Politecnico di Torino

Master's Degree in Aerospace Engineering

Academic year 2022/2023

Master's Degree Thesis

Analysis of aeronautical servo-actuator models with "Elements"

New multiphysics system simulation environment

Supervisors:
Prof. M. Dalla Vedova
Prof. P. Maggiore
Eng. D. Catelani

Candidate:
Filippo Drappero

July 2023

Abstract

The main objective of this Thesis is to analyse the numerical modelling of actuation systems for aeronautical applications, in particular for aircraft flight control. The Thesis involves the simulation of these system models in two different development environments, Elements and Simulink, with a comparison of the results. Elements is a brand new simulation tool, which was explored in depth during an internship at Hexagon MSC Software offices. Instead, the aim of this Thesis is to compare existing actuation system models developed in Simulink, a tool already established in the academic and industrial context, with the results provided by Elements simulations.

After a necessary introduction on the field of interest, such as Systems Engineering and the different simulation environments, a chapter focusing on the theory of flight controls and aircraft actuation systems is submitted.

Then, simple mass-spring-damper systems are introduced, with the purpose to explain with a practical example the main differences between Elements and Simulink. After that, a servo-actuator system is analysed from the point of view of Model-Based design, comparing simulation results from Elements with those from Simulink.

Before presenting the results and final considerations, the Thesis will present an interesting work applied to the case study of the previously illustrated actuation system, realised with co-simulation between two different software, exploiting multibody and multiphysics analysis. Indeed, with strong connections to other Hexagon products, such as Adams, and openness to other tools, Elements offers a powerful simulation experience that supports early decision-making and system integration activities in the Aerospace, Automotive, Robotics, Energy, Heavy machinery and other industries.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Systems Engineering

Systems Engineering is an interdisciplinary and integrative approach that uses systems principles and concepts, as well as scientific, technological, and management methods, to enable the successful realisation, use, and retirement of engineered systems. It combines a few techniques to ensure that the designed system meets all requirements. It focuses on a system's architecture, implementation, integration, analysis, and management throughout its lifecycle. It also takes into account the system's software, hardware, personnel, processes, and procedures.

Systems Engineering focuses on:

- Establishing, balancing, and integrating stakeholders' goals, purpose, and success criteria, as well as defining actual or anticipated customer needs, operational concept, and required functionality, early in the development cycle.

- Developing an appropriate lifecycle model, process approach, and governance structures while taking complexity, uncertainty, change, and variety into account.

- Developing and assessing alternative solution concepts and architectures.

- Defining and modelling requirements, as well as selecting a solution architecture for each phase of the project.

- Carrying out design synthesis as well as system verification and validation.

- While taking into account both the problem and solution domains, taking into account necessary enabling systems and services, identifying the role that the parts and their relationships play in the overall behaviour and

> performance of the system, and determining how to balance all of these factors to achieve a satisfactory outcome.

Systems Engineering facilitates, guides, and leads the integration of relevant disciplines and specialty groups into a cohesive effort, resulting in an appropriately structured development process that progresses from concept to production, operation, evolution, and eventual disposal. Systems Engineering considers both the business and technical needs of customers in order to provide a quality solution that meets the needs of users and other stakeholders, is fit for the intended purpose in real-world operation, and avoids or minimises unintended consequences. The goal of all Systems Engineering activities is to manage risk, which includes the risk of not delivering what the customer wants and needs, the risk of late delivery, the risk of over-budgeting, and the risk of unintended negative consequences. The degree to which such risk is reduced is one measure of the utility of Systems Engineering activities. In contrast, the level of excess risk incurred as a result of the absence of a System Engineering activity is a measure of acceptability (7).

Model-Based Systems Engineering (MBSE) is a formalised methodology used to support the requirements, design, analysis, verification, and validation phases of complex system development. In contrast to document-centric engineering, MBSE puts models at the centre of system design. The increased adoption of digital-modelling environments during the past few years has led to increased adoption of MBSE. In January 2020, NASA noted this trend by reporting that MBSE, "has been increasingly embraced by both industry and government as a means to keep track of system complexity."

MBSE in a digital modelling environment offers benefits that document-based systems engineering cannot. In a document-based approach, for example, many documents are created by various authors to capture the system's design from various stakeholder perspectives, such as system behaviour, software, hardware, safety, security, or other disciplines. A single source of truth for the system is built using a digital-modelling approach, in which discipline-specific views of the system are created using the same model elements.

A digital-modelling environment also creates a common standards-based approach to system documentation that can be programmatically validated to remove inconsistencies within the models and force all stakeholders to use a standard. This common modelling environment improves system analysis and reduces the number of defects commonly introduced in a traditional document-based approach. The availability of digitalized system data for analysis across disciplines allows for the consistent propagation of corrections and incorporation of new information and design decisions to all stakeholders. When MBSE is done correctly, the overall risk of development is reduced.

MBSE combines model and systems thinking.
A model is a simplified version of something (a graphical, mathematical, or physical representation of reality) that abstracts some of its complexities. This definition implies the use of rules or formality when simplifying, representing, or abstracting. A systems architect must model a system with less detail so that its structure and behaviour are obvious, and its complexity is manageable. In other words, models must adequately represent the system, and the system must validate the models.
Systems thinking is a way of considering a system not as a self-contained entity, but as a part of a larger system. Systems thinking is not the same as sticking to good plans, collecting statistics, or being methodical. The systems engineer observes the system from afar, investigates its boundaries, context, and lifecycle, takes notes on its behaviour, and identifies patterns. This method can assist the engineer in identifying issues (for example, a missing interaction, a missing step in a process, duplication of effort, or a missed opportunity for automation) and managing the complexity of a system. Although systems engineers must first break down and analyse the system (identifying parts and describing connections between them) systems thinking allows them to later synthesise the parts back into a coherent whole. Parts are not just connected to one another; they rely on one another to operate properly. This interconnectedness is emphasised by systems thinking. The activities of the system's subparts determine the system's behaviour. The systems engineer identifies feedback loops and causality patterns that may not be obvious at first by observing the system's interconnections. Systems thinking can help make issues more visible and easier to identify, balance the system, and manage the complexity of the system (8).

1.2 Elements: new digital-modelling environment

Systems simulations are tools for defining the inputs, outputs, and parameters of individual components as they interact with one another. So, instead of separately optimising each component, engineers can optimise the system as a whole using set functions that regulate the entire system. This can be achieved by the software modelling, computing, and evaluating integrated interdisciplinary and multiphysics systems using equations, lookup tables, 1D simulations, and other fast computing methods.

Elements, a brand-new simulation tool from Hexagon's Manufacturing Intelligence division, aids engineering teams in comprehending the behaviour of systems that are becoming more complex in contemporary products. Teams may quickly assess the effectiveness and viability of new design concepts using the software to inform more efficient product development and lower risk and cost.
Elements addresses the growing importance of multidisciplinary, integrated system development and end-to-end workflows. Customers across industries are demanding technologies that take into account multiple physics domains and address mechatronics challenges in a robust manner, fueling a growing trend towards system-level engineering. To accomplish this, all relevant engineering disciplines, control systems, and third-party supplier systems must be integrated to understand how the domains interact and how a design decision in one discipline affects others.

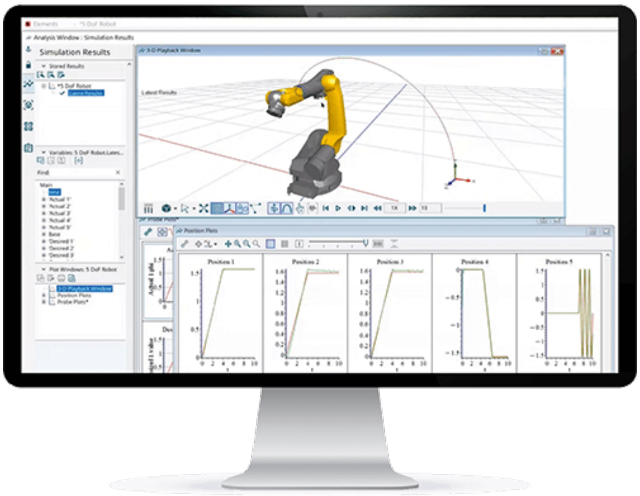


Figure 1.1: Elements Analysis Window Interface

To help engineers learn to use the Elements software, Hexagon streamlined the user interface (UI) by utilizing a drag-and-drop, flow-chart-based workflow to model physics, logics, systems and more.The model library used within this virtual environment is developed in the object-oriented and equation-based modelling language Modelica, which is an open and widely used standard for building customisable models. In fact, each component in the flow chart can be customized or selected from a variety of standard and user-made libraries of components. Moreover, such an acausal modelling language reduces the modelling effort and maximizes the flexibility and reusability.

Systems modelling is essential in the initial stages of design because it requires little data (often a CAD model, equation, or lookup table is sufficient, and a 3D mesh is not required). Elements speed up engineering development even further by ensuring that system-level models compute quickly and efficiently by optimising equations and code for fast calculation, but with no loss of fidelity. As the design evolves, and components and subsystems are detailed in specialised simulation software, the components can be routinely integrated into a larger system by adhering to the FMI standard and Hexagon's proprietary SmartFMU technology.

The traditional functional mock-up interface (FMI) standard allows information from one simulation software to be merged into another, resulting in one-directional or multidirectional co-simulation. However, the FMI/FMU appears as a "black box" to the various software options. The SmartFMU, on the other hand, provides users with access to schematics, documentation, and view/edit privileges. This makes the connection between the tools available to anyone with access to it. SmartFMU can have a greater influence on the design by increasing an engineer's visibility into the designs of other engineers. Whereas a traditional FMI/FMU must remain static as product designs iterate, the SmartFMU can be modified as each model changes.

Consider sharing information between an Elements and Adams simulation. Specifically, the Elements systems simulation can define a car's speed and engine output, which can then be fed into the Adams simulation to determine driving torque, which can then be fed back into Elements for further simulations. If either the Elements or Adams simulations are changed, whoever created the traditional FMI/FMU connection may need to create a new one, slowing the iterative process. SmartFMU, on the contrary, allows anyone with connection access to make the necessary changes, which helps to avoid resource bottlenecks, time-consuming switching between applications, and versioning issues.

Elements was developed in close collaboration with Maplesoft, leveraging its powerful maths engine to optimise system equations and provide Hexagon customers with fast, efficient simulation. Engineers in many industries benefit from new productivity-boosting solutions that significantly reduce simulation effort by combining the tried-and-tested Maplesoft system-level modelling technology with the extensive Hexagon CAE portfolio. Before committing designs to resource-intensive engineering processes, Elements allows engineers to quickly explore system behaviour and find better solutions to problems using simulation (12).

1.2.1 Elements and Simulink comparison

Simulink is a competitor simulation and model-based design software, also used for modelling, simulating, and analysing dynamic systems. Developed by MathWorks, Simulink is a popular choice for engineers and scientists looking to develop and test control systems for a wide range of applications. It allows users to build models using block diagrams and simulate them in real-time, making it a powerful tool for developing and testing complex systems. While both tools offer similar functionality, there are some key differences between them that are important to consider when choosing a simulation tool.

One of the biggest differences between Elements and Simulink is their user interfaces. Elements has a more modern and intuitive interface compared to Simulink, which can be more difficult to use for beginners. Elements has a streamlined workflow that makes it faster and more efficient to build and simulate models. In contrast, Simulink has a more complex interface that can be overwhelming for some users. While it offers a wide range of modelling and simulation tools, its interface can be cluttered, making it more difficult to navigate.

Another key difference between Elements and Simulink is the range of modelling tools available. Elements is a powerful multi-physics simulation software designed to provide a more comprehensive toolset for the analysis and optimization of complex mechanical systems, including those with multi-physics interactions. In contrast, Simulink is a valuable tool for modelling and simulating control systems, but it does not offer the same level of modelling capabilities as Elements. Although, it has a large library of pre-built blocks that can be used to create models quickly. While this is useful for some applications, it can limit the flexibility of the simulation tool for more complex systems.

Both Elements and Simulink offer strong simulation capabilities, although there are some differences between them. Elements is designed to handle large models with thousands of components and can run simulations in real-time. This makes it ideal for simulating complex systems, such as autonomous vehicles, where there are multiple sensors and control systems that need to be coordinated. In addition, Elements supports parallel processing, allowing for faster simulation times and improved productivity. The software also provides advanced visualization tools, enabling engineers to visualize and analyse simulation results in a user-friendly format. Simulink also has strong simulation capabilities, although it can struggle with larger models. It is better suited for simpler control systems, such as those used in industrial automation.

In conclusion, Elements and Simulink are both powerful simulation tools that can be used to model and simulate complex systems in the automotive, aerospace, and industrial automation industries. However, there are some key differences between them that are important to consider when choosing. Elements has a more modern and intuitive interface, a wider range of modelling tools, and is better suited for larger and more complex models. Simulink, on the other hand, has a simpler interface, a larger library of pre-built blocks, and is better suited for simpler control systems. Ultimately, the choice between Elements and Simulink will depend on the specific needs of the project and the level of complexity of the system being simulated.

Chapter 2

Actuation Systems for Flight Controls

2.1 Flight Controls

The motion of an aircraft is defined by its translational movement, in which it travels from one point to another in space and by the rotational movement around its three axes X, Y, Z which define pitch, roll and yaw as shown in Figure 2.1.

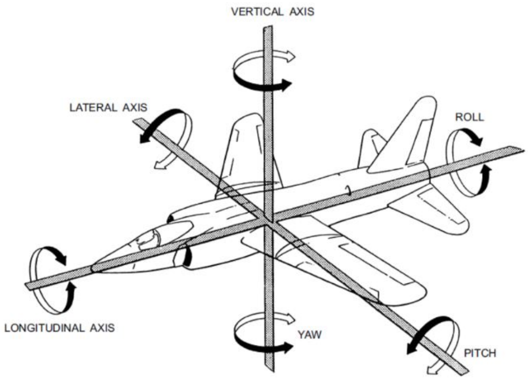


Figure 2.1: Vehicle control axes

Rotation around the axes just described allows control of the aircraft during all phases of flight. Pitch, roll and yaw movements occur through the movement of surfaces appropriately controlled by the pilot, which requires a thorough knowledge of aircraft aerodynamics and dynamics.

Aircraft flight control systems are the mechanisms that allow a pilot to control the movement and direction of an aircraft. These systems can be classified into two types, primary and secondary systems. In Figure 2.2 it is possible to identify the main control surfaces.

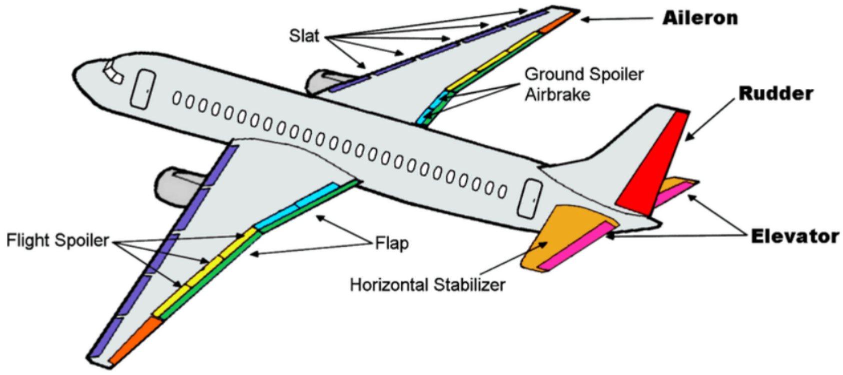


Figure 2.2: Main Flight Control Surfaces

Primary flight control systems are the primary means by which a pilot controls the movement of the aircraft by generating forces that induce a rotation of the aircraft in the plane. Primary flight commands must provide the pilot with relative force feedback proportional to the intensity of the high-frequency command. The loss of one or more primary flight controls can have disastrous effects, which is why they must be highly reliable. Traditional primary flight controls are:

- Ailerons: generate rotation around the longitudinal axis of the aircraft, control the roll motion by changing the angle of the wingtips.
- Elevators: generate rotation around the transverse axis of the aircraft, control the pitching motion by changing the angle of the horizontal stabilizer.
- Rudder: generate rotation around the vertical axis of the aircraft, control yaw motion by changing the angle of the vertical stabilizer.

Primary flight controls can be of the reversible or irreversible type:

- Reversible controls consist of a direct connection between the control lever and the control surface. They can be implemented by means of metal rods or with a system of non-rigid rope pulleys. With this type of control the pilot must compensate for the hinge moment with a suitable force; it is proportional to the size of the surface. The use of such controls can be problematic in the presence of large surfaces due to human force limitations.
- Irreversible commands are also called powered flight commands or position commands. They have the particularity of not requiring pilot action to compensate the hinge moment, but use line commands that transfer the position 'signal' to a servomechanism. This connection can be made mechanically (ropes or rods), electrically (fly-by-wire) or via optical fibre (fly-by-light). In these cases, it is necessary to provide the pilot with realistic feedback, i.e. to simulate the sensation of effort he would perceive with reversible

controls. A classic example of an irreversible flight control is shown in Figure 2.3, where it can be observed that it is realised by means of a hydraulic system that allows the oil to come under pressure in the servo-valve, which distributes it to the actuator chambers. chambers of the hydraulic actuator where the stem connected to the surface generates its movement.

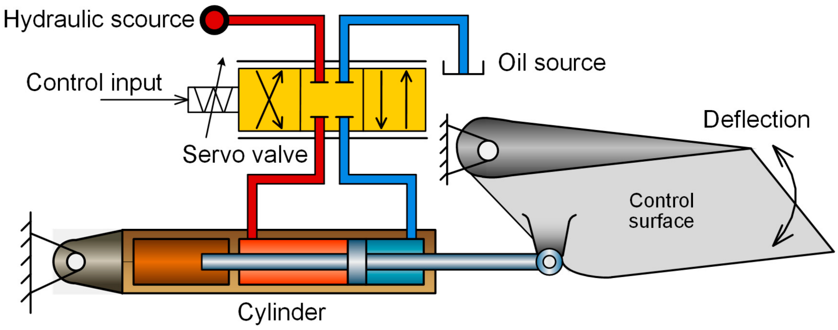


Figure 2.3: Irreversible (hydraulic) Flight Control

Secondary flight control systems are auxiliary components that assist in controlling the aircraft's flight path by supporting or augmenting the primary flight controls. These systems include flaps, slats, spoilers, and trim tabs. Flaps are located on the trailing edge of the wings and increase the lift and drag of the aircraft, allowing it to take-off and land at lower speeds. Slats are located on the leading edge of the wings and increase the lift of the aircraft at low speeds. Spoilers are used to decrease the lift of the wings, helping the aircraft to descend or slow down. Trim tabs are small surfaces used to adjust the primary flight controls' neutral position, helping to reduce the pilot's workload during extended flights.

The reliability and safety of primary and secondary flight control systems are critical for aircraft operations, and manufacturers must adhere to strict design and testing standards to ensure their proper functioning. Regular maintenance and testing of these systems are necessary to prevent failures and ensure their continued safe operation. Overall, both primary and secondary flight controls are essential for safe and efficient aircraft operations.

In modern aircraft, primary and secondary flight control systems are becoming increasingly complex, with the integration of advanced technologies such as fly-by-wire systems. These systems use electronic signals to replace traditional mechanical linkages, providing the pilot with more accurate feedback, greater precision and improved speed in terms of command response. These electronic systems are designed to supplement the primary and secondary flight control systems, providing an extra layer of safety and redundancy, but also reduced weight and easier maintenance.

2.2 Actuation Systems

Actuation systems are essential components in the design of modern large aircraft, providing precise control of critical systems such as flight control surfaces, landing gear, and engine components. In general, these systems measure their outputs and use them in a feedback loop to achieve the desired target, constantly updating the error value between command and current status (position, speed, force, torque, temperature, pressure, or electrical magnitude). The main solutions will be described briefly.

Flight controls, on modern aircraft, are usually operated in hydraulic or electric way:

- for Primary controls hydraulic actuation is used, with linear cylinder-piston motors or linear motors;
- the Secondary controls can be actuate using hydraulic or electric rotary motors.

During last decades, because of the development of reliable electric motors (with a high power/weight ratio) and the consolidation of innovative technologies, secondary electromechanical controls have established in the field of small powers (typically $<$ 7 kW) and they begin to be uses also for the implementation of primary commands, especially for UAVs and small aircrafts.
In this way it will be possible completely substitute hydraulic power with the electric one, which for now is starting to apply in actuators of EHA (Electro-Hydraulic Actuator).
The robustness, reliability and safety of actuation systems are critical for aircraft operations, and manufacturers must adhere to stringent design and testing standards to ensure their proper functioning. As the aerospace industry continues to evolve, the development of innovative actuation systems will remain a key area of focus for improving the performance and safety of aircraft.

Common servomechanisms used in aircraft's environment which will be describe in detail on paragraphs below are:

- Electromechanical actuation (EMA)
- Electro-Hydraulic actuation
- Hydromechanical actuation
- Electro-hydrostatic actuation (EHA)

Electromechanical Actuators

Electromechanical Actuation (EMA) shifts aircraft design towards the philosophy of "More Electric Aircraft," with increased safety, efficiency, cost savings, and significant pollution reduction. The hydraulic circuit in EMAs is completely replaced by mechanical solutions, typically reducers (both gearbox and nut-screw types), which eliminate the possibility of leakages: these issues are frequently difficult to detect and difficult to repair due to the complexity and limited accessibility of the fluid circuit.
The figure below depicts a possible simplified EMA structure for aerospace applications.

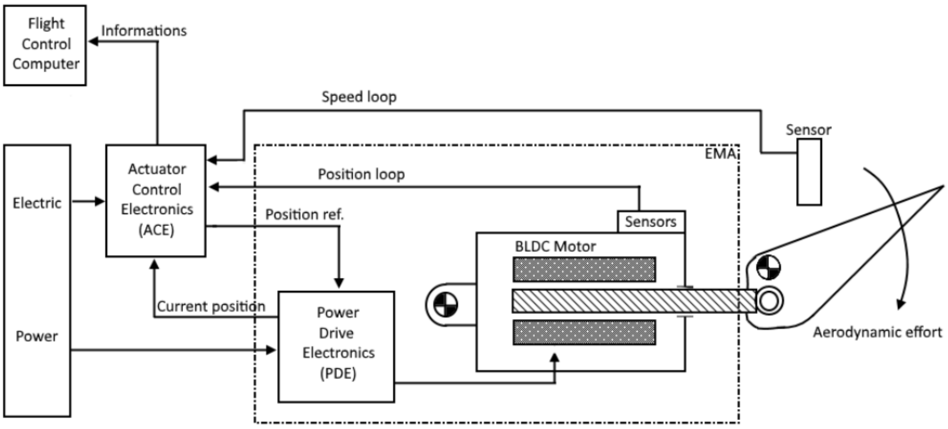


Figure 2.4: Electromechanical actuator scheme

The Actuator Control Electronics (ACE) is the main controller of the system and performs all calculations required to keep the error as close to zero as possible; all feedbacks collected by the sensors arrive in this device. ACE sends the reference position to the Power Drive Electronics (PDE) after calculating the current state of the end-effector using position and speed loops. PDE is typically made up of a three-phase inverter bridge, and its goal is to provide the correct power flow to the electric motor in order for it to reach the reference position.
The motor is primarily a brushless direct current (DC) motor that converts electrical input into mechanical rotational output for the reducer's gears. Motors in EMAs structures are typically very small, light, and fast for this task. A gear reducer is used to increase torque and decrease transmitted speed, which is usually followed by a ball or roller screw (Figure 2.5) that can convert a rotary motion into a translational one.

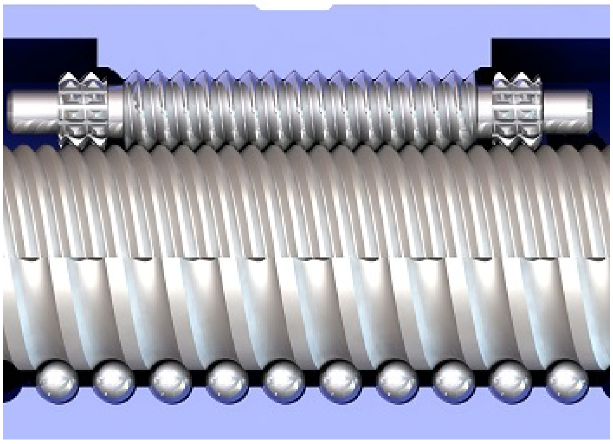


Figure 2.5: Ball screw and Roller screw comparison

A rolling body (spheres for the ball screw and shaped cylinders for the roller screw) is located between the rotating shaft and a nut in these devices. The friction is clearly very low, the loads can be very heavy, and it has a high wear resistance; however, there are some issues with the plays. The translation of these mechanisms allows the end-effector to move: in this simple example, it is represented by a flight control surface to which an aerodynamical effort is applied. Sensors are another important component of the system: they are required to detect the system's current state (position, speed, torque, force, and so on).

Electromechanical actuators are already used in military aircraft or secondary flight controls; their application to primary flight control requires some critical issues to be addressed. For starters, they have a very complex structure that necessitates a thorough understanding of kinematics and redundancy design; additionally, to increase efficiency and avoid jam problems, a different BLDC motor is required for each surface, but this increases the possibility of a critical failure. Another significant issue is the ability to maintain a determined position after the application of a heavier load: if the motor speed is close to zero and the torque is high, the joule effect dissipates all of the current given to the stator. An irreversible transmission could solve this problem.

Electro-Hydraulic Actuators

Nowadays, with large aircraft, the control of the primary flight surfaces is done with hydraulic actuators due to their simple and very efficient design. Electrohydraulic servomechanisms are the most used actuation system on modern civil and military aircraft where mechanical feedback is replaced by a fly-by-wire control system: replacing the mechanical line with an electric one to considerably reduce the weight and bulk on the aircraft. Here, trough pressurized fluid, power hydraulic is provided by Flight Control Computer signal of fly-by-wire architecture and not directly by pilot action.

The operating principle is based on the use of a three-phase power supply for the electric motor, which in turn drives a bi-directional, variable-speed hydraulic pump, which allows the hydraulic jack to move. The control comes from the fly-by-wire line, enters the ACE (Actuator Control Electronics) which is used as a comparator of the feedback signal coming from the LVDT, the signal thus compared provides the system error which is processed by the PDE (Power Drive Electronic) which is used to determine the necessary power supply to be sent to the variable speed motor, thus determining the direction and speed of rotation of the pump which in turn will generate the fluid flow rate and the pressure variation in the flow and return line.

The hydraulic line mainly consists of a low-pressure accumulator and valves that ensure that the maximum pressure that the system can reach is not exceeded.

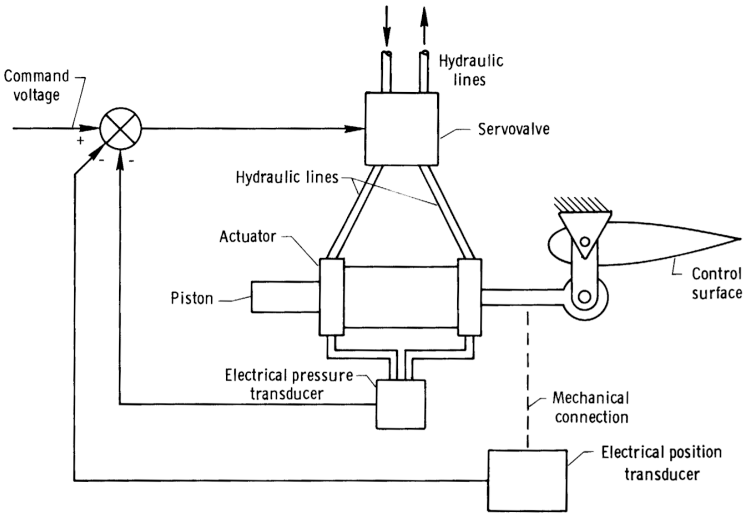


Figure 2.6: Electro-Hydraulic actuator scheme

The component that regulates hydraulic oil is the servo valve, which, in its simplest form laminates the fluid. The most adopted servo valves are:

- two stage "Flapper-Nozzle" servo valves (Figure 2.7);
- three stage "Jet pipe" servo valves (Figure 2.8);
- "Direct drive" valves DDV (Figure 2.9).

In Flapper-Nozzle servo valve there is an electrical component, called torque motor, which transmits the command instead of a mechanical lever. Electric torque motor moving the flapper, which is the first stage hydraulic valve and in this way the amount of fluid that flows from the ducts, so also the pressure,

changes.
Pressure regulated by the flapper acts on the spool of the second stage valve, distributing pressure to the actuator jack. The spool position is returned to the flapper mechanically through the feedback spring, while the piston position is measured by a Linear Variable Differential Transducer (LVDT) and transmitted to the control electronics.

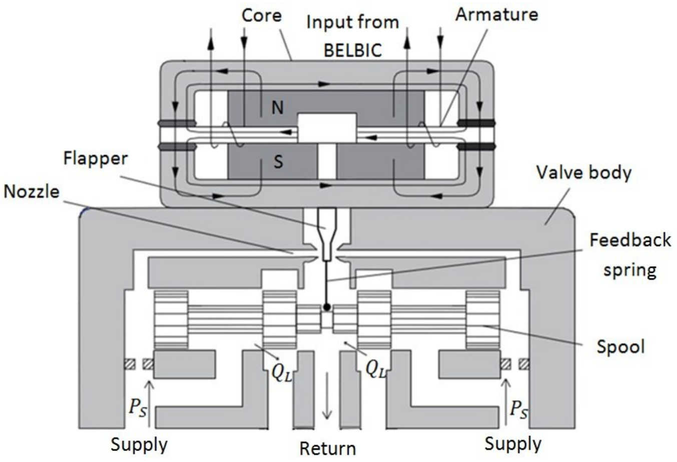


Figure 2.7: Flapper-Nozzle valve

Compared to Flapper-Nozzle, in Jet pipe servo valve a torque motor of a bigger entity directs a suitable flow of oil into two opposite chambers, capable of causing the spool to move due to the received overpressure. In this case, an oil flow is not deviated, but the efflux channel called jet pipe is appropriately oriented. Also, in this electrohydraulic system there is not direct feedback between the control valve and the actuator: the absolute position of the piston is surveyed by a Linear Variable Differential Transducer (LVDT) and sent to the control electronics.

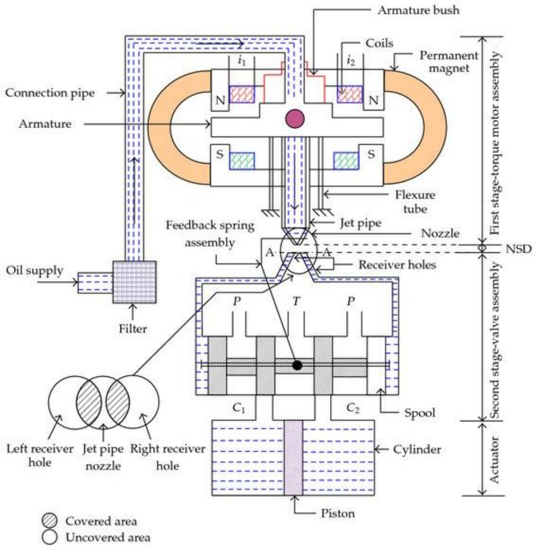


Figure 2.8: Jet pipe valve

Respect to Flapper nozzles, Jet pipe valve:

- has better performance;
- uses high-speed jet of orifice to change pressure into kinetic energy that is obtained in two receiver holes, then to control this energy for spool action;
- mouth is big, so debris or dirty particles cause small harm.

Finally, there are Direct Drive Valves (DDV) which allow to eliminate the system based on torque motor or jet, since the spool is moved directly by a dedicated electric motor, and they are the last frontier in the field of servo valves.

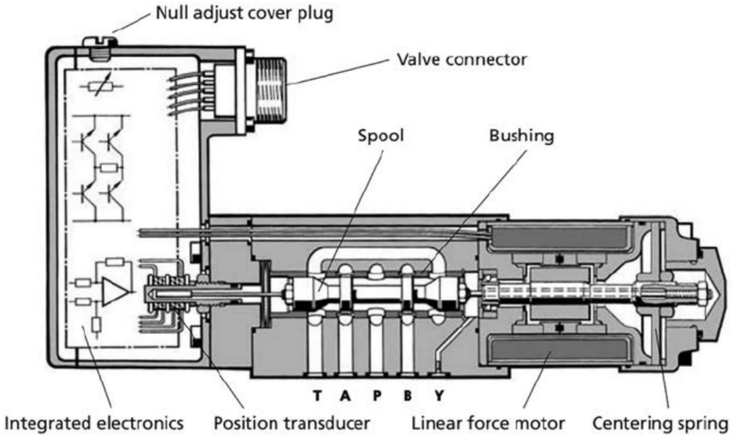


Figure 2.9: Direct drive valve

Electrohydraulic actuator is commonly used due to its high reliability and performance. Hydraulic power provides a high level of force and torque, allowing for responsive control of the aircraft's flight surfaces and high positioning efficiency. The electronic control system provides accurate control, ensuring that the aircraft responds exactly as intended by the pilot or flight control system.

They also offer a high flexibility, with the ability to be used in a wide range of flight control applications: they can be used in a variety of flight surfaces, including ailerons, elevators, and rudders.

This kind of actuator system is able of producing very high forces without backlash, but, in the other hand, it necessitates of a centralized hydraulic network that should be maintained at constant pressure (3000÷5000 psi) by hydraulic pumps relentlessly draining energy from the engines. The continuous consumption of energy leads to the heating of the hydraulic fluid, which then required a cooling system to maintain an acceptable temperature of the hydraulic fluid. The central hydraulic network also needs a system of pipes to deliver pressurized hydraulic fluid to actuators distributed throughout the aircraft, adding extra weight and occupying space. The large hydraulic network increases the risk of leaks and requires a large volume of hydraulic fluid. Traditional electrohydraulic actuators have remarkable power density (kW/kg) equipment wise but low power density when considering the power distribution network.

Hydromechanical Actuators

The complexity and weight of flight control systems greatly increase with the size and performance of the aircraft, but the use of hydromechanical actuators helps overcome these limitations. With hydraulic flight control systems, the size and performance limitations of the aircraft are due to economic reasons and not the pilot's muscular strength. In the early days, only partially assisted systems were used, so the pilot still had the feeling of aerodynamic loads on the control surfaces.

The simplest hydromechanical flight control system is mainly represented by a mechanical circuit, which connects the cockpit controls with the hydraulic circuits, composed by rods, cables, pulleys, and sometimes chains; and a hydraulic circuit, with hydraulic pumps, tanks, filters to maintain the fluid clean, pipes, valves (such as a selector valve, to control the direction of the flow, and a relief valve, to drain the excessive pressure in the circuit) and hydraulic actuators.

