

POLITECNICO DI TORINO

MASTER's Degree in AEROSPACE ENGINEERING



MASTER's Degree Thesis

Modeling of an aerospace servovalve for electrohydraulic servomechanisms

Supervisors

Prof. PAOLO MAGGIORE

Prof. MATTEO DALLA VEDOVA

Eng. PIER CARLO BERRI

Eng. GAETANO QUATTROCCHI

Candidate

EUGENIO CAREDDA

OCTOBER 2022

Acknowledgements

Pietro Chillemi
Alberto Caredda
Maria Teresa Chillemi
Daniel Rivello

Table of Contents

List of Figures	VII
1 Introduction	1
2 Servo Mechanism for Flight Control	3
2.1 Hydro-Mechanical Actuators	4
2.2 Electro-Mechanical Actuators	5
2.3 Electro-Hydrostatic Actuators	5
3 Servo-Valve for position command	8
3.1 Flapper-Nozzle Valve	8
3.2 Technical characteristics of valves and non-linearities	10
3.2.1 Lapp condition	12
3.2.2 Threshold	13
3.2.3 Magnetic Hysteresis	14
3.2.4 Friction	15
3.2.5 Backlash	19
3.2.6 Offset valve	21
4 Simulink Model	22
4.1 High-Fidelity Model	23
4.1.1 Hysteresis, offset and interferences in the torque motor	25
4.1.2 Electromechanical model of the servovalve	26
4.1.3 Fluid dynamic model	33
4.1.4 Mechanical model of the jack	40
4.2 Low-Fidelity Models	42
5 Development of Computational models	43
5.1 Developing of Database	44

5.1.1	Databases	45
5.2	Lookup table	49
5.3	Genetic Algorithm	51
5.3.1	Development of the object function	55
5.3.2	Genetic Algorithm development	56
5.3.3	Results	58
5.4	Neural network	58
5.4.1	Training algorithm	60
5.4.2	Modelization	62
5.4.3	Diagnostic	65
6	EHA testing	71
6.1	EHA genetic algorithm	71
6.2	EHA neural network	71
6.3	EHA commands	72
6.3.1	Simplified neural network	73
6.3.2	Neural network model improvement	76
7	Conclusions and future developments	82
	Bibliography	85

List of Figures

2.1	Primary and secondary flight controls	3
2.2	Scheme of a hydro-mechanical actuator	5
2.3	Scheme of a electro-mechanical actuator	6
2.4	Scheme of a electro-hydrostatic actuator	6
3.1	Flapper-Nozzle Valve	9
3.2	Valve characteristic curves	11
3.3	Flow rate and pressure gains	12
3.4	Lapp conditions and their responses	13
3.5	Lapp condition scheme	13
3.6	Threshold	14
3.7	Magnetic hysteresis	15
3.8	Behaviour of the friction force	16
3.9	Coulomb friction model	17
3.10	Karnopp model	18
3.11	Simulink of Karnopp model	19
3.12	Borello friction model	20
3.13	Simulink of Borello friction model	20
4.1	Model representation in Simulink of the High-Fidelity model	23
4.2	Input block sub-system	24
4.3	PID controller's scheme	25
4.4	Model of torque motor hysteresis	26
4.5	Offset current model	26
4.6	Servo valve's electromechanical model	27
4.7	Gain GM calculation	28
4.8	Torque motor's magnet circuit	29
4.9	Servo valve's first stage dynamical model	30
4.10	Servo valve's first stage dynamical model	31

4.11	Control valve scheme 4/3-way directional	33
4.12	Zero-lapping valve's section of passing fluid when $XS>0$. . .	36
4.13	Fluid Dynamic Model's subsystem	37
4.14	CRH's subsystem	38
4.15	Model of pipe capacity	40
4.16	Fluid dynamic model	41
4.17	Mechanical model of the jack	42
5.1	Data concentration in databases	45
5.2	Simulink model to create the characteristic curves	46
5.3	Characteristic curves with parameter variations	47
5.4	3D surface characteristic curves	49
5.5	Look-up Table Simulink Model	50
5.6	Look-up Table results	51
5.7	Genetic algorithm scheme	52
5.8	Genetic algorithm results	57
5.9	Neural Network performances	60
5.10	Neural Network error histogram	60
5.11	Neural Network regression	61
5.12	Neural Network structure	63
5.13	Neural Network with database 1	64
5.14	Neural Network with database 2	65
5.15	Neural Network with database 3	66
5.16	NN single Q	67
5.17	NN single Q=0	68
5.18	Neural Network scheme	69
6.1	EHA model with genetic algorithm	71
6.2	Result of gensim command	72
6.3	EHA model with neural network	72
6.4	Step 0.1	73
6.5	Sin 0.1 @ 10π	74
6.6	Sin 0.01 @ 10π	74
6.7	Simplified Neural Network Simulink Model with 4 input variables	75
6.8	Simplified Neural Network Simulink Model with 2 input variables	76
6.9	Simplified Neural Network Simulink Model with 2 input vari- ables, 3D plot	77
6.10	Step 0.1	78

6.11	Step 0.01	78
6.12	Ramp 0.1	79
6.13	Ramp 0.01	79
6.14	Sin 0.01 @ 2π	80
6.15	Sin 0.1 @ 10π	80
6.16	Sin 0.01 @ 10π	81

Chapter 1

Introduction

The purpose of this thesis is to investigate different methods of dynamic fluid modeling for an electro hydrostatic actuator's servo valve. The modeling methods will be based on four variables taken as representative of the servo valve conditions. Four models of the servo-valve will be used in this thesis. The first, based on a Simulink Model, was more complex and reliable and was used as a reference. The others are based on three different methods to replace runtime computation with simpler and faster methods. The analyzed techniques are: Lookup Table, Genetic Algorithm, Neural Network. Each method uses the same parameters to detect the presence of a fault. The physical quantities that were used to identify the degradation of the servo valve are the overlap and the radial clearance. These quantities were used to direct addressing the lookup table, to develop the prognostic algorithm and to train the neural network. The more exact first model was used as a reference.

The lookup table is based on a data structure that allows the result to be discovered more quickly than by calculating or searching for it, by matching one or more criteria. The genetic algorithm minimizes an objective function that is calculated from these quantities and makes the response of the simplified model match that of the first. An optimization problem's population of potential solutions is created through iterations leading to improved answers.

Neural Networks are a subfield of machine learning methods. They are inspired by the structure and biological network that composes animal brains, and are made up of node layers, each of which has an input layer, one or more hidden layers, and an output layer. Each node, or artificial neuron, is

linked to another and has its own weight and threshold. If the output of any particular node exceeds the given threshold value, that node is activated and begins transferring data to the network's next tier. Otherwise, no data is sent to the next network layer.

The purpose of this study will be twofold: having the model of a complete actuator, obtaining a simplified model of the servo valve that obtains the outlet pressure in relation to the above parameters, will allow to replace the simplified model within the complete model of the actuator. This will be useful in project phase, as it will be possible to test the actuator operation with servo valve degradation more easily. This functioning mode will be analyzed with all three methods, as being a function approximation, it can be performed by all of them.

The second aim of the study is to identify the degradation of the servo valve based on the flow and pressure values. This is done in order to implement a strategy aimed at identifying possible degradations that may lead to a system failure. This is known as Prognostic & Health Management (PHM). This discipline is important because of the increasing complexity and importance of aircrafts.

Companies can be more competitive by using components for their entire useful lives, up to the point that the component ceases to function. This allows maintenance scheduling to be optimized while not compromising the reliability and integrity of components. This problem is better performed by the neural network, as it is a pattern recognition problem, therefore it will not be analyzed with the other modeling methods.

The research will be conducted based on the servo valve flapper nozzle type, located upstream of a hydraulic actuator, as this component is more technologically sophisticated and hence more prone to fail, compromising the overall efficiency of the servo control.

Chapter 2

Servo Mechanism for Flight Control

The purpose of movable command surfaces is to allow the pilot to control and maneuver the aircraft from the cockpit. They can be used to alter the external shape of the wing surfaces in order to modify aerodynamic resultant forces and moments on the airplane. Flight commands can be divided into two categories: primary and secondary.

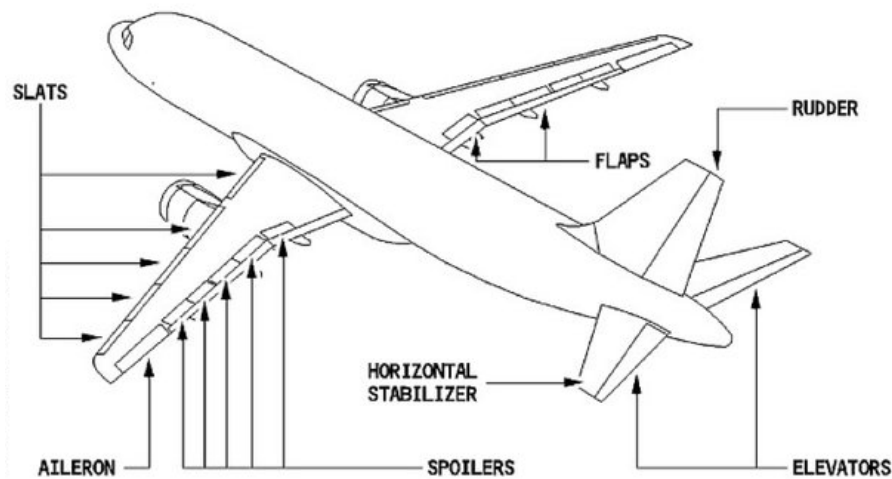


Figure 2.1: Primary and secondary flight controls

Examples of primary commands are ailerons, rudder and elevators. They

are continuously actuated during flight and must possess some basic characteristics. They must be instinctive, meaning that the direction of the command and the aircraft movement has to be consistent. Moreover, they must be proportional and reversible so that a control bars movement match to an accurate mobile surface deflection. They also has to be sensitive in order for the pilot to feel the resistance and stress on the mobile surfaces, in relation to their deflection and to the aircraft's velocity.

Secondary commands include leading edge extensions (slats), trailing edge extensions (flaps), tail plan and air brakes (spoilers). Their positioning is discrete and is usually modified by the pilot only during specific flight phases like the rolling phase, take-off, or landing. They must also be irreversible and have lower performance in terms of magnitude and actuation precision, due to their limited use.

Flight control actuation system is used to allow pilot to deflect mobile surfaces. It is used in large commercial aircrafts and is executed with hydraulic actuators due to their reliability, security, and robustness. Computer controls them by translating the pilot's analogical signal into a digital one. These actuators are not without disadvantages. They are heavy and complex, making it difficult to locate and repair a fault in the system.

It is becoming more difficult to reduce the weight of modern components because of the need to meet safety requirements. One solution is to use servo electro-hydrostatic actuators (EHA). These mechanisms allow for a greater compaction of components by using a hydraulic circuit that has smaller dimensions in comparison to a centralized one. A fixed displacement pump, an electric motor, and a servo-valve actuate the hydraulic actuator to convert electric energy into hydraulic energy. The lack of a hydraulic supply allows for space and weight savings. Another solution is to use electro-mechanical actuators (EMA), which are more reliable and stronger than the hydraulic ones. We will examine the main characteristics of EHA, EMA, and traditional hydro-mechanical actuators in the next paragraphs.

2.1 Hydro-Mechanical Actuators

The most common type of actuation system in the past was hydro-mechanical actuators. They consist of the control element, a valve and the actuator. A cinematic chain transmits the pilot's input to the valve, which operates by means of a three-centres lever. One of the levers is attached to the valve's

spool and allows it to move. The second lever connects to the control surface to provide mechanical feedback. Once commanded, the valve supply one chamber of the actuator and discharge the other. With this scheme, the valve is commanded with the geometrical difference between its commanded position and the surface position, in order to attain a stable displacement.

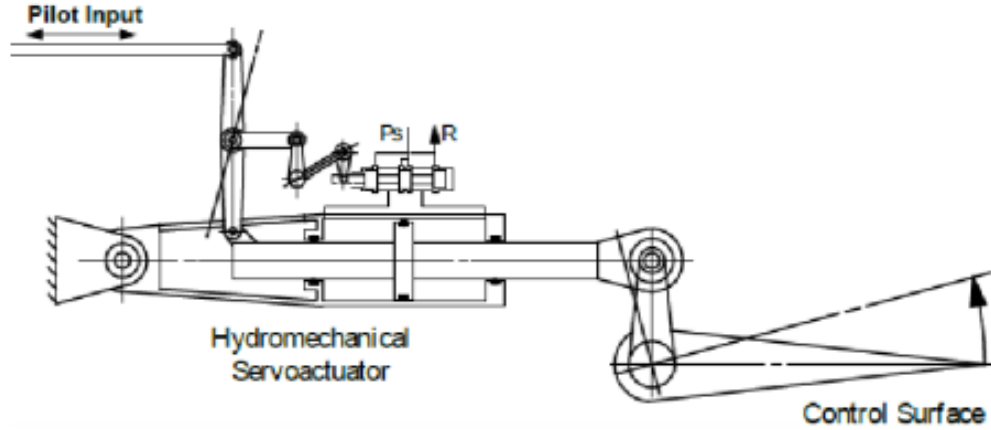


Figure 2.2: Scheme of a hydro-mechanical actuator

2.2 Electro-Mechanical Actuators

This machine converts electric power into mechanical power by using an electrical motor, a gear box, which reduces the RPMs and increases available torques, and a cinematic chain which converts rotatory motion into linear motion. This is usually done by using a ball screw to reduce friction. All-electric systems such as this are likely to be adopted by any aircraft in the near future. EMAs are used only as secondary commands due to a major downside: if a single grip occurs in the ball screw, this eventually prevents all the system from moving. This is frequently the result of a single grip in the ball screw.

2.3 Electro-Hydrostatic Actuators

This is the actuator that is considered in this thesis. This actuator uses a brushless motor to drive the pump. The pump is placed near the actuator. There are two types of pumps: one is a fixed displacement pump where the

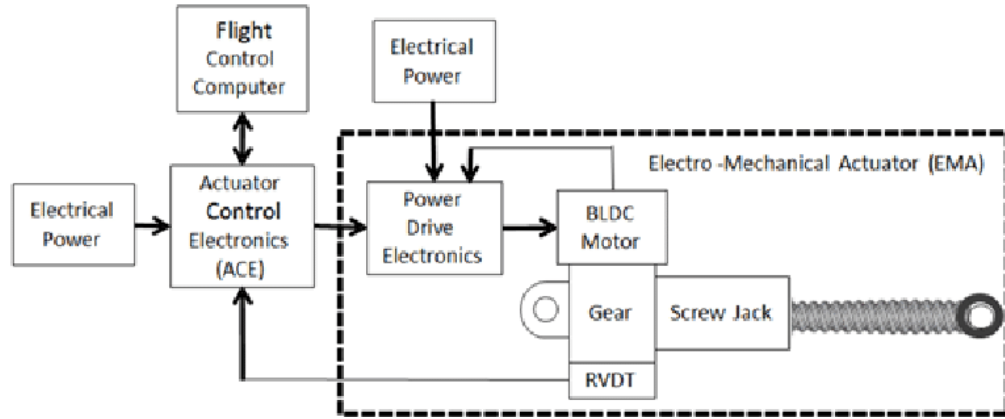


Figure 2.3: Scheme of an electro-mechanical actuator

speed of motor can be controlled, and the other is a variable displacement pump where regulation depends on the pump's plate. The actuator is close to the pump, which allows for weight savings and avoids heavy hydraulic lines. The actuator also has the advantage of transmitting pilot commands via fly-by wire lines. This avoids friction losses and linkage flexibility that are characteristic of reversible control cables. The feedback signal is received through either a LVDT (Linear Variable Differential Transformer) or RVDT (Rotary Variable Differential Transformer), which provide information about the instantaneous position to the ACE (Actuator Controller Electronics). The position error is then given to the PDE (Power Electronics), which regulates the motor's angular velocity.

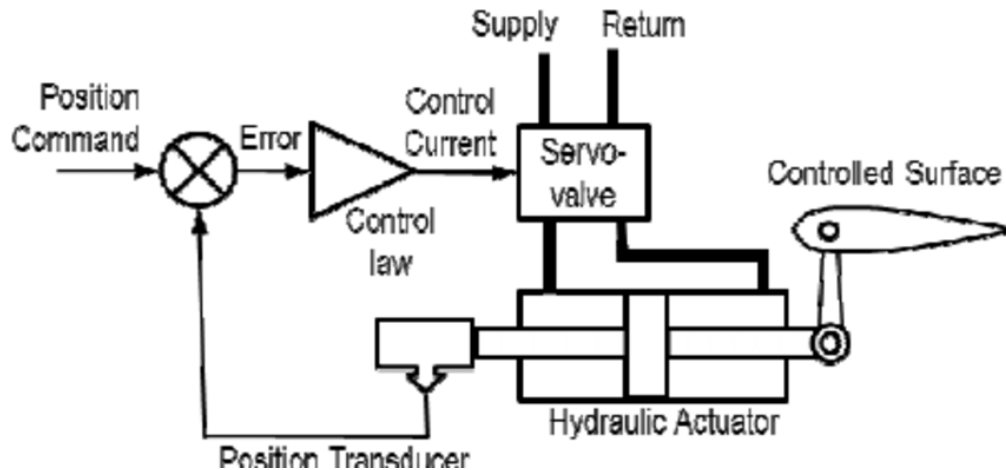


Figure 2.4: Scheme of an electro-hydrostatic actuator

This actuator is the subject of this thesis. It is driven by a brushless motor and has its pump located close to it. Pumps can be classified as fixed displacement pumps or variable displacement pumps. Fixed displacement pumps regulate speed by controlling the motor speed, while variable displacement pumps regulate speed based on the plate size. It is possible to reduce weight by placing the pump close to the actuator, hence avoiding heavy hydraulic lines. It also avoids friction losses and flexible links associated with reversible control lines because the pilot command is transmitted via fly-by-wire lines. A LVDT or a RVDT provides feedback about the instantaneous position of the jack to the ACE. By this method, the Power Drive Electronics (PDE) that controls the motor's angular velocity is provided with the position error.

Chapter 3

Servo-Valve for position command

A servo-valve's function is to control the flow rate of fluid entering the actuator. This is done to allow the control of the actuator's position and to any forces caused by the aerodynamic hinge moment to be equilibrated.

Since the second half of the 20th century, servo-valves have been extensively used in aviation, both for primary and secondary control.

The servo-valve is activated by an electric input signal from the control unit, which in this thesis is implemented as a PID controller, and transferred to its torque motor. The system uses the difference between the commanded and instantaneous positions of the jack, which is determined by a closed-loop structure, to cancel out the position error and return to the commanded position.

The next paragraph will describe the flapper-nozzle valve used in this thesis.

3.1 Flapper-Nozzle Valve

A flapper-nozzle valve is divided into two stages. The first stage is made of an electric torque motor, two flappers, and two nozzles. The second stage comprehends the sleeves, where the spool can be moved to open or close the four passageways. The valve can be controlled by the spool. Its movement is dependent on the differential pressure created by the nozzles.

According to the design of its sleeve it opens and closes two passageways:

one to supply a chamber for the jack, and one to discharge the other. The feedback-spring represents the feedback between these two stages.

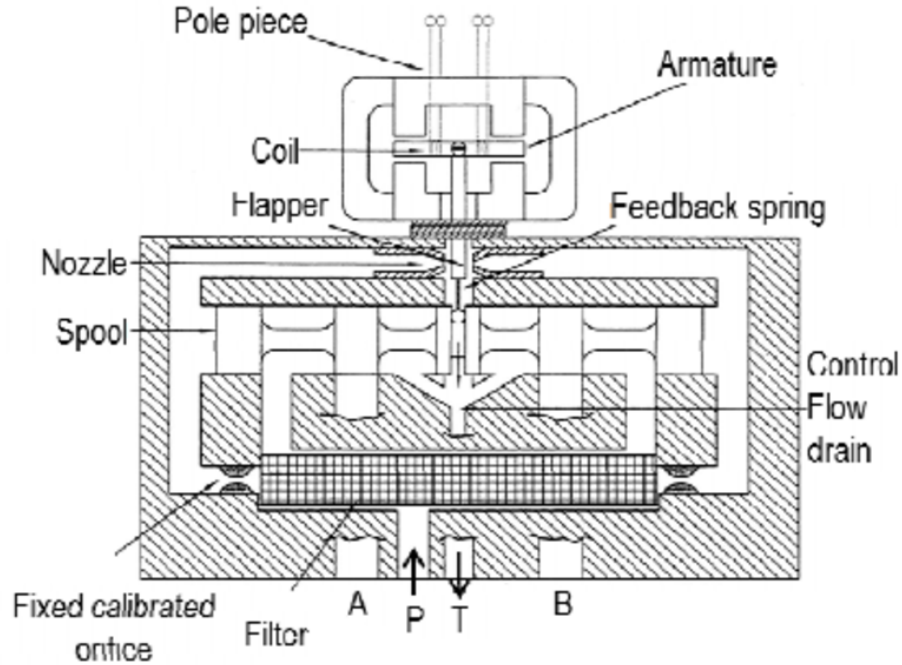


Figure 3.1: Flapper-Nozzle Valve

The stator in the torque-motor is composed of an upper and lower yoke, with a permanent magnet fixed on each of them. The rotor is composed of an armature with two coils, where the middle position matches with null current. Between the four airgaps, the magnetomotive force generated by the coils and permanent magnets is shared between the coils and the coils.

The flapper's upper portion is fixed in the middle of the armature, and its lower part is placed between two equally spaced nozzles. This means that any rotation of the rotating rotor will result in a rotation by the flapper that blocks one of the respective nozzles. This rotation is distinguished by the hinge spring as well as the bending feedback spring. The armature-flapper assembly, however, is supported by a flexible tube that prevents hydraulic fluid filters from the torque motor.

The flapper can be moved to the right or left, causing partial occlusion of the one nozzle. This will cause the fluid coming from one nozzle to be throttled and the other one to be unconstrained. A partial occlusion will result in an increase in the pressure reported in the chamber. The spool

will move in one direction due to differential pressure between the surfaces. Two passageways will be open to allow the spool to move relative to the null position. One will supply a chamber for the actuator, the other will connect to the tank. The hydrostatic force of differential pressure on the spool is not the only force that acts on it. It is also subject to feedback spring as shown in Figure 3.1.

The feedback spring acts as a force to cancel out the position error between the flapper (spool) and the flapper. This is an important contribution to stability: if the valve is commanded with a step signal input (e.g., a step signal), the flapper will block one of the nozzles, causing differential pressure on the chamber. Once the jack has reached the new command position, the error tends towards zero.

