SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY

TECHNICAL UNIVERSITY OF MUNICH

Bachelor's Thesis

Concept Development, Mechanical Design, and Programming of a Braking Device to Protect The Robot In A Test Facility To Analyze Effective Robot Masses

Abdalla Mahamid

SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY

TECHNICAL UNIVERSITY OF MUNICH

Bachelor's Thesis

Concept Development, Mechanical Design, and Programming of a Braking Device to Protect The Robot In A Test Facility To Analyze Effective Robot Masses

Author:	Abdalla Mahamid
Supervisor:	Prof. Dr.-Ing. Sami Haddadin
Advisor:	M.Sc. Robin Kirschner
Submission Date:	June 23, 2024

I confirm that this bachelor's thesis is my own work and I have documented all sources and material used.

Munich, June 23, 2024

Abdalla Mahamid

Abstract

The field of physical human-robot interaction (HRI) addresses safety concerns at the interface between humans and robots. The ISO 15066 standard specifies safety requirements for collaborative industrial robot systems, including velocity limitations to mitigate injuries [8]. The concept of "effective mass" plays a critical role in determining potential injury severity [8, 11]. As part of this research, a test stand has been developed to ascertain the effective mass of the robot (see figure 2.1). During the test, the robot impacts a pendulum at a specific position and speed, with data being captured by various sensors installed at different points on the test stand. However, a challenge arises with the test stand design: the pendulum swings back and collides with the robot after the initial impact. To address this issue and prevent further collisions, this work aims to develop, design, and implement a braking system to safeguard both the robot and the test stand.

The braking system was designed following the VDI-2222 engineering guidelines [14] and implemented using real-time control hardware and software provided by National Instruments. This system detects the pendulum's motion and activates the brakes to hold the pendulum, preventing it from swinging back. The solution incorporates a servo motor and a caliper with a bike disk brake, controlled by a computer algorithm that processes data from a motion sensor.

The testing of the braking system, which effectively brings the pendulum to a controlled stop and returns it to its initial position, has significant practical implications. The results of this thesis not only improve the current test stand but also lay the foundation for future scalability. There is potential for further system automation, incorporating advanced feedback mechanisms and adding a force sensor to continuously monitor the force applied to the pendulum rather than just at specific positions.

Contents

1. Introduction

Artificial Intelligence (AI) has evolved and become integral to our lives. It has a wide range of applications and usages, ranging from robots in the automotive industry assembly lines [17], to language models such as chatGPT, self-driving cars, and much more. Many AI-based robotic systems rely heavily on regular interaction between the end-user and the system. In such systems, the interface between humans and robots directly affects the system's safety and overall usability. As a result, HRI has become widely influential across many sectors. HRI is particularly relevant in dynamic and partially unknown systems, where human safety is of high priority [11].

The ISO 15066 standard is an example of a standard that defines roles and limitations to minimize hazards associated with robotic motion [8], such as, the limitation of the robot velocity that is correlated with injuries [11]. *Effective mass* is an important parameter contributing to the potential injury severity [8, 11]. Therefore, a precise, effective robot mass is required. As the robot manufacturers do not provide sufficient data to obtain an adequate value of effective mass [15], a rudimentary test stand was developed in the Robot Performance and Safety Lab at the Munich Institute of Robotics and Machine Intelligence (MIRMI) to determine a practical value.

The setup involves a test stand where a robot impacts a pendulum of known mass. The robot's effective mass is estimated by measuring the collision's force and velocity and assuming the momentum conservation [8, 11]. However, when the robot collides with the pendulum, the pendulum swings back and hits the robot. Constant hits, mainly when a more potent collision test is performed, will negatively impact the robot. Thus, a braking system has been developed to prevent the pendulum from swinging back.

The braking is designed following the VDI-2222 engineering guidelines [14]. These guidelines are used to design, engineer, and develop the braking system from a mechanical perspective. In addition to the guidelines, the *National Instrument Company (NI)*, a real-time embedded industrial controller and re-configurable IO Modules, provides hardware and software solutions that we use in our system.

The primary focus of this thesis is the systematic development and implementation of an automated braking system for the test stand. This system protects the test robots and equipment from damage. Therefore, as the functionality improves, more potent hits on the pendulum can be performed across the spectrum of force and velocity. Moreover, the thesis will not only enhance the functionality of the current test stand but also provide a more reliable method for measuring the effective mass of robots in various operational contexts.

The structure of this thesis is as follows: Chapter 2 provides background information about collaborative robots (cobots), the development of safety standards, the importance of effective robot mass in relation to safety, how it is measured, and the contributions of this thesis. Chapter 3 presents the problem being addressed by this thesis and analyzes it. It also explains the guidelines followed to overcome the problem and design the solution. Chapter 4 details the solution design and provides comprehensive information about the steps taken toward the solution. Chapter 5 focuses on the implementation of the designed solution. Chapter 6 includes the evaluation of the implementation and provides experimental results. Finally, in chapter 7, the thesis concludes with a summary.

2. State of the Art

The emergence of cobots has changed traditional industrial environments into shared spaces where humans and robots coexist and cooperate. The robots in the collaborative environment are designed to operate alongside humans, unlike the transitional robots, designed to operate in isolated environments with no shared spaces [3]. Therefore, this robot integration with humans poses unique safety challenges, where a potential injury from a human-robot collision can be a dangerous event for human life. On that account, the effective mass is one of the primary human safety parameters affecting the injury severity [8, 11]. Thus, much research has been done as a step forward to determine a precise, effective robot mass to be able to resolve the correlation between it and the injury magnitude. Accurately determining the effective robot mass is valuable data for analyzing and setting limits on various parameters, such as velocity and force, contributing to estimating the effective robot mass [8, 11].

2.1. Historical Perspective on Cobots

In the 1990s emerged the development of robot-human collaboration in shared workspaces [13]. The primitive cobots were simple and low-cost and designed to assist humans with basic tasks without any sophisticated development of safety as a priority [1]. Over time, the integration of the cobots demanded better safety standards as they coexist and work directly alongside humans. Therefore, the need for human safety in shared spaces with robots presented new challenges that need to be developed to keep the integration process. Any human safety issue can lead to the inhibition of that process and detain it.

2.2. Safety Standards in Robotics

Some international institutes discussed and designed human-robot collaborative safety. They developed protocols, rules, and standards, and they play an important role in ensuring the safety of human operators in shared work environments. The robots are more integrated into society and interact with humans beyond traditional industrial settings. Therefore, the necessity for rigorous safety protocols and standards becomes more evident. The International Organization for Standardization (ISO) provides comprehensive guidelines for the safety of HRI. A vital standard is ISO 15066 [8], a technical specification that guides safety requirements for collaborative industrial robot systems. It introduces the limits on the power and force of robots, and also the design of safety-related control systems, and the implementation of protective measures [8]. This thesis will focus on the primary parameter contributing to estimating a precise, effective robot mass, which will be the development of a test stand.

2.3. Robot Effective Mass and Kinematics

The concept of effective mass is crucial in the context of robotics, particularly when considering the interactions between robots and humans. The effective mass of a robot is defined as the perceived mass at the point of contact during an impact [11], and in our tests, it is the impact of a robot on a pendulum. The effective mass is an important parameter contributing to the potential injury severity [8, 11]. According to the paper "Notion on the Correct Use of the Robot Effective Mass in the Safety Context and Comments on ISO 15066 [11]" understanding the effective mass helps in designing safer robot systems, and it will provide a quantitative measure to assess and limit the kinetic energy transferred during unexpected collisions and thereby reducing the risk of injuries. This is crucial for human safety as it complies with some safety regulations and rules that define the limitation and another aspect of human-robot interaction in a shared workspace [19].

In addition to the effective robot mass, robot kinematics involves the study of motion without considering the forces that cause it [22]. As outlined in the "Springer Handbook of Robotics, chapter 2. Kinematics [20]" kinematics is very important in analyzing and synthesizing robot movements. It enables the design of robotic systems with the possibility of forecasting the position, orientation, speed, and acceleration of different robot parts. Accurate kinematic models are essential for controlling motion and achieving precise movements, which are critical for the safety and effectiveness of robotic applications. In robot design and control, kinematics is a crucial parameter in calculating the necessary joint configurations to achieve the desired positions and orientations of the robot. Understanding kinematics is essential for optimizing robot structures and mechanisms to improve efficiency and functionality in complex environments.

As robot kinematics and robot effective mass are important, the manufacturer often provides incompatible data regarding the dynamic priorities of their robots [15]. Therefore, an accurate prediction of the effective mass is a challenge to resolve. In addition, robotic kinematics focuses on describing the pose, velocity, and higher-order derivatives of rigid bodies within robotic mechanisms, excluding forces and torques [20]. Consequently, at MIRMI, we rely on the empirical method as a step toward the estimation of accurate, effective mass and kinematic pose, where these parameters are crucial to determining the injury magnitude. Hence, both robot kinematics and robot effective mass are part of the primary research area in our Lab.

2.4. Empirical Studies and Methodologies

Experimental setups can be an efficient method for estimating data that is not available from the manufacturer. They can provide essential insights into real-world performance and practical implications. In this section, we will elaborate on the empirical approach and methodology used in the field of robotics to capture data necessary for researching the safety of humans and the effectiveness of robots, particularly in human-robot interaction scenarios. To this degree, a test stand for determining the effective mass and other studies has been constructed. The test stand was designed primitively, where a pendulum hung on a mechanical spline connected to a two column (see figure 2.1). In the middle of the pendulum is a force sensor that measures the force of a mass impact of an object. In addition, an encoder is installed on the left side of the mechanical spline to measure the velocity and the position of the pendulum. This design allows a robot installed in front of the test stand to impact the pendulum, and through calculating force and velocity, the effective mass will be estimated [11].

However, the setup has a problem: as the robot collides with the pendulum, it swings and hits the robot back as it returns to the start position. Continued hits will damage the robot and setup over time, leading to a biased data capture that negatively impacts the precision of the estimated effective robot mass. Obtaining sufficient and precise data is a requirement to have an accurate estimation of effective mass. Thus, a proper and efficient setup is crucial to sustain unbiased data captures. Therefore, a braking system has to be developed and implemented into the setup to manage the experiments in the best way possible.

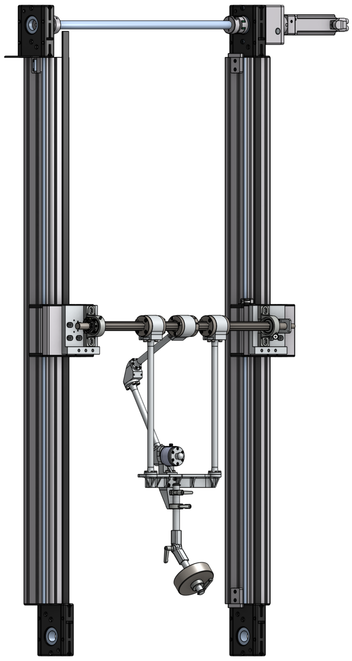


Figure 2.1.: Test Stand

2.5. Thesis Relevance

This thesis focuses on developing a braking system for a pendulum integrated within a robotic test environment. It is a significant improvement to prevent the pendulum from swinging back and potentially damaging the robot or the test stand during experiments when a robot collides with the pendulum to capture and analyze data and estimate its effective mass with its kinematics configurations.

2.5.1. Significance of the Braking System

The primary purpose of the braking system is to mitigate hazards associated with the pendulum's uncontrolled movements after a robot impacts it in the test environment. In applications where robots operate with the pendulum, the response must be controlled precisely to avoid accidental collisions leading to equipment damage or failure, which implies biased captured data and more expenses. The braking system is an improvement as a safety feature to the test environment and equipment that prevents them from damage or any hazards during the test process. As this braking system is integrated into the test stand, the following points can be achieved:

- **Safety Enhancement**: The braking system directly contributes to the overall safety of the test stand and the test robot by preventing unintended collisions. This is crucial for avoiding damaged equipment that can delay the test process, the research phase, and the overall costs.
- **Control System Design**: The braking system will be automated. As the pendulum moves, the braking system will stop it in a required position as defined in the requirement list (3.1) without any human interaction. Then, the system will wait for feedback to release the pendulum under a controlled condition and move it to the neutral position.
- **Scalability and Adaptation**: The principles and technologies developed for the braking system can be adapted and scaled for other robotic applications where moving objects interact with robots. The braking system is automated. As the system captures a move, the brakes should hold the object and wait for feedback (which is done manually, in our case, through the software). This step could also be automated when feedback is received from the robot itself, as the robot is integrated with the same system.

The foundational argument for showing that a braking system is pivotal has been established. This thesis will first analyze the problem and identify the challenges it faces. Secondly, it will develop and propose multiple solutions and adopt the first-rated one for implementation, where the thesis will show the technical side of the solution and the used hardware and software. Lastly, an evaluation will be conducted to review the problem and the solution. Finally, the thesis will contrive a conclusion.

2.6. Current Trends in Collaborative Robotics

The field of collaborative robots is experiencing significant advancements due to the integration of AI. Safety remains a critical focus in the development of cobots, which are designed to operate near humans. Several key trends and technological advancements are contributing to enhanced safety in collaborative robotics [3]. This technology allows for more capabilities, making it more efficient and safer for human interaction. Recent research has highlighted the expanding role of AI, especially in deep learning and reinforcement learning [3].

