UNIVERSITAT POLITÈCNICA DE VALÈNCIA

School of Industrial Engineering

Model-based and algorithmic tool development for the analysis of the Boom Suspension systems on the FluxJet vehicle.

Master's Thesis

Master's Degree in Industrial Engineering

AUTHOR: Herraiz Barca, David

Tutor: Solanes Galbis, Juan Ernesto

ACADEMIC YEAR: 2023/2024

Abstract

Modern mechatronic systems are indispensable in today's transportation engineering, yet their development is challenging due to the cross-domain complexity involving mechanical, electrical and software engineering aspects.

In this context, the innovative TransPod vehicle "FluxJet," a hybrid between an airplane and a train, represents such a complex mechatronic system. The FluxJet operates within a specialized infrastructure called the FluxWay. Magnetic forces generated by linear induction motors allow the FluxJet to lift and thrust within the FluxWay, which is maintained at a low ambient pressure near vacuum. This setup minimizes friction and air resistance, thereby enhancing the efficiency of the vehicle.

The overall "FluxJet" system includes several Boom Suspension (BS) subsystems that actively connect the vehicle's body to external components (payloads), such as linear motors, wheel systems, and current collectors. Given the variety of payloads and functional requirements, the design of the BS system must be adaptable across different FluxJet prototypes to meet specific operational and design constraints. This thesis aims to define a reusable, parameter-based design foundation for Boom Suspension systems.

The development process for the BS system is analysed using the V-Model as a Systems Engineering process. Incorporating Model-Based Systems Engineering (MBSE) methods helps establish the structure and behaviour of the mechatronic system and the relationship between requirements and solutions, thereby contributing transparency to the development process.

An algorithmic tool is programmed to enable system engineers to analyse the BS system behaviour. This tool is fundamental for assessing different configurations of the system and identifying locally optimal solutions, ensuring that the chosen design meets all requirements while being optimized for efficiency and performance.

Establishing this approach enhances the adaptability of the design process for BS systems in future iterations of the FluxJet and facilitates the rapid identification of effective solutions for new prototypes and payloads, reducing the time required for the design process while ensuring that each configuration meets specific operational constraints and performance requirements.

Table of contents

Abbreviations

Abbreviation	Definition
ARCADIA	Architecture Analysis and Design Integrated Approach
AWS	Arc Wheel System
BR	Boom Roll
BS	Boom Suspension
EMC	Electromagnetic Compatibility
ER	Extend/Retract
IEEE	Institute of Electrical and Electronics Engineers
INCOSE	International Council on Systems Engineering
IPP	Intermediate Power Pickup
ISO	International Organization for Standardization
LA	Logical Architecture
LCC	Life Cycle Cost
LE	Linear Engine
LM	Linear Motor
MAGLEV	Magnetic Levitation
MBSE	Model-Based Systems Engineering
NCOSE	National Council on Systems Engineering
OA	Operational Analysis
PA	Physical Architecture
PL	Payload
PPP	Primary Power Pickup
SA	System Analysis
SE	Systems Engineering
SOI	System Of Interest
SoS	System of Systems
WR	Wrist Roll
ZPP	high-Z Power Pickup

List of figures

List of tables

1. Introduction

The aim of this thesis is to analyse and optimise the development of transportation technologies to enhance future mobility systems. As society evolves and the demand for faster and more efficient modes of transportation intensifies, the necessity to explore and implement revolutionary technologies becomes increasingly apparent. This thesis specifically addresses the development of the boom suspension system within the TransPod vehicle, a crucial system of the vehicle. By focusing on this system, this thesis seeks to establish a structured methodology for system development that is both efficient and adaptable, contributing for future advancements in transportation technology.

The subsequent sections detail the technological background and motivation, followed by an introduction to the TransPod system.

1.1. Background and motivation

Mobility and transportation are crucial and vital services to society, consisting of interconnected systems designed to meet the mobility needs of both people and goods. These transportation systems are inherently complex, comprising both physical and organizational elements that interact and affect each other in both direct and indirect ways. From this viewpoint, the transportation system can be described as a dynamic, large-scale, interconnected, open, socio-technical system. However, current transportation methods, including rail, road, air, and waterborne transport, rely on established concepts. Over the years, improvements have primarily been evolutionary, aiming to achieve a safe, efficient, reliable, and accessible transportation network. In recent years, various new transportation concepts and technologies have emerged as highly promising. Disruptive transportation technologies – those with the potential to radically alter industries and societies (Millar, Lockett, and Ladd 2018, 254-260) – have become a significant focus of research and development (Herrera-Quintero et al. 2019). Ultra-high-speed tube transportation is one of those very promising transportation technologies that emerges as the future solution for long range distance transportation for passengers and freight (Barber et al. 2023).

Transportation companies pursuing the mentioned pioneering technologies are intensely competing to be the first to develop and launch these innovations. As a result, this industry is driven towards system development in higher quality, shorter time and at lower cost. However, these technologies are inherently complex, consisting of multiple interconnected systems across various categories – including mechanical, electrical, digital, pneumatic, and hydraulic components – that must function seamlessly together to achieve specific functions. This results in the central challenge of determining the best overall system solution for a specific development project among many possible overall system solutions.

This thesis introduces the FluxJet vehicle, developed by TransPod, with a focus on the rapid development of its crucial boom suspension (BS) system, that is primarily used to manoeuvre the vehicle's peripheral systems. Due to the various versions and design variations of the BS system, creating a model that simplifies and accelerates the design process is essential. The goal is to establish a reusable methodology that serves as a development baseline, applicable to future prototypes, thereby reducing the development time for subsequent BS systems. Additionally, a digital tool will be provided to enhance transparency and accessibility of the system's requirements, architecture, and variability. This approach is expected to not only streamline the BS system's development but also offer a versatile methodology that can be applied to other systems within the vehicle.

1.2. The TransPod system

Ultra-high-speed tube transportation is designed to allow near-supersonic intercity transportation, connecting cities with a fixed guideway infrastructure. Tube transportation is like rail transportation but uses vehicles travelling inside a round linear guideway (tube) rather than traditional railroad cars travelling along a flat linear guideway (railroad tracks). The vehicles can reach near-supersonic speeds thanks to the benefit of reduced air pressure inside the tube, which reduces friction at high speed. Tube transportation is under development by several transportation companies, including TransPod Inc., combining the latest techniques from the aerospace and rail industry.

TransPod, based in Canada, is developing the next generation of transportation technologies in cooperation with global aerospace and engineering partners. The company envisions a world in which people, cities, and businesses are connected through affordable and sustainable ultra-high-speed transportation (TransPod Inc. 2024).

The company has introduced the TransPod Line system, which represents a significant advancement over traditional tube-transportation concepts compared to previous types such as vacuum trains, traditional hyperloop, hypertube, etc. The TransPod system, illustrated in Figure 1, includes infrastructure (FluxWay) and vehicles (FluxJets).

Figure 1: The TransPod system, including infrastructure (FluxWay) and vehicles (FluxJets), source TransPod Inc. 2021

This advanced system reduces capital investment, enhances reliability, and improves operational performance, supporting its viability as an ultra-high-speed transportation solution. Utilising sustainable solar and electric energy and remaining unaffected by weather conditions, the TransPod system allows passenger and cargo travel at speeds exceeding those of aircraft. (TransPod Inc. 2021)

The FluxJet is like an aircraft (aerodynamic fuselage and structure), minus the wings, and operates like a train. The system is designed to have several vehicles travelling in each direction simultaneously. The vehicles are computer-controlled and operated autonomously, enhancing passenger safety by eliminating human piloting errors. The system operates in a near-vacuum enclosed tube environment, making air resistance much lower than in the outside environment. In the case of any evacuation emergency, the tube can be returned to atmospheric pressure by systems which rapidly allow the ordinary air pressure back inside the tube. The tube is constructed out of steel or alternatively composite materials. It can be built below ground, at ground level, or above ground suspended by concrete pillars. To achieve maximum operating speeds, the tube transport line is designed to be as vertically and horizontally straight as possible, avoiding sharp curves which would result in strong and uncomfortable G-force sensations for passengers. The vehicles are propelled by linear induction engines, which are powered entirely by electric energy. These powerful engines, operating in a near-vacuum environment, enable passenger and cargo vehicles to travel at over 1000 km/h.

1.3. The TransPod line

The TransPod line or FluxWay represents a significant advancement in vacuum tube transportation technology, designed to facilitate vehicle speeds exceeding 1000 km/h. The FluxWay utilizes a dual tube design supported by concrete pylons, with the height varying based on the line configuration and surrounding terrain. The infrastructure incorporates three principal types of engineered structures:

- Elevated guideway: this design minimizes land usage, allowing a smaller footprint compared to traditional rail tracks and facilitating unobstructed crossing for vehicles and agricultural machinery beneath. It enhances safety by eliminating crossing controls at roadway
- Ground-level guideway: this design integrates seamlessly with existing landscapes while maintaining efficient transport capabilities.
- Underground guideway: this design offers superior safety features; however, it incurs higher construction costs.

Compared to conventional high-speed rail systems, FluxWay's infrastructure offers notable advantages in terms of cost-efficiency and safety. Strategic design choices, such as the use of elevated guideways, not only enhance safety but also reduce costs by minimizing land disruption and associated expenses.

Significant technological innovations have been introduced by TransPod, focusing primarily on the vehicle itself to reduce dependency on extensive infrastructure enhancements. The system abandons traditional magnetic levitation (MAGLEV) technologies, which involve superconducting magnets that are expensive and technologically limiting in a vacuum environment. Instead, an innovative levitation and thrust system is embedded within the vehicle, eliminating the need for costly electromagnets along the track. This decreases the infrastructure costs associated with long track installations.

The vehicle technology enables levitation, propulsion, and high-speed power transfer without reliance on expensive MAGLEV systems or onboard batteries. This technology is designed for compatibility with renewable energy sources, including solar power, supplemented by connections to regional electrical grids.

Historically, MAGLEV technology, though pioneered in the 1970s and demonstrated in high-speed train projects like the German Transrapid and the Japanese Shinkansen, has seen limited application globally due to its excessive costs. TransPod's strategy of concentrating technological advancements on the vehicle rather than the track infrastructure sidesteps the economic and material limitations faced by traditional MAGLEV systems.

1.4. The TransPod vehicle

The TransPod vehicle or FluxJet, illustrated in Figure 2, was announced in July 2022 in Toronto, Canada (TransPod Inc. 2024).

Figure 2: The FluxJet, source TransPod Inc. 2024

To achieve more efficiency and higher speeds than trains, automobiles, and jets, the TransPod system contains aerodynamic and propulsion systems to overcome the following challenges in actual transportation:

- Friction resulting from ground contact: eliminated by magnetic levitation mechanisms.
- Aerodynamic friction between the air and the vehicle: lowered through reducing the ambient pressure in which the vehicles operate.

The combination of these factors allows electrically powered vehicles to travel faster than aircraft, using little energy, at ground level. This increases the typical speed limit of trains, above 1000 km/h, when cities are at least 500 km apart.

Contrary to long and heavy trains, FluxJet's are small units: a vehicle is 25 m long, capable of transporting up to 50 passengers, sitting in rows of seats separated by a central aisle. Cargo vehicles use the same structure as passenger vehicles, but the interior is designed to fit air cargo containers and/or pallets. One cargo vehicle can transport approximately 10 t of goods.

The main subcomponents of a vehicle include (TransPod Inc. 2021):

- Fuselage: The TransPod fuselage is made of composite materials like the construction of commercial aircraft today, therefore providing a similar structural integrity.
- Pressurisation: The cabin is pressurised for passenger comfort, like a jet aircraft. The vehicle systems are like existing commercial aircraft technology.
- Liquid cooling system: Heat is transferred out of electronics, propulsion drives, motors, and the passenger air conditioning system. The cooling system is like engine cooling systems in the automotive and aviation industries.
- Propulsion System: The vehicle is driven by electric power. Linear Induction Motors (LIM), a type of magnetic traction engine, use moving magnetic fields to generate force against the tube traction surface structure. These traction engines provide thrust to accelerate the vehicle and braking to decelerate and maintain the vehicle position along the guideway. The FluxJet is equipped with a self-canting mechanism which allows it to easily navigate in curves along the guideway, allowing for a smooth ride like high-speed rail systems.

- Levitation System: TransPod's technology controls a magnetic levitation system, to keep the vehicle levitated off the bottom of the tube guideway, creating a smooth ride, and reducing energy consumption due to an absence of mechanical friction between the vehicle and the infrastructure.
- Electrical Power Transmission: The electrical system receives electrical energy onboard the vehicle from “3rd-rail” conductors inside the tube. In that way, external power from the grid connections is delivered into the tube, to the conductors mounted inside the tube, and in turn to the vehicle as it runs along these conductors inside the tube. This technology endeavours to be the world’s fastest power transmission system due to the low pressure inside the tube, beating the capability of high-speed trains.
- Wheel System: The vehicle is equipped with a “landing gear” primarily used when the vehicle is grounded in loading and unloading areas while having the levitation operation off. Secondarily it is a safety system, in case of failure during the levitation operation.

Several of the systems described, such as the Linear Induction Motors used for propulsion and levitation, the power transmission systems, and the wheel systems, are peripheral, meaning they must be positioned outside the fuselage to perform their functions. To deploy and retract these peripheral systems, also referred to as payloads, the FluxJet is equipped with various boom suspension (BS) systems. The animation "TransPod Age of Innovation - July 2022" (TransPod Inc. 2024) illustrates these payloads deploying at various stages. The BS system is the SOI (System Of Interest) of this thesis.

2. State of the art

This section provides an in-depth analysis of the current state of the art in systems engineering. It begins with a thorough overview of the discipline, delving into its fundamental concepts, definitions, and structured processes. The aim is to highlight the significance of systems engineering in the design, implementation, and management of complex systems. The section also explores the latest methodologies and practices, with a particular focus on the advancements brought about by Model-Based Systems Engineering (MBSE). By examining models, simulations, and the MBSE approach in detail, this section underscores the critical role of these tools in modern engineering processes. Through presenting the latest standards, tools, and methodologies, it lays the groundwork for understanding the evolving landscape of systems engineering and its applications in mechatronic systems.

2.1. Systems Engineering

This section provides a concise overview of the discipline of Systems Engineering (SE). It introduces the core concepts of SE, offering clear definitions and outlining the structured process involved in designing, implementing, and managing complex systems. The emphasis is on the value added by the discipline in enhancing efficiency, reducing costs, and improving system reliability.

2.1.1. Fundamental definitions

Systems engineering, as defined by the International Council on Systems Engineering (INCOSE), encompasses an integrated set of elements, subsystems, and assemblies aimed at achieving a specific objective. These components include products, processes, people, information, techniques, facilities, services, and other supporting elements. (Walden et al. 2015)

The concept of a system includes both internal and external views. Externally, a system interacts with a set of external elements, known as the operating environment or context, which includes users or operators. Internally, it comprises the interconnected elements that function within a defined boundary. The system boundary acts as a demarcation line, distinguishing the system from its external environment and determining what is included within the system and what is not.

The functionality of a system is expressed through its interactions with the operating environment, particularly with its users. When considered as a cohesive unit of interacting components, a system's functionality emerges not only from these interactions but also from the organizational structure of its elements. This structure is referred to as system architecture, which according to ISO/IEC/IEEE 42010:2011 (ISO 42010:2011), includes the system's essential concepts or properties within its environment, its elements, their relationships, and the principles guiding its design and evolution.

Systems also possess attributes, observable characteristics represented symbolically as variables. These attributes, such as the airspeed of an aircraft, can be measured, assigning values to variables. A system is in a specific state when the values of its attributes are stable over a significant period. In SE, the system elements have processes (e.g., operations) in addition to attributes. These have the binary logical values of being either idle or executing. A complete description of a system state therefore requires values to be assigned to both attributes and processes. The dynamic behaviour of a system is the evolution of its state over time.

The primary method for problem-solving in SE utilizes the black box and white box system representations. The black box model emphasizes an external perspective, focusing on the system's observable attributes. Conversely, the white box model delves into an internal perspective, incorporating both the attributes and the structural details of the elements. It is essential to understand the interplay between these models to fully represent a system, including its external attributes, internal structure, and the scientific laws governing their relationships.

The concept of "partitioning a system" involves organizing system elements into a structured hierarchy where each element is categorized by its function within the overall system, rather than by the detailed nature of their interactions. This classification can designate elements as indivisible (atomic) or as complex enough to be systems, further decomposable into smaller parts.

Modern systems often form part of a larger, interconnected entity known as a System of Systems (SoS), which integrates both technological and human components. The SoS, defined by its elements' managerial and operational independence, achieves outcomes beyond the capabilities of its individual parts. Depending on the problem at hand, a systems engineer might address an issue by considering it either as a single system or as an SoS, leveraging the appropriate perspective for optimal problem-solving.

SoS are distinguished by their complex behaviours. Unlike complicated systems, where part interactions follow fixed rules enabling reliable predictions (e.g., automobiles), complex systems (e.g., air transport systems) exhibit self-organizing behaviours that lead to unexpected emergent patterns. Even simple systems can display significant complexity due to changing behaviours.

To effectively understand a complicated system, it is broken down recursively into simpler components until each is understood, and then reassembled to grasp the entire system. However, this method is inadequate for complex systems, where emergent properties vital to understanding the system are lost when parts are isolated. Instead, complex systems require an iterative, adaptive approach to comprehend the entire system within its context. Thus, SE balances linear, procedural techniques for managing complicatedness with holistic, nonlinear, iterative strategies essential for addressing complexity, a balance that must be dynamically maintained throughout the SE process.

2.1.2. The Systems Engineering process

SE has continuously evolved to address the increasing complexity of systems across various industries. This evolution has been accompanied by the development of standardized methods and processes, promoted by organizations like IEEE and ISO. These standards facilitate a common language and methodologies that enhance the efficiency and predictability of engineering outcomes.

At its core, SE is an interdisciplinary approach that focuses on defining customer needs and required functionalities early in the development cycle. It documents requirements, then proceeds with design synthesis and system validation, considering all aspects of the problem including operations, cost, schedule, performance, training, support, test, manufacturing, and disposal. The approach integrates all disciplines and specialty groups into a structured development process that moves from concept to production to operation, aiming to satisfy both the business and technical needs of customers with a high-quality product.

The SE process, shown in Figure 3, begins with the specification and design phase, where stakeholder needs are elicited and analysed to define the system's objectives. This involves detailing required functionalities, interfaces, and performance characteristics, synthesizing alternative solutions, and conducting trade-off analyses to select the optimal solution that balances all requirements. Following this, system integration and testing are crucial to ensure that all components function as intended and meet the requirements. Testing at various stages of integration helps identify and resolve issues early in the development process. SE is known for its iterative process of top-down synthesis, development, and operation of a real-world system that satisfies, in a near-optimal manner, the full range of system requirements.

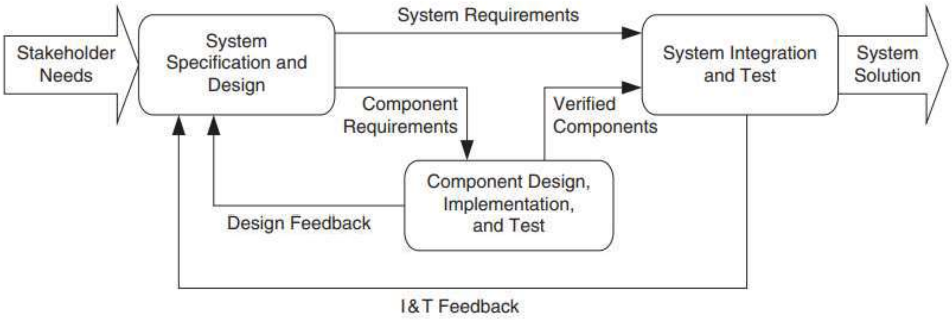


Figure 3: Simplified systems engineering technical processes, source Friedenthal, Moore, and Steiner 2014

Engaging a diverse group of stakeholders throughout the system lifecycle is crucial. This includes end-users, manufacturers, and maintenance personnel, as well as entities like government bodies influenced by regulations. The involvement of multidisciplinary teams ensures that all system aspects are considered and integrated effectively.

SE is traditionally a cornerstone in industries like aerospace and defence and is increasingly applied across various sectors including automotive, railway and telecommunications. This broad application is essential as SE integrates technical and management processes to ensure that development costs, schedules, and technical performance objectives are met.

The approach also emphasizes the importance of evaluating multiple solutions to ensure the best outcome. The solution-finding process is structured into defined phases to manage complexity and risk, utilizing a formal problem-solving cycle at each stage of development. This cycle typically involves identifying a problem, exploring alternative solutions, analysing, and evaluating these alternatives, and selecting the most suitable solution.

With its disciplined approach to design, integration, and testing, supported by evolving standards and a multidisciplinary team structure, SE effectively translates diverse stakeholder needs into efficient system solutions. It not only simplifies complexity but also ensures adaptability and sustainability of system designs across various sectors, making it an indispensable methodology in modern engineering.

2.1.3. Use and value of Systems Engineering

Systems Engineering traces its roots back to the 1930s and was established as a recognized discipline in 1954 (Walden et al. 2015). The discipline saw further institutionalization with the creation of the National Council on Systems Engineering (NCOSE) in 1990, which became the International Council on Systems Engineering (INCOSE) in 1995. A critical achievement occurred in 2008 with the harmonization of SE principles in the ISO/IEC/IEEE 15288 (ISO 15288:2015) standard, initially recognized in 2002. This standard, however, involves challenges due to the participation of multiple contractors and subcontractors within Systems of Systems (SoS) frameworks.

From its inception, SE has proven effective in managing complexity and change, which are increasingly prevalent in modern products, services, and societal structures. Mitigating risks associated with new or modified complex systems remains a core objective of SE. The importance of early decision-making in a system's lifecycle is crucial; decisions made without full comprehension can significantly impact the system's later stages.

The importance of early decision-making in a system's lifecycle is crucial; decisions made without full comprehension can significantly impact the system's later stages. Systems engineers are tasked with addressing these challenges decisively and prudently, as outlined in (Sheard 1996).

This is exemplified in Figure 4, which illustrates life cycle cost (LCC) trends from a study on US Department of Defence projects (Defence Acquisition University, 1993). For instance, at the concept stage, a system typically incurs 8% of the total LCC, with 80% of the LCC committed when only 20% of the actual cost has been accrued. This figure emphasizes the cost-effectiveness of addressing issues early in the lifecycle.