They could be classified as:

- Position hydraulic servo-system (with or without mechanical feedback);
- Speed hydraulic servo-system;
- Force hydraulic servo-system.

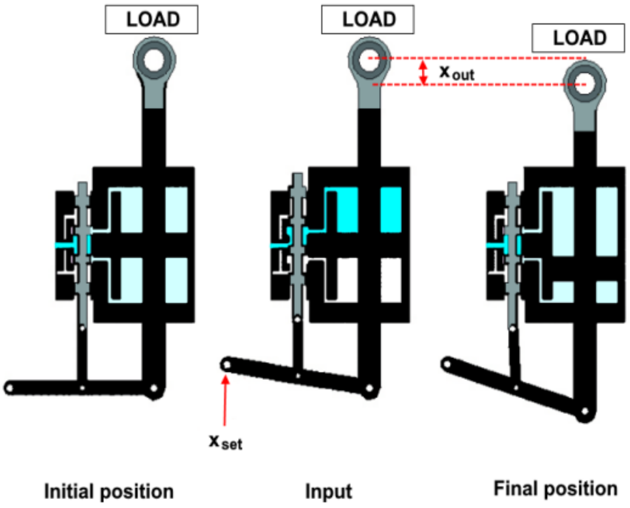


Figure 2.10: Hydromechanical actuator scheme

Pilot's command is transmitted via a mechanical linkage to a hydraulic valve, whose activation actuates the control surface. The three-centre lever consists to

a mechanical proportional controller because it multiplies the position error by a factor dependent by the lever's geometry. Then, this system is composed by a hydraulic cylinder and a proportional control valve.
A perturbation on the three-pivot control lever (x_{set}) causes the movement of the valve's spool: in this way one chamber of the cylinder and a tank (or generically the supplier) are connected. Fluid flows in the superior chamber of the cylinder, moving downwards the actuator's shaft and generating the displacement (x_{out}). At the initial and final position, when the input and output variable are steady, the valve closes all possible holes and maintains blocked the servosystem.

This solution is reversible: changing the direction of the input, so the direction of the spool motion, actuator's shaft will move towards the other side. If the shaft must sustain a greater aerodynamic load, the valve's spool opens a duct and supplies the needed pressurized flow to support it.
A system is reliable if is also insensible to an eventual disturbance; for a position control the disturbance is represented by a force. Hence, if the shaft must sustain a heavier load, the valve's spool opens a duct and supplies the needed pressurized flow to bear it: in this way the system has also a disturbance rejection.

This type of servomechanism is widely used due to its simplicity and reliability, it can be found on several operational aircraft.
Hydromechanical actuators can provide a large amount of force and torque, making them suitable for use in various mechanical systems in the aerospace industry. They also have a long service life, with minimal maintenance requirements, making them a cost-effective solution for various applications.
Another advantage is their ability to operate in a wide range of environmental conditions. They can operate at high altitudes, in extreme temperatures, and in high vibration and shock environments, making them suitable for use in a variety of aerospace applications. In conclusion, they are also highly compatible with modern control systems, which use digital signals to control the mechanical system, so these actuators can be easily integrated with the control system, providing precise and accurate control of the mechanical system.

Electro-Hydrostatic Actuators

In recent years, the will to make flights cheaper and greener have driven the aerospace industry to develop more electrical actuation system to implement in power-by-wire networks. Two examples of this philosophy are electro-hydrostatic (EHAs) and electromechanical actuators (EMAs). Nowadays, in civil transportation, EHAs and EMAs are widely used, both for first and secondary control systems. Electro-hydrostatic actuators are a power-by-wire type of motors that execute the movement of the control surfaces exploiting a localized hydraulic power obtained by the electrical power. These actuators could be divided in several groups according to their difference in control modes:

- Fixed Pump displacement and Variable Motor speed EHA (FPVM-EHA);
- Variable Pump displacement and Fixed Motor speed EHA (VPFM-EHA);
- Variable Pump displacement and Variable Motor speed EHA (VPVM-EHA).

In this paragraph are described only the FPVM-EHA because is the most popular thanks to its structural simplicity and high efficiency.

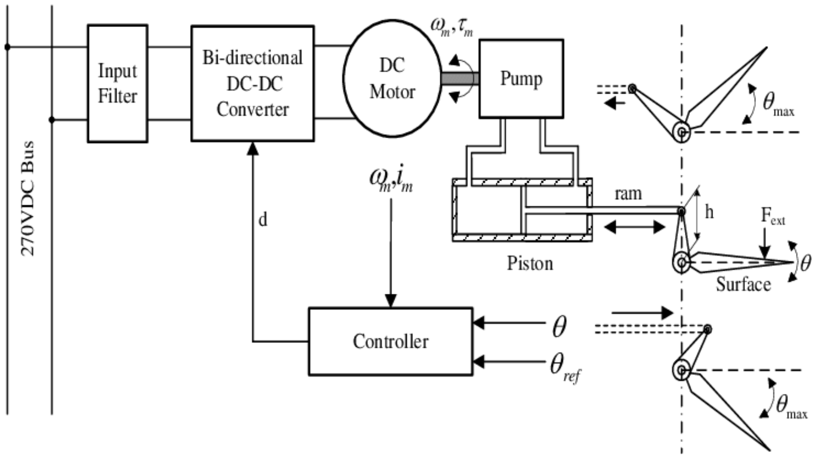


Figure 2.11: Electro-Hydrostatic actuator scheme

Thanks to the electro-hydrostatic actuators, it is possible to make the action on the controls even more prompt and dynamic, preserving the artificial sensitivity returned to the pilot. EHA also ensures structural simplicity and high efficiency. This system is composed by a servomotor variable speed (usually a Brushless DC) controlled by electronics drives, a bi-directional pump with variable speed and a linear or rotary actuator.

The electric machine transforms electrical power into mechanical power, and the axial piston pump generates the controlled pressure which is used by the actuator to control the surface movement. The fluid in the circuit is stored in a proper reservoir, composed by a low-pressure tank and two check valves able to maintain the minimum pressure required by the system. Close to reservoir there is a by-pass valve electronically controlled and two relief valves arranged for safety of the system. The pressurized oil in the circuit it's not provided by the on-board main hydraulic system, but the pump takes the needed fluid from the tank. For this type of actuator, the feedback ring needs a position sensor placed on the shaft.

This is not the best structure if the aim is to decrease aircraft's weight, but it's very effective to reduce the critical issue related to a centralized hydraulic system.

Furthermore, it allows maintenance and fuel cost savings while it assures the same precision and reliability of a traditional hydraulic system. Pilot's input and FCS input travel on the electrical cable of the fly-by-wire architecture, which is again lighter in term of weight with respect to a hydraulically driven signal transmission system.
Another advantage, other than reduced weight, is the absence of friction losses or delays due to the flexibility typical of reversible mechanical control lines. The command given by the pilot is than compared to a feedback signal measured by a Linear Voltage Differential Transducer LVDT or a Rotary Voltage Differential Transducer RVDT. The feedback signal of motor angular velocity is used to determine the exact amount of power needed. The error signal is then amplified and, filtered from noises, is used to control the actuation group.

The integration of EHA into aircraft systems requires careful consideration of the electrical, hydraulic, and mechanical components and their interactions. The control electronics must be carefully designed to ensure stable and precise control, while the hydraulic components must be selected to provide the necessary flow and pressure for the application. Additionally, safety and reliability considerations must be considered, with redundant systems and fail-safe mechanisms employed where necessary.
Overall, EHA represents an exciting advancement in aerospace actuation technology, providing precise and efficient control of critical aircraft systems. As the aerospace industry continues to evolve, EHA will likely play an increasingly important role in the design and operation of modern aircraft.

Chapter 3

Systems Modelling

3.1 Elements Overview

Elements is a modelling environment that enables users to build and simulate sophisticated multidomain physical systems. It allows for the development of component diagrams that graphically represent physical systems. Elements automatically generates model equations from a component diagram and executes high-fidelity simulations that represent the behaviour of a physical system employing advanced symbolic and numeric capabilities.
Elements can be used to create models that combine components from multiple engineering disciplines into a full system. It provides a large library of modelling components such as electrical, hydraulic, mechanical, pneumatic, and thermal devices, as well as sensors and sources and signal blocks. Customised components may also be created according to the requirements of modelling and simulation. Elements has a number of ready-to-use tools to assess model equations performing advanced analysis tasks like parameter optimisation. The models can also be translated into C code and exported into other tools and apps, such as those that support real-time simulation.

3.1.1 Physical Modelling in Elements

Physical modelling, also known as physics-based modelling, uses mathematics and physical laws to define the behaviour of a single engineering component or a system of interacting components. Although most engineering systems feature dynamics, the behaviour is generally characterised using Ordinary Differential Equations (ODEs). Elements offers the following capabilities to aid the quick and simple modelling process (13):

- **Topological or "Acausal" System Representation**
  Traditional modelling tools use a signal-flow method that needs system inputs and outputs to be explicitly stated. Elements, on the other hand,

makes it possible to link interrelated components using a topological model without being concerned about how signals flow between them.

- **Mathematical Model Formulation and Simplification**
  The symbolic capability of Elements automates the creation of system equations by mapping a topological representation to its mathematical form. Many mathematical simplification strategies occur when Elements formulates the system equations to remove any redundant equations. The expressions are then combined and reduced by the simplification tools to provide the minimal set of equations needed to describe a system without losing fidelity.

- **Advanced Differential Algebraic Equation Solvers**
  Algebraic constraints are introduced in the topological approach to model definition. Problems that combine ODEs with these algebraic constraints are called Differential Algebraic Equations (DAEs). The complexity of the DAE problem can vary depending on the nature of these constraints. The DAEs index provides a measure of the problem's complexity, so it increases with the index of the equations. The development of generalised solvers for complex DAEs is a hot topic in the symbolic computation domain. Elements solves high-index DAEs with innovative DAE solvers, that include cutting-edge symbolic and mathematical techniques.

- **Acausal and Causal Modelling**
  Actual engineered assemblies, such as motors and powertrains, are made up of a network of physically interacting elements. Block diagrams are frequently employed in development tools to model them. The lines connecting two blocks imply that they are linked by physical laws. When represented by software, block diagrams may either be causal or acausal.

  Many simulation tools are restricted to causal (or signal-flow) modelling. In these tools, a unidirectional signal, which is essentially a time-varying number, flows into a block. The block then performs a well-defined mathematical operation on the signal and the result flows out of the other side. This approach is useful for modelling systems that are defined purely by signals that flow in a single direction, such as control systems and digital filters. This approach is analogous to an assignment, where a calculation is performed on a known variable, or set of variables, on the right-hand side and the result is assigned to another variable on the left: $y := f(x)$.

  Modelling how real physical components interact requires a different approach. In acausal modelling, a signal from two connected blocks travels in both directions. The programming analogy would be a simple equality statement: $y = f(x)$.

The signal includes information about which physical quantities (for example, energy, current, torque, heat and mass flows) must be conserved. The blocks contain information about which physical laws (represented by equations) they must obey and, hence, which physical quantities must be conserved. Elements allows to use both approaches: a physical system (with acausal modelling) can be simulated together with the associated logic or control loop (with causal modelling) in a manner that suits either task best. When using the acausal modelling approach, it is useful to figure out the through and across variables of the component to be modelled. In general terms, a through variable outlines the flow of a conserved quantity and also establishes the flow direction for the sign convention of that quantity, an across variable indicates the driving force in a system.

3.2 Elements vs. Simulink Modelling

Before to discuss actuator systems on Elements, it's important to compare Elements and Simulink development environments, starting with the representation of two simple second-order dynamical systems:

- Mass-Spring-Damper system (1-DOF);
- two Mass-Spring-Damper system, or Harmonic Oscillator (2-DOF).

The first is a simple physical system used for the initial approaches to modelling while the second, as a first approximation, it can be representative of different mechanisms like a hydraulic jack adapted to move a mobile surface, subjected to any external load, through a yielding shaft.

3.2.1 Mass-Spring-Damper System

We will analyse a mechanical system of the MCK (Mass-Spring-Damper) type, i.e. a typical example of a linear system with second-order dynamics.
This system, apparently too simple to offer concrete applications, constitutes instead a valuable (even if elementary) model useful for the simulation of various mechanical systems (seismographs, accelerometers, mechanical gyroscopic wheels, structures in general...): it is the basic representative scheme of any mechanical oscillating system.
The system under consideration consists of a mass M moving on a smooth horizontal plane (i.e. in the absence of friction) under the combined action of an external force F, of the elastic return force produced by an ideal spring of stiffness K (i.e. a spring whose stiffness, unlike the real case, is constant with respect to the elongation x) and of the damping force due to a viscous damper with a dimensional damping coefficient C.

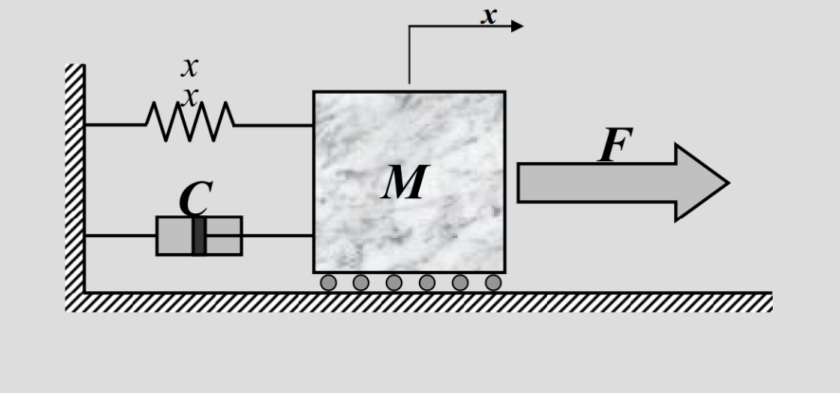


Figure 3.1: MCK System

Typically, in order to study these models, a system of equilibrium equations that governs the physical phenomenon is defined (in this case, this will consist of a single second-order differential equation in the x unknown that establishes the equilibrium of the forces acting on the mass), converted into the corresponding set of algebraic equations by means of Laplace transforms, and then translated into the corresponding block diagram.

At each instant, the forcing F is balanced by the sum of the inertia force, the spring's elastic return force and the damping force produced by the damper; this equilibrium condition can be formalised mathematically by obtaining:

$$M\frac{d^2x}{dt^2} + C\frac{dx}{dt} + Kx = F \tag{3.1}$$

Using the Laplace transform method, the above-mentioned second-order differential equation can be transformed into the simplest algebraic equation of the second degree in the variable s:

$$Ms^2\bar{x} + Cs\bar{x} + K\bar{x} = \bar{F} \tag{3.2}$$

From the latter, one can easily arrive at the corresponding Transfer Function:

$$G(s) = \frac{\bar{x}}{\bar{F}} = \frac{1}{Ms^2 + Cs + K} \tag{3.3}$$

To model the elementary block diagram, the differential equation must be rewritten as follows in order to obtain the expression for the mass linear acceleration:

$$\frac{d^2x}{dt^2} = \frac{F - Cdx/dt - Kx}{M} \tag{3.4}$$

Simulink model

The Simulink elementary block model is described as follows, together with the Transfer Function model (derived by directly relating the forcing term to the system response), in order to compare the results obtained from the two different solving approaches.

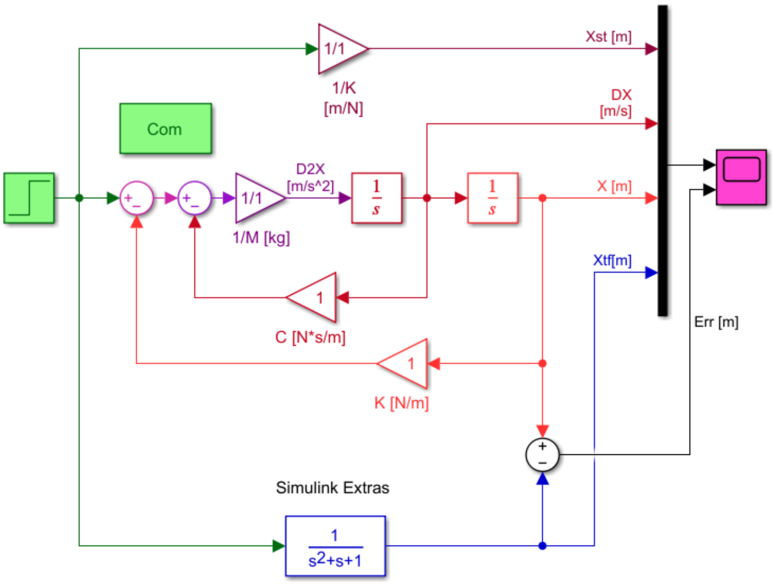


Figure 3.2: MCK Simulink model

In the following graph, it's possible to see the mass dynamics described by both the elementary block model and the Transfer Function, with the numerical error between the two approaches.

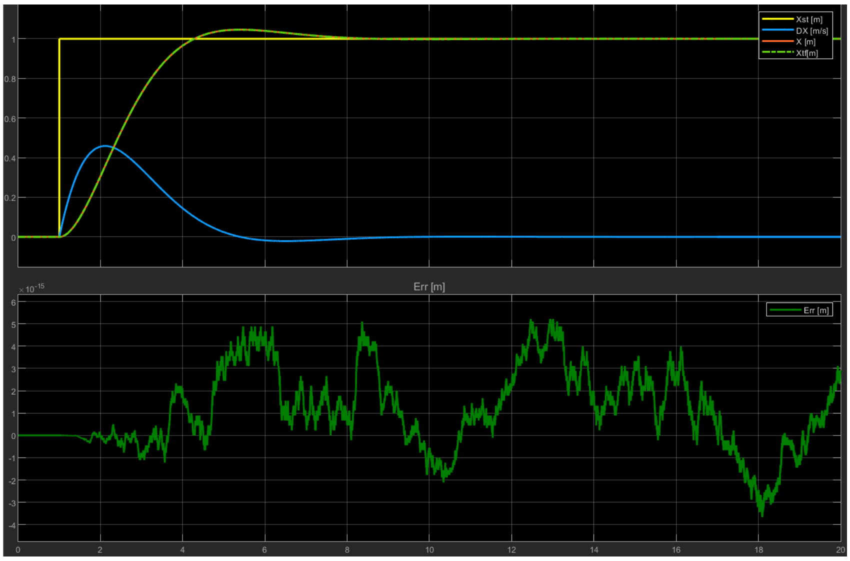
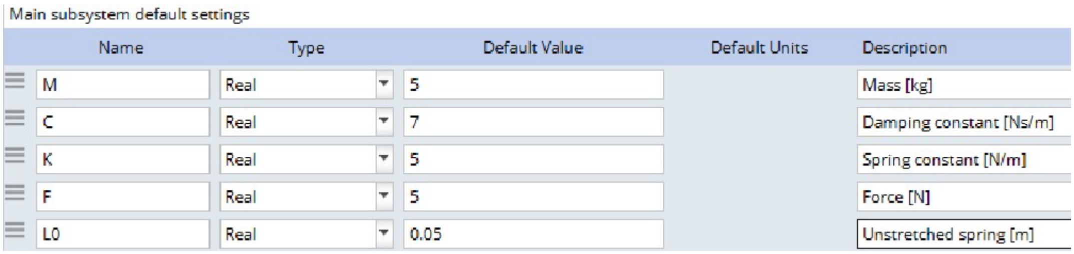


Figure 3.3: MCK Simulink results

The system is linear, and its reaction consists of an initial rise when force F is applied, followed by an overshoot when the system is at rest. The position subsequently declines (since the elastic return is greater than the force F), and finally the oscillatory dynamics is established, which attenuates and then settles on the position value commanded to infinity.
As a verification that the two dynamics are superimposed, it can possible see the position error calculated between x and x_{tf} positions. This error is of the order of 10^{-15}.

Elements models

For the Elements model, the same parameter set (shown in the table) was applied as in the Simulink simulation, with the aim of achieving the best possible match between the two models. Also the same solver has been set, using Euler first-order fixed-step resolution method (ode1), which is the simplest and most controllable possible. The time step is set at $1e^{-4}s$ in both simulation environments.



Main subsystem default settings

Name	Type	Default Value	Default Units	Description
M	Real	5		Mass [kg]
C	Real	7		Damping constant [Ns/m]
K	Real	5		Spring constant [N/m]
F	Real	5		Force [N]
L0	Real	0.05		Unstretched spring [m]

Table 3.1: MCK parameter set

Now, the different Elements models, built to replicate the Simulink one, are presented.
The first models are an exact reproductions of the Simulink elementary block diagram with the Transfer Function comparison.
Then, a physical model, built with 1D traslational components, is shown.

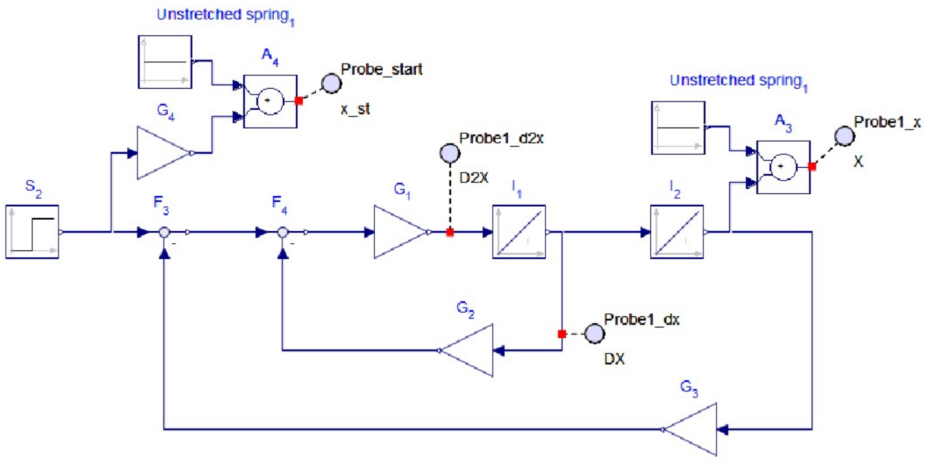


Figure 3.4: MCK Elements elementary block model

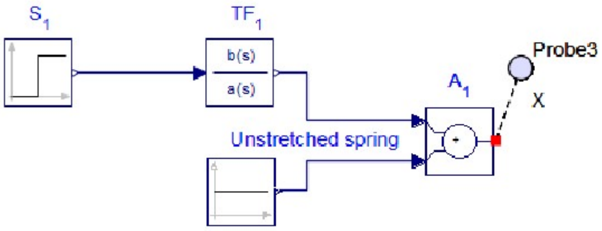


Figure 3.5: MCK Elements T.F. model

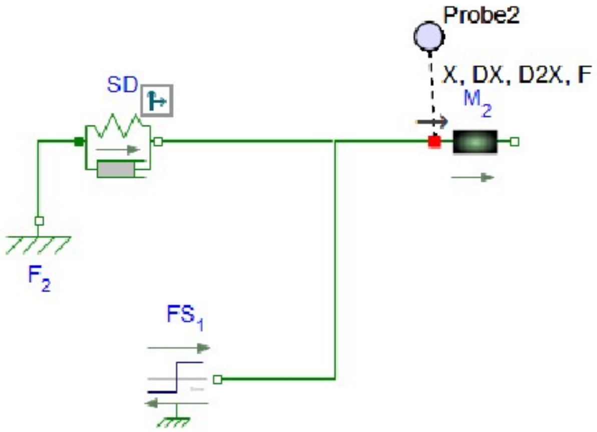


Figure 3.6: MCK Elements physical model

It can be seen that the models, in this case, include a constant value, defined as ’Unstretched spring’, which is added to the simulation results (Probe). This is an expedient carried out in order to have a better correspondence between these models and the one built using physical components, since the Unstretched spring value cannot be neglected when defining the equations that define the system.

It can also be seen how much simpler it is to develop a physical model on Elements. Only four blocks from the 1-D Translation library were used to represent the system:

- Fixed frame F_2
- Spring-Damper SD
- Step force source FS_1
- translational mass M_2

The results for the simulations of the three different models are presented below. It is evident that the plots follow the same path as the graphs produced by Simulink, a sign that the models are well developed.

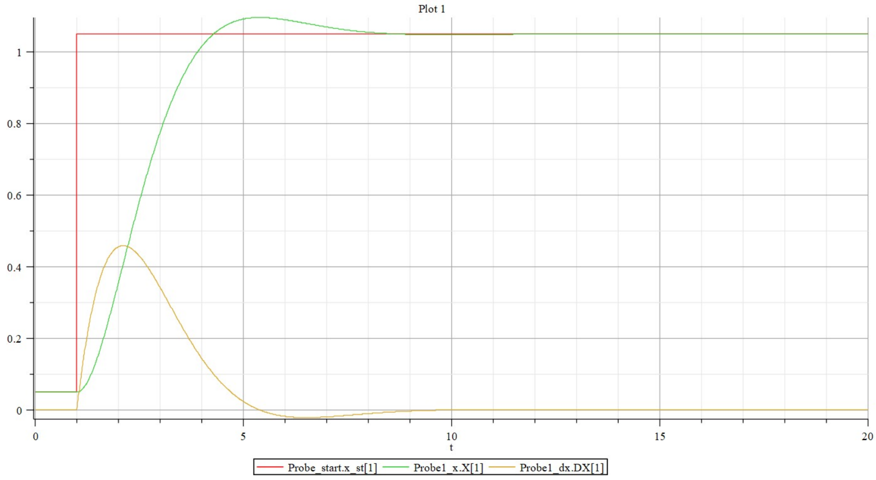


Figure 3.7: MCK, elementary block model results

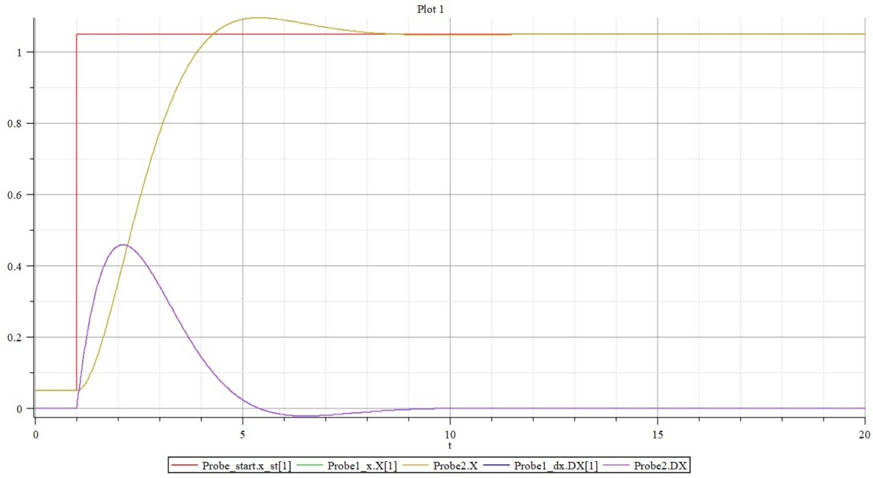


Figure 3.8: MCK, elementary block and physical models results comparison

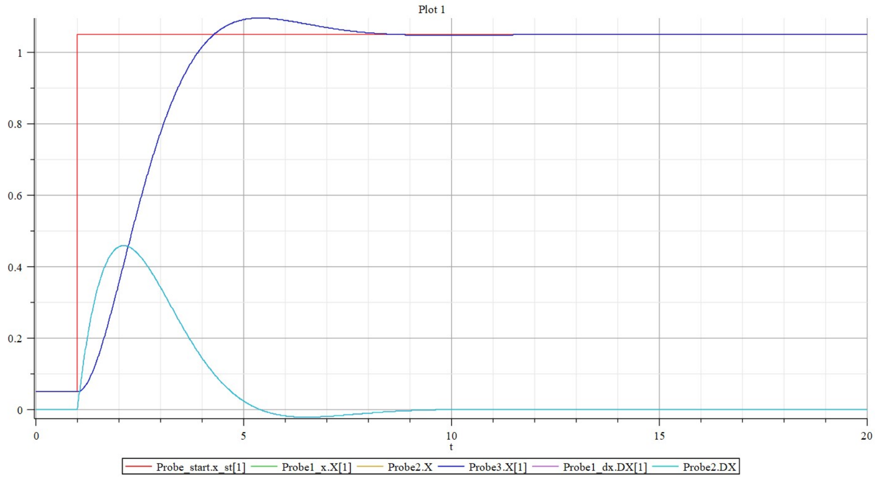


Figure 3.9: MCK, elementary block, T.F. and physical models results comparison

As a verification that the three dynamics are superimposed, it can possible see the position error calculated between the elementary block model with the physical model (1) and the elementary block model with the Transfer Function model (2). These errors are of the order of 10^{-15} and 10^{-16}.

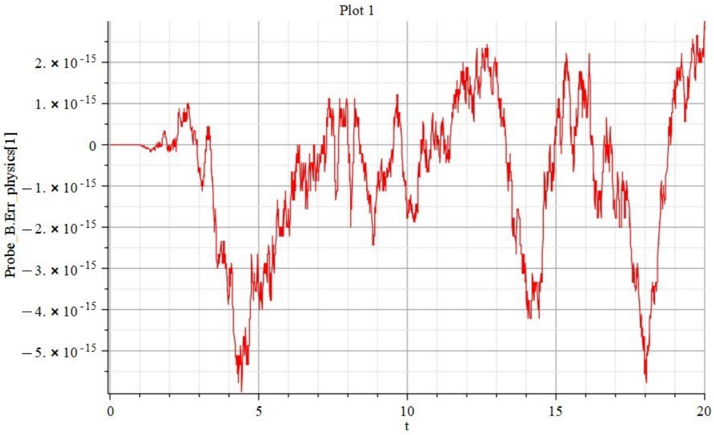


Figure 3.10: MCK Elements position error - 1

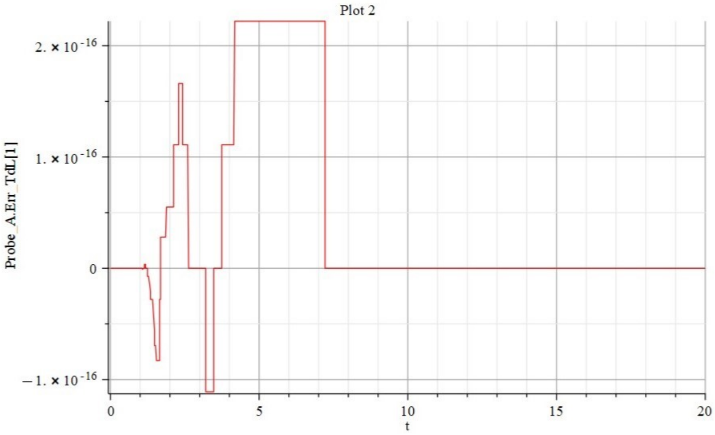


Figure 3.11: MCK Elements position error - 2

Another interesting feature is the Elements 3-D visualisation framework.
It enables the creation and animation of three-dimensional graphical representations of multibody mechanical system models. This environment is useful for examine the model's 3-D configuration and visually evaluate the system's performance under different scenarios and at different simulation start times.

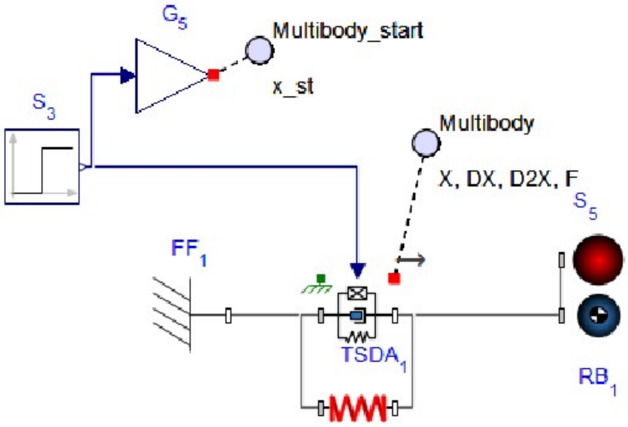


Figure 3.12: MCK Elements multibody mechanical model

As we can see, also this mass dynamic is superimposed to the other models' ones.

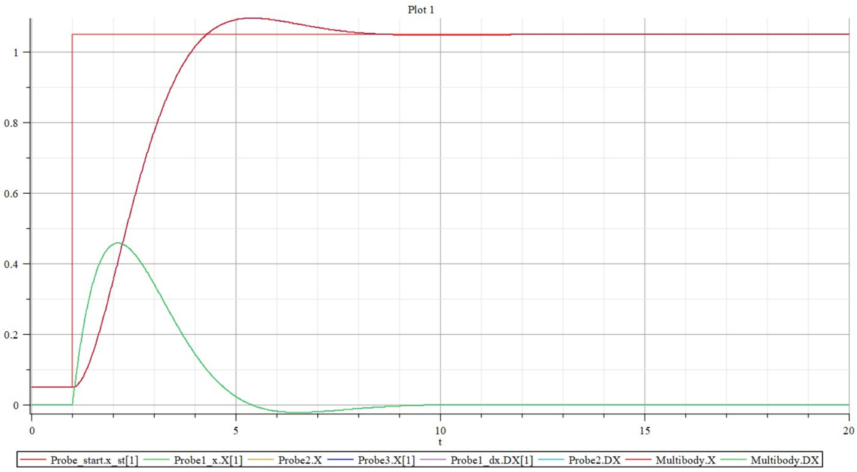


Figure 3.13: MCK Elements models results comparison

Two fragments of the 3-D graphical animation (starting time and elapsing time) of the multibody mechanical system model, produced in conjunction with the Elements simulation, are shown below.