The advances in cobots are not only due to AI but also to other technologies, such as sensors. Cobots are now equipped with advanced sensors that are aware and respond to human presence and movements. Examples of these sensors include cameras and LiDAR, which provide real-time data about the robot's surroundings. The robot's algorithms use this data to make appropriate decisions [3].

As previously mentioned in the thesis, advances in AI have played an essential role in the development of cobots. AI is also a key factor in ensuring safety. The machine learning

(ML) algorithm is vital in enhancing the safety of cobots by enabling predictive maintenance and anomaly detection [3]. ML is a subset of AI that involves statistical operations, allowing computers to learn from experience. The ML algorithms analyze data from the robot's sensors to predict potential failures and detect unusual behavior patterns that may indicate safety risks [3]. The same applies to reinforcement learning – an autonomous learning system that continuously adjusts behaviors based on feedback to maximize a reward [3]. This enables cobots to improve their interaction with human coworkers.

As we can see, the AI tries to prevent accidents from happening, but there is also a margin of error that can occur. In this case, an overall limitation, such as velocity, is still required to minimize an injury's severity.

2.6.1. Real-World Applications

The automotive assembly line is a well-known area where cobots are integrated to complete tasks such as wheel assembly and hood installation. In this scenario, cobots work alongside humans to assemble the wheels while workers handle other important tasks, such as repairing the hood [3]. Cobots are equipped with advanced sensors and force feedback mechanisms to ensure safe operation around human workers. They can detect human presence around them and adjust their movements to prevent accidents and injuries. Additionally, force feedback and collision detection allow the cobots to identify any unexpected resistance or contact, helping to prevent collisions with coworkers by halting their operations immediately upon sensing an obstruction [3].

Another example where the cobots are integrated is the logistics, where the cobots are increasingly used for tasks such as order picking and sorting in warehouses. For instance, cobots are used to help pick items from shelves and place them in designated shipping bins [3]. This speeds up the order fulfillment process and reduces the physical strain on human workers by handling heavy and repetitive tasks. The cobots used in logistics come with safety-rated monitored stops and hand-guiding features. These features enable human workers to manually guide the cobots when necessary, ensuring precise and safe operations. Additionally, the monitored stop feature ensures that the cobots can be quickly halted if a safety concern is detected [3].

3. Problem Analysis

Robots are applied to our lives in shared spaces as we observe, e.g., vacuum robots or in the automotive industry's assembly lines [17]. The HRI is a significant subject marked by the ability of robots to operate alongside humans without traditional physical barriers. This integration process introduces substantial challenges to ensuring human safety while interacting with robots, represented by injuries and accidents. A primary factor in the safety equation contributing to the potential injury severity is the robot's effective mass [8, 11].

3.1. The Challenge of Robot Effective Mass

The robot's effective mass is the mass experienced by an object or human during a collision. It depends on multi-factors such as the robot's kinematic and inertial characteristics, joint configuration, and much more in the motion-related and dynamical properties [11]. Hence, the effective mass is a crucial concept in human safety by collaborative robots[8].

3.1.1. Empirical Measurement Challenges

As estimating the effective robot mass is a challenge, primitive empirical measuring can be a practical approach to effective mass determination [11]. On the other hand, setting up an experimental stand presents numerous challenges:

- **Experimental complexity**: Setting up an experimental stand that derives accurate data regarding robots' dynamics and motion is challenging due to the need for high-precision measurement tools. In addition, it is necessary to ensure the tests' results are accurate and correct by comparing them to expected data for specific tests.
- **Experimental Costs**: Setting up a precise test stand can incur significant expenses, particularly when requiring costly hardware for accurate data derivation.
- **Safety risks**: Hazards for test engineers and tested devices need to be reduced. [6].
- **Logistics**: As we observe in the lab, the testers have to test multiple robots, and logistic challenges will crystallize by bringing and installing multiple robots to the test stand.

Therefore, these challenges should be considered when designing and constructing an empirical measurement test stand.

3.2. Implications of Inaccurate Effective Mass Estimation

A biased data capture can lead to an inaccurate estimation of effective robot mass that could significantly underestimate the potential impact force, which is a result of a wrong decision of a robot [21], posing a substantial risk to human safety [8].

The speed and power and force limitations in ISO 15066 [8] section 5.5.4: Speed and Separation Monitoring and section 5.5.5: Power and Force Limiting are significant to protecting human safety. Thus, any exceedance of these limits is not acceptable. Therefore, any outlier in the captured data can lead to a problem by estimating the correct values [21], leading to biased effective robot mass and safety issues.

As a logical implication of biased data capture, it would extend the testing phase and cause extra costs to the staff working and the resources used.

3.3. Improvement on the Test Stand

The existing test stand in the laboratory presents a challenge: when a robot impacts a pendulum, the pendulum swings back and hits the robot. A more substantial collision performed will obviously have adverse effects on the robot. Therefore, a braking system has been designed, developed, and implemented to prevent the pendulum from swinging. This system will enhance the quality and functionality of the test stand. It will imply more potent collision tests, which is essential for covering more of the spectrum of potential impacts, in addition to more precise results and fewer outlier captures (especially in the long run) yield more accurate results and reduce the outliers, particularly beneficial over the long run, allow for quicker execution of tests, reduce expenses by minimizing damage to the robots, and provide pendulum control.

3.4. Analysis of Requirements for a Suitable Solution

This thesis will discuss the obstacles of building a braking system on a test stand (2.1), where a robot impacts a pendulum. This pendulum must be prevented from swinging. As mentioned in the introduction, this thesis utilizes the VDI-2222 engineering guidelines. Section *3.3.2 Requirements and Main Tasks, Secondary Tasks* [14] demonstrates an essential step in the process, which is to determine the requirement of a problem that needs to be solved and ensure that the final solution meets not only functional specifications but also safety and reliability.

The requirements serve as the blueprint for the entire project, guiding us through each stage of development from conception to execution. By designing and implementing according to the predetermined requirements, we can organize each stage of development and avoid making unnecessary decisions that lead to confusion with the perspective and objective of the solution. The determination process of requirements is a foundational step in problem-solving, where analysis and understanding of the problem are crucial to designing the solution.

The table *Requirement List* (3.1) presents the requirements this thesis determined according to the *Verein Deutscher Ingenieure (VDI).*

As noted in figure (3.1), the requirement list categorizes requirements into three types found in the third column, "Type of Requirement". The categories are defined as follows:

- **Mandatory**: These are essential requirements that cannot be substituted or ignored.
- **Target requirement**: Important requirements that need to be fulfilled. However, alternative solutions are also permissible.
- **Optional**: Non-essential requirements that can be disregarded if necessary.

In addition, it has a "Supporting Documents", column (number 4) that refers to the source for "Descriptive Information Qualitative" and "Quantitative Numerical Information". These are represented as:

- **Test-Requirement**: Indicates fixed values or team preferences that are not essential to modify.
- **Optional-Requirement**: Chosen by the team for potential benefits to the project, these can be disregarded if deemed impractical.
- **CUI Devices AMT133Q-V**: Referenced in [4]
- **National Instruments**: Referenced in [16]
- **Doc ROS LabView**: Referenced in [2]

Column number six, titled "Possible weighting factors (1-5)", evaluates the team's perceived importance of each requirement for the existing test stand. In column number 5, titled "person", these are the reviewers for these requirements. Finally, the requirements and their values are detailed in the sections "Descriptive Information Qualitative" and "Quantitative Numerical Information".

3.4.1. Short Details of the Requirements

1. **Motion Detection capability**: It is a crucial component, as the system requires the capability to detect motion to a specific velocity. The threshold is set at $v\,m/s$.
2. **Motion-Detection Device**: It is also an essential component of the list. It is vital to possess the detection capability for any motion of the pendulum. The CUI Devices AMT133Q-V should fulfill the requirement as it has resolutions up to 4096 PPR.
3. **Control Unit**: It is hardware where it can govern the motor device that should control the caliper. It is capable of real-time control, allowing instantaneous adjustments to maintain stability and desired motion controls by sending signals and adjusting power levels sent to the motor.
4. **Hold**: The braking system must be able to hold up to 20kg for a pendulum length of 85cm. These specifications are crucial for selecting the motor that fulfills this requirement.
5. **Disc Radius Limitation**: In the existent test stand, the pendulum hung on a mechanical spline connected to a two-column as the figure (2.1) shows. The height of the mechanical spline can be adjusted. Therefore, if it is adjusted to the upper limit, the disc will limited to a 27cm radius due to the frame of the test stand.

6. **Shaft**: The mechanical spline shaft already exists on the test stand. It has a diameter of $14\,mm$ on the sides and $20\,mm$ in the middle.
7. **Optimal Location**: One requirement is to securely hold the pendulum. Consequently, a question arises: where is an optimal position to activate the brake? In conclusion, the best location to hold the pendulum is within a range where its velocity approaches zero. At this point, minimal force is required to stop the pendulum. The primary objective is to prevent the pendulum from swinging, which can be achieved by countering the force of gravity and the mass. Note that the robot might continue moving. Therefore, activate the brakes and hold the pendulum as far as possible from the start position.
8. **Prevent Over-Swing**: An issue arising from the proposed ideal position solution is the potential for over-swinging if the impact of the robot is too powerful. The system continuously monitors and analyzes the pendulum's position to prevent this event. If the pendulum exceeds the allowed angle, which is 135°, the system must promptly activate the brake mechanism to prevent further swinging.
9. **Controlled Return to Start Position**: After the brakes are activated and feedback is provided, the braking system must, under controlled conditions, return the pendulum to its start position.
10. **Human Feedback**: The pendulum can return to its starting position if the robot is not obstructing its path. This requirement is mandatory to determine when the braking system can begin releasing the pendulum. Without communication between the braking system and the robot, such as feedback on the robot's position, there is no mechanism to ascertain whether the robot is in the pendulum's path.
11. **Robot Feedback**: An optional requirement is to have a robot feedback providing its position or release permit.
12. **Communication with Robot**: Robot communication is facilitated through the LabVIEW ROS connection. This communication enables the robot to receive real-time data. Such communication could prove invaluable for the operation of a fully automated system.
13. **Control Algorithm**: To be able to meet certain requirements, an algorithm needs to be developed to analyze the incoming data and make the appropriate decision. For instance, when to activate the brakes or how to release the pendulum in a controlled condition to its start position.
14. **Main Energy Source**: The system requires an energy source to power its hardware components. This energy source operates on 220 volts of electricity.

Structure Part	Requirements Section			Access Section			Appendix
Information Gain	Descriptive Information Qualitative		Quantitative Numerical Information	Type Of Requirement	Source: Supporting Documents	Source: Personen	Possible weighting factors (1-5)
	Nr.	1	2	3	4	5	6
Requirements	1	**Motion Detection Capability**: Enable the detection of motion or impact on the pendulum using a highly sensitive motion detection device	v = 0.2 m/s	M	Test-Requirement	Robin Kirschner	5
	2	**Motion Detection Device**: CUI Devices AMT133Q-V	Resolutions up to 4096 PPR	M	CUI Devices AMT133Q-V	Robin Kirschner	3
	3	**Control Unit**: Interface between computer and motor	Real-Time Control	T	National Instruments	Robin Kirschner	3
	4	**Hold**: Hold the mass m with its length ℓ	m = 20 kg, ℓ = 85 cm	M	Test-Requierment	Robin Kirschner	5
	5	**Disc Max-Radius**: Limited from exiting structure	27 cm	M	Test-Requierment	Robin Kirschner	5
	6	**Shaft**: To build the break system	14 mm, 20 mm	T	Test-Requierment	Robin Kirschner	3
	7	**Optimal Location**: To ensure the robot avoids colliding with the pendulum again(if it continue moving), it should be stopped at its highest point of swing.	Velocity-Approximation = 0	O	Optional-Requirement	Robin Kirschner	2
	8	**Prevent Over-Swing**: Pendulum did not reaches velocity 0 before degree Θ ➡ stop it before over-swing.	Θ_{max} = 135°. Torque τ = $\ell(-mg)sin\theta$	T	LibreTexts Physics	Robin Kirschner	4
	9	**Controlled Return to Neutral Position**: Release the pendulum carefully to ensure it doesn't collide with the robot upon release	x_0 (Start position)	T	Test-Requierment	Robin Kirschner	4
	10	**Human Feedback**: Release or not	Boolean	M	Test-Requierment	Robin Kirschner	5
	11	**Robot Feedback**: General status feedback to analyze	Status: robot configuration, position or boolean.	O	Test-Requierment	Robin Kirschner	2
	12	**Communication with Robot**: To be able to receive data and send command	LabView ROS connection	O	Doc ROS LabView	Robin Kirschner	2
	13	**Control Algorithm**: An algorithm that analyze the incoming data and makes the appropriate decision	The decision faster than the return of the pendulum	M	Test-Requierment	Robin Kirschner	5
	14	**Main Energy Source**: Electricity	220v	M	Test-Requierment	Robin Kirschner	5

> Type Of Requirement: M - Mandatory,T - Target Requirement, O - Optional

Figure 3.1.: Requirement List

4. Solution Design

Human safety has become a crucial subject in robotics, so its development demands precise data for safe HRI. As we mentioned before, the effective robot mass is a primary parameter among others that contributes to the potential injury severity [8, 11]. On the other hand, the robot manufacturers do not provide sufficient data to obtain an adequate value of effective mass [15]. To fill the gap in the sufficiency of the data, a rudimentary test stand with a pendulum was engineered and developed to resolve it.
This thesis contributes to the engineering and development of the test stand by designing an automated braking system that prevents the pendulum from swinging during the testing. This step is crucial to improve the reliability and ease of the testing process.