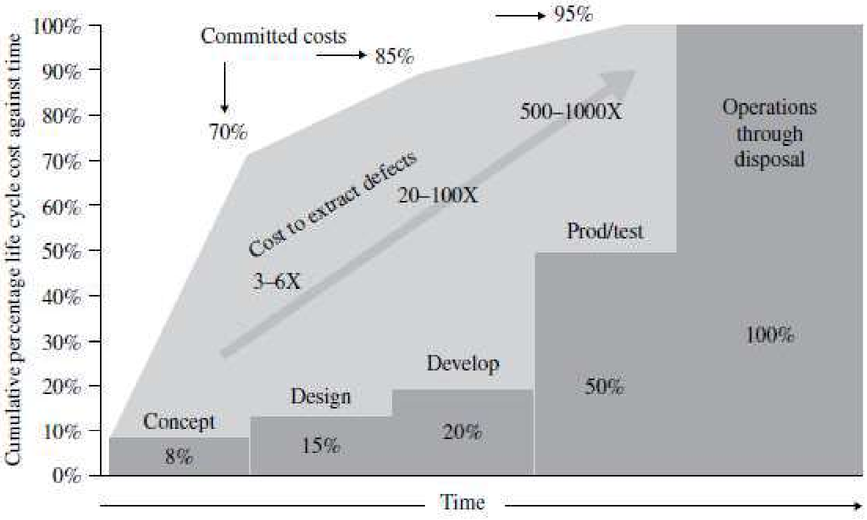


Figure 4: Life cycle cost over system lifetime, source Walden et al. 2015

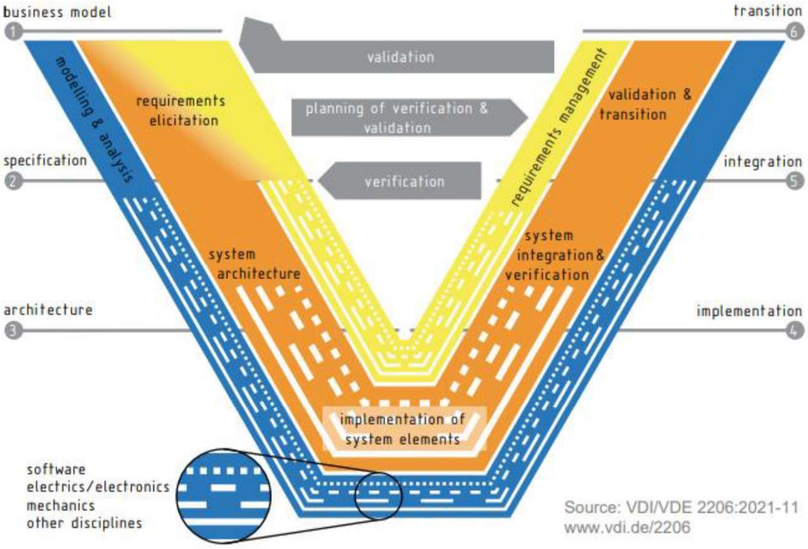
SE also enhances the informed execution of early project stages, transitioning from concept exploration to minimize risks associated with premature commitments. While the stages of modern product development may seem linear, they often recur throughout the lifecycle, bearing consistent consequences for poorly made decisions.

A 2012 study by the National Defence Industrial Association, IEEE, and Carnegie Mellon's Software Engineering Institute demonstrated that projects staffed with highly skilled SE professionals tend to achieve superior performance outcomes, affirming the value of SE in effective project management (Elm and Goldenson 2012).

In summary, SE is a critical discipline for developing complex systems, combining technical precision with comprehensive systems thinking. Its relevance and application continue to expand across various industries as systems grow more interconnected and complex.

2.1.4. Systems Engineering for mechatronic systems

Mechatronic products integrate mechanics, electronics, and software using interdisciplinary methods. Systems Engineering supports this with guidelines for development, such as the VDI Guideline 2206, "Design Methodology for Mechatronic Systems," first released in 2004 by the German Association of Engineers (VDI). This guideline adapts the V-Model from software engineering to mechatronics, covering the product creation process and highlighting the integration of all engineering disciplines. A revised version has been developed since 2016 and is shown in Figure 5.



Source: VDI/VDE 2206:2021-11
www.vdi.de/2206

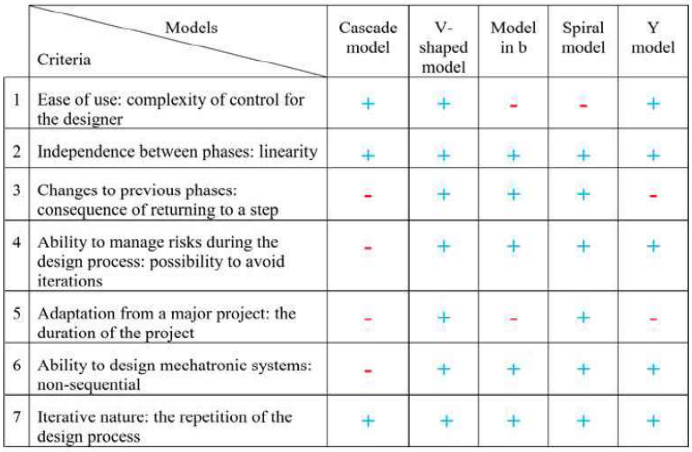
Figure 5: The new V-Model of VDI 2206, source Graessler and Hentze 2020

The V-Model structures the design methodology for mechatronic systems, emphasizing a comprehensive, multidisciplinary approach to product creation. It is adaptable to both traditional project management and agile methodologies, essential in modern engineering environments. The complexity of mechatronic system design arises from its integration of multiple domains. Designers must evaluate trade-offs and justify decisions based on analyses from various disciplines, ensuring coherence throughout the process.

Mechatronic system design is inherently complex due to its functional integration and multi-domain nature. At each design cycle stage, designers must evaluate trade-offs and justify choices based on analyses from various disciplines, ensuring coherence throughout the process. The authors in (Samon and Fotsa Tchouazong 2022, 71-78) conducted a comparative study of existing design models, which confirmed the V-Model's superiority for mechatronic system design, although it acknowledges areas for improvement, guiding future research. A comparative table of the different analysed models is shown in Figure 6.

Key elements of mechatronic systems include mechanical structures, electromechanical components (sensors and actuators), and control software. The "V" model outlines the design process from system-wide requirements to user-validated systems, ensuring coherence and traceability. It includes system design, discipline-specific design, and system integration phases, with continuous testing, verification, and validation. This approach reduces development costs and time by addressing incompatibilities early.

The revised V-Model has three coloured strands, each highlighting different parts of the design process. The central orange strand covers core activities and tasks. The inner yellow strand focuses on handling requirements. The outer blue strand is dedicated to modelling and analysis. These strands show the interconnected details of various disciplines like mechanics, electronics, software, pneumatics, hydraulics, optics, and human science.



	Models Criteria	Cascade model	V-shaped model	Model in b	Spiral model	Y model
1	Ease of use: complexity of control for the designer	+	+	-	-	+
2	Independence between phases: linearity	+	+	+	+	+
3	Changes to previous phases: consequence of returning to a step	-	+	+	+	-
4	Ability to manage risks during the design process: possibility to avoid iterations	-	+	+	+	+
5	Adaptation from a major project: the duration of the project	-	+	-	+	-
6	Ability to design mechatronic systems: non-sequential	-	+	+	+	+
7	Iterative nature: the repetition of the design process	+	+	+	+	+

Figure 6: Comparison of different design process models, source Samon and Fotsa Tchouazong 2022

Core activities start with gathering, structuring, and analysing stakeholder needs. These needs are documented in clear, technical language and translated into formal system requirements, refined throughout development.

In the system architecture and design phase, engineers create a comprehensive structure, including mechanical building structures, electronic circuit diagrams, and software program designs. They compare structural alternatives and allocate functions across disciplines iteratively to ensure effective system functionality.

Implementation involves detailed design and specification of system components. Tools like CAD and FEM are used for mechanical parts, and EDA or ECAD for electronic parts. Software elements are developed by converting algorithms into functional programs.

System integration and verification merge the implemented elements and subsystems to meet requirements, with thorough testing to ensure functionality. This step involves ongoing detailing and problem-solving.

Continuous validation and verification are central to the V-Model. This iterative process ensures all components meet specified requirements, considering both technical and human factors. Interdisciplinary teams and diverse stakeholder needs are involved to translate complex requirements into efficient solutions.

Finally, validation and transition processes confirm that the integrated system meets stakeholder needs and system requirements. Various validation methods ensure the system's functionality and readiness for handover.

The V-Model also emphasizes the importance of modelling and analysis for efficient, computer-based development and active requirement management. This comprehensive approach ensures that changes in requirements are systematically tracked, analysed, and integrated throughout the project.

In conclusion, the V-Model provides a robust framework for designing mechatronic systems by integrating mechanics, electronics, and software through a structured, interdisciplinary approach. Its emphasis on continuous validation, verification, and stakeholder involvement ensures coherence and efficiency throughout the development process. The V-Model's adaptability to both traditional and agile methodologies make it essential for addressing the complexities of modern engineering challenges.

2.2. Model-based Systems Engineering

The previous section provided a description of the systems engineering (SE) and the process used throughout the system lifecycle. This section delves into methods that cut across the SE processes, particularly MBSE, reflecting various aspects of the iterative and recursive nature of SE. It provides an overview of the foundational elements of this discipline, including models and simulations, the discipline itself, and the current MBSE tools.

2.2.1. Models and simulations

In the system engineering (SE) life cycle, models and simulations play a crucial role by validating system needs and behaviours before full-scale development and by providing a clear, coherent design for development and operational teams. This early application helps identify potential limitations and incompatibilities, thus avoiding higher costs and delays later, particularly during system operation. As systems increase in size or complexity, the value of modelling and simulation similarly grows. (Walden et al. 2015)

Models and simulations in the initial stages of the SE life cycle are aimed at gathering critical information before significant investment in design and development. They generate necessary data that is otherwise unavailable, supporting timely and cost-effective decision-making. These tools inform stakeholders of potential impacts, help in evaluating alternatives, and boost confidence in the system's capabilities. They also aid in understanding design requirements and the limits imposed by technology and management, while ensuring sustainability. (Walden et al. 2015)

Furthermore, models and simulations equip organizations and their suppliers with the insights to allocate appropriate personnel, methods, tools, and infrastructure for system realization. The complexity of models and simulations required varies depending on the competency of the staff and the novelty and scale of the system being developed. The effectiveness of these tools is judged by their timeliness, reliability, and ease of use and maintenance, aligning with the resources invested in their development, verification, validation, accreditation, operation, and maintenance. The benefits are proportionate to the gap between the problem's complexity and the staffing competencies. (Walden et al. 2015)

The term "model" has many definitions but refers to an abstraction or representation of a system, entity, phenomenon, or process of interest (DoD 5000.59-M, 1998). In SE system models play a vital role across the entire system lifecycle. They are instrumental for various purposes including the characterization of existing systems, evaluation of missions and system concepts, architectural design, and the detailing of requirements. Furthermore, these models aid in system integration, verification, training, and the evolution of designs. By encapsulating the key characteristics of the system, its operational environment, and interactions with other systems and operators, they enhance numerous lifecycle processes such as analysis, design, verification, validation, and operations. Models can be classified into two main types (Walden et al. 2015):

- Physical mock-ups: A model that represents an actual system, such as a model airplane or wind tunnel model, or a more abstract representation, such as a model that is often represented using a computer.
- Abstract models: An abstract model can have many different expressions to represent a system, entity, phenomena, or process, which vary in degrees of formalism.

    - Informal models: Systems can be represented using simple drawing tools or verbally. However, if the meanings of terms in these fewer formal representations are not universally understood, there can be a lack of precision and potential ambiguity. While these representations may be useful, they must meet specific criteria to be considered within the realm of modelling and simulation for SE.
    - Formal models: These can be subdivided into geometric, quantitative (i.e., mathematical), and logical (or conceptual) models.
        - Geometric models illustrate the spatial relationships within the system or entity.
        - Quantitative models express numerical relationships, such as mathematical equations, about the system or entity.
        - Logical models, sometimes known as conceptual models, delineate logical relationships within the system, such as whole-part relationships, interconnections between components, or sequence relationships among activities.

The term "simulation" refers to the use of a model, or multiple models, within a designated environment that facilitates the model's operation over time. Simulations are tools for examining the complex dynamic behaviours of systems, which may include software, hardware, human interactions, and physical phenomena. A computer simulation encompasses the analytical model represented as executable code, the input conditions and other input data, and the computing infrastructure required to run the simulation. Simulations in SE are categorized into three main types, each with distinct purposes and methodologies (Walden et al. 2015):

- Physical simulations use tangible models to replicate specific system attributes and environmental factors but are limited in scope and costly.
- Computer-based simulations vary based on the model of computation, handling a broad range of attributes and complexities, and emphasize modularity and reuse in different scenarios.
- Hardware-in-the-Loop simulations combine real-time interactions with hardware elements.

The choice of model or simulation for a particular phase of a system’s lifecycle depends on its intended use, the specific attributes of the system that are relevant, and the necessary level of model accuracy. Typically, a chosen model or simulation targets a specific subset of the system’s overall characteristics, such as timing, process behaviour, or various performance metrics. (Walden et al. 2015)

The scope of a model should align with its intended purpose, with the selection of model types and modelling languages tailored to specific needs. Key aspects to define in each model include breadth, depth, and fidelity (Walden et al. 2015):

- Breadth refers to how comprehensively the model covers system requirements, including functional, interface, and performance specifications.
- Depth indicates the level of system decomposition, from the overall system context down to individual elements.

- Fidelity involves the level of detail represented in the model, which can range from abstract depictions of interfaces to detailed, high-fidelity simulations that specify encoding or signal characteristics. Fidelity also encompasses the precision required in computational models, such as simulation time steps.

In a model-based approach, it is essential to integrate various models and simulations spanning multiple disciplines and domains effectively. System models, for example, delineate system components, while logical models of system architecture outline the partitioning and interconnections of these components. Quantitative models evaluate the properties of these elements to ensure they meet system requirements, and executable system models verify whether these elements behave as expected. Achieving semantic interoperability is vital for successful integration, ensuring that constructs across different models preserve consistent meaning. The implementation of standardized modelling languages, transformation techniques, and data exchange protocols supports this integration process. (Walden et al. 2015)

The quality of a model is measured by its adherence to specific modelling standards and its effectiveness in fulfilling its intended purpose. These standards vary depending on the type of model and may include criteria such as naming conventions, the use of annotations, the proper application of modelling constructs, and considerations for model reuse. (Walden et al. 2015)

In summary, models and simulations are pivotal in the system engineering lifecycle, enabling efficient management and risk mitigation of complex systems. This approach sets the foundations for Model-Based Systems Engineering (MBSE), which enhances these benefits by focusing on systematic model usage throughout the engineering process. The subsequent section delves into how MBSE builds on traditional modelling and simulation to drive innovation and consistency in system development.

2.2.2. The MBSE approach

Model-based Systems Engineering (MBSE) is an advanced approach in systems engineering that leverages continuous integration of system modelling to refine the development process of systems. According to the International Council on Systems Engineering (INCOSE) Systems Engineering Vision 2020 (INCOSE 2007), MBSE is defined as "the formalized application of modelling to support system requirements, design, analysis, verification, and validation activities beginning in the conceptual design phase and continuing throughout development and subsequent life cycle phases."

Model-Based Systems Engineering (MBSE) is frequently compared to the traditional document-based systems engineering (SE) approach. In the traditional approach, extensive information about the system is captured in various documents and artifacts, including specifications, interface control documents, system description documents, trade studies, analysis reports, and verification plans, procedures, and reports. The information contained within these documents is often difficult to maintain and synchronize, and difficult to assess in terms of its quality (correctness, completeness, and consistency).

In contrast, MBSE focuses on the creation and iteration of system models rather than static documents. This change in basic assumptions mirrors trends observed in other fields of engineering, such as mechanical and electrical, where there has been a significant move towards advanced Computer-Aided Design (CAD) and automated design tools in recent decades. MBSE enhances the ability to capture, analyse, share, and manage the information associated with the specification of a product, resulting in the following advantages (Walden et al. 2015):

- Improved communications among the development stakeholders (e.g., the customer, program management, systems engineers, hardware and software developers, testers, and specialty engineering disciplines), often located in varied geographical regions.
- Increased ability to manage system complexity by enabling a system model to be viewed from multiple perspectives and to analyse the impact of changes.
- Improved product quality by providing an unambiguous and precise model of the system that can be evaluated for correctness, and completeness and consistency.
- Enhanced knowledge capture and reuse of the information by capturing information in more standardized ways and leveraging built-in abstraction mechanisms inherent in model-driven approaches. This in turn can result in reduced cycle time and lower maintenance costs to modify the design.
- Improved ability to teach and learn SE fundamentals by providing a clear and unambiguous representation of the concepts.

Although MBSE presents many benefits, transitioning from a document-based approach introduces challenges, including the need for investments in new tools, training for teams, and modifications to project workflows. Nonetheless, the long-term advantages of improved efficiency (Borky and Bradley 2019), enhanced quality, and decreased risk surpass these initial costs. (Ramos, Ferreira, and Barcelo 2012)

Another crucial aspect of MBSE is its capability to integrate the system model with models from other engineering disciplines, such as mechanical, electrical, and software engineering. This integration facilitates a comprehensive system-level view and supports detailed analysis and simulation activities.

MBSE revolves around the next three critical components (Friedenthal, Moore, and Steiner 2014):

- System Model: At the heart of MBSE is the system model, which encompasses specifications, designs, analysis, and verification data. This model is dynamically updated and refined using model-based tools and methods.
- Model Repository: All model elements (requirements, designs, test cases, etc.) are stored in a structured model repository that allows for efficient management and a more uniform and standardized development process.
- Tools and Execution Environments: MBSE requires robust modelling tools that support the creation, modification, and analysis of system models. Some tools also support direct execution of models to simulate system behaviour.

The Agile MBSE Architecture, illustrated in Figure 7, is visualized as a series of interconnected pyramids, each representing different views and stages of system development. This structure effectively organizes and manages various system aspects.

The Requirements View captures all necessary system requirements, including functional requirements, performance constraints, and interface requirements. Use cases in this view represent the functional interactions between users and the system, depicting expected system behaviour in different scenarios. (Friedenthal, Moore, and Steiner 2014)

The Logical View represents the system's logical functions and their interactions. This includes logical control, logical data, and the logical sequence of operations necessary to meet the requirements. (Holt and Perry 2010)

The Physical View translates the logical architecture into physical components and connections. It encompasses hardware design (HW Design), mechanical engineering design (ME Design), and software design (SW Design), among others. (Friedenthal, Moore, and Steiner 2014)

The Integration and Test View ensures that the system meets all performance requirements. A testing strategy is established to verify system performance through specific tests, including unit tests, integration tests, and system tests, which assess how well the system performs under different operational conditions. (Estefan 2007, 14-26)

Work Artifacts are the tangible outputs produced during the development process. These include models, diagrams, documents, and other deliverables that capture and communicate the system's design and progress. (Holt and Perry 2010)

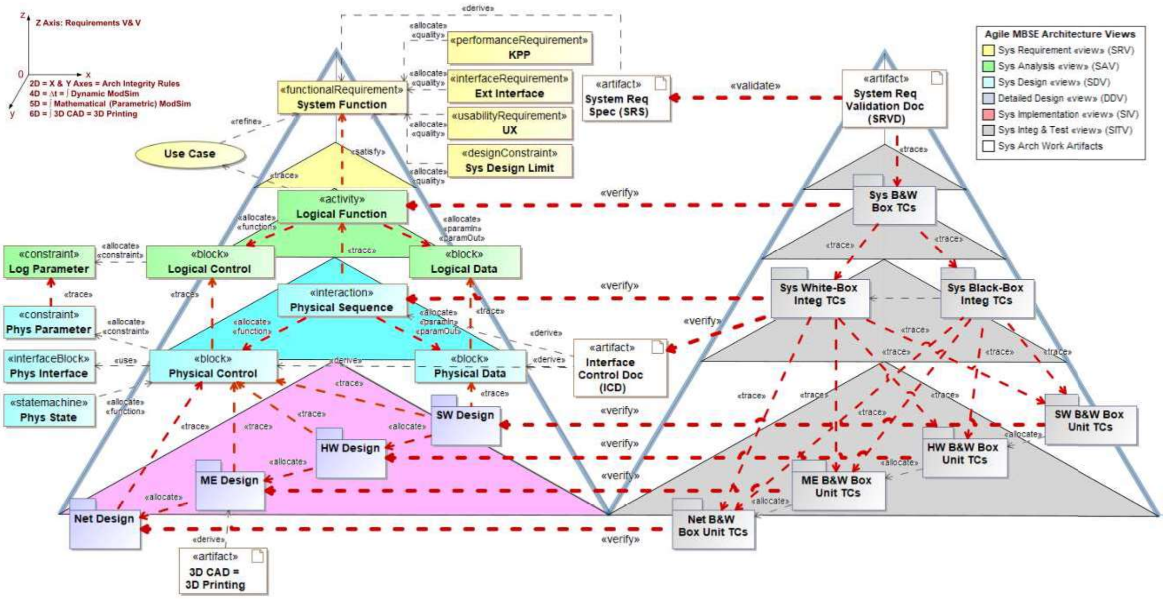


Figure 7: Agile MBSE architecture, source: PivotPoint Technology Corp. 2024

The Agile MBSE Architecture emphasizes iterative development and continuous integration. Each iteration builds on the previous one, incorporating feedback and addressing emerging requirements. The pyramids represent different layers of abstraction, from high-level requirements to detailed physical implementations. The interconnections between these layers highlight the flow of information and dependencies. (Friedenthal, Moore, and Steiner 2014, 46-230)

This architecture supports agile principles by providing a structured yet flexible framework that can adapt to changes and new insights throughout the project lifecycle. Effective collaboration and communication are essential in agile MBSE. The multiple views and artifacts ensure that all stakeholders have access to the necessary information, fostering a shared understanding of the system. (Estefan 2007, 14-26)

2.2.3. Commonly used MBSE tools

MBSE utilizes system models to facilitate various engineering tasks throughout the development process of systems.

There are two main types of models employed in MBSE (Bahill and Wymore 1999):

- Descriptive models: These models outline the logical relationships between system components and functions, detailing the system's logical and physical architecture.
- Analytical models: These models use equations, rules, and direct relationships to represent a system and analyse its characteristics, which are crucial for simulations to verify system performance.

A comprehensive system model often integrates both descriptive and analytical models to capture different dimensions of the system. For instance, system requirements might be specified in a descriptive model and analysed through an analytical simulation model to evaluate their fulfilment. These models also help assess safety, reliability, and performance perspectives.

Several MBSE tools are currently being used to support the MBSE activities. These tools include Cameo Systems Modeller (Dassault Systèmes), Rhapsody (IBM), Capella (Thales), Core (Vitech), ANSYS SCADE, and the Modelica suite.

Most of these tools use the standards modelling languages UML (Unified Modelling Language) and SysML (Systems Modelling Language), with Capella as a notable exception.

UML is used primarily in software engineering to model, visualize, and document the artifacts of both software and non-software systems. It encompasses a range of diagrams categorized into structural and behavioural types, helping to illustrate several aspects of a system. UML is essential for conceptualizing system architectures and facilitates clear communication among various project stakeholders, making it a foundational tool in software development processes.

SysML addresses the limitations of UML in modelling complex systems that may include hardware, software, information, processes, personnel, and facilities. Unlike UML, which is more focused on software-centric applications, SysML is more suitable for the holistic and interdisciplinary nature of modern complex systems in MBSE. SysML offers a broader modelling capability for systems engineering with extensions to handle requirements, parametric relationships (for performance, reliability, and other quantitative analysis), and more comprehensive system behaviour modelling.

Choosing the appropriate MBSE tool depends on the specific application or system under development. To effectively use MBSE, an appropriate combination of the modelling language, tool, and architecture framework is necessary. Additionally, a systematic process for developing system models is essential for ensuring transparency and a shared understanding of the system among engineering teams. Unfortunately, many available MBSE tools do not provide a modelling methodology.

