

Proyecto final

Caamiña Quineros, Daniela Beatriz Yapura, Cristian Alejandro

14 de mayo de 2021

Índice

1. Introducción	4
2. Objetivo	5
3. Túnel	6
3.1. Historia del Túnel UNPSJB	6
3.1.1. Motor y Variador de velocidad	7
4. Pruebas iniciales	8
4.1. Ecuación velocidad del aire	9
4.2. Filtros	10
4.3. Error observado	10
5. Desarrollo	12
5.1. Captura de datos por puerto serie	12
5.2. Estimación de la planta	13
5.2.1. Diagrama de trabajo	13
5.3. Método de estimación	14
5.4. PID inicial	14
6. Planos Eléctricos	14
7. Recomendaciones futuras	14
8. Conclusión	14
9. Bibliografía	14

Índice de figuras

2.1. Diagrama	5
3.1. Tunel uso recreativo	6
3.2. Clasificación túneles	6
3.3. Tunel UNPSJB	6
3.4. LS650	7
4.1. Placa con sensores	8
4.2. Datos y diversos filtros	10
4.3. Datos y filtro mediana	11
4.4. Curva aceleración y desaceleración	11
5.1. Pantalla Processing	13
5.2. Diagrama de bloques del procedimiento de modelado de la planta . .	13
5.3. Diagrama en bloques	14

1. Introducción

Actualmente, en el Laboratorio de Fluidos de la Universidad, se utiliza el Túnel del Viento para realizar el contraste de anemómetros y experimentos para distintas materias. Gran parte de estas aplicaciones requieren que se conozca la velocidad del fluido. Por lo tanto, variación de presión, humedad, presión atmosférica y temperatura son variables requeridas para lograr estimarla con mayor precisión. Cada variable debe ser medida de forma manual con sus respectivos instrumentos para luego ingresar estos valores a una tabla (generada de forma estadística) y obtener una estimación de la velocidad del fluido. Para realizar distintas mediciones, se utilizaba un control de velocidad a lazo abierto en el que se modifica la resistencia del motor, cambiando la velocidad del aire en pasos discretos. Actualmente, desde principios del año 2020 se utiliza un variador de velocidad de la marca Long Shenq. Realizar este proceso de forma manual, se torna engorroso y poco práctico para la realización de varias mediciones.

2. Objetivo

Generar un lazo de control que tenga como entrada la velocidad de referencia, y que contemple las perturbaciones externas del sistema en el cálculo de la velocidad de salida que se utilizará como lazo de realimentación. Adaptar la acción de control que ingresa al variador de velocidad (adquirido por el Laboratorio de Fluidos) para alimentar al motor y dejar en desuso el banco de resistencias que se utiliza. Además, realizar una interfaz gráfica para un mejor manejo y control del sistema.

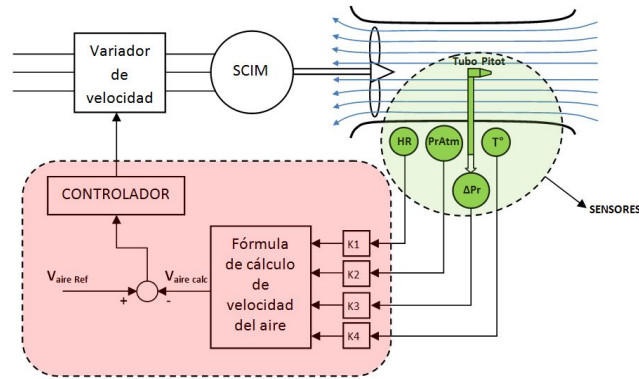


Figura 2.1: Diagrama

3. Túnel



Figura 3.1: Túnel uso recreativo

Un túnel de viento es una herramienta que puede tener dos fines hoy en día, ya sea para un uso recreativo o propósito científico. Como uso científico se utiliza para observar los efectos del movimiento de aire al rededor de objetos sólidos, como también para la calibración de anemómetros.

Los túneles de viento se pueden clasificar en túneles abiertos o cerrados y a su vez pueden ser verticales u horizontales.