The feedback spring applies a force to report that the spool is in the null position. This is the unique position that has null error. The feedback spring would not act on the spool and the jack would oscillate in the equilibrium position: this would cause a limit cycle.

Fluid contamination is a problem. Small metal parts could cause problems with the valve's functioning. These metal parts can get to the nozzles, and can cause damage due to supersonic velocities. They can also reach the ball at either end of the feedback spring or the spool, which can reduce the precision of your valve.

A filter is installed after the return and supply passageways. This valve is the most popular in commerce due to its high reliability and dynamic performance in terms reaction velocity.

3.2 Technical characteristics of valves and non-linearities

As mentioned before, valves and servovalves are used to control the fluid in the chambers of an actuator. The control is achieved by the position of the spool which closes or opens the passageways. The actuator's flow rate and pressure depend on the external conditions (load, pressures of supply/return). A useful way to explain all this is through the characteristic curves of a valve.

Figure 5.3 shows the characteristic curves of pressure $P_{12} - X_S$. Given a fixed flow rate Q_J (in our first example equal to $-6 \text{ m}^3/\text{s}$, $-3 \text{ m}^3/\text{s}$, $0 \text{ m}^3/\text{s}$, $3 \text{ m}^3/\text{s}$, $6 \text{ m}^3/\text{s}$), the correspondence of input X_S with output P_{12} can be explained.

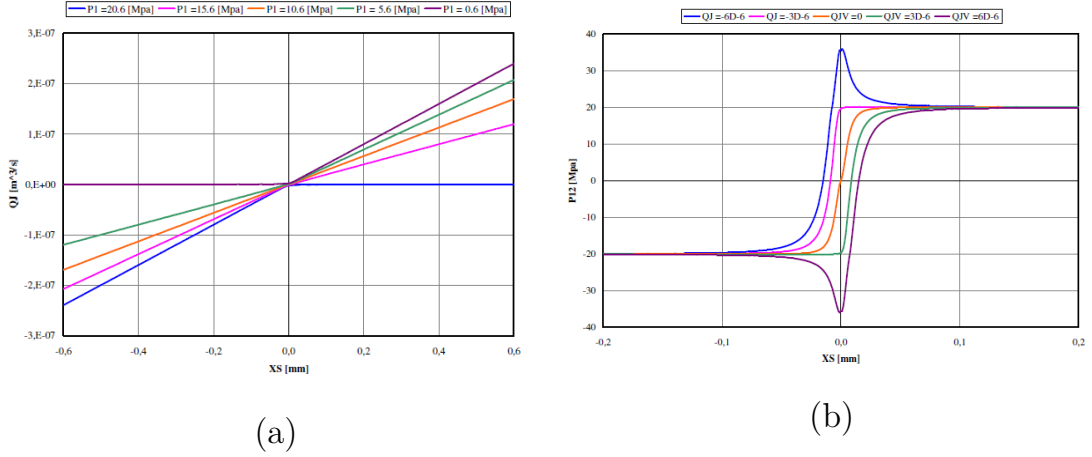


Figure 3.2: Valve characteristic curves

The control pressure is also a function of the valve's geometric characteristics, supply pressures and return pressures. It is striking to see that the pressure $P12$ can be almost equal to the supply tension for large spool displacements XS . However, it can take pressures greater than the supply pressure for smaller values of XS . This is typical for rapid closing of the spool where fluid velocity is maintained while the passageway is shut off, leading to water hammer.

Figure 5.3.a shows the characteristic curves flowrate $QJ - XS$. Given a fixed pressure $P12$ (equal to 0.6 MPa, 5.6 MPa, 10.6 MPa, 15.6 MPa, 20.6 MPa) correspondence of the input XS to the output QJ can be seen. Two different horizontal segments are visible in this example: the first shows the flow rate at $P1=0.6$ MPa and the second, $XS=20.6$ MPa. These correspond to situations where the pressure $P1$ equals the return pressure in either the first or second case. This results in a null flow rate since there is no differential pressure. Some simplifications can be useful: because this study is focused upon the proximity of the null pool position, it's possible to apply superposition effect to XS , taking into account two different contributions to the pressure and flowrate. These simplifications were made in consideration of the flow rate gain GQ and the pressure increase GP . The slope of $QJ-XS$ at pressures $P12$ is the flow rate gain, and the slope $P12-XS$ is the pressure gain at flow rates QJ .

Although this can provide a powerful way to analyze the performance of a valve during its initial phase, it may also be necessary to examine other

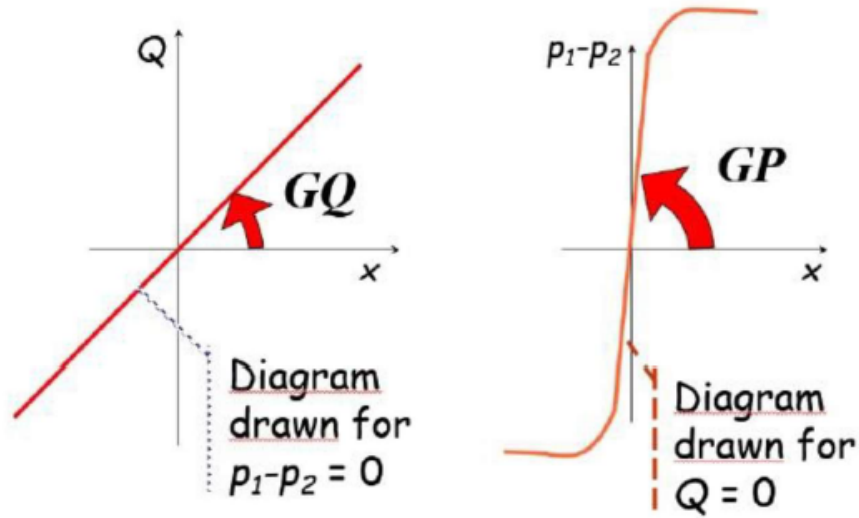


Figure 3.3: Flow rate and pressure gains

aspects of the valve. The main features of the valve are explained in the following paragraphs.

3.2.1 Lapp condition

A valve can make three different types of constructive choices depending on the lapp condition. Zero-lapped is a spool with the same dimensions as the passageway. In this case, there is only one position that closes the passageway completely. A displacement results in an immediate response from the valve in terms pressure and flow rate.

The purpose is to avoid non-linearities within the null location neighbourhood. In this manner, the pressure and flow rate gains are nearly constant in this area. An overlapped valve refers to a condition in which the spool is wider than the passageway. This is used to compensate for static leakage caused by clearance and creates a dead area where an axial displacement does not cause any flow rate. Instead, an underlapped valve is one where the spool is smaller than the passageway. This means that even at null, the passageway may not be completely closed.

The purpose of this is to prevent water hammer from rapid closing the valve. The valve will respond to pressure and flow rate even when the position is null. Below is a diagram showing the responses to inputs of each of the three conditions.

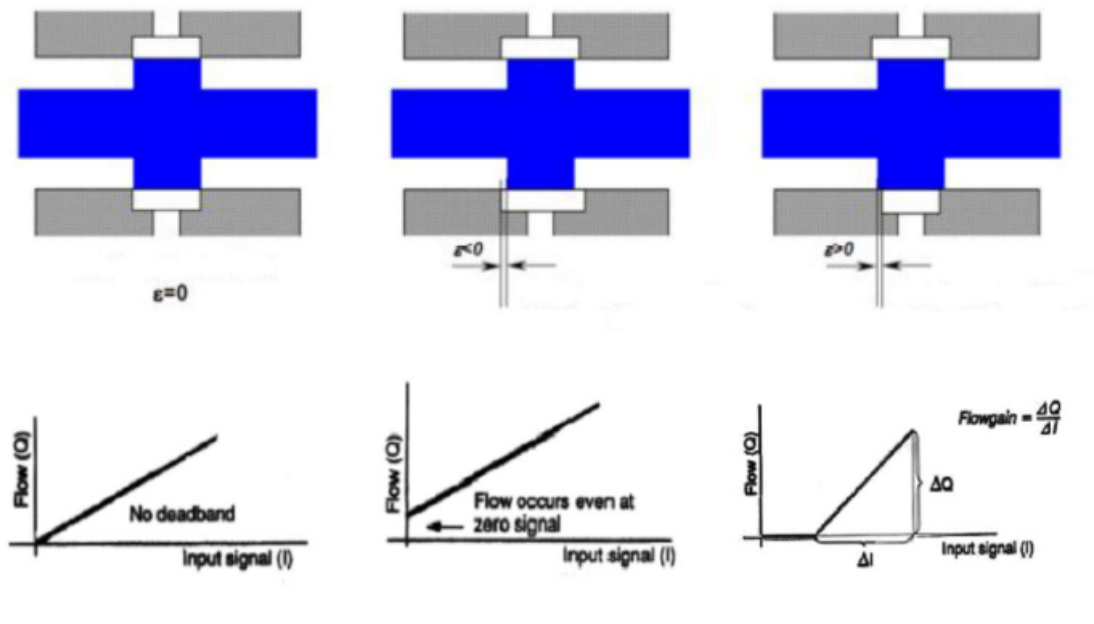


Figure 3.4: Lapp conditions and their responses

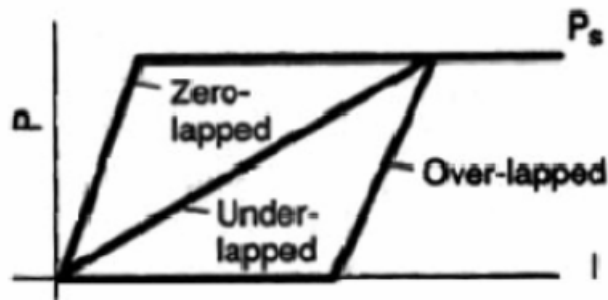


Figure 3.5: Lapp condition scheme

3.2.2 Threshold

Threshold can be defined as the minimum input current the valve needs to change the fluid flow rate during directional inversion. It can also be described as the minimal variation in current that causes a change in valve response. It is the threshold of how sensitive a valve to changes in input current.

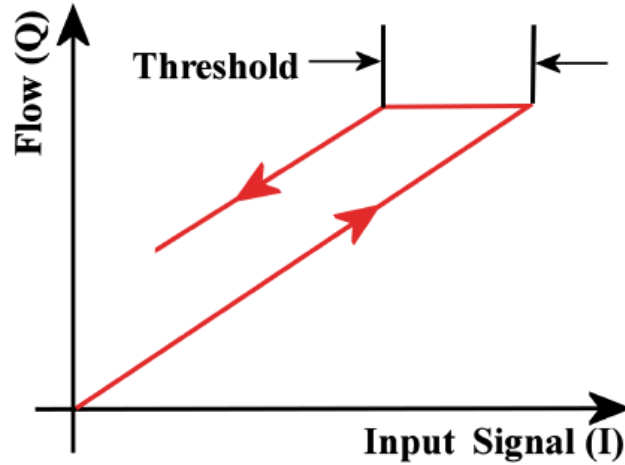


Figure 3.6: Threshold

3.2.3 Magnetic Hysteresis

Magnetic hysteresis, which is an important source for non-linearities, is defined as the ability of a system react to both instantaneous input values and to the time history of states. This is a characteristic of magnetic materials, and obviously all of their applications (transformers and electric motors, etc.). The graph below shows the typical response to a system with hypersteresis.

Initial magnetization of the system follows the first curve. However, when the current is increased, the magnetization does not increase in proportionally.

The magnetization B will attain saturation when the current I exceeds a certain level. This is because the Weiss domains in the material slowly align themselves to the magnetic field. As a result, there will be an increase in magnetization, until all are oriented in the same way and then the saturation.

If the current is reduced to zero, it is possible to see that the curve is not the same as the first. In particular, B won't be equal to 0, but will assume a positive value. During this phase, some domains of the Weiss will retain their previous orientation and thus, the residual magnetization of material will occur.

A current of the opposite sign is required to demagnetize the materials. If the current is applied for longer periods, the material will become magnetized in the opposite direction and reach a saturation value that is different from the first. The curve will close by decreasing the current to zero and increasing

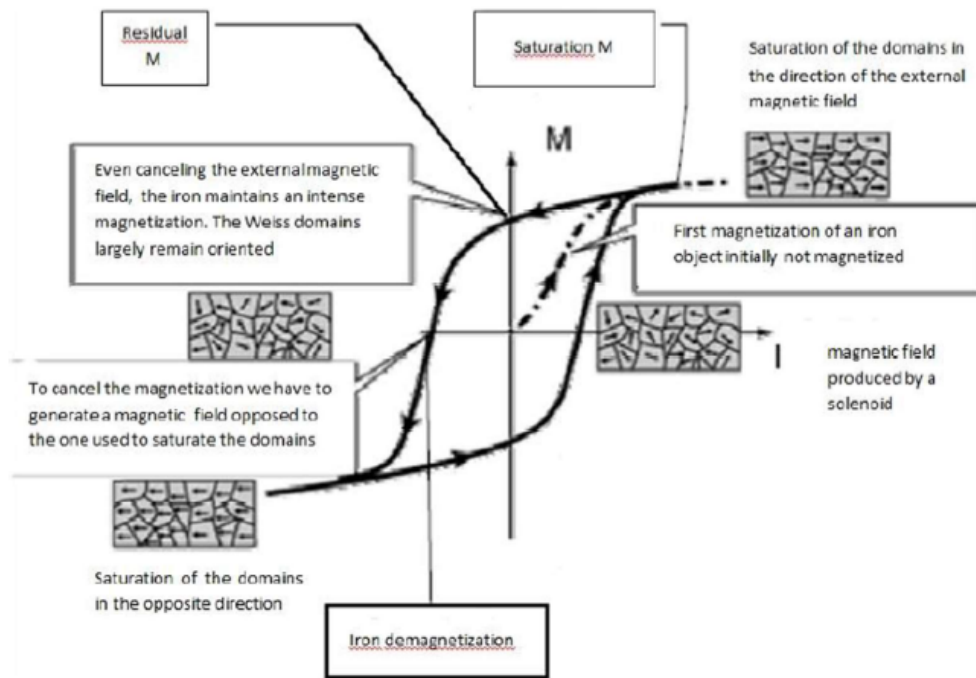


Figure 3.7: Magnetic hysteresis

it in opposite direction. This is shown in Figure 3.7.

The diagram of Hysteresis can look quite different depending on the material being analysed. Hard ferromagnetic substances usually have a higher residual magnetization than the soft and have therefore different applications.

All ferromagnetic materials may lose their magnetization through natural demagnetization (ductile iron case), or heat (recovery case of permanent magnets). The magnetic hysteresis is what affects the torque motor at the first stage. This happens even though the current is not applied anymore.

The thesis examines magnetic hysteresis in the system. It affects the torque motor at the first stage. Even though the current is stopped, the magnetic field continues its attraction to the armature.

3.2.4 Friction

Because friction is an important aspect of system analysis, it has a significant impact on the entire system. It is a strong nonlinear phenomenon that limits position accuracy in the aeronautical sector.

Friction is the force between two surfaces, which opposes their relative motion. Static friction will only be present if the surfaces have a low relative velocity. In all other cases dynamic friction will be present.

There are three types of friction. Sliding friction is when two surfaces slip. Rolling friction is when a body rolls on a surface. Viscous friction is when a body, or a fluid layer, has a relative velocity to the layer below. The static and dynamic friction coefficients can be compared using the following expressions.

$$F_s = \mu_s * N$$

$$F_d = \mu_d * N$$

Depending on the contact materials and n the normal ground-body reaction, μ_1 and μ_2 are the respective dimensionless static and dynamic friction coefficients. The following behaviour will be observed if a body comes in contact with a non-lubricated surface.

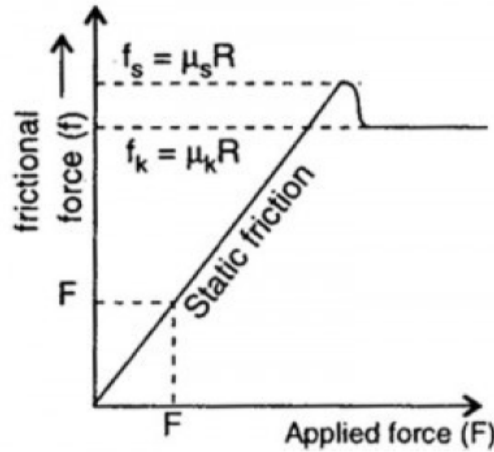


Figure 3.8: Behaviour of the friction force

Figure 3.8 shows that when a force less than friction applies to the body, it will remain in quiet. However, once the static friction reaches its maximum value, the body will start moving, and the friction will decrease to a nearly constant value equal to the dynamic friction.

To have a good friction model in this instance, you must distinguish four situations: Body that is still in quiet, body that moves, body that is in motion and body that stops moving.

The Coulomb model is one of the first friction models. It ignores local dynamic phenomena and provides a good global representation. It can be described using the following equations:

$$FF = \begin{cases} F_{att} & \text{if } v = 0 \wedge |F_{att}| \leq FSJ \\ FSJ * \text{sign}(F_{att}) & \text{if } v = 0 \wedge |F_{att}| > FSJ \\ FDJ * \text{sign}(v) & \text{if } v \neq 0 \end{cases}$$

This mathematical description shows every situation that has been identified before.

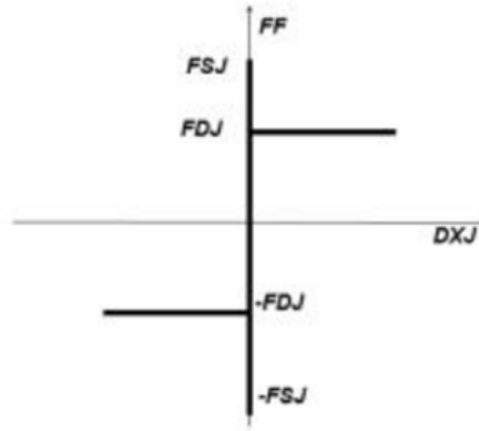


Figure 3.9: Coulomb friction model

Conventionally, the friction force (FF) is positive when it opposes a positive velocity. This model is not suitable to numerical simulations because of the discontinuity $v = 0$.

When the accelerations in these two cases are merged, two different velocity values of opposite sign are produced. The model cannot recognize when the speed reaches zero, and therefore, it cannot verify that the body has been arrested. This is important because, in quiet conditions, the active force generated by the body can be less than static friction.

These limits lead to the development and refinement of friction models. The Karnopp model corrects for the discontinuity in friction force due to null velocity by introducing a deadband at the origin. This dead band allows you to discern the cinematic differences between dynamic and static friction.

In particular, static friction can equilibrate external forces, keeping the

null value for acceleration, and thus avoiding the beginning. Below is the mathematical formula for the model:

$$FF = \begin{cases} \min(\max(-FSJ, F), FSJ) & \text{if } |v| \leq \varepsilon \\ v = 0 & \text{if } |v| \leq \varepsilon \\ FDJ * \text{sign}(v) & \text{if } |v| > \varepsilon \end{cases}$$

F is the driving force, e is the velocity at which the velocity is equal to 0. It is important to select the right value for e . If it is too high, simulation results will not be comparable to real data. On the other hand, if e is too low, the model cannot distinguish between static and dynamic states.

A value of e that is too high can cause calculations to be stuck in a cycle. Conversely, a value that is too low can lead to numerical instability. a graphic representation of Karnopp's friction model is shown in Figure 3.10.

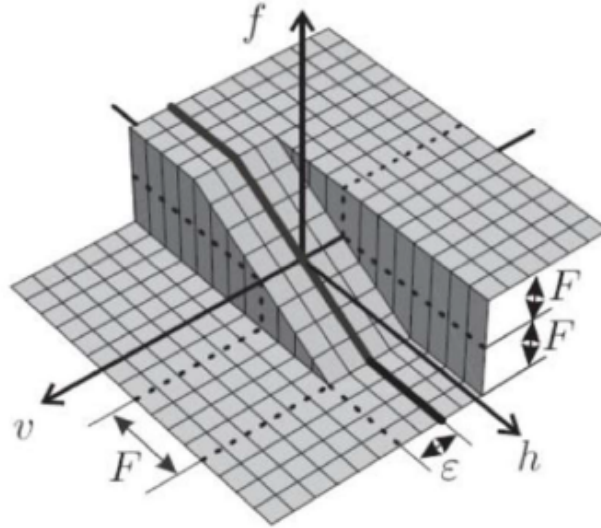


Figure 3.10: Karnopp model

The Simulink model can be derived from the equations that were previously discussed:

Professor of Politecnico di Torino Borello has developed a new model of friction. It surpasses the limitations of the Karnopp model. It can correctly distinguish the static from the adherent condition, and it can also discriminate the four cinematic scenarios described at the start of this paragraph. The mathematical and graphic representations of this model are described below:

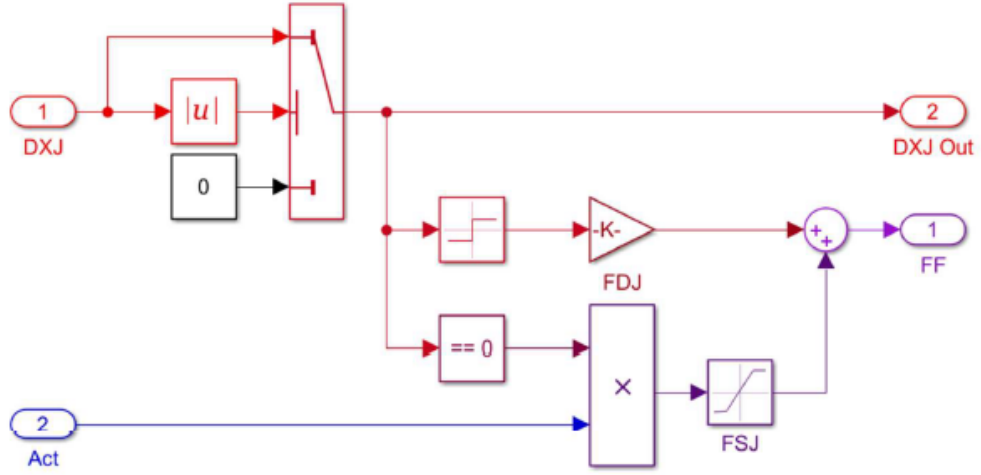


Figure 3.11: Simulink of Karnopp model

$$FF = \begin{cases} F_{att} & \text{if } v = 0 \wedge |F_{att}| \leq FSJ \\ FSJ * \text{sign}(F_{att}) & \text{if } v = 0 \wedge |F_{att}| > FSJ \\ FDJ * \text{sign}(v) & \text{if } v \neq 0 \end{cases}$$

This model does not require a dead band. In fact, if there is a change in velocity in an integration interval, the algorithm will reset the velocity, arresting it. If an external force is present but the body wouldn't arrest, at the next integration step, a decompensation between the forces is created, allowing the system to move again. These features make the model perform better than the literature.