4.1. Objectives and Structure

The primary goal is to design and implement an automated braking system for the test stand using a servo motor and a caliper with a bike disk brake. These parts are controlled by a computer that calculates the data flow from the sensors installed and takes necessary action. The solution design is structured as follows:

- **Functional Structure**: This section outlines the main components and their functions within the automated braking system. In addition, it provides a clear understanding of how the system operates and what each part contributes to its functionality.
- **Conceptual Solution**: This section discusses the various solutions considered during the development process and determines the most suitable solution for implementation using rating criteria. This approach ensures that the chosen solution aligns with the project's requirements (3.1) and constraints.
- **3D Modeling**: This section demonstrates the use of 3D Modeling to overcome obstacles encountered during the design and implementation process. It showcases how the physical components were modeled to fit together and function effectively.
- **Software Solution**: Finally, this section explains the software used to program and control the automated braking system and what programming languages, software platforms, and control algorithms are employed to manage operations and responses from and to hardware.

4.2. Functional Structure

The functional structure, in general, according to VDI 2222, is a crucial step in the design process, where the functions of a system are identified and organized in a hierarchical manner, which helps to understand the functionality and the contribution of each part of the system. It has three main elements that categorize the types of interactions and transformations occurring within a system [14]:

- **Storage**: It refers to the physical substances manipulated, transformed, or moved within the system.
- **Energy**: It refers to the power required to drive the system's functions, e.g., sources and energy transformations that enable the system to perform.
- **Information**: It refers to the data and signals processed, transmitted, and used to control the system.

In the thesis, we applied the functional structure to design the braking system for the test stand, as shown in Figure (4.1).

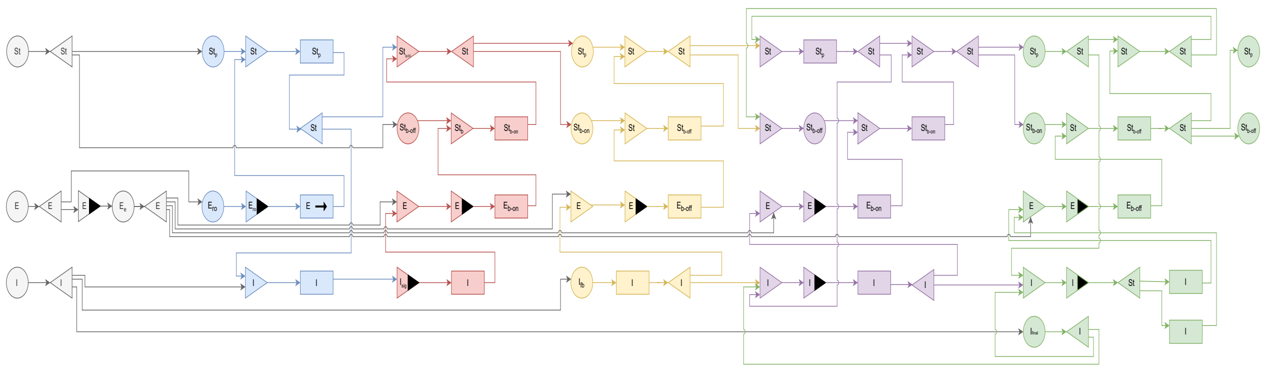


Figure 4.1.: Functional Structure

The functional structure of the braking system is designed to be functional and easy to understand. Therefore, it is divided into five phases, where each phase represents an event in one test. The five phases are as follows:

1. **Initial-States Phase**: This phase represents the source of the raw material for the three main elements, which are Material, Energy, and Information, that the system requires. For instance, the Material element includes the robot, pendulum, all brake parts, etc.
2. **Robot-Pendulum-Accident Phase**: This phase represents the initial movement of the robot and its collision with the pendulum, which causes the pendulum to start moving.
3. **First-Break-Reaction Phase**: This phase occurs when the braking system takes its first action to stop the pendulum from swinging back.
4. **Feedback-After-Accident Phase**: This phase occurs when the braking system waits for feedback to release the pendulum to its neutral position.
5. **Release-Pendulum Phase**: In the final phase, the functional structure demonstrates how the braking system releases the pendulum in a controlled condition.

To provide a detailed explanation of the functional structure, each phase will be explained separately to ensure a comprehensive understanding of how the braking system functions. As an introduction to the functional structure, Figure (4.2) illustrates the functionality of each individual function:

GENERAL OPERATIONS	MATERIAL	NR.	SYMBOL	FUNCTIONALITY
STORAGE	Storage	1	St	Source of all available substances
	Energy	2	E	Source of all available energy
	Information	3	I	Source of all available information
CONDUCT	Storage	4	St	Transfer substances to the destination
	Energy	5	E	Transfer energy to the destination
	Information	6	I	Transfer information to the destination
CONVERT	Storage	7	St	Convert storage to another form
	Energy	8	E	Convert energy to another form
	Information	9	I	Convert information to another form
SUMMATIVE	Storage	10	St	Connect multiple substances or a substance with other materials
	Energy	11	E	Connect multiple energies or energy and other materials together
	Information	12	I	Connect multiple pieces of information or information with other materials
DISTRIBUTIVE	Storage	13	St	Distribute substances to multiple destinations
	Energy	14	E	Distribute energy to multiple destinations
	Information	15	I	Distribute information to multiple destinationsĺ

Figure 4.2.: Function Details: Explanation of the purpose of each individual function

Initial-States Phase

In the initial state phase (4.3), all the raw materials are ready for use. The *Storage - St* represents all available material elements involved in the process. The elements in the braking system are the robot, pendulum, and brakes, which have two states: on and off.

The *Storage - E*, serving as the energy element, represents the energy source accessible to the system. Within the braking system, electricity storage E_e is conducted from the initial main source energy storage E.

The *Storage - I*, which is the information element, like the others, represents all the information needed to control the system. These are the signal, which is generated from the moving pendulum, the first feedback I_{fb}, and the last feedback I_{final}

These raw materials will be distributed as the triangle in Figure (4.3) presents. Each triangle faced to the left is a disruption function that serves the element as input. The element will be noted inside it.

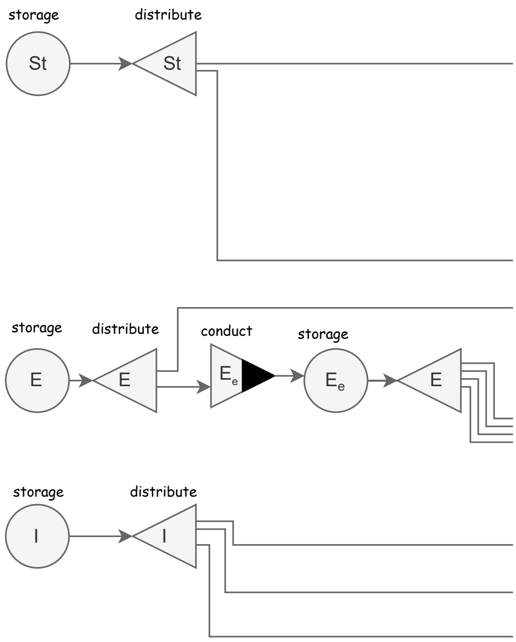


Figure 4.3.: Initial-States Phase

Robot-Pendulum-Accident Phase

In this phase (4.4), the material is represented by standing still pendulum storage St_p and the energy as storage E_{ro}, which is the moving robot as the available mechanical energy supplied from storage E in (4.3). The pendulum is in its neutral position as free mass, where no force affects it except gravity. When a robot impacts the pendulum, it represents the transmission of the mechanical energy from the moving robot to a standing still pendulum, where it initiates motion after the collision, and other forms like sound [9].

After the foremost energy is transmitted to mechanical energy to the standing still pendulum, which initiates swinging, a motion sensor triggers a signal that is transmitted and interpreted.

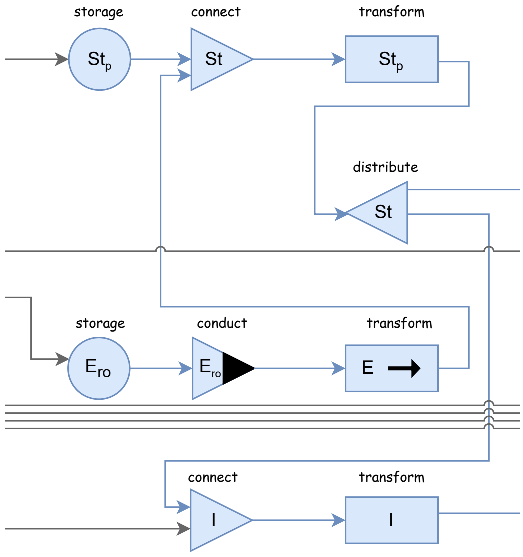


Figure 4.4.: Robot-Pendulum-Accident Phase

First-Break-Reaction Phase

In this phase (4.5), I_{sig} will conduct the information signal received from the motion sensor. It is transmitted and connected with electricity storage E_e from (4.3). This connection allows the servo to impact the brakes St_{b-off}. The St_{b-off} represents open caliper brakes, which means the caliper is not initiating any force to the disc. After the energy is transmitted to the St_{b-off}, the St_{b-off} starts to move and initiate force on the disc. The force initiation is represented by St_{b-on}, which means the brakes St_{b-off} transforms from off toward on.

The St_{b-on} is connected to the moving pendulum to prevent it from swinging. The blue arrow is the moving pendulum from (4.4) and the red arrow from St_{b-on} from (4.5).

Feedback-After-Collision Phase

In this phase (4.6), St_p represents the pendulum at rest, held by the brake mechanism. The incoming red arrows on St_p and St_{b_on} originated from (4.5) as an outcome from the last event. The I_{fb} represents the feedback information provided to the system to start releasing the

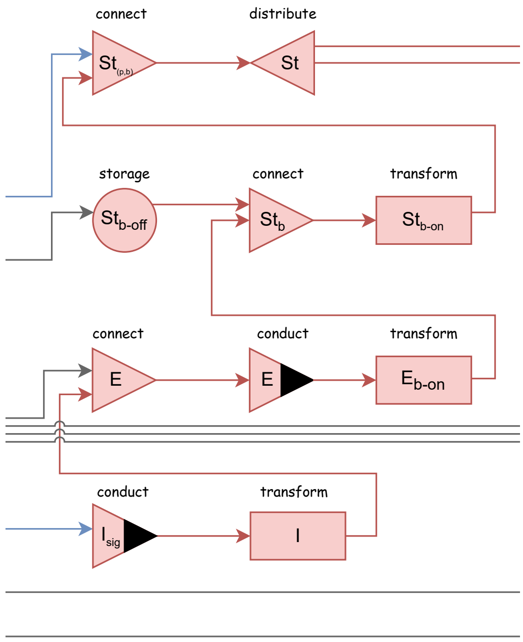


Figure 4.5.: First-Break-Reaction Phase

pendulum in a controlled condition to its neutral position. This feedback is transmitted and connected with the E_e energy from (4.3) to trigger the servo to reduce the force from the caliper on the disc. The caliper St_{b-on} is in the stillness, not moving initiating force to the disc. After the servo releases the caliper, the caliper starts moving, which is represented as St_{b-off} rectangular in Figure (4.6). The moving caliper St_{b-off} toward the "off" state releases the pendulum, and the pendulum initiates motion toward its neutral position caused by gravity.

Release-Pendulum Phase

The release-pendulum phase denotes the period in which the pendulum regains its neutral position in a controlled manner. The function structure in Figure (4.7) shows a moving pendulum represented by the purple rectangle St_p. This moving pendulum generates signal information from the encoder connected with the information feedback from the Feedback-After-Collision Phase (4.6) and information I_{final}. I_{final} represented in green circle in (4.7). This information is connected and transmitted in the following triangle to analyze and determine the subsequent course of action.

The yellow arrow is the information initiating the beginning of the release phase. The green arrow is a manual feedback given when the loop is terminated. The loop refers to the repetitive cycle of braking and releasing the pendulum within the release-pendulum phase. The manual feedback should given when the pendulum arrives at its neutral position and rest. The purple arrow is the encoder feedback that indicates whether the pendulum is in motion during the moving phase of the release-pendulum phase or not.

If the green arrow inputs are positive, the loop must be halted. If negative, it is disregarded. In the case of positive purple input, indicating pendulum motion, action is required. Otherwise, the negative input is disregarded.

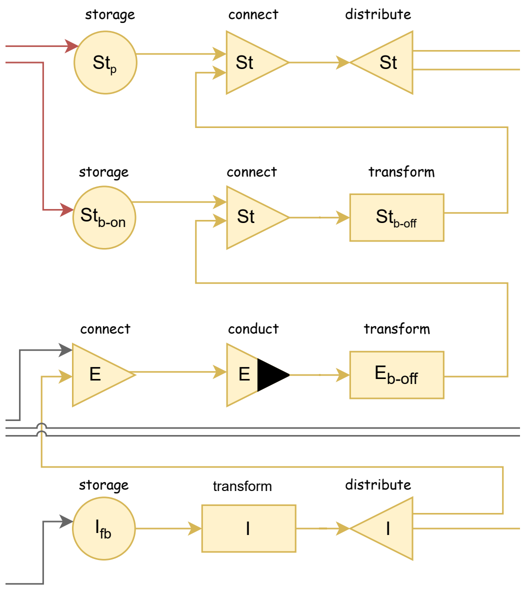


Figure 4.6.: Feedback-After-Collision Phase

In case a brake is required, then the electricity will be connected with the purple circle St_{b-off} to activate the brake by connecting the purple rectangle St_{b-on} and the pendulum in motion to hold it. The green input arrows on the connect purple triangle state are the feedback from the next part of the phase, where if the pendulum is still not in its neutral position as defined in the requirement list (3.1), then the pendulum considered still moving. If the pendulum is within the neutral position range, even in motion, it is regarded as a state where braking is unnecessary.