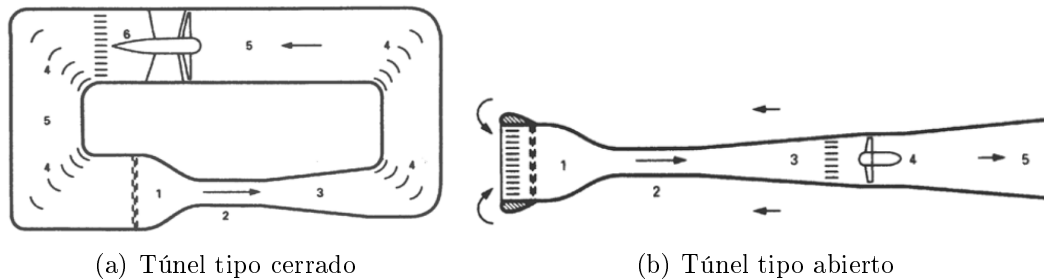


Figura 3.2: Clasificación túneles

3.1. Historia del Túnel UNPSJB

¹ El túnel aerodinámico del Laboratorio de Mecánica de Fluidos (LMF) de la Facultad de Ingeniería de la Universidad Nacional de la Patagonia San Juan Bosco (UNPSJB) es un circuito abierto (tipo Eiffel) con cámara de ensayos cerrada. Puede clasificarse como un túnel “pequeño de baja velocidad”, con una longitud total de 11m, una velocidad máxima de 18 m/s y una cámara de ensayos con un área de 0,8m².



Figura 3.3: Tunel UNPSJB

La entrada del túnel cuenta con canalizadores, comúnmente denominados "panal de abejas", que favorecen la formación de un flujo uniforme y homogéneo propiciando mejores resultados en los experimentos.

La cámara de ensayos es vidriada para poder observar con claridad el flujo y está incorporada en un módulo extraíble del túnel, lo cual permite fácil acceso para el armado de los distintos objetos a ensayar.

La variación de la velocidad del aire dentro de la cámara se consigue por dos vías: modificando la velocidad del motor para lograr una aproximación, y mediante la apertura de compuertas ubicadas entre el rodete y la zona de ensayo, para el ajuste fino. La toma aire desde el exterior a través de las compuertas actúa como by-pass, modificando el flujo principal del túnel y controlando su velocidad.

¹<http://www.ing.unp.edu.ar/mecanica/Paginas/Tunel.htm>

Los distintos ensayos que se realizan en el túnel son:

- Determinación de coeficientes de resistencia y sustentación de distintos cuerpos y perfiles aerodinámicos.
- Determinación de distribución de presiones a través de diferentes objetos como perfiles aerodinámicos, edificios, puentes, automóviles, etc.
- Visualización con humo del flujo a través de distintos obstáculos.
- Estudio del comportamiento dinámico de generadores eólicos.
- Calibración de anemómetros.

3.1.1. Motor y Variador de velocidad

Variador de velocidad

Es utilizado para controlar la velocidad de giro de un motor. Para regular las revoluciones, se debe tener en cuenta las características del motor, ya que este tiene una curva propia de funcionamiento. Un variador es capaz de generar elementos control de aceleración, frenado, seguridad, control del torque y operaciones que mejoran la eficiencia energética.

² El motor que se utiliza en para hacer funcionar el ventilador corresponde a uno de la marca AEG, de 30 kW de potencia. Mientras que el variador de velocidad, es de la marca Long Shenq.

Variador modelo LS650-4045 para una potencia de 45kW con salida trifásica. Capaz de ser controlado por entradas de corriente, tensión, PID interno, potenciómetro o de forma manual con botones en su panel frontal.



Figura 3.4: LS650

²http://www.ing.unp.edu.ar/mecanica/Archivos/Fluid/LMF/Informe_Proyecto_-_Túnel_de_Viento.pdf

4. Pruebas iniciales

I2C

Es un puerto y protocolo de comunicación serial, define la trama de datos y las conexiones físicas para transferir bits entre 2 dispositivos digitales. El puerto incluye dos cables de comunicación, SDA (Datos seriales) y SCL (reloj serial). Además el protocolo permite conectar hasta 127 dispositivos esclavos con esas dos líneas, con hasta velocidades de 100, 400 y 1000 kbits/s.

Para comenzar con las pruebas, lo primero que se realizó es la conexión de diversos sensores en una protoboard. Se obtuvieron datos para elegir cuál convenía utilizar, ya que se hicieron pruebas con varios sensores de temperatura, humedad y presión atmosférica.

Cabe destacar que para la comunicación de estos sensores se utilizó **comunicación I2C**, que permitió, a través de dos líneas de comunicación, conectar, en una primera instancia, 5 elementos al mismo bus.

