



UNIVERSIDAD  
**DE ATACAMA**

**FACULTAD TECNOLÓGICA**  
**DEPARTAMENTO DE ENERGÍAS**

**DISEÑO DE UN CONTROL PID DE NIVEL MEDIANTE EL USO DE MATLAB**

Proyecto de titulación presentado en conformidad a los requisitos para obtener el título de ingeniero de ejecución en instrumentación y automatización industrial.

Profesor guía: José Sebastián Rojas Gallardo

Oskar Kevin Andrés Villa Zuleta

Copiapó, Chile 2024

## INDICE

<b>CAPITULO I.....</b>	<b>1</b>
<b>INTRODUCCION .....</b>	<b>1</b>
<b>1.1 Generalidades .....</b>	<b>1</b>
<b>1.2. Objetivo general .....</b>	<b>2</b>
<b>1.3. Objetivos específicos .....</b>	<b>2</b>
<b>CAPITULO II .....</b>	<b>3</b>
<b>MARCO TEORICO .....</b>	<b>3</b>
2.1. Regulación automática .....	3
2.2. Sistemas de control .....	3
2.3. Tipos de control.....	5
2.4. Sintonización de controladores PID.....	7
2.5. Análisis de la respuesta transitoria de un sistema .....	8
<b>2.5.1. Estabilidad absoluta, estabilidad relativa y error en estado estacionario .....</b>	<b>9</b>
<b>2.5.2. Propiedades de la respuesta transitoria .....</b>	<b>10</b>
2.6. Modelado matemático de sistemas.....	11
2.7. Función de transferencia .....	12
2.8. Protocolo de comunicación OPC .....	13
2.9. Criterio de estabilidad de routh .....	13
2.10. Diagrama de bode .....	14
<b>CAPITULO III.....</b>	<b>15</b>
<b>DISEÑO DEL ENTORNO DE CONTROL .....</b>	<b>15</b>
3.1. Configuración pasarela OPC.....	15
3.2. Desarrollo en factory io.....	18
3.3. Desarrollo en matlab .....	21
<b>CAPÍTULO IV .....</b>	<b>25</b>
<b>IDENTIFICACION DEL SISTEMA .....</b>	<b>25</b>
4.1. Simulación control manual .....	25
4.2. Identificación del sistema.....	26
4.3. Sintonización PID .....	29

<b>CAPÍTULO V</b> .....	<b>34</b>
<b>OPTIMIZACION Y PRUEBAS DEL SISTEMA</b> .....	<b>34</b>
5.1. Primera simulación.....	34
5.2. Segunda simulación .....	35
5.3. Tercera simulación .....	37
<b>CAPÍTULO VI</b> .....	<b>41</b>
<b>INVERSIÓN Y OPTIMIZACIÓN DEL PROYECTO</b> .....	<b>41</b>
6.1. Costos de implementación .....	41
6.2. Visión futura de proyecto.....	42
<b>CAPITULO VII</b> .....	<b>44</b>
<b>CONCLUSIONES</b> .....	<b>44</b>
<b>BIBLIOGRAFIA</b> .....	<b>45</b>

## RESUMEN

El objetivo principal del trabajo desarrollado, es el de lograr realizar el diseño de un control automático de nivel utilizando software computacional, la problemática principal que se expuso es que al diseñar controles directamente sobre un proceso productivo real, es que existe la posibilidad de llevarlo a la inestabilidad mientras se realiza el diseño o sintonización y con esto provocar muchos tipos de pérdida en el mismo, es por ello que en este trabajo se buscó la manera de, a través de la función de transferencia del proceso, lograr diseñar un sistema de control sobre datos teóricos calculados por software, en vez de hacerlo directamente sobre el proceso real, para luego, traspasar todos esos datos calculados al proceso real, aunque debido a la dificultad para implementar un control en algo real, es que se eligió realizarlo en un entorno simulado. La manera de lograr este objetivo es, en un principio, diseñar el entorno de control en una computadora, esta etapa consistió en crear entornos de control en dos softwares como los son factory io y matlab, además de utilizar el software kepsserver para la intercomunicación entre los otros dos anteriormente mencionados. Luego se realizó la identificación del proceso, esta etapa consistió en la utilización del software matlab para recoger datos en tiempo real desde la simulación del software factory io y con ellos, estimar la función de transferencia del sistema, algo que matlab es capaz de hacer automáticamente, con la función del sistema, se procedió a diseñar el control para posteriormente analizar y realizar pruebas para optimizar el mismo y por último, se probó que los datos calculados funcionaran realmente sobre el proceso, se logró concluir que el diseño de un control automático usando software computacional fue satisfactorio toda vez que las comparaciones entre lo teórico versus lo simulado no tuvieron grandes diferencias.

## **ABSTRACT**

The main objective of the developed work is to achieve the design of an automatic level control using computer software. The main issue presented is that designing controls directly on a real productive process carries the risk of destabilizing it during the design or tuning phase, potentially causing various types of losses. Therefore, this work aimed to find a method to design a control system using the process's transfer function, based on theoretically calculated data by software, rather than directly on the real process. Then, all these calculated data can be transferred to the real process. However, due to the difficulty of implementing control in a real system, a simulated environment was chosen. The way to achieve this objective initially involved designing the control environment on a computer. This stage consisted of creating control environments in two software programs, namely Factory I/O and MATLAB, in addition to using Kepserver software for intercommunication between the two aforementioned software programs. Next, the process was identified. This stage involved using MATLAB software to collect real-time data from the Factory I/O software simulation and using it to estimate the system's transfer function, something MATLAB can do automatically. With the system's function, the control was designed, followed by analysis and testing to optimize it. Finally, it was confirmed that the calculated data worked effectively on the process. It was concluded that the design of an automatic control using computer software was satisfactory, as the comparisons between theoretical and simulated results showed no significant differences.

# CAPITULO I

## INTRODUCCION

### 1.1 Generalidades

El control automatico hoy en día es mas o menos importante dependiendo de la industria en la cual se aplique. En la industria minera, no es tan necesaria la precisión de un control si la comparamos por ejemplo con un laboratorio químico, una fabrica de alimentos o un sistema de algún observatorio astronómico, sin embargo, lo que si necesita un control automatico en la industria minera es estabilidad. Según consultas y experiencias recopiladas, la sintonización de controles automáticos se continua realizando por el famoso método de tanteo, ingreso de valores a la prueba y error hasta encontrar la estabilidad, este método a veces funciona y a veces no, lo que puede conllevar a dos posibles resultados, que los operadores de un proceso afectado trabajen de manera totalmente manual (agregando el factor del error humano) o que el tanteo sobre el proceso lo lleve a una inestabilidad la cual provoque perdidas productivas. Es por esta razón que se propone una manera de diseñar un control automatico sin la necesidad recurrir a la prueba y error sobre un proceso real, sino que hacerlo sobre un modelo matemático del mismo, utilizando software computacional, para lograr esto es necesario dividir la tarea en varios puntos, el primero es diseñar un entorno de control en una computadora sobre la cual trabajar, segundo, es utilizar el algún software computacional como matlab para identificar y analizar el proceso, esto se puede hacer a través de datos recopilados del mismo proceso, por ejemplo, en una planilla Excel, tercero, diseñar el control sobre la identificación del sistema realizada y cuarto, realizar pruebas con los datos calculados obtenidos en el sistema real.

La principal dificultad, es que como mantenedor del área eléctrica e instrumentación es difícil acceder a los sistemas de control de una compañía, por lo cual se opto por realizar todo este trabajo sobre un proceso simulado.

El desarrollo del trabajo se divide principalmente en: Diseño del entorno de control, identificación del sistema y optimización y pruebas del sistema.

## **1.2. Objetivo general**

- Diseñar un control automático de nivel utilizando matlab

## **1.3. Objetivos específicos**

- Diseñar el entorno de control entre los diversos softwares
- Realizar el análisis de la respuesta transitoria del control en lazo abierto.
- Estimar la función de transferencia del sistema.
- Sintonizar el control PID desde la función de transferencia del sistema.
- Analizar el comportamiento del sistema con los datos calculados.

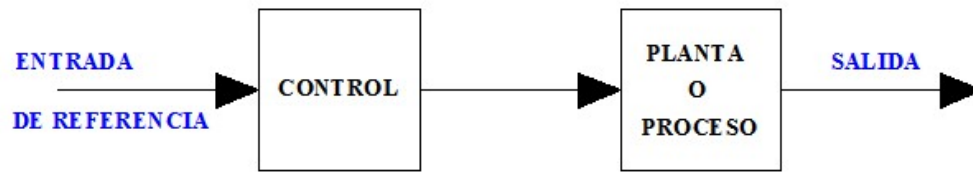
## **CAPITULO II MARCO TEORICO**

### **2.1. Regulación automática**

La regulación automática, también llamada teoría de control, estudia el comportamiento de los sistemas dinámicos, tratándolos como cajas o bloques con una entrada y una salida. En general, la entrada al sistema es una señal analógica o digital que se capta en algún punto del sistema. Los bloques intermedios representan las diversas acciones perturbadoras que afectan a la señal, como rozamientos en los actuadores, así como el efecto de los elementos de control interpuestos, los reguladores. Estos efectos se suelen representar mediante las funciones matemáticas que los describen, llamadas funciones de transferencia. La salida del sistema se llama referencia y corresponde al valor de la señal tras actuar sobre ella las anteriores funciones de transferencia. Cuando una o más de las variables de salida de un sistema tienen que seguir el valor de una referencia que cambia con el tiempo, se necesita interponer un controlador que manipule los valores de las señales de entrada al sistema hasta obtener el valor deseado de salida.

### **2.2. Sistemas de control**

Se puede definir un sistema de control como la combinación de elementos que, actuando sobre una planta o proceso, trata de fijar alguno de sus parámetros o de hacer que varíe, en el transcurso del tiempo, de una forma determinada que se predefine. Para representar esquemáticamente un sistema se utilizan los diagramas de bloques, en los que cada elemento o conjunto de elementos se simboliza con un bloque o rectángulo. Unas flechas indican el sentido de la información, que es único. La estructura más simple de un sistema de control es la que indica la siguiente figura.



**Figura N° 2.1. Sistema de control típico.**

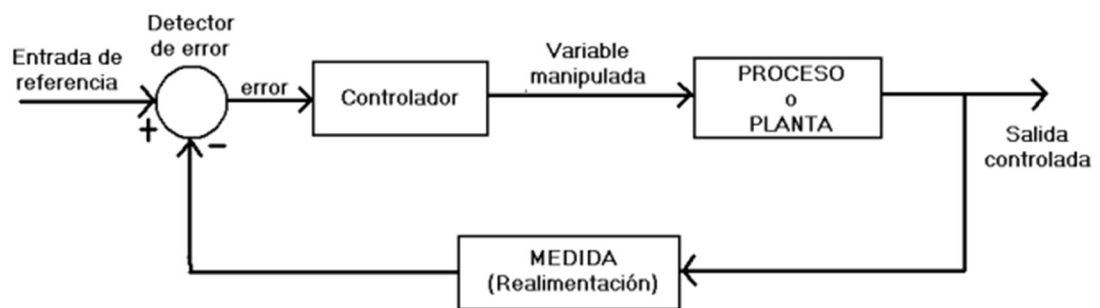
El control automático siempre está integrado por tres operaciones básicas, las cuales son:

- Medida: la medida de la variable a controlar se realiza mediante un sensor, a veces, combinado con un transmisor o un convertidor para adaptar una señal.
- Decisión: basándose en la medida, el controlador debe decidir qué hacer, para mantener una variable dentro de un valor deseado.
- Acción: como resultado de la decisión del controlador, el sistema debe emprender alguna acción. Normalmente la realiza el elemento final u órgano de control.

Los sistemas de control de lazo abierto constituyen el tipo más sencillo y económico de los sistemas de control y consta de dos partes: el controlador y el proceso (o planta) controlado. Una señal de entrada o comando se aplica al controlador, cuya salida actúa como señal actuante; la señal actuante controla el proceso de tal forma que la variable controlada se desempeña de acuerdo con lo preestablecido. En los casos simples, el controlador puede ser un amplificador, unión mecánica, filtro, u otro elemento de control. En los casos más complejos, el controlador puede ser una computadora. Debido a la simplicidad y economía de los sistemas de control en lazo (o bucle) abierto, se les encuentra en aplicaciones no críticas.

Lo que hace falta en el sistema de control en lazo abierto para que sea más exacto y más adaptable es una conexión o realimentación desde la salida hacia la entrada del sistema. Para obtener un control más exacto, la señal controlada debe ser realimentada y comparada con la entrada de referencia, y se debe enviar una señal actuante proporcional a la diferencia de la entrada y la salida a través del sistema para corregir el error. Un sistema con una o más trayectorias de realimentación se denomina sistema en lazo (o bucle) cerrado.

La figura 2.2 muestra un sistema simple en lazo cerrado, donde se puede destacar la presencia de un comparador o detector de error, así como un transductor o captador que toma la medida en la salida del sistema, para poderla comparar con la consigna o entrada de referencia. Cualquier diferencia entre estas dos señales constituye una señal de error o de actuación. El controlador recibe esta señal y modifica la llamada variable manipulada en tal sentido que obliga al proceso o planta a reducir el error original. El control realizará su acción correctora hasta que el error  $e(t)$  sea nulo, obligando, de este modo, a que la salida siga a la entrada de referencia (que representa la salida deseada). Está claro que, de este modo, se obtiene un sistema de control con un comportamiento totalmente automático, ya que no es precisa la intervención humana para adaptar la salida a la entrada, de ahí el nombre con que también se conocen los sistemas realimentados o de lazo cerrado: sistemas de control automático.



