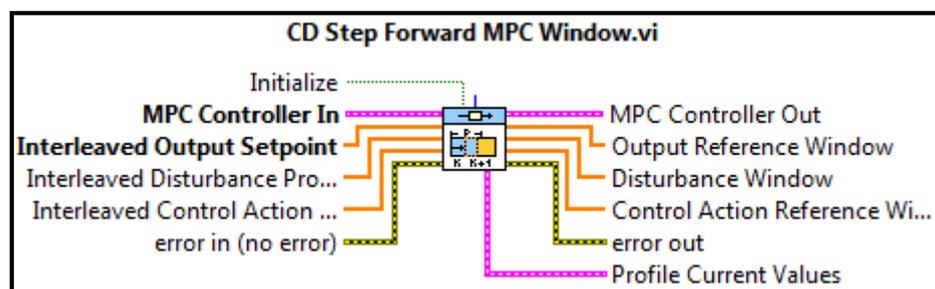


Figura 0.10: Función *Step Forward MPC Window*

Elaborado por: López, V y Paredes, A

a. Interleaved Output Setpoint

Se ingresa la referencia total de los valores de *setpoint* junto a sus valores iniciales de tiempo para toda la simulación.

b. Output Reference Window

Es la ventana de *setpoint* del tamaño del horizonte de predicción, que se irá actualizando a lo largo de la simulación como si fuese un horizonte deslizante a través del tiempo.

1.10.3. SIMULACIÓN DE LA PLANTA

Dentro del *toolkit Control Design* tenemos funciones para simulación, entre ellas, la que nos permite simular una planta que se encuentre definida en ecuaciones en espacio de estados es *Discrete State Space*, el icono en el diagrama de bloques se muestra en la Figura 0.11.

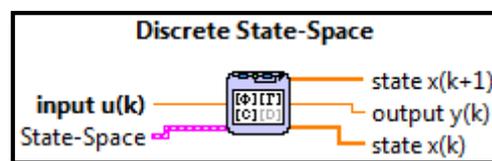


Figura 0.11: *Discrete State Space* en LabVIEW

Elaborado por: López, V y Paredes, A

a. Input $u(k)$

Es la entrada a la planta, en este caso es a dónde va la señal de control, puesto que el actuador se omite ya que el modelo matemático ya viene definido con los valores de corriente del mismo.

b. State-Space

Se ingresa el modelo matemático en ecuaciones en espacio de estados, esta función admite el modelo de manera explícita como entrada o de manera implícita editando las propiedades del mismo, en nuestro caso lo

haremos de forma explícita para que al editar las ecuaciones, el modelo dentro de la planta también se vea alterado.

c. Output $y(k)$

Es la salida del sistema, de igual manera se omite el sensor y los acondicionamientos pues el modelo viene definido con valores de corriente y puede ser usado directamente para la realimentación del controlador.

1.10.4. DIAGRAMA DE BLOQUES DE LA SIMULACIÓN

Detallados todos los elementos a utilizar para la simulación, se realiza la programación en LabVIEW, el diagrama de bloques completo se muestra en la Figura 0.12.

1.10.5. PANEL FRONTAL DE LA SIMULACIÓN

La visualización para el usuario, es decir, el panel frontal se ha dividido en tres pestañas por la gran cantidad de elementos y la limitación de espacio en la pantalla.

d. Modelo

En esta pestaña se muestra el *Control* de la ecuación, donde podremos editar las matrices A, B, C y D, así como el tiempo de muestreo. Y adicionalmente dos visualizadores para las ecuaciones en tiempo continuo y discreto respectivamente. La pestaña resulta como muestra la Figura 0.13.

e. MPC

Se agrupa todo lo relacionado a los parámetros antes vistos para el controlador predictivo, así como el perfil de *setpoint* usados en la simulación. En la Figura 0.14 se puede observar lo mencionado.

f. Gráficas

Se muestran dos gráficas, en la primera se visualiza el *setpoint vs process value*, y en la segunda el *control value*. Estos resultados se mostrarán al finalizar la simulación. Se ilustra la pestaña en la Figura 0.15.

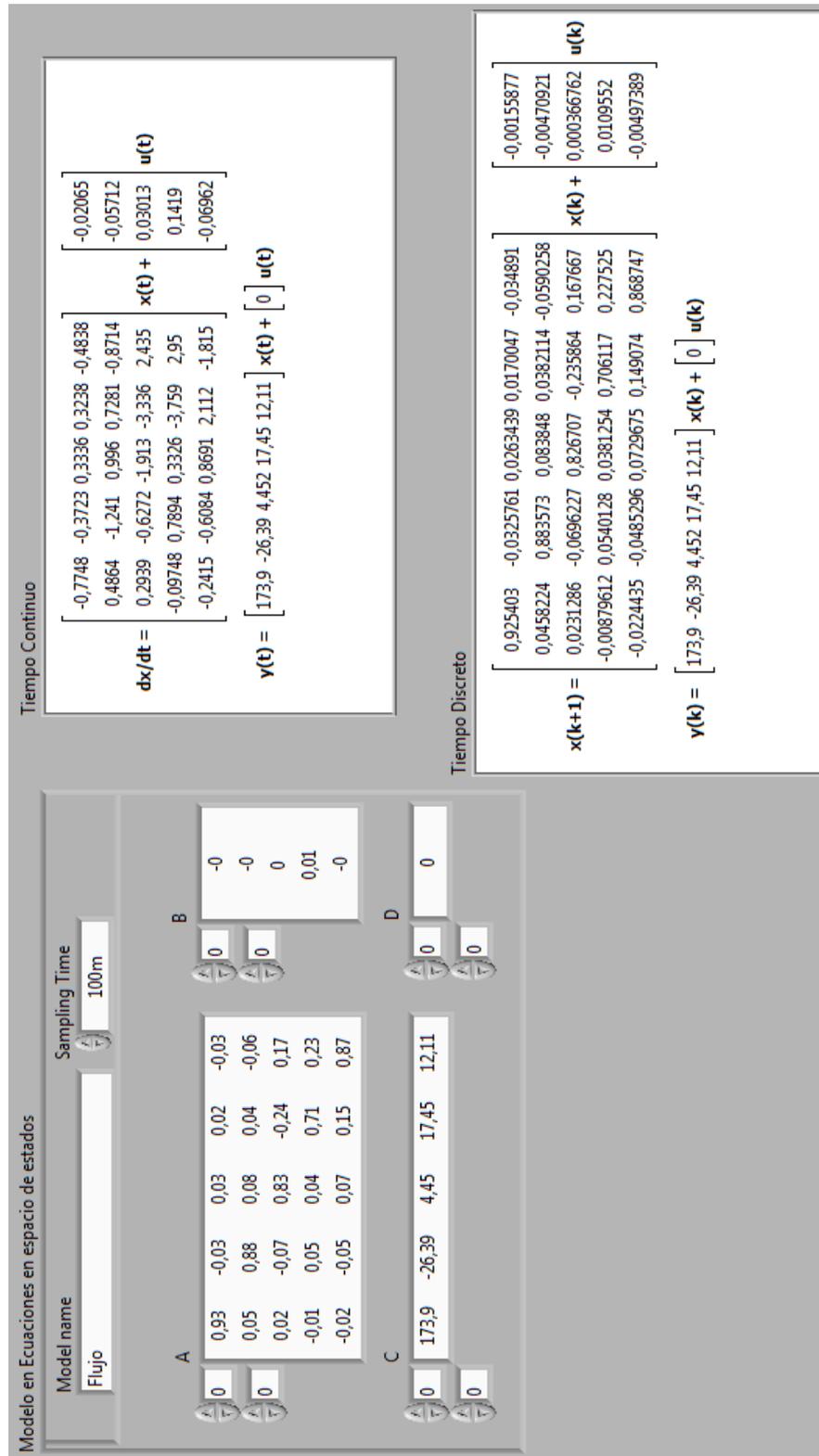


Figura 0.13: Pestaña Modelo en el panel frontal de la simulación

Elaborado por: López, V y Paredes, A

Tiempo de simulación (s) Tiempo de muestreo (s)

Parámetros del Controlador MPC

Prediction Horizon Control Horizon

Initial Window Integral Action?

Restricciones MPC (Dual)

u min init u min final u max final

u max init u max final u max final

y min init y min final y min final

y max init y max final y max final

du min init du min final du min final

du max init du max final du max final

stoping criteria

function tolerance

parameter tolerance

gradient tolerance

max iterations

max function calls

max time (sec)

Pesos de costo (MPC)

Output Error Weightings

Control Action Change Weightings

Control Action Error Weightings

MPC Final Conditions

Control Action

Output (y)

Disturbance (d)

State (x)

Control Action Change (Du)

Setpoint

Profile

time (s) Reference

Figura 0.14: Pestaña MPC en el panel frontal de la simulación

Elaborado por: López, V y Paredes, A

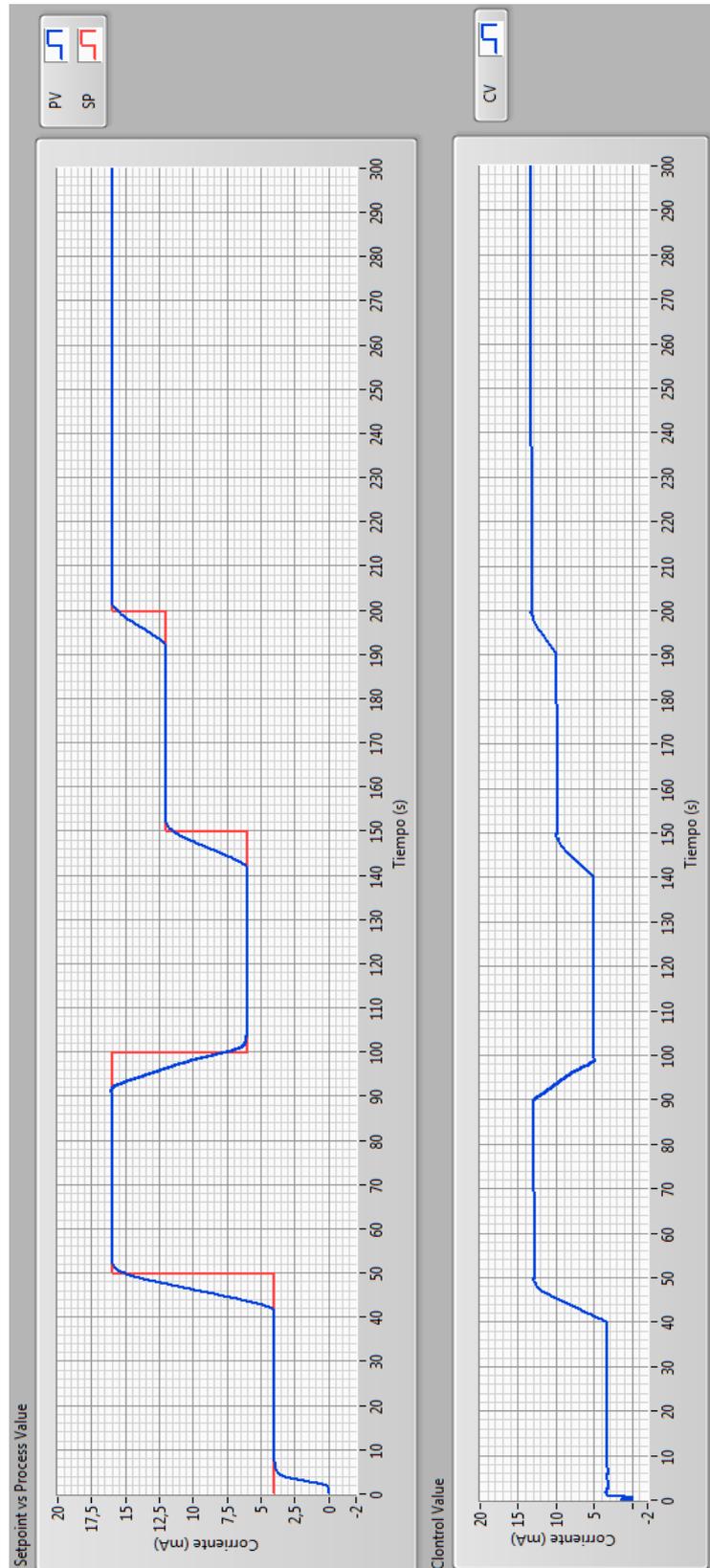


Figura 0.15: Pestaña Gráficas en el panel frontal de la simulación

Elaborado por: López, V y Paredes, A

1.11. COMPARACIÓN ENTRE MPC Y PID SIMULADOS

Para tener un punto de referencia al evaluar el control predictivo se lo compara con el controlador más usado como es el Proporcional Integral Derivativo (PID)², y en función a estos resultados determinar el desempeño del controlador bajo estudio.

1.11.1. TIEMPO DE ESTABILIZACIÓN

En la Figura 0.16 se puede observar el resultado de un escalón en el *setpoint* de 4 a 16 mA para los dos controles en el mismo tiempo, si bien se puede notar un adelanto de la respuesta del proceso al cambio de *setpoint* en el controlador predictivo es porque al tener todo el perfil de cambios, el MPC se puede adelantar al cambio que esta por suceder, ésta respuesta únicamente se da en la simulación, al momento de implementarlo la curva será similar pero estará en fase con el cambio de *setpoint* puesto que en la vida real no se puede anticipar cambios futuros.

De la Figura 0.16 (a), se puede notar que el tiempo que le toma al controlador predictivo estabilizar la señal hacia el nuevo valor consigna es de aproximadamente diez segundos.

Para el PID se puede observar en la Figura 0.16 (b) que el tiempo de estabilización desde el cambio de *setpoint* es de aproximadamente diecisiete segundos.

² La simulación del controlador PID se encuentra en los anexos, tanto el panel frontal como el diagrama de bloques.

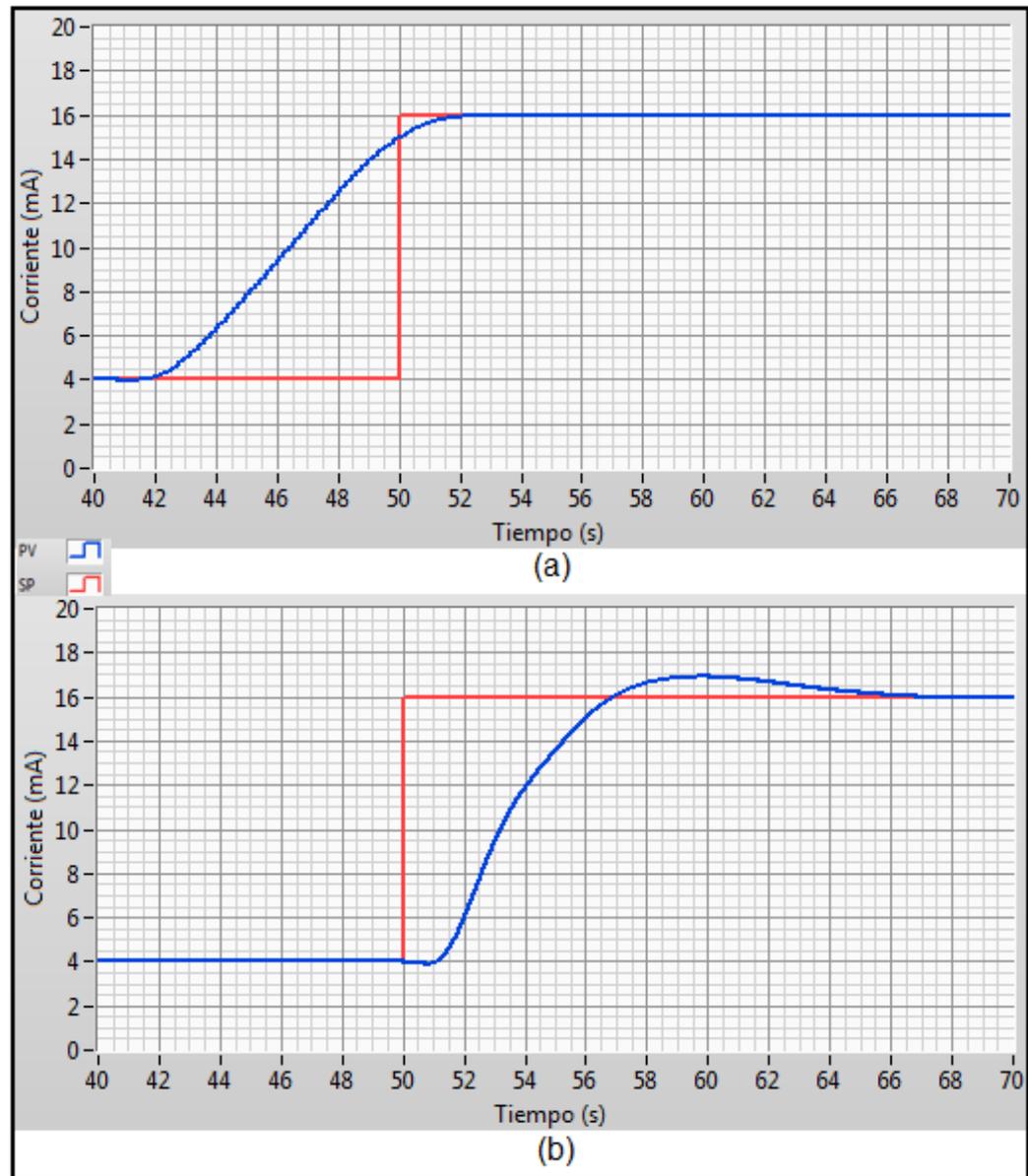


Figura 0.16: Setpoint vs Process Value, (a) Predictivo, (b) PID

Elaborado por: López, V y Paredes, A

1.11.2. SOBREPULSO

Para el controlador predictivo se puede apreciar en la Figura 0.16 (a) que prácticamente no hay sobreimpulso y la señal llega a estabilizarse en el valor fijado sin necesidad de pasarlo.

A su vez, el controlador PID tiene un sobreimpulso de aproximadamente seis por ciento, como se puede deducir de la Figura 0.16 (b).

1.11.3. ACCIÓN DE CONTROL

En la Figura 0.17 se puede observar las diferencias en las acciones de control, para el control predictivo se tiene una señal más estable y lineal comparándola con la acción del PID la cual tiene a ser más dinámica e irregular en el tiempo, esto por la acción de proporcionalidad del controlador, lo que le da esa característica; al contrario de la señal del predictivo que tiene un modelo de referencia y evita esas curvas en la acción de control.