Como paso siguiente se eligieron los sensores. Los elementos que se decidió utilizar generaban menor variación y menor error en comparación con el instrumento **TESTO 435** y estos fueron los siguientes:

- **BME280**: Sensor de presión atmosférica, temperatura y humedad relativa.
- **SI7021**: Sensor de temperatura y humedad relativa.
- **MPXV7002**: Sensor de presión diferencial.
- **ADS1115**: Convertidor analógico digital 16bits.

Una vez que se tuvo seguridad con los sensores elegidos se procedió a realizar una placa con estos elementos para que no se desconecten y no se produzcan errores como solía suceder mientras estaban en la protoboard.

A raíz de varias mediciones y realizando el contraste con el instrumento **TESTO**

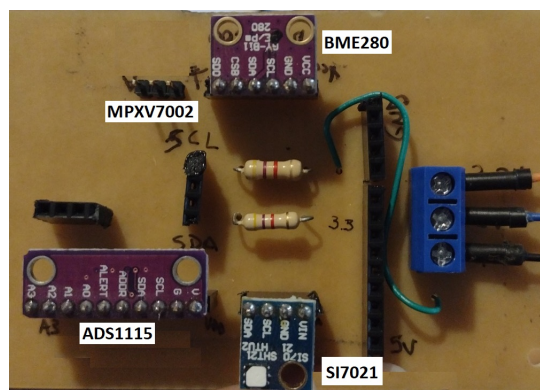


Figura 4.1: Placa con sensores

435, durante distintos días, se decidió agregar un offset al programa de Arduino para que luego el cálculo de la velocidad del aire no esté desfasado. **ver si se agrega esto**

Estas mismas pruebas sirvieron para probar y corroborar el funcionamiento del

MPXV7002, un sensor de diferencia de presión de alto costo en el país. Este sensor es capaz de medir de -2kPa a 2kPa en un rango de 4V (0.5 a 4.5V), para este proyecto se utilizó la parte positiva de los valores por lo que la resolución estaba establecida por los 2V de rango y 2kPa. Estos datos se ingresaron al programa Arduino a través de un conversor ADS 1115 con comunicación I2C.

No se utilizó el ADC interno del microcontrolador Arduino Uno ya que es de 10 bits y el externo de 15 bits más un bit de signo, este mismo posee un amplificador de ganancia programable (PGA) que establece la escala completa, es decir, indica el valor de referencia. En Arduino este valor viene determinado por el voltaje de referencia que en el caso de Arduino UNO es 5V. En el ADS1115 lo establece el PGA. Por defecto este valor de referencia es $\pm 6,144$ V, quiere decir que el valor de 32.677 (valor máximo con 15-bit) corresponde a 6,144 V.

Ejemplo Factor de escala de Arduino UNO:

$$Factor\ de\ escala = \frac{5\ V}{1023} = 0,0048875\ V = 4,88\ mV$$

Ejemplo Factor de escala de ADS1115:

$$Factor\ de\ escala = \frac{6,144\ V}{32677} = 0,0001875\ V = 0,1875\ mV$$

En la siguiente tabla se tiene un resumen de los valores posibles de factor de escala para ADS1115.

PGA	Referencia (V)	Factor de Escala (mV)
2/3	6,144	0,1875
1	4,096	0,125
4	1,024	0,0312
8	0,512	0,0156
16	0,256	0,0078

Aunque la mayor referencia sea de 6,144 V, el ADS1115 sólo puede medir el valor de alimentación (VDD) más 0,3 V por sus pines analógicos. **ver porq** pin ingresa la señal en proteus. Por lo tanto, si alimentamos el ADS1115 con 5V, sólo podremos medir tensiones por los pines analógicos hasta $5V + 0,3V$ es decir, 5,3V. Lo mismo ocurriría si alimentamos con 3,3V. En este caso el máximo sería 3,6V. **(con que** alimentamos? ver en proteus)

En el caso de este proyecto, se utilizó un valor de PGA de 1, tomando como voltaje de referencia 4,096V.

$$Factor\ de\ escala = \frac{4,096\ V}{32677} = 0,000125\ V = 0,125\ mV$$

4.1. Ecuación velocidad del aire

El Laboratorio de Mecánica de Fluidos, antes de comenzar con este proyecto utilizaba un archivo Excel para hacer la corrección de la velocidad del aire. En este se calcula matemáticamente la densidad del aire en función de la presión, temperatura y humedad atmosférica. Esta ecuación (1) fue desarrollada dentro del programa de Arduino para observar como dato final la velocidad del aire.