Simulink is easy to develop, and you can obtain the next scheme.

3.2.5 Backlash

Backlash refers to the distance between two mechanical components that allows them to move relative. It is the distance or angle that a mechanical component can move at a given speed without transmitting any force or couple to another component.

A time delay in the system can occur due to inversion of the input. This is one of the key factors that affects the system's performance. Because every manufacturing process results in tolerances in dimensions, it is an inevitable

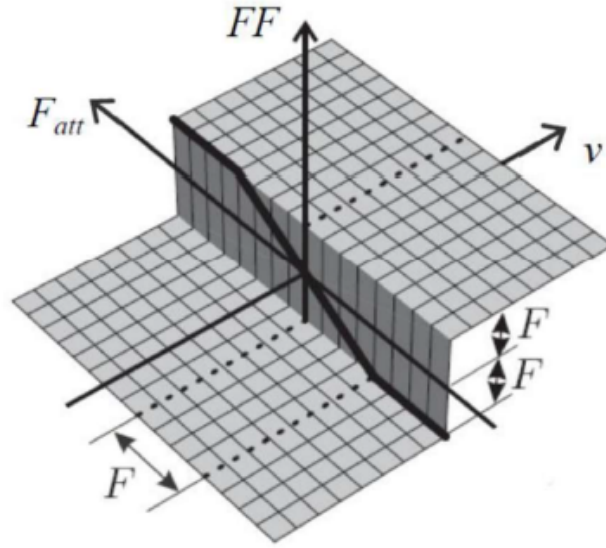


Figure 3.12: Borello friction model

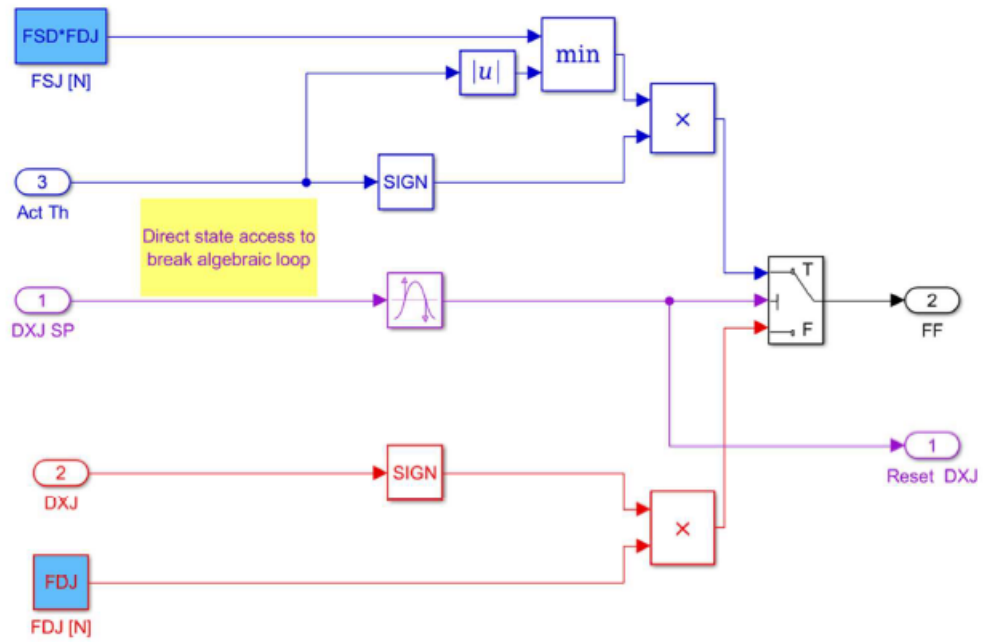


Figure 3.13: Simulink of Borello friction model

consequence. It is often present to allow coupling of multiple components or to compensate for the thermal expansion.

Backlash can also increase over the lifetime of a component due to continuous contact with other parts. This can lead to further degradation of system performance and a decrease in precision when positioning components.

3.2.6 Offset valve

An offset is usually a measurement of error relative to a reference value. The servovalve can cause damage to the electric motor even if it is not in use.

Even though the motor is not being deenergized, it can still receive minimal input current. This current can partially energize a torque motor, and it can open partially passageways of servovalve. It can also actuate the jack even though no input is available.

Chapter 4

Simulink Model

To get an estimate of the performance of each component, aeronautical projects were traditionally developed independently. Each component's interactions were tested on the ground and then flown. This ensured that any errors in the project phase would be costly in both time and money.

Aeronautical sectors, like the automotive, require instruments to demonstrate new solutions and improve the applications of existing components. This is why modern computers that are powerful and reliable can be a significant support to the design by simulating the system's interest behavior. The new design approach is to solve all problems before flying testing. This can be done through numerical simulations. This allows for the development of more precise models that can predict system dynamics and interactions. Models are also very useful in diagnostic and prognostic areas.

For example, it is possible to predict the useful life of a component by using a model-based approach. This allows you to compare the time response from sensors with that from models. A fault signal is sent to pilots when the differences exceed a security threshold. Once landed, the same fault signal can also be sent to maintenance. This will allow them to identify the location and entity of faults and eventually replace the damaged component. This allows components to be used for their full useful life without having to replace the entire mechanical assembly. It also saves time and money.

A number of steps are required to create a satisfying model. This is the first phase. It identifies all aspects, unique characteristics, and components of the system. It could be beneficial to simplify or approximate certain parts of the model due to their complexity. However, this will not affect its validity. It is important to find the right balance between usefulness and

complexity. Formulation of equations. After the mathematical model is developed, it's necessary to apply all laws and introduce the material and geometrical properties of the system. The result will be an independent set of differential equations that describes the system in all its major aspects (dynamic, chemical, thermal, and so forth). The equations must be solved.

It is important to choose carefully two aspects: The first is the order in which the resolution numerical method is used (ODE). A higher order may result in more precise simulation results, but it requires greater computational effort. The second is the integration time. This interval must not exceed the smallest time constant of a system, otherwise it could lead to numerical instability. A model must be validated by comparing its results with experimental data. If there is any inconsistency, the model must be revised.

Models can be used to save time and money but cannot replace experimental tests. Two models of the same EHA can be found in this thesis. The first model, High-Fidelity, was developed by Ing. It is a very accurate model and is used to reference the Low-Fidelity model. Although it does use some approximations it is faster and requires less calculation effort. This is why it is used in optimization algorithms (explained later). The MATLAB-Simulink 2016b version was used to develop the models.

4.1 High-Fidelity Model

This paragraph will describe the entire High-Fidelity System and its operation.

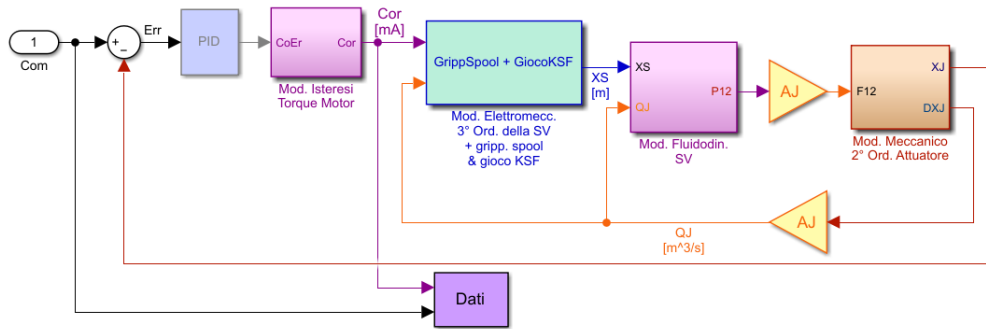


Figure 4.1: Model representation in Simulink of the High-Fidelity model

The system can be represented as a closed loop, with the variable XJ

representing the instantaneous position and the sine wave being feedbacked to calculate the error signal entered by the PID controller.

The command block is the first block. It is possible to provide different time domain reference signals to the system.

Five inputs are available: three standard signals (step and ramp), and two custom signals (chirp, a sinewave with variable frequency and Com signal consisting of different subsequent steps signals with variable amplitude).

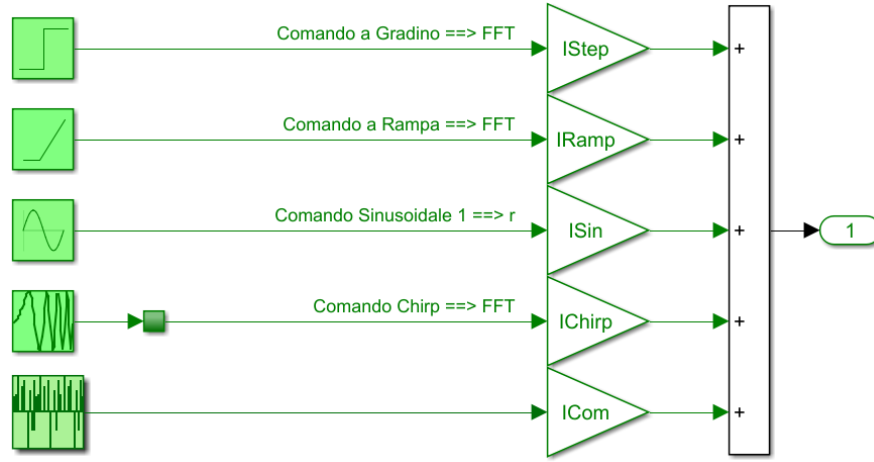


Figure 4.2: Input block sub-system

A PID controller is responsible for determining the error between the instantaneous jack position and the selected command. Its purpose is to stabilize the system's dynamic response. The PID controller consists of three contributions: integrational gain, proportional gain and derivative gain.

The proportional gain is treated as a non-null value in this instance. Therefore, the PID controller can only be used as a proportional controller due to the system's already granted stability. The error signal then feeds the hysteresis bloc of the torque motor. This block allows for both an offset on current input and disturbances due to external factors. The actual current supplied to the valve is represented by the output of the hysteresis bloc. The flapper-nozzles dynamic system is represented as a third-order system. It is made up of two sub-blocks, a second-order dynamic system for stage one and a first-order dynamic system for stage 2. The second stage also receives the flow rate QJ to calculate the flapper force. The output XS represents the spool position. However, it is also used to determine the differential pressure P12. To calculate the force that supplies the actuator,

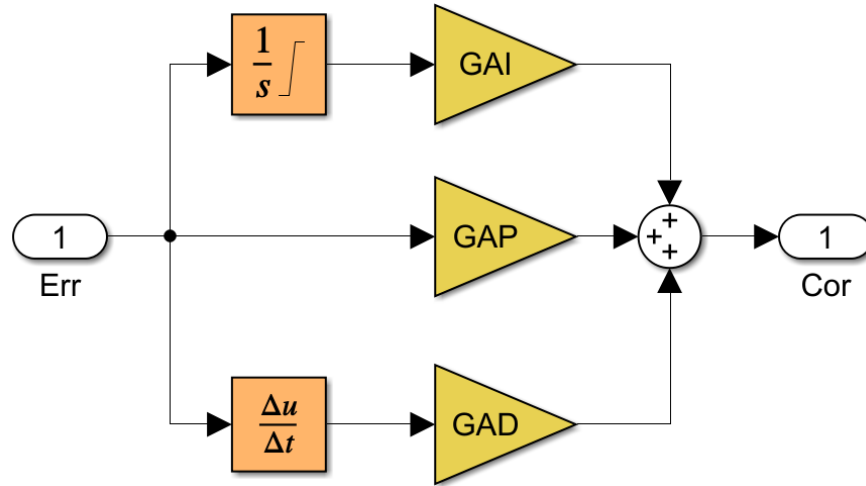


Figure 4.3: PID controller's scheme

the differential pressure is multiplied with the jack's area. Then it enters the last block, which is the second-order mechanical model of actuator. The second principle of dynamic calculates the position and velocity of the Jack, taking into account both external and aerodynamic forces as well as friction and displacement limits.

4.1.1 Hysteresis, offset and interferences in the torque motor

The block's purpose is to show disturbances and nonlinearity in the current entering the model of the torque motor. It is fed the control signal CoEr from the controller. This signal is then algebraically added to the values of His or Ofs. The signal will change according to its value. If it is rising, it will decrease His. However, if it falls, it will increase His. This is to indicate the hysteresis.

Instead, the block Ofs (shown in Figure. 4.5) allows you to set an offset value for the current. However, it is possible to also introduce an external source to interfere with the motor. This source is represented by a Band-Limited white noise block. Its contribution is a random signal of constant power spectral density. The resulting current will be saturated, if required, to the maximum and minimum values CorM or -CorM. These are the limits of current that can be fed the servovalve.

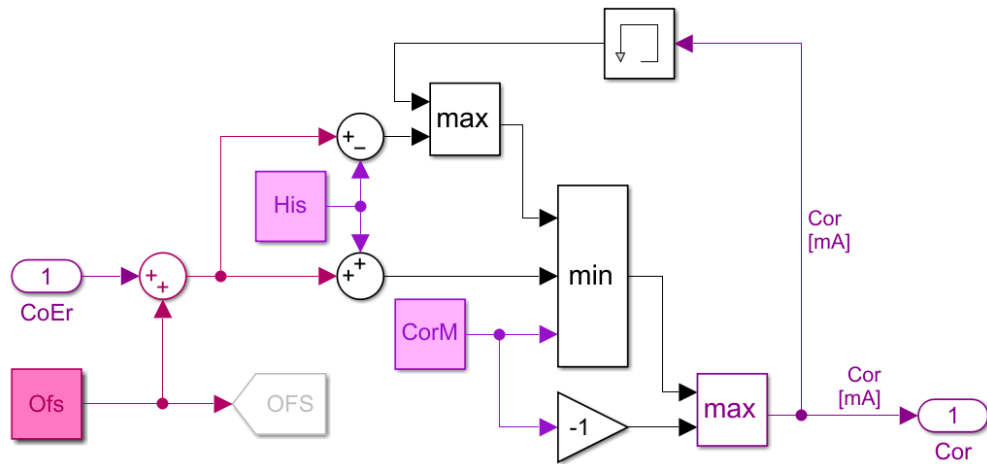


Figure 4.4: Model of torque motor hysteresis

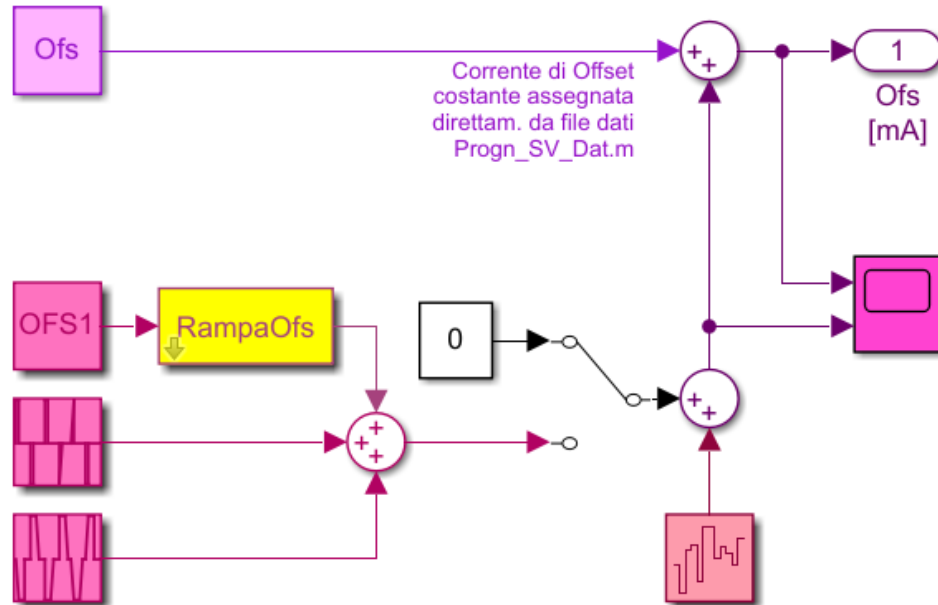


Figure 4.5: Offset current model

4.1.2 Electromechanical model of the servovalve

This block is the third-order electromechanical model for the servovalve. It is made up of two sub-blocks that describe the dynamic behaviour of each stage. Figure 4.6 shows that the signal *Cor* is sent from the block *GM* to

calculate the gain of torque motor (in (Nm/mA)/m).

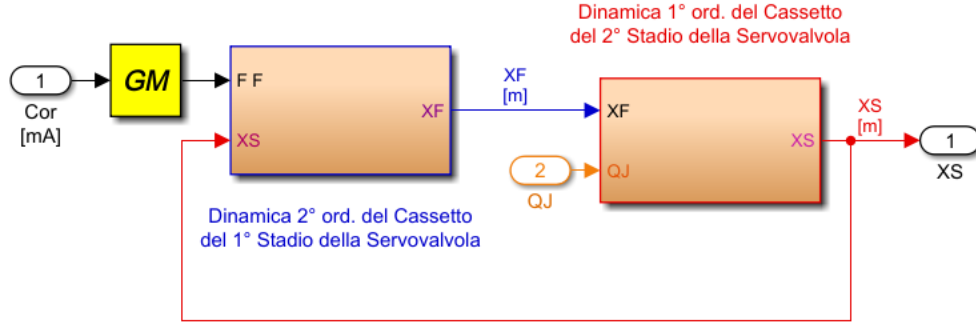


Figure 4.6: Servovalve's electromechanical model

There are two possible values for GM: The first is the fixed value that has already been calculated. The second considers the structure, configuration, and magnetic circuits of a motor. This corresponds to the next picture. The second branch of the calculation requires two inputs: Cor is the value of current from the previous block and XF is the position of the flapper that was feedbacked in the second order dynamical model. First, XF is multiplied to obtain the gain $(L_a/2)/l_s$. Here L_a is the length of an anchor and l_s is the arm of a spool. This calculates the rotation of the arms and multiplies it by $L_a/2$ to get the vertical displacement. These values are used for calculating the magnet flux in all airgaps.

$$\phi_1 = \frac{V_p}{2R_0} \frac{1 - \frac{XF}{lg}}{\lambda + \alpha * r} + \frac{Cor * n}{2R_0}$$

$$\phi_2 = \frac{V_p}{2R_0} \frac{1 + \frac{XF}{lg}}{\lambda + \alpha * r} - \frac{Cor * n}{2R_0}$$

Where:

n = winding's number in the armature

R_0 = airgap's reluctance in the null current condition

V_p = residual magnetic induction

lg = air-gap's total length

λ, α, r = correction factors considering edge effects of magnetic flux in armature's extremity

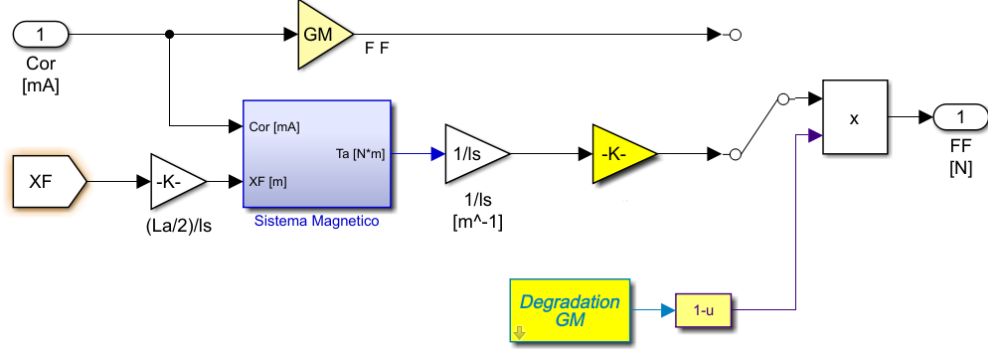


Figure 4.7: Gain GM calculation

The formulas above show that there are two components to magnetic flux. The first is the magnetic flux generated by the permanent magnets in the yokes with resistance:

$$R = \frac{2R_0}{1 \pm \frac{XF}{l_g}}$$

The second term, with an opposite sign, represents the flux created in coils.

The formula can be used to calculate the total torque once fluxes through airgaps have been calculated:

$$T = \frac{(\phi_1^2 - \phi_2^2) * La}{2Ag\mu_0}$$

Where:

La = anchor's length

Ag = airgap's section

μ_0 = vacuum's magnetic permeability

In order to determine the force applied to the flapper center, the torque is divided by the anchor length; a constant in time equal to GM is also added as another gain in order to obtain a stable model.

For the simulation of the behavior of the system in the presence of demagnetization of the torque motor, the force is multiplied by the value $1 - KGM$, where KGM is between 0 (normal condition of the torque motor) and 1 (completely demagnetized motor).

A model of the first stage of the servovalve will be analyzed in the next block.

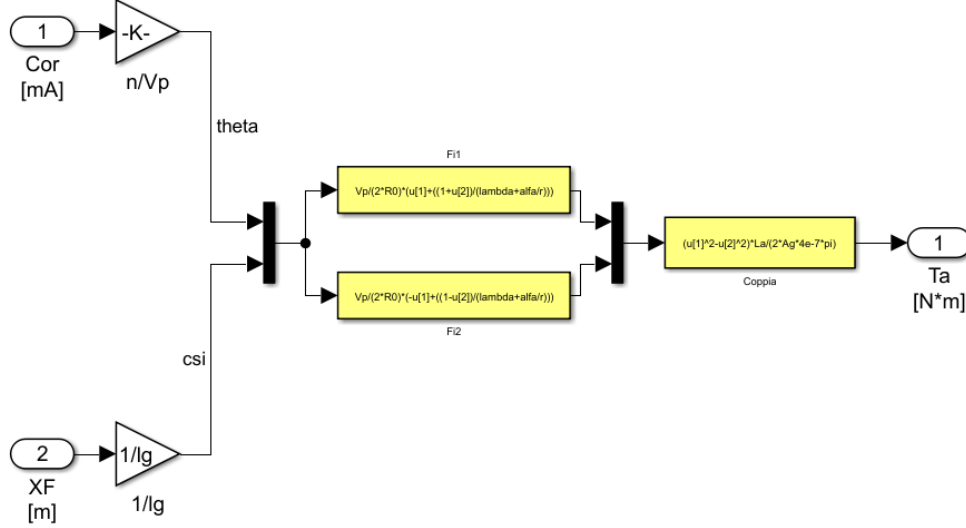


Figure 4.8: Torque motor's magnet circuit

The second order dynamic block contains the force applied to the flapper. However, this input is not the only one used. Also, the variable XS , which is the instantaneous position for the spool is feedbacked from second stage dynamic model.