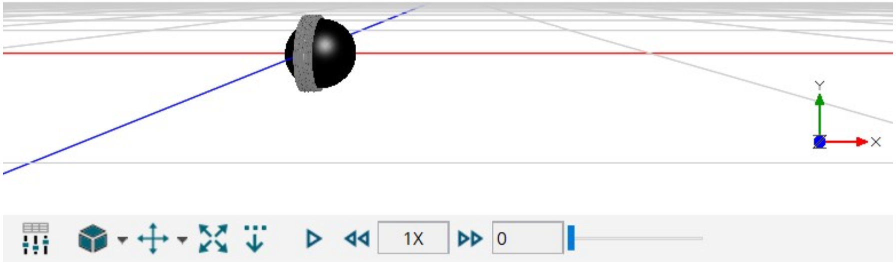


Figure 3.14: 3-D graphical animation $T_s = 0s$

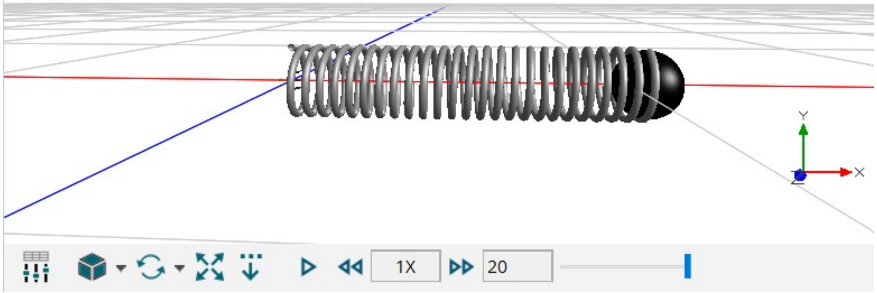


Figure 3.15: 3-D graphical animation $T_f = 20s$

3.2.2 Harmonic Oscillator System

Now, we will analyse a harmonic oscillator, i.e. a system consisting of two masses connected to each other by means of an ideal spring (with constant stiffness K_R and independent of its relative deformation, equal to the difference between the instantaneous position of the two masses $y - x$) and an ideal damper (with constant dimensional viscous damping coefficient C_R and independent of the corresponding relative velocity): the first mass will be subjected to the 'commanded' driving force F_M while the second mass will be subjected to the external load F_R. The entire system is subjected to absolute damping by means of a viscous damper connecting the first mass to a fixed reference. The mechanical configuration chosen, the type of loads applied and also the name of the different coefficients are consistent with the servomechanical applications that will be analysed later in the chapter.

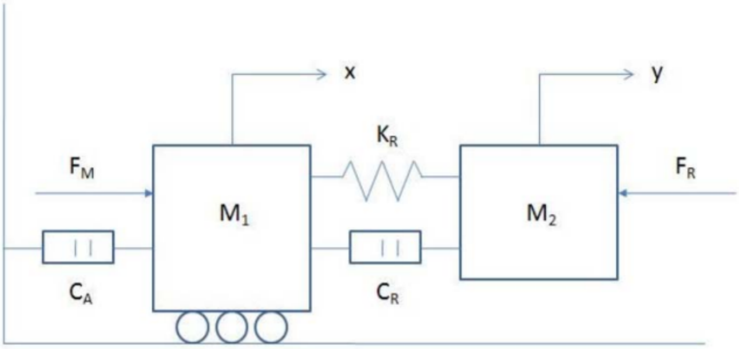


Figure 3.16: Harmonic Oscillator system

The dynamics of the system, clearly characterised by two translational degrees of freedom (relating, for example, to the instantaneous position of the centre of equilibrium of the two masses), is described by the following differential equations:

$$\begin{cases} M_1\ddot{x} + C_A\dot{x} = F_M K_R(x-y) - C_R(\dot{x}-\dot{y}) \\ M_2\ddot{y} = K_R(x-y) + C_R(\dot{x}-\dot{y}) - F_R \end{cases} \tag{3.5}$$

In order to obtain the system's elementary block model it's necessary to introduce a third equation:

$$F_M = G_M(Com - y) \tag{3.6}$$

Inspired by a servomechanical configuration, the F_M force applied to the mass M_1 is proportional, via a G_M gain, to the position error Err, i.e. the difference between a hypothetical commanded position Com (command input of the dynamic system) and the corresponding instantaneous position y of the mass M_2.

Simulink model

The three equations above represent the mathematical model of the problem under investigation, and allow the Simulink algorithm to be constructed, as with the previous MCK system.

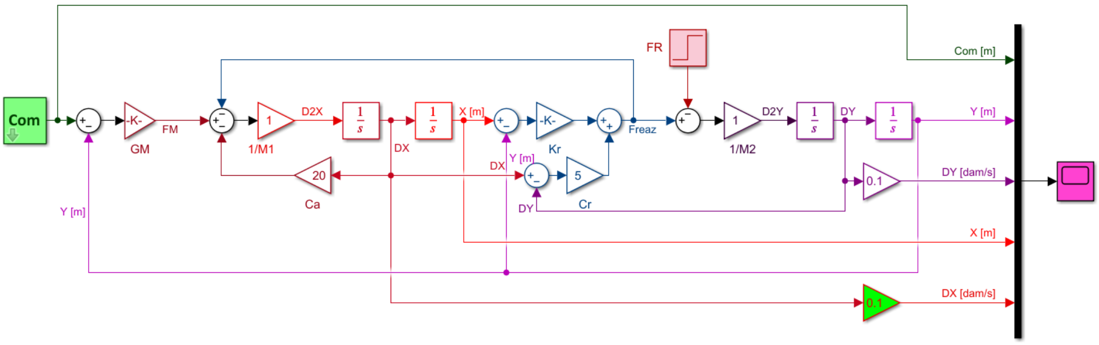


Figure 3.17: Harmonic Oscillator Simulink model

Thanks to the step command, mass M_1 begins to move, and the mass M_2 follows suit due to an elastic recall effect (at the beginning it starts quite slowly). The two masses establish an oscillatory dynamics that tends to decrease: in about 2 seconds, the oscillations of x and y are nearly the same.
Then, the system is at rest and reaches the commanded position.
When the load F_R is applied, the previously resting system restarts, and both the masses M_1 and M_2 move away from the equilibrium point reached. The actual equilibrium locations at which both masses are held will be different.

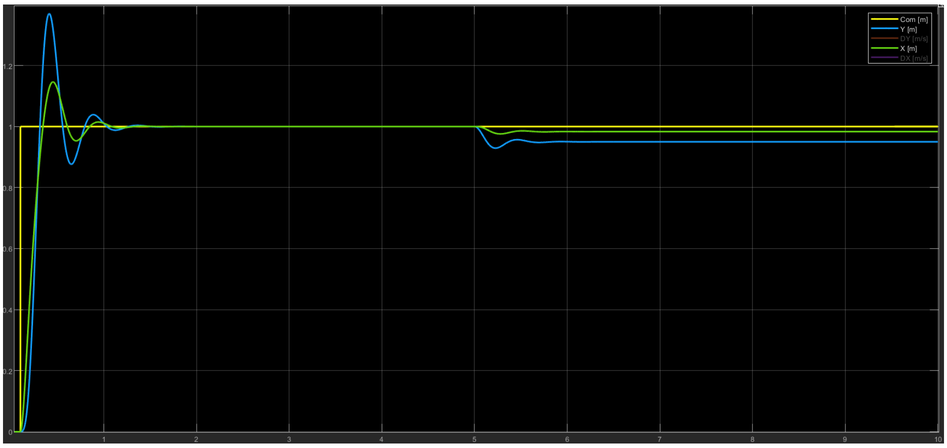


Figure 3.18: Harmonic Oscillator, Simulink position results

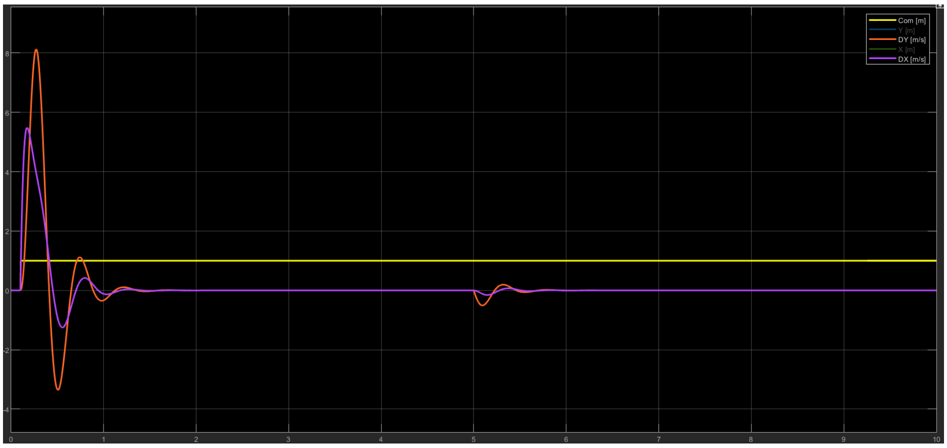
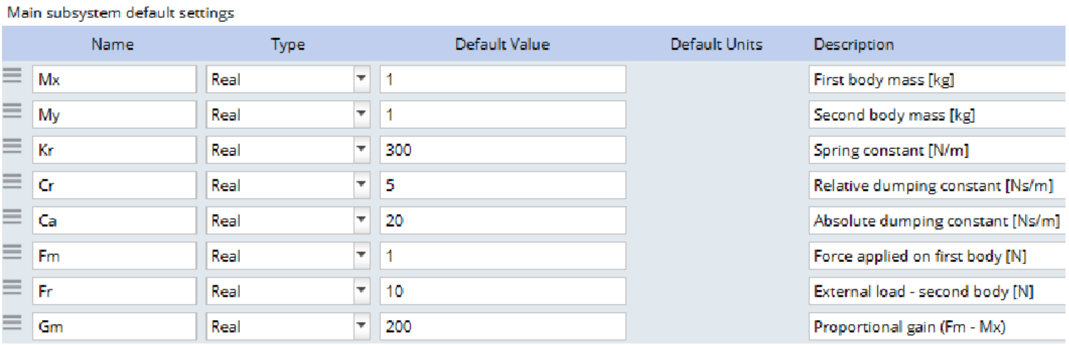


Figure 3.19: Harmonic Oscillator, Simulink velocity results

The external force F_R and the relative stiffness between the masses M_1 and M_2 define how far apart the final equilibrium positions are from the required position. In fact, decreasing the K_R of the spring reduces the divergence.

Elements models

For the Elements model, the same parameter set (shown in the table) was applied as in the Simulink simulation, with the aim of achieving the best possible match between the two models. Also the same solver has been set, using Euler first-order fixed-step resolution method (ode1), which is the simplest and most controllable possible. The time step is set at $1e^{-4}s$ in both simulation environments.



Main subsystem default settings

Name	Type	Default Value	Default Units	Description
Mx	Real	1		First body mass [kg]
My	Real	1		Second body mass [kg]
Kr	Real	300		Spring constant [N/m]
Cr	Real	5		Relative dumping constant [Ns/m]
Ca	Real	20		Absolute dumping constant [Ns/m]
Fm	Real	1		Force applied on first body [N]
Fr	Real	10		External load - second body [N]
Gm	Real	200		Proportional gain (Fm - Mx)

Table 3.2: Harmonic Oscillator parameter set

Now, the different Elements models, built to replicate the Simulink one, are presented. The first model are an exact reproduction of the Simulink elementary block diagram. Then, the physical model, built with 1-D traslational components, is shown.

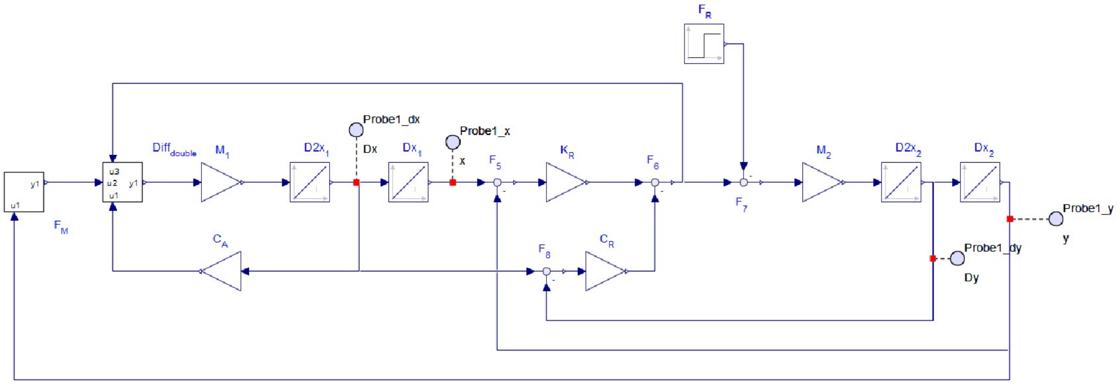


Figure 3.20: Harmonic Oscillator elementary block model

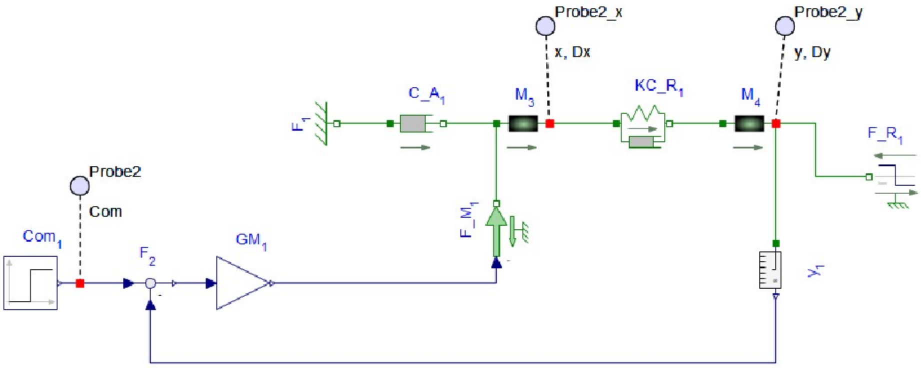


Figure 3.21: Harmonic Oscillator physical model

In this case, the Elements physical model representing the Harmonic Oscillator system consist of:

- Step source Com_1
- Feedback F_2 (position error between command input and y)
- Gain block GM_1
- Commanded force source FM_1, which converts the adimensional input signal in a force output signal $[N]$ that can interact with the physical model of the system
- Fixed frame F_1
- Translational damper CA_1
- Translational mass (1) M_3
- Spring-Damper system KCR_1
- Translational mass (2) M_4
- Position sensor y_1
- Step force source FR_1

The graphs presented below show the system dynamic simulated by the Elements elementary block model, which is exactly the same analysed on Simulink.

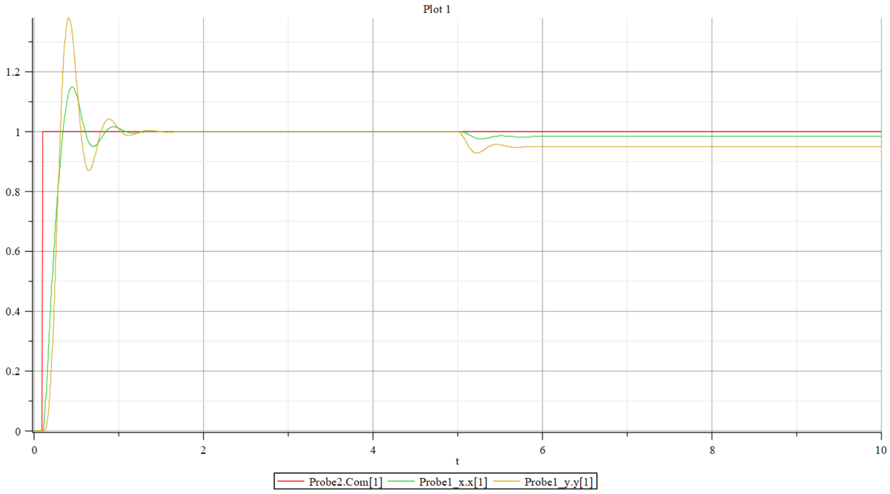


Figure 3.22: Harmonic Oscillator, Elements elementary block position results

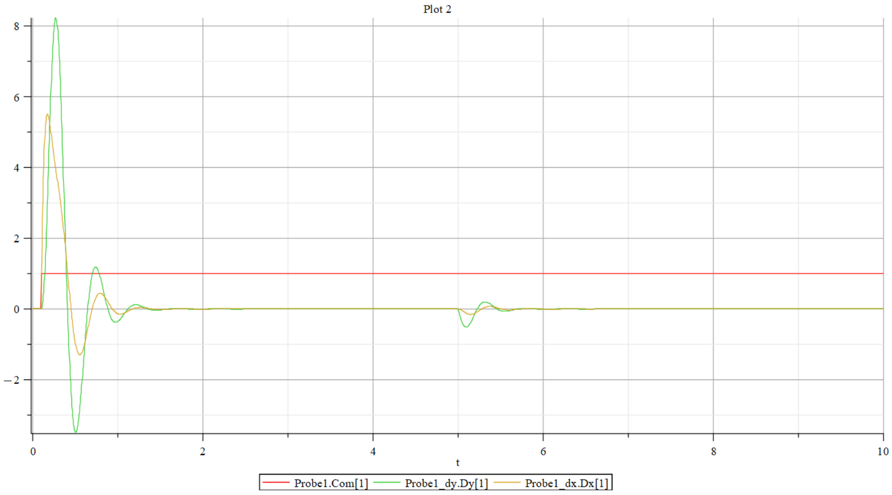


Figure 3.23: Harmonic Oscillator, Elements elementary block velocity results

It’s clear that the dynamics of the two masses described through Simulink and Elements are identically. It can possible see the position error about x and y calculated between Simulink and Elements position respectively.
(The comparison was performed by importing the Elements results into Matlab).
For x this error is of the order of 10^{-13}.

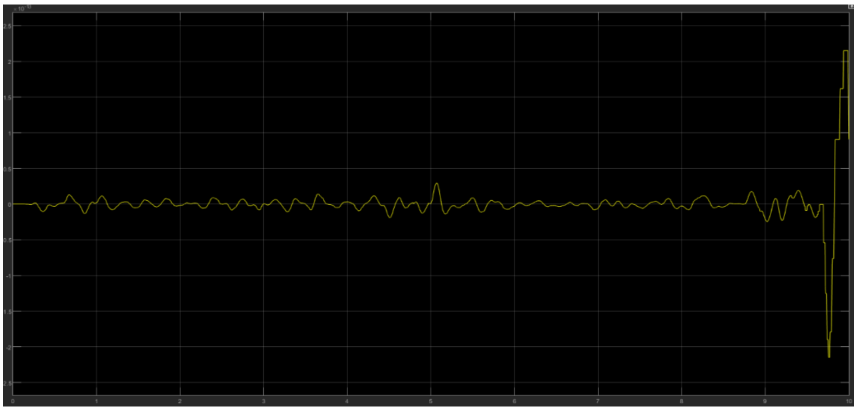


Figure 3.24: Position error (x) between Simulink and Elements

For y this error is of the order of 10^{-14}.

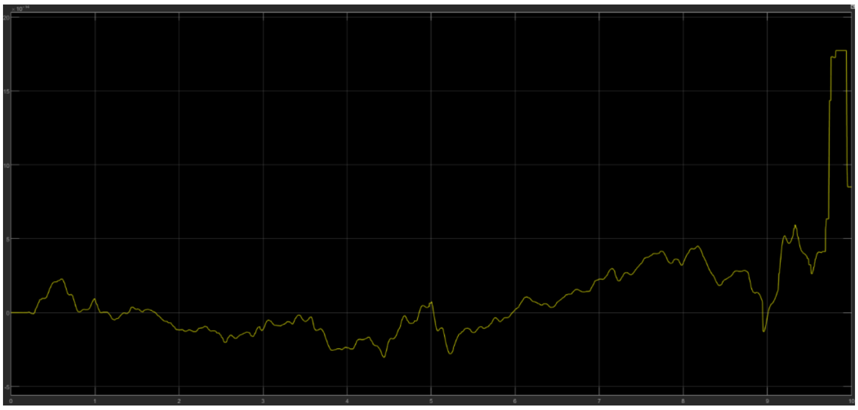


Figure 3.25: Position error (y) between Simulink and Elements

We can now focus on comparing the two different Elements models.
Graphs showing the results of the simulation conducted on both models (the one with elementary blocks and the one with physical blocks) are shown below.

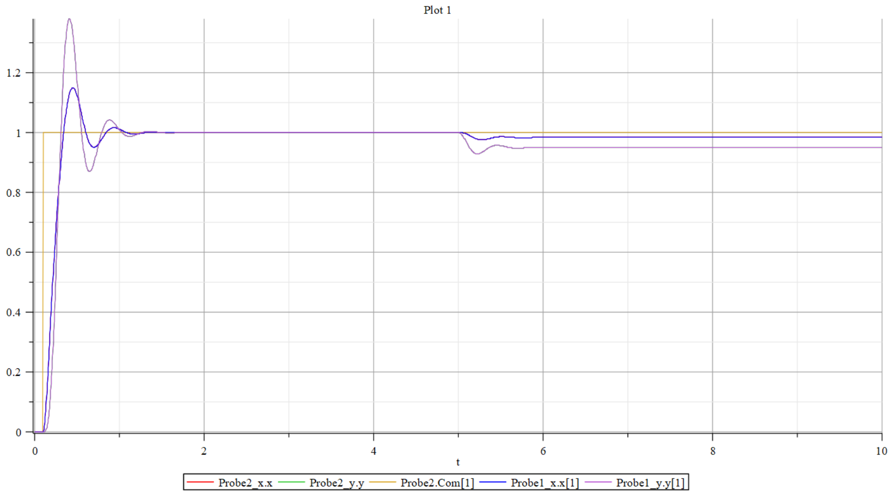


Figure 3.26: Harmonic Oscillator, Elements models position results

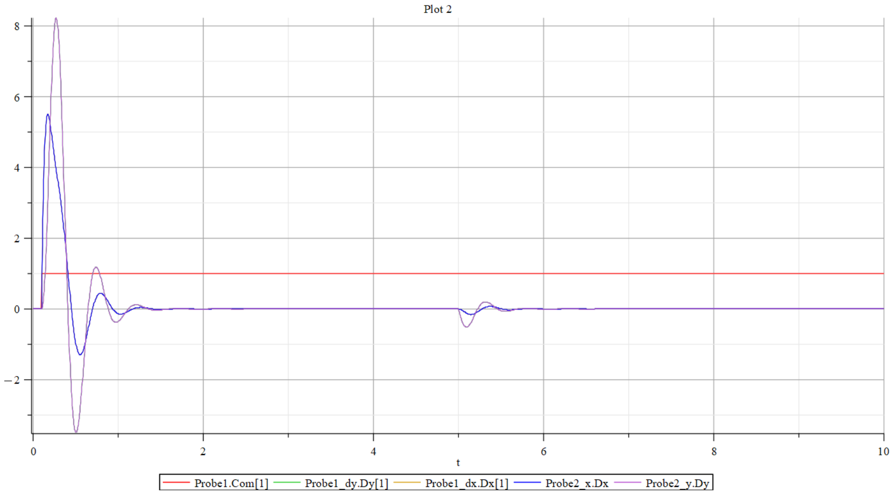


Figure 3.27: Harmonic Oscillator, Elements models velocity results

As a verification that the two dynamics are superimposed, it can possible see the position error about x and y calculated between the two models position respectively. For both x and y this error is of the order of 10^{-15}.

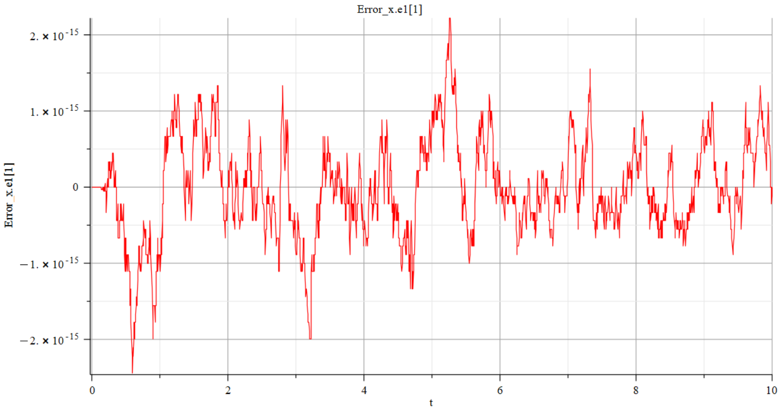


Figure 3.28: Position error (x) between Elements models

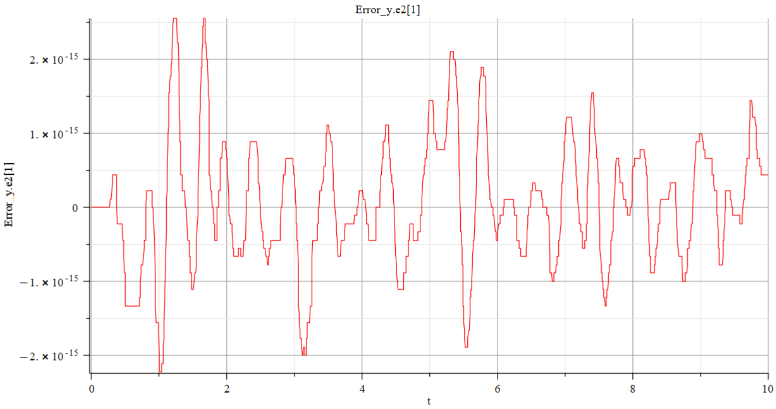


Figure 3.29: Position error (y) between Elements models

3.3 Electromechanical Actuator

On commercial or military aircraft currently in use, flight controls are usually hydraulically or electrically powered. For the primary controls, hydraulic boosting is generally used, with linear cylinder-piston motors or, more seldom, rotary motors. Secondary controls can be implemented using hydraulic or electric rotary motors. In recent decades, mainly as a result of the development of reliable electric motors with a high power-to-weight ratio and the consolidation of the necessary technologies, electromechanical secondary controls have established themselves in the field of small powers (typically <7kW) and, although with due caution, are also beginning to find practical uses for implementing primary controls (especially in the field of UAVs and small aircraft).
In this section, a typical primary flight control architecture realised with an electromechanical position servomechanism will be schematicallyexamined.

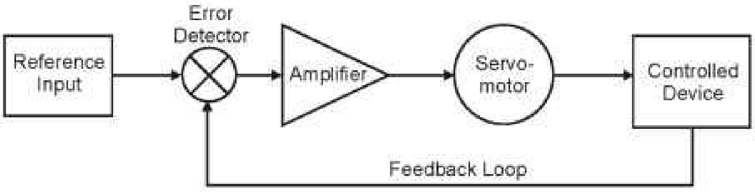


Figure 3.30: Schematic blocks representation of the examined EMA system

In order to develop the numerical model of the system, it will be assumed that the electric motor, gearbox, user and position transducer are connected to each other by means of an infinitely rigid transmission, thus neglecting line yielding, and free of mechanical backlash. In this way, it is possible to reduce all the inertias of the system to a single equivalent inertia J, and similarly, to consider a single viscous damping coefficient and equivalent friction c (operationally, all inertial and viscous terms relating to components of the system, as well as any external loads applied to the user itself, will be reduced to the motor shaft by means of the relative transmission ratios).

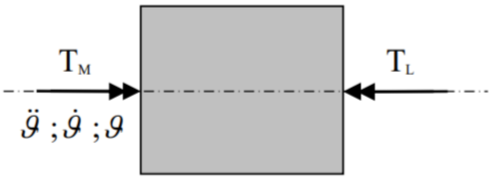


Figure 3.31: Schematic physical representation of the examined EMA system

The rotational dynamic equilibrium equation of the mechanical system just described will be:

$$J\ddot{\theta} + c\dot{\theta} = T_M - T_L \tag{3.7}$$

Where T_M is the driving torque produced by the rotor's magnetic induction field, T_L is the resistant torque produced by the external loads acting on the driven organ (suitably reduced to the rotor shaft) and θ the angular position of the mentioned rotor.
Bearing in mind that the first derivative of the angular position θ is equal to the rotational speed of the motor ω_M, the speed can be made explicit:

$$\dot{\omega} = \frac{T_M - T_L - c\omega_M}{J} \tag{3.8}$$

From which the baseline model for the system analysis can be derived in both Simulink and in Elements environment.

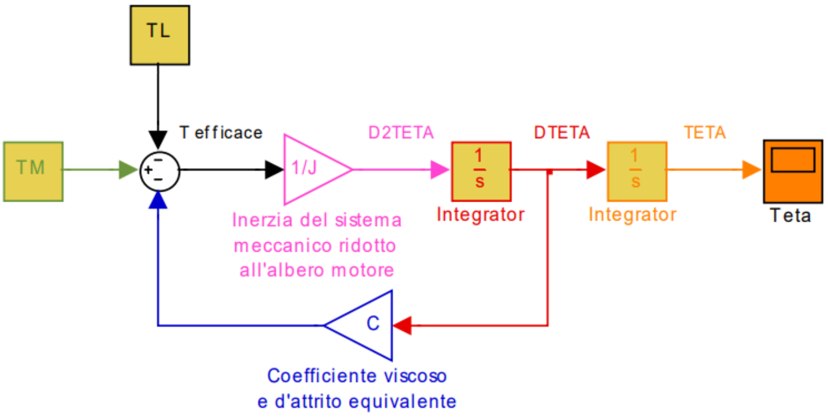


Figure 3.32: Simulink torque-driven model

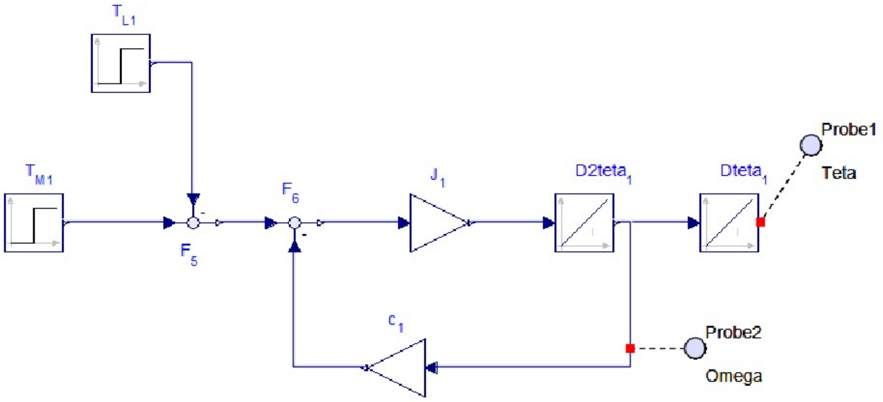


Figure 3.33: Elements torque-driven model

Assuming a Separately Excited DC motor, realised with a Permanent Magnet, the equation for the electromechanical characteristic of the electric motor can be deduced, i.e. that mathematical model which, from the supply voltage V_A, works out the corresponding value of the motor torque T_M as a function of the angular speed developed. The motor torque is proportional to the electric coil current I through a proportionality coefficient G_M (motor torque gain):

$$T_M = G_M I \tag{3.9}$$

Furthermore, the supply current is in turn proportional to the supply voltage V_A less the counter-electromotive force V_M according to a ratio $1/R$, where R is the ohmic resistance of the coils:

$$I = \frac{V_A - V_M}{R} \tag{3.10}$$

Since the electrical resistance of the coils is modelled as purely ohmic, the inductive effects are neglected, which, by delaying the dynamic response of the system, would slow down the variation of the current flowing in the motor.
In fact, according to the purely ohmic model, the current should evolve instantaneously in a fashion proportional to the supply voltage. Whereas, according to more advanced and realistic modelling, transients should develop, which can be represented, to a first approximation, by first-order dynamics (now neglected, due to their very rapid evolution, but taken up further on in the study):

$$V_A - V_M = RI + L\frac{dI}{dt} \tag{3.11}$$

Analysing the mechanical characteristic of the electric motor, it can be observed that the maximum torque is developed in stall condition ($\omega = 0$) and decreases until it is null at a maximum speed, known as the no-load speed. In fact, in proportion to the angular speed of the rotor ω, the counter-voltage (known as counter-electromotive force) increases, which tends to reduce the torque delivered by the motor.

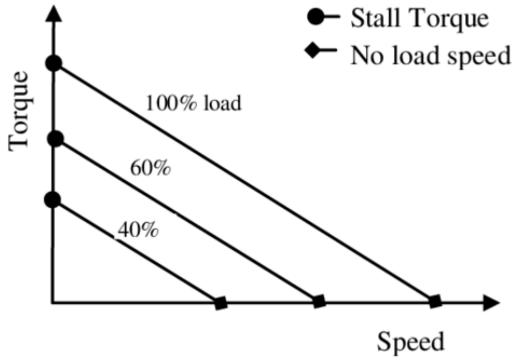


Figure 3.34: Mechanical characteristic of the Separately Excited DC motor

It is possible to integrate the above-mentioned electromechanical model as follows:

$$T_M = G_M\frac{V_A - k\omega_M}{R} \tag{3.12}$$

It is possible to think of F.C.E.M. as proportional to the angular speed of the electric motor by means of a constant k characteristic of the given motor type.

$$V_M = F.C.E.M. = k\omega_M \tag{3.13}$$

The model derived from the above considerations are shown below, in the block diagrams developed in both Simulink and Elements environment.

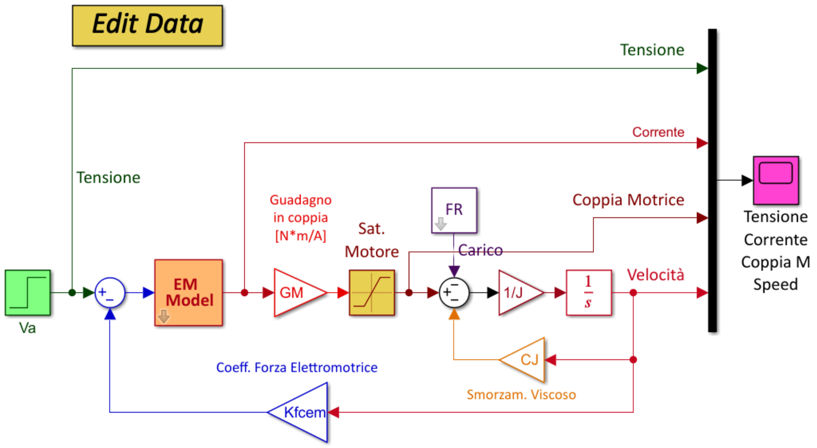


Figure 3.35: Simulink voltage-driven model

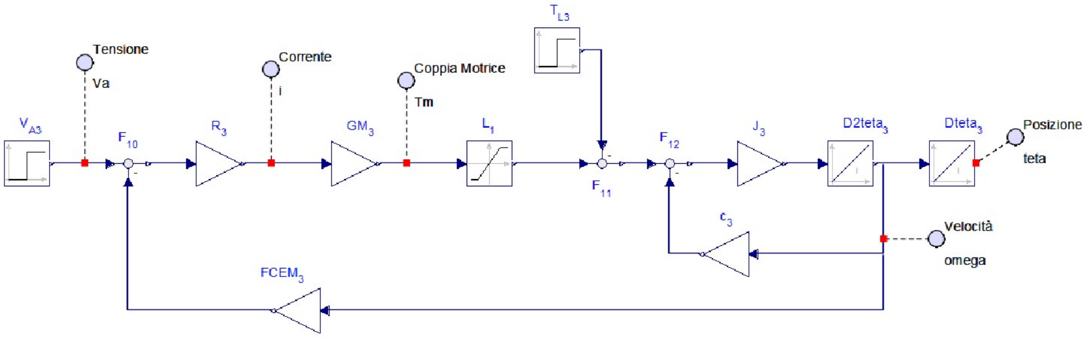


Figure 3.36: Elements voltage-driven model

To conclude the discussion of the servomechanism, it is necessary to close the position feedback by expressing the relationship that links the supply voltage of the rotor terminals with the corresponding position error, in other words, with the input of the driver, by defining the characteristic equation of the motor controller (power supply/amplifier unit that drives the electric motor).