$$\rho = \frac{3,48353 \cdot 10^{-3}\ kg\ K\ J^{-1} \cdot p \cdot (1 - 0,378 \cdot x_v)}{Z \cdot T} \quad (1)$$

$$x_v = \frac{(\alpha + \beta \cdot p + \gamma \cdot t^2) \cdot (1Pa \cdot e^{AT^2 + BT + C + D/T}) \cdot h / 100}{p}$$

$$Z = 1 - \frac{p}{T} \cdot [a_0 + a_1 t + a_0 t^2 + (b_0 + b_1 t)x_v + (c_0 + c_1 t)x_v^2] + (d + x_v^2) \frac{p^2}{T^2}$$

dónde:

- **p** [Pa] es la presión atmosférica medida,
- **t** [°C] es la temperatura medida,
- **T** [K] es la temperatura absoluta (**T**=**t** + 273,15 K)
- **h** [%] es la humedad relativa medida,
- y las constantes **A,B,C,D, α, β, γ, a₀, a₁, a₂, b₀, b₁, c₀, c₁, D**.

4.2. Filtros

Mediana

Es una técnica de filtrado digital no lineal que suele utilizarse para eliminar el ruido de una imagen o señal. La mediana es el número que está justo en el medio de un conjunto de datos ordenados de menor a mayor o de mayor a menor. La idea principal del filtro de mediana es recorrer la señal entrada, sustituyendo cada dato por la mediana de una ventana de “N” datos.

Una vez que se procedió a tomar diversos valores, se notó necesario la implementación de un filtro. Para esto se utilizaron varias librerías de Arduino para generar distintas pruebas (Figura 4.2).

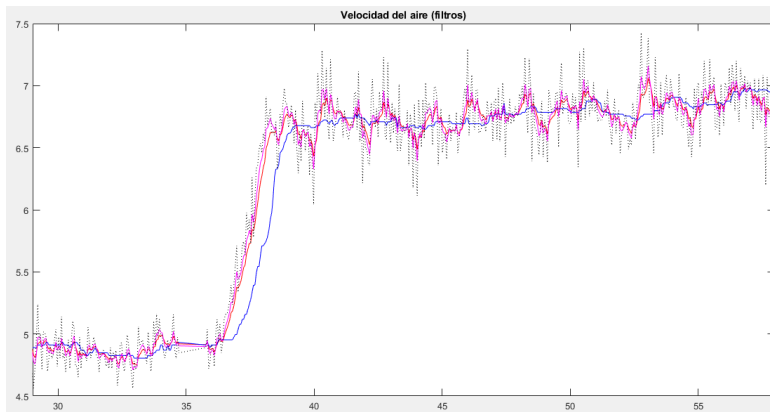


Figura 4.2: Datos y diversos filtros

Luego de varias pruebas, se eligió un filtro de mediana (Figura 4.3) con una ventana 40, que producía menor ruido en la velocidad del aire. Se implementó en el programa de Arduino utilizando la una función preestablecida.

4.3. Error observado

Las pruebas anteriores eran realizadas con el uso normal que se le daba al variador de velocidad utilizando el panel digital frontal. Luego, para comenzar con las pruebas de otros modos de funcionamiento del variador, se utilizó el potenciómetro frontal. Al utilizar este modo se originó error en la aceleración. Al hacer las averiguaciones pertinentes, esto se debió a una configuración interna del variador: si se utiliza con

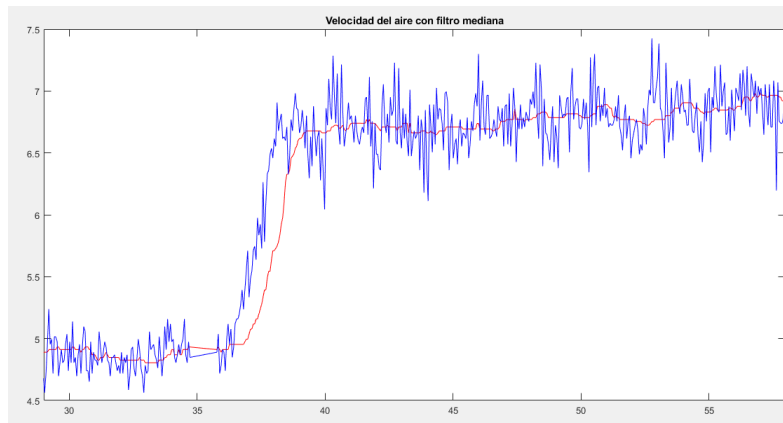


Figura 4.3: Datos y filtro mediana

el panel digital frontal, la curva de aceleración y desaceleración sigue una “s”, no realizando un cambio brusco en la velocidad del motor, en cambio, para el uso del potenciómetro u otro modo de funcionamiento la curva es lineal . Para revertir esto, se modificó el tiempo de aceleración y desaceleración a 20 segundos.