This feedback has a physical significance because it is the feedback provided by the feedback spring, which, proportionally to its stiffness, reports a force to flapper. The bending effect can be simulated by adding the spool position to that of the flapper. This is done due to the opposing reference systems. Multiplying this result for elastic constant yields the resultant force from the feedback spring.

A dead zone dynamic mask is also available to simulate the backlash between spool, ball. K_{gioco} , which is expressed in microns, can be used to simulate the degradation of this backlash. It can also be imitated from the associated data file. The intermediate closed loop branch is used to calculate the elastic return of the flapper by multiplying the position for KF , which

is the stiffness armature bolt.

The inner closed loop branch calculates the viscous reaction fluid has to flapper displacement. It is proportional to flapper velocity and the proportionality constant CF (in N/m/s), which is derived from the formula:

$$CF = 2 * ZF * \sqrt{MF * (KSF + KF)}$$

Where:

ZF = viscous coefficient (dimensionless)

MF = flapper's equivalent mass

KSF = feedback spring's translational stiffness linear term (between first and second stage of the seroalve)

KF = stiffness armature-bolt

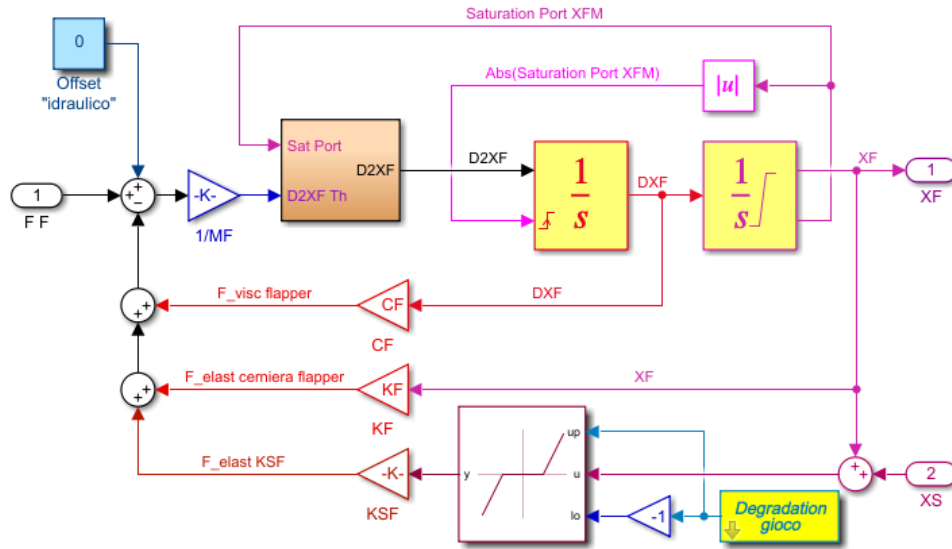


Figure 4.9: Servoalve's first stage dynamical model

To calculate the net force acting upon the flapper, these three forces are subtracted from FF . Divide it for the equivalent mass MF of the spool, and its acceleration is calculated. Two successively integrated calculations also determine the velocity and position of the spindle. The model is made

stronger by adding saturation. If the flapper reaches its limit position $+XFM$ or $-XFM$, then the displacement integrator switches on to $+1$ or -1 the saturation port. This value is then passed successively to an acceleration reset block to cancel any acceleration or velocity. You can also impose hydraulic offsets to shift the work point in another equilibrium state. It is a first-order dynamic model that represent the servovalve's second stage.

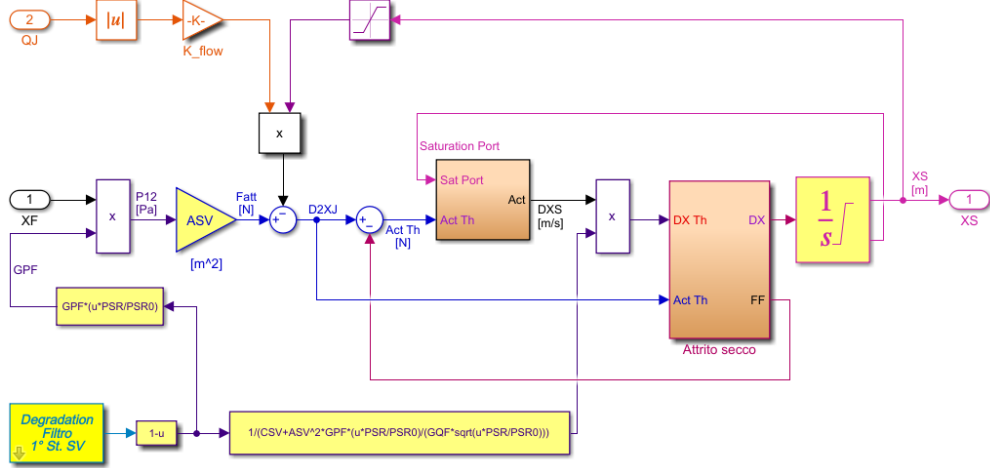


Figure 4.10: Servovalve's first stage dynamical model

The flapper and nozzle interaction have been simplified assuming that the valve behaves as a simple valve, in accordance with its own flow rate and pressure.

In terms of the dynamical behavior of the valve, the hydraulic force acting on the spool can be expressed as follows:

$$F_{idr} = (P_1 - P_2) * ASV$$

where ASV is the lateral area impacted by differential pressure.

A second order differential equation can be written considering all the forces acting on the spool:

$$M_{SV}\ddot{x}_S + C_S\dot{x}_S + K_{SV}(x_S - x_F) = F_{idr} - F_{frict} - F_{fl}$$

If we substitute the hydraulic force and the flow force, and neglect elastic force, we obtain the equation below:

$$M_{SV}\ddot{x}_S + C_S\dot{x}_S = ASV * P_{12} - F_{frict} - K_{flow} * QJ$$

Representing a differential equation of second order characterized by a time constant too small.

It's not possible to neglect the inertial effects in relation to the other effects, so an integration interval smaller than the smallest time constant of the model must be used, resulting in a longer calculation time.

In order to speed up the algorithm, the mass of the spool is neglected, resulting in a first order system ($P12$ is calculated by combining pressure gain and flow gain expression, and considering $QJ = \dot{x}_s * ASV$)

$$C_s \dot{x}_s = ASV * \left(x_F - \frac{\dot{x}_s * ASV}{GQF} \right) - F_{fric} - K_{flow} * QJ$$

and, if all terms are collected in \dot{x}_s :

$$\dot{x}_s * \left(C_s + ASV^2 \frac{GPF}{GQF} \right) = ASV * GPF * x_F - F_{fric} - K_{flow} * QJ$$

Borello's friction model uses the theoretical velocity of the spool as input: in this block friction force is the output, which is feedbacked and subtracted from theoretical force acting on flapper and actual flapper velocity. The position of the flapper is then obtained through integration, considering the limit positions $+XSM$ and $-XSM$.

In these calculations, degradation of the filter plays an important role, so a coefficient K_{intas} has been introduced, ranging from 0 (nominal filter conditions) to 1 (completely clogged filter conditions).

By using the following relation, $intas = 1 - K_{intas}$ has been used to correct the pressure gain in comparison to the nominal conditions:

$$GPF_{true} = GPF_0 * \left(intas * \frac{PSR}{PSR_0} \right)$$

Where:

PSR_0 = nominal supply-return differential pressure

PSR = actual supply-return differential pressure

$$GQF_{true} = GQF_0 * \sqrt{\left(intas * \frac{PSR}{PSR_0} \right)}$$

4.1.3 Fluid dynamic model

Ing. Dalla Vedova developed this model during his PhD research.

This paragraph will present the mathematical fluid-dynamic model for a proportional control servovalve. This model allows you to calculate flow rates to the actuator using a defined spool location XS .

It takes into account all geometric details and pressure drops through the cross sections. It only considers the fluid-dynamic portion and is valid regardless of how the spool moves. This allows it to be used to simulate both electromechanical and hydromechanical valves.

The fluid-dynamic model can be seen in Figure 4.11. It shows a typical configuration, with supply (S), return (R), and four ways. The flow through each section is calculated based on the position and geometry of the valve, as well as the differential pressure across them.

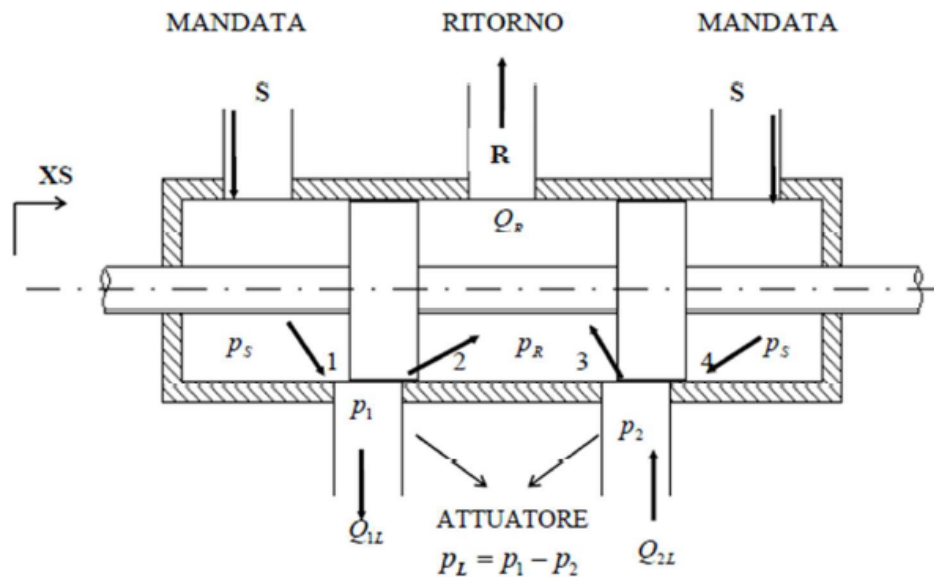


Figure 4.11: Control valve scheme 4/3-way directional

The differential pressure drop can be expressed in two ways. One is linear with flow rate and expresses laminar conditions, the other quadratic with flow rate expresses turbulent conditions. The viscous friction in the fluid is the cause of pressure loss. It is only present when the passing section of the fluid is covered by a spool.

In the second case, the discharge of kinematic energies through the passing

section causes pressure loss. It is always present as it is proportional to the passing area, which is greater than zero. These terms can be described as:

$$\Delta p_{lin} = RL \ Q$$

$$\Delta p_{quad} = R \ Q \ |Q|$$

where RQ is the pressure fall quadratic coefficient and RL is the pressure loss linear coefficient. You can express RQ and RL as:

$$RL = \frac{6 * \mu * l}{\pi * r * c^3 * \left[1 + \frac{3}{2} \left(\frac{e}{c}\right)^2\right]}$$

$$RQ = \frac{\rho}{2 * C_d^2 * A^2}$$

Where:

μ = dynamical viscosity

l = covering's length

c = radial clearance

e = eccentricity

ρ = fluid's density

C_d = efflux coefficient

A = passing section's area

The circumferential opening $w \simeq 2 * \pi * r$, ($r \gg c$) can also be introduced. RL can be used to express as follows, with $X0$ as the length of the covering and XS the spool position.

$$RL = \frac{12 * \mu * X0}{w * c^3 * \left[1 + \frac{3}{2} \left(\frac{e}{c}\right)^2\right]}$$

This will allow you to calculate the area of the passing sections and the covering $X0$.

When calculating $X0$, it is important to consider the lapping of valves: $OSSV$ refers to the lapping on the supply side and $ORSV$ refers to the lapping on the return side.

This notation also applies to flow rates: $Q1S$ is the flow rate of chamber 1 from the supply, $Q1R$ is the flow from chamber 1 from the return, and so on for $Q2S$ and $Q2R$.

$Q1L$ and $Q2R$ represent the net flow rates entering chamber 1 and chamber 2, respectively (these are assumed to always be positive as shown in Figure 4.11).

Each passing section will have the characteristic of an annular orifice with a length of $X0$. This is only true if the displacement of the spindle is greater than the lapping.

The lengths of each section can be expressed in the following:

$$X01S = OSSV - XS$$

$$X01R = ORSV + XS$$

$$X02S = OSSV + XS$$

$$X02R = ORSV - XS$$

Where:

$X01S$ = length of the supply to chamber 1 passageway

$X01R$ = length of the chamber 1 to return passageway

$X02S$ = length of the supply to chamber 2 passageway

$X02R$ = length of the chamber 2 to return passageway

Instead, the area of the passing section will assume a minimum value $w * c$ (annulus), if the corresponding lapping exceeds spool displacement XS . However, it will grow assuming a truncated conical area's value, as in Figure 4.12.

It is also possible to express the areas of the following sections as:

$$A1S = w * \sqrt{c^2 + (XS - OSSV)^2}$$

$$A1R = w * \sqrt{c^2 + (XS + ORSV)^2}$$

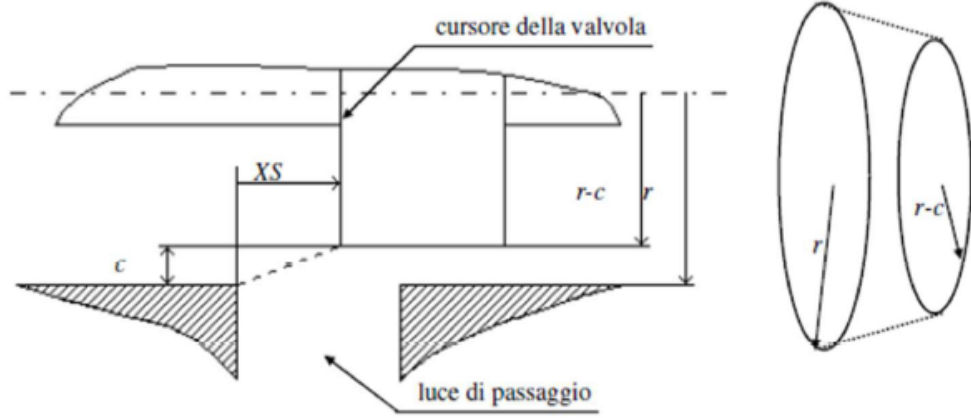


Figure 4.12: Zero-lapping valve's section of passing fluid when $XS > 0$

$$A2S = w * \sqrt{c^2 + (XS + OSSV)^2}$$

$$A2R = w * \sqrt{c^2 + (XS - ORSV)^2}$$

With the restriction to consider these areas equals $w * c$ when the flow rate of XS exceeds the correspondent lapping.

It is now possible to calculate the pressure drops across each area using flow rates:

$$\Delta p1S = (p_s - p_1) = \Delta p1S_{lin} + \Delta p1S_{quadr} = RL1S * Q1S + RQ1S * Q1S * |Q1S|$$

$$\Delta p1R = (p_1 - p_R) = \Delta p1R_{lin} + \Delta p1R_{quadr} = RL1R * Q1R + RQ1R * Q1R * |Q1R|$$

$$\Delta p2S = (p_s - p_2) = \Delta p2S_{lin} + \Delta p2S_{quadr} = RL2S * Q2S + RQ2S * Q2S * |Q2S|$$

$$\Delta p2R = (p_2 - p_R) = \Delta p2R_{lin} + \Delta p2R_{quadr} = RL2R * Q2R + RQ2R * Q2R * |Q2R|$$

These equations allow you to calculate flow rates through each area:

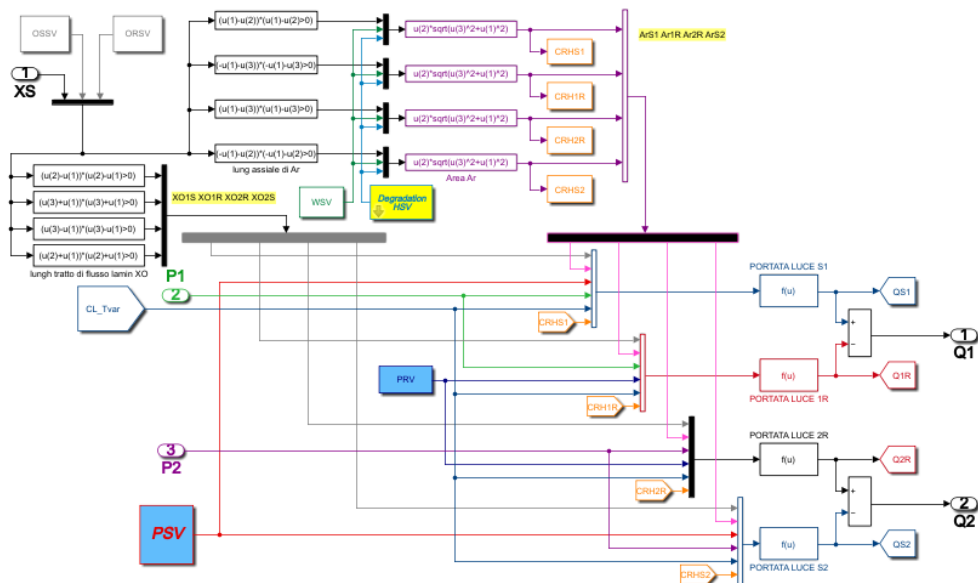
$$Q1S = \frac{-RL1S + \sqrt{RL1S^2 + 4 * RQ1S * |p_2 - p_1|}}{2 * RQ1S} * \text{SIGN}(p_s - p_1)$$

$$Q1R = \frac{-RL1R + \sqrt{RL1R^2 + 4 * RQ1R * |p_1 - p_R|}}{2 * RQ1R} * \text{SIGN}(p_1 - p_R)$$

$$Q_{2R} = \frac{-RL_{2R} + \sqrt{RL_{2R}^2 + 4 * RQ_{2R} * |p_2 - p_R|}}{2 * RQ_{2R}} * \text{SIGN}(p_2 - p_R)$$

After each flow rate has been calculated, the useful flow rates for the actuator can be obtained as an algebraic sum of all the flow rates through the relevant passing areas:

$$Q2L = Q2S - Q2R$$



This subsystem allows you to calculate the nominal efflux factor Cd

The first ($Cd0$), which is expressed as an empirical relation and takes into account the actual fluid density and cinematic viscosity through lookup tables, represents the nominal efflux rate. The second (C') is the correction of this first coefficient, determined from the variable geometry. The scheme is shown in Figure 4.14.

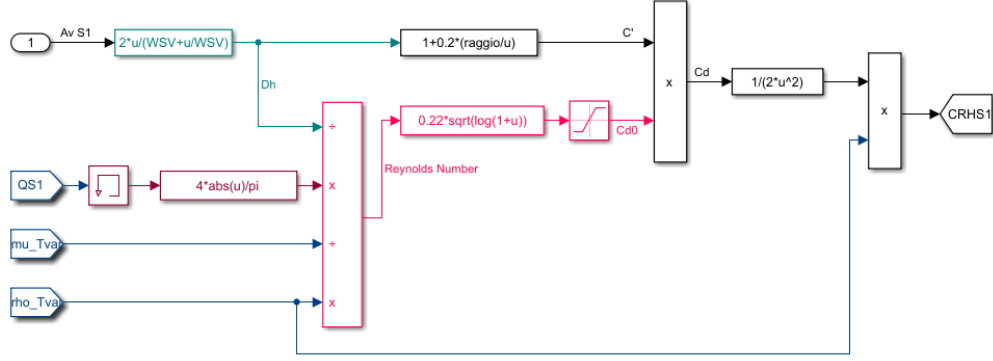


Figure 4.14: CRH's subsystem

Although the mathematical model described above allows you to calculate the characteristic curves for the flow rate and trend of regulated flows as functions of the spool position XS and pressures $P1$ and $P2$, it is not possible directly to determine the characteristic curves for pressures at a given flow rate.

This problem was solved by a calculation routine that simulates fluid compressibility in circuit after servovalve. It allows you to calculate the temporal derivatives of the regulated pressures at the two outlets of the valve.

They can be written as follows:

$$P1(t + DT) = P1(t) + [Q1L(t) - QM(t)] * \frac{Bet}{Vol} * DT$$

$$P2(t + DT) = P2(t) + [QM(t) - Q2L(t)] * \frac{Bet}{Vol} * DT$$

Where:

$P1, P2$ = outlet valve's regulated absolute pressures

$Q1L, Q2L$ = actuator's disposed flow rates

DT = sample time

Bet = fluid's compressibility coefficient of the fluid

Vol = hydraulic capacity's volume

QM = actuator's total absorbed flow

The last term can be expressed in terms of potential leakage:

$$QM(t) = AJ * DXJ(t) + Clk * [P1(t) - P2(t)]$$

These equations were modified to accurately simulate cavitation phenomena within the fluid, which can have a significant impact on the dynamic behavior of a system:

$$P1T(t + DT) = P1T(t) + [Q1L(t) - QM(t)] * \frac{Bet}{Vol} * DT$$

$$P1(t + DT) = MAX[PVap, P1t(t + DT)]$$

$$P2T(t + DT) = P2T(t) + [QM(t) - Q2L(t)] * \frac{Bet}{Vol} * DT$$

$$P2(t + DT) = MAX[PVap, P1t(t + DT)]$$

$P1T$, and $P2T$ are auxiliar terms that allow you to think about possible formation of vapour bubbles. It is possible to have memory for eventual negative pressure picks (even though these picks may be fictitious), that highlight the possibility of vapor bubble formation within the fluid.

This routine would cause calculation to lose the memory of oil vapour volume. It is formed in cavitation. The first must return to liquid to allow pressure to grow beyond $PVap$. This is equivalent to not considering vapor bubbles in fluid.

The pressure would then grow positively after cavitation. However, these assumptions reduce the cavitation effect, so that the real pressure grows only when $PVap$ is higher than the theoretical pressure.

The dynamical system calculates pressures by dividing the net flow rates of fluid capacities by the corresponding calculated pressures. This is a first-order model, so reducing the capacities will reduce the constant time τ .

In particular, if the capacities are too small, the model could be unstable due to incompatibility between constant time t , and sample time DT .

Below is the derived model for the pipe capacities Figure 4.15 shows the fluid dynamic model in its entirety. This model provided the basis for the characteristic curves that were used to design the Low-Fidelity fluid dynamic models.

