Primož Kovačič, BSc

Active Control Loading System for a Research Flight Simulator

MASTER'S THESIS

for the degree of

Master of Science

Master's programme Mechanical Engineering

submitted at

Graz University of Technology

Supervisor

Ao.Univ.-Prof. Dipl.-Ing. Dr.techn. Reinhard Braunstingl

Institute of Mechanics

Faculty of Mechanical Engineering and Economic Sciences

Graz, February 2018

Primož Kovačič, BSc

Active Control Loading System for a Research Flight Simulator

MASTERARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

Masterstudium Maschinenbau

eingereicht an der

Technischen Universität Graz

Betreuer

Ao.Univ.-Prof. Dipl.-Ing. Dr.techn. Reinhard Braunstingl

Institut für Mechanik

Fakultät für Maschinenbau und Wirtschaftswissenschaften

Graz, Februar 2018

Statuatory declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been either quoted literally or by content from the used sources. The text document uploaded to TUGRAZonline is identical to the present master's thesis.

Graz, ____________________ ____________________

(date) (signature)

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Das in TUGRAZonline hochgeladene Textdokument ist mit der vorliegenden Masterarbeit identisch.

Graz, am ____________________ ____________________

(Datum) (Unterschrift)

Abstract

In a flight simulator, the control loading system (CLS) provides forces on the control column, equivalent to those that would occur in a real aircraft and thus simulates the necessary haptic feedback for the pilot. Our goal is to develop and assemble an active CLS for the research flight simulator at TU Graz in order to improve the current simulation fidelity. Beforehand, the solution concepts should be implemented and evaluated on a small-scale test bench, which allows for rapid research and development. The presented master's thesis begins with a comprehensive overview of state of the art, followed by the development and commissioning of the CLS test bench. After the mechanical construction of the test bench was completed, an admittance type CLS controller was implemented on the selected B&R Automation control hardware including a Simulink model of the flight control system. The CLS performed excellently at recreating the dynamics of the flight controls model and compensated for the mechanism's friction very well. Some small, but noticeable imperfections due to the used gearbox remained in the simulation; therefore guidelines for future improvements are suggested in the discussion.

Kurzfassung

In einem Flugsimulator versorgt das Kraftrückführungssystem (Englisch: Control Loading System - CLS) den Steuerhebel mit Kräften, äquivalent zu denen die in einem realen Flugzeug auftreten würden und dadurch wird das nötige haptische Feedback für den Piloten simuliert. Unser Ziel ist die Entwicklung eines aktiven CLS für den Forschungsflugsimulator an der TU Graz, um die aktuelle Simulationstreue zu verbessern. Zuerst sollen die Lösungskonzepte auf einen kleineren Laborprüfstand implementiert und evaluiert werden, wodurch ein schneller Entwicklungsprozess unterstützt wird. Die vorliegende Masterarbeit beginnt mit einem umfassenden Überblick über den Stand der Technik, gefolgt von der Entwicklung und Kommissionierung des CLS Prüfstandes. Nachdem die mechanische Konstruktion des Prüfstandes abgeschlossen war, wurde ein Admittanz-Typ CLS Regler auf der ausgewählten B&R Automation Hardware implementiert, inklusive einem Simulink Model des Flugsteuerungssystems. Der entwickelte CLS konnte die Dynamik des Flugsteuerungsmodels ausgezeichnet nachbilden und auch sehr gut die vorhandene Reibung des Mechanismus kompensieren. Gewisse kleine, aber merkbare durch das verwendete Getriebe entstandene Störungen sind in der Simulation noch verblieben, deswegen werden Möglichkeiten für weitere Ausbesserungen in der abschließenden Diskussion vorgeschlagen.

Acknowledgements

First and foremost, I would like to thank my thesis supervisor Prof. Dr.techn. Reinhard Braunstingl. During my years at the university, I had the chance to be part of numerous fascinating projects related to aerospace engineering, and I am very grateful to him for every given opportunity, advice and his mentorship. The beginning of the thesis project contained many unknowns, and I really appreciate his trust and belief in the success of the project. I feel honoured to have been able to make a small contribution to the research flight simulator and have nothing but the utmost respect for his work as a researcher and educator.

The company B&R Automation very graciously supported our endeavours by providing us with discounted control hardware and offering a lot of helpful training and support during the commissioning phase. A special thank you goes to Mr Thomas Dicker and the rest of the B&R Education Network team for their kindness and motivation to educate the engineers of tomorrow. The company's willingness to support student projects was a vital contribution to get our project done right.

I would also like to thank Ing. Helmut Siegl from the institute for providing me with very useful practical advice in the lab and workshop. Many hardware and electronic components I was working with on the test bench were new to me, and his generously shared knowledge often helped me to clarify any doubts.

Last, but certainly not least, whom I cannot thank enough is my family: especially my loving parents, sister, grandparents and brother-in-law. They are and forever will be the most important people in my life that I look up to and the ones I know I can always count on. Without their unconditional love and support, I would not be who I am today.

Glossary

List of Abbreviations

AC Alternating current

ADC Analog Digital Converter

AFU Artificial Feel Unit

AS Automation Studio

BITD Basic Instrument Training Device

CG Centre of Gravity

CLS Control Loading System

CP Centre of Pressure

CPU Central Processing Unit

CS-FSTD(A) Certification Specifications for Aeroplane FSTDs

CS-FSTD(H) Certification Specifications for Helicopter FSTDs

DC Direct Current

DoF Degrees of Freedom

EASA European Aviation Safety Agency

FBD Function Block Diagram

FBW Fly-by-wire

FCS Flight Control System

FFS Full Flight Simulator

FNPT Flight Navigation Procedures Trainer

FS Flight Simulation

FSTD Flight Simulation Training Device

FTD Flight Training Device

HMI Human Machine Interface

IGBT Insulated Gate Bipolar Transistor

I/O Input/Output

PC Personal Computer

PCU Power Control Unit

PLC Programmable Logic Controller

PWM Pulse-Width Modulation

STO Safe Torque Off

ZFTT Zero Flight-Time Training

Contents

Chapter 1

Introduction

Flight simulation is one of the key training and engineering tools in the modern aerospace industry, because a virtual cockpit offers convenience, flexibility and cost advantages that simply cannot be matched by a real aircraft. The development of such simulation devices constantly strives to improve their fidelity to offer the highest degree of realism possible. The **Flight Simulation Research Platform at TU Graz** operates two fix-based flight simulators (wide-body and light aircraft), which are used in the research and development of cockpit environments as well as for teaching. A vital feature of the integrated systems and mechanical components is their high flexibility and the ability to accommodate additional equipment or features when conducting research rapidly.

One of the major engineering challenges in a flight simulator is an accurate replication of the control forces that the pilot experiences at the control column, wheel or pedals. This is the task of a **control loading system** (CLS), which can provide the necessary forces in various ways. Currently, the wide-body aircraft simulator at TU Graz is equipped with a passive CLS, which utilises mechanical spring and damper elements to approximate the control forces during flight. However, the achievable fidelity of a **passive** CLS is very limited, because its constant dynamic parameters cannot accurately represent the continuous changes of the aircraft's handling dynamics during different phases of flight. With the goal to continue improving the fidelity of the research flight simulator, an upgrade to an **active** CLS is currently in progress, which will incorporate digitally controlled electric actuators as flight control loaders. This will allow us to vary the control force parameters dynamically and also simulate many additional features (e.g. trimming, autopilot actuation, malfunctions etc.).

Although there are solutions available on the market, we opted to develop our own control loading system, because it better conforms with the mission of our research facility. The majority of the flight simulator's hardware and software results from years of development and improvement by the staff and students at the institute. This means that all components are open and available for modifications, which better facilitates the flexibility during research and provides a complete insight into the flight simulation working principles for teaching purposes. It would be preferred that the same approach is used for the implementation of the active CLS so that we would retain complete access and modification ability for the entire system.

The development of a CLS is a great challenge, because the human tactile sensory system is extremely sensitive and is able to detect tiny imperfections in the simulated force and motion, which almost never pose a problem for other typical motion control applications. The first prototype of the CLS (installed on the rudder pedals) presented us with some problems due to the mechanism's friction, which suggested that we need to perform more research for the future and more fidelity critical flight control channels (elevator, aileron). Therefore, we require a **experimental test bench platform**, where we could rapidly perform hands-on experiments and evaluate different mechanical components, control system architectures and flight controls simulation models in the attempt to optimise the CLS performance. Therewith, the **objectives** of my master's thesis are defined:

- Study and provide a comprehensive overview of state of the art.
- Mechanical design and setup of the CLS test bench.
- Choose and acquire appropriate control hardware that is future-proof.
- Commissioning of the CLS test bench.
- Develop and experiment with admittance type CLS control strategies.
- Develop a generic FCS model and implement it on the test bench.
- Identify the parameters that affect the CLS performance.
- Evaluate the strengths and weaknesses of the current configuration.
- Suggest guidelines and topics for further improvements and research.

In chapter 2, the thesis provides an overview of the fundamental topics and terminology, followed by chapter 3, where the process of the solution development is described. In chapter 4, the results are presented and discussed with the aid of experimental data and finally, chapter 5 concludes the project and offers a starting point for further development.

Chapter 2

State of the Art

The study and development of a flight simulator's control loading system combines multiple interdisciplinary subjects of *mechanical*, *aerospace*, *mechatronics* and *control systems engineering* as well as an understanding of the *human factors* at play. The following sections are a systematic overview of relevant topics with the goals to build a foundation knowledge, establish common terminology and provide insight into **state of the art**.

2.1 Flight Simulation

Although the beginnings of Flight Simulation Training Devices (FSTD) can be traced as far back as the late 1920s and their pioneer Edward Link, the true recognition and uprise of this technology were enabled through the enormous advances in digital computing during the Microelectronics revolution in the 1980s. [2, ch.1] Today, modern flight simulators represent indispensable tools for **flight training** as well as **aircraft engineering** therefore it is hard to imagine the current aerospace industry without them.

2.1.1 Flight simulation training devices

A significant amount of pilot training (civil and military) is performed in simulators because they offer many advantages in terms of:

- **safety**
  High-risk manoeuvres can be trained in the safe comfort of a training centre. [2, p.9]
- **cost**
  While the purchase and operational costs of a state of the art flight simulator might not be cheap, they are still much lower compared to operating an airborne aircraft.
- **training effect**
  - the flight instructor has full control over flight conditions
  - specific situations (e.q. equipment malfunctions) can be simulated on demand
  - manoeuvres (e.g. landing) can be repeated instantaneously

The main concerns for flight simulators used in flight training are:

- their **fidelity** (i.e. ability to accurately represent reality) and
- the thus achievable **training transfer** (i.e. 'effectiveness of training in a simulator' [2, p.11]; 'transferability of learned skills')

as different stages of training require specific degrees of realism. In order for the flight training time acquired in a simulator to be acknowledged by the authorities, strict and comprehensive device specifications have to be met, validated and certified.

EASA classifies FSTDs in the following categories [4][1]:

- **BITD** (Basic Instrument Training Device)
- **FNPT** (Flight Navigation Procedures Trainer) - *Levels I, II, MCC*
- **FTD** (Flight Training Device) - *Levels 1, 2*
- **FFS** (Full Flight Simulator) - *Levels A, B, C, D*

A **Level D FFS** provides the highest simulation realism available by featuring only the most advanced technologies such as visual system with collimated displays, 6 DoF motion platform, high dynamics control loading system etc. The achieved simulation fidelity is so excellent, that in certain circumstances aircraft type rating courses can be completed based on simulator flight experience alone, also known as "Zero Flight-Time Training" (ZFTT).

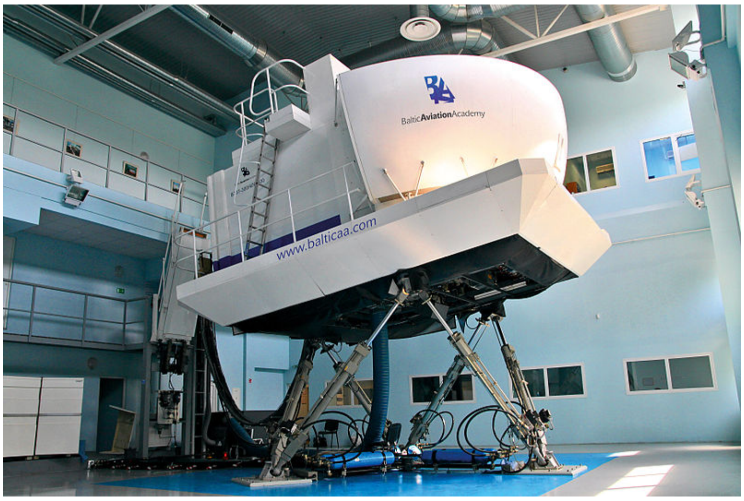


Figure 2.1: A modern FFS
[photo courtesy of Baltic Aviation Academy]

[1] CS-FSTD(A) regulates aeroplane and CS-FSTD(H) regulates helicopter simulators. The helicopter categories very minorly differ from the here listed aeroplane categories.

2.1.2 Flight simulation in engineering

Another important area of use for flight simulation is the **research and development** sector of the aerospace industry.

Scientific research of cockpit environments is usually conducted in a flight simulator because it provides complete control and repeatability of simulated situations.

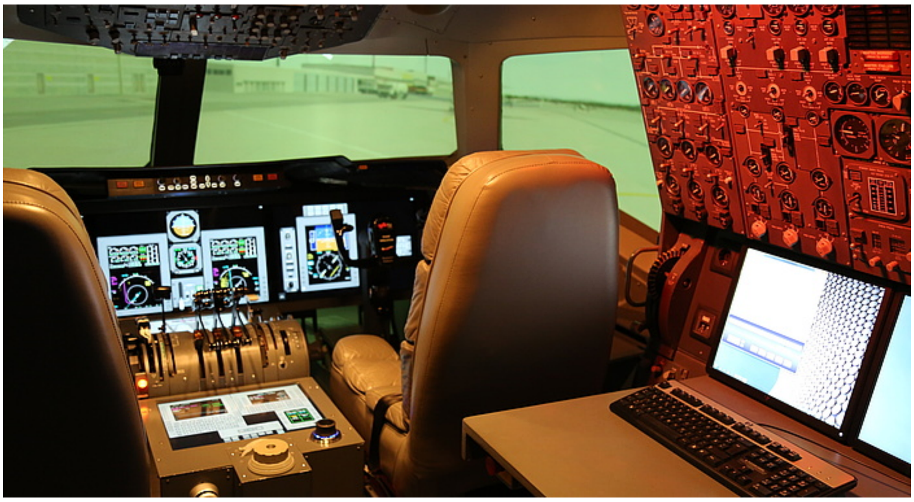


Figure 2.2: Research flight simulator of TU Graz
(fixed base, widebody aircraft) © TU Graz

The **engineering process** in general, is more and more inclined towards parallel development of systems and components with rapid prototyping and testing in the early phases of a project. Flight simulators are an excellent "playground" for aerospace engineers to explore and experiment with new ideas, which can be rapidly implemented and thoroughly evaluated in a time and cost-effective manner. During the extensive aircraft systems validation and trials, experiments can quickly be modified or repeated to capture the desired data efficiently. [2, p.13] Furthermore, the detected faults and even pilot's feedback can be taken into consideration already in the early stages of development when several design iterations can be improved upon in a virtual environment before a physical product even comes into existence.

These types of simulators are not legally required to undergo the certification process of a FSTD, and usually, a certain degree of realism is sacrificed in favour of high flexibility. Often, a motion cueing platform is omitted to retain a simple mechanical structure that is reconfigurable, modular and easily expandable to accommodate new technologies effortlessly.

2.1.3 The architecture of a flight simulator

Modern flight simulators usually posses a modular organization and consist of several task specific **subsystems** that are often distributed among multiple computers, which are connected in a local network. A generalized representation of a flight simulator's architecture is shown in Figure 2.3 [2, p.17]:

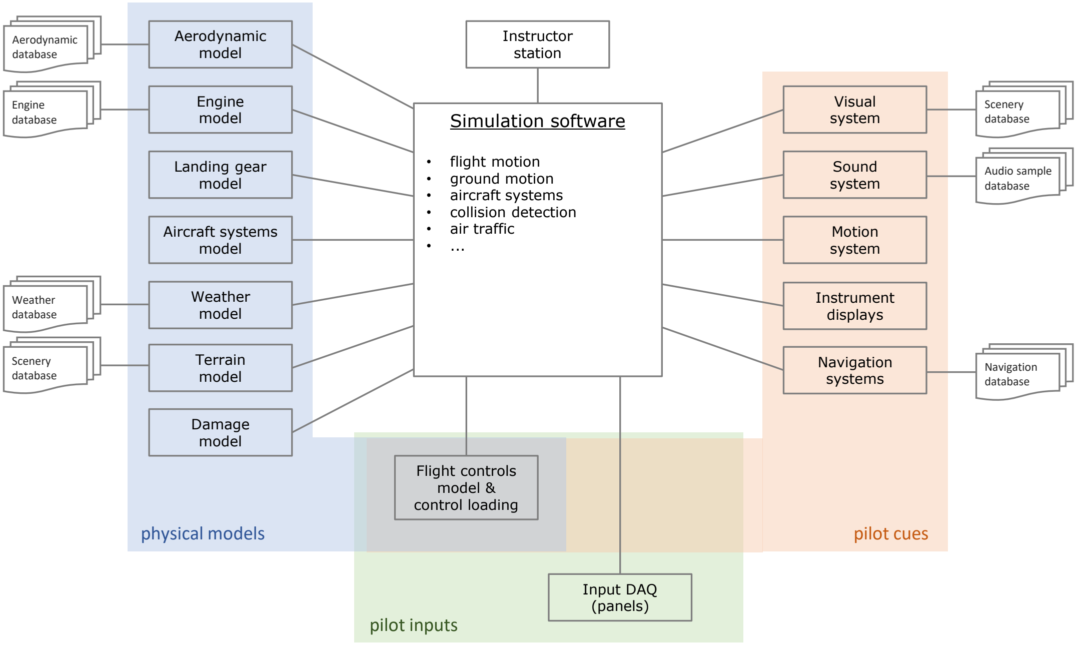


Figure 2.3: The architecture of a flight simulator

At the heart of every flight simulator lies the **simulation software**, which contains *numerical solvers* of differential equations, governed by various laws of physics and derived from the implemented **physical models**. Based on the acquired **pilot's input** data, the solvers have to provide a realistic system response solution in *real-time*. This data is then used to generate appropriate **pilot cues** (i.e. stimulate pilot's senses with visual, sound, motion and haptic cues). [2, p.27]
By now it should be evident, that a flight simulator is essentially a **virtual reality environment** based around the aircraft's Human Machine Interface (HMI) - the cockpit.

Notice, that the **control loading system** (CLS) is somewhat unique in its multifunctionality, since it not only acts as an interface for pilot's flight control inputs but also simultaneously provides a haptic cue over the same interface.

2.2 Flight Control Systems

As already mentioned, a flight simulator's CLS requires a *flight controls model*, thus naturally it is necessary to study the working principles of **real aircraft's flight controls**. (In this thesis, we focus our attention to *fixed-wing aircrafts* only.)

During the flight, the pilot manoeuvres the aircraft by varying the displacement of various **control surfaces**, which consequently deflect the passing slipstream and result in aerodynamic forces capable of changing the **aircraft's attitude**.

According to their function, we categorize **flight controls** into:

- **primary flight controls**
  $\hookrightarrow$ elevator, ailerons, rudder [2] $\Longrightarrow$ pitch, roll, yaw
- **secondary flight controls**
  $\hookrightarrow$ trim tabs, high-lift control (flaps, slats), spoilers, speed brakes... [11]

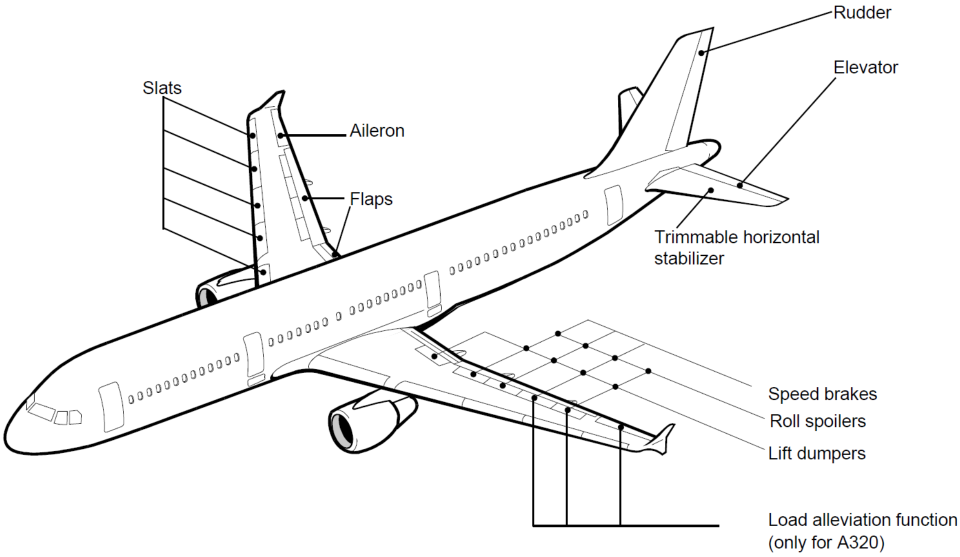

Rudder
Elevator
Slats
Aileron
Flaps
Trimmable horizontal stabilizer
Speed brakes
Roll spoilers
Lift dumpers
Load alleviation function (only for A320)

Figure 2.4: Flight control surfaces of an Airbus A320
[Airbus A319/A320/A321 Flight deck and systems briefing for pilots, 1998, p.5.4]

For the sake of completeness, the pilot also manoeuvres the aircraft with control over:

- **engine control systems** and
  $\hookrightarrow$ throttle/thrust, reverse thrust, auto-thrust system...
- **landing gear systems**
  $\hookrightarrow$ extend/retract landing gear, wheel brakes, nose-wheel steering... [11]

these systems, however, are based upon fundamentally different working principles.

[2] military fighter jets often feature (primary) flight control surfaces arranged in distinctly different and naturally unstable configurations to achieve exceptional agility (e.g. canard surfaces/foreplanes)

In contrast to the rest mainly *'discrete-event driven'* controls, the pilot provides precise correction inputs to the **primary flight controls** *continuously* throughout manual flight. An accurate representation of primary flight controls in a flight simulator is therefore especially crucial and heavily influences the achievable training transfer, which is why they shall be the primary focus in our CLS development.

The internal structure and functionality of a Flight Control System (FCS) vary according to the aircraft type and size, with many possible variations in the design details. Through history, as the performance demands grew, three basic principles of coupling control surfaces to the pilot's control column evolved.

2.2.1 Mechanical flight control system

Smaller sized aircrafts usually utilise a purely ***mechanical flight control system***, where the control surfaces are directly linked to the pilot's control column through a series of **mechanical linkages**, that consist of *push-pull control rods* and/or *cable and pulley systems* [11, p.9]. This divides the FCS into three distinct sections, often referred to as:

- **FWD** (components in the 'forward' section of the aircraft - including control column)
- **Linkage** (cable or control rod system)
- **AFT** (components in the aft section of the aircraft - including control surface)

When a control surface deflects during the flight, the resulting **aerodynamic forces are transferred directly** to the control column, tending to pull it back to its neutral position. For a given control surface size and geometry, the magnitude of this reaction force is roughly proportional to the **control surface deflection** δ (hence acting like a spring), and further amplified by the current **aerodynamic load** ($\propto \rho_{air} v^2$) [11, p.17], which the pilot experiences as stiffening of flight controls at higher velocities.

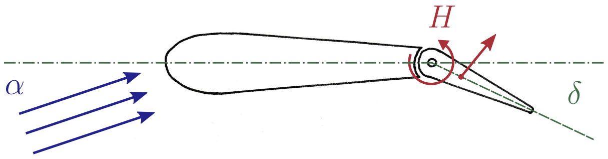


Figure 2.5: Forces on a control surface

Additionally, the angle of attack α, the sideslip angle β and many other secondary aerodynamic effects along with the inertia and aerodynamic damping of control surfaces, internal mechanism's inertia and friction, as well as disturbances such as turbulence wind gusts, aircraft acceleration, engine vibrations etc. all contribute to the resulting forces, felt by the pilot through the control column. All this information packed into the direct haptic feedback gives the pilot a good perception of the state of the aircraft and the resulted stress he is applying to its structure. Hence, the mechanical flight controls are also often referred to as **reversible flight controls**.