In contrast, Capella offers a comprehensive solution that facilitates designing a system along the entire development process: the ARCADIA (Architecture Analysis & Design Integrated Approach) methodology. Although ARCADIA is not tied to a specific tool, it is supported by the Capella workbench. Capella is developed and maintained by an open-source consortium (PolarSys, Eclipse Foundation) with a broad user base, and it has been successfully used by Thales and attracted industry attention due to its flexibility and integrated capabilities.

Capella enhances user experience by automatically inheriting model information from higher system levels, reducing manual workload, and maintaining model consistency. The Capella workbench includes tools for model development and complexity management, such as filters, replicable elements, copy-paste functionality, and functional decomposition and assignment, providing precise and user-friendly processes. Additionally, Capella manages functional and physical interfaces separately, which streamlines interface integration and simplifies the display, supported by a unique domain-specific modelling language (DSML). However, unlike tools such as MagicDraw (Dassault Systèmes) or Rhapsody (IBM), Capella lacks the capability to perform operational or functional simulation. (Guo et al. 2022, 146-154)

2.2.4. The ARCADIA methodology

ARCADIA (Architecture Analysis & Design Integrated Approach) is a model-based engineering method developed by Thales between 2005 and 2010 for systems, hardware, and software architectural design. Since 2011, it has been applied to numerous projects across various domains – including avionics, rail systems, defence systems in all environments, satellite systems, ground stations, and communication systems – and in many different countries.

ARCADIA is standardized under Z67-140 by AFNOR and establishes a collaborative environment among stakeholders to support a structured engineering process through the different design phases of a product. The system model represents the product and connects models at various levels, facilitating progressive, integrated, and coherent model development across the design process. ARCADIA is designed to define and validate the architecture of complex systems from an early phase to significantly enhance engineering and system integration effectiveness. A clear understanding of the system at all engineering levels is crucial for the early detection of architectural defects and incompatibilities.

This methodology emphasizes the use of functional need analysis to guide engineering activities. This involves converting requirements into functions and defining functional exchanges, states, and data flows. To facilitate user adoption, ARCADIA utilizes engineering concepts and language familiar to users. It also allows for the creation of a variety of diagrams based on the system model, which can represent different system views.

The methodology is characterized by its systematic breakdown of engineering tasks into core activities and perspectives progressively contributing to a solid architecture and analytical model as depicted in Figure 8. Following this, a comprehensive description of each phase within the ARCADIA methodology is provided (Voirin 2023):

- **Operational analysis (OA):** The initial phase of system development focuses on understanding stakeholders' needs to define stakeholder requirements. This involves identifying the system's "actors" and their specific operational requirements, without delving into system behaviour or functionality. The outcome is the operational architecture, which details operational scenarios, constraints, and capabilities, guiding subsequent system design and development.
- **System analysis (SA):** This phase ensures that the system meets the operational needs identified earlier and focuses on assessing the system's capabilities to define specific needs or use cases. It principally involves analysing the black-box functionality of the system, which means detailing the functions the system must perform and their interactions without delving into the specifics of the internal components. SA also develops the information flow between

these functions, which is essential for coherent system operation. Additionally, both functional and non-functional requirements are defined and formalized during this phase, aiding in the creation of an early system design to verify the feasibility of these requirements.

- **Logical architecture (LA):** This phase focuses on white-box analysis to determine how the system operates to meet the required performance identified in both OA and SA. It involves defining and assigning internal logical functions to specific logical components, thus forming a logical architecture that facilitates manageable progression to later development stages. Additionally, a breakdown of components is developed during this phase, considering both functional and non-functional behaviours to establish candidate architectures. This process is essential for designing a system that is both effective and feasible, balancing a range of competing system requirements and constraints.
- **Physical architecture (PA):** This last phase represents the transition from conceptual and logical design to the tangible, focusing on developing a technical solution based on the logical architecture. This phase emphasizes the implementation of physical components that perform the system's functions, allocating logical components into physical ones while considering technical and technological constraints. It outlines how the system is developed and built, guiding downstream engineering teams involved in component production, integration, and validation. A robust logical architecture can lead to multiple feasible physical architectures.

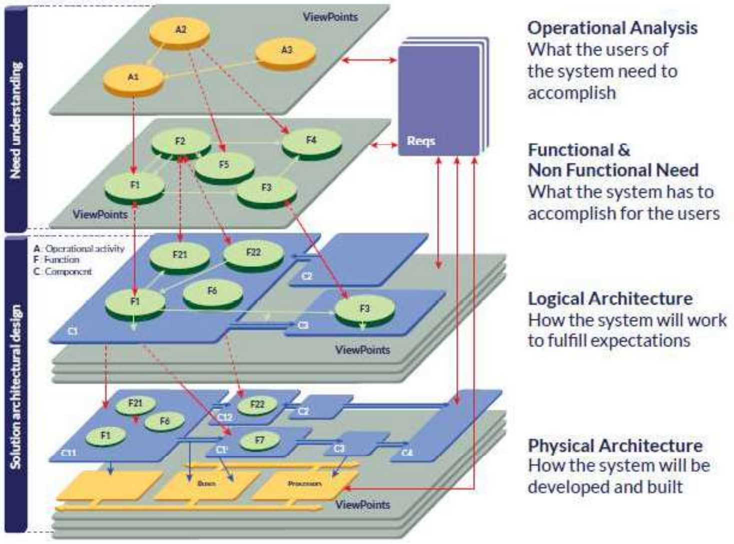


Figure 8: ARCADIA methodology overview, source: Thales Group 2024

Each core activity in the development process results in a distinct perspective of the architecture, together creating a detailed model that describes the system's architecture and its analysis. These stages involve rigorous checks for consistency – both external, across various system components and requirements, and internal, within the system's own architecture and design – to ensure that the solution is coherent, comprehensive, and aligned with the needs of users and stakeholders.

The architecture's compliance with all operational and system requirements is ensured through detailed multi-disciplinary analyses. These involve specialty and discipline-specific evaluations that assess the architecture's performance against a range of technical, operational, and functional expectations, helping to optimize it.

The ARCADIA methodology is designed with flexibility, allowing the sequence of activities to be adjusted based on real-world conditions and specific project demands. It incorporates iterations and feedback loops, which are vital for adjusting as new insights emerge throughout the development process. ARCADIA supports various procedural approaches, including top-down, bottom-up, and iterative processes, making it adaptable to diverse project requirements and constraints.

In summary, ARCADIA is a comprehensive, adaptable engineering methodology that methodically covers the entire lifecycle of system development from operational analysis to physical implementation. It emphasizes collaboration, consistency, and adaptability, ensuring the development of functional, efficient, and effective systems.

2.3. Mechatronic systems

The term "Mechatronics" was originally created in 1969 by Tetsuro Mori of the Japanese corporation Yaskawa Electric, which specialized in building mechanical factory equipment. At that time, Yaskawa Electric Corporation began integrating electronic features into its mechanical equipment and sought a technical term to describe this new technology. Consequently, Mori combined the words 'mechanical' and 'electronics' to create "Mechatronics." Yaskawa Company applied to register this word as a trademark and obtained the rights in 1972. The term officially appeared in dictionaries in 2005 and is now recognized as an essential term in the industry. (Milecki 2015)

The IEEE defines mechatronics as the “synergistic combination of precision mechanical engineering, electronic control, and systems thinking in the design of products and manufacturing processes" (Milecki 2015). While various definitions exist, they all agree on the interdisciplinary nature of mechatronics, involving mechanical systems (mechanical elements, machines, precision mechanics), electronic systems (micro-electronics, power electronics, sensors, and actuators), and information technology (systems theory, automation, software engineering, artificial intelligence), as illustrated in Figure 9.

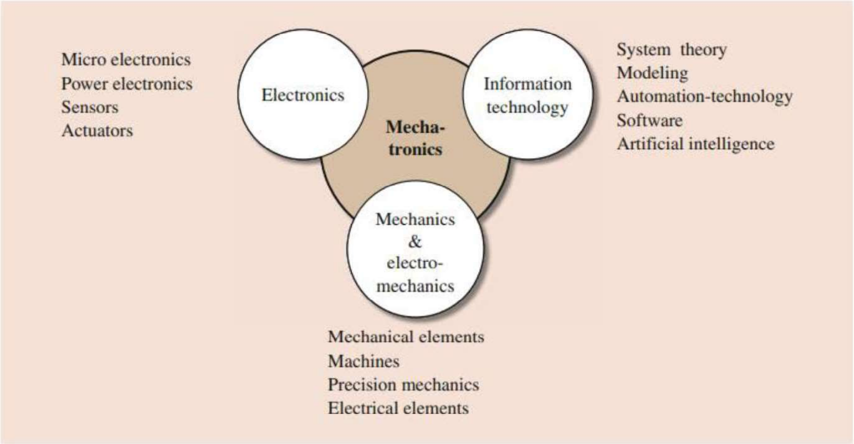


Figure 9: Synergetic integration of different disciplines in mechatronics, source: Nof 2023

Mechatronic systems are prevalent in various industries, including automotive, aerospace, manufacturing, and railways, due to their ability to enhance functionality, precision, and automation. Over the last decade, these systems have become increasingly complex, integrating more components and diverse technologies. System engineers aim to build the right systems correctly and on time while reducing costs.

The versatility and impact of mechatronic systems underscore their importance in advancing modern technology and innovation. These systems improve performance, add functionality, and provide autonomy, addressing the increasing demands of contemporary markets. Techniques aimed at improving system design are crucial for mastering the product life cycle, ensuring that mechatronic systems remain efficient and relevant.

In summary, mechatronics combines mechanics, electronics, and real-time computing, representing a rapidly developing and interdisciplinary field of engineering. The integration of these disciplines leads to the creation of sophisticated systems that enhance modern technology and innovation.

2.3.1. Evolution and design of mechatronic systems

Mechatronics is a multidisciplinary field that merges principles from mechanical engineering, electrical engineering, systems engineering, robotics, electronics, computer science, telecommunications, control engineering, and product engineering. This field focuses on the integrated design of products that combine mechanical and electronic components with intelligent control systems, aiming to create efficient, adaptable, and innovative solutions.

The development of mechatronic systems can be traced back to early mechanical systems that incorporated basic electrical components for enhanced control. The 20th century witnessed significant advancements, particularly with the advent of microcontrollers and integrated circuits, which played a crucial role in developing more sophisticated systems.

Early mechatronic systems featured simple electronic control units integrated into mechanical systems. This period laid the groundwork for more complex integrations in the following decades. The 1980s and 1990s saw rapid technological advancements, including the development of advanced microprocessors, sensors, and actuators. These technologies enabled the creation of more complex and efficient mechatronic systems, significantly expanding their capabilities and applications.

In recent years, the integration of artificial intelligence (AI) and machine learning has revolutionized mechatronics. These technologies allow systems to learn from data, adapt to changing conditions, and make intelligent decisions, further enhancing their capabilities. However, the development of mechatronic systems also presents challenges, such as the need for multidisciplinary expertise, complex system integration, and ensuring reliability and safety.

Mechatronics has evolved through various industrial revolutions, each contributing to its development and expansion:

- Industrial Revolution: introduced mechanization, waterpower, and steam power, laying the foundation for modern mechanical systems.
- Semiconductor Revolution: brought mass production, assembly lines, and electricity, leading to significant technological advances in engineering products' design and operation.
- Information Revolution: introduced computer automation, significantly impacting mechatronics by enabling more sophisticated control and integration of systems.
- Fourth Industrial Revolution: characterized by cyber-physical systems, has further pushed the boundaries of mechatronics, integrating advanced computing and communication technologies into engineering systems.

Mechatronic systems have developed through various stages, each representing a significant step in their complexity and capabilities (Sima 2022, 61-69):

- Primary Level: integrates mechanical signalling with mechanical action at the basic control level, such as fluid valves, switches, and relays.
- Secondary Level: incorporates microelectronics into electrically controlled devices, such as cassette recorders.
- Tertiary Level: involves advanced control strategies using microelectronics, microprocessors, and application-specific integrated circuits, such as microprocessor-based electric motors used in robotics.
- Quaternary Level: aims to improve system intelligence by introducing artificial neural networks and fuzzy logic, enhancing the system's ability to detect and isolate errors.

The design approach in mechatronics integrates sensors, actuators, signal conditioning, power electronics, decision-making algorithms, and computer hardware/software to create efficient and innovative systems.

Concurrent engineering and model-based design are essential for reducing time-to-market and improving design efficiency. This approach emphasizes the simultaneous development of different system components, rather than sequentially, to optimize performance and cost. Solutions are sought simultaneously across all domains, focusing on innovative approaches that optimize performance and cost. This involves evaluating possibilities in distinct parts of the systems and considering global optimization. Optimal design in mechatronics seeks synergistic and optimal solutions that cross domain boundaries rather than sequentially optimizing each domain.

Model-based design is a crucial aspect of mechatronic development, involving several phases from conceptual design to detailed subsystem design and controller design. The conceptual design phase involves evaluating different solutions based on system requirements and functions. Simulations with simple models capture the main dynamics, ensuring a close relation to physical reality. In the detailed design phase, more precise models are used for subsystems that need special attention. These models are often port-based to support reusability and polymorphic modelling. Controller design typically uses linearized models to simplify the complex non-linear models from previous phases. This phase focuses on ensuring that the control system meets performance specifications while maintaining a relation to the physical reality of the system.

Simulation software, such as MATLAB and Simulink, plays a vital role in the design and optimization of mechatronic systems. These tools enable engineers to test and refine designs before physical implementation, ensuring efficiency and effectiveness. Virtual environments also allow the evaluation of system components in a simulated setting. This approach is essential when neither the controller hardware nor the mechanical system is ready for physical testing, enabling the validation of design concepts and performance.

The demand for mechatronic systems continues to evolve, driven by market requirements and technological advancements. Emphasis on mastering the product life cycle and improving design techniques is vital in meeting these demands and advancing the field of mechatronics.

2.3.2. Mechatronic system components

The term "mechatronics" is a classification based on the technology used. From a functional point of view, a mechatronic system can be seen as a collaboration of two main parts:

- Operative part: This consists of mechanical and electromechanical components.
- Control part: This is made up of electronic, computer, and automatic technologies. It can be an open-loop or closed-loop control system.

Figure 10 illustrates the mechatronic process for a closed-loop control system.

A typical mechatronic system consists of (Van Amerongen 2007):

- Physical part, with at least electrical and mechanical components (components in other domains may be present as well).
- Sensors and actuators that couple the physical domain with the information processing domain.
- The information processing domain, realized in the form of analogue or digital electronics. In the latter case, the data processing takes place in the software.

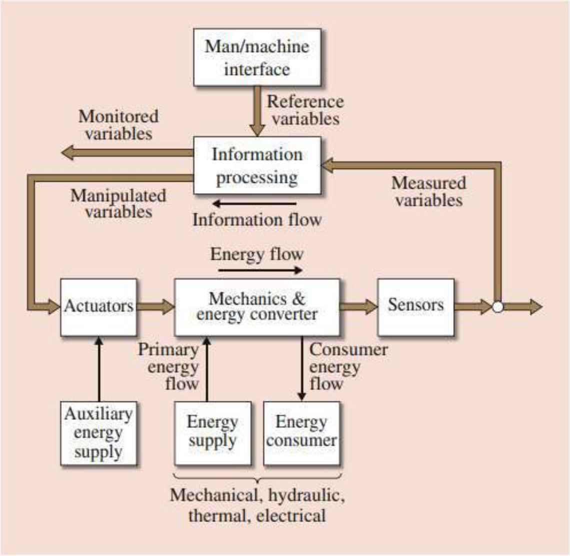


Figure 10: Schematic diagram of a mechatronic system process, source: Nof 2023

Most mechatronic systems involve motion, making actuators essential components. Actuators enable movement or action by converting energy into motion and force, resulting in acceleration and displacement. They require physical power to perform this function. There are diverse types of actuators, including hydraulic actuators, which use fluid pressure; pneumatic actuators, which utilize compressed air; and electric actuators, which rely on electrical energy. Each type is suited for specific applications based on the desired speed, force, and precision.

Sensors are devices, modules, machines, or subsystems of mechatronic systems, responsible for collecting data, detecting events or changes from the environment and providing feedback to the controllers. Several types of sensors are used, including temperature sensors, crucial in applications like HVAC systems and industrial processes; pressure sensors, essential for applications such as hydraulic or pneumatic systems; and position sensors, used in applications like robotics and automation systems.

Controllers process data from sensors and execute control algorithms to manage the behaviour of the system. Common controllers include PID controllers, which utilize proportional, integral, and derivative control actions to maintain desired system behaviour; microcontrollers, which are compact integrated circuits that perform computational tasks and control processes in real time; and Programmable Logic Controllers (PLCs), used in industrial automation for controlling machinery and processes. Controllers are increasingly incorporated into electromechanical devices, providing greater flexibility and control in system design. Software provides the platform for programming in case the controller is digital. The most common programming languages used for developing control algorithms and system software are C, C++, and Python.

Effective communication between components is essential for the seamless operation of mechatronic systems. The exchange of information between two components of the system is possible if there is a communication in common parameters, known as interfaces. The number and design of interfaces within a system significantly influence the simplicity, adaptability, and testability of a system. Interfaces, which are hardware and software, define the functionality of the system by enabling functions to transfer from one component to another.

Common communication protocols include the Controller Area Network (CAN) Bus, widely used in automotive and industrial applications for robust and efficient communication. Ethernet networks provide high-speed data transfer for complex systems requiring substantial amounts of data exchange. Wireless communication enables remote monitoring and control of mechatronic systems, enhancing flexibility and accessibility.

2.3.3. Classification and analysis of actuator technologies

An actuator is a device that converts input energy into mechanical energy. For the mechanical energy to be useful, the actuator must respond to a signal or stimulus and deliver work within a certain time and in a controllable manner. The work produced by an actuator is characterized by force and stroke, which are essential requirements for all actuators. Therefore, an actuator can be seen as a combination of technology and stimulus arranged in a specific configuration.

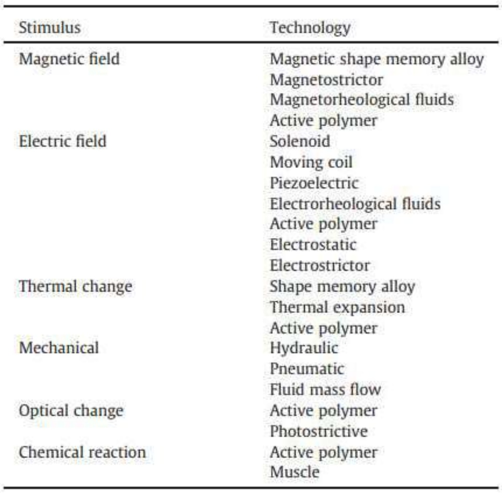
The actual market and research industry offer a wide variety of actuators. Actuators can be classified according to various criteria (Toshiyoshi and Westervelt 2006):

- Active vs. Semi-active: Semi-active actuators only dissipate energy through mechanical interaction with the controlled system, meaning their output mechanical power is not positive. Active actuators can either increase or decrease the energy level of the controlled system, with power flow being either positive or negative.
- Rotational vs. Translational: Translational actuators convert electrical energy into linear mechanical energy, with linear velocity and force as the conjugate variables defining power output. Rotational actuators convert electrical energy into rotational mechanical energy, with angular rate and torque as the conjugate variables.
- Hard vs. Soft: Soft actuators (pulling actuators) can only withstand traction forces and are inherently unidirectional but can be configured in antagonistic pairs for two-way actuation. Hard actuators (push-pull actuators) can sustain both traction and compression forces and are inherently bidirectional.
- Input Energy Domain: actuators can be classified according to the input energy used to deliver work. There are five main input energy domains: electrical, thermal, magnetic, chemical, and fluid.

Other classification criteria for actuators include whether the output motion is continuous or discontinuous.

Figure 11 presents a list of actuator technologies categorized by input domain.

Selecting actuators for a given application can be a complex task. The primary requirement is for the actuator to provide a specific amount of mechanical power. Therefore, the most critical factors in actuator selection are force (or torque), frequency (or speed), and stroke. Additional important considerations include response time, weight, precision, dimensions, and operating conditions such as temperature. (Poole and Booker 2011, 647-661)



Stimulus	Technology
Magnetic field	Magnetic shape memory alloy Magnetostrictor Magnetorheological fluids Active polymer
Electric field	Solenoid Moving coil Piezoelectric Electrorheological fluids Active polymer Electrostatic Electrostrictor
Thermal change	Shape memory alloy Thermal expansion Active polymer
Mechanical	Hydraulic Pneumatic Fluid mass flow
Optical change	Active polymer Photostrictive
Chemical reaction	Active polymer Muscle

Figure 11: Actuator technology categorisation by stimulus type, source: Poole and Booker 2011

Figure 12 provide an overview of the performance domains of existing actuator technologies based on these three fundamental parameters: force, frequency, and stroke:

- Force vs. Stroke: the first chart (top) shows the relationship between the force output and the stroke length for different actuator types. Actuators like hydraulics and pneumatics can deliver high force with relatively large strokes, making them suitable for heavy-duty applications. On the other hand, technologies such as piezoelectric and shape memory alloys provide lower force and shorter stroke, making them more appropriate for precision applications where minimal displacement is required.
- Stroke vs. Frequency: the second chart (middle) plots stroke length against frequency, highlighting the performance characteristics of actuators that must operate at different speeds. Actuators like piezoelectric devices can operate at high frequencies but have limited stroke lengths, making them ideal for applications that require rapid, small movements. Pneumatic and hydraulic actuators, which offer longer strokes but lower frequencies, are better suited for tasks requiring larger displacements at slower rates.
- Frequency vs. Force: the third chart (bottom) illustrates the relationship between frequency and force output. It shows how different actuators perform in maintaining force at various operational speeds. Piezoelectric and magnetostrictive actuators are suitable for high-frequency applications where moderate force is sufficient. In contrast, hydraulic and pneumatic actuators excel in scenarios that require high force at lower operational frequencies.

These charts offer a holistic view of the trade-offs between force, stroke, and frequency for different actuator technologies. This visualization aids in selecting the most appropriate actuator for specific engineering applications by aligning performance characteristics with design requirements, such as the need for high speed, large displacement, or significant force.

For the application of the analysed system in this thesis three main options meet the needs of the system and therefore are considered: electrical actuators (in the previous figures found as “moving coil transducers”), pneumatic actuators and hydraulic actuators.