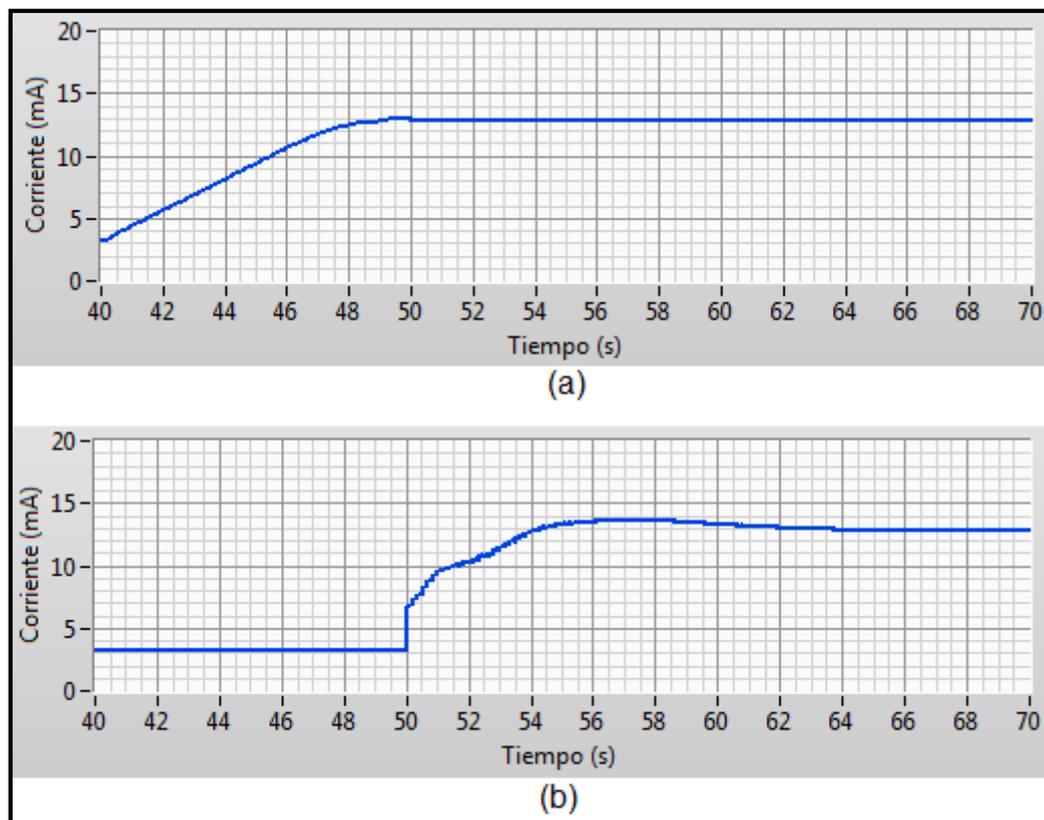


Figura 0.17: Señales de control, (a) Predictivo, (b) PID

Elaborado por: López, V y Paredes, A

Todo lo anteriormente analizado se puede resumir como indica la Tabla 0.1, con todos los datos obtenidos de la simulación, para su mejor comparación.

Tabla 0.1. Comparación entre MPC y PID simulados

	MPC	PID
Tiempo de estabilización aprox. (s.)	10	17
Sobreimpulso aprox. (%)	0	6
Estabilidad en la señal de control	Mejor Operación	Peor Operación

CAPÍTULO 4

IMPLEMENTACIÓN DEL CONTROLADOR PREDICTIVO BASADO EN MODELOS EN LA PLANTA DE FLUJO

La implementación del controlador en la planta de flujo se lo detallará en tres etapas, las cuales son: hardware, software. Para un mejor análisis en cada una de las partes de la implementación.

1.12. HARDWARE

Las características de los equipos físicos a utilizar se detallaron en el primer capítulo, por tanto, se trata en esta sección la implementación con los mismos.

La arquitectura física es un clásico lazo cerrado de control, como se muestra en la Figura 0.1 en diagrama de bloques.

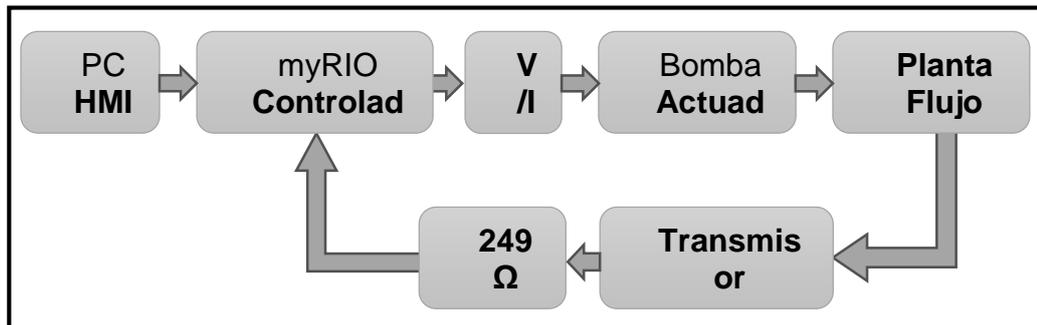


Figura 0.1: Diagrama de bloques del lazo de control predictivo

Elaborado por: López, V y Paredes, A

En la Figura 0.2 se puede observar la fotografía de la implementación con todos los equipos en el proceso de flujo (LAB-VOLT 3502) en el laboratorio de Redes Industriales y Control de Procesos de la Universidad de las Fuerzas Armadas ESPE extensión Latacunga.

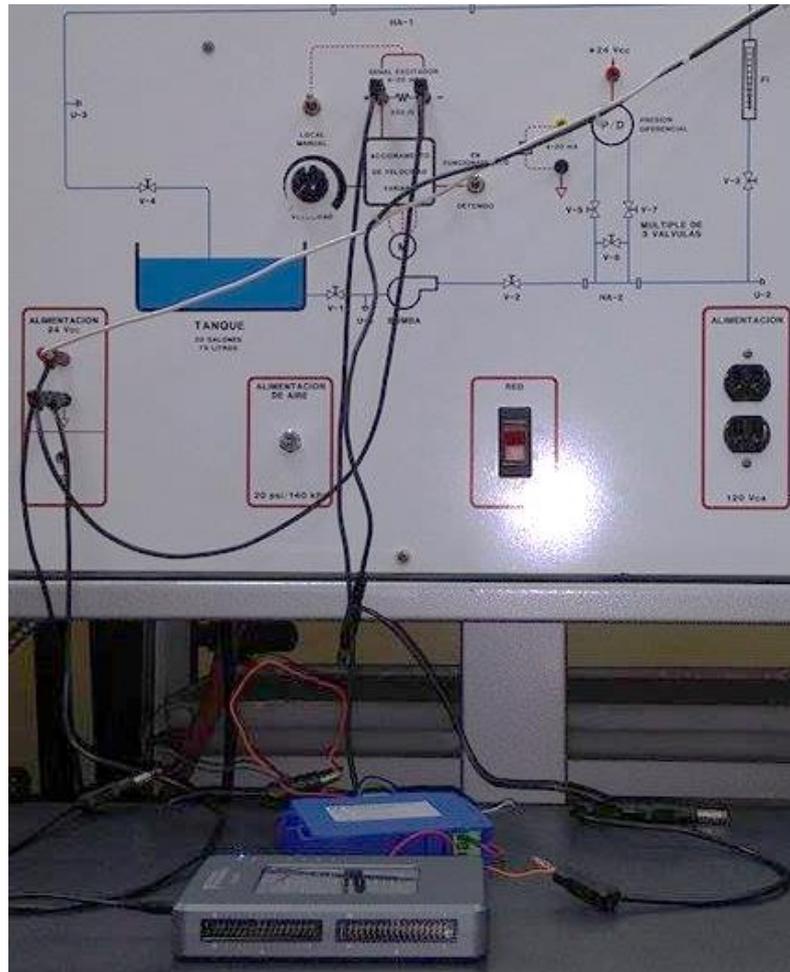


Figura 0.2: Fotografía de la implementación en la planta

Elaborado por: López, V y Paredes, A

1.12.1. COMUNICACIÓN HMI - CONTROLADOR

La comunicación entre el computador (donde se encuentra el HMI) y el dispositivo myRIO se lo puede realizar de manera USB o mediante la interfaz inalámbrica Wi-Fi. Para este controlador se decide utilizar el método inalámbrico por mayor comodidad y factibilidad.

Se crea una red virtual en el computador a la cual se conectará el dispositivo myRIO. Para el sistema operativo Windows se lo puede realizar con los siguientes comandos en el símbolo del sistema.

```
netsh wlan set hostednetwork mode=allow ssid="nombre" key="clave"
```

```
netsh wlan start hostednetwork
```

En la Figura 0.3 se puede observar la ejecución de los comandos mencionados en el símbolo del sistema sobre Windows 7, con el nombre de red *Tesis* y como contraseña *tesisrio*, la cual estará disponible para que el dispositivo myRIO se conecte.

```
C:\Windows\system32>netsh wlan set hostednetwork mode=allow ssid=Tesis key=tesisrio
El modo de red hospedada se estableció en permitir.
Se cambió correctamente el SSID de la red hospedada.
Se cambió correctamente la frase de contraseña de clave de usuario de la red hospedada.

C:\Windows\system32>netsh wlan start hostednetwork
Se inició la red hospedada.

C:\Windows\system32>pause
Presione una tecla para continuar . . .
```

Figura 0.3: Creación de una red mediante símbolo del sistema

Elaborado por: López, V y Paredes, A

Para conectar el equipo a la red creada en el computador, se usa la conexión física USB para configurar el adaptador inalámbrico del mismo, usando para esto el software de reconocimiento de NI MAX, en la Figura 0.4 se puede observar la ventana del software con el equipo conectado.

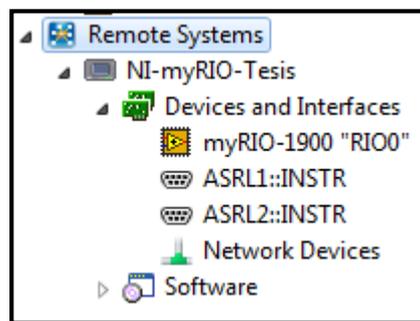


Figura 0.4: Dispositivo myRIO en el software de reconocimiento NI MAX

Elaborado por: López, V y Paredes, A

En la Figura 0.5 se muestra la configuración para que el dispositivo myRIO se conecte mediante Wi-Fi a una red inalámbrica, en este caso a la red llamada *Tesis*, usando un navegador de internet, para esto, dar clic derecho en el equipo y seleccionar la opción **Web Configuration**. De aquí

en adelante el equipo se conectará automáticamente a la red cada vez que sea encendido, sin la necesidad de la conexión física.

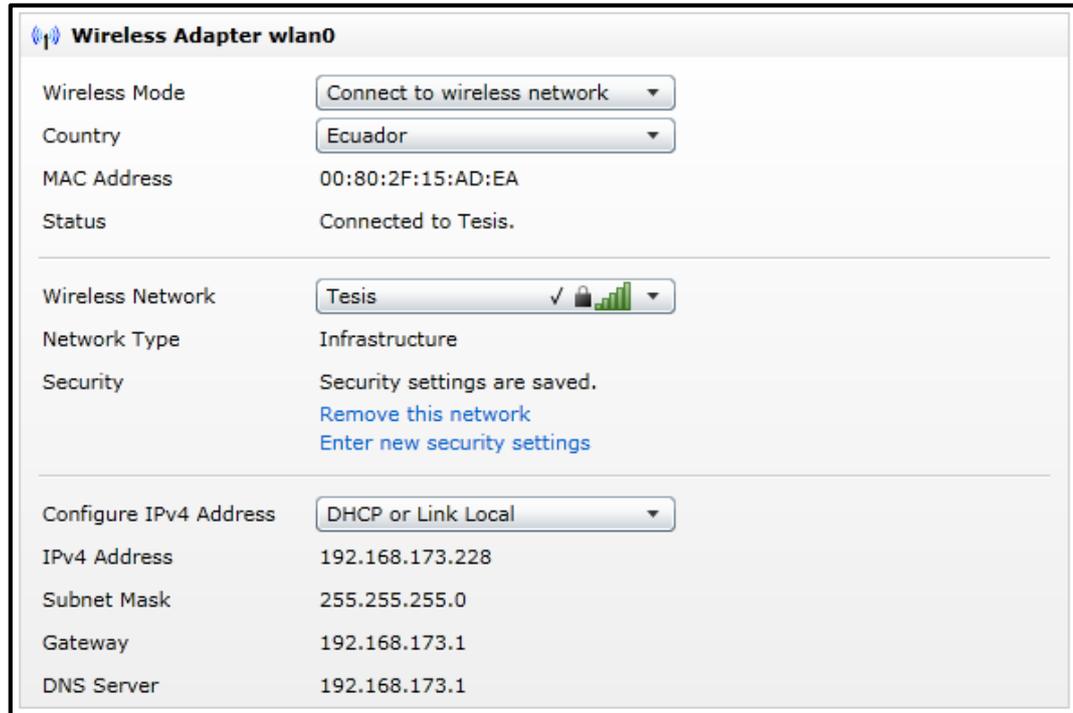


Figura 0.5: Configuración de conexión Wi-Fi para myRIO

Elaborado por: López, V y Paredes, A

1.12.2. CONVERSIONES VOLTAJE - CORRIENTE

Debido a que el dispositivo controlador posee únicamente salidas y entradas de voltaje y los procesos industriales, como la planta de flujo, manejan entradas y salidas de corriente, se tiene la necesidad de usar un convertor de voltaje a corriente (V/I) para conectar la señal de control (voltaje) al actuador (corriente), para este fin se usa el convertor Omega DRF-VDC³. Es un convertor con una amplia variedad de rangos de operación tanto en entrada como salida, por tanto se debe configurar los

³ El datasheet del convertor se encuentra disponible en la dirección web: <http://www.omega.com/manuals/manualpdf/M4139.pdf>

rangos de trabajo adecuado mediante los *jumpers* ubicados en la parte frontal del dispositivo, como muestra la Figura 0.6.

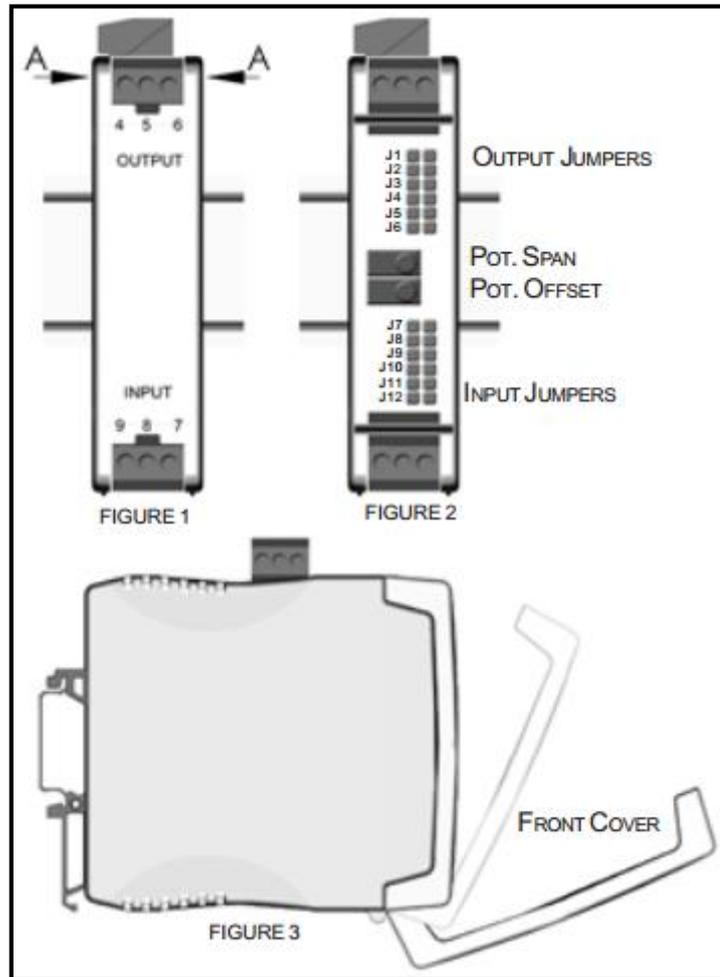


Figura 0.6: Ubicación de los *jumpers* de configuración del convertidor V/I [6]

La salida de corriente es la más común para procesos industriales, siendo 4 – 20 mA el rango a utilizar para el actuador, y en cuanto a la salida del controlador myRIO puede abarcar el rango de -10 a 10 Vdc, un valor cercano de trabajo para este convertidor es 7,5 Vdc. Para estas configuraciones de entrada/salida se deben ubicar los *jumpers* como indica la Figura 0.7.

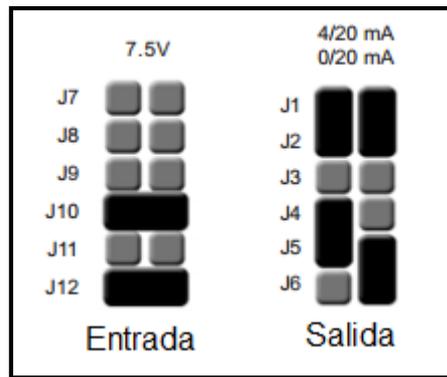


Figura 0.7: Configuración de los *jumpers* de entrada y salida [6]

Para la conversión de corriente a voltaje, es decir, la conexión del transmisor de presión diferencial (corriente) hacia el dispositivo myRIO (voltaje), se utiliza una resistencia de precisión de 249 ohmios en paralelo a los terminales de medición. Las entradas son diferenciales, por tanto el terminal negativo deberá conectarse a la tierra del circuito.

La conexión completa del controlador, para que tenga como entrada y salida señales de corriente se especifica en la Figura 0.8, obviando las respectivas alimentaciones de los equipos.