**Figura N° 2.2. Lazo cerrado de control**

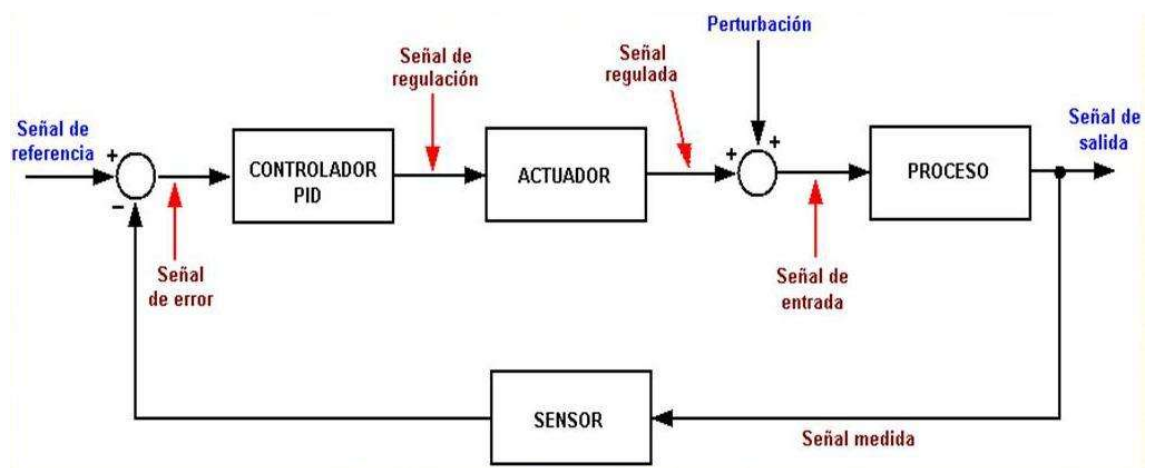
### 2.3. Tipos de control

Dentro de los diversos tipos de control existentes se pueden describir algunos como: Control todo-nada, flotante, proporcional de tiempo variable, proporcional, proporcional + integral, proporcional + derivativo, proporcional + integral + derivativo. Se centrará la información en los más importantes.

2.3.1. Control todo-nada. En la regulación todo-nada, el elemento de control adopta únicamente dos posiciones, abierto o cerrado, energizado o desenergizado, para un valor único de la variable controlada. Este tipo de control se caracteriza por un ciclo continuo de variación de la variable controlada.

2.3.2. Control proporcional. En el sistema de posición proporcional existe una relación lineal continua entre el valor de la variable controlada y la posición del elemento final de control. Es decir, la válvula se mueve en igual cantidad de magnitud por cada unidad de desviación o en otras palabras, la posición del elemento final de control es una copia inversa de la variable controlada.

2.3.3. Control proporcional, integral, derivativo. Un controlador o regulador PID es un dispositivo que permite controlar un sistema en lazo cerrado para que alcance el estado de salida deseado. El controlador PID está compuesto de tres elementos que proporcionan una acción Proporcional, Integral y Derivativa. Estas tres acciones son las que dan nombre al controlador PID. Este se muestra en la siguiente figura.



**Figura N° 2.3. Control PID**

#### A. Respuesta proporcional

El componente proporcional depende únicamente de la diferencia entre el punto de referencia y la variable del proceso. Esta diferencia se conoce como el término error. La ganancia proporcional ( $K_c$ ) determina la relación entre la respuesta de salida y la señal de error. Por ejemplo, si el término de error tiene una magnitud de 10, una ganancia proporcional de 5 produciría una respuesta proporcional de 50. En general, aumentar la

ganancia proporcional aumentará la velocidad de respuesta del sistema de control. Sin embargo, si la ganancia proporcional es demasiado grande, la variable del proceso comenzará a oscilar. Si  $K_c$  se aumenta aún más, las oscilaciones serán mayores y el sistema se volverá inestable e incluso puede oscilar fuera de control.

#### B. Respuesta integral

El componente integral suma el término de error en el transcurso del tiempo. El resultado es que incluso un término de error pequeño hará que el componente integral aumente lentamente. La respuesta integral aumentará continuamente con el tiempo a menos que el error sea cero, por lo que el efecto es llevar el error de estado estable a cero. El error de estado estable es la diferencia final entre la variable del proceso y el punto de referencia. Se produce un fenómeno llamado *windup* integral cuando la acción integral satura un controlador sin que el controlador conduzca la señal de error hacia cero.

#### C. Respuesta derivada

El componente derivado hace que la salida disminuya si la variable del proceso aumenta rápidamente. La respuesta derivada es proporcional a la tasa de cambio de la variable del proceso. Aumentar el parámetro de *tiempo derivado* ( $T_d$ ) hará que el sistema de control reaccione con más fuerza a los cambios en el término de error y aumentará la velocidad de respuesta del sistema de control en general. Los sistemas de control más prácticos utilizan un tiempo derivado muy pequeño ( $T_d$ ), porque la respuesta derivada es muy sensible al ruido en la señal de la variable del proceso. Si la señal de retroalimentación del sensor es ruidosa o si la tasa del ciclo de control es demasiado lenta, la respuesta derivada puede hacer que el sistema de control sea inestable

### 2.4. Sintonización de controladores PID

El proceso de establecer las ganancias óptimas para P, I y D para obtener una respuesta ideal desde un sistema de control se denomina *sintonización*. Existen diferentes métodos de sintonización de los cuales se discutirán el método de "adivinar y comprobar" y el método de Ziegler Nichols. Las ganancias de un controlador PID se pueden obtener con un método de prueba y error. Una vez que un ingeniero comprende la importancia de cada

parámetro de ganancia, este método se vuelve relativamente fácil. En este método, los términos I y D se establecen en cero primero y la ganancia proporcional se aumenta hasta que la salida del ciclo oscila. A medida que se aumenta la ganancia proporcional, el sistema se vuelve más rápido, pero debe tener cuidado de no volverlo inestable. Una vez que se ha establecido P para obtener una respuesta rápida deseada, el término integral se aumenta para detener las oscilaciones. El término integral reduce el error de estado estable, pero aumenta el sobreimpulso. Siempre es necesaria cierta cantidad de sobreimpulso para que un sistema rápido pueda responder de inmediato a los cambios. El término integral se modifica para lograr un mínimo error de estado estable. Una vez que P e I se han configurado para obtener el sistema deseado de control rápido con un mínimo error de estado estable, el término derivado se aumenta hasta que el ciclo es aceptablemente rápido hasta su punto de referencia. Aumentar el término derivado disminuye el sobreimpulso y produce una mayor ganancia con estabilidad, pero provocaría que el sistema fuera muy sensible al ruido. Muchas veces, los ingenieros necesitan intercambiar una característica de un sistema de control por otra para cumplir mejor con sus requisitos. El método Ziegler-Nichols es otro método popular para sintonizar un controlador PID. Es muy similar al método de prueba y error en el que I y D se establecen en cero y P se aumenta hasta que el ciclo comienza a oscilar. Una vez que comienza la oscilación, se anotan la ganancia crítica  $K_c$  y el período de oscilaciones  $P_c$ . P, I y D luego se ajustan según la figura 2.4. que se muestra a continuación.

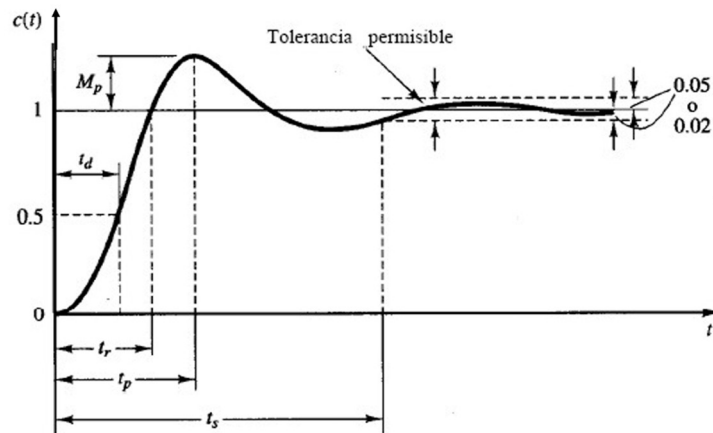
Control	P	Ti	Td
P	$0.5K_c$	-	-
PI	$0.45K_c$	$P_c/1.2$	-
PID	$0.60K_c$	$0.5P_c$	$P_c/8$

**Figura N° 2.4. Sintonizacion metodo ziegler-nichols**

## 2.5. Análisis de la respuesta transitoria de un sistema

La respuesta en el tiempo de un sistema de control consta de dos partes: la respuesta transitoria y la respuesta en estado estacionario. La respuesta transitoria se refiere a la que va del estado inicial a un estado estable. Por respuesta en estado estacionario se entiende la manera como se comporta la salida del sistema conforme el tiempo tiende a infinito.

El análisis de estas mencionadas respuestas del sistema se entiende como el estudio del comportamiento del mismo frente a los distintos tipos de entradas a la que este puede ser sometido, como, por ejemplo, entrada tipo escalón, impulso o rampa, a las que también se les conoce como señales de prueba. Un ejemplo se muestra a continuación en la figura 2.5.



**Figura N° 2.5. Análisis respuesta transitoria**

### 2.5.1. Estabilidad absoluta, estabilidad relativa y error en estado estacionario

Al diseñar un sistema de control, se debe ser capaz de predecir su comportamiento dinámico a partir del conocimiento de los componentes. La característica más importante del comportamiento dinámico de un sistema de control es la estabilidad absoluta, es decir, si el sistema es estable o inestable. Un sistema de control está en equilibrio si, en ausencia de cualquier perturbación o entrada, la salida permanece en el mismo estado. Un sistema de control lineal e invariante con el tiempo es estable si la salida termina por regresar a su estado de equilibrio cuando el sistema está sujeto a una condición inicial. Un sistema de control lineal e invariante con el tiempo es críticamente estable si las oscilaciones de

la salida continúa de forma indefinida. Es inestable si la salida diverge sin límite a partir de su estado de equilibrio cuando el sistema está sujeto a una condición inicial. En realidad, la salida de un sistema físico puede aumentar hasta un cierto grado, pero puede estar limitada por «detenciones» mecánicas, o el sistema puede colapsarse o volverse no lineal una vez que la salida excede cierta magnitud, por lo cual ya no se aplican las ecuaciones diferenciales lineales.

Entre los comportamientos importantes del sistema (aparte de la estabilidad absoluta) que deben recibir una cuidadosa consideración están la estabilidad relativa y el error en estado estacionario. Como un sistema de control físico implica un almacenamiento de energía, la salida del sistema, cuando este está sujeto a una entrada, no sucede a la entrada de inmediato, sino que muestra una respuesta transitoria antes de alcanzar un estado estacionario. La respuesta transitoria de un sistema de control práctico, con frecuencia, muestra oscilaciones amortiguadas antes de alcanzar un estado estacionario. Si la salida de un sistema en estado estacionario no coincide exactamente con la entrada, se dice que el sistema tiene un error en estado estacionario. Este error indica la precisión del sistema. Al analizar un sistema de control, se debe examinar el comportamiento de la respuesta transitoria y el comportamiento en estado estacionario.

### **2.5.2. Propiedades de la respuesta transitoria**

La respuesta transitoria se puede cuantificar siguiendo las siguientes propiedades o características:

- Tiempo de subida o crecimiento, esta característica se refiere al tiempo que el sistema se demora en alcanzar el 90% aproximadamente del valor seteado desde un valor bajo.
- Sobreimpulso, valor generalmente en porcentaje del valor máximo que toma la respuesta del sistema medida a partir de la unidad (del valor seteado). Se puede calcular según la siguiente ecuación.

$$\text{Pocentaje de sobreimpulso} = \frac{c(t_p) - c(\infty)}{c(\infty)} * 100\%$$

### **Ecuación N°2.1. Sobreimpulso**

- Tiempo de asentamiento, es el tiempo transcurrido desde la aplicación de una entrada de paso instantáneo ideal hasta el momento en que la salida ha entrado y permanecido dentro un error en estado estable cercano al valor seteado.
- Tiempo de retardo, el tiempo de retardo es el necesario para que la respuesta alcance inicialmente la mitad del valor final.
- Tiempo peak o peak, es el tiempo que demora el sistema en alcanzar el primer sobreimpulso, también puede ser expresado como factor del valor seteado.

## **2.6. Modelado matemático de sistemas**

Una de las definiciones más simples de modelo es la propuesta por (Lee, 2016): “Un modelo es una representación de la realidad”. Se puede definir el modelado de un determinado proceso como la obtención de un conjunto de funciones matemáticas que permiten representar, al menos de forma aproximada, el comportamiento de las variables de mayor interés del sistema bajo estudio.

Existen diferentes técnicas de modelado de sistemas dinámicos, pero de una forma muy esquematizada se pueden separar en dos grandes grupos: el modelado experimental y el modelado analítico. El modelado experimental hace uso de la respuesta del sistema ante determinadas entradas de prueba para obtener una función matemática que relaciona las variables de salida del sistema con las variables de entrada del mismo. Se puede decir que esta metodología de modelado “ve”, al sistema a estudiar, como una caja negra en la que no se analiza de ninguna forma el tipo de componentes que lo forman y que obtiene la expresión matemática que liga entradas y salidas sin conocer los elementos internos.

Por el contrario, el modelado analítico pretende obtener el conjunto de funciones matemáticas que representan el comportamiento del sistema, en base al estudio detallado de cada uno de los componentes que lo integran y sus interrelaciones. Esta metodología

es la que proporciona un mayor volumen de información acerca del funcionamiento del sistema, del conjunto de variables, así como de la interrelación entre las mismas.

La importancia del modelado matemático se puede establecer en que para muchos problemas es posible encontrar un controlador adecuado simplemente mediante prueba y error. Sin embargo, en muchos casos el enfoque de prueba y error no es factible, debido a complejidad, eficiencia, costo, o aún seguridad. En particular, es imposible mediante prueba y error responder a cuestiones como las siguientes antes de hacer pruebas:

- Dada una planta y un objetivo deseado de operación, ¿qué controlador puede alcanzarlo? ¿Se puede alcanzar el objetivo propuesto con algún controlador?
- Dados un controlador y una planta, ¿cómo operarán en lazo cerrado?
- ¿Por qué un lazo dado opera de la forma que lo hace? ¿Puede mejorarse? ¿Con qué controlador?
- ¿Cómo cambiaría la operación si se cambiaran los parámetros del sistema, o si las perturbaciones fueran mayores, o si fallara algún sensor?

Para responder sistemáticamente a estas cuestiones son necesarios modelos matemáticos. Los modelos matemáticos nos brindan los medios de capturar el comportamiento de un sistema sujeto a condiciones iniciales, entradas de control y perturbaciones mediante un conjunto de ecuaciones matemáticas.

La importancia de los modelos matemáticos radica en que pueden ser simulados en situaciones hipotéticas, ensayados en estados que serían peligrosos en el sistema real, y usados como base para sintonizar controladores.

## **2.7. Función de transferencia**

En la teoría de control, a menudo se usan las funciones de transferencia para caracterizar las relaciones de entrada-salida de componentes o de sistemas que se describen mediante ecuaciones diferenciales lineales invariantes en el tiempo. La función de transferencia de un sistema descrito mediante una ecuación diferencial lineal e invariante en el tiempo se define como el cociente entre la transformada de Laplace de la

salida (función de respuesta) y la transformada de Laplace de la entrada (función de excitación) bajo la suposición de que todas las condiciones iniciales son cero.

