



**UNIVERSIDAD TÉCNICA DE AMBATO**  
**FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E**  
**INDUSTRIAL**  
**POSGRADOS**

**PROGRAMA DE MAESTRÍA EN MATEMÁTICA APLICADA**  
**MODALIDAD DE TITULACIÓN PROYECTO DE DESARROLLO**

Trabajo de titulación previo a la obtención del grado académico de  
Magíster en Matemática Aplicada

**Tema:** “Modelación matemática para un control robusto de la planta  
Festo MPS-PA Compact Workstation mediante la normativa IEC-61499”

**Autor:** Ing. Sergio Patricio Bustos Pulluquitin

**Director:** Ing. Marcelo Vladimir García Sánchez, Ph.D.

Ambato – Ecuador

2021

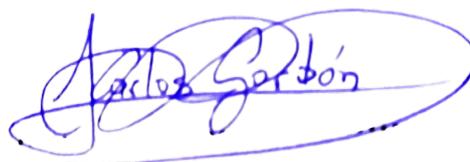
## APROBACIÓN DEL TRABAJO DE TITULACIÓN

A la Unidad de Titulación/Unidad Académica de Titulación de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial. El Tribunal receptor de la Defensa del Trabajo de Titulación presidido por Ing. Elsa Pilar Urrutia Urrutia Mg., e integrado por los señores: Ing. Carlos Diego Gordón Gallegos Ph.D. e Ing. Ernesto Antonio Ponsot Balaguer Ph.D., designados por la Unidad de Titulación de la Universidad Técnica de Ambato, para recibir el Trabajo de Titulación con el tema: “Modelación matemática para un control robusto de la planta Festo MPS-PA Compact Workstation mediante la normativa IEC-61499”, elaborado y presentado por el Ing. Sergio Patricio Bustos Pulluquitin, para optar por el Grado Académico de Magister en Matemática Aplicada; una vez escuchada la defensa oral del Trabajo de Titulación el Tribunal aprueba y remite el trabajo para uso y custodia en las bibliotecas de la Universidad Técnica de Ambato.



Ing. Elsa Pilar Urrutia Urrutia Mg.

Presidente y Miembro del Tribunal de Defensa



Ing. Carlos Diego Gordón Gallegos Ph.D.

Miembro del Tribunal de Defensa



Ing. Ernesto Antonio Ponsot Balaguer Ph.D.

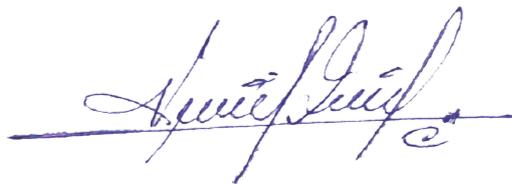
Miembro del Tribunal de Defensa

## AUTORÍA DEL TRABAJO DE TITULACIÓN

La responsabilidad de las opiniones, comentarios y críticas emitidas en el trabajo de titulación presentado con el tema: “Modelación matemática para un control robusto de la planta Festo MPS-PA Compact Workstation mediante la normativa IEC-61499”, le corresponde exclusivamente al: Ing. Sergio Patricio Bustos Pulluquitin, autor bajo la dirección del: Ing. Marcelo Vladimir García Sánchez, Ph.D, director del trabajo de investigación; y el patrimonio intelectual pertenece a la Universidad Técnica de Ambato.

A handwritten signature in blue ink, consisting of a large, stylized 'S' followed by the name 'SERGIO BUSTOS' written in a smaller, more legible script.

Ing. Sergio Patricio Bustos Pulluquitin

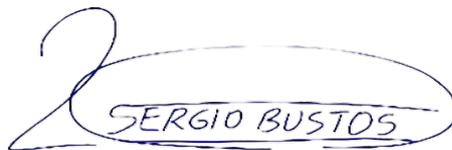
A handwritten signature in blue ink, featuring a highly stylized and cursive script that is difficult to decipher without context.

Ing. Marcelo Vladimir García Sánchez Ph.D.

## **DERECHOS DE AUTOR**

Autorizo a la Universidad Técnica de Ambato, para que el trabajo de titulación, sirva como un documento disponible para su lectura, consulta y procesos de investigación, según las normas de la Institución.

Cedo los derechos de mi trabajo de titulación, con fines de difusión pública, además apruebo la reproducción de éste, dentro de las regulaciones de la Universidad Técnica de Ambato.

A handwritten signature in blue ink. The signature consists of a large, stylized initial 'S' followed by the name 'SERGIO BUSTOS' written in capital letters. The entire signature is enclosed within a hand-drawn oval shape.

Ing. Sergio Patricio Bustos Pulluquitin

# Índice de figuras

1.	Sistema de variación de nivel $h$ en un tanque . . . . .	5
2.	Representación de un sistema no lineal . . . . .	6
3.	Elementos de la respuesta al escalón de un sistema de primer orden. . . . .	9
4.	Componentes de la respuesta de un sistema al escalón. . . . .	9
5.	Porcentaje de respuesta alcanzado para diversos valores de tiempo: $t = 1/a_0, t = 2/a_0, t = 3/a_0$ y $t = 4/a_0$ . . . . .	10
6.	Respuesta al escalón de tres diferentes sistemas de primer orden. . . . .	11
7.	Obtención de $t$ a partir de la respuesta al escalón de un sistema de primer orden. . . . .	11
8.	Recta ajustada. . . . .	13
9.	Diagrama de bloque de un sistema de control discreto básico. . . . .	14
10.	Números Fibonnaci. . . . .	16
11.	Gráfico de una función y aproximaciones alternativas al área bajo la curva en un solo intervalo de tiempo. . . . .	16
12.	El prototipo del sistema de datos muestreados. . . . .	19
13.	Salida $D/A$ para entrada de pulso unitario. . . . .	20
14.	Estrategia MPC. . . . .	22
15.	Estructura básica del MPC. . . . .	23
16.	Respuesta libre y forzada. . . . .	26
17.	Trayectoria de referencia. . . . .	27
18.	Ley de Control GPC. . . . .	34
19.	Restricciones en la señal de control. . . . .	35
20.	Bloque de Función. . . . .	46
21.	Bloque de Función Secuencia Interna. . . . .	47
22.	Aplicación Distribuida. . . . .	47
23.	Aplicación Distribuida. . . . .	48
24.	Modelo de Aplicación con PLC. . . . .	48
25.	Bloque de Función Básico. . . . .	49
26.	Bloque de Función Compuesto. . . . .	49
27.	Bloque de Función de Servicio. . . . .	50
28.	Entorno de desarrollo 4DIAC. . . . .	50
29.	Arquitectura OPC UA. . . . .	52
30.	Arquitectura OPC UA. . . . .	52

31.	Célula analógica FESTO MPS PA® Compact Workstation. . . . .	55
32.	Diagrama PI&D de nivel. . . . .	55
33.	Conexión física del controlador predictivo. . . . .	56
34.	Sistema de nivel representado por la ecuación (5). . . . .	56
35.	Diagrama experimento respuesta al escalón. . . . .	58
36.	Respuesta del sistema al escalón unitario. . . . .	58
37.	Datos a la respuesta al escalón. . . . .	59
38.	Gráfica aproximada a los datos recolectados. . . . .	60
39.	Respuesta al escalón en tiempo continuo y discreto. . . . .	61
40.	Tarjeta BeagleBone Black. . . . .	76
41.	Escenario general de desarrollo del software en 4DIAC. . . . .	77
42.	Estructura en C++ de SIFB GPC_CONTROL. . . . .	77
43.	Bloque de Función ANALOG_INPUT. . . . .	78
44.	Bloque de Función GPC_CONTROL. . . . .	79
45.	Bloque de Función Cliente OPC UA. . . . .	80
46.	Diagrama de Funciones Controlados GPC. . . . .	80
47.	Cliente OPC UA UAExpert. . . . .	81
48.	Escenario de aplicación del control de nivel. . . . .	81
49.	Mapeo de un simple escenario OPC UA. . . . .	82
50.	Atributos incluidos en un Nodo. . . . .	82
51.	Objeto de dispositivo que expone referencias. . . . .	83
52.	Diagrama de funciones Servidor OPC UA. . . . .	84
53.	Servidor OPC UA implementado. . . . .	85
54.	Aplicación de control en 4DIAC-IDE. . . . .	86
55.	Configuración del Sistema. . . . .	87
56.	Interfaz Hombre Máquina (HMI). . . . .	87
57.	Configuración del recurso HMI. . . . .	88
58.	Configuración de Recurso BBB_NIVEL. . . . .	88
59.	Distribución Física del Sistema Distribuido. . . . .	89
60.	Simple Servidor OPC UA. . . . .	90
61.	UAExpert Cliente OPC-UA. . . . .	91
62.	Prueba de funcionamiento 1. . . . .	91
63.	Variables del controlador. . . . .	92
64.	Variables del controlador. . . . .	92
65.	Prueba de funcionamiento 2. . . . .	92

66.	Variables del controlador. . . . .	93
67.	Variables del controlador. . . . .	93
68.	Variables del controlador. . . . .	93
69.	Experimento de setpoint=5 litros . . . . .	94
70.	Controlador GPC sin restricciones. . . . .	94
71.	Variables del controlador. . . . .	95
72.	Controlador GPC con restricciones. . . . .	95
73.	Variables del controlador. . . . .	95
74.	Error Cuadrático Medio. . . . .	96
75.	Archivos FORTE ubicados en la tarjeta Beaglebone Black. . . . .	104
76.	Carpeta TESIS_FINAL creada en /src/modules, contiene los archivos del nuevo módulo. . . . .	104
77.	Carpeta TESIS_FINAL contiene funciones y archivo CMakeLists.txt generados en la carpeta de módulo. . . . .	105
78.	Contenido del archivo CMakelists.txt. . . . .	105
79.	Contenido del archivo CMakelists.txt . . . . .	106
80.	Circuito de acondicionamiento de tensión para la entrada analógica. . . . .	107
81.	Circuito conversión de corriente 4-20mA a un voltaje 0-5V. . . . .	108
82.	Circuito de acondicionamiento de salida PWM. . . . .	109
83.	Tarjeta BeagleBone Black. . . . .	109
84.	Secuencia del comportamiento del GPC. . . . .	110
85.	Secuencia del comportamiento del GPC. . . . .	110
86.	Secuencia del comportamiento del GPC. . . . .	110
87.	Secuencia del comportamiento del GPC. . . . .	111
88.	Secuencia del comportamiento del GPC. . . . .	111
89.	Secuencia del comportamiento del GPC. . . . .	111
90.	Secuencia del comportamiento del GPC. . . . .	112
91.	Secuencia del comportamiento del GPC. . . . .	112

## Índice de tablas

1.	Tarjetas Embebidas y PLC compatibles con IEC-61499. . . . .	76
2.	Características de la tarjeta BBB. . . . .	76
3.	Descripción de las funciones del Servidor OPC UA. . . . .	80
4.	Descripción de las funciones del Servidor OPC UA. . . . .	85
5.	Análisis resultados regresión. . . . .	93
6.	Componentes electrónicos. . . . .	109
7.	Resultados experimentales. . . . .	113

## **AGRADECIMIENTO**

A mi padre y madre por todo el apoyo y confianza brindada.

A los docentes de la maestría por todos conocimientos brindados.

A Marcelo García por nuevamente aceptar ser tutor de mi proyecto de investigación.

Sergio Patricio Bustos Pulluquitin

## **DEDICATORIA**

El presente proyecto de investigación se lo dedico a mis padres Patricio y Solanda.

Sergio Patricio Bustos Pulluquitin

# UNIVERSIDAD TÉCNICA DE AMBATO

## FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL

### MAESTRÍA EN MATEMÁTICA APLICADA

#### INFORMACIÓN GENERAL

**TEMA:** “Modelación matemática para un control robusto de la planta Festo MPS-PA Compact Workstation mediante la normativa IEC-61499”

**AUTOR:** Ing. Sergio Patricio Bustos Pulluquitin

**DIRECTOR:** Ing. Marcelo Vladimir García Sánchez, Ph.D.

**LÍNEA DE INVESTIGACIÓN:**

- Matemática Computacional

**FECHA:**14-01-2021

#### RESUMEN EJECUTIVO

En el presente trabajo de investigación se realizó la implementación de un control GPC con restricciones en una planta de nivel FESTO MPS-PA Compact Workstation mediante el modelado de bloques de funciones, obteniendo así una aplicación de control industrial con lógica descentralizada el cual es uno de los objetivo del estándar IEC 61499. El IDE empleado compatible con IEC 61499 fue el 4DIAC IDE y el entorno ejecución (runtime) 4DIAC FORTE, además se utilizó la tarjeta BBB para la lectura y escritura de datos de sensores y actuadores respectivamente. El modelo matemático de la planta de nivel representado como función de transferencia fue encontrada mediante dos métodos: el primero utilizando ecuaciones diferenciales lineales y el segundo mediante datos experimentales de la respuesta al escalón de un sistema de primer orden. Se implementaron tres tipos de restricciones: en el incremento de control ( $\Delta u(t)$ ) para evitar esfuerzos innecesarios en el actuador y las restricciones de salida ( $y(t)$ ) y control ( $u(t)$ ) que evitarán daños a la planta, el algoritmo GQPA fue empleado para optimizar la función objetivo.

**Palabras Claves:** Bloque de Funciones, Control Predictivo, Eventos, Función Transferencia, Modelo Matemático, Optimización, Predicción, Restricciones, Sistemas Distribuidos, Sistema Lineal.

# TECHNICAL UNIVERSITY OF AMBATO

## FACULTY OF SYSTEMS, ELECTRONIC AND INDUSTRIAL ENGINEERING

### MASTER OF APPLIED MATHEMATICS

#### GENERAL INFORMATION

**THEME:** “Mathematical modeling for robust control of the Festo MPS-PA Compact Workstation plant according to IEC-61499”

**AUTHOR:** Ing. Sergio Patricio Bustos Pulluquitin

**DIRECTED BY:** Ing. Marcelo Vladimir García Sánchez, Ph.D.

**LINE OF RESEARCH:**

- Computational Mathematics

**DATE:** 14-01-2021

#### EXECUTIVE SUMMARY

In the present research work, the implementation of a GPC control with restrictions was carried out in a FESTO MPS-PA Compact Workstation level plant through the modeling of function blocks, thus obtaining an industrial control application with decentralized logic which is one of the objectives of the IEC 61499 standard. The IDE used compatible with IEC 61499 was the 4DIAC IDE and the runtime environment (runtime) 4DIAC FORTE, in addition the BBB card was used for reading and writing data from sensors and actuators respectively. The mathematical model of the level plant represented as a transfer function was found by two methods: the first using linear differential equations and the second using experimental data of the response to the step of a first order system. Three types of restrictions were implemented: in the control increment ( $\Delta u(t)$ ) to avoid unnecessary efforts in the actuator and the output restrictions ( $y(t)$ ) and control ( $u(t)$ ) that will prevent damage to the plant, the GQPA algorithm was used to optimize the objective function.

**Keywords:** Block of Functions, Predictive Control, Events, Transfer Function, Mathematical Model, Optimization, Prediction, Constraints, Distributed Systems, Linear System.

# 1. CAPÍTULO I

## PROBLEMA DE INVESTIGACIÓN

### 1.1. Introducción.

En el presente trabajo de investigación se realizó la implementación de un control GPC con restricciones en una planta de nivel FESTO MPS-PA Compact Workstation mediante el modelado de bloques de funciones, obteniendo así una aplicación de control industrial con lógica descentralizada el cual es uno de los objetivo del estándar IEC 61499 y el protocolo de comunicación OPC UA. El entorno de desarrollo empleado compatible con IEC 61499 fue el 4DIAC IDE y el entorno ejecución (runtime) 4DIAC FORTE, además se utilizó la tarjeta de desarrollo BBB para la lectura y escritura de datos de sensores y actuadores respectivamente.

El modelo matemático de la planta de nivel representado como función de transferencia necesaria para implementar el algoritmo del GPC fue encontrado mediante dos métodos: el primero utilizando ecuaciones diferenciales lineales basados en principios de la física clásica y el segundo mediante datos experimentales de la respuesta al escalón de un sistema de primer orden.

Se implementaron tres tipos de restricciones: en el incremento de control ( $\Delta u(t)$ ) para evitar esfuerzos innecesarios en el actuador y las restricciones de salida ( $y(t)$ ) y control ( $u(t)$ ) que evitarán daños al personal supervisor de planta y posibles pérdidas económicas, el algoritmo GQPA fue empleado para optimizar la función objetivo.

Se modelaron tres tipos de bloques de función para la implementación del GPC: el primero lee el valor del sensor de nivel y verifica el correcto funcionamiento del GPC, en el segundo se implementó el algoritmo de control con las respectivas restricciones y en el tercero se implementó un cliente OPC UA el cual publicará los datos en el servidor OPC UA.

El servidor OPC UA fue implementado en la estación de ingeniería (PC) el cual recibe los datos del bloque de función cliente OPC UA, éstos datos son leídos por dos clientes OPC UA uno para visualizar gráficamente el funcionamiento de las variables que interactúan en la aplicación y el otro cliente OPC UA envía los datos al actuador mediante PWM de la tarjeta BBB.

## 1.2. Justificación.

Mientras las empresas manufactureras luchan por competir en los impredecibles y cambiantes mercados globales, existe una urgencia cada vez mayor de mejorar la agilidad de los sistemas manufactureros. Para realizar productos competitivos e innovadores, las empresas deben diseñar y crear rápidamente nuevas formas de producción automatizada, tales niveles de automatización requieren la creación de grandes sistemas de control industrial y sistemas de logística empresarial. Una característica clave de todos estos nuevos sistemas será la capacidad de incorporarse rápidamente a los cambios, resultando en sistemas de fabricación ágiles [1]. Actualmente existe un creciente interés en las nuevas tecnologías y arquitecturas para crear la próxima generación de sistemas distribuidos para la automatización industrial [2].

La IEC ha desarrollado una norma específica IEC 61499 [3] que define cómo se pueden utilizar los FBs en procesos industriales distribuidos, sistemas de medida y control, éste trabajo puede ayudar a resolver parte del problema de integración semántica. En sistemas industriales, los FBs son un concepto establecido para definir componentes de software robustos y reutilizables. Un bloque de función puede proporcionar una solución de software a un pequeño problema, como el control de una válvula, una unidad principal de la planta o una línea de producción completa. Los FBs permiten encapsular los algoritmos industriales en tal forma que las personas que no son especialistas en software pueden entender y aplicarlos fácilmente [4].

El estándar IEC 61449 brinda las siguientes características:[3]

- Portabilidad: soportar e interpretar correctamente configuraciones y componentes software creadas por diferentes herramientas software.
- Interoperabilidad: los distintos dispositivos integrados pueden funcionar conjuntamente para llevar a cabo las funciones propias de las aplicaciones.
- Configurabilidad: cualquier dispositivo y sus componentes software pueden ser configurados por herramientas de software de múltiples proveedores.
- Reconfigurabilidad: supone la habilidad para adaptar el hardware y el software de control durante la operación del proceso.
- Distribución: permite distribuir componentes software en diferentes dispositivos hardware independientemente del proveedor, el cual, es un requisito necesario dado por la industria de la automatización.

Con la adopción exitosa de OPC en miles de productos, OPC es utilizada como interfaz estandarizada entre sistemas de automatización en diferentes niveles de la pirámide de automatización, además existe muchas áreas en las que los fabricantes quieren usar un estándar como OPC pero no pueden usarlo debido a la dependencia COM de OPC o debido a las limitaciones para el acceso remoto usando DCOM. El primer estándar OPC clásico, fue diseñado como interfaz para controladores de comunicación, lo que permite un acceso de lectura y escritura estandarizado a los datos actuales en dispositivos de automatización. El caso de uso principal fueron los sistemas HMI y SCADA que acceden a datos de diferentes tipos de hardware de automatización y dispositivos de varios proveedores [5].

La Arquitectura Unificada OPC nació del deseo de crear un verdadero reemplazo para todas las especificaciones basadas en COM existentes sin perder ninguna característica o rendimiento del OPC clásico. Además, debe cubrir todos los requisitos para interfaces de sistema independientes de la plataforma. La amplia gama de aplicaciones donde se utiliza OPC requiere escalabilidad desde sistemas integrados a través de SCADA y DCS hasta sistemas MES y ERP [5]. Los requisitos más importantes para OPC UA se enumeran en la Tabla ??.

Tabla 1: Beneficios del protocolo OPC UA.

Comunicación entre sistemas distribuidos	Modelado de datos
<ul style="list-style-type: none"> <li>■ Robustez y tolerancia a fallos</li> <li>■ Independencia de plataforma</li> <li>■ Escalabilidad</li> <li>■ Alto rendimiento</li> <li>■ Internet y firewalls</li> <li>■ Control de acceso y seguridad</li> <li>■ Interoperabilidad</li> </ul>	<ul style="list-style-type: none"> <li>■ Modelo común para todos los datos OPC</li> <li>■ Orientado a objetos</li> <li>■ Sistema de tipo extensible</li> <li>■ Meta información</li> <li>■ Datos y métodos complejos</li> <li>■ Escalabilidad de modelos simples a complejos</li> <li>■ Modelo base abstracto</li> <li>■ Base para otros modelos de datos estándar</li> </ul>

Controladores avanzados son cada vez más implementados a nivel industrial como es el caso del MPC que se ha desarrollado considerablemente en los últimos años, tanto dentro de la comunidad de investigación como en la industria. La razón de este éxito se puede atribuir al hecho de que el MPC es, quizás, la forma más general de plantear el problema del control de procesos en el dominio del tiempo. La formulación del MPC integra un control óptimo, control estocástico, control de procesos con tiempo muerto y control multivariable. Otra ventaja del MPC es que debido al horizonte de control finito utilizado, las restricciones y los procesos no lineales que se encuentran con frecuencia en la indus-

tria. Algunos resultados nuevos y muy prometedores en este contexto permiten pensar que esta técnica de control experimentará una mayor expansión dentro de esta comunidad en un futuro próximo. Por otro lado, aunque se han reportado varias aplicaciones tanto en la industria como en las instituciones de investigación, el MPC aún no ha alcanzado en la industria la popularidad que su potencial sugeriría [6].

### **1.3. Objetivos.**

#### **1.3.1. Objetivo general.**

- Modelar matemáticamente para realizar un control robusto de la planta Festo MPS-PA Compact Workstation mediante la normativa IEC-61499.

#### **1.3.2. Objetivos específicos.**

- Realizar el modelo matemático del modulo FESTO MPS-PA Compact Workstation para la variable de nivel.
- Diseñar bloques de funciones para la implementación del controlador robusto.
- Implementar el control robusto en el modulo FESTO MPS-PA Compact Workstation mediante la normativa IEC 61499.

## **2. CAPÍTULO II**

### **ANTECEDENTES INVESTIGATIVOS**

#### **2.1. Antecedentes investigativos.**

Actualmente el rápido desarrollo de la tecnología, como Big Data, MES, sensores inteligentes y sistemas de control, han sido razones para que la industria 4.0 tenga un profundo impacto en las empresas industriales. La automatización de bajo costo promueve arquitecturas rentables y nuevos enfoques de desarrollo para aumentar la flexibilidad y la eficiencia de las operaciones de producción en una planta industrial. Esto ha llevado a la adopción de estándares como la IEC 61499 y protocolos de comunicación OPC UA para colaborar a las empresas industriales a integrarse en la visión de la industria 4.0, por tal razón ha sido motivo de diversas investigaciones en muchos campos: En [7], se discuten tres temas principales a través del análisis de investigaciones recientes: a) cómo los sistemas existentes programados en IEC 61131-3 se pueden hacer la transición a sistemas basados en IEC 61499; b) cómo se ha integrado IEC 61499 con tecnologías habilitadoras para la automatización inteligente distribuida; y c) cómo los entornos de ingeniería para IEC 61499 han sido implementados.

Existen muchas aplicaciones de control a nivel académico basados en la normativa IEC 61499 como en: [8] donde se ha logrado la integración del runtime IEC 61499 FORTE con un hardware integrado de bajo costo, donde se proporcionan un conjunto de SIFBs capaces de encapsular el acceso a las señales de entrada salida y algoritmos de control. La aplicación fue implementada en una estación manipuladora de piezas y una cinta transportadora, de igual manera en [9] se presentó un enfoque para la implementación de un Sistema Ciberfísico (CPS) industrial basado en la integración del estándar internacional IEC 61499 para la automatización industrial distribuida. Se desarrolla la automatización de una planta industrial a escala de laboratorio para el tratamiento de fin de vida de placas de circuitos electrónicos y se demostró cómo es posible modelar sistemas complejos, creando estructuras agregadas de CPS, y controlarlas gracias a la inteligencia distribuida. Otro campo de aplicación se da en [10], donde se presenta la implementación de una aplicación DGI: Equilibrio de Carga, utilizando la arquitectura IEC 61499, la aplicación fue realizada en dispositivos de automatización programables comerciales

y en sistemas embebidos, de igual manera en [11] se presenta modelos y conceptos para desarrollar sistemas de control de redes eléctricas inteligentes basados en conceptos holónicos y los estándares abiertos IEC 61850 y IEC 61499, se presentó una aplicación de protección contra fallas simple que ilustra cómo IEC 61499 pueden usarse para modelar e implementar aplicaciones que cumplen con IEC 61850.

Existen también aplicaciones en el campo de la robótica donde [12] propone un framework y el método de implementación de CPPS basado en FB IEC 61499 para la interacción física humano-robot (pHRI) y finalmente en [13] se tiene un enfoque de control holónico distribuido presentado para el sistema de fabricación de neumáticos que se desarrolló sobre la base del concepto multi-agente que se conecta a un entorno de fabricación virtual o real mediante bloques de función desarrollados de acuerdo con la norma IEC 61499. La aplicación de control se aplicó en un entorno de fabricación virtual y real para la verificación de algoritmos de control.

La industria 4.0 presenta la aplicación de conceptos modernos en tecnología de la información y la comunicación en contextos industriales para crear productos más flexibles e innovadores. La cuarta revolución industrial sólo se puede lograr si se invierte el esfuerzo suficiente para introducir la interoperabilidad entre aplicaciones industriales. Por lo tanto el protocolo de comunicación OPC UA juega un papel muy importante dentro de la industria 4.0, por tal motivo, en [14] se propone una solución hacia el interfuncionamiento OPC UA y oneM2M. De igual manera en [15] se implementó una arquitectura CPPS a través de un servidor OPC UA central para la fabricación inteligente (Smart Manufacturing) basada en una red 5G, del mismo modo en [16] se presenta el diseño de un Sistema de Adquisición de Datos (DAS) dentro de un framework estructural presentado por la OPC UA, la implementación fue implementada en una célula FESTO.

Como ya se ha mencionado anteriormente el estándar IEC-61499 y el protocolo de comunicación OPC UA jugarán un papel muy importante en la industria 4.0, por lo que su unificación será muy importante. Investigaciones como la de [17] han presentado un enfoque para acceder a datos de campo en sistemas de automatización utilizando servidores OPC UA en arquitectura CPPS mediante aplicaciones IEC 61499 para procesos industriales de extracción de petróleo. Del mismo modo en [18] se presenta el diseño de un sistema de control distribuido empotrado realizado en una placa Raspberry PI, la cual entrega datos a un servidor OPC-UA todo dentro de la norma IEC-61499, la aplicación fue implementada en una maqueta FESTO, de igual manera en [19] se realiza la implementación de un sistema de clasificación de paquetes unificando estándar IEC-61499 y el protocolo de comunicación OPC UA.

El control MPC ha sido tema de investigación y desarrollo durante más de tres décadas, se han realizado innumerables aplicaciones tanto en la industria como a nivel educativo, por ejemplo en [20] se trató sobre la cadena de acciones que involucran la enseñanza, la práctica y la aplicación en laboratorio del MPC, donde se simularon algoritmos de control ideales agregando desajuste de modelos, perturbaciones y ruido de medición para un sistema de primer orden para controlar el nivel del agua, de igual manera en [21] se presenta y discute los últimos desarrollos del MPC para convertidores y variadores de potencia, además se describe el estado actual de esta estrategia de control y analiza las nuevas tendencias y desafíos que presenta cuando se aplica a los sistemas electrónicos de potencia. Otro campo de aplicación se da en [22] donde se proporciona una descripción general completa de los resultados de estabilidad de un EMPC: con y sin restricciones terminales, con restricciones promediadas, formulaciones con múltiples objetivos y restricciones terminales generalizadas, así como enfoques basados en Lyapunov y finalmente en [23] se ofrece una descripción general de los principales desarrollos en el área del control predictivo de modelos estocásticos (SMPC), proporcionando al lector diferentes algoritmos SMPC y los desafíos teóricos clave en el control predictivo estocástico sin una complejidad matemática.

En [24] se propone un prototipo de plataforma de control basada en MPC, la plataforma está estructurada según una arquitectura de control de dos niveles. En la capa inferior, los algoritmos MPC distribuidos controlan los equipos del sistema de producción. En la capa superior, un coordinador MPC gestiona los controladores de nivel inferior, aprovechando al máximo los avances más recientes en teoría de control híbrido, programación dinámica, optimización de enteros mixtos y teoría de juegos. La plataforma de control basada en MPC se presentará y se aplicó al caso de una planta de producción piloto. Una aproximación híbrida se da en [25] donde se presentó la reutilización de hardware de bajo costo basado en el desarrollo bajo el estándar IEC-61499 de un control que encapsula un algoritmo MPC y funciones de comunicación permitiendo que una tarjeta de desarrollo Raspberry Pi tome el control total de la operación del controlador lógico programable (PLC) involucrado en el control de una bomba de una estación de bombeo de oleoductos, lo que hace posible combinar las fortalezas físicas de un PLC con las capacidades de procesamiento de un controlador embebido.

## 2.2. Fundamentación teórica.

### 2.2.1. Modelado matemático de sistemas físicos.

Uno de los aspectos más importantes de la ingeniería es poder representar un fenómeno físico en forma matemática, ya que así es posible llevar a cabo un análisis cuantitativo del sistema y determinar sus características, su comportamiento y sus limitaciones; además, en dado caso, también será posible buscar alternativas para mejorar el funcionamiento del sistema. Para dar este paso, primero es necesario identificar la variable o variables que ocasionan el cambio en el sistema y después establecer una hipótesis empírica o basada en alguna ley física que permita representar al sistema en forma matemática[26].

- Sistema hidráulico.

Sea la ecuación de gasto  $q$  que relaciona la velocidad  $v$  con la que un fluido pasa a través de una tubería de cierta área  $a$ :

$$q = av = aC_q v_{teórica} \quad (1)$$

donde

$q$  = Gasto a través de la válvula ( $m^3/seg$ ).

$a$  = Área de la sección transversal por donde pasa el fluido ( $m^2$ ).

$v$  = Velocidad real con la que un fluido pasa a través de un orificio ( $m/seg$ ).

$C_q$  = Coeficiente de descarga (0,6 a 0,8),  $v = C_q v_{teórica}$  = velocidad real.

El producto de  $v_{teórica}$  por el coeficiente de descarga  $C_q$  proporciona la velocidad  $v$  real del fluido.

El principio de Torricelli[26] establece que la velocidad  $v$  de un fluido a través de un orificio de área  $a$  de un tanque lleno con un líquido hasta una altura  $h$  es igual a:

$$v = \sqrt{2hg} \quad (2)$$

Sea un tanque de área  $A$ , con un orificio de sección transversal  $a$  en el fondo del recipiente, según se indica en la Figura 1, se pretende obtener una expresión para la variación del nivel  $h$  con respecto al tiempo.

En relación con la ecuación (1), la cual indica el gasto que tiene que ver con la velocidad con la cual un fluido pasa a través de un orificio de área  $a$ :

$$q = av = \frac{V}{t}, (m^3/seg) \quad (3)$$

donde:

$t =$  Tiempo (*seg*).

$q =$  Gasto a través de la válvula ( $m^3/seg$ ).

$a =$  Área de la sección transversal por donde pasa el fluido ( $m^2$ ).

$v =$  Velocidad real con la que un fluido pasa a través de un orificio ( $m/seg$ ).

$V =$  Volumen del recipiente ( $m^3$ ).

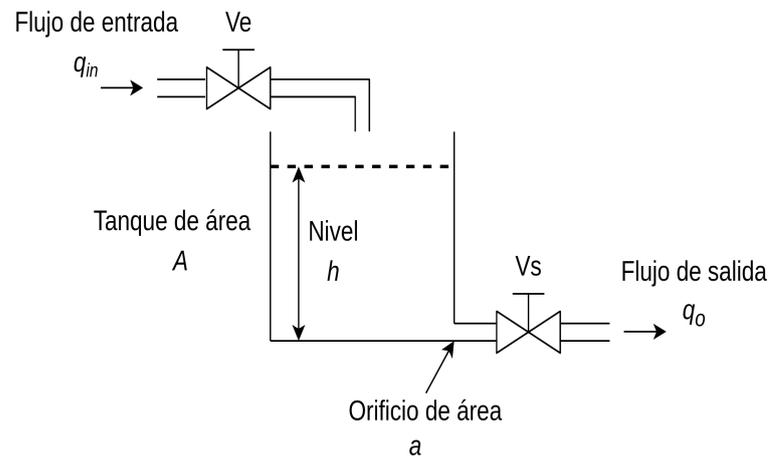


Figura 1: Sistema de variación de nivel  $h$  en un tanque

Para establecer la ecuación que define al sistema[26], se considera que la cantidad acumulada  $dV/dt$  es igual a la diferencia entre el flujo de entrada  $q_{in}$  y el flujo de salida  $q_o$ :

$$\frac{dV}{dt} = (\text{flujo de entrada}) - (\text{flujo de salida})$$

el área permanece constante, por lo cual:

$$A \frac{dh}{dt} = q_{in} - q_o$$

para el flujo de salida, se reescribe la ecuación (1) como:

$$A \frac{dh}{dt} = q_{in} - a C_q v_{teórico} \quad (4)$$

donde:

$h =$  Nivel del fluido almacenado ( $m$ ).

$A =$  Sección transversal del tanque ( $m^2$ ).

$a =$  Área del orificio de salida del fluido ( $m^2$ ).

$C_q v_{teórico} =$  Velocidad real  $= v$  ( $m/seg$ )

y tomando en cuenta la ley de Torricelli (2):

$$A \frac{dh}{dt} = q_{in} - aC_q \sqrt{2gh} \quad (5)$$

Para obtener la función de transferencia de un sistema, el primer requisito a satisfacer es que la ecuación sea lineal. La ecuación (5) es una ecuación diferencial no lineal, por lo que en principio no es posible llevar a cabo la transformación. Para obtener su correspondiente función de transferencia, habrá que linealizar dicha expresión.

■ Linealización de sistemas.

La serie de Taylor establece que cualquier función analítica puede representarse como una serie infinita de la forma[26]:

$$f(x) = f(c) + f'(c)(x - c) + f''(c) \frac{(x - c)^2}{2!} + \dots + f^n(c) \frac{(x - c)^n}{n!}$$

$$f(x) = \sum_{n=0}^{\infty} \frac{d^n f}{dx^n} \frac{(x - c)^n}{n!} \quad (6)$$

donde  $c$  es una constante alrededor de la cual se lleva a cabo la aproximación,  $(x - c)$  representa un intervalo, y la exactitud de la aproximación estará en función de qué tan pequeño sea el intervalo elegido. El objetivo es linealizar una ecuación no lineal, por ejemplo, la ecuación (5), alrededor de un punto específico denominado punto de operación.

Sea un sistema no lineal, como el mostrado en la Figura 2, donde  $y(t)$  es la respuesta de un sistema  $g(t)$  a una entrada  $x(t)$ :

$$y(x) = g(x(t))$$

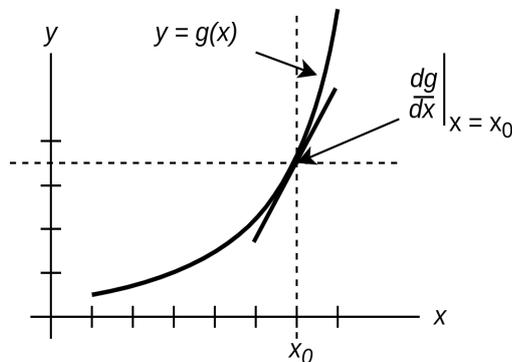


Figura 2: Representación de un sistema no lineal

Se designa un punto de operación  $x_0$  con un intervalo pequeño  $(x - x_0)$ , en un segmento casi lineal de la curva no lineal, pero como esa curva es continua puede representarse como serie de Taylor mediante la ecuación (6):

$$y = g(x) = g(x_0) + \frac{dg}{dx}(x - x_0) + \frac{d^2g}{dx^2} \frac{(x - x_0)^2}{2!} + \dots \quad (7)$$

donde la pendiente de la curva en el punto de operación es  $x_0$  ver en la Figura 2:

$$\left. \frac{dg}{dx} \right|_{x=x_0}$$

Una aproximación razonable de la ecuación (7) es:

$$y \approx g(x_0) + \frac{dg}{dx}(x - x_0) \quad (8)$$

siempre y cuando el intervalo  $(x - x_0)$  sea lo suficientemente pequeño.

La justificación analítica de la aproximación de la serie de Taylor, que lleva a la ecuación (8) se da a continuación[26].

Al hacer  $g(x_0) = y_0$  y sustituir en la ecuación (8) se tiene:

$$y \approx y_0 + m(x - x_0) \quad (9)$$

donde  $m$  es la pendiente de la tangente en el punto de operación  $x_0$ :

$$\therefore m = \frac{y - y_0}{x - x_0} = \frac{\Delta y}{\Delta x}$$

que es precisamente la definición analítica de la diferenciación.

### 2.2.2. Características de respuesta de los sistemas.

Los sistemas de primer orden son aquellos que quedan definidos por medio de ecuaciones diferenciales de primer orden; por ejemplo, un sistema térmico, hidráulico, eléctrico, etcétera[26]:

$$C \frac{dT}{dt} + \frac{1}{R}T = q_{in}, \quad A \frac{dh}{dt} + \frac{1}{R}h = q_{in}, \quad L \frac{di}{dt} + Ri = v(t).$$

Un sistema de primer orden es aquel que queda definido por una ecuación diferencial de primer orden:

$$a \frac{dy}{dt} + by = cr(t) \quad (10)$$

Al considerar constantes a los coeficientes  $a$ ,  $b$  y  $c$ , se determinará la función de transferencia  $G(s)$  del sistema que transforma al dominio  $s$  la ecuación (10), suponiendo cero

a la condición inicial  $y(0)$ . Para obtener  $G(s)$  en términos de los polos del sistema, se hace unitario el coeficiente de la mayor derivada contenida en la ecuación diferencial[26]:

$$sY(s) + \left(\frac{b}{a}\right)Y(s) = \left(\frac{c}{a}\right)R(s)$$

y

$$Y(s) = R(s) \frac{\left(\frac{c}{a}\right)}{s + \left(\frac{b}{a}\right)}$$

$$G(s) = \frac{Y(s)}{R(s)} = \frac{\left(\frac{c}{a}\right)}{s + \left(\frac{b}{a}\right)} = \frac{\left(\frac{c}{a}\right)}{s + a_0} \quad (11)$$

donde el polo de la función de transferencia  $G(s)$  es  $p = -a_0 = -(b/a)$ .

La ecuación (11) se reescribe como:

$$G(s) = \frac{\frac{c}{a} \frac{a}{b}}{s + \frac{b}{a} \frac{a}{b}} = \frac{\frac{c}{b}}{\left(\frac{a}{b}\right)s + 1} = \frac{K}{\tau s + 1} \quad (12)$$

donde

$K$  = Ganancia del sistema (factor de amplificación entre salida y entrada).

$\tau$  = Constante de tiempo del sistema (*seg*).

La ecuación (12) queda expresada en términos de la constante de tiempo  $\tau$  del sistema (al conocer  $\tau$  será posible determinar la velocidad de respuesta del sistema). Una vez establecida la función de transferencia  $G(s)$  del sistema, se procederá a determinar su respuesta o salida  $y(t)$  cuando se aplica una entrada escalón  $r(t) = AU(t)$ [26].

La respuesta del sistema se obtiene al reordenar la ecuación (11):

$$Y(s) = R(s)G(s) = \frac{A\left(\frac{c}{a}\right)}{s\left(s + \frac{b}{a}\right)}$$

al aplicar descomposición en fracciones parciales:

$$Y(s) = \frac{A\left(\frac{c}{a}\right)}{s\left(s + \frac{b}{a}\right)} = \frac{C_1}{s} + \frac{C_2}{s + \frac{b}{a}} = \frac{A\left(\frac{c}{b}\right)}{s} - \frac{A\left(\frac{c}{b}\right)}{s + \frac{b}{a}}$$

$$\mathcal{L}^{-1}\{Y(s)\} = y(t) = A\left(\frac{c}{b}\right)[1 - e^{-\left(\frac{b}{a}\right)t}] \quad (13)$$

o su equivalente en términos de la ganancia del sistema  $K = c/b$  y de su constante de tiempo  $\tau = a/b$ :

$$y(t) = AK[1 - e^{-\frac{t}{\tau}}] \quad (14)$$

Si el sistema es estable, su valor final será:

$$y(\infty) = \lim_{t \rightarrow \infty} y(t) = AK$$

De las ecuaciones (13) o (14), es posible determinar directamente la respuesta al escalón de un sistema de primer orden si se conoce  $A$ ,  $K$  y  $\tau$ .

Para determinar las características generales de la respuesta al escalón de un sistema de primer orden, se normalizará la ecuación (14) haciendo  $AK = 1$ ; además, se considerará que  $a_0 = 1/\tau$ , con lo cual se obtiene[26]:

$$y(t) = [1 - e^{-a_0 t}] \quad (15)$$

donde la respuesta  $y(t)$  del sistema consta de dos términos: un factor exponencial decreciente  $e^{-t}$  (el cual queda definido por las características particulares del sistema  $G(s)$ , denominado componente transitorio) y un valor constante igual a la unidad (que corresponde a la forma de onda de la entrada  $r(t)$  y recibe el nombre de régimen de estado estable). La gráfica de  $y(t)$  se observa en la Figura 3[26].