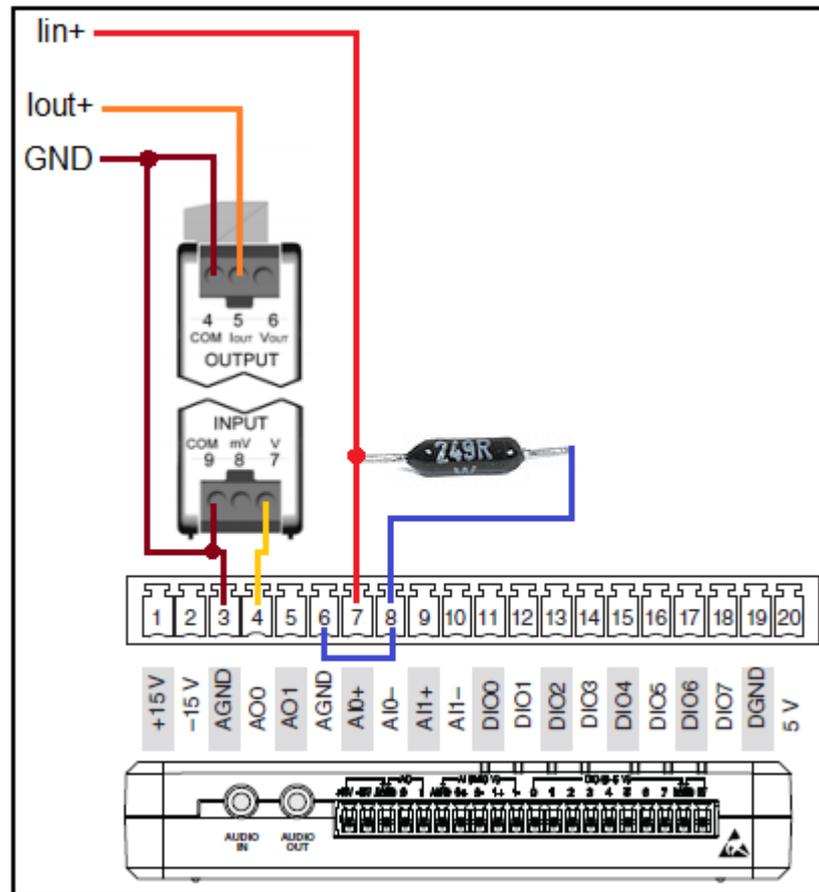


Figura 0.8: Conexión física entre myRIO, convertor y resistencia

Elaborado por: López, V y Paredes, A

Una vez acondicionado el controlador para que trabaje con señales de corriente, se puede conectar al actuador y al transmisor de presión diferencial de la planta, además de la alimentación para cada equipo y unir las tierras entre los mismos para un mejor desempeño.

1.13. SOFTWARE

Antes de poder implementar el controlador predictivo, se debe cargar los *toolkits* necesarios al dispositivo myRIO, para poder utilizar las herramientas sobre el sistema operativo del mismo. Para esto usamos el software NI MAX mediante una conexión física o inalámbrica.

En la Figura 0.9 se muestran las herramientas utilizadas en el equipo myRIO para la implementación del controlador predictivo.

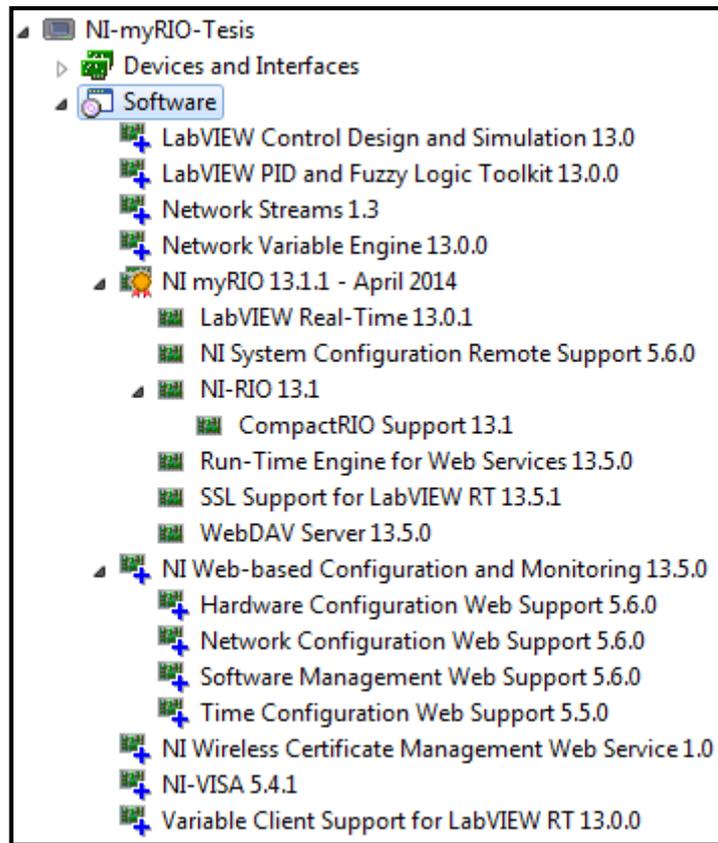


Figura 0.9: Lista de *toolkits* necesarios en el dispositivo myRIO

Elaborado por: López, V y Paredes, A

Como se detalló previamente, el controlador se lo va a implementar en el sistema operativo del equipo myRIO y la interfaz Humano-Máquina (HMI) en un computador, por tanto, se trabaja con un proyecto en el software LabVIEW, para comunicar ambos equipos mediante conexión inalámbrica Wi-Fi.

La estructura del proyecto se puede observar en la Figura 0.10, se puede notar los VI's para el HMI (PC.vi), el controlador (RT.vi) y el manejo de la FPGA del equipo (FPGA.vi), además de la librería para el manejo de los valores entre el dispositivo myRIO y el computador (Variables.lvlib).

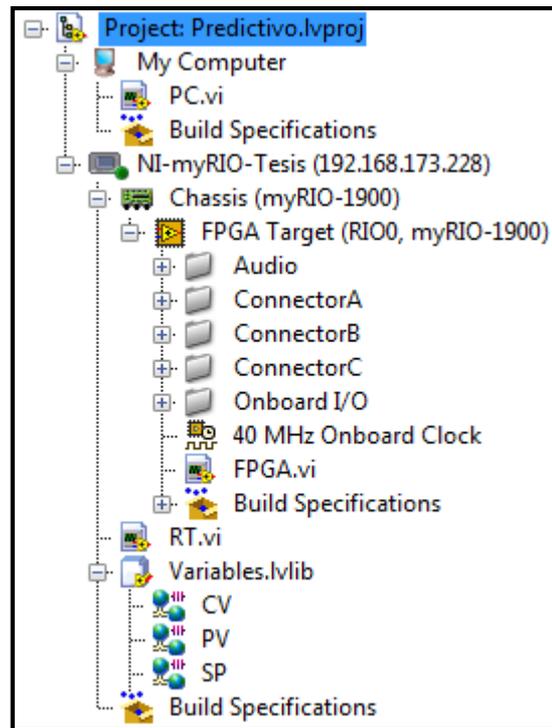


Figura 0.10: Esquema del proyecto “Predictivo” en LabVIEW

Elaborado por: López, V y Paredes, A

1.13.1. FPGA

Todas las entradas y salidas del equipo myRIO se encuentran conectadas y administradas por la FPGA integrada, por tanto, para poder manejarlas es necesario crear un VI que posteriormente es compilado para ser funcional. Se recomienda que el panel frontal respectivo sea lo más sencillo posible, para minimizar recursos de la FPGA, ya que este VI es transparente para el resultado final (HMI).

Para este proceso en particular, se requiere únicamente una entrada analógica (AI0) y una salida analógica (AO0). Por tanto el diagrama de bloques resulta como indica la Figura 0.11.

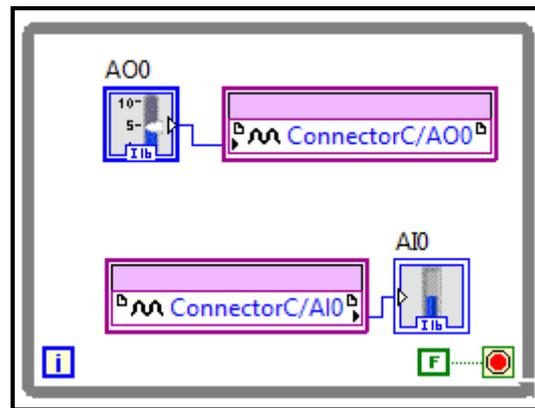


Figura 0.11: Diagrama de bloques del VI FPGA.vi

Elaborado por: López, V y Paredes, A

Como se puede notar en la Figura 4.11 los valores son de tipo *Int16*, es decir enteros, por lo que la lectura y escritura no es directamente en unidades de voltaje. La resolución en entrada y salida analógica de los conectores MSP es de doce bits con un rango de ± 10 V, por tanto el cálculo para convertir estos valores enteros a voltaje queda determinado como muestra la Ecuación (0.1).

$$V = ADC * 0.004883 \quad (0.1)$$

Donde:

V = valor en unidades de voltaje

ADC = valor de lectura – escritura en entero tipo *Int16*

En el lazo *while* se coloca una constante, puesto que este VI no necesita un control por ser transparente al usuario y será llamado desde otro programa.

El panel frontal para la FPGA se diseña de manera más sencilla, en la Figura 0.12 se distingue un controlador para la salida AO0 y un indicador para la entrada AI0, con sus límites máximos para los 10 V que maneja el conector MSP.

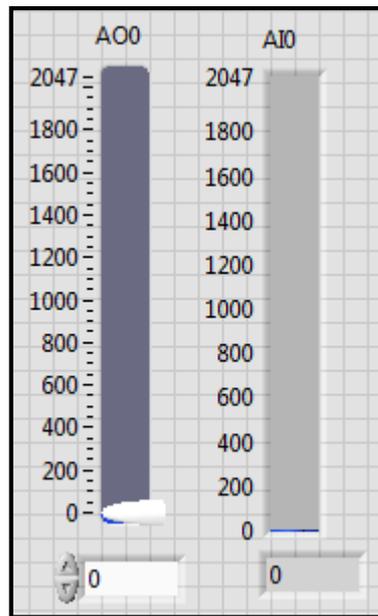


Figura 0.12: Panel frontal del VI FPGA.vi

Elaborado por: López, V y Paredes, A

1.13.2. CONTROLADOR

Todo el algoritmo de control predictivo se lo implementa en el sistema operativo en tiempo real del dispositivo myRIO, en el proyecto se lo denomina RT.vi.

EL diagrama de bloques completo del controlador se muestra en la Figura 0.13, En general es el programa para adquirir la señal y tratarla para tener una señal de salida en el actuador, y a partir del mismo se detalla cada etapa de la programación por partes.

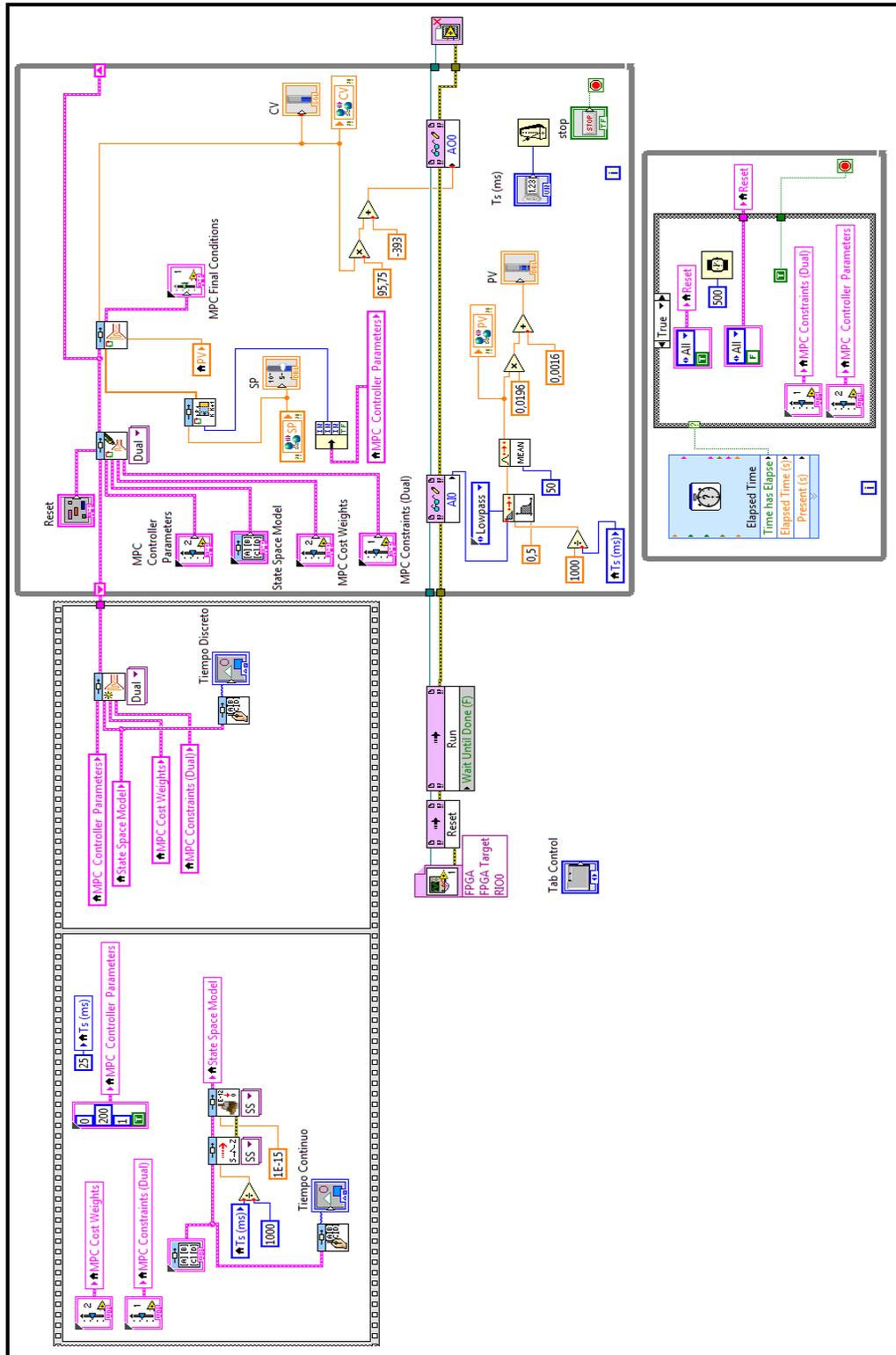


Figura 0.13: Diagrama del bloques del VI RT.vi

Elaborado por: López, V y Paredes, A

Todo lo que se encuentra fuera de los lazos repetitivos *while* son inicializaciones ya sea de variables, funciones u otro dispositivo.

En la Figura 0.14 se muestra las inicializaciones de los parámetros del controlador MPC, explicados anteriormente, así como del tiempo de muestreo expresado en milisegundos.

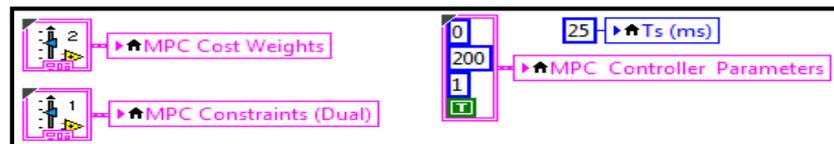


Figura 0.14: Inicialización de parámetros MPC y tiempo de muestreo

Elaborado por: López, V y Paredes, A

Como se mencionó en el segundo capítulo de este documento, el modelo en ecuaciones en espacio de estados se lo determinó en tiempo continuo para mayor versatilidad al momento de convertirlo a tiempo discreto, ya que nos permite cambiar el tiempo de muestreo sin necesidad de volver a estimar las ecuaciones, por tanto en LabVIEW se realiza este proceso antes de entrar en el lazo repetitivo. En la Figura 0.15 se observa el procedimiento para convertir un modelo en ecuaciones en espacio de estados (SS) a tiempo discreto dado un período de muestreo específico (T_s) expresado en segundos, de ahí que se haga la conversión de milisegundos a segundos mediante la división; en el proceso puede resultar cantidades muy pequeñas e insignificantes para el modelo final, por esto se elimina en la siguiente etapa los valores menores a $1E-15$.

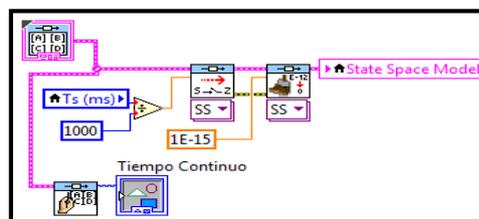


Figura 0.15: Conversión del modelo en tiempo continuo a tiempo discreto

Elaborado por: López, V y Paredes, A

Una vez inicializados los parámetros y el modelo matemático, se crea el controlador predictivo MPC, en la Figura 0.16 se muestra el uso de la función *Create MPC controller* para este fin, además de una función de visualización de ecuaciones para el modelo.

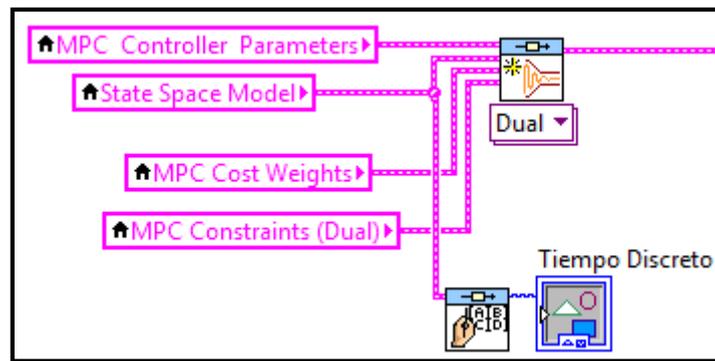


Figura 0.16: Creación del controlador predictivo basado en modelos

Elaborado por: López, V y Paredes, A

Antes de poder utilizar las entradas y salidas configuradas en la FPGA del equipo desde el programa que se encuentra en el sistema operativo en tiempo real (RT.vi), se debe abrir una conexión hacia la FPGA y con esa referencia poder manipular los controles que se encuentran en el programa (FPGA.vi), en la Figura 0.17 se ve la función *Open FPGA* con el VI ya cargado, además de unas funciones especiales como son *Reset* y *Run* que a pesar de no ser obligatorio ponerlas, es una manera más segura de inicializar el programa invocado.