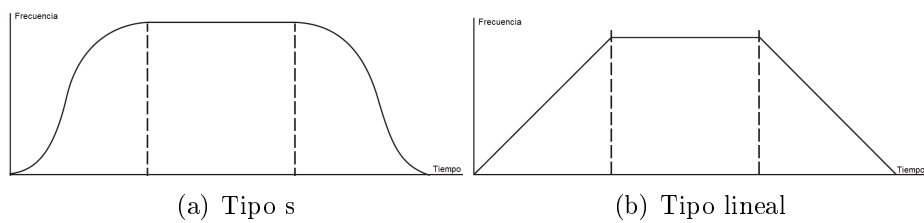


Figura 4.4: Curva aceleración y desaceleración

5. Desarrollo

Una vez corroborado que el error ya no ocurrió nuevamente, se procedió a ingresar la señal del lazo de corriente para controlar el variador de velocidad.

Para la comunicación del Arduino con el variador de velocidad se decidió utilizar un lazo de corriente de 0-20mA, este tiene ventajas sobre el lazo de tensión ya que el utilizado es más estable en largas distancias y más inmune a los ruidos eléctricos e interferencias electromagnéticas respecto al lazo de tensión. Normalmente, se utilizan lazos de corriente de 4-20mA para poder observar si hubiera fallas en el circuito, pero este modelo de variador, tiene el piso del lazo de corriente en 0mA. Cabe mencionar, que el variador podía ser controlado por cualquiera de los dos métodos de lazos.

A continuación se observará la placa interna del variador con sus respectivas borneras por donde ingresará la señal de corriente del lazo de control y el jumper que se tuvo en cuenta para que sea de corriente y no de tensión.

IMAGEEEN

La señal para controlar el variador de velocidad fue generada por una señal PWM estipulada a través de la librería `TIMEROne`. La instrucción `Timer1.initialize(period)` de Arduino inicializa el timer con el valor de `period`, este valor es el tiempo en el que se dispara el temporizador y en el caso de este proyecto es de 40 microsegundos. El otro comando que se utilizó fue `Timer1.pwm(pin, duty)` que establece el número de pin, en este caso pin 9 y “duty” es un valor entre 0 y 1023 establecido por la programación. Esta señal de salida, fue necesario transformarla al lazo de corriente utilizado para establecer la señal al variador de velocidad. Esto se realizó a través de una placa adaptadora con un filtro generada por nosotros.

PROTEUS PLACA

5.1. Captura de datos por puerto serie

Processing

Es un lenguaje de programación basado en Java, aunque hace uso de una sintaxis simplificada y de un modelo de programación de gráficos.

Para capturar los datos, primeramente se utilizó Matlab, como se necesitó visualizar los valores en tiempo real y luego guardar la tabla con vectores, este programa producía errores en la captación de datos. Para solucionar el inconveniente anteriormente nombrado, se utilizó un código generado en Processing. Con este código, se capturó los valores, se observaba el valor numérico estimado de la velocidad y se lograba ingresar el valor del escalón requerido para tomar la planta. (Figura 5.1)

