



**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**

**Facultad de Ingeniería en Electricidad y Computación**

**“MODELAMIENTO Y SIMULACIÓN DE UN SISTEMA ELECTRO-  
NEUMÁTICO”**

**INFORME DE MATERIA DE GRADUACIÓN**

Previo a la obtención del título de:

**INGENIERO EN ELECTRICIDAD ESPECIALIZACIÓN  
ELECTRÓNICA Y AUTOMATIZACIÓN INDUSTRIAL**

Presentada por:

Pedro Andres Bazurto Velasco

Mercedes Noemí Murillo Abril

**GUAYAQUIL – ECUADOR**

**AÑO: 2015**

## AGRADECIMIENTO

A Dios por brindarme sabiduría, salud y las fuerzas para superarme cada día. A mis padres por darme la oportunidad de cumplir con mis metas y ser ese apoyo constante para poder trazarme nuevas. A todos los profesores que me han guiado durante todos mis años de estudios en ESPOL. En especial al Ing. Manzur por compartir su pasión por la Ingeniería. A Mercedes mi compañera de trabajo por ayudarme a conseguir la conclusión de una etapa más de mi vida.

*Pedro Andres Bazurto Velasco*

## AGRADECIMIENTO

A Dios por el don de la vida, la salud y la ciencia. A mis padres y hermana por facilitarme su apoyo y guía constante a lo largo de mi carrera. A todas las personas que han contribuido con su granito de arena en mi formación como persona y profesional. Siempre las tendré presentes en mi mente, mi corazón y mi vida. Y de manera particular a Pedro Bazurto mi amigo y compañero con quien he contado incondicionalmente y compartido la experiencia de concluir esta etapa de mi vida.

*Mercedes Noemí Murillo Abril*

## DEDICATORIA

Por la sonrisa de mi madre que vale un millón. Por llenar de orgullo a mi padre. Por mis hermanos esperando ser un ejemplo para que ellos aprecien que con sacrificio y esfuerzo se puede conseguir cualquier meta que se propongan. Por todas las personas que creyeron en mí. Por todos mis amigos que siempre me apoyaron. Les dedico este trabajo.

*Pedro Andres Bazurto Velasco*

## DEDICATORIA

A Dios. A mis padres. A mi hermana. A mis profesores. A mis amigos y amigas. Y a quienes me impulsaron a ser mejor cada día. Les dedico este trabajo como muestra de que sin importar las circunstancias solo nosotros somos dueños de nuestro propio destino y amos de nuestros logros. Nada es imposible.

*Mercedes Noemí Murillo Abril*

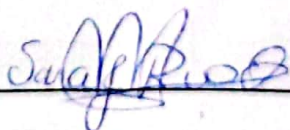
## TRIBUNAL DE SUSTENTACIÓN



---

Ph.D. Douglas Plaza G.

PROFESOR DE LA MATERIA DE GRADUACIÓN



---

M.Sc. Sara Ríos O.

PROFESOR DELEGADO POR LA UNIDAD ACADÉMICA

## DECLARACIÓN EXPRESA

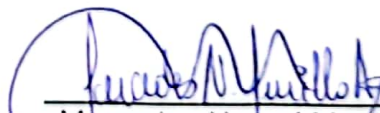
“La responsabilidad del contenido de este Informe, nos corresponde exclusivamente; y el patrimonio intelectual de la misma a la Escuela Superior Politécnica del Litoral”

(Reglamento de Graduación de la ESPOL)



---

Pedro Andres Bazarro Velasco



---

Mercedes Noemi Murillo Abril

## RESUMEN

El presente trabajo tiene como objetivo modelar y simular un sistema electro-neumático evaluando el desempeño de los programas de simulación a utilizar. Los tres programas en donde se implementó el modelo son Matlab/Simulink, Scilab/Xcos y OpenModelica. Como primer paso seleccionamos la válvula reguladora de presión como el sistema electro-neumático a modelar.

En la primera parte describimos los procedimientos realizados para obtener el modelo matemático del sistema electro-neumático. Uno de los requisitos para obtener dicho modelo es el conocimiento profundo del sistema, la válvula reguladora de presión, por lo que definimos sus características y funcionamiento. También identificamos las variables de interés del caso de estudio y delimitamos el alcance del modelo que obtuvimos.

Para la segunda parte trasladamos el lenguaje matemático a los diferentes entornos de programación. En este proceso definimos los parámetros útiles y las limitaciones que tomamos en cuenta para la simulación. Con la ejecución de estos programas obtuvimos resultados para hacer una evaluación cualitativa y cuantitativa del desempeño de los simuladores para compararlos y establecer sus ventajas y desventajas.



## ÍNDICE GENERAL

	Pág.
AGRADECIMIENTO .....	II
DEDICATORIA .....	IV
TRIBUNAL DE SUSTENTACIÓN .....	VI
DECLARACIÓN EXPRESA .....	VII
RESUMEN.....	VIII
ÍNDICE GENERAL .....	IX
ABREVIATURAS .....	XII
SIMBOLOGÍA .....	XIV
ÍNDICE DE FIGURAS.....	XVI
ÍNDICE DE TABLAS.....	XX
INTRODUCCIÓN.....	XXI
CAPÍTULO 1.....	1
1. ANTECEDENTES .....	1
1.1 Justificación.....	1
1.2 Objetivo general .....	2
1.3 Objetivos específicos .....	3

1.4 Metodología .....	3
1.5 Alcance .....	5
1.6 Resultados esperados .....	6
CAPÍTULO 2.....	7
2. MARCO TEÓRICO .....	7
2.1 Clasificación de los modelos matemáticos de los sistemas .....	8
2.2 Válvula electro-neumática reguladora de presión .....	10
2.3 Simuladores Propuestos .....	14
CAPÍTULO 3.....	17
3. MODELAMIENTO DE MATEMÁTICO DE LA VÁLVULA .....	17
3.1 Parámetros del sistema .....	18
3.2 Subsistema electromagnético.....	24
3.3 Sistema mecánico .....	29
3.4 Sistema neumático .....	31
CAPÍTULO 4.....	36
4. IMPLEMENTACIÓN DEL MODELO ELECTRO-NEUMÁTICO .....	36
4.1 Simulación en lenguaje gráfico del sistema electro-neumático .....	38
4.2 Simulación en lenguaje funcional del sistema electro-neumático .....	52
CAPÍTULO 5.....	58

5. Resultados .....	58
5.1 Subsistema Electromagnético .....	58
5.2 Respuesta del modelo matemático a un escalón de voltaje .....	61
5.3 Tiempo de ejecución y error en estado estable .....	72
CONCLUSIONES .....	76
RECOMENDACIONES.....	78
ANEXOS	
BIBLIOGRAFÍA	

## ABREVIATURAS

A	Amperios. Unidad de corriente en el sistema internacional.
bar	Bares. Unidad de presión.
F	Fuerza.
ISO	International organization for standardization.
l	Litros. Unidad de volumen en el sistema internacional.
mA	Miliamperios. Unidad de corriente. Décima parte de un amperio.
Matlab	Matrix Laboratory.
Máy.	Máximo.
min.	Minuto. Unidad de tiempo. Sexagésima parte de una hora.
Mín.	Mínimo.
mm	Milímetro. Unidad de longitud. Milésima parte de un metro.
N	Newton. Unidad de fuerza en el sistema internacional.
ODE	Ordinary differential equation.
OMEdit	Open modélica connection editor.
Pa	Pascales. Unidad de presión en el sistema internacional.
Prom.	Promedio.

- s Segundos. Unidad de tiempo en el sistema internacional.
- V Voltios. Unidad de voltaje en el sistema internacional.

## SIMBOLOGÍA

$b$	Coeficiente de amortiguamiento.
$C$	Conductancia Sónica.
$\gamma$	Constante adiabática de los gases.
$K_r$	Constante del resorte.
$F_{mag}$	Fuerza magnética.
$F_{atm}$	Fuerza producida por la presión atmosférica.
$F_{res}$	Fuerza del resorte.
$F_{fr}$	Fuerza de fricción.
$i$	Corriente del solenoide.
$R$	Constante universal de los gases.
$\rho_o$	Densidad del aire en condiciones referenciales.
$x_d$	Desplazamiento del cabezal.
$q_{m1}$	Flujo másico a la entrada de la cámara de baja presión.
$L$	Inductancia.
$u_o$	Permeabilidad magnética del aire.

$V_o$	Voltaje DC aplicado al solenoide.
$V_R$	Voltaje en los extremos de la resistencia.
$V_L$	Voltaje en los extremos de la inductancia.
$\mathfrak{R}$	Reluctancia magnética.
$b(x)$	Razón de la presión crítica.

## ÍNDICE DE FIGURAS

Figura 1.1: Metodología .....	5
Figura 2.1: Válvula electro-neumática reguladora de presión [1] .....	11
Figura 2.2: Curva característica de una válvula reguladora de presión .....	14
Figura 3.1: Componentes de un solenoide [8].....	20
Figura 3.2: Curvas características del solenoide proporcional .....	21
Figura 3.3: Subsistema electro–magnético .....	25
Figura 3.4: Curvas típicas de Fuerza vs Desplazamiento dependiendo de la forma de los entrehierros .....	27
Figura 3.5: Circuito eléctrico equivalente del solenoide .....	28
Figura 3.6: Diagrama de cuerpo libre en el diafragma .....	29
Figura 4.1: Programación del sistema electro-neumático a) Lenguaje gráfico entorno Simulink b) Lenguaje gráfico entorno Xcos.....	39
Figura 4.2: Programación del subsistema electro-magnético.....	41
Figura 4.3: Programación del subsistema mecánico a) Lenguaje gráfico entorno Simulink b) Lenguaje gráfico entorno Xcos.....	42
Figura 4.4: Programación del subsistema mecánico a) Condición inicial y límites de desplazamiento en Simulink b) Condición inicial y límites de desplazamiento en Xcos.....	43



Figura 4.5: Programación del subsistema neumático a) Lenguaje gráfico entorno Simulink b) Lenguaje gráfico entorno Xcos.....	44
Figura 4.6: Programación del subsistema neumático a) Evaluación de los polinomios de $b(x)$ y $C(x)$ en Simulink b) Evaluación de los polinomios de $b(x)$ y $C(x)$ en Xcos. ....	45
Figura 4.7: Programación del subsistema neumático a) $q_{m1}$ cuando $P_{reg}/P_{sup} \leq b$ en Simulink b) $q_{m1}$ cuando $P_{reg}/P_{sup} \leq b$ en Xcos. ....	46
Figura 4.8: Programación del subsistema neumático a) $q_{m1}$ cuando $0.999 > P_{reg}/P_{sup} > b$ en Simulink b) $q_{m1}$ cuando $0.999 > P_{reg}/P_{sup} > b$ en Xcos. ....	47
Figura 4.9: Programación del subsistema neumático a) $q_{m1}$ cuando $P_{reg}/P_{sup} \geq 0.999$ en Simulink b) $q_{m1}$ cuando $P_{reg}/P_{sup} \geq 0.999$ en Xcos. ....	48
Figura 4.10: Programación del subsistema neumático a) Conexión de los conmutadores de acuerdo a las condiciones de la Ecuación 3.13 en Simulink b) Conexión de los conmutadores de acuerdo a las condiciones de la Ecuación 3.13 en Xcos.....	49
Figura 4.11: <i>Solver options</i> en Simulink.....	50
Figura 4.12: <i>Simulation Setup</i> en Xcos.....	51
Figura 4.13: Declaración de constantes en OMEdit .....	53
Figura 4.14: Declaración de variables en OMEdit .....	54
Figura 4.15: Subsistema electro-magnético en OMEdit .....	54
Figura 4.16: Subsistema mecánico en OMEdit .....	55

Figura 4.17: Subsistema neumático en OMEdit .....	56
Figura 4.18: Ventana de configuración de la simulación en OMEdit .....	57
Figura 5.1: Fmag [N] vs Voltaje [V] .....	59
Figura 5.2: Fmag [N] vs xd[mm].....	60
Figura 5.3: Modelo electroneumático: a) Implementado en Simulink b) Implementado en Xcos .....	61
Figura 5.4: Presión de salida en función del tiempo en Simulink .....	62
Figura 5.5: Presión de salida en función del tiempo en Xcos .....	63
Figura 5.6: Presión de salida en función del tiempo en OpenModelica .....	63
Figura 5.7: Fuerza magnética en el solenoide en función del tiempo en Simulink .....	65
Figura 5.8: Fuerza magnética en el solenoide en función del tiempo en Xcos .....	65
Figura 5.9: Fuerza magnética en el solenoide en función del tiempo en OpenModelica .....	66
Figura 5.10: Desplazamiento del cabezal en función del tiempo en Simulink	66
Figura 5.11: Desplazamiento del cabezal en función del tiempo en Xcos .....	67
Figura 5.12: Desplazamiento del cabezal en función del tiempo en OpenModelica .....	67
Figura 5.13: Corriente del solenoide en función del tiempo en Simulink .....	68
Figura 5.14: Corriente del solenoide en función del tiempo en Xcos .....	69

Figura 5.15: Corriente del solenoide en función del tiempo en OpenModelica .....	69
Figura 5.16: Escalón de Voltaje en Simulink .....	70
Figura 5.17: Escalón de Voltaje en Xcos .....	70
Figura 5.18: Escalón de Voltaje en OpenModelica .....	71
Figura 5.19: Resumen de rendimiento para el modelo en Matlab/Simulink ...	73
Figura 5.20: Resumen de rendimiento para el modelo en Scilab/Scos .....	73
Figura 5.21: Resumen de rendimiento para el modelo en OpenModelica .....	74

## ÍNDICE DE TABLAS

Tabla 1: Resultados de la simulación en estado estable.....	75
Tabla 2: Estadística de los resultados simulados en estado estable .....	75

## INTRODUCCIÓN

Desde el inicio de los tiempos el ser humano ha buscado describir y comprender los fenómenos físicos que lo rodean, realizando cada vez un análisis más profundo y detallado. Donde los modelos matemáticos explican de una manera completa el comportamiento de éstos, ante situaciones difíciles de observar en la realidad.

El desarrollo de software ha facilitado la aplicación de métodos matemáticos para la solución de las ecuaciones complejas que suelen describir la dinámica de los sistemas. Lo cual nos lleva al plantearnos lo siguiente. ¿Qué plataforma es mejor para encontrar las soluciones de las ecuaciones que describen los sistemas?, en este documento no orientaremos al lector a escoger una u otra aplicación como la mejor, solo expondremos los resultados obtenidos en tres simuladores. Uno de software propietario (Matlab/Simulink) y otros de software libre (Scilab/Xcos y OpenModelica), marcando las características de cada uno y dejando al lector que determine sus propias conclusiones de acuerdo a su necesidad.

# CAPÍTULO 1

## 1. ANTECEDENTES

La implementación de muchos equipos, que se encuentran en los diferentes campos de aplicación, está basada en el desarrollo de prototipos. La mejora continua de estos prototipos se realiza de acuerdo a los estándares que deben cumplir previo a su aprobación. En este proceso de creación se han introducido los modelos matemáticos y los softwares de simulación que aportan un conocimiento profundo del prototipo permitiendo un ahorro de tiempo y dinero.

### 1.1 Justificación

El desarrollo de un prototipo depende de la capacidad del ingeniero para elaborar un modelo matemático, el cual prediga la respuesta del sistema. El modelamiento se aplica en todas las áreas de estudio, con la proyección de controlar la respuesta del sistema a través de sus

variables. El avance de las industrias ha demandado la combinación de varias áreas en la implementación de nuevos procesos. La electro-neumática nace de una de estas combinaciones, convirtiéndose en práctica habitual dentro de la industria. Por esto el desarrollo de modelos matemáticos para los sistemas electro-neumáticos, nos permite desarrollar aplicaciones industriales confiables y eficientes.

En busca de facilitar la descripción matemática, tenemos a nuestro alcance herramientas informáticas con interfaces sencillas. En donde se aplican métodos numéricos iterativos, para simular las respuestas de los modelos bajo diferentes escenarios. En ingeniería implementamos los modelos cuya respuesta, producto de una simulación, se ajusta a los requerimientos de la aplicación. El presente trabajo busca evaluar y analizar el desempeño de los programas, basándose en la rapidez y exactitud de la respuesta simulada.

## **1.2 Objetivo general**

Modelar y simular un sistema electro-neumático, evaluando el desempeño de los programas de simulación tales como: Matlab/Simulink, Scilab/Xcos y (Open) Modelica.

### 1.3 Objetivos específicos

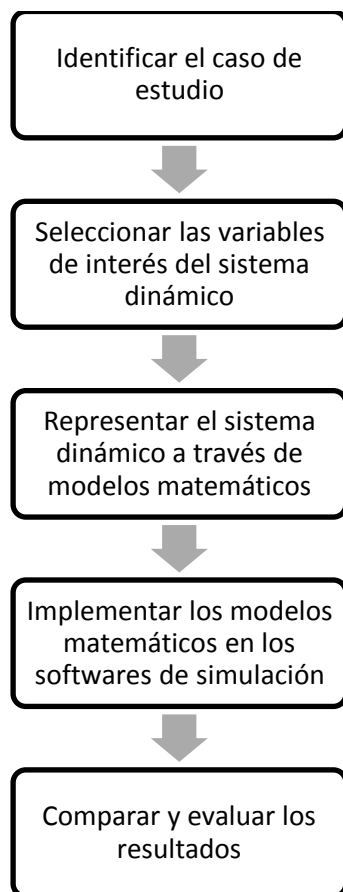
- Seleccionar un sistema electro-neumático de estudio.
- Identificar los principios físicos y las variables que describen al sistema.
- Establecer el modelo matemático del sistema escogido.
- Implementar los modelos matemáticos seleccionados en los softwares de simulación.
- Simular el modelo en los softwares propietarios y libres, comparando resultados.
- Evaluar la eficiencia de los entornos informáticos.

### 1.4 Metodología

El modelamiento y simulación de un sistema electro-neumático comienza con la investigación de las aplicaciones de la electro-neumática dentro de la industria. Existen procesos donde es necesario que la presión que se aplica se mantenga y sea regulada a valores específicos. Para cumplir con este tipo de requerimientos existen las válvulas reguladoras de presión operadas por señales de voltaje o corriente que regulan de manera proporcional la presión de salida.



La metodología que seguimos en el presente trabajo se muestra en la Figura 1.1. Como primer paso planteamos que la simulación será de una válvula electro-neumática reguladora de presión [1]. Ya seleccionado el sistema a simular en el paso dos de la Figura 1.1 diferenciamos las variables de interés como la presión a la salida de la válvula, la fuerza del solenoide y el movimiento del vástago interno para la regulación de la presión con las no relacionadas en nuestro caso de estudio. Tomaremos estas variables de interés para generar un conjunto de ecuaciones que describe matemáticamente el sistema a simular. Posteriormente, en el paso tres de la Figura 1.1, el cual se explica a fondo en el CAPÍTULO 3, realizaremos la descripción matemática de las características dinámicas del sistema. A esta descripción matemática se le llama modelo matemático donde la mayoría se representan en términos de ecuaciones diferenciales [2]. Los siguientes dos pasos de la Figura 1.1 se relacionan estrechamente ya que implementamos el modelo establecido en el paso anterior en los tres softwares planteados: Matlab/Simulink, Scilab/Xcos, OpenModelica. Posteriormente, documentamos los resultados para compararlos y evaluarlos concluyendo con el desarrollo de este proyecto.



**Figura 1.1:** Metodología

### 1.5 Alcance

Este documento pretende llevar a cabo un análisis comparativo de los resultados de la simulación de los diferentes entornos de simulación evaluados sobre un sistema electro–neumático. La modelación matemática del sistema será sometida a diferentes pruebas de simulación en los siguientes softwares: Matlab/Simulink, Scilab/Xcos y OpenModelica. Con estas pruebas observaremos las respuestas de las variables de interés. Una de las limitaciones es que el sistema

seleccionado no se encuentra a disposición para este estudio. Lo que impide verificar experimentalmente los resultados del modelo. Esto imposibilita el uso de todos los parámetros que emplean los fabricantes en la construcción de válvulas.

## **1.6 Resultados esperados**

Manteniendo la misma estructura de los modelos matemáticos esperamos desarrollar un análisis comparativo del desempeño del software propietario Matlab/Simulink y los programas libres Scilab/Xcos y OpenModelica evaluados sobre el sistema electro-neumático seleccionado. Además de comprobar si el modelo matemático de la válvula establecido se ajusta a las diferentes características de los modelos presentados en otros trabajos, citados en este documento, donde se utilizan válvulas reales.

Con los objetivos y la metodología planteada en este capítulo tenemos un panorama más claro para la obtención del modelo de la válvula electro-neumática reguladora de presión. Para complementar la comprensión del lector es importante definir las bases teóricas; de la elaboración del modelo, del sistema y los simuladores propuestos; las cuales abordaremos en el siguiente capítulo.