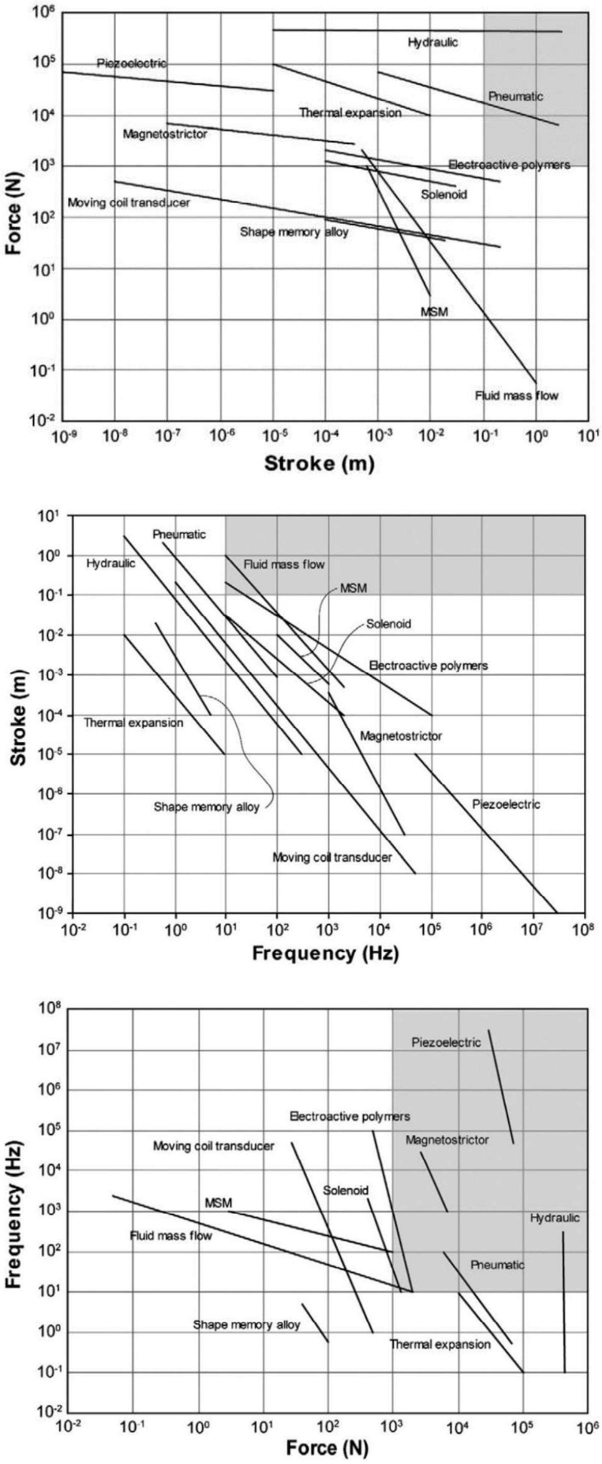


Figure 12: Stroke, force, and frequency relationship for direct single actuators, source: Poole and Booker 2011

2.3.3.1. Electric actuators

Electrical actuators are widely used due to their high efficiency, precision, and scalability. Among the distinct types of electrical actuators, geared drive motors and direct drive motors are two of the most employed options. Both types of actuators operate based on the principle of converting electrical energy into mechanical motion through the interaction of magnetic fields.

Geared drive motors typically consist of an electric motor, such as a DC motor or a servomotor, paired with a gear reduction system (i.e., planetary reduction, harmonic reduction, or cycloidal pinwheel reduction) to increase or reduce torque and control the speed of the output shaft. The gear system enables precise positioning and high torque capabilities while maintaining a compact form factor. However, the use of gears introduces backlash, which can affect the accuracy and repeatability of the actuator. Additionally, gears are subject to wear and tear, leading to reduced efficiency and the need for regular maintenance.

Direct drive motors eliminate the need for a gear system by directly coupling the motor output shaft to the load. This design typically lacks mechanical components such as gearboxes, which are subject to continuous friction, thereby eliminating backlash and reducing mechanical wear and potential failure points. Direct drive motors are often synchronous permanent magnet motors. The synchronous characteristics of these motors mean that the rotor rotates at the same frequency as the magnetic field generated by the stator winding, achieving efficient and accurate speed control. In direct drive designs, the extremely low rotor losses enable higher stator current loading and reduce the motor frame size. The elimination of slip improves both motor efficiency and the dynamic control performance of the entire drive, resulting in increased accuracy, responsiveness, and overall system reliability. The unparalleled precision and speed of direct drive motors have made them the preferred choice for a wide range of applications that demand high accuracy and rapid response. Despite these advantages, direct drive motors often require larger, more powerful motors to produce the desired torque, resulting in increased weight and size, as well as higher costs compared to geared drive motors.

Servomotors are crucial components in many electrical actuator systems, including both geared drive and direct drive motors. A servomotor consists of a motor coupled to a sensor for position feedback. The servomotor is controlled by a sophisticated controller, which uses the feedback sensor to ensure the motor operates at the desired speed and position.

2.3.3.2. Pneumatic actuators

Pneumatic actuators are widely used in various industrial automation applications due to their unique combination of reliability, flexibility, and cost-effectiveness. These actuators convert compressed air energy into linear or rotational mechanical movements, enabling precise control of position, speed, or force of output elements. Pneumatic actuators offer several advantages, such as fast response times, a high power-to-weight ratio, and intrinsic safety, as they use non-explosive, non-toxic air as a driving force. Despite their sensitivity to air pressure fluctuations and the necessity for reliable air supply systems, they remain popular in applications ranging from simple offline control to complex closed-loop systems.

However, pneumatic actuators also have some disadvantages that should be considered when selecting an actuator for a specific application. One significant limitation is their lower energy efficiency compared to electric actuators. The energy conversion process from electrical power to compressed air is inherently inefficient, and air leaks in the system can further exacerbate this issue.

Additionally, pneumatic actuators may exhibit less precise control due to the compressibility of air, making it difficult to achieve the same level of positioning accuracy as electric actuators. Moreover, pneumatic systems can be noisier than other actuation methods, which may not be suitable for certain environments. The noise produced by pneumatic actuators primarily results from the venting of exhaust air, which can be mitigated to some extent with the use of silencers or mufflers.

Despite these disadvantages, pneumatic actuators remain an indispensable component in many industrial automation systems, with their suitability determined by the specific requirements of each application.

2.3.3.3. Hydraulic actuators

Hydraulic actuators are devices that produce mechanical movements and forces using pressurized hydraulic fluids. They are commonly used in various industrial and automotive applications, offering high force and speed capabilities as well as precise motion control. The main functions of hydraulic transmission systems include transmitting and moving hydraulic oil, directing it into the oil cylinder cavity, and controlling the expansion or retraction of the oil cylinder piston rod to perform various tasks. Due to the incompressible nature of liquids, hydraulic actuators can exert significant forces. However, this same property often results in slower movements.

The advantages of hydraulic actuators include their high-power density, ruggedness, and durability. However, they also have some drawbacks, such as the potential for hydraulic fluid leaks, which can pose environmental and safety hazards. Despite these disadvantages, their versatility and reliability make them a popular choice for many applications, including heavy machinery, construction equipment, and aerospace systems.

Compared to electric motor drive systems, traditional hydraulic systems offer higher power density, making them a preferred choice for many applications. However, typical hydraulic systems using servo or proportional valves often exhibit low efficiency due to high energy loss from throttling. To address this, digital hydraulic technology has been developed, providing benefits such as reduced throttling loss, increased reliability, higher resistance to contaminants, and lower costs.

3. Thesis structure and methodological approach

This section outlines the methodology employed in this thesis, detailing the structured approach and sequential steps necessary to achieve the research objectives. The development of the FluxJet represents a significant engineering challenge due to its complexity as a System of Systems (SoS). Each subsystem must function flawlessly on its own and synergize with others to achieve optimal performance under varying operational conditions. This thesis focuses on the Boom Suspension (BS) subsystems of the FluxJet, hereafter referred to as the System Of Interest (SOI). These subsystems are critical for thrusting the vehicle, stabilizing it during high-speed operations, and supporting power transmission or wheel systems.

Given the variety of payloads and different prototypes to be developed, the development process must be adaptable to meet specific operational and design constraints. This thesis aims to define a systematic framework for analysing, simulating, optimizing, and validating the BS systems within the FluxJet vehicle. Achieving this target for complex mechatronic systems is challenging but highly beneficial. The Systems Engineering approach, specifically using Model-Based Systems Engineering (MBSE) methodologies, is employed to meet this challenge.

As concluded in Section 2.2.2, MBSE improves communication among stakeholders, enhances the ability to manage system complexity by viewing and analysing the system model from multiple perspectives, and boosts product quality through precise, evaluable models. It also enhances knowledge capture and reuse by standardizing information and leveraging model-driven abstraction mechanisms, leading to reduced cycle times. This thesis focuses on reducing development time of the BS by creating a reusable, parameter-based design foundation for these systems.

The principal task of this thesis is to apply Model-Based Systems Engineering (MBSE) to the Boom Suspension (BS) system through a systematic application of the V-Model in Systems Engineering, complemented by the ARCADIA methodology. This involves creating a system model essential for a structured development process based on specified parameters. Section 4 provides a detailed examination of the system model.

First, a thorough analysis of the system's requirements and constraints is conducted, involving the identification and documentation of stakeholder requirements, definition of system requirements (both functional and non-functional), and determination of design constraints and operational parameters. This step establishes a clear understanding of the system's needs and limitations.

Secondly, the system is analysed as a black box, defining the functions it must perform and their interactions without delving into the specifics of internal components. System requirements and design constraints are defined and formalized during this phase, aiding in the creation of an early system design to verify the feasibility of these requirements.

Next, the system is analysed as a white box to determine how it operates to meet the required performance. In this stage, a breakdown of components is developed, identifying which components perform which functions. Defining the logical architecture involves evaluating the numerous configurations of the system to determine the most efficient one. A tool is essential for assessing different configurations and identifying locally optimal solutions, ensuring that the chosen design meets all requirements while being optimized for efficiency and performance. Section 5 delves into the development of this tool.

Because the MBSE tool Capella does not support simulations or calculations of the system, MATLAB is used to create this tool. A MATLAB code is developed to enable system engineers to analyse the

BS system behaviour based on design parameters. Calculations are carried out to model the system's logical architecture, providing insights into its behaviour and performance. These calculations help visualize and analyse the system’s logical structure and expected performance under various conditions.

The MATLAB code also enables the system designer to perform a trade-off analysis to evaluate different design options and select the most optimal solution. This process ensures that the chosen design balances all critical factors and meets the system requirements efficiently. Three configurations are calculated in this thesis, and these calculations are validated through an analysis of the solutions.

A demonstration of the tool's capabilities is also provided. The design team can extend this tool to calculate more viable solutions, demonstrating its utility in achieving the primary goal of reducing development time for the BS systems. The design team is also able to apply this tool to develop BS systems for new prototypes needed in the project. A similar tool could be used for other subsystems of the vehicle.

By detailing the development process and simulation outcomes of the BS subsystems, this thesis contributes to the broader field of mechatronic system design in transportation. It demonstrates the feasibility of using MBSE for a more systematic and rapid development of the system, offering a methodological approach for future innovations in mechatronic systems. This thesis aims to inspire the creation of tools that allow rapid system development, which is crucial for companies competing in the development of disruptive technologies.

4. System model

This section focuses on developing a comprehensive system model of the Boom Suspension (BS) system. The term "model" encompasses all entities, relationships, diagrams, and architectures created during the system analysis using the Capella software, following the ARCADIA methodology. This model provides a structured representation of how the system functions and interacts with its environment, aiming to establish an adaptable framework applicable to different prototypes and payloads.

Before delving into the system analysis, an introduction to the BS system is provided, offering detailed explanations of its various components and configurations. This foundational knowledge is essential for understanding the subsequent analysis phases.

The development process begins with an Operational Analysis (OA), which captures and defines the system's needs from the stakeholders' perspective. This analysis is fundamental for identifying the system's capabilities, activities, and requirements, forming the basis for subsequent design phases.

Next, the System Analysis (SA) phase translates these operational needs into specific system requirements, establishing clear boundaries and consolidating the system's objectives.

Following this, the Logical Analysis (LA) phase breaks down the system into manageable components, defining their functions and interactions. This phase involves evaluating different logical architectures to determine the most efficient configuration. A tool is developed within this thesis to assist in this evaluation, enabling the analysis of various configurations to identify the optimal solution.

Finally, the Physical Analysis (PA) phase, although not the primary focus of this thesis, outlines the transition from logical to physical architecture.

By systematically addressing each phase of the development process, this section aims to provide a clear and structured approach to designing the BS system, ensuring it meets all operational requirements and constraints effectively.

4.1. Introduction to the BS System

The BS system is crucial for deploying and retracting various peripheral systems – collectively referred to as payloads – located outside the vehicle's fuselage. These payloads include linear induction motors, which are essential for propulsion and levitation, power transmission systems, and wheel systems. The peripheral systems are deployed to the operational position, which is close to the guideway. The guideway has a reaction surface, and the payload interacts with this reaction surface to achieve its functionality. This functionality varies depending on the specific payload.

The FluxJet is equipped with multiple boom suspension systems, each individually designed to securely hold and robotically manoeuvre its payloads according to their specific requirements and operational needs. As such, different configurations of the BS system are necessary to successfully accomplish the diverse functionalities and constraints of each payload.

Additionally, the BS system interacts with other components of the vehicle, such as the control computers, the vehicle structure, or the thermal management system, forming an integrated network that supports the FluxJet's operations.

The boom suspension systems serve five primary objectives:

1. Deploy and retract the payload between the retracted position, adjacent to the fuselage, and the standby position, adjacent to the guideway.
2. Move the payload continuously to conform to changes in guideway surface as the vehicle moves forward at high speed, enabling the fuselage to maintain a straight trajectory.
3. Passively yield during collisions between a payload and the guideway to avoid damage to both the fuselage and payload.
4. Cause the fuselage to execute a roll motion to bank into turns (only for Linear Engines)
5. Perform special manoeuvres when passing specific areas in the guideway, such as areas under maintenance or obstacles.

Examples of the different BS systems include:

- LE.BS (Linear Engine): the LE, illustrated in Figure 13, includes several components: a linear motor (LM), a boom suspension (BS), and a few other subsystems. The dot-notation indicates that the BS is a subsystem of the LE. In this case, the LE.BS is a specific design of BS that is adapted for the LE and is different from other BS's such as the AWS.BS.

Figure 13: Linear Engine system of the FluxJet, source: TransPod Inc. 2024

- PPP.BS (primary power pickup): non-contact electrode system, illustrated in Figure 14, that generates plasma in the gap between the electrode and the power rails.

Figure 14: Primary Power Pickup system of the FluxJet, source: TransPod Inc. 2024

- IPP.BS (intermediate power pickup): contact power brush pressed against a power rail.
- ZPP.BS (high-Z power pickup): partial contact power brush.
- AWS.BS (Arc Wheel System): the AWS is a wheel system, illustrated in Figure 15,used while the levitation operation is off or for safety reasons.

Figure 15: Arc Wheel System of the FluxJet, source: TransPod Inc. 2024

Certain BS systems (LE.BS and AWS.BS) carry structural load forces from their payload components. From one perspective, this type of BS moves its own payload (particularly when the payload is not active). From another perspective, the BS is moving the vehicle with respect to the guideway, for example when the payload is busy thrusting against the infrastructure and the force is transmitted through the BS to the vehicle fuselage.

Certain BS systems (PPP.BS, IPP.BS and ZPP.BS) does not carry the weight of the vehicle, but instead simply move power systems in place such as power contacts or non-contact power systems.

Various BS systems are specialized to meet the specific needs of different payloads based on criteria such as load forces, displacement range, and displacement velocity. TransPod has developed, and continues to develop, prototypes at various scales, resulting in an increased diversity of BS system types. Each BS system is subject to a range of design constraints such as weight, spatial requirements, cost, vacuum compatibility, electromagnetic compatibility (EMC), aerodynamic effects, reliability, availability, maintainability, and safety.

Although there are several types of payloads, this thesis endeavours to minimize distinctions among them, aiming to create a universal system development process. However, the most complex system, the LE.BS, is selected as a reference to simplify the adaptation of this process to less complex systems in the future.

4.2. Operational Analysis

The first phase of the development process is the Operational Analysis (OA), which focuses on capturing and defining the needs and context of the system from the perspective of its stakeholders.

The initial step in this stage is to gather and consolidate the operational needs from stakeholders. This involves collecting detailed information about the expectations of various stakeholders, such as users, customers, and other interested parties.

Subsequently, the focus shifts to defining what the users of the system need to accomplish. This involves identifying the specific tasks, goals, and objectives that users must achieve using the system.

Following this, the analysis identifies the entities and activities involved in the system's operation. Entities refer to the components or objects within the system, while activities outline the processes and actions that need to be performed.

By systematically capturing stakeholder needs and defining the system context, the OA stage lays the foundation for a well-designed and functional system that meets the specified requirements and expectations. This phase is critical as it forms the basis for defining the system's requirements and ensuring that the system fulfils the needs and expectations of its stakeholders.

4.2.1. Stakeholder analysis

A stakeholder octopus' diagram, see Figure 16, is used as a visual to map out the relationships and influence of different stakeholders within the SOI. It is used to identify all the parties interested in or affected by the SOI and illustrates how they are connected to the SOI.

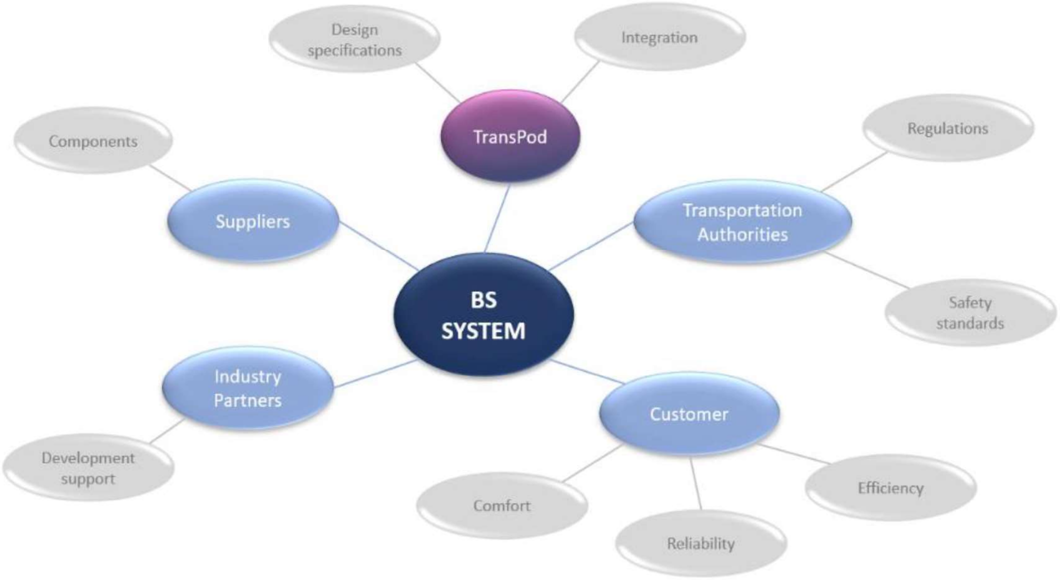


Figure 16: Stakeholder octopus' diagram

Stakeholders have critical roles, including specifying requirements, contributing technical expertise, offering user feedback, etc. Their constraints often involve budget limitations, regulatory compliance, technical feasibility, time restrictions, risk tolerance, etc. Effectively managing these stakeholder functions and constraints is essential for aligning project objectives with practical capabilities and ensuring the successful execution of projects. Therefore Table 1 provides the specific interactions between the SOI and its stakeholders, describing the contributions and exchanges from each stakeholder to the system.

Table 1: Stakeholders' functions and constraints

Stakeholder	Functions	Constraints
TransPod	Ensure system compatibility with overall vehicle design and operation.	Impose functions and requirements of the system.
Suppliers	Provides components and materials that meet system specifications.	Impose constraints based on the availability and limitations of the components or materials they provide.
Industry Partners	Contribute to the development of functional requirements and system specification.	
Transportation Authorities	Ensure the system meets specific safety and regulatory standards.	Impose regulations and safety standards.
Customer	Provides feedback to ensure the system meets performance expectations.	Define the functional performance expectations.

Due to the initial stages of development of the TransPod system, this thesis does not consider all stakeholders for the subsequent analysis. Suppliers for the subcomponents of the SOI have not yet been established, transportation authorities have not been defined as this thesis addresses a novel transport system, and there are no customers involved at this point for the same reason. Consequently, TransPod is the primary and sole stakeholder.

4.2.2. Stakeholder requirements

Stakeholder requirements are derived through close interaction with stakeholders, ensuring their needs are accurately identified and documented. For this project, only a select few stakeholder requirements have been thoroughly examined to maintain a focused scope.

Requirements can be categorized into two types: functional and non-functional (ISO 29148:2018). Functional requirements specify what are the actions that the system should perform, including tasks, data handling, processes, and other specific functionalities necessary for system operation. Conversely, non-functional requirements detail how these functions should be performed, focusing on the system's quality attributes such as performance, usability, reliability, and scalability. Additionally, non-functional requirements may encompass system constraints or standards that must be met, including limitations on weight and size.

Table 2 lists the stakeholder requirements. As mentioned before, this thesis focuses solely on TransPod as the stakeholder, so only its requirements are included. Additionally, a basic approach is taken due to the limited scope of the project.

The stakeholder requirements are translated into system requirements that serve as inputs for the next steps of the development process.

Table 2: Stakeholder requirements list

	ID	Description
Functional	STK_REQ_1	The system shall support mechanical forces generated by the payload or the vehicle.
	STK_REQ_2	The system shall move the payload between the retracted position and the operational position.
	STK_REQ_3	The system shall adapt the position of the payload to the reaction surface.
	STK_REQ_4	The system shall conform the position of the payload to the rotation of the vehicle.
	STK_REQ_5	The system shall avoid damage of the payload or the vehicle in case of contact or collision.
Non-functional	STK_REQ_6	The system shall operate in a vehicle speed range from 0 km/h to 1200 km/h.
	STK_REQ_7	The system shall operate in a temperature range from -30 °C to 50 °C.
	STK_REQ_8	The system shall operate in a pressure range from 8 Pa to 101325 Pa.

4.2.3. Operational capabilities

Three primary actors are identified during the OA phase: the BS system as the SOI, the payload or peripheral system, and the FluxJet fuselage, which encompasses the structural components of the core of the vehicle and all included elements.

Operational capabilities are derived in collaboration with the stakeholder, TransPod, based on the system's main objectives.

Firstly, the SOI is designed to mechanically link the vehicle's fuselage and the payload, carrying various loads. These loads may vary across different BS versions and can include the payload weight, forces due to payload functionality, vehicle weight (particularly for BS systems positioned at the lower part of the vehicle), aerodynamic forces, and inertia forces.

Secondly, the SOI dynamically connects the fuselage and the payload, facilitating payload movement. This involves moving the payload from a retracted position next to the fuselage to a standby position adjacent to the guideway, where the payload is ready for operation. The SOI must continuously adjust the payload position to maintain a specified distance from the designated reaction surface within the infrastructure, compensating for perturbations due to vehicle dynamics, fluid dynamics, or deviations in the FluxWay caused by temperature, wind, precipitation, geological vibration, ground movement, and settling (Janzen 2017, 8-17).

Another crucial capability relates to vehicle turns. To ensure passenger comfort or cargo integrity, the vehicle fuselage must bank upon entering a curve, a manoeuvre achieved through the SOI, when the payload is a linear motor. The rest of the payloads must be displaced with a roll angle relative to the fuselage during the turn.

Additionally, the SOI must protect the integrity of both the vehicle and payload in the event of collisions with objects or the reaction surface.

The resulting operational architecture is depicted in Figure 17.