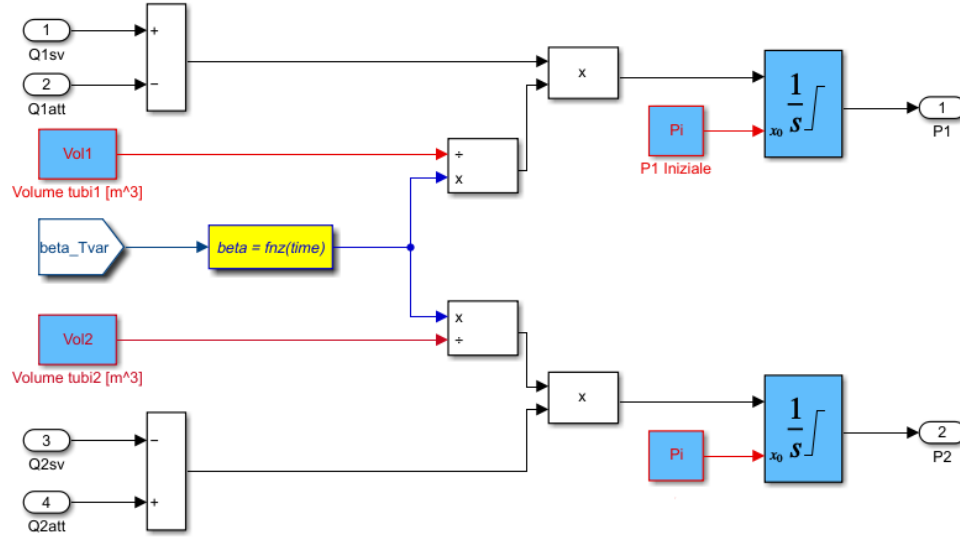


Figure 4.15: Model of pipe capacity

The complete fluid dynamic model is shown in Figure 4.16

The characteristic curves resulting from this model will be discussed in the next chapter.

4.1.4 Mechanical model of the jack

A mechanical model of the jack is the final block in the analysis. It takes the differential pressure P_{12} determined by the fluid-dynamic model and multiplies it by the equivalent surface of the jack, providing the input for the jack's model. It can be seen that the structure is very similar to that of the first stage of the servovalve model.

After subtracting dissipative forces from the input force, the actual force is calculated: through a manual switch, the sum of possible external and aerodynamic forces can be obtained based on the viscous force, given by the jack's velocity multiplied by the viscosity coefficient CJ .

Aerodynamic forces are calculated by multiplying the jack's position by the aerodynamic coefficient K_{aer} , which simulates the opposition between

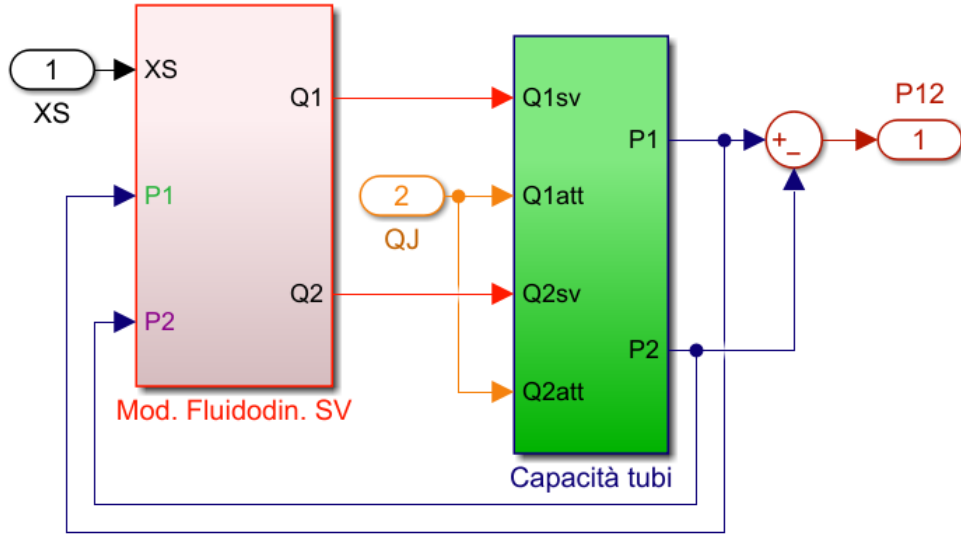


Figure 4.16: Fluid dynamic model

aerodynamics and displacement of mobile surfaces.

Following this, Borello's friction model receives the theoretical net force, the velocity of the jack and the saturation value of the integrator's saturation port, giving as output the real friction force as well as the first integrator's reset value.

Whenever the jack reaches the upper or lower limit of its displacement, a saturation block will nullify its velocity. Then the acceleration of the jack is obtained by dividing the net force by the mass of the jack, and with two successive integrations its velocity and displacement are calculated.

Additionally, a backlash block follows the second integrator in the closed loop, which simulates any possible backlash on the kinematic chain joining jack to mobile surfaces due to usury.

Assumed to be installed on the mobile surface, the sensor is subject to possible backlash and has been inserted into the closed loop.

The final step is to insert a quantizer block before the output XJ to consider the discretization issue resulting from digital acquisition.

Chapter 5

Development of Computational models

Due to the extensive development in architectural design and technological solutions, it is becoming increasingly challenging to evaluate new systems on board the aircraft. Today we need instruments that can prove the effectiveness of innovative solutions or make an existing component more efficient, no matter how complex they are. As computers have become increasingly powerful in recent decades, they have become able to perform this function by providing modules of calculus that are competitive enough to support the computing needs of the mathematical model of a system. As a result, designers are now able to build complex mathematical models using commonly available computational resources.

In the aerospace sector, as in many others, there has been a lot of effort put into developing complex systems and now has succeeded in reducing the great cost of tests by replacing and supporting them. By using prototypes, it's now possible to simulate a system's response in certain specific conditions, making it unnecessary to prove and verify the system.

Computing models can also be used to estimate a system's health condition and predict its deterioration with high degree of accuracy. A consequence of this is that an element can have a longer useful life, and the costs can be amortized more efficiently. This makes diagnostics and prognosis easier.

Modern models are useful not only during the flight but also during the operative period of the aircraft. An operative error can be determined by comparing a system's response with the expected one. Once this error has exceeded a certain threshold, the component is designated as failing, and the

pilots and maintainers are notified.

5.1 Developing of Database

The first step in developing the servo-valve models was to obtain a large amount of data from the High-Fidelity model of the servo-valve, in order to create a number of databases that can be used to train the various modeling methods.

Each database consists of a certain number of maps representing the servovalve characteristic.

Each map represent the servovalve characteristic for a certain combination of values representing the servovalve status. These values are covering and radial clearance.

Databases consist of 4 variables inserted into a 4-D matrix, defined by:

- Spool position
- Flow
- Covering
- Radial clearance

The definition of these four variables leads to the representation of the pressure value for these conditions.

The first variable, the spool position, is represented on the x-axis of each map, while the pressure is represented on the y-axis. Each map represents a certain number of flow rates and features a different combination of radial clearance and covering, therefore the choice of the last two variables defines which map we are analyzing and the total number of maps of each database is given by the product of these variables.

The high fidelity model provides the pressure values for a single combination of the four variables involved. The pressure starts from zero and tends to a constant value after a transition period.

The databases were created by blocking the covering and radial clearance variables for each single map, and by varying the flow rate and the spool position to obtain the characteristic curves of that single map.

Each map is therefore composed of a number of simulations given by the product of the number of spool positions analyzed by the number of flow

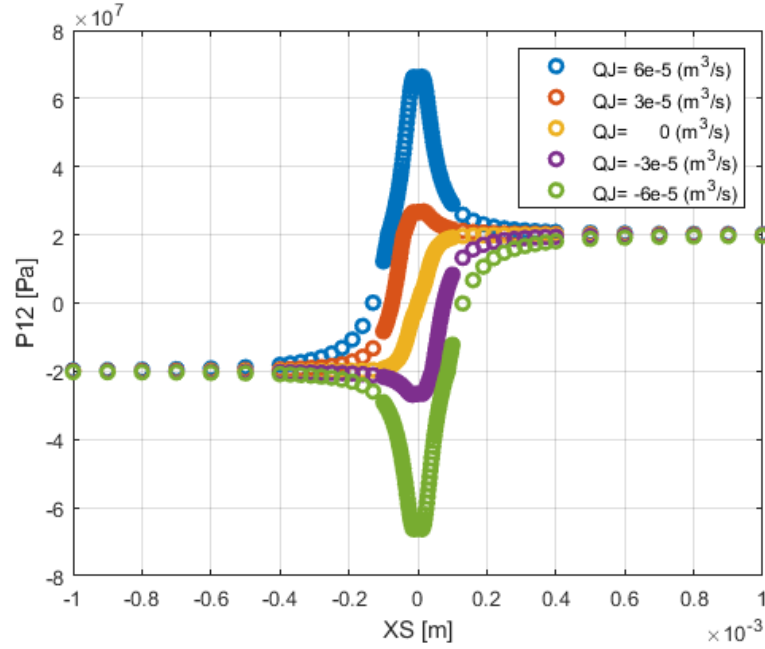


Figure 5.1: Data concentration in databases

rates considered, and each database is given by this number multiplied by the number of combinations of radial clearance and coverage of that database.

The simulations were performed with a simulation time of $1e-2$ and an integration step of $1e-6$.

Each map is composed considering 133 spool positions, with a higher density around the null position of the spool, to better represent the pressure gradients around it.

Figure 5.1 shows this distribution of the spool positions in the simulation of the servovalve map; we can see the lower density of XS in the extremes where there is a linear behavior of the pressure.

5.1.1 Databases

Several databases have been created to better address the issues related to each servovalve's performance modeling method, they are described in the following table and explained in the following paragraphs.

Databases					
Database	Spool positions	Flow Rates	Coverings	Radial Clearances	Total Data
1	133	5	5	5	16625
2	133	15	9	9	161595
3	133	15	9	9	161595
4	133	15	30	30	1795500
5	133	133	1	1	17689

Database 1

The first database was composed of 25 maps of the servo-valve.

Figure 5.2 shows the simulink model to simulate the 5 flow rates, in which the EHA high fidelity model discussed in chapter 4 is present in each block.

We can see in Figure 5.3 an example of some maps, and how the variation of the radial clearance and overlap parameters affect them.

This database was used for the development of both the lookup table and the genetic algorithm, but it proved unsuitable for neural network development.

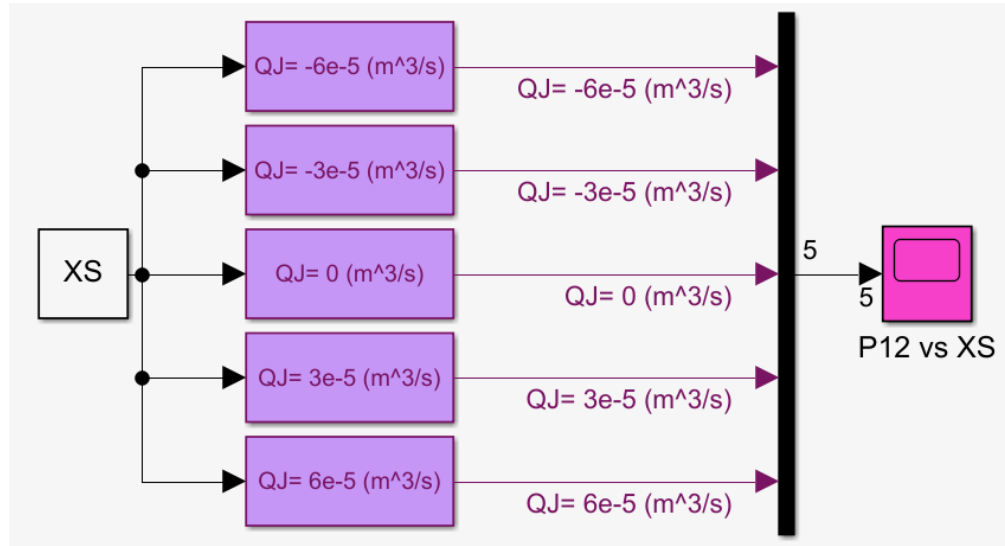


Figure 5.2: Simulink model to create the characteristic curves

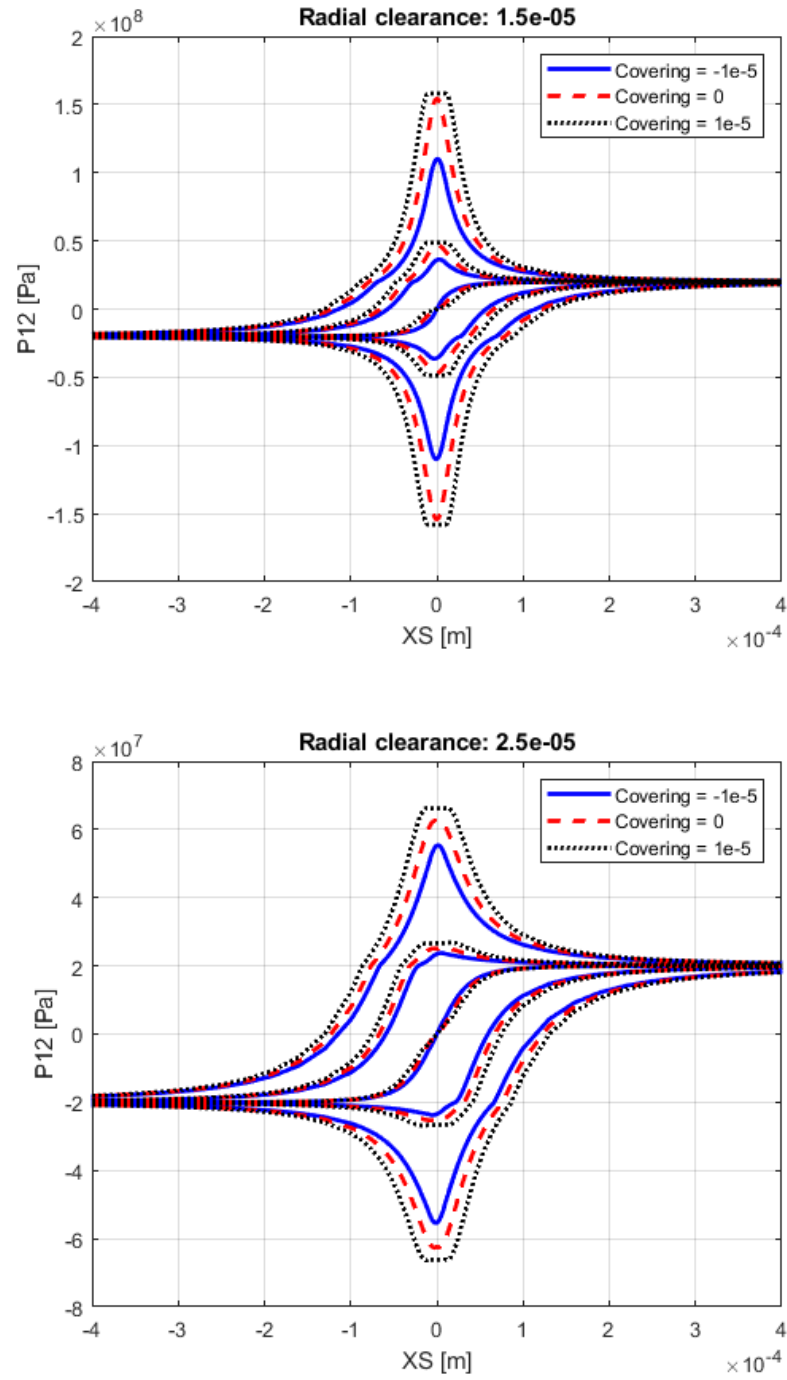


Figure 5.3: Characteristic curves with parameter variations

Database 2

Two problems were encountered in applying the first database to the neural network: the first is related to the size of the database, as the efficiency of applying a neural network to a non-linear problem is directly proportional to the size of the database.

The second problem is related to the way in which the database was developed, precisely to the differentiation between the four variables. We will see this problem more precisely in the paragraph on the neural network.

To improve the performance of the neural network, the combinations of the variables flow rate, radial clearance and coverage were therefore increased, and consequently the total number of data. The number of maps in this database is 81.

Database 3

In using the neural network as a tool for simulating the performance of the servovalve, by modeling the outlet pressure relative to the other parameters, it was noted that it was more difficult to correctly model the flow rate values close to zero flow, as opposed to those at greater range with large water hammers that the net is able to model with good accuracy.

To try to improve the performance of the network also for flow rates close to zero flow, a database was created with the same number of data but with flow rates limited to a lower range.

Database 4

The use of the neural network as a method of diagnosing the conditions of the servovalve made it necessary to create a database with a high number of combinations of games and coverings, i.e. a high number of maps, in order to allow the network to recognize the differences between these variables.

Database 5

To solve the problems encountered in the application of the neural network in the simulink model of the complete actuator, which will be discussed in the last chapter, the network has been simplified by removing the two variables of radial clearance and covering, and making the training matrix a square matrix with an equal number of spool positions XS and flow rate.

By plotting the results on a grid, we can visualize the surface representing the pressure values for any combination of XS and Q , which in the case of this database lays on a grid with a square base.

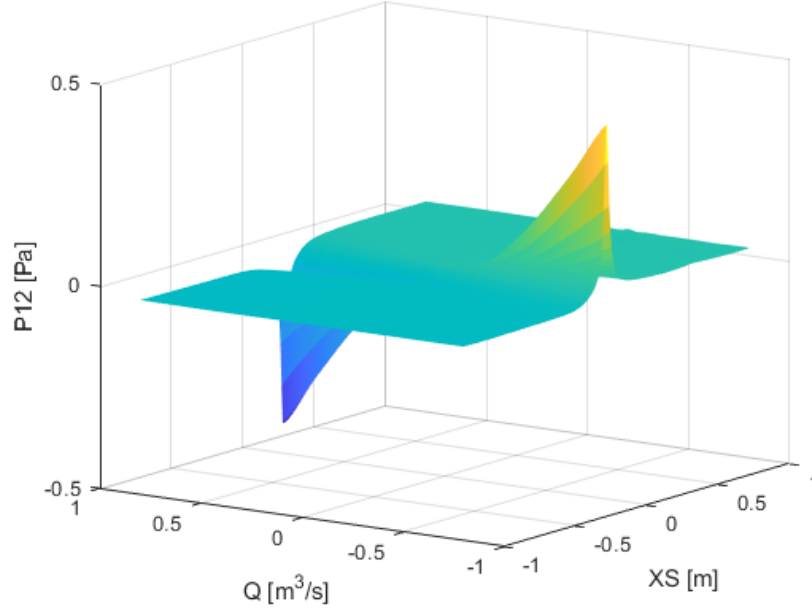


Figure 5.4: 3D surface characteristic curves

5.2 Lookup table

A Lookup Table is a set of data sorted in an array or in a matrix which, thanks to their structure, allow to associate an output data configuration to each admissible combination of input data.

In this way it is possible to replace *runtime* calculation operations with a simpler consultation operation.

The first computational method that was tried in order to replace the high-fidelity model, is the use of a lookup table. Using an array of data, a lookup table block approximates a mathematical function. Look-up tables with K inputs are digital memories that can implement any Boolean function over K variables.

Given a set of input values, a lookup operation retrieves the corresponding output values from the table. A lookup table represents an array of data

that maps inputs to outputs, hence representing an approximation of a mathematical function.

Simulink can estimate an output value when the look up table does not explicitly define the input values, through the following methods: interpolation, extrapolation, rounding. Using lookup table blocks may allow for speed gains when simulating a model, since table lookups and simple estimations could be faster than mathematical function evaluations.

This method will be only used to obtain the pressure for different clearance and coverage values than those used in the database, with which the board will be trained. a 4-dimensional lookup table was implemented through the specific simulink block shown in Figure 5.5.

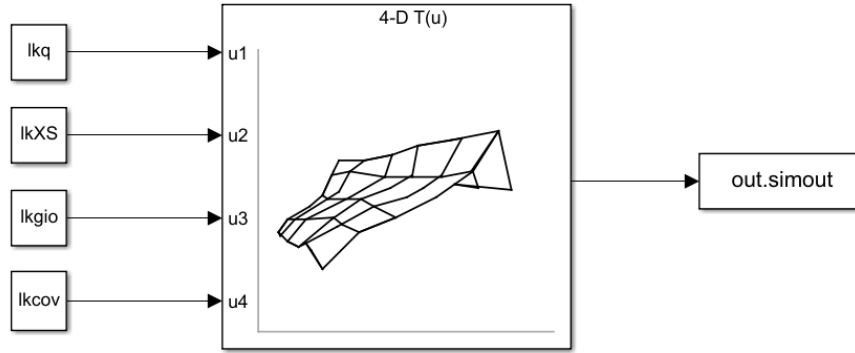


Figure 5.5: Look-up Table Simulink Model

To use the lookup table, the Table Data (in this case the pressure values) and four breakpoints (the arrays containing the four variables in play) will be set.

After having designed the table, we proceeded to test it with a different set of radial clearance and coverage from the database's set, through a matlab code that also calculated the percentage error. Results are shown in Figure 5.6.

The error range of this method is 5 – 10%, but since the time needed to simulate a single map is around 60 seconds, it is too slow to be used in the Simulink model of the complete actuator.

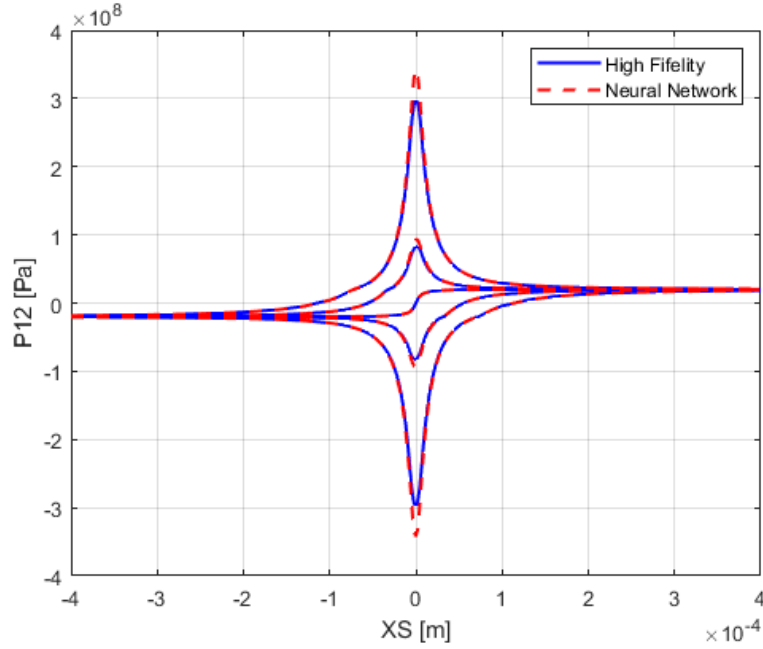


Figure 5.6: Look-up Table results

5.3 Genetic Algorithm

The second method analyzed is a genetic algorithm. This method should be faster but it needs a correct definition of the objective function and of some parameters in order to function correctly.