# CAPÍTULO 2

## 2. MARCO TEÓRICO

Un sistema es un conjunto de componentes que de manera simultánea actúan realizando un objetivo específico. Su interacción con el ambiente se realiza a través de dos grupos de variables: las que no dependen directamente de lo que sucede dentro del sistema y originan una salida son llamadas variables de entrada, y las variables que son generadas por el sistema referidas como de salida [3]. Todo sistema donde su entrada se mantenga invariante en un período de tiempo y presente un cambio significativo en sus variables de salida durante el mismo período, es considerado como un sistema dinámico [3] [2]. Los sistemas dinámicos se representan por un grupo mínimo de variables, las variables de estado, que describen en su totalidad la condición del sistema para cualquier instante de tiempo. Este conjunto de variables deben ser independientes entre sí [3].

Toda simulación es a su vez una representación de un modelo matemático el cual es un conjunto de ecuaciones que describen completamente las relaciones entre las variables de un sistema. En los sistemas físicos las ecuaciones diferenciales se utilizan para construir la mayoría de los modelos matemáticos [2]. Para el diseño de modelos matemáticos debemos tener en cuenta que: sean lo más simples posible, conserven la exactitud de sus resultados y solucione un problema en específico. Cada sistema es descrito de diferente forma en función de las características que se quieren analizar [3]. Dando como resultado un modelo diferente, sin embargo se los agrupa de acuerdo a características comunes.

## **2.1 Clasificación de los modelos matemáticos de los sistemas**

Existen muchos criterios de clasificación para los modelos matemáticos, sin embargo nombraremos los cuatro más comunes [3] los cuales se ven relacionados con un tipo de ecuación diferencial que describe al sistema. El primer criterio es la aplicación del principio de superposición que clasifica a los modelos matemáticos en lineales y no lineales. En los primeros el principio de superposición es aplicable y sus ecuaciones diferenciales son lineales, mientras que en los

segundos el principio no se puede aplicar y sus ecuaciones son no – lineales.

Los modelos también se pueden clasificar por el criterio de la dependencia de sus variables en el tiempo y espacio. El modelo se denomina distribuido cuando es dependiente tanto de las coordenadas espaciales como temporales y es descrito a través de ecuaciones diferenciales parciales. Cuando es independiente de las coordenadas espaciales pero dependiente del tiempo el modelo es de tipo agrupado y está descrito por ecuaciones diferenciales ordinarias. Los criterios de clasificación de los modelos también incluyen el estado de los parámetros con respecto al tiempo. Los modelos cuyas ecuaciones diferenciales contengan coeficientes (parámetros) constantes se los determinan como estacionarios. Mientras que los modelos cuyos parámetros varían en el tiempo son modelos llamados variantes en el tiempo [3].

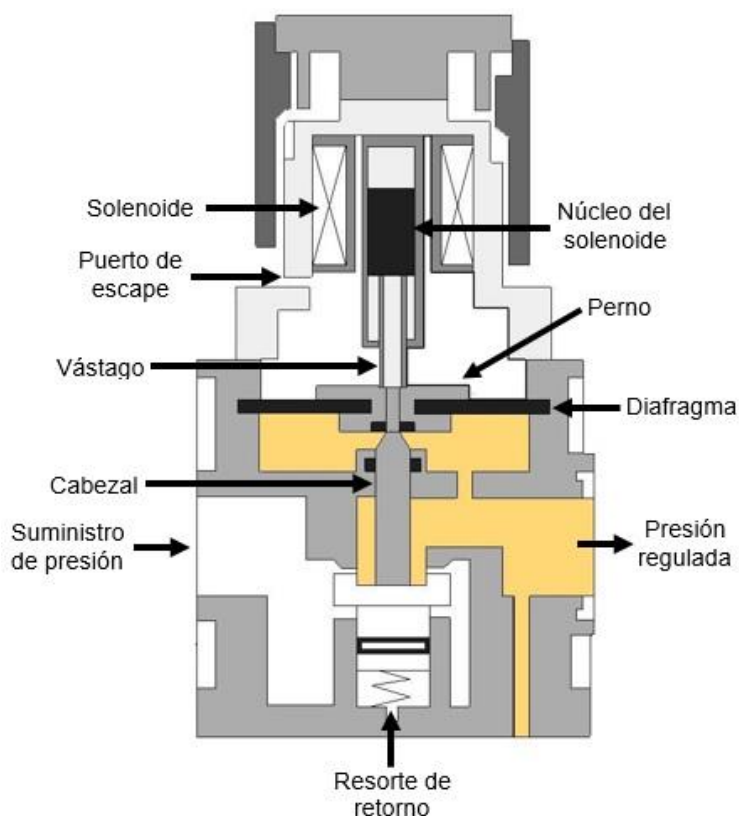
Como último criterio, tenemos la continuidad de las variables independientes. Si las variables dependientes se definen sobre un rango continuo de las variables independientes el modelo es continuo y sus ecuaciones son diferenciales. En cambio, si las variables

dependientes se definen solo para valores específicos de las variables independientes el modelo se denomina discreto y sus ecuaciones son diferenciales en ciertos periodos de tiempo. La clasificación expuesta en párrafos anteriores nos permitirá reconocer de manera más sencilla las características del modelo de la válvula electro-neumática reguladora de presión. A continuación, describiremos los elementos constitutivos y el funcionamiento de la válvula los cuales nos concederán un conocimiento profundo del sistema previo al análisis para obtener su modelo matemático [3].

## **2.2 Válvula electro-neumática reguladora de presión**

Los sistemas neumáticos son aquellos que transmiten señales y potencia utilizando como medio un gas, por lo general aire [2]. Su estudio se basa en la compresibilidad del aire y los efectos que produce en las diferentes variables termodinámicas como presión, volumen, temperatura, etc. La introducción de la electricidad a estos sistemas, ha permitido adoptar el término electro – neumática, dentro del mundo industrial. Con el objetivo de lograr un control eficiente de los fluidos a través de señales eléctricas.

Entre los componentes principales que conforman un sistema neumático tenemos los compresores que producen el aire comprimido, actuadores neumáticos o electro-neumáticos convirtiendo la energía neumática en mecánica y las válvulas neumáticas o electro-neumáticas que controlan la presión y/o el flujo. En este trabajo analizaremos una válvula electro-neumática reguladora de presión la cual es definida como un dispositivo que fija la presión de un sistema en un valor requerido, de manera proporcional a una entrada analógica de control como un voltaje [1]. La Figura 2.1 muestra la válvula electro-neumática reguladora de presión con cada una de sus partes internas.



**Figura 2.1:** Válvula electro-neumática reguladora de presión [1]

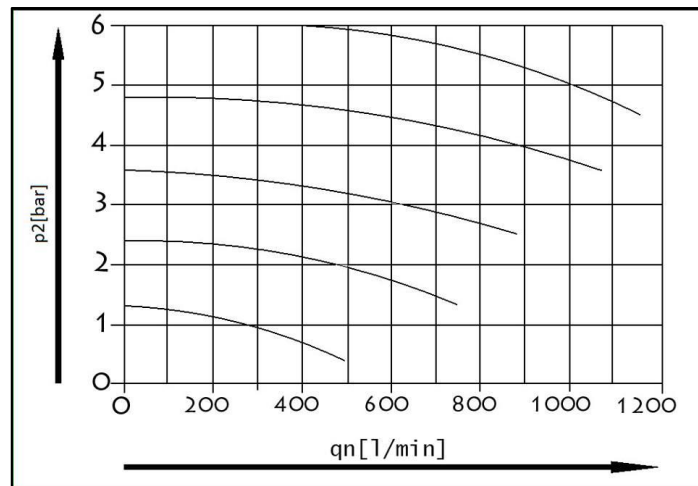


La función principal de esta válvula es permitir la variación de la presión de aire a través del diafragma (Figura 2.1), cuya posición resulta del equilibrio entre la fuerza variable generada por el solenoide proporcional (Figura 2.1) y la fuerza opuesta generada por la presión regulada. La sumatoria de fuerzas provoca un movimiento del diafragma permitiendo el desplazamiento del cabezal (Figura 2.1) el mismo que define el paso del flujo de aire a la cámara de presión regulada (Figura 2.1). La válvula posee un resorte de retorno (Figura 2.1) que restringe el movimiento del diafragma de acuerdo a su límite elástico.

El valor de la presión de salida de la válvula es considerado proporcional a la fuerza electromagnética y también de la corriente del solenoide [1]. Esta fuerza produce el movimiento del vástago (Figura 2.1) aplicando el control deseado de la presión dentro de la sumatoria de fuerzas en el diafragma. El perno (Figura 2.1) conecta el diafragma con el cabezal. Si la presión de salida es muy baja el vástago presiona el diafragma y el cabezal hacia abajo abriendo la restricción (Figura 2.1) lo que permite el paso de más aire aumentando la presión. En cambio, si la presión es muy alta, el diafragma se eleva y el paso de aire a través de la restricción del cabezal se reduce disminuyendo el paso de aire y a su vez la presión en su salida. Cuando la salida

sobrepase el valor ajustado la de-presurización se produce a través del puerto de escape [1].

La curva característica que explica el comportamiento de estos reguladores es similar para todos los modelos en el mercado, sin embargo la forma de onda depende de manera significativa de su construcción interna. En la Figura 2.2 podemos observar tres tipos de respuestas que corresponden a tres reguladores de presión diferentes. Además de su curva característica, los reguladores de presión poseen varios conceptos relacionados que se deben tomar en cuenta al momento de diseñar para lograr el funcionamiento que se esté buscando entre estos conceptos tenemos: la histéresis, sensibilidad, desplazamiento y supresión del punto cero, fallo de linealidad, precisión de repetición y adaptación del margen de presión [4]



**Figura 2.2:** Curva característica de una válvula reguladora de presión

Conocido el sistema y las variables que lo pueden describir es posible obtener diferentes estructuras de modelos matemáticos. Tener en cuenta el entorno donde los modelos se pondrán a prueba, en nuestro caso los simuladores propuestos, se logra una mejor interpretación de los resultados obtenidos en las simulaciones. Por esta razón es importante conocer los entornos informáticos que utilizaremos dando énfasis a sus similitudes y diferencias.

### 2.3 Simuladores Propuestos

En el ámbito de investigación y desarrollo de modelos numéricos existen diferentes simuladores, los cuales pueden dividirse en dos grupos: programas o lenguajes propietarios y programas o lenguajes

libres. Con el fin de comparar los resultados de los simuladores, utilizados en este trabajo, debemos tener conocimiento del origen de los programas sobre los cuales se implementará y probará el modelo.

MathWorks es una empresa que desarrolla programas de cálculo matemático. Estos programas aceleran descubrimientos, innovaciones y desarrollo de ingenieros y científicos en cualquier campo de estudio. Dentro de sus productos se encuentran MATLAB® y Simulink®. MATLAB® es un software de programación donde sus usuarios: desarrollan algoritmos, realizan cálculos numéricos, analizan y visualizan datos. Por otro lado Simulink® usa un lenguaje de programación gráfica para simular y diseñar modelos de sistemas dinámicos con múltiples variables y sistemas embebidos. Simulink permite incorporar algoritmos, desarrollados en MATLAB, en sus modelos y exportar los resultados de sus simulaciones a MATLAB para un análisis más completo [5].

Scilab Enterprises es el proveedor oficial de Scilab y su servicio de soporte técnico. Scilab es un programa libre y de código abierto que en un entorno computacional aplica el cálculo numérico en el desarrollo aplicaciones de ingeniería y científicas. Para una

programación gráfica Xcos, del mismo Scilab Enterprises, es ideal en el diseño de modelos para sistemas dinámicos. Xcos permite el diseño, carga, compilación y simulación de modelos [6].

Modelica® es un lenguaje libre, orientado a objetos y basado en ecuaciones para modelar complejos sistemas físicos. La promoción de este lenguaje se lleva a cabo por medio de la Asociación Modelica, una organización sin fines de lucro, no gubernamental. Esta asociación posee y administra los derechos incorporales relacionados con Modelica disponibles para la promoción del desarrollo industrial y la investigación.

Descritos los simuladores a utilizar podemos marcar diferencias importantes entre ellos. MATLAB/Simulink programas con licencias propietarias, Scilab/Xcos programas libres y de código abierto y Modelica un lenguaje de programación orientada a objetos abierto. Sin embargo todos ellos sirven para la investigación y desarrollo de aplicaciones. Lo que resulta en un punto importante el comparar su ejecución sobre un mismo sistema.

# CAPÍTULO 3

## 3. MODELAMIENTO DE MATEMÁTICO DE LA VÁLVULA

El funcionamiento de la válvula proporcional reguladora de presión se basa en varios principios físicos. Para obtener las ecuaciones que describen el sistema separamos el análisis en subsistemas. La válvula se compone de tres subsistemas: el subsistema electro–magnético, el subsistema mecánico y el subsistema neumático. En el primer subsistema examinamos el circuito electro – magnético del solenoide, señalado en la Figura 2.1, que produce la fuerza magnética. En el subsistema mecánico analizamos la sumatoria de fuerzas que ocurre en el diafragma y la fuerza resultante que desplaza el cabezal al punto donde el valor de presión de salida es la deseada. Finalmente, el subsistema neumático estudia como el flujo de aire se comporta a través de la válvula.

Para realizar el análisis de los subsistemas citados es necesario establecer los parámetros involucrados dentro del sistema y las condiciones de empleo de la válvula. Una vez fijados los parámetros y condiciones el modelo del sistema resulta más preciso. Debido a que no tenemos un dispositivo físico disponible para el estudio utilizamos los parámetros descritos en hojas de datos de fabricantes y otros documentos donde se describen las válvulas proporcionales reguladoras de presión.

### **3.1 Parámetros del sistema**

En el mercado existen algunos tipos de válvulas reguladoras de presión que se diferencian de acuerdo a las condiciones de su aplicación. Basándonos en el modelo presentado en el CAPÍTULO 2 describiremos los elementos constitutivos y las características de la válvula que utilizaremos de aquí en adelante para plantear el modelo y sus simulaciones.

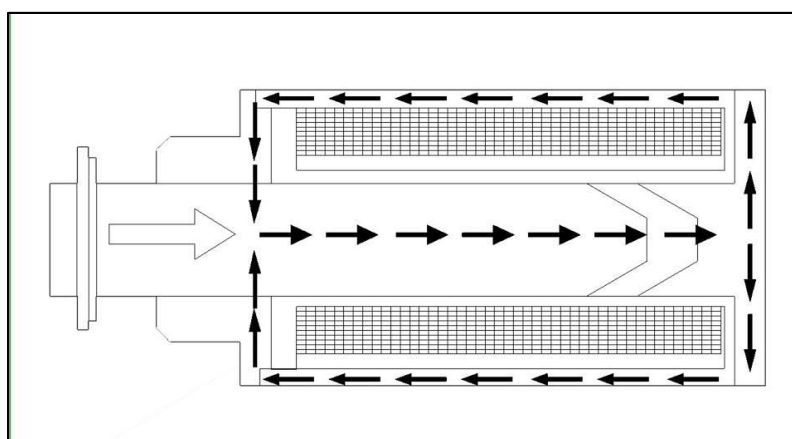
Como fue descrito en la sección 2.2 el solenoide regula la presión de salida aplicando una fuerza en el diafragma que desplaza al cabezal controlando el paso de aire. El solenoide es un dispositivo que convierte energía eléctrica en energía mecánica aprovechando el

efecto de la corriente eléctrica que al fluir a través de un conductor genera un campo magnético [7]. Este campo magnético produce la fuerza magnética que maneja el movimiento del núcleo.

Los solenoides se componen de la siguiente manera: cuerpo magnético, devanados de la bobina, núcleo y componentes operacionales para su funcionamiento. El cuerpo magnético contiene los devanados de la bobina y forma parte del camino del flujo magnético. El patrón de las líneas de flujo es toroidal (Figura 3.1). La eficiencia magnética está determinada por la longitud del camino del flujo, el entrehierro y la permeabilidad magnética de los materiales del solenoide [8]. La válvula en estudio tiene dos entrehierros señalados en la Figura 3.3: el entrehierro 1 entre el núcleo y la carcasa y el entrehierro 2 entre el núcleo y el tope fijo. El diseño de la válvula requiere que exista una eficiencia magnética máxima en el solenoide para esto el cuerpo magnético debe cumplir con dos requisitos. El primer requisito es que debe ser de acero, ya que el acero posee una permeabilidad magnética muy alta ( $\mu_{\text{acero}} \gg \mu_{\text{aire}}$ ), de tal forma que la reluctancia solo sea dada por los entrehierros. La permeabilidad magnética alta en el acero también evita la acumulación de flujo remanente y la histéresis magnética. El segundo requisito es que la geometría de la carcasa para nuestro diseño sea la de un solenoide



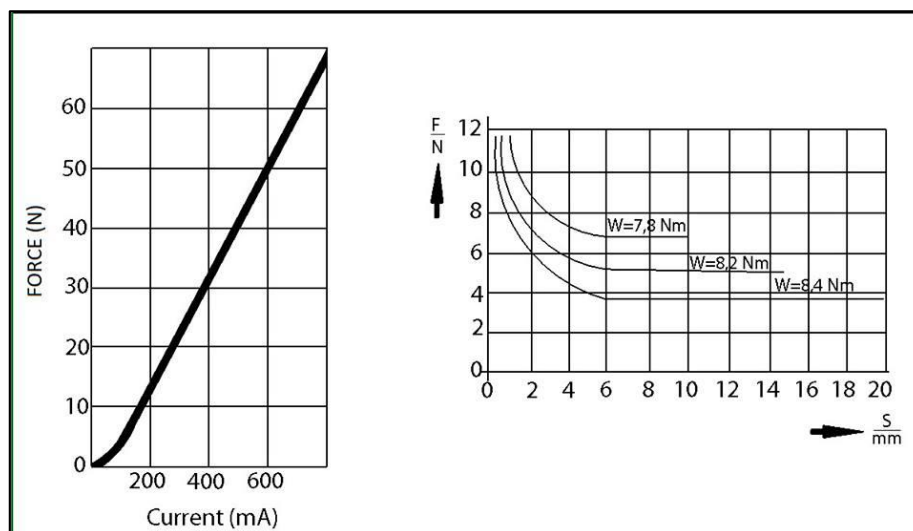
tubular permitiendo que las líneas de flujo que circulan alrededor del solenoide se concentren en la carcasa metálica y no se disipen en entrehierros externos como sucede en otros solenoides.



**Figura 3.1:** Componentes de un solenoide [8]

El núcleo se desplaza en el interior de la bobina el mismo que está hecho de plástico o de un material no magnético para que no intervenga en el recorrido del flujo (Figura 3.1). Los componentes operacionales que no intervienen en la generación de la fuerza magnética contribuyen en el funcionamiento del solenoide. Algunos de estos componentes son la tuerca de montaje, vástagos para push - pull y resortes para ubicar el núcleo en posición inicial y limitar su desplazamiento. La fuerza del solenoide depende en gran medida de la posición del núcleo ya que el entrehierro produce resistencia al flujo magnético [7]. Entonces es importante analizar las curvas características que describen el comportamiento del solenoide a

utilizar. La fuerza magnética con respecto al voltaje/corriente aplicado y al desplazamiento de la armadura.



**Figura 3.2:** Curvas características del solenoide proporcional

En la Figura 3.2 se pueden observar dos particularidades: en la curva  $F_{mag}$  vs *Voltaje/Corriente* la fuerza es directamente proporcional al voltaje/corriente aplicada y en la curva  $F_{mag}$  vs *Desplazamiento* muestra un intervalo donde la fuerza es constante a lo largo del desplazamiento. Esta curva puede ser modificada por el resultado de la geometría del tope fijo y del núcleo [8]. Las curvas mostradas clasifican el solenoide en un solenoide proporcional de un solo accionamiento. Las características mostradas se acogen a la información encontrada en las hojas de datos publicadas por los fabricantes de válvulas reguladoras de presión.

Adicionalmente las válvulas están formadas por más elementos que aseguran el correcto trabajo de la misma entre ellos tenemos el resorte. El resorte dentro de la válvula facilita que en el interior de la bobina la distancia entre el núcleo y el tope fijo no sea igual a cero inicialmente. Además compensa el desplazamiento debido a la presión atmosférica en el diafragma. Para esto es necesario precargar el resorte sobredimensionando su diseño, haciéndolo más largo, lo cual crea una deformación inicial. El solenoide de la válvula se acciona únicamente hacia abajo por lo que necesita un retorno a su posición inicial. El resorte, que actúa de forma lineal, cumple este objetivo comprimiéndose cuando se le aplica voltaje al solenoide y retornando a su extensión inicial cuando este voltaje es cero.