## 2.8. Protocolo de comunicación OPC

El OPC (OLE for Process Control) es un estándar de comunicación en el campo del control y supervisión de procesos industriales, basado en una tecnología Microsoft, que ofrece una interfaz común para comunicación que permite que componentes de software individuales interactúen y compartan datos. La comunicación OPC se realiza a través de una arquitectura Cliente-servidor. El servidor OPC es la fuente de datos (como un dispositivo hardware a nivel de planta) y cualquier aplicación basada en OPC puede acceder a dicho servidor para leer/escribir cualquier variable que ofrezca el servidor. Es una solución abierta y flexible al clásico problema de los drivers propietarios.

## 2.9. Criterio de estabilidad de routh

El problema más importante de los sistemas de control lineal tiene que ver con la estabilidad. Es decir, ¿en qué condiciones se vuelve inestable un sistema? Si es inestable, ¿cómo se estabiliza? Un sistema de control es estable si y sólo si todos los polos en lazo cerrado se encuentran en el semiplano izquierdo del plano S. La mayoría de los sistemas lineales en lazo cerrado tienen funciones de transferencia en lazo cerrado de la forma:

$$\frac{C(s)}{R(s)} = \frac{b_0s^m + b_1s^{m-1} + \dots + b_{m-1}s + b_m}{a_0s^n + a_1s^{n-1} + \dots + a_{n-1}s + a_n} = \frac{B(s)}{A(s)}$$

### Ecuación N°2.2. Función de transferencia

Donde las a y las b son constantes y m, n, el orden de la función (por ejemplo, primer, segundo, tercer orden). Un criterio simple, conocido como el criterio de estabilidad de Routh, permite determinar la cantidad de polos en lazo cerrado que se encuentran en el semiplano derecho del plano s sin tener que factorizar el polinomio.

El criterio de estabilidad de Routh dice si existen o no raíces inestables en una ecuación polinomial, sin tener que obtenerlas en realidad. Este criterio de estabilidad sólo se aplica a los polinomios con una cantidad finita de términos. Cuando se aplica el criterio a un sistema de control, la información sobre la estabilidad absoluta se obtiene directamente de los coeficientes de la ecuación característica.

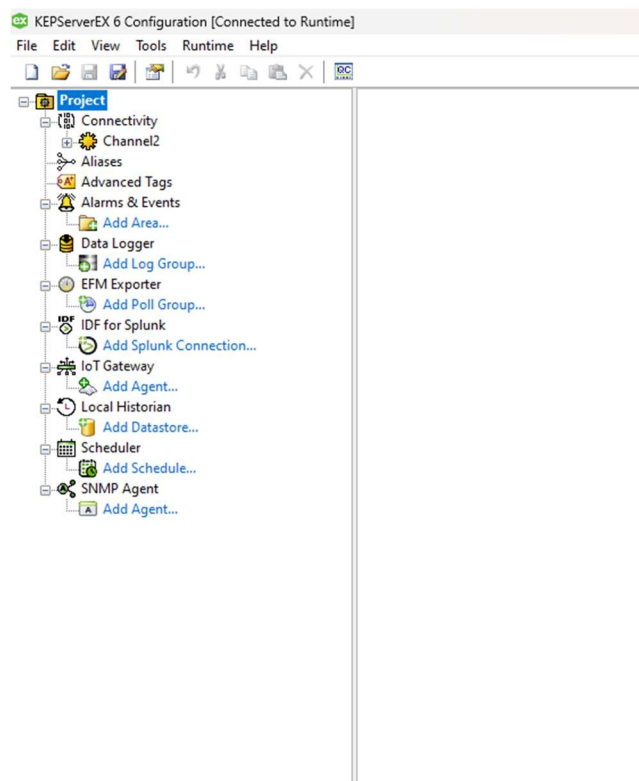
## **2.10. Diagrama de bode**

Un diagrama de Bode está formado por dos gráficas: una es la gráfica del logaritmo de la magnitud de la función de transferencia sinusoidal, y la otra es la gráfica del ángulo de fase; ambas se dibujan contra la frecuencia en escala logarítmica. La representación común de la magnitud logarítmica de  $G(j\omega)$  es  $20 \log G(j\omega)$ , donde la base del logaritmo es 10. La unidad utilizada en esta representación para la magnitud es el decibelio, por lo general abreviado dB. En la representación logarítmica, se dibujan las curvas sobre papel semilogarítmico, con la escala logarítmica para la frecuencia y la escala lineal para cualquier magnitud (en decibelios) o el ángulo de fase (en grados). (El rango de frecuencia de interés determina el número de ciclos logarítmicos que se requieren en la abscisa). La ventaja principal de utilizar el diagrama de Bode es que la multiplicación de magnitudes se convierte, en suma. Además, cuenta con un método simple para dibujar una curva aproximada de magnitud logarítmica. Se basa en aproximaciones asintóticas. Esta aproximación, mediante asíntotas (líneas rectas), es suficiente si sólo se necesita información general sobre la característica de la respuesta en frecuencia. Si se desea obtener curvas exactas, es fácil corregir las curvas asintóticas. Es muy útil ampliar el rango de bajas frecuencias mediante el uso de una escala logarítmica, debido a que las características de las bajas frecuencias son las más importantes en los sistemas prácticos. Aunque no es posible dibujar las curvas hasta una frecuencia cero, debido a la frecuencia logarítmica ( $\log 0\%$ ), esto no es un problema serio.

## CAPITULO III DISEÑO DEL ENTORNO DE CONTROL

### 3.1. Configuración pasarela OPC

El primer paso a realizar, mas que nada para tener cierto orden a la hora de armar el entorno donde se ejecutara la simulación es configurar el software donde se intercomunicaran los programas, esta es la pasarela OPC, aquí se configurara el servidor y se declararan las variables a utilizar dentro del proyecto, para ello se utilizara el software kepservers, el cual se muestra en la siguiente figura 3.1.



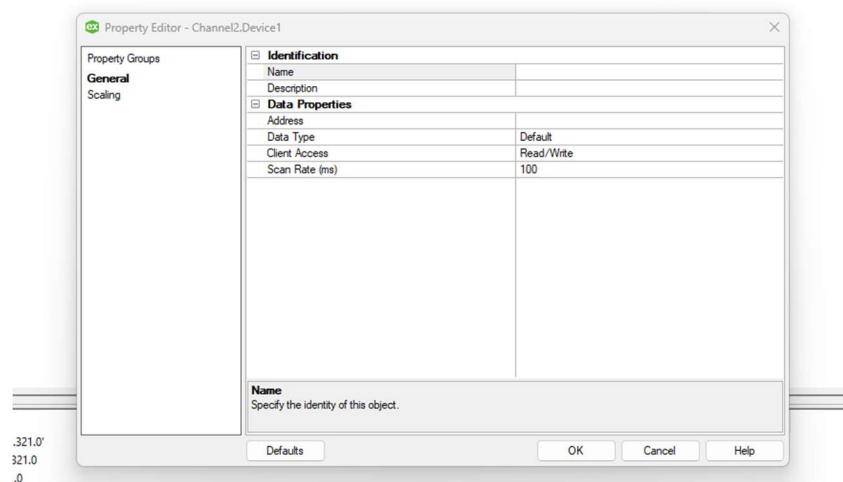
**Figura N° 3.1. Entorno kepservers**

Dentro de kepservers lo que se visualizará será una pantalla en blanco con un árbol de selección al costado izquierdo donde se debe seleccionar debajo de Project-connectivity,

agregar un canal. Se abrirá una ventana emergente donde hay que buscar en la lista desplegable OPC UA client que es el servidor a utilizar luego, se debe dar click en siguiente hasta finalizar.

Luego de haber agregado un canal se procede a crear las variables a utilizar dentro del proyecto en este caso en el software factory io. Para ello hay hacer click derecho sobre el cuadro en blanco y se verá como aparece una opción que dice “new tag”.

Al intentar agregar una variable se abrirá una ventana como la de la siguiente figura. N° 3.2.



**Figura N° 3.2. Creación de nueva variable**

En esta ventana se puede dar un nombre a la variable, una descripción, la dirección correspondiente, el tipo de dato de la variable, el privilegio del cliente sobre la variable y la tasa de escaneo de la misma. Los datos mas importantes son el nombre, la dirección y el tipo de dato. Para la dirección, el software OPC maneja las direcciones de todas las variables según el siguiente formato, Kxxxx, donde se debe anteponer una K mayúscula seguida de cuatro números comenzando por cuatro ceros. Luego para el tipo de dato se debe considerar el tipo a utilizar, para ello se puede ver en la web del software factory io los tipos de datos correspondientes a los elementos a utilizar, en este caso, las válvulas de alimentación – descarga y los sensores de nivel y flujo de salida, como se encuentran en la siguiente figura N° 3.3.

### Analog

Tag	Controller I/O	Type	Description
Tank # (Fill Valve)	Output	Float	[0, 10] V: fill valve positioning.
Tank # (Discharge Valve)	Output	Float	[0, 10] V: discharge valve positioning.
Tank # (Level Meter)	Input	Float	[0, 10] V: level meter value.
Tank # (Flow Meter)	Input	Float	[0, 10] V: flow meter value (10 V = 0.3543 m <sup>3</sup> /s).

**Figura N° 3.3. Tipo de dato factory io**

En la figura 3.3. se observa que el tipo de dato especificado es tipo flotante, por lo que es el que se debe seleccionar para la creación de variables en el servidor. Una vez definidos los parámetros se procede a crear la variable, para este caso se crearán cuatro que corresponderían los cuatro elementos anteriormente mencionados, los sensores de flujo y nivel y las válvula de alimentación y descarga. Otro detalle a tener en cuenta es el espacio de memoria que utilizan los tipos de datos, en este caso el tipo de dato flotante, según la indicación de ayuda del software kepsserver, el tipo de dato flotante ocupa un espacio de memoria de 32 bit o 4 byte, por lo que al momento de direccionar se debe dejar un espacio de 4 bytes para evitar el solapado de los datos.

Tag Name	Address	Data Type
Flow_int	K0000	Float
Level_int	K0004	Float
valve_alim	K0008	Float
valve_disch	K0012	Float

**Figura N° 3.4. Direccionamiento de variables**

En la figura N° 3.4. se puede apreciar como queda el direccionamiento de las variables a utilizar. Con esto ya no queda nada mas por realizar en el software kepsserver.

### 3.2. Desarrollo en factory io

Para la realización del proyecto se ejecuta un ejemplo de simulación de control de nivel utilizando distintos softwares computacionales como los son, factory io para la simulación del proceso, matlab para el control, modelado y diseño del sistema de control y kepsserver para la interconexión entre ambos softwares.

El diseño del entorno de control se debe comenzar con la configuración en el software factory io. Para ello se ejecuta el programa y se vera una interfaz como la de la siguiente figura N° 3.5.

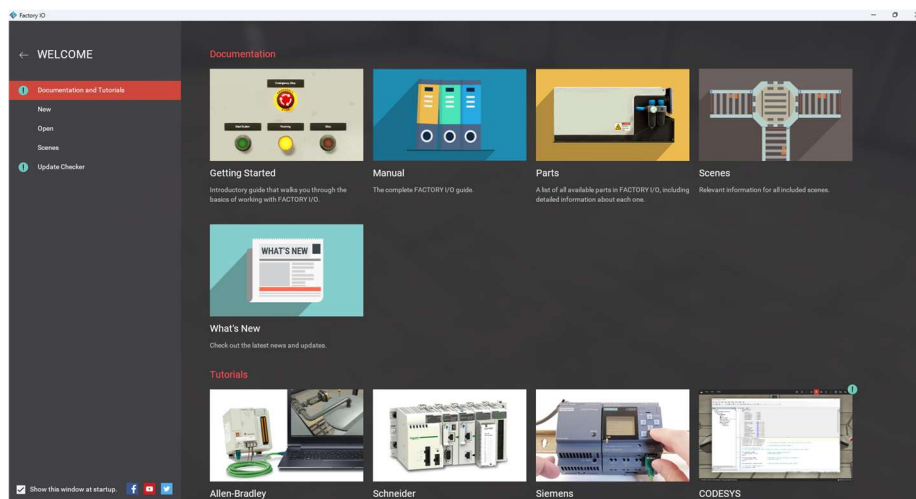


Figura N° 3.5. Factory io

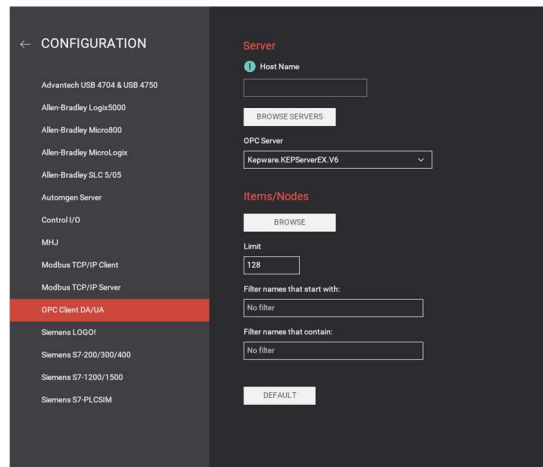
Haciendo click en el botón new, se abrirá un nuevo proyecto, la simplicidad de este programa para un control de nivel es grande ya que tiene un estanque ya integrado con todas las entrada-salidas asociadas. Dentro del programa se visualiza un entorno en 3d con un árbol de selección en el costado derecho que posee diversos elementos, dentro de estos elementos se puede encontrar el estanque a utilizar.