A genetic algorithm is a process inspired by Darwin’s principles of natural selection and genetics to solve optimization problems. Based on a set of solutions evolved in intervals of time, called generations, the evolution is guided by determining the optimal solution based on the “goodness” of the individual or generation; it is possible to determine the best individuals using the objective function, and use them to create successive generations. Each individual is unique in terms of characteristics and properties: these are the phenotypes. Those traits are determined by an invisible genetic setup, called genotypes, which are the basic building blocks of chromosomes.

Genetic algorithms have a cyclic base structure, so each cycle represents a generation, and each step represents a new generation. The first step is to create an initial population, and the following steps are repeated each time a new generation is born. Fitness function values guide selection of the

most suitable individuals, while crossover operations and mutations lead to creating new generations, which combine or modify the characteristics of the individuals.

Figure 5.7 shows the characteristic scheme of a genetic algorithm

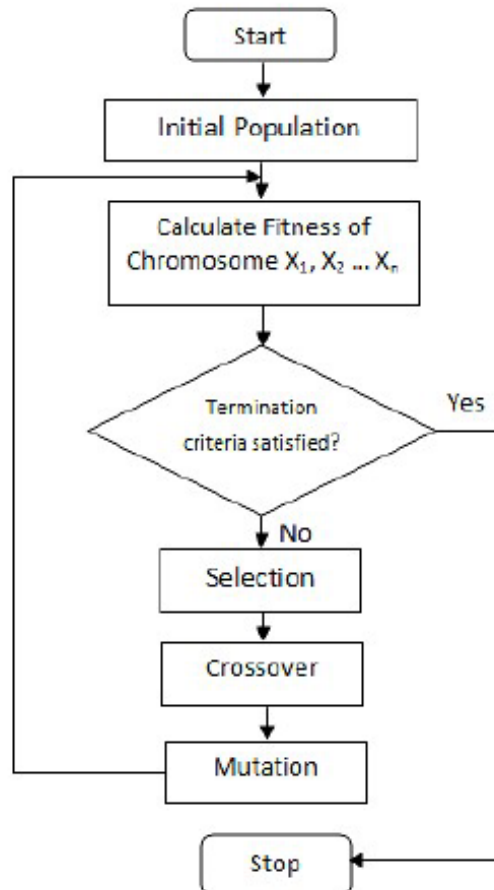


Figure 5.7: Genetic algorithm scheme

First, it is necessary to define the fitness function, which is the objective function that the algorithm should minimize and indicates how far an individual is from the optimal solution. For biological individuals, it is the external environment in which they can thrive or die. The objective function can have local and global minimums, so if the initial population is concentrated in the local minimum, the algorithm will reach this point. Therefore, it is essential to have a large enough number of individuals covering most of the domain, in order to explore all the possible solutions, but this number should

not be too high to make the algorithm very time consuming.

MATLAB provides several ways to obtain an initial population: the default option generates a random population that satisfies the bounds, if present, and is well-distributed within the domain. In addition, the initial population can be specified totally or partially, so that it has interesting individuals from the beginning. In case only a part of the initial population is defined, a function called *CreationFcn* will fill the remaining individuals.

Following the definition of the initial population, each individual is scored according to his fitness value, and a sub-population is chosen to generate new solutions. Different criteria are used for selection: the default option is the *rank scaling function*. It is a process of sorting and scaling each individual's raw score based on their position in each category. This is a conservative method, since also less fit individuals can be selected for reproduction, thereby making it easier to use genetic portions, but this method is slower to convert.

Besides these methods, there is also *proportional selection* that choose the best individuals according to their fitness function values: first the population's fitness function value is determined:

$$F_{max} = \sum_{i=1}^n f(x_i)$$

$$F_{min} = \sum_{i=1}^n \frac{1}{f(x_i)}$$

Where x represents the i th individual. The first value is used in the research of maximum fitness functions and the second for minimum fitness functions. With this value, we can determine the likelihood of selecting each individual according to fitness function.

$$p_i = \frac{f(x_i)}{F_{max}}$$

$$p_i = \frac{1}{F_{max}} \frac{1}{f(x_i)}$$

As shown here, the results are represented by a wheel divided into segments, with each segment representing a probability associated with a particular individual.

To select the individuals for reproduction, the wheel is spun k times. The new generation $P(n+1)$ is obtained from the previous one $P(n)$ through the following steps:

1. The best individuals are selected for reproduction, and the best ones are introduced into an intermediate population P1 and afterward into the mating pool
2. By applying the crossover operator to individuals in the mating pool, a new population P2 is generated
3. Using the mutation operator, the individuals P2 will generate a second population P3
4. As a result of replacement and survival selection, the new generation P(n+1) will consist mainly of individuals in P3 but may also include those in P(t) that have not been chosen for reproduction or the best ones

New chromosomes are created by the parents, thus the new solutions have a genetic bond with the parents, otherwise convergence would not be guaranteed. As a result of these recombination process, crossover is also said to reflect the exchange of homologue material between two or more chromosomes.

The following options are available for crossover: scattered, single point, two point, arithmetic, intermediate, heuristic

By using the *scattered function*, a random binary vector is created, in which a bit from the first parent is equal to 1 and a bit from the second one is equal to 0.

For example:

$$P1 = [12345678]$$

$$P2 = [abcdefgh]$$

Single point functions divide the genetic information from the parent genes into two parts by selecting an integer number I between 1 and n-1 bits, the first part being composed of the genes with an index less than i, and the second part being composed of the genes with an index greater than i. In the resultant child, the first part will come from the first parent and the second part will come from the second parent.

$$Reproduction = [10101010]$$

$$Child = [1b3d5f7h]$$

Two points crossover works similarly, but the two integer numbers are chosen in order to obtain three genetic portions. The offspring will, for example, contain the middle portion of the genetic code from one parent and the complementary part from another. In the examples so far, binary codification is used, while real numbers are used in the remaining options.

Using the arithmetic crossover approach, the children are computed as the weighted arithmetic mean of the two parents. Using the intermediate crossover approach, the children are calculated as the weighted average of the parents; we have the option to specify the parameter ratio.

Heuristic crossover produces a child which lies on the line containing the two parents, a short distance from the parent with a better fitness value, and a long distance from the parent with a worse fitness value.

For the intermediate crossover of the formula, it is possible to specify the same parameter ratio.

Mutation operator introduces minor changes in the genetic material to create new chromosome. An individual's chromosomes are modified by this operator in order to explore new areas of the domain of possible solutions that have not yet been explored: this is done in order to avoid the convergence into a local minimum. In contrast, excessive use of mutation results in an algorithm that is unable to converge. As a result, it is applied with a very small probability, oscillating between $[0.001, 0.01]$.

Several options are available: gaussian, uniform and adaptive feasible functions. In gaussian mutations, each entry of the parent vector receives a random number from a gaussian distribution with mean 0. The process of uniform mutation involves two steps: in the first step, the algorithm selects a fraction of an individual's vector entries for mutation, with each entry having an individual mutation probability. The second one involves replacing each selected entry with a random number drawn uniformly from a specified range. In adaptive feasible functions, directions are randomly generated that are adaptive, taking into account whether the previous generation was successful or unsuccessful, while satisfying bounds and linear constraints.

5.3.1 Development of the object function

The objective function must be written in such a way as to be able to represent the graph of the pressure with respect to the position of the

spool, for different flow rates. It must therefore be an algorithm with some coefficients and some parameters that will be exponents of the spool position and flow rate.

The designed algorithm is composed of three terms: the first is a function only of the spool position, which is inserted in a hyperbolic tangent. In this way it is possible to represent the pressure function at zero flow rate. In this term there are two coefficients.

The other two terms are fractions in which in the numerator there is the flow pressure raised to a certain odd power (to a unitary power for the first term and to a third power for the second), while the denominator occurs in both cases the sum of the position of the spool raised to two equal powers (to the second and to the fourth power).

These three terms have their own coefficient in both fractions, therefore by adding the two coefficients of the hyperbolic tangent there is a total of 8 coefficients, which will be the variables of the genetic algorithm.

The following is the algorithm of the objective function, which will allow the genetic algorithm to be trained through the error between the value of this function with the coefficients limited to varying within certain values, and the results of the high fidelity simulation.

$$c_1 * \tanh(c_2 * w_{xs}) + \frac{c_3 * w_q}{(1 + c_4 * w_{xs}^2 + c_5 * w_{xs}^4)} + \frac{c_6 * w_q^3}{(1 + c_7 * w_{xs}^2 + c_8 * w_{xs}^4)}$$

5.3.2 Genetic Algorithm development

The first step to be able to train the genetic algorithm was to normalize the data. The values of spool displacement, flow rate and pressure have been multiplied by coefficients in order to have all values in the order of the unit. Normalization will also be used for the neural network's data.

A matlab code was therefore written that would call the objective function described in the previous paragraph to use it within the matlab ga function. Within the same code, the options of the ga function have been set.

To allow a complete analysis and reduction of the error by the genetic algorithm, the following maximum values concerning the generations have been set:

Max Generations = 100000

Max Stall Generations = 50000

At this point the algorithm is able to reach convergence and to represent the general pressure trend, but without being equipped with an initial set of coefficients from which to start and also from upper and lower limits it cannot reach an acceptable value efficiency and error.

Initial coefficients

Through a trial & error process, a set of initial coefficients was selected that could already represent the characteristic map of the servovalve fairly faithfully, so that the genetic algorithm could start from it in the selection of the best coefficients. The set of coefficients that has been selected is the following:

$$\text{initial coefficients} = [2e7 \ 2e2 \ -1.5e7 \ 800 \ 10000 \ -8e5 \ 5000 \ 10000];$$

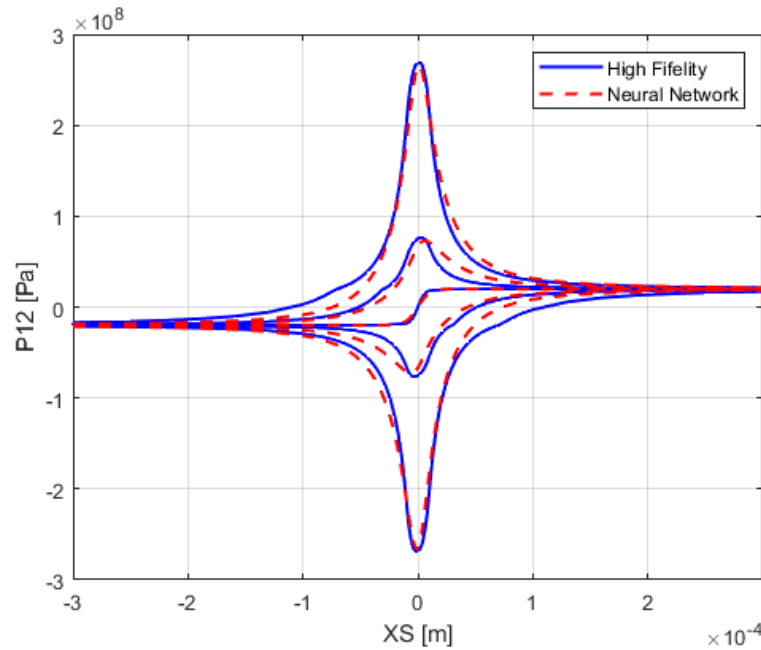


Figure 5.8: Genetic algorithm results

Upper and Lower Bonds

The following limit values have been selected, through a trial & error process, within which the coefficients can vary during the training of the genetic

algorithm:

lower bonds = [2e7 1e2 -1e9 -1e7 -1e7 -1e9 -1e7 -1e7];

upper bonds = [2e7 1e3 1e9 1e7 1e7 1e9 1e7 1e7];

5.3.3 Results

By operating the genetic algorithm with these parameters, it was possible to obtain the results shown in Figure 6.9.

The total error of pressure in Euclidean norm is 13%, which could be improved through a better choice of initial coefficients, upper and lower limits, and genetic algorithm options.

5.4 Neural network

In complex problems, artificial neural networks (ANNs) are becoming more common since they offer a greater degree of versatility than the traditional regression algorithms.

Even when it's not clear which is the best regression equation for the data or if the data are correlated at all, ANNs are capable of finding correlations between their inputs and outputs. In this study, shallow artificial neural networks in a feed-forward configuration were used.

A shallow ANN only contains one hidden layer. As the name implies, in a feed-forward network information is only passed from one input node to the next, and the processing of that information is determined only by the current input, unlike in a recurrent neural network.

Generally, this type of architecture is found in regression problems, and the accuracy can be improved by adding more input nodes or hidden nodes or by adding more hidden layers. In the figure, we see a simple shallow feed-forward ANN with inputs, neurons hidden in layers, and outputs.

The network is fully connected, since each layer's nodes are connected to the layer below. An optimum regression coefficient (often described as a desired mean square error) will be found after an adequate number of cycles, or until the networks fail too many validation tests.

During the validation phase, data that is not used for training is employed to ensure the network's performance (usually measured by the Mean Squared Error) on random data is still satisfactory after every training cycle.

An excessive number of validation checks failed consecutively (e.g. MSE of predictions on on-training data drops), likely means that the network has overfitted on the training data and that additional training would be ineffective.

Training an ANN typically involves a large number of sets of known inputs and a set of known targets.

As MATLAB and his Deep Learning Toolbox provide a variety of features and options at a reasonable performance level, we selected the software for programming and training the ANN.

In this thesis the neural network was used with a double purpose: the first was to obtain the pressure relative to the four variables of the database, in the same way as was done for the look-up table and the genetic algorithm.

The second use was of the diagnostic type, allowing the network to recognize which radial clearance and overlap values (representing, as already mentioned, the wear of the servovalve) were related to a certain series of spool position values, flow of flow rate and outlet pressure. In this way we tried to obtain a diagnostic tool capable of recognizing the state of health of the servovalve based on data that can be obtained in flight.

It must be taken into account that the characteristic maps obtained in the previous paragraph consist of a high number of spool positions and flow rates (hence a high number of pressures). On the contrary, the number of radial gaps and coverings, from which the number of these maps derives as we have said, is not very high due to the fact that increasing it means increasing the number of maps and therefore significantly increasing the number of simulations to be carried out.

This has led to the need to change the way in which the database was conceived, passing from a conception by points to a conception by function, allowing the network to associate a single pressure trend with the relative clearance and coverage values. In addition to this, the number of games and overlays in database 4 has been increased, as explained above.

In this way it was possible to use the network in both ways, with databases conceived by points in the modeling activity of the output pressure from the servovalve, with databases for pressure functions in the diagnostics activity of the servovalve conditions.

Both methods will be analyzed in the following paragraphs.

The following images show the results of the neural network in terms of error, performance and regression.

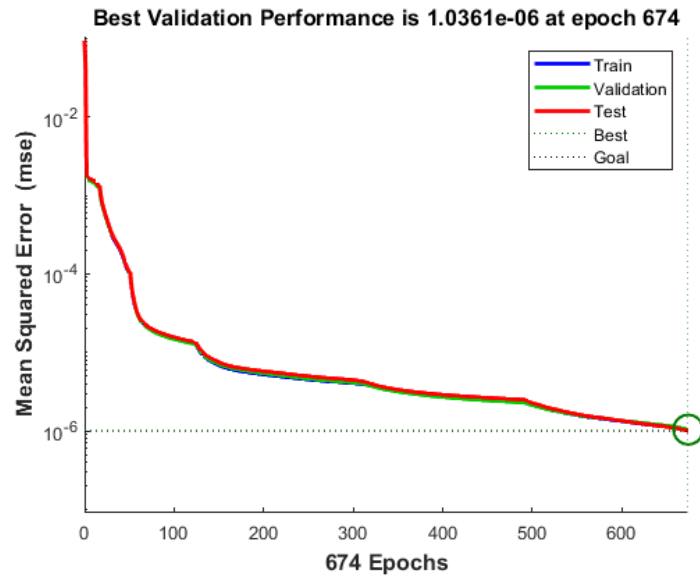


Figure 5.9: Neural Network performances

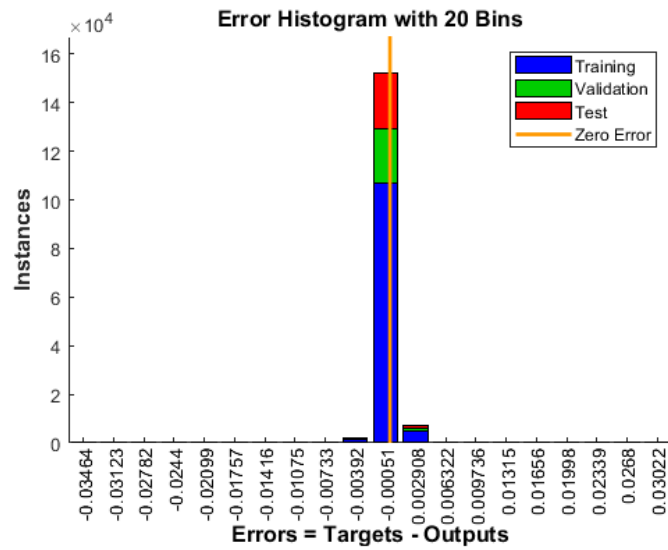


Figure 5.10: Neural Network error histogram

5.4.1 Training algorithm

First we need to select a training algorithm. It is not known initially which will be the best training function, as their effectiveness depends on several

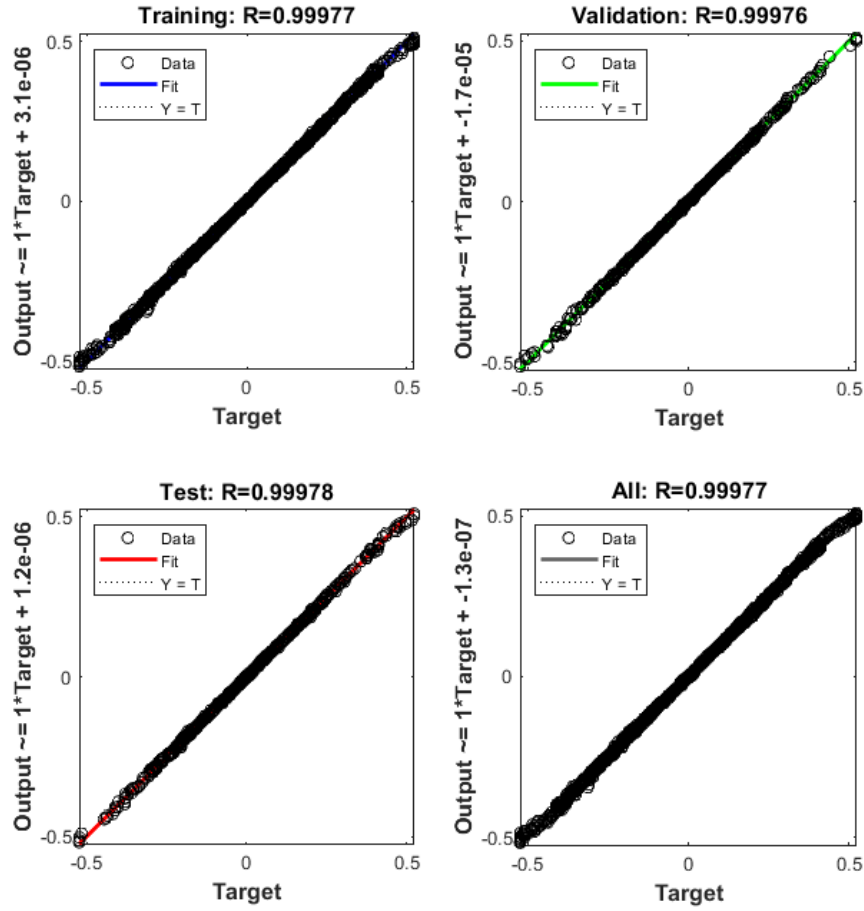


Figure 5.11: Neural Network regression

factors including the complexity of the problem analyzed, the size of the database, the error goal, the type of problem analyzed.

The types of problems that can be analyzed by the neural network are divided into pattern recognition (discriminant analysis) or function approximation (regression). About our cases, the analysis of the servovalve diagnostics can be considered a pattern recognition problem, while the modeling of the servovalve performance and pressure is a function approximation problem.

When it comes to function approximation problems, the Levenberg-Marquardt algorithm will have the fastest convergence rate when networks contain up to a few hundred weights. When precise training is required, this

advantage becomes particularly apparent. The mean square errors of `trainlm` are usually lower than those of any other algorithm tested. `Trainlm`'s advantage decreases as the number of weights in the network increases. `Trainlm`'s pattern recognition performance is also poor. It requires more storage space compared to the other algorithms.

Another training function that applies the Levenberg Marquardt optimization principle to weight and bias values is the `trainbr` function. It minimizes an error combination of squared errors and weights, determining the best combination for producing a generalized network.

Pattern recognition problems are best solved using the `trainrp` function. On the other hand, it fails to solve function approximation problems well. With a smaller error goal, its performance also declines. In comparison with other algorithms, the memory requirements for this algorithm are relatively low. For networks with a large number of weights conjugate gradient algorithm, such as `trainscg`, appear to do well. On function approximation problems, the SCG algorithm is almost as fast as the LM algorithm (even faster for large networks), and on pattern recognition problems is almost as fast as `trainrp`.

Conjugate gradient algorithms have relatively modest memory requirements, and their performance does not degrade as quickly as `trainrp`'s does when the error is reduced. In terms of performance, `trainbfg` is comparable to `trainlm`. It requires less storage, but as the nodes increases the computation required does not increase geometrically, since each iteration must compute a matrix inversion.

While `traingdx` has a much slower learning rate, it contains the same amount of data as `trainrp`, so for particular problems it can still be useful. Using a fast algorithm can result in inconsistent output data in certain situations, for example when an early stopping is used. It is possible to overshoot the point of minimal error on the validation set.

After several tests with the various functions, it was decided to use the `trainlm` function both for the simulation of the performance of the servovalve and for the diagnostics of its conditions.

5.4.2 Modelization

In this paragraph we will study the use of the neural network as a method of modeling the performance of the servovalve, simulating the outlet pressure as a function of the other parameters of the databases.

For the application of the databases to the neural network, it was necessary to apply some changes to them and set them in appropriate matrices.

First of all the data have been normalized in the interval $[-1, 1]$. Subsequently they were divided into two matrices, called data and labels. The first contains the variables linked to the related results in the labels matrix. In both matrices, the data of a single variable are all represented on a single row, with the number of rows given by the number of variables in the matrix. In the case of the servovalve performance modeling we are dealing with, the data matrix has four variables (spool position, flow rate, radial clearance, covering) and the labels matrix has only the output pressure.