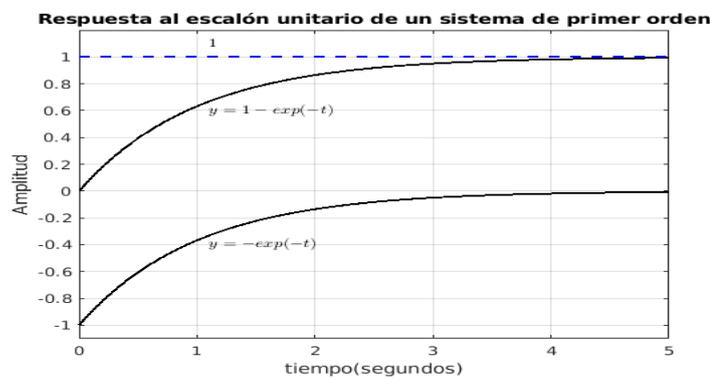


Figura 3: Elementos de la respuesta al escalón de un sistema de primer orden.

$$y(t) = y_{\text{régimen transitorio}}(t) + y_{\text{régimen de estado estable}}(t) \quad (16)$$

La Figura 4 indica los regímenes transitorio (o respuesta natural) y de estado estable (o respuesta forzada) de  $y(t) = 1 - e^{-t}$ [26].

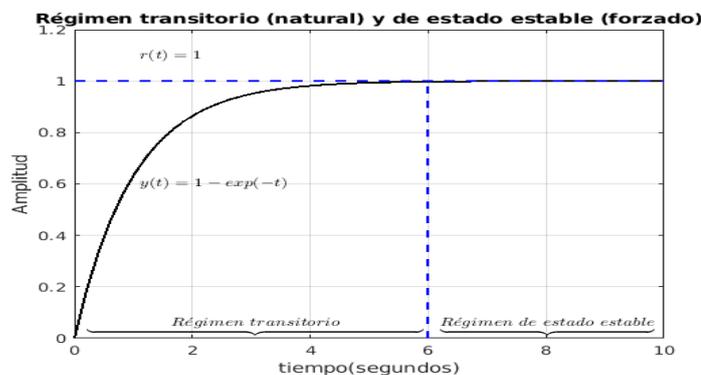


Figura 4: Componentes de la respuesta de un sistema al escalón.

- Constante de tiempo  $\tau$ .

Se considera nuevamente la ecuación (15), la cual será cuantificada para diversos valores de  $a_0$ . Para alcanzar el régimen de estado estable, teóricamente se requiere que  $t \rightarrow \infty$ ; sin embargo, resulta importante determinar un valor práctico de tiempo en el que el sistema alcance su valor final, por lo que se evaluará la ecuación (15) para distintos valores de tiempo:  $t = 1/a_0$ ,  $t = 2/a_0$ ,  $t = 3/a_0$  y  $t = 4/a_0$ [26]. Lo anterior se indica en la gráfica de la Figura 5:

$$y\left(t = \frac{1}{a_0}\right) = [1 - e^{-1}] = 63,212 \%y(\infty)$$

$$y\left(t = \frac{2}{a_0}\right) = [1 - e^{-2}] = 86,646 \%y(\infty)$$

$$y\left(t = \frac{3}{a_0}\right) = [1 - e^{-3}] = 95,502 \%y(\infty)$$

$$y\left(t = \frac{4}{a_0}\right) = [1 - e^{-4}] = 98,168 \%y(\infty)$$

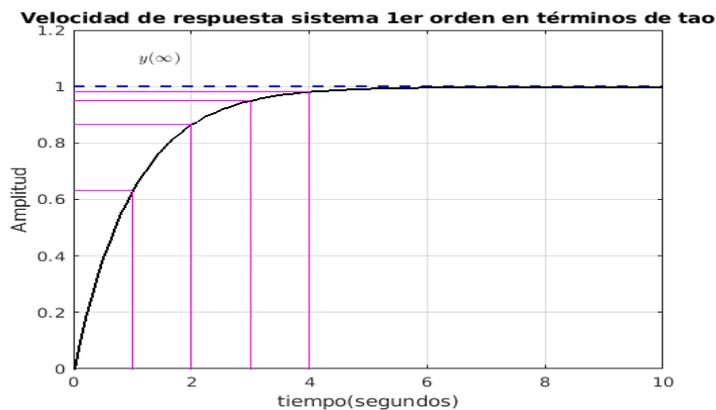


Figura 5: Porcentaje de respuesta alcanzado para diversos valores de tiempo:  $t = 1/a_0$ ,  $t = 2/a_0$ ,  $t = 3/a_0$  y  $t = 4/a_0$ .

Con respecto a la ecuación (12), se define al número  $\tau$  y se le denomina constante de tiempo del sistema (*seg*):

$$\tau = \frac{1}{\left(\frac{b}{a}\right)} = \frac{a}{b} \quad (17)$$

Para fines prácticos, se dice que el tiempo de asentamiento  $t_a$  es el tiempo que requiere el sistema para alcanzar su valor final, y corresponde a cuatro constantes de tiempo[26]:

$$t_a = 4\tau \quad (18)$$

La Figura 6 indica la respuesta al escalón de tres sistemas de primer orden para diferentes constantes de tiempo:

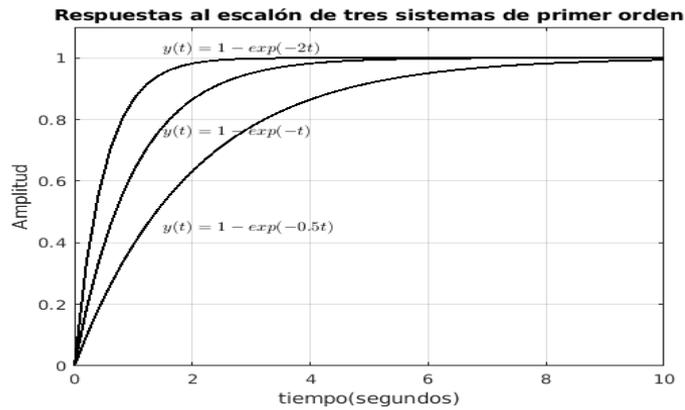


Figura 6: Respuesta al escalón de tres diferentes sistemas de primer orden.

$$G_1(s) = \frac{1}{s + 0,5} \text{ donde } \tau = \frac{1}{0,5} = 2\text{seg} \text{ y } t_a = 4\tau = 8\text{seg}$$

$$G_2(s) = \frac{1}{s + 1} \text{ donde } \tau = 1\text{seg} \text{ y } t_a = 4\tau = 4\text{seg}$$

$$G_3(s) = \frac{1}{s + 2} \text{ donde } \tau = \frac{1}{2} = 0,5\text{seg} \text{ y } t_a = 4\tau = 2\text{seg}$$

- Aproximación teórica de una curva de respuesta real.

Si se conoce la respuesta real de un sistema al escalón, es posible obtener la representación analítica del sistema en forma de función de transferencia  $G(s)$ . A continuación se definirán dos métodos para lograr tal propósito[26].

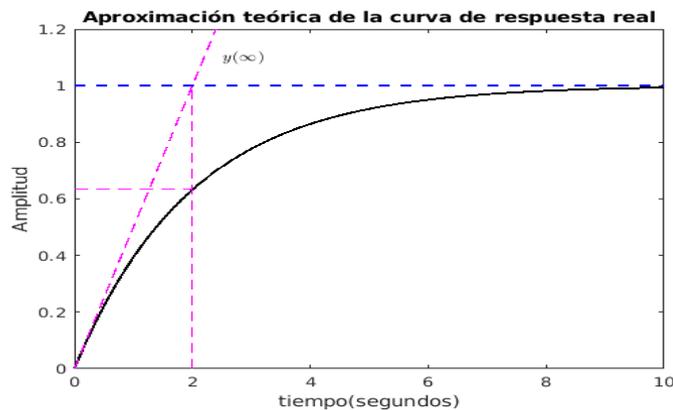


Figura 7: Obtención de  $t$  a partir de la respuesta al escalón de un sistema de primer orden.

1. Método de la constante de tiempo  $\tau$ .

Si se considera el hecho de que cuando transcurre una constante de tiempo el sistema ha alcanzado el 63,212 % de su valor final (como se indica en la Figura 7), se

procede a trazar una recta paralela a la abscisa  $t$  que corresponda al 63,212 % del valor final  $y(\infty)$  hasta que corte la curva de respuesta; en ese punto se proyecta una recta paralela a la ordenada hasta que corte el eje de tiempo  $t$ , que es el punto del valor de la constante de tiempo  $\tau$ [26].

## 2. Método de la pendiente máxima.

En este caso se traza una recta con pendiente máxima, desde el origen sobre la curva de respuesta hasta cortar la recta de valor final  $y(\infty)$ , y en el punto de cruce se traza una recta perpendicular a la abscisa  $t$ , hasta cortar el eje del tiempo, cuyo punto es el valor de la constante de tiempo  $\tau$ [26].

### 2.2.3. Método de mínimos cuadrados para regresión.

Supongamos que se tienen los siguientes datos para las variables  $x$  y  $y$ .

$x_1$	$x_2$	$\dots$	$x_n$
$y_1$	$y_2$	$\dots$	$y_n$

Esta situación se puede presentar en estudios experimentales, donde se estudia la variación de cierta magnitud  $x$  (por ejemplo, temperatura, nivel, presión, etc.) en función de otra magnitud  $y$ . De manera teórica, se espera que la relación entre estas variables sea lineal pero también es aplicable para otro tipos de ecuaciones no lineales, es decir, que haya una expresión del tipo  $y = mx + b$  que las relacione. El problema es determinar los valores de  $m$  y  $b$  que determinaron esta funcionalidad lineal entre  $x$  y  $y$ .

Si las mediciones hechas para  $x$  y  $y$  fueron exactas, todos los datos  $(x_i, y_i), i = 1, 2, \dots, n$  obtenidos representarían puntos del plano colocados (todos ellos) sobre una misma línea recta  $y$ , de hecho, bastarían dos de ellos para determinar la pendiente  $m$  y la ordenada al origen  $b$ . Sin embargo, en la práctica, resulta muy poco probable que esto ocurra. Ciertamente, al colocar todos los puntos  $(x_i, y_i), i = 1, 2, \dots, n$  en el plano veríamos que están distribuidos “en los alrededores de alguna recta”, la cual se podría trazar sencillamente con una regla, “al tanteo”, tratando de que todos quedaran lo “más cercanos posible” a la recta[27].

El método de mínimos cuadrados nos proporciona un criterio con el cual podremos obtener *la mejor* curva que representa a los puntos dados. Digamos que  $y = mx + b$  es esa recta. Como dijimos, sería deseable que se tuviera  $y_i = mx_i + b$  para todos los puntos  $(x_i, y_i)$  de  $i = 1, 2, \dots, n$ . Sin embargo, como, en general,  $y_i \neq mx_i + b$ , se pide que la

suma de los cuadrados de las diferencias (las desviaciones)  $y_i - (mx_i + b)$  sea la menor posible[27]. Es decir, se pide que

$$S = (y_1 - (mx_1 + b))^2 + (y_2 - (mx_2 + b))^2 + \dots + (y_n - (mx_n + b))^2 = \sum_{i=1}^n (y_i - (mx_i + b))^2$$

sea lo más pequeña posible. Los valores de  $m$  y  $b$  que cumplan con esta propiedad, determinan la recta  $y = mx + b$  que mejor representa el comportamiento lineal de los puntos  $(x_i, y_i)$ .

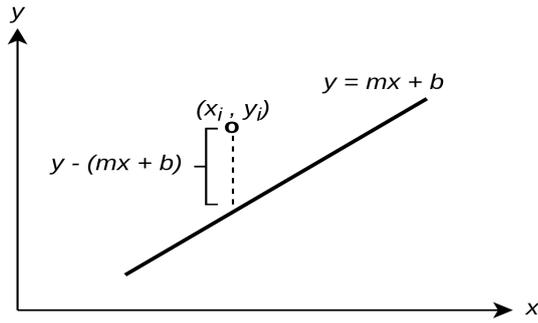


Figura 8: Recta ajustada.

Consideremos entonces la función  $f$  de las variables  $m$  y  $b$  dada por

$$f(m, b) = \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (19)$$

$$f(m, b) = \sum_{i=1}^n (y_i - (mx_i + b))^2$$

los puntos críticos de esta función se obtienen al resolver el sistema

$$\frac{\partial f}{\partial m} = \sum_{i=1}^n 2(y_i - mx_i - b)(-x_i) = -2 \sum_{i=1}^n x_i(y_i - mx_i - b) = 0$$

$$\frac{\partial f}{\partial b} = \sum_{i=1}^n 2(y_i - mx_i - b)(-1) = -2 \sum_{i=1}^n (y_i - mx_i - b) = 0$$

De la segunda ecuación obtenemos[27]

$$\sum_{i=1}^n y_i - m \sum_{i=1}^n x_i - \sum_{i=1}^n b = 0$$

de donde

$$b = \frac{1}{n} \sum_{i=1}^n y_i - m \left( \frac{1}{n} \sum_{i=1}^n x_i \right)$$

Llamemos  $\bar{x} = \sum_{i=1}^n x_i$ ,  $\bar{y} = \sum_{i=1}^n y_i$ . Estas son las medias aritméticas de los valores  $x_i$  y  $y_i$ , respectivamente. Entonces

$$b = \bar{y} - m\bar{x}$$

Sustituyendo en la ecuación  $\frac{\partial f}{\partial m} = 0$ , nos queda

$$m = \frac{\sum_{i=1}^n x_i(y_i - \bar{y})}{\sum_{i=1}^n x_i(x_i - \bar{x})}$$

En resumen, la función  $f(m, b) = \sum_{i=1}^n (y_i - (mx_i + b))^2$  tiene un (único) punto crítico para[27]

$$m = \frac{\sum_{i=1}^n x_i(y_i - \bar{y})}{\sum_{i=1}^n x_i(x_i - \bar{x})}, \quad b = \bar{y} - m\bar{x}$$

#### 2.2.4. Análisis de sistemas discretos.

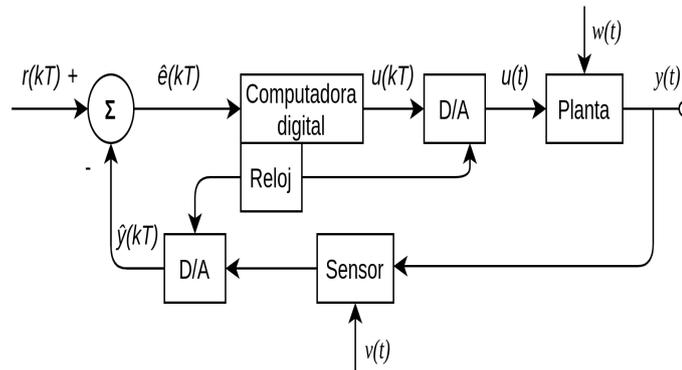


Figura 9: Diagrama de bloque de un sistema de control discreto básico.

- Ecuaciones de diferencia lineal.

Suponemos que el convertidor analógico-digital (A/D) de la Figura 9 toma muestras de la señal en momentos discretos y las envía a la computadora de modo que  $\hat{y}(kT) = y(kT)$ . El trabajo de la computadora es tomar estos valores de muestra y calcular de alguna manera las señales que se enviarán a través del convertidor digital-analógico (D/A). Supongamos que llamamos a las señales de entrada hasta la  $k$ -ésima muestra  $e_0, e_1, e_2, \dots, e_k$ , y las señales de salida antes de ese tiempo  $u_0, u_1, u_2, \dots, u_{k-1}$ . Para obtener la siguiente salida, la máquina calcula alguna función que podemos expresar en forma simbólica como[28]:

$$u_k = f(e_0, \dots, e_k; u_0, \dots, u_{k-1}) \quad (20)$$

Debido a que planeamos enfatizar las posibilidades elementales y dinámicas, asumimos que la función  $f$  en la ecuación (20) es lineal y depende sólo de un número *finito* de  $e$ s y  $u$ s anteriores. Así escribimos

$$u_k = -a_1 u_{k-1} - a_2 u_{k-2} - \dots - a_n u_{k-n} + b_0 e_k + b_1 e_{k-1} + \dots + b_m e_{k-m} \quad (21)$$

La ecuación (21) se denomina ecuación de recurrencia lineal o ecuación en diferencias y tiene muchas similitudes con una ecuación diferencial lineal. El nombre “ecuación en diferencias” se deriva del hecho de que podríamos escribir la ecuación (21) usando  $u_k$  más las diferencias en  $u_k$ , que se definen como[28]:

$$\begin{aligned} \nabla u_k &= u_k - u_{k-1} \\ \nabla^2 u_k &= \nabla u_k - \nabla u_{k-1} \\ \nabla^n u_k &= \nabla^{n-1} u_k - \nabla^{n-1} u_{k-1} \end{aligned} \quad (22)$$

si resolvemos la ecuación (22) para los valores de  $u_k$ ,  $u_{k-1}$  y  $u_{k-2}$  en términos de diferencias, encontramos

$$\begin{aligned} u_k &= u_{k'} \\ u_{k-1} &= u_k - \nabla u_k \\ u_{k-2} &= u_k - 2\nabla u_k + \nabla^2 u_k \end{aligned}$$

Por lo tanto, para una ecuación de segundo orden con coeficientes  $a_1$ ,  $a_2$  y  $b_0$  (dejamos  $b_1 = b_2 = 0$  para simplificar), encontramos que la ecuación en diferencia equivalente

$$a_2 \nabla^2 u_k - (a_1 + 2a_2) \nabla u_k + (a_2 + a_1 + 1) u_k = b_0 e_k$$

Aunque las dos formas son equivalentes, la forma recurrente de (21) es más conveniente para la implementación por computadora; Sin embargo, continuaremos refiriéndonos a nuestras ecuaciones como “ecuaciones en diferencias”. Si las  $a$ s y las  $b$ s en la ecuación. (21) son constantes, entonces la computadora está resolviendo una (CCDE).

Para resolver una CCDE necesitamos un tiempo inicial (valor  $k$ ) y algunas condiciones iniciales para caracterizar el contenido de la memoria de la computadora para ese momento. Por ejemplo, supongamos que tomamos el caso

$$u_k = u_{k-1} + u_{k-2} \quad (23)$$

e inicia en  $k = 2$ . Aquí no hay valores de entrada y para calcular  $u_2$  necesitamos conocer los valores (iniciales) de  $u_0$  y  $u_1$ . Supongamos que son  $u_0 = u_1 = 1$ . Los primeros

nueve valores son 1, 1, 2, 3, 5, 8, 13, 21, 34.... En la Figura 10 se muestra una gráfica de los valores de  $u_k$  versus  $k$  (Números Fibonnaci).

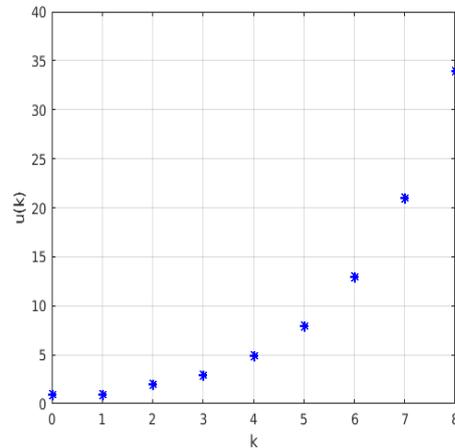


Figura 10: Números Fibonnaci.

Como ejemplo de los orígenes de una ecuación en diferencias con una entrada externa, consideramos la aproximación discreta a la integración. Suponga que tenemos una señal continua,  $e(t)$ , de la cual se bosqueja un segmento en la Figura 11, y deseamos calcular una aproximación a la integral

$$\mathcal{J} = \int_0^t e(t)dt, \quad (24)$$

utilizando solo los valores discretos  $e(0), \dots, e(t_{k-1}), e(t_k)$ . Suponemos que tenemos una aproximación para la integral desde cero hasta el tiempo  $t_{k-1}$  y la llamamos  $u_{k-1}$ . El problema es obtener  $u_k$  a partir de esta información. Tomando a la integral como el área bajo la curva  $e(t)$ , vemos que éste problema se reduce a encontrar una aproximación al área bajo la curva entre  $t_{k-1}$  y  $t_k$ . En la Figura 11 se representan tres alternativas. Utilizando el rectángulo de altura  $e_{k-1}$  o el rectángulo de altura  $e_k$ , o el trapecoide formado conectando  $e_{k-1}$  con  $e_k$  por una línea recta[28].

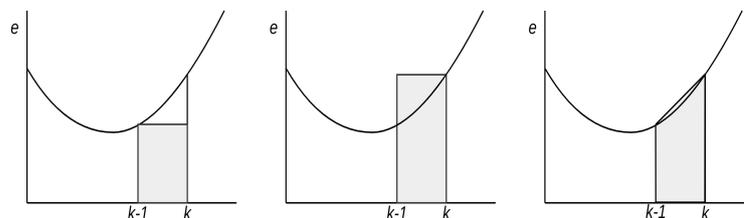


Figura 11: Gráfico de una función y aproximaciones alternativas al área bajo la curva en un solo intervalo de tiempo.

Si tomamos la tercera opción, el área del trapecoide es

$$A = \frac{t_k - t_{k-1}}{2}(e_k + e_{k-1}). \quad (25)$$

Finalmente, si asumimos que el período de muestreo,  $t_k - t_{k-1}$ , es una constante,  $T$ , nos lleva a una fórmula simple para la integración discreta llamada **regla del trapecoide**

$$u_k = u_{k-1} + \frac{T}{2}(e_k + e_{k-1}). \quad (26)$$

Si  $e(t) = 1$ , entonces  $e_k = kT$  y la sustitución de  $u_k(T^2/2)k^2$  satisface la ecuación (26) y es exactamente la integral de  $e$ . Si aproximamos el área bajo la curva por el rectángulo de altura  $e_{k-1}$ , el resultado se llama **regla rectangular hacia adelante**(forward rectangular rule) también llamado método de Euler y se describe por

$$u_k = u_{k-1} + Te_{k-1}$$

La otra posibilidad es la **regla rectangular hacia atrás** (backward rectangular rule), dada por

$$u_k = u_{k-1} + Te_k$$

Cada una de estas reglas de integración es un caso especial de nuestra ecuación en diferencias generales (21).

Así vemos que las ecuaciones en diferencias pueden ser evaluadas directamente por una computadora digital y que pueden representar modelos de procesos físicos y aproximaciones a la integración. Resulta que si las ecuaciones en diferencias son lineales con coeficientes que son constantes, podemos describir la relación entre  $u$  y  $e$  mediante una función de transferencia y, por lo tanto, obtener una gran ayuda para el análisis y también al diseño de controles lineales, constantes y discretos.

- Función de transferencia discreta.

La función de transferencia de sistemas lineales, constantes y discretos se obtiene mediante el método de análisis de transformada  $z$ [28].

Si una señal tiene valores discretos  $e_0, e_1, \dots, e_k, \dots$  definimos la transformación  $z$  de la señal como la función

$$\begin{aligned} E(z) &\hat{=} \mathcal{Z}\{e(k)\} \\ &\hat{=} \sum_{k=-\infty}^{\infty} e_k z^{-k}, \quad r_0 < |z| < R_0 \end{aligned} \quad (27)$$

y asumimos que podemos encontrar valores de  $r_0$  y  $R_0$  como límites en la magnitud de la variable compleja  $z$  para la cual converge la ecuación de serie (27). En (27) el límite inferior es  $-\infty$  de modo que se incluyen los valores de  $e_k$  en ambos lados de  $k = 0$ .

La transformada  $z$  tiene el mismo papel en sistemas discretos que la transformada de Laplace en el análisis de sistemas continuos[28]. Para encontrar la relación, procedemos por sustitución directa. Tomamos la definición dada por (27) y, de la misma manera, definimos la transformada  $z$  de la secuencia  $\{u_k\}$  como

$$U(z) = \sum_{k=-\infty}^{\infty} u_k z^{-k} \quad (28)$$

Ahora multiplicamos la ecuación (26) por  $z^{-k}$  y sumamos  $k$ , obtenemos

$$\sum_{k=-\infty}^{\infty} u_k z^{-k} = \sum_{k=-\infty}^{\infty} u_{k-1} z^{-k} + \frac{T}{2} \left( \sum_{k=-\infty}^{\infty} e_k z^{-k} + \sum_{k=-\infty}^{\infty} e_{k-1} z^{-k} \right) \quad (29)$$

De (28), reconocemos el lado izquierdo como  $U(z)$ . En el primer término de la derecha, dejamos  $k - 1 = j$  para obtener

$$\sum_{k=-\infty}^{\infty} u_{k-1} z^{-k} = \sum_{j=-\infty}^{\infty} u_j z^{-(j+1)} = z^{-1} U(z) \quad (30)$$

Mediante operaciones similares en el tercer y cuarto términos podemos reducir la ecuación. (29) a

$$U(z) = z^{-1} U(z) + \frac{T}{2} [E(z) + z^{-1} E(z)] \quad (31)$$

La ecuación (31) ahora simplifica una ecuación algebraica en  $z$  y las funciones  $U$  y  $E$ . Resolviéndola obtenemos

$$U(z) = \frac{T}{2} \frac{1 + z^{-1}}{1 - z^{-1}} E(z) \quad (32)$$

Definimos la relación de la transformada de la salida a la transformada de la entrada como la función de transferencia,  $H(z)$ . Por lo tanto, en este caso, la función de transferencia para la integración de la regla trapezoidal es[28]

$$\frac{U(z)}{E(z)} \triangleq H(z) = \frac{T}{2} \frac{z + 1}{z - 1} \quad (33)$$

Para la relación más general dada por la ecuación (21), se verifica fácilmente mediante las mismas técnicas que

$$H(z) = \frac{b_0 + b_1 z^{-1} + \dots + b_m z^{-m}}{1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_n z^{-n}}$$

y si  $n \geq m$ , podemos escribir esto como una razón de polinomios en  $z$  como

$$H(z) = \frac{b_0 z^n + b_1 z^{n-1} + \dots + b_m z^{n-m}}{z^n + a_1 z^{n-1} + a_2 z^{n-2} + \dots + a_n} \quad (34)$$

o

$$H(z) = \frac{b(z)}{a(z)}$$

- Modelos discretos de sistemas de datos muestreados.

La mayoría de los sistemas dinámicos que se van a controlar son sistemas continuos y, si son lineales, se describen mediante funciones de transferencia continua en la variable de Laplace  $s$ [28]. La interfaz entre los dominios continuo y discreto son los convertidores A/D y D/A como se muestra en la Figura 9, donde:

$r$  = Referencia o entradas de comando.

$u$  = Control o señal de entrada del actuador.

$y$  = Señal controlada o de salida.

$\hat{y}$  = Salida del instrumento o sensor, generalmente una aproximación o estimación de  $y$ .

$\hat{e} = r - \hat{y}$  = Error indicado.

$e = r - y$  = Error del sistema.

$w$  = Entrada de perturbaciones a la planta.

$v$  = Perturbación o ruido en el sensor.

A/D = Conversor analógico a digital.

D/A = Conversor digital a analógico.

En esta sección desarrollamos el análisis necesario para calcular la función de transferencia discreta entre las muestras que son recogidas por el convertidor A/D. La situación se dibuja en la Figura 12.



Figura 12: El prototipo del sistema de datos muestreados.

Se desea encontrar la función de transferencia discreta (mediante de la transformación  $z$ ) desde las muestras de entrada  $u(kT)$  a las muestras de salida  $y(kT)$  recogidas por el convertidor A/D. Aunque es posible que al principio sea confuso, seguimos la convención y llamamos a la función de transferencia discreta  $G(z)$  cuando la función de transferencia

continua es  $G(s)$ . Aunque  $G(z)$  y  $G(s)$  son completamente funciones diferentes, describen la misma planta, y siempre se mantiene el uso de  $s$  para la transformada continua y  $z$  para la transformada discreta[28]. Para encontrar  $G(z)$  solo necesitamos observar que  $y(kT)$  son muestras de la salida de la planta cuando la entrada es del convertidor  $D/A$ . En cuanto al convertidor  $D/A$ , asumimos que este dispositivo, comúnmente llamado retención de orden cero o  $ZOH$ , acepta una muestra  $u(kT)$  en  $t = kT$  y mantiene su salida constante en este valor hasta que se envía la siguiente muestra en  $t = kT + T$ . La salida constante a trozos del  $D/A$  es la señal  $u(t)$  que se aplica a la planta.

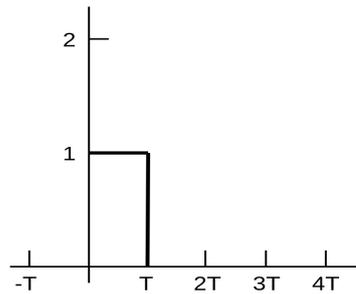


Figura 13: Salida  $D/A$  para entrada de pulso unitario.

La función de transferencia discreta es la transformada  $z$  de las muestras de la salida cuando las muestras de entrada son el pulso unitario en  $k = 0$ . Si  $u(kT) = 1$  para  $k = 0$  y  $u(kT) = 0$  para  $k \neq 0$ , la salida del convertidor  $D/A$  es un pulso de ancho  $T$  segundos y altura 1, como se indica en la Figura 13. Matemáticamente, éste pulso está dado por  $1(t) - 1(t - T)$ . Llamemos a la salida particular en respuesta al pulso que se muestra en la Figura 13  $y_1(t)$ , ésta respuesta es la diferencia entre la respuesta al escalón [a  $1(t)$ ] y la respuesta al escalón retardada [a  $1(t - T)$ ]. La transformada de Laplace de la respuesta al escalón es  $G(s)/s$ . Por lo tanto, en el dominio de transformación, la respuesta de pulso unitario de la planta es[28]

$$T_1(s) = (1 - e^{-Ts}) \frac{G(s)}{s} \quad (35)$$

y la función de transferencia requerida es la transformada  $z$  de las muestras de la inversa de  $Y_1(s)$ , que se puede expresar como

$$\begin{aligned} G(z) &= \mathcal{Z}\{Y_1(kT)\} \\ G(z) &= \mathcal{Z}\{\mathcal{L}^{-1}\{Y_1(s)\}\} \hat{=} \mathcal{Z}\{Y_1(s)\} \\ G(z) &= \mathcal{Z}\{(1 - e^{-Ts}) \frac{G(s)}{s}\} \end{aligned}$$

Ésta es la suma de dos partes, La primera es  $\mathcal{Z}\left\{\frac{G(s)}{s}\right\}$ , y la segunda es

$$\mathcal{Z}\left\{e^{-Ts}\frac{G(s)}{s}\right\} = z^{-1}\mathcal{Z}\left\{\frac{G(s)}{s}\right\}$$

porque  $e^{-Ts}$  es exactamente un retraso de un período. Por tanto, la función de transferencia es[28]

$$G(z) = (1 - z^{-1})\mathcal{Z}\left\{\frac{G(s)}{s}\right\} \quad (36)$$

### 2.2.5. Control predictivo basado en modelo.

El control predictivo basado en modelo (MBPC o MPC) se originó a fines de los años setenta y se ha desarrollado considerablemente desde entonces. El término Control Predictivo de Modelo no designa una estrategia de control específica, pero si una gama muy amplia de métodos de control que hacen uso explícito de un modelo del proceso para obtener la señal de control minimizando una función objetivo. Las ideas que aparecen en toda la familia del control predictivo son básicamente[6]:

- Uso explícito de un modelo para predecir la salida del proceso en instantes de tiempo futuros (horizonte).
- Cálculo de una secuencia de control minimizando una función objetivo.
- Estrategia de retroceso, de manera que en cada instante el horizonte se desplaza hacia el futuro, lo que implica la aplicación de la primera señal de control de la secuencia calculada en cada paso.

Los diversos algoritmos MPC sólo se diferencian entre sí en el modelo utilizado para representar (proceso y ruidos) y en la función de coste a minimizar. El MPC presenta una serie de ventajas sobre otros métodos, entre los que destacan[6]:

- Es particularmente atractivo para el personal con un conocimiento limitado de control porque los conceptos son muy intuitivos y al mismo tiempo, el ajuste es relativamente fácil.
- Se puede utilizar para controlar una gran variedad de procesos, desde aquellos con dinámicas relativamente simples hasta otros más complejos.
- El caso multivariable puede tratarse fácilmente.

- Introduce el control prealimentada (feed forward control) de forma natural para compensar las perturbaciones medibles.
- Fácil de implementar una ley de control lineal.
- Tratamiento de restricciones simple y se pueden incluir sistemáticamente durante el proceso de diseño.

La metodología de todos los controladores pertenecientes a la familia MPC se caracteriza por la siguiente estrategia, representada en la Figura 14.

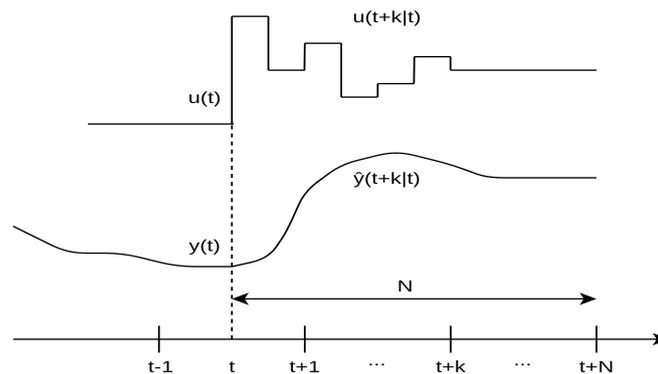


Figura 14: Estrategia MPC.

- Las salidas futuras para un horizonte determinado  $N$  (horizonte de predicción), se predicen en cada instante  $t$  utilizando el modelo de proceso. Estas salidas predichas  $y(t+k|t)$  para  $k = 1 \dots N$  dependerán de los valores conocidos hasta el instante  $t$  (entradas y salidas pasadas) y de las señales de control futuras  $u(t+k|t)$ ,  $k = 0 \dots N-1$ , las cuáles son las que se enviarán al sistema y serán calculadas[6].
- El conjunto de señales de control futuras se calculan optimizando un criterio determinado para mantener el proceso lo más cerca posible de la trayectoria de referencia  $w(t+k|t)$  (que puede ser el punto de referencia en sí o una aproximación cercana). Este criterio generalmente toma la forma de una función cuadrática de los errores entre la señal de salida predicha y la trayectoria de referencia predicha. El esfuerzo de control se incluye en la función objetivo en la mayoría de los casos. Se puede obtener una solución explícita si el criterio es cuadrático, el modelo es lineal y no hay restricciones; de lo contrario, se debe utilizar un método de optimización iterativo (Algoritmo GQP)[6].
- La señal de control  $u(t+k|t)$  se envía al proceso mientras se rechazan las siguientes señales de control calculadas, porque en el siguiente instante de muestreo  $y(t +$

1) ya se conoce y el paso 1 se repite con este nuevo valor y todas las secuencias se actualizan. Por lo tanto, la  $u(t + 1|t + 1)$  se calcula utilizando el concepto de retroceso del horizonte[6].

Para implementar esta estrategia, se utiliza la estructura básica que se muestra en la Figura 15. Se usa un modelo para predecir las salidas futuras de la planta, en función de los valores (pasados y actuales) y en las acciones de control futuras óptimas propuestas. El optimizador calcula éstas acciones teniendo en cuenta la función de costo (donde se considera el error de seguimiento futuro), así como las restricciones[6].

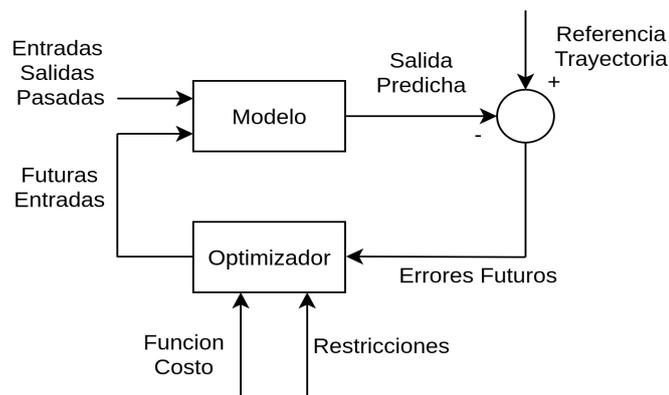


Figura 15: Estructura básica del MPC.

El modelo del proceso juega un papel decisivo en el controlador, debe ser capaz de capturar la dinámica del proceso para predecir los resultados futuros, además de ser fácil de implementar y comprender. Como el MPC no es una técnica única sino un conjunto de metodologías diferentes, existen muchos tipos de modelos utilizados en varias formulaciones.

Uno de los más populares en la industria es el Modelo de Respuesta al Escalón, que es muy simple de obtener ya que solo necesita la medición de la salida cuando el proceso se excita con una entrada tipo escalón. Es ampliamente aceptado en la industria porque es muy intuitivo y también se puede usar para procesos multivariables, de igual manera el modelo de función de transferencia es, quizás, el más extendido en la comunidad académica y se usa en la mayoría de los métodos de diseño de control, ya que es una representación que requiere solo unos pocos parámetros y es válida para todo tipo de procesos.

El optimizador es otra parte fundamental de la estrategia, ya que proporciona las acciones de control, si la función de costo es cuadrática, su mínimo se puede obtener como

una función explícita (lineal de entradas y salidas pasadas) y la trayectoria de referencia futura, en presencia de restricciones de desigualdad, la solución debe obtenerse mediante algoritmos numéricos más exigentes computacionalmente.

A continuación se indican los elementos del MPC:

Modelo de predicción.

El modelo es el pilar básico del MPC, un diseño completo debe incluir los mecanismos necesarios para obtener el mejor modelo posible el cual capturará completamente la dinámica del proceso y debe permitir calcular las predicciones. El uso del modelo del proceso está determinado por la necesidad de calcular la salida predicha en instantes futuros  $\hat{y}(t+k|t)$ . Las diferentes estrategias del MPC pueden usar varios modelos para representar la relación entre las salidas y las entradas medibles, algunas de las cuales son variables manipuladas y otras pueden considerarse perturbaciones medibles. También se debe tener en cuenta un modelo de perturbación para describir el comportamiento que no se refleja en el modelo de proceso incluido el efecto de las entradas no medibles, el ruido y los errores del modelo[6].

- Función de transferencia.

Empleada por el GPC, utiliza el concepto de función de transferencia  $G = B/A$  de modo que la salida está dada por[6]:

$$A(z^{-1})y(t) = B(z^{-1})u(t)$$

con

$$A(z^{-1}) = 1 + a_1z^{-1} + a_2z^{-2} + \dots + a_naz^{-na}$$

$$B(z^{-1}) = b_1z^{-1} + b_2z^{-2} + \dots + b_nbz^{-nb}$$

Así, la predicción viene dada por

$$\hat{y}(t+k|t) = \frac{B(z^{-1})}{A(z^{-1})}u(t+k|t)$$

Esta representación también es válida para procesos inestables y tiene la ventaja que solo necesita unos pocos parámetros.

- Modelo de perturbaciones.

La elección del modelo utilizado para representar las perturbaciones es tan importante como la elección del modelo de proceso. Un modelo muy utilizado es el Media Móvil

Integrada y Auto Regresiva Controlada (CARIMA) en el que las perturbaciones, es decir, las diferencias entre la salida medida y la calculada por el modelo, están dadas por[6]

$$n(t) = \frac{C(z^{-1})e(t)}{D(z^{-1})}$$

Donde el polinomio  $D(z^{-1})$  incluye explícitamente el integrador  $\Delta = 1 - z^{-1}$ ,  $e(t)$  es un ruido blanco de media cero y el polinomio  $C$  normalmente se considera igual a uno. Tenga en cuenta que al incluir un integrador se logra un control de estado estable sin compensación (offset-free).

Usando la ecuación diofántica

$$1 = E_k(z^{-1})D(z^{-1}) + z^{-k}F_k(z^{-1}) \quad (37)$$

Se tiene

$$n(t) = E_k(z^{-1})e(t) + z^{-k}\frac{F_k(z^{-1})}{D(z^{-1})}e(t) \quad n(t+k) = E_k(z^{-1})e(t+k) + F_k(z^{-1})n(t)$$

y la predicción será[6]

$$\hat{n}(t+k|t) = F_k(z^{-1})n(t) \quad (38)$$

Si la ecuación (38) se combina con un modelo de función de transferencia (como el que se usa en GPC), haciendo  $D(z^{-1}) = A(z^{-1})(1 - z^{-1})$ , la predicción de salida se puede obtener:

$$\hat{y}(t+k|t) = \frac{B(z^{-1})}{A(z^{-1})}u(t+k|t) + F_k(z^{-1})(y(t) - \frac{B(z^{-1})}{A(z^{-1})}u(t))$$

$$\hat{y}(t+k|t) = F_k(z^{-1})y(t) + \frac{B(z^{-1})}{A(z^{-1})}(1 - z^{-k}F_k(z^{-1}))u(t+k|t)$$

y usando (37) se obtiene la siguiente expresión para el predictor de  $k$  pasos adelante

$$\hat{y}(t+k|t) = F_k(z^{-1})y(t) + E_k(z^{-1})B(z^{-1})\Delta u(t+k|t)$$

- Respuesta libre y forzada.

Una característica típica de la mayoría de los MPC es el uso de conceptos de respuesta *libre* y *forzada*. La idea es expresar la secuencia de control como la suma de las dos señales[6]:

$$u(t) = u_f(t) + u_c(t)$$

La señal  $u_f(t)$  corresponde a las entradas pasadas y se mantiene constante e igual al último valor de la variable manipulada en instantes de tiempo futuros. Es decir:

$$u_f(t-j) = u(t-j) \quad \text{para} \quad j = 1, 2, \dots$$

$$u_f(t+j) = u(t-j) \quad \text{para} \quad j = 0, 1, 2, \dots$$

La señal  $u_c(t)$  se hace igual a cero en el pasado e igual a los siguientes movimientos de control en el futuro. Es decir:

$$u_c(t-j) = 0 \quad \text{para} \quad j = 1, 2, \dots$$

$$u_c(t+j) = u(t+j) - u(t-1) \quad \text{para} \quad j = 0, 1, 2, \dots$$

La predicción de la secuencia de salida se divide en dos partes como se puede ver en la Figura 16. Uno de ellos ( $y_f(t)$ ), la respuesta libre, corresponde a la predicción de la salida cuando la variable manipulada del proceso se iguala a  $u_f(t)$ , y la otra, la respuesta forzada ( $y_c(t)$ ), corresponde a la predicción de la salida del proceso cuando la secuencia de control se iguala a  $u_c(t)$ . La respuesta libre corresponde a la evolución del proceso debido a su estado actual, mientras que la respuesta forzada se debe a los futuros movimientos de control[6].