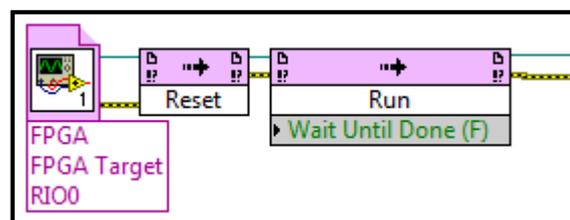


Figura 0.17: Proceso para abrir una conexión a la FPGA

Elaborado por: López, V y Paredes, A

Se tiene un lazo *while* adicional al de control, que se encuentra temporizado para actualizar algunos parámetros y restricciones del

controlador MPC, el cual se muestra en la Figura 0.187. Esta actualización de parámetros ocurre a los diez segundos de haber empezado el programa, un tiempo suficientemente amplio para que el proceso se estabilice, y se lo realiza por las condiciones iniciales del proceso y los conflictos que genera en las restricciones del controlador, como se enlista a continuación.

Variable del proceso, como se había determinado anteriormente los límites del proceso están comprendidos entre 4 – 20 mA (y min – y max), pero debido al uso del filtro para estabilizar la señal, inicialmente esta lectura empieza desde el valor de cero, y al estar fuera de los rangos del controlador, se produce conflictos en los cálculos y en la señal de control por corregir esta limitación, provocando que el procesador del dispositivo no reaccione adecuadamente o incluso perdiendo la conexión con el HMI, por esta razón se inicia los rangos en 0 – 20 mA y al cabo de diez segundos, el proceso está en sus rangos normales de trabajo, y se reconfigura estos límites a los adecuados.

Señal de control, otro problema se tiene en los límites de la señal hacia el controlador, que se especificó en 8 – 19 mA (u min – u max), ya que al iniciar el controlador, la señal en AO0 empieza en cero, estando fuera de los límites y ocasiona que el proceso no responda a la señal de control y el controlador realice esfuerzos excesivos por tratar de corregir este valor que está violando las restricciones impuestas, para solucionarlo, de igual manera empezamos con un rango de 0 – 20 mA, y al cabo de diez segundos, cuando el proceso está en su rango operativo se cambia a los valores normales.

Horizonte de predicción, más que un problema o restricción, se trata de optimizar los cálculos, pues como se detalló, se necesita de diez segundos hasta que el proceso se estabilice y esté en sus rangos de trabajo nominales, por tanto, es ineficiente tener un horizonte de predicción amplio que requiere más procesamiento, cuando el sistema aún no está iniciado como el modelo lo predice, por ello se inicia con un horizonte de predicción bajo, ahorrando cálculos hasta que el proceso sea estable, una vez que llega a este punto, podemos cambiar el horizonte al valor real.

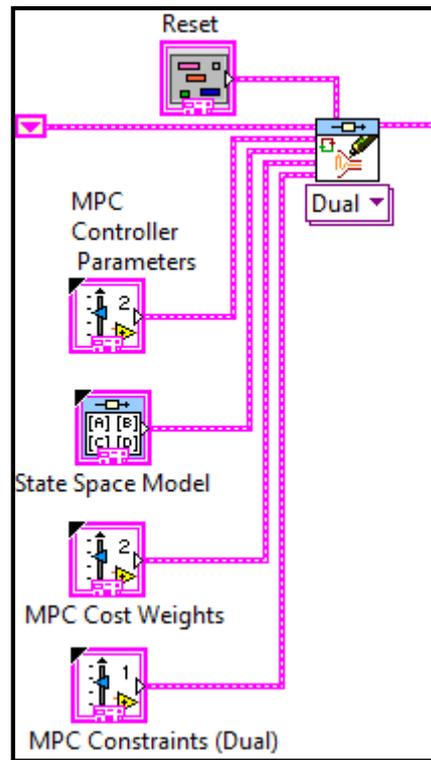


Figura 0.19: Configuración del controlador MPC dentro del lazo de control

Elaborado por: López, V y Paredes, A

La siguiente etapa a especificar es el acondicionamiento de las señales a las unidades requeridas, en este caso todas las variables (SP, PV y CV) están expresadas en miliamperios en un rango total de 4 – 20 mA. Todos los acondicionamientos son lineales, es decir, se usa la ecuación de la recta para ajustarlos.

Process Value, **para ajustar los rangos se tiene la entrada como los valores de lectura del ADC en la entrada AI0 y la salida como el rango requerido, en este caso 4 – 20 mA. Se realizó mediciones en ambos extremos para determinar el valor correspondiente en la entrada AI0, en la**

Tabla 0.1 se muestran los valores en (X, Y) para determinar la ecuación de ajuste.

Tabla 0.1. Datos para el acondicionamiento del *Process Value*

X (AI0)	Y (mA)
204	4
1020	20

Con los dos puntos y aplicando la ecuación de la recta, se puede determinar la expresión matemática para escalar la entrada a nuestros rangos en corriente, la Ecuación (0.2) muestra la operación necesaria.

$$y [mA] = 0.0196 * x [ADC] + 0.0016 \quad (0.2)$$

En la Figura 0.20 se ilustra el ajuste de los rangos mediante los cálculos descritos, además de dos etapas anteriores las cuales son: el filtrado de la señal en la frecuencia de 0.5 Hz, determinando la frecuencia de muestreo mediante T_s (ms); y un promedio de la señal con diez muestras para estabilizar mejor la respuesta de la misma.

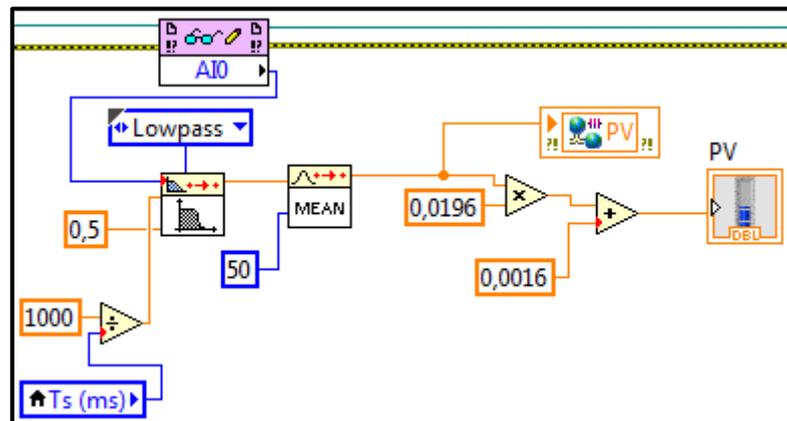


Figura 0.20: Acondicionamiento en LabVIEW del *Process Value*

Elaborado por: López, V y Paredes, A

Control Value, el controlador actúa sobre la planta en unidades de corriente, determinado así por el modelo matemático, por tanto los límites superior e inferior son 4 mA y 20 mA respectivamente como señal de control,

estos valores deben ajustarse a los rangos de la salida analógica AO0, se realizó una medición en corriente para determinar los valores correspondientes a dichos límites. En la Tabla 0.2 se muestran los resultados.

Tabla 0.2. Datos para el acondicionamiento del *Control Value*

X (mA)	Y (AO0)
4	-10
20	1522

De igual forma, aplicando la ecuación de la recta, se obtiene el cálculo necesario para el ajuste, mostrado en la Ecuación (0.3).

$$y [DAC] = 95.75 * x [mA] - 393 \quad (0.3)$$

El ajuste de rangos para poder ser usado en la salida analógica se detalla en la Figura 0.21, a pesar de no visualizarse, la señal viene del controlador MPC, que pasa a ser adaptada antes de ser usada por la FPGA para controlar el actuador.

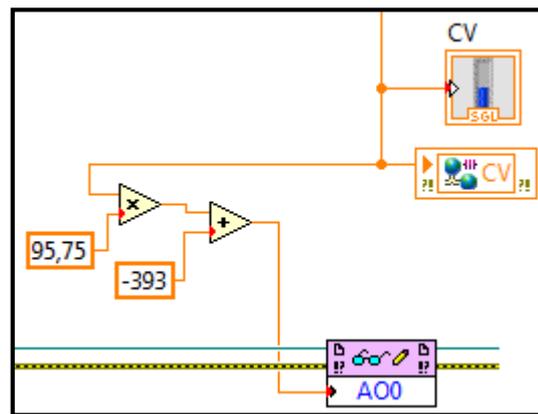


Figura 0.21: Acondicionamiento en LabVIEW del *Control Value*

Elaborado por: López, V y Paredes, A

Setpoint, dado que el setpoint es controlado en el HMI, no es necesario acondicionar en esta etapa, pues ya se encuentra en los rangos de corriente necesarios.

Al igual que en la simulación, es necesario tener una ventana de referencia (*setpoint*) de igual amplitud que el horizonte de predicción, para que se pueda llevar a cabo la diferencia (error) y se pueda calcular una acción de control adecuada. Pero difiere de la simulación, puesto que en la implementación se ajusta continuamente la referencia, no siendo previamente determinada como en el caso de una simulación. Para ello usamos la función *Update MPC Window* que se ilustra en la Figura 0.22, notándose que el *setpoint* es tomado de una variable local que toma el valor del HMI, y que se descompone el *cluster* de los parámetros del controlador MPC para determinar el tamaño del horizonte de predicción y calcular una ventana de referencia de igual extensión.

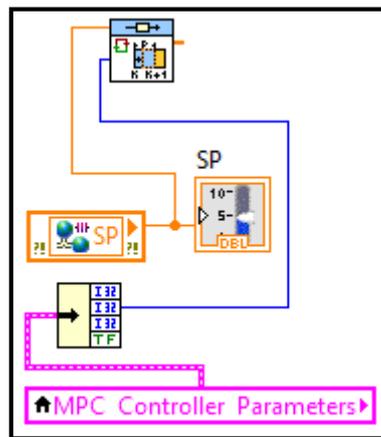


Figura 0.22: Función *Update MPC Windows* para el *setpoint*

Elaborado por: López, V y Paredes, A

Con todos los requisitos cumplidos, se puede realizar la implementación del controlador, como se muestra en la Figura 0.23, se observa que se usa una variable local para la variable PV (*Process Value*) para facilitar la conexión, además de un elemento extra designado *MPC Final Condition* que muestra información sobre los datos internos del controlador que pueden ser útiles, tales como: salida predicha, valor de estado, tasa de cambio de la variable del control.

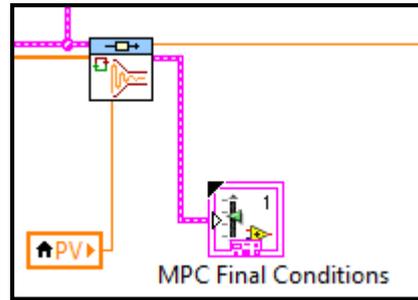


Figura 0.23: Implementación del controlador MPC

Elaborado por: López, V y Paredes, A

El panel frontal se divide en tres paneles denominados: Modelo, Controlador MPC y Variables. Los cuales se detallan a continuación.

Modelo, en esta pestaña se encuentra el modelado de la planta mediante ecuaciones en espacio de estados, tanto en tiempo continuo como discreto y su período de muestreo, como se ilustra en la Figura 3.13.

Controlador MPC, se encuentra todos los parámetros del controlador anteriormente detallados, se puede apreciar en la Figura 0.24 los mismos, tales como: Horizonte de predicción, restricciones, matrices de ponderación y como ayuda el indicador *MPC Final Condition*.

Ts (ms) 25

Parámetros del controlador MPC

Prediction Horizon 160 Control Horizon 1

Initial Window 0 Integral Action?

Reset All

Restricciones MPC (Dual)

u min init	u max init	u min final	u max final	y min init	y max init	y min final	y max final	du min init	du max init	du min final	du max final
0,00	0,00	8,00	0,00	0,00	0,00	19,00	0,00	0,00	0,00	0,00	0,00
0	0	0	0	0	0	0	0	0	0	0	0
0,00	0,00	2,00	0,00	0,00	0,00	22,00	0,00	0,00	0,00	-0,10	0,00
0	0	0	0	0	0	0	0	0	0	0	0
0,00	0,00	0,04	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
0	0	0	0	0	0	0	0	0	0	0	0

stopping criteria

function tolerance 1E-8

parameter tolerance 1E-6

gradient tolerance 1E+6

max iterations 10000

max function calls 10000

max time (sec) -1

Pesos de costo MPC

Output Error Weightings

0 0,022 30,000 30,000

0 30,000 30,000

Control Action Change Weightings

0 200,00 100,00 100,00

0 100,00 100,00

Control Action Error Weightings

0 0,00 0,00 0,00

0 0,00 0,00 0,00

Output Error Factors

0 0,00 0,00 0,00

Control Action Change Factors

0 0,00 0,00 0,00

Control Action Error Factors

0 0,00 0,00 0,00

MPC Final Conditions

Control Action 0 0,16

Output (y) 0 -0,00121255

Disturbance (d) 0 0

State (x) 0 -0,000104724

Control Action Change (Du) 0 0,04

Figura 0.24: Pestaña Controlador MPC en el panel frontal RT.vi

Elaborado por: López, V y Paredes, A

Variables, se encuentran los indicadores de las variables usadas en el proceso: *Setpoint (SP)*, *Process Value (PV)* y *Control Value (CV)* escalados de 4 a 20 mA en *Sliders*. Se muestra en la Figura 0.25.

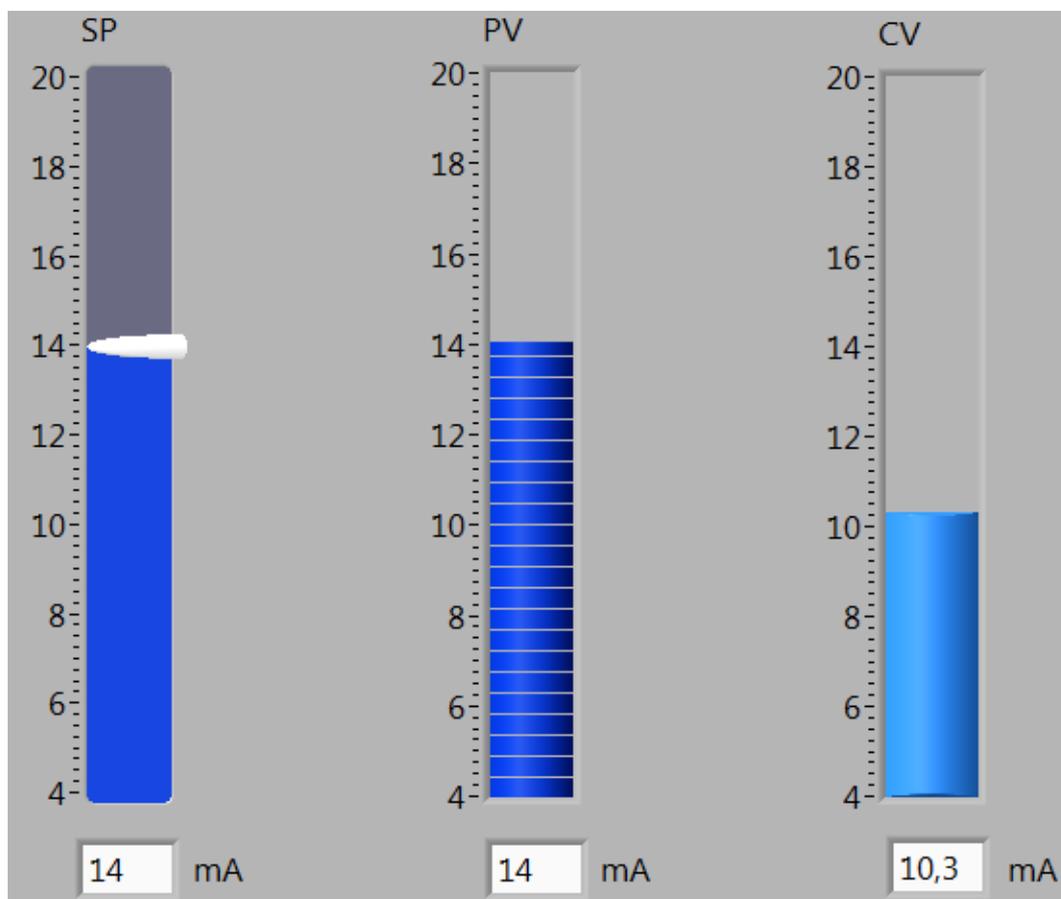


Figura 0.25: Pestaña Variables en el panel frontal RT.vi

Elaborado por: López, V y Paredes, A

1.13.3. INTERFAZ HUMANO – MAQUINA (HMI)

Este VI no tiene ninguna acción de control sobre el proceso, su función es la de visualización y manipulación del valor consigna (*setpoint*), está implementado sobre el computador, en el proyecto se denomina PC.vi.

El diagrama de bloques completo se muestra en la Figura 0.26, como se observa tiene indicadores, cuadros de gráficos y acondicionamientos, cada etapa se detalla de manera individual.