La válvula utiliza además un diafragma en el cuál se concentran todas las fuerzas que intervienen en el sistema. El diafragma es una membrana de caucho nitrílico reforzado por un disco central de aluminio rígido. El resorte es acoplado para que el movimiento máximo no estire el diafragma y pase su límite elástico [7]. Asimismo utiliza un elemento de seguridad incorporado cuando la presión sobrepasa el límite especificado. A través de un orificio en el diafragma la cantidad de aire en exceso es enviada al exterior por el puerto de escape

mostrado en la Figura 2.1. Si el regulador no ha sido seleccionado correctamente para las condiciones de operación este sistema de seguridad puede causar oscilaciones en la presión de salida.

El aire es la fuente de alimentación principal de la válvula reguladora de presión. Para esto plantearemos ciertos criterios útiles al momento del desarrollo y de la simulación del modelo matemático de la válvula. El movimiento del aire produce cambios que afectan a la presión, temperatura y volumen de manera conjunta. Para aproximar la relación de las tres variables utilizaremos la ley de los gases ideales tomando el aire como un gas ideal [9]. El aire que existe en la atmósfera contiene vapor por lo cual debe ser tratado anteriormente con secadores y filtros evitando su condensación. Además se realiza la suposición de que el aire comprimido se puede tratar como continuo y de carácter compresible. El movimiento de aire a través de la válvula puede causar pérdidas de fricción por viscosidad o por gravedad sin embargo la única fuerza de fricción que tomaremos en cuenta será la producida por el movimiento del núcleo.

Ahora examinaremos el comportamiento termodinámico del aire dentro de la válvula al momento de la apertura. La compresión del aire

llena las cámaras dentro de la válvula de una manera muy rápida lo que aproxima este proceso termodinámico a un proceso adiabático. En un proceso adiabático no existe transferencia de calor entre un sistema y su entorno lo cual puede suceder de dos formas: si el sistema está bien aislado, o si el proceso se efectúa con tal rapidez que no hay tiempo para que ocurra un flujo de calor apreciable [9]. La válvula reguladora de presión se encuentra dentro del segundo caso.

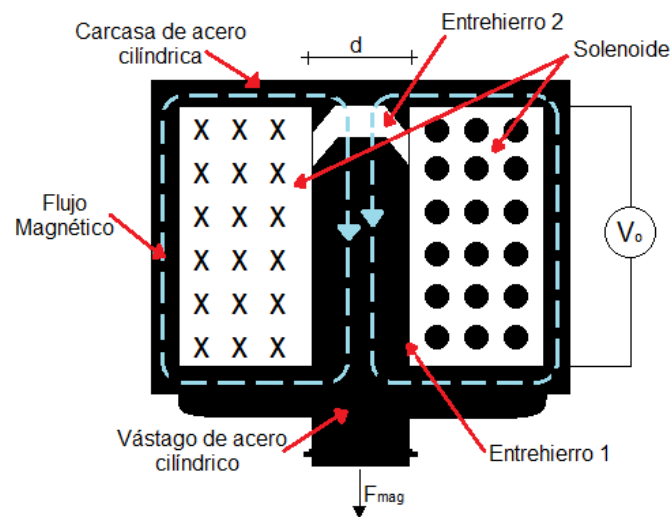
Los parámetros de diseño explicados anteriormente ofrecen una guía y un área de trabajo para el desarrollo del modelo matemático y de la posterior simulación de la válvula. A continuación se analizarán las ecuaciones de cada subsistema describiendo a la válvula de forma que exista una relación con las características planteadas para cada elemento de la misma.

### **3.2 Subsistema electromagnético**

En el subsistema electro-magnético obtenemos la respuesta en el tiempo de la fuerza magnética. Este subsistema se conforma de un circuito eléctrico y un circuito magnético. La carcasa, el vástago y los entrehierros 1 y 2, señalados en la Figura 3.3, componen el circuito magnético donde circula el flujo inducido por la corriente del circuito

eléctrico [10]. Como se puede apreciar en Figura 3.3 la longitud del entrehierro 1 es muy pequeña, del orden de las micras, por lo que no se considera para el cálculo de la reluctancia del camino magnético. La reluctancia total del circuito magnético es entonces la reluctancia del entrehierro 2. El entrehierro 2, señalado en la Figura 3.3, tiene un área cónica constante transversal al flujo; sin embargo su longitud aumenta o disminuye con el movimiento del vástago. Esta última condición hace que la reluctancia del sistema sea variable como se muestra en la siguiente ecuación:

$$\mathfrak{R} = \frac{d}{u_o A} = \frac{x_d}{u_o A_t} = \frac{x_d}{u_o (A_l + A_b)} \quad (3.1)$$



**Figura 3.3:** Subsistema electro-magnético

La magnitud de la fuerza magnética puede ser calculada por la ecuación propuesta [11]:

$$F_{mag} = \frac{i^2}{2} \frac{dL(x)}{dx} \quad (3.2)$$

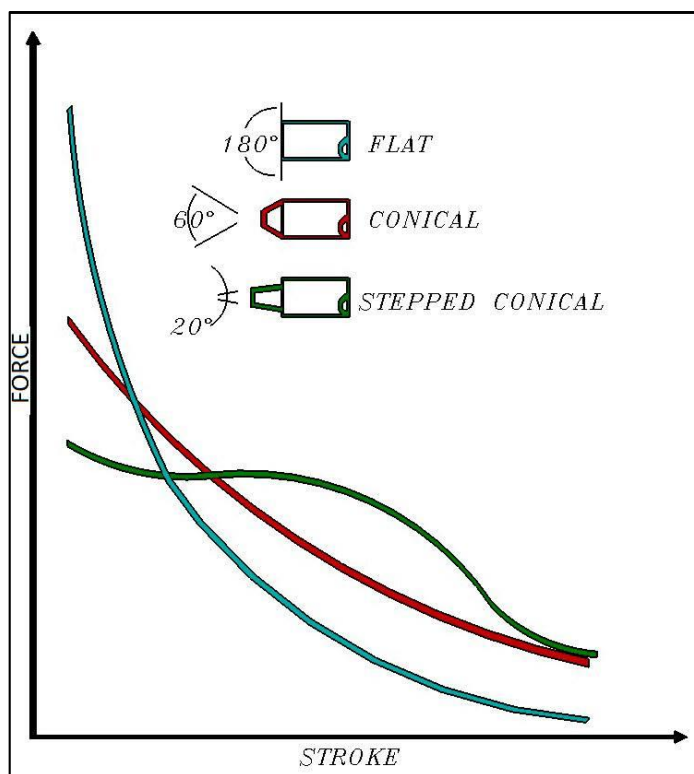
Esta fuerza depende de la intensidad de corriente  $i$ , que circula a través del solenoide, y de la derivada de la inductancia del sistema electromagnético  $dL(x)/dx$ . La inductancia se puede derivar conociendo la reluctancia total del sistema por la relación dada en la siguiente ecuación:

$$L(x) = \frac{N^2}{\mathfrak{R}} \quad (3.3)$$

Reescribiendo la fuerza magnética como se indica en la ecuación a continuación:

$$F_{mag} = \frac{i^2 N^2 u_o A_t}{2 x_d^2} \quad (3.4)$$

La forma cónica del entrehierro 2 (Figura 3.3) permite que el área transversal al flujo magnético sea mayor que si fuera cilíndrico lo cual incrementa la magnitud de la fuerza magnética. En la industria las formas cilíndricas y cónicas de los entrehierros son las más comunes. En el diseño se selecciona una forma u otra dependiendo de la respuesta de la fuerza magnética respecto al desplazamiento del entrehierro. La variación de la fuerza respecto a la forma del entrehierro se aprecia en la Figura 3.4 para todas las geometrías la fuerza es mayor mientras la longitud del entrehierro disminuye y viceversa [8].



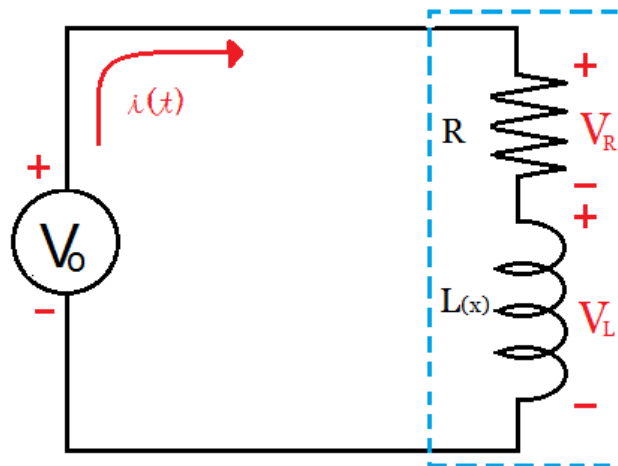
**Figura 3.4:** Curvas típicas de Fuerza vs Desplazamiento  
dependiendo de la forma de los entrehierros

El circuito eléctrico se compone de un solenoide de cobre y una fuente de voltaje como se indica en la Figura 3.3. En el equivalente de este circuito, mostrado en la Figura 3.5, se representa al solenoide con una resistencia y una inductancia en serie. El voltaje aplicado en los extremos del solenoide también forma parte del circuito y por lo tanto también se encuentra en la Figura 3.5 del equivalente eléctrico. Aplicando la segunda Ley de Kirchhoff al circuito descrito obtenemos la siguiente ecuación:

$$\sum V = V_o - V_R - V_L = 0 \quad (3.5)$$



Donde se incluyen el voltaje de alimentación ( $V_o$ ), la caída de tensión en la resistencia ( $V_R$ ) y el voltaje en los extremos de la inductancia ( $V_L$ ).



**Figura 3.5:** Circuito eléctrico equivalente del solenoide

El voltaje en los extremos de la resistencia lo calculamos con la ley de Ohm [10]. Para el inductor la caída de tensión en sus extremos depende de la variación de los enlaces de flujo en el tiempo. Los enlaces de flujo dependen de la corriente que atraviesa en el inductor y la distancia del entrehierro 2 (Figura 3.3) [12]. Con estas relaciones obtenemos la ecuación que depende tanto de la variación de corriente del circuito como de la distancia del entrehierro 2 ( $x_d$ ) como se muestra a continuación:

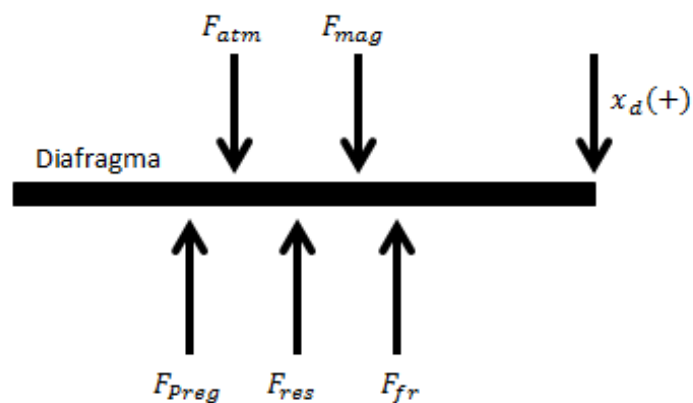
$$V_o - Ri(t) - L(x) \frac{di(t)}{dt} - i(t) \frac{dL(x_d)}{dt} = 0 \quad (3.6)$$

Con el análisis del subsistema electro-magnético obtuvimos las ecuaciones de la fuerza magnética y la relación entre la corriente del

solenoides y el desplazamiento del núcleo. Esta fuerza interviene en el análisis del subsistema mecánico en la sección 3.3. Acercándonos cada vez más a representar todo el sistema matemáticamente.

### 3.3 Sistema mecánico

La sumatoria de fuerzas que ocurre en el diafragma es de crucial interés en el análisis de la respuesta de la presión regulada. La rapidez con que la presión llegue al punto requerido depende en gran parte de la apertura o cierre del orificio por donde circula el aire. El área del orificio depende de la posición del cabezal. A través de la aplicación de las fuerzas en el diafragma señaladas en el diagrama de la Figura 3.6. Estas fuerzas son: la fuerza magnética  $F_{mag}$ , las fuerzas neumáticas (Presión atmosférica  $P_{atm}$  y Presión regulada o de salida  $P_{reg}$ ), la fuerza del resorte ( $F_{res}$ ) y la fuerza de fricción ( $F_{fr}$ ).



**Figura 3.6:** Diagrama de cuerpo libre en el diafragma

Por convención tomaremos como positivo el movimiento hacia abajo del cabezal representado por  $x_d$  (Figura 3.6) produce facilidad al interpretar los resultados. De acuerdo a la segunda Ley de Newton<sup>3</sup> tenemos:

$$\sum F = F_{Patm} + F_{mag} - F_{Preg} - F_{res} - F_{fr} = ma \quad (3.7)$$

La fuerza magnética es definida en la ecuación (3.4), las fuerzas neumáticas se amparan en el principio de presión en un fluido donde su superficie de contacto es el área efectiva del diafragma  $A_d$ , el resorte al ser lineal experimenta una fuerza proporcional al desplazamiento del cabezal y a su constante  $K_r$ , finalmente la fuerza de fricción se define proporcional a la velocidad del cabezal y a su constante de amortiguamiento  $b$  [9]. La ecuación (3.7) aplicando los conceptos señalados queda de la siguiente manera:

$$A_d P_{atm} + F_{mag} - A_d P_{reg} - K_r x_d - b \frac{dx_d}{dt} = m \frac{d^2 x_d}{dt^2} \quad (3.8)$$

La ecuación (3.8) representa la dinámica del cabezal y su respuesta en el tiempo en breves rasgos podemos notar que la fuerza magnética debe ser lo suficientemente grande como para sobreponerse al efecto de la fuerza provocada en la presión regulada y mantener la apertura del paso de aire. También podemos observar que en el momento que el sistema sea estable la fuerza magnética se equilibra con las fuerzas neumáticas y con la del resorte. Como hemos expresado la presión

regulada interviene de manera constante en el movimiento del cabezal además de ser la variable principal a controlar por la válvula. Por lo cual estudiaremos las ecuaciones neumáticas que dan origen a esta presión y los factores que participan dentro de la formulación de las mismas.

### 3.4 Sistema neumático

El flujo de aire que pasa por la válvula y la presión que produce se fundamenta en el principio de conservación de la masa, tomando en cuenta que el aire responde a un proceso adiabático [1] obtenemos la siguiente ecuación:

$$qm_1 - qm_2 = \frac{P_{reg}}{RT_{reg}} \frac{dV}{dt} + \frac{V}{\gamma RT_{reg}} \frac{dP_{reg}}{dt} \quad (3.9)$$

La válvula a la salida presenta una cámara donde consideramos que el volumen de aire será constante lo que modifica la ecuación (3.9) llevándonos a la ecuación mostrada:

$$qm_1 - qm_2 = \frac{V}{\gamma RT_{reg}} \frac{dP_{reg}}{dt} \quad (3.10)$$

Observamos que la tasa de cambio del volumen de control está dada en función de la presión regulada  $P_{reg}$ , el volumen de la cámara de presión regulada  $V$ , la constante adiabática  $\gamma$ , la constante de los gases  $R$  y temperatura de salida  $T_{reg}$ . La ecuación (3.10) introduce dos

términos  $qm_1$  que es el flujo másico a la entrada de esta cámara mediante un orificio y  $qm_2$  es el flujo másico a la salida. El flujo másico  $qm_1$  es el controlado por el movimiento del cabezal y puede ser calculado basándonos en el principio de la conservación de la energía y asumiendo que la alimentación desde el compresor toma el papel de un reservorio de aire donde se toma en cuenta que tanto la presión, la masa y la temperatura del mismo es constante, además que este orificio es uniforme [7].

Este cálculo es válido para orificios bien redondos sin contracción de flujo y velocidad inicial de casi cero pero este concepto tiene que ser ampliado para describir el flujo a través de orificios finos de bordes agudos donde se producen estas contracciones y existen pérdidas de flujo para detallar este efecto numéricamente introducimos el coeficiente de descarga  $c$  cuyo valor puede determinarse experimentalmente. Entonces el flujo másico  $qm_1$  se expresa de la ecuación que se muestra a continuación [2]:

$$\dot{m} = qm_1 = cA_o \frac{P_{sup}}{\sqrt{T_{sup}}} \sqrt{\frac{2\gamma}{(\gamma - 1)R} \left( \left( \frac{P_{reg}}{P_{sup}} \right)^{\frac{2}{\gamma}} - \left( \frac{P_{reg}}{P_{sup}} \right)^{\frac{\gamma+1}{\gamma}} \right)} \quad (3.11)$$

La ecuación (3.11) define nuevos términos como el área del orificio por donde fluye el aire  $A_o$  y  $P_{sup}$  que es la presión de entrada a la válvula.

Para ciertos valores de las variables de interés en la ecuación (3.11) el flujo másico se hace máximo y el valor de la presión donde se produce esta condición se conoce como presión crítica. Si sustituimos el valor de la presión de salida por la presión crítica obtenemos la velocidad crítica, que luego de aplicar las relaciones válidas para el proceso adiabático, resulta ser la velocidad del sonido [2]. Esto divide el flujo másico en dos expresiones definidas por la velocidad del flujo: si el aire experimenta la velocidad del sonido el flujo es sónico, sino el flujo es subsónico donde la velocidad es menor que la del sonido. La norma ISO 6358 trata sobre la determinación de las características del flujo másico en fluidos compresibles, para cualquier dispositivo neumático, obteniendo la siguiente ecuación [7]:

$$\dot{m} = q_{m1} = \begin{cases} P_{sup} C \rho_o \sqrt{\frac{T_o}{T_{sup}}} \sqrt{1 - \left( \frac{\frac{P_{reg}}{P_{sup}} - b}{1 - b} \right)^2} ; \frac{P_{reg}}{P_{sup}} > b \\ P_{sup} C \rho_o \sqrt{\frac{T_o}{T_{sup}}} ; \frac{P_{reg}}{P_{sup}} \leq b \end{cases} \quad (3.12)$$

De la ecuación (3.12) definimos el concepto de conductancia sónica  $C$  que es la relación entre el flujo másico que pasa por el componente neumático y la presión de entrada cuando la velocidad del aire ha alcanzado la velocidad del sonido y el flujo es sónico. La norma ISO incluye además variables que son definidas por valores en condiciones referenciales para la densidad  $\rho_o$  y para la temperatura  $T_o$ . También

debemos definir que la razón de la presión crítica  $b$  es la relación entre la presión de salida y la presión de entrada en que la velocidad del aire alcanza la velocidad del sonido [7]. Estas dos variables son indispensables para realizar el modelo neumático. De acuerdo a Varga y Honkola (2012) [13] la conductancia sónica y la razón de la presión crítica pueden ser determinadas como una función de la posición del cabezal, este tipo de mediciones son respaldadas con la norma ISO 6358 que determina dos formas de medición de las mismas.

Sin embargo el modelo presenta una nueva condición física cuando  $b$  se aproxima a la unidad el flujo se vuelve laminar, el número de Reynolds es bajo y el fluido puede ser considerado como incompresible [13]. Además presenta problemas numéricos en los algoritmos de integración de los simuladores. Entonces para representar el modelo de la mejor forma y no tener problemas con la simulación, se aproxima la ecuación subsónica a la siguiente ecuación:

$$q_{m1} = \begin{cases} K_1 P_{sup} \left(1 - \frac{P_{reg}}{P_{sup}}\right) \sqrt{\frac{T_o}{T_{sup}}} & ; \frac{P_{reg}}{P_{sup}} \geq 0.999 \\ P_{sup} C \rho_o \sqrt{\frac{T_o}{T_{sup}}} \sqrt{1 - \left(\frac{\frac{P_{reg}}{P_{sup}} - b}{1 - b}\right)^2} & ; 0.999 > \frac{P_{reg}}{P_{sup}} > b \\ P_{sup} C \rho_o \sqrt{\frac{T_o}{T_{sup}}} & ; \frac{P_{reg}}{P_{sup}} \leq b \end{cases} \quad (3.13)$$

Donde la ganancia  $K_1$  sirve para aproximar linealmente la ecuación (3.13) donde su valor es especificado en la ecuación a continuación:

$$K_1 = 1000 C \rho_o \sqrt{1 - \left(\frac{0.999 - b}{1 - b}\right)^2} \quad (3.14)$$

Una vez planteadas las ecuaciones que forman el modelo matemático para la válvula reguladora de presión luego del análisis de cada subsistema. Vamos a plantear la programación en bloques para Matlab/Simulink y Scilab/Xcos y la programación en código de OpenModelica. La simulación permitirá comprobar la validez del modelo y permitirá el estudio comparativo entre los programas. A continuación observaremos los entornos de programación y los cambios que se dan al programar entre un programa y otro.