**Figura N° 3.6. Estanque de nivel**

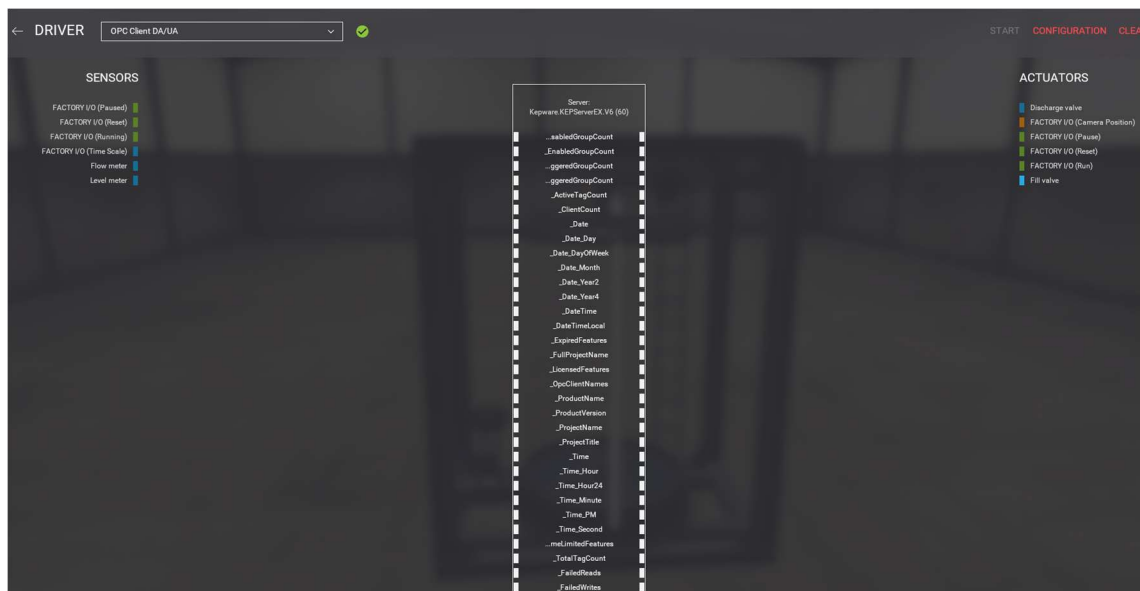
Observando la figura N°3.6, el sistema cuenta con dos válvulas, una de alimentación y una de descarga, así como dos sensores, uno de nivel del estanco y uno de flujo de salida del sistema. Adicionalmente pero no de manera obligatoria se le puede agregar un pequeño tablero donde se pueden adicionar algunas luces indicadoras y algunos display.

Una vez se tiene el entorno 3d se necesita interconectar los programas, para ello se presiona el botón file, luego la opción driver y dentro de la lista desplegable se selecciona el driver OPC client DA/UA y luego se procede a hacer click en configuración en la esquina superior derecha de la pantalla, así como se ve en la figura N° 3.7.



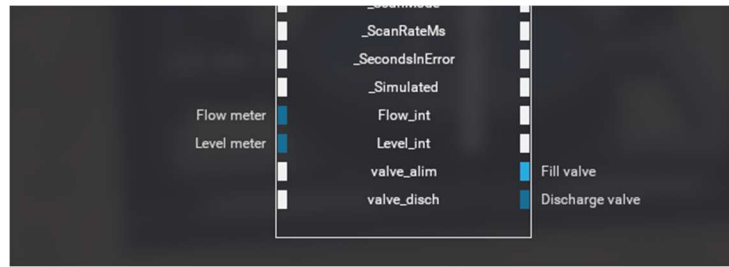
**Figura N° 3.7. Ventana de configuración**

Dentro del apartado de configuración se debe presionar en “buscar servidores” y luego en el recuadro “opc server” buscar el que diga “Kepware.KEPServerEX. V6” o similar. En la parte inferior se debe introducir 128 en el recuadro de limite, esto hace referencia a la cantidad de nodos o etiquetas existentes en el servidor, se selecciona esta cantidad en especifico debido a que existe una cantidad determinada de etiquetas o variable de sistema reservadas, entonces para asegurar que las que se han creado aparezcan en la lista, se colocan como límite 128 nodos. Luego se presiona el botón buscar en el apartado de ítems/nodos y con eso ya se puede volver atrás.



**Figura N° 3.8. Enlace de datos**

Si la configuración fue satisfactoria, se verá un cuadro con todas las variables del servidor como en la figura N° 3.8., normalmente las variables que se han creado se posicionaran en los últimos lugares de abajo, una vez se encuentren, solo queda enlazar las variables del servidor con las de nuestro proyecto, las que se ubican en las esquinas superiores izquierda y derecha según sea sensor o actuador.

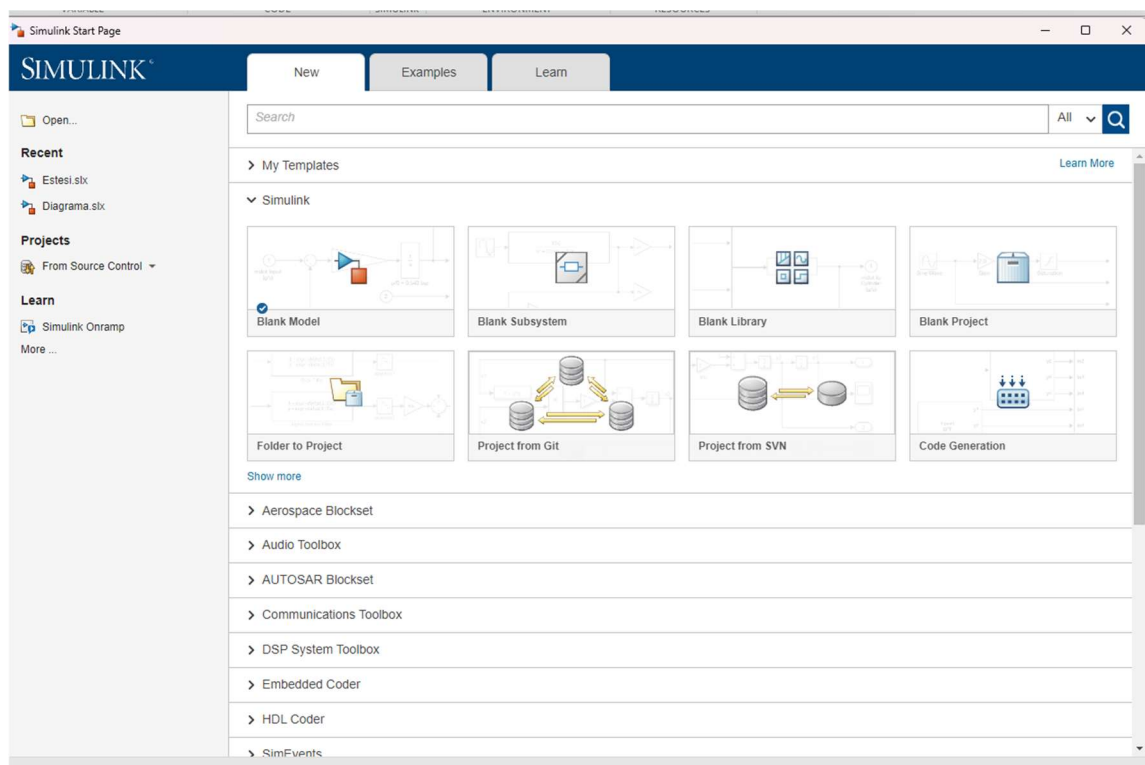


**Figura N° 3.9. Datos enlazados**

En la figura N°3.9. se ve como debería quedar el enlazado de los datos. Ya con esto se concluye la configuración en el software factory io.

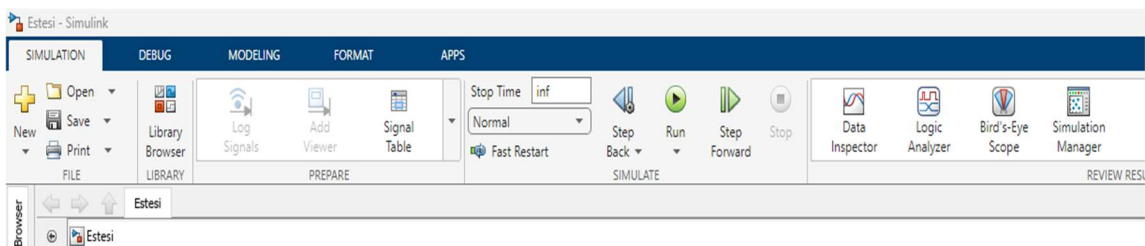
### 3.3. Desarrollo en matlab

Para comenzar la configuración en el entorno de programación de matlab se debe abrir el programa y buscar el complemento de simulink, como se ve en la figura N°3.10.



**Figura N° 3.10. Entorno matlab**

Con el software matlab abierto y una vez seleccionado el complemento de simulink lo que hay que hacer es abrir un nuevo modelo en blanco o “blank model”, como aparece en la figura. Una vez creado el proyecto se tendrá un gran espacio en blanco donde crear el diagrama, para lo que será el control de la simulación. Dentro del entorno se pueden observar diversos elementos como las pestañas superiores del programa, los iconos de apertura, guardado, el botono de buscador de librería, los cuadros respectivos a la ejecución de la simulación, etc. como se muestra en la figura N°3.11.



**Figura N° 3.11. Entorno de programación**

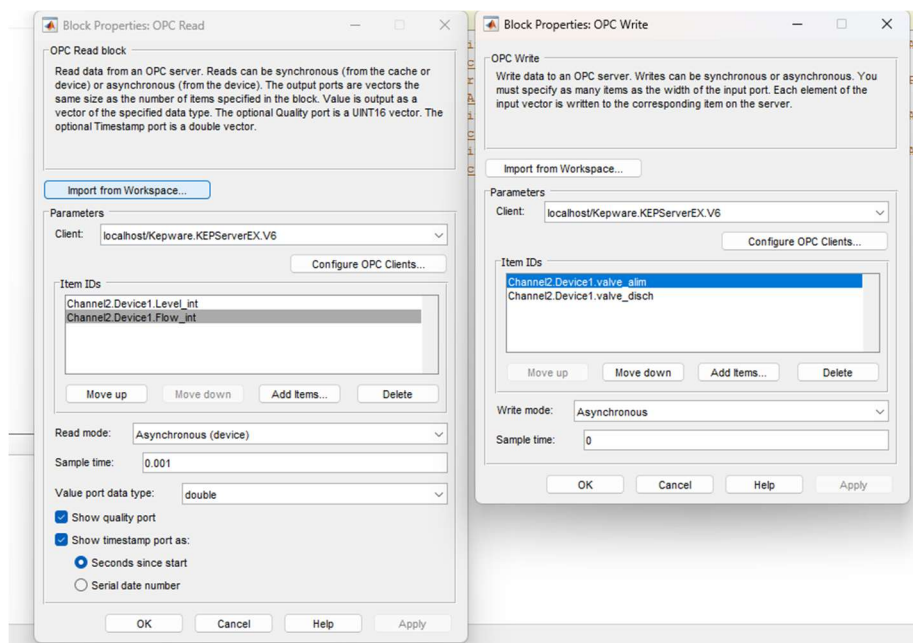
Lo más fácil para comenzar a construir el diagrama de control es hacer doble click sobre el espacio en blanco, ello desplegará un buscador donde simplemente se puede escribir el nombre del bloque y el buscador automáticamente lo encontrará.

Lo primero será buscar los bloques de comunicación OPC, para ello primero se busca el bloque llamado “opc configuration” que lo que hace es darle referencia de la configuración de clientes opc para modelos, control en pseudo tiempo real y comportamiento ante errores y eventos, es esencial para la comunicación. Luego se deben buscar bloques de escritura y lectura opc, estos, como su nombre lo dice, tienen la capacidad de escribir y leer los datos desde y hacia el servidor kepservers. Los bloques se muestran en la siguiente figura N°3.12.



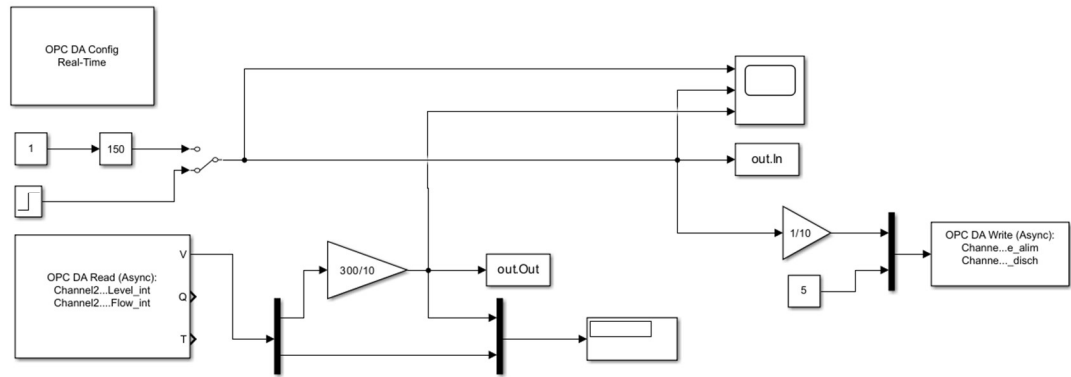
**Figura N° 3.12. Bloques de escritura y lectura opc**

A estos bloques se le deben asignar las respectivas variables ya configuradas en el servidor opc. Para ello se hace doble click en cada uno de estos bloques donde se abrirá una ventana de propiedades, en la ventana se debe primeramente configurar el cliente opc, para ello se debe agregar el server opc creado en el cual debería aparecer ya el listado de servidores disponibles, una vez agregado el server se deben agregar las variables a utilizar, para el caso se agregarán las variables de los sensores de flujo y nivel como entrada y las de control de las válvulas como salida, respectivamente en cada uno de los bloques asi como se ve el ejemplo en la figura 3.13.



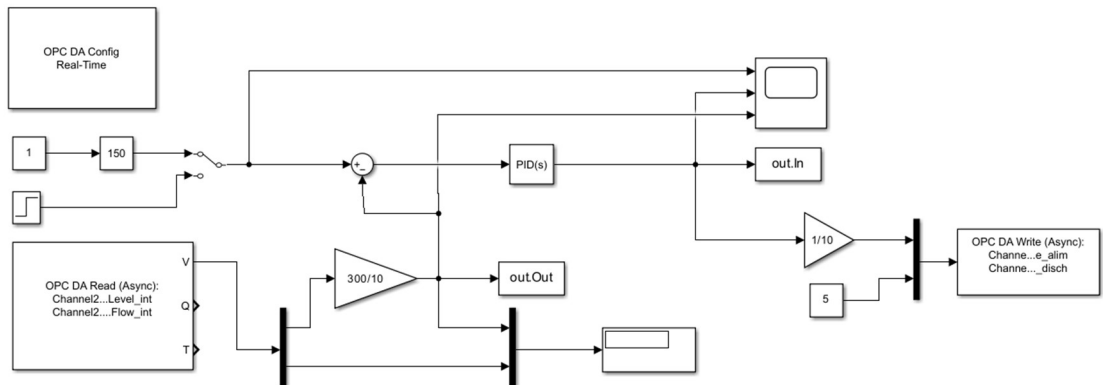
**Figura N° 3.13. Propiedades bloques opc**

Luego se continúa agregando bloques de: constantes, sumatoria para la realimentación del control pid, función escalón para el análisis de la respuesta transitoria, ganancia para el escalado correcto de las señales desde factory io, mux-demux para componer-descomponer señales grupales, osciloscopio para observar o graficar el comportamiento del sistema, bloques de traspaso de datos hacia el workspace de matlab para el uso de la optimización del sistema, el bloque de control PID, entre otros.