Most of the time during the flight, due to effects such as shifts in the effective position of CG and aerodynamic CP etc., a certain amount of constant control surface deflection is required to achieve steady-state flight conditions. [11, p.15-16] If a pilot would have to accomplish this by continuously applying a constant force to the control column, he would very quickly succumb to fatigue. Therefore most primary flight controls, but above all the elevator, have inbuilt **trim tabs** (or alternatively **trimmable stabilisers**) that utilise purely aerodynamic effects to essentially shift the neutral position of the control surface and thus relieve the load from the pilot and the internal linkage components. After correctly **trimming** the aircraft, the pilot can let go of the controls, while the aeroplane will continue in a steady-state flight. In certain cases, a specially mechanically linked tab surface can also perform the function of assisting the pilot with surface actuation (**servo tab**, **spring tab**) or do the opposite and inhibit erratic movement of unstable, too sensitive control surfaces (**anti-servo tab**).

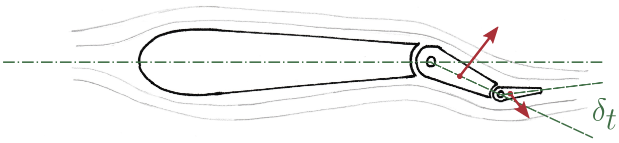


Figure 2.6: Working principle of a trim tab

To sum up, the resulting **hinge moment** H on the control surface (which is ultimately transferred to the control column via linkages) in a steady-state condition can be calculated with the help of the **hinge moment coefficient** C_H:

$$H \;=\; \tfrac{1}{2}\rho v^2 \, S \, \bar{c} \, C_H \tag{2.1}$$

Where:

- $\frac{1}{2}\rho v^2$ is the dynamic pressure
- S is the control surface area behind the hinge line
- $\bar{c}$ is the mean aerodynamic chord of the control surface of the same area and
- C_H is generally assumed to be linearly dependent on control surface deflection δ, tab deflection δ_t and the angle of attack α (or sideslip angle β for rudder controls):

$$C_H \;=\; \frac{\partial C_H}{\partial \delta}\,\delta \;+\; \frac{\partial C_H}{\partial \delta_t}\,\delta_t \;+\; \frac{\partial C_H}{\partial \alpha}\,\alpha \;+\; C_{H0} \tag{2.2}$$

[9]

2.2.2 Hydro-mechanical flight control system

As aircraft increased in size and the achievable velocity grew, the aerodynamic forces on control surfaces became too large to be handled by the pilot's muscle strength alone. A ***hydraulically powered mechanical flight control system*** incorporates hydraulic **power control units (PCU)**, with servo-valves connected to the control column again via mechanical linkages. The PCUs assist with or completely take over the actuation of the control surfaces, while still keeping them kinematically linked with the control column. [11]

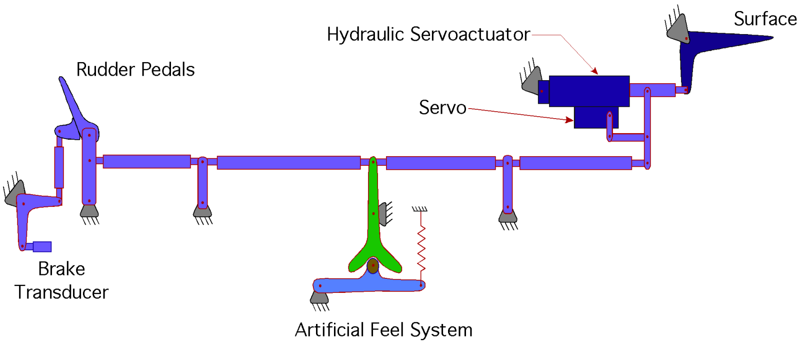


Figure 2.7: Schematic display of a hydraulically powered FCS
[http://daerospace.com/FlightControlSystems, accessed: 12.01.2018]

However, with the entire load from control surfaces now being intercepted by the servo system, the pilot no longer receives any direct force feedback to his control stick and is consequently unable to haptically perceive the aircraft's state (**irreversible flight controls**). By merely controlling the servo-valves he could easily command the actuators to apply enormous forces to the control surfaces and unknowingly over-stress the aircraft. To prevent this, the mechanical linkage subsystem connects to an **artificial feel unit (AFU)**, designated with the task of inhibiting the control column motion in various ways and herewith communicate the control input limitations to the pilot. Examples of AFUs include:

- *spring feel unit* - Incorporates a spring of predetermined stiffness, that provides a reaction force proportional to the control column displacement. [11, p.17]
- *q-feel unit* - Takes into account the difference between the pitot and static pressure to hydraulically reproduce the gain in control force proportional to the aerodynamic load ($\propto \rho_{air} v^2$). [11, p.17]
- *bob-weight* - A weight, built into the FCS of some fighter jets, exerts a large opposing force to the flight controls at High-G manoeuvres, thus physically preventing the pilot to exceed the g limits.

Additionally, a **stall warning stick shaker** is often attached to the control yoke, which monitors the aircraft's angle of attack and, in case of an imminent stall, warns the pilot with violent, unmistakable shaking of the control column along with a loud sound alarm. Should the pilot ignore the stall warning, a **stick pusher** can force a nose-down command to prevent airflow separation.

In pursuit of reducing the workload of the flight crew, **automatisation and stabilisation devices** were introduced in flight control systems, making it necessary to accommodate electrical as well as mechanical inputs to the PCU. The **autopilot** and stabilizers (e.g. **yaw damper**) all gain control access via **electrical signalling**, which can incorporate varying degrees of redundancy (simplex, duplex, quadruplex) depending on the severity of failure consequences. [11, p.21] Purely mechanical FCS can also offer auto-flight functions by including autopilot actuators, usually located in the AFT mechanical system.

2.2.3 Fly-by-wire flight control system

The proven reliability of digital technology as well as the strive to further reduce weight and maintenance lead to the idea of completely eliminating the elaborate mechanical linkages between control column and PCUs/control surfaces and transfer all actuation demands **exclusively** as **electrical signals over wires**. The state of the art ***fly-by-wire (FBW) flight control systems*** are nowadays a well-established technology, mainly found in commercial airliners and military aircraft.

With completely **non-existent mechanical link** between control column and control surfaces, it was no longer a necessity for their displacement/deflection to be kinematically coupled. This freed up the engineers to explore alternative flight control concepts, and indeed aircraft manufacturers developed **different philosophies** regarding the implementation of the fly-by-wire technology.

Boeing, for example, fitted traditional, coupled control yokes with actuators and feel units, which provide force feedback to the pilot and also back-drive the yokes when the autopilot is engaged. [14, ch.11] (analogous to FS CLS) Although the control surfaces and the control column are not physically connected, they retain a "virtual" kinematic link; hence Boeing's fly-by-wire implementation essentially replicates the functionality of traditional hydro-mechanical flight controls. This cuts down on the required transfer training for the pilots and retains their manual inputs as the top authority in the control loop.

In contrast, *Airbus* took advantage of this opportunity and introduced more fundamental changes to the cockpit by replacing the control yokes with uncoupled spring loaded **side-sticks** with no active force feedback. The haptic feedback is not necessary because all pilot inputs are first processed by the onboard computers before they are passed on to the control surfaces. During normal flight control laws, the side-stick positions do not represent deflections of control surfaces; instead, they translate into vertical load factor and roll rate demands, while the **flight-envelope-protection** automatically prevents the pilot from over-stressing the aircraft. [14, ch.12] These features further reduce the crew's workload and contribute to a flight control concept that is ready to accommodate future automatisation features easily.

2.3 Human Sensory and Motor Capabilities

The human body possesses incredibly versatile sensitivity and motor ability, still in many ways unmatched by machines to this day. Without knowing, through our interaction, we learn and remember the nature of the world around us in incredible detail and are therefore able to detect even very slight deviations of its attributes.

The designer of flight simulator's pilot cueing interfaces is usually bound by technical limitations *(e.g. visual system resolution and refresh rate, workspace of the motion platform, etc.)*, which can cause unnatural perceptions that diminish the training transfer. Sometimes, these technical shortcomings are successfully masked with the use of clever trickery that deceives pilot's senses *(e.g. washout filters in motion systems)*. Creating an accurate and convincing illusion of reality clearly demands a fundamental understanding of **the anatomy and physiology of the human body**.

Our bodies are thoroughly covered with diverse **sensory receptors** that gather information about various physical properties and are part of the **sensory nervous system**. Signals they produce are sent to the **central nervous system**, which performs the complex processing necessary to interpret this data as different **sensations/perceptions**. Table 2.1 shows five basic types of sensory receptors (classified according to the stimuli they detect) and the corresponding sensations they enable: [6]

Receptor type	**Sensations**
(1) mechanoreceptors	skin tactile and deep tissue sensibilities, hearing, equilibrium, arterial pressure
(2) thermoreceptors	cold, warmth
(3) nociceptors	pain
(4) eletromagnetic receptors	vision
(5) chemoreceptors	taste, smell, arterial oxygen,...

Table 2.1: Classification of Sensory Receptors
according to [6, p.560]

The classification of human senses is not universal and in literature a few interpretations with different terminologies can be found. Generally speaking, a pilot handling a control column will receive feedback about it's state mainly through his **haptic perception**, which includes:

- **tactile sensations** from the skin *(touch, pressure, vibration, tickle senses)* and
- **proprioceptive sensations** relating to the physical state of the body - particularly limbs *(position and relative movement sensations, tendon and muscle sensations...)*.

[8, p.2], [6, p.571]

These sensations are all enabled by different types of **mechanoreceptors**, each specialized for a specific range of mechanical stimuli. Johannsson (as cited in [8, p.3]) labels mechanoreceptors in the glabrous skin of the human hand by their:

- **adaptation speed** as:
  - Fast-adapting (FA) - specialized for fast changes in stimulus
  - Slow-adapting (SA) - specialized for static stimulus
- **receptive field size** as:
  - small with well defined borders (Type I)
  - large with diffuse borders (Type II)

Their **bandwidth** and peak sensitivity frequency have been thoroughly studied by many researchers through experiments. The upper frequency range of stimuli is captured by the **Pacinian Corpuscles** (FA, Type II mechanoreceptors), which can detect frequencies ranging from 20 to 1000 Hz with peak sensitivity in the 250-550 Hz range as stated by Bell et al. cited in [8, p.4]. Working together, the mechanoreceptors in the glabrous skin of the human hand are therefore able to cover a **broad spectrum of mechanical stimuli ranging from steady state to 1000 Hz**. [8, p.3]

Guided by the feedback from the sensory nervous system, humans are also capable of physically moving and manipulating the surrounding objects by exerting forces and torques on them using our **muscles and skeleton**. Very precisely controlled movements are made possible by the **motor nervous system** [6, p.543], however our (fingertip) movement and positioning is limited to **below 8-10 Hz** (according to Loomis, Lederman and Shimoga as cited in [8, p.4]). Evidently, the **actuation bandwidth** is dramatically lower compared to the much higher **sensory bandwidth**. This demonstrates a large **asymmetry in human sensory and motor capabilities**, which directly influences the way we interact with the environment. [8, p.4]

In an aircraft, the pilot is an integral part of the flight control loop, where his sensory and motor systems essentially act like any other sensors and actuators. The information about their bandwidth is significant when determining the requirements for the flight simulator's CLS. It allows us to estimate its required dynamic performance, and also assess to what degree the imperfections in the control force simulation will be tolerable.

2.4 Haptic Interfaces

A *haptic interface* is a device that provides the user with **artificial touch-based feedback** according to the performed movements, with the goal to simulate the sensations of direct contact with objects from a virtual or distant environment. It consists of a variety of sensors and actuators, usually combined in assemblies closely related to robotic devices. [8, p.6]

Such interfaces can range from something as simple as your smartphone briefly vibrating while you press a button on the touchscreen, to something very complicated such as a telemanipulator of an advanced surgical robot, where the surgeon requires accurate feedback of material's texture and other physical properties to be able to perform delicate procedures.

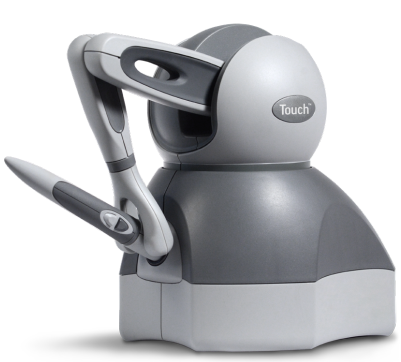


Figure 2.8: The Touch^{TM} Haptic Device by 3D Systems
(commercially available, universally usable)
[https://www.3dsystems.com/haptics-devices/touch, accessed: 7.1.2018]

During **physical interaction with real objects**, we exert forces on them which consequently result in object's motion. This **force-motion relationship** is governed by *Newton's laws of motion* and dependent on various mechanical parameters and secondary force sources such as inertia, stiffness, viscous damping, dry friction etc. With our high-bandwidth haptic sensory system, we can detect tiny variations of the resulting motion and create a perception/understanding of the object's structure and the external forces at play.

In a **virtual reality environment**, however, the previously mentioned effects have to be modelled and simulated to provide a real-time reference of how the force and motion of a virtual object should be coupled. The task of a **haptic interface** is then to mimic this virtually created force-motion relationship by controlling the output of its **actuators**. According to the control strategies used to achieve this goal, Kuchenbecker differentiates two main types of haptic interfaces: **admittance type** and **impedance type** [8, p.6], further examined in Table 2.2.

<table>
<tr><th colspan="2">Haptic Interfaces</th></tr>
<tr><th>admittance type</th><th>impedance type</th></tr>
<tr><td colspan="2"><i>control strategy</i></td></tr>
<tr><td>$\vec{v}_{hi} = Y_{env}\ \vec{F}_{user}$
• measure the user-exerted force $\vec{F}_{user}$
• control the required motion of the haptic interface $\vec{v}_{hi}$
• Y_{env}...virtual environment admittance</td><td>$\vec{F}_{hi} = Z_{env}\ \vec{v}_{user}$
• measure the user-imposed motion $\vec{v}_{user}$
• control the feedback force of the haptic interface $\vec{F}_{hi}$
• Z_{env}...virtual environment impedance</td></tr>
<tr><td colspan="2"><i>typical contruction</i></td></tr>
<tr><td>• strong, stiff, non-backdriveable mechanisms
• force sensor at the endpoint
• highly geared motors & position sensors at the base</td><td>• lightweight, backdriveable mechanisms
• DC motors & position sensors at the base</td></tr>
<tr><td colspan="2"><i>potential sources of deviations from the target</i></td></tr>
<tr><td>Y_{env}
• imperfect force sensing
• imperfect motion output
• mechanism friction, backlash and similar undesirable effects</td><td>Z_{env}
• imperfect motion sensing
• imperfect force output
• mechanism friction, backlash and similar undesirable effects</td></tr>
<tr><td colspan="2"><i>excel at</i></td></tr>
<tr><td>• inhibiting movement ($Y_{env} \approx 0$)
(by taking advantage of the mechanism's naturally large inertia & stiffness)</td><td>• near free-space motion ($Z_{env} \approx 0$)
(due to mechanism's naturally low inertia and low friction)</td></tr>
<tr><td colspan="2"><i>struggle with</i></td></tr>
<tr><td>• free-space motion ($Y_{env} \gg 0$)
(due to difficulties with disguising mechanism's naturally large inertia & stiffness)</td><td>• inhibiting movement ($Z_{env} \gg 0$)
(due to limitations in artificial replication of the large feedback forces required)</td></tr>
<tr><td colspan="2"><i>other pros & cons</i></td></tr>
<tr><td>+ stronger mechanisms are able to provide larger feedback forces
– high cost of accurate force sensors
– safety concerns for operating high inertia, non-backdriveable drive systems in direct contact with the user</td><td>+ simpler design
+ lightweight drive systems present a lower risk of injury</td></tr>
</table>

Table 2.2: Comparison of admittance and impedance type haptic interfaces according to Kuchenbecker [8, p.7-8]

2.5 Control Loading Systems

The **control loading system** (CLS) is the part of a flight simulator responsible for replicating the feedback forces on the control column that a pilot would experience during an actual flight. Since they provide a touch-based feedback from a virtual reality environment, control loading systems belong in the family of **haptic interfaces**.

2.5.1 Model of the flight control system

Simulation of the control column's haptic feedback begins with a **physical model** of the real **flight control system** (FCS), which serves as a reference of the desired system's dynamics. It describes the correlation between the torque the pilot exerts on the control column T_{pilot} and the resulting control column motion, here described as a state vector consisting of the (rotational) position and velocity $[\varphi \quad \dot{\varphi}]^T$. Furthermore, since the FCS dynamics and parameters vary according to the aircraft's state (e.g. velocity, angle of attack etc.) and secondary trim controls, the FCS model also has to constantly exchange parameter data with the rest of the flight simulation software.

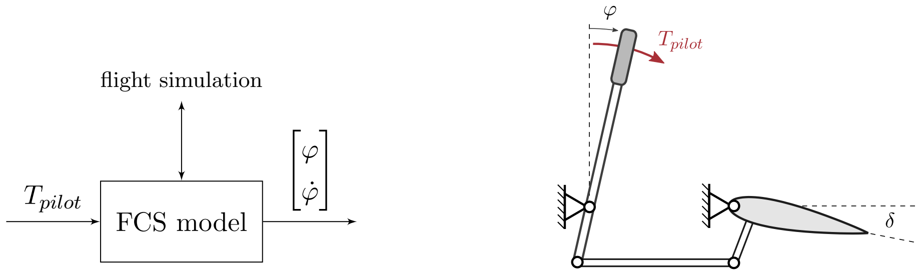


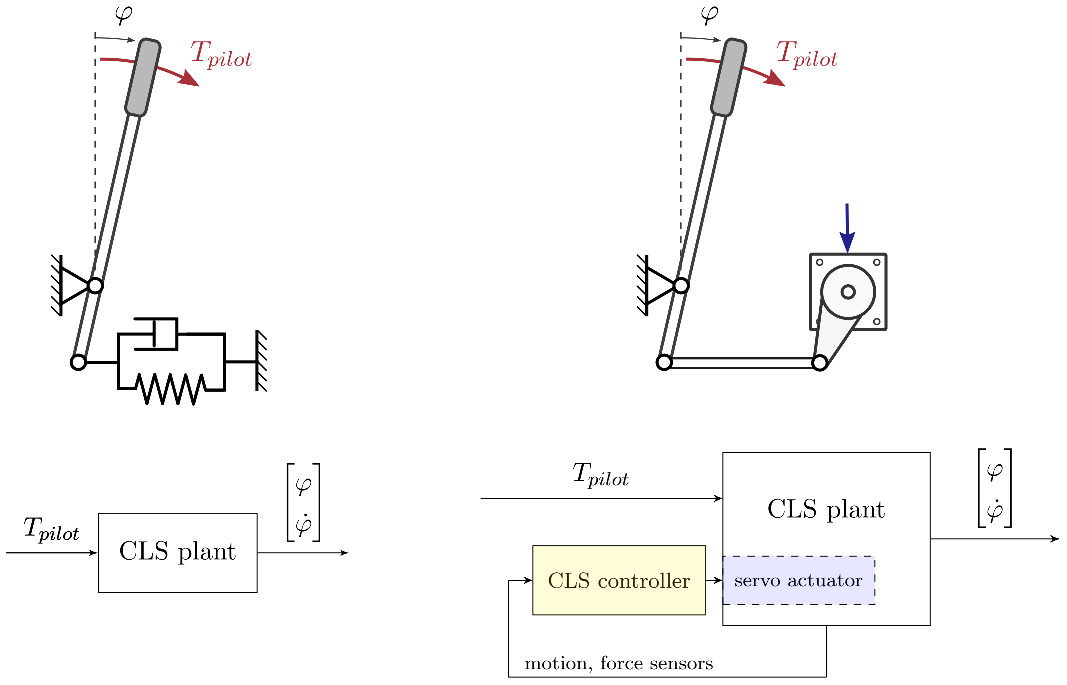
Figure 2.9: Model of the flight control system

To reverse engineer the internal structure of the FCS, various modelling techniques can be applied. *High fidelity flight simulators* will feature **detailed FCS models**, where all *system components* (e.g. hydraulic pistons and valves, mechanical linkages, AFUs), *physical effects* (e.g. experimentally verified aerodynamics, cable stretching, hysteresis, deadbands etc.) and even *malfunction scenarios* are meticulously modelled and extensively validated to ensure the highest degree of realism.

For *academic and research purposes* on the other hand, the FCS is often simplified as a second order **mass-spring-damper system** [5] due to its highly desired linear dynamics in the control system analysis. Expanding the linear model with a variable stiffness (correlated to dynamic pressure), additional local non-linearities such as hard stops or detents, as well as including the ability to offset the zero load position (for the trimming function) [2, p.22], already creates a reasonable approximation of the flight controls dynamics (as it will be demonstrated in section 3.5).

2.5.2 Passive and active control loading

After determining the target FCS dynamics, it is necessary to design a control loading device that will be able to mimic this behavior. This can be achieved in several ways:



(a) passive CLS

(b) active CLS

Figure 2.10: Passive and active CLS

The simplest approach is to implement a **passive CLS**, where the engineer designs the CLS plant in such a way, that it most closely resembles the FCS model. This is usually achieved with an assembly of springs and dampers attached to the control column, which results in an affordable and low maintenance solution. However, the CLS plant parameters remain fixed therefore in most cases a passive system will not be able to accurately portray the changing control column dynamics during different phases of flight. Applications of passive CLS include: passive fly-by-wire control sticks, non-automated throttle quadrants and simplified primary flight controls.

To be able to vary the dynamic behaviour of flight controls during simulation, we have to introduce a **servo actuator** into the CLS plant, that replaces the passive springs and dampers previously attached to the control column. An **active CLS** can be reprogrammed to resemble a (theoretically) arbitrary FCS dynamics by merely changing the structure or parameters of the CLS controller. The **actuator** connected to the control column can either be a **hydraulic** piston or an **electric** motor. In the past, mostly hydraulic actuators were used due to their high power density and excellent dynamic characteristics. However, the high maintenance of hydraulics peripherals and advances in the electric motor drive technology enabled that today, most control loading systems feature a single electric motor drive per channel. [2, p.22] This results in a very simple, clean and low maintenance construction design, that is flexible, reprogrammable and therefore able to simulate flight controls of various aircraft types.

2.5.3 Active CLS controller architectures

The control system diagram of an active CLS seen in Figure 2.10b is a very generalized one, since actually many variations in the details of the **control system architecture** are available.

An **impedance type CLS**, as already explained in Table 2.2, will measure the user-imposed motion and based on the FCS model determine the required reaction force/torque the actuator needs to provide. This simplest active configuration does not include a force sensor at the control column and directly controls the force/torque output of the actuator by supplying a specific electric current or hydraulic pressure. This means, the user has to move the mechanism, while the actuator will merely be able to inhibit this motion and dissipate energy. Seeing that all inertia and friction present in the CLS plant mechanism have to be overcome by the user, the CLS mechanism itself needs to be lightweight, low-friction and back-driveable. This control design is mostly used for active side-sticks and automated throttle quadrants, where lower forces are present in the system (due to short leverages and one-hand operation), therefore small sized motors can be placed in the direct proximity of the control stick (sometimes even in a direct-drive configuration).

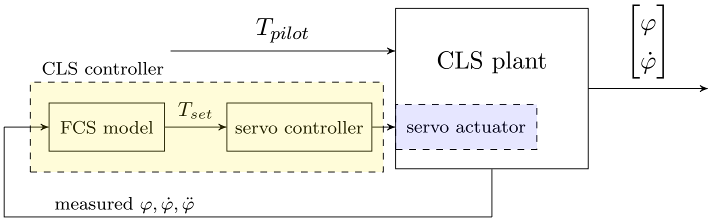

T_{pilot}
CLS plant
$\begin{bmatrix}\varphi\\\dot{\varphi}\end{bmatrix}$
CLS controller
FCS model
T_{set}
servo controller
servo actuator
measured $\varphi, \dot{\varphi}, \ddot{\varphi}$

Figure 2.11: Impedance type active CLS

In the case of coupled control yokes with long leverages (and also rudder pedals), however, large forces are involved and demanded from the actuators. Large sized motors and often included reduction gears will inevitably result in more inertia and an unacceptable amount of friction in the CLS mechanism. The solution is an **admittance type CLS**, where a **force sensor**, placed at the end of the mechanism near the control column, accurately measures & estimates the force applied by the pilot and a motion demand is placed on the servo controller. This way, even if the pilot exerts a small force on the control yoke, the force sensor can detect it and the FCS model sends the motion demand to the servo controller, which is able to apply the necessary torque to compensate for the CLS mechanism's friction. Being able to not only dissipate but also input work when needed, this configuration can mask the existing friction in the CLS plant and provide the larger forces on the control column.