# CAPÍTULO 4

## 4. IMPLEMENTACIÓN DEL MODELO ELECTRO-NEUMÁTICO

En el CAPÍTULO 3 obtuvimos el modelo matemático del sistema. El modelo se construyó con ecuaciones diferenciales, resultado del análisis de los subsistemas, cuya resolución matemática se obtiene con la aplicación de métodos numéricos avanzados y nos muestran la respuesta en el tiempo de las variables del modelo. La comprensión y control del comportamiento del sistema se simplifica con la respuesta en el tiempo de sus variables de interés. Implementando el modelo en los simuladores propuestos se ejecutan algoritmos de resolución y se obtienen las gráficas de dichas variables. La simulación del modelo la realizamos en tres software de simulación. En Simulink y Xcos utilizamos el lenguaje gráfico de programación que compone su interfaz. Utilizamos el lenguaje de código propietario de Matlab (Anexo 1) y el lenguaje de código libre de

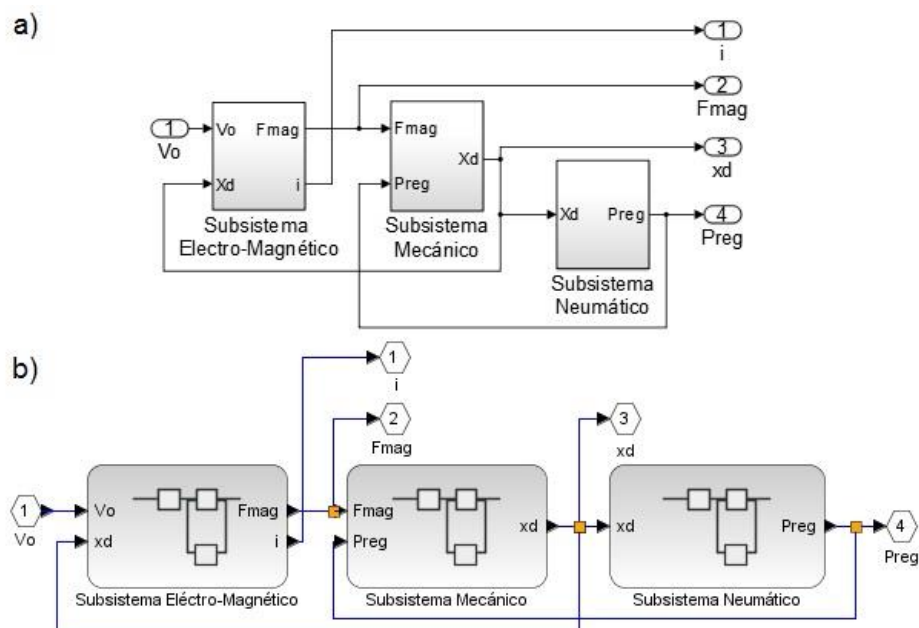
Scilab (Anexo 2) para declarar las constantes del sistema. Debido a que los entornos de simulación, Matlab/Simulink y Scilab/Xcos, tienen interfaces similares detallaremos su implementación en la sección 4.1 resaltando sus diferencias. Para el tercer programa, OpenModelica, la programación se basa en ubicar las ecuaciones que conforman el modelo en lenguaje de código y observar la respuesta en el tiempo de las variables de interés declaradas dentro del código. Las gráficas y los valores que se originan luego de la aplicación de métodos numéricos permiten realizar la evaluación de los simuladores y establecer una comparación entre ellos.

Así como se analizó el sistema en el CAPÍTULO 3 planteamos la simulación en subsistemas. Agrupamos las ecuaciones de los subsistemas en súper bloques reduciendo el número de bloques en la ventana del modelo. Estas agrupaciones nos permitieron también comprender el comportamiento de cada subsistema por separado y su influencia dentro del modelo matemático completo. Los softwares mencionados tiene la particularidad de desarrollar modelos a través de bloques que representa variables físicas, sin embargo estos bloques se encuentran en librerías las cuales no se utilizaron para el desarrollo de estas simulaciones. Por lo tanto se programó en los simuladores los sistemas físicos como expresiones matemáticas. El rango de estas

expresiones se definió en función de las limitaciones físicas del sistema especificadas en el CAPÍTULO 3.

#### **4.1 Simulación en lenguaje gráfico del sistema electro-neumático**

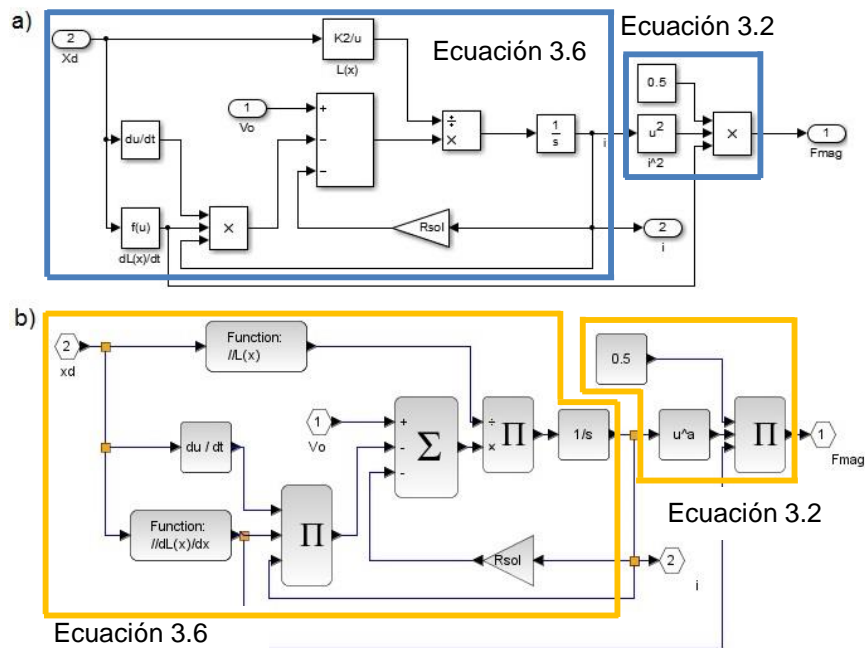
La simulación en lenguaje gráfico se realizó en el software Matlab/Simulink versión 8.1 (Figura 4.1.a) y Scilab/Xcos versión 5.5.2 (Figura 4.1.b). Podemos notar que la construcción del modelo se hizo con tres súper bloques que representan a cada subsistema descrito en el CAPÍTULO 3. De cada bloque salen las variables de interés de cada subsistema. La Figura 4.1 muestra cómo, para ambos casos, entre los subsistemas existen retroalimentaciones naturales que influyen directamente en la respuesta del subsistema electro-magnético y el subsistema mecánico. Si ingresamos en cada súper bloque se puede apreciar en detalle cada ecuación que compone cada subsistema



**Figura 4.1:** Programación del sistema electro-neumático a) Lenguaje gráfico entorno Simulink b) Lenguaje gráfico entorno Xcos.

En el bloque del subsistema electro-magnético (Figura 4.2) implementamos la ecuación (3.6) cuya solución es la respuesta en el tiempo de la corriente del solenoide proporcional  $i$ . Para armar esta ecuación despejamos la derivada de la corriente de la sumatoria de voltajes y la integramos. La ecuación (3.2), formada por la multiplicación de tres bloques, utiliza los valores de esta corriente para describir el comportamiento de la fuerza magnética  $F_{mag}$ . La ecuación (3.2) y (3.6), que se detallan en la Figura 4.2, contiene funciones definidas por el usuario que dependen de la inductancia y de su derivada respectivamente. El término de la inductancia depende del desplazamiento  $x_d$  el cual para evitar problemas de convergencia

dentro de la simulación lo limitamos al menor valor posible diferente de cero así como se expresó en la sección 3.1 y como se señala en la Figura 4.4. La derivada ha sido escrita en función de su valor absoluto ya que para la ecuación (3.2) solo nos interesa su valor numérico. Estas dos ecuaciones rigen el comportamiento del subsistema electromagnético donde claramente la corriente que se produce al aplicar un voltaje DC  $V_0$  en el solenoide proporcional es la principal variable de interés. El desplazamiento del vástago es una variable que afecta al sistema, pero se genera en el subsistema mecánico, jugando un papel importante en la respuesta en el tiempo de la fuerza magnética  $F_{mag}$ .



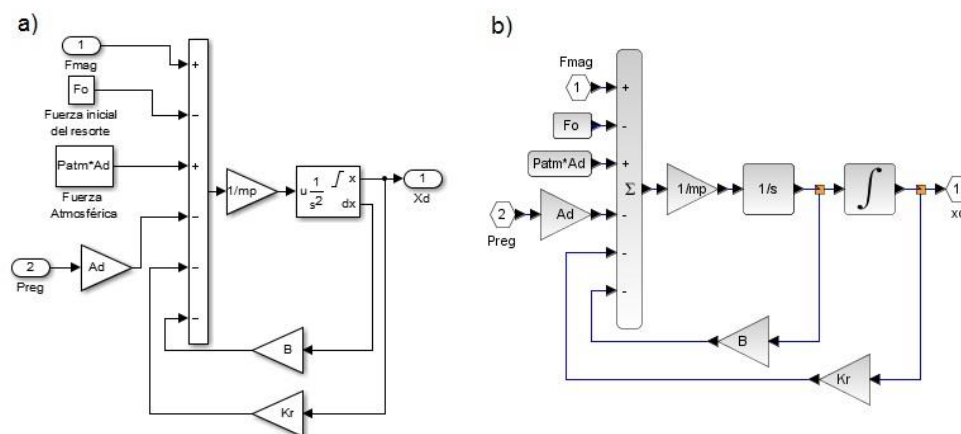
**Figura 4.2:** Programación del subsistema electro-magnético

a) Lenguaje gráfico entorno Simulink b) Lenguaje gráfico entorno

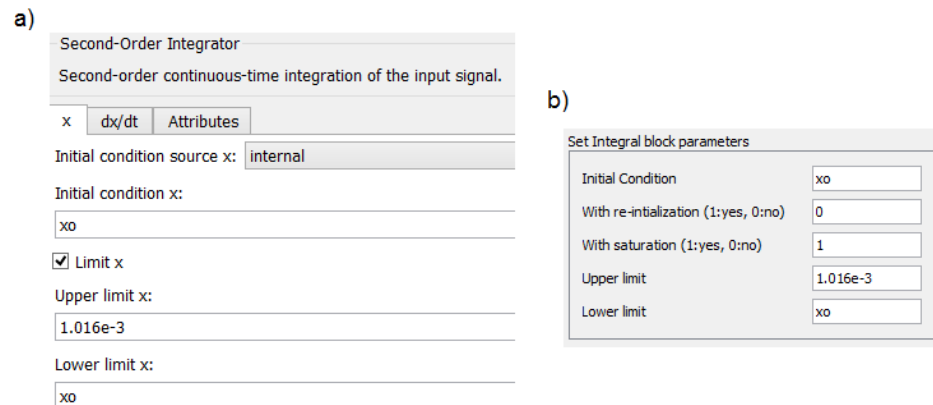
Xcos.

El bloque del subsistema mecánico lo describimos con la ecuación (3.8) cuya programación se muestra en la Figura 4.3. La ecuación (3.8) es una ecuación diferencial de segundo orden para su manejo en el programa despejamos la segunda derivada del desplazamiento y aplicamos un bloque, de doble integral en Matlab/Simulink y dos bloques de integral en Scllab/Xcos, donde cerramos los dos lazos del subsistema que se muestran en la Figura 4.3. Este subsistema genera el valor del desplazamiento  $x_d$  por lo tanto aquí es donde se definen sus límites. El límite inferior lo definimos por el mínimo valor del entrehierro especificado para el solenoide tubular SP1414P1200 [8].

El límite superior lo determinamos teóricamente como la suma del desplazamiento inicial más el valor máximo establecido en la publicación de Varga y Honkola (2012). Dentro de las simulaciones definimos estos valores tal como se muestra en la Figura 4.4 y en los programas del Anexo 1 y el Anexo 2 respectivamente. Dentro del subsistema mecánico también observamos una fuerza inicial del resorte definida en el Anexo 1 y el Anexo 2. Esta fuerza representa la precarga del resorte expresada en la sección 3.1. A su vez se muestran las fuerzas neumáticas que intervienen en el diafragma.



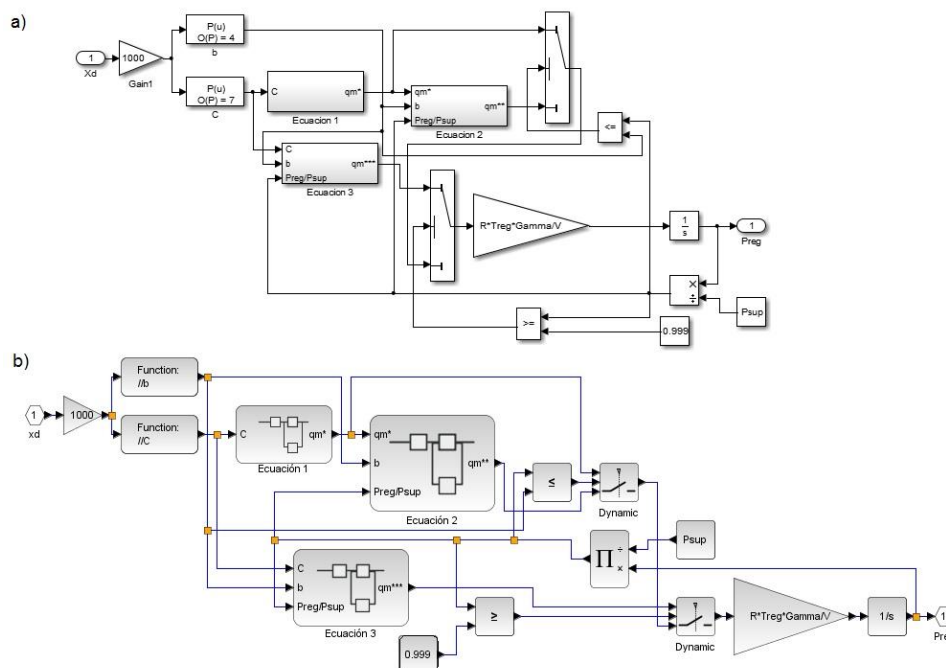
**Figura 4.3:** Programación del subsistema mecánico a) Lenguaje gráfico entorno Simulink b) Lenguaje gráfico entorno Xcos.



**Figura 4.4:** Programación del subsistema mecánico a) Condición inicial y límites de desplazamiento en Simulink b) Condición inicial y límites de desplazamiento en Xcos.

Finalmente consideramos el subsistema neumático descrito por las ecuaciones (3.13) y (3.14). La programación en Matlab/Simulink y Scilab/Xcos divide el subsistema neumático en tres subsistemas internos donde se expresa la ecuación del flujo másico  $q_{m1}$  por segmentos de acuerdo a la relación entre la presión de salida con respecto a la de entrada. Cada subsistema posee un segmento de la ecuación (3.13). La Figura 4.5 muestra las conexiones entre subsistemas internos y con los conmutadores que eligen la salida correspondiente de cada subsistema al integrador para obtener el valor de la presión de salida. La misma que es utilizada como realimentación en el subsistema mecánico.





**Figura 4.5:** Programación del subsistema neumático a) Lenguaje gráfico entorno Simulink b) Lenguaje gráfico entorno Xcos.

La ecuación (3.13) posee dos términos que son únicos para cada válvula. Recordando lo citado en la sección 3.4: La conductancia sónica y la razón de la presión crítica pueden ser determinadas como una función de la posición del cabezal [13]. Los estudios realizados por Varga y Honkola (2012) [14] determinaron mediante pruebas una expresión válida para la conductancia sónica  $C$  y la razón de la presión crítica  $b$ . Estas pruebas son realizadas a una válvula real, dispositivo que no poseemos, por lo tanto utilizaremos los datos publicados [13] para la simulación del sistema neumático. Las pruebas fueron realizadas sobre la válvula Festo VPPM-6L-L-1-G18-0L6H donde se  $b$

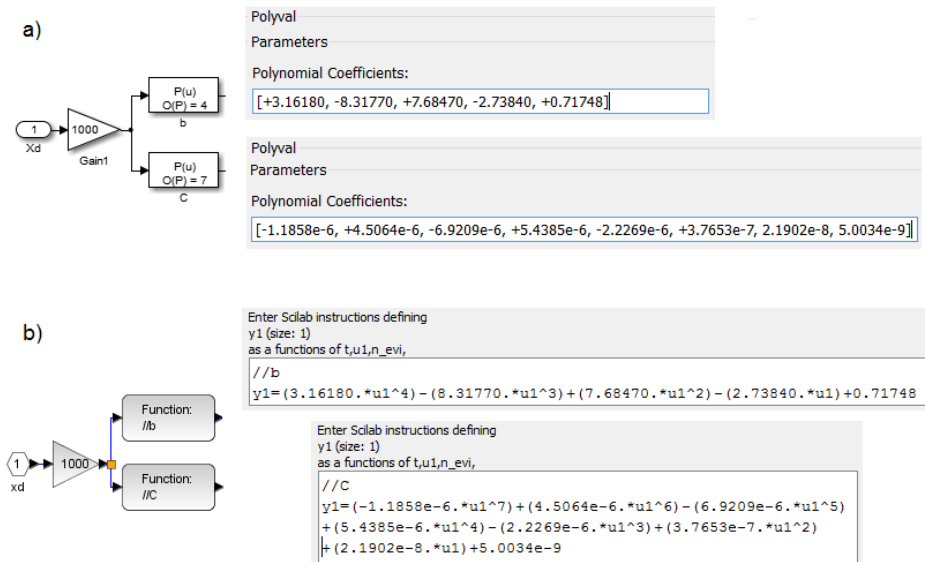
es un polinomio de cuarto orden, como se muestra en la siguiente ecuación:

$$b(x) = 3.1618x^4 - 8.3177x^3 + 7.6847x^2 - 2.7384x + 0.71748 \quad (4.1)$$

Además  $C$  es un polinomio de séptimo orden indicado en la ecuación posterior:

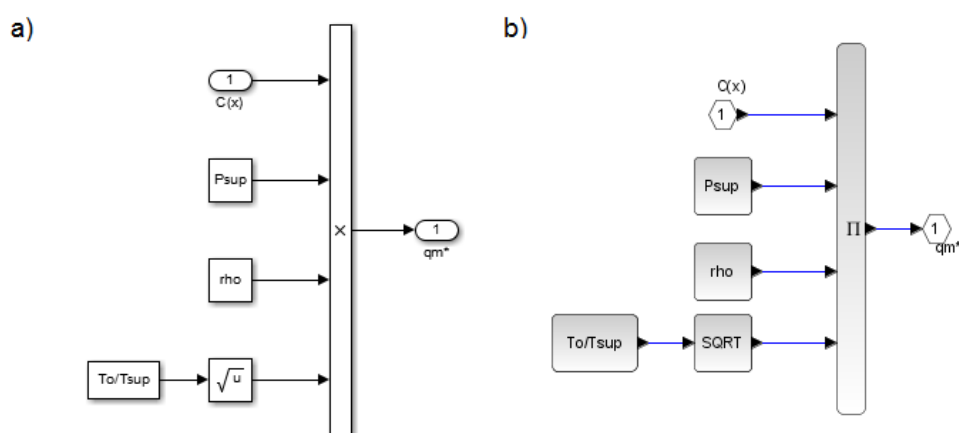
$$\begin{aligned} C(x) = & -(1.1858 * 10^{-6})x^7 + (4.5064 * 10^{-6})x^6 \\ & - (6.9209 * 10^{-6})x^5 + (5.4385 * 10^{-6})x^4 \\ & - (2.2269 * 10^{-6})x^3 + (3.7653 * 10^{-7})x^2 \\ & + (2.1902 * 10^{-8})x + 5.0034 * 10^{-9} \end{aligned} \quad (4.2)$$

La evaluación de los polinomios se observa en la Figura 4.6.



**Figura 4.6:** Programación del subsistema neumático a) Evaluación de los polinomios de  $b(x)$  y  $C(x)$  en Simulink b) Evaluación de los polinomios de  $b(x)$  y  $C(x)$  en Xcos.