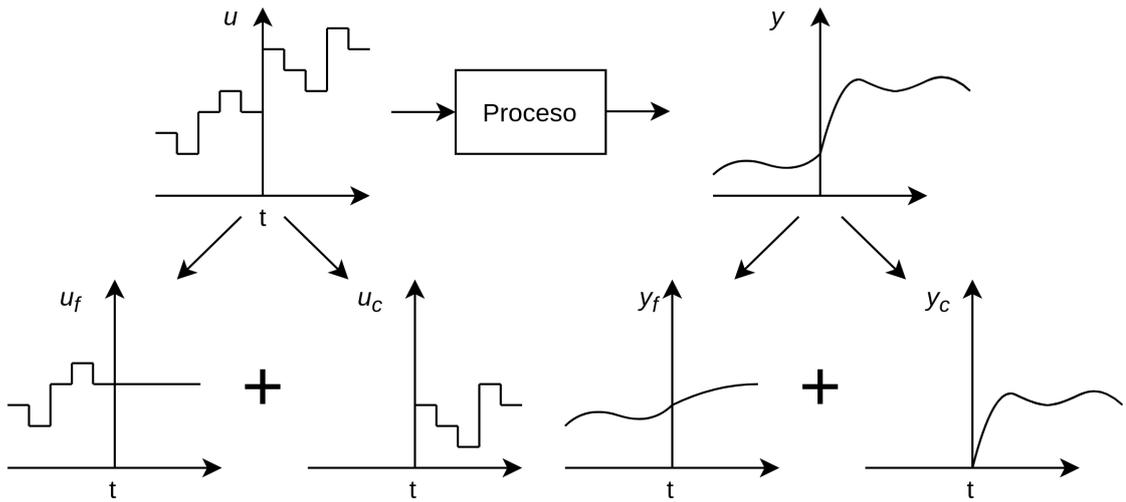


Figura 16: Respuesta libre y forzada.

Función objetivo.

Los distintos algoritmos MPC proponen diferentes funciones de coste para la obtención de la ley de control. El objetivo general es que la salida futura ( $y$ ) en el horizonte considerado debe seguir una señal de referencia determinada ( $w$ ) y, al mismo tiempo, el esfuerzo de control ( $\Delta u$ ) necesario para hacerlo debe ser penalizado. La expresión general para tal función objetivo será[6]:

$$J(N_1, N_2, N_u) = \sum_{j=N_1}^{N_2} \delta(j) [\hat{y}(t+j|t) - w(t+j)]^2 + \sum_{j=1}^{N_u} \lambda(j) [\Delta u(t+j-i)]^2 \quad (39)$$

En la función de costo se puede considerar:

- Parámetros:  $N_1$  y  $N_2$  son los horizontes de costo mínimo y máximo y  $N_u$  es el horizonte de control. El significado de  $N_1$  y  $N_2$  es bastante intuitivo, si se toma un valor elevado de  $N_1$  es porque no tiene importancia si hay errores en los primeros instantes, ésto originará una respuesta fluida del proceso. Los coeficientes  $\delta(j)$  y  $\lambda(j)$  son secuencias que consideran el comportamiento futuro, generalmente se consideran valores constantes. Por ejemplo, es posible obtener un peso exponencial de  $\delta(j)$  a lo largo del horizonte usando[6]:

$$\delta(j) = \alpha^{N_2-j}$$

Si se dá a  $\alpha$  un valor entre 0 y 1, los errores más alejados del instante  $t$  se penalizan más que los cercanos a él, dando lugar a un control más suave con menos esfuerzo. Si, por otro lado,  $\alpha > 1$  los primeros errores están más penalizados, provocando un control más estricto.

- Trayectoria de referencia: Una de las ventajas del control predictivo es que si se conoce a priori la evolución futura de la referencia, el sistema puede reaccionar antes de que el cambio se haya realizado de manera efectiva, evitando así los efectos del retraso en la respuesta del proceso. La evolución futura de la referencia  $r(t+k)$  se conoce de antemano en muchas aplicaciones, como robótica, servos o procesos por lotes. En la minimización 39, la mayoría de los métodos suelen utilizar una trayectoria de referencia  $w(t+k)$  que no necesariamente tiene que coincidir con la referencia real[6].

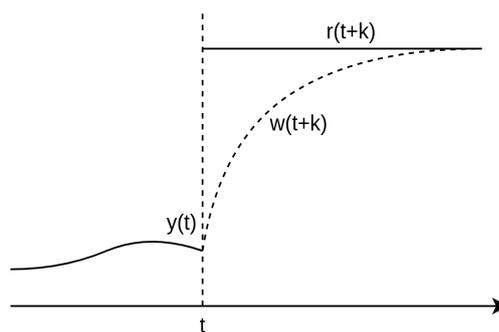


Figura 17: Trayectoria de referencia.

- Restricciones: En la práctica, todos los procesos están sujetos a limitaciones. Los actuadores tienen un campo de acción limitado así como una velocidad de respuesta determinada. Todo esto hace necesaria la introducción de restricciones en la función

a minimizar. Normalmente se considerarán límites en la amplitud, velocidad de respuesta es la señal de control y límites en la salida[6]:

$$\begin{aligned} u_{min} &\leq u(t) \leq u_{max} & \forall t \\ du_{min} &\leq u(t) - u(t-1) \leq du_{max} & \forall t \\ y_{min} &\leq y(t) \leq y_{max} & \forall t \end{aligned}$$

al agregar estas restricciones a la función objetivo, la minimización se vuelve más compleja, de modo que la solución no puede obtenerse explícitamente como en el caso no restringido.

Obtención de la ley de control.

Para obtener valores  $u(t+k|t)$  es necesario minimizar el funcional  $J$  de la ecuación 39. Para hacer esto, los valores de las salidas pronosticadas  $\hat{y}(t+k|t)$  se calculan en función de los valores pasados (de entradas y salidas) y de futuras señales de control, haciendo uso del modelo elegido y sustituido en la función de costo, obteniendo una expresión cuya minimización conduce a los valores buscados. Se puede obtener una solución analítica para el criterio cuadrático si el modelo es lineal y no hay restricciones, de lo contrario, debería utilizarse un método iterativo de optimización. Cualquiera que sea el método, obtener la solución no es fácil porque habrá  $N2 - N1 + 1$ . Para reducir este grado de libertad, se puede imponer una cierta estructura a la ley de control. Esta estructura de la ley de control a veces se impone mediante el uso del concepto de horizonte de control ( $Nu$ ) utilizado en el DMC, GPC entre otros, que consiste en considerar que luego de cierto intervalo  $Nu < N2$  no hay variación en las señales de control propuestas, esto es[6]:

$$\Delta u(t+j-1) = 0 \quad j > Nu$$

lo que equivale a dar pesos infinitos a los cambios en el control desde un determinado instante. El caso extremo sería considerar  $Nu$  igual a 1 con lo que todas las acciones futuras serían iguales a  $u(t)$ .

- Control Predictivo Generalizado.

La idea básica de GPC es calcular una secuencia de señales de control futuras de tal manera que minimice una función de costo de múltiples etapas definida sobre un horizonte de predicción. El índice a optimizar es la expectativa de una función cuadrática que mide

la distancia entre la salida prevista del sistema y alguna secuencia de referencia predicha sobre el horizonte más una función cuadrática que mide el esfuerzo de control.

La mayoría de las plantas de entrada y salida única (SISO), al considerar la operación alrededor de un set-point y después de una linealización, puede ser descrito por[6]

$$A(z^{-1})y(t) = z^{-d}B(z^{-1})u(t-1) + C(z^{-1})e(t)$$

Donde  $u(t)$  y  $y(t)$  son la secuencia de control y salida de la planta y  $e(t)$  es un ruido blanco de media cero.  $A$   $B$  y  $C$  son los siguientes polinomios en el operador de desplazamiento hacia atrás (backward)  $z^{-1}$ .

$$A(z^{-1}) = 1 + a_1z^{-1} + a_2z^{-2} + \dots + a_{na}z^{-na}$$

$$B(z^{-1}) = b_0 + b_1z^{-1} + b_2z^{-2} + \dots + b_{nb}z^{-nb}$$

$$C(z^{-1}) = 1 + c_1z^{-1} + c_2z^{-2} + \dots + c_{nc}z^{-nc}$$

Este modelo se conoce como un Controlador de Media Móvil Auto Regresiva (CARMA) y está dado por[6]:

$$A(z^{-1})y(t) = B(z^{-1})z^{-d}u(t-1) + C(z^{-1})\frac{e(t)}{\Delta} \quad (40)$$

con

$$\Delta = 1 - z^{-1}$$

Por simplicidad, el polinomio  $C$  se elige como 1. Observe que si  $C^{-1}$  se trunca entonces se puede absorber en  $A$  y  $B$ .

El algoritmo del GPC consiste en aplicar una secuencia de control que minimiza una función de costo multietapa de la forma

$$J(N_1, N_2, N_u) = \sum_{j=N_1}^{N_2} \delta(j)[\hat{y}(t+j|t) - w(t+j)]^2 + \sum_{j=1}^{N_u} \lambda(j)[\Delta u(t+j-i)]^2 \quad (41)$$

donde  $\hat{y}(t+j|t)$  es una predicción óptima de  $j$  pasos adelante de la salida del sistema en los datos hasta el tiempo  $t$ ,  $N_1$  y  $N_2$  son los horizontes de coste mínimo y máximo,  $N_u$  es el horizonte de control,  $\delta(j)$  y  $\lambda(j)$  son secuencias de ponderación y  $w(t+j)$  es la futura trayectoria de referencia.

El objetivo del control predictivo es calcular la secuencia de control futura  $u(t), u(t+1), \dots$  de tal manera que la salida futura de la planta  $y(t+j)$  se acerque a  $w(t+j)$ . Esto se logra minimizando  $J(N_1, N_2, N_u)$ .

Para optimizar la función de costo se obtendrá la predicción óptima de  $y(t + j)$  para  $j \geq N_1$  y  $j \leq N_2$ . Considere la siguiente ecuación diofántica:

$$1 = E_j(z^{-1})\tilde{A}(z^{-1}) + z^{-j}F_jz^{-1} \quad \text{con} \quad \tilde{A}(z^{-1}) = \Delta A(z^{-1}) \quad (42)$$

Los polinomios  $E_j$  y  $F_j$  se definen unívocamente con grados  $j - 1$  y  $na$  respectivamente. Se pueden obtener dividiendo 1 por  $\tilde{A}(z^{-1})$  hasta que el resto se pueda factorizar como  $z^{-j}F_j(z^{-1})$ . El cociente de la división es el polinomio  $E_j(z^{-1})$ .

Si la ecuación (40) se multiplica por  $\Delta E_j(z^{-1})z^j$ .

$$\tilde{A}(z^{-1})E_j(z^{-1})y(t + j) = E_j(z^{-1})B(z^{-1})\Delta u(t + j - d - 1) + E_j(z^{-1})e(t + j) \quad (43)$$

Considerando (42), la ecuación (40) se puede escribir como:

$$(1 - z^{-j}F_j(z^{-1}))y(t + j) = E_j(z^{-1})B(z^{-1})\Delta u(t + j - d - 1) + E_j(z^{-1})e(t + j)$$

que se puede reescribir como:

$$y(t + j) = F_j(z^{-1})y(t) + E_j(z^{-1})B(z^{-1})\Delta u(t + j - d - 1) + E_j(z^{-1})e(t + j) \quad (44)$$

Como el grado del polinomio  $E_j(z^{-1}) = j - 1$ , los términos de ruido en la ecuación (44) están todos en el futuro. Por tanto, la mejor predicción de  $y(t + j)$  es:

$$\hat{y}(t + j|t) = G_j(z^{-1})\Delta u(t + j - d - 1) + F_j(z^{-1})y(t)$$

donde  $G_j(z^{-1}) = E_j(z^{-1})B(z^{-1})$

Considere que los polinomios  $E_j$  y  $F_j$  se obtuvieron dividiendo 1 por  $\tilde{A}(z^{-1})$  hasta que el resto de la división pueda factorizarse como  $z^{-j}F_j(z^{-1})$ .

Estos polinomios se pueden expresar como[6]:

$$F_j(z^{-1}) = f_{j,0} + f_{j,1}z^{-1} + \dots + f_{j,na}z^{-na}$$

$$E_j(z^{-1}) = e_{j,0} + e_{j,1}z^{-1} + \dots + e_{j,j-1}z^{-(j-1)}$$

Suponga que se usa el mismo procedimiento para obtener  $E_{j+1}$  y  $F_{j+1}$ , es decir, dividir 1 por  $\tilde{A}(z^{-1})$  hasta que el resto de la división se pueda factorizar como  $z^{-(i+1)}F_{j+1}(z^{-1})$  con

$$F_{j+1}(z^{-1}) = f_{j+1,0} + f_{j+1,1}z^{-1} + \dots + f_{j+1,na}z^{-na}$$

Está claro que solo otro paso de la división realizado para obtener los polinomios  $E_j$  y  $F_j$  debe tomarse para obtener los polinomios  $E_{j+1}$  y  $F_{j+1}$ . El polinomio  $E_{j+1}$  vendrá dado por:

$$E_{j+1}(z^{-1}) = E_j(z^{-1}) + e_{j+1,j}z^{-j}$$

con  $e_{j+1,j} = f_{j,0}$

Los coeficientes del polinomio  $F_{j+1}$  se pueden expresar como[6]:

$$f_{j+1,i} = f_{j,i+1} - f_{j,0}\tilde{a}_{i+1} \quad i = 0 \dots na - 1$$

El polinomio  $G_{j+1}$  se puede obtener de forma recursiva de la siguiente manera:

$$G_{j+1} = E_{j+1}B = (E_j + f_{j,0}z^{-j})B$$

$$G_{j+1} = G_j + f_{j,0}z^{-j}B$$

Es decir, el primer coeficiente  $j$  de  $G_{j+1}$  será idéntico a los de  $G_j$  y los restantes coeficientes estarán dados por:

$$g_{j+1,j+i} = g_{j,j+i} + f_{j,0}b_i \quad i = 0 \dots nb$$

Para resolver el problema del GPC, el conjunto de señales de control  $u(t), u(t+1), \dots, u(t+N)$  debe obtenerse para optimizar la expresión (41). Como el sistema considerado tiene un tiempo muerto de  $d$  períodos de muestreo, la salida del sistema estará influenciada por la señal  $u(t)$  después del período de muestreo  $d+1$ . Los valores  $N_1, N_2$  y  $N_u$  que definen el horizonte se pueden definir por  $N_1 = d+1, N_2 = d+N$  y  $N_u = N$ . Note que no tiene sentido hacer  $N_1 < d+1$  ya que los términos agregados a la expresión (41) solo dependerán de las señales de control pasadas. Por otro lado, si  $N_1 > d+1$  los primeros puntos en la secuencia de referencia, no se tendrán en cuenta.

Ahora considere el siguiente conjunto de  $j$  predicciones óptimas por delante:

$$\hat{y}(t+d+1|t) = G_{d+1}\Delta u(t) + F_{d+1}y(t)$$

$$\hat{y}(t+d+2|t) = G_{d+2}\Delta u(t+1) + F_{d+2}y(t)$$

⋮

$$\hat{y}(t+d+N|t) = G_{d+N}\Delta u(t+N-1) + F_{d+N}y(t)$$

que se puede escribir como[6]:

$$\mathbf{y} = \mathbf{G}\mathbf{u} + \mathbf{F}(z^{-1})y(t) + \mathbf{G}'(z^{-1})\Delta u(t-1) \quad (45)$$

donde

$$\mathbf{y} = \begin{bmatrix} \hat{y}(t+d+1|t) \\ \hat{y}(t+d+2|t) \\ \vdots \\ \hat{y}(t+d+N|t) \end{bmatrix}$$

$$\mathbf{u} = \begin{bmatrix} \Delta u(t) \\ \Delta u(t+1) \\ \vdots \\ \Delta u(t+N-1) \end{bmatrix}$$

$$\mathbf{G} = \begin{bmatrix} g_0 & 0 & \cdots & 0 \\ g_1 & g_0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ g_{N-1} & g_{N-2} & \cdots & g_0 \end{bmatrix}$$

$$\mathbf{G}'(z^{-1}) = \begin{bmatrix} (G_{d+1}(z^{-1}) - g_0)z \\ (G_{d+2}(z^{-1}) - g_0 - g_1z^{-1})z^2 \\ \vdots \\ (G_{d+N}(z^{-1}) - g_0 - g_1z^{-1} - \dots - g_{N-1}z^{-(N-1)})z^N \end{bmatrix}$$

$$\mathbf{F}(z^{-1}) = \begin{bmatrix} F_{d+1}(z^{-1}) \\ F_{d+2}(z^{-1}) \\ \vdots \\ F_{d+N}(z^{-1}) \end{bmatrix}$$

Observe que los dos últimos términos de la ecuación (45) solo dependen del pasado y se pueden agrupar en  $\mathbf{f}$ , lo que conduce a:

$$\hat{\mathbf{y}} = \mathbf{G}\mathbf{u} + \mathbf{f}$$

Observe que si todas las condiciones iniciales son cero, la respuesta libre  $\mathbf{f}$  también es cero. Si se aplica un paso unitario a la entrada en el tiempo  $t$ ; es decir

$$\Delta u(t) = 1, \Delta u(t+1) = 0, \dots, \Delta u(t+N-1) = 0$$

la secuencia de salida esperada  $[\hat{y}(t+1), \hat{y}(t+2), \dots, \hat{y}(t+N)]^T$  es igual a la primera columna de la matriz  $\mathbf{G}$ . Es decir, La primera columna de la matriz  $\mathbf{G}$  se puede calcular como la respuesta al escalón de la planta cuando se aplica un escalón unitario a la variable manipulada. El término de respuesta libre se puede calcular de forma recursiva mediante[6]:

$$\mathbf{f}_{j+1} = z(1 - \tilde{A}(z^{-1}))\mathbf{f}_j + B(z^{-1})\Delta u(t-d+j)$$

con  $\mathbf{f}_0 = y(t)$  y  $\Delta u(t+j) = 0$  para  $j \geq 0$

La expresión (41) se puede escribir como:

$$\mathbf{J} = (\mathbf{G}\mathbf{u} + \mathbf{f} - \mathbf{w})^T (\mathbf{G}\mathbf{u} + \mathbf{f} - \mathbf{w}) + \lambda \mathbf{u}^T \mathbf{u} \quad (46)$$

donde:

$$\mathbf{w} = [w(t+d+1) \quad w(t+d+2) \cdots w(t+d+N)]^T$$

La ecuación (47) se puede escribir como:

$$J = \frac{1}{2} \mathbf{u}^T \mathbf{H} \mathbf{u} + \mathbf{b}^T \mathbf{u} + \mathbf{f}_0. \quad (47)$$

donde:

$$\mathbf{H} = 2(\mathbf{G}^T \mathbf{G} + \lambda \mathbf{I})$$

$$\mathbf{b}^T = 2(\mathbf{f} - \mathbf{w})^T \mathbf{G}$$

$$\mathbf{f}_0 = (\mathbf{f} - \mathbf{w})^T (\mathbf{f} - \mathbf{w})$$

El mínimo de  $J$ , asumiendo que no hay restricciones en las señales de control, se puede encontrar haciendo que el gradiente de  $J$  sea igual a cero, lo que lleva a[6]:

$$\mathbf{u} = -\mathbf{H}^{-1} \mathbf{b} = (\mathbf{G}^T \mathbf{G} + \lambda \mathbf{I})^{-1} \mathbf{G}^T (\mathbf{w} - \mathbf{f}) \quad (48)$$

Observe que la señal de control que se envía realmente al proceso es el primer elemento del vector  $\mathbf{u}$ , que viene dado por:

$$\Delta u(t) = \mathbf{K}(\mathbf{w} - \mathbf{f}) \quad (49)$$

donde  $\mathbf{K}$  es la primera fila de la matriz  $(\mathbf{G}^T \mathbf{G} + \lambda \mathbf{I})^{-1} \mathbf{G}^T$ . Esto tiene un significado claro, que se puede derivar fácilmente de la Figura 18: si no hay errores de predicción en el futuro, es decir, si  $(\mathbf{w} - \mathbf{f}) = 0$ , entonces no hay movimiento de control, ya que el objetivo se cumplirá con la libre evolución del proceso. Sin embargo, en el otro caso, habrá un incremento en la acción de control proporcional (con un factor  $\mathbf{K}$ ) a ese error futuro. Tenga en cuenta que la acción se toma con respecto a errores *futuros*, no errores *pasados*, como es el caso de los controladores de retroalimentación convencionales. Observe que solo se aplica el primer elemento de  $\mathbf{u}$  y el procedimiento se repite en el siguiente tiempo de muestreo.

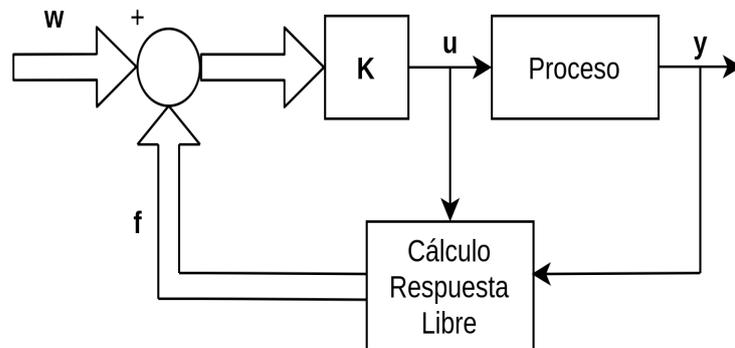


Figura 18: Ley de Control GPC.

- Restricciones en el MPC[6].

Los controladores donde se considera que todas las señales poseen un rango ilimitado, no son muy realistas porque en la práctica todos los procesos están sujetos a restricciones. Los actuadores tienen un rango de acción y una velocidad de respuesta limitada, como es el caso de las válvulas de control que están limitadas por una posición completamente cerrada y completamente abierta y una velocidad máxima de respuesta. Razones constructivas y / o de seguridad, así como el rango del sensor, provocan límites en las variables del proceso. Además, en la práctica, los puntos de operación de las plantas están determinados para satisfacer metas económicas y se encuentran en la intersección de ciertas limitaciones. Las limitaciones de salida se deben principalmente a razones de seguridad, y deben controlarse de antemano porque las variables de salida se ven afectadas por la dinámica del proceso. Las variables de entrada (o manipuladas) siempre pueden mantenerse limitadas por el controlador recortando la acción de control a un valor que satisfaga las restricciones de amplitud y velocidad de respuesta[6].

Las acciones de control de GPC se obtuvieron calculando el vector  $\mathbf{u}$  de incrementos de control futuros que minimiza una función objetivo cuadrática dada por:

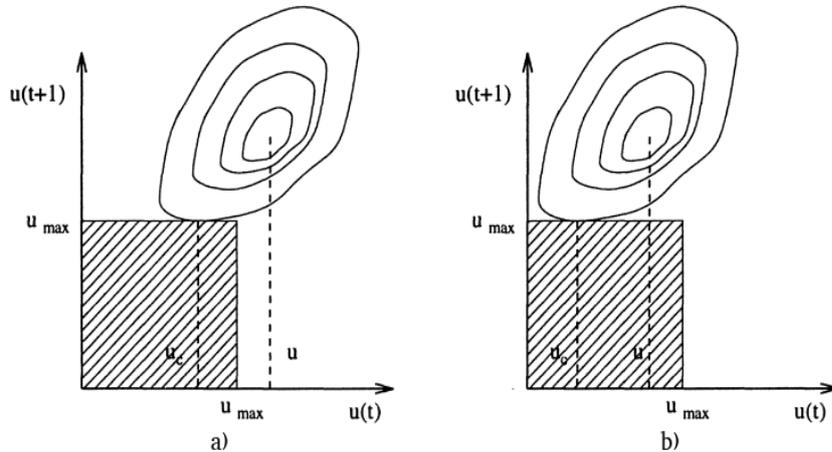


Figura 19: Restricciones en la señal de control.

$$J(\mathbf{u}) = \frac{1}{2} \mathbf{u}^T \mathbf{H} \mathbf{u} + \mathbf{b} \mathbf{u} + \mathbf{f}_0 \quad (50)$$

La solución óptima de este problema se encuentra resolviendo la ecuación lineal:

$$\mathbf{H} \mathbf{u} = -\mathbf{b}$$

En la práctica, la forma normal de utilizar un MPC es calcular  $u(t)$  como se describió anteriormente y aplicarlo al proceso. Si  $u(t)$  viola la restricción, se satura hasta sus límites,

ya sea por el programa de control o por el actuador. El caso de  $u(t + 1), \dots, u(t + N)$  que viola las restricciones ni siquiera se considera, ya que en la mayoría de los casos éstas señales ni siquiera se calculan.

Para ilustrar este punto, considere los casos de violaciones de restricciones que se muestran en la Figura 19 de un problema de GPC con un horizonte de control de dos. La Figura 19a muestra el caso donde  $u(t) > u_{max}$ . En este caso, la forma normal de operar sería aplicar  $u_{max}$  al proceso en lugar de  $u_c$  donde se alcanza el mínimo de  $J$  cuando se consideran las restricciones. En el caso que se muestra en la Figura 19b,  $u(t)$  no viola las restricciones y se aplicaría al sistema en lugar de la señal óptima  $u_c$  que debería aplicarse cuando se tengan en cuenta las restricciones[6].

- La restricción del incremento de control esta definida como:

$$\mathbf{u}_{min} \leq \Delta \mathbf{u} \leq \mathbf{u}_{max}$$

Esa restricción debemos expresarla en la forma de desigualdad que se mostró en la ecuación patrón:

$$\mathbf{A}\Delta \mathbf{u} \leq \mathbf{b}$$

Tomemos el primer intervalo:

$$\Delta \mathbf{u} \leq \mathbf{u}_{max}$$

Representándolo de forma matricial tenemos:

$$\begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+N_u-1) \end{bmatrix} \leq \begin{bmatrix} \Delta u_{max} \\ \Delta u_{max} \\ \vdots \\ \Delta u_{max} \end{bmatrix}$$

Entonces tenemos que para el incremento de control máximo esta dado por:

$$\mathbf{I}_{Nu \times Nu} \Delta \mathbf{u} \leq u_{max} \mathbf{1}$$

Por otro lado tomando la restricción del incremento de control mínimo:

$$\mathbf{u}_{min} \leq \Delta \mathbf{u}$$

en forma matricial tendríamos que

$$\begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+N_u-1) \end{bmatrix} \geq \begin{bmatrix} \Delta u_{min} \\ \Delta u_{min} \\ \vdots \\ \Delta u_{min} \end{bmatrix}$$

en forma reducida podemos expresarlo como:

$$\mathbf{I}_{N_u \times N_u} \Delta \mathbf{u} \geq u_{min} \mathbf{1}$$

Como el sentido de la desigualdad para el caso del incremento de control mínimo no es el mismo que el de la formula patrón, multiplicamos por menos uno en ambos lados de la desigualdad para cambiarle el sentido a la misma:

$$(-\mathbf{I}_{N_u \times N_u}) \Delta \mathbf{u} \leq (-u_{min} \mathbf{1})$$

Asi conseguimos llevarla a la misma representación. Por último vamos a mezclar los dos extremos de cada restricción en una restricción única.

Creo una matriz con un bloque de matrices identidades, multiplicado por  $\Delta u$  menor igual un bloque de vectores de  $U_{maximos}$  sobre un bloque de vectores de  $U_{minimos}$  asi:

$$\begin{bmatrix} \mathbf{I} \\ \dots \\ -\mathbf{I} \end{bmatrix} \Delta \mathbf{u} \leq \begin{bmatrix} u_{min} \mathbf{1} \\ \dots \\ -u_{min} \mathbf{1} \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \\ -1 & 0 & \dots & 0 \\ 0 & -1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & -1 \end{bmatrix} \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+N_u-1) \\ \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+N_u-1) \end{bmatrix} \leq \begin{bmatrix} u_{max} \\ u_{max} \\ \vdots \\ u_{max} \\ -u_{min} \\ -u_{min} \\ \vdots \\ -u_{min} \end{bmatrix}$$

- La restricción de la señal de control esta definida como:

$$u_{min} \leq u(k) \leq u_{max} \quad \forall k \in [0, N_u-1)$$

Podemos representar en forma matricial la ley de control

$$u_{min} \mathbf{1} \leq \begin{bmatrix} 1 & 0 & \dots & 0 \\ 1 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \dots & 1 \end{bmatrix} \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+N_u-1) \end{bmatrix} + \begin{bmatrix} u(k-1) \\ u(k-1) \\ \vdots \\ u(k-1) \end{bmatrix} \leq u_{max} \mathbf{1}$$

De forma rápida, podemos representar la matriz triangular inferior que acompaña el incremento de control como  $\mathbf{T}$ , así la restricción está representada por:

$$\begin{bmatrix} \mathbf{T} \\ \dots \\ -\mathbf{T} \end{bmatrix} \Delta \mathbf{u} \leq \begin{bmatrix} \mathbf{1}u_{max} - \mathbf{1}u(k-1) \\ \dots \\ -\mathbf{1}u_{min} + \mathbf{1}u(k-1) \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & \dots & 0 \\ 1 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \dots & 1 \\ -1 & 0 & \dots & 0 \\ -1 & -1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & \dots & -1 \end{bmatrix} \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+N_u-1) \\ \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+N_u-1) \end{bmatrix} \leq \begin{bmatrix} u_{max} - u(k-1) \\ u_{max} - u(k-1) \\ \vdots \\ u_{max} - u(k-1) \\ -u_{min} + u(k-1) \\ -u_{min} + u(k-1) \\ \vdots \\ -u_{min} + u(k-1) \end{bmatrix}$$

Las restricciones son hechas en base al modelo y debemos tener un muy buen modelo para garantizar que las restricciones de salida se cumplan.

- La restricción de salida debe representarse en función de  $\Delta u$ .

$$\mathbf{y}_{min} \leq \mathbf{y} \leq \mathbf{y}_{max} \quad \forall k \in [N_1, N_2)$$

sabemos que la salida se representa por:

$$\mathbf{y} = \mathbf{G}\mathbf{u} + \mathbf{f}$$

$$\mathbf{1}y_{min} \leq \mathbf{G}\mathbf{u} + \mathbf{f} \leq \mathbf{1}y_{max}$$

$$\mathbf{1}y_{min} - \mathbf{f} \leq \mathbf{G}\mathbf{u} \leq \mathbf{1}y_{max} - \mathbf{f}$$

Así la restricción viene dada por:

$$\begin{bmatrix} \mathbf{G} \\ \dots \\ -\mathbf{G} \end{bmatrix} \Delta \mathbf{u} \leq \begin{bmatrix} \mathbf{1}y_{max} - \mathbf{f} \\ \dots \\ -\mathbf{1}y_{min} + \mathbf{f} \end{bmatrix}$$

$$\mathbf{G} = \begin{bmatrix} g_0 & 0 & \cdots & 0 \\ g_1 & g_0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ g_{N-1} & g_{N-2} & \cdots & g_0 \\ -g_0 & 0 & \cdots & 0 \\ -g_1 & -g_0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ -g_{N-1} & -g_{N-2} & \cdots & -g_0 \end{bmatrix} \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+N_u-1) \\ \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+N_u-1) \end{bmatrix} \leq \begin{bmatrix} y_{max} - f(k) \\ y_{max} - f(k) \\ \vdots \\ y_{max} - f(k) \\ -y_{min} + f(k) \\ -y_{min} + f(k) \\ \vdots \\ -y_{min} + f(k) \end{bmatrix}$$

### 2.2.6. Programación cuadrática QP.

La optimización se ha expandido en todas direcciones a un ritmo asombroso durante las últimas décadas. La optimización ha sido una herramienta básica en todas las áreas de matemática aplicada, ingeniería, medicina, economía y otras ciencias[29].

La serie de Taylor es una herramienta esencial para analizar una función cuadrática. Expresa una función cuadrática en un nuevo punto en términos de su valor en otro punto y cantidades funcionales cambiadas.

Sea  $\mathbf{x}$  un  $n$ -vector con  $x = (x_1, \dots, x_n)'$ . Una función cuadrática general de  $n$  variables,  $f(\mathbf{x})$ , puede estar escrito

$$f(\mathbf{x}) = \sum_{i=1}^n c_i x_i + \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \gamma_{ij} x_i x_j, \quad (51)$$

Donde  $c_i$  y  $\gamma_{ij}$  son constantes. Podemos asumir sin pérdida de generalidad que  $\gamma_{ij} = \gamma_{ji}$ . Sea

$$\mathbf{c} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix} \quad \text{y} \quad \mathbf{C} = \begin{bmatrix} \gamma_{ji} & \gamma_{ji} & \cdots & \gamma_{ji} \\ \gamma_{ji} & \gamma_{ji} & \cdots & \gamma_{ji} \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_{ji} & \gamma_{ji} & \cdots & \gamma_{ji} \end{bmatrix}$$

Entonces  $\mathbf{c}$  es un  $n$ -vector,  $\mathbf{C}$  es una matriz simétrica  $(n, n)$ , y (52) puede ser escrito en la forma más compacta

$$f(\mathbf{x}) = \mathbf{c}'\mathbf{x} + \frac{1}{2}\mathbf{x}'\mathbf{C}\mathbf{x}. \quad (52)$$

El gradiente de  $f$  en  $x$  es el  $n$ -vector

$$\begin{bmatrix} \frac{\partial f(x)}{\partial x_1} \\ \frac{\partial f(x)}{\partial x_2} \\ \vdots \\ \frac{\partial f(x)}{\partial x_n} \end{bmatrix}$$

y se denota por  $g(\mathbf{x})$ . Entonces se puede escribir como[29]:

$$g(\mathbf{x}) = \mathbf{c} + \mathbf{C}\mathbf{x} \quad (53)$$

La matriz Hessiana de  $f$  en  $x$  es la matriz  $(n, n)$  cuyo elemento  $(i, j)$  es

$$\frac{\partial^2 f(x)}{\partial x_i \partial x_j},$$

y se denota por  $H(\mathbf{x})$ .

$$H(\mathbf{x}) = \mathbf{C} \quad (54)$$

La serie de Taylor de  $h$  sobre un punto dado  $x_0$  es

$$h(x) = h(x_0) + \frac{1}{1!} \frac{dh(x_0)}{dx} (x - x_0) + \frac{1}{2!} \frac{d^2h(x_0)}{dx^2} (x - x_0)^2 + \frac{1}{3!} \frac{d^3h(x_0)}{dx^3} (x - x_0)^3 \dots \quad (55)$$

Usando las cantidades vectoriales que acabamos de desarrollar, existe una serie de Taylor análoga para  $f(\mathbf{x})$  como en (54). Sea  $\mathbf{x}_0$  un  $n$ -vector dado. La serie Taylor para  $f$  acerca de  $\mathbf{x}_0$  es

$$f(\mathbf{x}) = f(\mathbf{x}_0) + g'(\mathbf{x}_0)(\mathbf{x} - \mathbf{x}_0) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_0)' \mathbf{C}(\mathbf{x} - \mathbf{x}_0) \quad (56)$$

Estrechamente relacionado con (57) está la identidad.

$$g(\mathbf{x}) = g(\mathbf{x}_0) + \mathbf{C}(\mathbf{x} - \mathbf{x}_0)$$

■ Algoritmo General de Programación Cuadrática: GQPA[29].

Considere el problema del modelo

$$\left. \begin{array}{l} \text{Minimizar : } \mathbf{c}^T \mathbf{x} + \frac{1}{2} \mathbf{x}^T \mathbf{C} \mathbf{x} \\ \text{Sujeto a : } \mathbf{a}_i^T \mathbf{x} \leq \mathbf{b}_i, \quad i = 1, \dots, m, \\ \mathbf{a}_i \mathbf{x} = \mathbf{b}_i, \quad i = m + 1, \dots, m + r, \end{array} \right\} \quad (57)$$

Suponga que  $\mathbf{C}$  es semidefinida positiva y que  $\mathbf{a}_{m+1}, \dots$ , son linealmente independientes. Sea  $f(\mathbf{x}) = \mathbf{c}^T \mathbf{x} + \frac{1}{2} \mathbf{x}^T \mathbf{C} \mathbf{x}$ . Ahora formulamos un algoritmo para la solución de 58 que construye una secuencia de puntos  $x_0, x_1, x_2, \dots$  de acuerdo a  $x_{j+1} = x_j - \sigma_j s_j, j = 0, 1, \dots$  y después de un número finito de pasos cualquiera determina una solución óptima para (58) o determina que (58) no tiene límites desde abajo. Llamaremos  $\mathbf{s}_j$  y  $\sigma_j$  la dirección de búsqueda y el tamaño del paso, respectivamente, en la iteración  $j$ . Para cualquier  $x \in R$ , sea

$$I(x) = \{i | \mathbf{a}_i^T \mathbf{x} = \mathbf{b}_i, 1 \leq i \leq m + r\}$$

de modo que  $I(x)$  denota el conjunto (desordenado) de índices de restricciones activas en  $x$ . Por definición, cada uno de  $m + 1, \dots, m + r$  está en  $I(x)$ . En la iteración  $j$ , sea  $K_j$  un subconjunto ordenado de  $I(x_j)$  tal que  $\mathbf{a}_i$ , para todo  $i \in K_j$ , son linealmente independientes. Nos referimos a  $K_j$  como el conjunto activo en la iteración  $j$ . Suponga primero que  $x_j$  no es un punto cuasi estacionario. Elegimos  $s_j$  tal que  $s_j$  sea una solución óptima para

$$\min\{f(x_j - s) \mid \mathbf{a}_i^T(x_j - s) = \mathbf{b}_i, \text{ Para todo } i \in K_j\}, \quad (58)$$

siempre que (59) tenga una solución óptima. Sea  $\mathbf{A}_j^T$  la matriz cuyas columnas son aquellas  $\mathbf{a}_i$  con  $i \in K_j$  y ordenada según  $K_j$ . De la serie de Taylor (57), la función objetivo para (59) puede escribirse

$$f(\mathbf{x}_j - \mathbf{s}_j) = f(\mathbf{x}_j) - \mathbf{g}_j^T \mathbf{s}_j + \frac{1}{2} \mathbf{s}_j^T \mathbf{C} \mathbf{s}_j,$$

donde  $\mathbf{g}_j = g(\mathbf{x}_j)$ . Las restricciones se vuelven

$$\mathbf{A}_j \mathbf{s}_j = 0.$$

Omitiendo el término constante  $f(\mathbf{x}_j)$  en la función objetivo, (59) es equivalente a

$$\min\{-\mathbf{g}_j^T \mathbf{s}_j + \frac{1}{2} \mathbf{s}_j^T \mathbf{C} \mathbf{s}_j \mid \mathbf{A}_j \mathbf{s}_j = 0\}. \quad (59)$$

#### Teorema 1

Sea  $f(\mathbf{x}) = \mathbf{c}^T \mathbf{x} + \frac{1}{2} \mathbf{x}^T \mathbf{C} \mathbf{x}$  convexo. Entonces  $\mathbf{x}_0$  es una solución óptima para  $\min\{\mathbf{c}^T \mathbf{x} + \frac{1}{2} \mathbf{x}^T \mathbf{C} \mathbf{x} \mid \mathbf{A} \mathbf{x} = \mathbf{b}\}$  si y solo si  $\mathbf{A} \mathbf{x}_0 = \mathbf{b}$  y hay un  $r$ -vector  $\mathbf{u}$  con  $-g(\mathbf{x}_0) = \mathbf{A}' \mathbf{u}$

Según el Teorema 1,  $\mathbf{s}_j$  es óptimo para (60) si y solo si hay un vector  $\mathbf{v}_j$  que con  $\mathbf{s}_j$  satisface

$$\begin{aligned} \mathbf{C} \mathbf{s}_j + \mathbf{A}_j^T \mathbf{v}_j &= \mathbf{g}_j \\ \mathbf{A}_j \mathbf{s}_j &= 0 \end{aligned}$$

Estos se pueden escribir en forma de matriz particionada como

$$\begin{bmatrix} \mathbf{C} & \mathbf{A}_j^T \\ \mathbf{A}_j & 0 \end{bmatrix} \begin{bmatrix} \mathbf{s}_j \\ \mathbf{v}_j \end{bmatrix} = \begin{bmatrix} \mathbf{g}_j \\ 0 \end{bmatrix}. \quad (60)$$

Sea

$$\mathbf{H}_j = \begin{bmatrix} \mathbf{C} & \mathbf{A}_j^T \\ \mathbf{A}_j & 0 \end{bmatrix}$$

la matriz de coeficientes de (61). Si  $\mathbf{H}_j$  no es singular, llamamos  $\mathbf{s}_j$  en la solución de (61) la dirección de Newton en  $\mathbf{x}_j$ . Habiendo obtenido  $\mathbf{s}_j$  de (61), calculamos el tamaño

de paso máximo factible  $\hat{\sigma}_j$  para  $\mathbf{s}_j$  y establezca  $\sigma_j = \min\{1, \hat{\sigma}_j\}$ . Si  $\sigma_j = 1$ , entonces  $x_{j+1}$  es un punto cuasi estacionario para (58). Si  $\sigma_j < 1$ , entonces alguna restricción  $l \notin K_j$  se activa en  $\mathbf{x}_j + 1$  y aumentamos  $K_j$  con  $l$  para obtener el nuevo conjunto activo  $K_j + 1$ . Si hay restricciones  $\rho_j$  en el conjunto activo en la iteración  $j$ , entonces en como máximo  $n - \rho_j$  tales iteraciones se determinará un punto cuasi estacionario o habrá  $n$  restricciones en el conjunto activo en la iteración  $j + n - \rho_j$ . En este último caso, el punto correspondiente es un punto extremo. Dado que un punto extremo es un caso especial de un punto cuasi estacionario, de ello se deduce que debe obtenerse un punto cuasi estacionario en como máximo  $n - \rho_j$  iteraciones.