We will see how to use the neural network as a servovalve diagnostic method it will be necessary to set the matrix in a different way.

Before proceeding with the simulations it is necessary to select the architecture of the neural network, i.e. the composition and the number of hidden layers from which the network will be composed. The hidden layers are composed of a certain number of neurons and are included between the input and output layers given by the number of variables of the two matrices labels and data.

For the use of the neural network as a modeling method of the servovalve, the following architecture composed of three hidden layers has been selected through trial & error: $[21 \ 15 \ 3]$. the scheme of the chosen architecture is shown in figure 5.12.

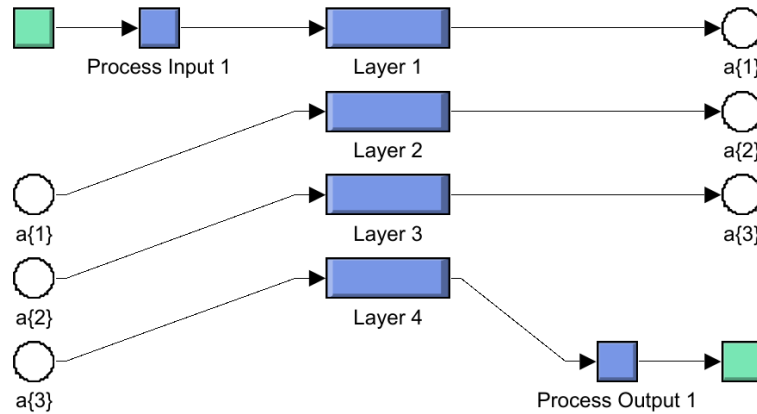


Figure 5.12: Neural Network structure

Training with database 1

Training with database 1 leads to an error of 7-8 %.

As we can see from Figure 5.13, the error is mainly in the calculation of the water hammer peaks and in the modeling of the function for zero flow.

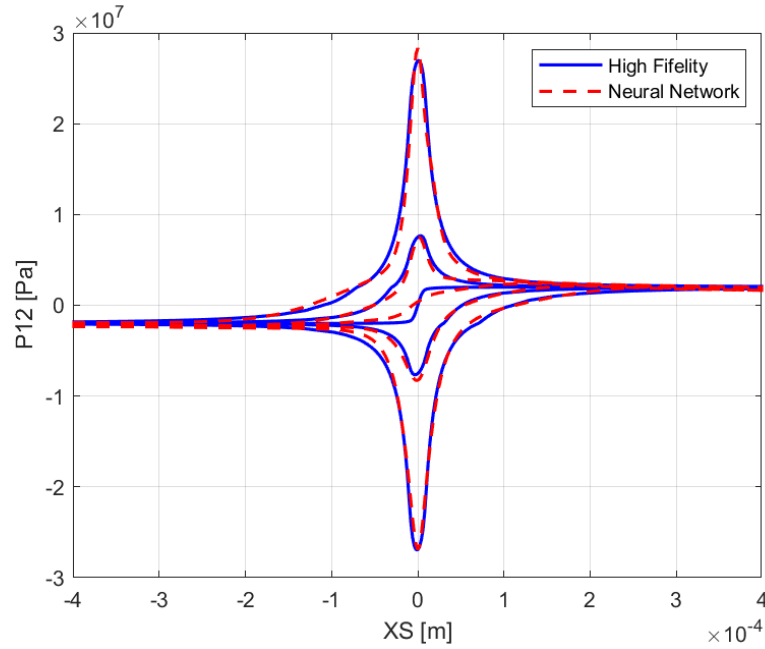


Figure 5.13: Neural Network with database 1

Training with database 2

To improve the results of the neural network, it was decided to increase the number of ranges with database 2.

Repeating the training the error dropped to 2-3%, but as we can see in Figure 5.14 the network still has difficulty in modeling the pressure relative to the zero flow, while the water hammer modeling is correct.

Training with database 3

At this point it was decided to decrease the flow range in order to allow the network to train and better understand the pressure behavior for low flow rates, and this was done with database 3.

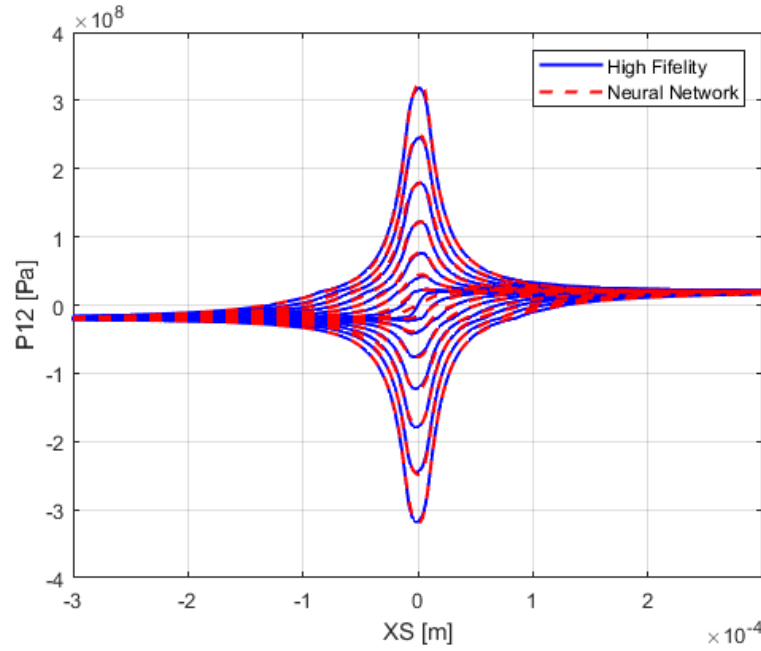


Figure 5.14: Neural Network with database 2

The error has always remained around 2-3 %, while as we can see from Figure 5.15 the ability to model the flow rates close to zero has increased to the detriment of the water hammer modeling.

We can better visualize the results by plotting the individual curves. In Figure 5.16 we can see the curve for a single flow rate, while in Figure 5.17 the curve for the zero flow rate.

We can see how the network is able to correctly model the output pressure from the servovalve, given the inputs considered, and will therefore be used in the next chapter within the simulink model of complete actuator.

5.4.3 Diagnostic

We will now analyze the second use of the neural network addressed in this thesis, that is the diagnostics of the servovalve.

Initially we will provide in input to the neural network the positions of the spool, the flow rates and the pressures of a single map, in order to obtain covering and radial clearance related to it.

The databases used up to now, however, have been categorized point by point, which is not suitable for the purpose we are going to address in

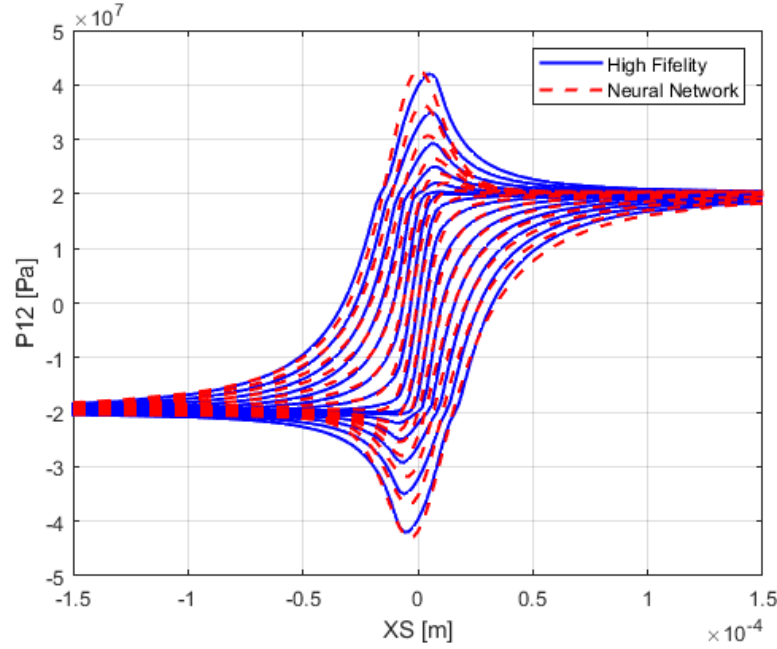


Figure 5.15: Neural Network with database 3

this paragraph. Trying to use the databases used for pressure modeling, modifying the labels and data matrices in order to obtain the inputs and outputs discussed above, unacceptable results are obtained with all the databases used so far, with errors from 30 % upward.

For the diagnostic use of the neural network we will therefore build the given input matrix in a different way, providing the individual pressure curves and the relative flow rate, removing the variable relating to the position of the spool as it is always the same.

To further improve the results of the network, database 4 with 900 combinations of games and coverings will be used, in order to allow the network to train better on their recognition.

The new data matrix will therefore have a size of 134x13500, with the rows given by the 133 pressures of a single curve plus the relative flow rate value, and the columns given by the 900 maps multiplied by the 15 flows of which each is composed.

Given the complexity of the input and output matrices, the training was carried out with a single hidden layer made up of 10 neurons. The scheme of the network is shown in Figure 5.18.

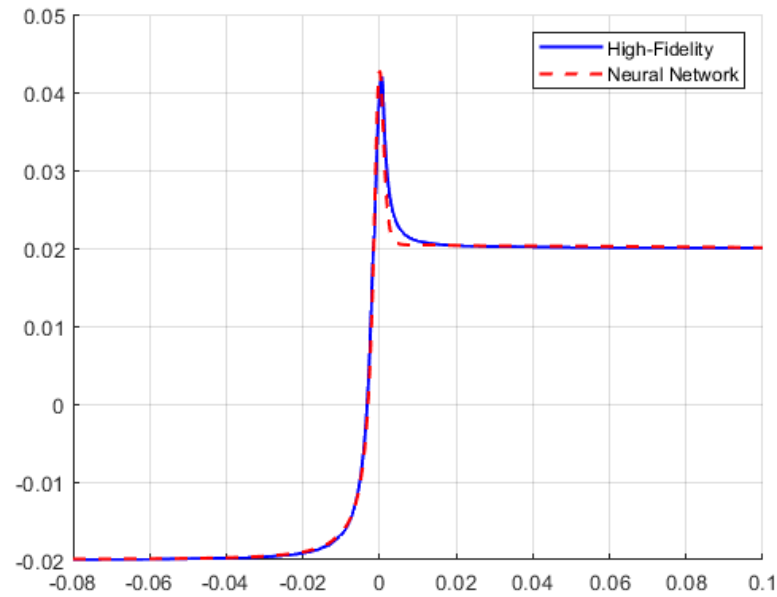


Figure 5.16: NN single Q

The results of the training with the new configuration of the given input matrix are contained in the range of 1% and therefore perfectly acceptable.

The simulations of the neural network for 10 radial clearance and coverage values are listed and compared with the original values below. In each row the radial clearance and then the overlap are shown.

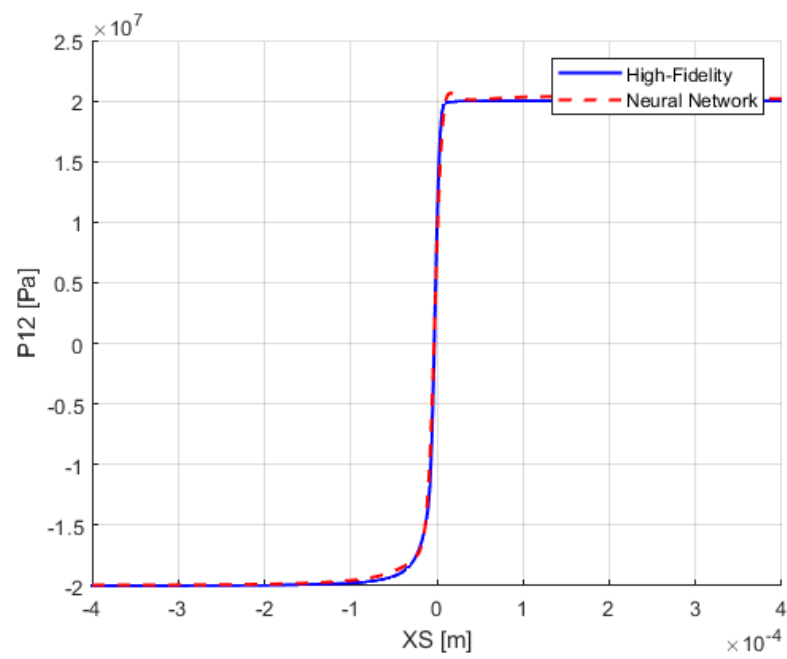


Figure 5.17: NN single $Q=0$

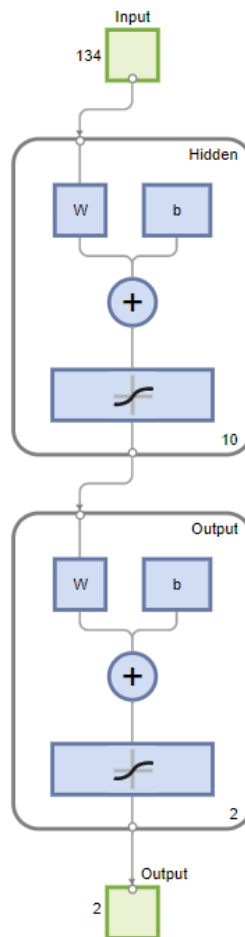


Figure 5.18: Neural Network scheme

Data	Prediction
-1.0000	-0.9844
0.5000	0.4832
-0.7931	-0.7827
0.4741	0.4689
-0.5862	-0.5841
0.4483	0.4491
-0.3793	-0.3787
0.4138	0.4224
-0.1724	-0.1740
0.3879	0.3904
0.0345	0.0335
0.3621	0.3596
0.2414	0.2421
0.3276	0.3186
0.4483	0.4449
0.3017	0.2974
0.6552	0.6534
0.2759	0.2767
0.9310	0.9282
0.5000	0.4832

Table 5.1: Covering and Radial Clearance prediction

Chapter 6

EHA testing

6.1 EHA genetic algorithm

In this paragraph we will insert the genetic algorithm obtained together with its coefficients inside a Simulink block in the form of a matlab code block, and we will replace it in the model of the complete actuator.

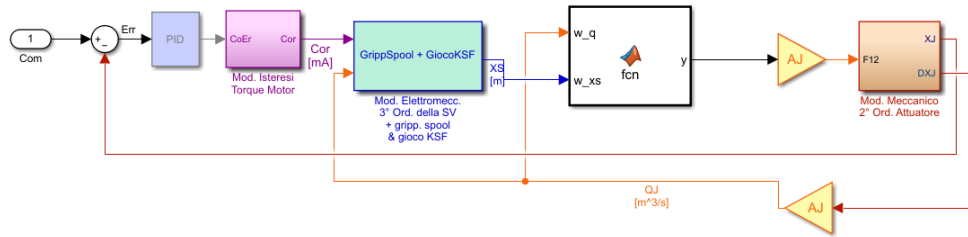


Figure 6.1: EHA model with genetic algorithm

6.2 EHA neural network

Having obtained good results in simulating the output of the pressure of the servovalve given the flow rate, the position of the spool and the wear values of the servovalve radial clearance and overlap, we want to see what can happen if we insert this modeling through neural network in the SIMULINK model of the complete actuator. To do this, the MATLAB suite allows us to create a Simulink block simply with the gensim(net) command, as shown in

Figure 6.2. This block inserted in the complete actuator's model is shown in Figure 6.3.

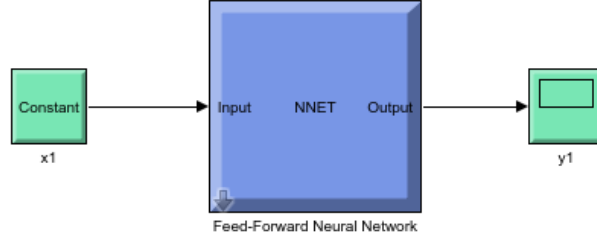


Figure 6.2: Result of gensim command

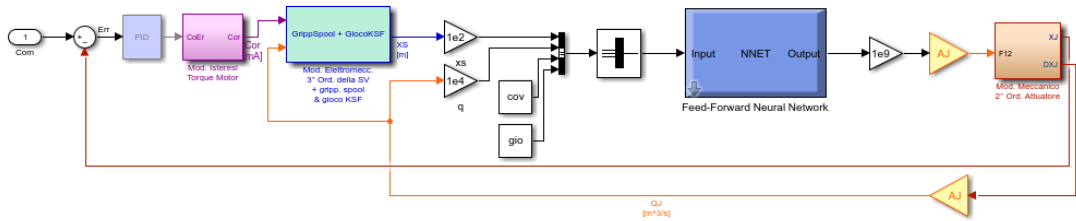


Figure 6.3: EHA model with neural network

6.3 EHA commands

In this section we will test the EHA models described so far (the high fidelity model, the genetic algorithm model and the neural network model) with different types of commands.

The commands, that are listed below, will be given to the three models (High-Fidelity, Genetic Algorithm, Neural Network), and their results will be compared through graphs of the output pressure over time.

Commands:

Step [mm] = 0.1 , 0.01

Ramp [mm] = 0.1 , 0.01

$\text{Sin [mm, Hz]} = 0.1 @5\text{pi} , 0.1 @10\text{pi} , 0.01 @5\text{pi} , 0.01 @10\text{pi}$

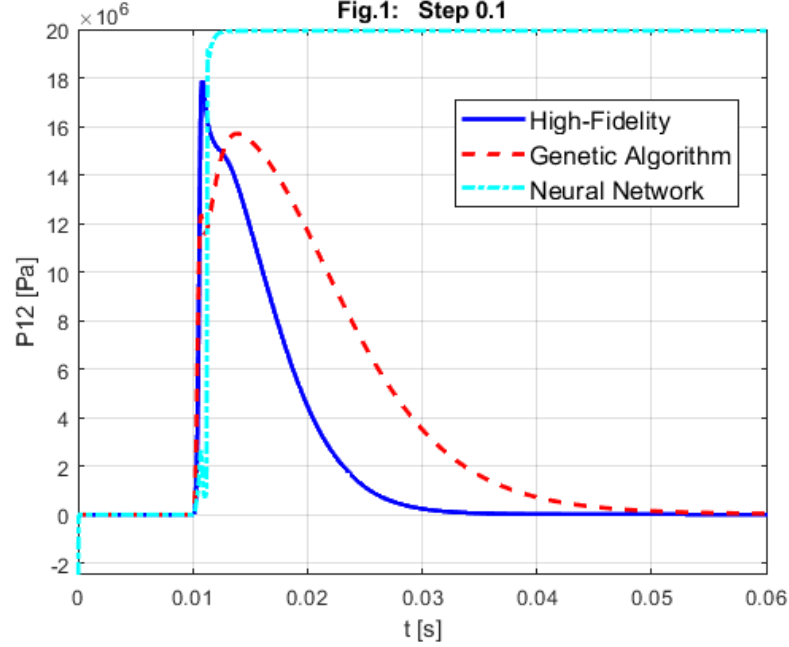


Figure 6.4: Step 0.1

As we can see from the graphs, the servomechanism model with the genetic algorithm is able to simulate the behavior of the High-Fidelity model in broad terms, even if with a certain degree of error both in the peak values and in the response times.

The neural network model, on the other hand, is not able to simulate the behavior of the actuator. Its trend mirrors that of an on/off, reaching the peak value of the command and keeping it. When there is a command inversion, the neural network model responds with a constant value change with a delay time compared to the High-Fidelity model

6.3.1 Simplified neural network

Being the insertion of the neural network into the complete actuator model and its simulation a more complex process than the modeling of the single maps of the servovalve, the ability of the neural network block simulink to address the problem was initially tested.

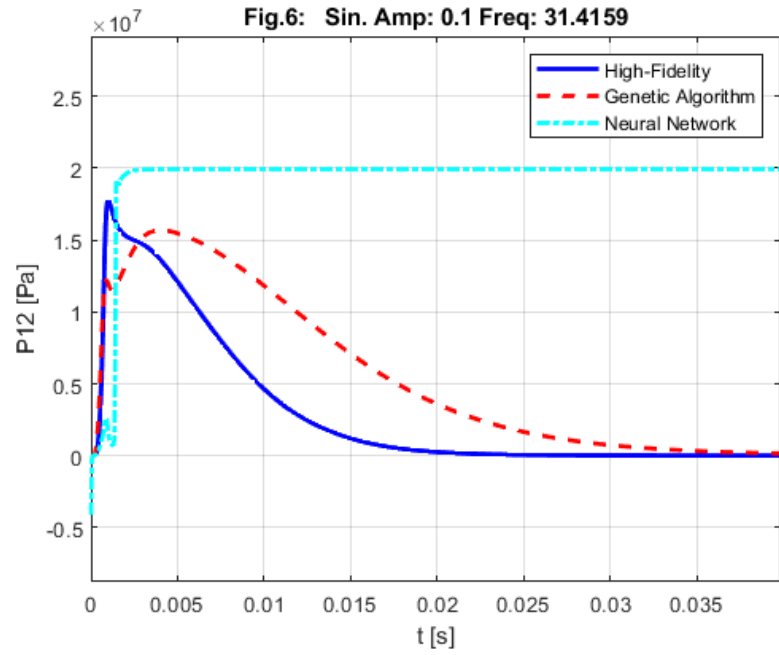


Figure 6.5: Sin 0.1 @ 10π

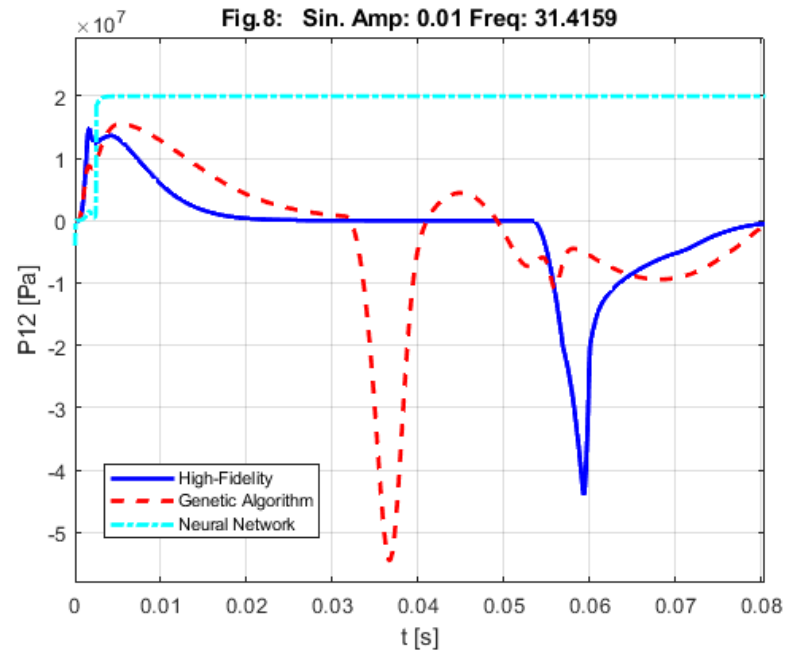


Figure 6.6: Sin 0.01 @ 10π

To do this, the block generated with the gensim function has been inserted in a simplified simulink model in which the values of XS and Q have been passed as inputs, instead of being taken in feedback as in the complete actuator model.