In the green area, the I triangle connects the feedback from the storage St_p and the feedback I_{final} in addition to the purple input, which indicates the state where the phase of braking has been taken. The barking phase in the release-pendulum phase is defined as the part of the loop where the brake should be turned on. The other part is when the brake is released.

If the feedback I_{final} is positive, then after the information is conducted, it will activate the stop-looping path to transmit information to stop the action of the brake by releasing it and stopping the cycle. If the input is negative, it follows the path to connect with energy, transmitting it to the storage unit St_{b-on} to release the brake, allowing the pendulum to continue swinging back.

Finally, suppose the pendulum reaches its predetermined neutral position and receives the finished feedback. In that case, both the pendulum St_p and the brake St_{b-off} will settle into their respective green circle states.

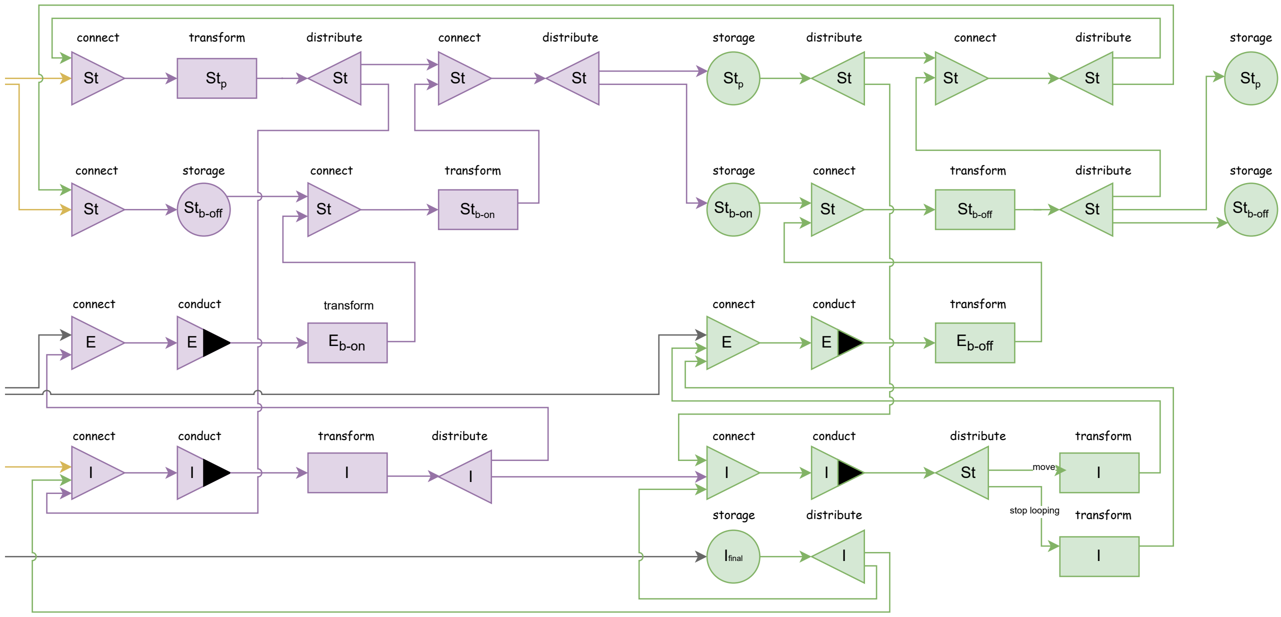


Figure 4.7.: Release-Pendulum Phase

4.3. Conceptual Solution

The conceptual solution in the context of VDI 2222 involves systematically developing and evaluating potential solutions to a given engineering problem. Also, it is a step toward a solution. As this thesis mentioned before, the VDI 2222 is a guideline for designing and developing a structured approach to problem-solving. Thus, this section describes the conceptual solution for the automated braking system of an existing pendulum test stand.

In the following, this thesis will present the conceptual solution table with a combination of potential solutions. After that, this combination will be evaluated using predefined criteria. The criteria are performance, compatibility, reliability, complexity, availability, efficiency, and cost.

4.3.1. Potential Solutions

The dotted line is the connection among different nodes. Each node represents a partial solution for a function. For instance, the *information conduct* in the Figure (4.8) has three possible partial solutions. Each part is considered a node. From one node, we can find a path to another node. The visited nodes on a path from the top to the bottom are considered the combination of a possible solution. Different paths mean different solutions. To select the best possible solution, an evaluation system will be followed to choose the best assessment solution.

Next, the thesis will detail the evaluation system, where each criterion is rated on a scale of 1 to 5, offering five different rating levels. The criteria are as follows:

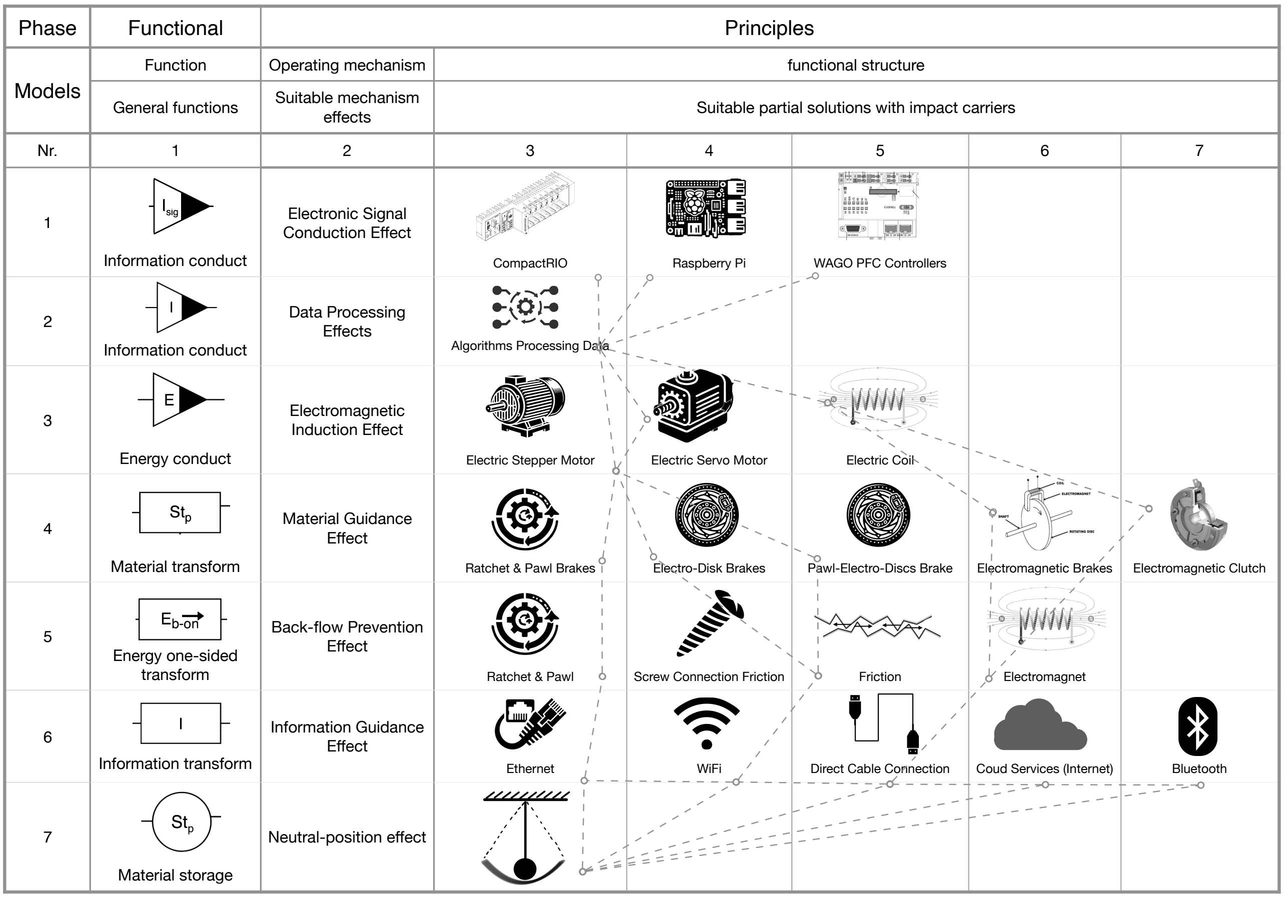


Figure 4.8.: Conceptual Solution. References for the icons used are as follows: (1,3) in [7], (1,5) in [5], and (4,6) in [12]

- **Performance**:
  - **1.0**: Very weak, does not meet all the mandatory requirements.
  - **2.0**: Insufficient, falls short of the mandatory requirements.
  - **3.0**: Marginal might be adequate but with little room for exceptions.
  - **4.0**: Satisfactory, meets the requirements but may be overpowered.
  - **5.0**: Excellent, perfectly matches our performance needs.
- **Compatibility**:
  - **1.0**: Completely incompatible, unsuitable for our use case.
  - **2.0**: Potentially compatible, but major infrastructure changes required.
  - **3.0**: Compatible with our infrastructure but needs customization and proxy.
  - **4.0**: Compatible with our infrastructure, it requires integration effort.
  - **5.0**: Seamless compatibility, no additional integration required.
- **Reliability**:
  - **1.0**: Unreliable, unusable.

  - **2.0**: Somewhat reliable, not dependable enough.
  - **3.0**: Moderately reliable.
  - **4.0**: Reliable but with some safety concerns in exceptional cases.
  - **5.0**: Highly reliable for our specific use case.
- **Complexity**:
  - **1.0**: Extremely complex, potentially unsolvable.
  - **2.0**: Complex and challenging for our use case.
  - **3.0**: Moderately complex, but time constraints limit exploration.
  - **4.0**: Not overly complex but requires integration and optimization time.
  - **5.0**: Simple, quick integration and implementation.
- **Availability**:
  - **1.0**: Unavailable, no production options.
  - **2.0**: Unavailable, only customized production possible.
  - **3.0**: Unavailable due to supply chain issues or long delivery times.
  - **4.0**: Available with moderate delivery times.
  - **5.0**: Readily available with short delivery times or already in our lab.
- **Efficiency**:
  - **1.0**: Highly inefficient given its capabilities.
  - **2.0**: Inefficient with some capability.
  - **3.0**: Inefficient but capable.
  - **4.0**: Efficient but slightly less capable.
  - **5.0**: Highly efficient with good capability.
- **Cost**:
  - **1.0**: Extremely expensive and not justifiable.
  - **2.0**: Very costly, too expensive for practical use.
  - **3.0**: Expensive relative to its capabilities.
  - **4.0**: Not overly expensive considering its capability and quality.
  - **5.0**: Cost-effective due to both capability and quality or provided at no cost.

4.3.2. Rating Potential Solutions

The potential solution discussed in the previous section will undergo evaluation. Six possible combinations of solutions have been selected within the morphological chart (see Figure 4.9). Rows 1-7 indicate the average rating for each element in the solution. In the last row, *Final Rating (Normalized)* indicates the average rating of a solution normalized.

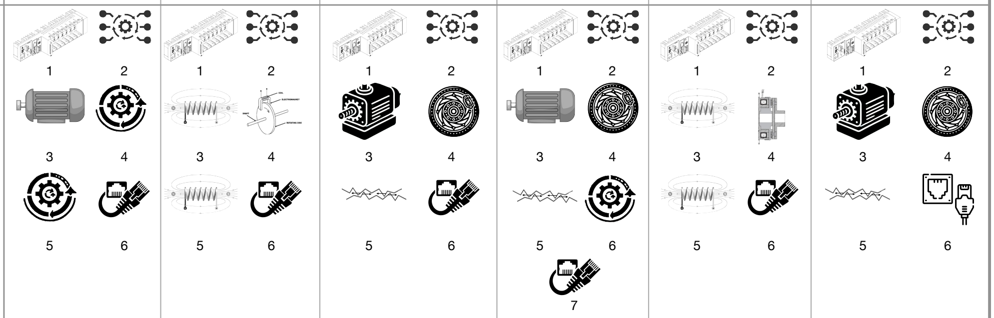
Solutions / Part-Solution	Nr.	Overall solutions composed of effect carriers principle sketches					
		1	2	3	4	5	6
Design Principle Sketch	-						
Electronic Signal Conduction, Data Processing, Electromagnetic Induction, Material Guidance, Back-flow Prevention, Information Guidance ,Neutral-position	1	4.62	4.62	4.62	4.62	4.62	4.62
	2	5.00	5.00	5.00	5.00	5.00	5.00
	3	4.29	2.29	4.29	4.29	2.29	4.29
	4	4.44	2.37	5.00	3.99	2.76	5.00
	5	4.53	2.36	5.00	5.00	2.36	5.00
	6	5.00	5.00	5.00	4.53	5.00	4.77
	7	-	-	-	5.00	-	-
Final Rating (Normalized)	-	4.65	3.61	4.82	4.63	3.67	4.78

Figure 4.9.: Morphological Chart: Refer to (4.8) for the icons references

In the figures' tables labeled as Energy One-Sided Transform (A.5), Energy Conduct (A.3), Information Transform (A.6), Information Conduct Data (A.2), Information Conduct Signal (A.1), and Material Transform (A.4), every individual element has been meticulously evaluated, and their average ratings have been precisely computed. These average values have been recorded in the column titled *Final Rating*. These averages serve as the ratings referred to in (4.9).