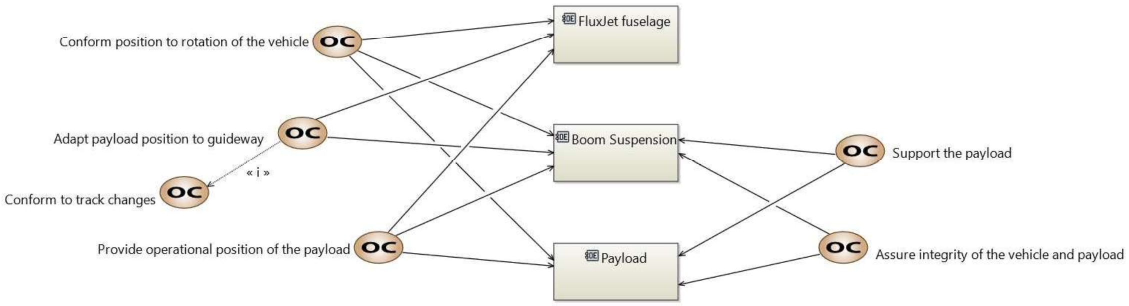


Figure 17: Operational capabilities architecture diagram

4.2.4. Operational activities

Next, the operational activities required to achieve the capabilities identified during the OA phase are detailed. Using the Capella tool, diagrams have been constructed to illustrate these activities for each capability, offering a comprehensive understanding of the interactions and processes involved.

As shown in Figure 18, the BS consistently supports the payload weight, serving as the sole mechanical link to the fuselage. Upon receiving a deploy command, the BS moves the payload to the operational position. When activated, the payload generates forces that vary in magnitude depending on the type of payload installed. For example, linear motors produce remarkably high forces, while power pick-up systems generate low forces. Regardless of the force magnitude, the BS must support these forces effectively. Once the payload is deactivated, the BS reverts to solely supporting the payload weight.

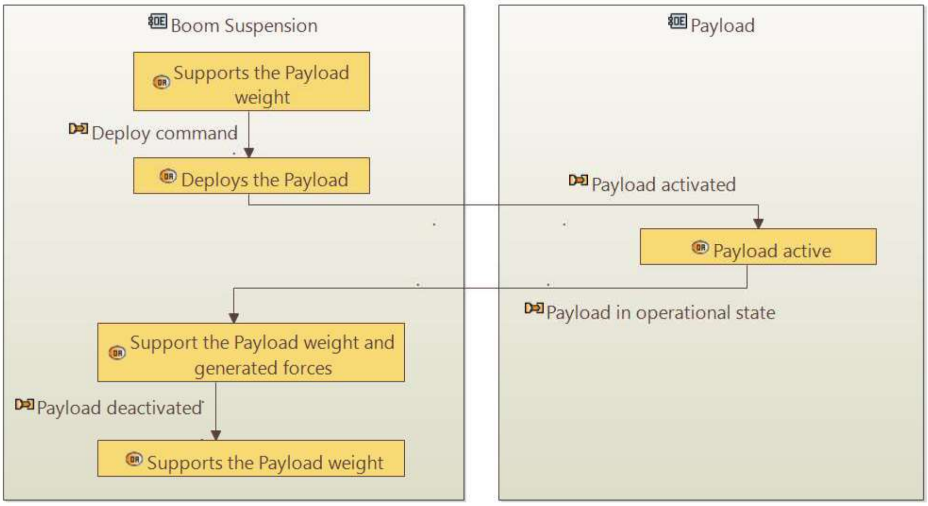


Figure 18: Operational activity diagram for supporting the payload

Figure 19 illustrates the activities involved in positioning the payload, including its deployment and retraction. When the vehicle travels at high speed through the guideway, the BS must continuously adjust the payload position to accommodate guideway deviations and fuselage movements. This adjustment ensures a consistent separation between the payload and the reaction surface.

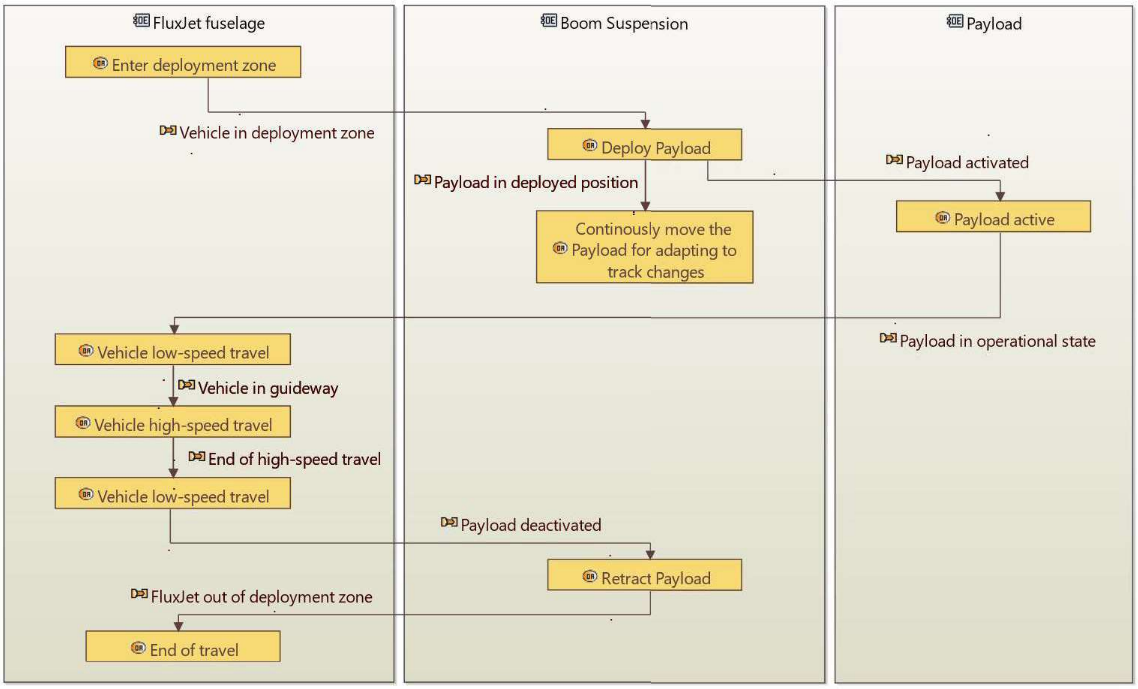


Figure 19: Operational activity diagram for providing operational position of the payload and adapting payload position to guideway

When the FluxJet enters a curve, the BS must adjust the payload, specifically linear motors, to generate forces that induce a roll motion in the fuselage. This adjustment must be maintained throughout the curve until the vehicle exits. These activities are depicted in Figure 20.

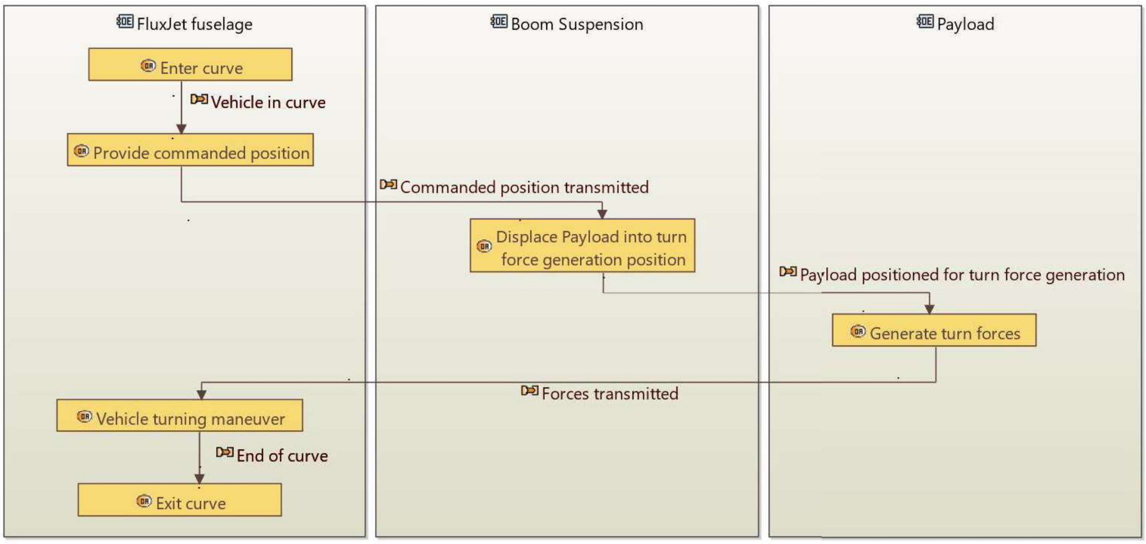


Figure 20: Operational activity diagram for vehicle fuselage rotation[1]

As illustrated in Figure 21, if the payload contacts the guideway or collides with an object, the BS must respond to these undesirable events. Upon encountering such a force, if it exceeds a certain threshold (not necessarily an electronic threshold), the BS yields to mitigate the force, causing the payload to move toward the fuselage. Once the force is no longer present, the payload must be returned to its operational position to resume its function, provided it is not malfunctioning after the event.

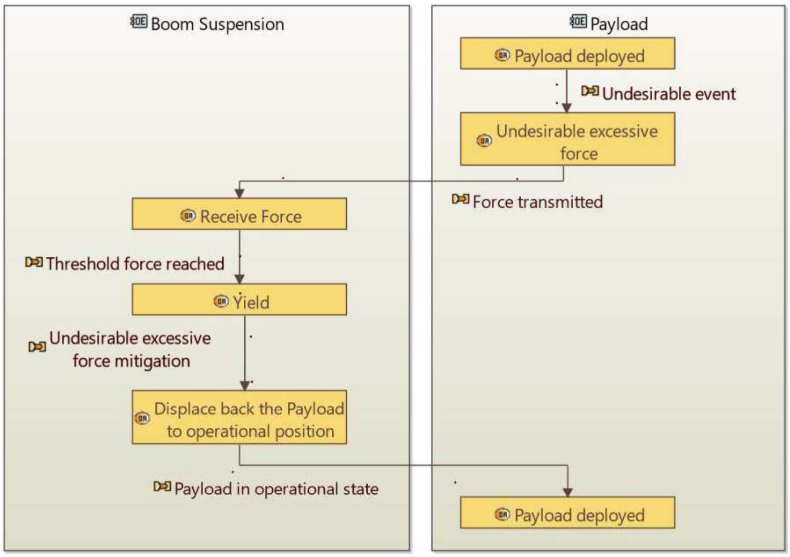


Figure 21: Operational activity diagram for assuring integrity of the vehicle and payload

Defining the activities that the SOI must perform is fundamental for facilitating the identification and specification of its essential functions. This groundwork is critical in the subsequent system analysis, where these functions are thoroughly outlined and examined.

---

[1] Specific diagram in case the payload is a linear motor.

4.3. System analysis

The System Analysis (SA) phase is essential in translating operational requirements into formal system requirements. This section marks the initial step where the focus shifts from solution-neutral operational analysis to defining the specific system under consideration.

The primary objectives of SA include identifying system boundaries to clearly delineate what is included within the system and what lies outside it. This boundary setting is crucial for understanding the scope and context of the system's operation. Another key objective is consolidating requirements by gathering and integrating all relevant stakeholder requirements to ensure the system meets their needs. This involves refining stakeholder requirements into specific, actionable system requirements. Additionally, defining system objectives establishes what the system needs to achieve, specifying the functions the system must provide to fulfil stakeholder needs effectively. Lastly, modelling functional behaviour involves developing models that represent the system's functional architecture to visualize how the system operates and to identify potential issues early in the development process.

This phase emphasizes maintaining an open design approach. While the focus is on defining what the system must do by considering it as a black box, how the functions of the systems are achieved is left for subsequent phases. By concentrating on what the system must accomplish rather than how it is designed, the solution remains flexible and adaptable to various design possibilities.

Before starting the system analysis, it is necessary to explain some fundamental concepts. These concepts are crucial for understanding the system's environment and how it operates. Without a clear understanding of these basics, the system requirements would be difficult to comprehend. Therefore, the next subsection addresses these key concepts to provide a solid foundation for the system analysis.

4.3.1. Fundamental concepts

The FluxJet vehicle follows the ENU axes convention, allowing the vehicle six degrees of freedom in a three-dimensional space. The vehicle can rotate along the normal axis Z (Yaw), transverse axis Y (Pitch), and longitudinal axis X (Roll), while displacing in each respective direction. Figure 22 provides a visual representation of the coordinate system and movements.

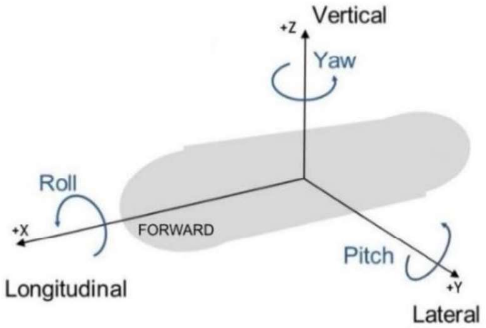


Figure 22: TransPod’s coordinate system

The SOI shall be capable of three distinct motions, as shown in Figure 23,to accomplish STK_REQ_2, STK_REQ_3 and STK_REQ_4:

- Extension/Retraction (ER): This motion extends or retracts the payload in and out from the vehicle, bridging the gap between the fuselage and the guideway.

- Boom Roll (BR): This involves rotating the entire suspension system around a longitudinal axis located at the base of the BS system. This motion is necessary to compensate for the vehicle's roll movement, as the guideway remains fixed.
- Wrist Roll (WR): This motion rotates the payload around a longitudinal axis at the end of the BS system. It is essential because, during a boom roll motion, the payload no longer faces the reaction surface. This motion ensures that the payload remains parallel to the reaction surface.

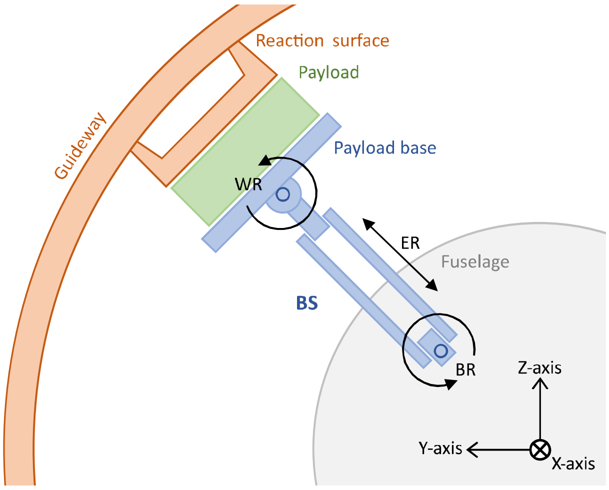


Figure 23: Motions of the BS system

The SOI shall support different forces to accomplish STK_REQ_1. These forces can be the vehicle weight, the payload (PL) weight, forces generated because of the functionality of the payload or other minor forces as aerodynamic or inertia forces. To reduce complexity all these forces except for the payload weight are summarized into three orthogonal forces illustrated in Figure 24: longitudinal force, radial force, and centrifugal force.

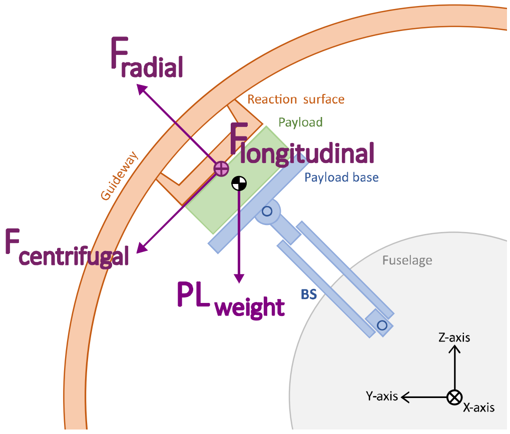


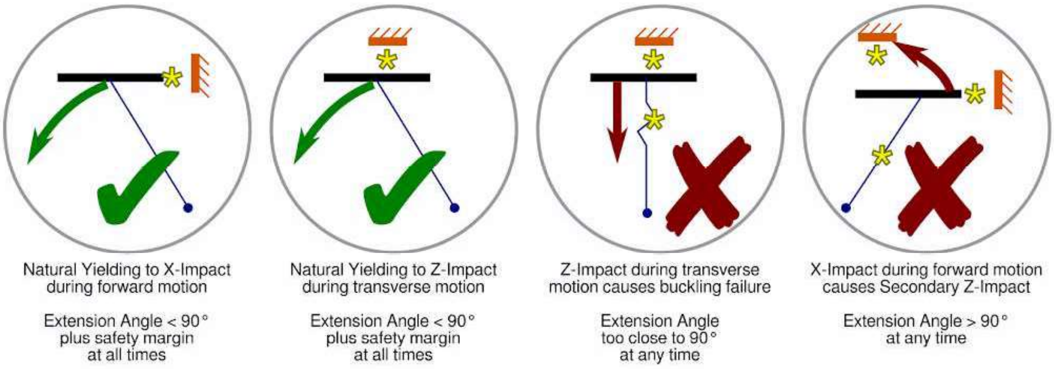
Figure 24: Payload forces

The SOI shall implement a passive yielding function to accomplish STK_REQ_5. Yielding shall protect against two types of faults:

- Longitudinal collisions in the x-direction
- Transverse collisions in the y- or z-directions

The cause of the impact or contact could be caused by debris on the guideway, broken offset guideway components sticking out of position or erroneous extension of the BS.

Figure 25 illustrates collisions in different directions and how the system should react.



Natural Yielding to X-Impact during forward motion
Extension Angle < 90° plus safety margin at all times

Natural Yielding to Z-Impact during transverse motion
Extension Angle < 90° plus safety margin at all times

Z-Impact during transverse motion causes buckling failure
Extension Angle too close to 90° at any time

X-Impact during forward motion causes Secondary Z-Impact
Extension Angle > 90° at any time

Figure 25: Yielding function description

4.3.2. System requirements

Stakeholder requirements are translated into system requirements now that the operational environment where the system functions is well understood. The system requirements are designed to be unambiguous, clear, unique, consistent, stand-alone, and verifiable (ISO 29148:2018). This parameter-based approach is fundamental for the subsequent development of the system.

Some of these parameters are defined by the specific requirements of the payload functionality or the vehicle's operation, such as the forces generated by the payload or the range of movement of the vehicle. Others must be calculated based on these initial parameters, such as the motion ranges of the system itself. One of the goals of this thesis is to assist in calculating these parameters. All variable parameters are indicated as “XXX” because a parameter-based approach is being developed that can be used for different variants of the SOI. For each variant different parameter values can apply.

The identified system requirements are listed in Table 3. While this list could be much more extensive if requirements for maintainability, reliability, regulations, etc., were included, this thesis focuses specifically on the functional system requirements.

Table 3: System requirements list

ID	Description
SYS_REQ_1	The system shall operate in a vehicle speed range from 0 km/h to 1200 km/h.
SYS_REQ_2	The system shall operate in a temperature range from -30 °C to 50 °C.
SYS_REQ_3	The system shall operate in a pressure range from 8 Pa to 101325 Pa.
SYS_REQ_4	The system shall forward the generated forces by the payload to provide the longitudinal motion of the fuselage.[2]
SYS_REQ_5	The system shall forward the generated forces by the payload to provide the roll motion of the fuselage.[2]
SYS_REQ_6	The system shall provide the operational position of the payload for a maximal translational skew (combination of lateral and vertical movement) of the fuselage of XXX mm.
SYS_REQ_7	The system shall provide the operational position of the payload for a maximal roll angle of the fuselage of XXX °.
SYS_REQ_8	The system shall be designed for a vehicle fuselage diameter of XXX mm.
SYS_REQ_9	The system shall be mounted on an angular position of XXX °.
SYS_REQ_10	The system shall be designed for a reaction surface diameter of XXX mm.
SYS_REQ_11	The system shall support the payload weight.
SYS_REQ_12	The system shall support a maximum positive longitudinal force XXX N generated by the payload.
SYS_REQ_13	The system shall support a maximum negative longitudinal force XXX N generated by the payload.
SYS_REQ_14	The system shall support a maximum positive radial force XXX N generated by the payload.
SYS_REQ_15	The system shall support a maximum negative radial force XXX N generated by the payload.
SYS_REQ_16	The system shall support a maximum positive centrifugal force XXX N generated by the payload.
SYS_REQ_17	The system shall support a maximum negative centrifugal force XXX N generated by the payload.
SYS_REQ_18	The system shall have a passive yielding function against collisions in the longitudinal direction.
SYS_REQ_19	The system shall be capable of absorbing longitudinal collisions energies up to XXX J.
SYS_REQ_20	The system shall have a passive yielding function against collisions in the transverse direction (y- or z-direction).
SYS_REQ_21	The system shall be capable of absorbing transverse collisions energies up to XXX J.
ER motion	
SYS_REQ_22	The system shall extend and retract the payload in the transverse direction: ER motion.
SYS_REQ_23	The ER motion shall displace the payload between an extension/retraction distance of XXX and XXX mm.
SYS_REQ_24	The ER motion shall have a linear velocity of XXX mm/s.
SYS_REQ_25	The ER motion shall have an extension/retraction acceleration of XXX mm/s^2.
SYS_REQ_26	The ER motion shall have a precision of XXX mm.
BR motion	
SYS_REQ_27	The system shall rotate the payload around a longitudinal axis, located at the base of the system: BR motion.
SYS_REQ_28	The BR motion shall have an angular displacement between XXX and XXX °.
SYS_REQ_29	The BR motion shall have an angular velocity of XXX °/s.
SYS_REQ_30	The BR motion shall have an angular acceleration of XXX $°/s^2$.
SYS_REQ_31	The BR motion shall have a precision of XXX °.
WR motion	
SYS_REQ_32	The system shall rotate the payload around a longitudinal axis, located at end of the system: WR motion.
SYS_REQ_33	The WR motion shall have an angular displacement between XXX and XXX °.
SYS_REQ_34	The WR motion shall have an angular velocity of XXX °/s.
SYS_REQ_35	The WR motion shall have an angular acceleration of XXX $°/s^2$.
SYS_REQ_36	The WR motion shall have a precision of XXX °.

---

[2] Specific requirement in case the payload is a linear motor.

4.3.3. System architecture

The final product of this design phase is the system architecture diagram. This diagram is an integrative representation of the system analysis. It is important to note that there can be multiple solutions for this diagram; however, clarity in its representation is essential.

Figure 26 illustrates the complete system architecture diagram resulting from the SA. This diagram represents the architecture of a system with a linear motor as a payload, and therefore includes specific functions unique to this type of BS. This choice is made because the LE.BS is the most complex. System architectures for different payloads would be similar but simpler. A larger version of this diagram is available in Appendix 1. The functional architecture is explained in detail using three additional figures (Figure 27, Figure 28 and Figure 29) that highlight different parts of the diagram.