Ahora suponga que  $\sigma_j = 1$  de modo que  $\mathbf{x}_j + 1$  es un punto cuasiestacionario. Los multiplicadores para las restricciones activas en  $\mathbf{x}_j + 1$  pueden obtenerse de  $\mathbf{v}_j$  en la solución de (61) como sigue. Por definición,  $\mathbf{g}_{j+1} = \mathbf{c} + \mathbf{C}\mathbf{x}_{j+1}$ , de modo que desde la primera partición de (61),

$$\mathbf{g}_{j+1} = \mathbf{c} + \mathbf{C}\mathbf{x}_{j+1} = \mathbf{A}_j^T \mathbf{v}_j.$$

Es decir,

$$-\mathbf{g}_{j+1} = \mathbf{A}_j^T (-\mathbf{v}_j), \quad (61)$$

y los multiplicadores de las restricciones activas pueden obtenerse negando los componentes de  $\mathbf{v}_j$ . El vector  $(m+r)$ -dimensional de multiplicadores  $\mathbf{u}_j + 1$  se puede obtener estableciendo los componentes asociados con restricciones inactivas a cero y usando  $K_j$  para establecer el resto en el componente apropiado de  $-\mathbf{v}_j$ .

Sea  $k$  tal que

$$(\mathbf{u}_{j+1})_k = \min\{(\mathbf{u}_{j+1})_i | i = 1, \dots, m\}. \quad (62)$$

Si  $(u_{j+1})_k \geq 0$ , entonces  $x_{j+1}$  es una solución óptima. Si  $(u_{j+1})_k \leq 0$ , procedemos eliminando la restricción  $k$  del conjunto activo. Es decir, obtenemos  $K_{j+1}$  de  $K_j$  eliminando  $k$ . Si  $\mathbf{H}_{j+1}$  no es singular, continuamos determinando la nueva dirección de Newton  $\mathbf{s}_{j+1}$  de (61) con  $j$  reemplazado por  $j + 1$ . Queda por considerar el caso de que  $\mathbf{H}_{j+1}$  sea singular. Suponga que  $\mathbf{H}_{j+1}$  es singular. Por definición de singularidad, hay un vector distinto de cero, que cuando se multiplica por  $\mathbf{H}_{j+1}$ , da cero. Deje que este vector sea  $(\mathbf{s}_{j+1}^T, \mathbf{v}_{j+1}^T)^T$ . Entonces

$$\mathbf{H}_{j+1} = \begin{bmatrix} \mathbf{s}_{j+1} \\ \mathbf{v}_{j+1} \end{bmatrix} = \begin{bmatrix} \mathbf{C} & \mathbf{A}_{j+1}^T \\ \mathbf{A}_{j+1} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{s}_{j+1} \\ j+1 \end{bmatrix} = \begin{bmatrix} \mathbf{C}\mathbf{s}_{j+1} & \mathbf{A}_{j+1}^T \mathbf{v}_{j+1} \\ \mathbf{A}_{j+1} \mathbf{s}_{j+1} & 0 \end{bmatrix} = 0$$

así que

$$\mathbf{C}\mathbf{s}_{j+1} + \mathbf{A}_{j+1}^T \mathbf{v}_{j+1} = 0 \quad \text{y} \quad (63)$$

$$\mathbf{A}_{j+1}\mathbf{s}_{j+1} = 0 \quad (64)$$

Multiplicar (64) a la izquierda con  $\mathbf{s}_{j+1}^T$  y usando (65) da

$$\mathbf{s}_{j+1}^T \mathbf{C}\mathbf{s}_{j+1} + (\mathbf{A}_{j+1}\mathbf{s}_{j+1})^T \mathbf{v}_{j+1} = \mathbf{s}_{j+1}^T \mathbf{C}\mathbf{s}_{j+1} = 0$$

Dado que  $\mathbf{C}$  es semidefinito positivo, esto implica que  $\mathbf{C}\mathbf{s}_{j+1} = 0$ . Con 64, esto implica que  $\mathbf{A}_{j+1}^T \mathbf{v}_{j+1} = 0$ . Pero como  $\mathbf{A}_{j+1}$  tiene rango de fila completo, debe ser el caso que  $\mathbf{v}_{j+1} = 0$ . Por tanto,  $\mathbf{H}_{j+1}$  singular implica que hay una  $\mathbf{s}_{j+1} \neq 0$  tal que

$$\mathbf{C}\mathbf{s}_{j+1} = 0 \quad \text{y} \quad \mathbf{A}_{j+1}\mathbf{s}_{j+1} = 0 \quad (65)$$

Además, sea  $k$  el índice de la restricción que se eliminó del conjunto activo en la iteración anterior. Entonces

$$\mathbf{A}_j^T = \left[ \mathbf{A}_{j+1}^T, \mathbf{a}_k \right]. \quad (66)$$

Si  $\mathbf{a}_k^T \mathbf{s}_{j+1} = 0$ , luego de (66)  $\mathbf{A}_j \mathbf{s}_{j+1} = 0$  y  $\mathbf{C}\mathbf{s}_{j+1} = 0$ . Esto implicaría que

$$\mathbf{H}_j = \begin{bmatrix} \mathbf{s}_{j+1} \\ 0 \end{bmatrix} = 0,$$

lo que a su vez implica que  $\mathbf{H}_j$  es singular. Esto es una contradicción y establece que  $\mathbf{a}_k^T \mathbf{s}_{j+1} \neq 0$ . Reemplazando  $\mathbf{s}_{j+1}$  con  $(\mathbf{a}_k^T \mathbf{s}_{j+1})^{-1} \mathbf{s}_{j+1}$ , ahora podemos asumir que

$$\mathbf{a}_k^T \mathbf{s}_{j+1} = 1 \quad (67)$$

Finalmente, de (62), (63), (66), (67) y (68)

$$\mathbf{g}_{j+1}^T \mathbf{s}_{j+1} = -(\mathbf{u}_{j+1})_k \mathbf{a}_k^T \mathbf{s}_{j+1} = -(\mathbf{u}_{j+1})_k,$$

así que  $\mathbf{g}_{j+1}^T \mathbf{s}_{j+1} > 0$ . Pero luego de la serie de Taylor (57) y (66)

$$\begin{aligned} f(\mathbf{x}_{j+1} - \sigma \mathbf{s}_{j+1}) &= f(\mathbf{x}_{j+1}) - \sigma \mathbf{g}_{j+1}^T \mathbf{s}_{j+1} + \frac{1}{2} \sigma^2 \mathbf{s}_{j+1}^T \mathbf{C}\mathbf{s}_{j+1} \\ &= f(\mathbf{x}_{j+1}) - \sigma \mathbf{g}_{j+1}^T \mathbf{s}_{j+1} \\ &\rightarrow -\infty \text{ como } \sigma \rightarrow +\infty. \end{aligned}$$

De (66),  $\mathbf{x}_{j+1} - \sigma \mathbf{s}_{j+1}$  es factible para

$$\min \left\{ -\mathbf{g}_{j+1}^T \mathbf{s}_{j+1} + \frac{1}{2} \mathbf{s}_{j+1}^T \mathbf{C}\mathbf{s}_{j+1} \mid \mathbf{A}_{j+1} \mathbf{s}_{j+1} = 0 \right\} \quad (68)$$

para todo  $\sigma \geq 0$ . Esto implica que eliminar la restricción  $k$  del conjunto activo da como resultado la siguiente dirección encontrando que el subproblema (69) es ilimitado

desde abajo. Sea  $\hat{\sigma}_{j+1}$  el tamaño de paso máximo factible para  $s_{j+1}$ . Si  $\hat{\sigma}_{j+1} = +\infty$ , entonces (58) no tiene límites desde abajo. Si  $\hat{\sigma}_{j+1} < +\infty$ , sea  $l$  el índice de la restricción restrictiva. Establecer  $\sigma_{j+1} = \hat{\sigma}_{j+1}$  da como resultado  $f(\mathbf{x}_{j+2}) < f(\mathbf{x}_{j+1})$  (siempre que  $\hat{\sigma}_{j+1} > 0$ ). Además, aumentar  $K_{j+1}$  con  $l$  para dar  $K_{j+2}$  da como resultado que  $\mathbf{H}_{j+2}$  no sea singular. Por tanto, nunca puede haber dos instancias consecutivas en las que  $\mathbf{H}_j$  sea singular. A lo largo del algoritmo, establecemos  $\gamma_j = 1$  si  $\mathbf{H}_j$  es no singular y  $\gamma_j = 0$  si  $\mathbf{H}_j$  es singular.

Para iniciar el algoritmo, requerimos un punto factible  $x_0$  y  $K_0 \subseteq I(x_0)$  satisfaciendo el Supuesto 1:

1.  $\mathbf{a}_i$  para todo  $i \in K_0$  son linealmente independientes,
2.  $\mathbf{s}'\mathbf{C}\mathbf{s} > 0$  para todo  $\mathbf{s} \neq 0$  con  $\mathbf{A}_0\mathbf{s} = 0$

A continuación se representa el proceso que sigue el algoritmo GQP para encontrar el mínimo de la función cuadrática[29].

Problema modelo:

$$\text{Minimizar : } \mathbf{c}^T\mathbf{x} + \frac{1}{2}\mathbf{x}^T\mathbf{C}\mathbf{x}$$

$$\text{Sujeto a : } \mathbf{a}_i^T\mathbf{x} \leq \mathbf{b}_i, \quad i = 1, \dots, m,$$

$$\mathbf{a}_i\mathbf{x} = \mathbf{b}_i, \quad i = m + 1, \dots, m + r,$$

Inicialización:

Comience con cualquier punto factible  $\mathbf{x}_0$  y conjunto activo  $K_0 \subseteq I(\mathbf{x}_0)$  de modo que se satisfaga el Supuesto 1. Calcular  $f(\mathbf{x}_0) = \mathbf{c}^T\mathbf{x}_0 + \frac{1}{2}\mathbf{x}_0^T\mathbf{C}\mathbf{x}_0$ ,  $\mathbf{g}_0 = \mathbf{c} + \mathbf{C}\mathbf{x}_0$  y establezca  $j = 0$ .

Paso 1: Cálculo de la dirección de búsqueda  $\mathbf{s}_j$ .

Sea

$$\mathbf{H}_j = \begin{bmatrix} \mathbf{C} & \mathbf{A}_j^T \\ \mathbf{A}_j & 0 \end{bmatrix}.$$

Si  $\mathbf{H}_j$  no es singular, vaya al Paso 1.1. De lo contrario, vaya al Paso 1.2.

Paso 1.1:

Calcule la dirección de Newton  $\mathbf{s}_j$  y los multiplicadores  $v_j$  de la solución de las ecuaciones lineales

$$\mathbf{H}_j \begin{bmatrix} \mathbf{s}_j \\ \mathbf{v}_j \end{bmatrix} = \begin{bmatrix} \mathbf{g}_j \\ 0 \end{bmatrix}.$$

Establezca  $\gamma_j = 0$  y vaya al Paso 2.

Paso 1.2:

Calcule  $\mathbf{s}_j$  tal que  $\mathbf{A}_j \mathbf{s}_j = 0$ ,  $\mathbf{C} \mathbf{s}_j = 0$  y  $ak^T \mathbf{s}_j = 1$ . Establezca  $\gamma_j = 1$  y vaya al Paso 2.

Paso 2: Cálculo del tamaño del paso  $\sigma_j$ .

Establecer  $\tilde{\sigma} = 1$  si  $\gamma_j = 0$  y  $\tilde{\sigma} = +\infty$  si  $\gamma_j = 1$ . Si  $a_j^T s_j \geq 0$  para  $i = 1, \dots, m$  establecer  $\tilde{\sigma} = +\infty$ . De lo contrario, calcule el índice más pequeño  $l$  y  $\hat{\sigma}_j$  tal que

$$\hat{\sigma}_j = \frac{a_l^T x_j - b_l}{a_l^T s_j} = \min \left\{ \frac{a_l^T x_j - b_l}{a_l^T s_j} \mid \text{para todo } i \notin K_j \text{ con } a_i^T s_j < 0 \right\}.$$

Si  $\tilde{\sigma}_j = \hat{\sigma}_j = +\infty$  imprima el mensaje “la función objetivo es ilimitada desde abajo” y pare. Caso contrario, establezca  $\sigma_j = \min\{\tilde{\sigma}_j, \hat{\sigma}_j\}$  y vaya al Paso 3.

Paso 3: Actualizar.

Establecer  $\mathbf{x}_{j+1} = \mathbf{x}_j - \sigma_j \mathbf{s}_j$ ,  $\mathbf{g}_{j+1} = \mathbf{c} + \mathbf{C} \mathbf{x}_{j+1}$ , y  $f(\mathbf{x}_{j+1}) = \mathbf{c}^T \mathbf{x}_{j+1} + \frac{1}{2} \mathbf{x}_{j+1}^T \mathbf{C} \mathbf{x}_{j+1}$ .

Si  $\sigma_j = \hat{\sigma}_j$ , vaya al Paso 3.1. Caso contrario, vaya al Paso 3.2.

Paso 3.1:

Establezca  $K_{j+1} = K_j + l$ , forme  $\mathbf{A}_{j+1}$  y  $\mathbf{H}_{j+1}$ , reemplace  $j$  con  $j + 1$  y vaya al Paso 1.1.

Paso 3.2:

Calcule el vector multiplicador  $\mathbf{u}_{j+1}$  de  $-\mathbf{v}_j$  y  $K_j$  y calcule  $k$  tal que

$$(u_{j+1})_k = \min\{(u_{j+1})_i \mid i = 1, \dots, m\}.$$

Si  $(u_{j+1})_k \geq 0$ , luego deténgase con la solución óptima  $\mathbf{x}_{j+1}$ . Caso contrario, establezca  $K_{j+1} = K_j - k$ , forme  $\mathbf{A}_{j+1}$  y  $\mathbf{H}_{j+1}$ , reemplace  $j$  con  $j + 1$  y vaya al Paso 1.

Ilustramos el algoritmo GQPA aplicándolo un ejemplo que se encuentra en la sección de la aplicación correspondiente.

### 2.2.7. Estándar IEC-61499.

Los grandes sistemas con una inteligencia centralizada que controla toda la planta se están transformando en sistemas distribuidos. En el sistema distribuido, las partes individuales tienen inteligencia y pueden comunicarse entre sí sin problemas, entonces el sistema actúa como un todo. El estándar IEC 61499 define un lenguaje de modelado específico para desarrollar soluciones de control industrial distribuido. IEC 61499 se extiende de IEC 61131-3 al mejorar la encapsulación de componentes de software para una mayor reutilización, proporcionando un formato independiente del proveedor y simplificando el

soporte para la comunicación de controlador a controlador. Su funcionalidad de distribución y el soporte inherente para la reconfiguración dinámica proporcionan la infraestructura requerida para la industria 4.0 y las aplicaciones industriales de IoT[30].

IEC 61499 define un lenguaje de modelado que es similar a los antiguos FB, sin embargo, este nuevo lenguaje está orientado a sistemas distribuidos. Por lo tanto, permite modelar todo el sistema, incluso si está compuesto de partes más pequeñas (PICs individuales, tarjetas embebidas). Los FBs se utilizan para encapsular completamente la funcionalidad y la aplicación se crea conectando los FB individuales. Además, el estándar define un modelo para representar los dispositivos en un sistema y sus conexiones. Si la aplicación se divide en varios dispositivos, todos los FB de la aplicación pueden asignarse a sus respectivos dispositivos[30].

- Interfaz FB.

La Figura 20 muestra la nueva apariencia del FB en IEC 61499. El FB encapsula la funcionalidad deseada, las entradas están a la izquierda y las salidas a la derecha, pero a diferencia de los FB de IEC 61131-3 la interfaz diferencia eventos y datos. Los eventos activan las funcionalidades de los FB, que luego usan los datos que están disponibles en las entradas de datos. Los eventos y las conexiones de datos no son compatibles, lo que significa que no puede conectarlos de ninguna manera.

Cada entrada de evento está conectada a varias entradas de datos a través de líneas y cuadrados pequeños. Del mismo modo, cada salida de evento está conectada a varias salidas de datos. Esta conexión define qué entradas/salidas de datos se actualizan cuando se produce un evento de entrada/salida. El comportamiento del FB depende del Control de Ejecución de Eventos (ECC). El ECC es una máquina de estado, que recibe los eventos de entrada, según el estado actual[30].

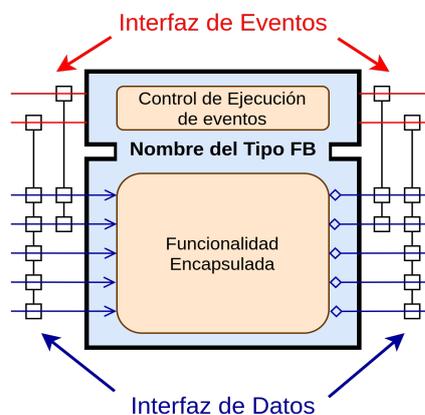


Figura 20: Bloque de Función.

- Secuencia interna de FB.

La Figura 21 indica cómo se activa un FB para ejecutar su funcionalidad. Se ejecuta una secuencia de pasos cuando llega un evento al FB[30].

1. Un evento de entrada llega al FB.
2. Las entradas de datos relacionadas con el evento entrante se actualizan.
3. El evento se pasa al ECC.
4. Dependiendo del tipo de FB y del ECC, la funcionalidad interna se activa para la ejecución.
5. La funcionalidad interna finaliza la ejecución y proporciona nuevos datos de salida.
6. Los datos de salida relacionados con el evento de salida se actualizan
7. Se envía un evento de salida.

Los pasos 4 al 7 pueden repetirse varias veces. Aunque se puede activar un evento de salida, este no tiene por qué ser el caso.

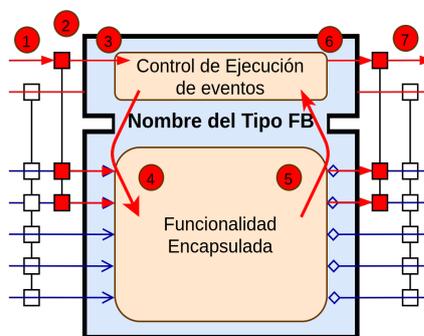


Figura 21: Bloque de Función Secuencia Interna.

- Aplicación distribuida.

Definimos ahora FBs y su comportamiento. En la Figura 22 se indica un ejemplo para una aplicación creada de acuerdo con IEC-61499. Puede ver que los datos y los eventos no están conectados entre sí[30].

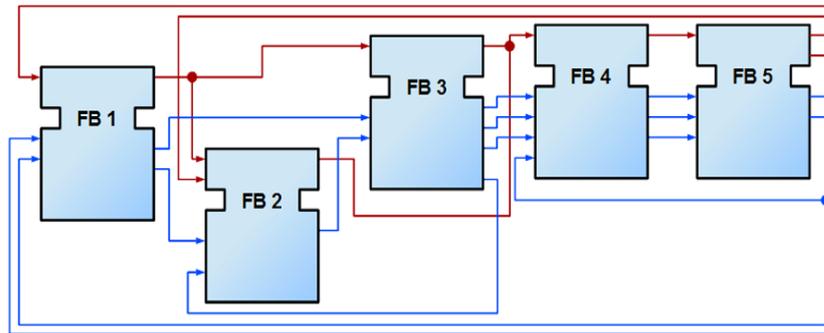


Figura 22: Aplicación Distribuida.

Como el estándar permite el modelado de sistemas distribuidos, la aplicación no necesita ejecutarse en un solo dispositivo, se puede dividir e implementar en varios dispositivos (PLC). Incluso puede tener muchas aplicaciones, que luego se distribuyen en muchos dispositivos. La vista Modelo del Sistema ayuda al usuario a diseñar esta distribución. Se ilustra en la Figura 23. Un dispositivo puede comprender varios recursos, que se pueden imaginar como hilos dentro de un dispositivo. Para ser precisos, los FBs se cargan en un recurso, no en el dispositivo en sí[30].

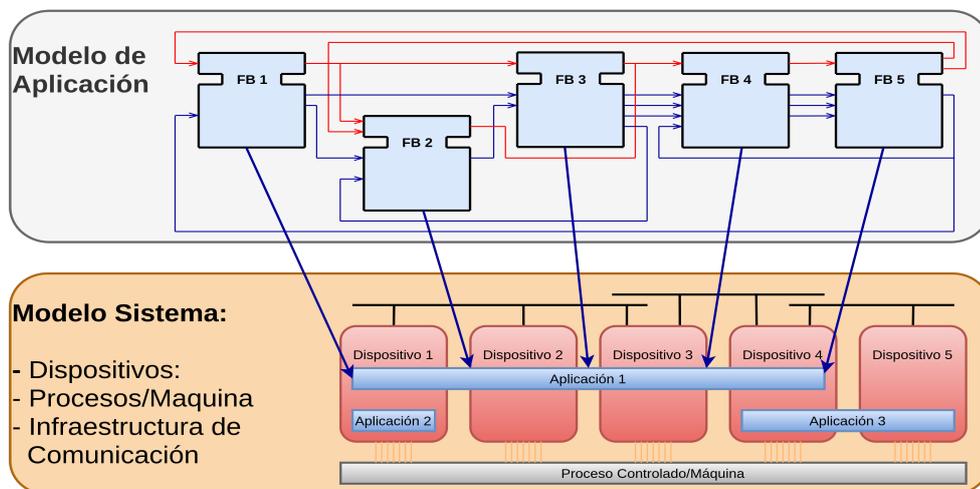


Figura 23: Aplicación Distribuida.

En la Figura 24 se observa que se conectan varios PLC y tarjetas embebidas y los bloques de funciones se distribuyen entre ellos. Por supuesto, tenemos que investigar cómo se implementan realmente las conexiones interrumpidas o cómo se accede a las entradas/salidas de hardware reales[30].

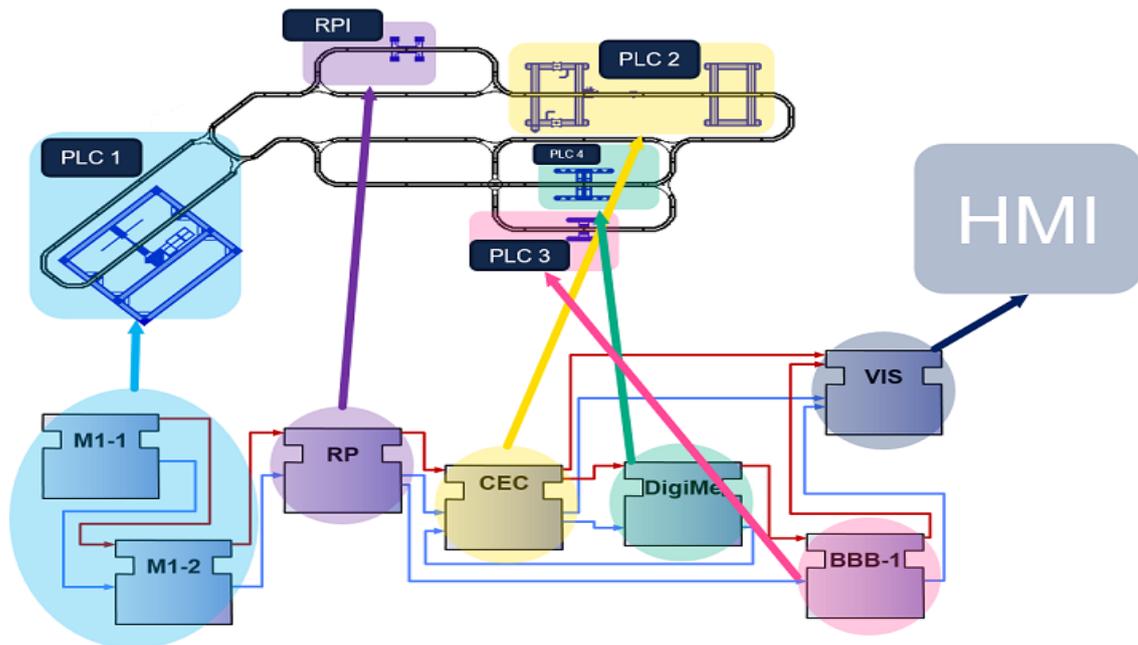


Figura 24: Modelo de Aplicación con PLC.

IEC 61499 define tres tipos de FBs que se puede usar al desarrollar su aplicación:

1. Bloque de funciones básicas (BFB).

En el BFB puede definir una máquina de estados usando el ECC, el cual decide qué algoritmo se ejecuta en función de su estado y los eventos de entrada. La Figura 25 indica un FBs con un ejemplo ECC. Los cuadros grises (Alg1-Alg3) en el ECC son los algoritmos encapsulados dentro de los bloques de funciones. Estos algoritmos son escritos por el usuario utilizando programación orientada a objetos (OOP). Los cuadros rosados (EO1,EO2) son los eventos de salida que se activan cuando se accede al estado. Las transiciones entre los estados (cuadros marrones) se muestran como flechas. Están conectados a un evento determinado, a menos que la transición se denomine "1", en cuyo caso no se necesita ningún evento y el control de ejecución salta directamente al siguiente estado. Por ejemplo, si el FB está en el estado START y llega un evento EI1, el FB salta al Estado1, ejecuta Alg1 y genera un evento EO1[30].

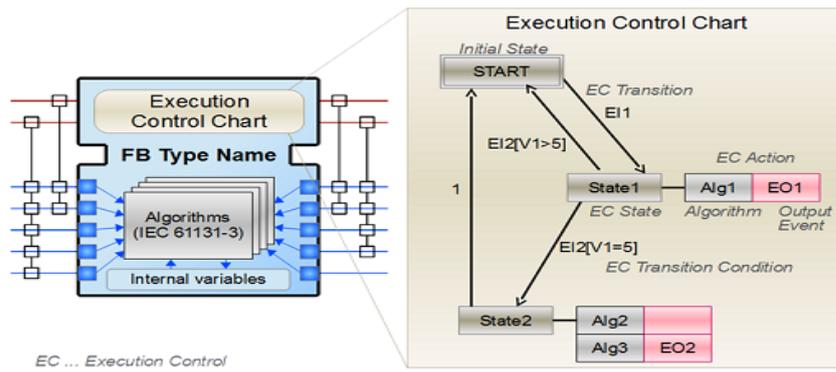


Figura 25: Bloque de Función Básico.

2. Bloque de funciones compuestas (CFB).

El CFB simplemente tiene una red interna de otros FB[30].

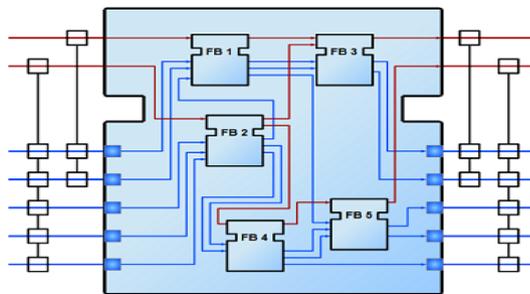


Figura 26: Bloque de Función Compuesto.

3. Bloque de funciones de servicio (SFB).

Los SFBs son FBs necesarios para acceder a partes específicas del hardware. Como se vio anteriormente, la misma aplicación se puede implementar en varios dispositivos. Para eso, se necesitan los SFBs. Estos tipos de FBs se activan no solo por un evento entrante sino también por el hardware[30].

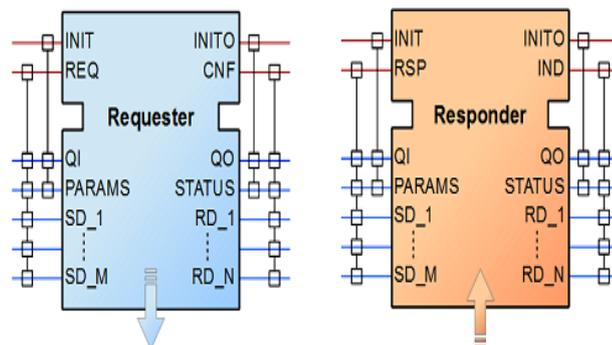


Figura 27: Bloque de Función de Servicio.

4diac IDE - Entorno de desarrollo compatible con IEC 61499.

El 4diac IDE es un entorno de ingeniería extensible, compatible con IEC 61499 para aplicaciones de control distribuido. Las aplicaciones modeladas se pueden descargar a dispositivos de campo distribuidos de acuerdo con los medios definidos por la norma IEC 61499. La definición de capacidad de hardware permite modelar el hardware de control y sus interconexiones a través de redes[30].

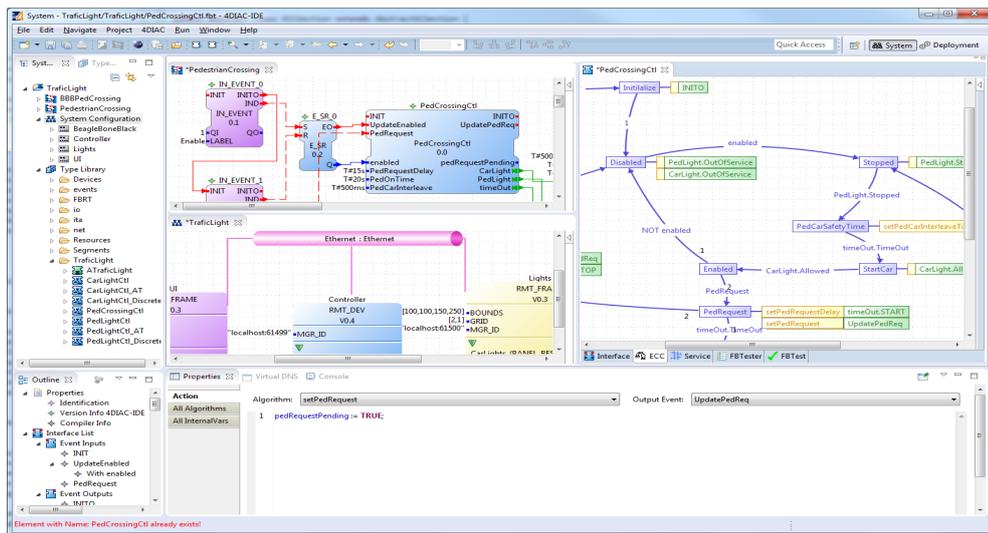


Figura 28: Entorno de desarrollo 4DIAC.

El 4diac IDE se basa en el framework de Eclipse, los sistemas basados en IEC 61499 siguen un diseño centrado en la aplicación, lo que significa que la aplicación del sistema general se crea al principio. Cada aplicación se crea interconectando los FB deseados en términos de una red de FB (FBN). Tan pronto como se conozca la estructura del hardware se puede agregar a la configuración del sistema de un proyecto y la aplicación ya existente puede distribuirse en los dispositivos disponibles[30].

FORTE 4diac - Entorno de Ejecución compatible con IEC 61499.

El 4diac FORTE es una pequeña implementación portátil de un entorno de tiempo de ejecución (runtime) IEC 61499 dirigido a pequeños dispositivos de control integrados (16/32 bits), implementado en C++. Es compatible con la reconfiguración en línea de sus aplicaciones y la capacidad de ejecución en tiempo real de todos los tipos de FB proporcionados por el estándar IEC 61499.

4diac FORTE es compatible con todos los tipos de datos, estructuras y matrices elementales IEC 61131-3 edición 2. Proporciona una arquitectura escalable que permite que 4diac FORTE se adapte a las necesidades de su aplicación. Las aplicaciones pueden consistir en cualquier elemento IEC 61499 como bloques de funciones básicas (BFB), blo-

ques de funciones compuestas (CFB), bloques de funciones de interfaz de servicio (SIFB), adaptadores y subaplicaciones. 4diac FORTE proporciona una infraestructura de comunicación flexible a través de las llamadas capas de comunicación[30].

### 2.2.8. OPC Arquitectura unificada.

OPC es el estándar de interoperabilidad para el intercambio seguro y confiable de datos en el espacio de automatización industrial y en otras industrias, garantiza el flujo continuo de información entre dispositivos de múltiples proveedores.

El estándar OPC es una serie de especificaciones desarrolladas por proveedores de la industria, usuarios finales y desarrolladores de software. Éstas especificaciones definen la interfaz entre clientes y servidores, así como servidores y servidores, incluido el acceso a datos en tiempo real, monitoreo de alarmas y eventos, acceso a datos históricos y otras aplicaciones.

Cuando el estándar se lanzó por primera vez en 1996, su propósito era abstraer protocolos específicos de PLC (como Modbus, Profibus, etc.) en una interfaz estandarizada que permite que los sistemas HMI/SCADA se interconecten con un "intermediario" quién convertiría las solicitudes de lectura/escritura genéricas-OPC en solicitudes específicas del dispositivo y viceversa. Como resultado, surgió toda una industria artesanal de productos permitiendo a los usuarios finales implementar sistemas, todos interactuando sin problemas a través de OPC[31].

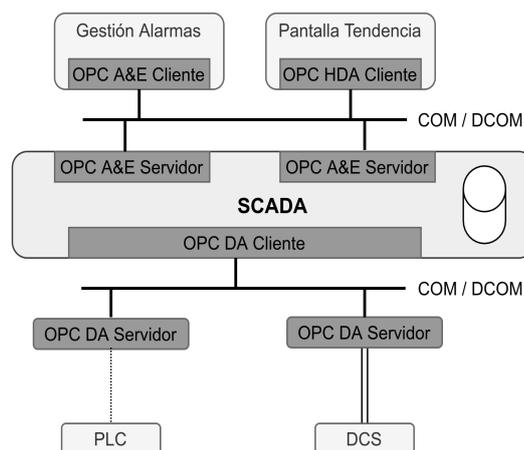


Figura 29: Arquitectura OPC UA.

Con la introducción de arquitecturas orientadas a servicios en los sistemas de fabricación, surgieron nuevos desafíos en seguridad y modelado de datos. La Fundación OPC desarrolló las especificaciones OPC UA para abordar estas necesidades y al mismo tiempo

proporcionó un arquitectura de plataforma abierta de tecnología rica en características[31], en la Figura 30 se indica el escenario de una aplicación con OPC UA .

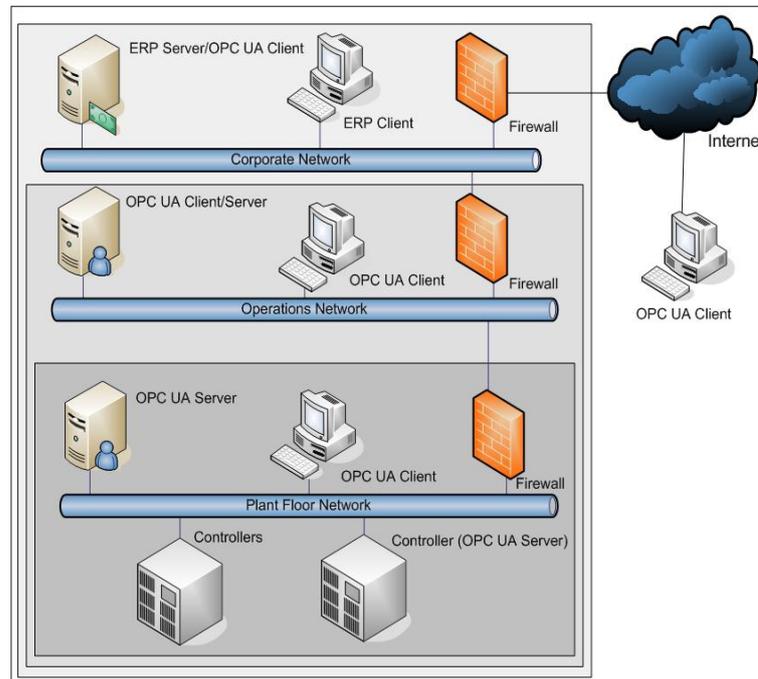


Figura 30: Arquitectura OPC UA.

La arquitectura unificada OPC UA lanzada en 2008, es una arquitectura orientada a servicios independiente de la plataforma que integra toda la funcionalidad de las especificaciones individuales de OPC Clásico en un framework extensible.

Este enfoque de múltiples capas logra los objetivos de la especificación del diseño original de[32]:

- Equivalencia funcional: todas las especificaciones COM OPC Clásico se asignan a OPC UA.
- Independencia de la plataforma: desde un microcontrolador embebido hasta una infraestructura basada en la nube.
- Seguro: cifrado, autenticación y auditoría.
- Extensible: capacidad de agregar nuevas funciones sin afectar las aplicaciones existentes.
- Modelado integral de información: para definir información compleja.

## **3. CAPÍTULO III**

### **MARCO METODOLÓGICO**

#### **3.1. Ubicación.**

La implementación del Control Predictivo Generalizado con restricciones basado en la normativa IEC 61499 fue realizado en el proceso de nivel de la célula analógica FESTO MPS PA® Compact Workstation de los laboratorios de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial de la Universidad Técnica de Ambato.

#### **3.2. Equipos y materiales.**

Hardware:

- Célula analógica FESTO MPS PA® Compact Workstation.
- Tarjeta de desarrollo Beagle BoneBlack.
- Estacion ingeniería (PC).
- Componentes electrónicos (Circuito de acople y conexión).
- Arduino UNO.

Software:

- 4DIAC IDE.
- 4DIAC FORTE.
- Matlab.
- Arduino IDE.
- Mathematica.
- RStudio.

### **3.3. Tipo de investigación.**

#### **3.3.1. Investigación bibliográfica.**

La búsqueda bibliográfica es la primera etapa del proceso investigativo y proporciona el conocimiento ya existente, a través de una amplia búsqueda de: información, conocimientos y técnicas sobre una cuestión determinada. La presente investigación tendrá esta modalidad debido a que se realizará una búsqueda de información escrita y un análisis de la misma sobre el tema recurriendo a fuentes como libros, revistas, artículos científicos, tesis doctorales y en la web, ayudando así al cumplimiento de los objetivos planteados.

#### **3.3.2. Investigación aplicada.**

Es de carácter aplicativo ya que busca la generación de conocimiento con aplicación directa a un específico sector tecnológico. Además busca la utilización de conocimientos de varias áreas especializadas de la ingeniería como las matemáticas, electrónica y la programación, con el propósito de implementarlos de forma práctica para desarrollar un sistema de control predictivo industrial distribuido.

### **3.4. Idea a defender.**

Es factible implementar un sistema de control predictivo generalizado de nivel con restricciones en una célula analógica FESTO MPS PA® Compact Workstation, mediante modelamiento de Bloques de Funciones basados en la normativa IEC 61499 para el desarrollo de control distribuido acorde a la industria 4.0.

### **3.5. Población o muestra.**

Por las características de la presente investigación no se requiere población y muestra.

### **3.6. Recolección de información.**

Para el presente trabajo la recolección de información se realizó mediante la selección de información relevante, obtenida de fuentes como: libros, revistas, artículos científicos, tesis doctorales y en la web mismos que aportarán con el material para el desarrollo de la investigación, además se recolectó información mediante una prueba experimental.

Los libros e investigaciones relacionadas con el tema propuesto se encontraron en la página web Google Académico (<https://scholar.google.es/schhp?hl=es>), los cuales fueron

descargados en las páginas web Library Genesis (<http://libgen.li/>) y sci-hub (<https://sci-hub.se/>) respectivamente.

El software compatible con la normativa IEC-61499 fue descargado en la página web 4diac (<https://www.eclipse.org/4diac/>), de igual manera el sistema operativo para la BBB fue descargado de la página web beagleboard.org (<https://beagleboard.org/latest-images>).

Varias librerías fueron utilizadas para el desarrollo del sistema de control predictivo distribuido implementado, la librería para manipulación de pines de la BBB fue descargada de la página web Black Lib (<http://blacklib.yigityuce.com/>), de igual manera la librería para el desarrollo del servidor y cliente OPC UA en C++ fue descargada de la página web open62541 (<https://open62541.org/>) y para el cliente OPC UA en python fue descargado de la página web (<https://github.com/FreeOpcUa>), finalmente la librería para realizar la optimización cuadrática fue descargada de la página web (<https://github.com/liuq/QuadProgpp>)

### **3.7. Procesamiento de la información y análisis estadístico.**

El modelo de regresión exponencial fue utilizado como herramienta para encontrar la ecuación de respuesta al escalón del sistema, de igual manera se aplicaron estadísticos de prueba para aceptar o rechazar hipótesis nulas referentes a los parámetros encontrados de la regresión exponencial implementada.

### **3.8. Resultados alcanzados.**

Se logró la implementación del Controlador Predictivo Generalizado con tres tipos de restricciones, en el proceso de nivel de la planta analógica FESTO MPS PA® Compact Workstation, mediante modelamiento de Bloques de Funciones basados en la normativa IEC 61499 para el desarrollo de control distribuido acorde a la industria 4.0.

## **4. CAPÍTULO IV**

### **RESULTADOS Y DISCUSIÓN**

#### **4.1. Análisis de resultados.**

##### **4.1.1. Desarrollo de la propuesta.**

Actualmente la necesidad de automatizar los procesos industriales utilizando sistemas de control distribuidos con el fin de obtener una mayor eficiencia en la producción y servicios ha sido motivo para llevar a cabo el presente proyecto de investigación. Se implementó un Controlador Predictivo Generalizado basado el estándar IEC-61499, la cual busca conseguir un alto grado de control distribuido y a la vez normalizar el desarrollo de código de manera que éste sea independiente de la máquina sobre la que se ejecuta.

Para la implementación del sistema de control industrial se seleccionó la maqueta analógica FESTO® MPS-PA Compact Workstation representada en la Figura 31 del laboratorio de hidráulica y neumática de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial de la Universidad Técnica de Ambato, debido a los diferentes procesos existentes en la misma, ideales para implementar una gran variedad de controladores.

La célula analógica FESTO® MPS-PA Compact Workstation dispone de procesos didácticos para realizar controladores de nivel, caudal, presión y temperatura. El diseño de los sensores y actuadores permite el uso de controladores continuos (P, PI, PID) y discontinuos (controladores de dos puntos)[33]. El diseño básico de la célula FESTO para la implementación del controlador se presenta a continuación:

- Componentes mecánicos: 2 depósitos de líquido, sistema de tubos, válvula de bola para regular.
- Sensores: sensor de ultrasonido de nivel.
- Actuadores: bomba de agua.



Figura 31: Célula analógica FESTO MPS PA® Compact Workstation.

En el diagrama PI&D de la Figura 32 se indica los componentes relacionados al proceso de control de nivel, donde la bomba P101 suministra un fluido desde un depósito de almacenamiento B101 a un depósito B102 a través de un sistema de tuberías. El nivel del fluido dentro del tanque B102 se monitoriza con un sensor analógico ultrasónico B101 en el punto de medición LIC B101 y se lee como valor real. El valor debe mantenerse en un cierto nivel incluso cuando se producen perturbaciones o cambios en el punto de consigna o setpoint. Para realizar perturbaciones se puede abrir o cerrar parcial o totalmente la válvula de bola V110.