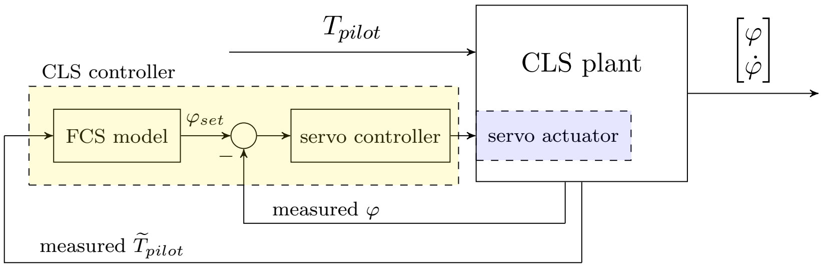


Figure 2.12: Admittance type active CLS
(position-loop)

In Figure 2.12 we can see that the CLS controller's structure consists of two cascaded control loops. The **outer loop** contains the FCS model and determines how the control column should behave based on the measured load. The **inner loop** receives the target signal generated by the outer loop and controls the servo actuator in such a way that the control column accurately tracks the FCS model output.

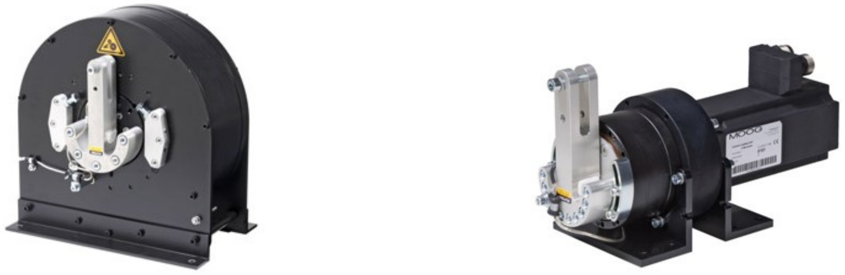
Furthermore, in their research, Gerretsen, Mulder and van Paassen [5] also differentiate among **position-loop**, **velocity-loop** and **force-loop** CLS architectures, according to whether the the target signal for the inner loop is a position, velocity or force demand. The position-loop and velocity-loop are similar admittance type architectures and function as shown in Figure 2.12. Meanwhile, the described force-loop architecture is actually an impedance type CLS; however, it features a force sensor in the linkage as well. This force sensor provides a feedback for the servo controller in the inner loop and allows the actuator to regulate its output torque directly at the control column, which is an alternative solution to solve the drawbacks of the previously described impedance type CLS without a force sensor.

The specific control system layout chosen in this thesis along with its performance and limitations shall be discussed in detail in section 3.6.

2.5.4 Hardware components, available market solutions and challenges

The today's state of the art commercial control loading products mostly utilise digitally controlled **electric servo motors** and control architectures featuring force sensors. Due to the already discussed extremely sensitive and high-bandwidth human sensory system, the control loops usually run at **iteration rates of several kilohertz** to ensure that the pilot feels a smooth motion of the control column.

The company **Moog** as one of the market leaders, for example, offers a variety of control loading actuators. The highest fidelity can be achieved with their 'high dynamic control loaders', which feature high torque, electric servo motors in a direct-drive set-up. [1] This eliminates the need for a reduction gearbox, which inevitably introduces more friction into the system and can cause small motion control imperfections felt by the pilot. The more affordable 'medium dynamic control loaders' trade some performance and fidelity by taking advantage of a gearbox to be able to use smaller, lower torque motors.



(a) High dynamic control loader

(b) Medium dynamic control loader

Figure 2.13: Control loading devices by Moog [1]

The **challenges** in the design of control loading devices include:

- guaranteeing stability of control loops for various FCS model parameters,
- eliminating the unwanted cogging torque of the motor,
- reducing the motion control imperfections due to gearbox etc.

The mechanical linkage between the control loading actuator and the control column needs to be stiff (to enable realistic hard stop simulation) and as lightweight as possible (to avoid control stability issues).

The influence of accurate FCS simulation on the achievable training transfer makes the CLS an important part of the flight simulator's certification process. Additionally, strict **safety regulations** (often requiring redundant control systems) have to be met, since powerful and potentially dangerous actuators are operated in direct contact with the user.

Chapter 3

Solution development

After an overview of the fundamental theory, it is now time to focus on the **design process** and the steps to the **commissioning** of our **control loading system test bench**. First, we established a basic solution concept based on the past experience as well as future goals and formulated a list of functionality and performance requirements. Keeping this in mind, we designed and built the CLS test bench and selected properly sized hardware components. Choosing the right control hardware, that would be robust and adaptable for future research & development as well as directly transferable to the full-sized assembly was one of the critical decisions that had to be made. Next, I continued by creating and analysing the necessary models of the CLS test bench plant and the FCS model used in the simulation of flight controls. Last, but not least, I developed a control system strategy and implemented it on the test bench, to be able to further study and discuss the controller architecture, tuning, stability and performance.

3.1 Solution concept and requirements

The primary flight controls in the wide-body aircraft research flight simulator at TU Graz were initially all equipped with a **passive CLS**, containing an assembly of mechanical springs and dampers. In the strive to further improve flight simulator's fidelity, we are currently in the process of **gradually upgrading** simulator's flight controls **to a fully active CLS**, by replacing the original spring-damper assemblies with electric motors.

The first flight controls channel, to be upgraded with an electric actuator were the **rudder pedals**, which were the logical place to start because the pilot's feet are a lot less susceptible to imperfections in the simulation in comparison to his hands. The active rudder pedals CLS was **already implemented** in the simulator **by previous collaborators** and it features a zero-cogging AC servo motor from Infranor with a planetary gearbox and chain transmission, connected to the rudder pedals via a push-pull rod linkage. The control system architecture resembles an **impedance type** haptic interface, implemented on low-cost, custom-made, Arduino-based hardware. (See [7].) This configuration was a success in the case of rudder pedals; however, the remaining friction in the transmission (that cannot be concealed without using a force sensor) presents a concern as we tackle the upgrade of the elevator and aileron flight controls, which require a higher fidelity.

Before proceeding with further work on other flight control channels, we decided to reevaluate, thoroughly study and experiment with other **CLS controller architectures** on a **small scale test bench**, which could be in operation without causing flight simulator downtime. I was confident, the solution to our friction problem is an additional force sensor at the control column, which could detect even small pilot force inputs and allow for an **admittance type CLS**, while preserving the previous **mechanical construction** (consisting of a geared AC servo motor and a mechanical linkage to the control column). Moreover, the wish to implement more advanced CLS control algorithms and more detailed FCS models, resulted in the need to also acquire more powerful, reliable and specialised **control hardware**.

List of requirements for the CLS test bench:

- mechanical design and components:
  - mechanical construction analogous to the full-scale implementation and compatible with its constraints
  - the ability to reconfigure or accommodate alternative hardware components
  - simple manufacturing and assembly (t-slot profiles)
  - easy servicing accessibility
- control hardware:
  - achievable outer loop iteration rates of several kHz
  - directly transferable to the future full-scale implementation
  - TCP/IP protocol support for seamless integration in the existing flight simulator network
  - expandable with the remaining flight control channels
  - sufficient computing power for a parallel simulation of multiple CLS channels
  - modular design for simple component exchanging
  - fast, user-friendly programming and diagnostics
(gentle learning curve that supports short-term student projects)
  - a variety of supported programming languages/techniques

3.2 Test bench construction and hardware selection

3.2.1 Mechanical design and assembly

The mechanical assembly of the small-scale test bench needs to resemble a design, that will be transferable to the **full-scale version** of the CLS. The accessibility and space constraints in the wide-body aircraft flight simulator at TU Graz will require the electric servo drives to be located at the front of the cockpit shell and for the elevator and rudder controls, the existing connections on the control column could be utilised to link the servo via a push-pull control rod resulting in a simple **four-bar linkage** mechanism.
(Note, that the linkage to the aileron control wheel will inevitably pose additional design challenges, due to the relative movement of the elevator controls. Meanwhile, the lower torques required for the thrust levers might allow for an even simpler and more compact transmission solution. Nevertheless, for now, the test bench mechanical design shall resemble the four-bar linkage solution, implementable on the elevator and rudder channels.)

The **final test bench design** is the result of an iterative process, heavily influenced by component selection and sizing (see the following subsections). The **base frame** was built from standard t-slot aluminium extrusions, resulting in a simple and flexible assembly with excellent accessibility to all inbuilt components. The **moving mechanical components** (schematically shown in Figure 3.1) ultimately form our **CLS plant**: the *electric servo motor* with an attached *planetary gearbox* connects to the (1-axis) *control column* via a *four-bar linkage*, whose connecting-rod (coupler) features a *load cell* that measures the force present in the rod. These moving parts should be designed as stiff and lightweight as possible (to reduce control loop stability issues later on). The **conductive mounting plate** accommodates all necessary *control hardware* and *electrical wiring* in an unobstructed and compact manner.

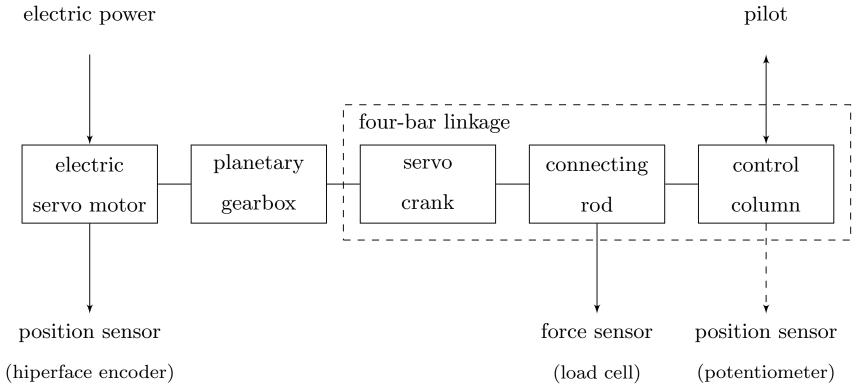


Figure 3.1: A schematic diagram of the test bench CLS plant

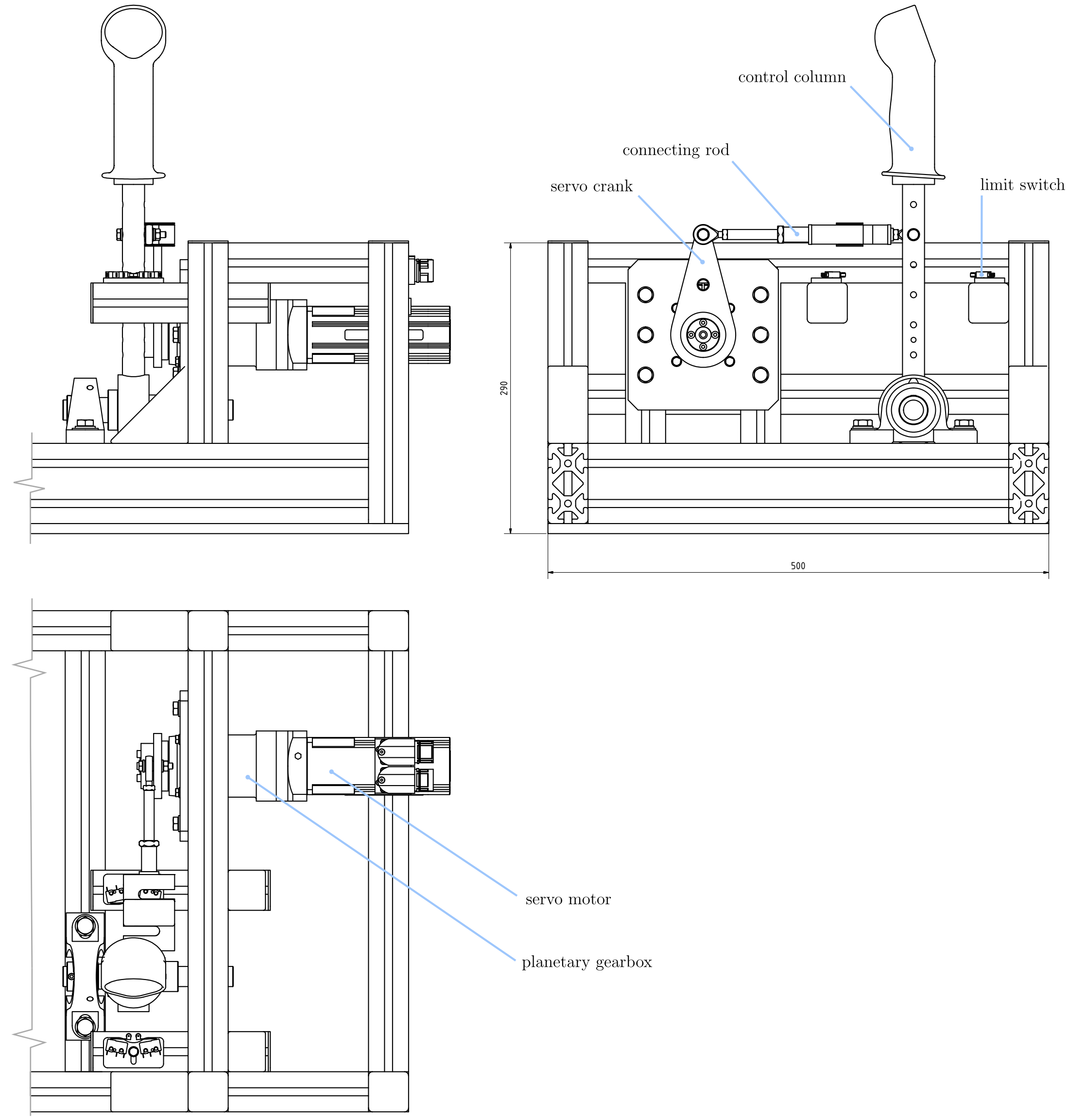


Figure 3.2: CAD mock-up of the CLS test bench mechanism

Dimension	Unit	Value
control column (axis-hand grip)	mm	~ 300
control column (axis-coupler connection)	mm	175
connecting rod length	mm	209
servo crank arm length	mm	100
control column deflection limits	deg	± 20

Table 3.1: Basic dimensions

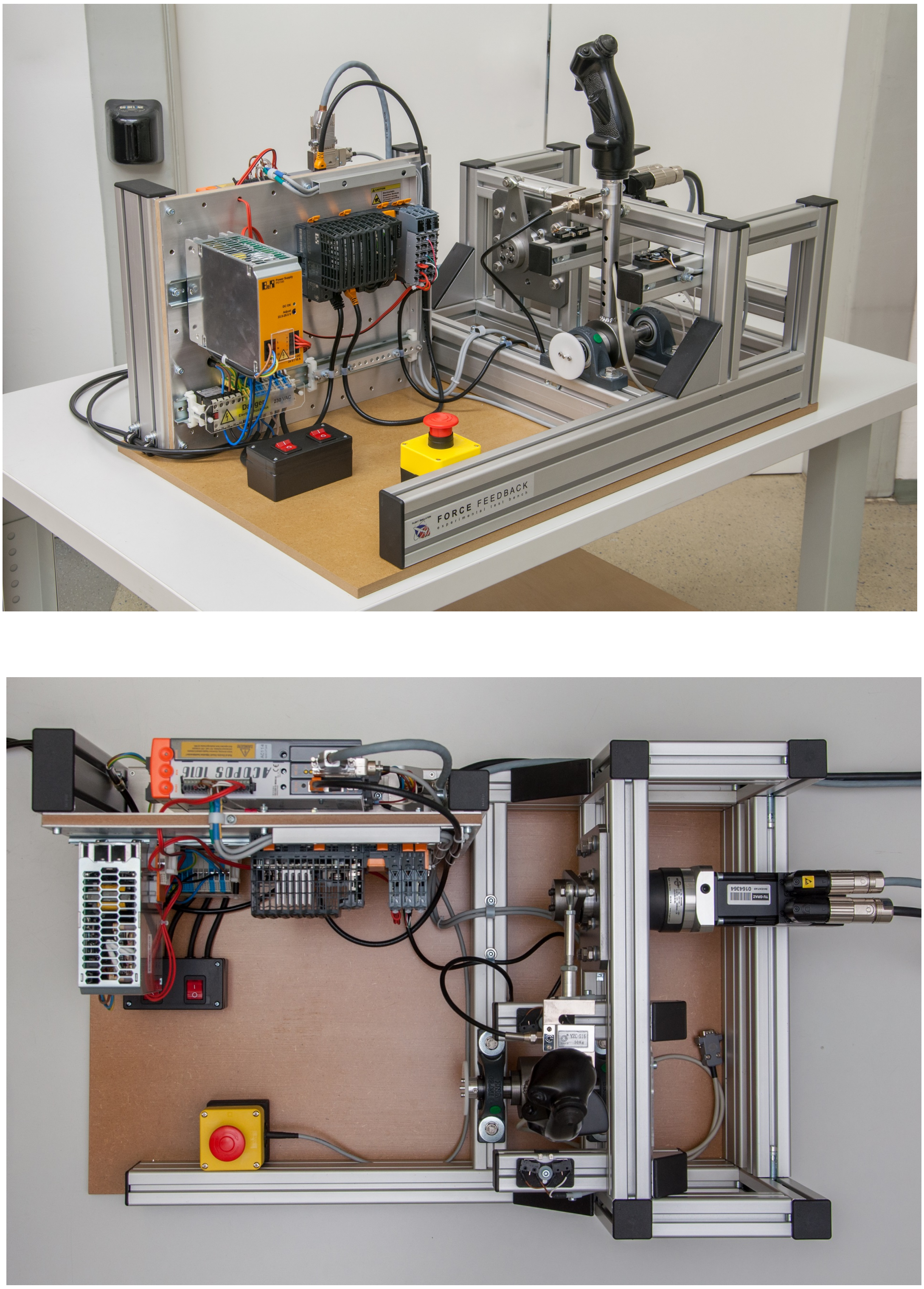


Figure 3.3: CLS test bench

3.2.2 Servo drivetrain

Previous good experience lead to the decision to again choose an appropriate servo motor from the **Infranor XtraforsPrime** series. These are compact size, high torque and speed, permanent-magnet synchronous motors with a slotless stator design resulting in zero cogging torque, which is especially welcome for CLS applications. Adding a planetary reduction gearbox will allow for a more compact and reasonably priced motor, that is still able to satisfy the relatively high torque demand.

The **sizing** of the servo drivetrain configuration begins by estimating the required **continuous torque** the motor will have to provide. For our purely experimental purposes, we wish for at least 100 N of continuous static feedback force to the pilot's hand at the control column grip. An **Infranor XtraforsPrime FP0105 motor** (stall torque 1.0 Nm) coupled with a **Neugart PLN070 planetary gearbox** (gear ratio $i_{gbx} = 25$ in two stages, backlash < 5 arcmin) are going to be up to this task. Assuming a constant fourbar-linkage ratio of $i_{fb} = 1.75$ and gearbox efficiency $\eta_{gbx} = 0.85$, we can estimate the torque, the servo can provide on the control column as

$$T_{cc} = T_{motSt}\, i_{gbx}\, \eta_{gbx}\, i_{fb} = 1.0 \text{ Nm} \cdot 25 \cdot 0.85 \cdot 1.75 = 37.19 \text{ Nm}$$

and the resulting static force on the pilot's hand is obtained by dividing T_{cc} with the lever distance of the grip

$$F_{ph} = \frac{37.19 \text{ Nm}}{0.3 \text{ m}} = 124.0 \text{ N}$$

which satisfies (and exceeds) our minimum continuous force demand.

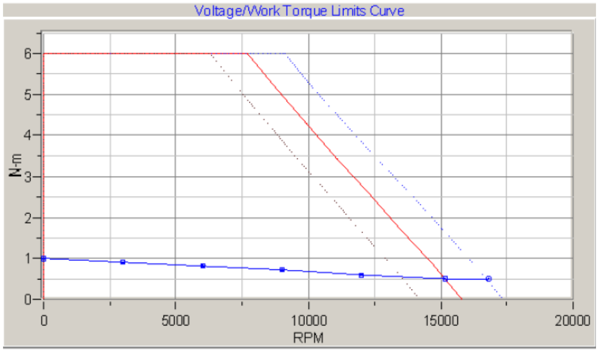


Figure 3.4: Torque-speed chart

(Infranor XtraforsPrime FP0105)

[Service Manual XtraforsPrime v 11.06]

Characteristic	Units	Value
rated torque (at 3000rpm)	Nm	0.90
stall torque	Nm	1.00
peak torque	Nm	6.00
max motor speed	rpm	11400
rated current (at 3000rpm)	A	3.11
stall current	A	3.45
peak current	A	20.38
supply voltage	V	230
N° of pole pairs	-	4
torque constant K_t	Nm/A	0.29
EMF constant K_e	Vs/rad	0.17
cogging torque	Nm	0
rotor inertia	kgm^2	4.0e-5

Table 3.2: Motor technical data

(Infranor XtraforsPrime FP0105)

[Service Manual XtraforsPrime v 11.06]

Next, we have to check the **dynamic performance** of the servo motor. The corresponding **torque-speed chart** (see Figure 3.4) gives a general insight by showing the maximal available torque as a function of motor's speed. The blue line represents limits for continuous loads, while the red line shows the achievable peak performance for short, intermittent time periods.

A key aspect of motor sizing and the achievable dynamic performance is the drivetrain's **inertia ratio**, which compares the *motor's inertia* with the *load's reflected inertia* to the motor axis:

$$\text{inertia ratio} = \frac{I_{loadRef}}{I_{motor}}$$

For most applications with moderate performance requirements, drive manufacturers recommend inertia ratio of around 5. For uncritical situations is not uncommon to use higher ratios of 10, 100 or above, however, if excellent dynamic performance and control loop stability are required (as they are in our case), one should tend towards smaller inertia ratios of 2 or even below. [12]

The inertia ratio for our CLS test bench is calculated based on Table 3.3. Inertias of separate components were taken from the respective data sheets or calculated from the CAD model. (For this rough estimation, the four-bar linkage ratio is assumed to be constant. The inertia contributed by the coupler, whose motion is not rotational, was approximated as two point masses at the connecting joints, rotating together with the crank or the control column.)

Component	Inertia	Gearing ratio (to motor axis)	Reflected inertia (to motor axis)
	I [kgm^2]	i [-]	$I_{Ref} = I/i^2$ [kgm^2]
motor	$4.00 \cdot 10^{-5}$	1	$4.00 \cdot 10^{-5}$
planetary gearbox	$2.49 \cdot 10^{-5}$	1	$2.49 \cdot 10^{-5}$
servo crank	$9.40 \cdot 10^{-4}$	25	$0.15 \cdot 10^{-5}$
coupler part 1	$1.25 \cdot 10^{-3}$	25	$0.20 \cdot 10^{-5}$
control column	$3.51 \cdot 10^{-2}$	$25 \cdot 1.75 = 43.75$	$1.83 \cdot 10^{-5}$
coupler part 2	$1.15 \cdot 10^{-2}$	$25 \cdot 1.75 = 43.75$	$0.60 \cdot 10^{-5}$

Table 3.3: Inertias of the motor and load components

We determined the inertia ratio of $5.27/4.00 = 1.32$, which should give us a dynamic and high performing motion system. Note, that a certain inertia margin should be included when sizing the servo drivetrain. As the pilot's hand rests on the control column, the mass of the limb will certainly add to total load inertia that the motor has to move.

Ultimately, we are interested in the **bandwidth** of the servo actuator when it is attached to the rest of the CLS plant. The FCS (which the actuator is trying to simulate) usually consists of mechanical components such as springs and hard stops, that can contain high-frequency contents in their response. In order to be able to recreate these dynamics and achieve good fidelity of the CLS, the bandwidth of the CLS actuator should be at least ten times larger than the bandwidth of the FCS it is simulating. [13] Additionally, we should avoid any problematic **resonant frequencies** in the plant by designing the linkage (gearbox, four-bar linkage) as stiff as possible.

3.2.3 Sensors

The plant incorporates several sensing devices which are used in the control loops or for the safety functions.

- **Hiperface motor encoder**
  The Hiperface SKM36 multiturn absolute encoder, mounted directly on the motor shaft, measures its rotation in 4096 increments/revolution and is able to track its absolute position for 4096 revolutions. The motor encoder monitors the position of the stator coils relative to the rotor (phasing offset), which a vital information when driving a synchronous motor. Additionally, it provides a feedback signal for servo's velocity and position control loops. We will also use the motor encoder to estimate the control column's position by assuming a rigid and low backlash linkage between them.