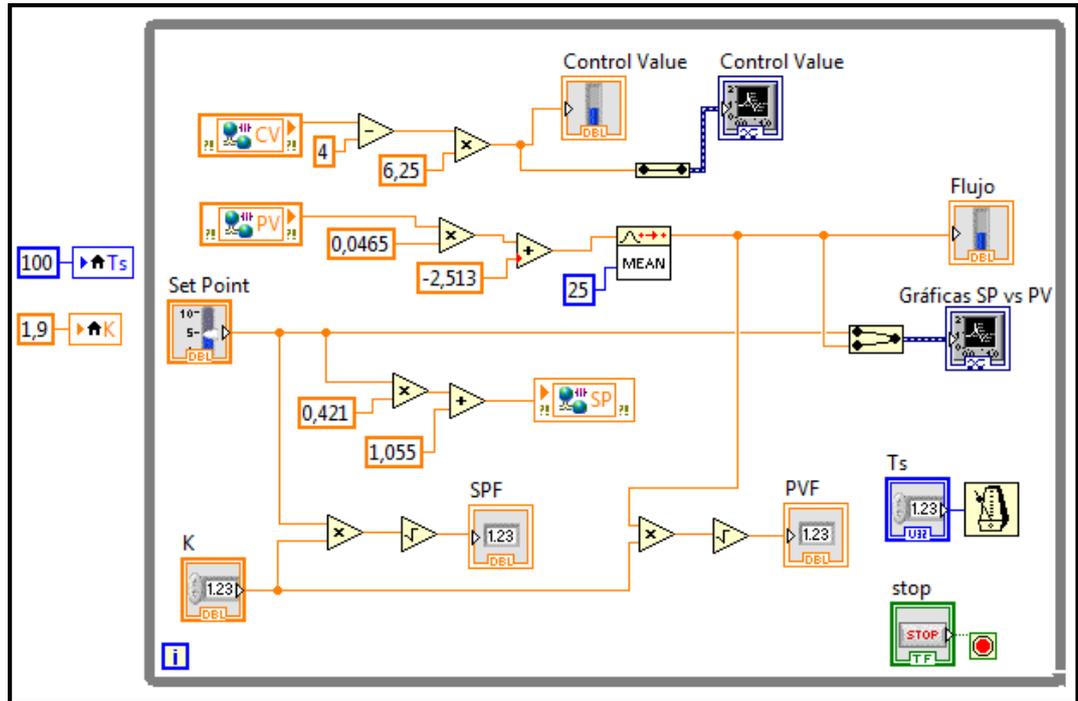


Figura 0.26: Diagrama del bloques del VI PC.vi

Elaborado por: López, V y Paredes, A

Fuera del lazo repetitivo se tiene la inicialización del tiempo de muestreo en 100 ms., y una constante de conversión igual a 1,9. El proceso de flujo mide el caudal de manera indirecta mediante un transmisor de presión diferencial, por tanto es necesario una operación matemática para obtener la medida en unidades de flujo. La relación entre presión diferencial y caudal se aprecia en la Ecuación (0.4).

$$Q^2 = k * (P.D.) \quad (0.4)$$

Donde:

Q = caudal

k = constante de conversión

$P.D.$ = presión diferencial

Dicha constante depende de varios factores físicos, tales como: ancho de la tubería, densidad del fluido, temperatura, presión ambiental, etc. Que están lejos de poder ser cuantificados para la conversión, por tanto se

estima dicho valor de manera empírica mediante mediciones, obteniendo así una constante $k = 1,9$. Si bien no se obtiene la conversión exacta, se obtienen resultados cercanos a los indicados por el rotámetro de la estación de flujo. La expresión resultante para obtener caudal se indica en la Ecuación (0.5).

$$Q = \sqrt{1.9 * (P. D.)} \quad (0.5)$$

En la Figura 0.27 se ilustra las operaciones en el software LabVIEW para la conversión de unidades, de presión diferencial a caudal.

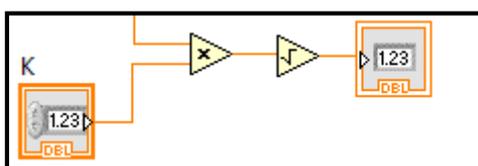


Figura 0.27: Conversión de presión diferencial a caudal en LabVIEW

Elaborado por: López, V y Paredes, A

Setpoint, dado que el valor consigna se manipula desde ésta interfaz, se encuentra en unidades de presión diferencial y flujo, pero el controlador necesita obtener el valor en miliamperios, se hace necesario un ajuste mediante la ecuación de la recta. La Tabla 0.3 muestra los valores mínimos y máximos para determinar la expresión.

Tabla 0.3. Datos para el ajuste del Setpoint inH_2O mA

X (inH_2O)	Y (mA)
7	4
45	20

Aplicando la ecuación de la recta con los puntos dados, la Ecuación (0.6) muestra las operaciones necesarias para la conversión.

$$y [mA] = 0,421 * x [inH_2O] + 1,055 \quad (0.6)$$

En la Figura 0.28 se muestra los cálculos en el diagrama de bloques para realizar la conversión de rangos de presión diferencial a corriente.

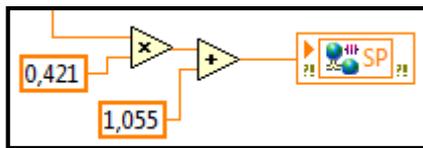


Figura 0.28: Conversión del *setpoint* inH₂O – mA en PC.vi

Elaborado por: López, V y Paredes, A

Process Value, sucede lo contrario que con el valor consigna, pues la medición se lleva a cabo en el controlador, que tiene unidades de corriente y se requiere ajustar a unidades de presión diferencial, los puntos a utilizar se muestran en la siguiente tabla.

Tabla 0.4. Datos para el ajuste del *Process Value* mA – inH₂O

X (mA)	Y (inH ₂ O)
5	7
20	45

Con los puntos anteriores, la Ecuación (0.7) muestra las operaciones para el ajuste de rangos.

$$y [inH_2O] = 0,0465 * x [mA] - 2,513 \quad (0.7)$$

La implementación de la expresión de acondicionamiento en LabVIEW se puede observar en la Figura 0.29.

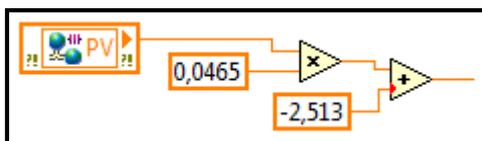


Figura 0.29: Conversión del *Process Value* mA – inH₂O en PC.vi

Elaborado por: López, V y Paredes, A

Control Value, el acondicionamiento de la acción de control es más sencilla, el rango del controlador es de 4 – 20 mA, la mejor manera de visualizarlo es en porcentaje (0 – 100), por tanto el ajuste necesario es como indica la Ecuación (0.8).

$$y [\%] = (x [mA] - 4) * 6,25 \quad (0.8)$$

La ejecución del cambio de rango en el programa HMI, se visualiza en la Figura 0.30.

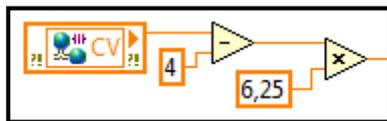


Figura 0.30: Conversión del *Control Value* mA - % en PC.vi

Elaborado por: López, V y Paredes, A

El panel frontal del HMI en su totalidad se observa en la Figura 0.31, mostrando los indicadores, controles y cuadros de gráficos señalados en el diagrama de bloques.

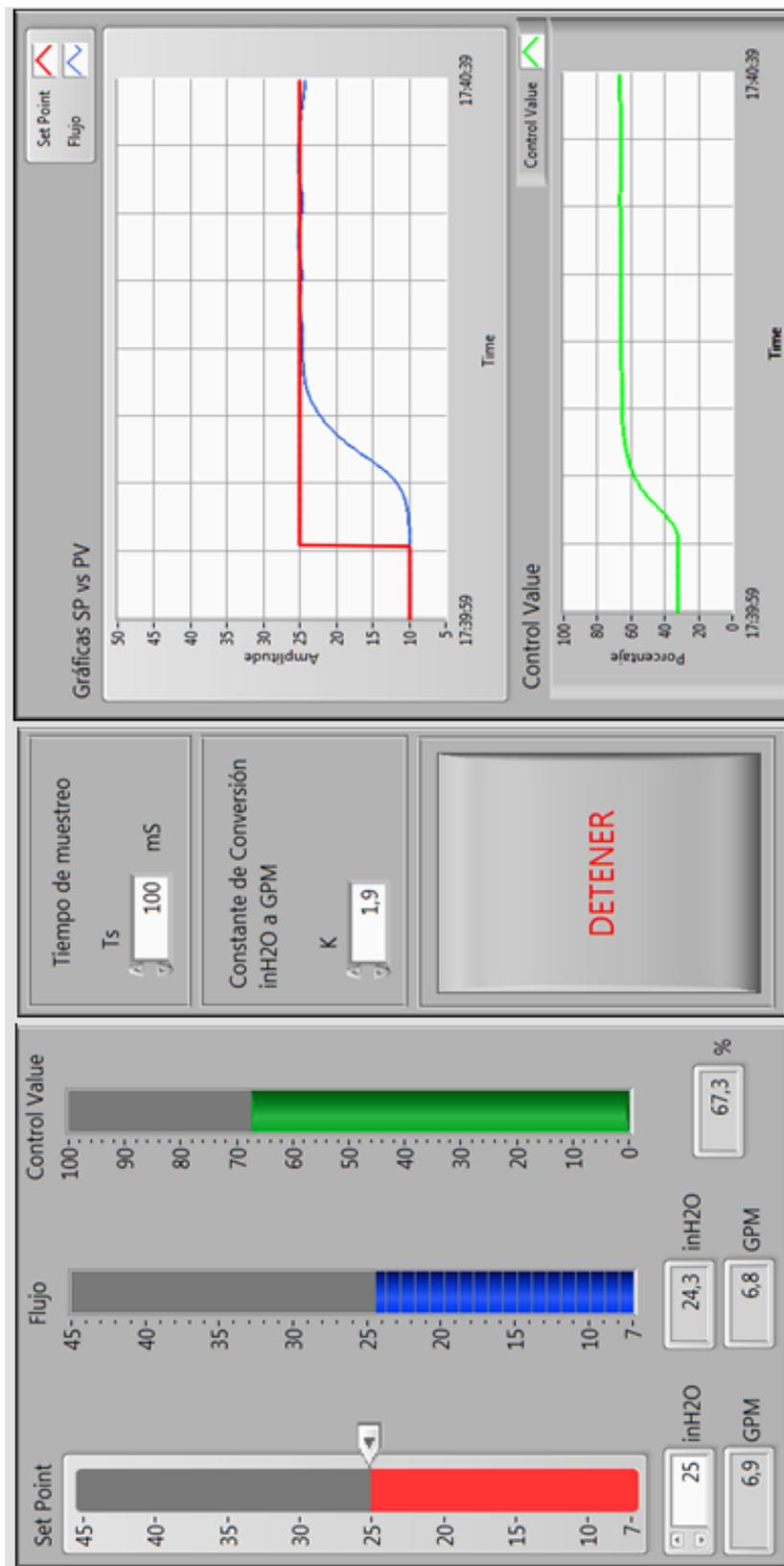


Figura 0.31: Panel frontal del VI PC.vi

Elaborado por: López, V y Paredes, A

CAPÍTULO 5

ANÁLISIS DE RESULTADOS

Para determinar los resultados del controlador predictivo basado en modelos, como primera etapa se realiza una comparación con los resultados obtenidos en la simulación, para comprobar el grado de coincidencia y validar los resultados de la simulación respecto a los reales. Posteriormente, para apreciar las características del controlador, se lo compara con un control realimentado Proporcional Integral Derivativo (PID), implementado sobre el mismo proceso de flujo, puesto que es el control más utilizado y conocido, por lo que será nuestra referencia y punto de partida para evaluar el desempeño del control predictivo objeto de este estudio.

Para cada etapa de análisis, se realizó pruebas en seis puntos diferentes a lo largo del rango del proceso, y el promedio de dichos experimentos se toma como valor representativo y por tanto el resultado final.

1.13.4. COMPARACIÓN ENTRE CONTROL PREDICTIVO REAL Y SIMULADO

En la Figura 0.1 se visualiza la respuesta del proceso ante un cambio en el *setpoint* de 10 a 25 inH₂O en la implementación, lo que corresponde a 5,26 a 11,58 mA en la simulación, notándose las similitudes en ambos controladores tanto en tiempo de estabilización y sobreimpulso.

El tiempo de estabilización aproximado en la simulación es de diez segundos, y para la implementación es cercano a catorce segundos, siendo tiempos semejantes en ambos casos, validando el resultado obtenido por simulación.

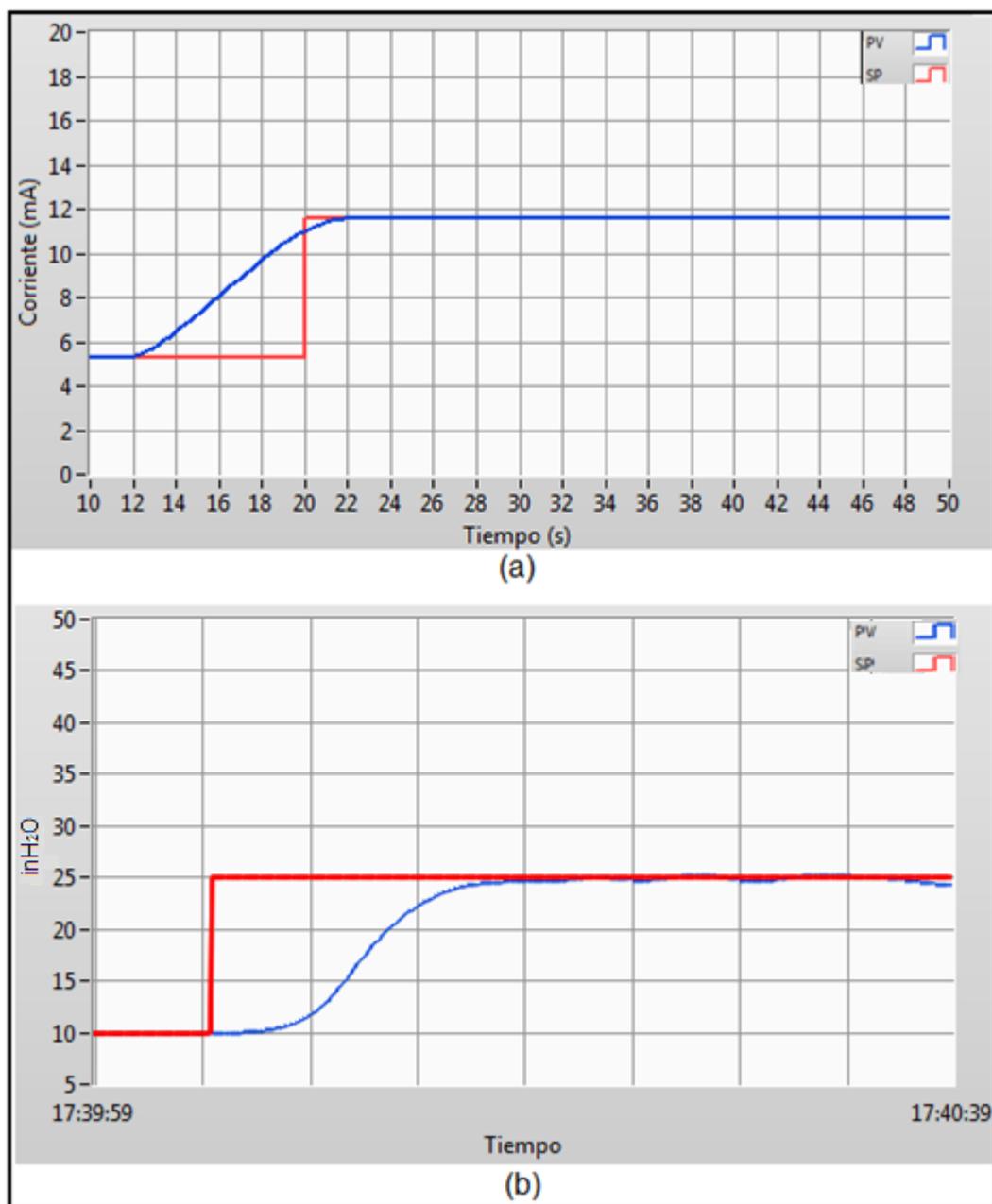
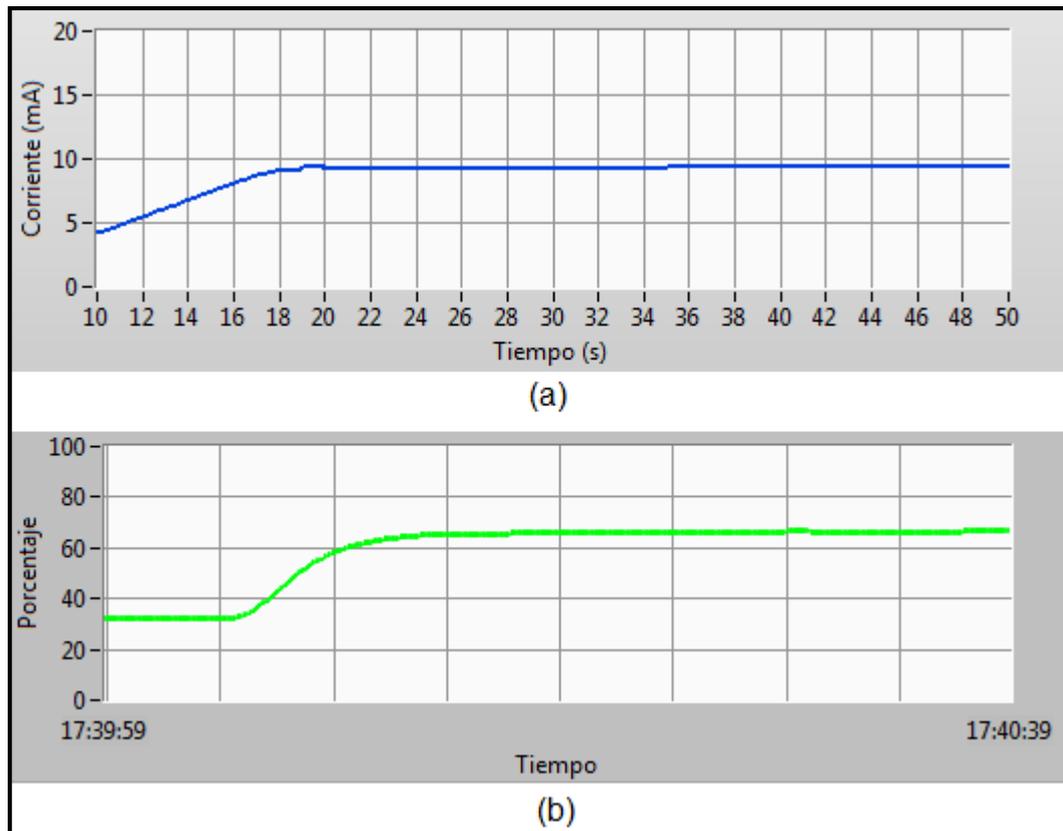


Figura 0.1: Setpoint 10 – 25 inH₂O control MPC, (a) Simulado, (b) Real

Elaborado por: López, V y Paredes, A

Las acciones de control resultan similares entre sí, resultando notorio la aproximación entre la pendiente en el control simulado y en el real, como se deduce de la Figura 0.2.



**Figura 0.2: Acción de control de 10 – 25 inH₂O en el controlador MPC,
(a) Simulado, (b) Real**

Elaborado por: López, V y Paredes, A

Las gráficas para el resto de cambios en el *setpoint* se pueden examinar en el **¡Error! No se encuentra el origen de la referencia..**

Los resultados están resumidos en la Tabla 0.1 mostrando las diferencias entre un control simulado y un real.