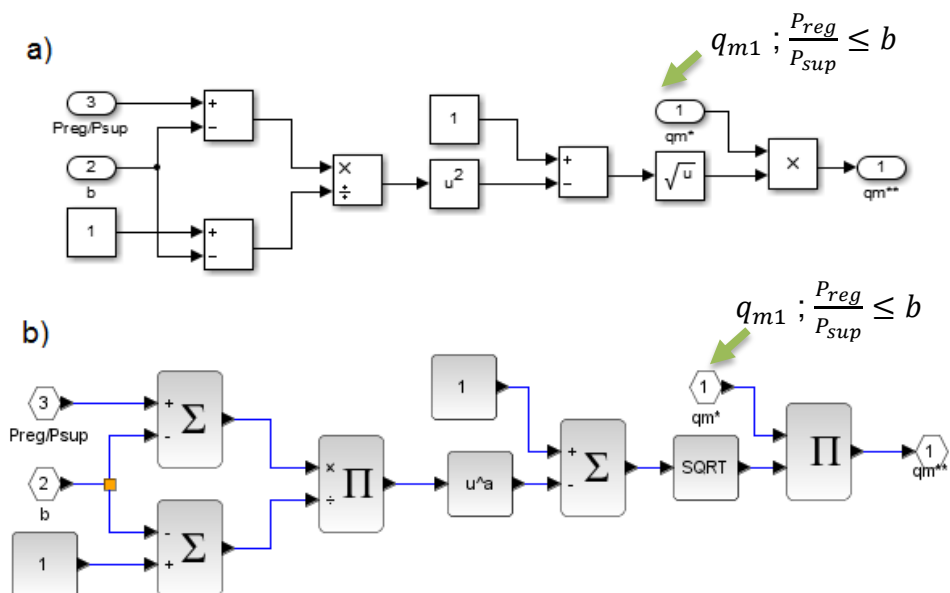
Los ambientes de programación definen de forma similar las funciones  $b(x)$  y  $C(x)$  dentro de las simulaciones. Ahora analizaremos los subsistemas internos. De acuerdo a la ecuación (3.13) en su primer segmento cuando  $\frac{P_{reg}}{P_{sup}} \leq b$  el flujo másico no depende de  $P_{reg}$  así como lo observamos en la Figura 4.7. La entrada de la conductancia sónica  $C$  y el resto de sus componentes expresados con las constantes que se detallan en el Anexo 1 y el Anexo 2.



**Figura 4.7:** Programación del subsistema neumático a)  $q_{m1}$  cuando  $P_{reg}/P_{sup} \leq b$  en Simulink b)  $q_{m1}$  cuando  $P_{reg}/P_{sup} \leq b$  en Xcos.

En el segundo segmento de la ecuación (3.13) existen los términos  $\frac{P_{reg}}{P_{sup}}$  y  $b$ . Dentro del cálculo del flujo másico en la Figura 4.8 se manifiesta que ambos términos entran al subsistema. Además existe una particularidad, si prestamos atención en la ecuación (3.13), esta parte de la ecuación es el segmento calculado en la Figura 4.7 multiplicado por una raíz. Entonces el cálculo de este segmento de la

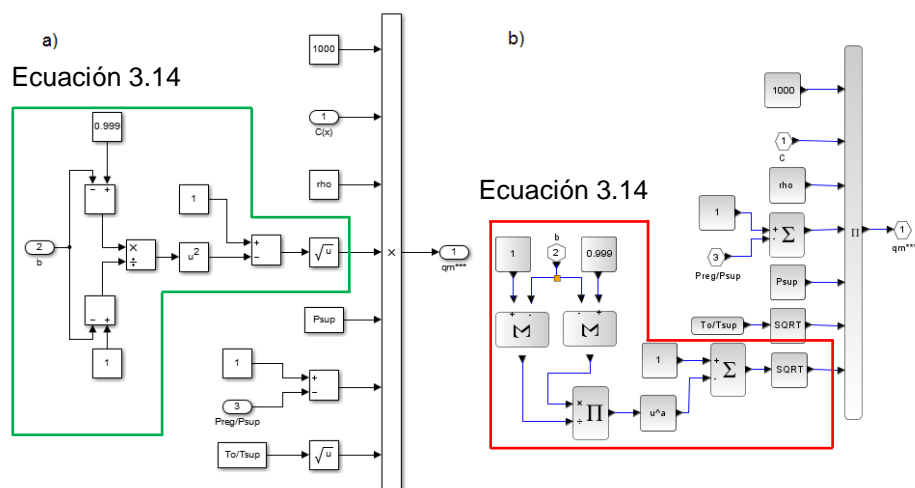
ecuación consiste en armar, con bloques de operaciones matemáticas, los términos dentro de la raíz y multiplicarla por el término anterior como se muestra en la Figura 4.8.



**Figura 4.8:** Programación del subsistema neumático a)  $q_{m1}$  cuando  $0.999 > P_{reg}/P_{sup} > b$  en Simulink b)  $q_{m1}$  cuando  $0.999 > P_{reg}/P_{sup} > b$  en Xcos.

Como fue descrito al final de la sección 3.4 existe un problema de convergencia cuando  $P_{reg} \rightarrow P_{sup}$ . La derivada del flujo másico con respecto al cambio del diferencial de presión, tiende al infinito si la presión diferencial ( $\Delta P = P_{reg} - P_{sup}$ ) se aproxima a cero [7]. Para solucionar este problema dentro de los simuladores se introdujo un tercer término que es la linealización del segmento anterior. Conservando la misma respuesta en la salida de presión. La Figura 4.9 presenta al tercer segmento formado por bloques donde se señala

el término  $K_1$ , ecuación (3.14), que elimina el problema de convergencia y aproxima la curva a su respuesta real en las condiciones máximas de trabajo.

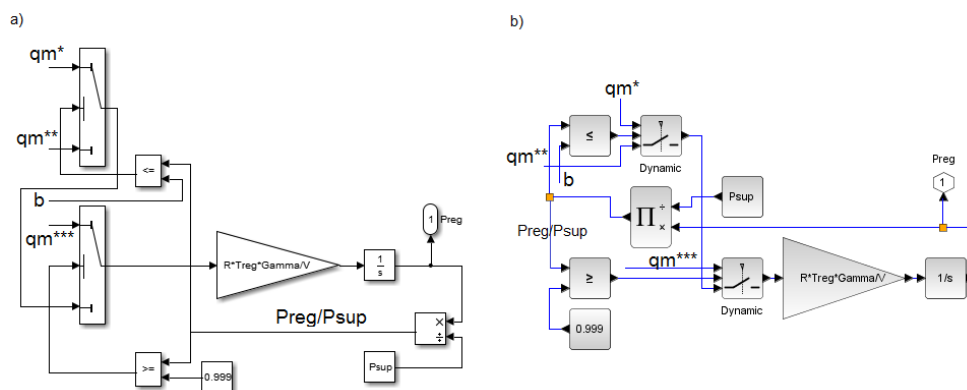


**Figura 4.9:** Programación del subsistema neumático a)  $q_{m1}$  cuando  $P_{reg}/P_{sup} \geq 0.999$  en Simulink b)  $q_{m1}$  cuando  $P_{reg}/P_{sup} \geq 0.999$  en Xcos.

En las figuras anteriores describimos la respuesta del flujo másico en función de  $P_{reg}$ . Sin embargo como se especificó en la sección 3.4 existe una relación entre los flujos másicos a la entrada y a la salida de la cámara de baja presión con la presión regulada de salida mostrada en la ecuación (3.10). La simulación del modelo consiste en observar la respuesta dinámica de la presión de salida. Para lograr este objetivo asumiremos que no existe salida de flujo de la válvula, es decir  $q_{m2}$  es igual a cero, como se muestra en la siguiente ecuación:

$$qm_1 - \overbrace{qm_2}^0 = q_{m1} = \frac{V}{\gamma RT} \frac{dP_{reg}}{dt} \quad (4.3)$$

Esta condición permite que la presión llegue a su valor máximo estableciendo el tiempo mínimo en que la presión se estabiliza, su curvatura, entre otras características útiles al momento de diseñar y seleccionar un controlador, lo cual no es parte de este trabajo. La Figura 4.10 muestra los dos conmutadores utilizados para definir el flujo másico de acuerdo a las condiciones establecidas en la ecuación (3.13). Este valor será integrado para hallar la  $P_{reg}$  del sistema.



**Figura 4.10:** Programación del subsistema neumático a) Conexión de los conmutadores de acuerdo a las condiciones de la Ecuación (3.13) en Simulink b) Conexión de los conmutadores de acuerdo a las condiciones de la Ecuación (3.13) en Xcos

Para terminar con la programación de la simulación establecimos los parámetros que utilizará el software para proporcionar los resultados. Empezamos con Matlab/Simulink. La Figura 4.11 muestra el cuadro de los parámetros de la simulación. Primero definimos el tipo de *solver*

para nuestra simulación. Utilizamos *Variable – step*, el cual varía cada paso dependiendo de la dinámica del modelo, lo que provoca que se acorte el tiempo de simulación. Es recomendado para modelos donde existen rápidos cambios de estado o que contiene discontinuidades [5]. Para escoger el *solver* debemos tener en cuenta que es el que determina el paso de la simulación y aplica un método numérico para resolver el conjunto de ecuaciones diferenciales ordinarias que representan el modelo [5]. Hemos considerado el método *ode45 (Dormand-Prince)* que utiliza el método numérico *Runge - Kutta* para la resolución de ecuaciones diferenciales. También modificamos el campo de *Relative tolerance* que determina el máximo error aceptable para el modelo. El resto de valores son proporcionados por defecto en Matlab/Simulink.

Solver options	
Type:	Variable-step
Max step size:	auto
Min step size:	auto
Initial step size:	auto
Number of consecutive min steps:	1
Solver:	ode45 (Dormand-Prince)
Relative tolerance:	1e-6
Absolute tolerance:	auto
Shape preservation:	Disable All

**Figura 4.11:** *Solver options* en Simulink

En la Figura 4.12 tenemos el setup de Scilab/Xcos. Se denota una diferencia principal no existe la definición del tipo de *solver*, sino que existe una opción llamada *solver kind* el cual define el tipo y el método con el cual el programa trabajará la simulación al mismo tiempo. Para nuestro caso la simulación utiliza el tipo *CVode (Variable – coefficients*

*ODE solver*) el mismo que es un método numérico que resuelve Ecuaciones diferenciales ordinarias con valores iniciales. Maneja el método de integración implícito de “*Fórmulas de diferencias hacia atrás*” (BDF) o también llamado *Corrector de Adams* y el método iterativo de Newton. El resto de valores se definen por defecto [6] así con en Matlab/Simulink.

Final integration time	6.0E01
Real time scaling	0.0E00
Integrator absolute tolerance	1.0E-06
Integrator relative tolerance	1.0E-06
Tolerance on time	1.0E-10
Max integration time interval	1.00001E05
Solver kind	Sundials/CVODE - ADAMS - NEWTON
Maximum step size (0 means no limit)	0.0E00
Set Context	

**Figura 4.12:** *Simulation Setup* en Xcos

Una vez definidos los parámetros de la simulación para dos de los simuladores donde la programación es gráfica empezaremos con la programación en OpenModelica. El programa es nuevo para nosotros sin embargo ofrece la facilidad que para modelar nuestro sistema necesitamos un código que contengan las ecuaciones y nos facilite la respuesta de sus variables en el tiempo. Es más sencillo definir y graficar valores que en los simuladores mencionados anteriormente. Para esto hay que familiarizarse con el software para explotar todas sus funciones.



## **4.2 Simulación en lenguaje funcional del sistema electro-neumático**

Como fue planteado en la sección 2.3 el programa OpenModelica se basa en ecuaciones para modelar complejos sistemas físicos ofreciendo dos tipos de programación: gráfica y por código. Dentro del paquete que ofrece OpenModelica se encuentra el programa OMEdit (OpenModelica Connection Editor) que presenta la opción de realizar una programación tanto en bloques como en lenguaje de código. Para desarrollar la simulación del modelo realizamos la programación en lenguaje de código debido a que en las ecuaciones que describen la válvula existen términos que en los programas anteriores fueron definidos por bloques de funciones opción que no presenta OMEdit. Además el simulador presenta la ventaja que al programar no requiere de palabras reservadas para aplicar métodos matemáticos en el cálculo de los valores las variable en cada ecuación marcando una diferencia con los otros programas.

De la misma forma que se planteó en los simuladores anteriores. La programación de la simulación ha sido dividida en tres subsistemas. La programación en OMEdit se define por clases. Cada clase tiene su propia característica. Como vamos a programar un modelo

matemático utilizaremos la clase *model*. Para empezar vamos a declarar las constantes que forman parte de las ecuaciones que describen cada subsistema así como lo muestra la Figura 4.13. Para declarar constantes utilizamos la palabra *parameter* y el tipo de dato que representa cada constante. Como todos los datos deben ser del mismo tipo se utilizaremos *Real* como estándar.

```
//Declaración de constantes
parameter Real Vo = 10; // Escalón de voltaje [V]
parameter Real Rsol = 13.758; // Resistencia del solenoide [ohmios]
parameter Real K2 = 1.9487e-5; // Constante de la inductancia
parameter Real Ad = 7.8540e-5; // Area del diafragma [m2]
parameter Real Patm = 1.013e5; // Presión atmosférica [Pa]
parameter Real b = 1.00; // Coeficiente de amortiguamiento N/(m/s)
parameter Real mp = 1.00e-3; // Masa del vástago [Kg]
parameter Real Kr = 100.00; // Constante del resorte [N/m]
parameter Real Psup = 2e5; // Presión de alimentación de la válvula [Pa]
parameter Real To = 293.15; // Temperatura referencial del aire [K]
parameter Real Tsup = 293.15; // Temperatura del aire en la alimentación [K]
parameter Real rho = 1.185; // densidad referencial del aire [kg/m3]
parameter Real V = 0.08; // Volúmen de la camara de baja presión [m3]
parameter Real R = 288; // Constante universal de los gases J/(kg.K)
parameter Real Gamma = 1.4; // Constante adiabática de los gases
parameter Real Treg = 291.15; // Temperatura a la salida del aire
parameter Real coef1[:] = {0.71748, -2.7384, 7.6847,
-8.3177, 3.1618}; // Coeficientes de la función b(x)
parameter Real coef2[:] = {5.004e-9, 2.1902e-8, 3.7653e-7, -2.2269e-6,
5.4385e-6, -6.9209e-6, 4.5064e-6, -1.1858e-6}; // Coeficientes de la función C(x)
```

**Figura 4.13:** Declaración de constantes en OMEdit

Como siguiente punto realizaremos la declaración de variables dentro del sistema. Es importante saber que al programar en OMEdit el simulador requiere que se evidencie el mismo número de ecuaciones y de variables. Si esto no se cumple el programa no simula y devuelve un error. Mediante programación podemos limitar los valores de las variables y darles un valor inicial. Esto lo aplicamos para el desplazamiento del cabezal el cuál no puede ser cero para evitar inductancia infinita. Todo lo expresado se puede observar en la Figura 4.14.

```

Real x(start = 0.064e-3, min = 0.064e-3); // Desplazamiento
//del cabezal
Real i(start = 0); // Corriente del solenoide
Real Fo(start = 0); // Fuerza inicial del resorte
Real Fmag(start = 0); // Fuerza magnética
Real xprima(start = 0); // Velocidad del cabezal
Real xd(start = 0); // Desplazamiento del núcleo del solenoide
Real bx(start = 0); // Función de la razón de presión crítica
Real Cx(start = 0); // Función de la conductancia sónica
Real m(start = 0); // Primer término de la ecuación 3.13
Real K1(start = 0); // Ecuación 3.14
Real Preg(start = 0); // Presión regulada
Real div(start = 0); // Razón entre Preg y Psup
Real qml(start = 0); // Flujo másico a la entrada de la
//cámara de baja presión

```

**Figura 4.14:** Declaración de variables en OMEdit

Con las variables y sus constantes definidas plantearemos las ecuaciones que conforman cada subsistema. Para hacerlo la clase *model* posee la palabra reservada *equation*. Debajo de esa palabra ubicaremos todas las ecuaciones, principales e intermedias, que conforman el modelo. Iniciamos con el subsistema electro-magnético descrito por las ecuaciones (3.4) y (3.6). Como observamos en la Figura 4.15 solo basta con escribir las ecuaciones debajo de la palabra *equation* sin ser necesario el uso de palabras reservadas para su resolución.

```

equation

// Subsistema Electro-magnético

Vo - i * Rsol - i * (K2 / xd ^ 2) * der(xd) -
K2 / xd * der(i) = 0; // Ecuación 3.6

Fmag = 0.5 * i ^ 2 * (K2 / xd ^ 2); // Ecuación 3.3

```

**Figura 4.15:** Subsistema electro-magnético en OMEdit

En el subsistema mecánico existen ecuaciones intermedias que deben ser planteadas como por ejemplo la fuerza precargada en el resorte. En los simuladores analizados anteriormente al momento de integrar el desplazamiento se limitaba en los valores de trabajo de  $x_d$  en el bloque integrador. Para lograr este efecto en OMEdit utilizamos un *if conditional* para simular el efecto del saturador cuando sobrepasa el valor requerido. Este efecto se observa dentro de la válvula con limitadores como el resorte. La programación del sistema mecánico se muestra en la Figura 4.16

```
// Subsistema Mecánico

xprima = der(x); // derivada del desplazamiento del cabezal

Fo = Patm * Ad - Kr * 0.064e-3; // Ecuación de la fuerza inicial del resorte

Fmag - Fo + Patm * Ad - Preg * Ad - Kr * x -
b * xprima - mp * der(xprima) = 0; // Ecuación 3.8

if x < 1.016e-3 then // limitador del desplazamiento
  xd = x;
else
  xd = 1.016e-3;
end if;
```

**Figura 4.16:** Subsistema mecánico en OMEdit

Para simular el sistema neumático primero debemos evaluar  $x_d$  en las funciones de  $b(x)$  y  $C(x)$  para lograrlo debemos utilizar una función externa al código utilizado. La función *PolyEval* (Anexo 4) se encuentra dentro de OMNotebook perteneciente a OpenModelica. Para poder utilizarla es necesario abrirla en el mismo proyecto y hacerle un llamado así como se muestra en la Figura 4.17. Con los valores de las funciones  $b(x)$  y  $C(x)$  procedemos a programar la ecuación (3.13).

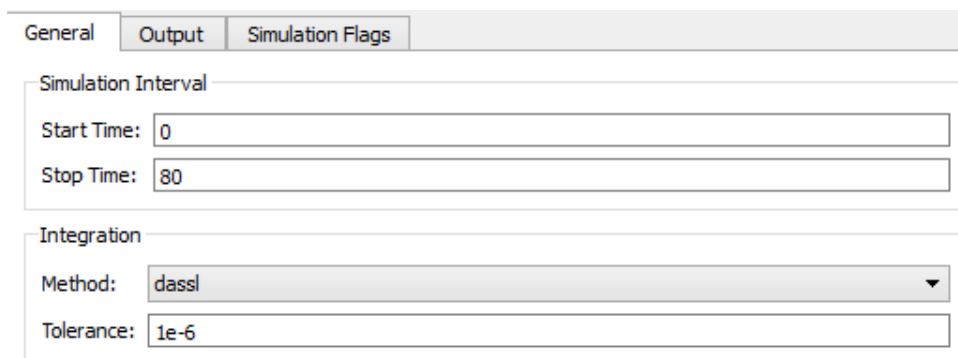
Con el objetivo de realizar una programación ordenada establecimos dos variables intermedias:  $m$  que es el primer segmento de la ecuación (3.13) y  $K_1$  que es la ecuación (3.14). Para establecer la ecuación (3.13) que describe a  $q_{m1}$  utilizamos otro *if conditional* definiendo los tres segmentos de la ecuación y sus condiciones comparadas con la razón entre  $\frac{P_{reg}}{P_{sup}}$  expresada con la variable *div*. Finalmente aplicamos la Ecuación (4.3) para hallar la Presión regulada a la salida  $P_{reg}$  así como lo señala la Figura 4.17.

```
// Subsistema Neumatico
bx = PolyEval(coef1, xd * 1000); // Evaluacion funcion b(x)
Cx = PolyEval(coef2, xd * 1000); // Evaluacion funcion C(x)
m = Psup * Cx * rho * sqrt(To / Tsup); // Primer segmento Ecuacion 3.13
K1 = 1000 * Cx * rho * sqrt(1 - ((0.999 - bx) / (1 - bx)) ^ 2); // Ecuacion 3.14
div = Preg / Psup; // Razon entre Preg y Psup
if div <= bx then // Ecuacion 3.13
    qm1 = m;
elseif div > bx and div < 0.999 then
    qm1 = m * sqrt(1 - ((div - bx) / (1 - bx)) ^ 2);
elseif div >= 0.999 then
    qm1 = K1 * Psup * (1 - div) * sqrt(To / Tsup);
else
    qm1 = 0;
end if;
Gamma * R * Treg / V * qm1 - der(Preg) = 0; // Ecuacion 4.3
```

**Figura 4.17:** Subsistema neumático en OMEdit

El código explicado se encuentra en el Anexo 3. Observamos que no apareció la exigencia de escribir palabras reservadas para graficar las señales. El programa las ubica dentro de la pestaña *plotting* donde se encuentran todas las variables declaradas anteriormente las mismas que se pueden observar en la ventana que aparece en dicha pestaña.