The morphological chart contains six solutions, as shown in columns 1 to 6 (refer to Figure 4.9). According to the guideline provided in VDI 2222 [14], the top three solutions need to be identified for further consideration. Based on the evaluation, the best three solutions, with ratings of 4.65, 4.82, and 4.78, are located in columns one, three, and six, respectively. These solutions will be advanced to the next stage for further assessment.

After selecting the best three promising solutions, it is time to start the construction of the best-rated solution. During the construction, problems may appear that can only be identified after the beginning, e.g., problems with a supply chain. Sometimes, these problems can not be identified in advance, and there is no solution in the near future. Thus, the next best-rated solution will be selected and proceeded to construction.

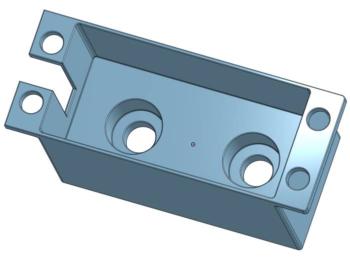
In this thesis, the solution noted in column 3 of the Figure (4.9) was the highest-rated one. It proved successful, with the construction process going smoothly and only encountering minor issues. We were able to solve these problems by using 3D Modeling for some components (refer to section 3D Modeling 4.4). The solution consists of the following: the control unit CompactRio (refer to Hardware Component 5.2.1), a software solution for the control algorithm (see section Software Solution 4.5), a servo motor (refer to Hardware Implementation 5.2.1), a bike disc brake with a caliper (refer to Mechanical Implementation 5.1), friction as energy conduction, and a direct cable connection to the computer.

4.4. 3D Modeling

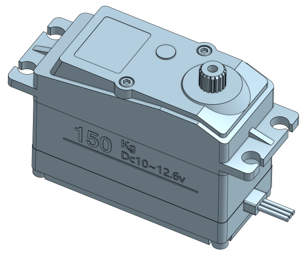
3D Modeling is a development step in mechanical design that is considered a solution to an obstacle faced by assembling the best-rated selected parts of the braking system. 3D Modeling is not obligatory according to a specific guideline VDI 2222 [14] in the or ISO 15066 [8] but remains an optional step that can be pursued if required. It serves as a tool for visualizing and refining solutions to complex engineering problems. This step is particularly valuable when assembling components from various sources to create a prototype, such as the braking system discussed in this thesis. 3D Modeling allows engineers to simulate the assembly process, identify potential fit or interference issues, and ensure that all parts work harmoniously together. This thesis encountered two problems with the mechanical solution design, and it will present the solution in the following subsections. All the designed parts were ordered as aluminum material for greater resilience and durability.

4.4.1. Servo Mounting Frame

The servo motor (4.10b) must be mounted on the right side of the test stand, and it is shaped like a cuboid. The servo has a dimension measuring 30 $mm \times 31\ mm \times 65\ mm$, along with an additional frame section where four screws with their nuts secure it. Besides the servo, there is also a mounted frame (4.10a) that is constructed from scratch using *OnShape* software as a tool. It has four pores to fasten the servo and two extra aperture for $M8$ cylinder screws to mount it to the side of the test stand. The outer dimensions of the mounting frame shown in Figure (4.10a) are as follows: it has a height of 45 mm, a width of 33 mm, and a length of 85 mm. At the bottom, there are two apertures with different diameters, 15 mm and 5.5 mm, designed to accommodate screws discreetly and allow the servo slide to fit perfectly in place. Additionally, there are four screw pores with a diameter of 6 mm to securely fasten the servo to the mounting frame. The base of the frame has a thickness of 15 mm, providing ample space to conceal the heads of the$M8$ cylinder screws.



(a) Servo Mounting Frame



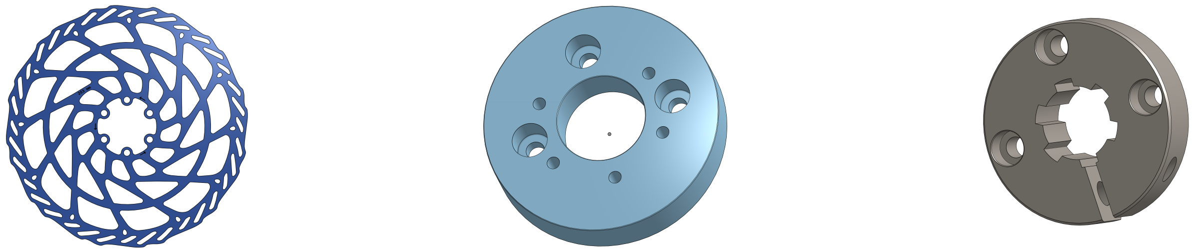
(b) Servo Motor [18]

Figure 4.10.: Servo Mounting Frame and Servo Motor

4.4.2. Servo Mounting Frame

The disc brake must be mounted to the spline shaft, as the pendulum is also mounted to it. When the spline shaft stops, the pendulum stops. Therefore, the disc brake must be fastened to the spline shaft. When the caliper holds the disc, the shaft stops moving. To achieve this, the clamping ring for the spline shaft slides around the shaft (see Figure 4.12a) as a solution since the disc brake does not fit the shaft and its shape. The disc brake will be mounted with the clamping ring to turn when the shaft turns. However, the issue with the clamping ring and the disc brake is that there are no specially designed parts available that can be ordered to fit both the clamping ring and the disc brake. Therefore, an interface has to be designed to adapt both the disc and the clamping ring. The disc adapter, as the Figure (4.11b), is the part that is set between the disc and the clamping ring so both can be mounted together.

The dimensions of the disc adapter are as follows: it has a diameter of 80 mm and a thickness of 14.6 mm. The five small holes are for mounting the disc, and the other three apertures are for fastening the clamping ring. The three apertures are designed with two different diameters, 12 mm and 6.3 mm, providing ample space to conceal the heads of the $M6$ cylinder screws.



(a) Disc Brake (b) Adapter (c) Clamping Ring

Figure 4.11.: Disc, Clamping Ring, and its Adapter

4.4.3. Final Assembly

The following figures are the final assembly that presets how the disc brake, adapter, and clamping ring should be assembled and mounted together in addition to the servo motor and its mounted frame.

(a) Disc Assembly (b) Servo Mounted Frame Assembly

Figure 4.12.: Assembly Components: Disc Assembly and Servo Mounted Frame Assembly

4.5. Software Solution

The software solution for the automated braking system is crucial for ensuring accurate control and response to the sensor data. The software manages the servo motor and caliper interactions to achieve the desired braking effect. The Software requirements are real-time processing of sensor data to detect pendulum motion and its velocity, control algorithms to activate and release the brake based on the pendulum's position and velocity, and safety checks to prevent overswing and ensure the pendulum returns to its neutral position. These software requirements can be achieved by LabVIEW, a graphical programming environment developed by National Instruments. This software can communicate with the servo motor through the interface control unit *CpmpactRio* (cRio) with *NI 9263*. The software will command the cRio NI 9263 to deliver x volt to the servo motor.

4.5.1. PWM Simulation, FPGA Target Interface, and Control Algorithm

Pseudo Algorithm - (FPGA)servoController.vi (5.3.1)

The servo motors used as caliper controllers work with PWM signals. PWM represents a signal as a rectangular wave. The NI 9263 delivers an analog signal, which is not the type of signal that servo motors will react to. Therefore, a PWM signal simulation has to be developed to replicate it. Thus, the thesis developed an algorithm to achieve the PWM signal.

Algorithm 1: Pseudo Algorithm for Simulating PWM Signal

```
Import global variable pulseWidth;
while true do
    if button clicked then
        exit while;
    send maxVolt volts to the servo;
    if pulseWidth in range 2200 - 2500 then
        sleep pulseWidth microseconds;
    else if pulseWidth > 2500 then
        sleep 2500 microseconds;
    else
        sleep 2200 microseconds;
    send minVolt volts to the servo;
    sleep Hz microseconds;
```

The $maxVolt$ and $minVolt$ values are the voltage levels needed to send to the servo (see subsection 5.2.1).

The $pulseWidth$ value must be in the range of 2200 to 2500 microseconds as a limit to the servo motor this thesis uses (see subsection 5.2.1). This value will determine the rotation of the servo, where 1500 is the middle of the rotation ability. 2200 is the minimum, and 2500 is the maximum. The ability of the servo is from 500 - 2500, but it is limited as there is no need to rotate a greater angle; if it rotates more, then it will try to pull the cable more, but there is no option to pull, resulting in damage to the servo.

The Hz value is the loop clock to simulate the Hz of the signal, where it is from 50-300 Hz (see subsection 5.2.1).

Pseudo Algorithm - servoChannelTargetReference.vi (5.3.2)

Algorithm 2: Pseudo Algorithm for Communicating with FPGA Target

Import global variables $pulseWidth, trigger$
while *true* **do**
    if *trigger* **then**
        exit while;
if *pulseWidth between 2200 and 2500* **then**
    (FPGA)servoController (pulseWidth);
else
    do nothing;

The servoChannelTargetReference.vi (5.3.2) is a file created to be an interface between the servoControl.vi (5.3.1) and the (FPGA)servoController.vi (5.3.3). When there is a need to brake, the pulseWidth of the servo must be changed. The arguments pulseWidth and true constant must passed to the servoChannelTargetReference to establish a connection with the (FPGA)servoController.vi and pass the pulseWidth. The initial value of the pulse width that considers the brakes are off is 2499 microseconds. The location and position of the servo motor determine this value.

The environment of servoChannelTargetReference.vi has two global variables: pulseWidth and trigger. Both are defined in different VIs. When the program is running, it will wait for the trigger. When the trigger is on, it will pass the pulseWidth after the validation to the FPGA-Target. The validation of the pulseWidth's value will determine if it is in the range of 2200 and 2500. If it is valid, the program passes the pulse width to the FPGA-Target; if not, it ignores the command.

Pseudo Algorithm - servoControl.vi (5.3.3)

This VI controls the braking system and its behavior. It initiates all the actions and the decisions. servoControl.vi is divided into three main parts, where each part is considered a phase. The first phase is the observation phase, where the program observes the motion of the pendulum. The second phase is the braking phase, where after the detection of the motion in the first phase, the second phase will initiate the braking to prevent the pendulum from swinging. This phase will end with feedback given to the program. The feedback marks the start of phase three, the release of the pendulum. This phase involves releasing the pendulum under controlled conditions.

The program utilizes four global variables: $velocity$, $force$, $trigger$, and $pulseWidth$. The $pulseWidth$ is defined and assigned within the program, while the others are defined and assigned in different VIs.

During the first phase, the program waits until the $trigger$ is true and the $velocity$ is greater than $minVelocity$. The $trigger$ indicates that motion has been detected. Action is required when the $velocity$ exceeds $minVelocity$ as per the requirement list (3.1).

In the second phase, the program initializes the $pulseWidth$ to 2200 microseconds. This value is determined based on various tests. If the value is smaller, the servo motor rotates in an attempt to pull the brake's cable further. However, due to physical limitations, the cable cannot be pulled any further. In this scenario, the motor will continue attempting to pull until it reaches the position of the pulseWidth. This event can damage the internal components of the servo motor.

Therefore, for the setup of the test stand and after several tests, the 2200 microseconds pulse width is determined. After the initialization, the program checks the validity of the $force$, where if the force is greater or equal to the $maxForce$, then this is an error case, where the robot continues to move further after the first collision. In this case, the baking system must release the brakes to prevent damage to the test stand and the robot. The release pulse width is 2500 microseconds. If the $force$ is valid, the system will activate the brakes to prevent the pendulum from swinging. When the brakes are active, the system waits for feedback to release the pendulum. In our system, the feedback is designed to be received from a human. When the human presses the "release-the-pendulum" button, then the program will continue to the third phase.

In the third phase, the release phase, the program observes the pulse width and the pendulum's velocity. If the $pulseWidth$ is greater than the maximum allowed pulse width, then it will be assigned with $maxPulseWidth$, and if it is less than the allowed $pulseWidth$, then it will be assigned with the $minPulsWidth$. In addition, if the $velocity$ is less than $minValue$, the program assigns 2 to the $case$ variable. If it is greater than $maxVelocity$, the $case$ variable is assigned 1. Otherwise, the $case$ is assigned 0. Next, the program has three cases:

- **Case 0**: No changes should be made. This occurs when the $velocity$ is between the minimum and maximum, indicating controlled movement.
- **Case 1**: The $velocity$ exceeds the allowed maximum, where a need to decrease the pulse width to increase the brake's force.
- **Case 2**: The pendulum's velocity is slower than allowed. Therefore, the pulse width must be increased to release the brake further.

In cases 1 and 2, the program calls the servoChannelTargetReference VI to pass the modified pulse width.