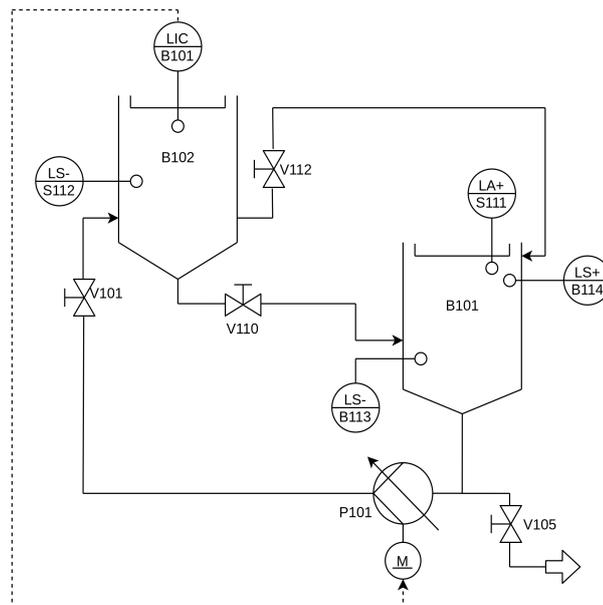


Figura 32: Diagrama PI&D de nivel.

Implementado el controlador predictivo, se tiene como resultado la configuración que se detalla en la Figura 33, posteriormente se indicará la función de cada uno de los componentes (software y hardware) del sistema.

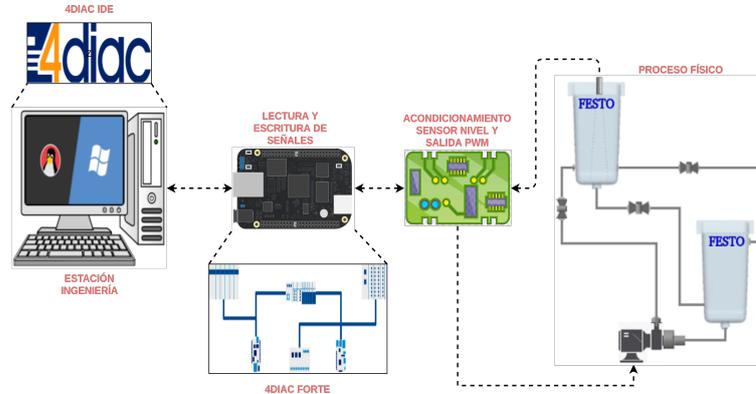


Figura 33: Conexión física del controlador predictivo.

#### 4.1.2. Modelo matemático de planta de nivel.

El modelo matemático para el proceso de nivel de la célula analógica FESTO® MPS-PA Compact Workstation se representa en la ecuación (5) y se encuentra en la fundamentación teórica del Capítulo II, pero necesitamos obtener una expresión del modelo matemático linealizado, así como la función de transferencia para el sistema de nivel definido por la ecuación (5) que representa al sistema de la Figura 34:

$$A \frac{dh}{dt} = q_{in} - aC_q \sqrt{2gh}$$

El término que hace no lineal a la ecuación (5) es el factor

$$q_0 = aC_q \sqrt{2gh} \quad \text{o bien,} \quad q_0 = K\sqrt{h} \quad (69)$$

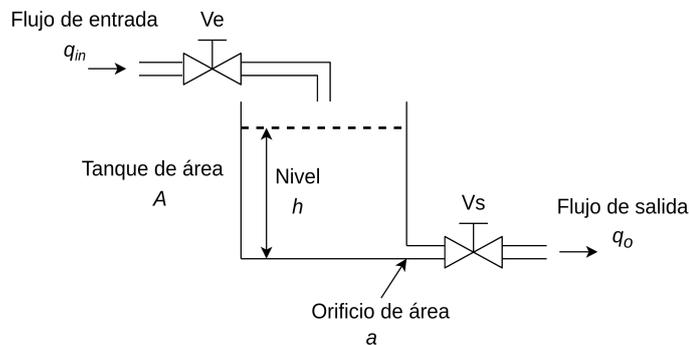


Figura 34: Sistema de nivel representado por la ecuación (5).

por lo que se linealizará alrededor del punto de operación  $h_0$ .

Al describir la ecuación (9) en términos de las variables propias del sistema hidráulico tenemos:

$$q_0 \approx q(h_0) + \left. \frac{dq}{dt} \right|_{h=h_0} (h - h_0) \quad (70)$$

La derivada de  $q_0$  de la ecuación (70) evaluada en  $h = h_0$  es:

$$\left. \frac{dq_0}{dh} \right|_{h=h_0} = \left. \frac{1}{2} K h^{-1/2} \right|_{h=h_0} = \frac{K}{2\sqrt{h_0}} = \frac{1}{R_h} \quad (71)$$

donde  $R_h =$  Resistencia hidráulica, por lo que (71) se convierte en

$$q_0 \approx q(h_0) + \frac{1}{R_h} (h - h_0) \quad (72)$$

Si se sustituye (73) en (5):

$$A \frac{dh}{dt} = q_{in} - \left[ q(h_0) + \frac{1}{R} (h - h_0) \right]$$

y se reagrupan términos, obtenemos:

$$A \frac{dh}{dt} + \frac{1}{R} (h - h_0) = q_{in} - q(h_0) \quad (73)$$

Las variables de desviación (diferencia entre el valor de la variable y su valor en estado estable) se utilizará para obtener la representación final linealizada del sistema.

Al introducir las variables de desviación  $Q$  y  $H$ :

$$Q = q_i - q(h_0) \quad y \quad H = h - h_0$$

además  $A \frac{d(h-h_0)}{dt} = A \frac{dH}{dt}$ , ya que  $h_0$  es constante, la ecuación(74) se convierte en:

$$A \frac{dH}{dt} + \frac{1}{R} H = Q \quad (74)$$

donde

$$q_0 = \frac{h}{R} \quad (75)$$

La ecuación (75) es una expresión linealizada, razón por la cual puede transformarse al dominio  $s$ :

$$sH(s) + \frac{1}{AR} H(s) = \frac{1}{A} Q(s)$$

$$G(s) = \frac{H(s)}{Q(s)} = \frac{\frac{1}{A}}{s + \frac{1}{AR}} \quad (76)$$

Verificado que el proceso de nivel de la célula analógica FESTO® MPS-PA Compact Workstation es un sistema de primer orden, procedemos a encontrar la respectiva función de transferencia mediante un experimento de entrada al escalón aplicado al proceso de nivel. En la Figura 35 se representa el diagrama del experimento.

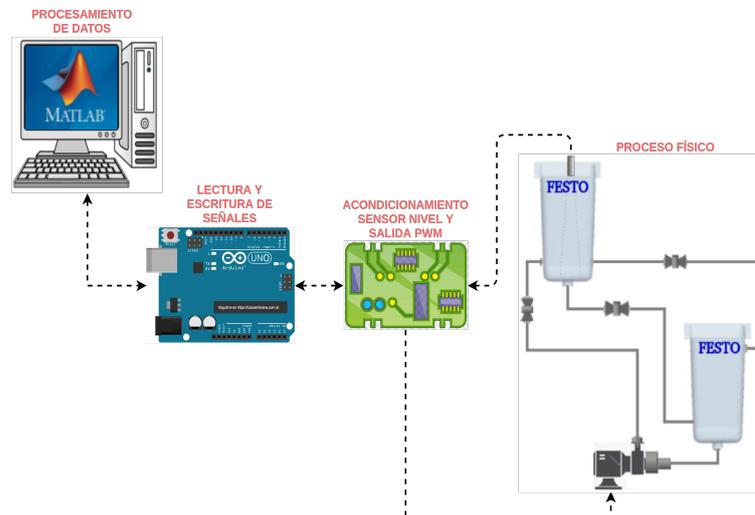


Figura 35: Diagrama experimento respuesta al escalón.

Mediante la ecuación de respuesta del sistema (14) y aplicando una entrada  $r(t) = 1U(t)$  se obtiene que  $A = 1$ , del mismo modo se encontraron los valores de  $K$  y  $\tau$  mediante resultados experimentales para lo cual se aplicó una entrada escalón de valor  $A$  al sistema y se recolectaron los datos.

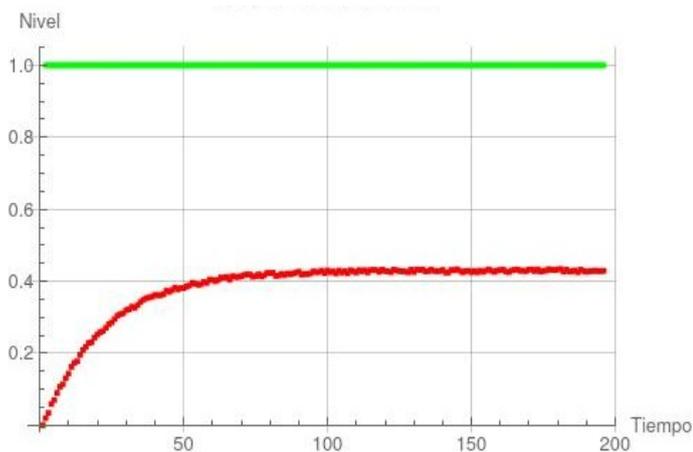


Figura 36: Respuesta del sistema al escalón unitario.

Los datos se obtuvieron experimentalmente con ayuda de un microcontrolador Arduino y el software de programación Matlab, los cuales están representados en la Tabla

7 de los anexos y se los puede visualizar en la Figura 37, donde se indica las variables *Tiempo* y *Nivel* alcanzados.

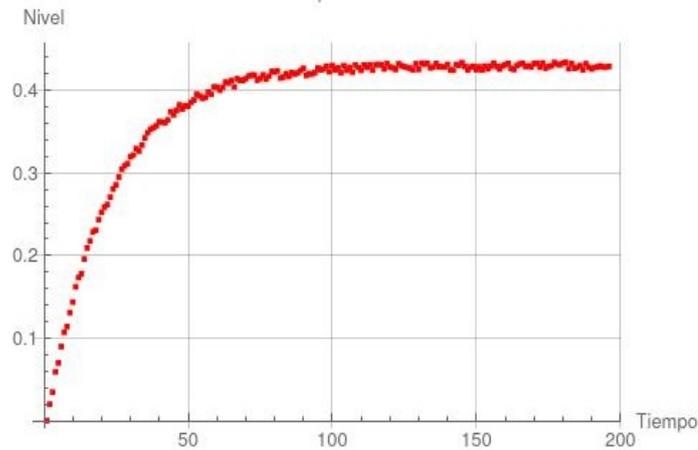


Figura 37: Datos a la respuesta al escalón.

El objetivo fue encontrar la curva que mejor se ajusta a los datos experimentales, mediante el método de mínimos cuadrados obtendremos los valores de  $K$  y  $\tau$ . Considerando la ecuación (14) correspondiente a un sistema de primer orden y reemplazándola en la ecuación de mínimos cuadrados (19) obtenemos

$$\hat{Nivel}(t) = AK(1 - e^{-\frac{t}{\tau}})$$

$$f(K, \tau) = \sum_{i=1}^n (Nivel_i - \hat{Nivel}_i)^2$$

además, reemplazamos los valores experimentales  $T$ ,  $Nivel$  y el valor del escalón  $A$ , obteniendo así una función cuadrática que se encuentra en función de las variables  $K$  y  $\tau$ :

$$f(K, \tau) = \sum_{i=1}^n (Nivel_i - AK(1 - e^{-\frac{t}{\tau}}))^2$$

$$f(K, \tau) = \sum_{i=1}^{195} (Nivel_i - 1K(1 - e^{-\frac{T_i}{\tau}}))^2$$

El propósito es encontrar el valor mínimo de esta función cuadrática, por tal motivo procedemos derivar parcialmente con respecto a cada una de las variables e igualarlas a cero.

$$\frac{\partial f(K, \tau)}{\partial k} = 0$$

$$\frac{\partial f(K, \tau)}{\partial \tau} = 0$$

Se obtiene dos ecuaciones con dos incógnitas y se procede a resolver el sistema de ecuaciones, logrando los valores de:

$$K = 0,429973 \quad \text{y} \quad \tau = 21,6245$$

Finalmente estos valores son reemplazados en (14) obteniendo la ecuación de respuesta del sistema y su gráfica se representa en la Figura 38, de igual manera reemplazando los valores de  $K$  y  $\tau$  en (12) obtenemos la función de transferencia.

$$y(t) = 0,429973(1 - e^{-\frac{t}{21,6245}})$$

$$G(s) = \frac{0,429973}{21,6245s + 1}$$

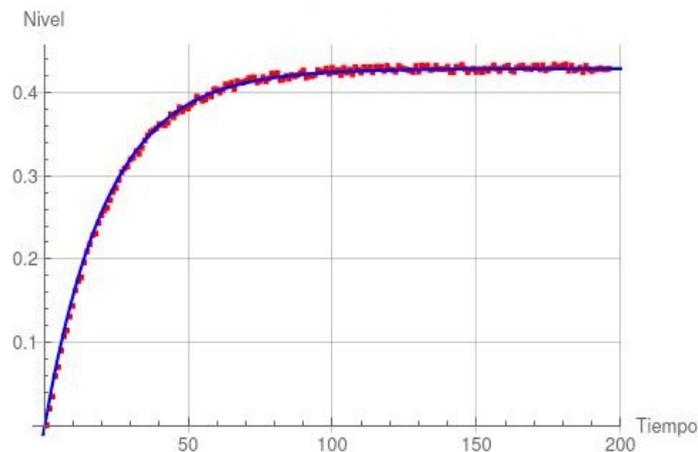


Figura 38: Gráfica aproximada a los datos recolectados.

El algoritmo del sistema de control fue implementado en un computador embebido, por tal motivo en la implementación del controlador predictivo se trabajó con ecuaciones en tiempo discreto.

Considerando la función de transferencia continua del sistema

$$G(s) = \frac{0,42}{21,62s + 1} = \frac{0,0194}{s + 0,0463}$$

procedemos a separar términos, obteniendo:

$$\frac{G(s)}{s} = \frac{0,0194}{s(s + 0,0463)} = \frac{0,419}{s} - \frac{0,419}{s + 0,0463}$$

$$\frac{G(s)}{s} = \frac{0,0769}{s(s + 0,1923)} = \frac{0,4}{s} - \frac{0,4}{s + 0,1923}$$

aplicamos la fórmula de la ecuación (36)

$$G(z) = (1 - z^{-1})\mathcal{Z}\left\{\frac{G(s)}{s}\right\}$$

la función correspondiente en tiempo continuo es:

$$\mathcal{L}^{-1}\left\{\frac{G(s)}{s}\right\} = 0,419(1(t)) - 0,419(e^{-0,0463t}1(t))$$

además considerando que el tiempo de muestreo fue cada segundo se tiene  $0,419(1(kT)) - 0,419(e^{-0,0463t}1(kT))$  y la transformada  $z$  es:

$$\mathcal{Z}\left\{\frac{G(s)}{s}\right\} = 0,419\left(\frac{z}{z-1}\right) - 0,419\left(\frac{z}{z - e^{-0,0463*0,4}}\right)$$

$$\mathcal{Z}\left\{\frac{G(s)}{s}\right\} = z\left(\frac{0,0076}{(z-1)(z-0,9817)}\right)$$

finalmente calculamos la transformación deseada como:

$$G(z) = \frac{z-1}{z} \frac{z(0,0076)}{(z-1)(z-0,9817)}$$

$$G(z) = \frac{0,0076}{z-0,9817}$$

En la Figura 39 se representa la respuesta al escalón unitario de la función de transferencia en tiempo continuo y tiempo discreto

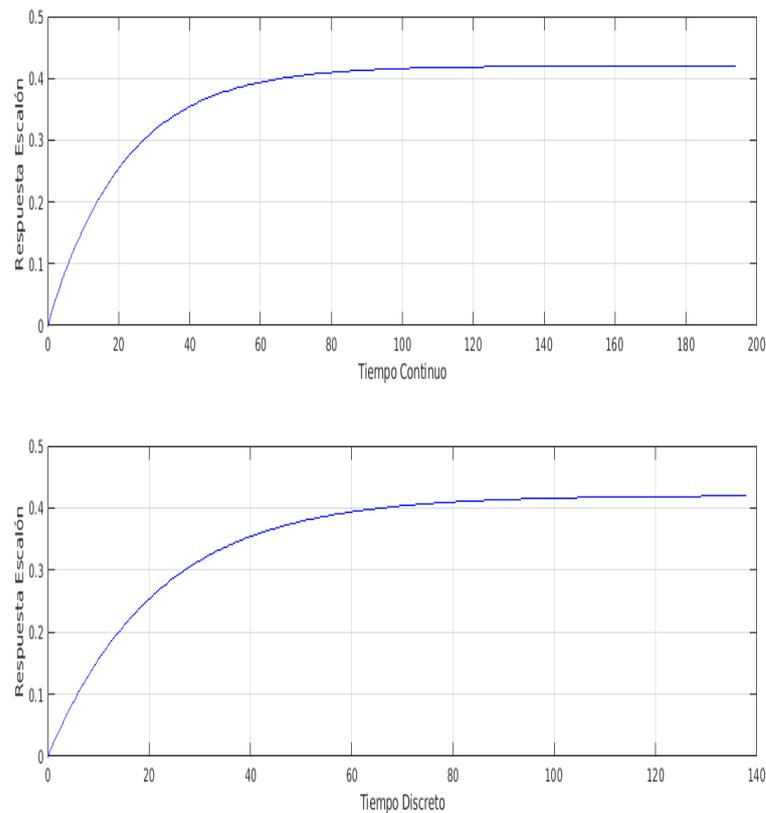


Figura 39: Respuesta al escalón en tiempo continuo y discreto.

### 4.1.3. Implementación del control GPC.

Para demostrar la implementación del Controlador Predictivo Generalizado con y sin restricciones para el caso de estudio del sistema de primer orden de la planta de nivel se obtuvo la siguiente equivalencia discreta.

$$(1 + az^{-1})y(t) = (b_0 + b_1z^{-1})u(t - 1) + \frac{e(t)}{\Delta}$$

En este ejemplo, el retardo  $d$  es igual a 0 y el polinomio de ruido  $C(z^{-1})$  se considera igual a 1.

Para la implementación del algoritmo de control del GPC descrita en el Capítulo II, se tiene los resultados numéricos para los parámetros  $a = -0,9817$  y  $b_0 = 0,0076$ , siendo los horizontes  $N_1 = 1$  y  $N = N_u = 3$ . Como se ha demostrado los valores predichos de salida del proceso en el primer horizonte se calcularon y se reescribieron en forma de la ecuación (45) y luego la ley de control se calculó usando la expresión (49) para el GPC sin restricciones y con el algoritmo GQPA para el GPC con restricciones, donde se eligió un instante  $k$  para la demostración del algoritmo, las restricciones implementadas se indican a continuación:

- Restricción en el incremento de control.  $(\Delta u(t))$ .

$$\begin{bmatrix} \mathbf{I} \\ \dots \\ -\mathbf{I} \end{bmatrix} \Delta \mathbf{u} \leq \begin{bmatrix} u_{min} \mathbf{1} \\ \dots \\ -u_{min} \mathbf{1} \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} \Delta u_1 \\ \Delta u_2 \\ \Delta u_3 \\ \Delta u_1 \\ \Delta u_2 \\ \Delta u_3 \end{bmatrix} \leq \begin{bmatrix} 0,1 \\ 0,1 \\ 0,1 \\ -0,1 \\ -0,1 \\ -0,1 \end{bmatrix}$$

- Restricción en la señal de control  $(u(t))$

$$\begin{bmatrix} \mathbf{T} \\ \dots \\ -\mathbf{T} \end{bmatrix} \Delta \mathbf{u} \leq \begin{bmatrix} \mathbf{1}u_{max} - \mathbf{1}u(k - 1) \\ \dots \\ -\mathbf{1}u_{min} + \mathbf{1}u(k - 1) \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ -1 & 0 & 0 \\ -1 & -1 & 0 \\ -1 & -1 & -1 \end{bmatrix} \begin{bmatrix} \Delta u_1 \\ \Delta u_2 \\ \Delta u_3 \\ \Delta u_1 \\ \Delta u_2 \\ \Delta u_3 \end{bmatrix} \leq \begin{bmatrix} 13,5000 - 0,5001 \\ 13,5000 - 0,5001 \\ 13,5000 - 0,5001 \\ -0 + 0,5001 \\ -0 + 0,5001 \\ -0 + 0,5001 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ -1 & 0 & 0 \\ -1 & -1 & 0 \\ -1 & -1 & -1 \end{bmatrix} \begin{bmatrix} \Delta u_1 \\ \Delta u_2 \\ \Delta u_3 \\ \Delta u_1 \\ \Delta u_2 \\ \Delta u_3 \end{bmatrix} \leq \begin{bmatrix} 12,9999 \\ 12,9999 \\ 12,9999 \\ 0,5001 \\ 0,5001 \\ 10,5001 \end{bmatrix}$$

- Restricciones en la señal de salida ( $y(t)$ ).

$$\begin{bmatrix} \mathbf{G} \\ \dots \\ -\mathbf{G} \end{bmatrix} \Delta \mathbf{u} \leq \begin{bmatrix} \mathbf{1}y_{max} - \mathbf{f} \\ \dots \\ -\mathbf{1}y_{min} + \mathbf{f} \end{bmatrix}$$

$$\begin{bmatrix} 0,0077 & 0,0000 & 0,0000 \\ 0,0153 & 0,0077 & 0,0000 \\ 0,0227 & 0,0153 & 0,0077 \\ -0,0077 & -0,0000 & -0,0000 \\ -0,0153 & -0,0077 & -0,0000 \\ -0,0227 & -0,0153 & -0,0077 \end{bmatrix} \begin{bmatrix} \Delta u_1 \\ \Delta u_2 \\ \Delta u_3 \\ \Delta u_1 \\ \Delta u_2 \\ \Delta u_3 \end{bmatrix} \leq \begin{bmatrix} 5,5000 - 0,0149 \\ 5,5000 - 0,0185 \\ 5,5000 - 0,0220 \\ -0 + 0,0149 \\ -0 + 0,0185 \\ -0 + 0,0220 \end{bmatrix}$$

$$\begin{bmatrix} 0,0077 & 0,0000 & 0,0000 \\ 0,0153 & 0,0077 & 0,0000 \\ 0,0227 & 0,0153 & 0,0077 \\ -0,0077 & -0,0000 & -0,0000 \\ -0,0153 & -0,0077 & -0,0000 \\ -0,0227 & -0,0153 & -0,0077 \end{bmatrix} \begin{bmatrix} \Delta u_1 \\ \Delta u_2 \\ \Delta u_3 \\ \Delta u_1 \\ \Delta u_2 \\ \Delta u_3 \end{bmatrix} \leq \begin{bmatrix} 5,4851 \\ 5,4815 \\ 5,4780 \\ 0,0149 \\ 0,0185 \\ 0,0220 \end{bmatrix}$$

El procedimiento para la implementación del algoritmo del Controlador Predictivo Generalizado con y sin restricciones consta de cuatro pasos, los cuales se indican a continuación:

1. Salida de la planta  $y(t)$ .

Considerando función de transferencia en tiempo discreto obtenemos los valores de  $a = -0,9817$  y  $b_0 = 0,0076$ , por lo tanto nuestra ecuación en diferencia queda representada por:

$$y(k) = b_0 u(k-1) - ay(k-1)$$

$$y(k) = 0,0076u(k-1) + 0,9817y(k-1)$$

donde

$y(k)$  es la salida real de la planta en un instante discreto  $k$ .

$u(k-1)$  es la señal de control en un instante  $k-1$ .

$y(k-1)$  es la salida real de la planta en un instante discreto  $k-1$ .

2. Coeficientes al escalón, respuesta libre y predicción.

Los polinomios predictores  $E_j(z^{-1})$ ,  $F_j(z^{-1})$  desde  $j = 1$  hasta  $j = 3$  se calcularon resolviendo la ecuación diofántica (42), con

$$\tilde{A}(z^{-1}) = A(z^{-1})(1 - z^{-1}) = 1 + 1,9817z^{-1} + 2,9454z^{-2}$$

En éste caso donde el horizonte no es demasiado grande, los polinomios se pueden obtener directamente dividiendo 1 por  $\tilde{A}(z^{-1})$ .

$$\begin{array}{r|l} +1,0 & \left| \begin{array}{l} +1,0 - 1,9817z^{-1} + 0,9817z^{-2} \\ +1 + 1,9817z^{-1} + 2,9454z^{-2} \end{array} \right. \\ \hline -1,0 \quad +1,9817z^{-1} \quad -0,9817z^{-2} & \\ \hline \quad +1,9817z^{-1} \quad -0,9817z^{-2} & \\ \hline \quad -1,9817z^{-1} \quad +3,9271z^{-2} \quad -1,9454z^{-3} & \\ \hline \quad \quad +2,9454z^{-2} \quad -1,9454z^{-3} & \\ \hline \quad \quad -2,9454z^{-2} \quad 5,8369z^{-3} \quad -2,8915z^{-4} & \\ \hline \quad \quad \quad +3,8915z^{-3} \quad -2,8915z^{-4} & \end{array}$$

Los valores obtenidos son:

$$\begin{array}{ll} E_1 = 1,000 & F_1 = 1,9817z^{-1} - 0,9817z^{-2} \\ E_2 = 1,000 + 1,9817z^{-1} & F_2 = 2,9454z^{-2} - 1,9454z^{-3} \\ E_3 = 1,000 + 1,9817z^{-1} + 2,9454z^{-2} & F_3 = 3,8915z^{-3} - 2,8915z^{-4} \end{array}$$

con estos valores y el polinomio  $B(z^{-1}) = 0,0076$ , los valores de  $G_i(z^{-1})$  son:

$$G_1 = 0,0077$$

$$G_2 = 0,0077 + 0,0153z^{-1}$$

$$G_3 = 0,0077 + 0,0153z^{-1} + 0,0227z^{-2}$$

por lo que las salidas predichas se pueden escribir como:

$$\hat{\mathbf{y}} = \mathbf{G}\mathbf{u} + \mathbf{F}(z^{-1})y(k)$$

$$\hat{\mathbf{y}} = \mathbf{G}\mathbf{u} + \mathbf{f}$$

donde

$$\mathbf{G} = \begin{bmatrix} 0,0077 & 0,0000 & 0,0000 \\ 0,0153 & 0,0077 & 0,0000 \\ 0,0227 & 0,0153 & 0,0077 \end{bmatrix}$$

$$\begin{bmatrix} \hat{y}(t+1|t) \\ \hat{y}(t+2|t) \\ \hat{y}(t+3|t) \end{bmatrix} = \begin{bmatrix} 0,0077 & 0,0000 & 0,0000 \\ 0,0153 & 0,0077 & 0,0000 \\ 0,0227 & 0,0153 & 0,0077 \end{bmatrix} \begin{bmatrix} \Delta u(t) \\ \Delta u(t+1) \\ \Delta u(t+2) \end{bmatrix} + \begin{bmatrix} 0,0077\Delta u(t-1) + 1,9817y(t) - 0,9817y(t-1) \\ 0,0153\Delta u(t-1) + 2,9454y(t) - 1,9454y(t-1) \\ 0,0227\Delta u(t-1) + 3,8915y(t) - 2,8915y(t-1) \end{bmatrix}$$

### 3. Incremento de control $\Delta u(t)$ .

El siguiente paso es calcular  $\mathbf{H}^{-1}\mathbf{b}$ . Si  $\delta = 2$  y  $\lambda = 5$

$$(\mathbf{G}^T \delta \mathbf{I} \mathbf{G} + \lambda \mathbf{I})^{-1} \mathbf{G}^T \delta \mathbf{I} = \begin{bmatrix} 0,0031 & 0,0061 & 0,0091 \\ 0,0000 & 0,0031 & 0,0061 \\ 0,0000 & 0,0000 & 0,0031 \end{bmatrix}$$

Es necesario calcular  $\Delta u(t)$ , por lo que solo la primera fila de la matriz es utilizada

$$\mathbf{K} = [0,0031 \quad 0,0061 \quad 0,0091]$$

obteniendo la siguiente expresión para la ley de control:

$$\Delta u(t) = [0,0031 \quad 0,0061 \quad 0,0091] \begin{bmatrix} w(t+1) - 1,9817y(t) + 0,9817y(t-1) \\ w(t+2) - 2,9454y(t) + 1,9454y(t-1) \\ w(t+3) - 3,8915y(t) + 2,8915y(t-1) \end{bmatrix}$$

$$\Delta u(t) = -0,0595y(t)+0,0412y(t-1)+0,0031w(t+1)+0,0061w(t+2)+0,0091w(t+3)$$

donde  $w(t+i)$  es la trayectoria de referencia que puede considerarse constante e igual al setpoint actual.

Para encontrar el  $\Delta u(t)$  del GPC sin restricciones procedemos a resolver

$$\Delta u(t) = \mathbf{K}(\mathbf{w} - \mathbf{f})$$

los vectores  $\mathbf{f}$  y  $\mathbf{w}$  serán calculados cada iteración por lo tanto se eligió un instante  $k$  para demostrar el funcionamiento del algoritmo de control y optimización.

$$\mathbf{f} = \begin{bmatrix} 0,0149 \\ 0,0185 \\ 0,0220 \end{bmatrix} \quad \text{y} \quad \mathbf{w} = \begin{bmatrix} 11,0000 \\ 11,0000 \\ 11,0000 \end{bmatrix}$$

por lo tanto se tiene un incremento de control para el instante  $k$  de:

$$\Delta u(t) = \begin{bmatrix} 0,0031 & 0,0061 & 0,0091 \end{bmatrix} \left( \begin{bmatrix} 11,0000 \\ 11,0000 \\ 11,0000 \end{bmatrix} - \begin{bmatrix} 0,0149 \\ 0,0185 \\ 0,0220 \end{bmatrix} \right)$$

$$\Delta u(t) = 0,2003$$

Para encontrar el  $\Delta u(t)$  del GPC con restricciones partimos de la ecuación vectorial de la función objetivo a minimizar  $J(\Delta \mathbf{u}) = \frac{1}{2} \Delta \mathbf{u}^T \mathbf{H} \Delta \mathbf{u} + \mathbf{b}^T \Delta \mathbf{u}$  representada como:

$$J(\Delta \mathbf{u}) = \frac{1}{2} \Delta \mathbf{u}^T (\mathbf{G}^T \delta \mathbf{I} \mathbf{G} + \lambda \mathbf{I}) \Delta \mathbf{u} + ((\mathbf{f} - \mathbf{w})^T \delta \mathbf{I} \mathbf{G})^T \Delta \mathbf{u}$$

reemplazamos valores para encontrar la matriz  $\mathbf{H}$

donde:

$$\mathbf{H} = \mathbf{G}^T \delta \mathbf{I} \mathbf{G} + \lambda \mathbf{I}$$

$$\delta \mathbf{I} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix} \quad \text{y} \quad \lambda \mathbf{I} = \begin{bmatrix} 5 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 5 \end{bmatrix}$$

$$\mathbf{H} = \begin{bmatrix} 5,0016 & 0,0009 & 0,0003 \\ 0,0009 & 5,0006 & 0,0002 \\ 0,0003 & 0,0002 & 5,0001 \end{bmatrix}$$

además

$$\mathbf{b} = (\mathbf{f} - \mathbf{w})^T \delta \mathbf{I} \mathbf{G}$$

$$\mathbf{b} = (\mathbf{f} - \mathbf{w})^T \begin{bmatrix} 0,0154 & 0,0000 & 0,0000 \\ 0,0305 & 0,0154 & 0,0000 \\ 0,0454 & 0,0305 & 0,0154 \end{bmatrix}$$

los vectores  $\mathbf{f}$  y  $\mathbf{w}$  serán calculados cada iteración, por lo tanto se eligió un instante  $k$  para demostrar el funcionamiento del algoritmo de control y optimización.

$$\mathbf{f} = \begin{bmatrix} 0,0149 \\ 0,0185 \\ 0,0220 \end{bmatrix} \quad \text{y} \quad \mathbf{w} = \begin{bmatrix} 11,0000 \\ 11,0000 \\ 11,0000 \end{bmatrix}$$

El vector  $\mathbf{b}$  queda definido como:

$$\mathbf{b} = \left( \begin{bmatrix} 0,0149 \\ 0,0185 \\ 0,0220 \end{bmatrix} - \begin{bmatrix} 11,0000 \\ 11,0000 \\ 11,0000 \end{bmatrix} \right)^T \begin{bmatrix} 0,0154 & 0,0000 & 0,0000 \\ 0,0305 & 0,0154 & 0,0000 \\ 0,0454 & 0,0305 & 0,0154 \end{bmatrix}$$

$$\mathbf{b} = \begin{bmatrix} -1,0021 & -0,5041 & -0,1690 \end{bmatrix}$$

Considerando la matriz Hessiana  $\mathbf{H}$ , el vector  $\mathbf{b}$  y las restricciones, precedemos a minimizar la función objetivo mediante el algoritmo GQP para el instante  $k$  seleccionado. La formula general de la función cuadrática a minimizar está representada por:

$$f(\Delta u_1, \Delta u_2, \Delta u_3) = 2,5008\Delta u_1^2 + 5,5003\Delta u_2^2 + 2,5001\Delta u_3^2 + 0,0009\Delta u_1\Delta u_2$$

$$+ 0,0003\Delta u_1\Delta u_3 + 0,0002\Delta u_2\Delta u_3 - 1,0021\Delta u_1 - 0,5041\Delta u_2 - 0,1690\Delta u_3$$

y está sujeta a:

Restricción en el incremento de control ( $\Delta u(t)$ ).

$$\begin{aligned} +1,0000\Delta u_1 & \leq 0,1000 \\ & +1,0000\Delta u_2 & \leq 0,1000 \\ & & +1,0000\Delta u_3 & \leq 0,1000 \\ -1,0000\Delta u_1 & \leq 0,1000 \end{aligned}$$

$$-1,0000\Delta u_2 \leq 0,1000$$

$$-1,0000\Delta u_3 \leq 0,1000$$

Restricción en la señal de control ( $u(t)$ ).

$$+1,0000\Delta u_1 \leq 13,000$$

$$+1,0000\Delta u_1 + 1,0000\Delta u_2 \leq 13,000$$

$$+1,0000\Delta u_1 + 1,0000\Delta u_2 + 1,0000\Delta u_3 \leq 13,000$$

$$-1,0000\Delta u_1 \leq 0,5000$$

$$-1,0000\Delta u_1 - 1,0000\Delta u_2 \leq 0,5000$$

$$-1,0000\Delta u_1 - 1,0000\Delta u_2 - 1,0000\Delta u_3 \leq 0,5000$$

Restricciones en la señal de salida ( $y(t)$ ).

$$+0,0077\Delta u_1 \leq 5,4851$$

$$+0,0153\Delta u_1 + 0,0077\Delta u_2 \leq 5,4815$$

$$+0,0227\Delta u_1 + 0,0153\Delta u_2 + 0,0077\Delta u_3 \leq 5,4780$$

$$-0,0077\Delta u_1 \leq 0,0149$$

$$-0,0153\Delta u_1 - 0,0077\Delta u_2 \leq 0,0185$$

$$-0,0227\Delta u_1 - 0,0153\Delta u_2 - 0,0077\Delta u_3 \leq 0,0220$$

Procedemos a minimizar la función objetivo mediante al algoritmo de optimización.

$$\mathbf{H} = \begin{bmatrix} 5,0016 & 0,0009 & 0,0003 \\ 0,0009 & 5,0006 & 0,0002 \\ 0,0003 & 0,0002 & 5,0001 \end{bmatrix}$$

$$\mathbf{b} = \begin{bmatrix} -1,0021 \\ -0,5041 \\ -0,1690 \end{bmatrix}$$

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & \textcircled{1} \\ 0 & 1 & 0 & \textcircled{2} \\ 0 & 0 & 1 & \textcircled{3} \\ -1 & 0 & 0 & \textcircled{4} \\ 0 & -1 & 0 & \textcircled{5} \\ 0 & 0 & -1 & \textcircled{6} \\ 1 & 0 & 0 & \textcircled{7} \\ 1 & 1 & 0 & \textcircled{8} \\ 1 & 1 & 1 & \textcircled{9} \\ -1 & 0 & 0 & \textcircled{10} \\ -1 & -1 & 0 & \textcircled{11} \\ -1 & -1 & -1 & \textcircled{12} \\ 0,0077 & 0 & 0 & \textcircled{13} \\ 0,0153 & 0,0077 & 0 & \textcircled{14} \\ 0,0227 & 0,0153 & 0,0077 & \textcircled{15} \\ -0,0077 & 0 & 0 & \textcircled{16} \\ -0,0153 & -0,0077 & 0 & \textcircled{17} \\ -0,0227 & -0,0153 & -0,0077 & \textcircled{18} \end{bmatrix} \quad \text{y} \quad \mathbf{d} = \begin{bmatrix} 0,1 \\ 0,1 \\ 0,1 \\ 0,1 \\ 0,1 \\ 0,1 \\ 13,000 \\ 13,000 \\ 13,000 \\ 0,5000 \\ 0,5000 \\ 0,5000 \\ 5,4851 \\ 5,4815 \\ 5,4780 \\ 0,0149 \\ 0,0185 \\ 0,0220 \end{bmatrix}$$

**Inicialización:**

$$\Delta \mathbf{u}_0 = \begin{bmatrix} -0,1000 \\ -0,1000 \\ -0,1000 \end{bmatrix}, \quad K_0 = \{4, 5, 6\}, \quad \mathbf{A}_0 = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix},$$

$$\mathbf{H}_0 = \begin{bmatrix} 5,0016 & 0,0009 & 0,0003 & -1,0000 & 0 & 0 \\ 0,0009 & 5,0006 & 0,0002 & 0 & -1,0000 & 0 \\ 0,0003 & 0,0002 & 5,0001 & 0 & 0 & -1,0000 \\ -1,0000 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1,0000 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1,0000 & 0 & 0 & 0 \end{bmatrix},$$

$$g_0 = \begin{bmatrix} -1,5024 \\ -1,0043 \\ -0,6691 \end{bmatrix}, \quad j = 0.$$

**Iteración 0**

**Paso 1:**

$\mathbf{H}_0$  es no singular.

Vaya al Paso 1.1.

**Paso 1.1:**

$$\text{Resolviendo } \mathbf{H}_0 \begin{bmatrix} \mathbf{s}_0 \\ \mathbf{v}_0 \end{bmatrix} = \begin{bmatrix} \mathbf{g}_0 \\ 0 \end{bmatrix} \text{ resulta } \mathbf{s}_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \text{ y } \mathbf{v}_0 = \begin{bmatrix} 1,5024 \\ 1,0043 \\ 0,6691 \end{bmatrix}.$$

Establezca  $\gamma_0 = 0$ .

**Paso 2:**

$$\tilde{\sigma}_0 = 1$$

$$\hat{\sigma}_0 = +\infty$$

$$\sigma_0 = \min\{1, +\infty\} = 1.$$

**Paso 3:**

$$\Delta \mathbf{u}_1 = \begin{bmatrix} -0,1000 \\ -0,1000 \\ -0,1000 \end{bmatrix}, \quad \mathbf{g}_1 = \begin{bmatrix} -1,5024 \\ -1,0043 \\ -0,6691 \end{bmatrix}$$

Vaya al Paso 3.2.

**Paso 3.2:**

$$u_1 = (0, 0, 0, -1,5024, -1,0043, -0,6691)', \quad (u_1)_4 = \min\{0, 0, 0, -1,5024, -1,5024, -1,0043, -0,6691\} = -1,5024, \quad k = 4, \quad K_1 = \{5, 6\} \quad \mathbf{A}_1 = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}, \quad j = 1 \quad \mathbf{H}_1 = \begin{bmatrix} 5,0016 & 0,0009 & 0,0003 & 0 & 0 \\ 0,0009 & 5,0006 & 0,0002 & -1,0000 & 0 \\ 0,0003 & 0,0002 & 5,0001 & 0 & -1,0000 \\ 0 & -1,0000 & 0 & 0 & 0 \\ 0 & 0 & -1,0000 & 0 & 0 \end{bmatrix}$$

**Iteración 1**

**Paso 1:**

$\mathbf{H}_1$  es no singular.

Vaya al Paso 1.1.

**Paso 1.1:**

$$\text{Resolviendo } \mathbf{H}_1 \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{v}_1 \end{bmatrix} = \begin{bmatrix} \mathbf{g}_1 \\ 0 \end{bmatrix} \text{ resulta } \mathbf{s}_1 = \begin{bmatrix} -0,3004 \\ 0 \\ 0 \end{bmatrix} \text{ y } \mathbf{v}_1 = \begin{bmatrix} 1,0040 \\ 0,6690 \end{bmatrix}.$$

Establezca  $\gamma_1 = 0$ .

**Paso 2:**

$$\tilde{\sigma}_1 = 1,$$

$$\hat{\sigma}_1 = \min\left\{\frac{-0,2}{-0,3004}; -; -; -; -; -\right\}$$

$$\sigma_1 = \min\{1; 0,6658\} = 0,6658.$$

**Paso 3:**

$$\Delta \mathbf{u}_2 = \begin{bmatrix} -0,1 \\ -0,1 \\ -0,1 \end{bmatrix} - 0,6658 \begin{bmatrix} -0,3004 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0,1 \\ -0,1 \\ -0,1 \end{bmatrix}, \quad \mathbf{g}_2 = \begin{bmatrix} -0,5020 \\ -1,0041 \\ -0,6690 \end{bmatrix}$$

Vaya al Paso 3.1.

**Paso 3.1:**

$$K_2 = \{5, 6, 1\}, \quad \mathbf{A}_2 = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \\ 1 & 0 & 0 \end{bmatrix}, \quad j = 2,$$

$$\mathbf{H}_2 = \begin{bmatrix} 5,0016 & 0,0009 & 0,0003 & 0 & 0 & 1,0000 \\ 0,0009 & 5,0006 & 0,0002 & -1,0000 & 0 & 0 \\ 0,0003 & 0,0002 & 5,0001 & 0 & -1,0000 & 0 \\ 0 & -1,0000 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1,0000 & 0 & 0 & 0 \\ 1,0000 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

**Iteración 2**

**Paso 1:**

$\mathbf{H}_2$  es no singular.

Vaya al Paso 1.1.

**Paso 1.1:**

$$\text{Resolviendo } \mathbf{H}_2 \begin{bmatrix} \mathbf{s}_2 \\ \mathbf{v}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{g}_2 \\ 0 \end{bmatrix} \text{ resulta } \mathbf{s}_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \text{ y } \mathbf{v}_2 = \begin{bmatrix} 1,0041 \\ 0,6690 \\ -0,5020 \end{bmatrix}.$$

Establezca  $\gamma_2 = 0$ .