- **Position potentiometer**
  The precision potentiometer is an optional sensor, that can directly measure the position of the control column and outputs an analogue signal. Our goal is to accomplish all position measurements with the motor encoder alone; therefore the potentiometer is not actively used, but merely an additional redundant sensor available for experimentation purposes.

- **Load cell**
  The load cell is located in the connecting rod of the four-bar linkage and consists of a strain gauge in a full Wheatstone bridge configuration. The measured voltage difference signal is proportional to the longitudinal force present in the connecting rod. The sensor attempts to measure the pilot's currently exerted force; therefore it is advantageous that it is placed as close to the control column as possible before any major friction or plant dynamics can severely influence the measurement. According to the calculated maximal expected force in the connecting rod, we chose a 50 kg s-beam load cell (2 mV/V) that can measure tension and compression and will easily cover the required measuring range. Out of curiosity, we opted for a low-cost product for the initial experimentation, which surprisingly turned out to offer adequate performance and precision for our purposes.

- **Limit switches**
  The two travel limit switches (positive and negative) provide a safety function by preventing an accidental collision between the control column and the stop blocks. To ensure fail-safe operation, it is crucial they are wired as 'normally closed' switches. During normal operation, limit switches are never triggered, because beforehand a virtual hard stop of the CLS should be reached.

- **Emergency stop**
  The emergency stop button, located in an unobstructed area, implements a Safe Torque Off (STO) function that can instantly remove the entire torque from the motor, should the system start behaving in an unexpected and dangerous manner.

3.2.4 Control system hardware

The control algorithm enhancements, we were striving to achieve on our CLS called for **industrial grade control hardware**, that would offer more computing power, reliability and also be able to implement all the necessary safety functions. I was pleased to discover that all our requirements could be met by hardware from a local manufacturer **B&R Automation**, which offers a wide selection of automation products and provided us with excellent support during the education and commissioning process.
Listed below are some of the **highlighted features** of their product line, that convinced us it was the right choice:

- **highly modular and expandable hardware design**
  Since we wish to transfer the same control hardware to the full-scale CLS implementation in the future, it was crucial that the system would be able to accommodate additional channels as it grows. All B&R products are compatible with each other and the interconnecting I/O modules allow for a very customisable control system that has the ability to interface virtually any sensor signal or standardised communication protocol if necessary.

- **high performance computing and communication abilities**
  Motion control demands real-time (deterministic) computing and the especially high iteration rates required for a smooth and stable CLS operation were the key feature that needed to be met. The performance bottleneck is usually the communication line between the PLC and the servo drive, which will be part of the outer loop in our CLS control system. Here, B&R uses a self-developed, deterministic, real-time *Ethernet Powerlink* protocol, which can achieve excellent cycle times of 400 μs.

- **B&R Automation Studio software**
  The developer or user gains full access to the control hardware through the PC application 'B&R Automation Studio'. This very powerful, intuitive and easy to learn software covers the whole range of tasks: programming, simulation, commissioning, diagnostics etc. If the PC is connected to the control hardware via ethernet, the user can monitor the processes on online devices in real time.

- **industry standard libraries**
  The Automation Studio as a programming environment supports all the PLCopen automation industry standard languages (e.g. structured text, ladder diagrams, function block diagrams etc.) as well as the lower level C or C++ languages. It also already provides the programmer with rich function libraries for diverse tasks (e.g. PLCopen motion control library), which greatly simplifies and accelerates the application development.

- **Matlab Simulink Target**
  Most modern control system developments heavily rely on the indispensable Matlab & Simulink tools that enable a rapid system development and simulation. B&R's Target

for Simulink has the ability to automatically generate a hardware-ready C code from a Simulink model of the controller. This incredible feature allows the engineer to intuitively and easily design high-level, hardware independent logic, that can be immediately tested on the real hardware. The convenient operation in external mode even allows connecting to an online target device to tune program parameters or plot variables in real time.

- **extensive documentation and education materials**
  At university, the students are usually involved in short-term projects, and it is vital that the skills for operating the hardware can be quickly and easily obtained. B&R Automation Academy provides extensive high-quality learning materials in the form of documentation and online tutorials as well as organised training seminars.

After carefully refining our requirements and consulting with the B&R Automation specialists, the devices listed in Table 3.4 were selected for our CLS control system. The corresponding wiring diagram can be seen in Figure 3.5.

Device	**Short technical specification***
PLC CPU X20CP1585	”X20 CPU, ATOM 1.0 GHz, 512 MB DDR2 RAM, 1 MB SRAM, removable application memory: CompactFlash, 1 insert slot for X20 interface modules, 2 USB interfaces, 1 RS232 interface, 1 Ethernet interface 10/100/1000 Base-T, 1 POWERLINK interface”
I/O module AI1744	”X20 analog input module, 1 full-bridge strain gauge input, 24-bit converter resolution, 5 kHz input filter”
I/O module AI4632	”X20 analog input module, 4 inputs, ±10 V or 0 to 20 mA, 16-bit converter resolution, configurable input filter, oscilloscope functions”
Servo drive ACOPOS 8V1016.50-2	”ACOPOS servo drive, 3x 110-230 V / 1x 110-230 V, 3.6 A, 0.7 kW, line filter, integrated braking resistor and electronic secure restart inhibit” + plug-in module POWERLINK V2 interface (8AC114.60-2) + plug-in module HIPERFACE interface (8AC121.60-1)
Power supply 0PS1100.1	”24 VDC power supply, 1 phase, 10 A, input 100 to 240 VAC, wide range, top-hat rail installation”

Table 3.4: Chosen B&R Automation control hardware
[*technical specifications quoted from B&R Automation datasheets]

The X20CP1585 **PLC CPU** is the heart of the control system, which runs a deterministic, real-time operating system Automation Runtime. At power-up, the program stored on the CompactFlash card is loaded, and the assigned tasks are cyclically executed. The Powerlink port is used to establish a cyclic connection to other B&R devices such as servo drives, external I/O modules and other CPUs. Through the ethernet port, the device can be connected to a network, from where it is accessible from a PC to perform commissioning and diagnostics in Automation Studio or to exchange data with other devices via various protocols (e.g. TCP/IP).

The **I/O modules** can be connected directly to the PLC housing in series and at any time additional modules can be added to the chain. Our control system currently only possesses two modules. AI1744 is the amplifier and ADC for the signal coming from the load cell's full-bridge strain gauge, while the AI4632 is a general 4 channel ADC, that could be used for reading the position potentiometer signal or any other analogous signal.

The **servo drive** receives motion demands (position, velocity or torque) from the PLC via the Powerlink according to the PLCopen standard. It possesses its own CPU, that can cyclically perform position, velocity and torque control of the motor axis and as a state machine also handle discrete events (e.g. limit switch, error messages, homing etc.). The servo drive needs to provide the synchronous AC motor with a 3 phase voltage of varying frequency. To do this, first the rectifier converts the AC supply voltage (constant 50 Hz frequency) into DC, which is stored in the DC bus. From there the inverter can utilise Pulse-width modulation (PWM) and send short bursts of the DC bus voltage to the motor that can approximate an AC with an arbitrary frequency. The control of this power electronics is a relatively complex task that has already been solved internally by the manufacturer. We only need to worry about the performance and stability of the velocity and position controllers, that are arranged in a cascade control configuration (detailed description in section 3.6) and require the current motor shaft position (measured by the motor encoder) as feedback.

The control system devices were installed on the conductive mount plate with a top-hat rail and wired according to the diagram in Figure 3.5. (This diagram is an abstract schematic and does not show the wiring details, which can be found in respective manuals. All signal cables were properly shielded according to the recommendations in the manual.)

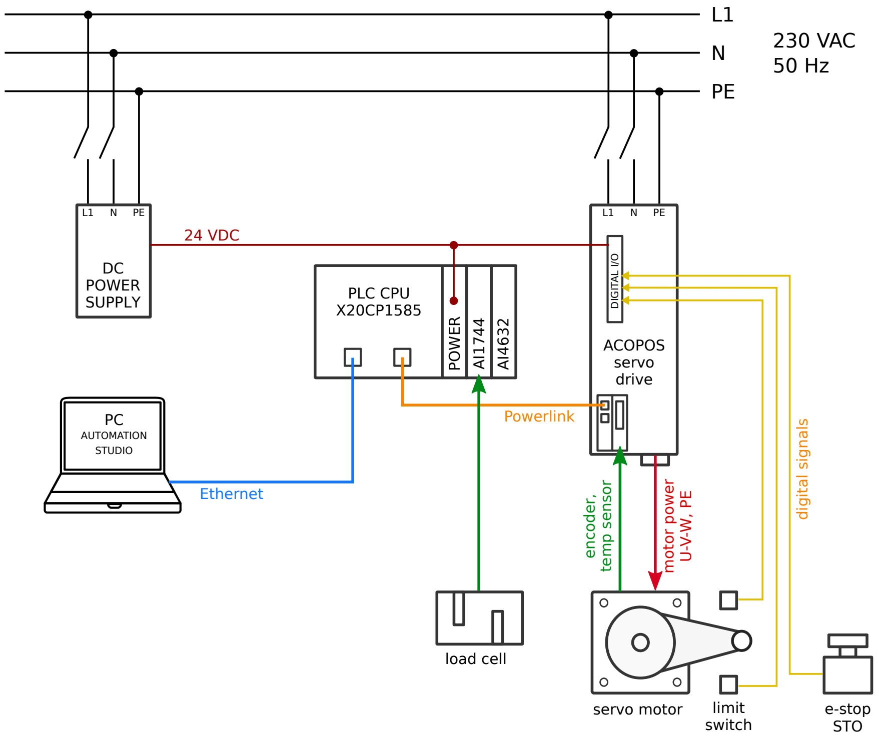
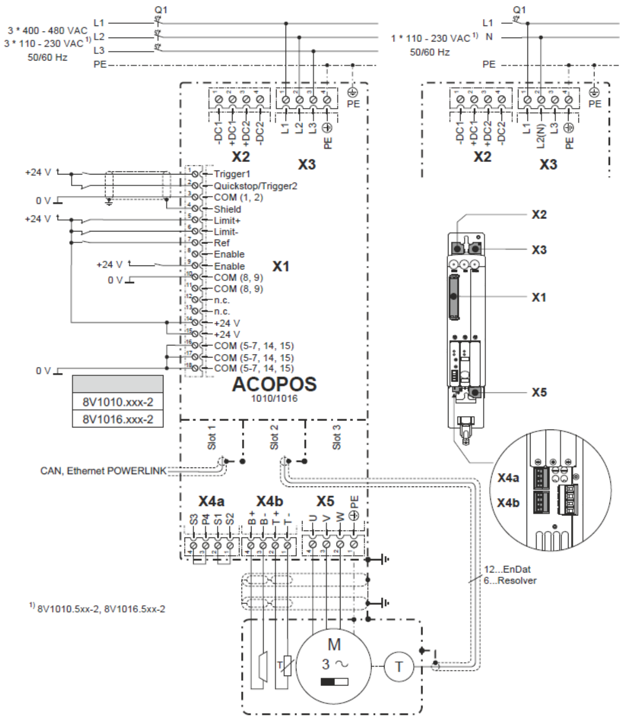


Figure 3.5: Wiring diagram of the CLS test bench control system



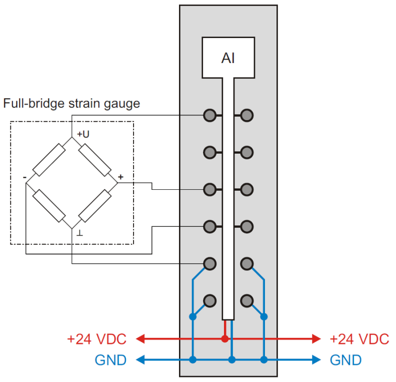


Figure 3.6: Detailed wiring servo drive

[B&R Automation 8V1016.50-2 Datasheet]

Figure 3.7: Detailed wiring load cell

[B&R Automation X20AI1744 Datasheet]

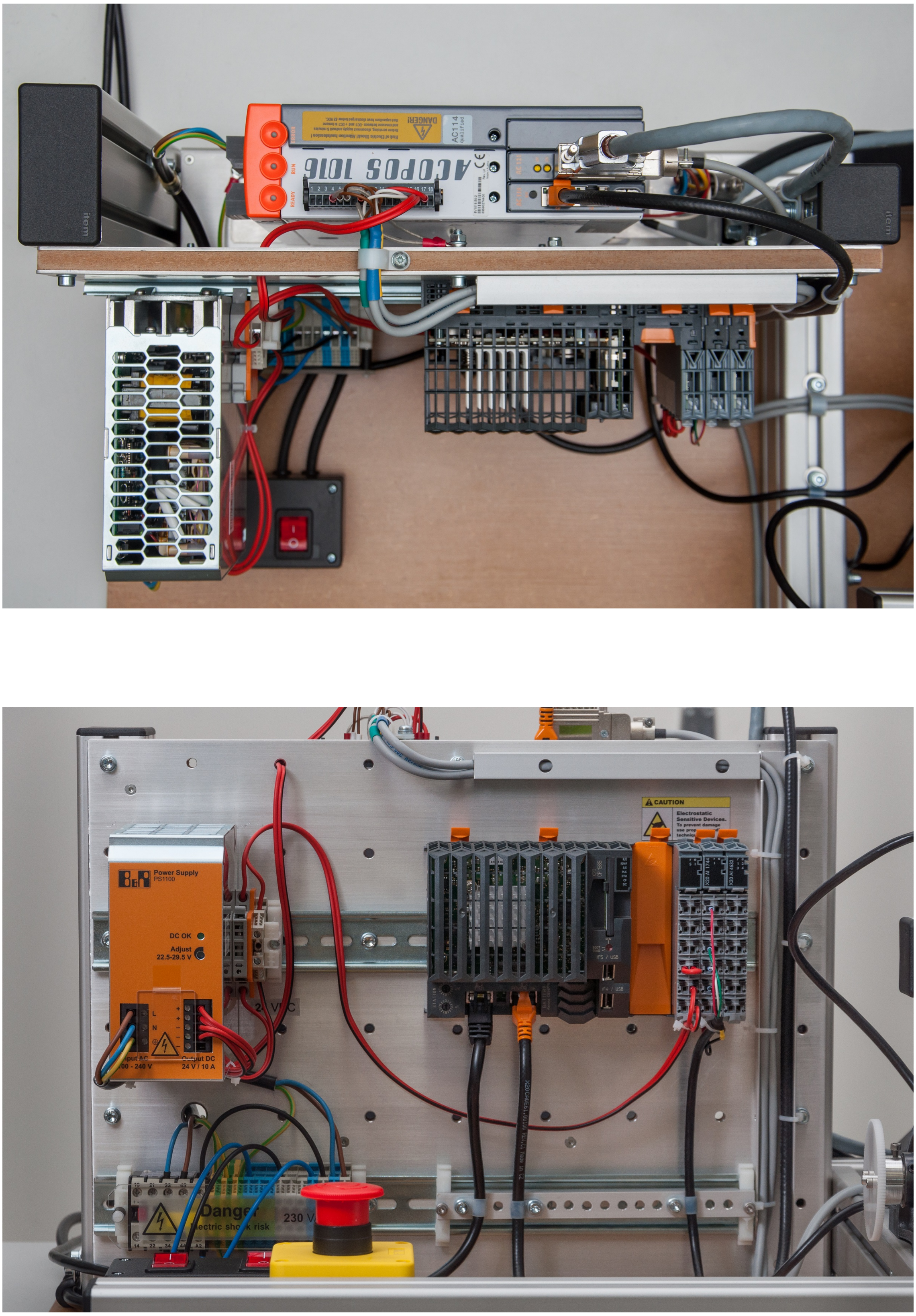


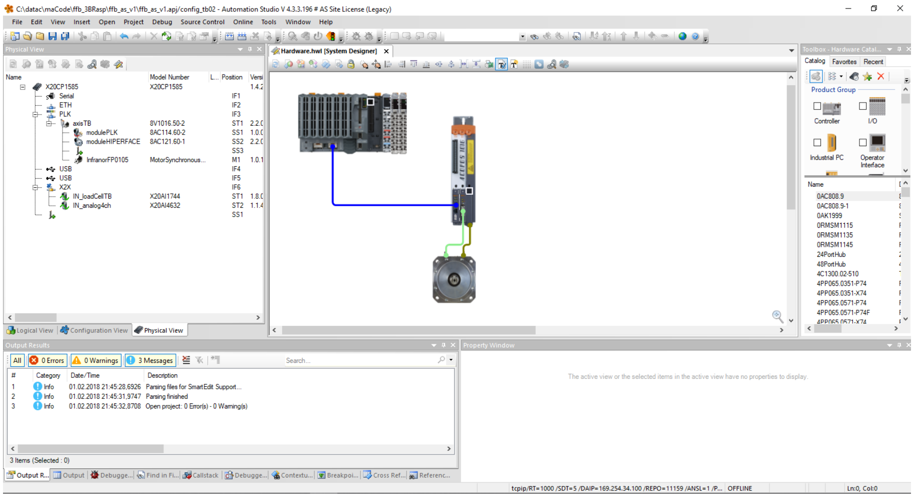
Figure 3.8: Control system of the CLS test bench

3.3 Test bench commissioning

After all devices have been securely mounted and wired, they are ready for their **initial commissioning**. Several initial procedures need to be completed before the first programs can be executed on the hardware.

3.3.1 PLC setup

Every new B&R control system commissioning starts with a new 'project' inside the **Automation Studio**. First, the user has to select the currently used devices and modules and arrange them in a **hardware configuration**. Various device settings can be adjusted at this point to enable the required features of each module. Next, the PLC can be powered on and an **online connection** can be established directly through the Automation Studio if the target hardware is connected to the PC's network via ethernet.



C:\datac\maCode\ffb_3BRasp\ffb_as_v1\ffb_as_v1.apj/config_tb02 - Automation Studio V 4.3.3.196 # AS Site License (Legacy)

Name	Model Number	L...	Position	Versi
X20CP1585	X20CP1585			1.4.2
Serial			IF1	
ETH			IF2	
PLK			IF3	
axisTB	8V1016.50-2		ST1	2.2.(
modulePLK	8AC114.60-2		SS1	1.0.(
moduleHIPERFACE	8AC121.60-1		SS2	2.2.(
			SS3	
InfranorFP0105	MotorSynchronous...		M1	1.0.1
USB			IF4	
USB			IF5	
X2X			IF6	
IN_loadCellTB	X20AI1744		ST1	1.8.(
IN_analog4ch	X20AI4632		ST2	1.1.4
			SS1	

#	Category	Date/Time	Description
1	Info	01.02.2018 21:45:28,6926	Parsing files for SmartEdit Support...
2	Info	01.02.2018 21:45:31,9747	Parsing finished
3	Info	01.02.2018 21:45:32,8708	Open project: 0 Error(s) - 0 Warning(s)

3 Items (Selected : 0)

The active view or the selected items in the active view have no properties to display.

tcpip/RT=1000 /SDT=5 /DAIP=169.254.34.100 /REPO=11159 /ANSL=1 /P... OFFLINE Ln:0, Col:0

Figure 3.9: Hardware configuration in Automation Studio

3.3.2 Servo commissioning

The servo motor's initial setup took some effort because we were coupling a servo motor and a drive of two different manufacturers. The motor's parameter data had to be meticulously checked to ensure compatibility and the encoder cable connector required a specially made adapter. The reason, we went through this effort is that it gave us the advantages of both: the zero-cogging Infranor motor as well as the B&R high-speed Powerlink connection to the PLC.

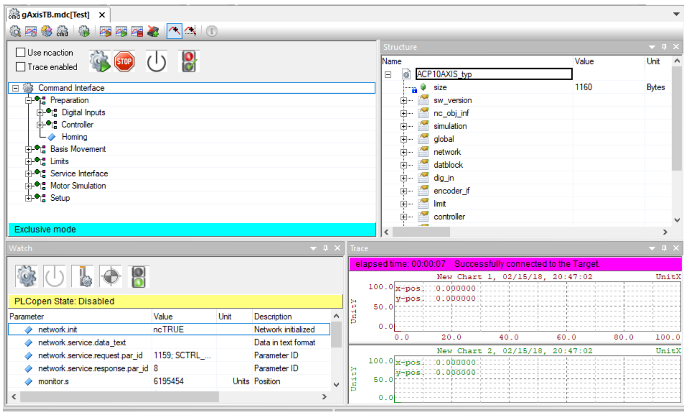
The rest of the setup was a simple and user-friendly experience completed exclusively inside the Automation Studio:

1. **create a new motion axis**

   A new motion axis is created in the AS project, which creates and manages the motion control objects required for seamless communication with the servo drive.

2. **provide motor data**
   Motor's technical data such as torque and voltage constants, number of pole pairs, maximum current, speed and temperature, encoder type and resolution etc. have to be specified, so that the servo drive can appropriately control the motor's current as well as read its position and temperature.

3. **define motion parameters and limits**
   Motion control applications according to the PLCopen standard refer to the axis position in so-called 'PLCopen units'. They can be arbitrarily chosen for each axis and even allow to include transmission ratios. In our case, I decided to assign 3600 [PLCopen units] to represent 1 revolution of the motor shaft. When the PLCopen units are established, we can use them to define software motion limits such as positive and negative position limits, maximum velocity and acceleration, jolt time etc. Additionally, digital inputs on the servo drive are configured to provide the drive with the information of a triggered limit switch or emergency stop.



□ Use ncaction
□ Trace enabled

- Command Interface
  - Preparation
    - Digital Inputs
    - Controller
    - Homing
  - Basis Movement
  - Limits
  - Service Interface
  - Motor Simulation
  - Setup

Name	Value	Unit
ACP10AXIS_typ		
size	1160	Bytes
sw_version		
nc_obj_inf		
simulation		
global		
network		
datblock		
dig_in		
encoder_if		
limit		
controller		

PLCopen State: Disabled

Parameter	Value	Unit	Description
network.init	ncTRUE		Network initialized
network.service.data_text			Data in text format
network.service.request.par_id	1159; SCTRL_...		Parameter ID
network.service.response.par_id	8		Parameter ID
monitor.s	6195454	Units	Position

elapsed time: 00:00:07 Successfully connected to the Target.

New Chart 1, 02/15/18, 20:47:02
x-pos, 0.000000
y-pos, 0.000000

New Chart 2, 02/15/18, 20:47:02
x-pos, 0.000000
y-pos, 0.000000

Figure 3.10: NC Test window in AS

4. **power on**
   With completed motor and motion parameter tables, the drive can be powered on and further operated via the **NC Test** window (part of AS), that allows the user to further configure and test-operate the axis. All the following steps can be performed inside the NC Test environment.

5. **phasing**
   The purpose of phasing a synchronous motor is to determine the commutation offset, which describes how the motor encoder is positioned relative to the stator coils. This is an essential parameter for synchronous motor control, and the advantage of an absolute motor encoder is that this procedure (although automatised) only has to be performed once (and not at each motor startup as this is the case with incremental encoders).

6. **homing**
   Homing procedure determines the reference zero position (0 PLC units) of the motion axis. Various automated methods can be utilised to get a consistent reference of an

absolute position at each servo startup. Absolute encoders again have the advantage of retaining the information about their absolute position and can implement simple offset homing method.

7. **controller tuning**
   Servos receive position or velocity commands, which can be tracked by utilising position and velocity controllers that are implemented on the servo drive. To guarantee stability and optimal performance the controller parameters have to be tuned. (Also see section 3.6)

8. **performing movements**
   After the setup is finished, the servo can perform its first test movements directly from the NC Test window.

3.3.3 Load cell calibration

The load cell signal that is proportional to the applied force will be interpreted by the ADC as a 'count' integer. To convert these units to force measured in Newtons, a calibration of the sensor needs to be performed.