**Figura N° 3.14. Sistema de control manual**

Para el análisis de la respuesta transitoria se debe utilizar un control en lazo abierto como el de la figura N° 3.14., luego una vez obtenidos los datos se puede dejar como el de la figura N° 3.15. el cual ya sería el funcionamiento en automático.



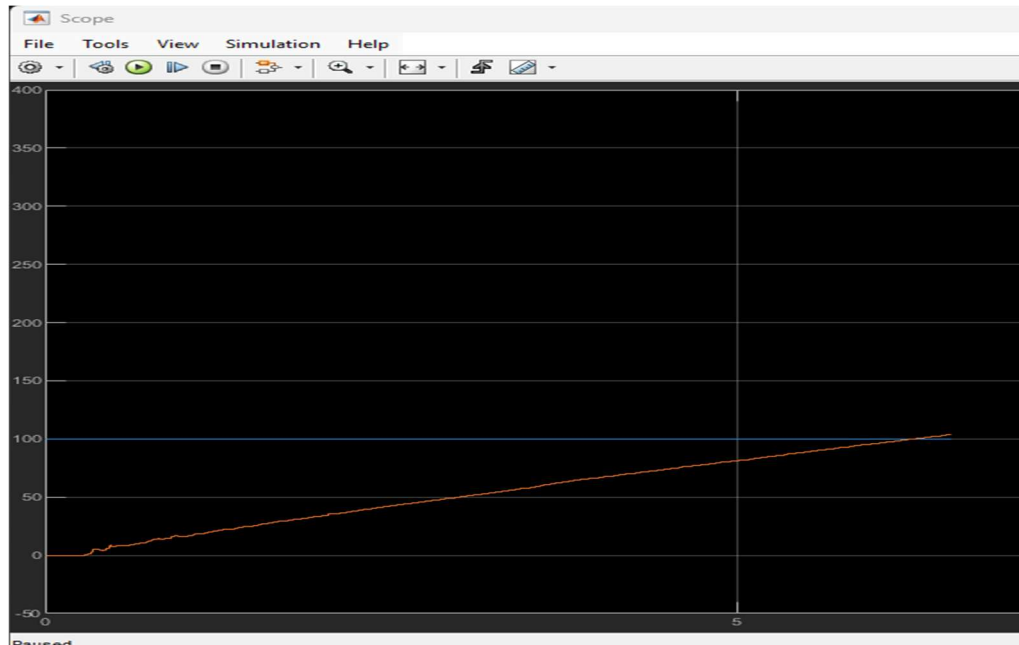
**Figura N° 3.15. Sistema de control automatico**

## CAPÍTULO IV IDENTIFICACION DEL SISTEMA

### 4.1. Simulación control manual

Una vez teniendo el entorno de control listo se puede comenzar a realizar pruebas, para ello se puede dar inicio o “run” a los softwares factory io y matlab (kepsver se encuentra corriendo siempre).

Utilizando el sistema de control manual de la figura N° 3.14., lo que se hará, será estudiar la respuesta el sistema en lazo abierto frente a una señal de prueba escalón, cabe destacar que aquí la variable a controlar es el nivel del estanque y para ello se controla como salida la válvula de alimentación, esto es meramente arbitrario ya que también se podría controlar el flujo y utilizar como elemento de control la válvula de descarga, sin embargo como el objetivo del proyecto se enfoca en el control de un proceso mas que en el proceso en sí, sirve esta configuración. Si se hace doble click en el osciloscopio se abrirá una ventana donde se graficarán las variables que estén ingresadas previamente, en este caso la entrada escalón y la variable de nivel.



**Figura N° 4.1. Gráfico comportamiento del sistema en lazo abierto.**

Como se observa en la figura anterior N°4.1. donde se tiene la señal escalón de prueba en azul y el nivel en salmón, se debe detener la simulación una vez el nivel haya alcanzado el valor final del escalon, lo que en este caso tomo alrededor de seis segundos. Con los bloques “out.In” y “out.Out”, se registran y envían los datos al espacio de trabajo de matlab donde se hará la identificación del sistema y la estimación del control PID.

Desde el espacio de trabajo en matlab ya se puede observar el registro de la primera simulación.

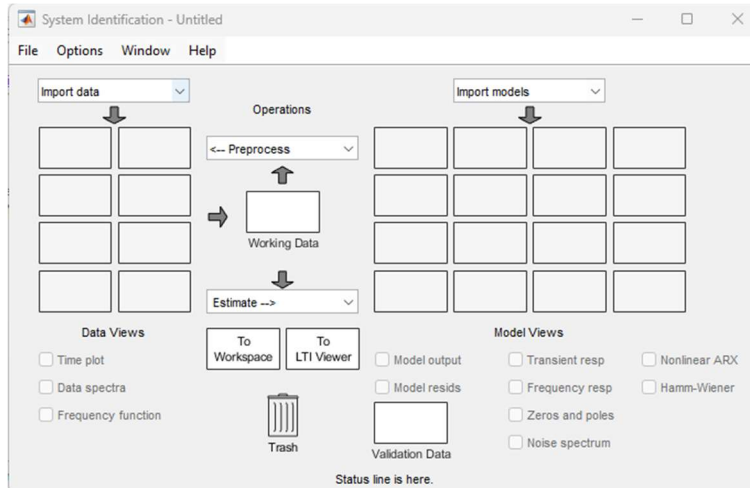


**Figura N° 4.2. Registro datos simulados**

En la columna derecha de la figura anterior N° 4.2. se ve el registro de datos de la simulación, si se hace doble click se abrirán las variables en el recuadro central, las que interesan son la de entrada que en este caso es la función escalon y la salida que sería el nivel, importante es fijarse que los valores de dichas variables sean iguales, es decir, que tengan la misma cantidad de datos, en este caso se observa que la entrada tiene 5658 datos y la salida uno menos, para igualarlos podemos hacer doble click en el valor de la entrada lo que nos abrirá un formato similar al libro de Excel, se debe borrar una fila y ya se tendrán igual cantidad de datos en ambas variables.

#### **4.2. Identificación del sistema**

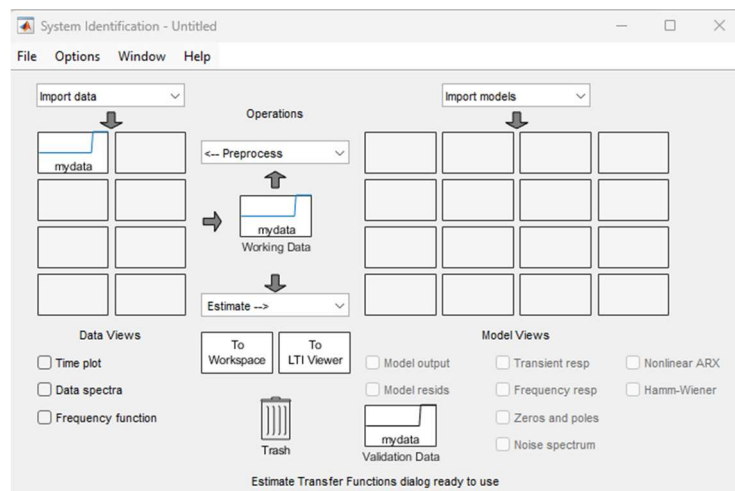
Ahora para realizar la identificación del sistema se debe usar dicha aplicación dentro de matlab, para ello en las pestañas superiores se debe buscar la llamada “apps” y ahí “identificación de sistema”, si se hace click se abrirá un cuadro como el mostrado en la figura N° 4.3. a continuación.



**Figura N° 4.3. Identificación de sistema**

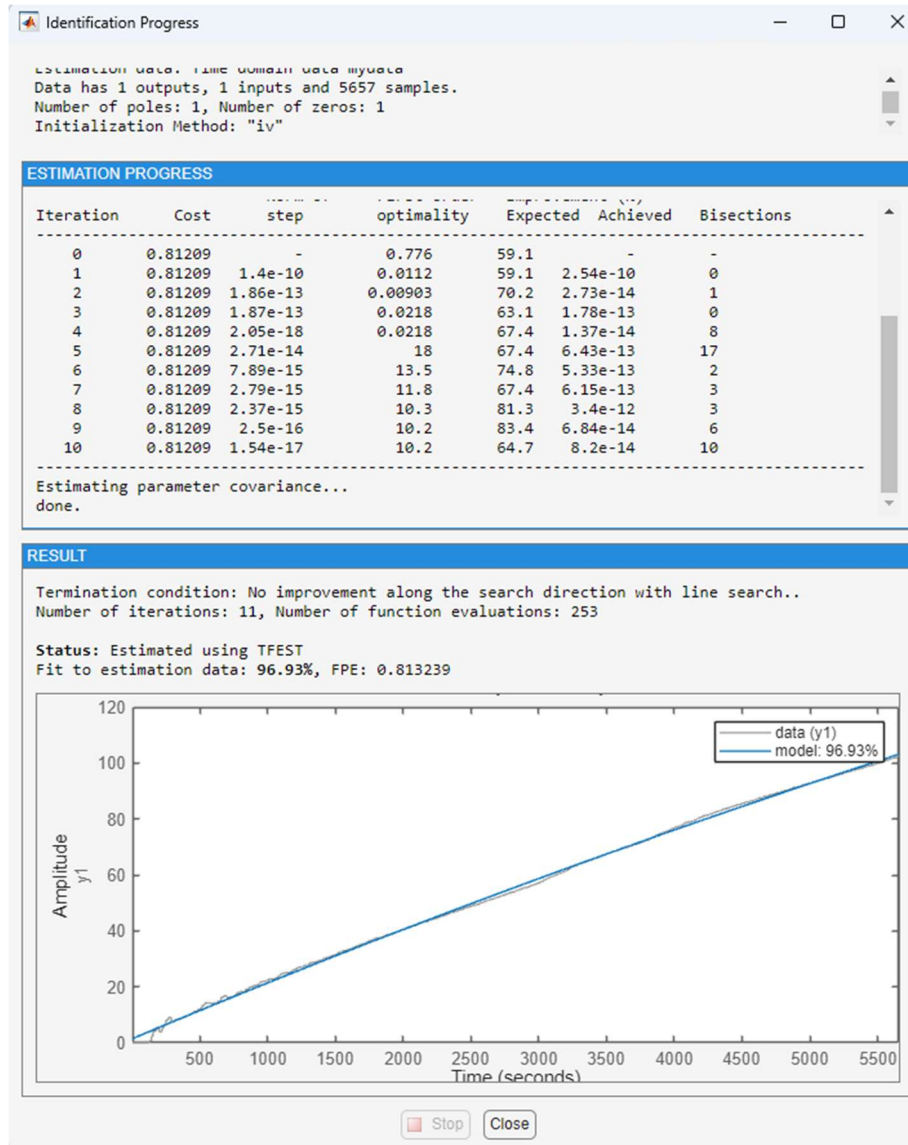
El primer paso es importar los datos recogidos en el espacio de trabajo, esto se hace buscando en la lista desplegable de la esquina superior izquierda de la figura anterior, en esta lista se observa el tipo de registro capturado, en este caso son datos registrados en el dominio del tiempo por lo que se debe seleccionar esta opción, una vez seleccionada se abrirá otro recuadro donde hay que anotar en entrada y salida el nombre, tal cual, de la entrada y salida registrada, out.In y out.Out.

Si se realiza bien debería aparecer en uno de los recuadros de la izquierda el registro, así como en la siguiente figura N°4.4.



**Figura N° 4.4. Datos importados**

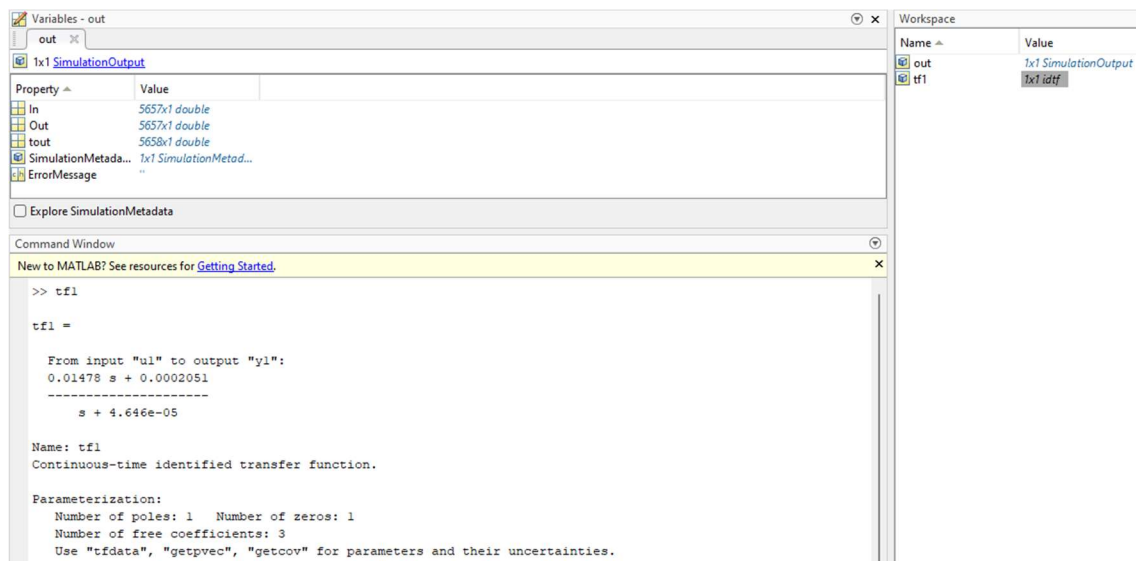
Lo siguiente sería seleccionar en la lista desplegable que dice “estimate” y ahí seleccionar la opción de modelos de función de transferencia, en un nuevo recuadro que aparecerá hay que elegir el nombre del modelo, así como el número de polos y ceros de la función, al ser un proceso simple podemos asumir que se trata de un modelo de primer orden por lo que se debe seleccionar los polos y ceros en uno, se hace click al botón de estimar y el software arrojará la primera estimación del sistema.



**Figura N° 4.5. Función de transferencia**

En este nuevo recuadro como el de la figura N° 4.5. se mostrarán los resultados de la estimación de la función de transferencia del sistema y una comparación grafica entre los datos registrados y el modelo calculado, se observa que en la primera estimación se obtuvo una aproximación de 96.93% lo cual es bastante bueno, como referencia, un 88% de aproximación ya es un buen valor.