**Paso 2:**

$$\tilde{\sigma}_2 = 1$$

$$\hat{\sigma}_2 = +\infty$$

$$\sigma_2 = \min\{1, +\infty\} = 1.$$

**Paso 3:**

$$\Delta \mathbf{u}_3 = \begin{bmatrix} 0,1000 \\ -0,1000 \\ -0,1000 \end{bmatrix}, \quad \mathbf{g}_3 = \begin{bmatrix} -0,5020 \\ -1,0041 \\ -0,6690 \end{bmatrix}$$

Vaya al Paso 3.2.

**Paso 3.2:**

$$u_1 = (-0,6690; 0; 0; 0; -0,5020; -1,0041)',$$

$$(u_1)_4 = \min\{-0,6690; 0; 0; 0; -0,5020; -1,0041\} = -1,0041, \quad k = 6,$$

$$K_3 = \{6, 1\} \quad \mathbf{A}_3 = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}, \quad j = 3$$

$$\mathbf{H}_3 = \begin{bmatrix} 5,0016 & 0,0009 & 0,0003 & 0 & 1,0000 \\ 0,0009 & 5,0006 & 0,0002 & 0 & 0 \\ 0,0003 & 0,0002 & 5,0001 & 1,0000 & 0 \\ 0 & 0 & 1,0000 & 0 & 0 \\ 1,0000 & 0 & 0 & 0 & 0 \end{bmatrix}$$

**Iteración 3****Paso 1:**

$\mathbf{H}_3$  es no singular.

Vaya al Paso 1.1.

**Paso 1.1:**

$$\text{Resolviendo } \mathbf{H}_3 \begin{bmatrix} \mathbf{s}_3 \\ \mathbf{v}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{g}_3 \\ 0 \end{bmatrix} \text{ resulta } \mathbf{s}_3 = \begin{bmatrix} 0 \\ -0,2008 \\ 0 \end{bmatrix} \text{ y } \mathbf{v}_3 = \begin{bmatrix} -0,6690 \\ -0,5018 \end{bmatrix}.$$

Establezca  $\gamma_3 = 0$ .

**Paso 2:**

$$\tilde{\sigma}_3 = 1,$$

$$\hat{\sigma}_1 = \min\{-; \frac{-0,2}{-0,2008}; -; -; -; -\}$$

$$\sigma_1 = \min\{1; 0,996\} = 0,996.$$

**Paso 3:**

$$\Delta \mathbf{u}_4 = \begin{bmatrix} 0,1 \\ -0,1 \\ -0,1 \end{bmatrix} - 0,996 \begin{bmatrix} -0 \\ -0,2008 \\ 0 \end{bmatrix} = \begin{bmatrix} 0,1 \\ 0,1 \\ -0,1 \end{bmatrix}, \quad \mathbf{g}_4 = \begin{bmatrix} -0,5018 \\ -0,0040 \\ -0,6690 \end{bmatrix}$$

Vaya al Paso 3.1.

**Paso 3.1:**

$$K_4 = \{6, 1, 2\}, \quad \mathbf{A}_4 = \begin{bmatrix} 0 & 0 & -1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad j = 4,$$

$$\mathbf{H}_4 = \begin{bmatrix} 5,0016 & 0,0009 & 0,0003 & 0 & 0 & 1,0000 \\ 0,0009 & 5,0006 & 0,0002 & -1,0000 & 0 & 0 \\ 0,0003 & 0,0002 & 5,0001 & 0 & -1,0000 & 0 \\ 0 & -1,0000 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1,0000 & 0 & 0 & 0 \\ 1,0000 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

**Iteración 4**

**Paso 1:**

$\mathbf{H}_4$  es no singular.

Vaya al Paso 1.1.

**Paso 1.1:**

$$\text{Resolviendo } \mathbf{H}_4 \begin{bmatrix} \mathbf{s}_4 \\ \mathbf{v}_4 \end{bmatrix} = \begin{bmatrix} \mathbf{g}_4 \\ 0 \end{bmatrix} \text{ resulta } \mathbf{s}_4 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \text{ y } \mathbf{v}_4 = \begin{bmatrix} 0,0040 \\ 0,6690 \\ -0,5018 \end{bmatrix}.$$

Establezca  $\gamma_4 = 0$ .

**Paso 2:**

$$\tilde{\sigma}_4 = 1$$

$$\hat{\sigma}_4 = +\infty$$

$$\sigma_4 = \min\{1, +\infty\} = 1.$$

**Paso 3:**

$$\Delta \mathbf{u}_5 = \begin{bmatrix} 0,1000 \\ 0,1000 \\ -0,1000 \end{bmatrix}, \quad \mathbf{g}_5 = \begin{bmatrix} -0,5018 \\ -0,0040 \\ -0,6690 \end{bmatrix}$$

Vaya al Paso 3.2.

**Paso 3.2:**

$$u_5 = (-0,0040; -0,5018; 0; 0; 0; -0,6690)',$$

$$(u_5)_6 = \min\{-0,0040; -0,5018; 0; 0; 0; -0,6690\} = -0,6690, \quad k = 6,$$

$$K_5 = \{1, 2\} \quad \mathbf{A}_5 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad j = 5$$

$$\mathbf{H}_5 = \begin{bmatrix} 5,0016 & 0,0009 & 0,0003 & 1,0000 & 0 \\ 0,0009 & 5,0006 & 0,0002 & 0 & 1,0000 \\ 0,0003 & 0,0002 & 5,0001 & 1,0000 & 0 \\ 1,0000 & 0 & 0 & 0 & 0 \\ 0 & 1,0000 & 0 & 0 & 0 \end{bmatrix}$$

**Iteración 5**

**Paso 1:**

$\mathbf{H}_5$  es no singular.

Vaya al Paso 1.1.

**Paso 1.1:**

$$\text{Resolviendo } \mathbf{H}_5 \begin{bmatrix} \mathbf{s}_5 \\ \mathbf{v}_5 \end{bmatrix} = \begin{bmatrix} \mathbf{g}_5 \\ 0 \end{bmatrix} \text{ resulta } \mathbf{s}_5 = \begin{bmatrix} 0 \\ 0 \\ -0,1338 \end{bmatrix} \text{ y } \mathbf{v}_5 = \begin{bmatrix} -0,5018 \\ -0,0040 \end{bmatrix}.$$

Establezca  $\gamma_5 = 0$ .

**Paso 2:**

$$\tilde{\sigma}_5 = 1,$$

$$\hat{\sigma}_5 = \min\{-; -; \frac{-0,2}{-0,1338}; -; -; -\}$$

$$\sigma_5 = \min\{1; 1,4948\} = 1.$$

**Paso 3:**

$$\Delta \mathbf{u}_6 = \begin{bmatrix} 0,1 \\ 0,1 \\ -0,1 \end{bmatrix} - 1,0000 \begin{bmatrix} 0 \\ 0 \\ -0,1338 \end{bmatrix} = \begin{bmatrix} 0,1 \\ 0,1 \\ 0,0338 \end{bmatrix}, \quad \mathbf{g}_6 = \begin{bmatrix} -0,5018 \\ -0,0040 \\ 0,0000 \end{bmatrix}$$

Vaya al Paso 3.1.

**Paso 3.2:**

$$u_5 = (0,5018; 0,0040; 0; 0; 0; 0)',$$

$$(u_5)_6 = \min\{0,5018; 0,0040; 0; 0; 0; 0\} = 0, \quad k = 6,$$

$$(u_5)_6 \geq 0; \text{ Parar con la solución óptima } \Delta \mathbf{u}_6 = \begin{bmatrix} 0,1 \\ 0,1 \\ 0,0338 \end{bmatrix}$$

El valor  $\Delta u(t)$  tomado será el de la primera posición del vector  $\Delta \mathbf{u}$  para el caso ejemplificado el valor de 0,1.

4. Salida de control  $u(t)$ .

La señal de control es una función de la referencia deseada y de entradas y salidas pasadas, la cual está dada por:

$$u(t) = u(t - 1) - 0,0595y(t) + 0,0412y(t - 1) + \\ + 0,0031w(t + 1) + 0,0061w(t + 2) + 0,0091w(t + 3) \quad (77)$$

**4.1.4. Selección del software y hardware IEC-41699.**

Para la implementación del controlador predictivo se optó por la selección de software y hardware que soporte la estructura de modelado distribuido basado en la IEC-41699 y en el protocolo de comunicación OPC-UA.

El entorno de desarrollo (IDE) Open Source optado fue el 4DIAC (4DIAC-IDE) versión 1.8.2, el cual es un IDE robusto y tiene soporte oficial de Eclipse 4diac. Contiene un test y debug a través de un display online que posibilita establecer y leer valores variables de forma remota[34].

La plataforma de ejecución (runtime) elegida fue el 4DIAC (4DIAC-RTE, FORTE) a causa de la versatilidad para seleccionar el hardware, no depende de la plataforma en la que se ejecute y trabaja sobre sistemas operativos como Windows o Linux, tiene soporte oficial de Eclipse 4diac y es compatible con muchas tarjetas embebidas y PLCs. FORTE soporta la reconfiguración on-line de sus aplicaciones y la ejecución en tiempo real de los FB ofrecidos por el estándar IEC-61499[34].

Una vez seleccionado el software que se empleará para la implementación del sistema de control, se procede a la selección del sistema embebido o PLC compatible con el software antes seleccionado. 1.

Tabla 2: Tarjetas Embebidas y PLC compatibles con IEC-61499.

Tarjetas embebidas	Controladores Lógicos Programables (PLC)
BeagleBone Black	Bachmann electronic M1 PLC
Digi Connect ME (ARM7)	CONMELEON C1
KIPR's CBC v2 robot controller	Raspberry-SPS
Lego Mindstorms EV3 (ARM7)	WAGO PFC200
Raspberry PI	

Se descartaron las posibilidades de utilizar los PLCs debido a que el controlador será implementado en un sistema didáctico de control de nivel en un ambiente no industrial.

El sistema embebido optado fue la BBB de Texas Instruments porque es una plataforma de desarrollo de bajo costo respaldada por la comunidad para desarrolladores y aficionados, además consta con pines de entrada y salida analógicas necesarios para nuestra aplicación[34]. En la Figura 40 se representa la tarjeta BBB.



Figura 40: Tarjeta BeagleBone Black.

Tabla 3: Características de la tarjeta BBB.

Beagle BoneBlack
Procesador AM335x 1GHz ARM® Cortex-A8
Almacenamiento flash integrado eMMC de 4 GB y 8 bits
Acelerador de gráficos 3D
Acelerador de punto flotante NEON
microcontroladores 2x PRU de 32 bits

#### 4.1.5. Implementación de la metodología 4DIAC.

Para llevar a cabo la implementación de la metodología 4DIAC, se han utilizado varias herramientas software y hardware previamente seleccionadas. Inicialmente se ha empleado la herramienta de ingeniería 4DIAC que trabaja con una máquina virtual de Java. Esta herramienta permite editar y ejecutar proyectos basados en IEC-61499 bajo el runtime FORTE y el runtime FBRT. Además, se ha utilizado el editor de texto NANO como una herramienta de programación C++.

La Figura 41 ilustra el escenario general para implementar SIFBs usando las herramientas 4DIAC y el editor de texto NANO C++. Las estructuras vacías de los SIFBs implementadas en C++ serán editadas con los correspondientes algoritmos y librerías de usuario para ser compilados y finalmente ejecutados, la estructura del SIFB GPC\_CONTROL.cpp que se ilustra en la Figura 42 incluye una declaración de tipos de eventos de entrada y salida además de datos de entrada y salida. Esta estructura se exporta como ficheros .cpp y .h con estructura FORTE.

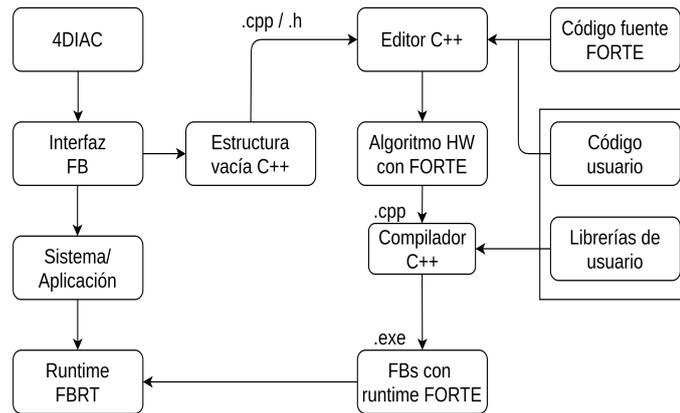


Figura 41: Escenario general de desarrollo del software en 4DIAC.

```

Abrir  GPC_CONTROL.h  Guardar
~/Documents/FORTE/GENERADO/EXPORTADOS
1 #include "GPC_CONTROL.h"
2 #ifdef FORTE_ENABLE_GENERATED_SOURCE_CPP
3 #include "GPC_CONTROL_gen.cpp"
4 #endif
5
6 DEFINE_FIRMWARE_FB(FORTE_GPC_CONTROL, g_nStringIdGPC_CONTROL)
7
8 const CStringDictionary::TStringId FORTE_GPC_CONTROL::scm_anDataInputNames[] = {g_nStringIdI, g_nStringIdSETPOINT};
9 const CStringDictionary::TStringId FORTE_GPC_CONTROL::scm_anDataInputTypeIds[] = {g_nStringIdBOOL, g_nStringIdREAL};
10 const CStringDictionary::TStringId FORTE_GPC_CONTROL::scm_anDataOutputNames[] = {g_nStringIdQ, g_nStringIdOUTPUT, g_nStringIdCONTROL,
11 g_nStringIdINCREASE_CONTROL};
12 const CStringDictionary::TStringId FORTE_GPC_CONTROL::scm_anDataOutputTypeIds[] = {g_nStringIdBOOL, g_nStringIdREAL, g_nStringIdREAL,
13 g_nStringIdREAL};
14 const TforteInt16 FORTE_GPC_CONTROL::scm_anEIWithIndexes[] = {0, 2};
15 const TdataIOID FORTE_GPC_CONTROL::scm_anEIWith[] = {0, 255, 0, 1, 255};
16 const CStringDictionary::TStringId FORTE_GPC_CONTROL::scm_anEventInputNames[] = {g_nStringIdINIT, g_nStringIdREQ};
17 const TforteInt16 FORTE_GPC_CONTROL::scm_anEventInputIndexes[] = {0, 2, -1};
18 const SFBInterfaceSpec FORTE_GPC_CONTROL::scm_sfbInterfaceSpec = {
19 2, scm_anEventInputNames, scm_anEIWith, scm_anEIWithIndexes,
20 2, scm_anEventOutputNames, scm_anEOWith, scm_anEOWithIndexes, 2, scm_anDataInputNames, scm_anDataInputTypeIds,
21 4, scm_anDataOutputNames, scm_anDataOutputTypeIds,
22 0, 0
23 };
24
25
26 void FORTE_GPC_CONTROL::executeEvent(int pa_nEIID){
27 switch(pa_nEIID){
28 case scm_nEventINITID:
29 break;
30 case scm_nEventREQID:
31 break;
32 }
33 }
34 }
35 }
36 }
  
```

Figura 42: Estructura en C++ de SIFB GPC\_CONTROL.

Para el manejo de la entrada analógica, implementación del algoritmo de control y cliente OPC UA se desarrolló un conjunto de FBs que contienen la programación necesaria empleada en la aplicación. Los FBs desarrollados encapsulan algoritmos mediante la implementación código de usuario y librerías dentro de su codificación en C++. Existe gran cantidad de librerías desarrolladas específicamente para el manejo de puertos GPIO, ANALÓGICOS y PWM de la BBB, para nuestro caso se utilizó la librería BlackLib.

## 1. Bloque de Función ANALOG\_INPUT.

Este FB fue diseñado para la lectura del valor del sensor analógico de nivel empleado en la aplicación.

En la Figura 43 se representa el FB ANALOG\_INPUT, el cual consta de dos eventos en la entrada: el evento INIT se ejecuta al iniciar el FB y el evento REQ que se ejecuta periódicamente para leer el valor del sensor analógico, del mismo modo se tiene dos eventos de salida: el evento INITO indica que se completó la inicialización del FB y el evento CNF que indica que se ejecutó correctamente el evento de entrada REQ.

Además el FB consta de una entrada de datos tipo STRING llamada NAME, en donde se encuentra el nombre del sensor de la entrada analógico y una salida de datos tipo REAL nombrada INPUT\_VALUE la cual indica el valor medido por el sensor analógico.

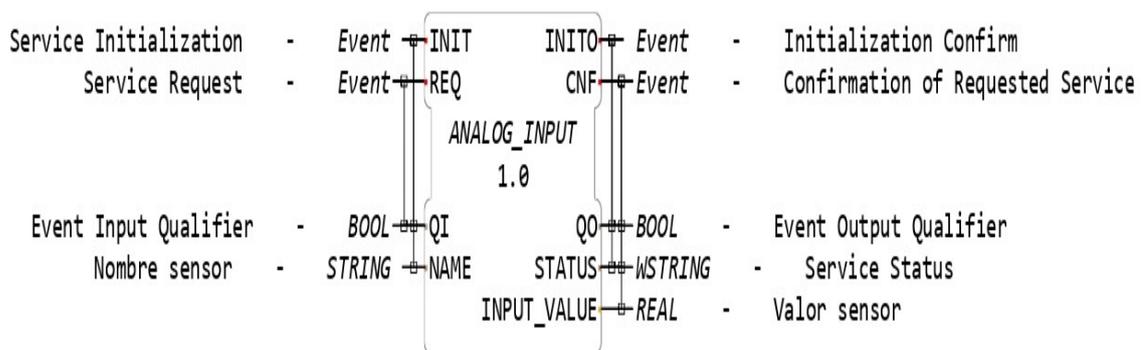


Figura 43: Bloque de Función ANALOG\_INPUT.

## 2. Bloque de Función GPC\_CONTROL.

Este FB fue diseñado para implementar el algoritmo de control del GPC con las tres restricciones antes mencionadas.

En la Figura 44 se representa el FB GPC\_CONTROL, el cual consta de dos eventos en la entrada: el evento INIT se ejecuta al iniciar el FB y el evento REQ que se ejecuta periódicamente para leer el valor del setpoint deseado y realizar los diferentes algoritmos de control implementados en el GPC, del mismo modo se tiene dos eventos de salida: el evento INITO indica que se completó la inicialización del FB y el evento CNF que indica que se ejecutó correctamente el evento de entrada REQ.

Además el FB consta de una entrada de datos tipo REAL nombrada SETPOINT

la cual lee el valor del setpoint. De igual manera consta de 3 salidas tipo REAL nombradas OUTPUT, CONTROL e INCREASE\_CONTROL las cuales indican los valores de señales de: la salida de la planta, señal de control e incremento en la señal de control respectivamente.

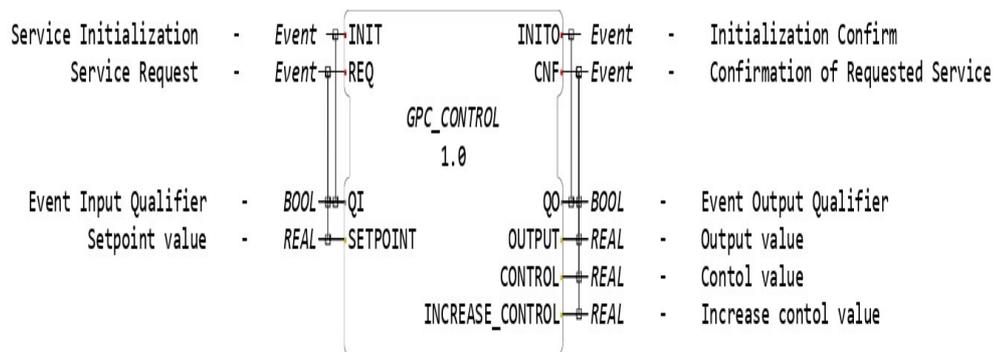


Figura 44: Bloque de Función GPC\_CONTROL.

### 3. Bloque de Función OPC\_UA\_CLIENT.

Este FB fue diseñado para implementar un cliente OPC UA el cual publicará los diferentes valores de las variables del sistema en el servidor OPC UA.

En la Figura 45 se representa el FB OPC\_UA\_CLIENT, el cual consta de dos eventos en la entrada: el evento INIT se ejecuta al iniciar el FB y el evento REQ que se ejecuta periódicamente para leer el valor del setpoint, el valor de salida del controlador, la señal de control y el valor del sensor analógico, del mismo modo se tiene dos eventos de salida: el evento INITO indica que se completó la inicialización del FB y el evento CNF que indica que se ejecutó correctamente el evento de entrada REQ.

Además el FB consta de 4 entradas de datos tipo REAL nombradas VALUE\_SETPOINT, VALUE\_OUTPUT, VALUE\_CONTROL, VALUE\_SENSOR las cuales leen los valores deseados de: setpoint, salida de la planta, señal de control y el valor del sensor analógico respectivamente.

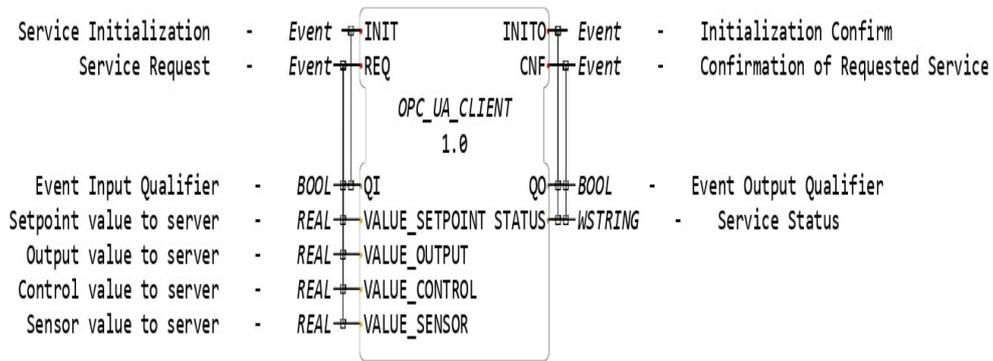


Figura 45: Bloque de Función Cliente OPC UA.

En la Figura 46 se presentan las funciones utilizadas en el Bloque de Función GPC \_CONTROL, donde se crearon cuatro funciones principales y una función secundaria: En la función Read\_Sensor() se lee el valor capturado del sensor de nivel, la función Write\_PWM() envía el valor de pwm al actuador, la función Algoritmo\_GPC() contiene los 4 pasos descritos anteriormente. La función secundaria Quadprog() resuelve la minimización de la función objetivo, finalmente todas estas funciones son invocadas en la función principal main(). La descripción de las funciones se presenta a continuación en la Tabla 3.

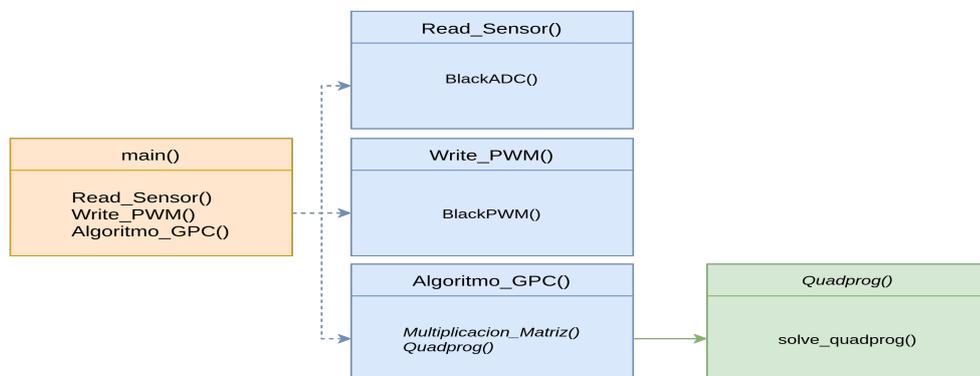


Figura 46: Diagrama de Funciones Controlados GPC.

Tabla 4: Descripción de las funciones del Servidor OPC UA.

Función	Descripción
QuadProg()	Desarrolla el algoritmo de optimización cuadrática
Algoritmo_GPC()	Desarrolla el algoritmo de control del GPC
Write_PWM()	Escribe el valor que será enviado como PWM
Read_Senro()	Lee el valor del sensor de nivel

#### 4.1.6. Desarrollo servidor OPC UA.

Para el desarrollo del servidor OPC UA será útil instalar un cliente OPC UA (UAExpert) que tiene una interfaz gráfica, el cual permite examinar el modelo de información del servidor OPC UA desarrollado, el cual se indica en la Figura 47.

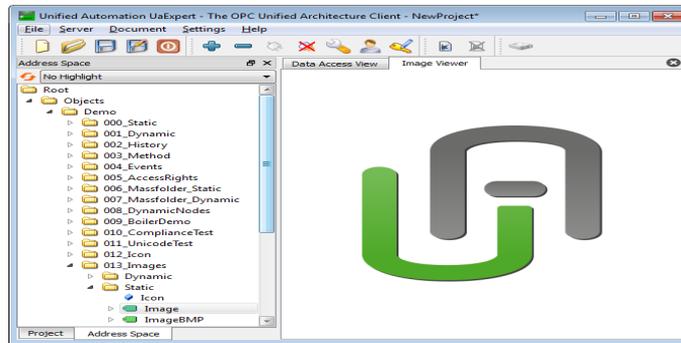


Figura 47: Cliente OPC UA UAExpert.

Como ya se ha mencionado anteriormente la aplicación modelada fue un control predictivo GPC de nivel. En la Figura 48 se indica una aplicación típica de control de nivel (escenario de aplicación OPC UA), donde se puede observar las señales de: Setpoint, Control, Salida y Sensor. En esta sección no indicaremos detalles sobre la lógica interna del controlador, pero se puede visualizar que consta con un sensor (nivel), un actor (bomba de agua), módulo de control (BBB) y una estación de ingeniería (computador).

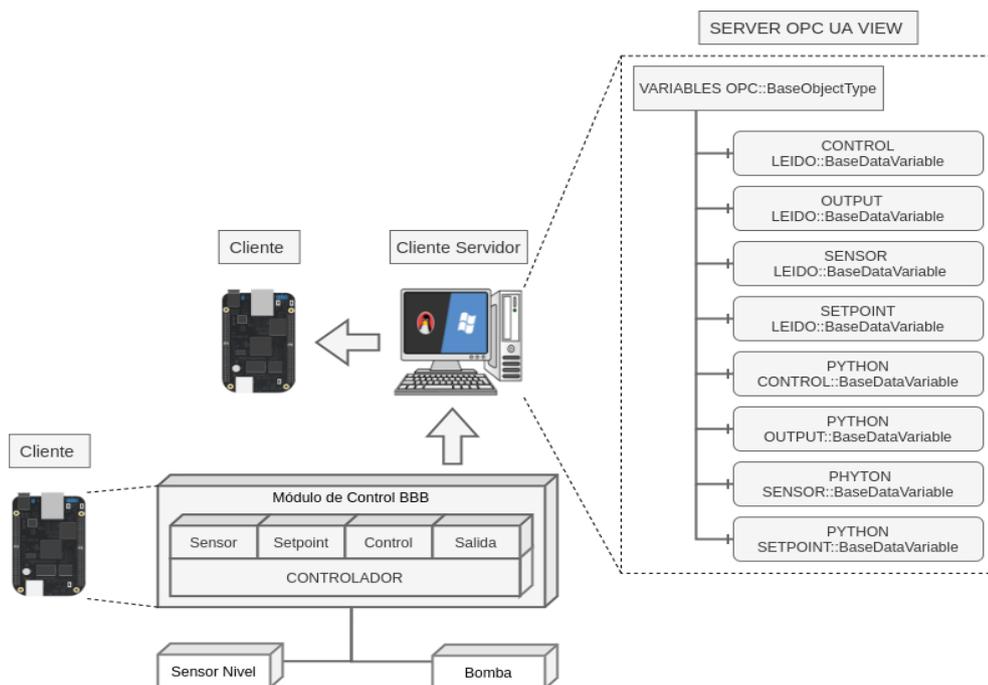


Figura 48: Escenario de aplicación del control de nivel.

En un OPC clásico, se proporcionaría un servidor OPC DA que permite acceder a los valores medidos y establecer los puntos de ajuste, para nuestra aplicación se desea proporcionar información similar en un servidor OPC UA, su AddressSpace se vería similar al de la Figura 49. Aquí, solo el BaseObjectType y BaseDataVariableType son utilizados y la Referencia estándar HasComponent.

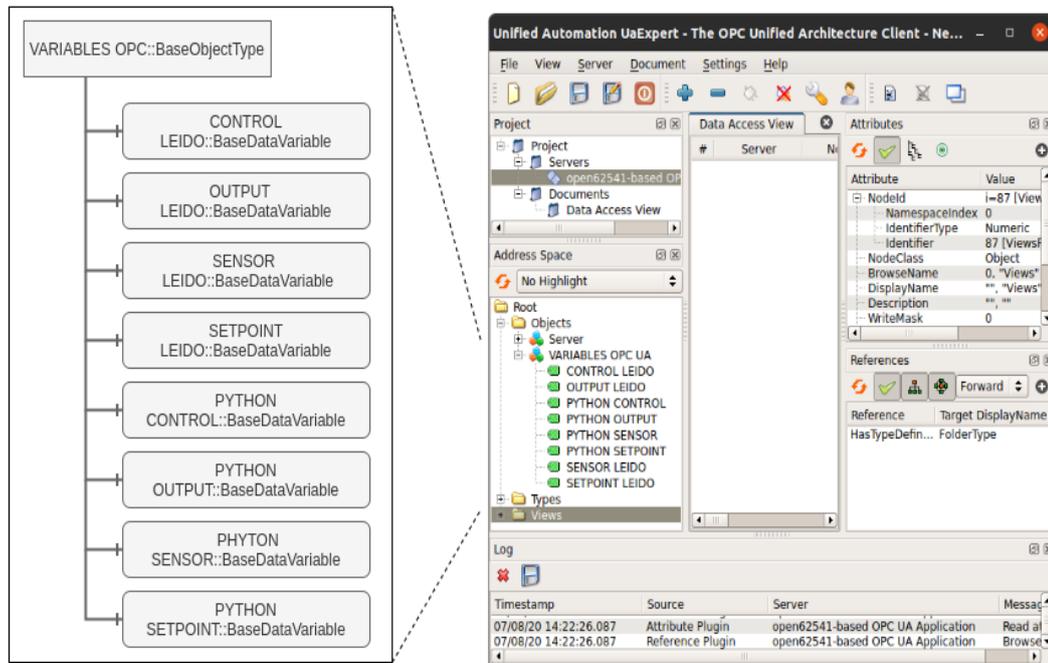


Figura 49: Mapeo de un simple escenario OPC UA.

Los Atributtes de un Node se pueden colocar dentro del elemento gráfico. Esto se ejemplifica usando un Object como se indica en la Figura 50, pero se puede aplicar en cualquier NodeClass.

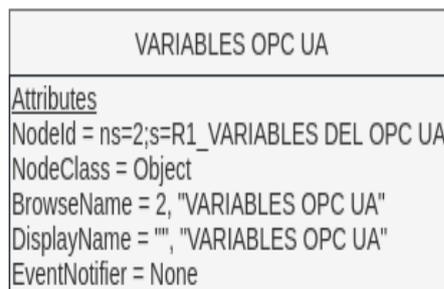


Figura 50: Atributos incluidos en un Nodo.

En la Figura 51 se representa mediante notación gráfica OPC UA la aplicación implementada, donde se observa que el Object VARIABLE OPC UA tiene 6 DataVariables con sus respectivos Atributos.

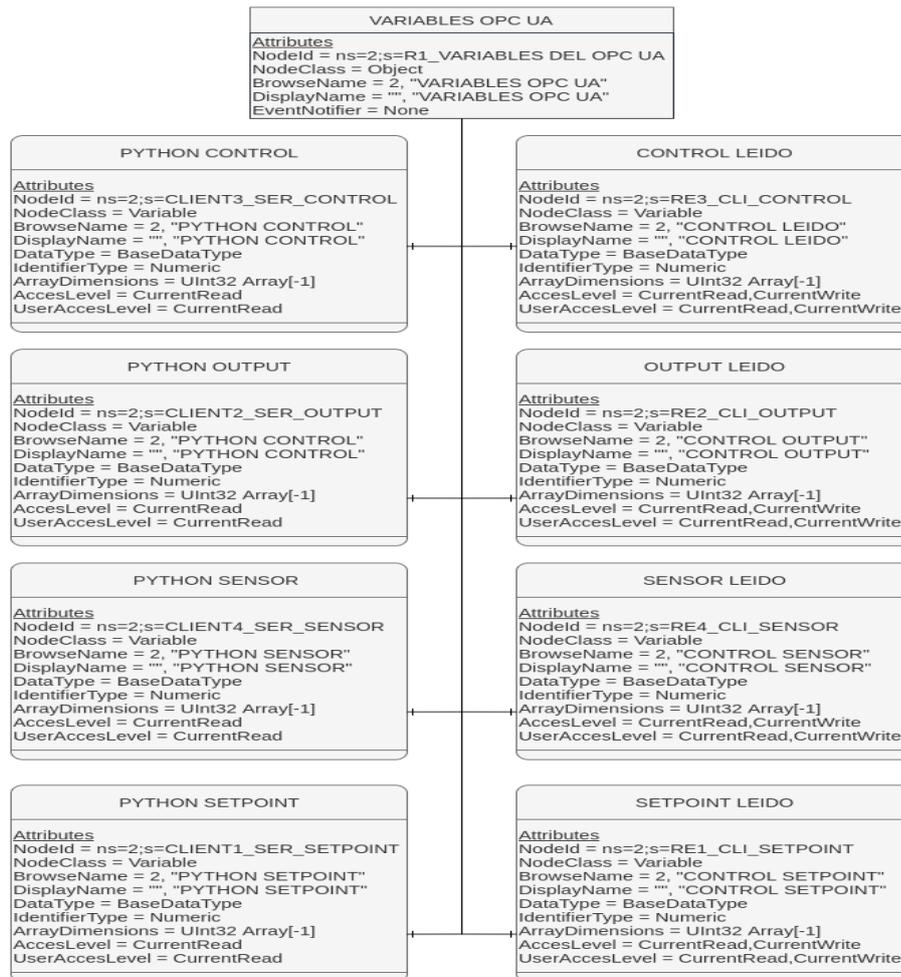


Figura 51: Objeto de dispositivo que expone referencias.

El NodeClass Object VARIABLES OPC UA desarrollado se observa en la interfaz gráfica del cliente OPC UA UAExpert como se indica en la Figura 53, además se indican las variables del controlador (NodeClass Variable) empleadas en el desarrollo del sistema.

Los valores de las NodeClass Variable indicadas a continuación son leídas por el servidor OPC UA, las cuales son enviados por el FB cliente OPC\_UA\_CLIENT que se encuentra ejecutándose en el 4DIAC IDE.

- SETPOINT LEÍDO.- Lee el valor del setpoint deseado.
- CONTROL LEÍDO.- Lee el valor de la señal de control.
- OUTPUT LEÍDO.- Lee el valor de la señal de salida que es enviada al actuador.
- SENSOR LEÍDO.- Lee el valor del sensor de nivel.

Las valores de las NodeClass Variable indicadas a continuación son puestas a disposición por el servidor OPC UA, éstos valores podrán ser leídos por los clientes OPC UA

en nuestro caso para visualizar los valores en un HMM2.

- PYTHON SETPOINT.- Escribe el valor del setpoint deseado.
- PYTHON CONTROL.- Escribe el valor de la señal de control.
- PYTHON OUTPUT.- Escribe el valor de la señal de salida que es enviada al actuator.
- PYTHON SENSOR.- Escribe el valor del sensor de nivel.

En la Figura 52 se presentan las funciones creadas para el desarrollo del Servidor OPC-UA, éstas funciones se describen en la Tabla 4.



Figura 52: Diagrama de funciones Servidor OPC UA.

Tabla 5: Descripción de las funciones del Servidor OPC UA.

Función	Descripción
Function_Read_Setpoint()	Lee el valor del setpoint enviado desde el Bloque de Función Cliente OPC UA
Function_Send_Setpoint()	Lee el valor del setpoint enviado desde el Bloque de Función Cliente OPC UA y lo pone a disposición para la escritura del valor en cualesquier Cliente OPC UA
Function_Read_Control()	Lee el valor del control enviado desde el Bloque de Función Cliente OPC UA
Function_Send_Control()	Lee el valor del control enviado desde el Bloque de Función Cliente OPC UA y lo pone a disposición para la escritura del valor cualesquier Cliente OPC UA
Function_Read_Salida()	Lee el valor del salida enviado desde el Bloque de Función Cliente OPC UA
Function_Send_Salida()	Lee el valor del salida enviado desde el Bloque de Función Cliente OPC UA y lo pone a disposición para la escritura del valor en cualesquier Cliente OPC UA
Function_Read_Sensor()	Lee el valor del sensor enviado desde el Bloque de Función Cliente OPC UA
Function_Send_Sensor()	Lee el valor del sensor enviado desde el Bloque de Función Cliente OPC UA y lo pone a disposición para la escritura del valor en cualesquier Cliente OPC UA
Variables_OPC_Read_Setpoint()	Actualiza la variable setpoint en el servidor OPC UA enviada desde el Bloque de Funcion Cliente OPC UA
Variables_OPC_Write_Setpoint()	Actualiza la variable setpoint en el servidor OPC UA enviada desde el Bloque de Funcion OPC UA dispuesta para cualesquier Cliente OPC
Variables_OPC_Read_Control()	Actualiza la variable control en el servidor OPC UA enviada desde el Bloque de Funcion Cliente OPC UA
Variables_OPC_Write_Control()	Actualiza la variable control en el servidor OPC UA enviada desde el Bloque de Funcion OPC UA dispuesta para cualesquier Cliente OPC
Variables_OPC_Read_Salida()	Actualiza la variable salida en el servidor OPC UA enviada desde el Bloque de Funcion Cliente OPC UA
Variables_OPC_Write_Salida()	Actualiza la variable salida en el servidor OPC UA enviada desde el Bloque de Funcion OPC UA dispuesta para cualesquier Cliente OPC
Variables_OPC_Read_Sensor()	Actualiza la variable sensor en el servidor OPC UA enviada desde el Bloque de Funcion Cliente OPC UA
Variables_OPC_Write_Sensor()	Actualiza la variable sensor en el servidor OPC UA enviada desde el Bloque de Funcion OPC UA dispuesta para cualesquier Cliente OPC
Variables_OPC()	Obtiene el ID del nodo asignado por el servidor
stopHandler()	Detiene la ejecución del servidor OPC UA
main()	Contiene todas las funciones creadas

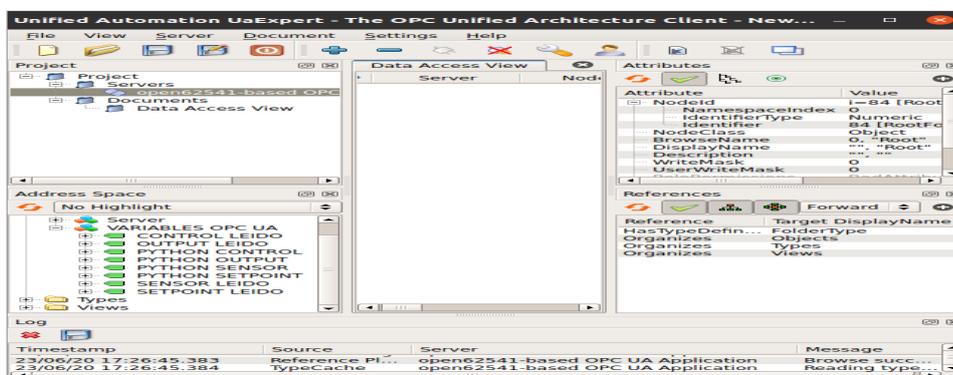


Figura 53: Servidor OPC UA implementado.

#### 4.1.7. Diseño de la aplicación en 4DIAC-IDE.

En la Figura 54 se representa la aplicación de control realizada en el entorno de desarrollo 4DIAC-IDE, contiene dos tipos de dispositivos detallados a continuación:

El primer dispositivo será la estación HMI la cual está representada en color morado, éste dispositivo permitirá la manipulación del setpoint deseado, adicionalmente indicará los valores de salida de planta, señal de control, valor del sensor analógico e incremento en la señal de control.

El segundo dispositivo será la tarjeta BBB y se encuentra de color marrón, la cual contiene los FBs para la entrada analógica, algoritmo de control GPC y cliente OPC UA.