The modelling of the aforementioned comparator/amplifier is usually carried out by neglecting the delays due to the dynamics of the various electronic components and adopting a dynamic model with instantaneous response, which directly correlates the position error Err to the V_A voltage by means of an appropriate amplification gain G_A:

$$V_A = ErrG_A \tag{3.14}$$

Substituting all the relations obtained, it results in:

$$\dot{\omega} = \frac{G_M \frac{ErrG_A - k\omega_M}{R} - T_L - c\omega_M}{J} \tag{3.15}$$

The corresponding electromechanical servomechanism model is derived from this equation, in both Simulink and Elements environment.

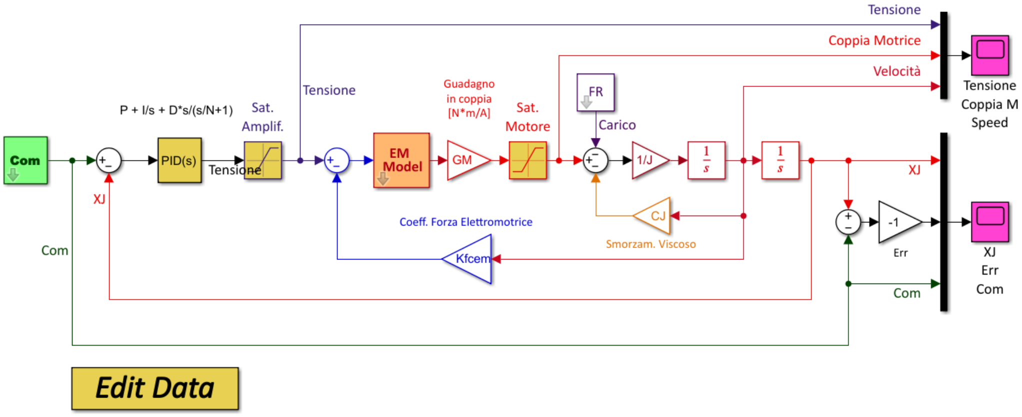


Figure 3.37: Simulink electromechanical servomechanism model

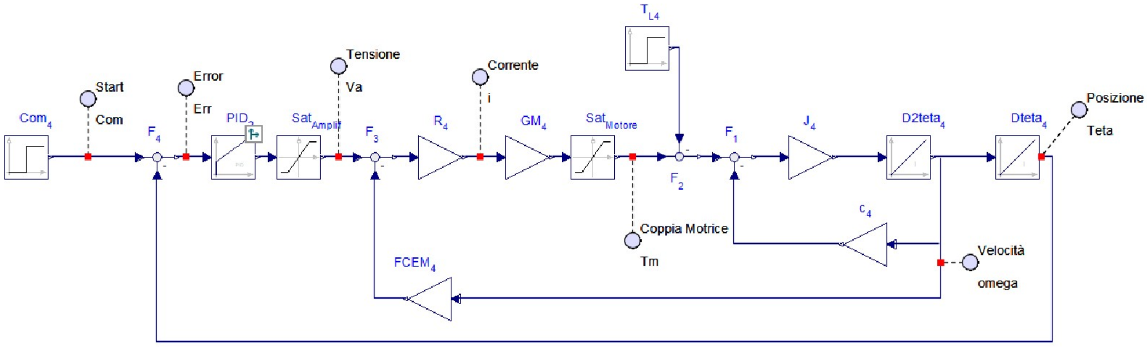


Figure 3.38: Elements electromechanical servomechanism model

Taking a close look at the block diagram, it is immediately clear that there are two 'Saturation' blocks and one 'PID' block (these blocks are included in both the Simulink and Elements libraries), which are not reflected in the mathematical model shown above.

The second block comes under the field of controlling and will be explored further with practical examples.

The former, on the other hand, make it possible to limit the value of the input signal within a given band, enabling simulations to be carried out to evaluate the dynamic response of the system subject to saturation phenomena.

Before going any further, a few remarks should be made about the non-linear nature of the physical system examined:

- the occurrence of saturation phenomena in the signal produced by the amplifier and in the flow of magnetic induction in the electric motor's irons, introduces effects into the system's dynamics that cannot be simulated in purely linear models;
- in order to ensure acceptable accuracy in simulations, it is often necessary (and appropriate) to take such non-linearities into account by introducing appropriate non-linear blocks;
- the adoption of a non-linear model does not necessarily entail an increase in model complexity; in fact, the introduction of appropriate non-linearities (representative of the model's physics) makes it possible to obtain, with relatively simple models, particularly accurate simulations.

Looking again at the mechanical characteristic of the electric motor, it can be seen that as the supply voltage increases, the curves shift towards higher torques:

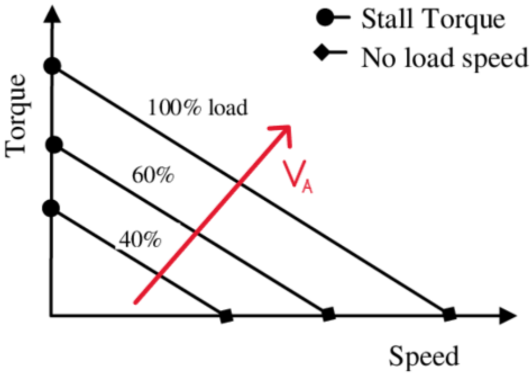


Figure 3.39: Ideal mechanical characteristic of the electric motor

Actually, due to the aforementioned phenomena of saturation of the magnetic induction flux of the rotor field, it can be seen that, at a certain point, while continuing to increase the coil current (i.e. the V_A supply voltage), there will no longer be a corresponding increase in the torque delivered by the motor.

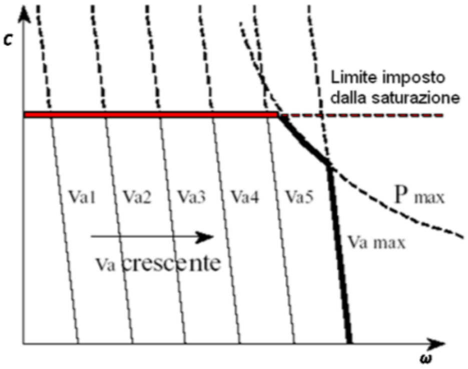
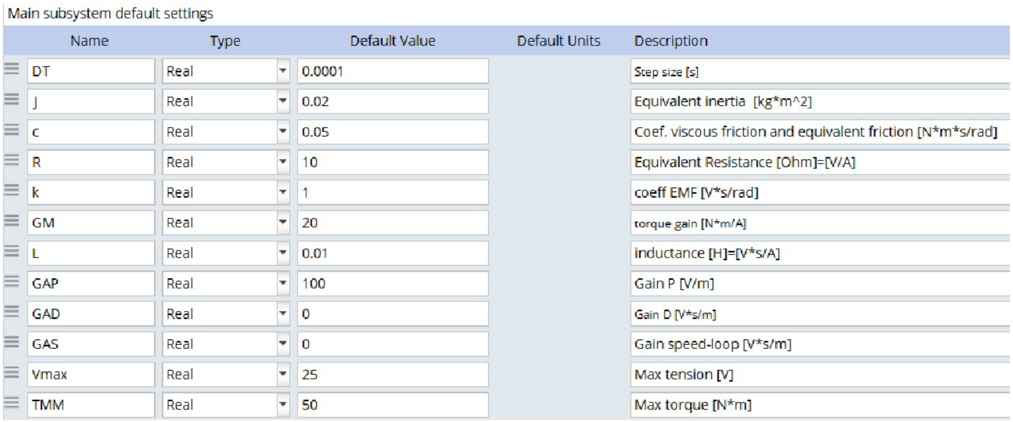


Figure 3.40: Real mechanical characteristic of the electric motor

In the remainder of the chapter, it will first compare the output values obtained from the simulations of the models previously illustrated (Figures 3.37 - 3.38) in the two environments.
Next, a model built with physical components in Elements will be described. After comparing the latter model with the elementary block ones, the peculiarities of the modelled electromechanical servomechanism will be examined by analysing its dynamic response under various command and load conditions.

The data used for the simulations are summarised in the table shown below.
The parameters that vary under different command and load conditions will be specified in each case.



Main subsystem default settings

Name	Type	Default Value	Default Units	Description
DT	Real	0.0001		Step size [s]
J	Real	0.02		Equivalent inertia [kg*m^2]
c	Real	0.05		Coef. viscous friction and equivalent friction [N*m*s/rad]
R	Real	10		Equivalent Resistance [Ohm]=[V/A]
k	Real	1		coeff EMF [V*s/rad]
GM	Real	20		torque gain [N*m/A]
L	Real	0.01		inductance [H]=[V*s/A]
GAP	Real	100		Gain P [V/m]
GAD	Real	0		Gain D [V*s/m]
GAS	Real	0		Gain speed-loop [V*s/m]
Vmax	Real	25		Max tension [V]
TMM	Real	50		Max torque [N*m]

Table 3.3: EMA parameter set

The dynamic response produced by the non-linear EMA system due to a step command of unit amplitude and an external load of 25% of the stall load is illustrated below.

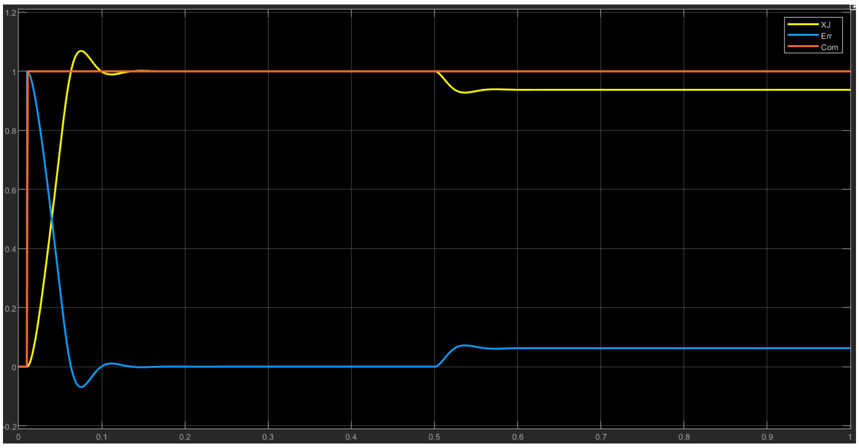


Figure 3.41: Simulink EMA response to unit step and 25% external load - 1

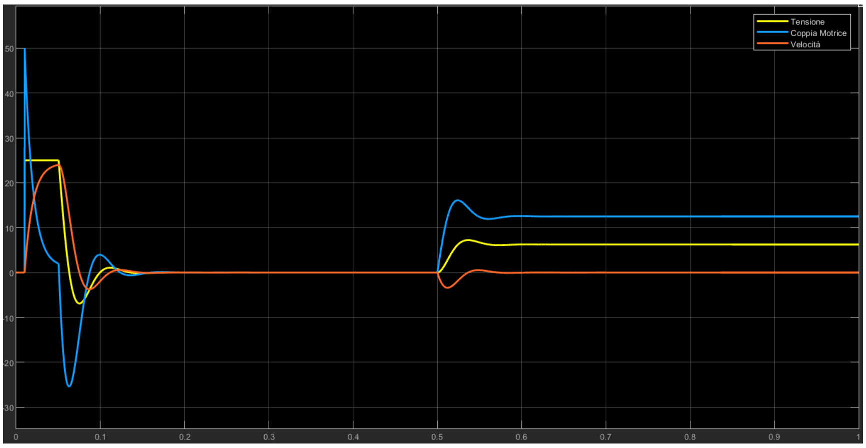


Figure 3.42: Simulink EMA response to unit step and 25% external load - 2

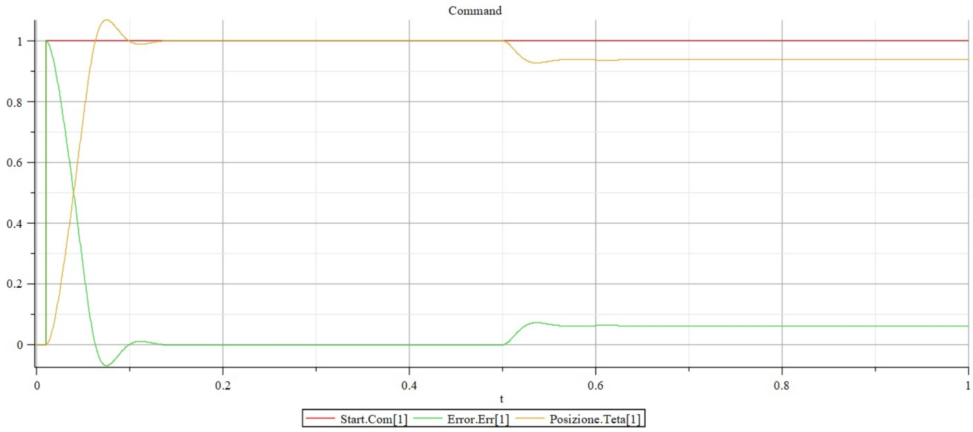


Figure 3.43: Elements EMA response to unit step and 25% external load - 1

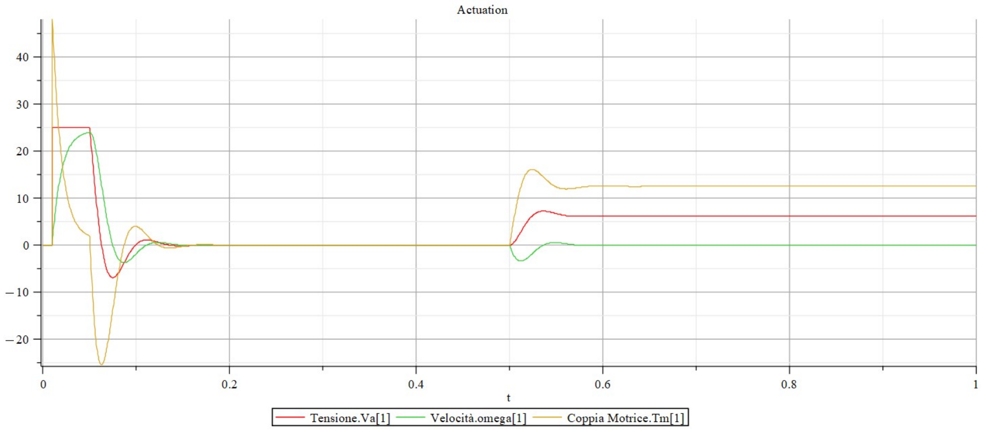


Figure 3.44: Elements EMA response to unit step and 25% external load - 2

Analysing the figures, it can be seen that at the instant the command step is applied, voltage and motor torque reach saturation value due to the high position error. As long as the torque is in saturation, the dynamic model of the system (second order) degrades to a simple first-order model since, in this condition, whatever is upstream of the saturation block is unable to influence the dynamics of the system (under such conditions, the saturation block is seen downstream as a simple constant input, while any feedback loops closing upstream of this block do not affect its response).

As the velocity increases, the FCEM increases and, consequently, the effective voltage and thus the drive torque decreases. When the error is reduced appropriately, the voltage also returns to the linear range, and thus the EMA response once again assumes the typical pattern of a subcritical second-order system converging asymptotically to the commanded position.

In fact, in the Figure 3.41, one can observe the course of the position (yellow curve) which, following the step command (red curve), emerges at a horizontal tangent and, after a stretch with a linear trend (i.e. an almost constant velocity, in which the voltage is in saturation and the system evolves as a first-order dynamic in velocity instead of second-order in position: it develops a velocity asymptotically tending to the steady state value for which the equilibrium between commanded voltage, FCEM and the corresponding viscous effects is realised), it converges back to the commanded position with a damped oscillatory dynamic (i.e. a linear second-order).

By applying an external load equal to 25% of the stall load for $t = 0.5[s]$, it can be seen how, at the instant the load is applied, the motor withdraws from the commanded position and, at the same time, the motor torque increases abruptly until it reaches a constant value that balances the said load. Obviously, if the external load is less than the maximum drive torque that can be developed by the motor, the system will reach an equilibrium configuration (at steady state, the error will be at a constant value).

After this brief analysis of the system dynamics represented by the graphs above, it can be clearly seen that the results generated by the two calculation environments correspond with a more than good approximation, as verified in the previous paragraphs.

We can now introduce the model that will be the object of study, built using the Elements libraries featuring physical components.

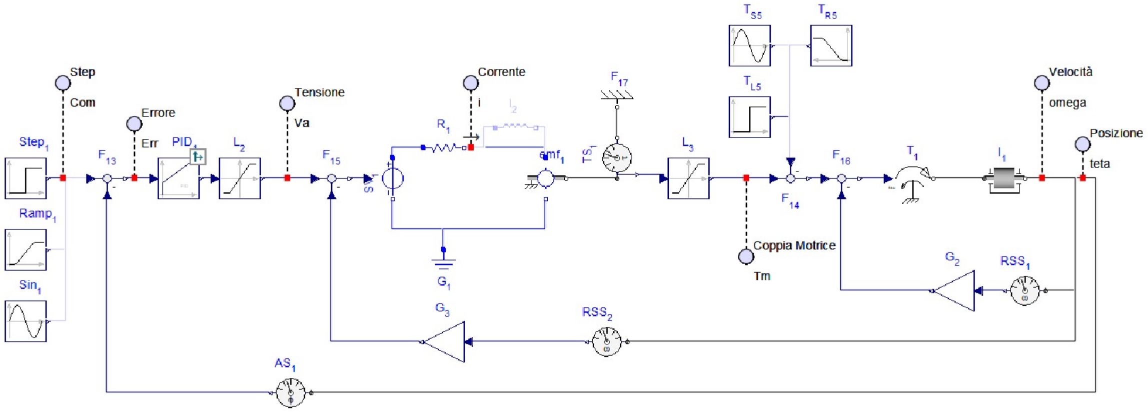


Figure 3.45: Elements EMA physical model

On the left side, there is a multiple choice signal source, with which different types of commands (step, ramp, sine wave...) can be simulated.
The PID and Saturation blocks remain identical to the previous model.
Blocks belonging to the Elements Electric library (Voltage Generator, Resistance, Inductance, Current/Torque Converter, Ground) were used to model the DC motor, characterised by a blue colour.
The output torque of the electric motor is measured by a torque sensor, so that the scaled value can go to the input of the Saturation block and be added to the external load.
At this point, the resulting dimensionless value is converted back into a torque, which acts on the rotating inertial mass component.
To close the loop, the angle θ and angular velocity ω are measured from the rotating mass (the latter will also be multiplied by the friction coefficients in a feedback loop).

The simulation results of the current model are shown below, together with those of the elementary block one. It can be seen that the graphs are exactly superimposed.

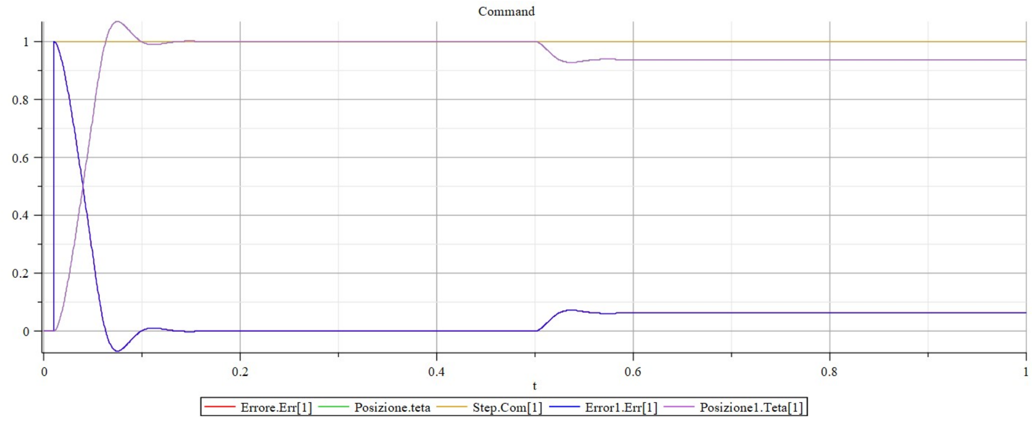


Figure 3.46: Elements EMA response comparison (unit step and 25% external load) - 1

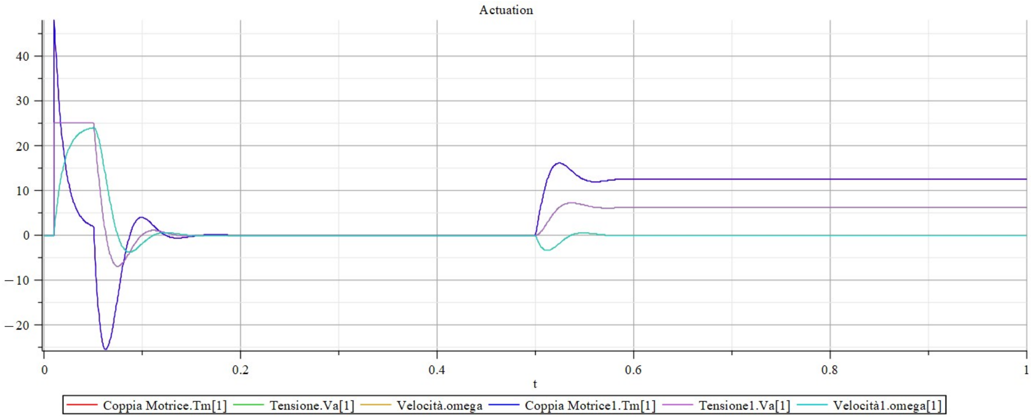


Figure 3.47: Elements EMA response comparison (unit step and 25% external load) - 2

As a verification that the two dynamics are superimposed, it can possible see the angular position error calculated between the two models θ respectively. The error is of the order of 10^{-15}.

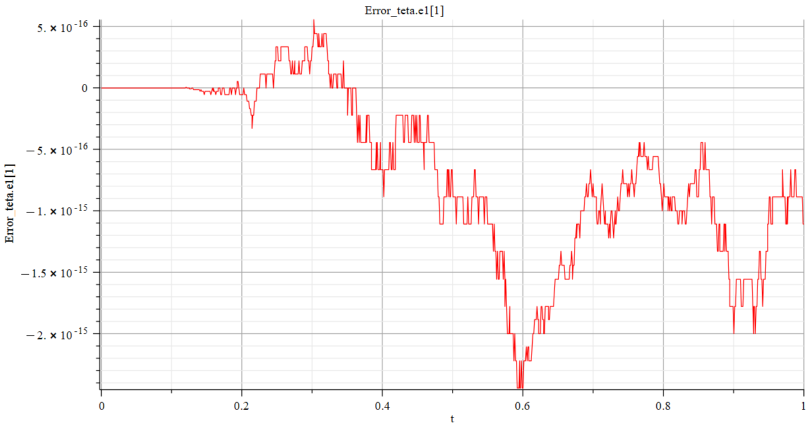


Figure 3.48: Theta error between EMA physical and elementary block models

After this comparison between different types of models representative of the system under study, other scenarios can be tested varying the simulation parameters.

DC Motor: Ohmic vs. Ohmic-Inductive

Within the model, the system's DC electric motor can be represented by the Resistance R (Ohmic), or by the sum of the Resistance R and the Inductance L (Ohmic-Inductive). For ease of description, the reference model is purely ohmic: this results in a faster response and overestimates the stability of the system, but can be accepted as a first approximation in certain cases. To establish when the inductance can be neglected, the coefficient $\tau = \frac{L}{R}$ is introduced: if this parameter is sufficiently small with respect to the characteristic time of the system, then the purely ohmic motor approximation can be accepted.

The results of three different scenarios with varying inductance are shown below:

- $L = 0.01H$

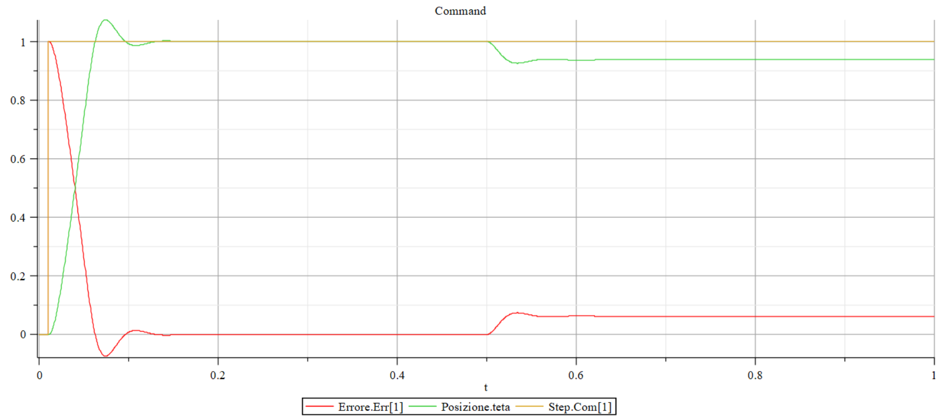


Figure 3.49: Ohmic-Inductive $(L = 0.01H)$ response - 1

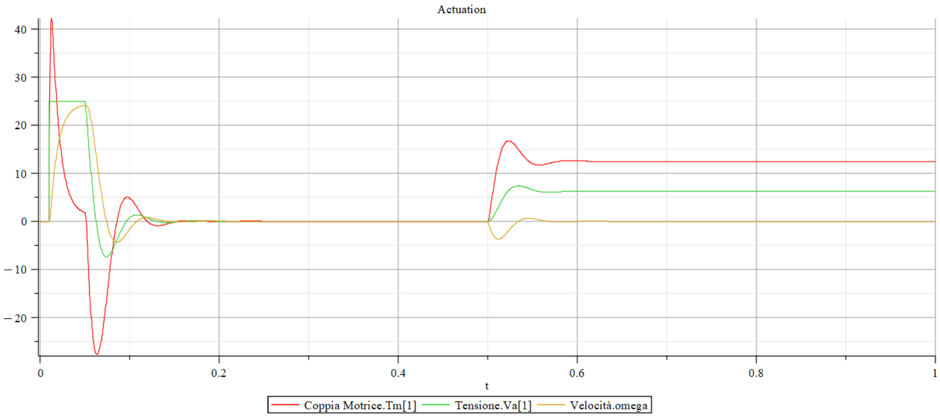


Figure 3.50: Ohmic-Inductive $(L = 0.01H)$ response - 2

With such a small value of inductance, the parameter τ is negligible with respect to the characteristic time of the system and the purely ohmic motor approximation is verified.

If we zoom in on the point at which the step command is actuated, however, for the angular velocity ω, we can see how the two dynamics differ. As mentioned earlier, the ohmic model has a faster response, while the ohmic-inductive model has a horizontal tangent caused by the inductance that retards the system's response.

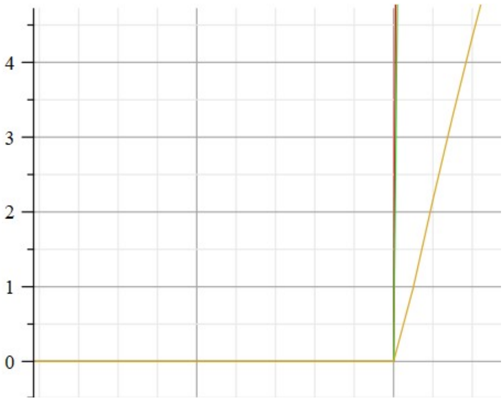


Figure 3.51: Angular velocity ω detail (Ohmic, $R = 10\Omega$)

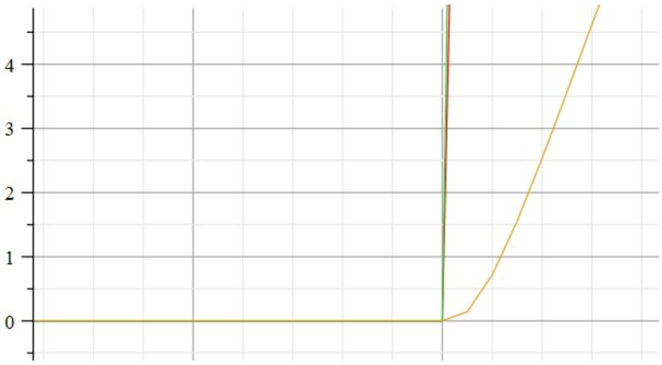


Figure 3.52: Angular velocity ω detail (Ohmic-Inductive, $R = 10\Omega$; $L = 0.01H$)

- $L = 0.03H$

  A detail of the system's response is shown below. It can be seen how, by increasing the value of the inductance (and thus of τ), the dynamics are altered compared to the purely ohmic model.

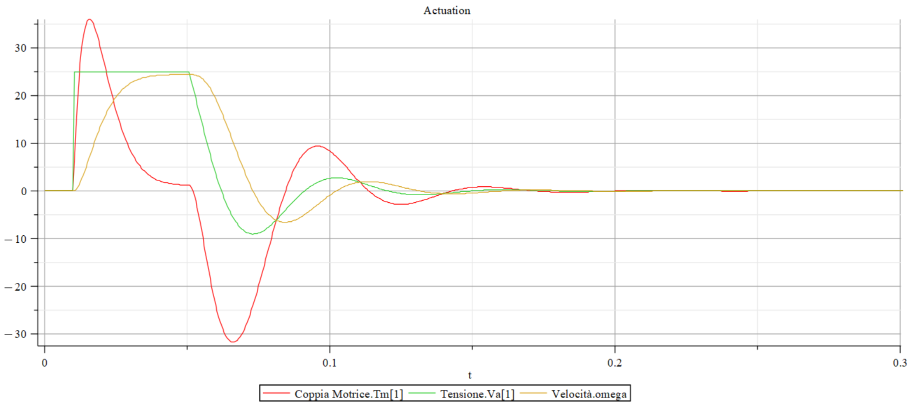


Figure 3.53: Ohmic-Inductive ($L = 0.03H$) response detail

- $L = 0.1H$
  Finally, we test the system with an inductance value one order of magnitude higher.

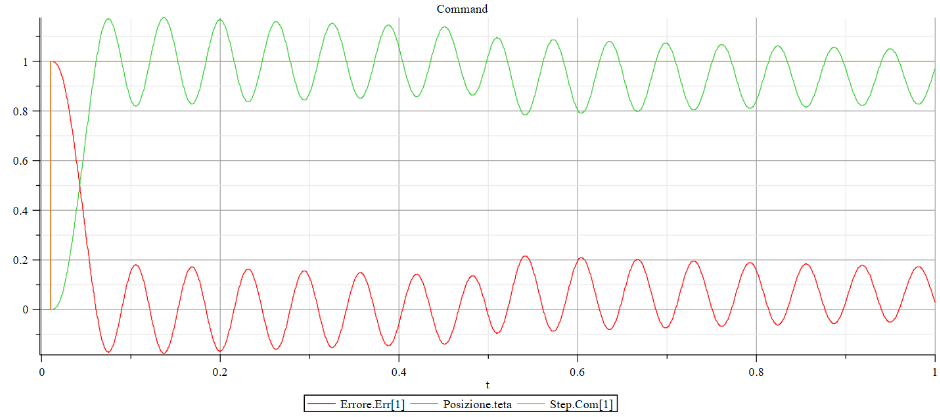


Figure 3.54: Ohmic-Inductive $(L = 0.1H)$ response - 1

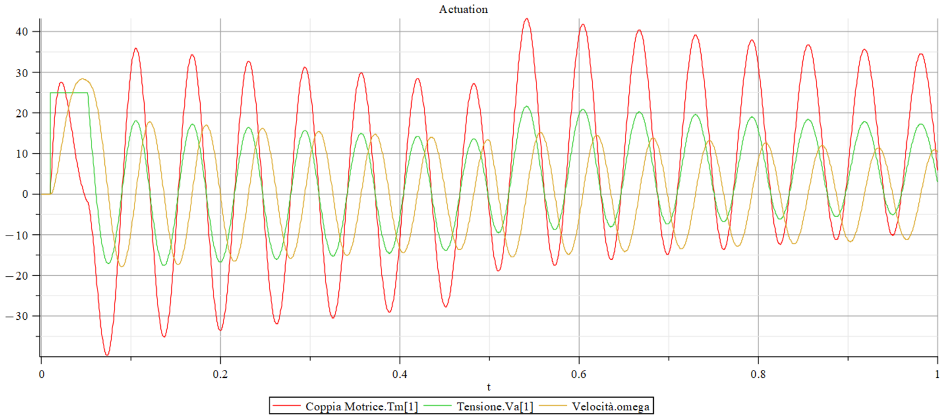


Figure 3.55: Ohmic-Inductive $(L = 0.1H)$ response - 2

With this inductance value, the system is no longer stable.

Input Command Signal

In the reference model, the position command for system actuation is defined by a step signal, which at a certain time t assumes a non-zero value towards which the system must converge. This type of model is ideal, since in reality the system's actuation velocity depends not only on the amplitude of the command, but also on external factors such as bearing wear, system friction, aerodynamic load acting on the driven surface, etc. Taking a trailing-edge high-lift device (flap) as an example, the pilot does not have a continuous selector to actuate the surfaces, but rather a discrete command. The control of the mechanism is therefore a step command, but the position controller translates that command by means of a ramp. This is done to prevent symmetrical control surfaces, in this case the flaps, subject to different boundary conditions, from giving an unexpected roll moment. The ramp command, with a suitably defined slope, allows two symmetrical surfaces to actuate without leading to an excessively uneven actuation, which can be compensated for by the autopilot.

We will now look in detail at an example of actuation with a ramp control, having a slope $r = 1 rad/s$.

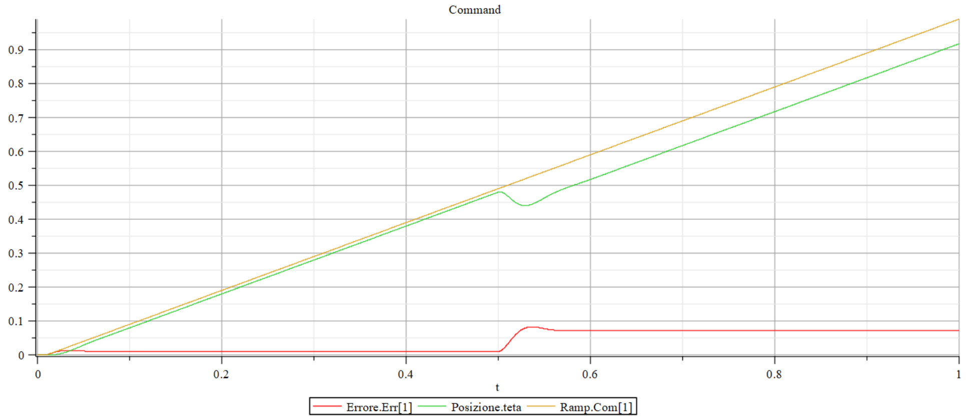


Figure 3.56: Ramp input command ($r = 1 rad/s$) response - 1

In this command, which is typical for flaps, the position is linearly varied over time, which is equivalent to imposing a certain extraction velocity (useful if desired to avoid asymmetrical surfaces actuation).