Algorithm 3: Pseudo Algorithm for Controlling the Pendulum

```
Import global variables velocity, force, trigger
Create global variable pulseWidth;
while do
    if velocity > minVelocity & trigger is true then
        exit while;
while do
    pulseWidth ← 2200;
    if force ≤ maxForce then
        servoChannelTargetReference (true, pulseWidth) /* brake */;
    else
        servoChannelTargetReference (true, 2500) /* don't brake */;
    if return-to-neutral-position = true then
        exit while;
while do
    if pulseWidth greater than maxPulseWidth then
        pulseWidth ← 2500;
    else if pulseWidth less than minPulseWidth then
        pulseWidth ← 2200;
    if velocity between minVelocity2 and maxVelocity then
        case ← 0;
    else if velocity > maxVelocity then
        case ← 1;
    else
        case ← 2;
    if case = 0 then
        do nothing;
    else if case = 1 then
        pulseWidth ← pulseWidth * 0.995;
        servoChannelTargetReference (true, pulseWidth);
    else
        pulseWidth ← pulseWidth * 1.001;
        servoChannelTargetReference (true, pulseWidth);
    if finish = true then
        exit while;
```

5. Implementation

Implementing the automated braking system for the test stand involves mechanical and software components. This section details the process and methodologies for developing and integrating these components. This thesis will divide the implementation into three sections. The first section is the mechanical implementation, where all the mechanical parts are installed and mounted together. The second section is the hardware implementation. In this section, the thesis shows how the hardware components interconnect and communicate together. The last section is the software implementation, which shows how the algorithms are implemented and what software is used to accomplish it.

5.1. Mechanical Implementation

The mechanical implementation focuses on integrating a servo motor and a caliper with a bike disk brake to effectively control the pendulum's swinging. The main components include:

- **Bike Disc Brake**: G2 CleanSweep, 6-Loch, 200 mm
- **Caliper**: SRAM BB7

The servo motor, caliper, and disc brake are mounted on the same level as Figure (5.1) presents. When the servo motor's outer gear rotates, it pulls the cable, causing the caliper to close on the disc. This action stops the disc, which implies that the pendulum is prevented from swinging.

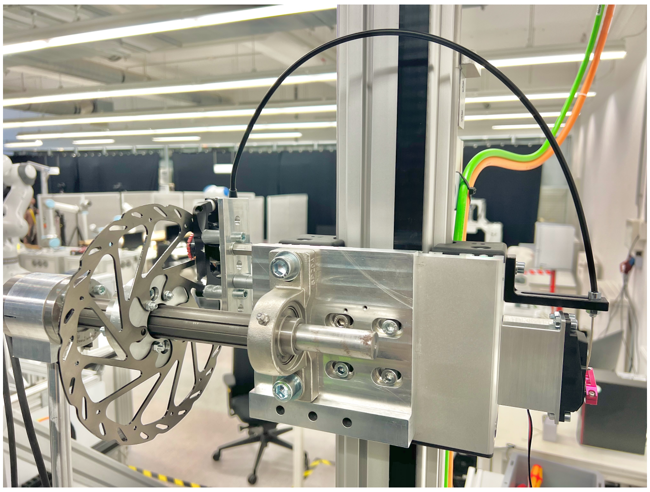


Figure 5.1.: Assembly

5.2. Hardware Implementation

CompactRio (cRio) NI 9263 is a real-time controller that functions as an interface between the computer and the servo motor. The NI 9263 is an extension module of the cRio system, primarily responsible for outputting analog signals for real-time updates. However, as previously mentioned in this thesis, servo motors cannot directly receive and interpret these analog signals. They require a PWM signal. For a detailed explanation of how this issue is resolved, please refer to chapter *Solution Design* section *Software Solution - 4.5.1. Pulse-Width Modulation (PWM) Signal Simulation*(4.5.1).

5.2.1. Hardware Component

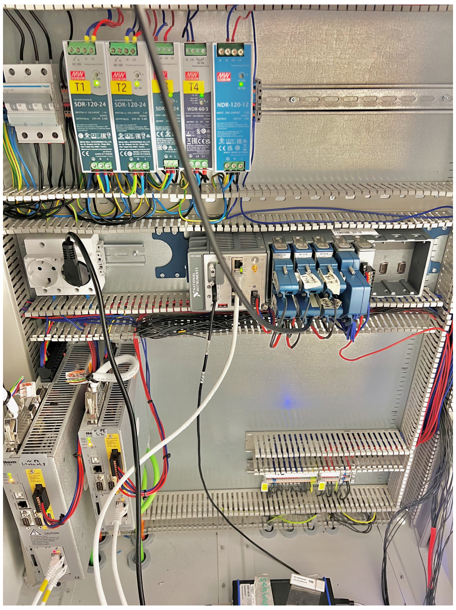
In Figure (5.2a), there are two main hardware components that are crucial for the braking system: the cRio and its extension NI 9263 and the MEAN WELL power supply NDR-120-12.

- **Servo Motor**: The model number is DS51150, operating at 12.6V, generating a torque of 173 kg-cm (approximately 16.97 Nm). The control system utilizes PWM with a pulse width range of 500 to 2500 microseconds pulse width. The definition of the neutral position is the pulse width of 1500 microseconds. Additionally, it operates within a frequency range of 50 to 330 Hz [10].
- **cRio and NI 9263**: It is the real-time control unit and its extension, the NI 9263. The cRio is connected to the computer, where it can send commands to the servo motor through the NI 9263.
- **Power Supply**: The manufacturer is MEAN WELL, and its model number is NDR-120-12. Its main functionality is to receive 230 volts and output 12 volts.

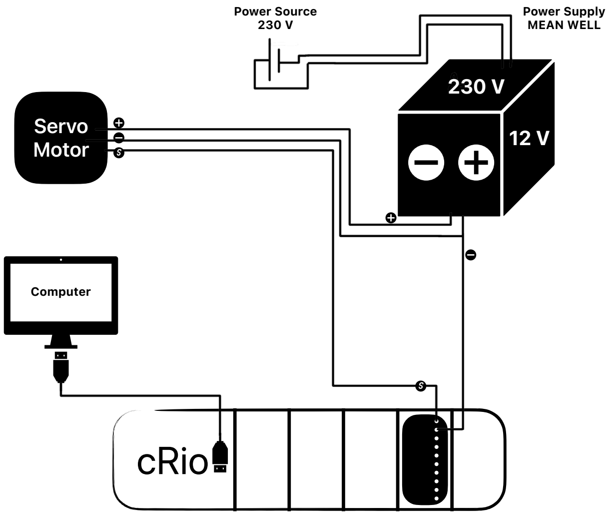
The components are interconnected to construct the electrical circuit that supplies electricity to the servo motor and connects it to the cRio. In the next section, the thesis will explain how the hardware collaborates to enable computer control of the servo motor.

5.2.2. Coherence Across Hardware Components

The Figure (5.2b) presents an abstract and simplified circuit overview of the interconnection among the hardware components of the Figure (5.2a). The servo motor is supplied with the required electricity and operates on a 12-volt power supply from the MEAN WELL power source. It is connected to the power supply via positive and negative cables. The third cable is a signal cable that commands the servo through a PWM signal (see servo motor 5.2.1). The cRio sends the commands through NI 9263 and is connected to a computer that controls the commands. The servo motor's signal cable is connected to the NI 9263 AO0 output, where the computer commands the cRio through NI 9263 AO0 to send signals to the servo. Although the NI 9263 is an analog module and sends an analog signal to the servo motor, the servo motor can only interpret the PWM signal. Refer to Section Software Solution (4.5) to understand how this issue is resolved.



(a) Electrical Circuit



(b) Simple Circuit

Figure 5.2.: Electrical and Simple Circuits

5.3. Software Implementation

The software implementation for the automated braking system encompasses the development of algorithms and control programs that manage the interactions between the various hardware components. The software used is LabVIEW, a graphical programming environment from National Instrument that produces the cRio. In this section, the thesis will present the programs and the algorithm that control the braking system. The program is divided into three parts, where each part is a VI:

- **(FPGA)servoController.vi**: The FPGA target file is created on the NI-cRIO-9045-01F88613 inside the Chassis (cRIO-9045) with the name (FPGA)servoController.vi. This program is responsible for simulating the PWM signal sent to the servo. It is connected to Mod5/AO0, where Mod5 is the NI 9263 module, and AO0 is the output signal connected to port 0 (see figure 5.2b).
- **ServoChannelTargetReference.vi**: It is located on the NI-cRIO-9045-01F88613. This program acts as an interface, connecting the main control program with the FPGA target. It facilitates communication between (FPGA)servoController.vi and servoControl.vi. Any read or write operations to the FPGA target are executed through ServoChannelTargetReference.vi.
- **servoControl.vi**: In this file, the control algorithm of the braking system is implemented.

5.3.1. (FPGA)servoController.vi

In this program, the algorithm (1) is implemented to simulate the PWM signal connected to the servo motor, as explained above. Firstly, a while loop is created, as shown in Figure (5.3). This

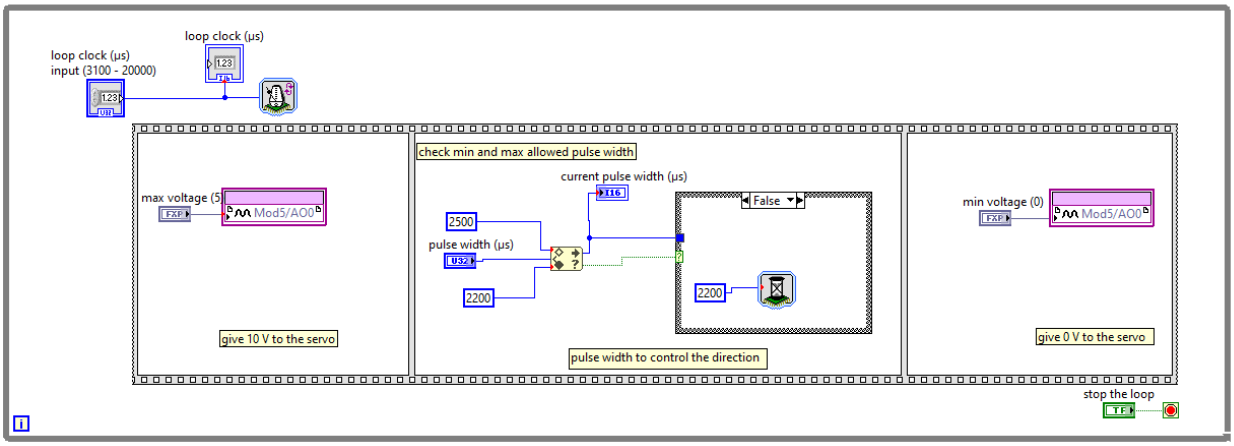


Figure 5.3.: FPGA Control Algorithm

loop is programmed to operate at a frequency ranging from 50 to 330 Hz, the frequency range the servo can operate within. Then, a flat sequence is used to organize which operation must be executed and which must wait until the others are finished. The flat sequence is divided into three sequences. The first sequence passes the argument max voltage to the Mod5/AO0, where the servo motor is connected, and then we send the pulse, depending on its value, as the algorithm specifies. Therefore, we create the case structure, an If-Else statement operation, to differentiate whether the passing pulse width is valid. After that is finished, the program, in the third sequence, will send 0 voltage to the servo. At this point, the sequence is finished, and the program starts over.