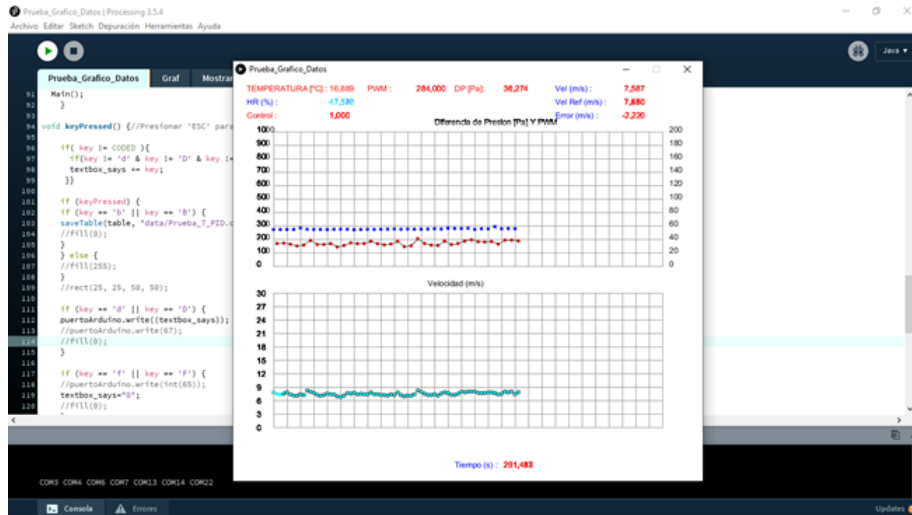


Figura 5.1: Pantalla Processing

5.2. Estimación de la planta

5.2.1. Diagrama de trabajo

Para realizar la estimación de la planta se genera un diagrama de bloques del procedimiento que se siguió de forma resumida.

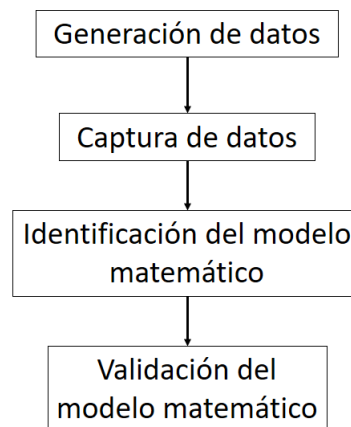


Figura 5.2: Diagrama de bloques del procedimiento de modelado de la planta

Generación de datos: se hizo diversas pruebas para obtener la mayor cantidad de información en la respuesta.

Captura de datos: A través del puerto serie y con Processing se realiza el almacenamiento de los datos de respuesta del sistema ante el estímulo de las señales de excitación. Posteriormente, el análisis de los datos y la generación de las gráficas correspondientes es realizado por medio de rutinas de código implementadas en Matlab.

Identificación del modelo matemático: Se utilizaron varios métodos numéricos para la estimación.

Validación del modelo matemático: una vez obtenida la mejor estimación, se efectuó una validación adicional a partir de la comparación de datos experimentales con los teóricos generados por escalones.

5.3. Método de estimación

ver si ponemos este diagrama

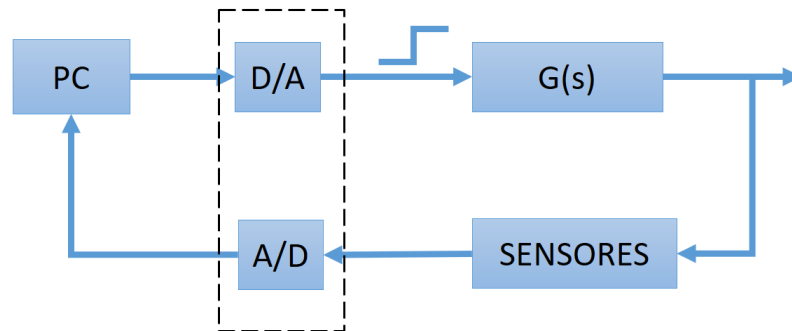


Figura 5.3: Diagrama en bloques

Una vez que se determinó valores de ventana de filtro, velocidad de conmutación de PWM, tiempos de aceleración y desaceleración, etc. Se utilizó el modo de ingreso de señal por lazo de corriente en el variador de velocidad. Para realizar la estimación de la planta se obtuvieron y guardaron tablas de datos con Processing, generando diversas señales con distintos escalones de entrada. PONER PRUEBAS VALOR DEL ESCALON, VALOR DE FRECUENCIA.

El archivo generado fue un “.csv” para el cual luego fue necesario utilizar un código de Matlab para obtener cada vector según los datos que poseía cada uno. Con estos datos, se realizó la estimación de la planta por los métodos de ... y de..... concluyendo que el método JSadas de cuarto orden era suficiente.

1.GRAFICO Y ECUACION DE JASKAC. 2.Grafico de los graficos con los valores marcados que se tomaron

Para corroborar la elección de la planta se genreó un archivo en simulink generando como entrada los esclaoes que se colocaron en el sistema.

IMAGEN SIMULINK

FUNCION DE TRANSFERENCIA

5.4. PID inicial

6. Planos Eléctricos

7. Recomendaciones futuras

8. Conclusión

9. Bibliografía