This was done to test the neural network's ability to perform what it was designed for even in the presence of a simultaneous variation of XS and Q . In the construction of the servovalve maps, in fact, only the value of XS was made to vary, maintaining a fixed flow rate each time.

As suspected, the neural network used so far is not able to simulate a simultaneous variation of XS and Q , and as shown in Figure 6.7 it does not give any output pressure results.

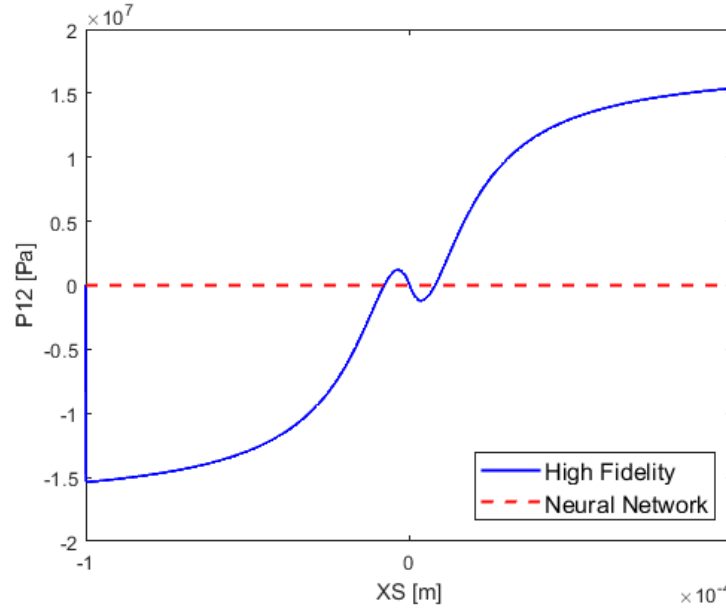


Figure 6.7: Simplified Neural Network Simulink Model with 4 input variables

To try to improve the performance of the neural network within the complete actuator model, it was decided to simplify it by removing the two variables of radial play and coverage, making the network more efficient. The database 5 is in fact composed only of the variables XS and Q , and the removal of the other two variables allowed to extend the range to 133 spool positions and 133 flow rates.

As we can see in Figure 6.8, with this new configuration the neural network

is able to simulate, albeit with a certain error, a simultaneous variation of XS and Q .

This new neural network block was then replaced in the complete actuator, with the results shown in the next paragraph.

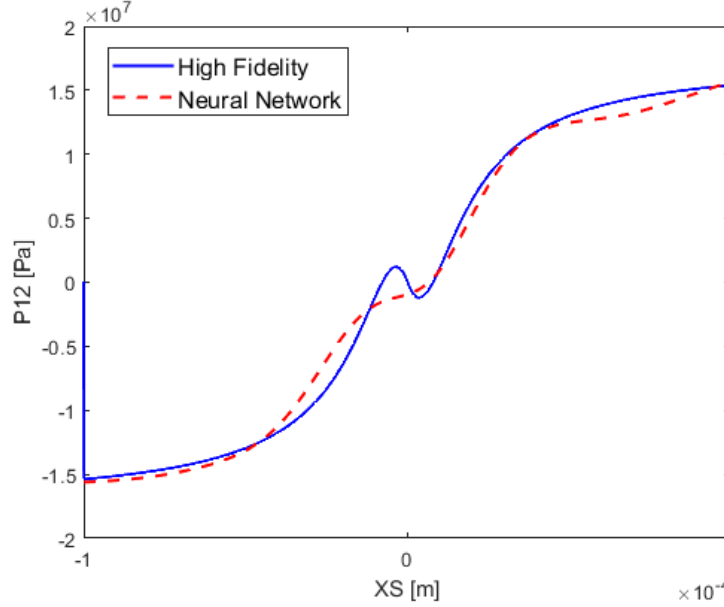


Figure 6.8: Simplified Neural Network Simulink Model with 2 input variables

6.3.2 Neural network model improvement

As we can see from the following images, by modifying the neural network we have obtained acceptable results for many of the input commands.

We can distinguish some types of behavior:

In some cases both the neural network and the genetic algorithm simulate the implementation with good consistency with the High-Fidelity model; this is the case of Step 0.1 shown in Figure 6.10, and Sin with amplitude 0.1 and frequency 2π shown in Figure 6.14. This usually happens when there is a positive pressure spike.

On the other hand, when the pressure of the High-Fidelity model peaks downwards, as in Figure 1, Figure 2 and Figure 3, an error of the neural network and of the genetic algorithm is noted both in the peak value and in the time of the impulse.

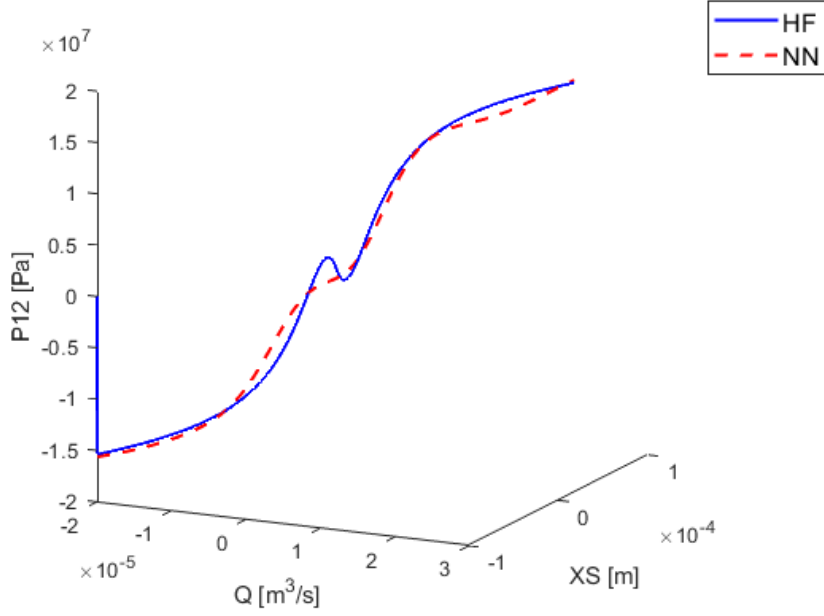


Figure 6.9: Simplified Neural Network Simulink Model with 2 input variables, 3D plot

We can also note that in some cases, as in Figure 6.12, the genetic algorithm has a damped oscillatory behavior typical of a second order system.

In general, however, we can say that the replacement of the High-Fidelity model of the servovalve has given consistent results that can certainly be improved in the future.

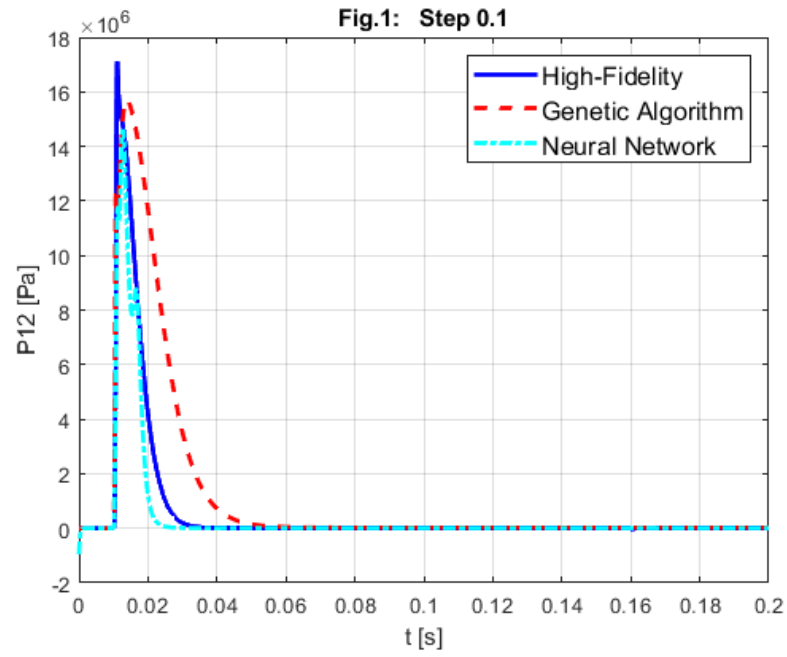


Figure 6.10: Step 0.1

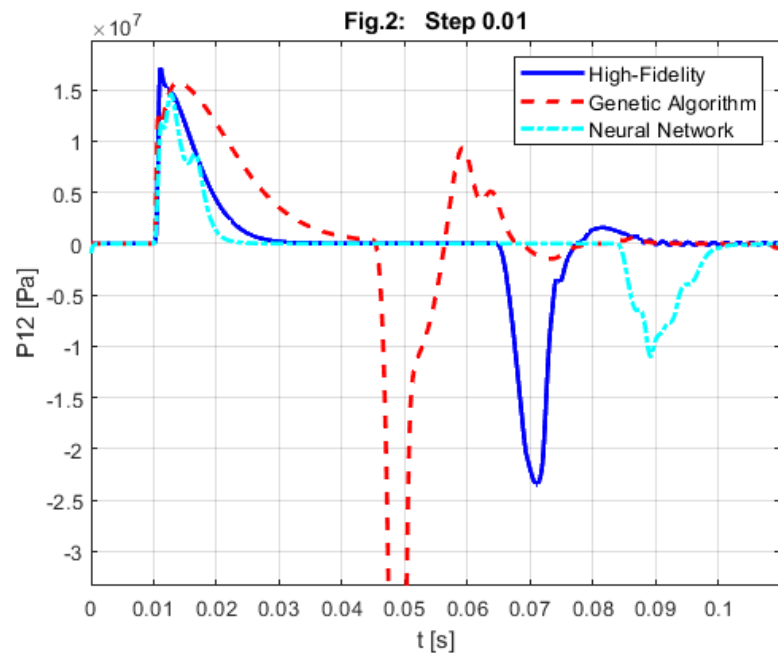


Figure 6.11: Step 0.01

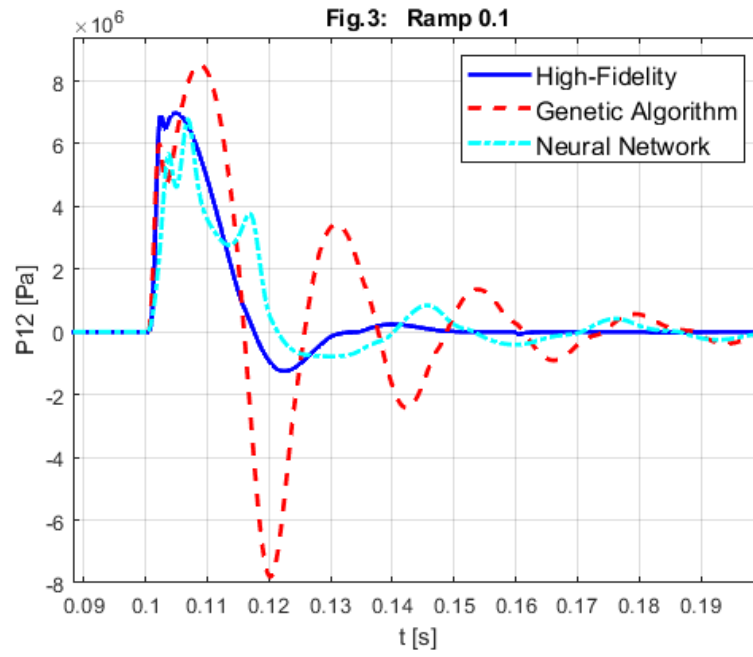


Figure 6.12: Ramp 0.1

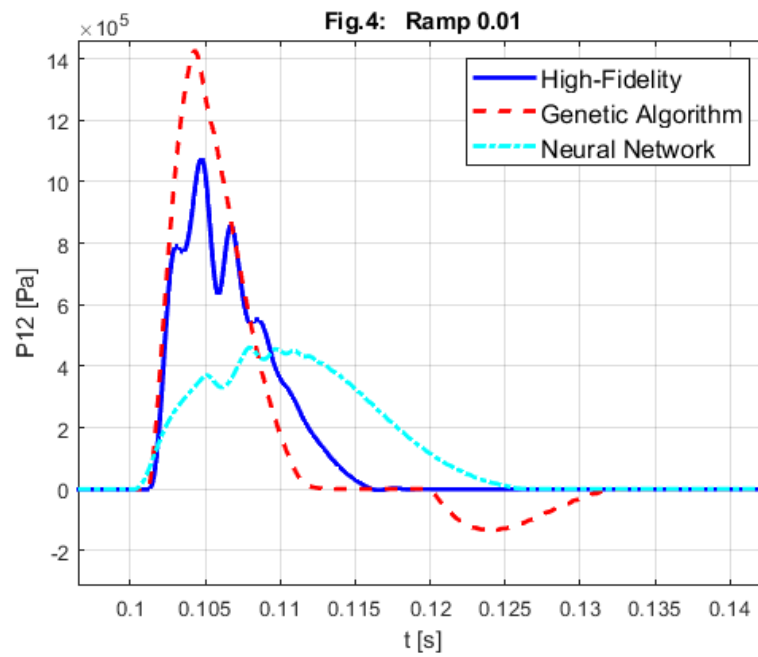


Figure 6.13: Ramp 0.01

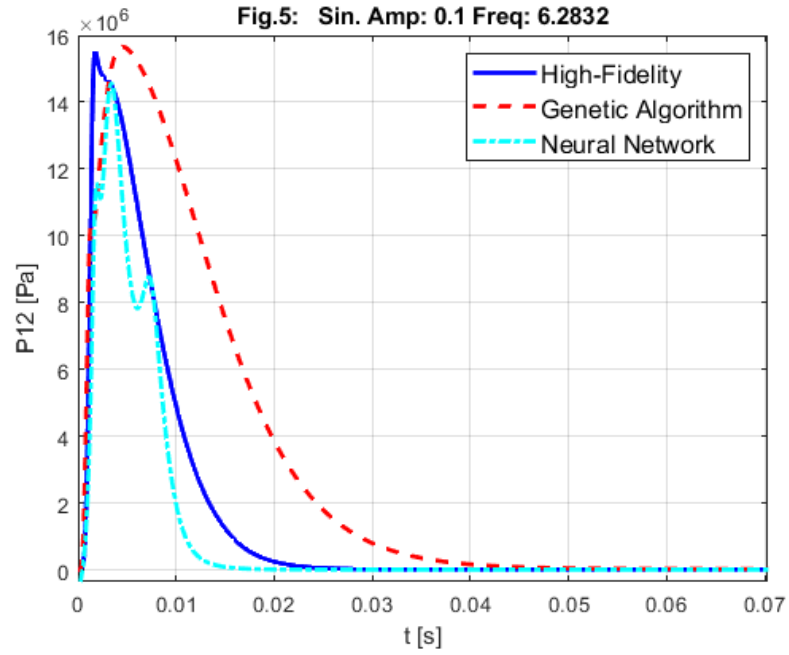


Figure 6.14: Sin 0.01 @ 2π

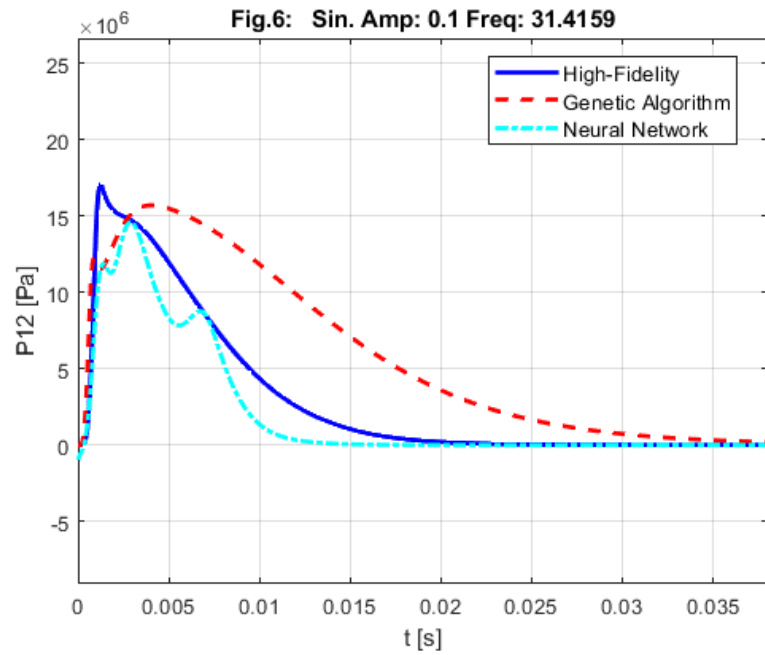


Figure 6.15: Sin 0.1 @ 10π

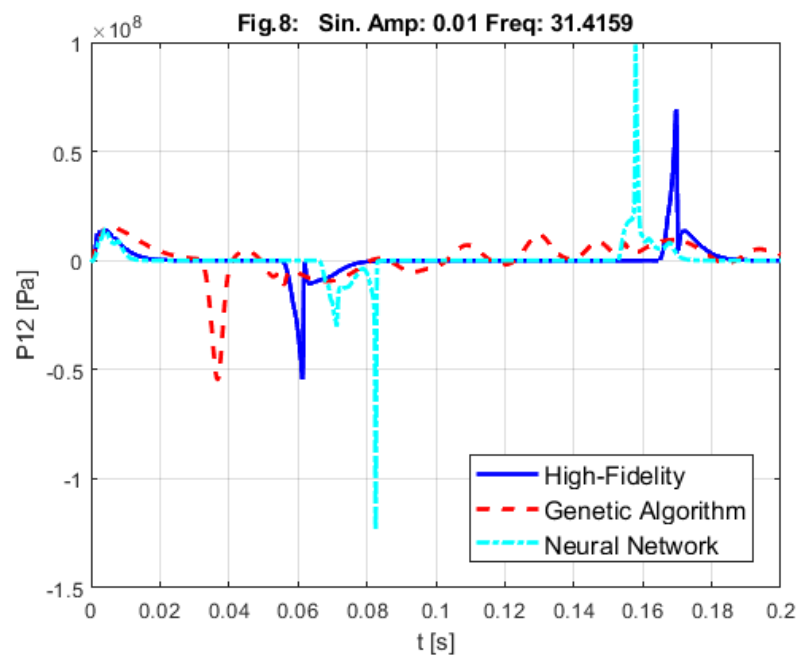


Figure 6.16: Sin 0.01 @ 10π

Chapter 7

Conclusions and future developments

The main objective of this thesis was to model the performance of an electrohydraulic servovalve. To do this, three different methods were compared, each with its own particularities.

The lookup table has proved to be an effective method but with too long calculation times to be used within the complete actuator model.

The genetic algorithm was based on the design of the objective function. Although the choice made in this thesis proved to be effective in simulating the characteristic maps of the servovalve, the implementation in the model of the complete actuator showed some difficulty in following the pressure trend. Its performance can certainly be improved through a different design of the objective function, particularly by introducing high coefficients with odd powers to the denominator of the function.

The neural network was used both as a method of modeling the performance of the servovalve, showing good results and being implemented within the model of the complete actuator, and also as a method of diagnostics of the servovalve, obtaining a method capable of recognizing the internal state of the servovalve by analyzing its pressure and flow rate.

The diagnostic results turned out to be effective, with 1% error rate. As a future development, this method could be tested with experimental data to test its operational capacity.

In the implementation of the genetic algorithm and neural network methods in the complete actuator model, there have been some issues regarding the method in which the neural network was conceived. Despite overcoming

these problems, the results of the actuator outlet pressure modeling still have errors both in simulating pressure peaks and their timing.

However, the applicability of these methods inside the complete actuator has been demonstrated, managing to simulate the pressure trend. Subsequent studies may focus on improving the simulation performance of the complete actuator.

The following table collects the results of the three methods of modeling the performance of the servovalve, represented by the error in Euclidean norm.

	Lookup Table	Genetic Algorithm	Neural Network
Error	5-10%	10-15%	1-5%

Bibliography

- [1] Borello, L., Dalla Vedova, M. D. L., Jacazio, G., Sorli, M. (2009). A prognostic model for electrohydraulic servovalves. In Annual Conference of the Prognostics and Health Management Society (pp. 1-12).
- [2] Dalla Vedova, M., D., L. (2007). Modelli Numerici Semplificati delle Caratteristiche Fluidodinamiche di Servovalvola. Tesi di dottorato, Cap. 5.
- [3] Dalla Vedova, M., D., L. (2007). Modello Numerico Particolareggiato delle Caratteristiche Fluidodinamiche di Valvola. Tesi di dottorato, Cap. 4.
- [4] Dalla Vedova, M., D., L., Maggiore, P. (2020). Advanced numerical modeling for systems engineering: theory and applications. Politecnico di Torino, Scuola di Dottorato.
- [5] Dalla Vedova, M., Jacazio, G., Maggiore, P., Sorli, M. (2010). Identification of precursors of servovalves failures for implementation of an effective prognostics. In Proceedings of the 4th International Conference on Recent Advances in Aerospace Actuation Systems and Components (pp. 111-126).
- [6] Fu, J., Maré, J. C., Fu, Y. (2017). Modelling and simulation of flight control electromechanical actuators with special focus on model architecting, multidisciplinary effects and power flows. Chinese Journal of Aeronautics, 30(1), 47-65.
- [7] Maggiore, P. (2020). Comandi di volo. ATA 27. Politecnico di Torino, Corso di “Sistemi di bordo aerospaziali”.
- [8] Romano, A. (2011) Modelli per la Prognostica delle Avarie Multiple di Comandi di volo Primari. Tesi di Laurea Magistrale.
- [9] Singh, S., Chand, A., Lal, S. P. (2013, September). Improving spam

- detection using neural networks trained by memetic algorithm. In 2013 Fifth International Conference on Computational Intelligence, Modelling and Simulation (pp. 55-60). IEEE.
- [10] Simi, A. (2007). Implementazione di algoritmi genetici multiobiettivo distribuiti in ambiente Matlab®.