5.3.2. ServoChannelTargetReference.vi

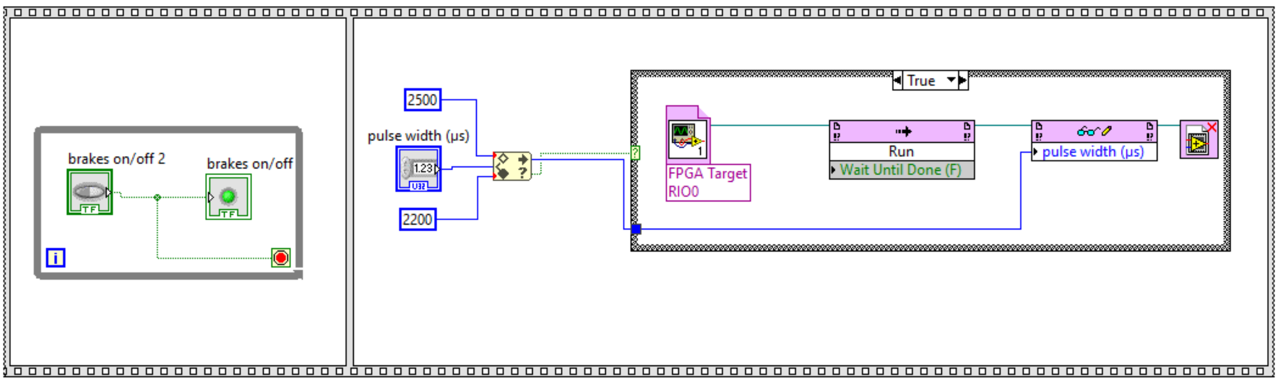


Figure 5.4.: FPGA Target

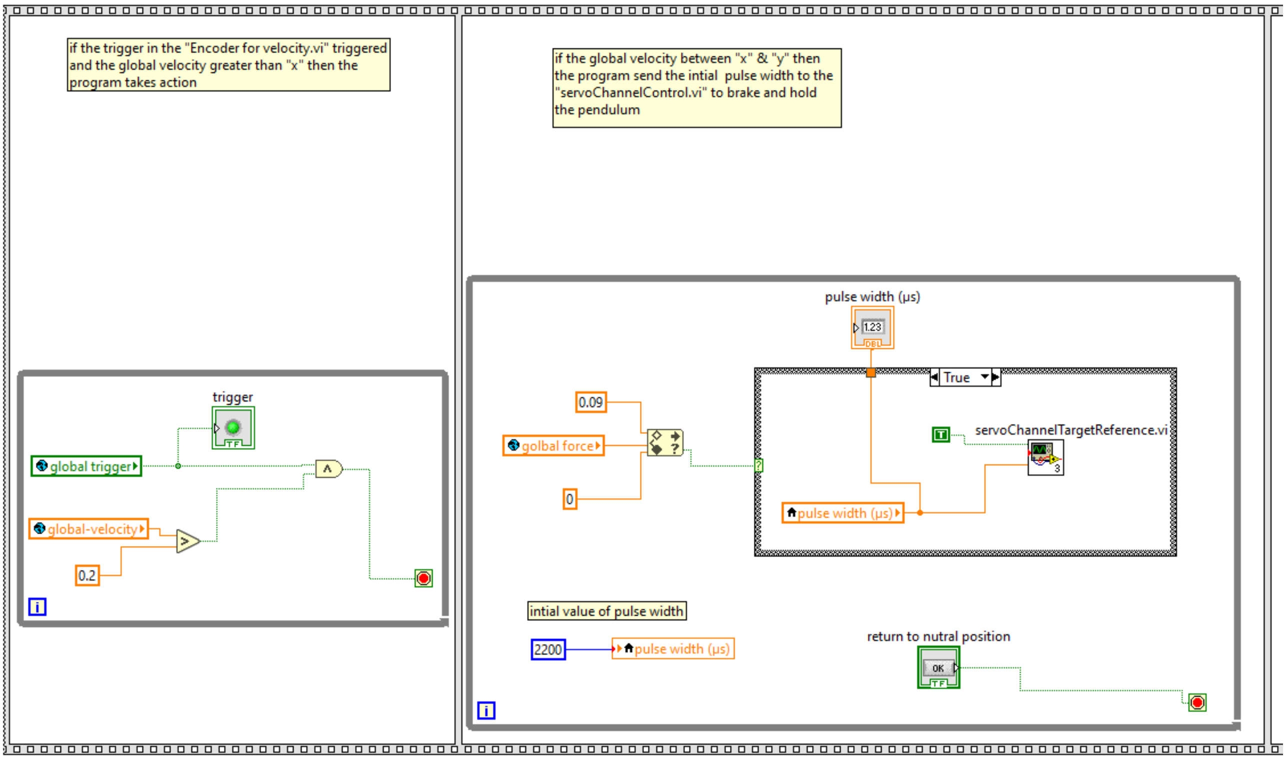
This program (figure 5.4) is an interface algorithm (2) to communicate with the FPGA target. It is built in a flat sequence of two parts. The first part is a while loop running and continually waiting for the trigger that indicates when the brakes must be activated. When the button is triggered, the loop will terminate, and the next part of the program will begin. The program will pass the argument $pulseWidth$ to the FPGA target in the second part. Before passing it, the argument must be validated to ensure it is within the allowed range. If it is not within the range, no action will be taken. It will be forwarded to the target if it is within the range.

5.3.3. servoControl.vi

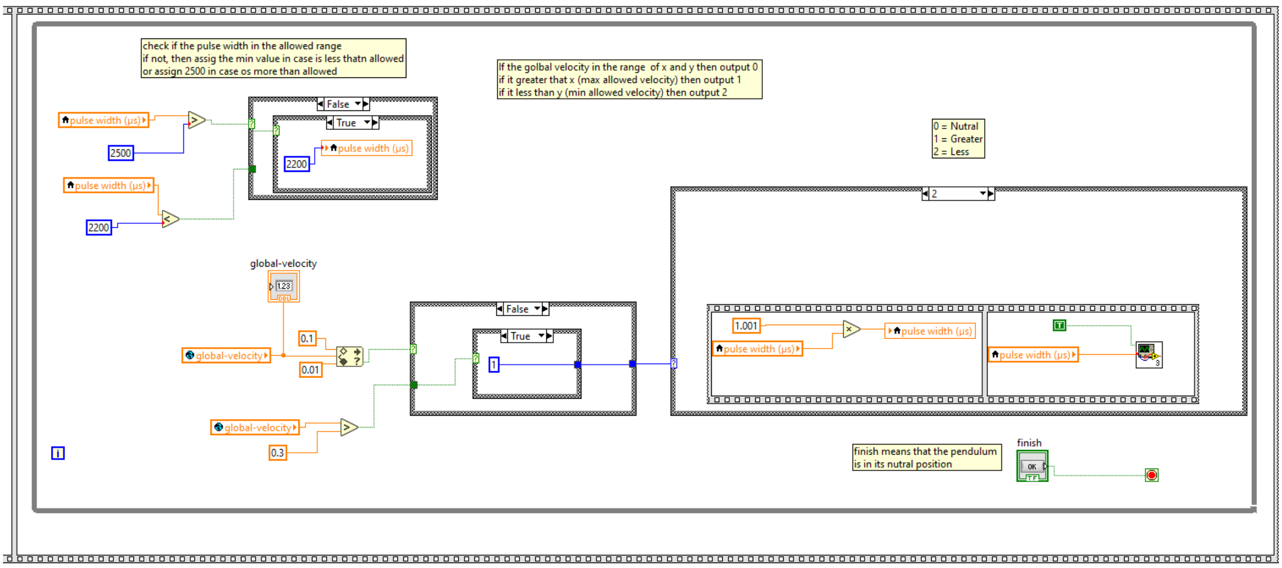
The program (5.5) is responsible for controlling the brakes. This program is implementing the algorithm (3) and is divided into three parts. In the first part, which is in Figure (5.5a), a global trigger observes the motion of the pendulum. When the pendulum is moving, the trigger will be true. Additionally, there is a global velocity variable that monitors the pendulum's velocity. When the velocity exceeds the minimum threshold, action is required. Therefore, if the operator outputs true, the loop will terminate, and the program will switch to the next flat sequence.

The second flat sequence (see figure 5.5a) is responsible for the initial brake reaction to hold the pendulum. It consists of a while loop that continually sends a pulse width value of 2200 if no force is detected from the force sensor. If any force is detected, the brakes must release the pendulum to prevent damage to the test stand or robot. Therefore, a global force variable is introduced to monitor the force received from the sensor, and each loop cycle tests the validity of this variable. If it is within the allowed range, the brakes remain activated. If not, the brakes must be released. The pulse width for the braking action is 2200 microseconds, and the release is 2500 microseconds. The program will switch to the following sequence when feedback is received through the button "return to the neutral position".

The final sequence (5.5b) is responsible for returning the pendulum to its starting position in a controlled manner. The program monitors the pulse width and the pendulum's velocity in each loop cycle to check. If the pulse width is greater than the allowed pulse width, then it will be assigned the upper limit, which is 2500, and if it is less than the minimum allowed value, then it will be assigned 2200, which is the lower limit. In addition, if the velocity is within the allowed range. If the velocity is within the range, the program will do nothing by assigning the second case structure a value of 0. If the velocity exceeds the maximum allowed value, a value of 1 will be assigned to the case structure, and the pulse width will be decreased by $0.995 \times \text{pulseWidth}$. Otherwise, the case structure will be assigned a value of 2, and the pulse width will be increased by $1.001 \times \text{pulseWidth}$.



(a) First & Second Phases



(b) Third Phase

Figure 5.5.: Servo Control Algorithm

6. Evaluation

In this chapter, the thesis will present the results of experiments performed on the braking system and its reliability. There are two types of tests. First, a manual test will be conducted on the pendulum, where it will be pushed by hand. This test will be performed twice: once with the braking system disabled and once with the braking system enabled. The thesis will then present the differences between the two test cases, focusing on the velocity in relation to time over a period of time. The second type of test involves the robot *Franka Emika Roboter* with fixed velocity colliding with the pendulum n times. Velocity data in relation to time will be captured, and the mean of the n test data will be calculated. Then, the data will be calculated and analyzed to demonstrate the reliability of the braking system. This kind of test will be done with two different fixed velocities, 20% and 50% of the maximum robot velocity.

6.1. Manual Test

In Figure (6.1), there are two graphs. The red line chart shows the pendulum's velocity with a brake, indicating a 150.53 N force applied. The green line chart represents the velocity without brakes, where a force of 131.084 N is applied. The green line chart exhibits a wave-like behavior, indicating that the pendulum's velocity gradually decreases as it swings due to the influence of gravity. On the other hand, the red line chart shows that once the velocity exceeds 0.8 m/s, it immediately drops to zero. This suggests that the brake is engaged after the pendulum is set in motion and then brought to a velocity of 0. After the second 9, the chart depicts motion when the velocity is not zero, signifying that the feedback to release the pendulum is given, allowing it to move in a controlled manner back to its neutral position.

The comparison reveals the distinct behavior of the pendulum's velocity under natural effects like gravity and air friction versus when the brakes are applied, as evidenced by the velocity hitting zero following the initial spike.

6.2. Reliability Tests

In this section, the thesis will demonstrate the reliability of the braking system. Ten tests are performed on the pendulum using a robot at 20% of its maximum speed, where the average force applied is 47.298 N. The results are presented in Figure (6.2a). Another nine tests are conducted on the pendulum with a robot at 50% of its maximum speed, where the average force applied is 68.049 N, and the results are shown in Figure (6.2b). The charts in both figures show the average of all the test data collected during the tests. This means that in Figure (6.2a), the velocity in relation to time is captured from the ten tests, and the average velocity for each

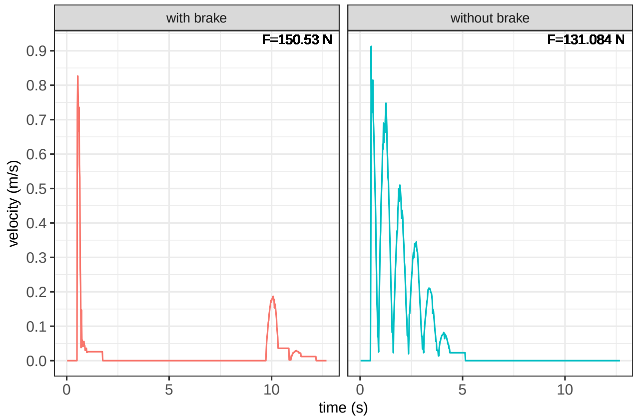
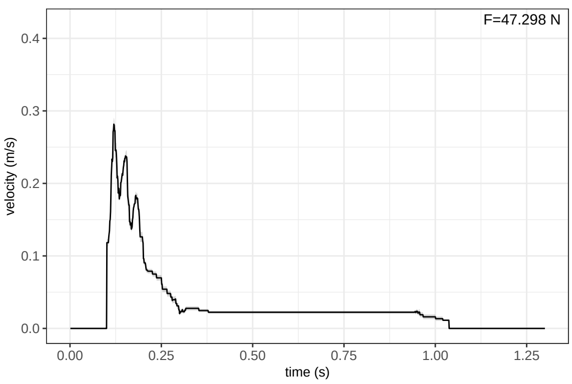


Figure 6.1.: Graphs for manual tests with and without the brake

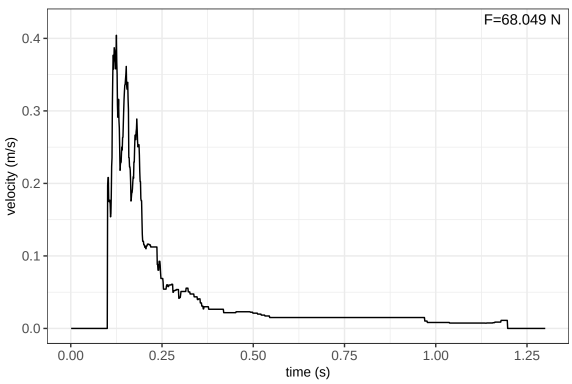
time is presented on the graph. The time is captured in 1 ms intervals. Besides that, an issue was observed in both figures (see Figure 6.1). The encoder is experiencing an offset error each time the velocity reaches zero. Approximately 0.75 seconds before reaching zero, there is a non-zero value, which then transitions to zero as expected. This non-zero value appears as a constant small velocity for that duration. Testing on the encoder is necessary to identify any potential issues.

Due to issues with the encoder, we cannot determine the average time it takes from the point the pendulum initiates a motion till it stops from the charts or the captured data. As a rough estimate, we predict that it will stop around the second 0.25 in Figure (6.2a) and around the second 0.35 in Figure (6.2b). However, it's important to note that these values are only estimations and may not be accurate. As previously mentioned, it's necessary to test the encoder to identify and address the issue that caused this offset to happen.

The analysis in this thesis demonstrates that the braking system is functioning effectively, as evidenced by the figures (6.1, 6.2). The system reliably detects motion and prevents the pendulum from swinging back and hitting the robot, thus meeting the requirement despite the issue in the encoder. This represents an improvement in the performance of the test stand.



(a) Test at 20% of the robot's maximum speed



(b) Test at 50% of the robot's maximum speed

Figure 6.2.: Graphs of the braking system reliability tests on the pendulum using a robot

7. Conclusion

This thesis was dedicated to the development of a braking system to protect robots in a test facility designed to analyze effective robot masses. The primary goal was to prevent the pendulum from swinging when the impact on the pendulum occurred during tests. The issue encountered was that when a robot hit the pendulum to collect data for estimating the effective robot mass, the pendulum would swing as expected and hit the robot as it came to a stop. Therefore, a braking system was designed to prevent a second collision when the pendulum returns and hits the robot. The braking system, as the main focus of this research, will protect the robot and the test equipment from damages that may lead to deviation and incorrect estimation of the effective robot mass over time.