Differently to what happens with a step command, in this case the system does not start quickly to reset the error, but there is an initial transient in which the error grows to a value that causes the controlled surface to move.

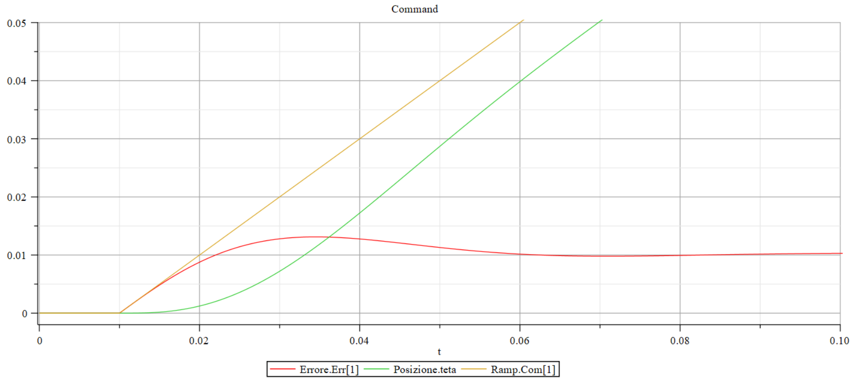


Figure 3.57: Ramp input command ($r = 1rad/s$) response detail - 1

The one shown in the figures is a typical ramp actuation: the error, after a short transient and an overshoot, goes to convergence and the system position settles to a value parallel to the assigned command. The difference between the command and the actual position remains constant because the controller is only proportional.

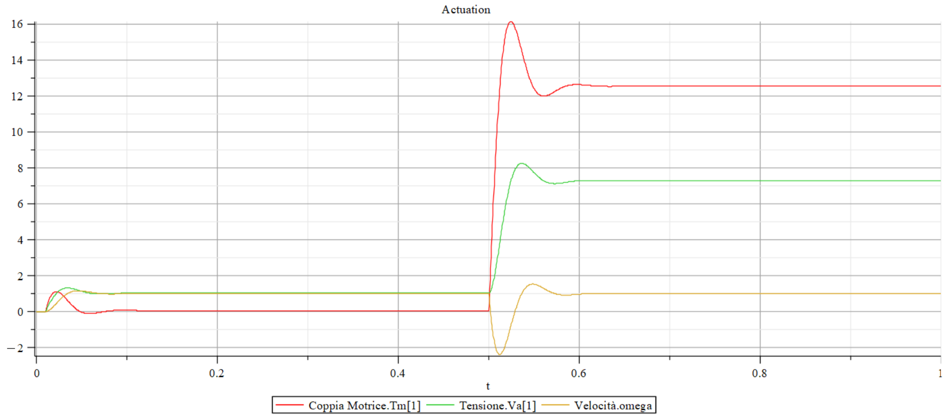


Figure 3.58: Ramp input command ($r = 1rad/s$) response - 2

As the error increases, the supply voltage and drive torque increase. The velocity evolves more slowly until a constant value is reached (the value obtained coincides with the slope of the ramp).

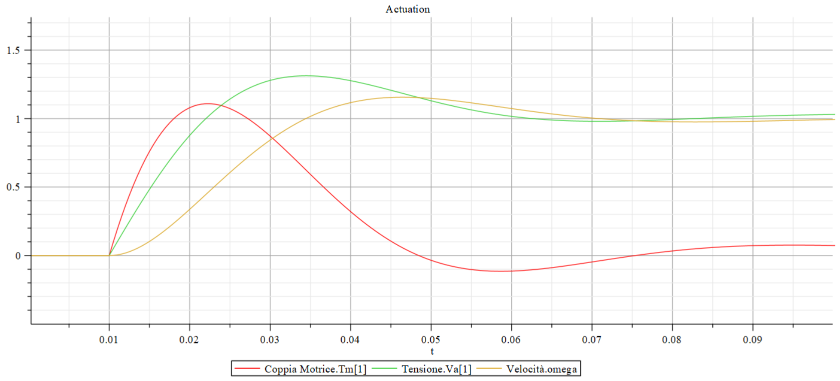


Figure 3.59: Ramp input command ($r = 1rad/s$) response detail - 2

Next, we examine periodic commands, in particular the sinusoidal command, which is typical of control surfaces that perform high-frequency compensation (e.g. the Eurofighter's canards).
For this type of analysis, we increase the simulation time to appreciate the results and, for simplicity, we impose no external load.

- Semiamplitude $A = 0.5rad$ and frequency $f = 2\pi\, rad/s = 1Hz$

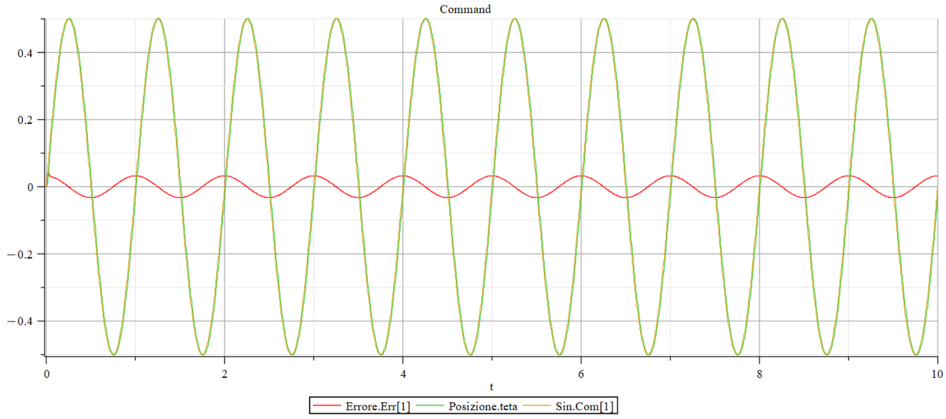


Figure 3.60: Sin input command ($A = 0.5rad/s$; $f = 1Hz$) response

The sinusoid can be thought of as a ramp that varies over time, so, as can be easily guessed, the actual position is in pursuit of the commanded one, always with a slight delay (as can be seen in Figure 3.61).

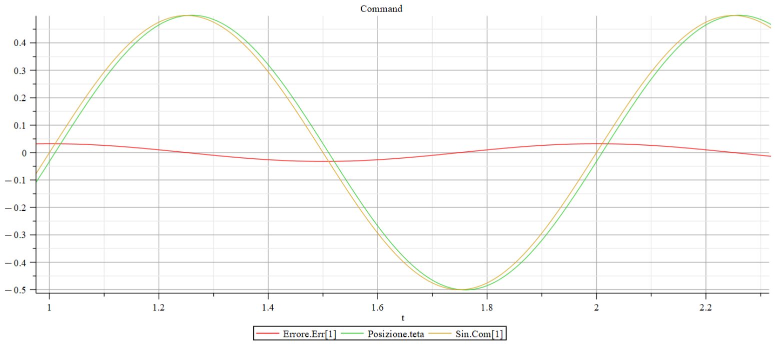


Figure 3.61: Sin input command ($A = 0.5rad/s$; $f = 1Hz$) response detail

- Semiamplitude $A = 0.5rad$ and frequency $f = 5\pi\, rad/s = 2.5Hz$
  In this case, increasing the command frequency increases the phase shift between commanded and actual position.

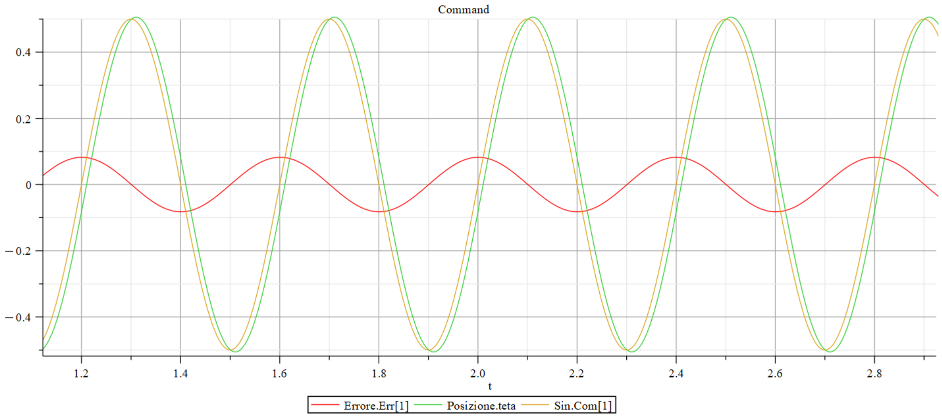


Figure 3.62: Sin input command ($A = 0.5rad/s$; $f = 2.5Hz$) response

- Semiamplitude $A = 0.5rad$ and frequency $f = 20\pi\, rad/s = 10Hz$
  In this case, the phase shift increases significantly. The commanded and actual positions are almost in quadrature, and being close to the cutoff frequency, an attenuation phenomenon can be appreciated.

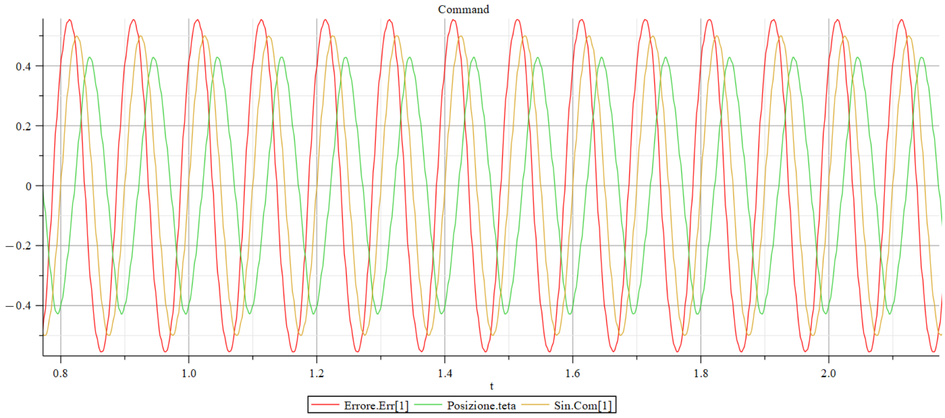


Figure 3.63: Sin input command ($A = 0.5rad/s$; $f = 10Hz$) response

- Semiamplitude $A = 0.5rad$ and frequency $f = 40\pi\, rad/s = 20Hz$
  In this case the response is extremely different. The actual position is very attenuated, because the system is in antiphase (the phase shift is about 180^o) and part of the control is performing an antagonistic behaviour: the disturbance, instead of being attenuated, is being amplified.

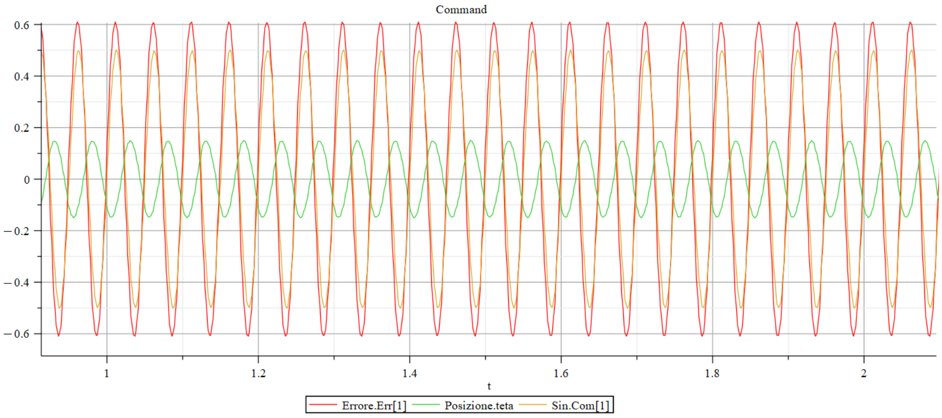


Figure 3.64: Sin input command ($A = 0.5rad/s$; $f = 20Hz$) response

- Semiamplitude $A = 0.05rad$ and frequency $f = 10\pi\, rad/s = 5Hz$
  If the amplitude of the sine wave is reduced, there is no appreciable difference in the response. This happens because the command is small enough not to saturate the system.

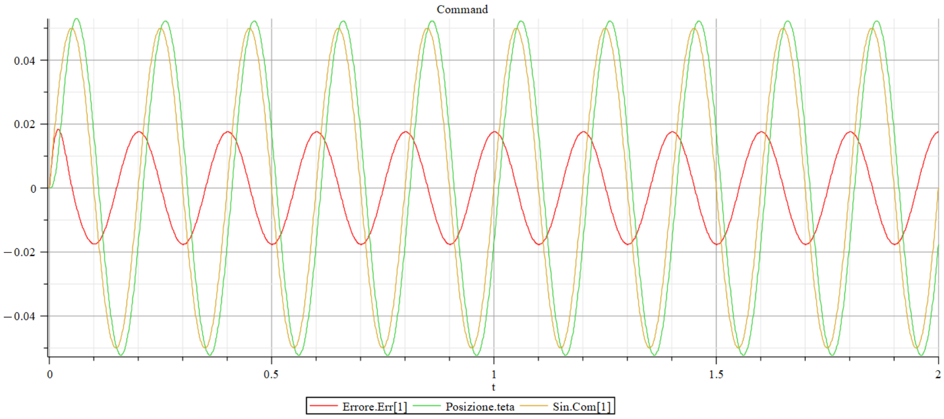


Figure 3.65: Sin input command ($A = 0.05rad/s$; $f = 5Hz$) response

- Semiamplitude $A = 5rad$ and frequency $f = 10\pi\, rad/s = 5Hz$
  If, on the other hand, the amplitude of the sine wave is increased, there is no longer a sine wave as a response. What is shown in Figure 3.66 is a connected sawtooth, which indicates that the amplifier has entered saturation, and the system responds linearly.

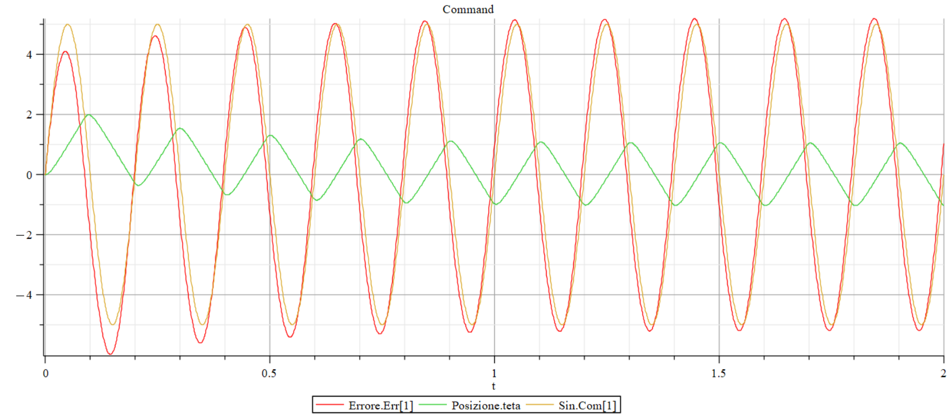


Figure 3.66: Sin input command ($A = 5rad/s$; $f = 5Hz$) response - 1

Increasing the command amplitude means imposing higher velocities on the actuator during the same period. The system, however, is limited by the supply voltage provided by the amplifier, which goes into saturation.

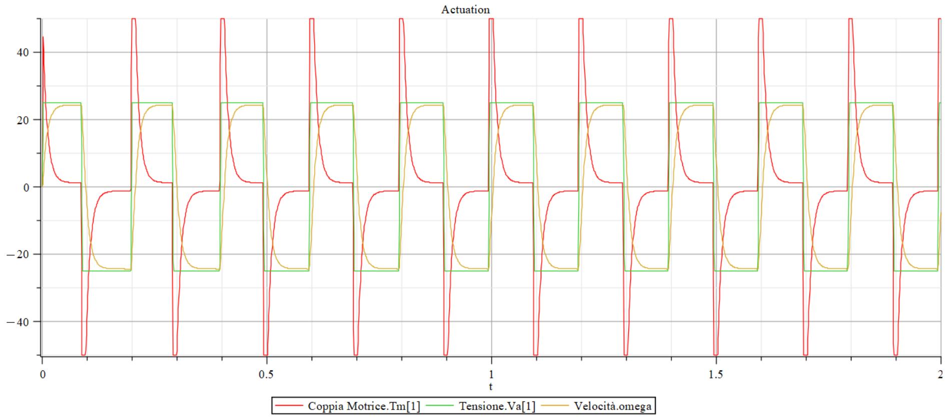


Figure 3.67: Sin input command ($A = 5rad/s$; $f = 5Hz$) response - 2

External Load

We will now reset the input as a step command, and we vary the external load applied to the actuator, in order to analyse the system's response to these new conditions. As a reminder, in the reference model, the external load was defined as $T_L = 25\% TMM$ ($TMM = 50Nm$, the maximum torque that can be developed by the mechanism,equal to the stall load), applied at $t = 0.5s$.

- $T_L = 50\% TMM = 25Nm$ at $t = 0.5s$
  As the load increases, the steady-state error under load obviously also increases: the error is greater because it has to compensate for a higher load.

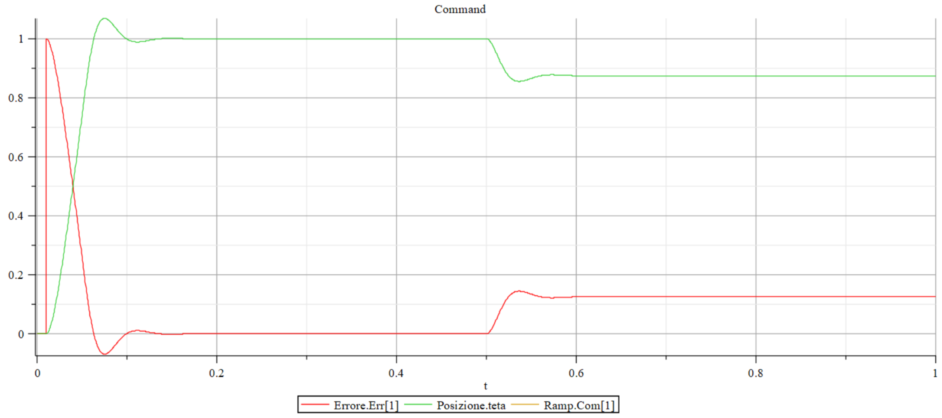


Figure 3.68: External load ($T_L = 50\% TMM$; $t = 0.5s$) response

- $T_L = 75\% TMM = 37.5Nm$ at $t = 0.5s$
  By further increasing the external load, a linear dependency between load and error is observed. The dynamics are fast enough to contrast the effect of the load acting on the moving surface and reach the equilibrium.

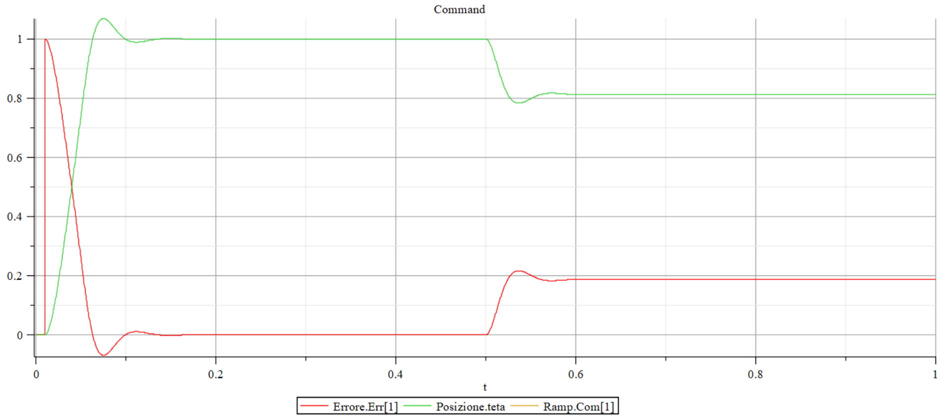


Figure 3.69: External load ($T_L = 75\% TMM$; $t = 0.5s$) response

- $T_L = TMM = 50Nm$ at $t = 0.5s$
  If, on the other hand, the external load is imposed equal to the maximum torque that can be delivered by the actuator, the situation changes.

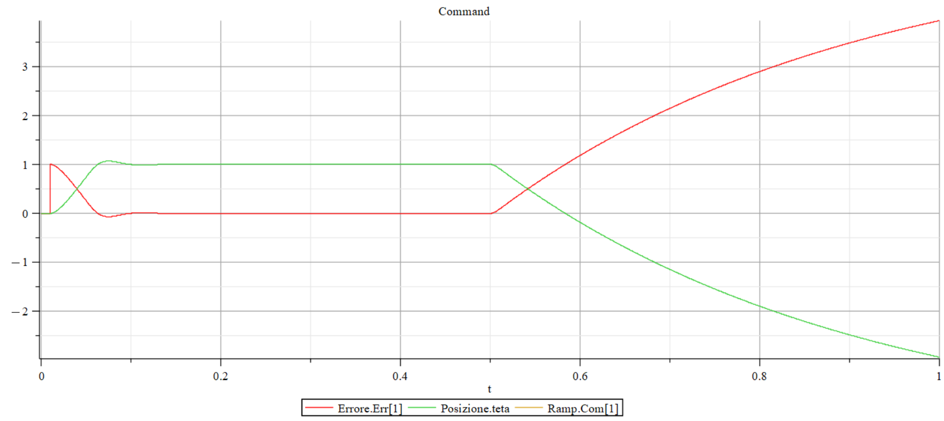


Figure 3.70: External load ($T_L = TMM$; $t = 0.5s$) response - 1

From the equilibrium position, the electric motor receives a very high load (equal to the stall load), which causes the system to regress. As it regresses, the error increases and so does the torque, which quickly reaches its saturation/maximum value (the system is degraded to first order).
In this condition, the resistant torque is balanced by the torque provided by the motor, but in the meantime the system has developed a rather high velocity due to the disturbance. The established equilibrium between the external load and the driving torque is thus disturbed by the viscous actions.

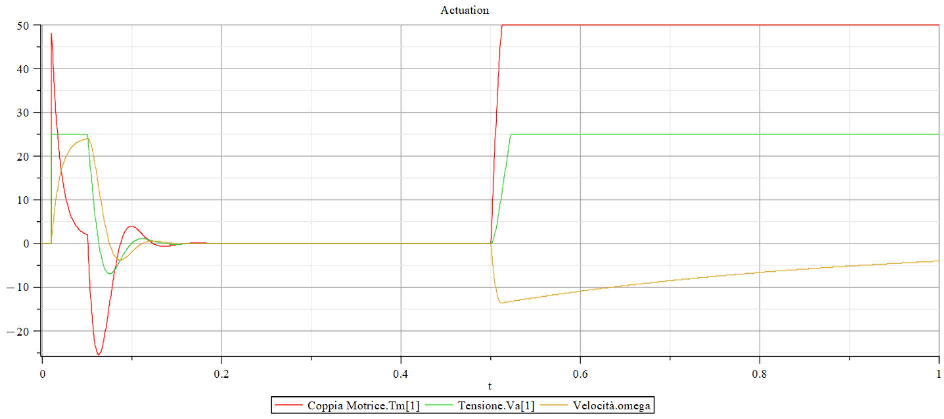


Figure 3.71: External load ($T_L = TMM$; $t = 0.5s$) response - 2

Another peculiarity of this case under consideration is the difference between the time constants. At the beginning, when the step command is given, the system responds very quickly, whereas when the disturbance arrives, the transient to reach equilibrium is much longer.
This is due to the fact that two different degradations come into play. At

start-up, due to the error, the supply voltage goes into saturation while the drive torque remains in the linear range. Instead, when the external disturbance occurs, it is the driving torque that goes into saturation first, and the supply voltage reaches saturation as the moving surface continues to recede under the load.
The system, degraded to first order, will therefore continue to function, but only under the effect of the viscous action of the mechanism's bearings, because the contribution of the counter-electromotive force is inhibited by saturation.

- $T_L = 51Nm$ at $t = 0.5s$

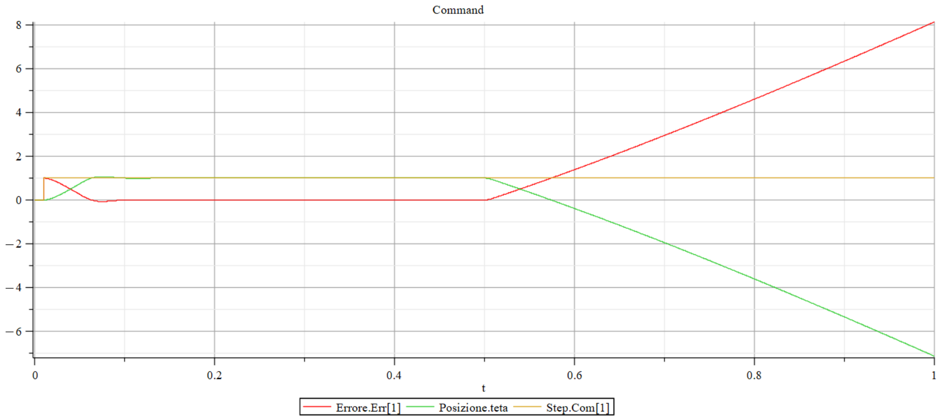


Figure 3.72: External load ($T_L = 51Nm$; $t = 0.5s$) response - 1

By applying an external load that exceeds the maximum torque that the motor can deliver, the system is no longer able to balance the disturbance and the actuation velocity, after a long transient, settles at a constant negative value. Again, the system goes into saturation, but after the transient, it no longer reaches a condition of static equilibrium. When the transient is exhausted, the excess $1Nm$ of the load is compensated by the only remaining counter-reaction loop, the viscous one.
The value of the residual velocity multiplied by the coefficient of viscous friction is exactly equal to $1Nm$.

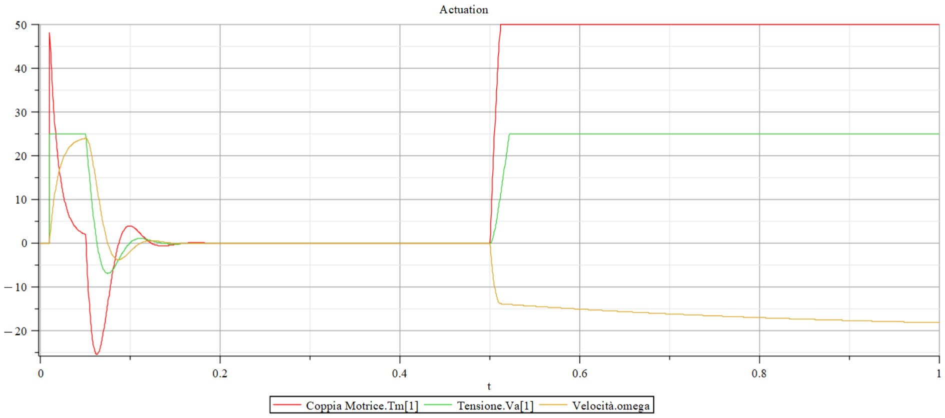


Figure 3.73: External load ($T_L = 51Nm$; $t = 0.5s$) response - 2

PID Control Logic

Whenever, with the same external load, if a reduction in the corresponding error value is required (i.e. the system is required to be more rigid to loads), the EMA can be modified in various ways:

- adopt a motor with a higher torque gain
  (which means modifying the design, increasing both weight and size),
- increase the gain of the proportional GAP controller
  (which means increasing the supply voltage for the same error),
- add a GAI integral term (suitably limited) into the PID control logic.

A Proportional-Integral-Derivative (PID) controller is a feedback-based control loop mechanism that is commonly employed in industrial control systems and other applications that require actively modulated control. A PID controller constantly calculates an error value $e(t)$ as the difference between a desired set-point (SP) and a measured process variable (PV) and makes a correction based on proportional, integral, and derivative terms (denoted P, I, and D, respectively), as the name suggests. The capacity to use the three control terms on the controller output to apply accurate and optimal control is what characterises the PID controller.

The proportional term P is always present in control logic because it is the factor that, multiplied by the measured error, allows the desired setpoint to be reached. Since the controller requires an error to generate the proportional output response, using proportional control alone results in an error between the setpoint and the process value. Under steady-state process conditions, an equilibrium is achieved with a constant SP-PV 'offset'.

The integral term I The I takes into account past values of the SP-PV error and integrates them over time to produce the GAI integration term. For example, if there is a residual SP-PV error after applying proportional control, the integral term attempts to eliminate the residual error by adding a control effect due to the cumulative historical value of the error. When the error is eliminated, the integral term ceases to grow. In this way, the proportional effect decreases as the error decreases, but is offset by the increasing integral effect.

The derivative term D The D is the best estimate of the future trend of the SP-PV error, based on its current rate of change. It is sometimes called 'anticipatory control', as it effectively attempts to reduce the effect of the SP-PV error by exerting a controlling influence generated by the rate of change of the error. The faster the change, the greater the controlling or damping effect.

Returning then to the configuration of the reference model, we will analyse the effects of the PID block on the system's response.

First, let's start by investigating the influence of the proportional term.

- $GAP = 500$

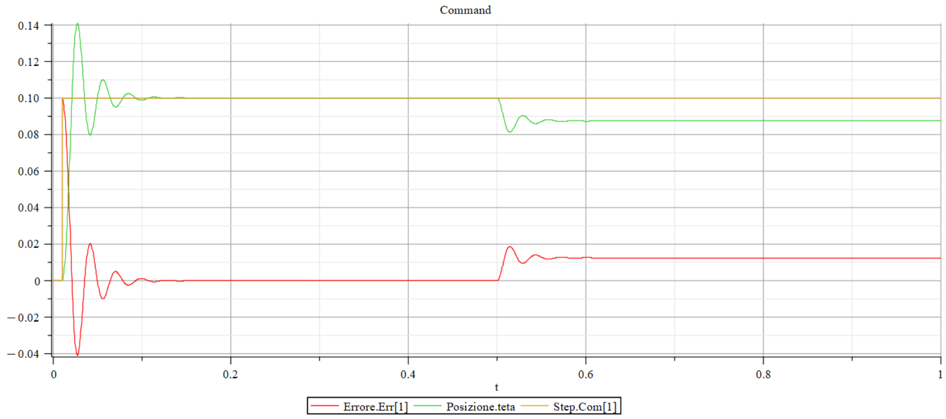


Figure 3.74: PID ($GAP = 500$) response

- $GAP = 2500$
  By increasing the proportional term, it can be seen that the error is reduced, but, at the same time, large overshoots and undershoots arise, which smooth out after an oscillating transient.

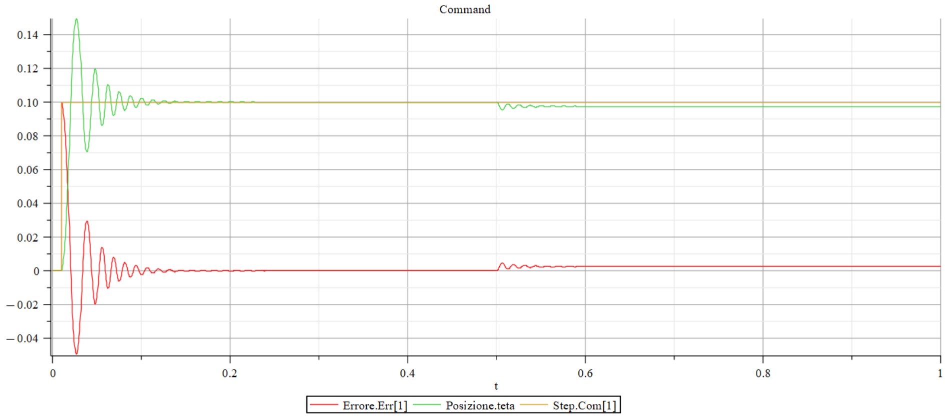


Figure 3.75: PID ($GAP = 2500$) response

- $GAP = 10000$
  Taking the proportional term to the limit leads to instability: the figure below shows the physical limit cycle in which the system is no longer able to respond to the external load.

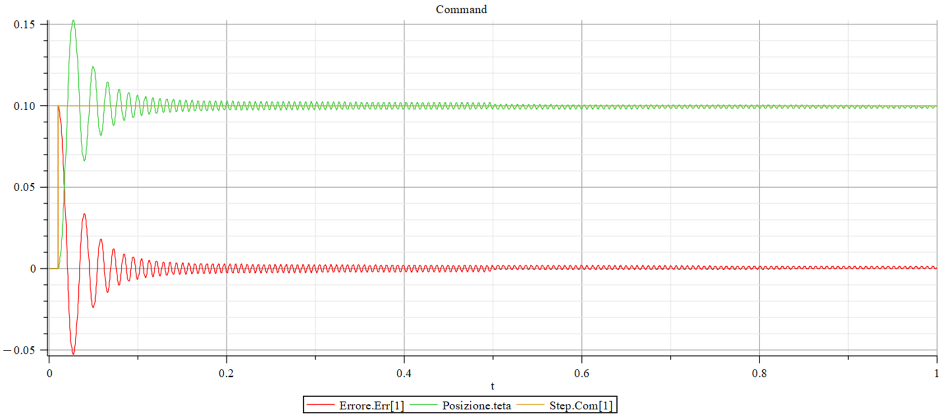


Figure 3.76: PID ($GAP = 10000$) response

In conclusion, what is desired by loading the proportional term is a high impedance to external loads, but without leading to instability.

Next, we introduce the integral term, used in conjunction with proportional one to compensate for long-term disturbances.

- $GAI = 1000$, $GAP = 100$
  By adopting a PI control, after a certain transient depending on the value of the integral term set, the error is reset to zero.