Para dar paso a la simulación debemos plantear el tiempo de simulación y el método numérico. Utilizaremos el método por defecto del programa. El *dassl* (*differential/algebraic system solver*) utilizado para solución de ecuaciones implícitas y para sistemas de ecuaciones diferenciales ordinarias. El tiempo de simulación es el mismo establecido para las pruebas y la tolerancia al error de simulación es la misma que en los otros simuladores. La ventana de configuración del OMEdit se muestra en la



The image shows a screenshot of the OMEdit simulation configuration window. At the top, there are three tabs: 'General', 'Output', and 'Simulation Flags'. The 'Simulation Interval' section contains two input fields: 'Start Time' with the value '0' and 'Stop Time' with the value '80'. The 'Integration' section contains a 'Method' dropdown menu set to 'dassl' and a 'Tolerance' input field set to '1e-6'.

**Figura 4.18:** Ventana de configuración de la simulación en OMEdit

Establecidos los parámetros y la programación para los tres simuladores propuestos. Analizaremos los resultados ofrecidos por los programas en varios sentidos. Formas de gráficas, tiempos de ejecución y exactitud de resultados. Estos resultados marcarán diferencias claras entre los programas seleccionados. Permittedo dar un criterio sobre las ventajas y desventajas entre cada uno.

# CAPÍTULO 5

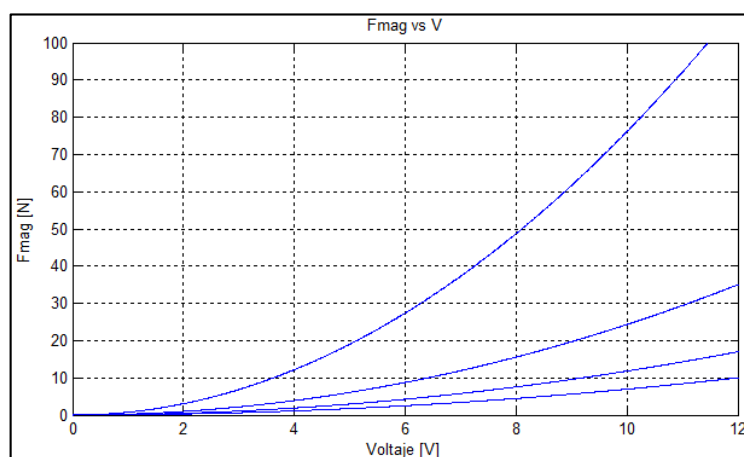
## 5. Resultados

En el CAPÍTULO 4 describimos la implementación del modelo matemático en los simuladores propuestos especificando los parámetros de simulación. En este capítulo recopilamos los resultados de las simulaciones y realizamos el análisis en base a las condiciones y valores teóricos detallados en capítulos anteriores. Interpretamos las respuestas del modelo simulado para verificar si el resultado de la simulación se ajusta al comportamiento de una válvula reguladora de presión. Además en esta sección también contrastamos los resultados entre los simuladores así como su eficiencia al ejecutar las simulaciones.

### 5.1 Subsistema Electromagnético

El subsistema electromagnético detalla el efecto que se produce cuando un voltaje DC se aplica a un solenoide con el objetivo de producir una fuerza magnética. En la sección 3.1 detallamos en base

a un estudio teórico como deben ser las gráficas de la Fuerza magnética  $F_{mag}$  con respecto al Voltaje/Corriente aplicada  $V/I$  y con respecto al desplazamiento  $x_d$ . En la Figura 5.1 observamos la primera de las gráficas. Teóricamente, la Fuerza magnética es proporcional para todo  $V > 0$ . Sin embargo podemos apreciar que la proporcionalidad se puede aproximar a cada curva de la Figura 5.1, existiendo una zona limitada donde esto se cumple. Esto causa problemas al momento de realizar un control, no obstante puede lograrse esta condición al aplicar un voltaje PWM al solenoide.

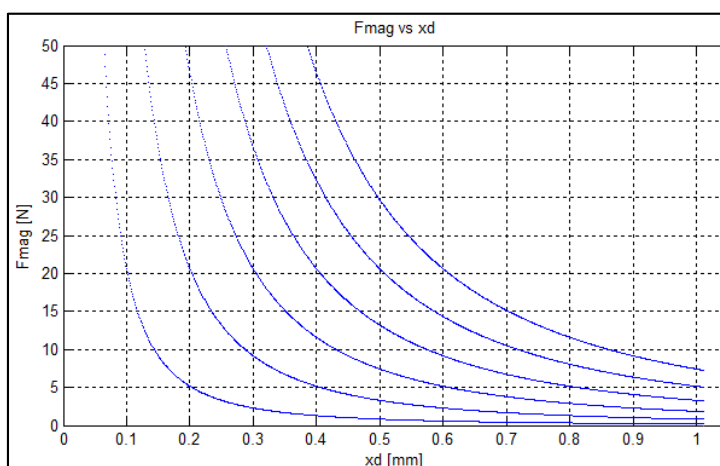


**Figura 5.1:** Fmag [N] vs Voltaje [V]

La Figura 5.2 describe el comportamiento de la Fuerza magnética con respecto al desplazamiento para diferentes valores de Voltaje nos fijamos que la fuerza es inversamente proporcional al desplazamiento. Además mientras menos voltaje se aplique un mayor segmento de la curva es horizontal. Esto se interpreta físicamente a la necesidad de conservar una fuerza constante mientras aumenta el desplazamiento



para no estirar el diafragma de tal forma que este no pierda sus propiedades elásticas tal como se detalla en la sección 3.1. Otro punto muy importante que nos expresa la Figura 5.2 es que cuando haya un retroceso debido a la fuerza que ejerce la presión regulada en la salida el solenoide aumenta su fuerza evitando que el orificio sea cerrado en su totalidad para permitir el paso de aire.

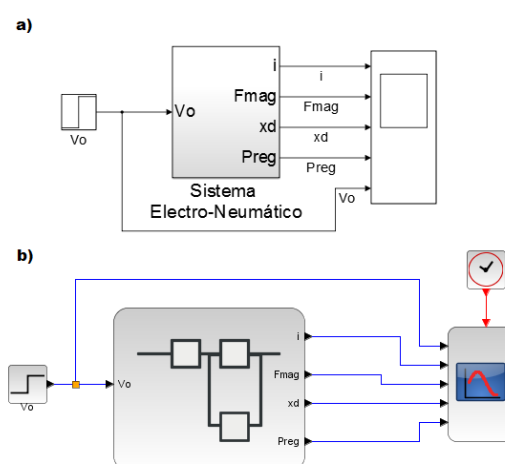


**Figura 5.2:** Fmag [N] vs xd[mm]

La Figura 5.1 y la Figura 5.2 muestran que el modelo matemático implementado representa el comportamiento del solenoide. El funcionamiento adecuado del solenoide es parte fundamental para la validez del modelo de la válvula. Ahora analizaremos las variables de interés del modelo completo. Desde el punto de vista dinámico, para  $q_{m2} = 0$ , examinaremos la forma de curva resultado de la simulación en los tres programas. De igual manera prestando atención a las gráficas de las variables de interés determinamos si las formas de onda tienen coherencia con lo desarrollado en el CAPÍTULO 3.

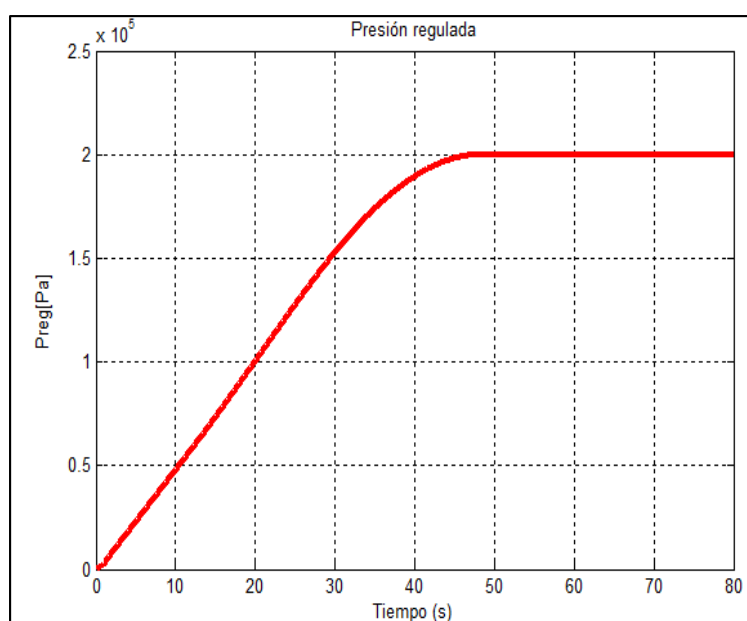
## 5.2 Respuesta del modelo matemático a un escalón de voltaje

En la Figura 5.3 observamos el bloque completo que representa el modelo en Simulink y Xcos. Este bloque en su interior posee cada subsistema detallado en el CAPÍTULO 3. Como podemos notar en la Figura 5.3 se grafican en los osciloscopios tanto de Simulink como de Xcos las cuatro variables de interés que hemos definido para este trabajo. Para OpenModelica como se indicó en la implementación basta con definir las variables para poder visualizarlas en el tiempo. La señal escalón de voltaje aplicado al sistema  $V_o$ , la corriente del solenoide  $i$ , la fuerza magnética del solenoide  $F_{mag}$ , el desplazamiento del cabezal  $x_d$  y la presión regulada de salida  $P_{reg}$ . Analizaremos cada señal basándonos en los análisis teóricos realizados en el CAPÍTULO 2 y el CAPÍTULO 2 para demostrar su validez.

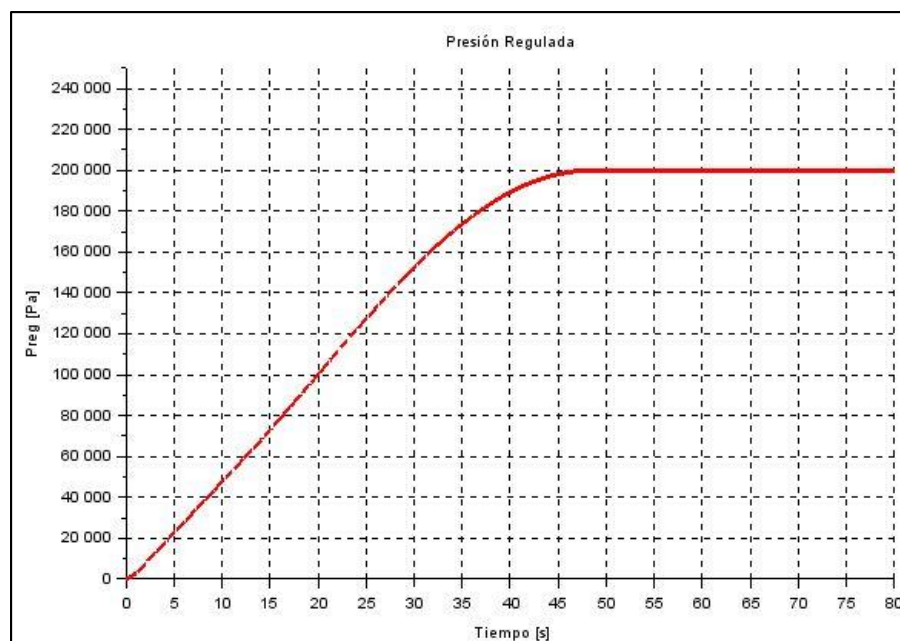


**Figura 5.3:** Modelo electroneumático: a) Implementado en Simulink  
b) Implementado en Xcos

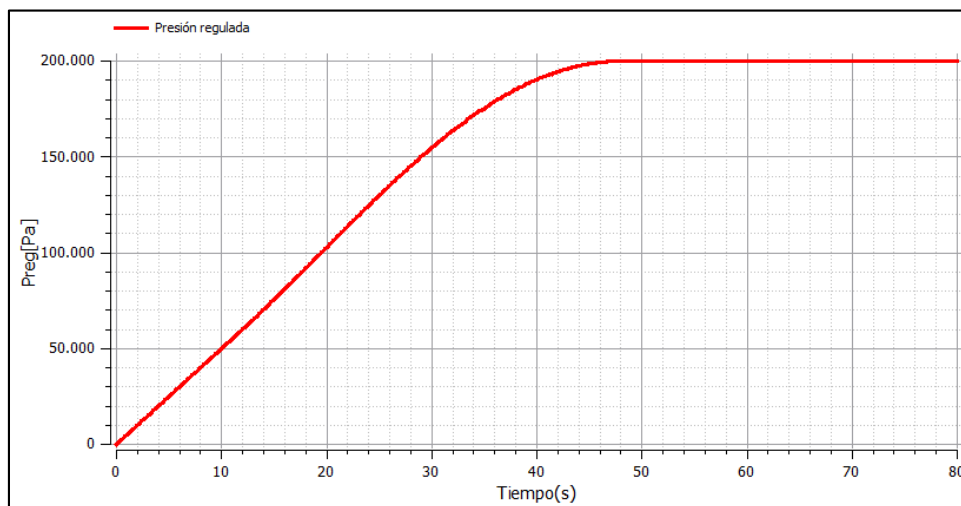
La Figura 5.4 (Simulink), Figura 5.5 (Xcos) y la Figura 5.6 (OpenModelica) nos muestran la Presión regulada a la salida de la válvula en función del tiempo observada. La gráfica obedece a lo expresado en la ecuación (3.13) al inicio al no haber una retroalimentación por parte de  $P_{reg}$  es simplemente proporcional al tiempo debido al integrador. Luego empieza a tomar una forma curva debido al segundo y tercer segmento de la ecuación (3.13). La gráfica se apoya también en los estudios de Varga y Honkola [13] [14] sobre válvulas reguladoras de presión. Como se expresa en el Anexo 1 la presión de alimentación de la válvula es  $P_{sup} = 2 \times 10^5 [Pa]$  entonces apreciamos como la presión regulada de salida se estabiliza en ese valor en un tiempo de 50 s aproximadamente.



**Figura 5.4:** Presión de salida en función del tiempo en Simulink



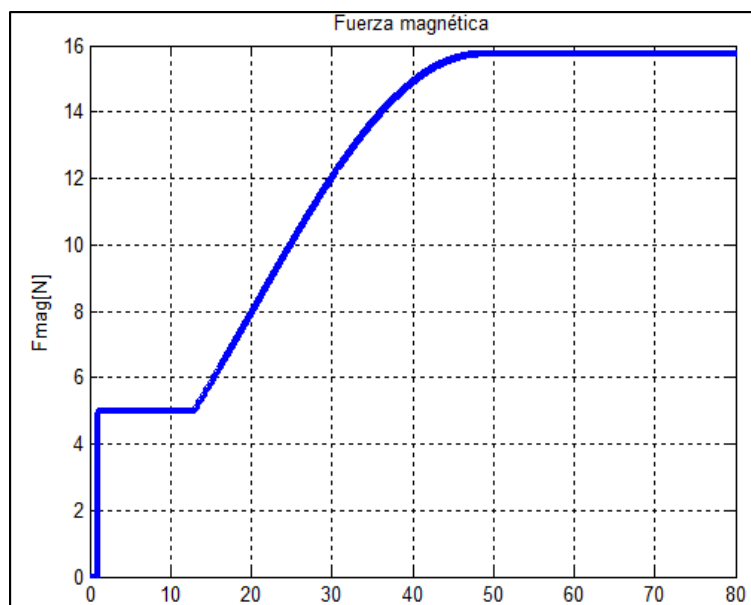
**Figura 5.5:** Presión de salida en función del tiempo en Xcos



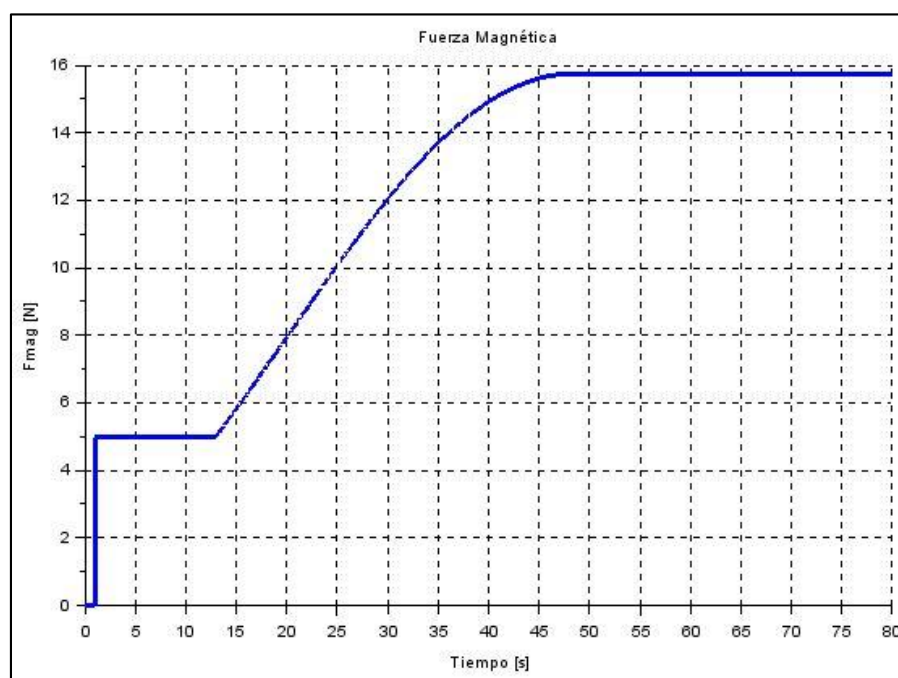
**Figura 5.6:** Presión de salida en función del tiempo en OpenModelica

Ahora para describir el desplazamiento que causa la apertura del orificio para alcanzar la respuesta de las figuras anteriores analizaremos la Fuerza magnética (Figura 5.7, Figura 5.8, Figura 5.9) y el desplazamiento (Figura 5.10, Figura 5.11, Figura 5.12). Como

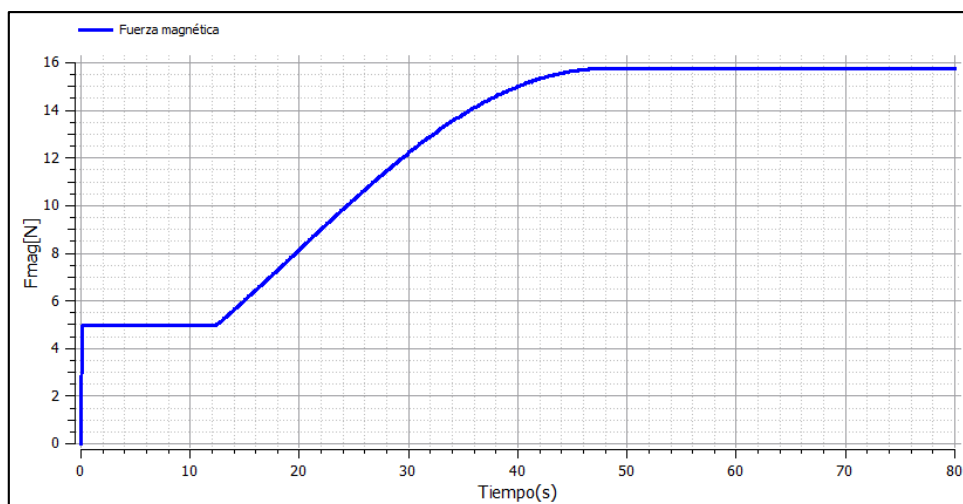
observamos al inicio de las figuras, cuando se da el escalón de Voltaje ambas llegan a un valor máximo. Tal como se enunció en la sección 3.1 el objetivo es lograr la mínima fuerza para el máximo desplazamiento. Luego cuando la presión  $P_{Reg}$  en la cámara de baja presión empieza a aumentar se produce una fuerza neumática sobre el diafragma descrito en la sección 3.3. En las figuras que señalan el desplazamiento (Figura 5.10, Figura 5.11, Figura 5.12) se distingue el efecto que produce esta fuerza cuando empieza a aumentar y sobrepasa la fuerza del solenoide. El desplazamiento matemáticamente disminuye en su valor sin sobrepasar los límites establecidos en la sección 4.1. Sin embargo físicamente de la curva se interpreta que el movimiento se produce en sentido contrario cerrando la apertura del orificio de la cámara de baja presión en la válvula. Como especificamos en la Figura 5.2 si el desplazamiento disminuye la fuerza magnética empieza a aumentar para mantener la apertura y por ende el paso de aire. Tal como se observa en las figuras de la Fuerza magnética (Figura 5.7, Figura 5.8, Figura 5.9). Finalmente notamos que cuando la presión regulada de salida llega al valor máximo la Fuerza magnética y el desplazamiento se estabilizan.