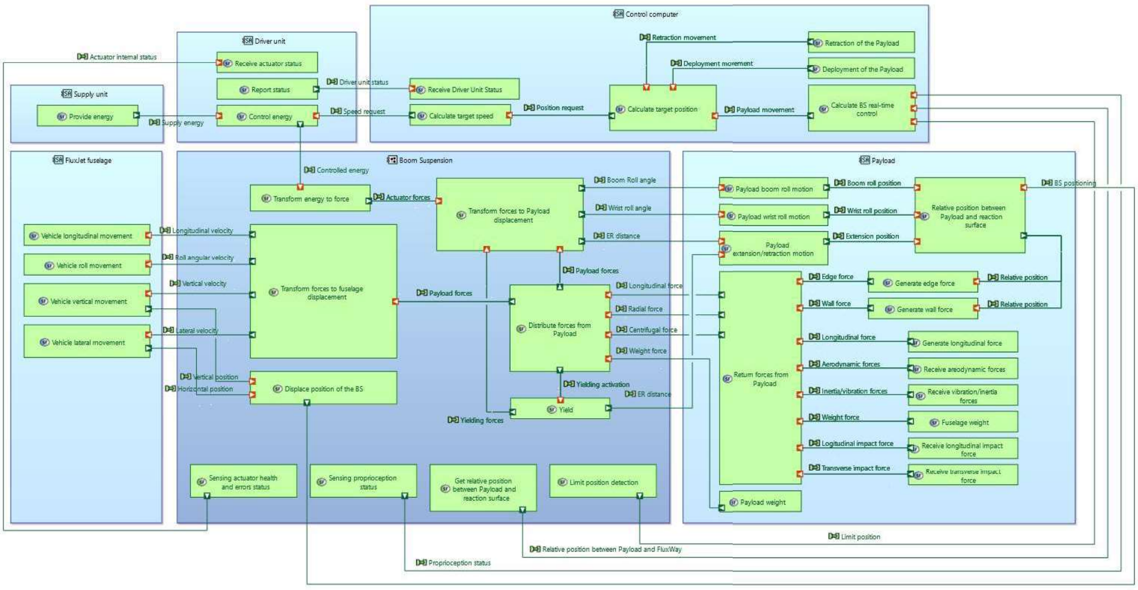


Figure 26: System architecture diagram – overview

The system functionality involves several actors, which are external entities that exchange information, signals, or physical interactions with the SOI. These actors include the BS system, payload, FluxJet fuselage, control computer, driver unit, and supply unit.

The control computer receives position information about the payload from the BS sensors. Using this information, it calculates the speed requirements for the driver units, enabling the movement of the payload to specific positions. A defined gap between the payload and the reaction surface is maintained during operation, even at very high speeds, thanks to a feedback control loop managed by the control computer.

Some positions, such as the retracted and deployed positions, are stationary and do not depend on the reaction surface's actual position. These positions are reached when the payload is not yet in operational state. The control computer also monitors the status of the driver units, for example to react against malfunctions.

While the SA does not define specific solutions to achieve system functionality, actuators have been identified as the logical solution for moving the payload. Although internal actuators are not yet illustrated in the system architecture to maintain a black-box perspective, the external driver units that control them are included. For simplification, a common actor represents the plurality of driver units that might be used.

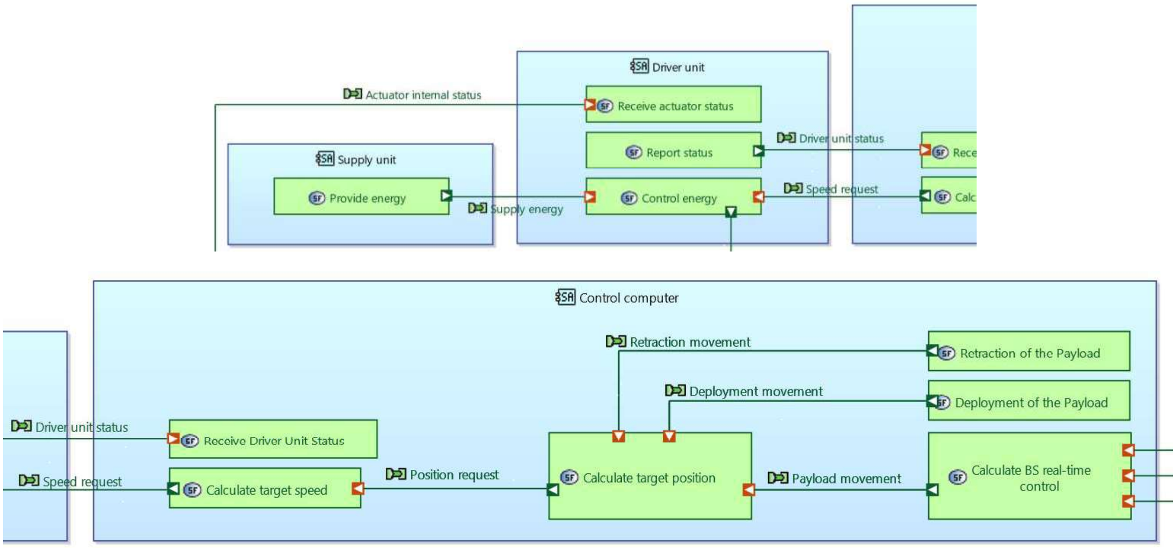


Figure 27: System architecture diagram – detail of supply unit, driver unit and control computer

The driver unit converts control signals from the control computer into the physical motion of the actuator, using energy supplied by the supply unit. Additionally, it receives direct feedback from the actuators regarding their health and error status.

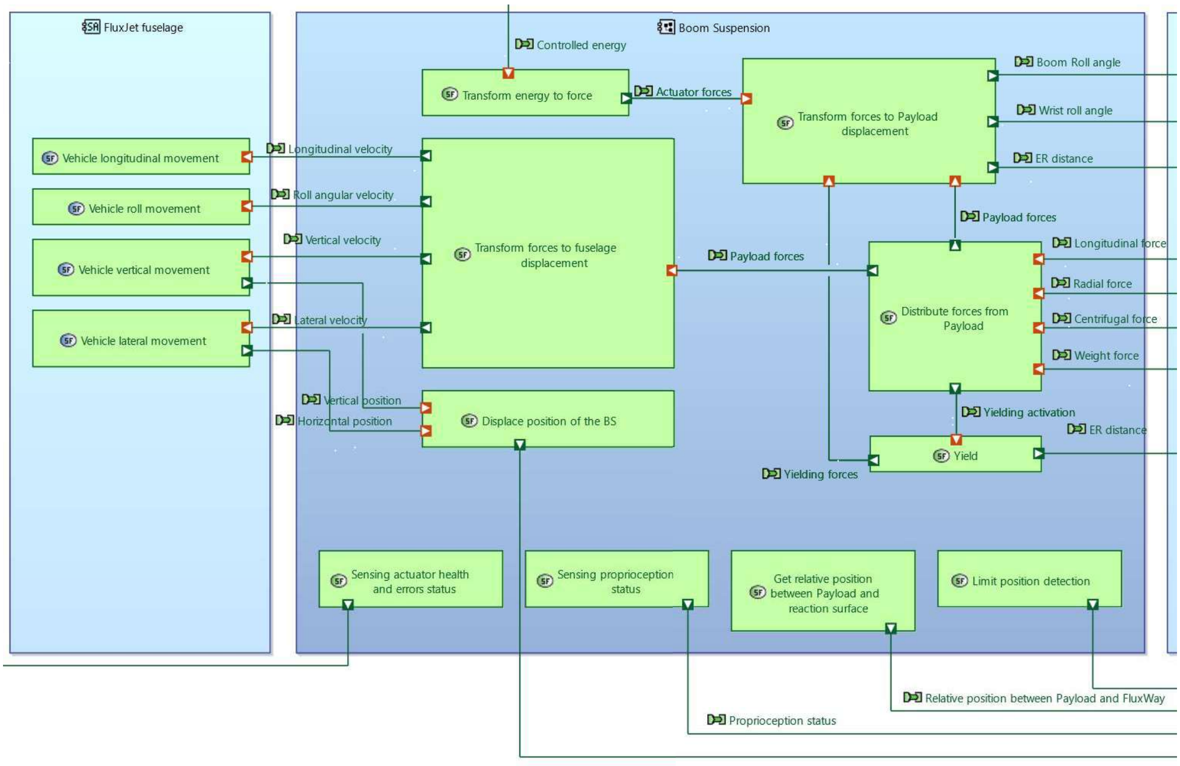


Figure 28: System architecture diagram – detail of SOI (BS) and FluxJet fuselage

The BS system has several critical functions. It must provide the necessary motions – ER, BR and WR – to position the payload accurately and ensure it remains close to the reaction surface. This is achieved by converting the controlled energy supplied by the drivers into forces that cause physical displacement.

Additionally, the BS system must support the forces that move the fuselage. Since the BS system is fixed to the fuselage, any movement of the fuselage changes the position of the BS system (including the payload) relative to the reaction surface.

The BS system also needs to include a yielding function, which retracts the payload in the event of collisions.

Furthermore, some sensors are essential for the system's functionality:

- Proprioception Sensors: These sensors provide critical feedback on the position, movement, and force exerted by the actuator. They help the control system understand the exact state of the actuator, enabling precise control and adjustments. This feedback ensures smooth and accurate movements, enhances the system's responsiveness to changing conditions, and prevents damage by avoiding excessive force or misalignment.
- Distance Sensors: These sensors measure the gap between the payload and the reaction surface, providing data for feedback control.
- Limit Position Sensors: These sensors detect when the payload surpasses predetermined limits, triggering an immediate system reaction to prevent damage.

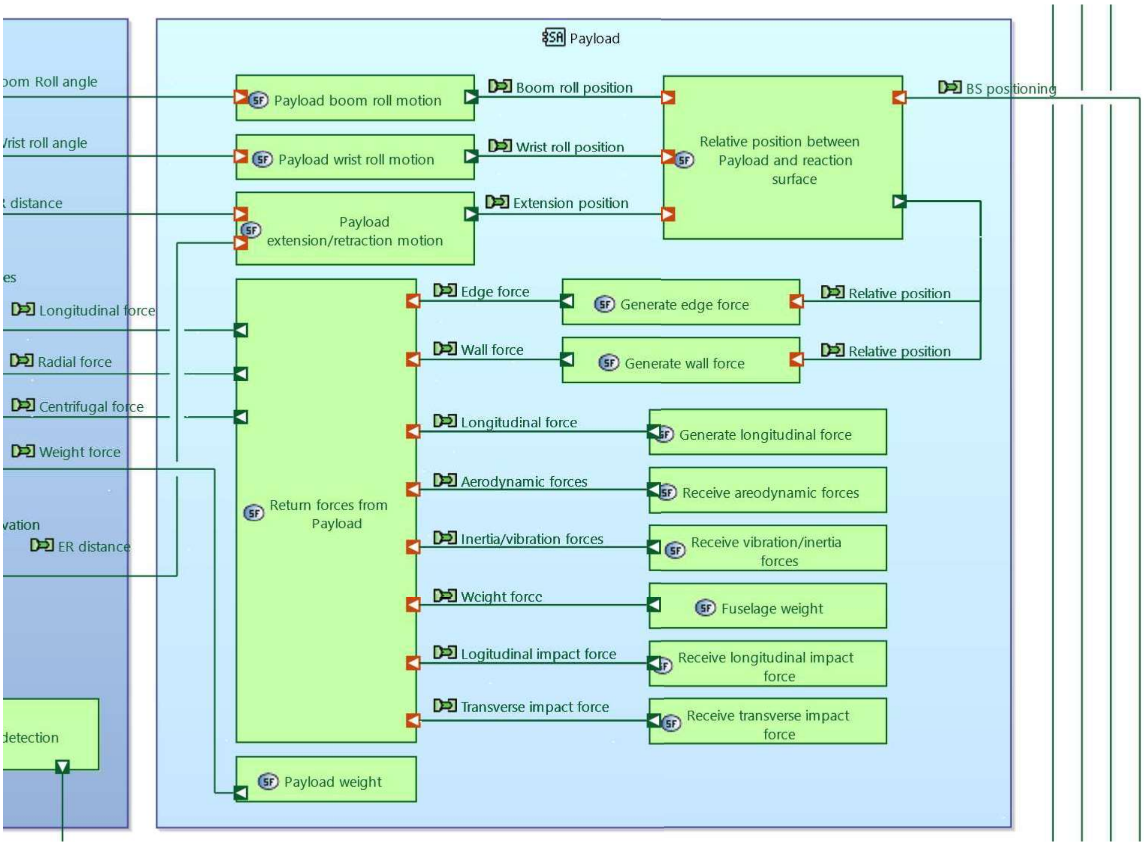


Figure 29: System architecture diagram – detail of payload

The payload generates the edge, wall, and longitudinal forces necessary for moving the fuselage through magnetic field interactions between the payload and the guideway. It is also subject to other forces, such as aerodynamic and vibration forces, and in undesirable situations, collision forces. The weight of the fuselage can significantly affect the BS system, particularly for BS units located on the lower part of the fuselage. All these forces are summarized into three orthogonal forces when analysing the SOI, simplifying its analysis. The weight of the payload is always present, as the SOI must always support the payload, regardless of its position or functioning state.

The following Figure 30 illustrates the system architecture of the SOI, incorporating functional chains. Functional chains depict sequences of functions and are essential for understanding the functional flow and dependencies among various system components. By segmenting the system's operations, functional chains provide a clear view of interrelated processes, enabling detailed analysis of functional dependencies and overall operational efficiency.

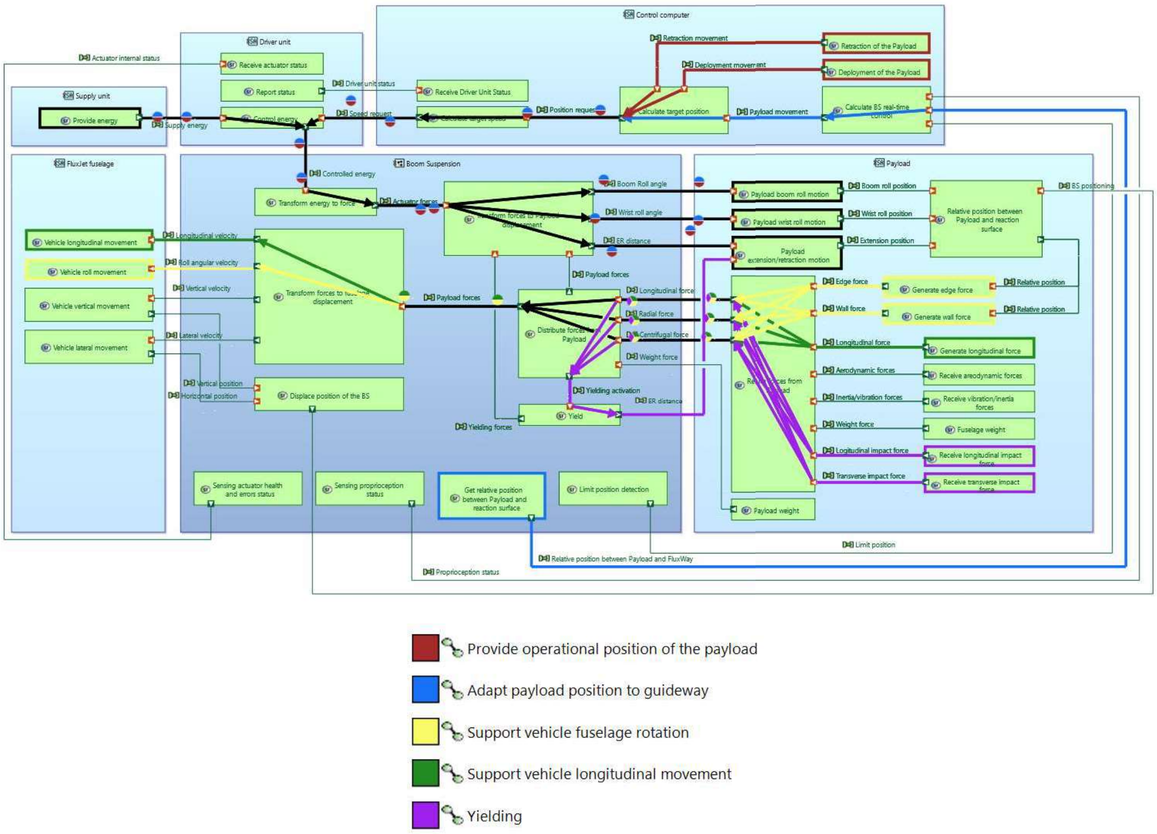


Figure 30: System architecture diagram – functional chains

The system architecture diagram outlines the boundaries of the SOI and clearly defines how the SOI interacts with external actors. Completing this phase provides a clear picture of the SOI's functions and connections. This sets the stage for the next development phase, allowing for a detailed examination of the SOI's internal components and their interactions to ensure the system works effectively.

4.4. Logical analysis

The high-level functions and requirements previously presented can be organized into logical blocks to develop a logical architecture of the SOI. Several design considerations for the system are clear. The system must include a moveable structure to achieve the motions of extension and retraction (ER), boom roll (BR), and wrist roll (WR) for the payload. This structure is composed of beams to support the payload forces, whether from external sources or its own weight. It includes main beams and active beams, with the active beams connected to actuators and the main beams providing support and stability to the payload.

Figure 31 illustrates a potential solution used in a version of the FluxJet, though various configurations can meet the system requirements. Some motions, particularly BR and WR, can be achieved using rotary actuators or axial actuators combined with other mechanisms. The number of actuators and their methods of achieving the different motions can vary. These motions can be executed using individual actuators or multiple actuators capable of combined movements. The placement of these actuators, whether flat or inclined, and the beam lengths can also be adapted, resulting in a logical architecture with numerous potential solutions.

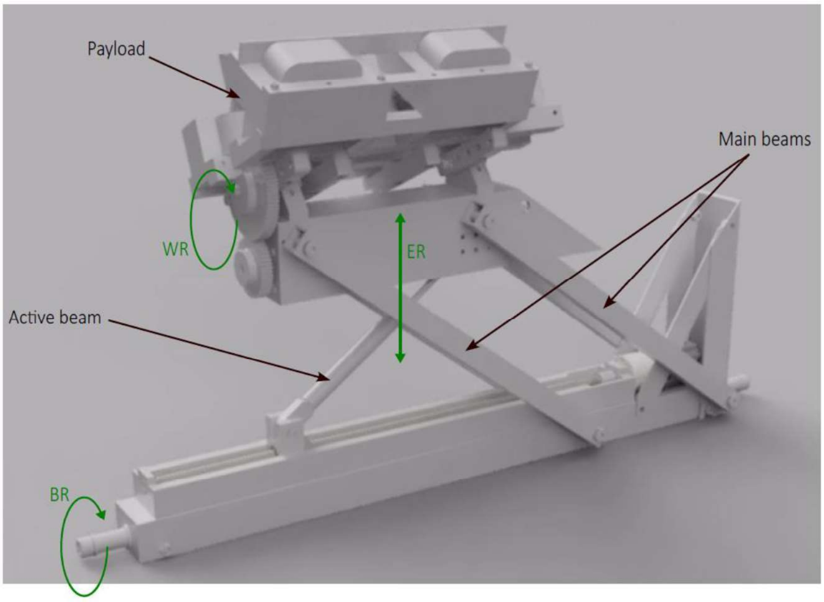


Figure 31: CAD representation of LE.BS of a FluxJet prototype

This thesis considers three main possible logical architectures:

- Logical architecture 1: Independent configuration
- Logical architecture 2: Dependent in-line configuration
- Logical architecture 3: Dependent transverse configuration

The three solutions are explained with further detail in the following Sections 4.4.1, 4.4.2 and 4.4.3.

The variety of viable solutions presents a challenge in the system development. The system designer must choose between them, and this decision cannot be arbitrary. Therefore, for the BS system design, it is crucial to have a tool that aids the designer in making this decision. This project's scope includes developing such a tool to analyse different configurations, perform trade-off analyses, and select the optimal solution from a global perspective. The development of this tool is addressed in Section 5.

4.4.1. Logical architecture 1: Independent configuration

The independent configuration system variant, illustrated in Figure 32, includes several main beams designed to support the payload. The primary main beam is the one connected to the active beam, while the remaining main beams are considered secondary main beams. Although only one additional row of secondary main beams is illustrated in the figure, there could be multiple rows to distribute the forces across more main beams.

The main beams are connected to the vehicle structure and the payload via fulcrums. The active beam is connected at one end to the main beam through a joint and at the other end to an axial actuator. When the axial actuator displaces, it causes a displacement of the active beam, subsequently moving the primary main beam and resulting in the extension and retraction of the payload.

The BR motion of the payload is accomplished using a second actuator, which rotates the entire represented structure. The WR motion of the payload can be achieved using a third actuator, which rotates only the payload. The BR and WR actuators are not illustrated as they are not further examined in this thesis.

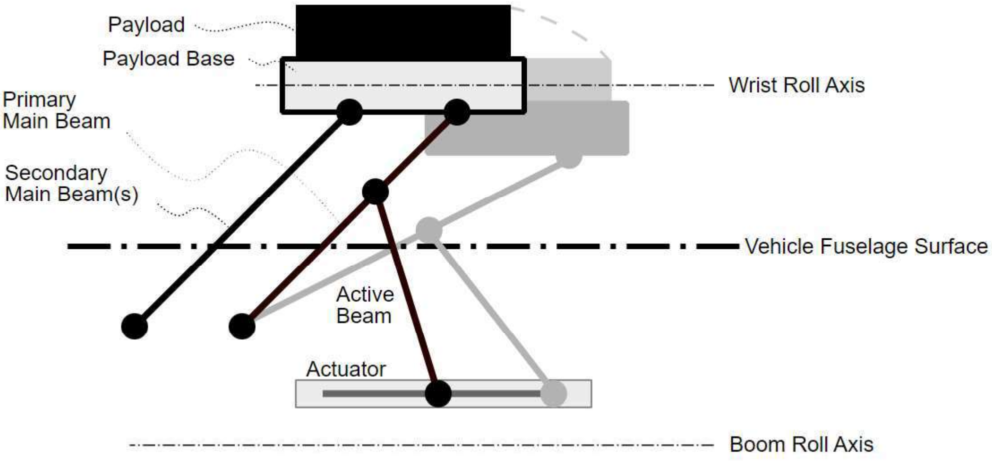


Figure 32: 2D representation of independent configuration of the BS

4.4.2. Logical architecture 2: Dependent in-line configuration

The dependent in-line configuration system variant, illustrated in Figure 33, is analogous to the independent configuration and includes several main beams to support the payload, designated as primary and secondary main beams as in the last configuration.

In this solution, the primary main beam is connected directly to two axial actuators, therefore there are no active beams or in other words the two actuators are integrated into the active beams. The displacement of these actuators allows for both the ER motion and the BR motion of the payload. When both actuators extend or retract simultaneously, the payload is also extended or retracted outwards or inwards into the vehicle. Conversely, when one actuator extends while the other retracts, it induces a lateral displacement, resulting in a boom roll angle. Thus, the system can facilitate the payload movement around the BR axis.

The WR mechanism remains the same as in the independent configuration.