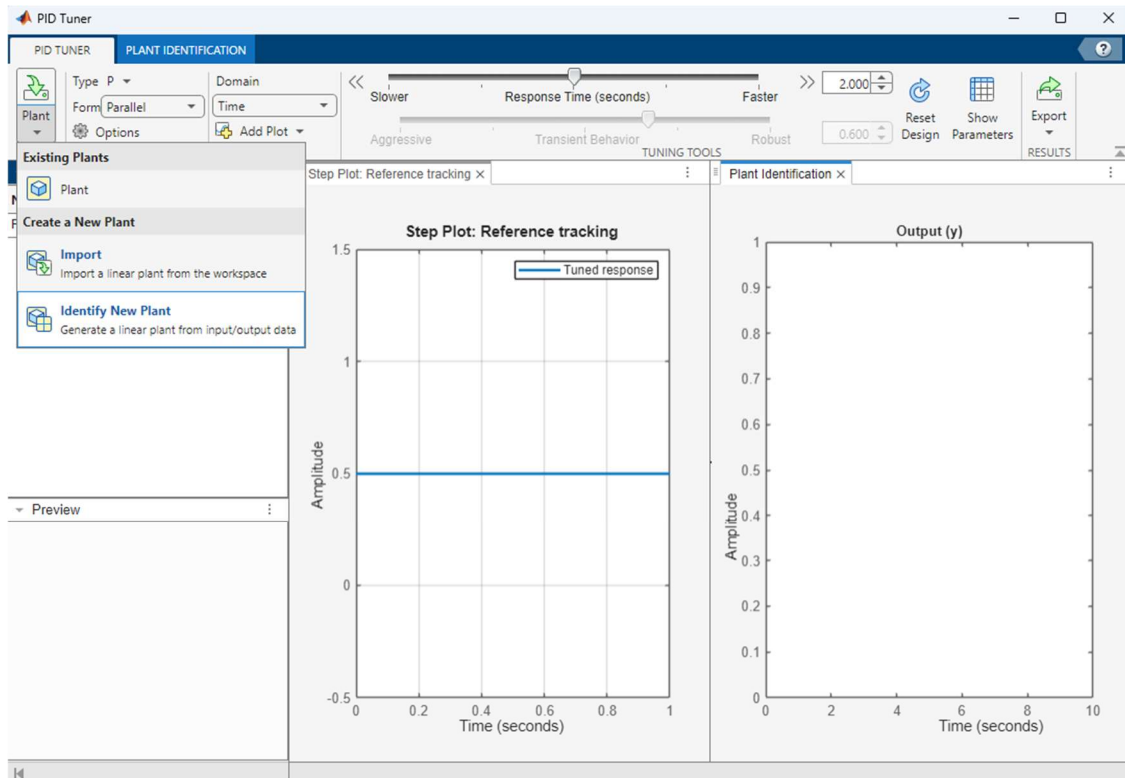
Ahora que la función de transferencia del sistema ya fue estimada, aparecerá como en uno de los recuadros de la derecha, se puede arrastrar el cuadro hacia el que dice “to workspace” para llevar la función al espacio de trabajo de matlab, volviendo al espacio de trabajo de matlab se arrastran los datos de la función a la ventana de comandos para ver los parámetros de dicha función, tal como se ve en la figura N° 4.6.



**Figura N° 4.6. Parametros funcion de transferencia**

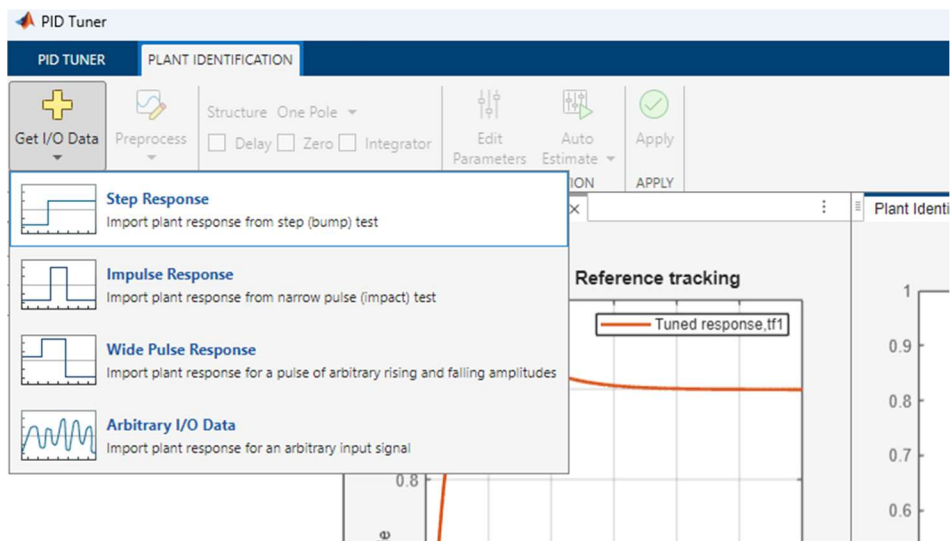
### 4.3. Sintonización PID

Ahora hay que realizar un control pid sobre la función de transferencia calculada, para ello en las pestañas superiores se selecciona “pid tuner” se abrirá una nueva ventana donde lo primero será importar la “planta”, es decir, la función de transferencia que representa al proceso.



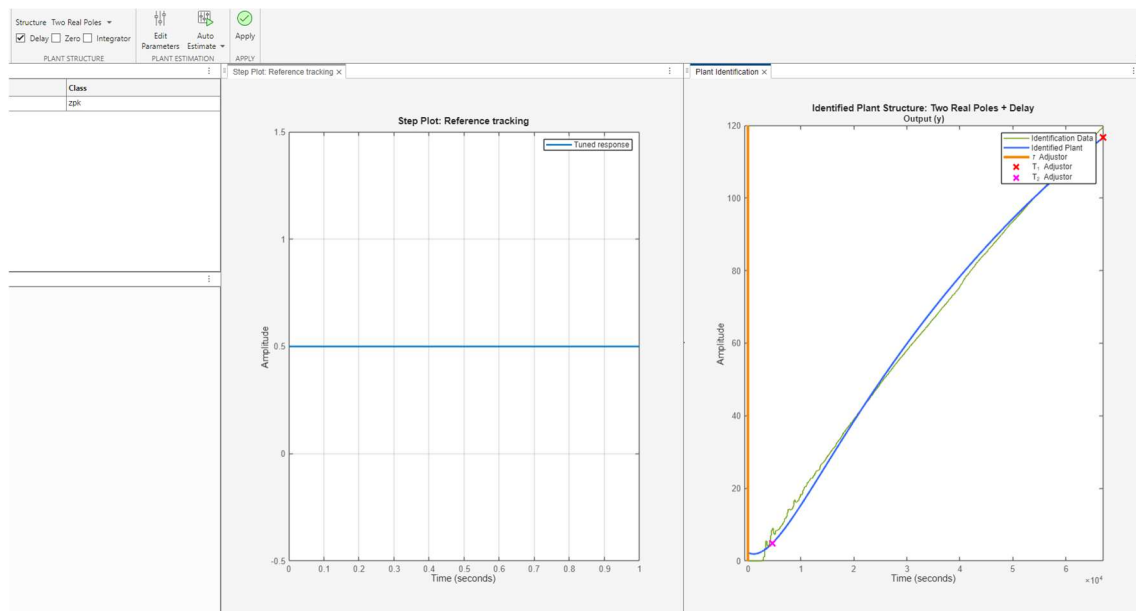
**Figura N° 4.7. Sintonización pid**

Para importar los datos se selecciona en la esquina superior izquierda el botón “plant” y ahí “identificar nueva planta”. Esta nueva identificación de planta corresponde ya a la estimación de los parámetros del control pid que luego se probara en la simulación.



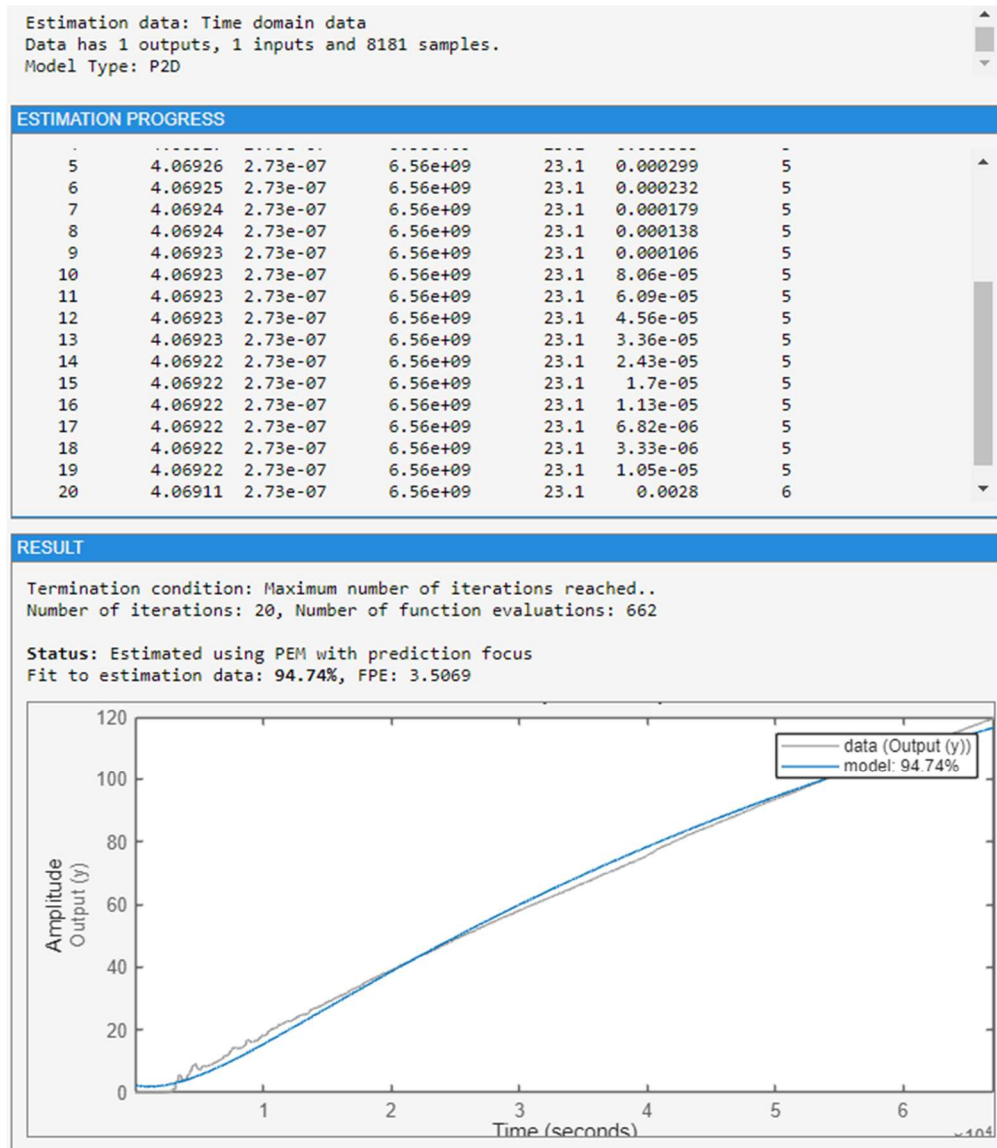
**Figura N° 4.8. Obtencion de datos desde funcion de transferencia**

En la nueva pestaña de identificación de planta lo que hay que hacer es obtener los datos del sistema, como se ve en la figura anterior N° 4.8., se presiona el botón “Get I/O Data”, y se debe seleccionar la opción de respuesta al escalon ya que esta es la que se ha usado como método de prueba, en una nueva ventana se deben ingresar los datos de prueba de la señal escalon tal cual se habian utilizado en la primera identificación del sistema, una amplitud de cien, offset cero, elegir la señal de salida, out.Out y el tiempo de muestreo similar al que le tomo a la primera identificación.



**Figura N° 4.9. Identificación planta**

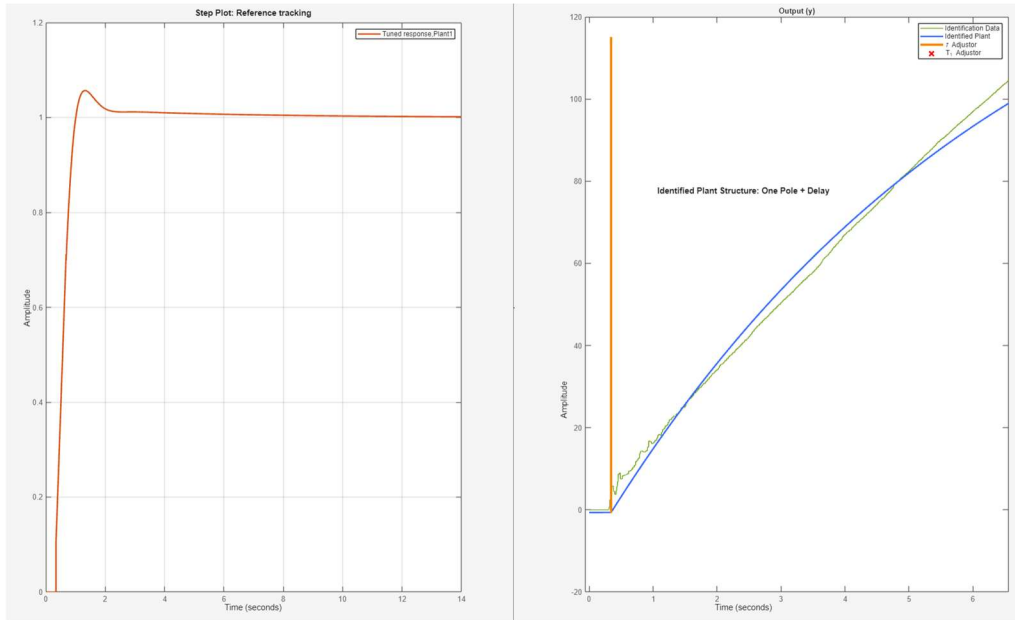
Una vez cargados los datos se verán dos gráficos, como en la figura anterior N° 4.9. el de la izquierda será para la respuesta del sistema calculada con los parámetros PID y a la derecha la comparación entre los datos registrados y los calculados similar a la primera identificación realizada, en la parte superior habrá un botón llamado “autoestimate” el cual al presionarlo ejecutará un cálculo inicial donde se puede ver la aproximación del calculo respecto al proceso simulado, se debe buscar una aproximación similar a la primera obtenida, que si recordamos fue de 96 % aproximadamente, acá para mejorar la estimación se puede experimentar cambiando la estructura de un polo a dos, agregando delay e incluso moviendo manualmente los puntos marcados con x en el grafico de la derecha.