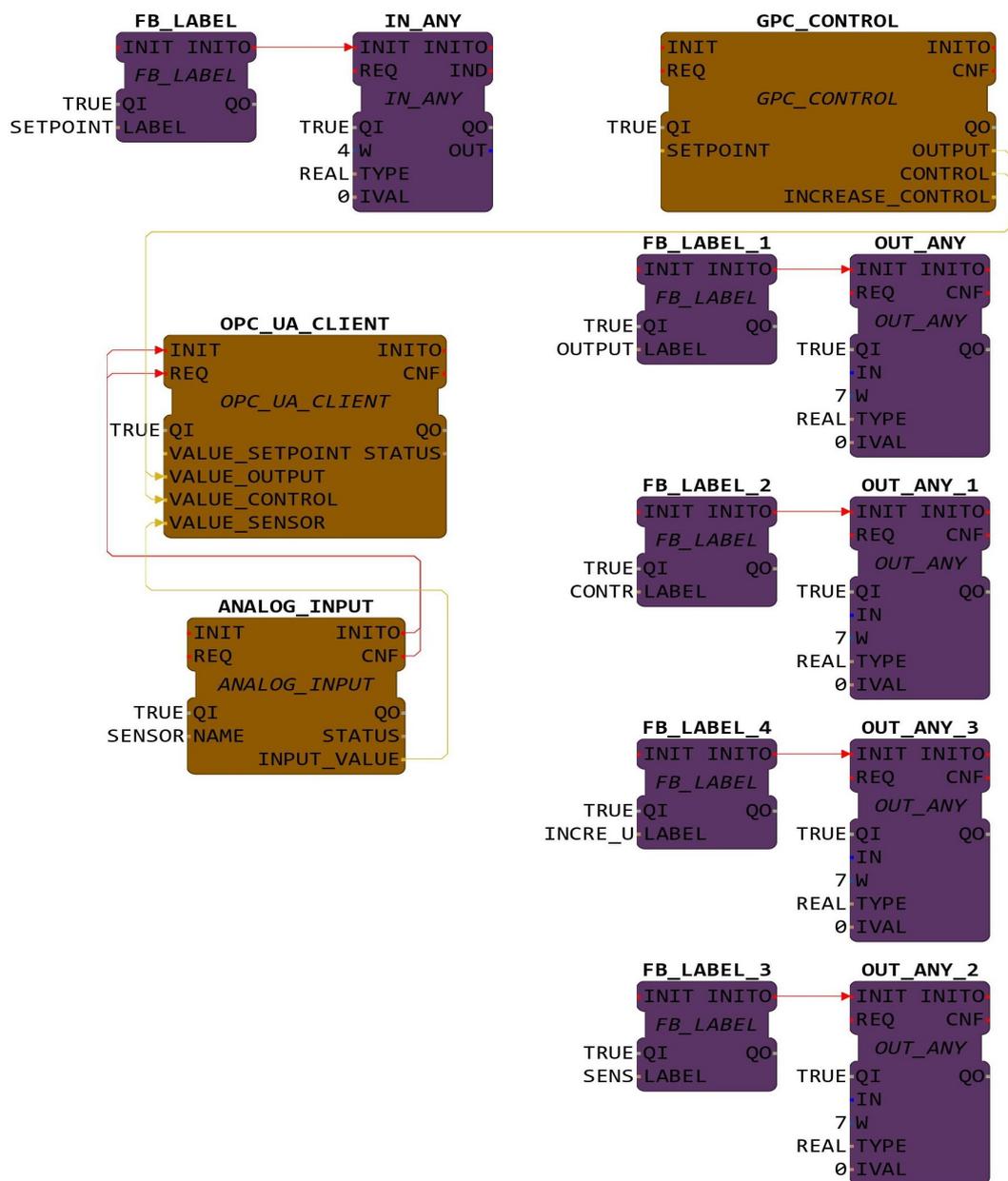


Figura 54: Aplicación de control en 4DIAC-IDE.

El sistema de comunicación se configura con dos dispositivos con un único recurso cada uno y comunicados a través de una red ethernet como se indica en la Figura 55.

El primer dispositivo es un FBRT\_WINDOWS con un único recurso para un panel gráfico (PANEL\_RESOURCE) denominado HMI, a este dispositivo le corresponde la dirección IP 192.168.0.11:61499. El segundo dispositivo es la BBB con un único recurso para un tipo embebido simple EMB\_RES denominados BBB\_NIVEL a este dispositivo le corresponde la dirección IP 192.168.0.10:61499.

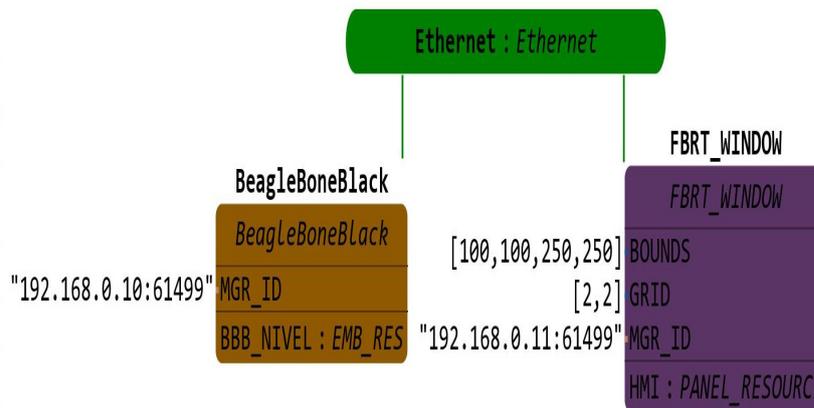


Figura 55: Configuración del Sistema.

Los FBs de color morado mapeados sobre el recurso HMI, permitirá obtener la interfaz gráfica (HMI) con el fin de observar el correcto funcionamiento del controlador. Este recurso se descargará sobre el runtime FBRT de 4DIAC-IDE, el cual se ejecuta en la estación de ingeniería (PC) para visualizar la aplicación HMI. Este recurso contiene FBs de tipo PUBLISH quienes envían la información al recurso BBB\_NIVEL y FBs tipo SUBSCRIBE quienes reciben la información del recurso BBB\_NIVEL.

Este recurso al ser descargado en el runtime FBRT del 4DIAC-IDE proporciona una interfaz gráfica muy sencilla la cual se presenta en la Figura 56.

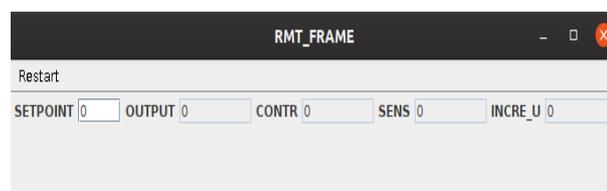


Figura 56: Interfaz Hombre Máquina (HMI).

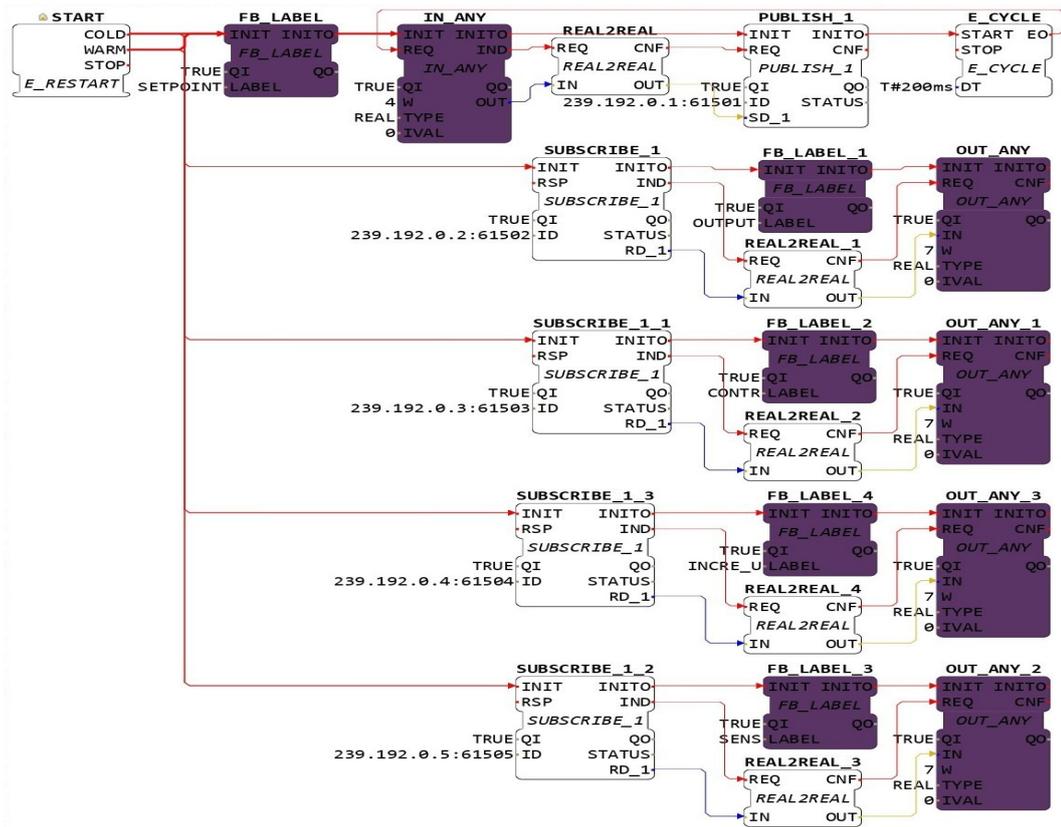


Figura 57: Configuración del recurso HMI.

El recurso BBB\_NIVEL posee los FBs diseñados para el GPC, manipulación de la entrada analógica y cliente OPC UA, el cual se descargará sobre el runtime FORTE a través del 4DIAC-IDE, además contiene FBs de tipo PUBLISH quienes envían la información al recurso HMI y FBs tipo SUBSCRIBE quienes reciben la información del recurso HMI.

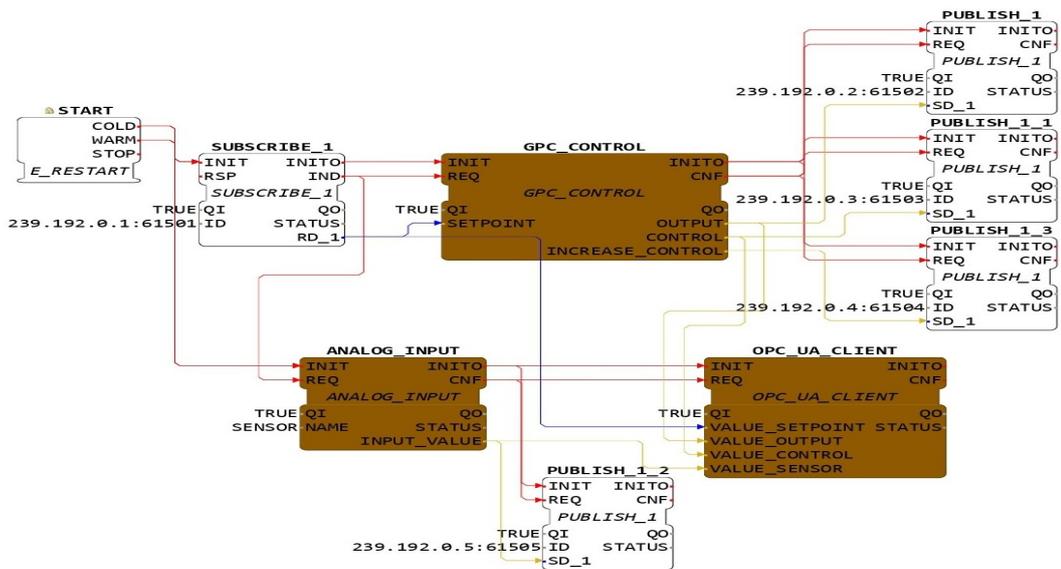


Figura 58: Configuración de Recurso BBB\_NIVEL.

#### 4.1.8. Distribución física del sistema implementado.

La distribución física del sistema de control desarrollado se representa en la Figura 59, está compuesta por una tarjeta BBB, una estación de ingeniería (PC) y la planta Festo MPS-PA Compact Workstation, la aplicación tiene una conexión de red ethernet.

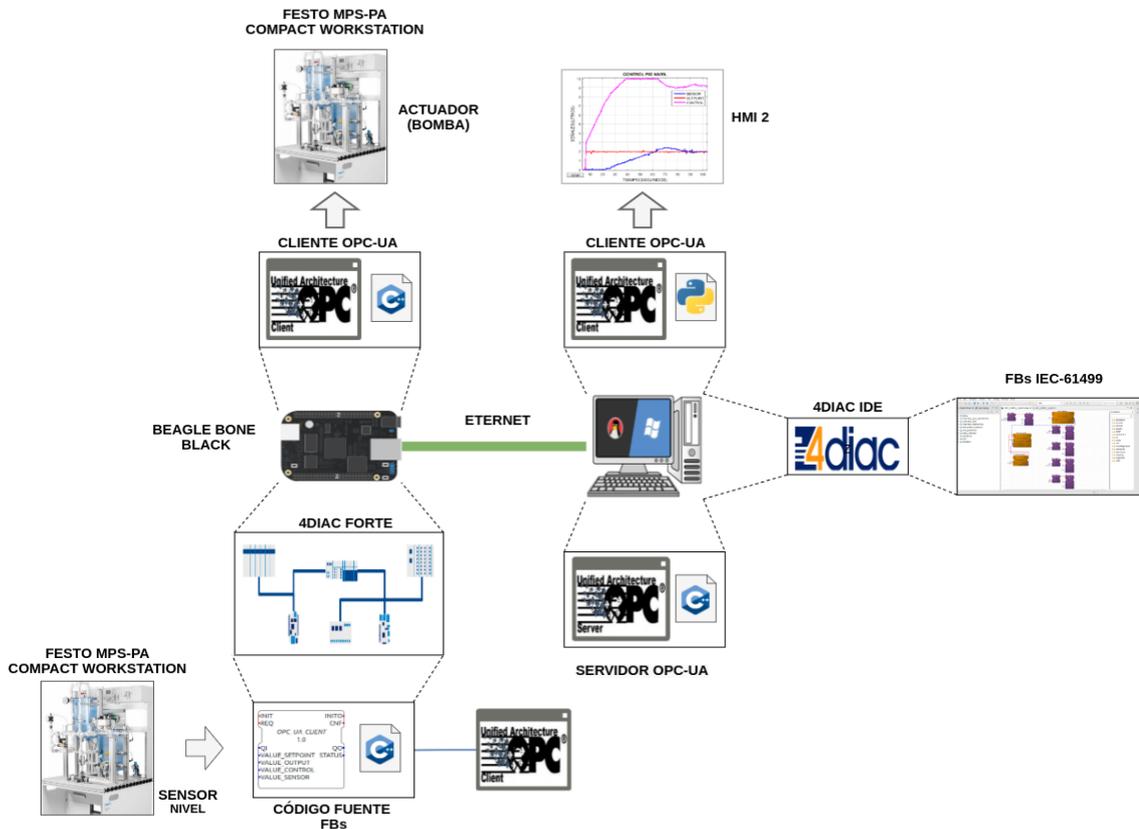


Figura 59: Distribución Física del Sistema Distribuido.

En la PC se ejecuta:

- 4DIAC-IDE.- Se emplea para la modificación, descarga y conexión de los FBs del sistema distribuido con sus respectivas aplicaciones, donde además se ejecuta una aplicación HMI-1 para visualizar las variables principales del proceso e introducir el parámetro de referencia (setpoint) deseado.
- Servidor OPC UA.- Encargado de publicar y suscribir los valores de las variables empleados en el sistema distribuido.
- Cliente OPC UA.- Ejecuta un código fuente implementado en python para una aplicación HMI-2 y visualizar las variables principales del proceso en tiempo real.

La tarjeta BBB se encuentra interactuando con el sensor y actuador de la planta Festo MPS-PA Compact Workstation, en donde además se ejecuta lo siguiente:

- 4DIAC-FORTE.- Contiene todos los códigos fuente de los FBs utilizados en la aplicación.
- Cliente OPC UA.- Ejecuta un código fuente implementado en C++ para enviar la señal de PWM hacia el actuador (bomba).

#### 4.1.9. Pruebas de funcionamiento.

El controlador GPC de nivel implementado mediante la normativa IEC 61499 y el protocolo de comunicación OPC UA tiene una alto grado de control distribuido el cual es uno de los objetivos a cumplir por la normativa y el protocolo antes mencionado. Los valores de las variables empleadas en la aplicación, las cuales son enviadas al servidor OPC UA mediante un cliente OPC UA (publicador), pueden ser visualizadas en cualquier tipo de PC conectada a nuestra red local mediante la implementación de un cliente OPC UA (subscriber). En la aplicación implementada se ejecutó el servidor OPC UA y el cliente UAExpert en la estación de ingeniería(PC) y un cliente OPC-UA en la tarjeta BBB. La distribución física se representa en la Figura 60,

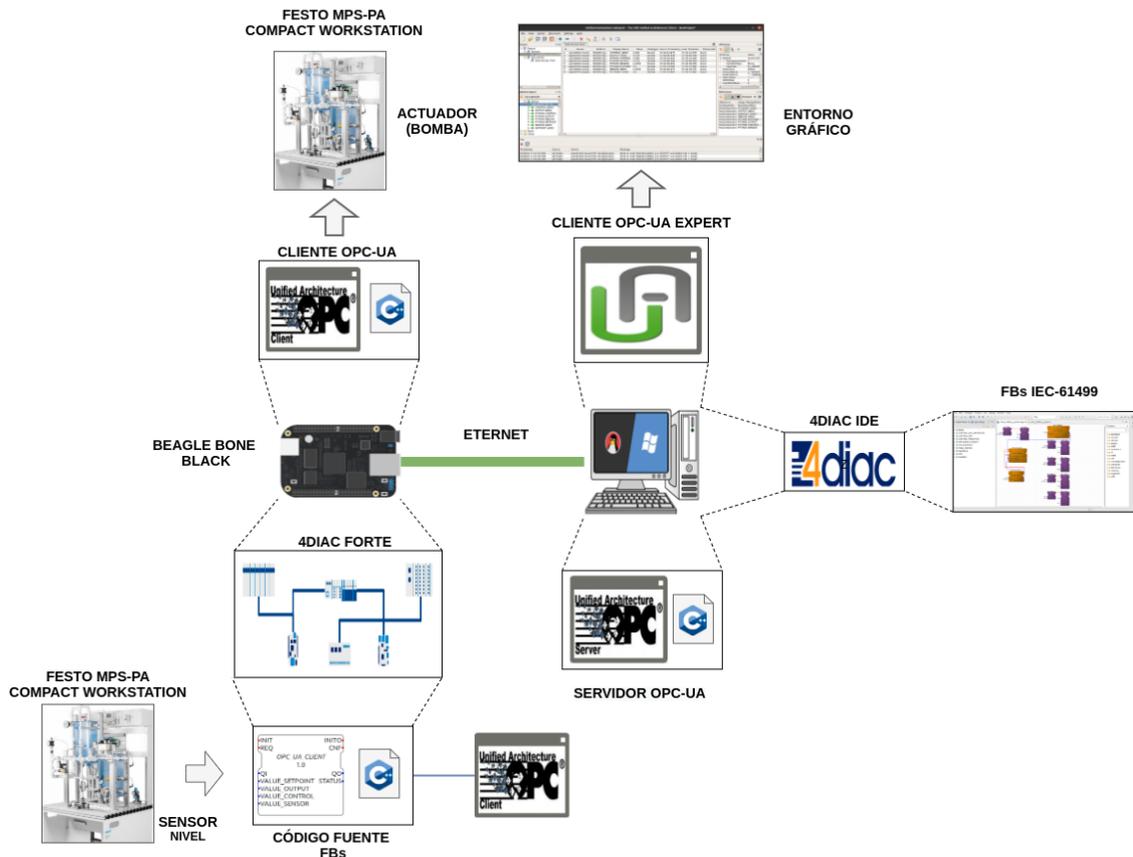


Figura 60: Simple Servidor OPC UA.

En la Figura 61 se indica la correcta comunicación entre el servidor OPC UA, FB cliente OPC UA y cliente UAExpert, el servidor OPC UA tiene la capacidad de recibir datos cualesquier cliente OPC-UA y al mismo tiempo los pone a disposición para que estos datos sean leídos por otro cliente OPC-UA.

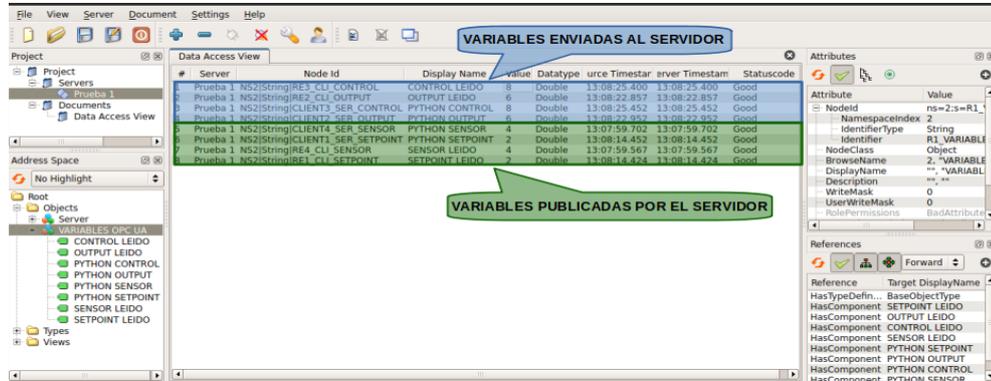


Figura 61: UAExpert Cliente OPC-UA.

■ Prueba de funcionamiento 1.

Para verificar el funcionamiento del GPC, se realizó una prueba de funcionamiento 1 y se asignó un valor de 4.5 litros al setpoint, la señal de salida ( $y(t)$ ) alcanza el valor deseado, la señal de control ( $u(t)$ ) no tiene un incremento brusco debido a las restricciones en el incremento de control ( $\Delta u(t)$ ) el cual no supera el valor de 0.1 y -0.1 cuando existe un incremento o un decremento en el setpoint respectivamente, además los valores de la señal de salida ( $y(t)$ ) y control ( $u(t)$ ) no sobrepasan los valores de 5.5 y 13.5 unidades respectivamente, el comportamiento de esta prueba se indica en las Figuras 62, 63 y 64.



Figura 62: Prueba de funcionamiento 1.

```

k= 8 Referencia= 4.5 Ym= 0.0128708 max_inc_u= 0.081752 CONTROL= 0.57379
ejec: Adafruit_BBIO: pwm_set_duty_cycle: P8_13 0.128708 OK
k= 9 Referencia= 4.5 Ym= 0.0170535 max_inc_u= 0.0816527 CONTROL= 0.655443
ejec: Adafruit_BBIO: pwm_set_duty_cycle: P8_13 0.170535 OK
k= 10 Referencia= 4.5 Ym= 0.0217883 max_inc_u= 0.0815435 CONTROL= 0.736986
ejec: Adafruit_BBIO: pwm_set_duty_cycle: P8_13 0.217883 OK

```

Figura 63: Variables del controlador.

```

k= 395 Referencia= 4.5 Ym= 4.5708 max_inc_u= -0.00131565 CONTROL= 10.9342
ejec: Adafruit_BBIO: pwm_set_duty_cycle: P8_13 45.708035 OK
k= 396 Referencia= 4.5 Ym= 4.57135 max_inc_u= -0.00132485 CONTROL= 10.9329
ejec: Adafruit_BBIO: pwm_set_duty_cycle: P8_13 45.713512 OK
k= 397 Referencia= 4.5 Ym= 4.57188 max_inc_u= -0.00133365 CONTROL= 10.9316
ejec: Adafruit_BBIO: pwm_set_duty_cycle: P8_13 45.718792 OK

```

Figura 64: Variables del controlador.

- Prueba de funcionamiento 2.

Para la prueba de funcionamiento 2, se asignó un valor de 6.0 litros al setpoint, la señal de salida ( $y(t)$ ) no alcanza el valor deseado debido a la restricción en la señal de salida ( $y(t)$ ) la cual no debe superar el valor de 5.5 litros, además la señal de control ( $u(t)$ ) no supera el valor 13.5 unidades y no existe un incremento brusco en la señal de salida ( $y(t)$ ) debido a las restricciones en el incremento de control ( $\Delta u(t)$ ) que no supera el valor de 0.1 y -0.1 cuando existe un incremento o un decremento en el setpoint respectivamente, el comportamiento de esta prueba se indica en las Figuras 65, 66, 67 y 68.



Figura 65: Prueba de funcionamiento 2.

```

k= 12 Referencia= 6 Ym= 0.0401079 max_inc_u= 0.1 CONTROL= 1.1
ejec: Adafruit_BBIO: pwm_set_duty_cycle: P8_13 0.401079 OK
k= 13 Referencia= 6 Ym= 0.047844 max_inc_u= 0.1 CONTROL= 1.2
ejec: Adafruit_BBIO: pwm_set_duty_cycle: P8_13 0.478440 OK
k= 14 Referencia= 6 Ym= 0.0562084 max_inc_u= 0.1 CONTROL= 1.3
ejec: Adafruit_BBIO: pwm_set_duty_cycle: P8_13 0.562084 OK

```

Figura 66: Variables del controlador.

```

k= 339 Referencia= 6 Ym= 5.5 max_inc_u= 0 CONTROL= 13.0714
ejec: Adafruit_BBIO: pwm_set_duty_cycle: P8_13 54.999996 OK
k= 340 Referencia= 6 Ym= 5.5 max_inc_u= 0.000128162 CONTROL= 13.0715
ejec: Adafruit_BBIO: pwm_set_duty_cycle: P8_13 54.999996 OK
k= 341 Referencia= 6 Ym= 5.5 max_inc_u= -0.00012987 CONTROL= 13.0714
ejec: Adafruit_BBIO: pwm_set_duty_cycle: P8_13 55.000004 OK

```

Figura 67: Variables del controlador.

```

k= 288 Referencia= 6 Ym= 5.47996 max_inc_u= 5e-09 CONTROL= 13.5
ejec: Adafruit_BBIO: pwm_set_duty_cycle: P8_13 54.799564 OK
k= 289 Referencia= 6 Ym= 5.48362 max_inc_u= 5e-09 CONTROL= 13.5
ejec: Adafruit_BBIO: pwm_set_duty_cycle: P8_13 54.836231 OK
k= 290 Referencia= 6 Ym= 5.48722 max_inc_u= 5e-09 CONTROL= 13.5
ejec: Adafruit_BBIO: pwm_set_duty_cycle: P8_13 54.872227 OK

```

Figura 68: Variables del controlador.

## 4.2. Discusión.

Para el análisis estadístico de la ecuación del sistema

$$y(t) = 0,429973(1 - e^{-\frac{t}{21,6245}})$$

se utilizó el software de programación RStudio para encontrar y verificar los valores de  $K$  y  $\tau$ .

Tabla 6: Análisis resultados regresión.

Parámetros	Estimados	Error Estandar	Valor t	Pr(< t )
$K$	4.300e-01	2.777e-04	1548.5	<2e-16
$\tau$	2.162e+01	7.670e-02	281.9	<2e-16

En la Tabla 5 se indica los resultados mas significativos del análisis de regresión, los valor de probabilidad Pr para  $K$  y  $\tau$  son menores que el nivel de significancia 0,05 por

lo cual se rechaza la hipótesis nula la cual afirma que los parámetros  $K$  y  $\tau$  son nulos. Además se tiene un error estándar residual de 0,002972 en 194 grados de libertad. Para visualizar el comportamiento de la señal de salida de la planta se realizó dos experimentos con un setpoint igual a 5 litros, se capturaron los datos y se elaboró un gráfico de cajas como se indica en la Figura 69, donde se observa que el comportamiento de las señales en los dos experimentos son prácticamente iguales.

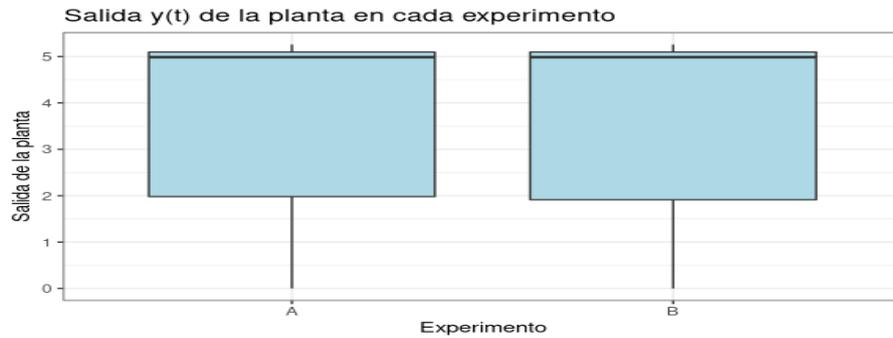


Figura 69: Experimento de setpoint=5 litros .

Para verificar de los beneficios del controlador GPC con restricciones se desarrolló dos experimentos, en primer lugar se implementó un controlador GPC sin restricciones como se indica en la Figura 70. El incremento de control ( $\Delta u(k)$ ) alcanza un valor de 0,3257 en un instante  $k$  seleccionado, el cual exige al actuador reduciendo el ciclo de vida del mismo, además se observa que la señal de salida ( $y(k)$ ) alcanza valores de 11 y 12 litros los cuales están fuera de rango del tanque provocando daños a la planta y pérdidas económicas.

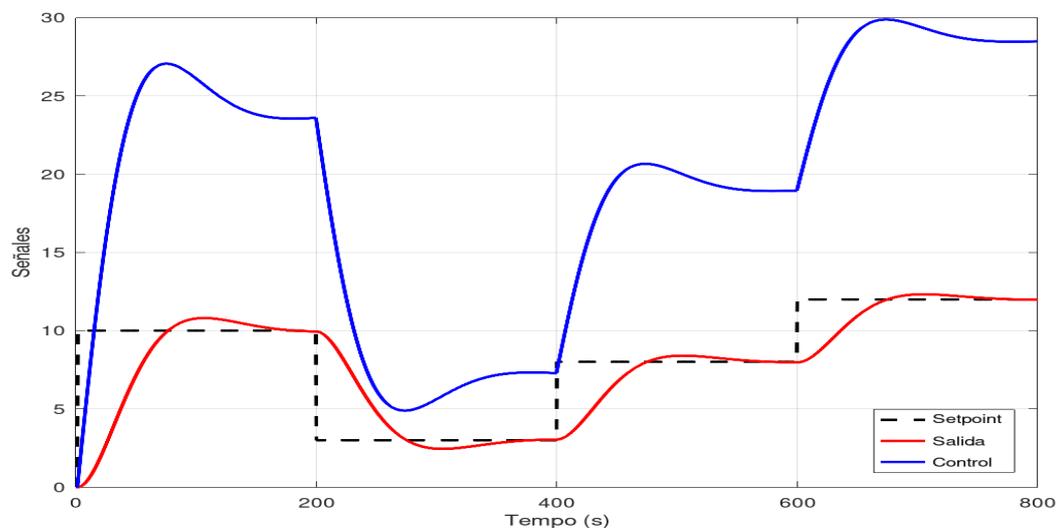


Figura 70: Controlador GPC sin restricciones.

```

k= 10 Referencia= 6 Ym= 0.171325 max_inc_u= 0.331728 u[k]=3.06547
ejec: Adafruit_BBIO: pwm_set_duty_cycle: P8_13 1.713248 OK
k= 11 Referencia= 6 Ym= 0.211154 max_inc_u= 0.328843 u[k]=3.39431
ejec: Adafruit_BBIO: pwm_set_duty_cycle: P8_13 2.111545 OK
k= 12 Referencia= 6 Ym= 0.254448 max_inc_u= 0.325785 u[k]=3.7201
ejec: Adafruit_BBIO: pwm_set_duty_cycle: P8_13 2.544481 OK

```

Figura 71: Variables del controlador.

Para evitar los inconvenientes antes mencionados se implementó las restricciones en el controlador GPC. El incremento de control ( $\Delta u(k)$ ) ya no alcanza el valor de 0,3257 en el instante  $k$ , ahora está restringido a un valor de 0,1000, evitando así el desgaste prematuro del actuador, además se observa que la señal de salida ( $y(k)$ ) no alcanza los valores de 11 y 12 litros ya que las señales de salida ( $y(k)$ ) y control ( $u(k)$ ) están restringidas a valores de 5,5 litros y 13,5 unidades respectivamente.

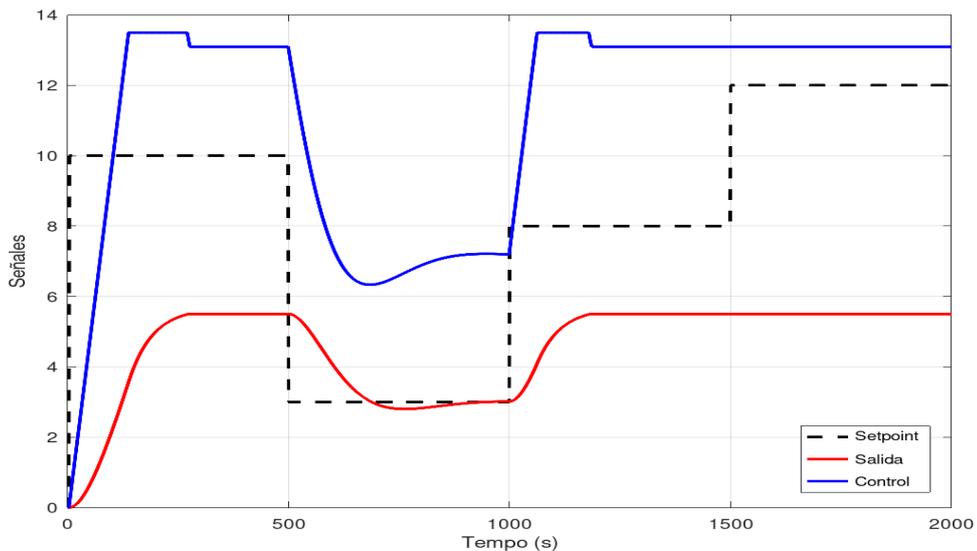


Figura 72: Controlador GPC con restricciones.

```

k= 12 Referencia= 6 Ym= 0.0401079 max_inc_u= 0.1 CONTROL= 1.1
ejec: Adafruit_BBIO: pwm_set_duty_cycle: P8_13 0.401079 OK
k= 13 Referencia= 6 Ym= 0.047844 max_inc_u= 0.1 CONTROL= 1.2
ejec: Adafruit_BBIO: pwm set duty cycle: P8_13 0.478440 OK
k= 14 Referencia= 6 Ym= 0.0562084 max_inc_u= 0.1 CONTROL= 1.3
ejec: Adafruit_BBIO: pwm_set_duty_cycle: P8_13 0.562084 OK

```

Figura 73: Variables del controlador.

En la Figura 74 se representa el error cuadrático medio (MSE) y la Raíz del Error

Cuadrático Medio (RMSE) entre la señal se salida de la planta y la señal capturada por el sensor, el valor del MSE es pequeño por lo que se concluye que el comportamiento de la planta es satisfactorio.

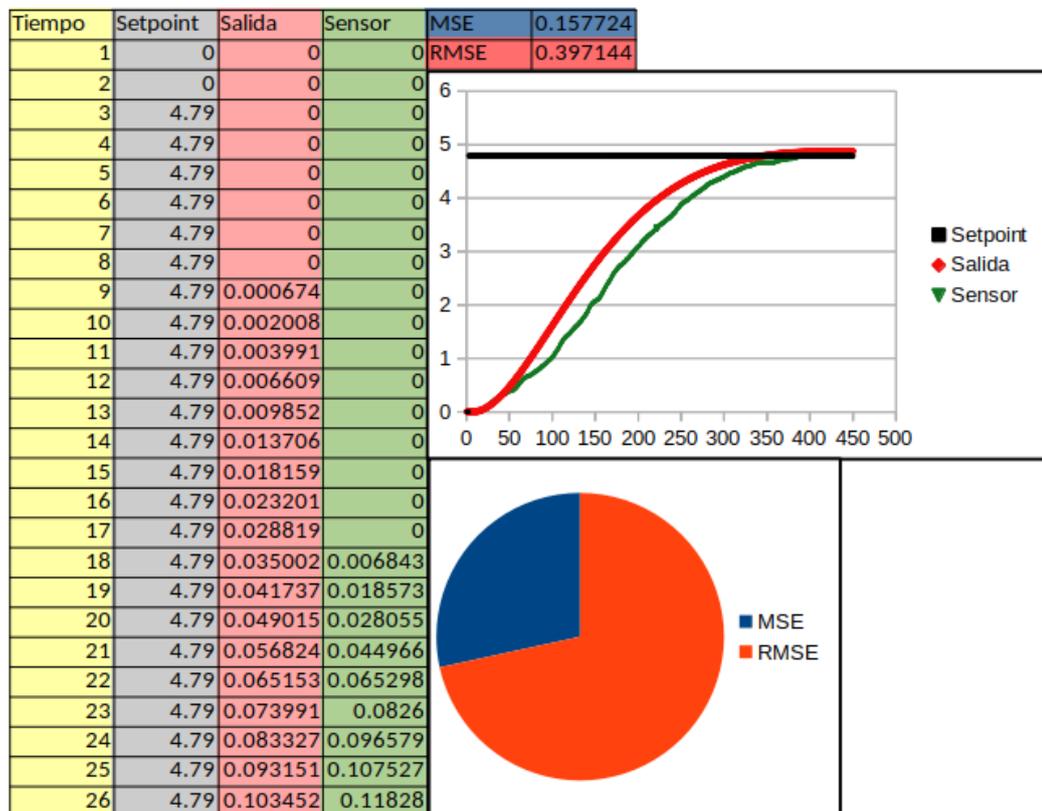


Figura 74: Error Cuadrático Medio.

## 5. CAPÍTULO V

### CONCLUSIONES, RECOMENDACIONES

#### 5.1. Conclusiones.

- Se determinó que la función de transferencia del proceso de nivel necesaria para realizar el controlador predictivo se puede determinar mediante dos métodos: el primero utilizando ecuaciones diferenciales lineales basados en principios de la física clásica y el segundo mediante datos experimentales de la respuesta al escalón de un sistema de primer orden, los dos métodos utilizados son igualmente aceptados para la implementación del controlador GPC, en nuestro caso se utilizó el primer método para verificar el orden de la función de transferencia, posteriormente se utilizó el segundo método encontrando los valores de  $\tau$  y  $K$  mediante un experimento de entrada escalón unitario a la planta y reemplazando estos valores en la ecuación de una función de transferencia de primer orden.
- Se logró implementar los algoritmos de control necesarios para la aplicación en los FB descargados en el runtime 4DIAC-FORTE que fueron diseñados en el entorno de desarrollo 4DIAC-IDE. El algoritmo de control GPC con restricciones implementado en el proceso de nivel se ejecutó correctamente alcanzando los valores deseados de setpoint y no sobrepasando las restricciones implementadas en las pruebas de funcionamiento del sistema, demostrando así la factibilidad de utilizar FB basados en la norma IEC-61499 para implementar algoritmos de control en procesos industriales.
- Se comprobó que las restricciones implementadas en el controlador GPC son necesarias para un mejor desempeño de la planta. La restricción en el incremento de control ayuda a no forzar en demasía al actuador (motor) manteniendo así el ciclo de vida del mismo, además las restricciones en las señales de salida y control ayudan en aspectos de seguridad al personal supervisor del proceso evitando así posibles daños personales, materiales y pérdidas económicas por fallos en el funcionamiento del sistema.
- De acuerdo con los objetivos planteados, podemos concluir que el Entorno de Desa-

rrrollo 4DIAC-IDE y el runtime FORTE han demostrado ser herramientas confiables para la implementación de sistema de control de lazo cerrado en tiempo real aplicando la normativa IEC-61499 en plantas de procesos industriales didácticos. La implementación del sistema de control GPC de nivel en la célula analógica FESTO® MPS-PA Compact Workstation se considera un paso más en el objetivo de demostrar los beneficios que ofrece el estándar IEC-61499 y el protocolo de comunicación OPC UA, además los resultados alcanzados indican que se puede continuar con las investigaciones con el fin de aprovechar todo el potencial que el estándar IEC-61499 y el protocolo de OPC UA ofrece a la industria.

## **5.2. Recomendaciones**

- Para utilizar el segundo método de identificación de la función de transferencia mediante datos experimentales de la respuesta al escalón de un sistema, primeramente debemos identificar el orden del sistema para lo cual se recomienda utilizar el primer método y encontrar la función de transferencia mediante ecuaciones diferenciales lineales.
- Utilizar un lenguaje de programación de alto nivel para realizar las primeras pruebas de funcionamiento del controlador GPC con restricciones, el algoritmo posteriormente serán programado en los FB que utiliza un lenguaje de programación orientado a objetos C++.
- En primer lugar implementar un controlador GPC sin restricciones, una vez verificado el correcto funcionamiento del controlador, implementar las restricciones necesarias en el controlador GPC.
- Utilizar el cliente OPC UA Expert el cual tiene y una interfaz gráfica amigable para verificar el correcto funcionamiento en conjuntos de la normativa IEC 61499, el controlador GPC con restricciones y el protocolo de comunicación OPC UA.

## 6. BIBLIOGRAFÍA

- [1] G.-J. Cheng, L.-T. Liu, X.-J. Qiang, and Y. Liu, “Industry 4.0 development and application of intelligent manufacturing,” in *2016 international conference on information system and artificial intelligence (ISAI)*, pp. 407–410, IEEE, 2016.
- [2] B. Ślusarczyk, “Industry 4.0: Are we ready?,” *Polish Journal of Management Studies*, vol. 17, 2018.
- [3] I. E. Commission, “International electrotechnical commission std (2005) iec 61499: Function blocks, part 1-4.” Available: <http://www.iec.ch>.
- [4] R. W. L. Alois Zoitl, *Modelling Control Systems Using IEC 61499*. IET Control Engineering Series 95, The Institution of Engineering and Technology, 2nd ed. 2014 ed., 2014.
- [5] W. M. a. Matthias Damm, Stefan-Helmut Leitner, *OPC unified architecture*. Springer-Verlag Berlin Heidelberg, 1 ed., 2009.
- [6] C. B. E. F. Camacho, *Model Predictive Control*. Advanced textbooks in control and signal processing, Springer, 1999.
- [7] G. Lyu and R. W. Brennan, “Towards iec 61499 based distributed intelligent automation: A literature review,” *IEEE Transactions on Industrial Informatics*, 2020.
- [8] M. V. García, F. Pérez, I. Calvo, and G. Morán, “Building industrial cps with the iec 61499 standard on low-cost hardware platforms,” in *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA)*, pp. 1–4, IEEE, 2014.
- [9] S. Spinelli, A. Cataldo, G. Pallucca, and A. Brusafferri, “A distributed control architecture for a reconfigurable manufacturing plant,” in *2018 IEEE Industrial Cyber-Physical Systems (ICPS)*, pp. 673–678, IEEE, 2018.
- [10] S. Patil, V. Vyatkin, and B. McMillin, “Implementation of freedm smart grid distributed load balancing using iec 61499 function blocks,” in *IECON 2013-39th Annual Conference of the IEEE Industrial Electronics Society*, pp. 8154–8159, IEEE, 2013.