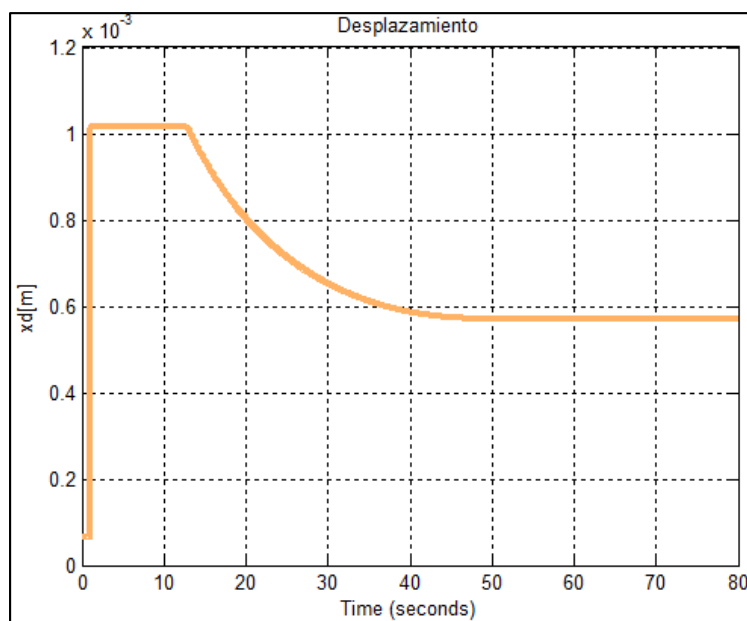
**Figura 5.7:** Fuerza magnética en el solenoide en función del tiempo en Simulink



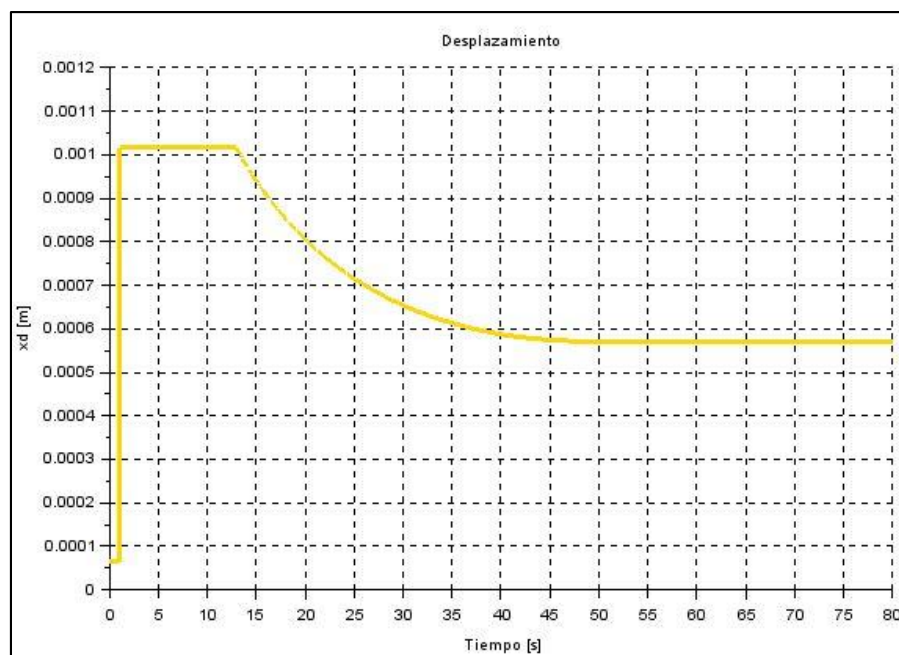
**Figura 5.8:** Fuerza magnética en el solenoide en función del tiempo en Xcos



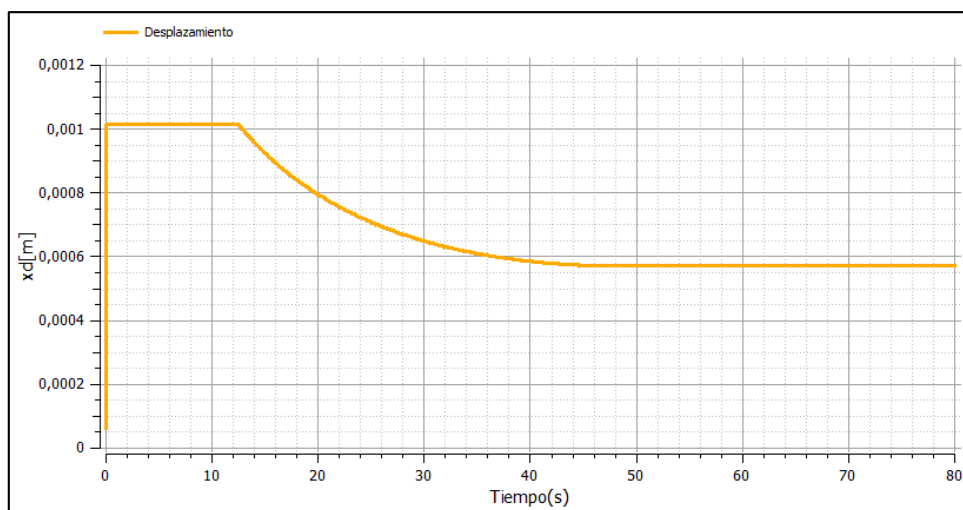
**Figura 5.9:** Fuerza magnética en el solenoide en función del tiempo en OpenModelica



**Figura 5.10:** Desplazamiento del cabezal en función del tiempo en Simulink



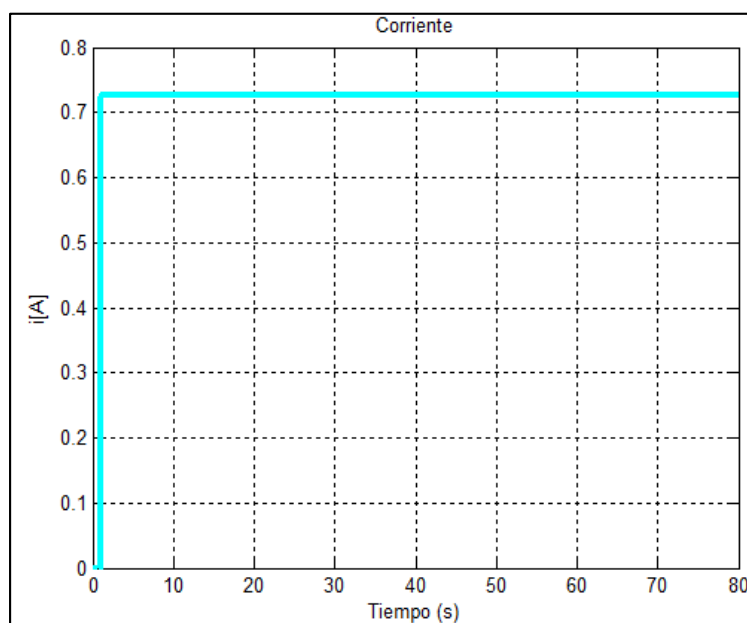
**Figura 5.11:** Desplazamiento del cabezal en función del tiempo en Xcos



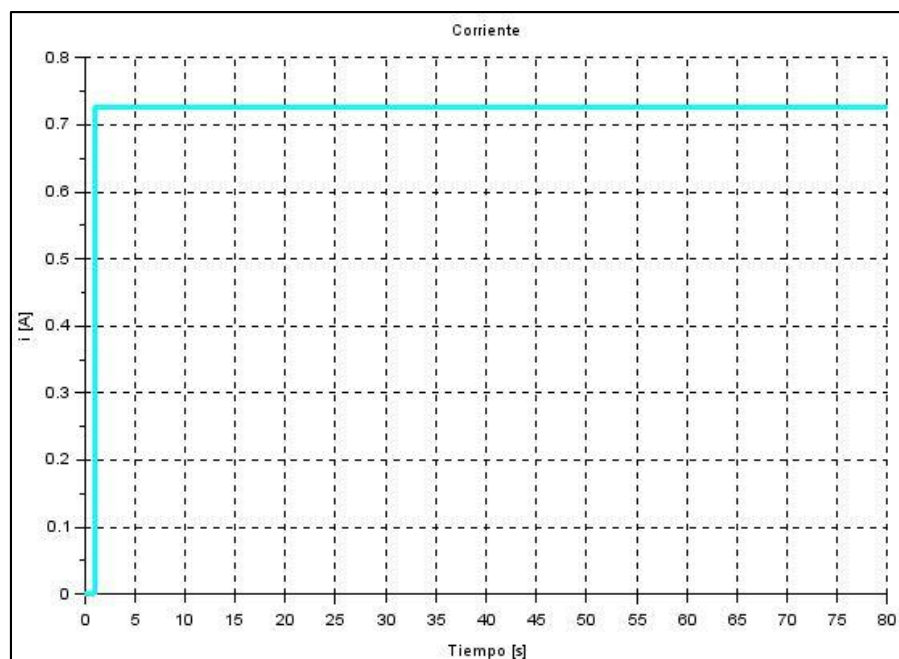
**Figura 5.12:** Desplazamiento del cabezal en función del tiempo en OpenModelica



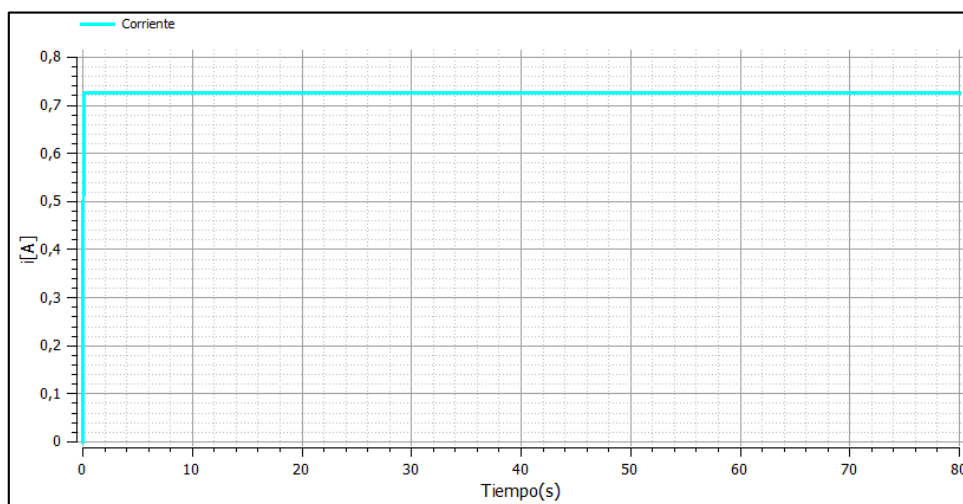
Analizaremos las dos variables eléctricas del sistema, la corriente en el solenoide (Figura 5.13, Figura 5.14, Figura 5.15) y la señal escalón de voltaje (Figura 5.16, Figura 5.17, Figura 5.18). Una de las principales ventajas de alimentar la válvula con voltaje DC es que obtenemos un valor fijo de corriente evitando sonidos molestos por interferencias así como lo muestran las figuras de la corriente (Figura 5.13, Figura 5.14, Figura 5.15). Notamos también que aunque dentro de la ecuación (3.6) la corriente depende de la inductancia y esta a su vez del desplazamiento su respuesta en el tiempo es indistinta de cambios en el desplazamiento. Dando una dominancia mayor por parte del voltaje aplicado que por el desplazamiento del cabezal.



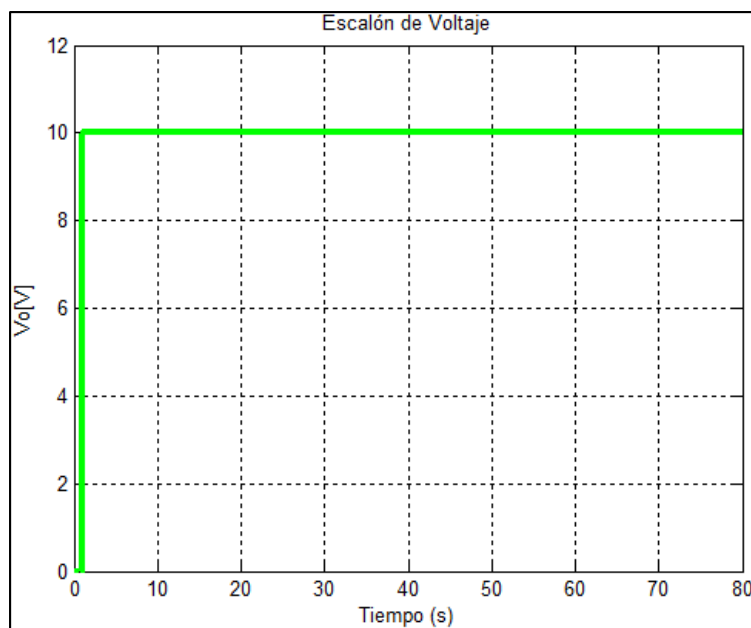
**Figura 5.13:** Corriente del solenoide en función del tiempo en Simulink



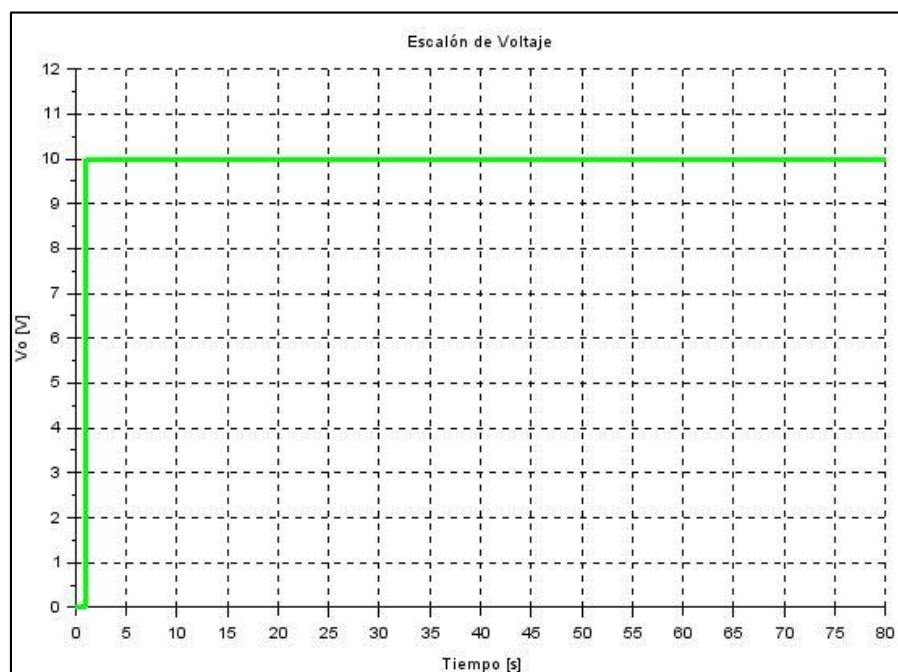
**Figura 5.14:** Corriente del solenoide en función del tiempo en Xcos



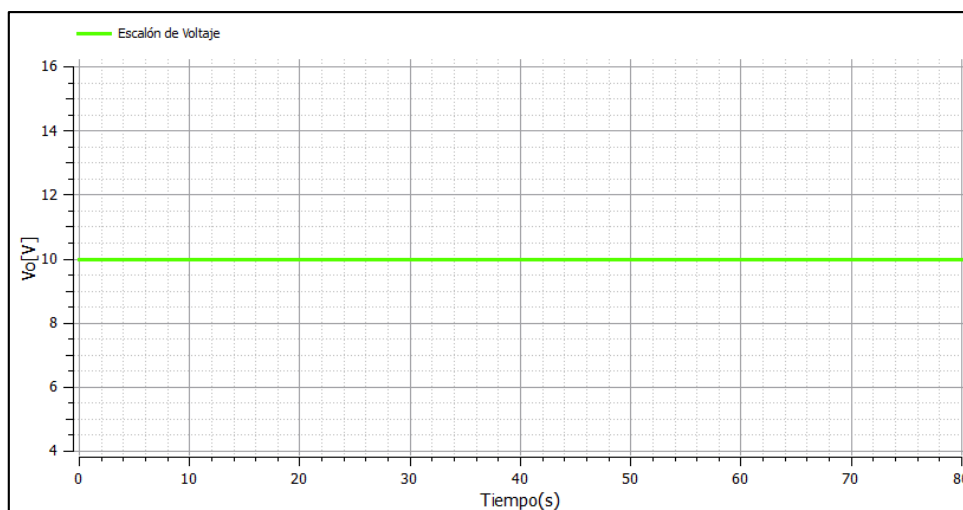
**Figura 5.15:** Corriente del solenoide en función del tiempo en OpenModelica



**Figura 5.16:** Escalón de Voltaje en Simulink



**Figura 5.17:** Escalón de Voltaje en Xcos



**Figura 5.18:** Escalón de Voltaje en OpenModelica

Observamos que las curvas simuladas en Matlab/Simulink, Scilab/Xcos y OpenModelica poseen la misma forma de onda. Se estabilizan en los mismos valores analizados anteriormente. Para comprobar la exactitud de las respuestas entre los simuladores es necesario realizar un análisis de error entre sus valores aplicando métodos estadísticos. El análisis del error dictará la exactitud de los resultados que proveen los simuladores luego de aplicar los métodos numéricos. Los datos que ofrecen los programas con respecto a tiempos de simulación y uso de la memoria del ordenador son factores que también debemos consultar antes de emitir un criterio sobre los programas. Los resultados adquiridos formaran parte del análisis comparativo del rendimiento de los softwares seleccionados.

### 5.3 Tiempo de ejecución y error en estado estable

Dentro de cualquier ámbito el tiempo y el error son parámetros importantes al medir eficiencia. En este trabajo medimos el tiempo de ejecución de los modelos implementados en los simuladores. Mientras que obtuvimos el error absoluto y relativo, de los valores de las variables, entre cada programa. Estas dos medidas nos permitieron sustentar los criterios de diferenciación entre un simulador y otro.

El tiempo de ejecución de los simuladores se obtiene de diferente manera para cada simulador. En Matlab utilizamos el botón *run and Time* para ejecutar el programa *CONSTANTES.m* del Anexo 1 y medimos su tiempo de ejecución. Para medir el tiempo total, la carga de las constantes más la ejecución del modelo en Simulink, utilizamos las dos últimas líneas de comando para mandar a correr el modelo en Simulink desde la función *CONSTANTES.m*. Obtuvimos los resultados mostrados en la Figura 5.19 con un tiempo total de 11.750 segundos de ejecución. En el simulador Scilab declaramos como una función el programa a ejecutar y mediante a las cuatro últimas líneas de código del Anexo 2 evaluamos su rendimiento. Los resultados presentados por el programa se detallan en la Figura 5.20 con un total de 195.32 segundos de ejecución. Para OpenModelica el resumen de la ejecución se obtiene de manera automática al ejecutar la simulación.

Los resultados se muestran en la Figura 5.21 donde se tiene un total de tiempo de ejecución de 0.0185742 segundos.

### Profile Summary

Generated 11-May-2015 12:39:10 using cpu time.