**Figura N° 4.10. Datos estimados**

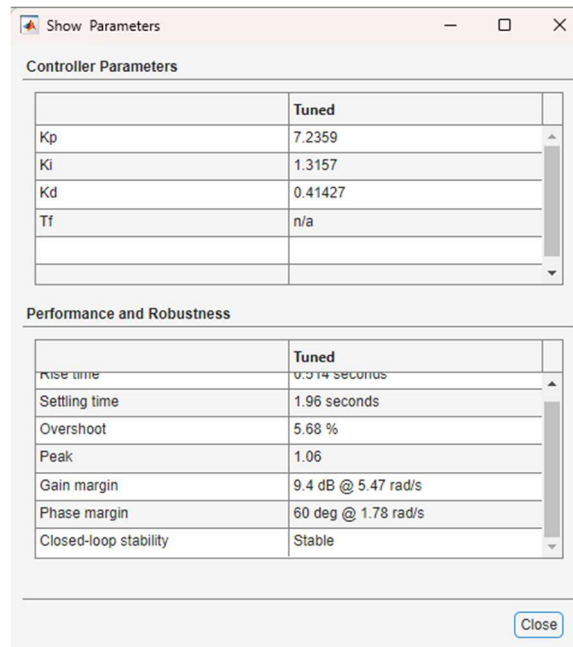
Una vez obtenido una aproximación adecuada, en este caso un 94 % según la figura anterior N° 4.10. Se presiona el botón de “aplicar” en la parte superior y así obtener los datos de la respuesta para el control PID.

Con la estimación aplicada se obtiene una curva característica de la respuesta del sistema ante un control PID, así como se ve en la siguiente figura N° 4.11.



**Figura N° 4.11. Sintonizacion PID**

Luego, al presionar el botón “mostrar parámetros”, se pueden observar los parámetros PID del sistema calculado, además de los valores de la respuesta del sistema, nos muestra el tiempo de subida, el de asentamiento, el porcentaje del sobreimpulso, el factor del peak y adicional nos muestra si esta respuesta es estable.



**Figura N° 4.12. Parámetros sistema**

## CAPÍTULO V OPTIMIZACION Y PRUEBAS DEL SISTEMA

### 5.1. Primera simulación

Ahora que se poseen los parámetros del sistema calculado, se puede comenzar a probarlos y verificar el comportamiento del sistema simulado a comparación del calculado por computadora. Para ello hay que ingresar al bloque PID en simulink los mismos parámetros obtenidos en la sintonización PID, datos mostrados en la figura 4.12. Se hace doble click en el bloque mencionado y se ingresan los valores de la acción proporcional, tiempo integral y tiempo derivativo. El bloque PID posee una variable llamada coeficiente de filtro, predeterminadamente debe dejarse este valor en cero si es que no lo está ya.

Una vez ingresados los valores, se ejecuta la primera simulación para ver los resultados. Para ello se debe dar en el botón “play” de los softwares matlab y factory io, a través del osciloscopio que se había integrado en el control, se puede observar la gráfica del sistema, así como en la figura 5.1.

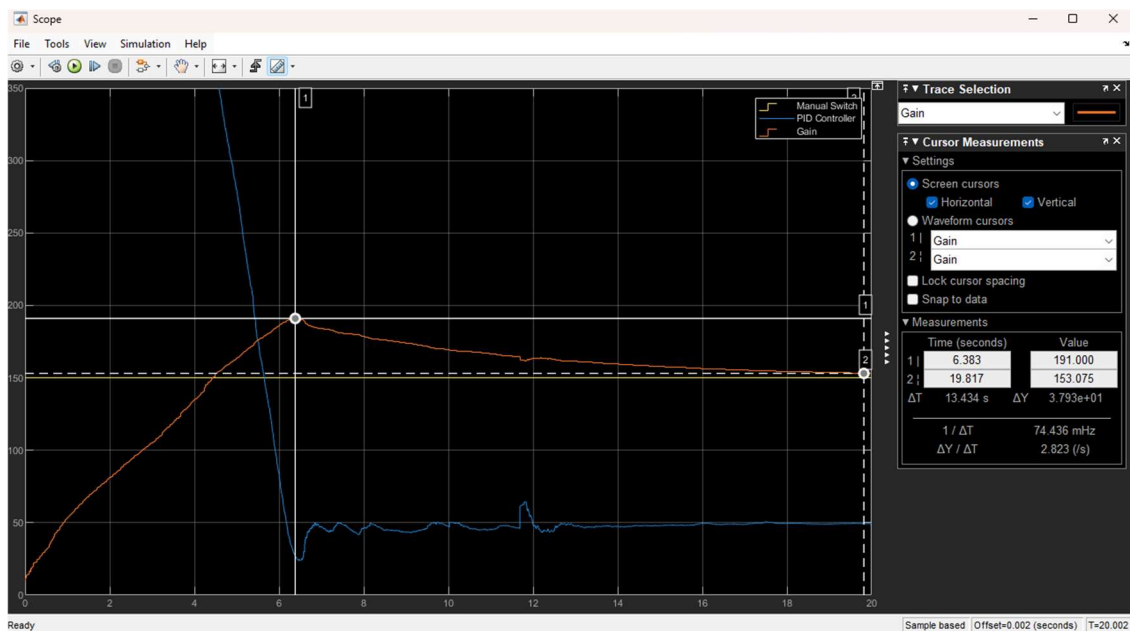


Figura N° 5.1. Primera simulación control PID

A primera vista, se observa una similitud del comportamiento del sistema simulado respecto al calculado, el siguiente paso será realizar una tabla comparativa de ambos sistemas para poder cuantificar con datos numéricos el control automático, para esto se pondrán los valores ya calculados desde el sistema calculado y podremos usar las herramientas del gráfico del osciloscopio para obtener los valores desde el sistema simulado. Recordaremos las características de cada uno de estos valores.

- Tiempo de subida: tiempo aproximado que demora el sistema en llegar al 90% del valor seteado.
- Tiempo de asentamiento: Tiempo que demora el sistema en alcanzar un valor que se mantenga estable cerca del valor seteado
- Sobreimpulso: Valor generalmente en porcentaje del valor máximo que toma la respuesta del sistema medida a partir de la unidad (del valor seteado).

**Tabla 5.1. Comparación primera simulación**

	Sistema calculado	Sistema simulado
Tiempo de subida	0.514 segundos	4.019 segundos
Tiempo de asentamiento	1.96 segundos	15.089 segundos
Sobreimpulso	5.68 %	27.3%

A pesar de que se podría considerar una buena respuesta en la primera simulación del sistema, los valores del sistema simulado se encuentran un poco alejados del calculado. Anteriormente se había mencionado la posibilidad de optimizar el sistema calculado para ajustar la respuesta del mismo. Lo que se hará, será ajustar el sistema calculado y probar nuevos valores en la simulación.

## **5.2. Segunda simulación**

Se realiza un nuevo ajuste al sistema calculado, en virtud de que se ha observado que la respuesta del sistema simulado es algo lenta a comparación de la primera estimación, es que se ajusta el sistema calculado para tener una respuesta más lenta y se toman los nuevos parámetros. Los valores calculados se muestran en la siguiente figura N° 5.2.

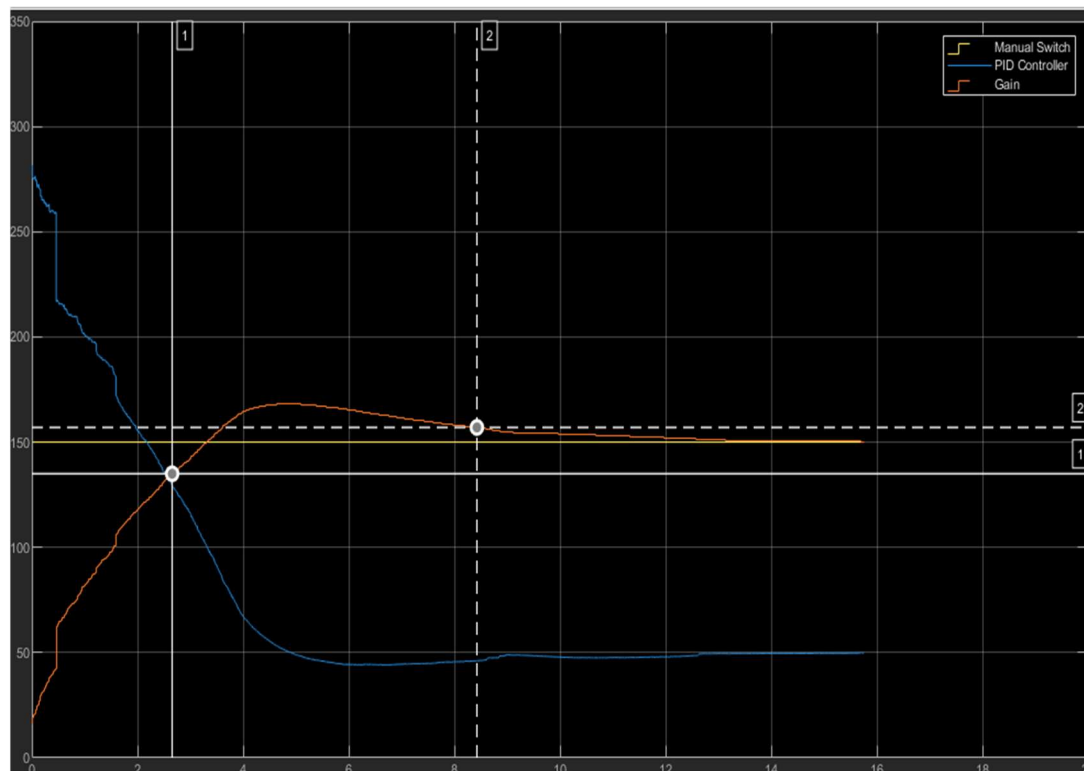
Controller Parameters	
	Tuned
Kp	2.1042
Ki	0.60416
Kd	0
Tf	n/a

Performance and Robustness	
	Tuned
Rise time	2.33 seconds
Settling time	12 seconds
Overshoot	8.82 %
Peak	1.09
Gain margin	18.7 dB @ 4.51 rad/s
Phase margin	67.1 deg @ 0.563 rad/s
Closed-loop stability	Stable

**Figura N° 5.2. Parámetros para segunda simulación**

En la siguiente figura N° 5.3. se puede visualizar la grafica con la respuesta del sistema para la segunda simulación realizada.



**Figura N° 5.3. Segunda simulacion control PID**

A continuación, se agrega la tabla comparativa para la segunda simulación.

**Tabla 5.2. Comparacion segunda simulacion**

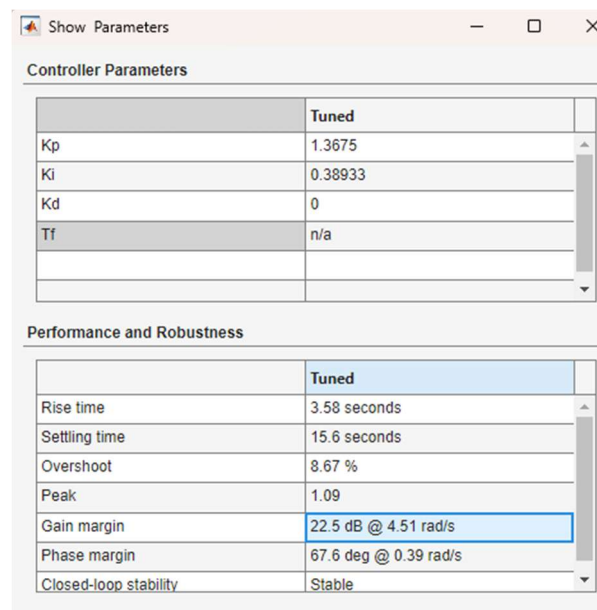
	Sistema calculado	Sistema simulado
Tiempo de subida	2.33 segundos	2.656 segundos
Tiempo de asentamiento	12 segundos	8.427 segundos
Sobreimpulso	8.82 %	12%

Según la tabla 5.2. los valores comparados en este caso se aproximan mucho mas a los calculados, cabe destacar que el nuevo ajuste para el sistema calculado arrojo valores solo de acción proporcional e integral, esto quiere decir que modifíco el control PID inicial en un control PI, esto se puede deber a la sencillez del control de nivel.

A pesar de la proximidad de los valores obtenidos, se realizará una tercera simulación para corroborarlos.

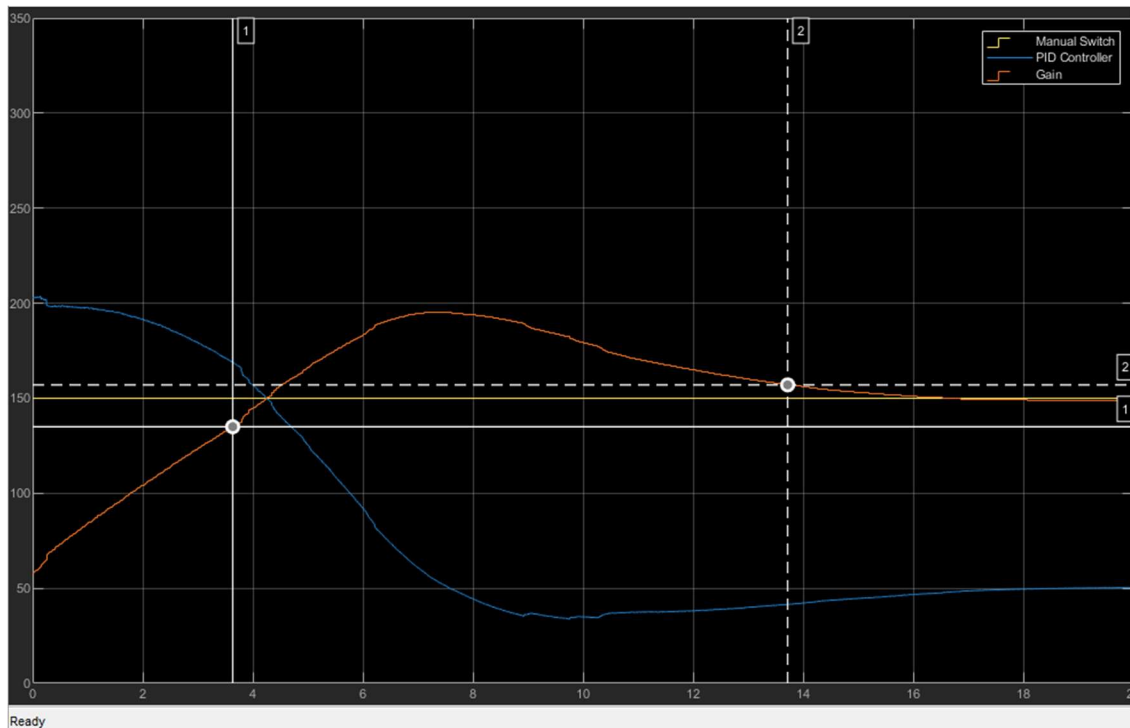
### 5.3. Tercera simulación

Con un nuevo ajuste al sistema calculado se obtienen nuevos parámetros y se ingresan para una nueva simulación. Los valores calculados se muestran en la siguiente figura N° 5.4.