- [11] V. Vlad, C. Buzduga, and C. Ciufudean, “An approach to developing power grid control systems with iec 61850, iec 61499 and holonic control,” *WSEAS Trans Syst*, vol. 13, pp. 503–9, 2014.
- [12] B. Yao, Z. Zhou, L. Wang, W. Xu, J. Yan, and Q. Liu, “A function block based cyber-physical production system for physical human–robot interaction,” *Journal of manufacturing systems*, vol. 48, pp. 12–23, 2018.
- [13] M. Jovanović, S. Zupan, and I. Prebil, “Holonic control approach for the “green”-tyre manufacturing system using iec 61499 standard,” *Journal of Manufacturing Systems*, vol. 40, pp. 119–136, 2016.
- [14] S. Cavalieri, S. Mulè, and M. G. Salafia, “Enabling opc ua and onem2m interworking,” in *2020 IEEE International Conference on Industrial Technology (ICIT)*, pp. 278–285, IEEE, 2020.
- [15] J. Kim, G. Jo, and J. Jeong, “A novel cpps architecture integrated with centralized opc ua server for 5g-based smart manufacturing,” *Procedia Computer Science*, vol. 155, pp. 113–120, 2019.
- [16] H. F. Bano, C. A. Garcia, A. Cabrera, E. X. Castellanos, J. E. Naranjo, and M. V. Garcia, “Low-cost embedded system for shop floor communications and control based on opc-ua,” in *The International Conference on Advances in Emerging Trends and Technologies*, pp. 3–12, Springer, 2019.
- [17] M. V. García, E. Irisarri, F. Pérez, E. Estévez, and M. Marcos, “An open cpps automation architecture based on iec-61499 over opc-ua for flexible manufacturing in oil&gas industry,” *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 1231–1238, 2017.
- [18] M. García, “Desarrollo de comunicación opc-ua en sistemas cpps de bajo costo,” in *Congreso de Ciencia y Tecnología ESPE*, vol. 10, pp. 220–225, 2015.
- [19] W. Dai, Y. Song, Z. Zhang, P. Wang, C. Pang, and V. Vyatkin, “Modelling industrial cyber-physical systems using iec 61499 and opc ua,” in *2018 IEEE 16th International Conference on Industrial Informatics (INDIN)*, pp. 772–777, IEEE, 2018.
- [20] D. Honc, R. Sharma, A. Abraham, F. Dušek, and N. Pappa, “Teaching and practicing model predictive control,” *IFAC-PapersOnLine*, vol. 49, no. 6, pp. 34–39, 2016.

- [21] S. Vazquez, J. Rodriguez, M. Rivera, L. G. Franquelo, and M. Norambuena, “Model predictive control for power converters and drives: Advances and trends,” *IEEE Transactions on Industrial Electronics*, vol. 64, no. 2, pp. 935–947, 2016.
- [22] T. Faulwasser, L. Grüne, M. A. Müller, *et al.*, “Economic nonlinear model predictive control,” *Foundations and Trends® in Systems and Control*, vol. 5, no. 1, pp. 1–98, 2018.
- [23] A. Mesbah, “Stochastic model predictive control: An overview and perspectives for future research,” *IEEE Control Systems Magazine*, vol. 36, no. 6, pp. 30–44, 2016.
- [24] A. Cataldo, I. C. Bertolotti, and R. Scattolini, “Model predictive control tools for evolutionary plants,” in *Factories of the Future*, pp. 39–56, Springer, Cham, 2019.
- [25] C. A. Garcia, E. X. Castellanos, J. Buele, J. Espinoza, D. Lanas, C. Beltran, and M. V. Garcia, “Mpc under iec-61499 using low-cost devices for oil pipeline system,” in *2018 IEEE 16th International Conference on Industrial Informatics (INDIN)*, pp. 659–664, IEEE, 2018.
- [26] R. H. Gaviño, *Introducción a los Sistemas de Control: Conceptos, Aplicaciones y Simulación con MATLAB*. Prentice Hall, 2010.
- [27] C. P. Ruiz, *Calculo Vectorial (Spanish Edition)*. Prentice Hall, 1996.
- [28] M. L. W. Gene F. Franklin, J. David Powell, *Digital Control of Dynamic Systems, 3rd Edition*. 3 ed., 1997.
- [29] M. J. Best, *Quadratic Programming with Computer Programs*. CRC, 2017.
- [30] 4DIAC. Available at <http://www.eclipse.org/4diac/>.
- [31] W. M. a. Matthias Damm, Stefan-Helmut Leitner, *OPC unified architecture*. Springer-Verlag Berlin Heidelberg, 1 ed., 2009.
- [32] OPCFOUNDATION. Available at <https://opcfoundation.org/>.
- [33] FESTO. Available at <https://www.festo-didactic.com/>.
- [34] S. Bustos, “Desarrollo de un sistema de control industrial basado en el estándar iec-61499,” 2017.

## 7. ANEXOS

### Lista de símbolos

$(M)_{ij}$  elemento  $ij$  de la matriz  $M$

$(\cdot)_T$  transpuesta de  $(\cdot)$

$(v)_i$   $i^{th}$  -elemento del vector  $v$

$\Delta$   $1 - z^{-1}$

$\hat{\cdot}$  valor esperado

$\hat{x}(t + j|t)$  valor esperado de  $x(t + j)$  con información disponible en el instante  $t$

$\|\mathbf{v}\|_Q$   $\sqrt{\mathbf{v}^T Q \mathbf{v}}$

$\mathbf{0}$  matriz de dimensiones apropiadas con todas las entradas iguales a cero

$\mathbf{0}_{p \times q}$  ( $p \times q$ ) matriz con todas las entradas iguales a cero

$\mathbf{1}$  vector de columna con todas las entradas iguales a uno

$\mathbf{I}$  matriz identidad de dimensiones apropiadas

$\mathbf{I}_{n \times n}$  ( $n \times n$ ) matriz identidad

$s$  variable compleja utilizada en la transformada de Laplace

$z$  operador de desplazamiento hacia adelante y variable compleja utilizada en la transformación  $z$

$z^{-1}$  Operador de cambio hacia atrás

### Parámetros y variables del controlador

$N_1$  menor valor del horizonte de predicción

$N_2$  mayor valor del horizonte de predicción

$N_2$  número de puntos del horizonte de predicción ( $N = N_2 - N_1$ )

$N_2$  horizonte de control ( $N_u$ )

- $\delta$  factor de ponderación para seguimiento trayectoria
- $\lambda$  factor de ponderación para incrementos de control
- $\mathbf{G}_{N_{123}}$  Matriz de predicción del GPC con horizontes  $(N_1, N_2, N_3)$
- $\mathbf{f}$  vector de respuesta libre prevista
- $\mathbf{u}$  vector de futuros incrementos de control para el horizonte de control
- $\mathbf{w}$  vector de referencias futuras
- $\mathbf{y}$  vector de salidas previstas para el horizonte de predicción
- $\tilde{\mathbf{A}}(z^{-1})$  polinomio  $\tilde{\mathbf{A}}(z^{-1})$  multiplicado por  $\Delta$

## Lista de acrónimos

- 4DIAC** Framework for Distributed Industrial Automation and Control, Framework para la Automatización y Control Industrial Distribuido
- AD** Analog-to-Digital Converter, Conversor Análogo Digital
- CARIMA** Controlled Auto-Regressive and Integrated Moving Average, Controlada Auto Regresiva y Media Móvil Integrada
- CARMA** Controller Auto-Regressive Moving-Average, Controlador de Media Móvil Auto Regresiva
- CCDE** Constant Coefficient Difference Equation, Ecuación de Diferencia de Coeficiente Constante
- CCDE** Constant-Coefficient Difference equation, Ecuación de Diferencia de Coeficiente Constante
- CPS** Cyber Physical Systems, Sistemas Ciber-Físicos
- CTO** Total Cost of Ownership, Costo Total de Propiedad
- DA** Digital-to-Analog Converter, Conversor Digital Análogo
- DGI** Distributed Grid Intelligence, Red Inteligente Distribuida

**DMC** Dynamic Matrix Control, Control de Matriz Dinámica

**DSC** Distributed Control System, Sistema de Control Distribuidos

**FB** Function Block, Bloque de Función

**FBD** Function Block Diagram, Diagrama de bloque de funciones

**FORTE** 4DIAC Runtime Environment, Ambiente de ejecución para 4DIAC

**GCC** GNU Compiler Collection, Colección de Compiladores GNU

**GPC** Generalized Predictive Control, Control Predictivo Generalizado

**GQPA** General Quadratic Programming Algorithm, Algoritmo General de Programación Cuadrática

**HMI** Human Machine Interface, Interfaz Hombre Máquina

**HRC** Human Robot Collaboration, Colaboración Humano Robot

**IC** Information Components, Componentes de Información

**IDE** Integrated Development Environment, Entorno de Desarrollo Integrado

**IEC** International Electrotechnical Commission, Comisión Electrotécnica Internacional

**IL** Instruction List, Lista de Instrucciones

**IMC** Intelligent Mechatronic Component, Componente Mecatrónico Inteligente

**IoT** Internet of Things, Internet de las Cosas

**IP** Internet protocol, Protocolo de Internet

**LD** Ladder Diagram, Lenguaje escalera

**M2M** Machine to machine communication, Comunicación Máquina a Máquina

**MES** Manufacturing Execution System, Sistemas de Ejecución de Producción

**MPC** Model Predictive Control, Control Predictivo basado en Modelo

**MPLv2** Mozilla Public License v2.0, Licencia Pública de Mozilla v2.0

**NFC** Near Field Communication

**OLE** Object Linking and Embedding, Vinculación e Vincrustación de Objetos

**OPC** Open Platform Communications,Plataforma de comunicacion Abierta

**OPC DA** Open Platform Communications Data Access,Plataforma de comunicacion Abierta Acceso de Datos

**pHRI** Physical Human–Robot Interaction, Interacción Física Humano-Robot

**PID** Proportional Integral Derivative, Proporcional Integral y Derivativo

**PLC** Programmable Logic Controller, Controlador lógico programable

**PWM** Pulse Width Modulation, Modulación por Ancho de Pulso

**QP** Quadratic Programming, Programación Cuadrática

**SCADA** Supervisory Control And Data Acquisition, Supervisión, Control y Adquisición de Datos

**SDK** Software Development Kit, Equipo de Desarrollo de Software

**SFC** Sequential Function Chart, Bloques de función secuenciales

**SIFB** Service Interface Function Block, Bloque de Función de Interfaz de Servicio

**SISO** Single-Input Single-Output, Una Entra Una Salida

**SOA** Service Oriented Architecture, Arquitectura Orientada a Servicios

**ZOH** Zero-Order-Hold, Retención de Orden Cero

### **7.0.1. Incorporación de 4DIAC-FORTE en la BBB.**

4DIAC-FORTE es el entorno de ejecución(runtime) instalado en el dispositivo de control para nuestro caso la tarjeta BBB, la forma en que se implementan los FBs creados en FORTE es a través de la implementación de sus archivos fuente en la carpeta que contiene los archivos base de FORTE.

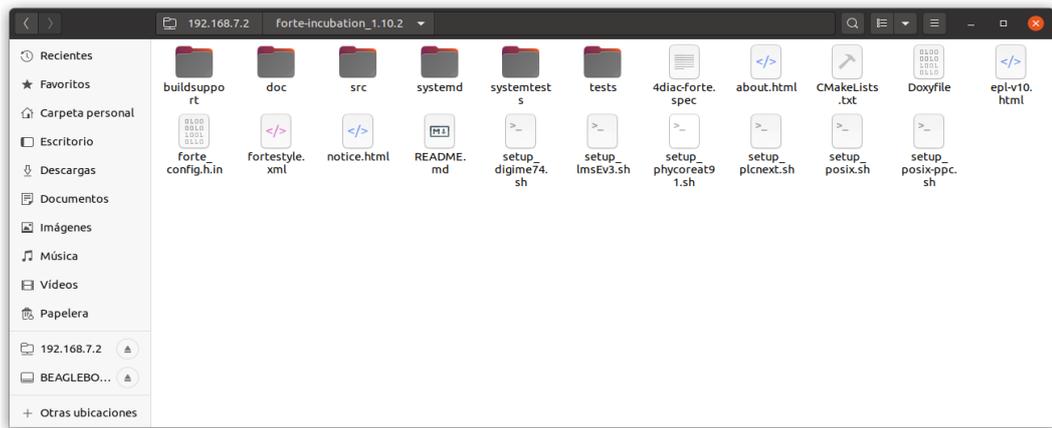


Figura 75: Archivos FORTE ubicados en la tarjeta Beaglebone Black.

Para utilizar los FBs desarrollados junto a FORTE es necesario crear un nuevo módulo en los archivos base de FORTE, donde se adicionarán sus archivos exportados. Para esto es necesario ingresar en la carpeta contenedora de FORTE y ubicarse en la dirección `/src/modules`, donde se debe crear una carpeta (en el caso de este proyecto `TESIS_FINAL`) como se ve en la Figura 76.

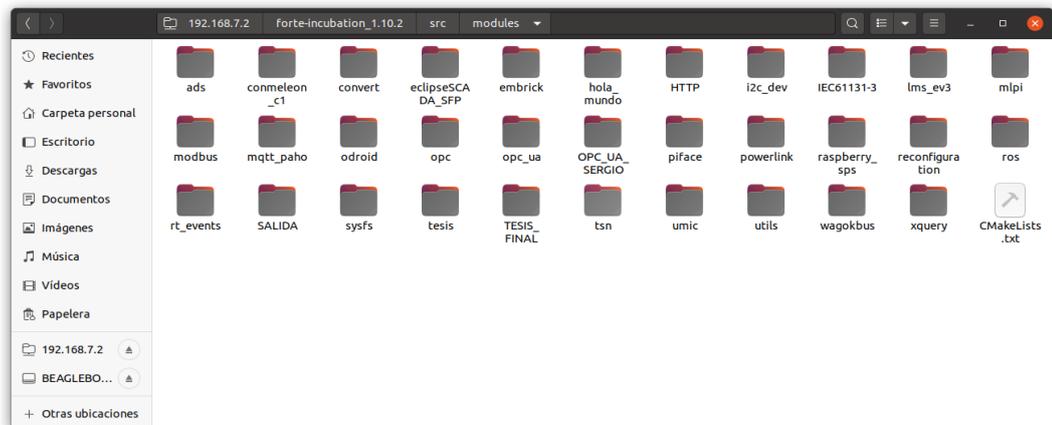


Figura 76: Carpeta `TESIS_FINAL` creada en `/src/modules`, contiene los archivos del nuevo módulo.

Dentro de la carpeta de módulos generada se debe añadir una carpeta en donde se almacenarán los archivos exportados y las librerías de usuario (en el caso de este proyecto `BlackLib`) y un archivo de texto con el nombre “`CMakeLists`” como se muestra en la Figura 77. El archivo `CMakeLists` contiene las directrices de compilación del nuevo módulo, por tal motivo debemos escribir una cabecera de información y la codificación de compilación como se indica en la Figura 78.

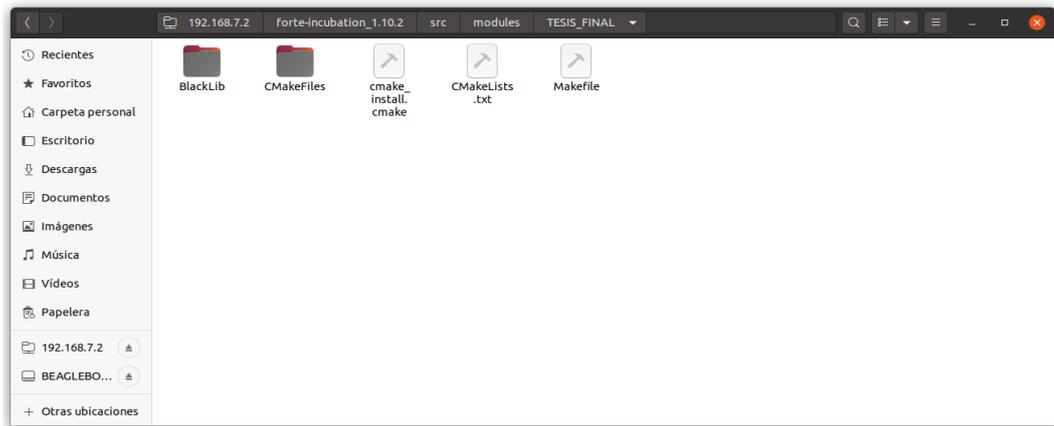


Figura 77: Carpeta TESIS\_FINAL contiene funciones y archivo CMakeLists.txt generados en la carpeta de módulo.

Las directrices de compilación `forte_add_sourcefile_h` (archivos `.h`) y `forte_add_sourcefile_hcpp` (archivos `.cpp`) deben agregarse para cada par de archivos exportados que se agregaron en la carpeta que contiene el resto de los archivos.

```

CMakeLists.txt
~/Documentos/TESSISERGIO/EXPORTADOS/AUX_TESIS
Guardar
1 #ARCHIVO CMAKELISTS
2
3 forte_add_module(TESIS_FINAL "TESIS_FINAL")
4
5 forte_add_link_directories(S(CMAKE_CURRENT_SOURCE_DIR)/BlackLib/v4_0/LIBRERIA)
6
7 forte_add_include_directories(S(CMAKE_CURRENT_SOURCE_DIR)/BlackLib/v4_0/LIBRERIA)
8
9 forte_add_subdirectory(BlackLib/v4_0/LIBRERIA)
10
11 #BLOQUE DE FUNCIONES
12 forte_add_sourcefile_h(BlackLib/v4_0/LIBRERIA/ANALOG_OUTPUT.h)
13 forte_add_sourcefile_hcpp(BlackLib/v4_0/LIBRERIA/ANALOG_OUTPUT.cpp)
14
15 forte_add_sourcefile_h(BlackLib/v4_0/LIBRERIA/ANALOG_INPUT.h)
16 forte_add_sourcefile_hcpp(BlackLib/v4_0/LIBRERIA/ANALOG_INPUT.cpp)
17
18 forte_add_sourcefile_h(BlackLib/v4_0/LIBRERIA/OPC_UA_CLIENT.h)
19 forte_add_sourcefile_hcpp(BlackLib/v4_0/LIBRERIA/OPC_UA_CLIENT.cpp)
20
21 forte_add_sourcefile_h(BlackLib/v4_0/LIBRERIA/DMC_CONTROL_RESTRICTED.h)
22 forte_add_sourcefile_hcpp(BlackLib/v4_0/LIBRERIA/DMC_CONTROL_RESTRICTED.cpp)
23
24 #LIBRERIAS
25 forte_add_sourcefile_h(BlackLib/v4_0/LIBRERIA/BlackCore.h)
26 forte_add_sourcefile_hcpp(BlackLib/v4_0/LIBRERIA/BlackCore.cpp)
27
28 forte_add_sourcefile_h(BlackLib/v4_0/LIBRERIA/Adc.h)
29 forte_add_sourcefile_hcpp(BlackLib/v4_0/LIBRERIA/Adc.cpp)
30
31 forte_add_sourcefile_h(BlackLib/v4_0/LIBRERIA/PWM.h)
32 forte_add_sourcefile_hcpp(BlackLib/v4_0/LIBRERIA/PWM.cpp)
33
34 forte_add_sourcefile_h(BlackLib/v4_0/LIBRERIA/BlackDef.h)
35 forte_add_sourcefile_h(BlackLib/v4_0/LIBRERIA/BlackErr.h)
36
37 forte_add_sourcefile_h(BlackLib/v4_0/LIBRERIA/open62541.h)
38 forte_add_sourcefile_h(BlackLib/v4_0/LIBRERIA/open62541.cpp)

```

Figura 78: Contenido del archivo CMakeLists.txt.

Una vez obtenido los archivos base de FORTE con la ayuda del CMake controlamos el proceso de compilación de los archivos.

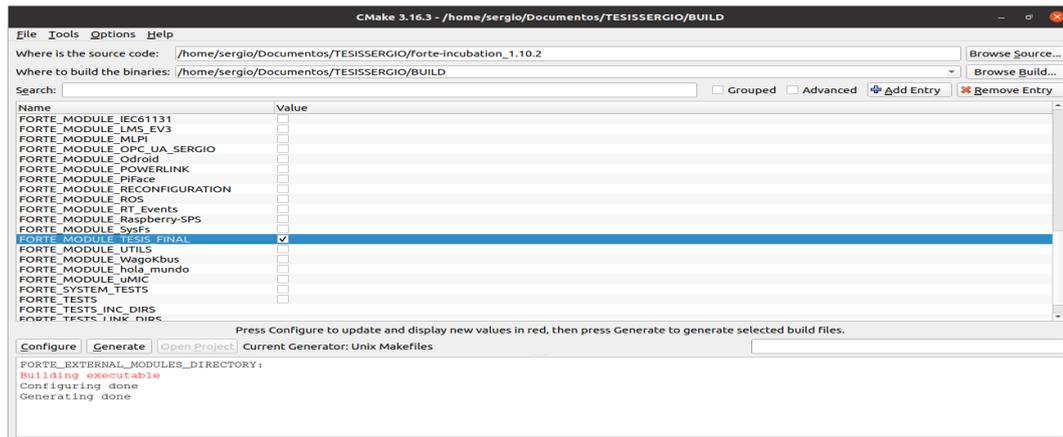


Figura 79: Contenido del archivo CMakelists.txt

## 7.0.2. Circuito electrónico de acople: protección, entradas analógicas y salidas PWM tarjeta BBB

En vista de la imposibilidad de la tarjeta BBB de ser utilizada directamente en aplicaciones industriales debido a sus limitaciones estructurales, es necesario desarrollar un circuito electrónico de acople que sirva como puente de conexión entre los circuitos de control y de potencia.

En la aplicación desarrollada se manipularon las entradas analógicas y las salidas PWM de la tarjeta BBB, las entradas analógicas admiten un rango de 0 a 1,8 Voltios y las señales PWM funcionan a 3,3 Voltios con una frecuencia de 50Hz.

En la implementación del controlador se conectará un sensor analógico con medidas en rango de 0-5 Voltios y un actuador (bomba) que funciona con un voltaje de 15 Voltios el cual será manipulado con la señal PWM[34].

- Acondicionamiento de tensión entrada analógica[34].

El circuito acondiciona el rango de tensión para una entrada analógica de la BBB que va de (0.0-1.8) Voltios a un rango de (0.0-5.0) Voltios mediante un divisor de tensión como se indica en la Figura 80.

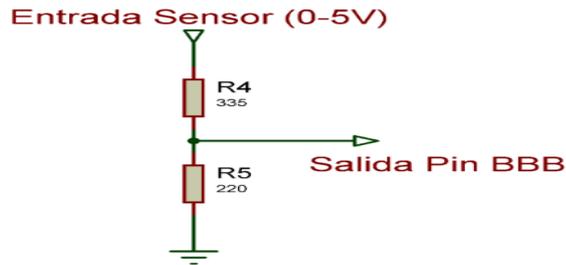


Figura 80: Circuito de acondicionamiento de tensión para la entrada analógica.

Cálculos y consideraciones:

Para el cálculo de la resistencias  $R_4$  se consideró una  $R_5 = 220\Omega$ ,  $V_{sal} = 1,8V$  y  $V_{ent} = 5V$ .

$$V_{Sal} = \left(\frac{R_5}{R_4 + R_5}\right)V_{Ent}$$

$$1,8V_{ol} = \left(\frac{220\Omega}{R_4 + 220\Omega}\right)5V_{ol}$$

$$R_4 = 335\Omega$$

■ Circuito de acondicionamiento sensor de nivel[34].

El sensor de nivel analógico ultrasónico empleado proporciona una señal estándar de 4-20 miliamperios (mA) para lo cual se realizó un circuito de acondicionamiento de señal para una conversión a voltaje de 0.0-5.0 Voltios. Se tiene los siguientes circuitos:

Seguidor de voltaje.

Proporcionando un efecto de aislamiento de la salida respecto de la señal de entrada, anulando los efectos de la “carga”.

Restador de voltaje.

El objetivo es obtener un valor de 0 Voltios cuando el voltaje en la salida del seguidor de tensión sea el valor mínimo. El circuito está representado en la Figura 81.

Cálculos y consideraciones:

Para el cálculo de la resistencias  $R_9$  se consideró una  $R_8 = 1000\Omega$ ,  $R_7 = 10000\Omega$ ,  $V_{salida} = 1,2V$ ,  $V_{entrada} = 4,4V$  y un  $V_{(5V)} = 5V$ .

$$V_{Salida} = V_{Entrada} \frac{R_9}{R_8} - V_{(5V)} \frac{R_9}{R_7}$$

$$1,2V = 4,4V \frac{R_9}{1000\Omega} - 5V \frac{R_9}{10000\Omega}$$

$$R_9 = 342\Omega$$

Amplificador no inversor de voltaje.

La finalidad es obtener una señal de voltaje de 0-5 Voltios del sensor de nivel analógico ultrasónico.

Cálculos y consideraciones:

Para el cálculo de la resistencias  $R_{11}$  se consideró una  $R_{12} = 1000\Omega$ ,  $V_{salida} = 5,0V$  y un  $V_{entrada} = 1,2V$ .

$$V_{Salida} = V_{Entrada} \left(1 + \frac{R_{12}}{R_{11}}\right)$$

$$5,0V = 1,2V \left(1 + \frac{1000\Omega}{R_{11}}\right)$$

$$R_{11} = 316\Omega$$

El circuito final de acondicionamiento del sensor del nivel está representado en la Figura 81.

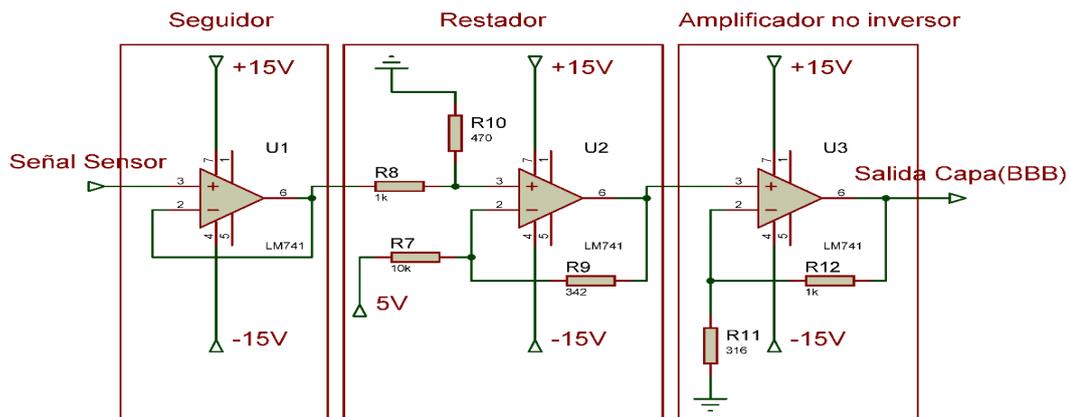


Figura 81: Circuito conversión de corriente 4-20mA a un voltaje 0-5V.

- Circuito de acondicionamiento de salida PWM[34].

El objetivo del circuito es manipular la bomba de agua de 24V con la salida PWM de la BBB la cual trabaja con un voltaje de 3.3V insuficiente para la manipulación de la bomba. La Figura 82 representa el circuito diseñado.



### 7.0.3. Secuencia del proceso de control GPC.

A continuación se indica el desempeño del GPC al aplicar un setpoint de 6 litros y se visualiza el comportamiento de las variables en las dos pantallas.

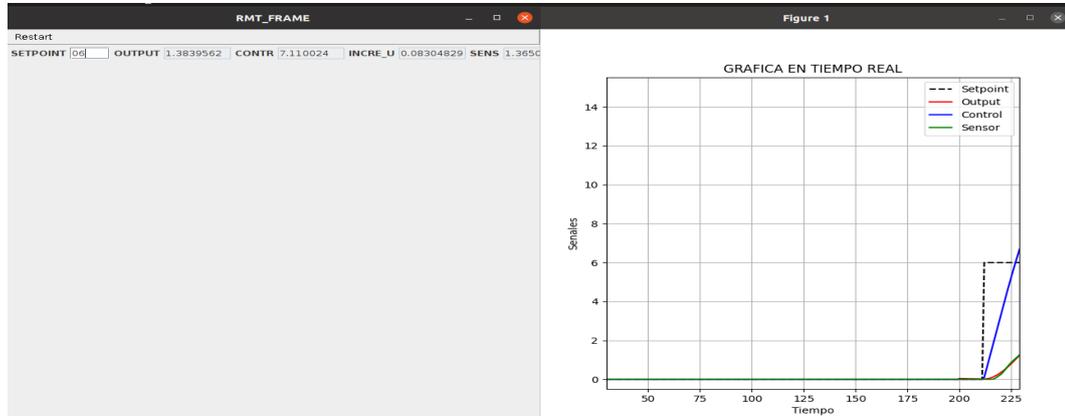


Figura 84: Secuencia del comportamiento del GPC.

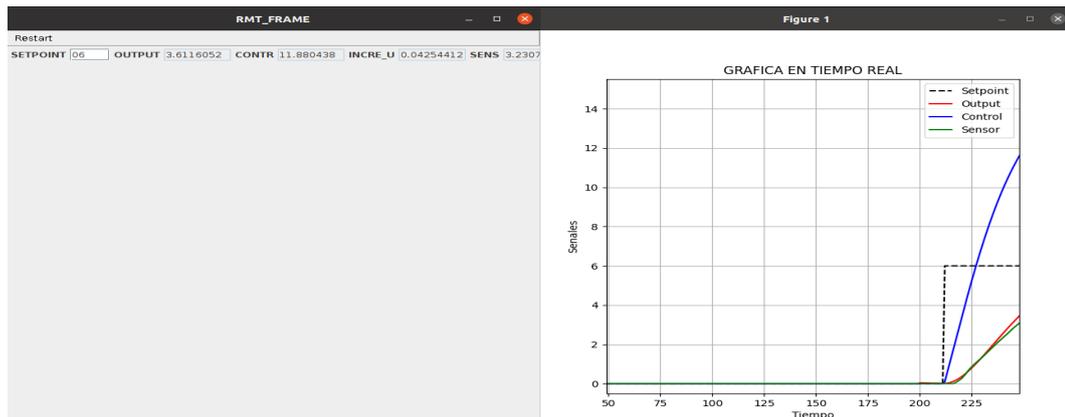


Figura 85: Secuencia del comportamiento del GPC.

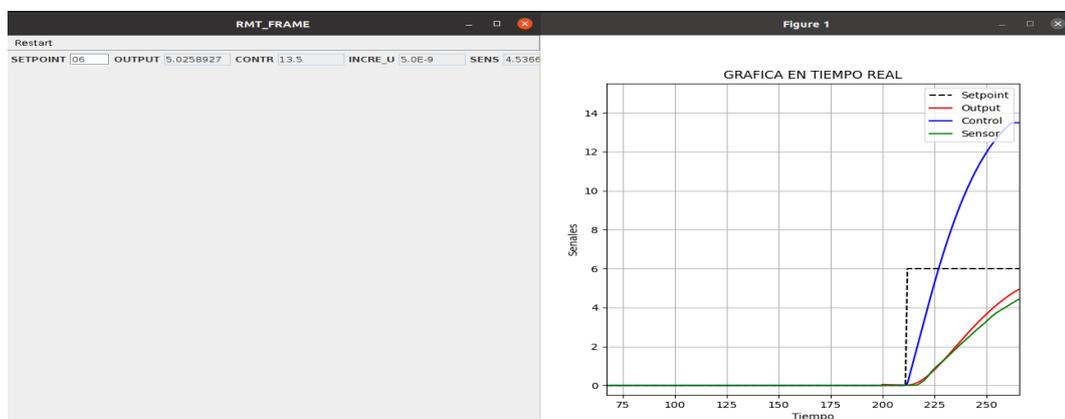


Figura 86: Secuencia del comportamiento del GPC.

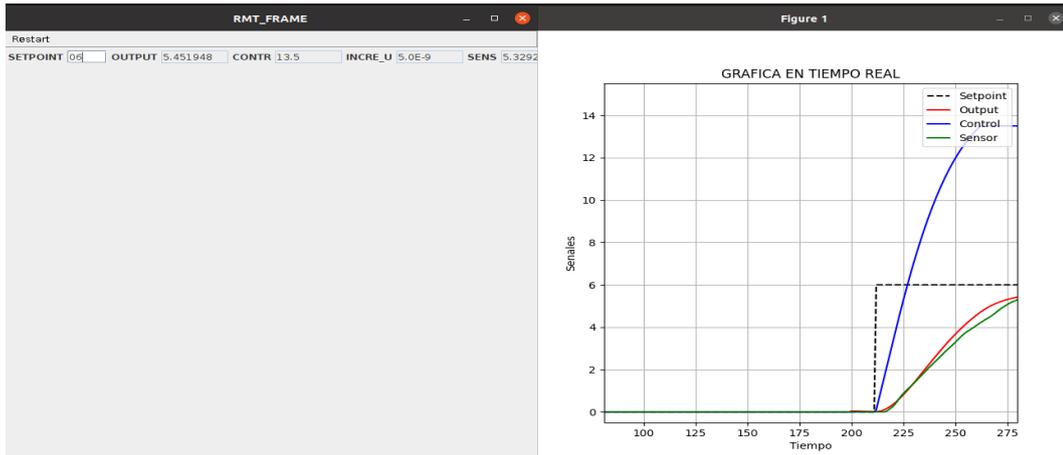


Figura 87: Secuencia del comportamiento del GPC.

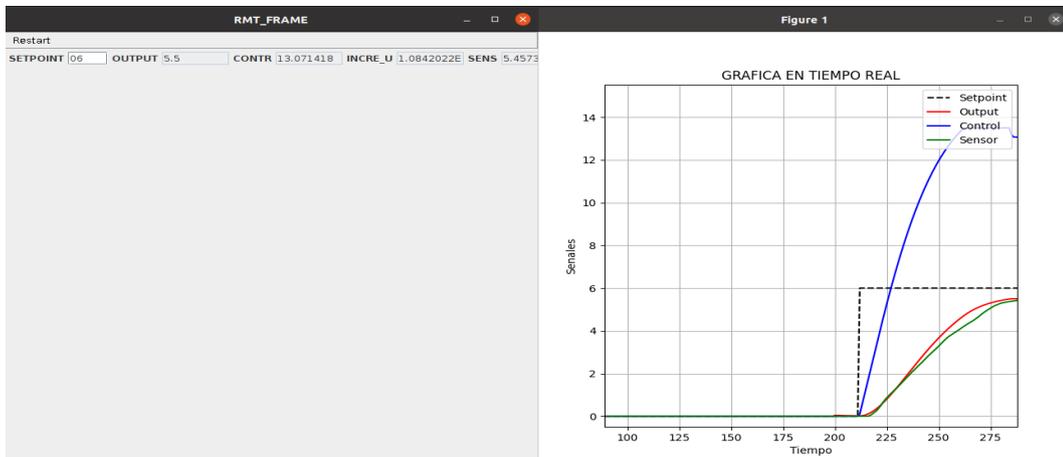


Figura 88: Secuencia del comportamiento del GPC.

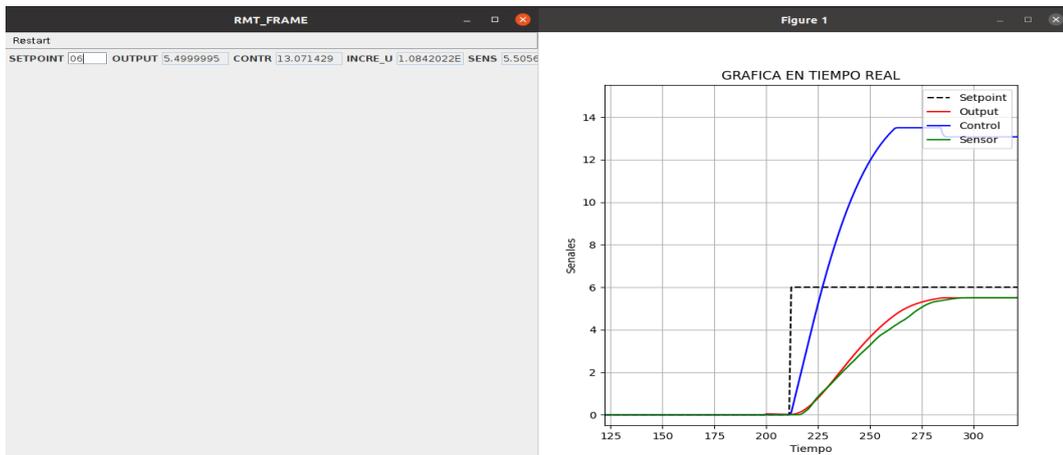


Figura 89: Secuencia del comportamiento del GPC.

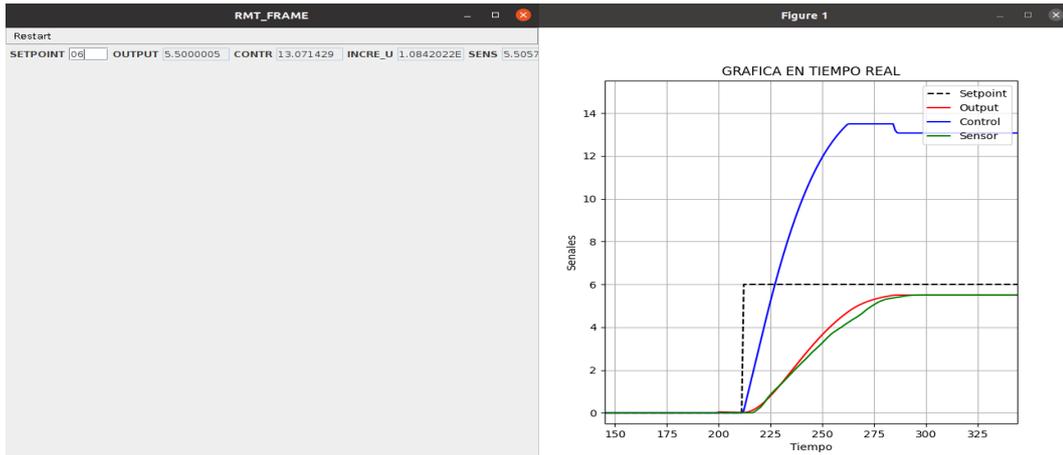


Figura 90: Secuencia del comportamiento del GPC.

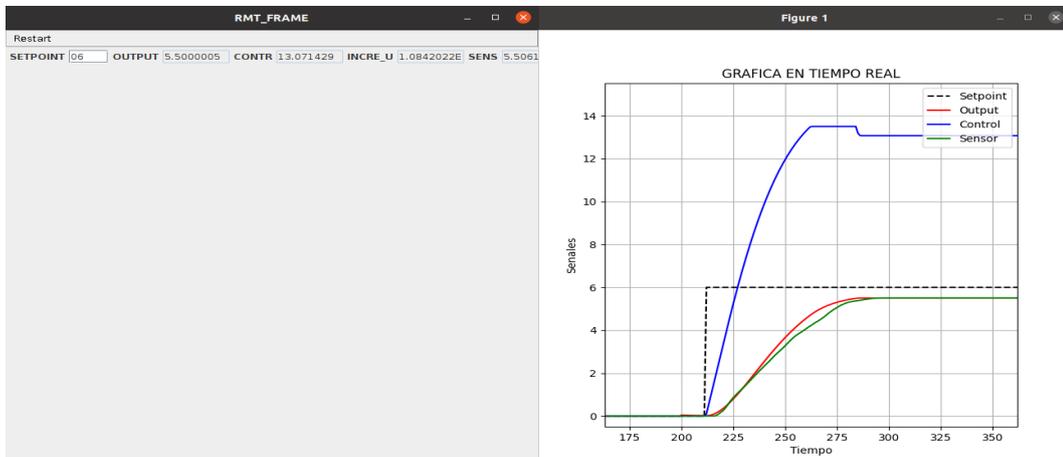


Figura 91: Secuencia del comportamiento del GPC.

#### 7.0.4. Resultados experimentales de entrada el escalón unitario.

En la tabla 7 se indica los valores capturados del experimento de entrada al escalón del sistema.

Tabla 8: Resultados experimentales.

T	Nivel	T	Nivel	T	Nivel	T	Nivel	T	Nivel	T	Nivel	T	Nivel
0	0.000	30	0.321	60	0.400	90	0.418	120	0.427	150	0.428	180	0.434
1	0.019	31	0.329	61	0.403	91	0.420	121	0.425	151	0.424	181	0.426
2	0.034	32	0.326	62	0.410	92	0.420	122	0.432	152	0.429	182	0.432
3	0.059	33	0.333	63	0.408	93	0.421	123	0.430	153	0.426	183	0.426
4	0.069	34	0.342	64	0.412	94	0.427	124	0.428	154	0.429	184	0.428
5	0.089	35	0.348	65	0.404	95	0.424	125	0.427	155	0.433	185	0.430
6	0.107	36	0.353	66	0.414	96	0.426	126	0.426	156	0.429	186	0.424
7	0.113	37	0.354	67	0.412	97	0.429	127	0.424	157	0.425	187	0.432
8	0.130	38	0.357	68	0.412	98	0.422	128	0.432	158	0.429	188	0.428
9	0.143	39	0.362	69	0.414	99	0.430	129	0.425	159	0.431	189	0.425
10	0.162	40	0.361	70	0.417	100	0.424	130	0.432	160	0.432	190	0.427
11	0.173	41	0.361	71	0.418	101	0.427	131	0.431	161	0.426	191	0.428
12	0.177	42	0.364	72	0.418	102	0.421	132	0.432	162	0.425	192	0.429
13	0.195	43	0.374	73	0.412	103	0.429	133	0.427	163	0.430	193	0.428
14	0.209	44	0.370	74	0.414	104	0.424	134	0.428	164	0.431	194	0.428
15	0.217	45	0.375	75	0.418	105	0.427	135	0.432	165	0.432	195	0.429
16	0.228	46	0.382	76	0.413	106	0.422	136	0.428	166	0.428		
17	0.230	47	0.377	77	0.417	107	0.430	137	0.428	167	0.429		
18	0.243	48	0.381	78	0.423	108	0.427	138	0.428	168	0.428		
19	0.252	49	0.381	79	0.422	109	0.424	139	0.431	169	0.432		
20	0.258	50	0.385	80	0.423	110	0.430	140	0.424	170	0.432		
21	0.261	51	0.388	81	0.415	111	0.427	141	0.424	171	0.428		
22	0.270	52	0.395	82	0.416	112	0.430	142	0.431	172	0.433		
23	0.280	53	0.393	83	0.420	113	0.424	143	0.430	173	0.426		
24	0.285	54	0.390	84	0.416	114	0.431	144	0.433	174	0.429		
25	0.295	55	0.391	85	0.421	115	0.431	145	0.429	175	0.429		
26	0.304	56	0.398	86	0.419	116	0.431	146	0.424	176	0.433		
27	0.308	57	0.395	87	0.421	117	0.426	147	0.428	177	0.431		
28	0.311	58	0.404	88	0.424	118	0.432	148	0.429	178	0.431		
29	0.319	59	0.404	89	0.426	119	0.428	149	0.425	179	0.433		