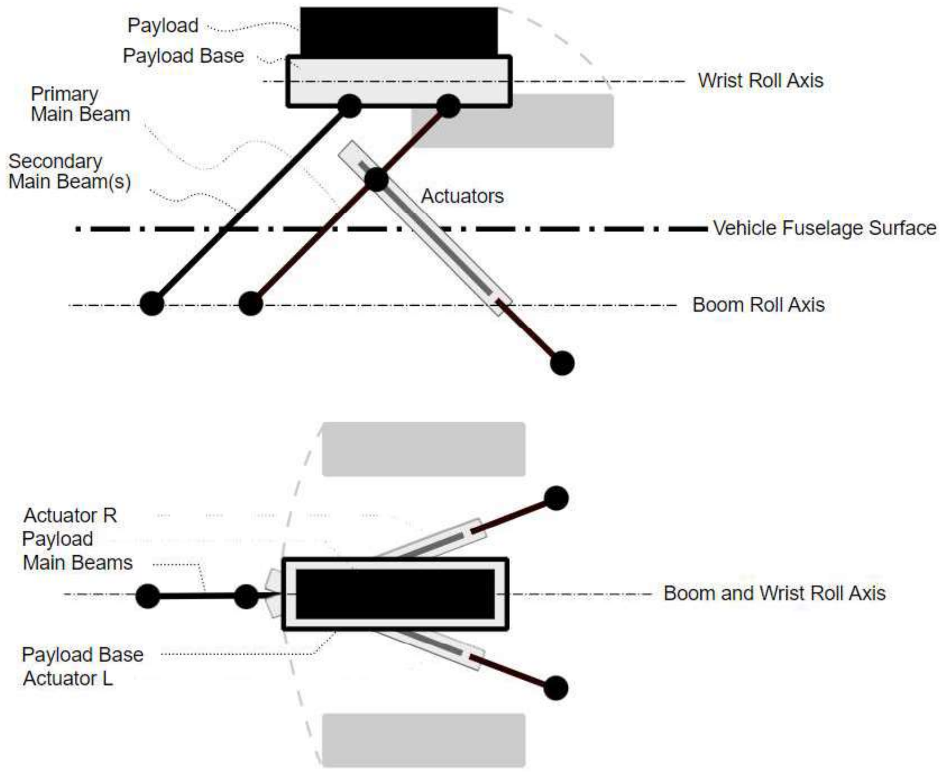


Figure 33: 2D representation of dependent in-line configuration of the BS

4.4.3. Logical architecture 3: Dependent transverse configuration

The dependent transverse configuration system variant, illustrated in Figure 34, is highly like the dependent in-line configuration. In this configuration, there are two active beams connected to the primary main beam on one side and to an axial actuator on the other. The displacement of these actuators enables both ER motion and BR motion of the payload, analogous to the dependent in-line configuration. The actuators are positioned in the vehicle's transverse plane in an inclined orientation. The WR mechanism remains unchanged from the independent configuration.

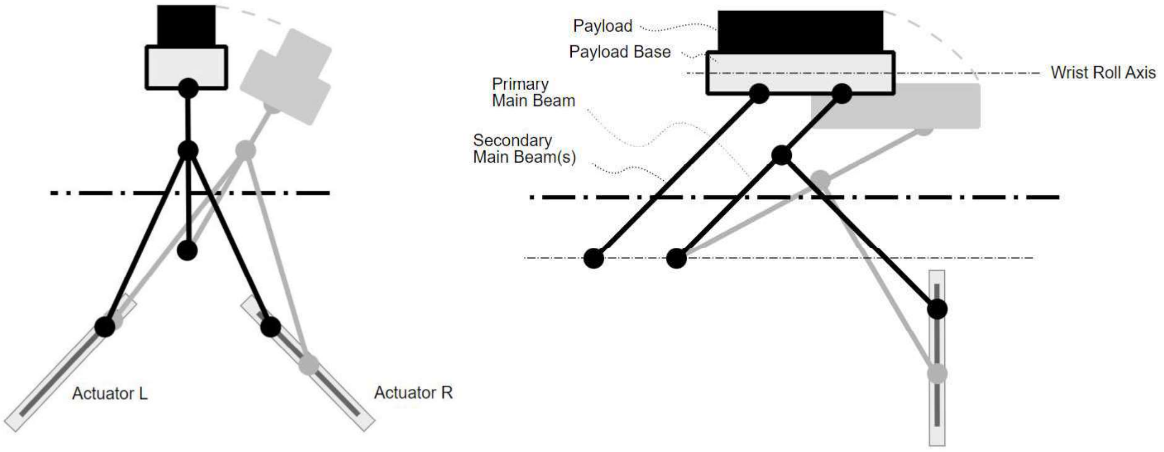


Figure 34: 2D representation of dependent transverse configuration of the BS

4.4.4. Components requirements

While other components must also be specified, this thesis focuses primarily on the actuators. Actuators are critical for the system's operation and are sourced from suppliers. Precise specifications are necessary to ensure the correct products are obtained. The table lists the actuator requirements essential for this specification in line with Section 2.3.3.

Table 4: Actuator requirements list

ID	Description
COMP_REQ_1	The actuator shall have a minimum stroke distance of XXX mm.
COMP_REQ_2	The actuator shall reach a minimum velocity of XXX mm/s.
COMP_REQ_3	The actuator shall achieve a maximum force of XXX mm/s.
COMP_REQ_4	The actuator shall reach a minimum acceleration of XXX mm/s^2.
COMP_REQ_5	The actuator shall have at least a precision of XXX mm.

It is important to note that the specified units may vary if rotary actuators are used; in such cases, millimetres are replaced by degrees.

As with the system requirements, the specific required values are currently unknown ("XXX"). A key objective of this thesis is to assist in calculating these values. The parameter values vary for each vehicle prototype, BS system, system variant, and specific actuator.

Other components must be also specified but for the scope of the thesis, the focus is set on the actuators.

4.4.5. Components breakdown structure

The components breakdown structure for the FluxJet system is presented in Figure 35. This diagram provides a detailed hierarchical representation of the main components and subsystems involved in the FluxJet and the BS operation.

The BS system includes various actuators, a yielding mechanism, the dynamic structure, and sensors that are essential for deploying and controlling the payload. The detailed breakdown of the BS system components highlights their specific functions and integration points within the overall FluxJet system. This serves as a foundation for mapping the distinct functions the system must achieve to the corresponding components responsible for executing them.

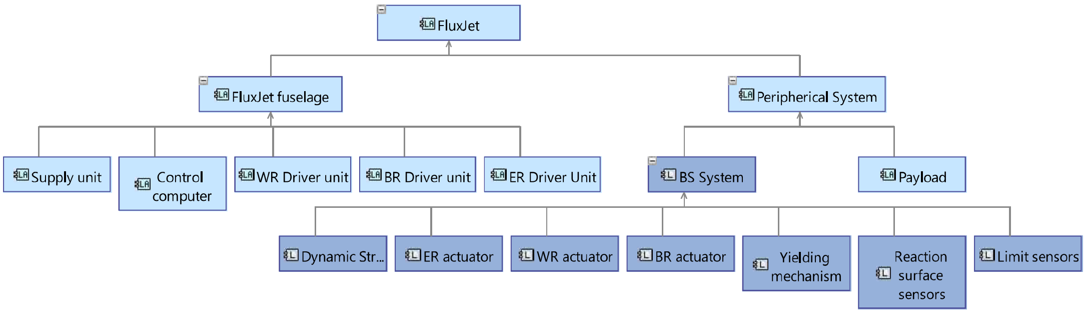


Figure 35: Components breakdown structure diagram

This diagram varies slightly for the three different logical architecture solutions. In the dependent configurations, there are no separate ER and BR actuators. Instead, two coupled actuators execute both motions. In this solution, these actuators are defined as ER-BR actuators.

4.4.6. Logical architecture

Upon completing the logical analysis, a logical architecture of the system is developed to compactly display all functions and interactions of the SOI. The complete logical architecture diagram is available in Appendix 2. This section focuses specifically on the SOI, as the differences in the external components outside of the BS are minimal compared to the previously seen system architecture diagram.

The main development within the model occurs in the SOI itself during the logical analysis, transitioning the perspective of the SOI from a black box to a white box approach. As in the system architecture, this diagram is created for the most complex version of the BS system, which has a linear motor as payload.

Figure 36 illustrates the logical architecture of the SOI, demonstrating that three actuators – ER, BR, and WR – control the motions of the payload by converting energy into displacement. This energy can be electrical, pneumatic, or hydraulic, depending on the actuator type. The proprioceptive sensory function, defined during the system analysis, is achieved by the actuators, which send information to the control computer to calculate the target movements of the payload. Additionally, the actuators directly inform the driver of their status, including health and errors.

Another critical component of the SOI is the movable structure composed of beams and joints. Although no further detail is provided for this component, its inclusion is vital as it manages the forces from the payload and fuselage. The interaction between these forces enables the movement of the FluxJet in longitudinal, vertical, and lateral directions, as well as roll rotation. The vehicle's position affects the BS system, influencing the overall system's position.

To achieve the yielding function, a yielding mechanism is necessary. An example is a gas spring aligned with the active beam, allowing the payload to retract under excessive forces.

Additionally, limit position sensors must be installed to prevent specific system or payload positions, and sensors to measure the distance to the reaction surface. Both types of information, along with proprioception data, are used to calculate the target movements of the payload.

Different logical architectures are possible depending on the system configurations seen in Sections 4.4.1, 4.4.2 and 4.4.3. However, this diagram is applicable to the three configurations, with minor differences for the ER and BR actuators in the dependent configurations. In this case, they are referred to as ER-BR actuators, where the ER and BR motion functions depend on both actuators.

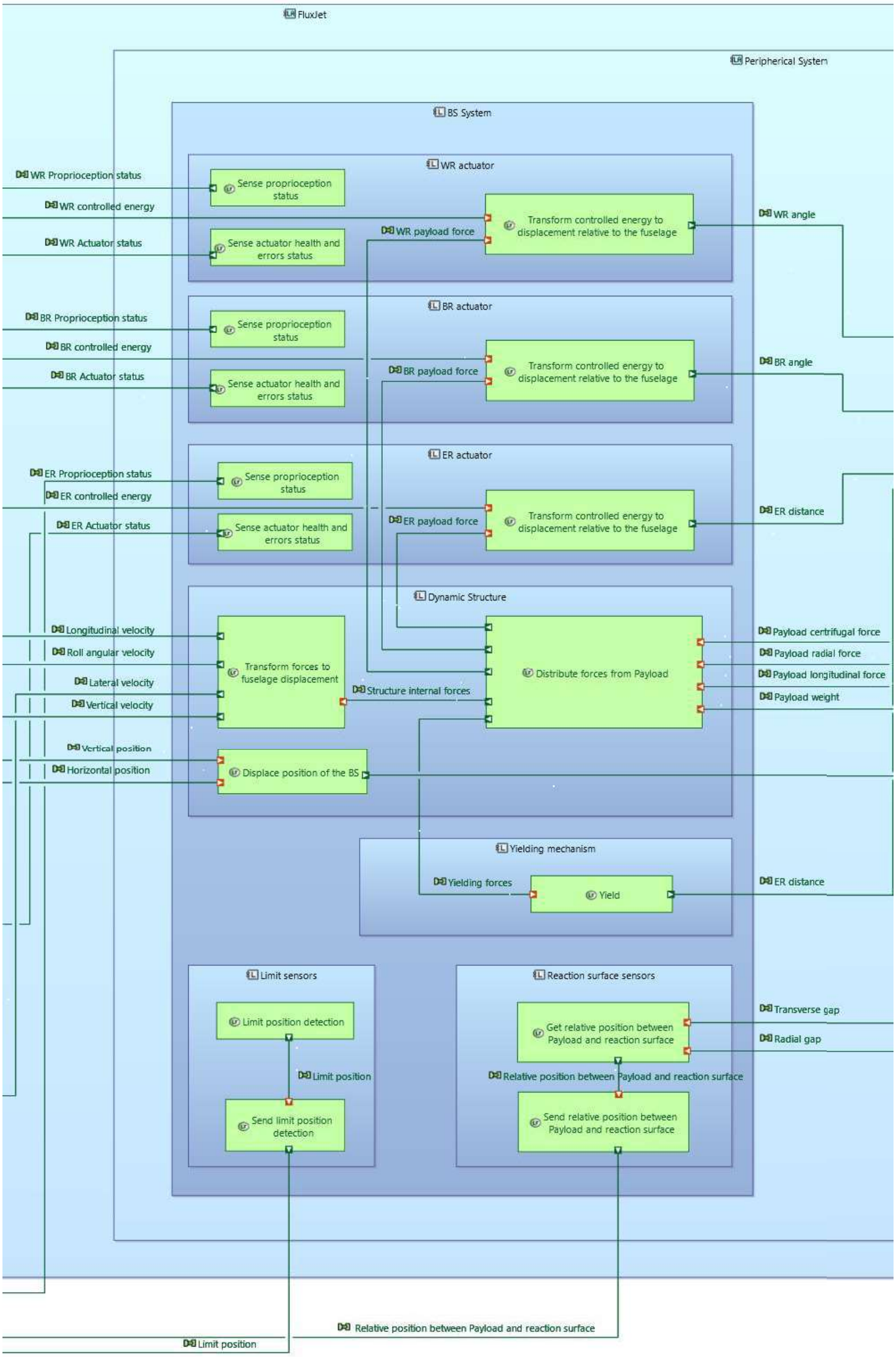


Figure 36: Logical architecture diagram – detail of SOI (BS)

4.5. Physical analysis

The physical analysis (PA) consists in converting the logical architecture into physical architecture which is composed of physical components of the system and their interfaces. However, in this thesis, although the logical architecture is analysed, a specific logical architecture is not finalized.

In this thesis an algorithm that evaluates and analyses various logical solutions is developed, facilitating a universal design process applicable to different variants of the SOI, dependent on specific requirements and constraints. This approach inherently leads to numerous potential physical solutions based on the specific logical architecture. It is critical to note that the production of specific physical solutions is not the aim; rather, the groundwork is being laid for a methodology that can adapt to various parameters and conditions.

The establishment of a logical solution typically follows the selection of physical components; however, this stage is intentionally excluded in this thesis. The exclusion is based on several key considerations. At this point in the research, there is no immediate need to specify physical components, as the goals are centred around conceptual development and simulation. This focus allows for the maintenance of clarity in crafting a versatile algorithm without delving into the complexities of physical implementation, which would significantly broaden the scope of the project.

The diversity and complexity of potential physical solutions further justify this approach. For example, the variety in actuators – ranging from electric, pneumatic, to hydraulic systems – indicates that each set of requirements could lead to different physical implementations. Addressing the multitude of possible physical configurations would complicate the thesis beyond its manageable scope.

It is important to clarify that the exclusion of PA from the thesis does not imply that such analysis is unnecessary in practice. Once a logical structure is finalized, the selection of specific physical components becomes crucial. Diverse types of actuators and other components can fulfil the logical functions identified through the logical analysis, and PA plays a vital role in evaluating these potential solutions.

This strategic decision aligns with the objectives of the thesis, ensuring that the research progresses within a well-defined framework, paving the way for future studies where detailed PA may be integrated once the algorithmic groundwork has been established.

5. Logical architecture analysis tool

As noted in the previous section, there are multiple possibilities for developing the system's logical architecture. This project focuses on three configurations discussed earlier. These configurations are considered the most likely to be used in various versions of the vehicle. However, many other options could have been explored.

To analyse different logical architectures, a tool is essential for the system's design. This tool must enable the design team to rapidly evaluate different configurations, thereby reducing the development time of the BS system. This is crucial as there are, for the moment, at least five distinct types of payloads and six different versions of the vehicle to develop, ranging from small demonstrator vehicles to full-scale models.

The objective is to create a tool that helps design the BS system with arbitrary parameters based on each payload requirements. There are many design parameters that influence the system, and the design team can adjust these to achieve an efficient design that meets all requirements. Given the complexity of the system, there might not be a single optimal solution. Therefore, the tool must facilitate trade-off analysis between various parameters.

The algorithm's general concept is to calculate the system's behaviour for a defined list of design parameters. These parameters can also be analysed over a range of values, observing how changes in these parameters affect the system's behaviour.

The tool must calculate the forces, range of movement, and velocity of the actuators, which are critical parameters for specifying the required actuator, as discussed in 4.4.4. Given the importance of the ER and BR motions, which are the most demanding and impactful on the BS system development, this thesis focuses primarily on these motions and its actuators. The WR motion is addressed in later stages of the design process.

MATLAB is used to perform the system calculations. The output includes the different forces involved in the system, aiding in the selection of the best configuration among those analysed. Given the considerable number of parameters, the calculations must be very rapid. Therefore, simulations using MATLAB add-ons such as Simulink or Multibody are not feasible. The goal is to develop a quick algorithmic code capable of calculating each configuration in tenths of a second or less, allowing for the evaluation of hundreds or thousands of configurations with different parameters in a reasonable time.

5.1. Prime and design parameters for the different BS configurations

First, a classification of the parameters involved in the algorithm is conducted:

- Inputs: There are two types of inputs:
    - Prime Parametric Requirements: These requirements are provided to the BS design team as fixed values that are fundamental for the design of the BS system. These parameters are independent of the BS system design but are dependent on the type of payload and prototype.
    - Design Parameters: These are the variables that a BS designer can adjust to create a design that meets the requirements. A large variety of these parameters can be modified to test the efficacy of different designs. Although as much parallelism as

possible is maintained between the different system configurations, these parameters are specific to each configuration.

- Time-Varying Variables: These are variables that change over time as the vehicle operates, such as forces and actuator movements. They represent operational conditions.
- Outputs: These results are calculated by the tool to aid in design decisions. The outputs vary with each configuration and operational condition and differ slightly for each logical architecture.

The following Table 5 detail the prime parametric requirements:

Table 5: List of prime parametric requirements

Notation	Parameter	Unit
$BS_{\text{angular position}}$	Angular position of the BS system[3]	°
PL_{mass}	Payload mass	kg
$FJ_{\text{fuselage radius}}$	FluxJet fuselage radius	mm
$FJ_{\text{max translational skew}}$	FluxJet maximal translational skew[3]	mm
$FJ_{\text{max roll angle}}$	FluxJet maximal roll angle[3]	°
$RS_{\text{radial distance}}$	Radial distance of the reaction surface	mm

Figure 37 shows the independent configuration including design parameters and time-varying variables:

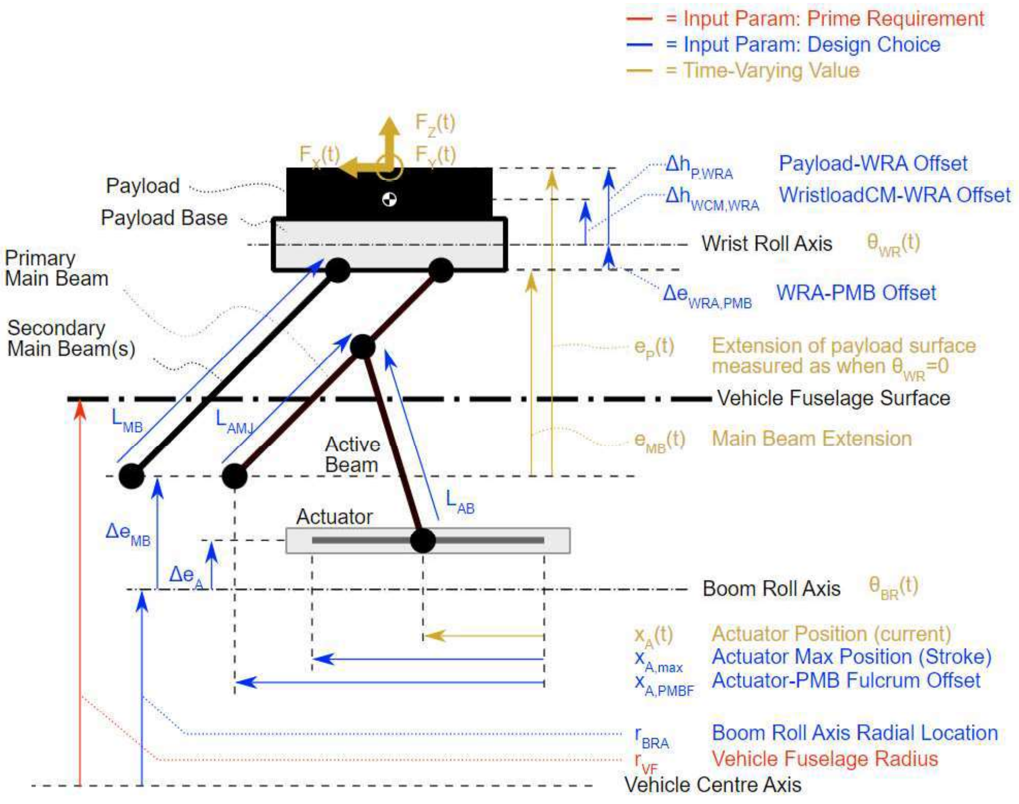


Figure 37: Geometry fundamentals and variables for the independent configuration

---

[3] See Figure 41: Defined positions along the ER motion

Figure 38 shows the dependent in-line configuration including design parameters and time-varying variables:

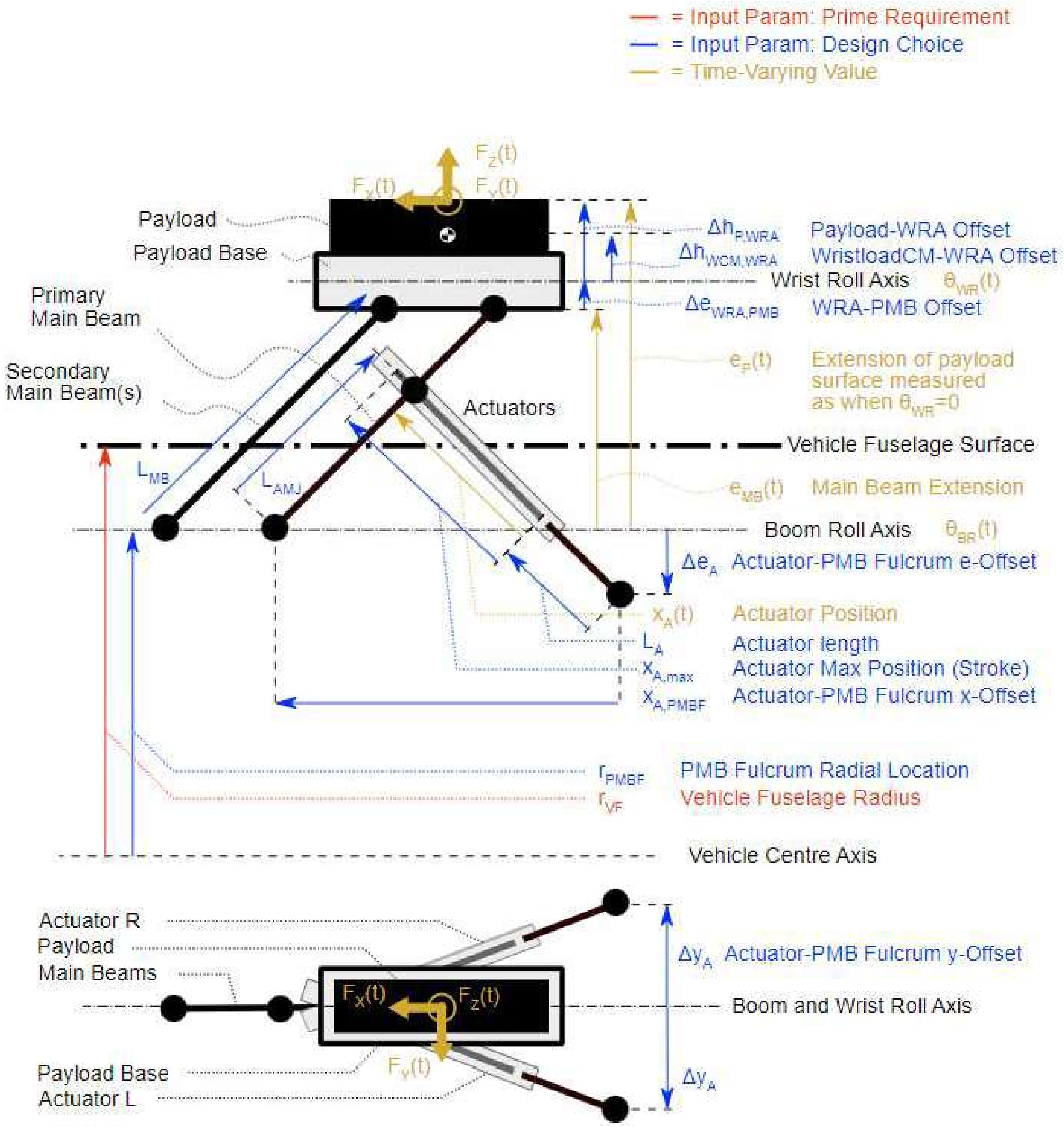


Figure 38: Geometry fundamentals and variables for the dependent in-line configuration