Calibration is done by applying static loads (weights of known mass) to the force sensor and record the ADC's output for each load. The recorded signal should be averaged over a time period to reduce statistical errors due to signal noise. Based on the measured data points, we can create a linear mapping (calibration gain and offset) that converts the incoming ADC count to the corresponding force in [N]:

$$\text{force [N]} = \frac{\text{ADC count} - \text{calibration offset}}{\text{calibration gain}} \tag{3.1}$$

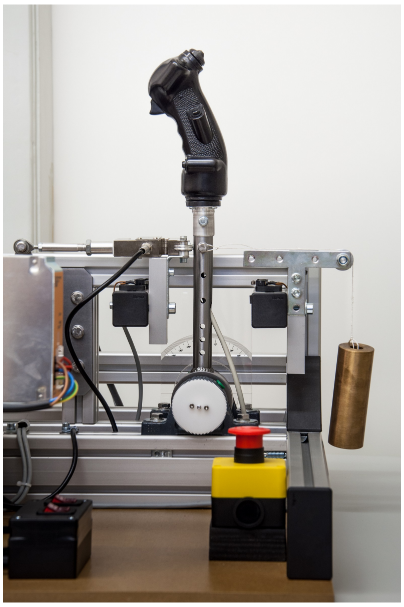


Figure 3.11: Load cell calibration setup

The calibration is performed when the sensor is already mounted in the assembly. The servo crank is locked, and the weight load is applied to the control column in its neutral (vertical) position via a pulley system. However, this method introduces a small offset error in the force measurement. A small amount of static friction that is present in the control column's axis bearings results in a **static hysteresis** effect. This means that the force measured by the load cell will be offset by this static friction, the direction of which will depend on the direction from which the control column arrived at the neutral calibration position. To correct for this effect, we repeat the measurement of the zero-load signal twice more. First, we drive the control column to the neutral position from the positive direction and record the signal, then we repeat the same measurement from the negative direction. The mean value of these two signals is our new corrected calibration offset.

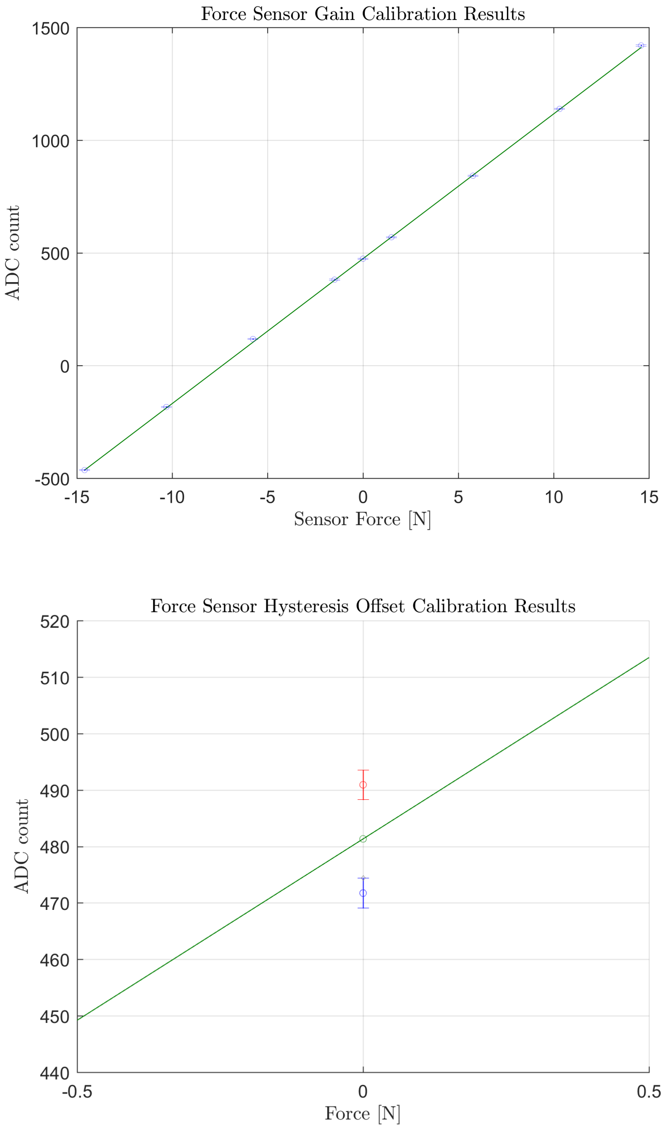


Figure 3.12: Load cell calibration results

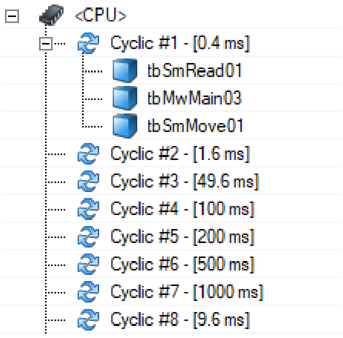
3.3.4 Creating and executing application programs

The last step of the commissioning is to program, configure and execute the application programs on the control hardware. The actual logic of the CLS controllers is a topic for section 3.6, at this point we shall only look at the general organisation layout and steps required for deploying a program on the control hardware.

Our control system consists of two CPU units, connected via Powerlink:

1. the central X20CP1585 PLC CPU
   - reads load cell signal
   - executes routines (outer loop)
   - sends PLCopen motion demands to the servo drive via Powerlink
   - exchanges flight simulation parameter data via ethernet
2. servo drive CPU
   - reads motor encoder signal
   - receives PLCopen motion demands from the PLC via Powerlink
   - executes servo control loops
   - monitors motor status and manages safety functions
   - sends motor encoder position to the PLC via Powerlink

The program for the servo drive has already been automatically created as we configured and commissioned the servo. Programming the central PLC CPU, however, requires some further work. It runs the real-time operating system Automation Runtime, which is designed to cyclically execute programs, referred to as **tasks**. Tasks are classified by their **cycle time** and **priority** into **task classes**, which determine how often and in what order the tasks should be executed. If the CPU is not capable of completing the task within a predetermined time frame, a **cycle time violation** will occur, and the process will be terminated. Real-time behaviour is absolutely essential for critical motion control applications, where a delayed program response is useless or even dangerous.



- <CPU>
  - Cyclic #1 - [0.4 ms]
    - tbSmRead01
    - tbMwMain03
    - tbSmMove01
  - Cyclic #2 - [1.6 ms]
  - Cyclic #3 - [49.6 ms]
  - Cyclic #4 - [100 ms]
  - Cyclic #5 - [200 ms]
  - Cyclic #6 - [500 ms]
  - Cyclic #7 - [1000 ms]
  - Cyclic #8 - [9.6 ms]

Figure 3.13: Task classes in AS

Tasks we created, have access (read or write) to the declared **variables**, which can be assigned to different available I/O channels (provided by the currently used modules) in the I/O mapping table or shared with other devices via Powerlink or ethernet. According to the accessibility of variables between different tasks, AS differentiates between global and local variables.

In our case, the PLC CPU executes the routine associated with the outer loop of our CLS, which I have divided into 3 subtasks:

1. **read**
   - demands the current encoder position and velocity data from the servo drive
   - programmed as a function block diagram (FBD) using the PLCopen Motion Control library blocks
2. **main**
   - executes the CLS open-loop routine (see section 3.6) based on current sensor data
   - automatically generated code from a Matlab Simulink model (featuring special B&R interface blocks)
3. **move**
   - sends motion commands determined in the previous task to the servo drive
   - programmed as a function block diagram (FBD) using the PLCopen Motion Control library blocks

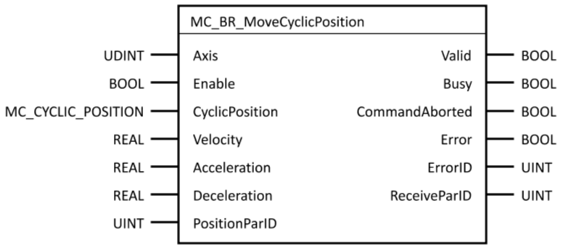


Figure 3.14: Motion control library function block 'MoveCyclicPosition'

(used in the FBD, conform with the PLCopen standard)

[Automation Help 4.3]

These tasks are all executed within 400 μs (at 2.5 kHz), as necessary for the CLS outer loop (also see section 3.6). In the future, as the CLS control system becomes a part of the flight simulator network, an additional task will be added to the PLC for TCP/IP communication in which the necessary parameters will be exchanged with the rest of the flight simulation. This task will, however, be assigned with a larger cycle time and lower priority, since the flight simulation software runs at considerably lower iteration rates (around 100 Hz). Furthermore, as more flight control axes are upgraded to active control loading, the PLC will have to manage such set of tasks for each axis in parallel.

3.4 CLS plant modelling

During the control systems design, we will require an understanding of different aspects of the CLS plant's dynamics. In this section, we model and derive the physical relationships that will be useful further on.

3.4.1 Simscape model

In the early phase of the design process, a simplified physical model of the CLS plant was created in Simulink Simscape. It allowed running a few proof of concept simulations before the real hardware was available.

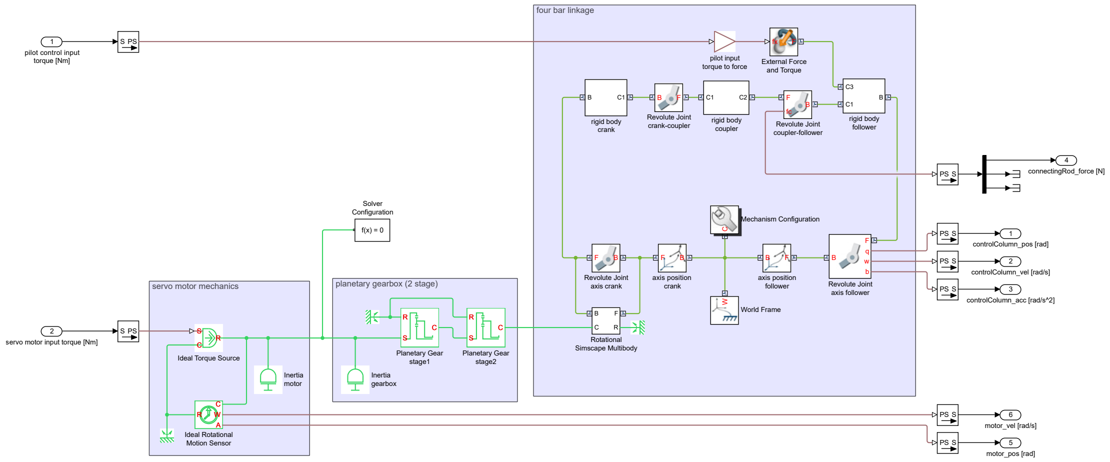


Figure 3.15: Simscape model of the CLS plant

3.4.2 Kinematic analysis of the linkage

We need the ability to transform between the motion of the motor's axis and the control column, which are interconnected with each other via a **linkage system** that consists of a gearbox and a four-bar linkage:

- The **planetary gearbox** provides a constant gearing ratio of $i_{gbx} = 25$ and gives a *linear* kinematic correlation between its input and output (provided that we neglect the small backlash).
- The **four-bar linkage** on the other hand, has a *non-linear* kinematic correlation between the crank and the follower (=control column). We will analytically derive these non-linear correlations and see for what conditions we could use a *linearised* approximation.

For the kinematic analysis, we will assume the entire linkage to be rigid and backlash-free.

Non-linear kinematic analysis of the four-bar linkage

We wish to derive the exact kinematic correlations of the four-bar linkage shown in Figure 3.16, so that for a given motion state of the crank $(\varphi_c, \dot{\varphi}_c, \ddot{\varphi}_c)$ we can directly calculate the motion state of the follower $(\varphi_f, \dot{\varphi}_f, \ddot{\varphi}_f)$ and vice versa.

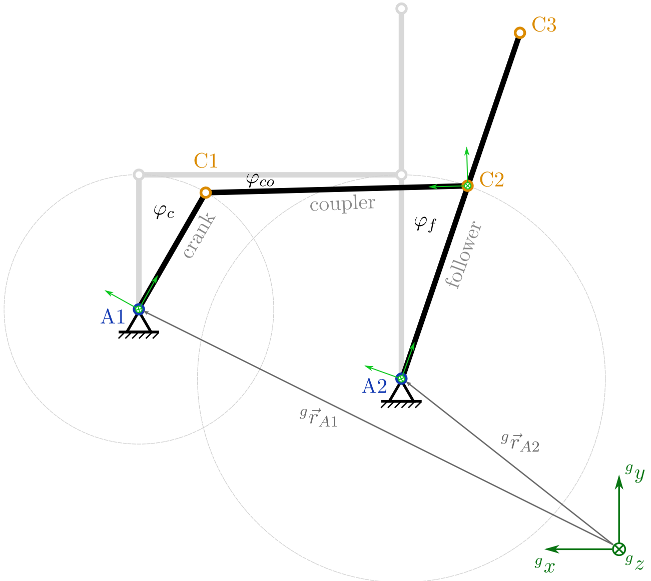


Figure 3.16: Kinematics of the four-bar linkage

The **crank** and the **follower** rotate around globally fixed axes A1 and A2 respectively and are interconnected via a **coupler** rod with revolute joints at connection points C1 and C2. The follower in our case is the control column and point C3 marks the position of the pilot's hand.

Suppose we are given the fixed positions of axes A1 and A2 in global coordinates: ${}^g\vec{r}_{A1}$, ${}^g\vec{r}_{A2}$ as well as the positions of the connection points C1 and C2 in the local, body-fixed coordinate systems for each body: ${}^c\vec{r}_{c1}$, ${}^f\vec{r}_{c2}$, ${}^f\vec{r}_{c3}$, ${}^{co}\vec{r}_{c1}$, ${}^{co}\vec{r}_{c2}$.
We can write the global positions of points C1 and C2 in terms of their rotation around axes A1 and A2:

$$ {}^g\vec{r}_{c1} = {}^g\vec{r}_{A1} + R_z(\varphi_c)\, {}^c\vec{r}_{c1} \tag{3.2} $$

$$ {}^g\vec{r}_{c2} = {}^g\vec{r}_{A2} + R_z(\varphi_f)\, {}^f\vec{r}_{c2} \tag{3.3} $$

where R_z is the rotation matrix around z-axis:

$$ R_z(\varphi) = \begin{bmatrix} \cos\varphi & -\sin\varphi & 0 \\ \sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix} $$

The distance between points C1 and C2 remains constant, due to the rigid coupler that connects them, which gives the **constraint equation**:

$$({}^{g}\vec{r}_{c1} \;-\; {}^{g}\vec{r}_{c2}) \cdot ({}^{g}\vec{r}_{c1} \;-\; {}^{g}\vec{r}_{c2}) \;\overset{!}{=}\; |\,{}^{co}\vec{r}_{c1} \;-\; {}^{co}\vec{r}_{c2}\,|^2 \;=:\; h^2 \tag{3.4}$$

Inserting 3.2 and 3.3 in the constraint equation will result in an expression in the form of:

$$A(\varphi_c)\,\cos\varphi_f \;+\; B(\varphi_c)\,\sin\varphi_f \;+\; C(\varphi_c) \;=\; 0 \tag{3.5}$$

According to [10] this equation can be solved for φ_f as follows: introduce an angle γ, so that:

$$\cos\gamma = \frac{A}{\sqrt{A^2+B^2}} \quad \text{and} \quad \sin\gamma = \frac{B}{\sqrt{A^2+B^2}} \quad \Longrightarrow \quad \gamma = \text{atan2}(B, A)$$

which inserted in equation 3.5 gives:

$$\cos\gamma\,\cos\varphi_f \;+\; \sin\gamma\,\sin\varphi_f \;+\; \frac{C}{\sqrt{A^2+B^2}} \;=\; 0$$

Formula for cosine of angle difference yields:

$$\cos(\varphi_f \;-\; \gamma) \;+\; \frac{C}{\sqrt{A^2+B^2}} \;=\; 0$$

$$\Longrightarrow \quad \boxed{\varphi_f \;=\; \gamma \pm \text{acos}\left(\frac{-C}{\sqrt{A^2+B^2}}\right)} \tag{3.6}$$

The equation gives two real solutions for φ_f (at a given φ_c) when $-1 \leq \frac{-C}{\sqrt{A^2+B^2}} \leq 1$.

After obtaining the solution for the angle position, we can further differentiate the function to obtain expressions for velocity and acceleration correlations:

$$\begin{aligned}
\text{position:} \qquad & \varphi_f = \underline{f_1(\varphi_c)} \qquad \textit{(obtained in Equation 3.6)} \\
\text{velocity:} \qquad & \dot{\varphi}_f = \frac{d\varphi_f}{dt} = \frac{d\varphi_f}{d\varphi_c}\,\frac{d\varphi_c}{dt} = \underline{f_2(\varphi_c)\,\dot{\varphi}_c} = f_3(\varphi_c, \dot{\varphi}_c) \\
\text{acceleration:} \qquad & \ddot{\varphi}_f = \frac{d\dot{\varphi}_f}{dt} = \frac{d}{dt}\left(\frac{d\varphi_f}{d\varphi_c}\right)\dot{\varphi}_c + \frac{d\varphi_f}{d\varphi_c}\,\ddot{\varphi}_c = \\
& = \frac{d^2\varphi_f}{d\varphi_c^2}\,\dot{\varphi}_c^2 + \frac{d\varphi_f}{d\varphi_c}\,\ddot{\varphi}_c = \underline{f_4(\varphi_c)\,\dot{\varphi}_c^2 + f_2(\varphi_c)\,\ddot{\varphi}_c} = f_5(\varphi_c, \dot{\varphi}_c, \ddot{\varphi}_c)
\end{aligned}$$

The analytic operations were performed in Matlab's *Symbolic Math Toolbox*, where the exact solution for the given defined motion range was found. However, the elaborate symbolic expressions are not practical to implement in a program for a PLC, since the large number of trigonometric operations would unnecessarily load the CPU. The exact solutions of functions f_1, f_2, f_3 and f_4 were therefore approximated with polynomials, which can be more optimally implemented in a Simulink model. Figure 3.17 shows plots of the analytic solutions overlaid by the fitted polynomial approximations in red.

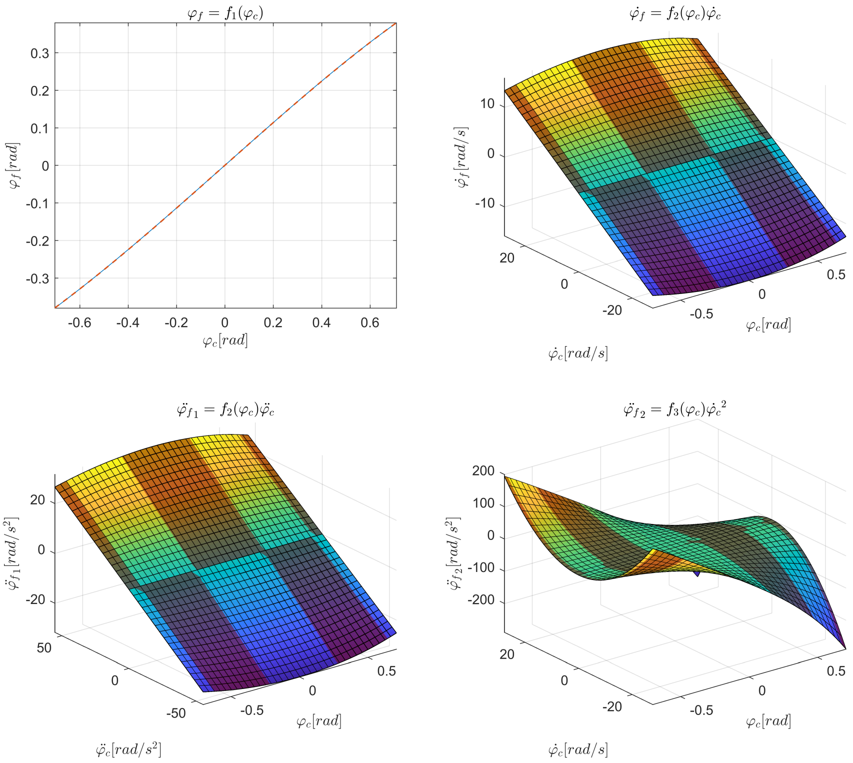


Figure 3.17: Results of the four-bar linkage analysis

Now the correlation between φ_f and φ_c is known, we can easily determine the angle of the coupler φ_{co} as well. Additionally, I calculated the linkage ratio $i_{fourbar} = T_f/T_c$ as a function of position. I repeated the exact same process to obtain a solution to the reversed problem and find φ_c, $\dot{\varphi}_c$ and $\ddot{\varphi}_c$ for given φ_f, $\dot{\varphi}_f$ and $\ddot{\varphi}_f$.

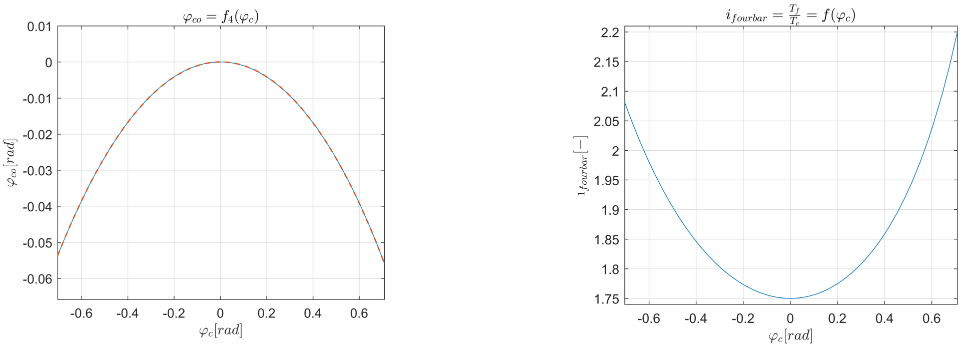


Figure 3.18: Results of the four-bar linkage analysis

Linearised kinematics of the four-bar linkage

From the plots on the previous page, it is clear that in certain cases our chosen geometry allows the four-bar linkage kinematics to be assumed as linear (around the zero position) with a constant linkage ratio of $R_{fb} = 1.75$. This will especially be utilised during the control system analysis.

Transformation between motor axis and control column movement

To transform the motion state of the control column in [rad] to the motion state of the motor axis in [PLCopen units], we apply the chain of transformations shown in Table 3.5.

Linkage part	**Transformation**
four-bar linkage control column [rad] → servo crank [rad]	non-linear polynomials or linear factor $R = 1.75$
planetary gearbox servo crank [rad] → motor axis [rad]	$R = 25$
motor axis [rad] → motor axis [rev]	$R = 1/2\pi$
motor axis [rev] → motor axis [PLCopen units]	$R = 3600$

Table 3.5: Kinematic tranformation chain of the CLS linkage

3.4.3 Kinetic analysis of the linkage

Figure 3.19 shows a free body diagram of the four-bar linkage with acting forces and torques, that will be used later on. The yellow arrows represent friction torques in the revolute joints.

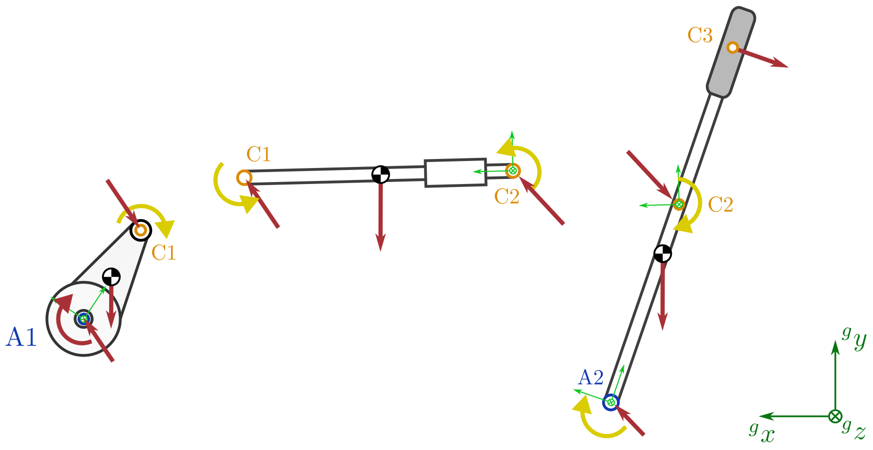


Figure 3.19: Free body diagram of the four-bar linkage

3.4.4 Added limb inertia

It is important to realise, that in addition to the mechanical structure also the pilot's body can directly contribute to the CLS plant dynamics. Let us briefly examine the **biomechanics of handling a control column**, which is very roughly modelled in Figure 3.20. The pilot will hold onto the control column grip with his hands and mainly use the muscles in his upper arm to perform the necessary movement. Meanwhile, the majority of the mass of his forearm will have the chance to rest on the control column and move with it, which consequently introduces additional inertia to the flight controls (or in our case the CLS plant).

As noted in Figure 3.20, the effect of this added limb inertia will vary according to how the pilot holds the grip of the flight controls. According to our simplified model, a **loose grip** will act like a soft spring between the control column and the limb mass, thus reducing the energy transfer between the two bodies. However, if the pilot is **tightly squeezing** the grip, this will result in a rigid connection, equivalent to a very stiff spring and the inertia of the resting forearm will become an integrated part of the CLS plant that the actuator has to move together with the rest of the mechanism.