The development process involved a comprehensive analysis of the problem, identification of requirements, and evaluation of various potential solutions following the engineering guideline VDI 2222 [14]. The chosen solution was implemented using mechanical and software components, utilizing real-time control hardware and software provided by National Instruments. In addition to the mechanical and software solutions, 3D modeling was also used to overcome some obstacles faced by mechanical design. This solution made the process more flexible, resulting in faster processing.

Our evaluation showed that an improvement was evident in both manual (6.1) and automated (6.2) tests, where the system reliably controlled the pendulum's motion, to control the pendulum and reduced the risk of the damage to the robot. The design provides precise control over the pendulum's motion as the tests are conducted. Once the system receives feedback, the pendulum will be released and returned to the starting position. These contributions address the research questions posed at the beginning of the thesis and provide a robust solution to the challenges identified.

As this thesis enhanced the test stand and developed and implemented a braking system, certain improvements could be made. First is the integration of force sensors in the caliper. Utilizing data captured from the sensor can be valuable for monitoring the force applied in the caliper and analyzing it in real-time. If the force is not as expected, it is highly likely that an error has occurred and the robot continues to move. In such cases, the braking system should release the caliper, preventing damage to the test facility and the robot. The second is motor enhancement. As the current motor is a servo motor (5.2.1), a more robust motor could be used to allow for a more substantial test and a heavier pendulum weight. Finally, it is to build an interconnection between the braking system and the test robot. Improvements in the software can enable communication between the robot and the brake system. This communication allows for feedback from the robot, for example, indicating that the pendulum is ready to be released. Additionally, the communication mechanism also enables the scalability of the test facility and a fully automated version.

8. References

[1] Janis Arents et al. “Human–robot collaboration trends and safety aspects: A systematic review”. In: *Journal of Sensor and Actuator Networks* 10.3 (2021), p. 48.

[2] Tufts Baxter. *ROS for LabVIEW Software*. 2024. URL: https://github.com/tuftsBaxter/ROS-for-LabVIEW-Software (visited on 04/20/2024).

[3] Alberto Borboni et al. “The Expanding Role of Artificial Intelligence in Collaborative Robots for Industrial Applications: A Systematic Review of Recent Works”. In: *Machines* (2023). URL: https://www.mdpi.com/2075-1702/11/1/111.

[4] CUI Devices. *AMT133Q-V | Modular Incremental Rotary Encoder*. 2024. URL: https://www.cuidevices.com/product/motion-and-control/rotary-encoders/incremental/modular/amt133q-v (visited on 03/20/2024).

[5] RP Group. https://www.rp-group.com/notlicht/zubehoer-systeme/mc-bmsm/data-sheet-for-mc-bmsm-and-mc-bms-ext_de-rev.001-170521-r.pdf. (Visited on 06/13/2024).

[6] Sami Haddadin et al. “Towards the Robotic Co-Worker”. In: *Proceedings of the International Symposium on Robotics Research*. German Aerospace Center (DLR) and KUKA Roboter GmbH. Wessling, Germany: Springer, Oct. 2011, pp. 261–282.

[7] National Instruments. *NI 9157/9159 Getting Started Guide*. https://www.ni.com/docs/de-DE/bundle/ni-9157-9159-getting-started/page/install-module.html. (Visited on 06/13/2024).

[8] *ISO/TS 15066:2016: Robots and robotic devices – Collaborative robots*. https://www.iso.org/standard/62996.html. International Organization for Standardization, 2016. (Visited on 04/12/2023).

[9] George Jackson. *What happens to kinetic energy in a car crash?* 2024. URL: https://physics-network.org/what-happens-to-kinetic-energy-in-a-car-crash/ (visited on 05/19/2024).

[10] Kaufland. *Servo Motor DS51150*. 2024. URL: https://www.kaufland.de/product/464594907/ (visited on 05/26/2024).

[11] Robin Jeanne Kirschner et al. “Notion on the correct use of the robot effective mass in the safety context and comments on ISO/TS 15066”. In: *2021 IEEE International Conference on Intelligence and Safety for Robotics (ISR)*. IEEE. 2021, pp. 6–9.

[12] Oriental Motor Asia Pacific Pte. Ltd. *AC Motor Fundamentals*. https://www.orientalmotor.com.sg/om/technical/ac-motors/ac-motor-fundamentals.html. 2024. (Visited on 06/13/2024).

[13] Eloise Matheson et al. “Human–robot collaboration in manufacturing applications: A review”. In: *Robotics* 8.4 (2019), p. 100.

[14] *Methodisches Entwickeln von Lösungsprinzipien*. Blatt 1. 2222. VDI-Richtlinie. VDI-Gesellschaft Entwicklung, Konstruktion Vertrieb. Verein Deutscher Ingenieure. Düsseldorf, June 1997.

[15] Giovanni Gerardo Muscolo et al. “A Method for the Calculation of the Effective Center of Mass of Humanoid Robots”. In: *Proceedings of the 11th IEEE-RAS International Conference on Humanoid Robots*. Conference held on October 26–28. Bled, Slovenia: IEEE, Oct. 2011. DOI: 10.1109/ICHR.2011.6128468.

[16] National Instruments. *CompactRIO*. 2024. URL: https://www.ni.com/de/shop/compactrio.html (visited on 02/23/2024).

[17] David Sanders and Alexander Gegov. “AI tools for use in assembly automation and some examples of recent applications”. In: *Assembly Automation* 33.2 (2013), pp. 184–194.

[18] *Servo Motor DS51 150kg 80kg 60 kg Servomotor*. https://grabcad.com/library/servo-motor-ds51-150kg-80kg-60-kg-servomotor-1. (Visited on 05/25/2024).

[19] Bruno Siciliano and Oussama Khatib. “Springer Handbook of Robotics”. In: ed. by Bruno Siciliano and Oussama Khatib. 2nd ed. Springer, 2016. Chap. 69.2.2 Safety Standards for Human–Robot Interaction, p. 1845.

[20] Bruno Siciliano and Oussama Khatib. “Springer Handbook of Robotics”. In: ed. by Bruno Siciliano and Oussama Khatib. 2nd ed. Springer, 2016. Chap. 2 Kinematics, p. 11.

[21] Bruno Siciliano and Oussama Khatib. “Springer Handbook of Robotics”. In: ed. by Bruno Siciliano and Oussama Khatib. 2nd ed. Discussing the impact of outliers on least-squares estimates and robust statistics. Springer, 2016. Chap. Robotics Foundation-Sensing and Estimation-Robust Estimation Methods, p. 104.

[22] Scientific Times. “Biomechanics - Unveiling The Science Behind Athletic Movement”. In: *Scientific Times* (2023). URL: https://scientifictimes.org/biomechanics/ (visited on 06/17/2024).

A. Detailed Rating

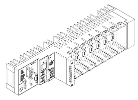
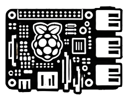
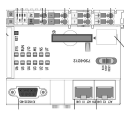
Function	Nr.	Name	Component	Metrics	Rating	Final Rating
Information Conduct (Signal)	1	 CompactRIO	1.1	Performance	5.0	4.67
				Compatibility	5.0	
				Reliability	4.5	
				Complexity	4.2	
				Availability	5.0	
				Efficiency	4.5	
				Cost	4.5	
	2	 Raspberry Pi	1.2	Performance	3.5	3.87
				Compatibility	2.0	
				Reliability	3.8	
				Complexity	4.0	
				Availability	5.0	
				Efficiency	3.8	
				Cost	5.0	
	3	 WAGO PFC Controllers	1.3	Performance	5.0	3.93
				Compatibility	2.0	
				Reliability	4.5	
				Complexity	3.5	
				Availability	4.0	
				Efficiency	4.5	
				Cost	4.0	

Figure A.1.: Information Conduct Signal. Refer to (4.8) for the icon references

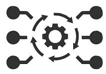
Function	Nr.	Name	Component	Metrics	Rating	Final Rating
Information Conduct (Data)	4	 Algorithms Processing Data	2.1	THE ONLY OPTION	5.0	5.00

Figure A.2.: Information Conduct Data

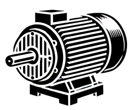
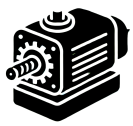
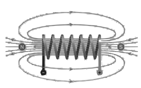
Function	Nr.	Name	Component	Metrics	Rating	Final Rating
Energy Conduct	5	 Electric Stepper Motor	3.1	Performance	4.5	4.29
				Compatibility	4.5	
				Reliability	4.0	
				Complexity	4.0	
				Availability	5.0	
				Efficiency	4.0	
				Cost	4.0	
	6	 Electric Brushless Motor	3.2	Performance	4.0	4.26
				Compatibility	4.5	
				Reliability	4.5	
				Complexity	4.0	
				Availability	5.0	
				Efficiency	4.3	
				Cost	3.5	
	7	 Electric Coil	3.3	Performance	2.0	2.29
				Compatibility	2.0	
				Reliability	1.5	
				Complexity	3.0	
				Availability	3.0	
				Efficiency	1.5	
				Cost	3.0	

Figure A.3.: Energy Conduct

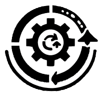
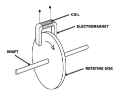
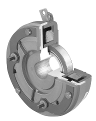
Function	Nr.	Name	Component	Metrics	Rating	Final Rating
Material Transform	8	 Ratchet & Pawl Brakes	4.1	Performance	4.5	4.44
				Compatibility	4.5	
				Reliability	4.8	
				Complexity	4.0	
				Availability	4.0	
				Efficiency	4.8	
				Cost	4.5	
	9	 Electro-Discs Brakes	4.2	Performance	5.0	5.00
				Compatibility	5.0	
				Reliability	5.0	
				Complexity	5.0	
				Availability	5.0	
				Efficiency	5.0	
				Cost	5.0	
	10	 Pawl-Electro-Discs Brake	4.3	Performance	4.8	3.99
				Compatibility	4.0	
				Reliability	4.5	
				Complexity	3.8	
				Availability	2.0	
				Efficiency	5.0	
				Cost	3.8	
	11	 Electromagnetic Brakes	4.4	Performance	2.8	2.37
				Compatibility	2.8	
				Reliability	2.0	
				Complexity	3.0	
				Availability	2.0	
				Efficiency	1.0	
				Cost	3.0	
	12	 Electromagnetic Clutch	4.5	Performance	3.0	2.76
				Compatibility	2.0	
				Reliability	3.8	
				Complexity	2.0	
				Availability	3.0	
				Efficiency	3.0	
				Cost	2.5	

Figure A.4.: Material Transform. Refer to (4.8) for the icon references

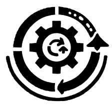
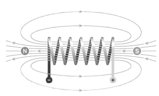
Function	Nr.	Name	Component	Metrics	Rating	Final Rating
Energy One-Sided Transform	13	 Ratchet & Pawl	5.1	Performance	4.8	4.53
				Compatibility	4.0	
				Reliability	4.5	
				Complexity	4.8	
				Availability	4.6	
				Efficiency	4.5	
				Cost	4.5	
	14	 Screw Connection Friction	5.2	Performance	3.5	3.30
				Compatibility	3.0	
				Reliability	3.0	
				Complexity	2.5	
				Availability	3.8	
				Efficiency	3.8	
				Cost	3.5	
	15	 Friction	5.3	Performance	5.0	5.00
				Compatibility	5.0	
				Reliability	5.0	
				Complexity	5.0	
				Availability	5.0	
				Efficiency	5.0	
				Cost	5.0	
	16	 Electromagnet	5.4	Performance	2.5	2.36
				Compatibility	2.0	
				Reliability	2.5	
				Complexity	2.0	
				Availability	3.0	
				Efficiency	2.0	
				Cost	2.5	

Figure A.5.: Energy One-Sided Transform

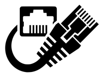
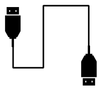
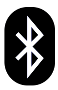
Function	Nr.	Name	Component	Metrics	Rating	Final Rating
Information Transform	17	 Ethernet	6.1	Performance	5.0	4.77
				Compatibility	4.5	
				Reliability	4.8	
				Complexity	4.5	
				Availability	5.0	
				Efficiency	4.8	
				Cost	4.8	
	18	 WiFi	6.2	Performance	4.0	3.96
				Compatibility	4.0	
				Reliability	3.8	
				Complexity	4.0	
				Availability	4.0	
				Efficiency	4.0	
				Cost	3.9	
	19	 Direct Cable Connection	6.3	Performance	5.0	5.00
				Compatibility	5.0	
				Reliability	5.0	
				Complexity	5.0	
				Availability	5.0	
				Efficiency	5.0	
				Cost	5.0	
	20	 Coud Services (Internet)	6.4	Performance	2.5	2.29
				Compatibility	1.0	
				Reliability	2.5	
				Complexity	2.0	
				Availability	3.5	
				Efficiency	2.5	
				Cost	2.0	
	21	 Bluetooth	6.5	Performance	3.5	3.00
				Compatibility	2.5	
				Reliability	2.0	
				Complexity	3.5	
				Availability	3.0	
				Efficiency	3.0	
				Cost	3.5	

Figure A.6.: Information Transform