<u>Function Name</u>	<u>Calls</u>	<u>Total Time</u>
<u>CONSTANTES</u>	1	11.750 s

**Figura 5.19:** Resumen de rendimiento para el modelo en Matlab/Simulink

```

Scilab 5.5.2 Console
File Edit Control Applications ?
Scilab 5.5.2 Console
|1|0   |7 | 40:  xo = (1.016 - 0.952) * 0.001;//VALOR MÍNIMO DEL ENTREHIERRO
|1|0   |3 | 41:  dd = 0.01;//DÍAMETRO DEL DIAFRAGMA
|1|0   |3 | 42:  Kr = 100;//CONSTANTE DEL RESORTE
|1|0   |9 | 43:  Ad = %pi * ((dd/2)^2);//ÁREA DEL DIAFRAGMA
|1|0   |3 | 44:  mp = 0.001;//MASA DEL VÁSTAGO
|1|0   |9 | 45:  Fo = Patm * Ad - Kr * xo;//FUERZA PRECARGADA DEL RESORTE
|1|0   |0 | 46:
|1|0   |1 | 47:  //EJECUCIÓN DEL MODOELO DESDE SCILAB
|1|0.09|3 | 48:  xcoss('MODELO_MATEMATICO.zcos')
|1|0.86|4 | 49:  importXcosDiagram('C:\Users\Mechita\Google Drive\MODELAMIENTO :
|1|0   |4 | 50:  typeof(scs_m)
|1|0   |6 | 51:  scs_m.props.context
|1|194.37|5 | 52:  xcoss_simulate(scs_m, 4);
|1|0   |0 | 53:
|1|0   |0 | 54:  endfunction
-->
<

```

**Figura 5.20:** Resumen de rendimiento para el modelo en Scilab/Scos

```

Output  Compilation
C:/Users/Usuario/AppData/Local/Temp/OpenModelica/OMEdit/prueba.exe -port=52632 -
logFormat=xml -w -lv=LOG_STATS
LOG_STATS      | info      | ### STATISTICS ###
LOG_STATS      | info      | timer
|              | |         | | 0.00291916s      reading init.xml
|              | |         | | 0.00172686s      reading info.xml
|              | |         | | 0.00136378s      pre-initialization
|              | |         | | 9.06624e-005s [ 0.5%] initialization
|              | |         | | 9.40836e-006s [ 0.1%] steps
|              | |         | | 0.00249407s [ 13.4%] creating output-file
|              | |         | | 7.44116e-005s [ 0.4%] event-handling
|              | |         | | 0.000302778s [ 1.6%] overhead
|              | |         | | 0.0142391s [ 76.7%] simulation
|              | |         | | 0.0185742s [100.0%] total
LOG_STATS      | info      | events
|              | |         | | 6 state events
|              | |         | | 0 time events
LOG_STATS      | info      | solver: DASSL
|              | |         | | 600 steps taken
|              | |         | | 819 calls of functionODE
|              | |         | | 126 evaluations of jacobian
|              | |         | | 23 error test failures
|              | |         | | 0 convergence test failures
LOG_STATS      | info      | ### END STATISTICS ###

```

**Figura 5.21:** Resumen de rendimiento para el modelo en OpenModelica

Otro de los aspectos importantes al momento de definir el simulador más confiable es el error de las mediciones. Para este estudio, como lo señalamos anteriormente, no tenemos a disposición una válvula física con las características descritas. Por este motivo decidimos calcular los promedios y desviaciones estándar entre los datos de los simuladores en estado estable. Estas medidas estadísticas nos permiten visualizar el porcentaje de error entre los valores simulados. Los resultados de estos análisis los realizamos para las cuatro variables de salida. En la Tabla 1 ordenamos los valores simulados de cada programa en estado estable. Mientras que en la Tabla 2 presentamos el promedio, la desviación estándar y el porcentaje de

dicha desviación. Con estas medidas estadísticas podemos cuantificar la diferencia de resultados en estado estable de los tres simuladores.

**Tabla 1:** Resultados de la simulación en estado estable

<b>Resultados de la Simulación</b>	<b>MATLAB</b>	<b>SCILAB</b>	<b>MODELICA</b>
Corriente [A]	0,7272	0,7268	0,7268
Fuerza Magnética [N]	15,7587	15,7881	15,7588
Desplazamiento [mm]	5,7150	5,7128	5,7153
Presión [Pa]	20000,0000	20000,0000	199999,9990

**Tabla 2:** Estadística de los resultados simulados en estado estable

<b>Estadística de los Resultados</b>	<b>Prom.</b>	<b>Desviación Estándar</b>	<b>Porcentaje de Desviación</b>
Corriente [A]	0,7269	0,0002	0,0275%
Fuerza Magnética [N]	15,7685	0,0169	0,1072%
Desplazamiento [mm]	5,7144	0,0014	0,0239%
Presión [Pa]	199999,9997	0,0006	0,0000003%

Ya presentados todos los resultados de las simulaciones los contrastamos unos con otros. Esta comparación nos brindó bases para sustentar los criterios presentados en este trabajo sobre cada simulador. Las conclusiones y recomendaciones de este trabajo presentadas en las dos siguientes secciones son propias de los autores y su percepción respecto a los resultados puede diferir con el criterio del lector. Sin embargo creemos que las hemos fundamentado de la mejor manera en este informe.



# CONCLUSIONES

En los capítulos anteriores hemos definido las ecuaciones del modelo matemático, los parámetros de simulación y los resultados del mismo. Datos que nos permiten emitir un criterio sobre nuestro trabajo. Fundamentados en lo presentado en el CAPÍTULO 5.

Con los resultados obtenidos de los simuladores no podemos demostrar la validez del modelo matemático. Para validar el modelo se requiere de datos experimentales usando una válvula con las características descritas en este trabajo. Sin embargo los valores simulados nos permiten evaluar el desempeño de los programas basándonos en los tiempos de ejecución. OpenModelica tiene el menor tiempo de ejecución entre los tres simuladores. Esto se debe a que la implementación del modelo se realizó con lenguaje funcional. Mientras que en Matlab/Simulink y Scilab/Xcos, se utilizaron bloques

que requieren llamadas a funciones internas aumentando el tiempo de ejecución.

Matlab/Simulink ofrece un lenguaje gráfico con mayor facilidad de manejo y menor tiempo de ejecución que Scilab/Xcos. Los bloques de Scilab/Xcos son complicados de programar y sus parámetros para la presentación de datos requieren ser modificados antes de cada ejecución contrario a Matlab/Simulink que admite cambios luego de cada simulación. Open Modelica tiene un lenguaje funcional más sencillo de implementar que los otros dos simuladores pues no requiere de palabras reservadas para la resolución de ecuaciones.

La desviación estándar calculada para los valores en estado estable de cada variable de interés es: corriente ( $\sigma = 0,0002$ ), fuerza Magnética ( $\sigma = 0,0169$ ), desplazamiento ( $\sigma = 0,0014$ ) y presión regulada ( $\sigma = 0,0006$ ). Como se observa en la Tabla 2 el porcentaje de desviación es menor al 1% para cada variable. Por lo tanto, a pesar de ejecutarse con diferentes *solvers*, los simuladores tienen el mismo resultado en estado estable.

# RECOMENDACIONES

Para la elaboración de este trabajo existieron diversos tipos de limitaciones que dificultaron el proceso. Presentamos las recomendaciones a tomar en cuenta de acuerdo a nuestra experiencia.

Este tipo de trabajos requiere disponibilidad del sistema físico a modelar. Por lo que en futuras versiones de esta materia se deben plantear los proyectos en función de los sistemas con los que cuenta la universidad. Se debe implementar el modelo en el mismo lenguaje, funcional o gráfico, para tener un criterio más sólido de evaluación de los simuladores.

El requerimiento de la válvula puede ser sobrellevado siempre y cuando se tengan datos obtenidos por otros autores para la verificación del modelo y de la posterior simulación. Entonces es importante una correcta selección

bibliográfica que servirá para validar los resultados y para el desarrollo del trabajo.

El software de simulación Open Modelica debe familiarizarse a los estudiantes dentro de las materias profesionales de su malla curricular. Este programa es de código abierto, tiene un gran rendimiento y permite una clara relación entre las ecuaciones definidas y el comportamiento de los sistemas.

.

**ANEXOS**

## ANEXO 1

### CÓDIGO DE MATLAB CON LAS COSTANTES DEL SISTEMA

```
clear
clc
%CONSTANTES DEL SOLENOIDE TUBULAR SP1414P1200
%CONSTANTES GEOMÉTRICAS
d = 0.01; %DIÁMETRO DEL VÁSTAGO
y = (sqrt(3)*d)/(sqrt(3)+2); %DIÁMETRO INTERNO DEL VÁSTAGO
Rmay = d/2; %RADIO MAYOR
rmen = y/2; %RADIO MENOR
g = sqrt(y^2+(Rmay-rmen)^2); %GENERATRIZ
%ÁREA TRANSVERSAL DEL ENTREHIERRO
A1 = pi*g*(Rmay-rmen); %ÁREA LATERAL DEL CONO
TRUNCADO
A2 = pi*(y/2)^2; %ÁREA DE LA BASE MENOR DEL
C.T.
At = A1+A2; %ÁREA TOTAL
%CONSTANTES MAGNÉTICAS
N = 500; %NÚMERO DE VUELTAS DEL SOLENOIDE
uo = 4*pi*1e-7; %PERMEABILIDAD MAGNÉTICA DEL AIRE
K1 = 1/(uo*At); %CONSTANTE DE LA RELUCTANCIA DEL
ENTREHIERRO
K2 = (N^2)/K1; %CONSTANTE DE LA INDUCTANCIA
%CONSTANTES ELÉCTRICAS
Rsol = 13.758; %RESISTENCIA DEL SOLENOIDE
%CONSTANTES NEUMÁTICAS DE LA VÁLVULA FESTO VPPM-6L-L-1-G180L6H
Patm = 1.013e5; %PRESIÓN ATMOSFÉRICA EN Pa
Psup = 2e5; %PRESIÓN DE AIRE DE ALIMENTACIÓN DE LA VÁLVULA
Tsup = 293.15; %TEMPERATURA DEL AIRE DE ENTRADA DE LA VÁLVULA
To = 293.15; %TEMPERATURA DEL AIRE EN CONDICIONES
REFERENCIALES ISO6358
rho = 1.185; %DENSIDAD DEL AIRE EN CONDICIONES REFERENCIALES
ISO6358
Gamma = 1.4; %CONSTANTE ADIABÁTICA
R = 288; %CONSTANTE DE LOS GASES IDEALES
Treg = 291.15; %TEMPERATURA DEL AIRE EN EL PUNTO DE SALIDA
V = 0.08; %VOLUMEN DE SALIDA DE LA VÁLVULA

%CONSTANTES MECÁNICAS DE LA VÁLVULA FESTO VPPM-6L-L-1-G180L6H
B = 1; %CONSTANTE DE FRICCIÓN
xo = (1.016-0.952)*1e-3; %VALOR MÍNIMO DEL
ENTREHIERRO
dd = 0.01; %DÍAMETRO DEL DIAFRAGMA
Kr = 100; %CONSTANTE DEL RESORTE
Ad = pi*(dd/2)^2; %ÁREA DEL DIAFRAGMA
mp = 1e-3; %MASA DEL VÁSTAGO
Fo = (Patm*Ad)-(Kr*xo); %FUERZA PRECARGADA DEL
RESORTE
%CARGA Y SIMULACIÓN DEL MODELO
open_system('MODELO_MATEMATICO.slx')
sim('MODELO_MATEMATICO.slx')
```

## ANEXO 2

### CÓDIGO DE SCILAB CON LAS CONSTANTES DEL SISTEMA

```
//DECLARACIÓN DE FUNCIÓN PARA EVALUAR RENDIMIENTO DEL SIMULADOR
function [K1, K2, Rsol, Patm, Psup, Tsup, rho, Gamma, R, Treg, V, B, xo, Kr, Ad, mp,
Fo]=CONSTANTES()

//CONSTANTES DEL SOLENOIDE TUBULAR SP1414P1200

//CONSTANTES GEOMÉTRICAS
d = 0.01; //DIÁMETRO DEL VÁSTAGO
y = (sqrt(3)*d)/(sqrt(3)+2); //DIÁMETRO INTERNO DEL VÁSTAGO
Rmay = d/2; //RADIO MAYOR
rmen = y/2; //RADIO MENOR
g = sqrt(y^2+(Rmay-rmen)^2); //GENERATRIZ

//ÁREA TRANSVERSAL DEL ENTREHIERRO
A1 = %pi*g*(Rmay-rmen); //ÁREA LATERAL DEL CONO TRUNCADO
A2 = %pi*(y/2)^2; //ÁREA DE LA BASE MENOR DEL C.T.
At = A1+A2; //ÁREA TOTAL

//CONSTANTES MAGNÉTICAS
N = 500; //NÚMERO DE VUELTAS DEL SOLENOIDE
uo = 4*%pi*1e-7; //PERMEABILIDAD MAGNÉTICA DEL AIRE
K1 = 1/(uo*At); //CONSTANTE DE LA RELUCTANCIA DEL ENTREHIERRO
K2 = (N^2)/K1; //CONSTANTE DE LA INDUCTANCIA

//CONSTANTES ELÉCTRICAS
Rsol = 13.758; //RESISTENCIA DEL SOLENOIDE

//CONSTANTES NEUMÁTICAS DE LA VÁLVULA FESTO VPPM-6L-L-1-G180L6H
Patm = 1.013e5; //PRESIÓN ATMOSFÉRICA EN Pa
Psup = 2e5; //PRESIÓN DE AIRE DE ALIMENTACIÓN DE LA VÁLVULA
Tsup = 293.15; //TEMPERATURA DEL AIRE DE ENTRADA DE LA VÁLVULA
To = 293.15; //TEMPERATURA DEL AIRE A CONDICIONES REFERENCIAL ISO 6358
rho = 1.185; //DENSIDAD DEL AIRE A CONDICIONES REFERENCIAL ISO 6358
Gamma = 1.4; //CONSTANTE ADIABÁTICA
R = 288; //CONSTANTE DE LOS GASES IDEALES
Treg = 291.15; //TEMPERATURA DEL AIRE EN EL PUNTO DE SALIDA
V = 0.08; //VOLUMEN DE SALIDA DE LA VÁLVULA

//CONSTANTES MECÁNICAS DE LA VÁLVULA FESTO VPPM-6L-L-1-G180L6H
B = 1; //CONSTANTE DE FRICCIÓN
xo = (1.016-0.952)*1e-3; //VALOR MÍNIMO DEL ENTREHIERRO
dd = 0.01; //DIÁMETRO DEL DIAFRAGMA
Kr = 100; //CONSTANTE DEL RESORTE
Ad = %pi*(dd/2)^2; //ÁREA DEL DIAFRAGMA
mp = 1e-3; //MASA DEL VÁSTAGO
Fo = (Patm*Ad)-(Kr*xo); //FUERZA PRECARGADA DEL RESORTE

//EJECUCIÓN DEL MODOELO DESDE SCILAB
xcos MODELO_MATEMATICO.zcos
importXcosDiagram("C:\Users\Mechita\Google Drive\MODELAMIENTO SISTEMA
ELECTRONEUMÁTICO\SIMULACIONES\SCILAB\MODELO_MATEMATICO.zcos")
typeof(scs_m)
scs_m.props.context
```

```
xcos_simulate(scs_m,4);  
  
endfunction  
//EVALUACIÓN DEL SIMULADOR  
add_profiling("CONSTANTES");  
CONSTANTES();  
profile(CONSTANTES)  
showprofile(CONSTANTES)
```



### ANEXO 3

## CÓDIGO DE OPENMODELICA PARA EL MODELO MATEMÁTICO DE LA VÁLVULA REGULADORA DE PRESIÓN

model prueba "Modelo Matematico Valvula Electro-neumatica reguladora de  
presion"

//Declaracion de constantes

parameter Real Vo = 10; // Escalon de voltaje [V]

parameter Real Rsol = 13.758; // Resistencia del solenoide [ohmios]

parameter Real K2 = 1.9487e-5; // Constante de la inductancia

parameter Real Ad = 7.8540e-5; // Area del diafragma [m2]

parameter Real Patm = 1.013e5; // Presion atmosferica [Pa]

parameter Real b = 1.00; // Coeficiente de amortiguamiento N/(m/s)

parameter Real mp = 1.00e-3; // Masa del vastago [Kg]

parameter Real Kr = 100.00; // Constante del resorte [N/m]

parameter Real Psup = 2e5; // Presion de alimentacion de la valvula [Pa]

parameter Real To = 293.15; // Temperatura referencial del aire [K]

parameter Real Tsup = 293.15; // Temperatura del aire en la alimentacion [K]

parameter Real rho = 1.185; // densidad referencial del aire [kg/m3]

parameter Real V = 0.08; // Volumen de la camara de baja presion [m3]

parameter Real R = 288; // Constante universal de los gases J/(kg.K)

parameter Real Gamma = 1.4; // Constante adiabatica de los gases

parameter Real Treg = 291.15; // Temperatura a la salida del aire

parameter Real coef1[:] = {0.71748, -2.7384, 7.6847, -8.3177, 3.1618}; //

Coeficientes de la funcion b(x)

parameter Real coef2[:] = {5.004e-9, 2.1902e-8, 3.7653e-7, -2.2269e-6, 5.4385e-6, -  
6.9209e-6, 4.5064e-6, -1.1858e-6}; // Coeficientes de la funcion C(x)

Real x(start = 0.064e-3, min = 0.064e-3); // Desplazamiento del cabezal

Real i(start = 0); // Corriente del solenoide

Real Fo(start = 0); // Fuerza inicial del resorte

Real Fmag(start = 0); // Fuerza magnetica

Real xprima(start = 0); // Velocidad del cabezal

Real xd(start = 0); // Desplazamiento del nucleo del solenoide

Real bx(start = 0); // Funcion de la razon de presion critica

Real Cx(start = 0); // Funcion de la conductancia sonica

Real m(start = 0); // Primer termino de la ecuacion 3.13

Real K1(start = 0); // Ecuacion 3.14

Real Preg(start = 0); // Presion regulada

Real div(start = 0); // Razon entre Preg y Psup

Real qm1(start = 0); // Flujo masico a la entrada de la camara de baja presion

```

equation
// Subsistema Electro-magnetico
Vo - i * Rsol - i * (K2 / xd ^ 2) * der(xd) - K2 / xd * der(i) = 0; // Ecuacion 3.6
Fmag = 0.5 * i ^ 2 * (K2 / xd ^ 2); // Ecuacion 3.3 // Subsistema Mecanico
xprima = der(x); // derivada del desplazamiento del cabezal
Fo = Patm * Ad - Kr * 0.064e-3; // Ecuacion de la fuerza inicial del resorte
Fmag - Fo + Patm * Ad - Preg * Ad - Kr * x - b * xprima - mp * der(xprima) = 0; //
Ecuacion 3.8
if x < 1.016e-3 then // limitador del desplazamiento
xd = x;
else
xd = 1.016e-3;
end if;
// Subsistema Neumatico
bx = PolyEval(coef1, xd * 1000); // Evaluacion funcion b(x)
Cx = PolyEval(coef2, xd * 1000); // Evaluacion funcion C(x)
m = Psup * Cx * rho * sqrt(To / Tsup); // Primer segmento Ecuacion 3.13
K1 = 1000 * Cx * rho * sqrt(1 - ((0.999 - bx) / (1 - bx)) ^ 2); // Ecuacion 3.14
div = Preg / Psup; // Razon entre Preg y Psup
if div <= bx then // Ecuacion 3.13
qm1 = m;
elseif div > bx and div < 0.999 then
qm1 = m * sqrt(1 - ((div - bx) / (1 - bx)) ^ 2);
elseif div >= 0.999 then
qm1 = K1 * Psup * (1 - div) * sqrt(To / Tsup);
else
qm1 = 0;
end if;
Gamma * R * Treg / V * qm1 - der(Preg) = 0; // Ecuacion 4.3
end prueba;

```

## ANEXO 4

### FUNCIÓN POLYEVAL

```
function PolyEval
input Real a[:];
input Real x;
output Real y;
protected
Real xpower;
algorithm
y := 0;
xpower := 1;
for i in 1:size(a, 1) loop
y := y + a[i] * xpower;
xpower := xpower * x;
end for;
end PolyEval;
```

## BIBLIOGRAFÍA

- [1] Felipe Cruz, Victor De Negri, and Raul Guenther, "Mathematical Modeling of an Electropneumatic Pressure Regulator Servo-Valve," 2004.
- [2] Ogata Katsuhiko, *Dinámica de sistemas*. México: Prentice-Hall Hispanoamericana, 1987.
- [3] Bohdan Kulakowski, John Gardner , and J. Lowen Shearer, *Dynamic Modeling and Control of Engineering Systems*. New York: Cambridge University Press, 2007.
- [4] FESTO, "Válvulas proporcionales reguladoras," 241, 2008.
- [5] Matlab. Mathworks. [Online]. [www.mathworks.com](http://www.mathworks.com)
- [6] Scilab Enterprises. Scilab. [Online]. [www.Scilab.org](http://www.Scilab.org)
- [7] Modelon AB, *Modeling of Pneumatic Systems*. Suecia, 2010.
- [8] BICRON Electronics Company, "DC SOLENOIDS Technical guide and Applications Standard and Custom Solenoids for OEM Applications," 1205-13339,.
- [9] Hugh Young and Roger Freedman, *Física Universitaria*.: Pearson, 2009.
- [10] William Hayt and John Buck , *Teoría Electromagnética*. México: McGrawHill, 2007.
- [11] A.E Fitzgerald, Charles Kingsley, and Stephen Umans , *Máquinas Eléctricas*. México: McGrawHill.
- [12] Sarang Lunge , Shailaja Dr. Kurode, and Bhaskar Chhibber, "Proportional Actuator from On Off Solenoid Valve using Sliding Modes," 2013.
- [13] Z Varga and P Honkola, "Mathematical model of pneumatic proportional valve," 2012.
- [14] Z Varga and P Honkola, "Determination of flow rate characteristics for pneumatic valves," , 2012.