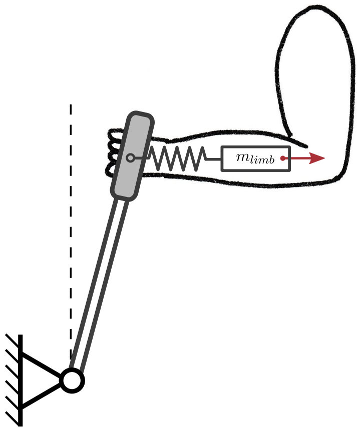


Figure 3.20: Added limb inertia to the control column

CLS controllers, therefore, have to be designed robust enough to account for these very inconsistent inertia fluctuations. How they can affect the performance and stability of the control loops shall be discussed in section 3.6 and demonstrated in chapter 4.

3.5 Flight control system modelling

The FCS model is the core of the CLS controller and dictates which forces will the pilot feel on the control column. The complexity of the flight controls model varies according to the flight simulator's purpose and fidelity requirements. The most advanced simulators will feature a very detailed and realistic FCS model, which includes the functionality of the unique design features of a specific aircraft type. At TU Graz, however, the research flight simulation platform is kept universal, where most simulation models represent a generic aircraft of its category.

3.5.1 Two-body model

The two-body model simulates the flight controls FWD and AFT systems as separate bodies, connected via a flexible linkage (cable or rod system), which results in two degrees of freedom: deflection of the control column φ (FWD system) and deflection of the control surface δ (AFT system).

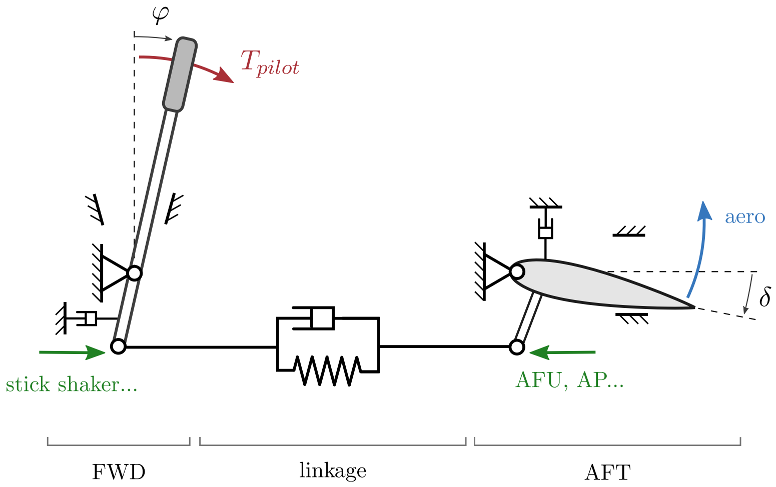


Figure 3.21: Two body FCS model

FWD system:

- pilot torque T_{pilot}
- inertia
- damping, friction
- travel limit
- stick shaker

Miscellaneous:

- stick pusher
- unbalance force

Linkage:

- stiffness
- damping
- backlash

AFT system:

- aerodynamic forces on control surface (hinge moment H, aerodyn. damping, gusts)
- artificial feel unit (AFU) (spring-feel unit, q-feel unit)
- inertia
- damping, friction
- travel limit
- autopilot (AP) actuators
- centering unit

The list above contains the variety of (passive or active) components, parameters and inputs that can contribute to model's whole dynamics. Which effects are implemented in the model depends on which features are present in the real aircraft's FCS.

Notice, that in a **mechanical** FCS the aero forces will be directly transferred to the control column, while in a **hydro-mechanical** FCS, the power control unit will intercept them, and the pilot will only feel the loads produced by the artificial feel unit.

The **travel limits** are usually designed in such a way that first the travel limit of the AFT system is reached, after which the FWD system can still travel a small amount before reaching its hard stop. If the pilot continues to apply force after the AFT travel limit is reached, he will be able to notice the stretching of the cable linkage.

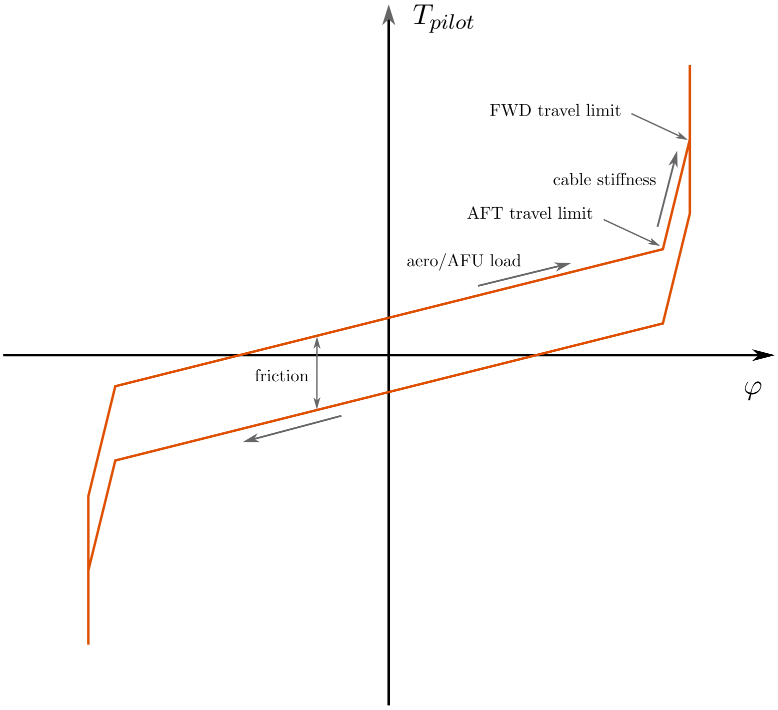


Figure 3.22: Typical static torque-displacement diagram for the control column

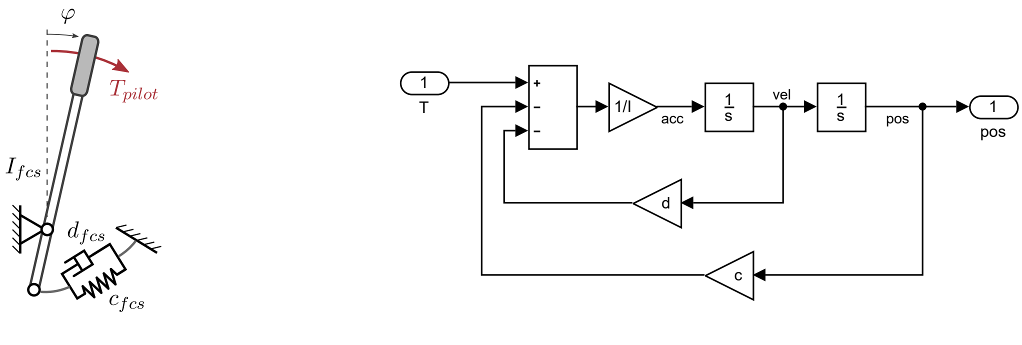
3.5.2 Single body model

We can further simplify our model because, for most purposes, a single body FCS model with one degree of freedom will be s sufficiently accurate representation of real flight controls. We start off by modelling the flight controls as a linear second order system, which is most suitable for control system analysis. In continuation, this linear model shall be further expanded with additional functionality features (e.g. trimming, hard-stops, etc.), that often introduce discontinuities and non-linearities, but are necessary for practical implementation in a flight simulator and can be added according to the simulation requirements. Note, that in a single body model, the loads produced by the elements are all referenced to the control column's axis.

Linear mass-spring-damper system

We start off by deriving the motion equation and transfer function of the rotational mass-spring-damper system, which resembles the basic dynamics of the aircraft's FCS at the control column's reference frame:

- **Inertia** I_{fcs}: represents the total inertia of the FCS mechanics (control column, internal linkages, control surfaces...).
- Rotational **spring** with stiffness c_{fcs}: represents the behaviour of the aerodynamic load on the control surfaces due to their deflection δ. For reversible mechanical flight controls with a constant linkage ratio $R = \delta/\varphi$ between the control column and control surface, equations 2.1 and 2.2 give: $c_{fcs} = \frac{RH}{\varphi} = R^2\frac{H}{\delta} = R^2\ \frac{1}{2}\rho v^2\ S\ \bar{c}\ \frac{\partial C_H}{\partial \delta}$. In the case of irreversible flight controls, an analogous dynamic behaviour is ensured by the AFU.
- Rotational **damper** with damping coefficient d_{fcs}: represents the aerodynamic damping of the control surface as well as mechanical damping of the internal mechanism.
- Applied **torque** T: mainly represents the torque, exerted by the pilot T_{pilot}, but can also include inputs from autopilot actuators etc.



(a) abstract model of the FCS

(b) simulink block diagram

Figure 3.23: Mass-spring-damper model

Based on the conservation of angular momentum, we can write:

$$I_{fcs}\,\ddot{\varphi}\;+\;d_{fcs}\,\dot{\varphi}\;+\;c_{fcs}\,\varphi\;=\;T$$

Applying the Laplace transform gives:

$$I_{fcs}\,\varphi\,s^2\;+\;d_{fcs}\,\varphi\,s\;+\;c_{fcs}\,\varphi\;=\;T$$

Rearranging, we can write the FCS transfer function $F(s)$, outputting control column's angle position φ:

$$F(s) = \frac{\varphi(s)}{T(s)} \;=\; \frac{1}{I_{fcs}\,s^2 + d_{fcs}\,s + c_{fcs}}$$

$$\implies \quad \varphi(s) \;=\; \frac{T(s)}{I_{fcs}\,s^2 + d_{fcs}\,s + c_{fcs}} \tag{3.7}$$

As expected, we end up with the well known second-order dynamic system. While inertia I_{fcs} and damping d_{fcs} mostly remain constant, the stiffness c_{fcs} will change proportionally with the dynamic pressure $\frac{1}{2}\rho v^2$ during simulation. The bode plot below shows, how changing the stiffness affects the system's frequency response.

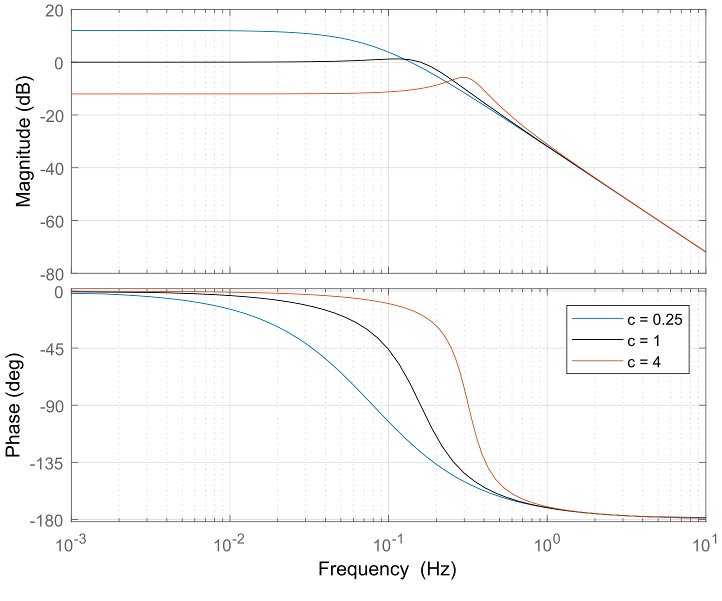


Figure 3.24: Second order system bode plot

($I_{fcs} = 1$, $d_{fcs} = 1$, c_{fcs} varies)

For a velocity-loop or force-loop CLS architecture, the output of the FCS model has to be rearranged to: [5]

$$\varphi\,s \;=\; \frac{T\;-\;c_{fcs}\,\varphi}{I_{fcs}\,s\;+\;d_{fcs}} \tag{3.8}$$

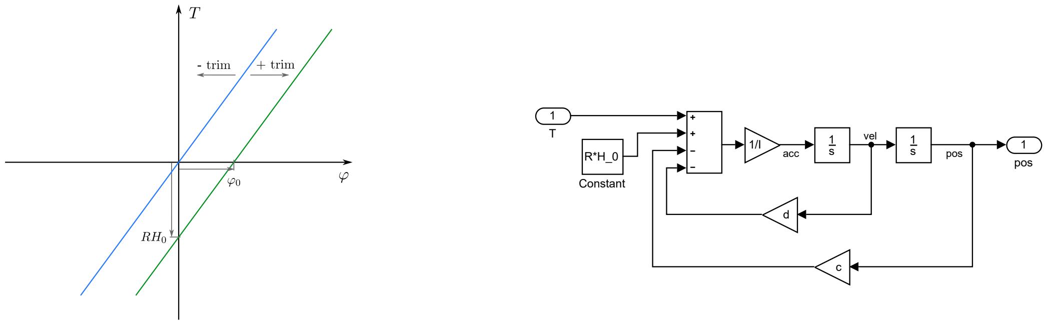
$$T \;=\; I\,\varphi\,s^2\;+\;d\,\varphi\,s\;+\;c\,\varphi \tag{3.9}$$

Neutral position offset

Recalling the Equation 2.2, it shows that the **trim tab deflection** δ_t, as well as the **angle of attack** α (or sideslip angle β for rudder control) will produce additional components of the hinge moment H_0:

$$H_0 \;=\; \tfrac{1}{2}\rho v^2\; S\,\bar{c}\;\left[\frac{\partial C_H}{\partial \delta_t}\,\delta_t \;+\; \frac{\partial C_H}{\partial \alpha}\,\alpha \;+\; C_{H0}\right]$$

This will effectively offset the control surface zero-load deflection δ_0 and consequently the neutral control column position φ_0 as shown in the figure below.



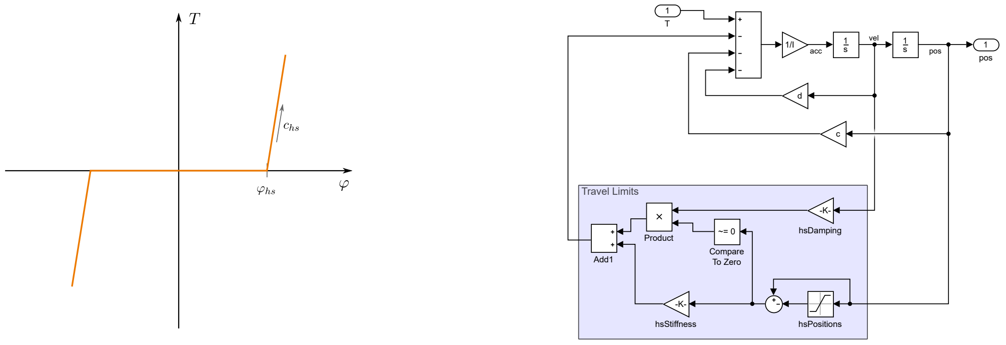
(a) static torque-displacement characteristic

(b) simulink block diagram

Figure 3.25: Control column neutral position offset

Travel limits

The CLS also has to be able to simulate virtual hard stops in the system to limit the range of control column's movement. This is done by applying an additional (large) stiffness c_{hs} and damping d_{hs} when a defined position of the hard stop φ_{hs} is reached, which introduces a discontinuity in the system. Virtual hard stops can feel very realistic to the user, provided that the linkage connecting the control column to the CLS actuator is stiff enough. If we wish to replicate the behaviour shown in Figure 3.22, we can include two travel limits in series. The first on with c_{hs} of the cable stiffness and a second stiffer one to represent the FWD hard stop.



(a) static torque-displacement characteristic

(b) simulink block diagram

Figure 3.26: Implementation of a positive and negative travel limit

Additional input torques

The system's input torque T is an algebraic sum consisting of multiple sources:

- First and foremost, the torque exerted on the control column by the **pilot** himself/herself T_{pilot}
- The **autopilot** can take over the actuation by applying the torque T_{ap} on the system.
- The **stall warning stick shaker** can also exert an additional periodic torque input T_{sw} when triggered. Its rotating unbalanced mass will produce a sinusoidal excitation with a predefined frequency and amplitude.
- Other **disturbances** that are able to further excite the system can be summed up in T_{dis}. (Such as turbulence wind gusts, accelerations, vibrations etc.)

Note that this general, simplified model behaves as if all torques take effect directly at the control column's axis. This simplification needs to be kept in mind when generating the simulation signals for the autopilot, stick shaker or disturbances.

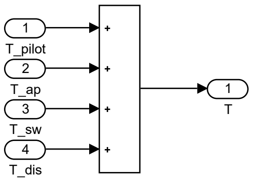


Figure 3.27: System inputs

Coulomb friction

Especially engine thrust control levers will often contain larger amounts of dry coulomb friction, which can be easily added to the Simulink model. It will, however be left out of our control systems analysis due to its non-linear behaviour.

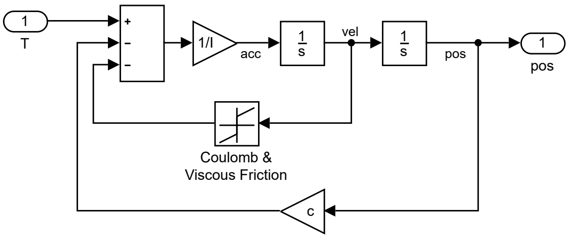
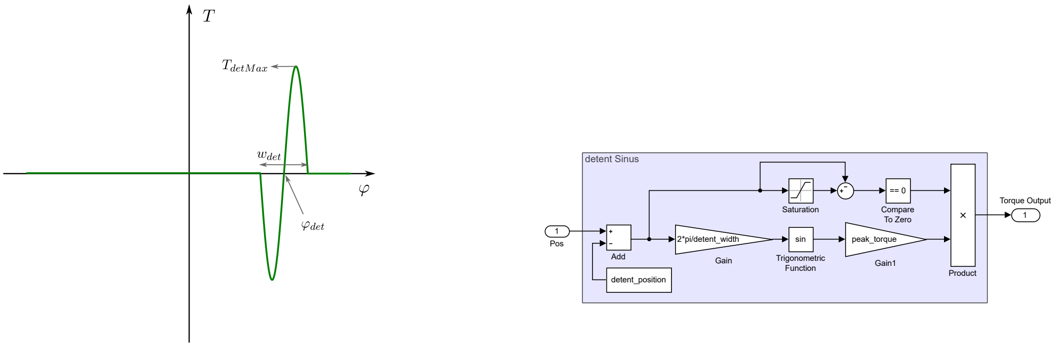


Figure 3.28: Mass-spring-damper model with coulomb and viscous friction

Detents

Most engine control levers and secondary flight controls will feature detents, marking specific important lever positions with a haptic feedback. Mechanically they are usually constructed as a spring-loaded ball entering a conical notch, however, in a simulator they can be included in the FCS model and the CLS is then able to virtually simulate them.

They are usually modelled as antisymmetric force/torque curves in the direct proximity of the detent location. An example can be seen in Figure 3.29, where I modelled the torque curve as a sine curve section with defined detent location φ_{det}, width w_{det} and peak torque T_{detMax}. Alternative curve shapes can be developed to more accurately model the internal detent geometry.



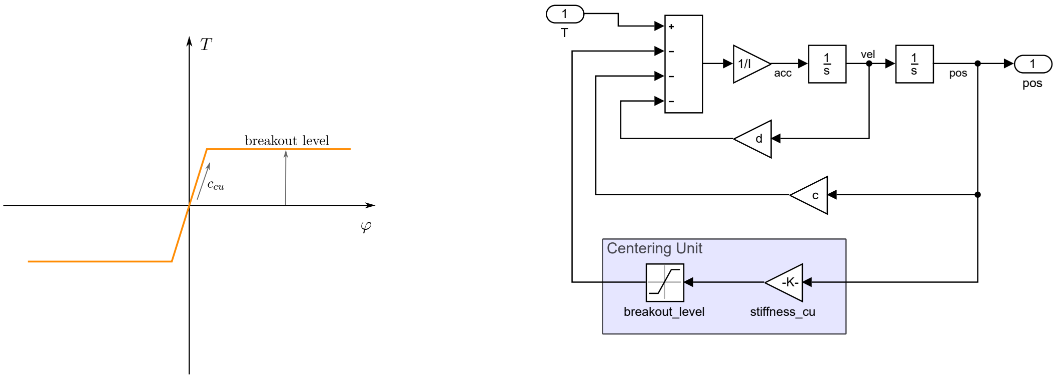
(a) static torque-displacement characteristic

(b) simulink block diagram

Figure 3.29: Implementation of a sinus-shaped rotational detent

Centering unit

The centering unit, when part of the FCS, provides a breakout force and can be very simply modelled as a spring with a saturation at the breakout level.



(a) static torque-displacement characteristic

(b) simulink block diagram

Figure 3.30: Implementation of a centering unit

Unbalance force

Mechanically unbalanced control surfaces or an unbalanced mass in the internal linkage will result in a small pull on the control column, that is constant or position dependent. The unbalance effect, however, only becomes apparent when the aircraft is grounded, and no airflow is present over the control surfaces ($c_{fcs} = 0$), causing the control column's tendency to fall towards one of its travel limits.

Parameter exchange with the host computer

Although the test bench is not yet connected to the rest of the flight simulation, we should consider which parameters have to be exchanged between the simulation software (on host computer) and the FCS model (on CLS controller). This communication runs at much slower cyclic rates compared to the CLS control loops, therefore only parameters that are not critical for the CLS real-time performance are exchanged.

FCS model receives from the simulation software:

- the aerodynamic hinge moment gradient (c_{fcs}) and offset (RH_0)
- autopilot commands (engage/disengage, reference signal)
- diverse trigger signals (e.g. stick shaker, stick pusher, malfunctions simulation)

while the FCS simulation returns:

- the control surface deflection δ, which is a required input for the flight simulation software.

3.6 Control system strategy

In subsection 2.5.3 a fundamental overview of the basic CLS control strategies was already presented. Now, we shall focus on the details of the control system architecture implemented on our CLS test bench. The goal of this thesis was to explore the **admittance type** control loading. The position and the velocity loop variants were both successfully implemented and tried out, however, we focus our further analysis mainly on the **position loop** architecture, shown below.

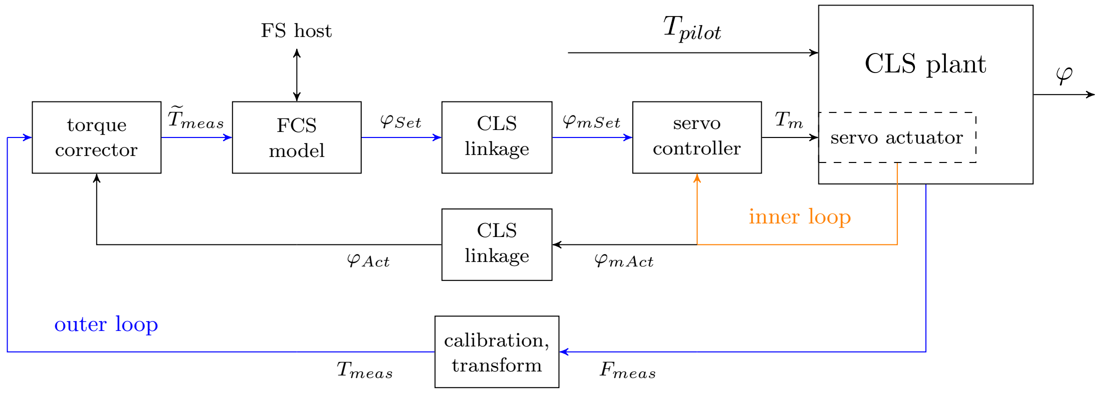


Figure 3.31: CLS control architecture implemented on the test bench

The CLS control system consists of two main parts:

- the **inner loop** and
- the **outer loop**

3.6.1 Inner loop

The inner loop has the task of controlling the servo motor's motion after receiving a cyclic command produced by the outer loop. In the case of the position loop architecture, this command will be a demanded angular position of the motor axis in [PLCopen units].

In our case, the inner loop is executed directly on the CPU of the ACOPOS servo drive, which already has a motion controller implemented. This pre-installed servo controller uses a **cascaded control concept**, which consists of three nested control loops for current/torque, velocity and position control of the servo. In cascade control, the manipulated variable of a higher-level controller becomes the set point for a lower-level controller. [3]

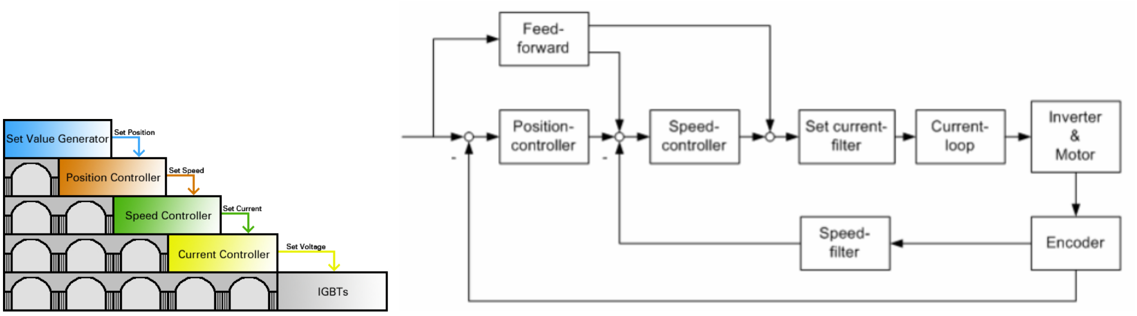


Figure 3.32: Cascaded structure of the B&R ACOPOS servo controller [3]

- **Current controller** is the innermost control loop that runs with the fastest cycle time of just 50 μs. For an AC synchronous motor the produced torque is proportional to its current; therefore this loop is essentially controlling the torque output of the motor. The inner structure and parameters of the controller are automatically determined based on the configured motor data so that the appropriate PWM signal is generated by the IGBTs and sent to the stator coils.