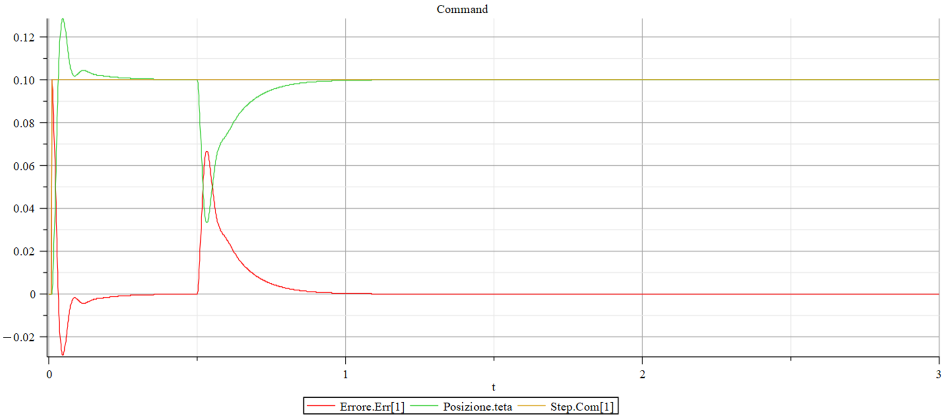


Figure 3.77: PID ($GAI = 1000$, $GAP = 100$) response

- $GAI = 5000$, $GAP = 100$
  Increasing the gain of the integral reaches the setpoint faster.
  Thus, the system's response becomes quicker, but at the same time, the value of the supply voltage (proportional to the integral of the error in time) increases, which can lead to complications in the actuation.
  The presence of the integral tends to reduce the stability margin of the system, resulting in the peaks seen in the figure below.

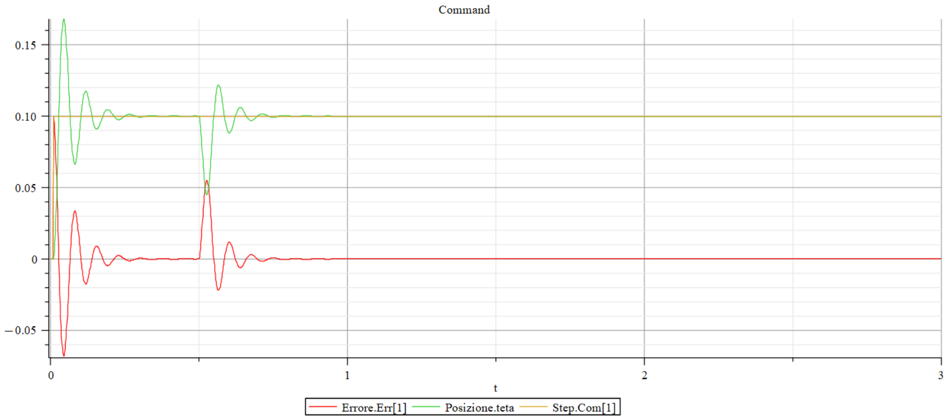


Figure 3.78: PID ($GAI = 5000$, $GAP = 100$) response

- $GAI = 10000$, $GAP = 100$
  Taking the integral term to the limit leads to instability: the figure below shows the physical limit cycle in which the system is no longer able to respond to the external load.

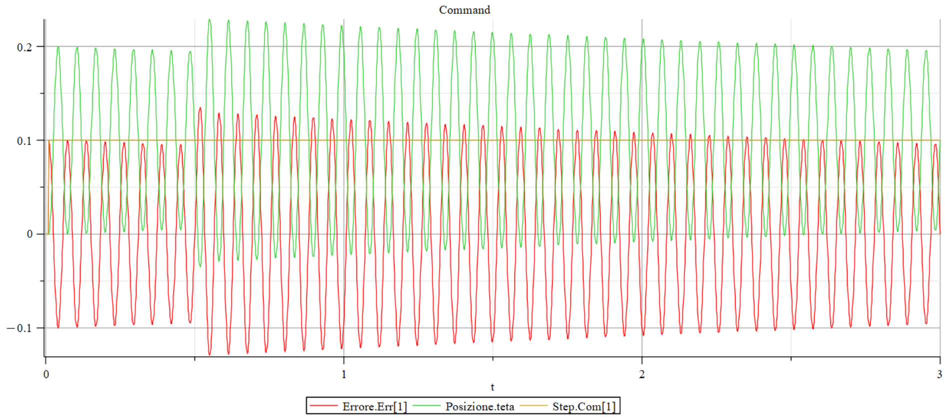


Figure 3.79: PID ($GAI = 10000$, $GAP = 100$) response

When using PI control logic, attention must be paid to the value of the integral. For this purpose, filters, known as anti-windup methods, are added to prevent the system from saturation due to the integral term.

Finally, we add the derivative term to the control logic, used to return the response to the stability margin.

- $GAD = 1$, $GAI = 100$, $GAP = 100$
  By introducing a small value of derivative gain, the oscillations created by the integral term (which reduces the stability margin) are dissipated.

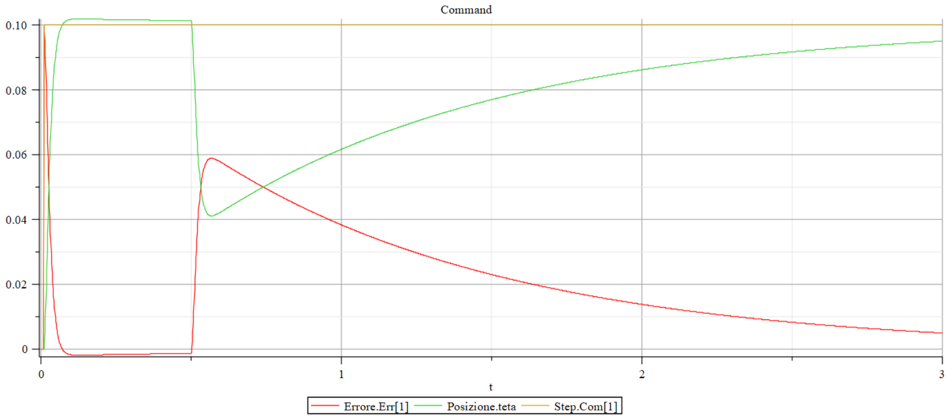


Figure 3.80: PID ($GAD = 1$, $GAI = 100$, $GAP = 100$) response

> Instead of the peaks previously seen, there is a small overshoot that slowly attenuates due to the integral term.

The use of the PID algorithm does not guarantee optimal control of the system or its stability. The controller's response can be described in terms of responsiveness to an error, the degree to which the system overshoots a setpoint and the degree to which the system oscillates.
Although a PID controller has three control terms, some applications require only one or two terms to provide adequate control. This is achieved by setting unused parameters to zero and is called a PI, PD, P or I controller in the absence of the other control actions.
Thus, when using PID control logic, it is necessary to be careful about the values one assigns to the three terms because the system is very sensitive to their influence. The balance of these effects is achieved by calibrating the tuning constants to produce the optimum control function, and must be derived for each control application. Approximate values of the constants can usually be initially entered knowing the type of application, but are normally refined, by 'bumping' the process into practice, introducing a setpoint change and observing the system's response.

Chapter 4

Co-Simulation

By combining several simulation disciplines, co-simulation offers engineers unique, more thorough, and holistic performance insight. MSC CoSim can integrate everything from acoustics to multibody dynamics (MBD), Computational Fluid Dynamics (CFD), structural analysis, and explicit crash dynamics. Being capable of rapidly and easily examining solutions with co-simulation, for instance, at a time when the engineering team is not yet committed to any particular strategy should allow for meeting performance criteria with lighter suspension, which can enhance a vehicle's fuel efficiency.

The MSC CoSim engine aims to operate as a co-simulation interface for the direct integration between different solvers and disciplines within a multiphysics framework. Engineers can use MSC CoSim to create co-simulation models between Adams, Marc, MSC Nastran, scFLOW, and scSTREAM. However, Hexagon supports a variety of co-simulation approaches in addition to the CoSim engine, such as the Functional Mock-up Interface (FMI), Adams Marc Co-Simulation Interface (ACSI), and more.

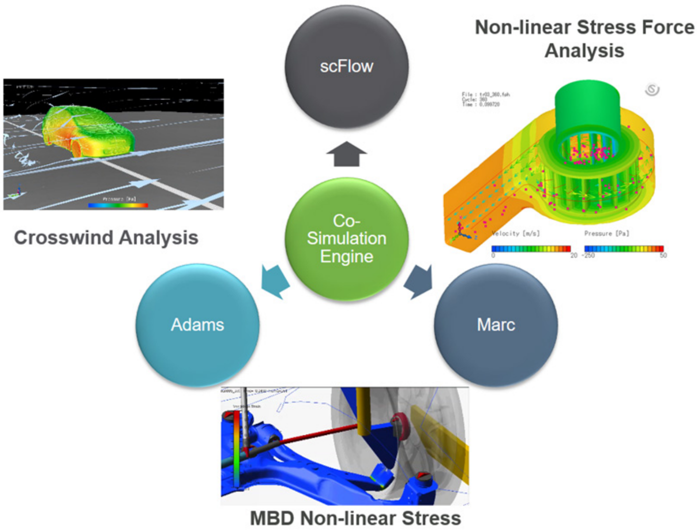


Figure 4.1: MSC CoSim engine

Engineers can use Hexagon's solutions in two ways, depending on the type of analysis: Co-Simulation (applying several physics to the model at the same time) or Chained Simulation (passing load case results from one analysis to the next). Chained simulation enables CAE engineers from several divisions to successively combine different fields and increase the overall simulation accuracy. For example, providing the road load data from an Adams Full Vehicle model to the downstream MSC Nastran model for the stress and durability tests.
Co-simulation, on the other hand, is of two types:

- Co-simulation o Discrete mode: Specifies that one application solve the mechanical system equations and the control application solve the control system equations. The two applications exchange inputs and outputs at predetermined time interval also known as the communication interval or co-simulation interval.
  For most analyses, the discrete mode is generally the more efficient simulation method. It is faster and can handle complex models better than continuous mode. You should use continuous mode when equations solved in the control system would cause a large coupling effect on the Adams data. For example, you might prefer to use the continuous mode if your analysis requires a very small time step.
  To preserve the proper dynamics for a mechanical system, discrete mode should sample the mechanical system at least five times greater than the highest frequency of interest. If the time step is too small to sample at five times the highest frequency, then you should use continuous mode.

- Function Evaluation or Continuous mode: Specifies that the control application solve both the mechanical and control system equations.
  In continuous mode, the control system package solves the equations of motion for the combined controls and mechanical dynamics system. Adams formulates the equations of motion for the mechanical subsystem and provides information to the controls package as needed. Typically Adams simulations involve solving a set of differential (equations of motion) and algebraic (constraint and applied force) equations, known as a DAE system. The Adams integrators are specially tuned for efficient solution of this type of problem. Using coordinate partitioning, Adams can also formulate the equations of motion based on a minimum independent set of system states. This results in a compact ordinary differential equation (ODE) formulation of the equations of motion. Typically this system of equations is more difficult to solve numerically than the DAE method described above.
  The benefit of this ODE formulation is that the state values and time derivatives can be easily communicated to the controls package. Using this information, the controls package can form a combined ODE system describing

> the controls and mechanical states. This is the problem that the controls package then solves with its own integration techniques. When a new set of state values is found in the controls package, Adams is called to find the time derivatives of these states, based on the ODE formulation that Adams has created for the mechanical states. This is why it is known as "function evaluation" mode. Adams simply sets up the equations of motion and provides time derivative information for the states in this mode.

To perform co-simulations between Adams and Elements, FMI approach must be addressed. Functional Mock-up Interface (FMI) is an open standard to support the exchange of simulation models and coupling between different tools. It provides two user scenarios: Model Exchange and Co-Simulation. Currently, both Adams and Elements provides the support for FMI Co-Simulation (both Parent and Child mode) and Model Exchange (Parent mode).

In the FMI Parent mode, Adams (or Elements) could import external models that conform to the FMI CoSimulation/Model Exchange standard. The model is also called Functional Mock-up Unit (FMU). The user needs to specify the communication interval in case of co-simulation and could then perform simulation using Adams (or Elements) as the parent.

In the FMI Co-Simulation Child mode, the user could export an Adams (or Elements) model as an FMU, and then use it in any software which supports the FMI Co-Simulation parent mode.

The FMU file is a zip formatted file which contains a model description file in XML format, a binary file for the model as well as optional resource files and source codes for the binary file. The model description file provides information such as input/output and capability flags.

Hexagon products support SmartFMU standard, too.

The SmartFMU is an enhanced Functional Mock-up Unit file that simplifies inspection and editing of the contents of an FMU. This standard extends the FMI specification to include model files, a model schematic diagram and extended documentation. The primary application of a SmartFMU is to facilitate interfaces between Adams, Elements and Easy5. Typically, a FMU is a 'black box' that is difficult to inspect & modify. The SmartFMU makes it easy to inspect, modify and update the contents of the FMUs that are shared between the tools. A SmartFMU is intended to simplify the tasks of model creation and modification, and it makes it easy to view and edit the FMU source directly from within the host tool (12).

4.1 Adams: Multibody Analysis

Multibody analysis is the process of simulating the motion and interaction of multiple bodies or components within a system. In the context of actuation systems, multibody analysis can be used to predict and optimize the performance of complex mechanical systems, such as aircraft flight control systems.
Multibody system analysis software is an essential component of modern Mechanical Computer Aided Engineering (MCAE) practise since it allows you to model the operating performance of a product design before manufacturing a prototype. As a result, product development expenses are reduced, more alternative designs can be evaluated, and the time it takes to bring a new product to market is cut.

Adams is an extremely capable modelling and simulation environment. It can be used to create and simulate a model of any mechanical system with moving parts. You can test your model in the same contexts that your final product will be exposed to. Additionally, you may check your model against test data and refine it with flexible parts, realistic geometry, and controls. Finally, you may identify the critical characteristics that influence your product design and determine the optimum combination of design values to achieve the best product design. The best part is that you can do all of this over and over again, saving time and money.
Adams provides a graphical interface for setting up and running multibody simulations, it supports engineers in analysing the dynamics of component motion and the distribution of loads and forces in mechanical systems by reproducing the behavior of the systems in a virtual environment. Product designers frequently fail to understand real system performance until late in the development cycle. Mechanical, electrical, and other subsystems are validated against their specific requirements during the systems engineering process, but full-system testing and validation occurs later, resulting in riskier and more costly rework and design modifications than those performed earlier.
Adams, the world's most popular and commonly used Multibody Dynamics (MBD) environment, boosts engineering efficiency and lowers product development costs by enabling early system-level design validation. Engineers may better optimise product designs for performance, safety, and comfort by evaluating and managing the intricate relationships between disciplines such as motion, structures, actuation, and controls. It is optimised for large-scale issues, taking advantage of high performance computer systems, and has extensive analysis capabilities. It runs non-linear dynamics in a fraction of the time required by FEA solutions by applying multibody dynamics solution technology. Adams simulations increase FEA accuracy by providing a better understanding of how loads and

forces vary throughout a complete range of motion and operating situations. It offers optional modules that let users to integrate mechanical components, pneumatics, hydraulics, electronics, and control system technologies to create and test virtual prototypes that accurately reflect the interactions between multiple subsystems (12).

4.1.1 Adams Controls

In addition to its standalone capabilities, Adams also supports co-simulation with other software tools, such as control systems design software. Co-simulation allows to integrate the mechanical system model with control system models and simulate the behavior of the entire system under different conditions. This approach enables engineers to optimize the mechanical actuation system design while taking into account the behavior of the control system, resulting in more accurate and reliable system performance.
Adams Controls is an Adams plug-in that allows you to easily include advanced control systems into mechanical models. It includes modelling elements that transport data to and from the control system model.

The mechanical designer and the controls designer typically start from the same concept but apply distinct kinds of software tools in the traditional design process of a mechanical system with controls. As a result, each designer develops a model for the same problem.Each model is then verified and tested, and the two models are brought together for the first time during physical prototype testing. If an issue arises during the interaction between the control design and the mechanical design, the engineers must refine either the control design or the mechanical design, or both, and then repeat the entire verification process.

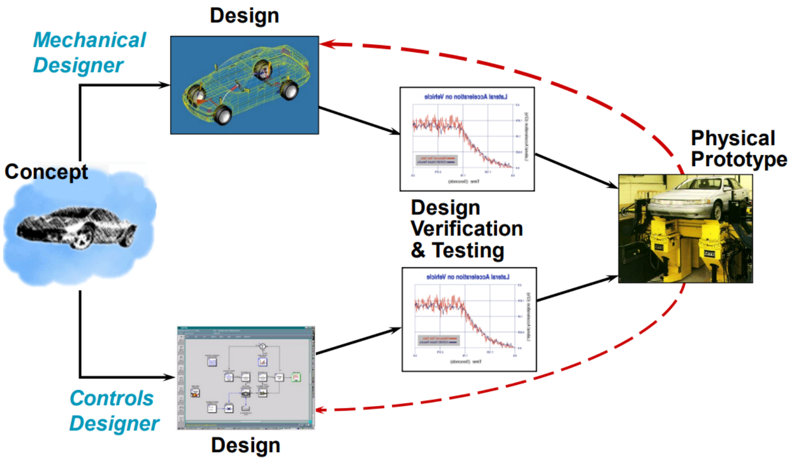


Figure 4.2: Traditional design process

The two designers may employ the same mechanical model with Adams Controls.

They can also validate the overall impact of a control system on a nonlinear, non-rigid model with just one database. The physical testing procedure is significantly optimised, and the chance of the control logic being incompatible with the physical system is eliminated (13).

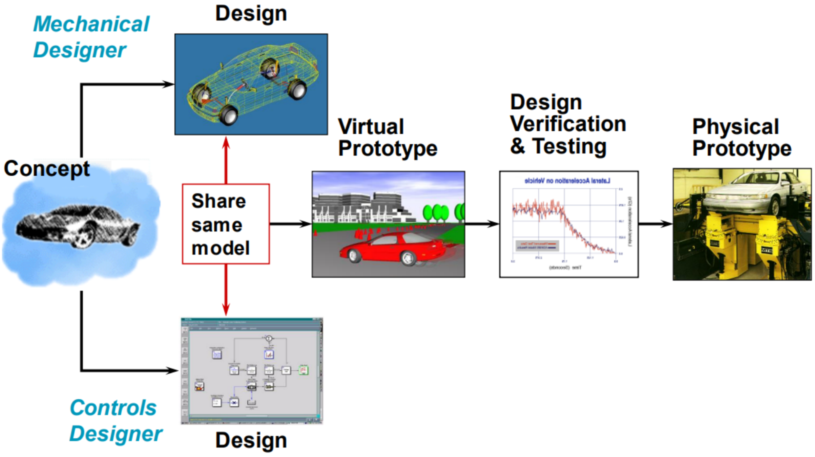


Figure 4.3: Design process with Adams Controls

The process of combining controls within a mechanical system can be described in a four-step plan.

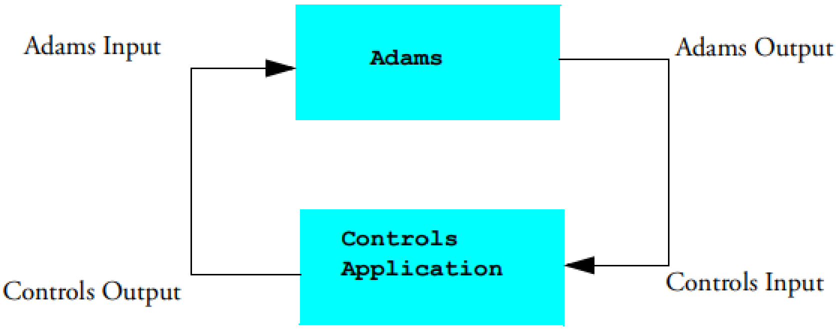


Figure 4.4: Adams Controls four-step design process

1. Build the Model
   To begin working with Adams Controls, create or import an Adams model. The model should be comprehensive, including all required geometry, constraints, forces, and measurements.

2. Create the Adams Inputs and Outputs
   The outputs define the variables that pass to the controls application (the Adams model output is the input to the controls system). The inputs specify the variables that return to Adams (the controls application's output) and therefore establish a closed loop between Adams and the controls application. Every input and output must be configured as a state variable.

Following this, you export the plant system files from Adams for integration into the control simulation software.

3. Build the Block Diagram
   Build the block diagram of the control system with Easy5, Elements or Matlab/Simulink, including the Adams plant within the block diagram.

4. Simulate the Model
   Simulate the mechanical model combined with the control system using one of several available methods, illustrated above.

This design process will be described in detail on the following pages, through a practical example applied to high lift devices.

4.2 High Lift Devices: Flaps Actuation

As the clean wing design is optimised for the cruise speed regime, today's commercial aircraft are designed with high lift devices to offer lift augmentation at low speeds during take-off and landing. The advent of jet engines in commercial aircraft required an increase in cruise speed capabilities in order to operate the jet engine at its optimum setting based on the engine's characteristics. The enhanced cruising speed was achieved through wing sweep and airfoil improvement; additional performance optimisations resulted in greater wing loads. However, increased wing loading implies larger lift coefficients at low speeds, whereas wing sweep reduces wing lift at low speeds. More powerful high lift devices were crucial to keeping take-off and landing speeds within the allowed ranges.
Plan flaps gave way to Fowler flaps with single, double, and even triple slots as wing trailing edge devices. Since then, with the only necessity for low speed performance, the trend in high lift system research has been to accomplish high levels of lift with simpler devices and optimised actuation systems in order to reduce complexity, weight, and maintenance costs. As a result, modern high lift actuation systems primarily consist of a mechanical gearbox shaft system that distributes mechanical energy from a central motor to rotary or ballscrew actuators situated along the gearbox shaft and moving the high lift surfaces. There are further high lift actuation systems that employ synchronised hydraulic actuators with common control valves. All of these configurations ensure synchronous deployment of all flap and slat panels, although they limit functional flexibility.

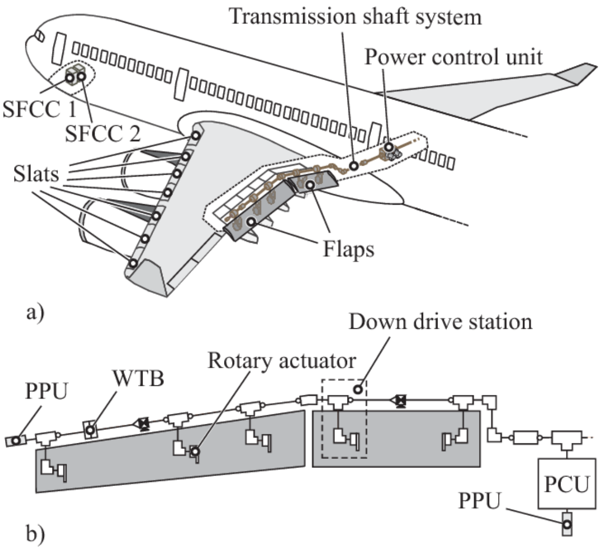


Figure 4.5: High Lift actuation system

Today's high lift system architectures' optimisation capabilities have reached a plateau and are restricted to minor local gains. An improved high lift system architecture with distributed active-controlled flap actuators allows for the deployment of extra trailing edge features, with benefits on the aircraft level and advantages in manufacture and assembly. Therefore, the distributed high lift system design can be considered a technological step change that opens up future performance improvements. Once further flexibility is introduced, there is the possibility of subsequent high lift system technology evolutions by extending other functionalities with greater confidence in this design.

Figure 4.5 depicts a high lift actuation system. The central hydraulic power control unit (PCU) supplies the power required to operate the flap panels on each wing. The mechanical power is transmitted to the rotary actuators, which move the flaps on the tracks, via a mechanical gearbox shaft. This shaft system is made up of shaft parts that have gearboxes for larger direction shifts, as well as system torque limiters, wing tip brakes, universal joints, plunge joints, and spline joints to compensate for wing bending and temperature effects. Each wing's gearbox system has five driving stations. The outboard flap has three drive stations, whereas the inboard flap has two. A down-drive gearbox, a down-drive shaft, an input gearbox with a torque limiter, a cross shaft, and the rotary actuator comprise each drive station. An interconnection strut links each wing's inner and outer flaps and operates as an additional load path in the case of a drive link disconnect at drive stations 1 (inboard), 2 and 3.
Two slat flap control computers (SFCC) collect data acquired from various analogue and discrete sensors to regulate and monitor the high lift system.The mechanical transmission shaft ensures synchronised deployment of all flap panels but limits practical flexibility. This sort of mechanical gearbox shaft system has a significant number of components with distinct part numbers and requires a large design-engineering and installation effort.There are further high lift actuation systems that use synchronised hydraulic actuators with common control valves; each panel is coupled to two actuators that are supplied by redundant hydraulic systems but controlled by a common mode control valve. Mechanical synchronisation ensures that all flap panels deploy at the same time.

An advanced high lift system architecture with distributed active-controlled flap actuators is a technological leap rather than an evolutionary step in high lift system technology. The concept of a highly flexible system with independent controlled flap panels as an enabler for additional functionality for the wing trailing edge, as well as advantages in design-engineering, fabrication, and assembly, motivated the creation of this topology.
The distributed flap drive system (DFDS) guidelines, defined from the reference aircraft, include installation space, loads, aero and system performance requirements, and weight competitiveness. The fully independent flap actuation system consists of an electromechanical actuator at each drive station and no mechan-

ical linkage (transmission shaft) between them. Thus, synchronous deployment or controlled differential deployment has to be ensured electronically by actuator control electronics (ACE) and slat flap control computers (SFCC).

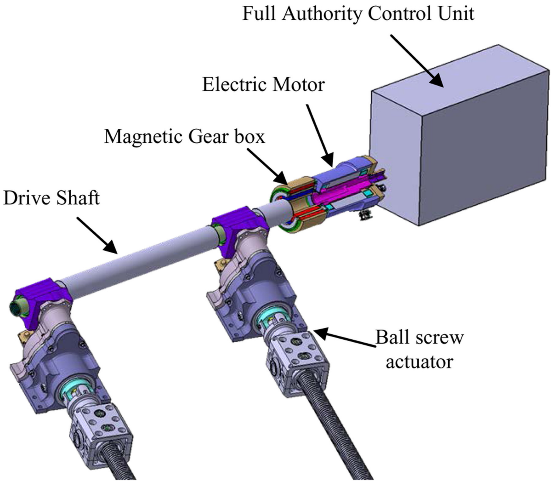


Figure 4.6: Distributed active-controlled flap actuator

In contrast to the conventional PCU architecture, in which two (hydraulic) motors are coupled by a speed-summing differential gearbox, the DFDS system's electric motors are developed with built-in redundancy. In a conventional motor, one motor phase is made up of multiple series- or parallel-connected windings distributed across several slots. Each stator tooth has one focused winding to achieve integrated redundancy. Furthermore, in order to be genuinely independent from other coils, each of these coils must be activated autonomously. As a result, a failure such as an open or short circuit in one coil cannot spread or influence the next coils, making the equipment fault resilient to some extent.

While a typical solution with active or standby redundancy would result in 200% weight and 50% performance for a single electric failure, when compared to a single standard motor designed for 100% performance, intrinsic redundancy should result in 120–140% weight and approximately 70–90% performance for a single electric failure. Obviously, the machine size and the number of coils have a significant impact on this.

Power electronics (PCE) and actuator control electronics (ACE) manage and track the flap panel actuators. The ACE and PCE form a cascade control loop structure, with the PCE controlling current and speed and the ACE controlling position and trajectory targets. The ACEs ensure that both actuators of one flap panel deploy synchronously, preventing twist or skew of the flap panel. The position command is received by the ACEs from the SFCCs, which regulate and monitor the entire high lift system and ensure synchronous or controlled differential deployment of all flap panels (9).

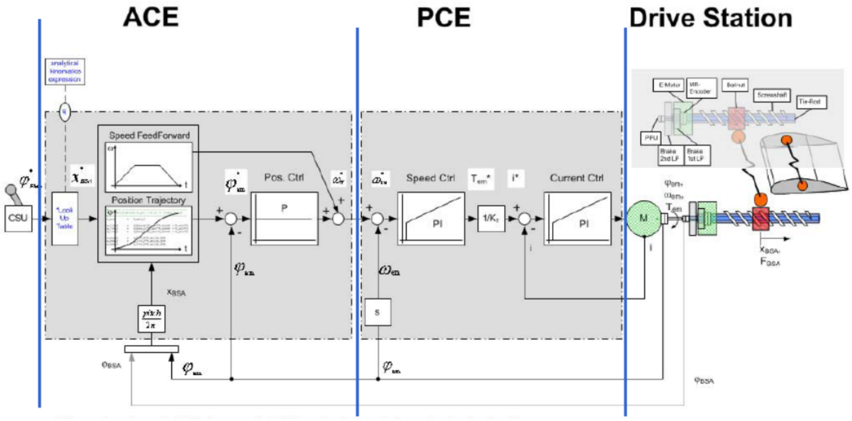


Figure 4.7: Flap actuator control structure

4.2.1 Case Study

The design of movable actuation mechanisms, as well as the related control surface systems, is heavily influenced by wing design, the aerodynamic performance of the wing in different flight states, and the functions that the movable must address. High lift systems, in particular, are frequently made up of complex structures and mechanical drive systems. While traditional high lift devices were developed to reconfigure the wing in fixed configurations to match different flight phases in take-off, approach, and landing, advanced high lift devices offer a higher level of functional flexibility to allow enhanced cruise performance as well as continuous control of span-wise lift distribution to reduce loads on the wing structure. The addition of further capabilities to a device might involve the development of innovative system components as well as an adaptation of the system architecture. Both traditional and advanced high lift system designs require considering system safety (i.e., redundancy standards and monitoring operations) and occurring loads.

Accurate testing efforts are required to assess the feasibility of designs in operational situations. However, testing is usually only viable when physical prototypes are accessible. Furthermore, system-level tests involve not only prototypes of individual components but the entire system. Also, the interaction of systems and structural elements of the wing and control surfaces can have an effect on the behaviour and functionality of control and monitoring capabilities. This frequently results in fairly delicate rig assemblies. The test rig itself introduces new degrees of complexity, incurs additional expenses, and increases the lead time.

As a result, the tendency is towards a broader variety of virtual testing activities in order to acquire a more in-depth understanding of system behaviour in different operational phases. These include, in addition to nominal functioning, failure events in which very transient loads must be accounted for in the design of actuators, drive shafts, mechanisms, and structures. Establishing verification loops at the system level early in the design process allows for the identification of faults or vulnerabilities in the architecture design, reducing the risk of major revisions later in the design stage.

The term virtual testing describes the evaluation of properties and behavior of the system under test through numerical simulation. Virtual testing for structural mechanics mainly focuses on the methods used for modelling and simulation in the structural mechanics domain. For flight control actuation systems, the basis for virtual testing is the assurance of system properties with respect to underlying system requirements. The choice of the modelling and simulation technique depends on the type of result, which shall be obtained. Moreover, the objective of the virtual tests defines, where most emphasis on the process has to lie. If, e.g., virtual testing is applied to gain knowledge of the system a priori to building prototypes in early development phases, credibility of the modelling techniques as well as experience may sufice for the intended use. If, on the other hand, virtual testing shall be used in product qualification and certification, extensive care must be taken considering the validity of the models and the remaining uncertainties (10). For the present work, the focus is on co-simulation methods between Adams, for multibody analysis, and Elements, for control analysis, verifying the integration between the two numerical simulation environments. Therefore, the testing process will be milder than that used in a design process or, even more so, in a certification process, precisely because the aim of this work is not to analyse virtual methods of developing and testing flight control systems, but to verify and validate the use of this new multiphysics modelling software, exploiting co-simulation capability.

The flight control system development process requires the combined efforts of multiple areas: vehicle dynamics, flight loads, structures, mechanisms, control systems and electrical or hydraulic systems. It is a very complex system that in the past was designed in separate environments, integrated and physically tested only at the end of the development and prototyping phase, causing a significant design cost. But now, thanks to advances in virtual integration technologies, it is possible to have a complete environment for the design and analysis of flight controls:

- Effect of wing warping on mechanism due to structural deflection
- Distributed aero loads vs. point load
- Environmental disturbances – thermal, aerodynamics
- Test complete aircraft and flight controls through flight maneuvers

- Nonlinear controls that effect flying qualities and overall safety
- Evaluate hydraulic actuation and effect of back force
- Effect of preload condition, excessive friction or system stiffness in mechanism design
- Perform tradeoff studies of surface flexibility, track shape, drive system authority, and configurations
- Perform systems integration before physical assembly

This allows designers to reduce risk and physical prototyping, which translates into important time and cost saving.

Within the MSC Software - Hexagon company portfolio, several tools are available to facilitate this task.

Adams, the mechanical motion simulation tool, is useful for multibody dynamics and kinematics, rigid and flexible body dynamics, and interfaces with CAD, FEM, control systems, and more.

Easy5 in the past, now Elements, are system-level modelling tools based on a schematic block diagram view of engineering systems. They facilitate modelling, design, and linear and non-linear analysis applying a multi-disciplinary and multi-physics approach (hydraulics, pneumatics, controls, electric drives, mechanics, and thermal controls).

Nastran (with Patran as pre-post) is the world's most popular and reliable finite element analysis solver. It allows simulation from a single component to complex assemblies under different conditions and supports superelements. It offers a full set of linear static and dynamic analysis capabilities, as well as a range of implicit and explicit nonlinear analysis capabilities. In addition,it allows coupling between various disciplines, such as thermal, structural, and fluid interaction.

Cradle is an advanced multiphysics computational fluid dynamic simulation tool used to analyse complex thermal and fluid phenomena. It uses Navier-Stokes equations for fluid dynamics and other specific models for complex phenomena. It also makes it possible to predict the performance of components before performing experimental tests, which saves a great deal of time and money. It can be implemented in a wide range of industries, as CFD can study any system that involves fluid dynamics and heat transfer.

By bringing all these tools together, it is possible to perform a multidisciplinary analysis, starting with the definition of the geometry of the model using CAD software, followed by the verification and compensation of the kinematics and dynamics of the mechanisms, and then the analysis of structural stresses and strains under load by integrating flexible body models, then adding the control logic of the actuation systems, and, finally, the aerodynamic loads for the entire system.

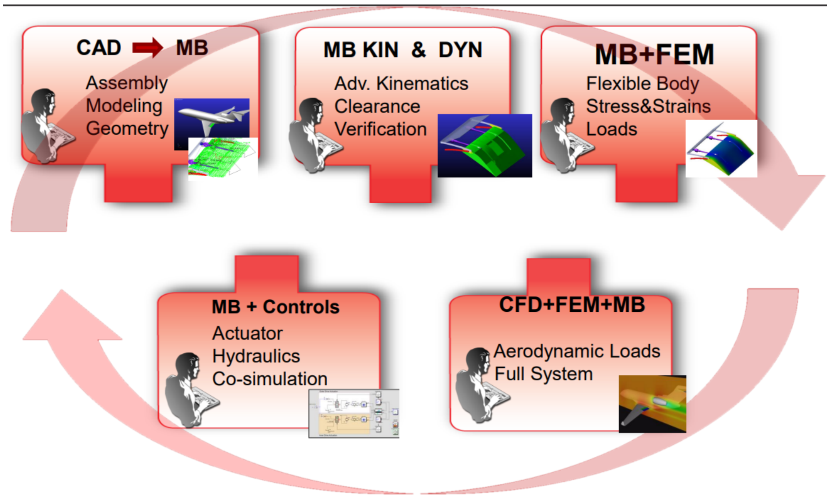


Figure 4.8: MSC Software multidiscipline analysis

In Adams, it is possible to directly import the geometries of assemblies modelled on CAD software, assign inertia, and perform preliminary analyses.