Tabla 0.1: Comparación entre MPC simulado y MPC implementado

Setpoint (inH ₂ O)	Tiempo de estabilización aprox. (s)		Porcentaje de sobreimpulso aprox. (%)	
	Simulado	Real	Simulado	Real
10 – 25	10	15	0	0
25 – 40	10	10	0	2
40 – 25	11	10	0	0
25 – 10	11	21	0	0
10 – 40	11	12	0	0
40 – 10	12	26	0	0
Promedio	10,83	15,66	0	0,33

1.13.5. COMPARACIÓN ENTRE MPC Y PID IMPLEMENTADOS

En la Figura 0.3 se muestra la respuesta de un controlador predictivo y un control PID al cambio en el *setpoint* de 10 – 25 inH₂O, mostrados en la misma escala de tiempo y amplitud, notándose la diferencia entre la variable a controlar, siendo más estable y teniendo un tiempo de establecimiento menor, debido a la característica principal del controlador predictivo de tener como referencia un modelo matemático para anticipar la respuesta futura, el control PID al basarse únicamente en el error no puede adecuar su acción de control para mejorar la salida del proceso, en la Figura 0.4 se puede observar las diferencias en la acción de control de los controladores.

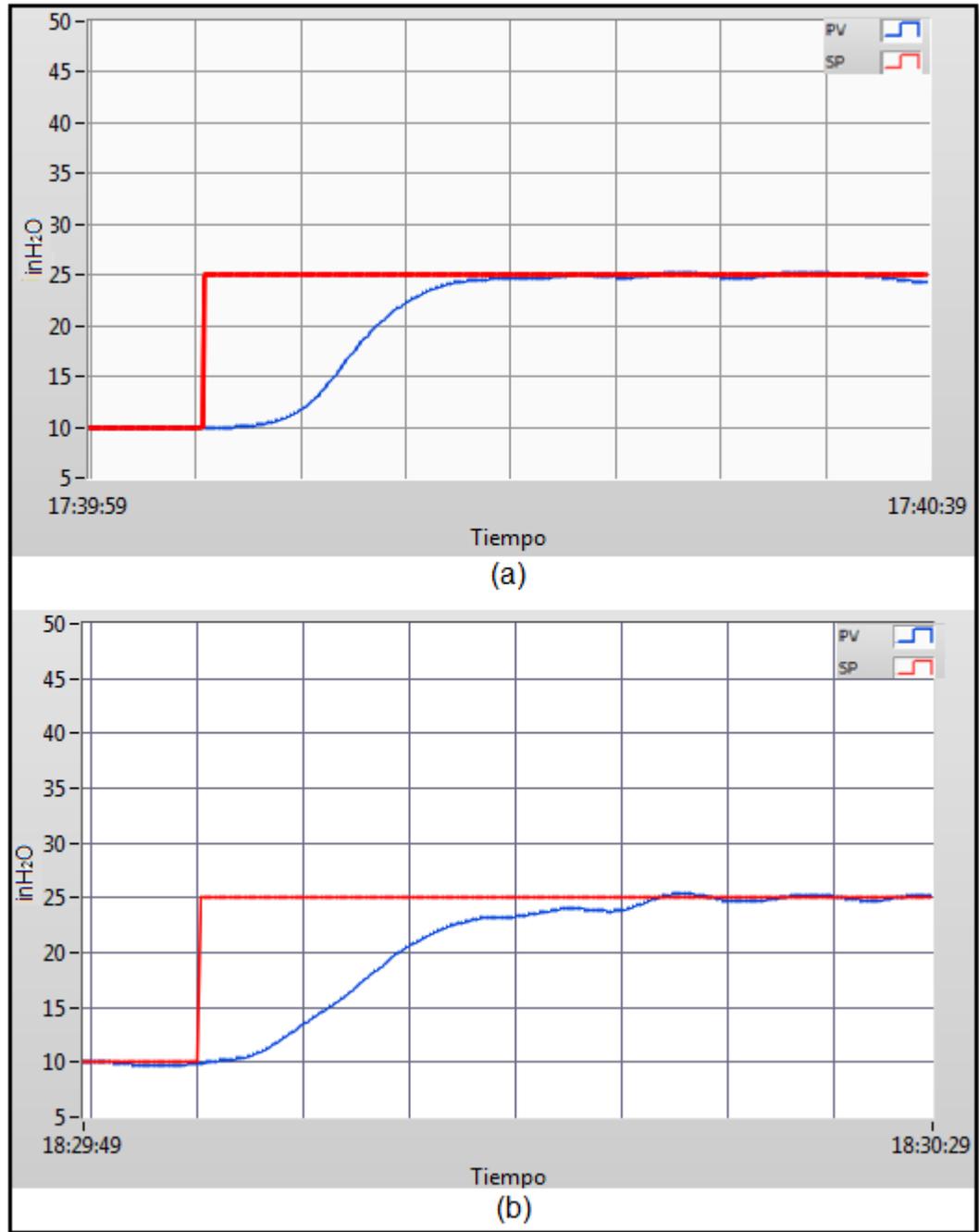


Figura 0.3: Setpoint 10 – 25 inH₂O, (a) MPC, (b) PID

Elaborado por: López, V y Paredes, A

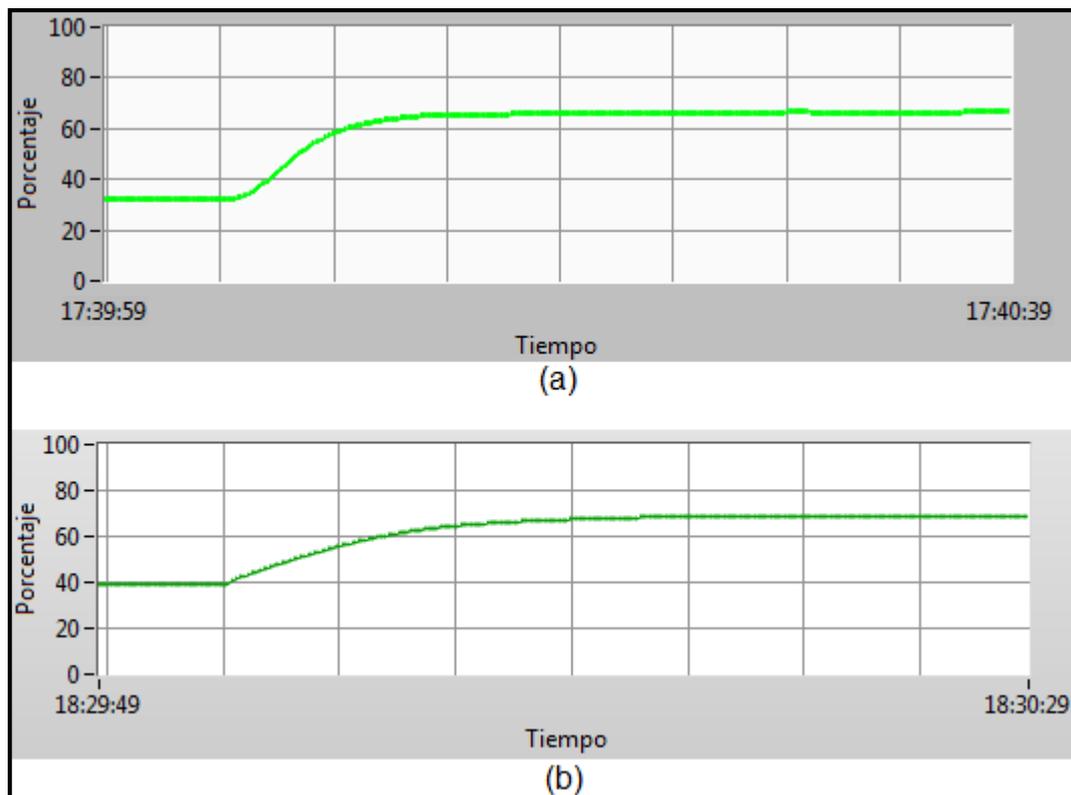


Figura 0.4: Acción de control 10 – 25 inH₂O, (a) MPC, (b) PID

Elaborado por: López, V y Paredes, A

Para un cambio de referencia de 25 – 40 inH₂O en el proceso de flujo se observa en la Figura 0.5 que los dos controladores poseen sobreimpulso, debido a que el sistema es muy sensible en niveles altos, a pesar de esta característica, el MPC demuestra tener un porcentaje de sobreimpulso y tiempo de estabilización menor respecto al PID. Las acciones de control mostradas en la Figura 0.6 aclaran el comportamiento del proceso para cada control.

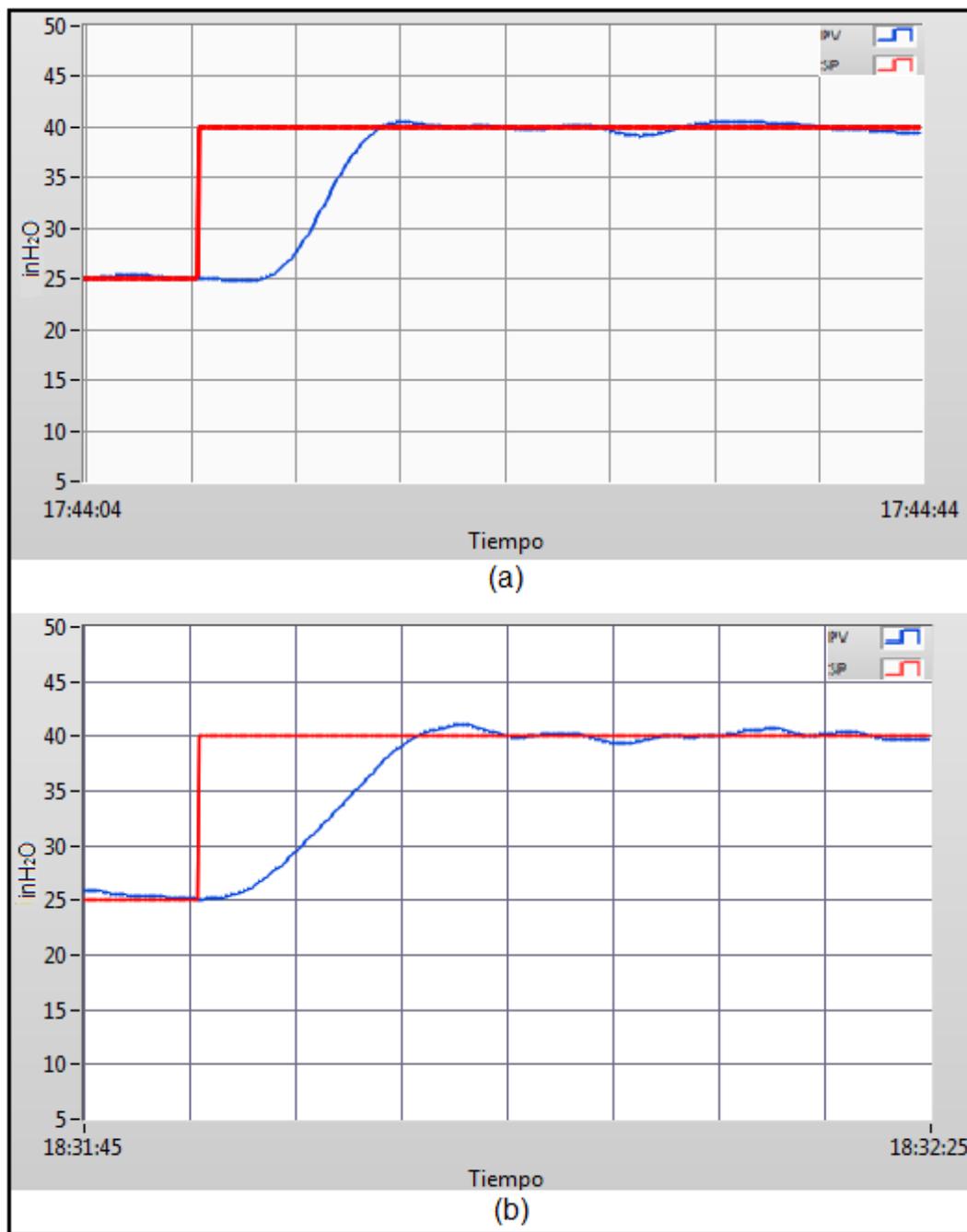


Figura 0.5: Setpoint 25 – 40 inH₂O, (a) MPC, (b) PID

Elaborado por: López, V y Paredes, A

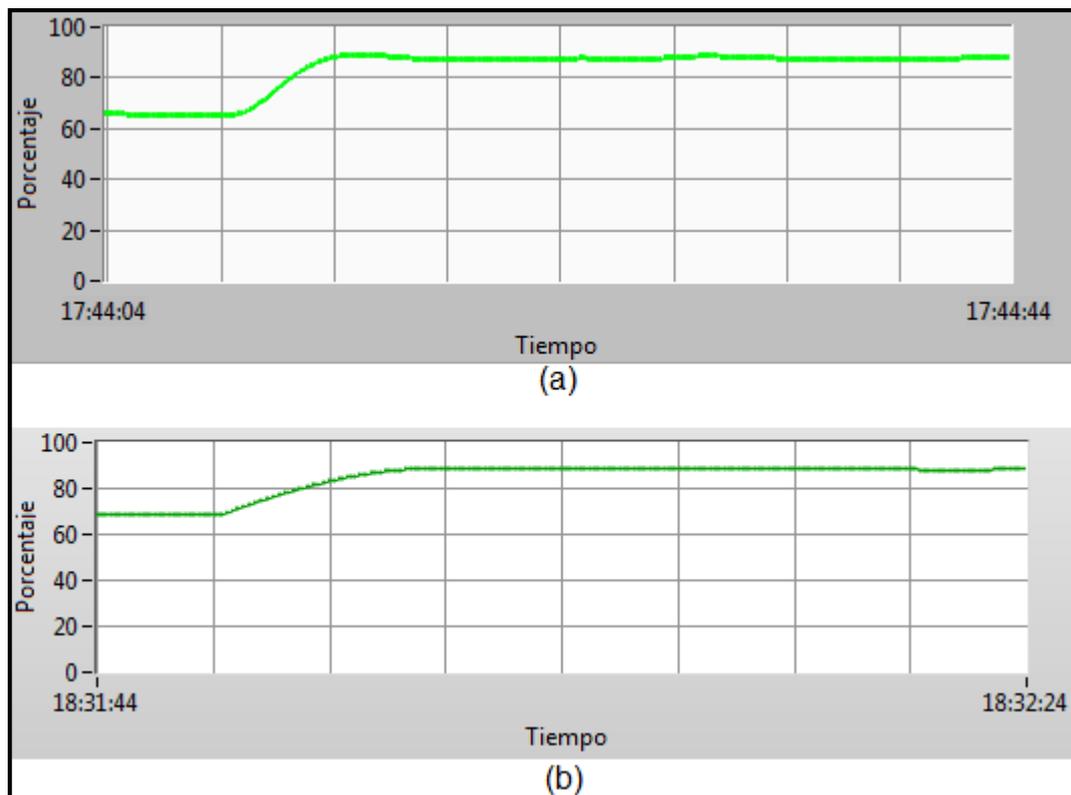


Figura 0.6: Acción de control 25 – 40 inH₂O, (a) MPC, (b) PID

Elaborado por: López, V y Paredes, A

Ante un cambio descendente del *setpoint* de 40 – 25 inH₂O, se aprecia en la Figura 0.7 una respuesta notablemente más estable y con menor tiempo de establecimiento del MPC respecto al PID, incluyendo que el controlador predictivo carece de sobreimpulso frente a su análogo. Las señales del *Control Value* en la Figura 0.8 manifiestan la razón de las diferencias entre ambas respuestas.