- **Velocity controller** is the next loop in the cascade, which runs with a cycle time of 200 μs. Its internal structure is a basic PI controller with an anti-windup measure. The speed filter differentiates the encoder's position signal to determine the actual velocity of the motor axis, which is compared to the setpoint given by the position controller. Because the speed controller runs at a faster iteration rate in comparison to the position controller, an interpolated velocity set value is used every other iteration.

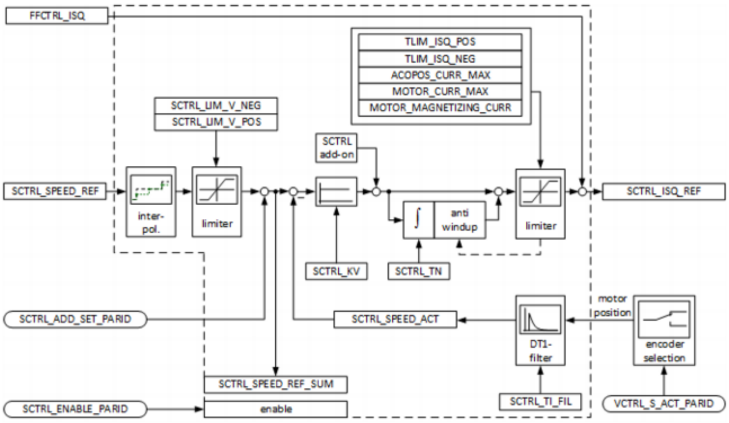


Figure 3.33: Internal structure of the velocity controller

- **Position controller** is the outermost loop of the cascade, which repeats its cycle every 400 μs. Its internal structure is also a PI controller with an anti-windup and is very similar to the velocity controller. The set position value comes from the set value generator, which in our case is cyclically received from the PLC via Powerlink.

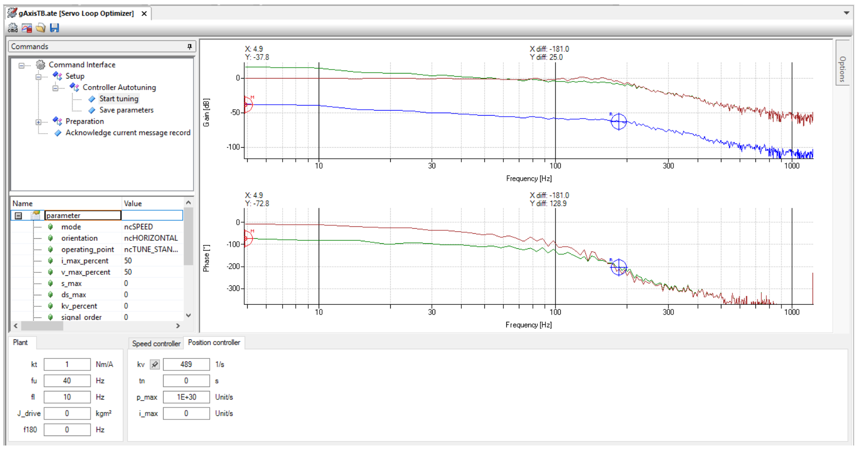
- The servo controller can also implement **feed-forward** pathways to further optimise the servo's performance. The feed-forward signal can account for example for system's inertia or a known load by providing this information directly to the current controller. We could implement this to our advantage because the torque exerted by the pilot on the control column (T_{pilot}) essentially presents a significant disturbance load for the inner loop controller. Since we already measure this torque with the installed load cell, we could easily compensate for it in the feed-forward signal.

Note, that when the servo controller operates in **position mode**, all cascaded control loops will be active, however, if **velocity mode** is selected, the outermost position loop will be deactivated, and for **torque mode** only the current controller will remain running.

Servo controller tuning

Before putting the servo motor into operation, the controller parameters need to be tuned to ensure stable and optimal control performance of the inner loop. In a cascaded control structure, the individual controllers have to be tuned by starting from the innermost loop outwards. The current controller is already tuned automatically according to the provided motor data, while the velocity and position controller require more attention.

For this purpose, B&R Automation Studio includes the tool **Servo Loop Optimizer**, which allows the user to set the control parameters based on frequency responses. The auto-tuning function provides a starting point, which can be further fine-tuned manually. Observing the open-loop frequency response, we have to set proportional and integral gains in such a way, that certain gain and phase margins remain reserved. We can increase the proportional gain until the loop becomes unstable and then reduce it back to reserve the desired stability margin. An integral gain can be added to ensure a zero steady-state error. The results of the tuning procedure can be checked by observing the system's step response. Note, that if the inner loop controller performance is pushed towards stability limits, this could also cause stability problems later on in the outer loop. (See subsubsection 3.6.2.)



Name	Value
parameter	
mode	ncSPEED
orientation	ncHORIZONTAL
operating_point	ncTUNE_STAN...
i_max_percent	50
v_max_percent	50
s_max	0
ds_max	0
kv_percent	0
signal order	0

Plant

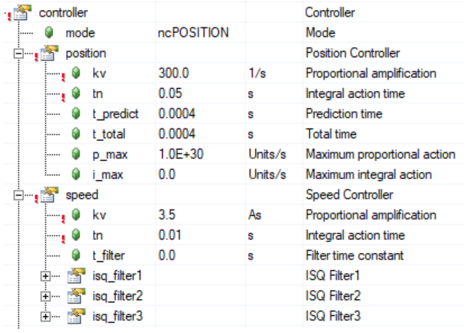
kt	1	Nm/A
fu	40	Hz
fl	10	Hz
J_drive	0	kgm²
f180	0	Hz

Position controller

kv	489	1/s
tn	0	s
p_max	1E+30	Unit/s
i_max	0	Unit/s

Figure 3.34: Position controller tuning in the Servo Loop Optimizer

Figure 3.35 shows an example of the tuned servo controller parameters and the corresponding position command ’step’ response. (Due to the software motion limits and jolt time setting, the set value generator does not actually provide a perfect step function.)



controller			Controller
mode	ncPOSITION		Mode
position			Position Controller
kv	300.0	1/s	Proportional amplification
tn	0.05	s	Integral action time
t_predict	0.0004	s	Prediction time
t_total	0.0004	s	Total time
p_max	1.0E+30	Units/s	Maximum proportional action
i_max	0.0	Units/s	Maximum integral action
speed			Speed Controller
kv	3.5	As	Proportional amplification
tn	0.01	s	Integral action time
t_filter	0.0	s	Filter time constant
isq_filter1			ISQ Filter1
isq_filter2			ISQ Filter2
isq_filter3			ISQ Filter3

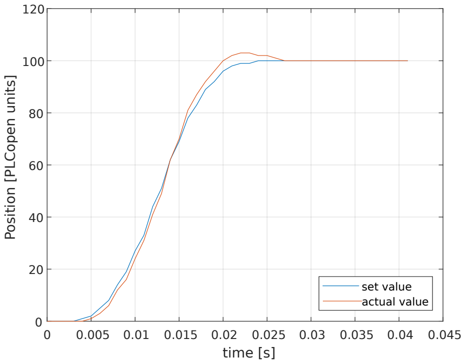


Figure 3.35: Servo controller parameters and ’step’ response

3.6.2 Outer loop

The outer loop receives the signal from the load cell, which is used to determine the torque that the pilot is exerting on the control column. The measurement is used by the FCS model to simulate the motion response of the control column. In the case of a position loop control architecture, the simulated control column position is first transformed to the motor axis domain, and this reference value is sent to the servo controller in the inner loop.

In our case, the outer loop routine runs on the CPU of the central PLC and is divided into several subtasks as described in subsection 3.3.4. The main program is created as a Simulink model from which the C code for PLC task is created, which is iterated every 400 μs.

Pilot torque measurement

The purpose of the installed load cell is to measure the torque that the pilot is exerting on the control column. However, we know that the load cell actually measures the longitudinal force in the four-bar connecting rod (at the control column’s side). This means that the sensor signal we detect does not measure pilot’s torque directly but also contains components of other dynamic effects that need to be addressed. Let’s recall the free body diagram of the control column to examine the forces and torques acting upon it.

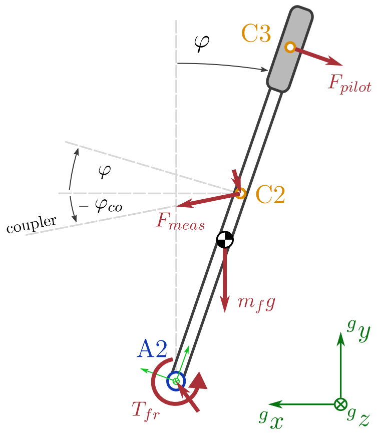


Figure 3.36: Control column free body diagram

- The load cell which is located directly at the control column will measure the force F_{meas} in joint C2, which produces a torque

$$T_{meas} = {}^f r_{c2y}\, F_{meas} \cos(\varphi - \varphi_{co})$$

  around control column's axis. (The angle φ_{co} remains small enough it could be neglected.)

- The force applied by the pilot to the control column will result in a torque T_{pilot}
- An unbalanced weight of the control column will produce a static, position dependent torque.
- In joint C2 there acts another force perpendicular to F_{meas}, which mainly exists due to the weight of the connecting rod and can be assessed together with the position dependent unbalance force. (The forces due to connecting rod's motion will also contribute slightly to this force, but remain neglected at this point.)
- The friction in the control column's axis could me modelled as a combination of viscous (velocity proportional) and coulomb friction: $T_{fr} = \mu_{fwd}\, sign(\dot{\varphi}) + d_{fwd}\, \dot{\varphi}$
- Friction in joint C2 is neglected.

Applying Newton's second law for rotating bodies, we can obtain T_{pilot} as:

$$T_{pilot} = T_{meas} - T_{stat}(\varphi) + \mu_{fwd}\, sign(\dot{\varphi}) + d_{fwd}\, \dot{\varphi} + I_{fwd}\, \ddot{\varphi} \tag{3.10}$$

where the index 'fwd' denotes the CLS plant forward of the load cell. Based on this relationship, we see that if we would be able to predict the unwanted components in the measured signal, we could calculate an estimated pilot torque $\widetilde{T}_{pilot}$.

$T_{stat}(\varphi)$ is the static, position dependent torque due to the unbalanced weight, which can be easily measured, identified and eliminated. I measured the static force on the sensor at different positions of control column, while ensuring that $T_{pilot} = 0$, which means that the only contributing components to F_{meas} were the $T_{stat}(\varphi)$ and the coulomb friction. From the plot below we notice the static hysteresis due to coulomb friction and the overlaid unbalance force, which we can easily approximate with a linear function.

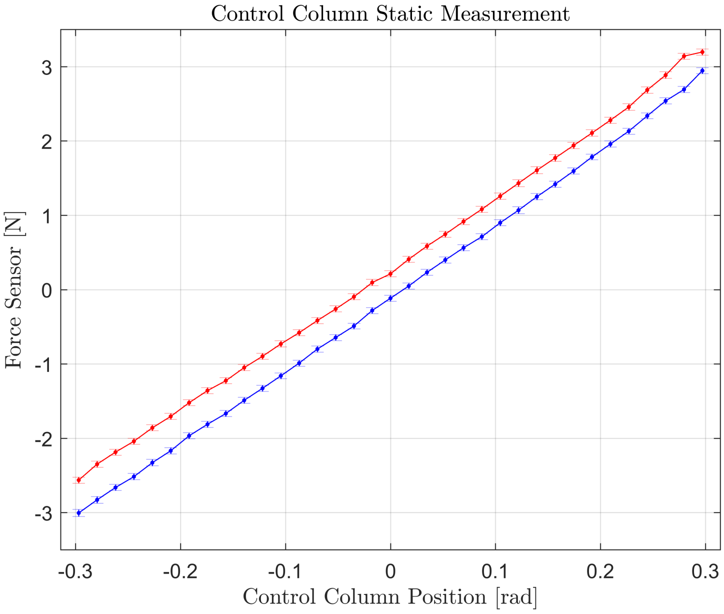


Figure 3.37: Static force measurement

After we have successfully identified the static unbalance torque T_{stat} in the measured signal, we could proceed and try to remove the damping, inertial and friction components as well by performing further experiments. While ensuring that $T_{pilot} = 0$, we would set the CLS mechanism in various random motion with the servo motor, while collecting the signals from the load cell and position sensor.

a) Based on Eq. 3.10 we could use a grey box modelling approach and use the experimental data to perform a parameter identification of μ_{fwd}, d_{fwd} and I_{fwd}, which would allow us to estimate T_{pilot}.

b) If the internal mechanics of the CLS plant is not easily accessible and geometric parameters are unknown, we could also model the fwd CLS dynamics as a black box (e.g. a neural network). We could use the experimental data to train the neural network to reproduce the F_{meas} signal from the φ, $\dot{\varphi}$, $\ddot{\varphi}$ signals when $T_{pilot} = 0$ and thus allow us to estimate and remove the unwanted components in the load cell signal.

However, in most cases, the effort of estimating the T_{pilot} might not be necessary, which will be argumented next.

Assume, that we provide our FCS model directly with the T_{meas} signal, which in addition to T_{pilot} still contains the inertial and damping components of the CLS control column:

$$\begin{aligned} T_{meas} &= T_{pilot} - d_{fwd}\,\dot{\varphi} - I_{fwd}\,\ddot{\varphi} \\ &= T_{pilot} - (d_{fwd}\,s + I_{fwd}\,s^2)\,\varphi \end{aligned} \tag{3.11}$$

We can simplify and redraw our CLS control system's outer loop as follows:

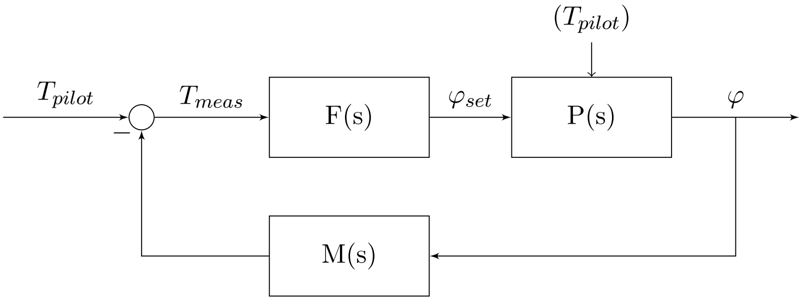


Figure 3.38: Simplified outer loop diagram

where:

- F(s) is the transfer function of our simple, linear FCS model given as:

$$F(s) = \frac{\varphi(s)}{T(s)} = \frac{1}{I_{fcs}\,s^2 + d_{fcs}\,s + c_{fcs}} \tag{3.12}$$

- P(s) is the closed-loop transfer function of the already tuned inner loop. (We assume, that the inner loop is able to perfectly compensate the disturbance load T_{pilot}.)

- M(s) represents the inertial and damping components of the CLS control column, that appear in the T_{meas} signal, which we already derived in Eq. 3.11:

$$M(s) = I_{fwd}\,s^2 + d_{fwd}\,s \tag{3.13}$$

The closed-loop transfer function of the entire CLS system can be determined as:

$$\frac{\varphi}{T_{pilot}} = \frac{F(s)\,P(s)}{1 + F(s)\,P(s)\,M(s)}$$

If we further assume, that the bandwidth of P(s) is significantly larger compared to bandwidth of F(s), we can use the approximation $F(s)\,P(s) \approx F(s)$ (as suggested in [13]), which gives:

$$\frac{\varphi}{T_{pilot}} = \frac{F(s)}{1 + F(s)\,M(s)} \tag{3.14}$$

Inserting equations 3.12 and 3.13 in 3.14 finally yields:

$$\frac{\varphi}{T_{pilot}} = \frac{1}{(I_{fcs} + I_{fwd})\, s^2 + (d_{fcs} + d_{fwd})\, s + c_{fcs}} \tag{3.15}$$

We can clearly see that if the inertia and damping of the control column are retained in the measured signal T_{meas}, its dynamics will simply be added to the simulated FCS dynamics. In most flight simulators, the installed control column very accurately represents the mechanics of the actual control column (FCS forward system) in a real aircraft and often even original parts are used in the simulator cockpit. This means that in such cases it is not necessary to suppress the dynamics of the simulator's control column, but rather use it to our advantage. We should exclude the dynamics of the control column (inertia, damping, friction) from our FCS model, so that when the simulated dynamics are combined with the real dynamics of the simulator's control column, the transfer function of the entire CLS (Equation 3.15) will yield the exact FCS behaviour, we originally strived for.

The previously mentioned methods for T_{pilot} estimation would only be of importance if the dynamics of the simulator's control column is substantially different from the one in the real aircraft we wish to simulate, which is almost never the case.

Stability analysis

The stability of the inner loop was already ensured during the servo controller tuning, however, the stability of the outer loop still needs to be analysed. Let us recall the simplified outer loop diagram in Figure 3.38. According to the Nyquist stability criterion we can evaluate the stability of the closed-loop by observing the frequency response of the open-loop transfer function, which for the system in Figure 3.38 can be written as:

$$\begin{aligned} G(s) &= M(s)\, F(s)\, P(s) \\ &= \frac{I_{fwd}\, s^2 + d_{fwd}\, s}{I_{fcs}\, s^2 + d_{fcs}\, s + c_{fcs}}\, P(s) \end{aligned}$$

If we plot the frequency response of the the first part of the open loop $M(s)\, F(s)$ for different ratios of inertia $R_I = I_{fwd}/I_{fcs}$ (see Figure 3.39), we notice that this inertia ratio will affect the magnitude at higher frequencies. From tuning of the inner loop, we know that the phase crossover frequency of $P(s)$ will occur further right, when the magnitude of $M(s)\, F(s)$ should already be established as $20\, log(I_{fwd}/I_{fcs})$. Therefore we can write the stability criterion for the CLS outer loop as [13]:

$$20\, log\left(\frac{I_{fwd}}{I_{fcs}}\right) + M_{P180} < 0 \tag{3.16}$$

where M_{P180} is the magnitude of $P(s)$ at the phase crossover frequency.

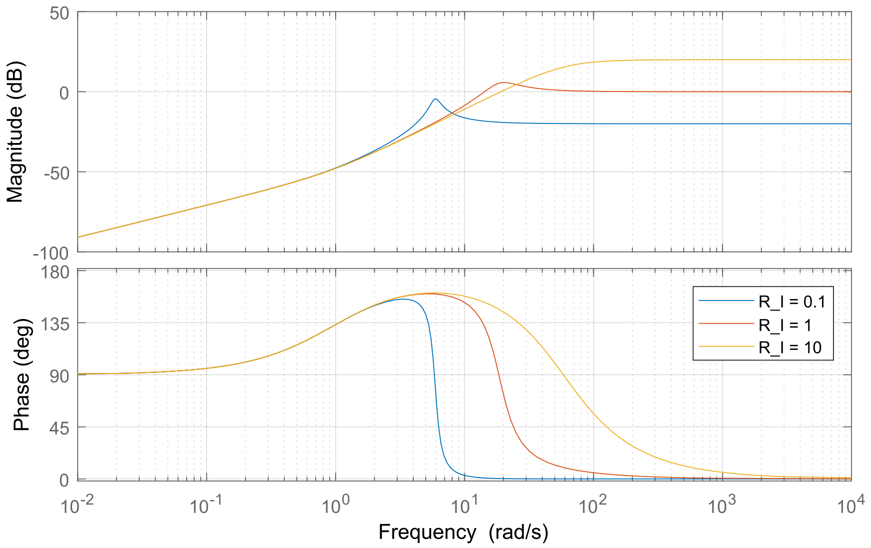


Figure 3.39: Bode plot $M(s)F(s)$

Looking at the stability criterion in Equation 3.16, we see that the inertia ratio I_{fwd}/I_{fcs} has to be kept below a certain value in order to guarantee the outer loop's stability. If all other parameters remain constant, we will approach the outer loop instability by:

- lowering I_{fcs}
  Which means that FCS below a certain inertia cannot be simulated.

- increasing I_{fwd}
  This is a very fluctuating parameter, because as the pilot firmly squeezes the control column, its effective inertia I_{fwd} will suddenly significantly increase. (See subsection 3.4.4) This means that we will have to account enough stability margin that such increase of I_{fwd} will not lead to instability.

- increasing M_{P180}
  When tuning the inner loop, choosing a larger gain margin will lower M_{P180} and thus improve the stability behaviour of the outer loop, however the compromise is a less responsive and dynamic inner loop controller.

Fortunately, in most aircraft (but especially larger airliners), the inertia of the aft FCS will be significantly larger in comparison to the inertia of the control column, which should help to avoid the instability of the outer loop. The behaviour when approaching outer loop's instability region will be demonstrated with experiments on the test bench in chapter 4.

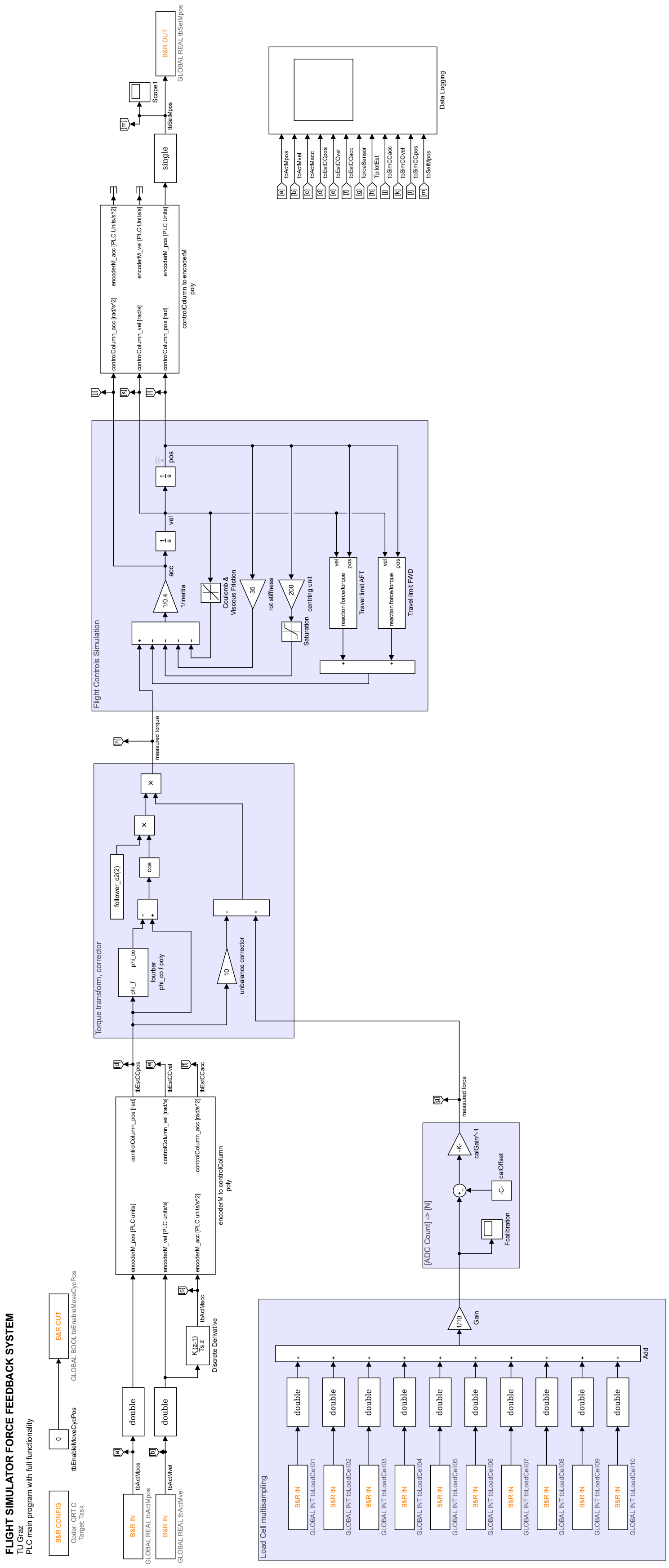


Figure 3.40: Simulink block diagram of the outer loop program

(ready for C code generation)

3.6.3 Safety considerations

The CLS is a powerful, high-torque motion system that is being operated in direct contact with the person seated in the cockpit. An unexpected movement of the control column (e.g. due to a hardware malfunction or software bug) has the potential of causing serious injury to the user, therefore it is important that we consider the safety aspects of the CLS and integrate the corresponding functionality in the system.