With regard to kinematics, evaluation of applied motion, motor torque, and actuator force can be carried out, taking into account external forces, weight, inertia, and friction, plus kineto-static performance evaluation, backlash determination, and preliminary failure analysis.

On the other hand, with regard to dynamics, assessment of actuator forces, evaluation of structural loads, friction, compliance, verification of assembly errors, structural deformation, fault investigation, and verification.

By integrating a finite element model into the multibody analysis, it is possible to perform a dynamic analysis on flexible bodies. This provides additional information regarding the effect of the fully flexible model (structural loads, errors, vibrations, and deformations), a stress and strain evaluation, and, finally, a durability and fatigue analysis.

Lastly, thanks to the Adams Controls plug-in, it is possible to perform mechatronic analysis via co-simulation. A control system can be defined by realistic actuator modelling and evaluating its interaction with the multibody assembly. This integration between control systems and multibody analysis will be discussed in detail within this chapter. First, however, it is necessary to define the configuration of the falp model to be examined.

For kinematic actuation design, the starting points are aerodynamic requirements, which are closely related to geometric requirements. That is, define the coordinates of point A (in Figure 4.9) for take-off and landing, means giving geometric constraints for the coordinates x_A , y_A.

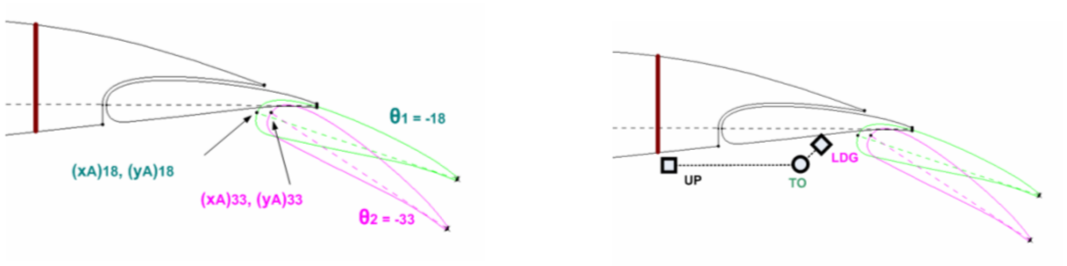


Figure 4.9: Flap geomentrical contraints

Next, the objectives to be met by each flap configuration available for this purpose must be identified:

- Meet aerodynamic requirements, identify flap point coordinates for cruise, take-off and landing (based on the motion law)
- Avoid interferences between the flap and the wing housing, considering that a minimum gap is required
- Operate in the allowed range considering links and bearings (internally of spar profile and considering the fairing profile) and maintain flap integrity (position of bearings and constraints out of flap profile)

Once the project objectives have been established, all valid configurations that may allow for different solutions are identified.

1. Configuration:

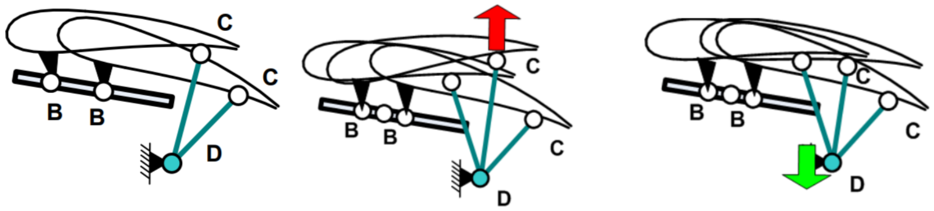


Figure 4.10: Rear Link with Straight Track

2. Configuration:

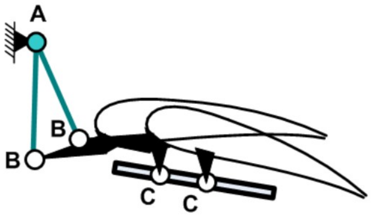


Figure 4.11: Fore Link with Straight Track

3. Configuration:

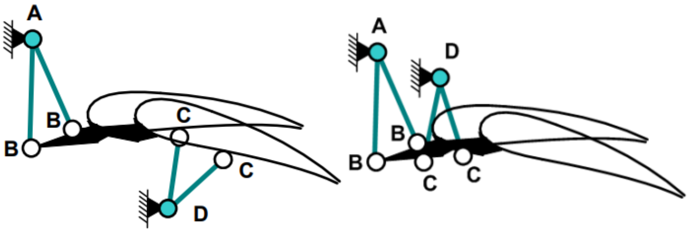


Figure 4.12: Fore Link with Rear Link or Fore Link

4. Configuration:

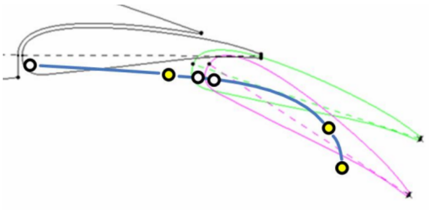


Figure 4.13: Curved track

5. Configuration:

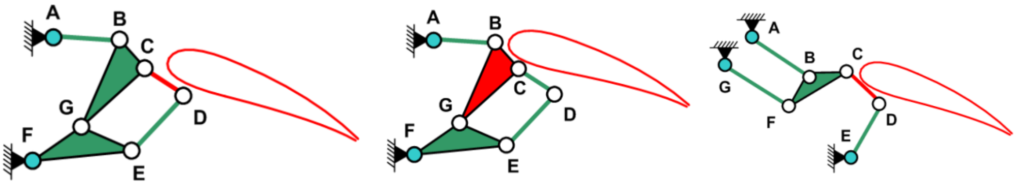


Figure 4.14: 6-bar Linkage

The preliminary design phase consists of choosing the kinematic mechanism. Thus, a simplified flap assembly is imported from CAD to obtain geometric information for the positions and interferences of the bearings and linkages, and to define the flap trajectory for extraction and retraction in cruise, take-off and landing. The CADs imported into Adams are shown here.

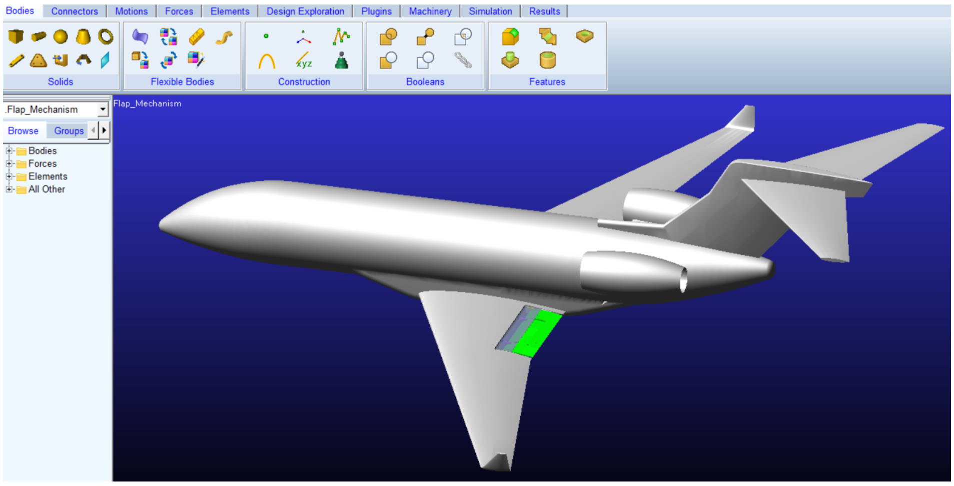


Figure 4.15: Aircraft CAD model

A simple Adams model of the flap is built, fully parameterised in terms of points and coordinates for the joints and links of each configuration to be studied.

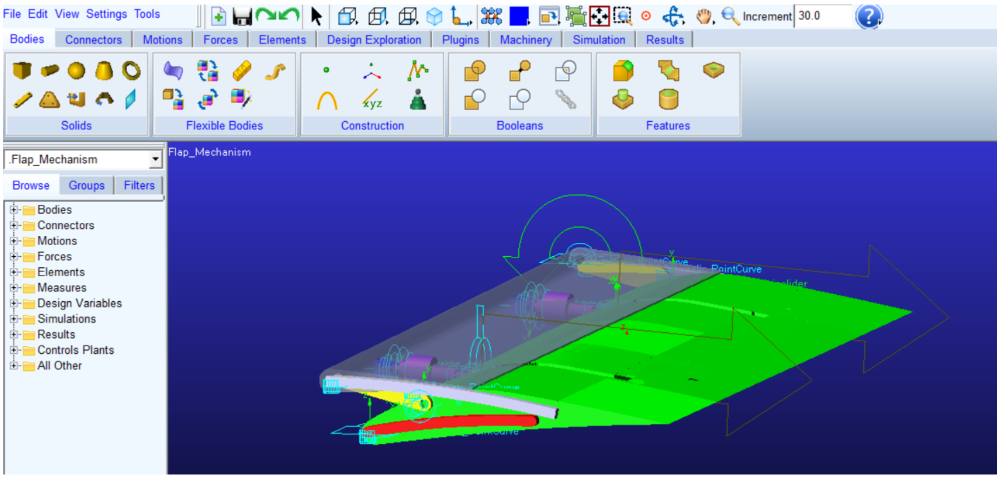


Figure 4.16: Adams flap model

The flap system consists of an internal flap, a double screw and hydraulic, electric or hybrid motors (depending on the actuation type defined during the design process). In Figure 4.17, it is possible to have a look at the different elements that make up the model.

Measurements and Objectives are defined (full movement allowed, no locked position, no interferences, no boundary and trajectory violations), then DOEs and optimisations are performed using Adams and Adams/Insight to define the best kinematic mechanism.

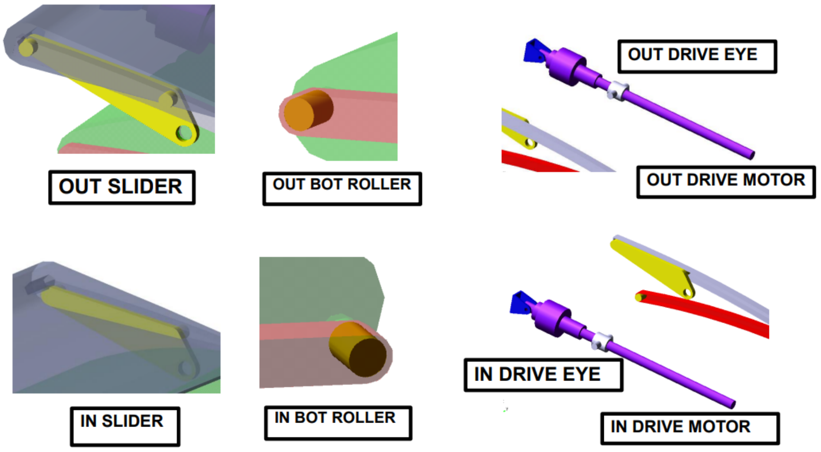


Figure 4.17: Flap model elements

With Adams, a kinematic analysis can subsequently be performed to verify functionality, perform preliminary actuator sizing, as well as clearances and failure analysis. However, the kinematic model has its limits: it does not incorporate flexible bodies and relies on ideal actuations (Motion). This leads to a dynamic analysis, which is based on a more realistic model, taking into account inertia, aerodynamic loads, vibrations, failures and disturbances. Hydraulic, electrical or other actuating systems can then be implemented. All this results in an increase in complexity, parameters and greater knowledge of physics (11).
Once the Adams multibody analysis has been completed, the actuation system design can be addressed. For this work, the actuation system model is executed with Elements.

To perform co-simulation between Adams and Elements, two alternatives are available :

1. Export a Controls Plant from Adams and import it into Elements as a building block for the actuation model.

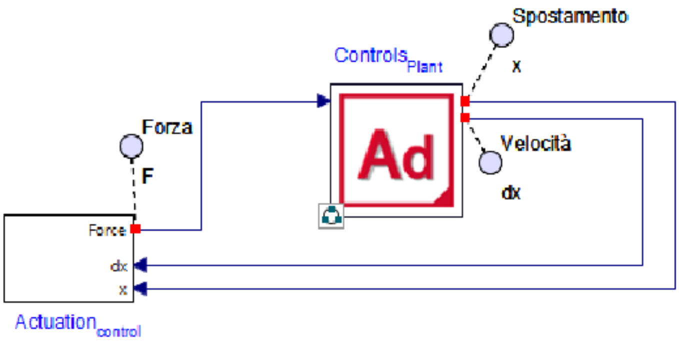
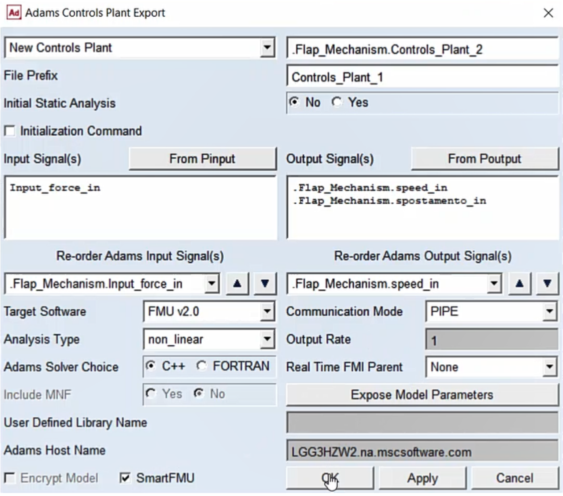


Figure 4.18: Elements flap model block diagram

The Adams block consists of a black box that receives as input the actuating force calculated by the control subsystem, and has as output the displacement x and velocity dx of the drive eye, which runs along the drive motor. Within this block the link to the multibody model is made to perform the co-simulation.



Adams Controls Plant Export

New Controls Plant | .Flap_Mechanism.Controls_Plant_2

File Prefix | Controls_Plant_1

Initial Static Analysis | ● No ○ Yes

□ Initialization Command

Input Signal(s) [From Pinput]

```
Input_force_in
```

Output Signal(s) [From Poutput]

```
.Flap_Mechanism.speed_in
.Flap_Mechanism.spostamento_in
```

Re-order Adams Input Signal(s)

.Flap_Mechanism.Input_force_in

Re-order Adams Output Signal(s)

.Flap_Mechanism.speed_in

Target Software: FMU v2.0

Analysis Type: non_linear

Adams Solver Choice: ● C++ ○ FORTRAN

Include MNF: ○ Yes ● No

Communication Mode: PIPE

Output Rate: 1

Real Time FMI Parent: None

User Defined Library Name

Adams Host Name: LGG3HZW2.na.mscsoftware.com

□ Encrypt Model ☑ SmartFMU

Figure 4.19: Adams Controls Plant Export Window

Investigating the '$Actuation_{control}$' subsystem, we see an electromechanical actuation system very similar to the model seen in the previous chapter (Figure 3.45).

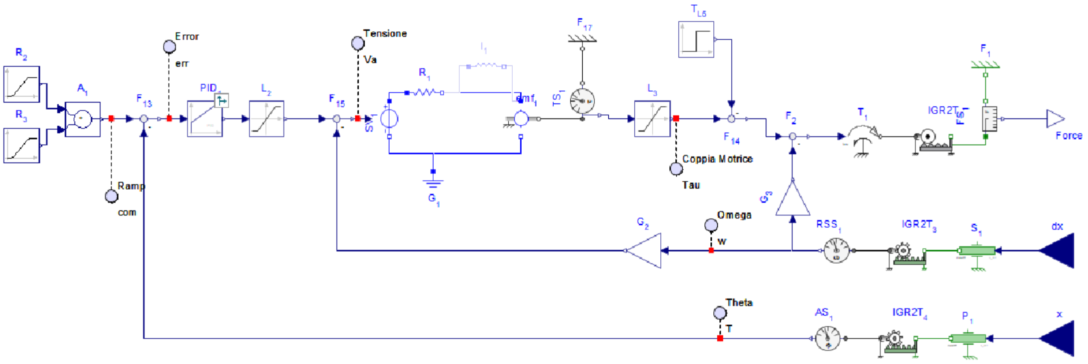
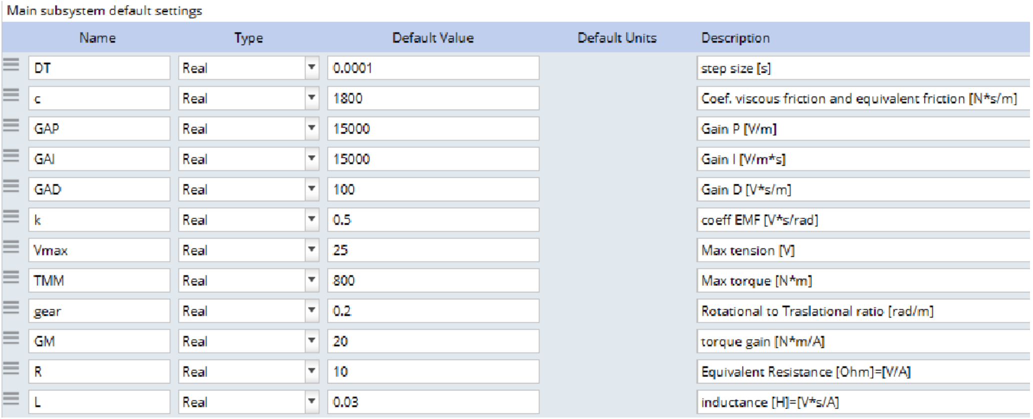


Figure 4.20: $Actuation_{control}$ subsystem block diagram

The main differences between stand-alone model and co-sim one are now outlined. Firstly, the command input, as in the former, three input types were tested, while in the latter, a single input type is used, which is the ramp input, defined in order to guarantee a deployment from a retracted position and a subsequent retraction from an extended position. Secondly, the actuation drive transmission. This is because in the stand-alone model, the dynamics are rotational (i.e., torque T_m, angular displacement θ, and angular velocity ω), whereas in the current model, we must convert the dynamics from rotational to translational, because the input and output parameters defined on the drive eye component of the multibody model refer to a displacement x, a velocity dx, and a force F. To do this, the 'Ideal Gear Rotational to Translational' $IGR2T$ components have been added to the block diagram, which allow the force to be converted to torque and the displacement and velocity measurements to be converted from translational to angular. The last difference between the models is the absence of the inertial component within the cosimulation control subsystem, because the actuator and flap inertial components are located within the multibody model. Once the actuator model has been built, the parameters of the various components in the Workspace can be defined.



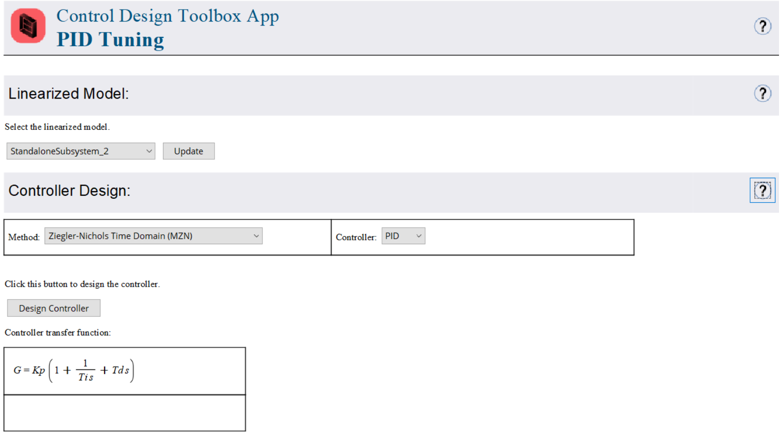
Main subsystem default settings

Name	Type	Default Value	Default Units	Description
DT	Real	0.0001		step size [s]
c	Real	1800		Coef. viscous friction and equivalent friction [N*s/m]
GAP	Real	15000		Gain P [V/m]
GAI	Real	15000		Gain I [V/m*s]
GAD	Real	100		Gain D [V*s/m]
k	Real	0.5		coeff EMF [V*s/rad]
Vmax	Real	25		Max tension [V]
TMM	Real	800		Max torque [N*m]
gear	Real	0.2		Rotational to Traslational ratio [rad/m]
GM	Real	20		torque gain [N*m/A]
R	Real	10		Equivalent Resistance [Ohm]=[V/A]
L	Real	0.03		inductance [H]=[V*s/A]

Table 4.1: $Actuation_{control}$ subsystem parameter set

After setting the ramp command so that the device deploys in $4.5s$, followed by $3.5s$ of equilibrium, and then a retraction of a further $4.5s$, followed again by $3.5s$ of equilibrium, the simulation parameters can be set: $16s$ duration with a fixed step Euler solver, 10000 plot points and $0.001s$ as communication interval between Elements and Adams.

In an ondesign configuration, without external loads T_L applied, different results can be obtained by changing the PID gain values.
Those shown in the Table 4.1 were obtained by means of the PID Tuning Control Design Toolbox App, an advanced feature integrated in Elements that allows optimised PID controller for the system, with Ziegler-Nichols design method.



Control Design Toolbox App

PID Tuning

Linearized Model:

Select the linearized model:

StandaloneSubsystem_2 Update

Controller Design:

Method: Ziegler-Nichols Time Domain (MZN)

Controller: PID

Click this button to design the controller.

Design Controller

Controller transfer function:

$G = Kp\left(1+\frac{1}{Ti\,s}+Td\,s\right)$

Figure 4.21: Elements PID Tuning Control Design Toolbox App

We also obtain, as a result, that for values above $GAP = 800000$ the system reaches a limit cycle and goes into instability.

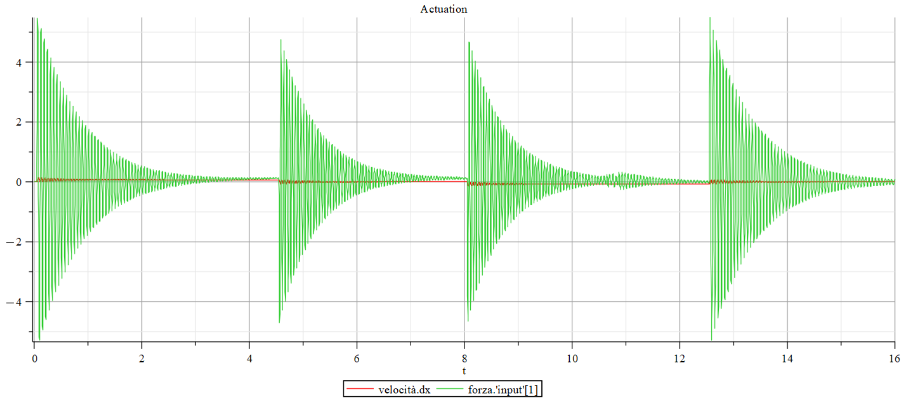


Figure 4.22: Flap actuation ondesign ($GAP = 800000$)

Below are the simulation results with $GAI = 0$ and $GAD = 0$.

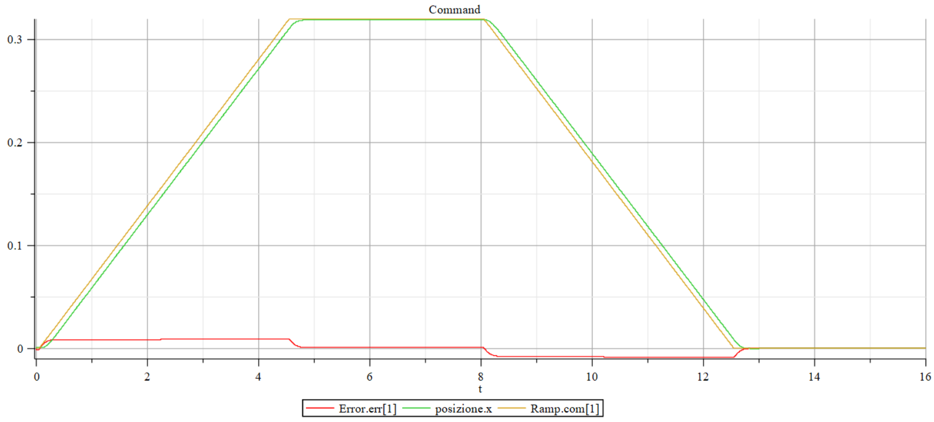


Figure 4.23: Flap actuation ondesign ($GAI = 0$, $GAD = 0$) - 1

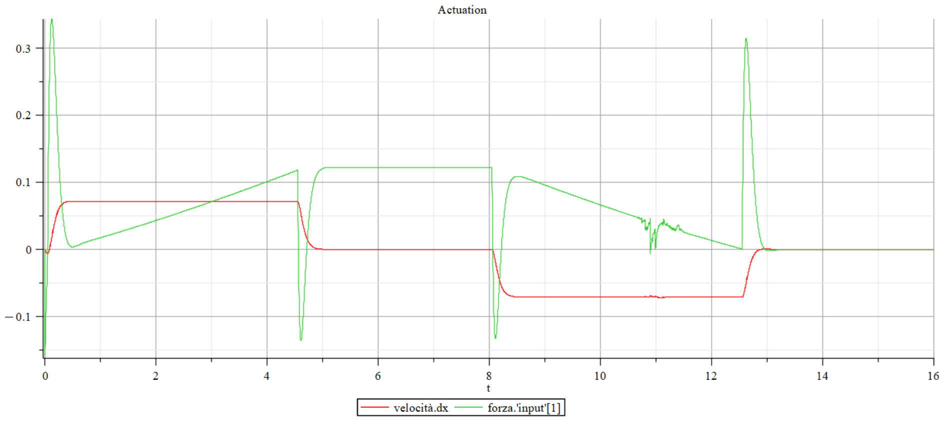


Figure 4.24: Flap actuation ondesign ($GAI = 0$, $GAD = 0$) - 2

As can clearly be seen, without an integral term, the actual flap position follows the input command without ever reaching convergence.
Instead, by resetting the original values of $GAI = 15000$ and $GAD = 100$, we obtain a position value that, over time, will tend to reduce the gap with the imposed command to zero.

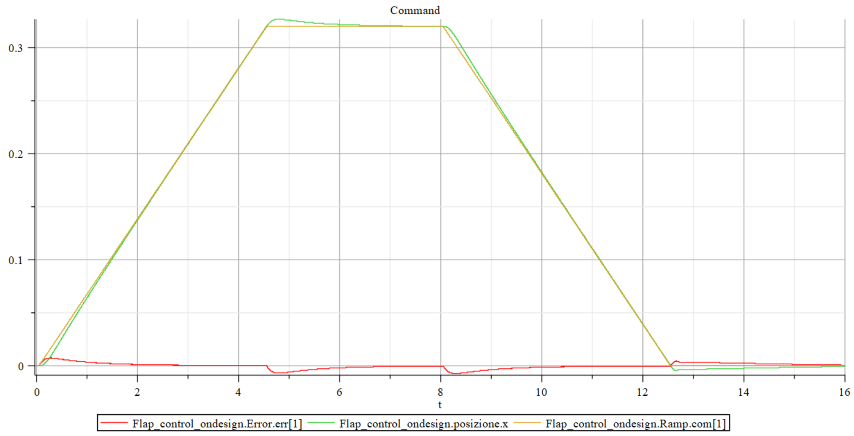


Figure 4.25: Flap actuation ondesign - 1

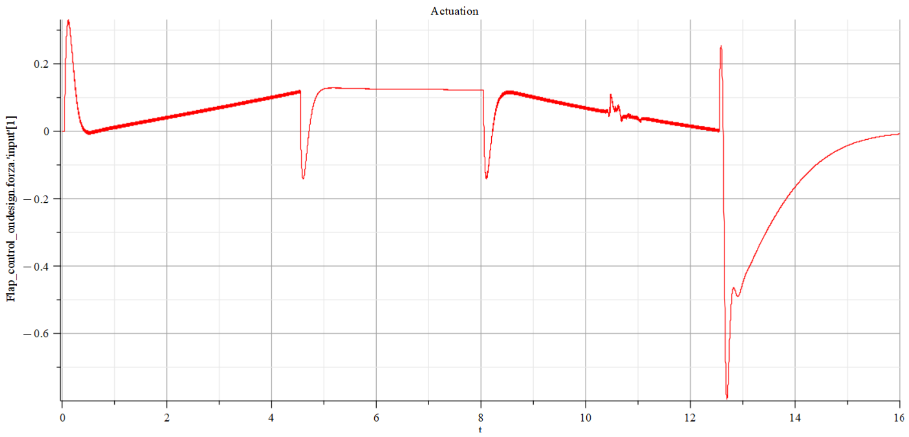


Figure 4.26: Flap actuation ondesign - 2

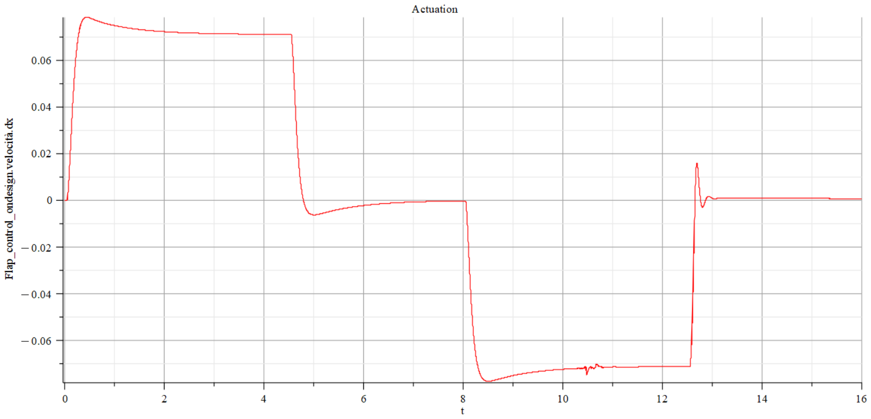


Figure 4.27: Flap actuation ondesign - 3

In Figure 4.25 we can see how, in the sections where the force acts to drive the flap to the commanded position ($t = 0.5 - 4.5, 8.5 - 12.5s$), the plot is segmented and, in $t = 10.5s$, there is even a slight local instability.
The first characteristic is due to a different number of plot points set within the Elements simulation (10000) and in the Adams simulation (5000). In fact, by simulating twice as many points as in multibody analysis, Elements certainly has greater precision in computing, but when communicating with the other solver, it does not receive an equal amount of data to return consistent results.

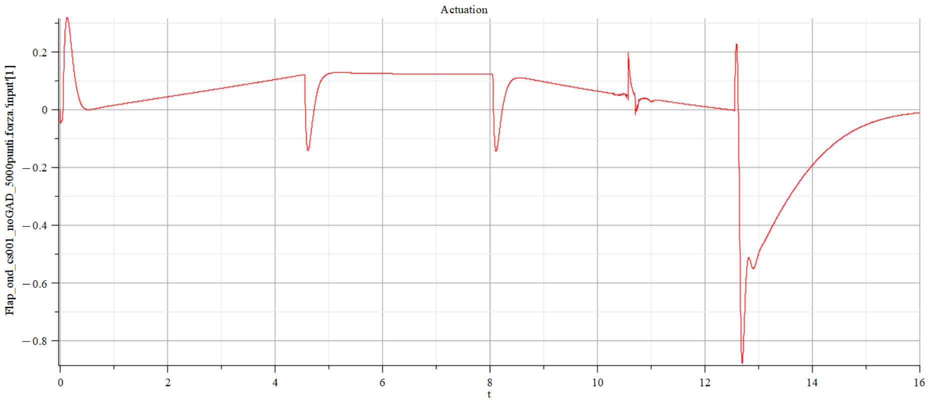


Figure 4.28: Flap actuation ondesign (5000 plot points)

Although the plot now has a more linear trend due to the correction made, the previously noted local instability persists. The cause of this instability is not to be attributed to the values assigned to the integral and derivative terms of the PID controller because it could already be observed in Figure 4.23. This would therefore be a local numerical instability related to the coupling of the equations solved by the two softwares during co-simulation, which, at that given point in the compilation, receives incongruent values between the two solvers.

Furthermore, we can see how the simulation would behave if we reduced the communication interval to $0.0001s$.

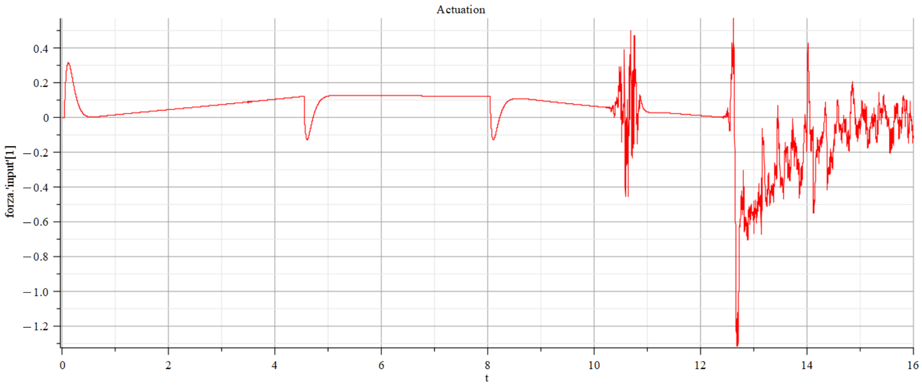


Figure 4.29: Flap actuation ondesign (communication interval = $0.0001s$)

As can clearly be seen, with a communication interval of a different magnitude from the simulation step size, in this case ten times smaller, the time required for calculation increases enormously, also leading to various instabilities in the solution.

The effects of the interaction between the actuation system and the flap structure are not modelled, but external loads can be added to the model to allow the actuation control response to be assessed in offdesign conditions. These aerodynamic loads are modelled as single contingency loads. In this work, four external loads of different intensities were introduced at specific times during the simulation. The loads are expressed as a percentage of the maximum torque that can be developed by the actuator (the stall torque):

(a) 40% of the maximum torque, applied in $t = 1.5s$.

(b) −30% of the maximum torque, applied in $t = 3s$.

(c) −30% of maximum torque, applied in $t = 9.5s$

(d) 40% of maximum torque, applied at $t = 11s$

The results are shown as follows, having set the same simulation parameters as the analysis previously executed in the absence of loads.