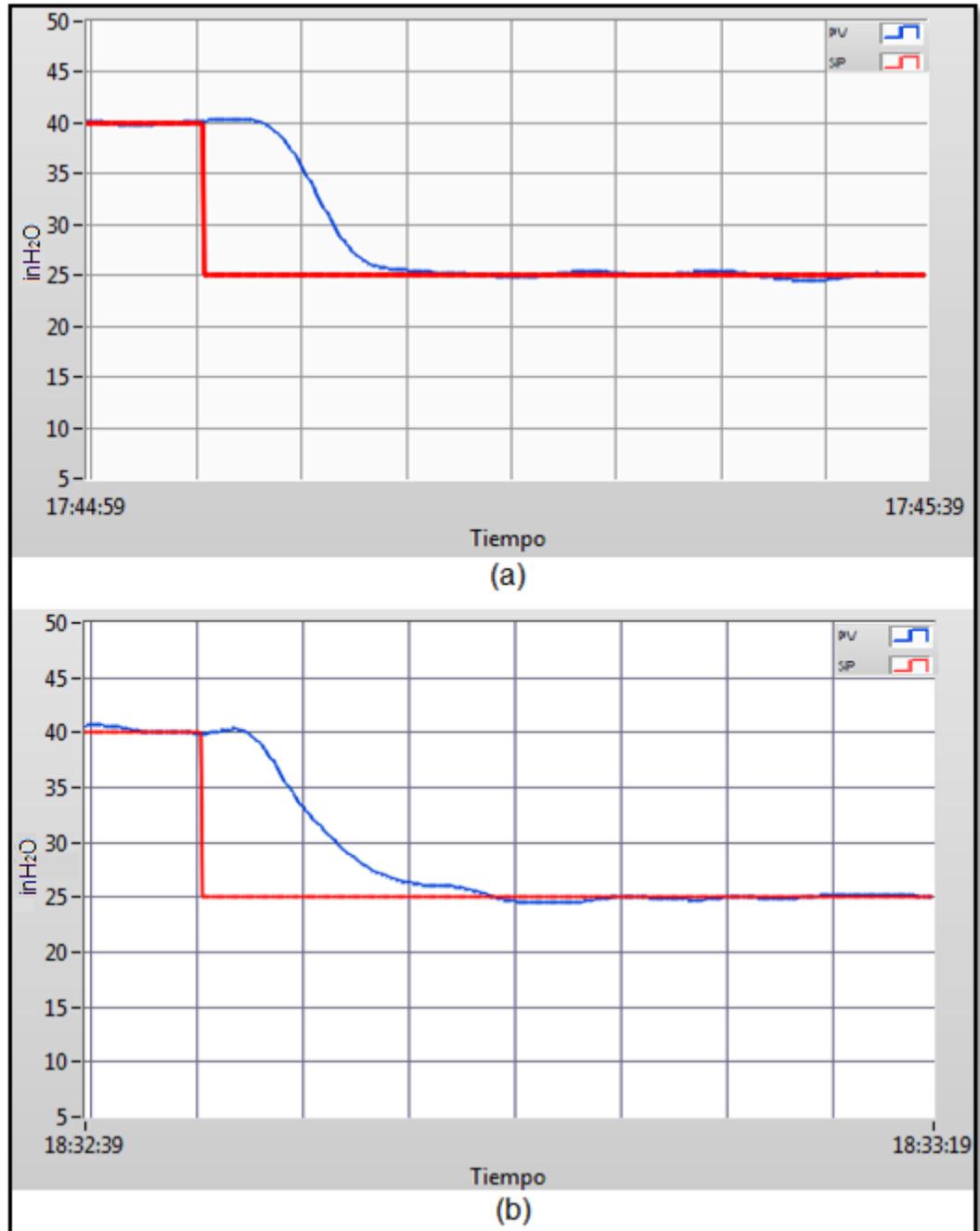


Figura 0.7: Setpoint 40 – 25 inH₂O, (a) MPC, (b) PID

Elaborado por: López, V y Paredes, A

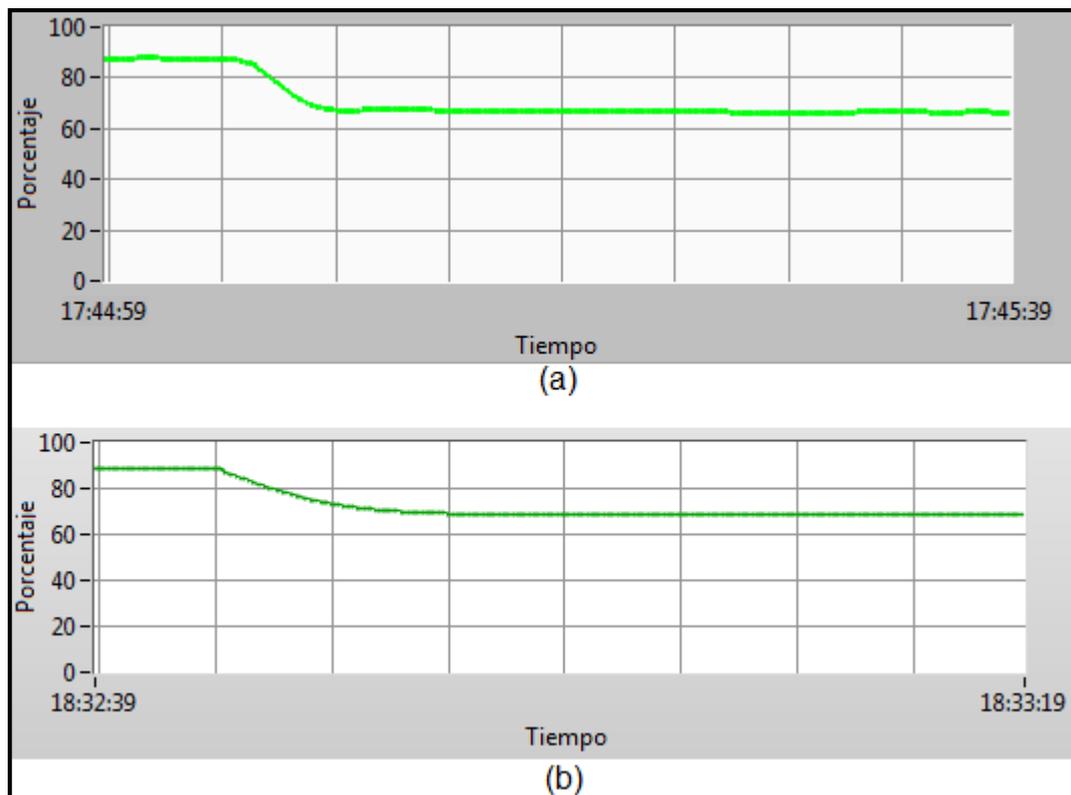


Figura 0.8: Acción de control 40 – 25 inH₂O, (a) MPC, (b) PID

Elaborado por: López, V y Paredes, A

En la Figura 0.9 se puede apreciar la similitud en las respuestas de los controladores bajo estudio, con una pequeña diferencia de tiempo de estabilización a favor del control MPC, implicando que los algoritmos de control resultan similares para este particular rango del proceso, notándose además las semejanzas en las acciones de control correspondientes expuestas en la Figura 0.10.

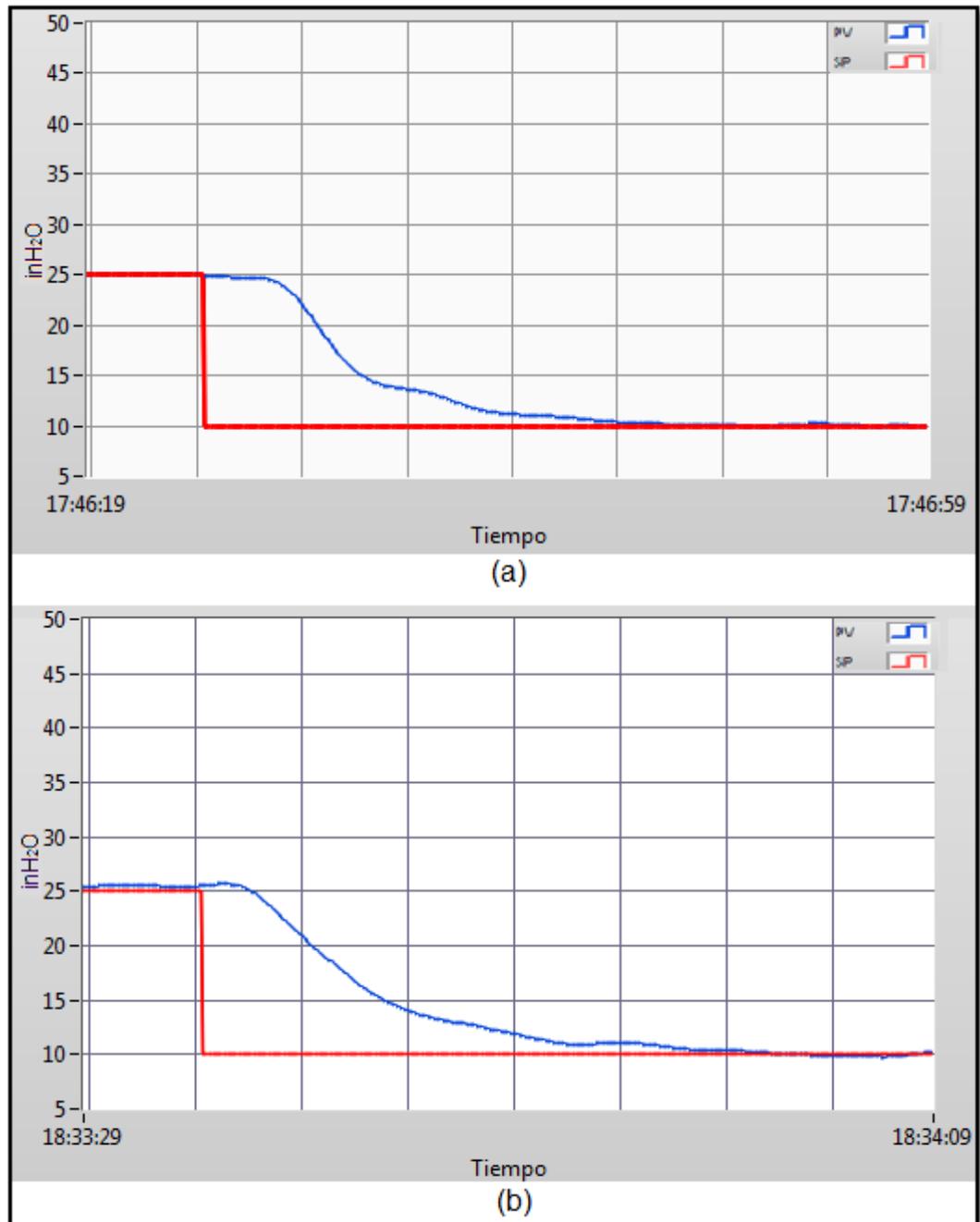


Figura 0.9: Setpoint 25 – 10 inH₂O, (a) MPC, (b) PID

Elaborado por: López, V y Paredes, A

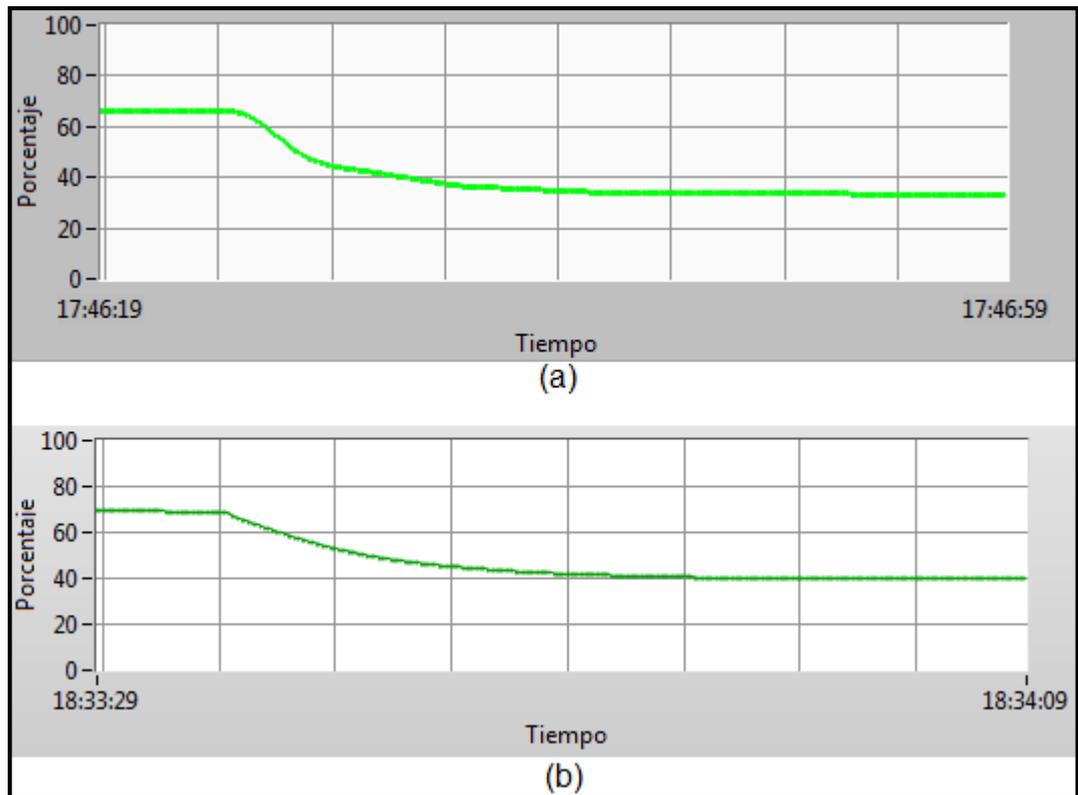


Figura 0.10: Acción de control 25 – 10 inH₂O, (a) MPC, (b) PID

Elaborado por: López, V y Paredes, A

En cambios ascendentes el controlador predictivo demuestra su ventaja respecto al controlador proporcional-integral-derivativo, de la Figura 0.11 se deduce que la respuesta del MPC no tiene sobreimpulso respecto al control tradicional, dando lugar a una mejora en el tiempo de establecimiento del proceso y en las acciones del actuador ilustradas en la Figura 0.12 frente al cambio de 10 – 40 inH₂O.

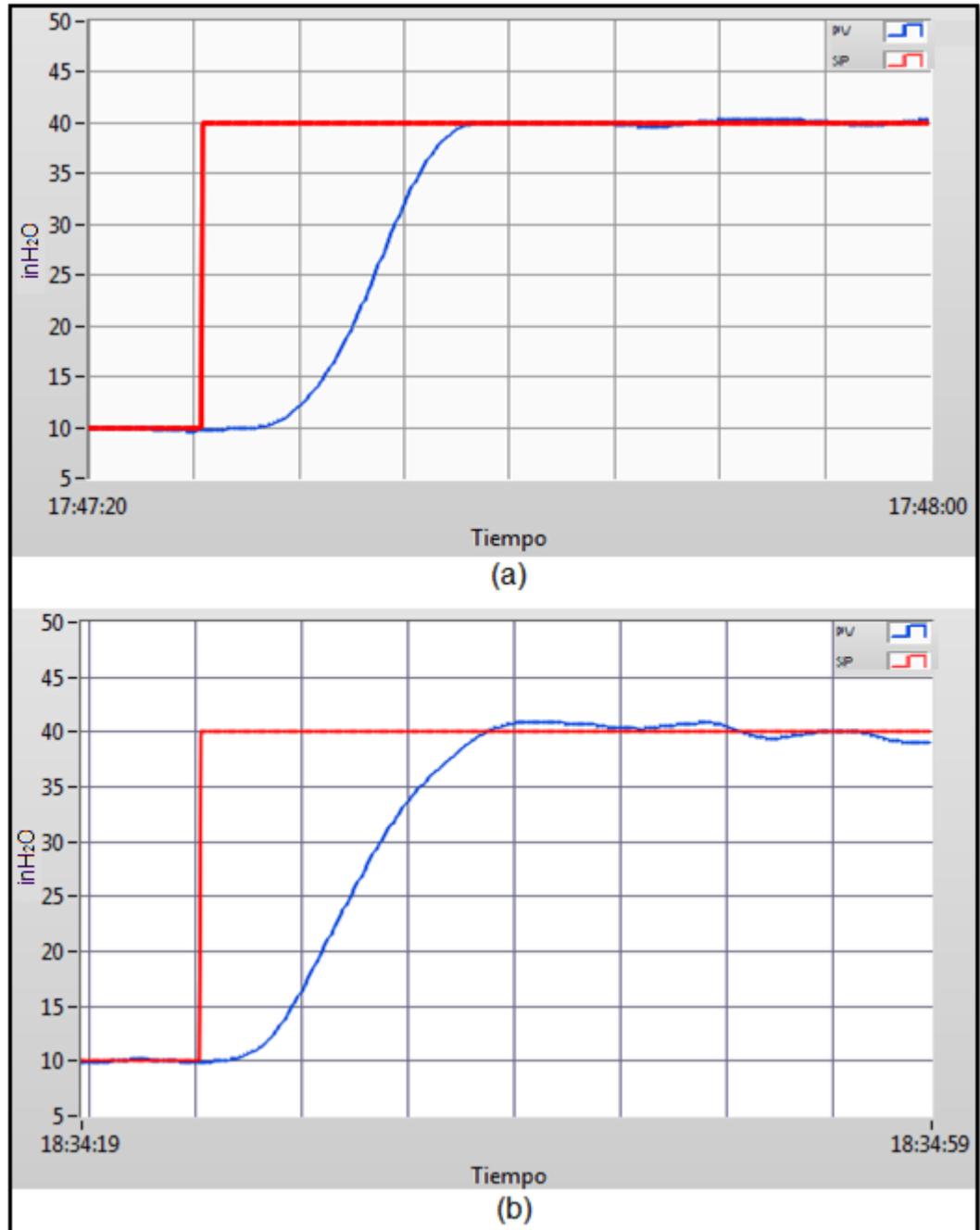


Figura 0.11: Setpoint 10 – 40 inH₂O, (a) MPC, (b) PID

Elaborado por: López, V y Paredes, A

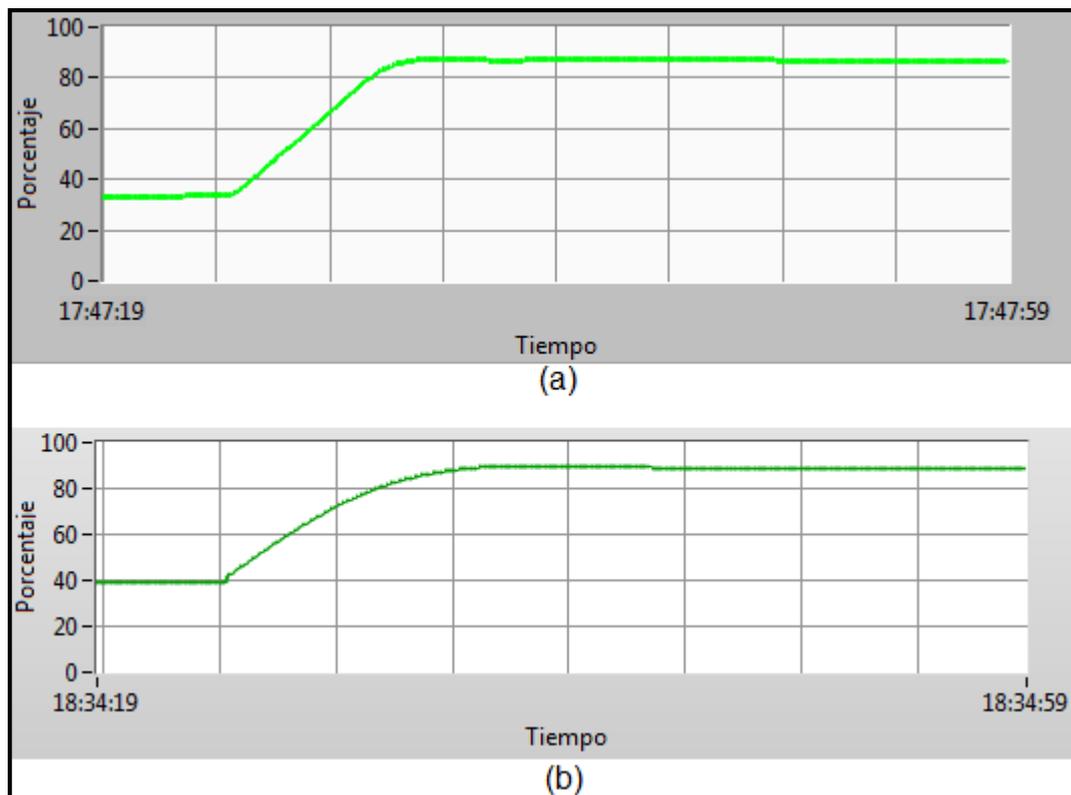


Figura 0.12: Acción de control 10 – 40 inH₂O, (a) MPC, (b) PID

Elaborado por: López, V y Paredes, A

Para el cambio hacia niveles cercanos al límite inferior, nuevamente se destaca la similitud en los controladores, como la Figura 0.13 muestra, llegando incluso a tener tiempos de estabilidad mínimamente diferenciables, a pesar de notar una diferencia en las señales de control como se indica en la Figura 0.14, el resultado en la respuesta del proceso de flujo resulta ser análogo para los controles MPC y PID.