On the test bench we already installed an **emergency stop** button, which can immediately trigger a **safe torque off** (STO) of the actuator in case the user encounters a threat. Especially during experimenting, when the control loops are not yet perfectly tuned, it is essential that the operator can immediately disable the motion system at any time.

When the CLS will be implemented in full scale in the flight simulator, the STO e-stop buttons should still be available throughout the cockpit for emergency situations. We should, however, look for additional improvements that attempt to prevent such events, because during flight simulation the crew's main focus is on other tasks, which will increase their reaction time in the case of an emergency.

We could consider in what way to limit the motor's torque T_m and angular velocity ω_m to avoid high risk situations. We already limited the motors torque and velocity output to certain values T_m^{lim} and ω_m^{lim} during the servo commissioning, however, due to the large torque and dynamic performance demands, these limits have to be set relatively high and thus provide little protection. Generally, a motion system is most dangerous when it is moving fast with a lot of of torque, which means that the actuator is producing a lot of power.

$$P_m = T_m \, \omega_m$$

If we define an additional power limit P_m^{lim} for the motor, the motion system will be able to generate a lot of torque at slow speeds (giving enough reaction time for a STO), as well as move fast provided that movement requires relatively little torque (resulting in lower damage/injury in case of a collision). The resulting operating area of the motor in the (four quadrant) torque-speed chart should be restricted as shown in Figure 3.41. The power limit is only enforced in the driving quadrants, when the motor is actually providing power into the system.

Furthermore, we could also continuously monitor and analyse the measured signals. Any unusually large spikes in the measured force (that did not occur due to expected events - e.g. a virtual hard stop defined in the FCS model) could be an indication of a dangerous collision situation, that would automatically trigger a STO of the motor. In addition, any larger motion tracking error of the servo controller suggests that the system is not operating as it should.

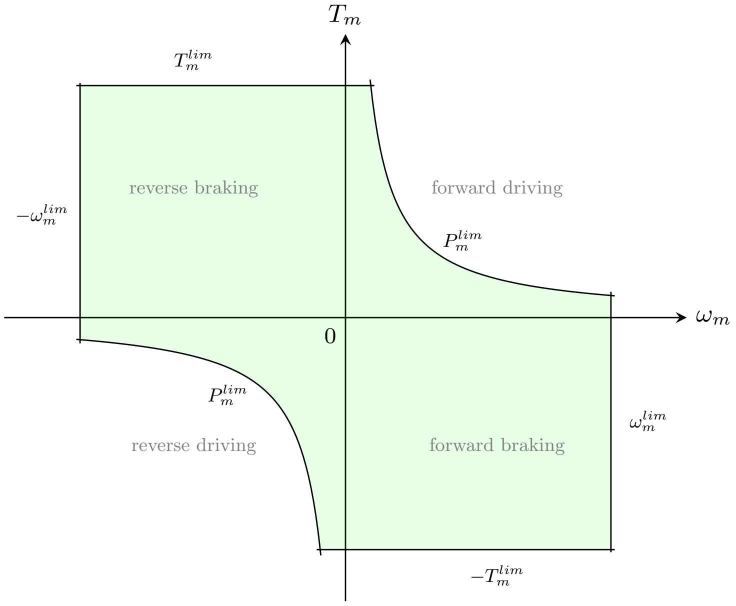


Figure 3.41: Motor safety limits

Chapter 4

Results and discussion

In this chapter, the **performance** of the developed control loading system will be **evaluated** by performing a series of experiments on the test bench. The acquired data will be presented, and the conclusions based on the observations will be discussed.

4.1 Masking mechanism's friction

The main motivation for installing a force sensor and implementing an admittance type CLS was to mask the large amount of friction present when back-driving the mechanism - predominantly the planetary reduction gearbox. With the new CLS control system architecture, the mechanism's friction is compensated for by the servo motor and, from the user's perspective, successfully and completely eliminated.

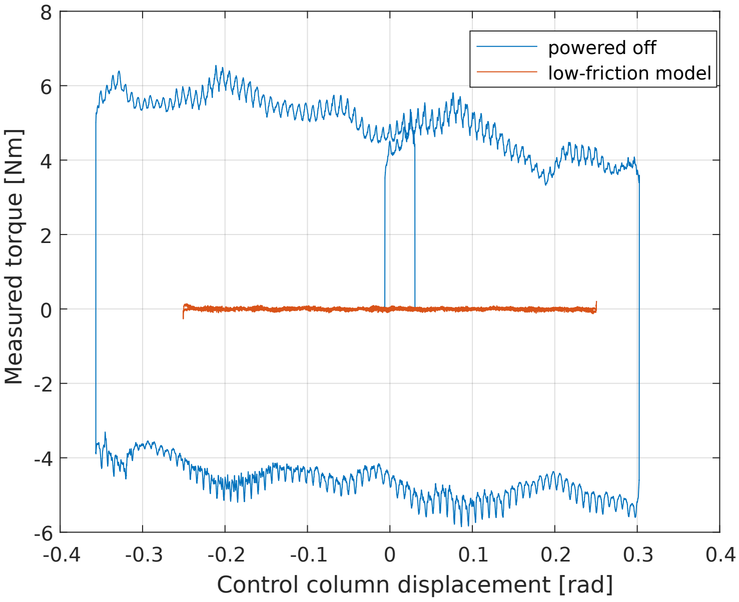


Figure 4.1: Successfully masked mechanism's friction

This is demonstrated in the torque-displacement plot above, which shows how much torque the user had to apply to overcome friction and move the control column. I measured the torque T_{meas} on the control column with the load cell and plotted it against the control column's displacement. The measurement was repeated twice: First with the actuator completely powered off to show the mechanism's natural resistance while being back-driven, and then again with the CLS simulating a very low-friction model to demonstrate its ability to conceal the existing friction.

We can see that with the actuator powered off, a substantial breakout force is required to move the control column because a large amount of friction is present in the system. Additionally, the meshing and friction in the gearbox result in a jerky motion. On the other hand, when the CLS was running and simulating a very low-friction model, the effort required to move the control column was practically non-existent, and the motion was a lot smoother, which demonstrates the sensitive dynamics and precise inputs we are able to simulate on this system.

4.2 Validating the FCS model

We need to validate, that the CLS indeed behaves in the same way as the FCS model simulation that runs in its core.

As already mentioned in section 3.5, the dynamics of a FCS can be effectively summarized with two diagrams:

- The **static force-displacement diagram** shows the relationship between the static control force and the control column's displacement φ. Experimentally we obtain this diagram by very slowly moving the control column through its entire travel range and measuring the forces and displacements that occur during this process. This plot allows us to identify *spring elements*, *hinge moments*, *travel limits*, *static friction (hysteresis)* etc.
- The **free response plot** shows the displacement of the control column over time after it has been brought to a certain displacement and rapidly released to freely move. This plot therefore reveals us more information about the *inertia* and *damping* of the FCS.

These measurements need to be performed in a real aircraft, to acquire the data based on which the FCS model parameters can be determined. After the FCS model is deployed on the CLS, the experiments are repeated in the flight simulator to validate, that the behaviour of flight controls in the simulator is identical to the real aircraft. For mechanical flight controls where the aerodynamic forces play a role, this process has to be repeated for different flight states.

The models deployed on our test bench do not yet represent a specific real aircraft type;

therefore I merely simulated different FCS models with various features and checked that the CLS was able to track the model's output successfully. Instead of the control force at the control column's grip, I measured the torque detected by the load cell T_{meas}, which (for steady state conditions and low friction in column's axis) well approximates T_{pilot}.

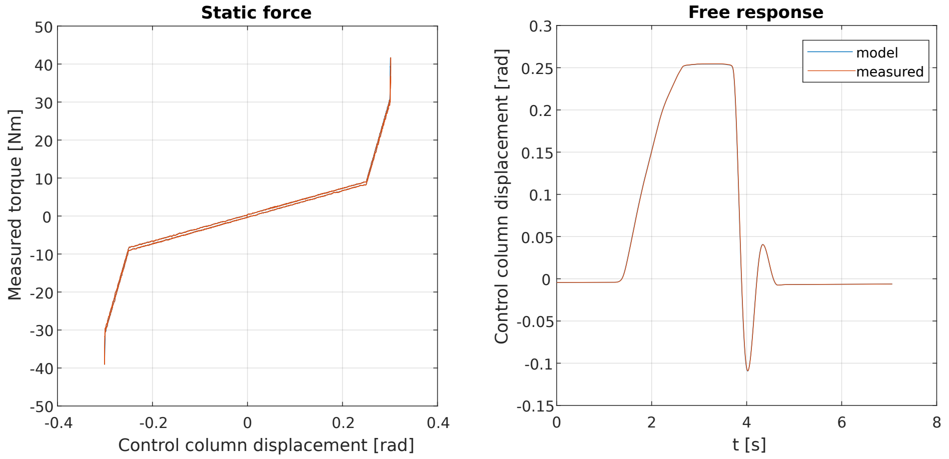


Figure 4.2: FCS model validation 1

($I_{fcs} = 0.3$, $c_{fcs} = 35$, $d_{fcs} = 1.5$, $\mu_{fcs} = 0.25$ ...)

Figure 4.2 shows a FCS model with a relatively low amount of friction, which results in a narrow hysteresis (see the static plot) and an under-damped free response. In the static plot, we can clearly identify the two travel limits in each direction: the first represents the travel limit of the control surface, after which the relatively flexible cable linkage can still be stretched. The second is the stiff travel limit of the control column itself. Notice, that the CLS has done a very good job at tracking the FCS model output position, since the two plots (model output position and actual measured position) are perfectly aligned in all diagrams.

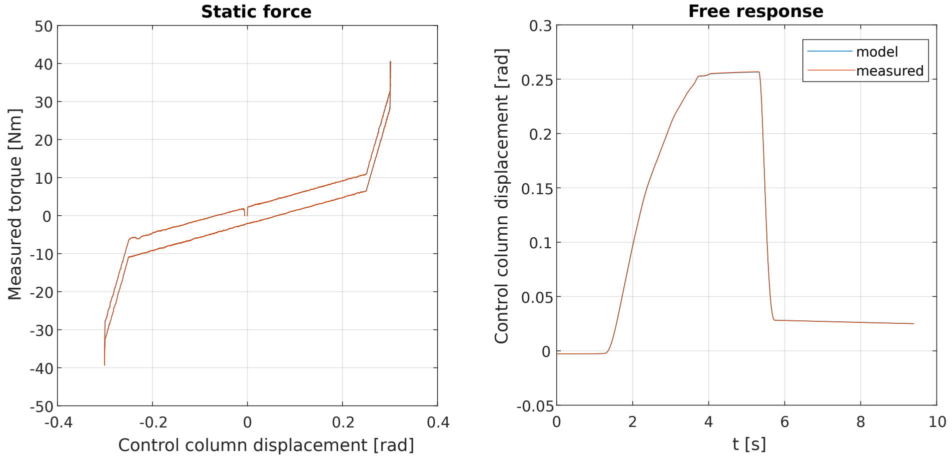


Figure 4.3: FCS model validation 2

($I_{fcs} = 0.4$, $c_{fcs} = 35$, $d_{fcs} = 4.0$, $\mu_{fcs} = 2.0$ ...)

The second FCS model in Figure 4.3 possesses more friction and damping, which is directly noticeable from the wider hysteresis and a strongly damped free response without overshoots. The large amount of static friction has also prevented the control column to freely return to its initial position.

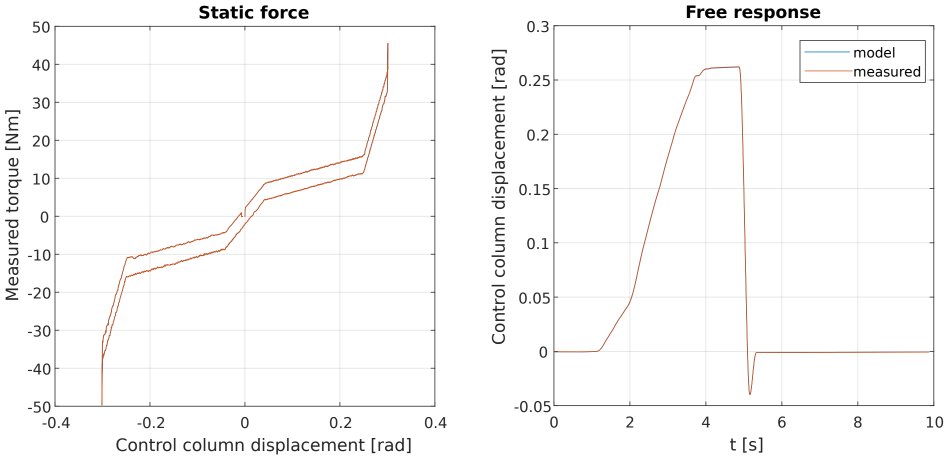


Figure 4.4: FCS model validation 3
($I_{fcs} = 0.4$, $c_{fcs} = 35$, $d_{fcs} = 4.0$, $\mu_{fcs} = 2.0$ ...)

The last model presented in Figure 4.4 features a centering unit, which is essentially an additional spring that is only effective in the proximity of the neutral position.

4.3 Motion tracking

In the next experiment, the CLS was rapidly moved in a sinusoidal fashion back and forth, to check how well the CLS will track the fast-changing motion of the model's output. From Figure 4.5 it is clear, that the motion system performs this task very well.

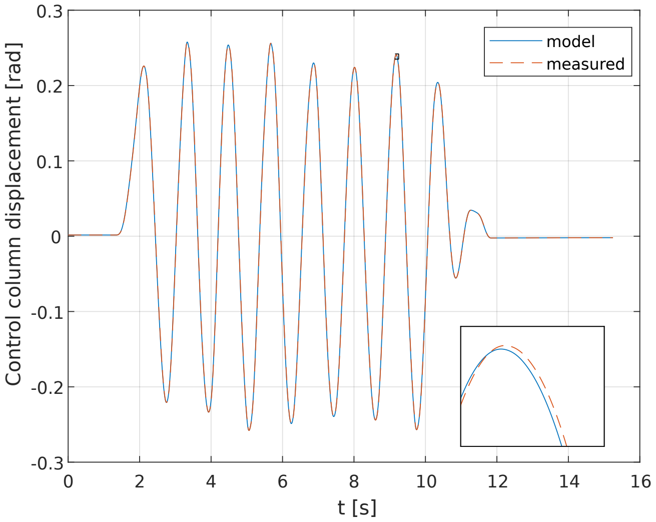


Figure 4.5: Sinusoidal motion tracking

4.4 Stability limits

In subsection 3.6.2 we discussed the stability of the outer loop, which will now be demonstrated experimentally. I loaded a FCS model with a low simulated inertia I_{fcs}, which resulted in a reduced stability margin. If the control column is pulled with a loose grip, the system will remain stable. However, if the pilot tightly squeezes the control column during operation, the effective inertia of the control column (I_{fwd}) will increase to the point when outer loop becomes unstable. This results in very noticeable vibrations and oscillations of the control column that can be seen in Figure 4.6.

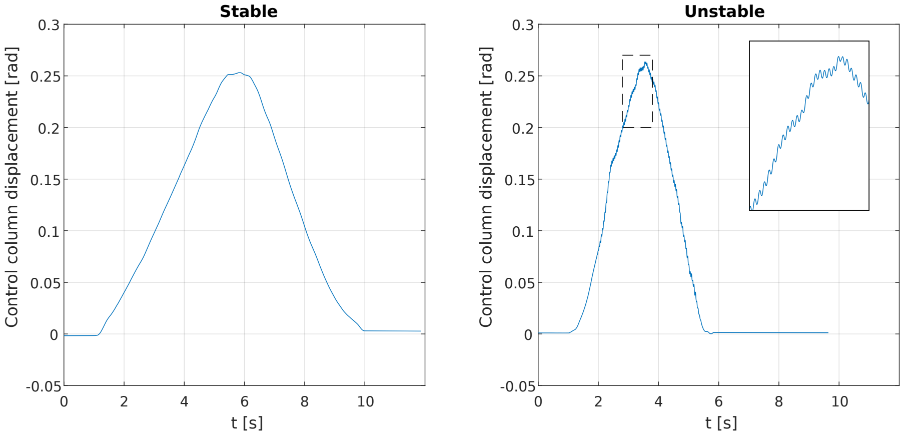


Figure 4.6: Moderate instability of the outer loop
($I_{fcs} = 0.25$, $c_{fcs} = 35$, $d_{fcs} = 1.5$, $\mu_{fcs} = 0.25$)

If we continue to decrease the simulated inertia I_{fcs}, the outer loop stability margin will become even smaller and squeezing of the control column will result in much more violent shaking and severe vibrations of the control column, which cannot be kept under control anymore (see Figure 4.7).

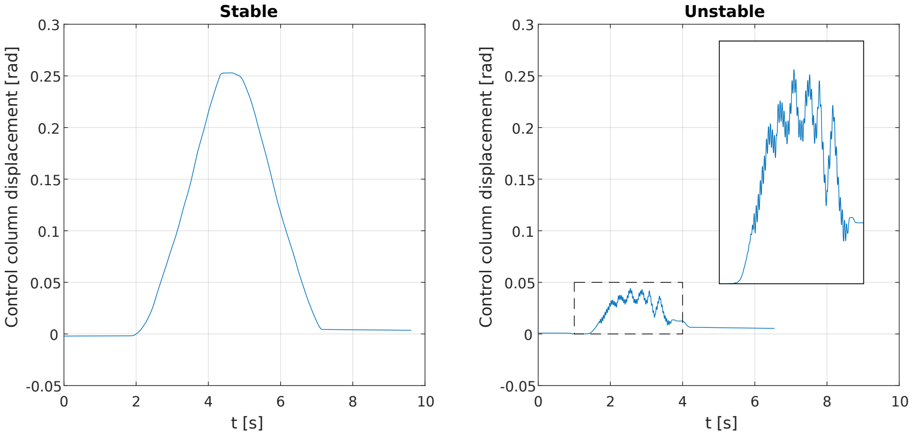


Figure 4.7: Severe instability of the outer loop
($I_{fcs} = 0.15$, $c_{fcs} = 35$, $d_{fcs} = 1.5$, $\mu_{fcs} = 0.25$)

4.5 Gearbox imperfections

If one moves the control column with the CLS powered off, a jerky, unsmooth motion of the column will be noticeable (see Figure 4.1). The cause for this lies in the reduction gearbox, and the effect is especially pronounced when a gearbox is back-driven. The meshing gears create complex, inconsistent friction contacts, which can be felt as irregularities in the transmission.

Once the CLS is powered on, the motion is much smoother in comparison, however, still, certain imperfections from the gearbox can be felt, which is not desirable. These imperfections are in fact small tracking errors of the servo controller, which cannot compensate for the small non-linear gear meshing effects accurately enough - especially when a large load torque is applied on the control column. In a typical motion control application this error remains unnoticeable; however, the extremely sensitive human haptic perception is able to detect it. This is the very reason why the flight simulators of highest fidelity will rather incorporate motors in a direct drive configuration and avoid using geared motors for the most critical flight control channels. However, these types of motors are usually not cogging-free, and therefore this disturbance also needs to be compensated for.

I am confident that the current geared motor CLS solution can be further improved upon and meet the fidelity requirements of our research flight simulator after some additional experimentation is made. I suggest that in the future, we avoid using a two-stage planetary gearbox, as the undesirable friction effects get multiplied. Choosing a slightly larger motor with a lower gearing ratio would most likely improve the current situation.

A potential improvement could also be a modification in the servo control architecture. Non-linear disturbances such as the mentioned gearbox imperfections and cogging torque have a very reproducible, deterministic nature. If their behaviour is predictable, we could attempt to compensate for these disturbances with a feed-forward control for which we would require a physical model of these disturbances to calculate the compensating feed-forward torque. However, due to the complexity of friction contact physics, this is a nearly impossible and impractical task to complete. As described by Velthius in [16], [15] a solution might be to utilise learning feed-forward control, where a neural network is incorporated in the feed-forward path. The network learns where and how the tracking errors occur and attempts to minimise them by applying a corresponding feed-forward torque to the servo controller. In his research, Velthius successfully eliminated most of the cogging and gearbox imperfections, which makes this approach a promising starting point for our further research.

Chapter 5

Conclusion

The thesis project posed an intriguing personal challenge for me because it involved the entire development process of the presented CLS test bench. A significant amount of effort and attention to detail went into acquiring compatible hardware components and studying them to make sure they function together properly. It was an incredible pleasure to finally see the individual parts come together in a functioning and well-performing control loading system.

I am confident that the newly introduced B&R Automation control hardware is a step in the right direction for our further control loading endeavours, because it offers a lot of advantages in terms of performance, ease of operation as well as safety. We have established a CLS platform, that has the potential to evolve into a truly advanced, state of the art system packed full of features and simulation possibilities.

The additional force sensor and an admittance type CLS controller showed excellent performance improvement compared with the previous impedance type control loading already implemented on the rudder pedals. The friction of the mechanism can now be successfully compensated for and the features of the FCS model such as travel limits, detents etc. are very realistically simulated. This means that (nearly) arbitrary flight control system dynamics can be instantly simulated by merely reconfiguring and updating the Simulink model, which is a very useful feature - especially for a reasearch flight simulator. The remaining force feel imperfections due to the gearbox should be improved, when a more suitable transmission solution is found and installed. With the now well-functioning force sensor placed at the control column, even non-backdrivable transmissions could be considered as an alternative.

I am very pleased with the outcome of the project, because significant improvements were achieved compared to the existing CLS, while the the new system also offers more opportunities and potential for easier and more comfortable further development.

In the future, more research and development will be conducted on the test bench to further improve the performance and integrate the CLS in the flight simulation itself. The following list contains guidelines for future development:

- improve the force-feel imperfections due to the current gearbox - alternative transmission / feedforward compensation?
- implement and evaluate the performance of the alternative force-loop architecture (see [5])
- establish a connection from the CLS PLC to the flight simulation network
- implement parameter exchange between the simulation software and FCS model
- incorporate the autopilot functionality and flight controls back-driving
- develop the same CLS working principles to the full-scale flight control channels

List of Figures

List of Tables

Bibliography

[1] Moog control loading systems. `http://www.moog.com/products/control-loading-systems.html`. Accessed: 2018-01-20 20:15.

[2] D. Allerton. *Principles of Flight Simulation.* John Wiley & Sons Ltd, 2009.

[3] B&R Automation. *Motion Control Concept and Configuration.*

[4] European Aviation Safety Agency. *Certification Specifications for Aeroplane Flight Simulation Training Devices*, July 2012.

[5] A. Gerretsen, M. Mulder, and M. M. R. van Paassen. Comparison of position-loop, velocity-loop and force-loop based control loading architectures. 2005.

[6] J. Hall. *Guyton and Hall Textbook of Medical Physiology.* Saunders Elsevier, 12 edition, 2011.

[7] T. Krauss. Ruderkraftsimulation am flugsimulator der tu graz. Master's thesis, TU Graz, 2015.

[8] K. J. Kuchenbecker. *Characterizing And Controlling The High-Frequency Dynamics Of Haptic Interfaces.* PhD thesis, Stanford University, 2006.

[9] F. Lutze. Stability and control stick free characteristics. `http://www.dept.aoe.vt.edu/~lutze/AOE3134/`. Accessed: 2018-01-25 19:30.

[10] J. M. McCarthy. Four-bar linkage analysis. `https://synthetica.eng.uci.edu/mechanicaldesign101/McCarthyNotes-2.pdf`. Accessed: 2018-02-06 21:20.

[11] I. Moir and A. Seabridge. *Aircraft Systems.* John Wiley & Sons Ltd, 3 edition, 2008.

[12] M. Pelletier. Servo motor sizing concepts. *Design News*, 2009.

[13] P. Qi, H. Wang, and J. Han. Stability of position-loop based hydraulic control loading system. 2010.

[14] C. Spitzer, editor. *The Avionics Handbook.* CRC Press LLC, 2001.

[15] W. Velthius. *Learning Feed-Forward Control.* PhD thesis, 2000.

[16] T. V. Wubbe Velthius. Learning control of a flight simulator stick. 1998.