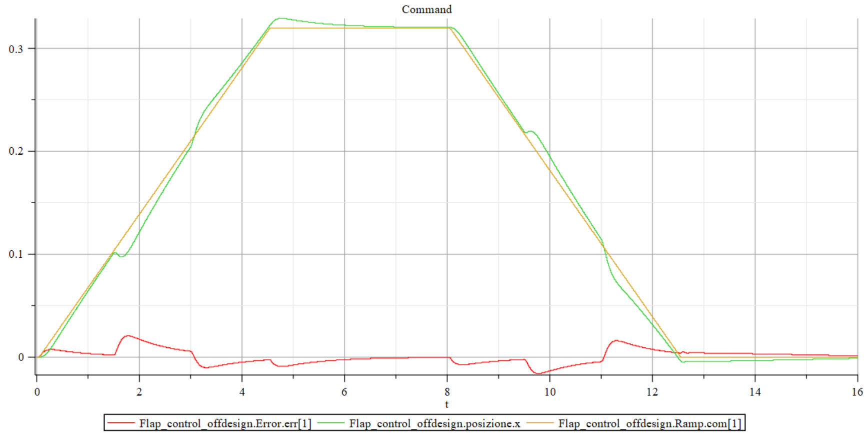


Figure 4.30: Flap actuation offdesign - 1

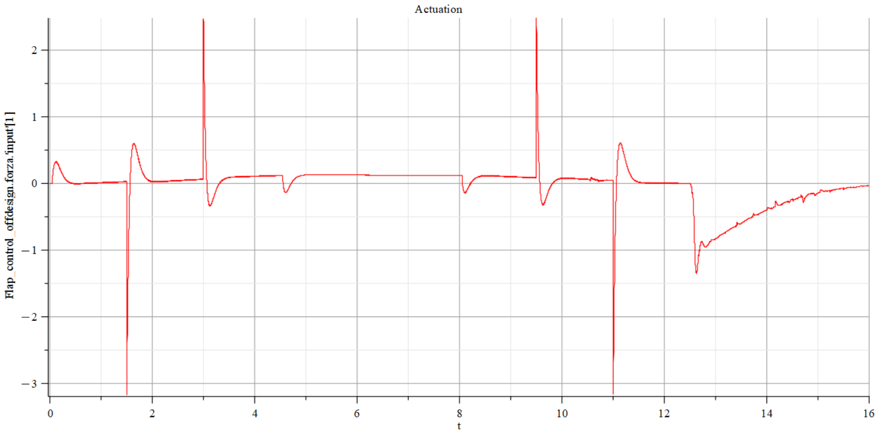


Figure 4.31: Flap actuation offdesign - 2

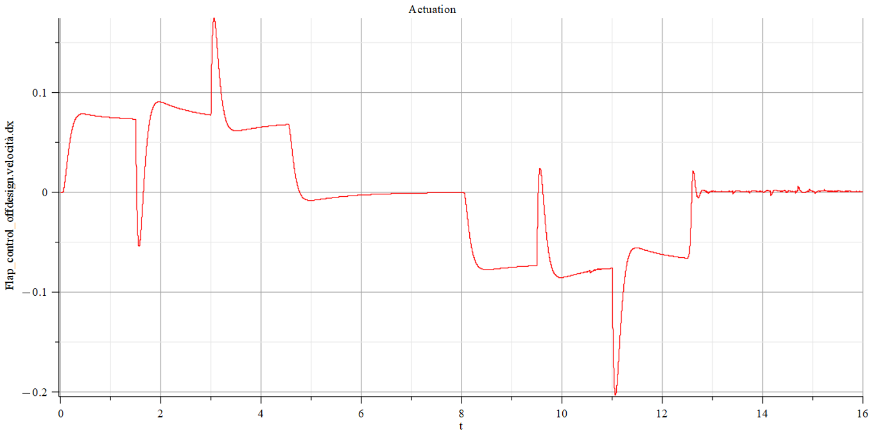


Figure 4.32: Flap actuation offdesign - 3

In this case we see peaks in the force diagram due to the loads applied at different instants of time. By zooming in on some of the details of the plot, we see the same phenomena as in the previous case without loads: in Figure 4.33, we see how the force during the ramp command is represented by a broken line, and in Figure 4.34, we see a fluctuation in the values during the flap stabilisation phase after retraction.

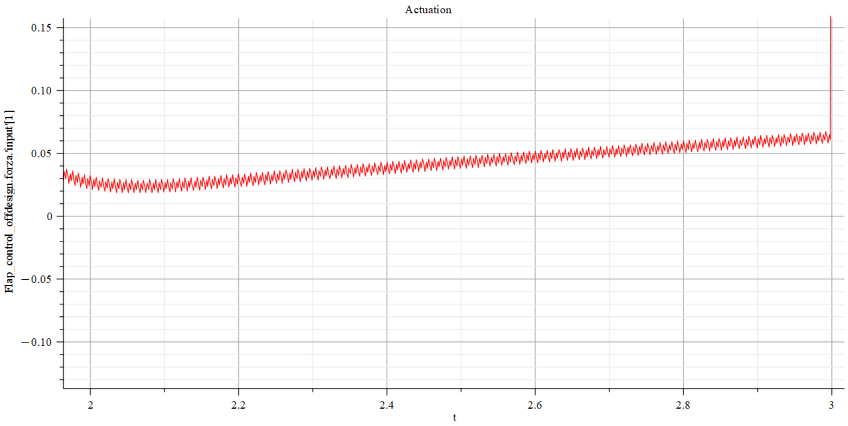


Figure 4.33: Flap actuation offdesign - 2
$t = 2 - 3s$ detail

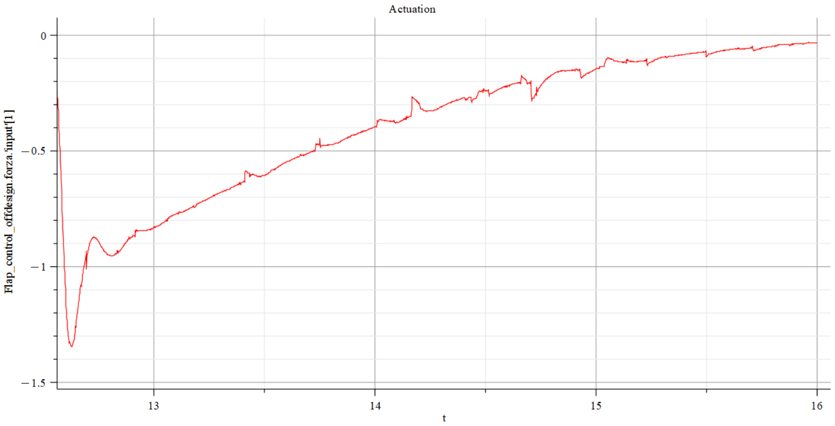


Figure 4.34: Flap actuation offdesign - 3
$t = 13 - 16s$ detail

In accordance with what has been shown before, the number of plot points of the simulation performed by Elements is reduced to match the number of plot points of the multibody analysis, in order to obtain a stable co-simulation. After having run a further analysis with this new simulation parameter, we can compare the two curves and see that the fluctuations and, of course, the broken lines have stabilised.

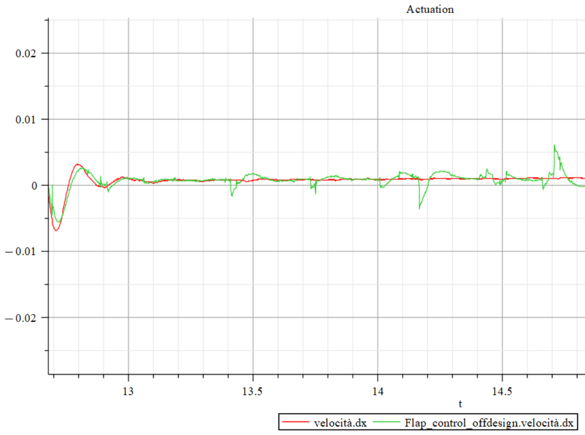


Figure 4.35: Flap actuation offdesign - 10000 vs 5000 plot points comparing

In conclusion, the different overshoots and undershoots of the commanded flap position can be evaluated, as can the ondesign and offdesign configurations with each other.

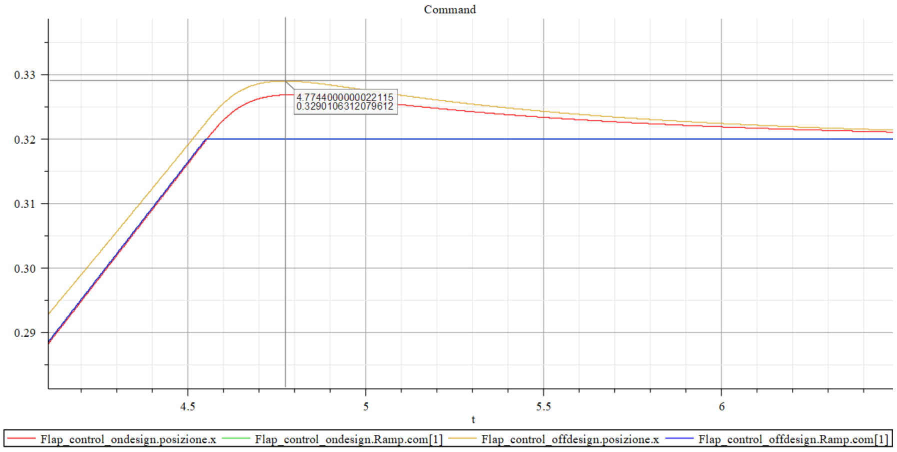


Figure 4.36: Flap actuation overshoot comparing - 1

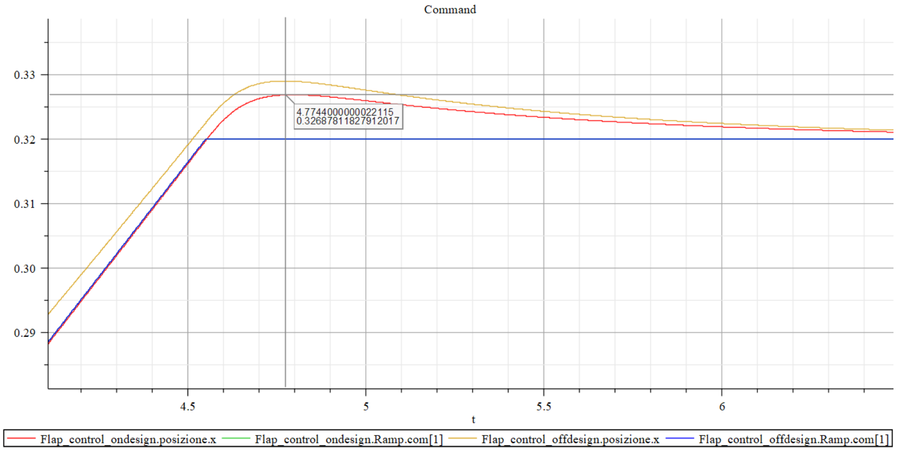


Figure 4.37: Flap actuation overshoot comparing - 2

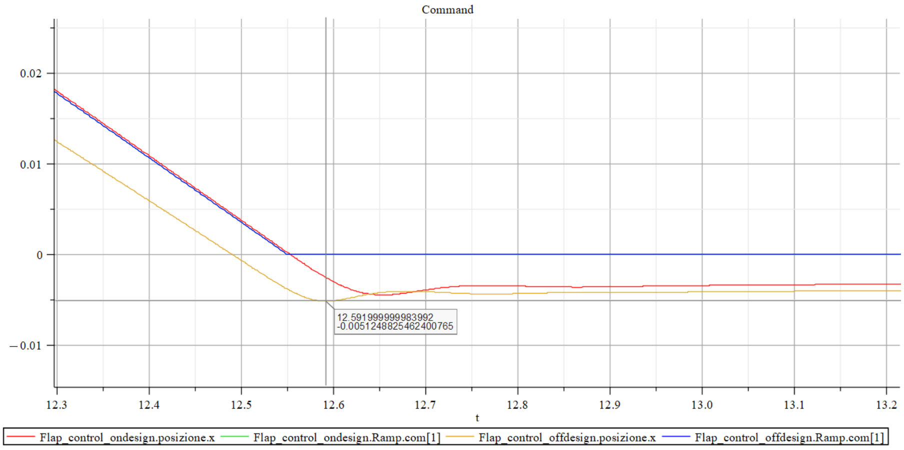


Figure 4.38: Flap actuation undershoot comparing - 1

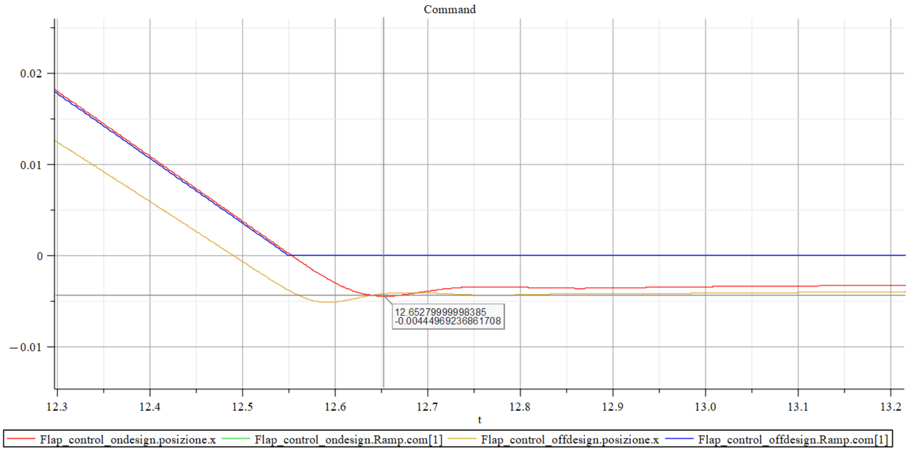


Figure 4.39: Flap actuation undershoot comparing - 2

Obviously, what is observed is that the offdesign configuration shows greater overshoots and smaller undershoots than the ondesign configuration. This is because, as loads are applied, the integral term of the PID controller has less time to settle the actual position relative to the commanded one.

2. Export the actuation subsystem from Elements and import it into Adams as a control system for the multibody model.

The multibody model of the flap, already seen in Figure 4.16, contains within it all the data relating to the mass, inertia, friction and other physical parameters of the various mechanical and structural components.
Through the integration in Adams of the control system developed in Elements, it is possible to obtain the following as co-simulation output:

(a) an animation of the physical-geometric model

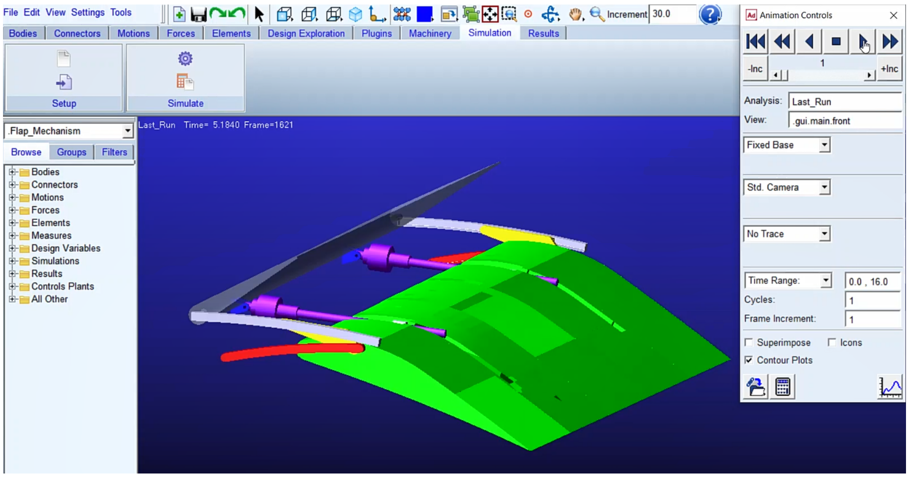
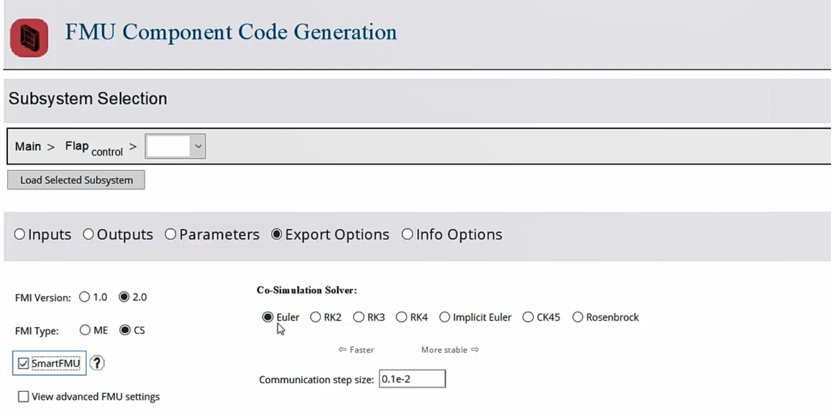


Figure 4.40: Flap actuation: Adams animation (frame 1621/5000)

(b) graphs of the same quantities displayed within Elements (displacement x, velocity dx and force F)

(c) the graphs of other quantities displayed in Adams thanks to its multibody analysis capability (displacements, velocities, accelerations, forces, torques, friction and so on)

To integrate the actuation control system within Adams, it is necessary to export a *.fmu* from the Elements model through the FMU Component Code Generation App, an Elements built-in tool.



FMU Component Code Generation

Subsystem Selection

Main > Flap control >

Load Selected Subsystem

○ Inputs ○ Outputs ○ Parameters ● Export Options ○ Info Options

FMI Version: ○ 1.0 ● 2.0

FMI Type: ○ ME ● CS

☑ SmartFMU ?

☐ View advanced FMU settings

Co-Simulation Solver:

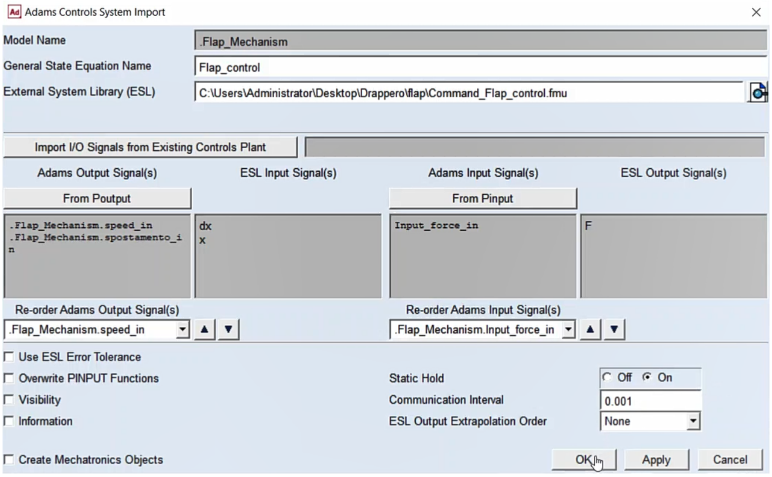
● Euler ○ RK2 ○ RK3 ○ RK4 ○ Implicit Euler ○ CK45 ○ Rosenbrock

⇦ Faster More stable ⇨

Communication step size: 0.1e-2

Figure 4.41: Flap actuation control system *.fmu* generation

Then, you can import the *.fmu* into Adams as an External System Library (ESL). Having done so, it is possible to run a co-simulation with Adams.


Adams Controls System Import

Model Name: .Flap_Mechanism

General State Equation Name: Flap_control

External System Library (ESL): C:\Users\Administrator\Desktop\Drappero\flap\Command_Flap_control.fmu

Adams Output Signal(s)	ESL Input Signal(s)	Adams Input Signal(s)	ESL Output Signal(s)
.Flap_Mechanism.speed_in .Flap_Mechanism.spostamento_in	dx x	Input_force_in	F

Re-order Adams Output Signal(s): .Flap_Mechanism.speed_in

Re-order Adams Input Signal(s): .Flap_Mechanism.Input_force_in

□ Use ESL Error Tolerance

□ Overwrite PINPUT Functions

□ Visibility

□ Information

□ Create Mechatronics Objects

Static Hold: ○ Off ● On

Communication Interval: 0.001

ESL Output Extrapolation Order: None

Figure 4.42: Flap actuation control system import

In this case, an equivalent model to the previously used electromechanical actuator was built, consisting only of the control logic (PID block, Limiter) and the input blocks for the external loads of the offdesign configuration.

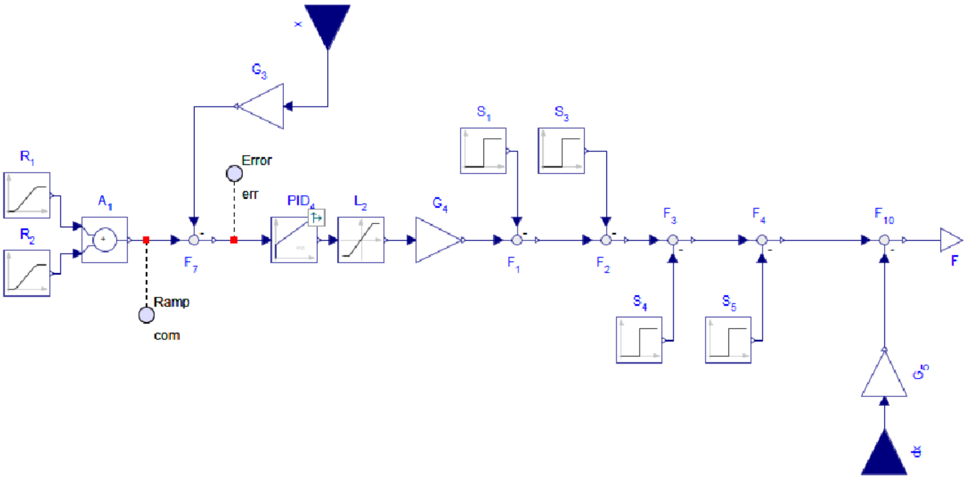


Figure 4.43: Imported actuation control subsystem

Going to analyse the quantity plots within the Adams post-processor, we will see how the trends of the curves are perfectly superimposable on those returned by the co-simulation launched in Elements.

For the ondesign configuration, with no external loads:

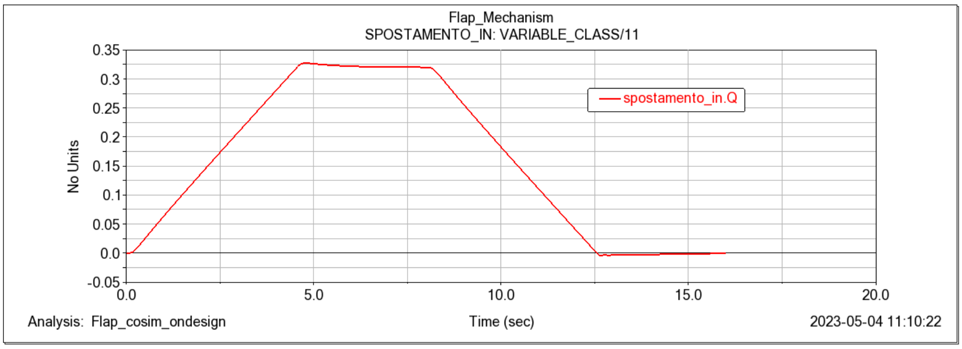


Figure 4.44: Adams post-processor: Flap actuation ondesign - 1

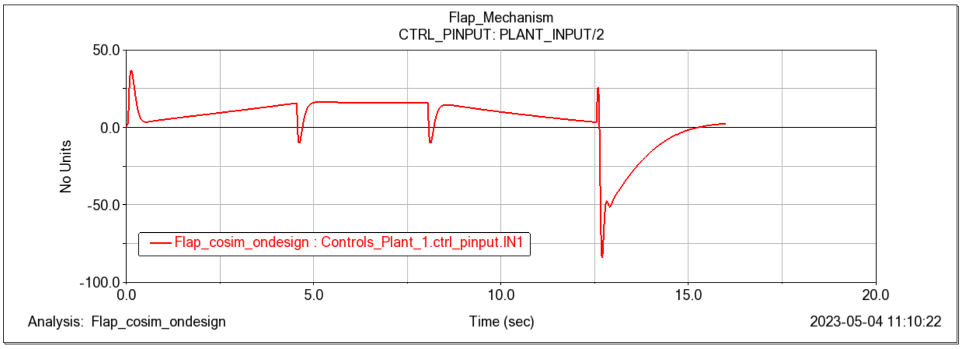


Figure 4.45: Adams post-processor: Flap actuation ondesign - 2

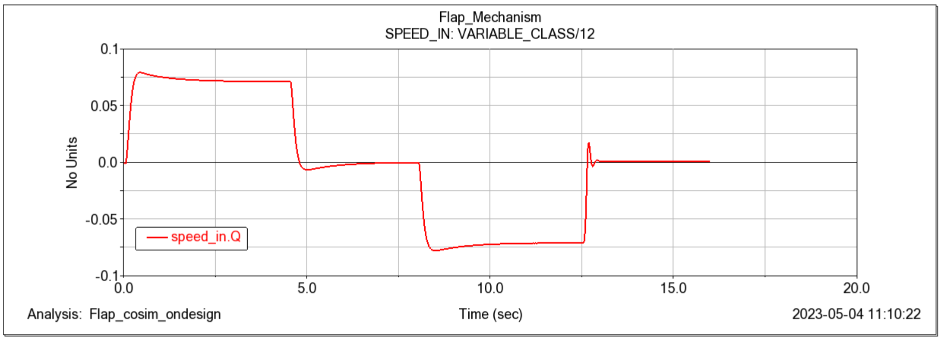


Figure 4.46: Adams post-processor: Flap actuation ondesign - 3

For the offdesign configuration, with external loads:

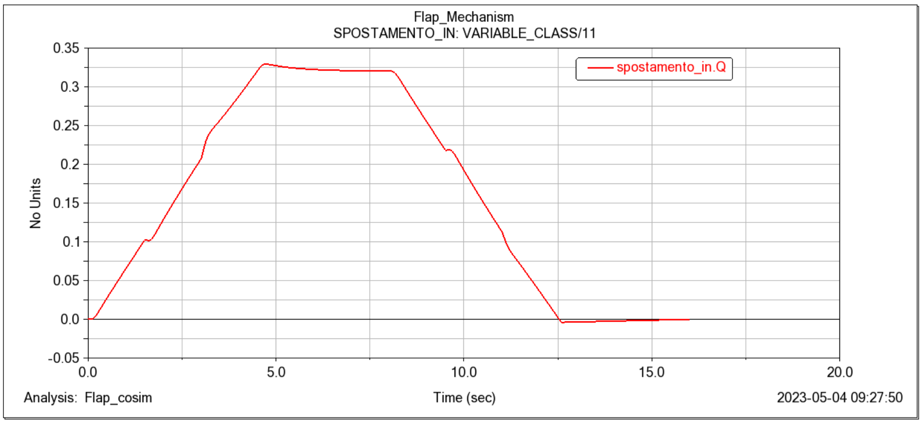


Figure 4.47: Adams post-processor: Flap actuation offdesign - 1

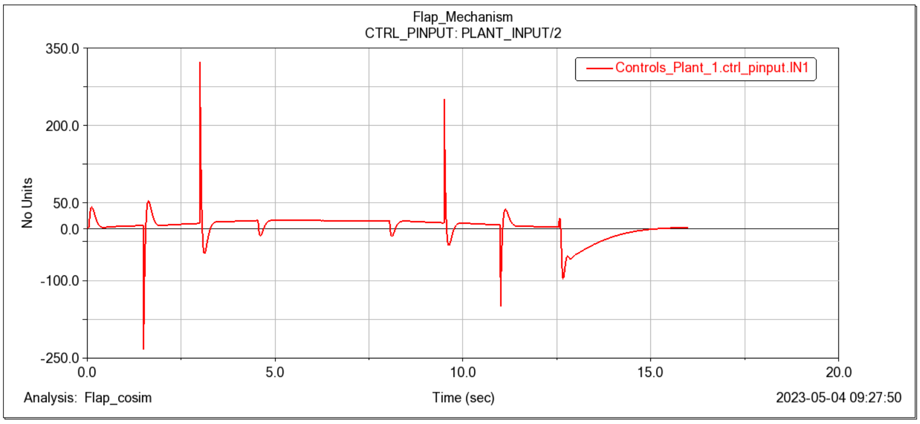


Figure 4.48: Adams post-processor: Flap actuation offdesign - 2

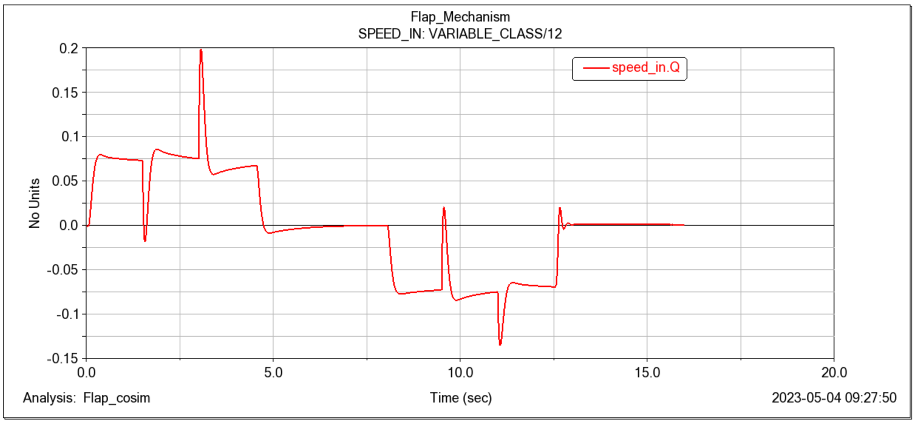


Figure 4.49: Adams post-processor: Flap actuation offdesign - 3

Chapter 5

Conclusions

From the results obtained, it is evident that both Elements and Simulink are valid for the correct implementation of dynamic systems models.
Both simulating environments have applicability in the aerospace systems field, but their suitability may vary depending on specific use cases and requirements. Simulink's extensive block libraries, MATLAB integration, and industry adoption make it well-suited for complex aerospace system simulations, especially when detailed control algorithm development is involved. On the other hand, Elements' open-source nature, support for the Modelica language, and multi-domain capabilities make it a compelling choice for modelling physical phenomena in aerospace systems, particularly for research, academic, or open innovation projects.
Ultimately, the choice between Simulink and Elements depends on factors such as desired features, performance requirements, integration needs, and the expertise available within the organisation or user community.

Obviously, one feature to be taken into account is the greater integrability between Elements and Adams (both Hexagon products) with regard to co-simulation. Indeed, we have seen how co-simulation can be helpful in the development process of a new product. Although the purpose of this work was only to test Elements, as a new multiphysics simulation environment on the scene, and to validate its co-simulation capabilities with Adams, it is worth remembering that the potential of Hexagon products is much broader than shown in the previous pages.
The aim of future work could be to investigate in depth the co-simulation and integration of different simulation tools in order to obtain more precise results.
An even more realistic flap model can be realised by implementing elasto-backlash and a friction model to the actuation system and then using Patran&Nastran, for structural analysis, and Cradle, for fluid dynamic analysis. By integrating a finite-element model into the flap structure, more detailed simulations can be obtained to assess how flexible bodies influence actuation mechanisms and to take fatigue analysis into account. By then extending the simulation with CFD anal-

ysis, the behaviour of the entire system under aerodynamic loads can be studied. As a result, this can have very useful impacts on the design, testing and qualification processes.

Currently, the design of test methods is carried out manually. Model-Based Test methods, on the other hand, might be designed using a full system model to automatically generate and set up test configurations and test cases. This would raise the level of automation and ensure that virtual tests were covered thoroughly. Virtual tests are an efficient technique to get system information during the early stages of design. The virtual process's end-to-end capabilities enable rapid assessment of design changes and test variants, and results can be tracked back to the underlying system requirements. All of this can be accomplished without a great deal of manual, susceptible-to-mistakes activity, thanks to the automation of model integration stages. In terms of modelling and simulation, the process chain can cover virtual test configurations with models in its current implementation, but these configurations can be further developed in the future to enable multi-disciplinary co-simulation with a multi-body representation of flap mechanisms, a flap structural model, and distributed aerodynamic loads. Through virtual testing, a deeper understanding of the system's design can be acquired. Transient loads under failure conditions, in particular, can be examined at the interface between the flap structure and mechanisms, allowing this data to be fed back into further design phases of the testing process. The process implementation is designed to be easily scalable, allowing the number of virtual test methods to be raised in response to the needs of use cases.

Whenever using these methods to support product qualification, the methodologies' and simulation models' reliability must be verified. To estimate and minimise the uncertainties in the parameters and models, intensive work is required. To fully realise the promise of virtual testing, validation procedures and hybrid testing approaches must be researched.

Bibliography

[1] **Paolo Maggiore and Matteo D.L. Dalla Vedova,** ***Lectures notes from 'Modellazione, simulazione e sperimentazione dei sistemi aerospaziali' course***, Politecnico di Torino, 2022

[2] **Stefano Re,** ***Development and comparison of prognostic methodologies applied to Electromechanical Servosystems (EMA) for aerospace purposes***, M.D. Thesis, A.Y. 2018

[3] **Francesco Nudo,** ***Numerical modeling of servomechanisms: comparison between different development environments***, M.D. Thesis, A.Y. 2020

[4] **Eric Olson,** ***Aircraft actuation technologies: how do electrohydraulic, electrohydrostatic and electromechanical actuators work?***, Globalspec, 2019

[5] **Guan Qiao, Geng Liu, Zhenghong Shi, Yawen Wang, Shangjun Ma and Teik C Lim,** ***A review of electromechanical actuators for More/All Electric aircraft systems***, Journal of Mechanical Engineering Science, 2017

[6] **Federal Aviation Administration,** ***Pilot's Handbook of Aeronautical Knowledge***, FAA-H-8083-25B, 2016

[7] **INCOSE,** ***Systems Engineering Definition***, International Council on Systems Engineering
`https://www.incose.org/about-systems-engineering/system-and-se-definition/systems-engineering-definition`

[8] **Shevchenko, N.,** ***An Introduction to Model-Based Systems Engineering (MBSE)***, Carnegie Mellon University, Software Engineering Institute's Insights (blog), 2020
`https://insights.sei.cmu.edu/blog/introduction/model-based-systems-engineering-mbse/`

[9] **Recksiek, M.,** ***Advanced high lift system architecture with distributed electrical flap actuation***, Proceedings of the 2nd International Workshop on Aircraft System Technologies, 2009

[10] **Hollmann, R. W., Schäfer, A., Bertram, O., & Rädel, M.,** ***Virtual testing of multifunctional moveable actuation systems***, CEAS Aeronautical Journal, 13(4), 2022

[11] **Daniele Catelani,** ***Workshop 'High-Lift Device, Multibody, FEM, CFD, Controls'***, MSC Software, 2021

[12] **Hexagon AB,** ***Product groups by the Manufacturing Intelligence division, Computer Aided Engineering software (CAE software)***
`https://hexagon.com/products/elements`
`https://hexagon.com/products/adams`
`https://hexagon.com/products/msc-cosim`

[13] **Hexagon AB,** ***Elements and Adams Product Documentations***, 2022