**Figura N° 5.4. Parametros para tercera simulacion**

Se visualiza el grafico de la tercera simulación realizada en la siguiente figura N°5.5.



**Figura N° 5.5 Tercera simulacion control PID**

Y por último se realiza la tabla comparativa.

**Tabla 5.3. Comparacion tercera simulacion**

	Sistema calculado	Sistema simulado
Tiempo de subida	3.58 segundos	3.635 segundos
Tiempo de asentamiento	15.6 segundos	13.719 segundos
Sobreimpulso	8.67 %	30%

Con esta tercera simulación se logró ajustar los tiempos de subida y de asentamiento aún más próximos a lo calculado, sin embargo, el sobre impulso del sistema se alejó bastante de lo esperado.

Considerando que la aproximación en la identificación del sistema no es del 100 % y que alcanzar un valor perfecto es imposible, es que se elegirá los parámetros de la segunda simulación realizada y se harán pruebas al 25, 50, 75 y 100 % del setpoint para verificar en el rango completo de trabajo, la estabilidad del sistema.

Con el setpoint al 25% del valor máximo se llena la siguiente tabla

**Tabla 5.4 Comparación al 25%**

	Sistema calculado	Sistema simulado
Tiempo de subida	2.33 segundos	2.319 segundos
Tiempo de asentamiento	12 segundos	7.5 segundos
Sobreimpulso	8.82 %	6.66%

Con el setpoint al 50% del valor máximo se llena la siguiente tabla

**Tabla 5.5 Comparación al 50%**

	Sistema calculado	Sistema simulado
Tiempo de subida	2.33 segundos	3.615 segundos
Tiempo de asentamiento	12 segundos	12.406 segundos
Sobreimpulso	8.82 %	17.8%

Con el setpoint al 75% del valor máximo se llena la siguiente tabla

**Tabla 5.6 Comparación al 75%**

	Sistema calculado	Sistema simulado
Tiempo de subida	2.33 segundos	4.5 segundos
Tiempo de asentamiento	12 segundos	11.854 segundos
Sobreimpulso	8.82 %	14.3%

Con el setpoint al 100% del valor máximo se llena la siguiente tabla

**Tabla 5.7 Comparación al 100%**

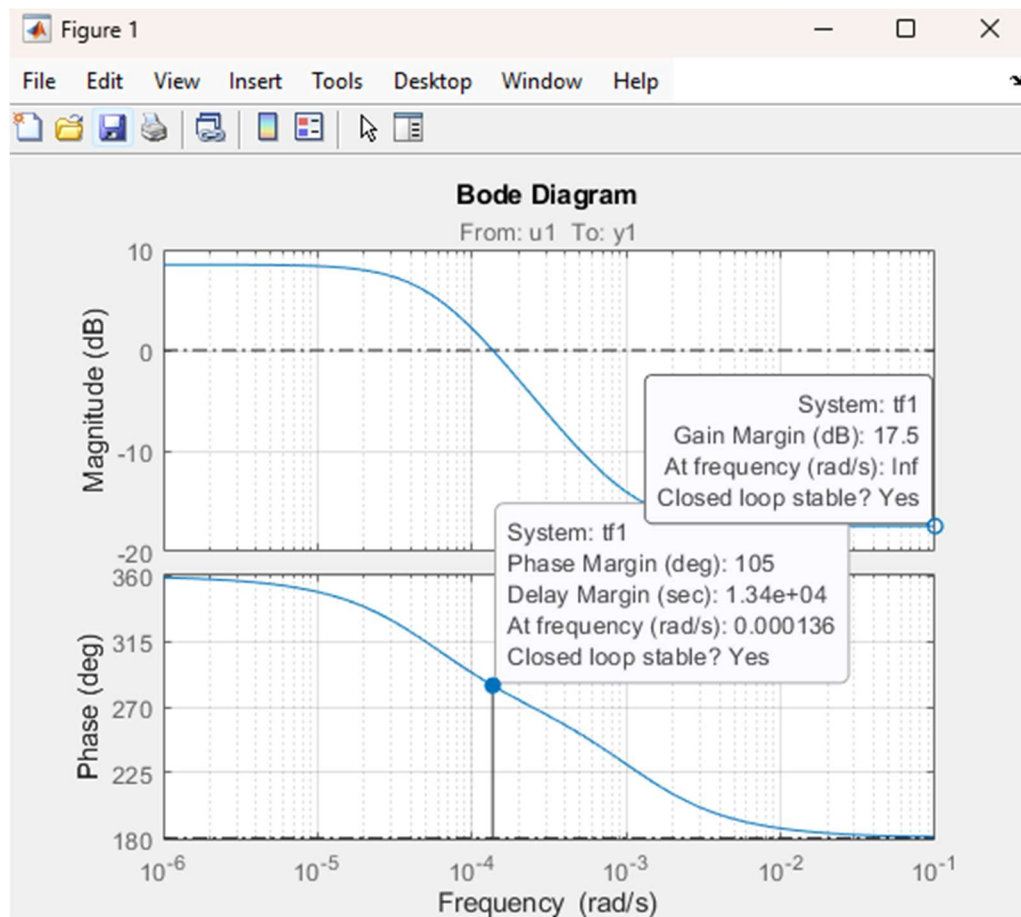
	Sistema calculado	Sistema simulado
Tiempo de subida	2.33 segundos	2.438 segundos
Tiempo de asentamiento	12 segundos	9.083 segundos
Sobreimpulso	8.82 %	3.46%

La respuesta del sistema se mantuvo generalmente estable en el rango de funcionamiento, sin embargo, para determinar la estabilidad de un sistema no basta con mirar solamente los valores entregados por un gráfico, se debe aplicar algo que en control automático se conoce como criterio de estabilidad, podemos aplicar arbitrariamente el diagrama de bode, afortunadamente el software matlab posee esta herramienta de la cual podemos hacer uso.

- Para realizar esto hay que nuevamente ejecutar una identificación del sistema, esta vez, de la respuesta en lazo cerrado, para luego ejecutar un simple comando en la ventana de comandos de matlab. El código es el siguiente: `Bode(tf1), grid`

El comando bode ejecuta el criterio de estabilidad, tf1 es la variable donde se guarda la función de transferencia una vez la estimamos y el comando gris es para graficar en una grilla el comando.

Una vez ejecutado debería abrirse una nueva ventana mostrando el resultado del criterio aplicado, como se observa en la siguiente figura N°5.6.



**Figura N° 5.6 Diagrama de bode**

Como parámetros generales de un diagrama de bode, se establece que para que un sistema sea estable (aunque el recuadro de la figura 5.6. lo diga) el margen de ganancia debe ser mayor a 1.7 y el margen de fase debe ser mayor a 30, por lo que se puede concluir que el sistema en lazo cerrado es estable.

## **CAPÍTULO VI INVERSIÓN Y OPTIMIZACIÓN DEL PROYECTO**

### **6.1. Costos de implementación**

En un entorno industrial real, la implementación de un sistema de control optimizado puede generar beneficios significativos en la producción. La automatización y ajuste preciso de variables como el nivel y flujo dentro de los procesos productivos pueden mejorar la eficiencia operativa, reducir tiempos de inactividad y minimizar el consumo innecesario de recursos. Esto, a su vez, contribuye a una mayor estabilidad del proceso y una reducción en pérdidas productivas.

Para la implementación de este proyecto sobre un proceso real, la empresa o compañía deberá adquirir la licencia oficial del software MATLAB, una herramienta fundamental para el diseño, simulación y optimización de sistemas de control. MATLAB permite realizar análisis avanzados de procesos industriales, facilitando la identificación de modelos matemáticos, la sintonización de controladores y la evaluación de la respuesta dinámica de los sistemas. Su integración con otros entornos de simulación y comunicación industrial lo convierte en una opción versátil para la automatización de procesos. Los costos de adquisición de licencia son los siguientes:

- MATLAB estándar duración perpetua – 2535 USD
- MATLAB estándar duración anual – 1015 por año

Dependiendo del uso que se requiera, se puede elegir una licencia o la otra y así asumir distintos costos. Si bien este costo representa una inversión inicial, su recuperación puede lograrse mediante la optimización de los procesos controlados. En una planta concentradora de cobre, por ejemplo, la implementación de un sistema de control eficiente puede traducirse en una mayor estabilidad operativa, reducción de tiempos de inactividad y optimización del consumo de insumos como reactivos y energía. La automatización precisa del nivel y flujo dentro de los circuitos de molienda y flotación puede mejorar la recuperación del mineral, evitando pérdidas por variaciones en la alimentación o errores en la dosificación. Además, al minimizar la intervención manual

en la sintonización de los controles, se reduce el riesgo de ajustes incorrectos que podrían comprometer la eficiencia del proceso. En este contexto, la inversión en MATLAB no solo se justifica por sus capacidades técnicas, sino también por el impacto positivo en la rentabilidad y sostenibilidad de la operación minera.

## **6.2. Visión futura de proyecto**

Este trabajo de titulación ha demostrado la viabilidad de diseñar y sintonizar un control PID utilizando simulación computacional, lo que permite evaluar su desempeño sin comprometer la estabilidad de un proceso real. Aunque en esta instancia la implementación se ha llevado a cabo en un entorno simulado, el mismo enfoque puede trasladarse a aplicaciones industriales concretas, proporcionando una herramienta confiable para el diseño de sistemas de control automático.

El método de sintonización propuesto no se limita únicamente al control de nivel; su aplicación es extensible a cualquier tipo de proceso que opere con controladores PID. Sistemas de temperatura, presión, caudal, velocidad y hasta el posicionamiento de actuadores pueden beneficiarse de este enfoque al permitir la determinación de parámetros óptimos sin necesidad de interferir directamente en la operación de la planta. Esto resulta especialmente útil en industrias donde la experimentación directa es riesgosa o costosa, como la minería, la química y la manufactura.

En una visión futura, este proyecto podría evolucionar hacia una metodología estándar dentro de la industria, integrando simulaciones avanzadas con datos reales para mejorar la precisión en la identificación de sistemas. Además, con el desarrollo de tecnologías como la inteligencia artificial y el aprendizaje automático, es posible incorporar modelos predictivos que ajusten automáticamente los parámetros del controlador PID en función de cambios en las condiciones del proceso.

La implementación en un entorno real no solo es posible, sino altamente recomendable, especialmente en industrias que buscan optimizar sus procesos sin comprometer la producción. Al validar los parámetros de control en una simulación antes de aplicarlos a la planta, se minimizan los riesgos asociados con la puesta en marcha,

asegurando estabilidad y eficiencia desde el primer momento. Este enfoque permite que las empresas adopten estrategias de automatización más seguras, mejorando su competitividad y reduciendo costos operativos. Así, lo que comenzó como una simulación académica puede convertirse en una solución aplicada para la optimización de procesos industriales en el futuro.

## **CAPITULO VII CONCLUSIONES**

De manera específica, se logró diseñar el entorno de control entre los diversos softwares, estableciendo una integración funcional y efectiva. Además, se realizó el análisis de la respuesta transitoria del sistema en lazo abierto, identificando sus características principales y evaluando su desempeño sin control. A partir de este análisis, se estimó la función de transferencia del sistema, obteniendo una representación matemática que permitió modelar su comportamiento. Con esta función, se llevó a cabo la sintonización del controlador PID, aplicando métodos que garantizaron un ajuste óptimo y eficiente. Finalmente, se analizó el comportamiento del sistema utilizando los datos calculados, verificando la estabilidad y precisión del control implementado.

En términos generales, se logró el diseño del control automático de nivel en un entorno simulado utilizando MATLAB como herramienta principal. Aunque la idea inicial era sintonizar el control automático sin realizar pruebas sobre el proceso real, se identificó la importancia de contrastar las estimaciones del sistema simulado con datos provenientes del proceso real, para garantizar que los cálculos realizados por el software se aproximen lo más posible a la realidad. Sin embargo, las pruebas realizadas demostraron que el sistema no se inestabilizó en ningún momento, evidenciando que las estimaciones del modelo computacional son suficientemente precisas para evitar la necesidad de ajustes completamente manuales. Esto resulta especialmente relevante en aplicaciones donde se requiere alta precisión, como el control de nivel o pH en una celda de flotación de cobre, la carga de alimentación en un molino SAG, entre otros, ya que optimizar controles manualmente puede incrementar el riesgo de inestabilidad y generar pérdidas productivas.

Por último, MATLAB se destacó como una herramienta integral en este proyecto, facilitando el diseño, análisis, control y optimización del sistema, gracias a su versatilidad y la amplia gama de funciones que ofrece, haciendo más eficiente el trabajo en proyectos de automatización.

## BIBLIOGRAFIA

- Creus, A. (2010). Regulación automática. En Instrumentación industrial (8va ed., pp. 499-642). Alfaomega.
- Ogata, K. (2010). Análisis de la respuesta transitoria y estacionaria. En Ingeniería de control moderna (5ta ed., pp. 159-268). Pearson.
- Ogata, K. (2010). Análisis y diseño de sistemas de control por el método de la respuesta en frecuencia. En Ingeniería de control moderna (5ta ed., pp. 398-566). Pearson.
- MathWorks. (s.f.). Bode. Centro de ayuda. <https://la.mathworks.com/help/control/ref/dynamicsystem.bode.html>
- DRONIC-32. (2021, enero 18). Implementación de OPC en Simulink para identificación y control de planta con Factory I/O+LabView [Video]. YouTube. <https://www.youtube.com/watch?v=ZvdaQfa8jRs&t=3576s>