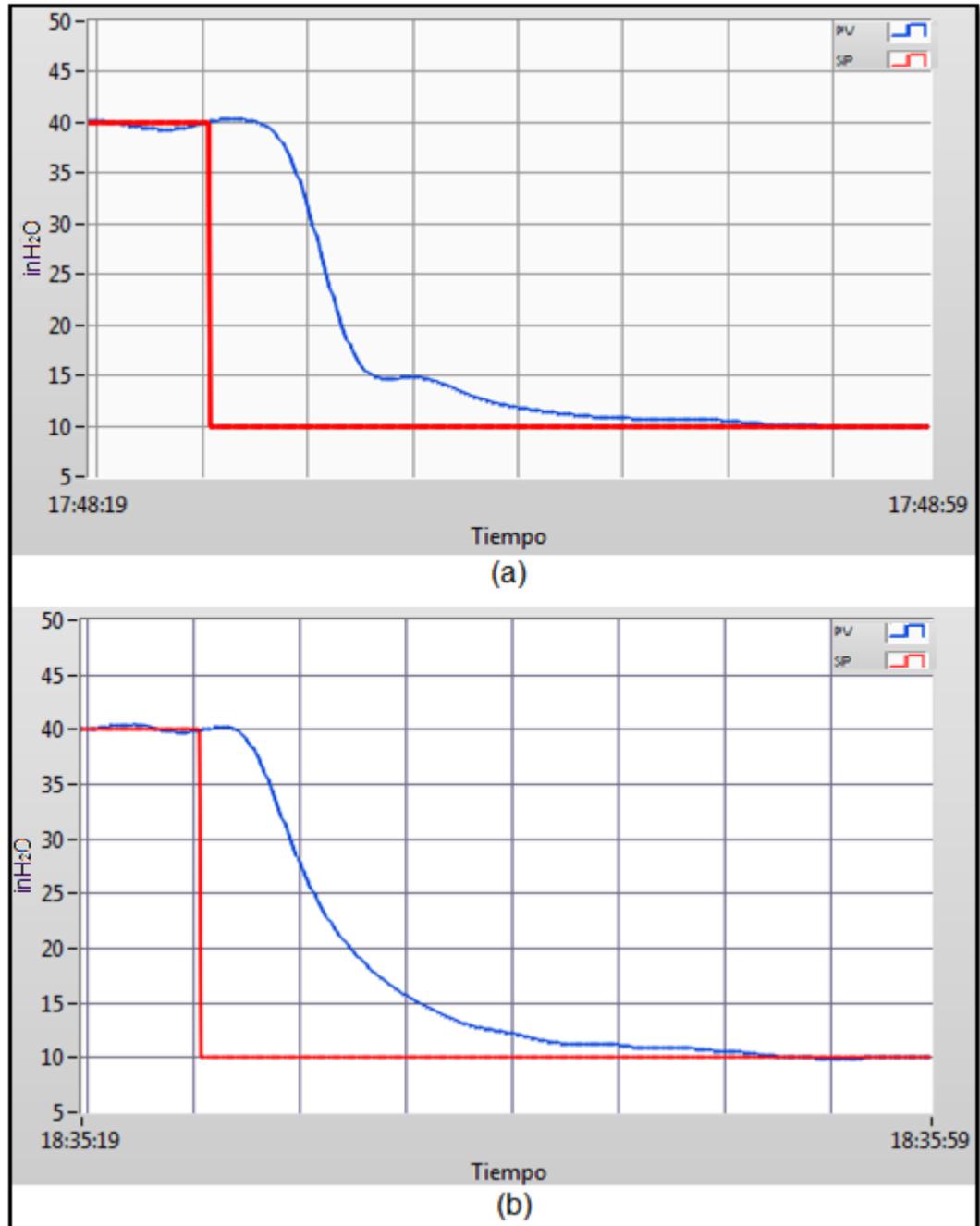


Figura 0.13: Setpoint 40 – 10 inH₂O, (a) MPC, (b) PID

Elaborado por: López, V y Paredes, A

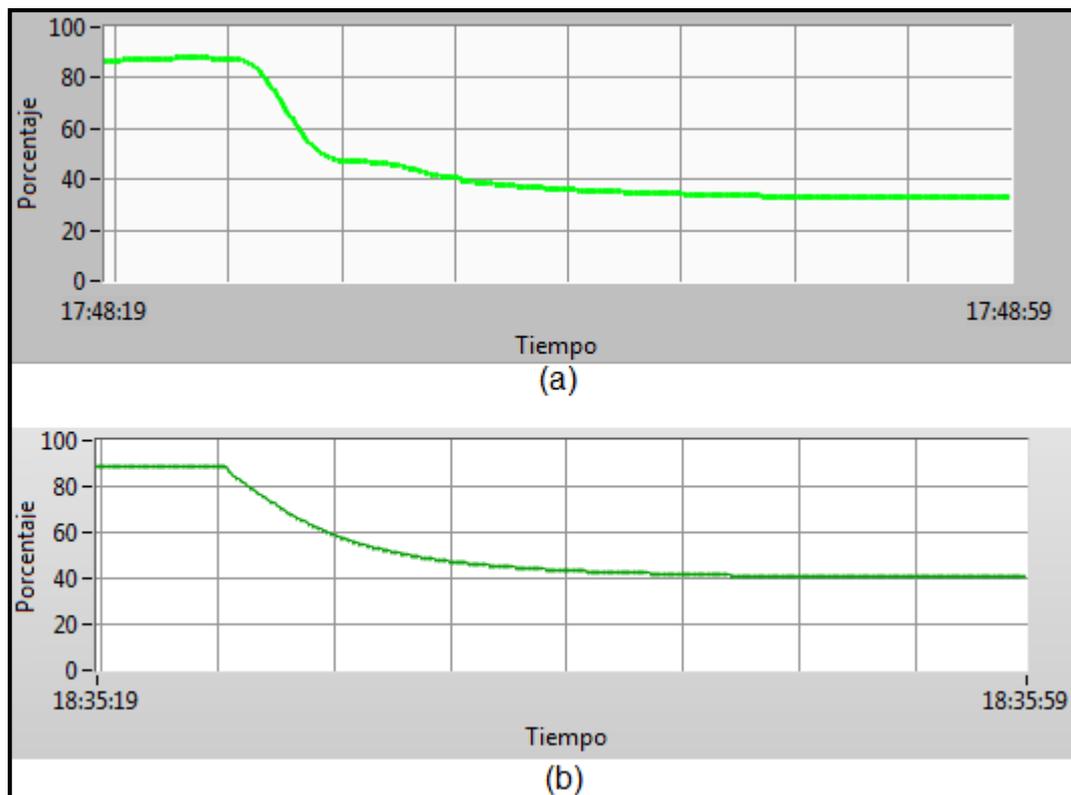


Figura 0.14: Acción de control 40 – 10 inH₂O, (a) MPC, (b) PID

Elaborado por: López, V y Paredes, A

De los ensayos anteriores, se condensa los resultados en la Tabla 0.2, expresando cuantitativamente las diferencias y similitudes de un control predictivo basado en modelos (MPC) y un control proporcional-integral-derivativo (PID) realizados sobre un proceso industrial de flujo.

Tabla 0.2. Comparación entre controles MPC y PID implementados

Setpoint (inH₂O)	Tiempo de estabilización aprox. (s)		Porcentaje de sobrepulso aprox. (%)	
	MPC	PID	MPC	PID
10 – 25	15	22	0	2
25 – 40	11	15	2	5
40 – 25	10	18	0	3
25 – 10	20	27	0	0
10 – 40	13	20	0	5
40 – 10	26	27	0	0
Promedio	15,83	21,50	0,33	2,5

Basados en las medias aritméticas se puede señalar una mejora en el tiempo de establecimiento de aproximadamente cinco segundo a favor del control MPC y una reducción cercana al dos por ciento en el nivel de sobrepulso en relación al control PID.

CAPÍTULO 6

CONCLUSIONES Y RECOMENDACIONES

1.14. CONCLUSIONES

La respuesta que se obtuvo de un control predictivo basado en modelos realizado sobre un proceso de flujo en relación a un control tradicional como el PID es marcadamente superior, destacando las características principales como: menor tiempo de estabilización para cambios en los valores de referencia, disminución del porcentaje de sobreimpulso siendo cercano a cero para el MPC y una señal de control más estable y eficiente para el actuador.

Una parte integral del controlador predictivo es el modelo matemático del sistema a controlar, por tanto, la estimación del mismo es parte fundamental del procedimiento, el grado de coincidencia entre el proceso real y el estimado influye de manera significativa en los resultados finales del controlador, pudiendo mejorarlo notablemente o incluso agravar las respuestas del sistema frente a otros tipos de controles más simples. Por consiguiente esta primera etapa del desarrollo debe ser ejecutada correctamente.

Las simulaciones resultan muy favorables para entender el funcionamiento de un control predictivo, resultando más cómodo ejecutar pruebas bajo el proceso modelado y entender cada uno de los parámetros que componen a un controlador MPC antes de proceder a implementarlo de manera física en el sistema real, se demostró que los resultados de un control simulado respecto a las respuestas reales del sistema son semejantes, dependiendo el grado de acierto de la coincidencia entre el modelo real y el estimado.

El controlador predictivo no se enmarca únicamente en mantener el proceso en un nivel fijado, como es el enfoque de un control tradicional, además busca optimizar los recursos para hacerlo y maximizar la eficiencia del sistema completo, resaltando las restricciones que constan como

parámetros del MPC para evitar valores inadecuados que deban ser evadidos en el proceso, limitando el funcionamiento del sistema en sus rangos adecuados, de igual manera confinando los niveles del actuador a sus valores nominales, impidiendo daños en el equipo.

Los parámetros de optimización disponibles en el controlador MPC, hacen que se pueda obtener un control adecuado a los requerimientos, a diferencia de controladores que no permiten manipular características como el nivel de estabilidad, pendientes de cambio ascendentes y descendentes para la señal de control e influencia del error, adecuando así la respuesta necesaria para cada proceso en particular y consiguiendo adaptar la acción de control a las necesidades del equipo actuador.

Como característica adicional entre un controlador predictivo y un control tradicional, el tiempo de implementación es menor para los controles básicos, principalmente aquellos que no dependen de un modelo matemático de la planta y se basan exclusivamente en cálculos del error para determinar una acción de control, en consecuencia el tipo de control a efectuar depende del sistema y los recursos necesarios para hacerlo.

El control predictivo ha visto un lento avance de aceptación a nivel industrial, principalmente por los equipos necesarios para ejecutarlo, necesitando dispositivos con capacidad de procesamiento superiores a los normalmente usados en controles industriales, con el avance tecnológico esto ha dejado de ser un impedimento y este estudio demuestra que los equipos actuales tienen las características necesarias para su implementación.

1.15. RECOMENDACIONES

La estimación del modelo matemático del proceso es una fracción muy trascendental dentro de un control MPC, por ende, se recomienda culminar esta etapa con los mejores resultados posibles, realizando las verificaciones y correcciones necesarias antes de proceder con la siguiente fase de la implementación.

Se recomienda dimensionar adecuadamente los límites de trabajo del proceso así como los rangos del actuador, debido a que un funcionamiento del sistema o acción de control fuera de las restricciones configuradas en el MPC podría llevar a un mal funcionamiento del mismo o a respuestas inestables del proceso.

Las compilaciones necesarias para los programas sobre la FPGA requieren de un tiempo prolongado, incluso si los cambios realizados son mínimos, por tanto, se pide realizar la programación sencilla y estable, que no requiera cambios a futuro, los procedimientos que se crean van a modificarse continuamente, hacerlo en los VI sobre el *chasis* de la myRIO o sobre el computador de ser el caso.

BIBLIOGRAFÍA

- [1] L. Ljung, System Identification: Theory for the user, Perason Education, 1998.
- [2] M. E. L. Guillén, «Identificación de Sistemas. Aplicación al modelado de un motor de continua,» UTPL, Loja.
- [3] D. R. Ramírez, «esi2,» Perspectiva General del Control Predictivo Min-Max, [En línea]. Available: <http://www.esi2.us.es/~danirr/CursoDoct.pdf>. [Último acceso: 23 Septiembre 2014].
- [4] M. d. I. M. Perez, «Dialnet,» Control Predictivo con modos deslizantes, 04 Junio 2004. [En línea]. Available: <https://www.google.com.ec/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0CB0QFjAA&url=http%3A%2F%2Fdialnet.unirioja.es%2Fdescarga%2Ftesis%2F152.pdf&ei=IH0oVL aDKpPKggTjjoGICA&usg=AFQjCNEizhX1npXQ3XxmTb6OGOE4 QvfhgQ&bvm=bv.76247554,d.eXY>. [Último acceso: 20 Septiembre 2014].
- [5] N. Instrument, «National Instrument,» [En línea]. Available: <http://sine.ni.com/psp/app/doc/p/id/psp-1166/lang/es>. [Último acceso: 12 Agosto 2014].
- [6] Omega, «omega,» 2005. [En línea]. Available: <http://www.omega.com/manuals/manualpdf/M4139.pdf>. [Último acceso: 21 Octubre 2014].

- [7] N. Instrument, «National Instrument,» [En línea]. Available: <http://sine.ni.com/nips/cds/view/p/lang/es/nid/1644>. [Último acceso: 10 Agosto 2014].
- [8] H. A. M. Díaz, «LICEUS,» Mayo 2003. [En línea]. Available: <http://www.liceus.com/cgi-bin/ac/pu/AutomatizacionMedioambiental.pdf>. [Último acceso: 20 07 2014].
- [9] J. Qin y T. Badgwell, «AN OVERVIEW OF INDUSTRIAL MODEL,» *Research Gate*, vol. I, nº 1, p. 1, 1997.
- [10] G. Goebel, «UFSC,» UFSC, 1 Junio 2003. [En línea]. Available: http://www.das.ufsc.br/~camponog/Disciplinas/DAS-5341/Fuzzy_logic/Intro_fuzzy_logic.pdf. [Último acceso: 19 07 2014].
- [11] M. Fernández, «AIE,» [En línea]. Available: http://www.aie.cl/files/file/comites/ca/abc/Control_Avanzado.pdf. [Último acceso: 15 Julio 2014].
- [12] MathWorks, «MathWorks,» Mathworks, [En línea]. Available: <http://www.mathworks.com/videos/introduction-to-system-identification-toolbox-68901.html>. [Último acceso: 15 Agosto 2014].

LINKOGRAFÍA

- [1] L. Ljung, System Identification: Theory for the user, Perason Education, 1998.
- [2] M. E. L. Guillén, «Identificación de Sistemas. Aplicación al modelado de un motor de continua,» UTPL, Loja.
- [3] D. R. Ramírez, «esi2,» Perspectiva General del Control Predictivo Min-Max, [En línea]. Available: <http://www.esi2.us.es/~danirr/CursoDoct.pdf>. [Último acceso: 23 Septiembre 2014].
- [4] M. d. I. M. Perez, «Dialnet,» Control Predictivo con modos deslizantes, 04 Junio 2004. [En línea]. Available: <https://www.google.com.ec/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0CB0QFjAA&url=http%3A%2F%2Fdialnet.unirioja.es%2Fdescarga%2Ftesis%2F152.pdf&ei=IH0oVLaDKpPKggTjjoGICA&usg=AFQjCNEizhX1npXQ3XxmTb6OGOE4QvfhgQ&bv m=bv.76247554,d.eXY>. [Último acceso: 20 Septiembre 2014].
- [5] N. Instrument, «National Instrument,» [En línea]. Available: <http://sine.ni.com/psp/app/doc/p/id/psp-1166/lang/es>. [Último acceso: 12 Agosto 2014].
- [6] Omega, «omega,» 2005. [En línea]. Available: <http://www.omega.com/manuals/manualpdf/M4139.pdf>. [Último acceso: 21 Octubre 2014].
- [7] N. Instrument, «National Instrument,» [En línea]. Available: <http://sine.ni.com/nips/cds/view/p/lang/es/nid/1644>. [Último acceso: 10 Agosto 2014].

- [8] H. A. M. Díaz, «LICEUS,» Mayo 2003. [En línea]. Available: <http://www.liceus.com/cgi-bin/ac/pu/AutomatizacionMedioambiental.pdf>. [Último acceso: 20 07 2014].
- [9] J. Qin y T. Badgwell, «AN OVERVIEW OF INDUSTRIAL MODEL,» *Research Gate*, vol. I, nº 1, p. 1, 1997.
- [10] G. Goebel, «UFSC,» UFSC, 1 Junio 2003. [En línea]. Available: http://www.das.ufsc.br/~camponog/Disciplinas/DAS-5341/Fuzzy_logic/Intro_fuzzy_logic.pdf. [Último acceso: 19 07 2014].
- [11] M. Fernández, «AIE,» [En línea]. Available: http://www.aie.cl/files/file/comites/ca/abc/Control_Avanzado.pdf. [Último acceso: 15 Julio 2014].
- [12] MathWorks, «MathWorks,» Mathworks, [En línea]. Available: <http://www.mathworks.com/videos/introduction-to-system-identification-toolbox-68901.html>. [Último acceso: 15 Agosto 2014].

UNIVERSIDAD DE LAS FUERZAS ARMADAS – ESPE
INGENIERIA EN ELECTRÓNICA E INSTRUMENTACIÓN

CERTIFICACIÓN

Se certifica que el presente trabajo fue desarrollado por la Srta. LÒPEZ CAISAGUANO VERÒNICA GABRIELA y el Sr. PAREDES ANCHATIPAN ALEX DARWIN.

LÒPEZ CAISAGUANO, VERÒNICA.

C.C. 150084984-7

PAREDES ANCHATIPÁN, ALEX.

C.C. 050361493-5

ING. JACQUELINE, LLANOS
DIRECTOR

ING. DIEGO, ORTIZ
CODIRECTOR

ING. FRANKLIN, SILVA
DIRECTOR DE LA CARRERA

DR. RODRIGO, VACA
SECRETARIO ACADÉMICO