Figure 39 shows the dependent transverse configuration including design parameters and time-varying variables:

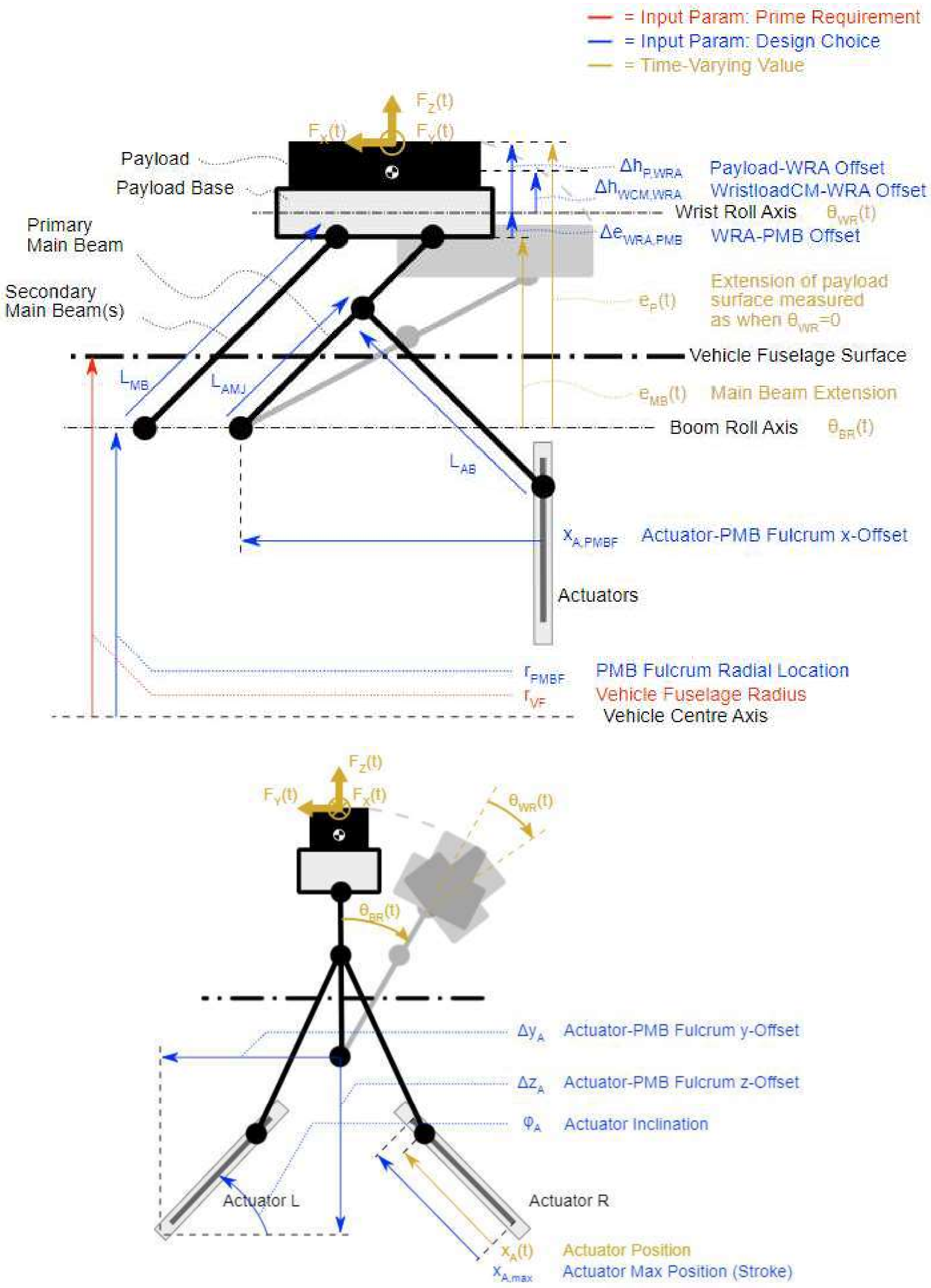


Figure 39: Geometry fundamentals and variables for the dependent transverse configuration

5.2. The algorithm

This section explains the construction of the algorithm, illustrated in Figure 40, and the calculations involved. The code is divided into sections, starting with the input configuration. The system designer must first configure the algorithmic tool, selecting the BS system configuration (independent, dependent in-line, or dependent transverse). The designer can choose to either calculate a single design, which provides detailed insights into the system's behaviour (individual analysis mode), including plots of displacements, actuator forces, and structural forces, or analyse the system's behaviour under the variation of a design parameter (parameter analysis mode). This parameter is set by specifying its minimum and maximum values and the number of evenly spaced points within this range.

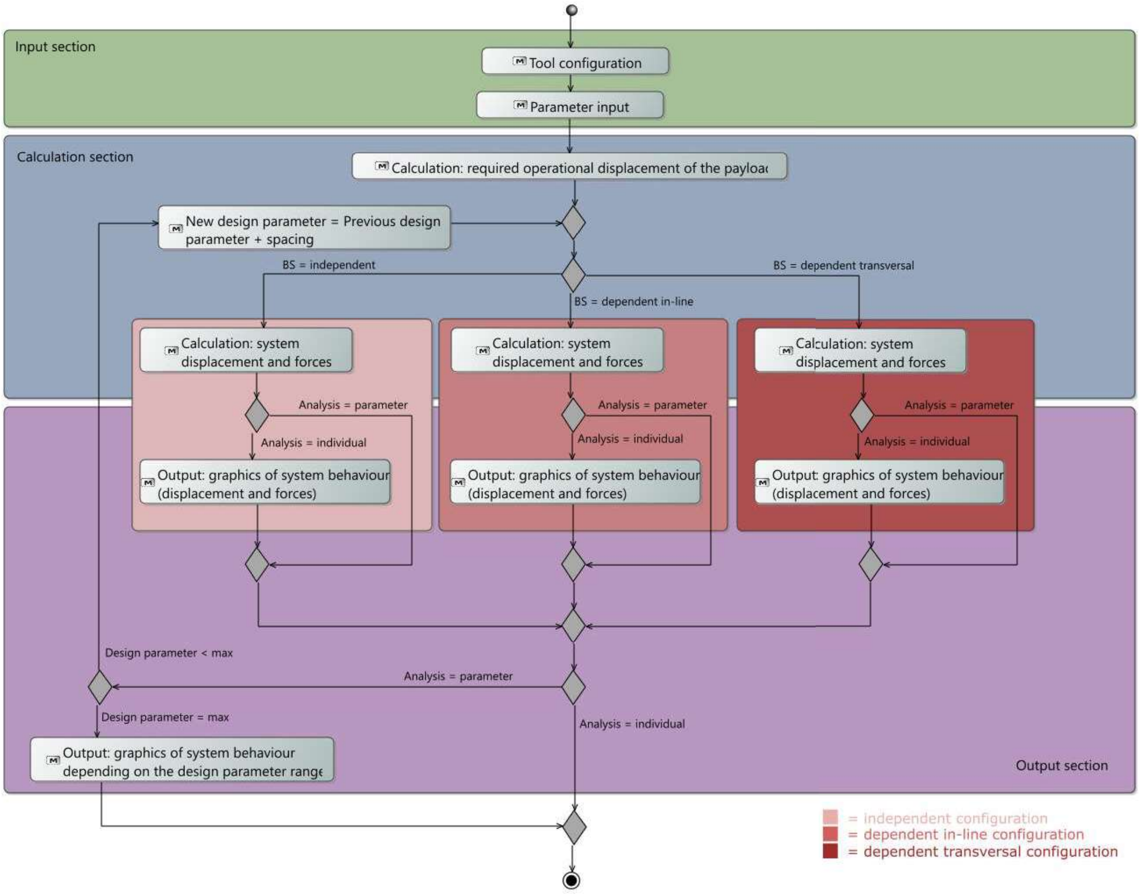


Figure 40: Flow diagram of the MATLAB algorithm used for the logical architecture analysis tool

The algorithm uses a discrete evaluation approach, calculating functions at specific points to control the resolution and accuracy of the results precisely. The computation resolution, defined by the number of points, can be set by the designer. Once the tool is configured, the primary parametric requirements and design parameters are entered.

The algorithm then runs either once, if configured for the individual analysis mode, or in a loop, if configured for the parameter analysis mode. In the latter case, the code iterates over the defined range of values for the selected parameter, increasing each iteration the value of the parameter from the specified minimum value to maximum value. The following Table 6 displays the code's speed for different analyses and configurations given various accuracy resolutions.

Table 6: Algorithm performance for different system configurations, analysis approaches and resolutions

Computation time (s)	**Individual analysis (Figures plotting included)**			**Parameter analysis (20 iterations)**		
Resolution (Accuracy)	50 (Low)	100 (Medium)	200 (High)	50 (Low)	100 (Medium)	200 (High)
Independent configuration	0.55-0.60	0.60-0.65	0.65-0.70	0.10-0.11	0.11-0.12	0.12-0.14
Dependent in-line configuration	0.45-0.50	0.50-0.55	0.70-0.75	0.53-0.55	1.80-1.90	6.80-7.00
Dependent transverse configuration	0.50-0.55	0.55-0.60	0.85-0.90	0.38-0.40	1.27-1.30	4.85-5.00

The ranges of the ER, BR, and WR motions are calculated for the worst-case scenario to determine if the system can achieve the required operational positions of the payload. This calculation also helps visualize the full operational movement achievable by the actuators and the actual range of motion needed. This approach is applied to the displacements performed by parameterized actuators already defined in the system design. For WR (and BR in the independent configuration), which play a secondary role in the design, calculations are performed for the required range of movement rather than the entire range of movement of an actuator.

The system design calculations involve determining the actuator forces needed to displace the payload and the mechanical structure forces supported by the beams for each operational position. The velocity of the actuators is not calculated directly but indirectly by determining the ratio between the displacement of the payload and the displacement of the actuator. This results in a relationship between the payload movement and the actuator movement. Knowing the required speed of the payload over a certain period allows for the determination of the actuator's speed over the same period, resulting in the velocity that the actuator must achieve.

The following sections explain these calculations in detail. The calculations for each logical architecture are maintained as similar as possible.

5.2.1. Calculation of the required operational displacement of the payload

The fundamental ranges of motion required for the Boom Suspension (BS) are calculated to determine the minimum and maximum ER, BR and WR motion values necessary for the BS to move the payload and meet the operational requirements. These calculations ensure that the payload can be positioned correctly in all relative positions between the vehicle and the infrastructure, which are defined by the prime parametric requirements. The required motions do not depend on the system configuration. The maximum and minimum ER, BR, and WR values are internal variables of the algorithm, helping to verify whether the chosen configuration satisfies the requirements.

Extension-Retraction motion

The payload must be moved from the retracted position to operational positions where it interacts with the FluxWay as illustrated in Figure 41. The guideway has a reaction surface, and through this interaction, depending on the payload, magnetic levitation and propulsion are achieved, or energy is acquired.

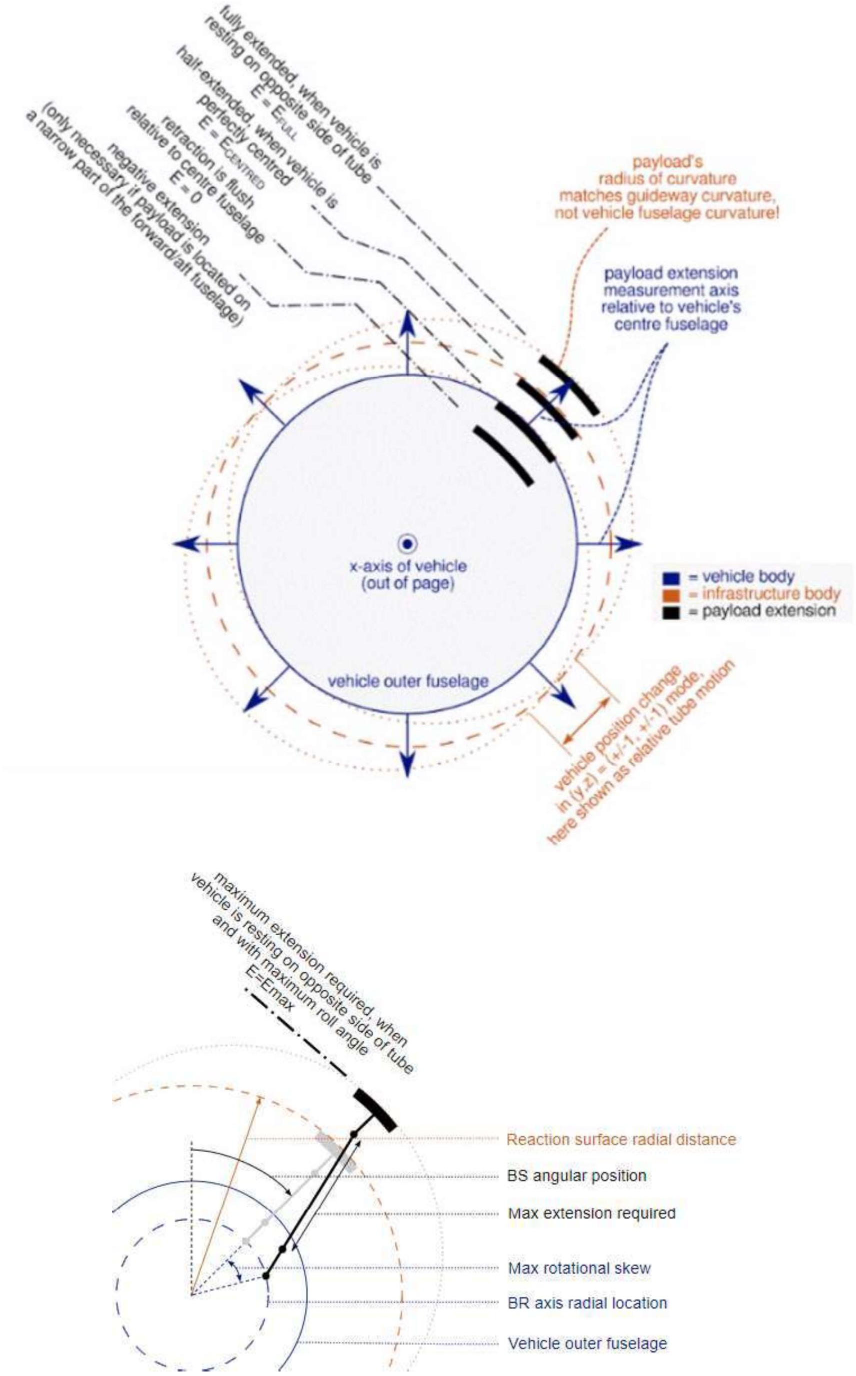


Figure 41: Defined positions along the ER motion

There are several defined relevant positions for this motion:

- Minimum Extension Required: The payload is in the retracted position when not in use, protected from collisions outside the vehicle. Due to the variable diameter of the fuselage along the longitudinal position, this extension might be less than the extension aligned with the fuselage. The minimum extension may be the same as the fuselage extension at the centre of the vehicle.

- Fuselage Extension: The payload is aligned with the fuselage diameter at the centre of the vehicle in its retracted position.
- Centred Extension: The payload is in the operational position, interacting with the reaction surface of the FluxWay. The FluxJet is centred in the tube in this position.
- Fully Extended: The payload is in the operational position, but the FluxJet is not centred in the tube. The fully extended position occurs when the FluxJet deviates both horizontally and vertically from the centre of the tube, with maximum translational displacement in the YZ-plane in the opposite direction to the reaction surface.
- Maximum Extension Required: Like the fully extended position, but with the vehicle adopting the maximum roll angle. This is the most unfavourable case and represents the maximum extension the BS system must achieve for the payload to reach the reaction surface at every vehicle position relative to the infrastructure.

Boom Roll motion

Due to the roll rotation of the FluxJet, the payload must rotate along a longitudinal axis to reach the reaction surface at every vehicle position. A maximum and minimum BR angles are calculated. It is assumed that the system and the reaction surface are aligned in the centred position. The extrema occurs when the FluxJet displaces with maximal translational skew in the direction of the reaction surface and the vehicle is rotated at the maximum roll angle as shown in Figure 42. Because of the symmetry of the vehicle, system, and infrastructure, the minimum and maximum BR angles have the same absolute values.

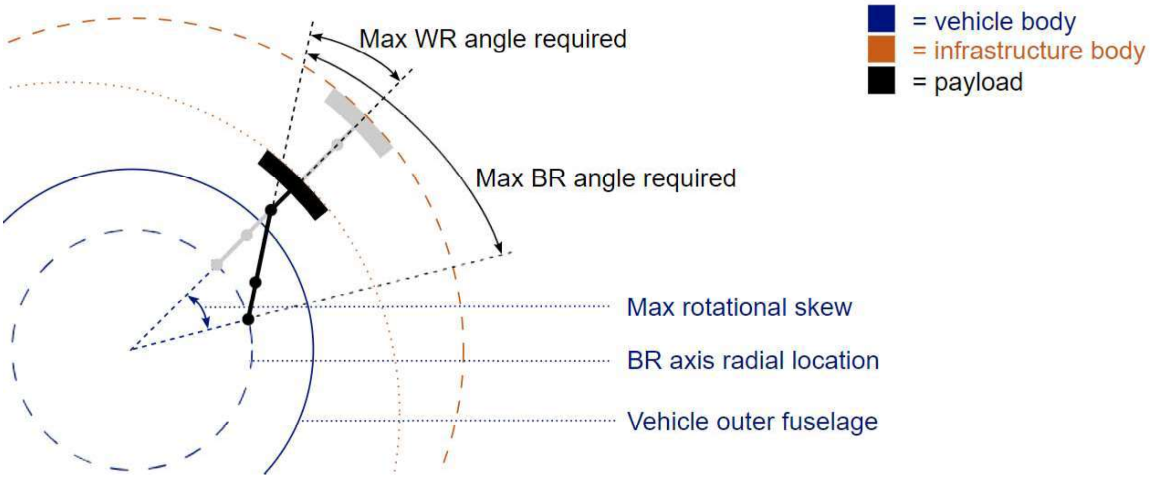


Figure 42: Maximum BR and WR angle calculation

Wrist Roll motion

The WR is required to realign the payload with the reaction surface when the system experiences a BR displacement. The worst-case scenario for calculating the maximum and minimum values for the WR is the same as for the BR, as shown in Figure 42. However, this does not mean the values are identical due to the distinct positions of the rotation axes. Again, because of the system's symmetry, the minimum and maximum WR angles have the same absolute values.

5.2.2. Calculations for the independent configuration

The system of equations for this system configuration addresses a two-dimensional problem. The algorithm begins by calculating the extension of the payload based on the displacement of the ER actuator. This calculation is performed geometrically by determining the intersection points of two circles with centres at the primary main beam fulcrum and the actuator's end as shown in Figure 43. The radii are the distances between the primary main beam fulcrum and the joint (L_{AMJ}), and the active beam length (L_{AB}), respectively. The intersection with the highest Z-coordinate is selected for each position as the correct one. This intersection point determines the joint's position, which is then used to calculate the payload position. This calculation is repeated for the entire range of the actuator's stroke.

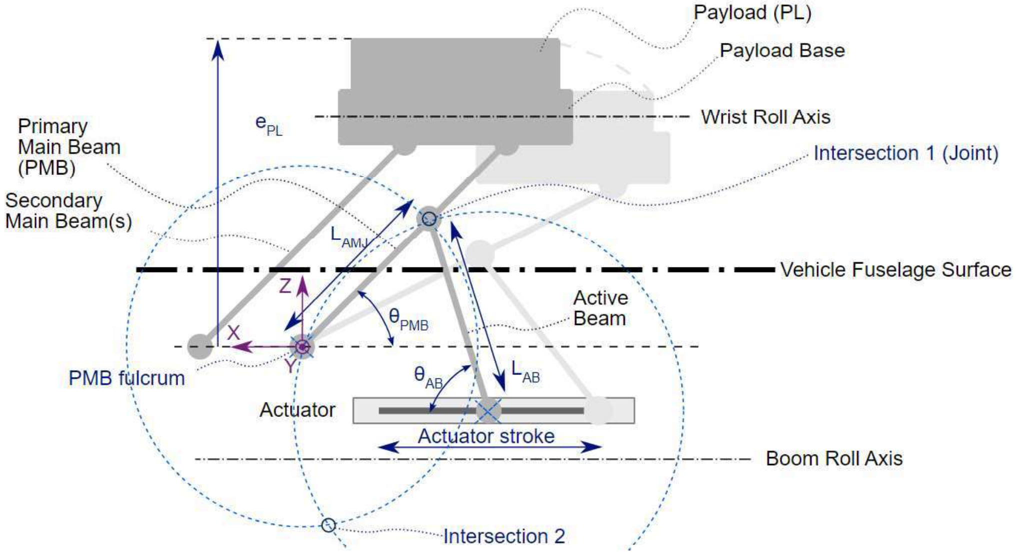


Figure 43: Calculation of joint and payload position in the independent configuration

Once the joint's position is known, the beams' inclinations (θ_{PMB} and θ_{AB}) can be easily calculated, leading to the determination of the payload extension (e_{PL}). A specific actuator position, illustrated in Figure 44, must be avoided when singularity occurs, which happens if the payload is extended too much, making the active beam perpendicular to the stroke. Exceeding this point compromises system stability in this configuration. Therefore, the BS design must prevent this singularity within this system configuration.

Subsequently, the forces that the system must passively endure and actively exert are calculated using trigonometry, based on the ER displacement. All the following mentioned forces are illustrated in Figure 45. Two primary forces are considered: the external forces acting on the payload (F_X, F_Y and F_Z) due to its function and the payload own weight (W_{PL}). The external forces move with the payload and therefore their vectors are independent of the WR angle, BR angle, and ER position. In contrast, the direction of the payload weight vector depends on the BR and WR angles assumed by the system. Therefore, the forces experienced by the system vary with the BR and WR angles and depend on the axial actuator displacement.

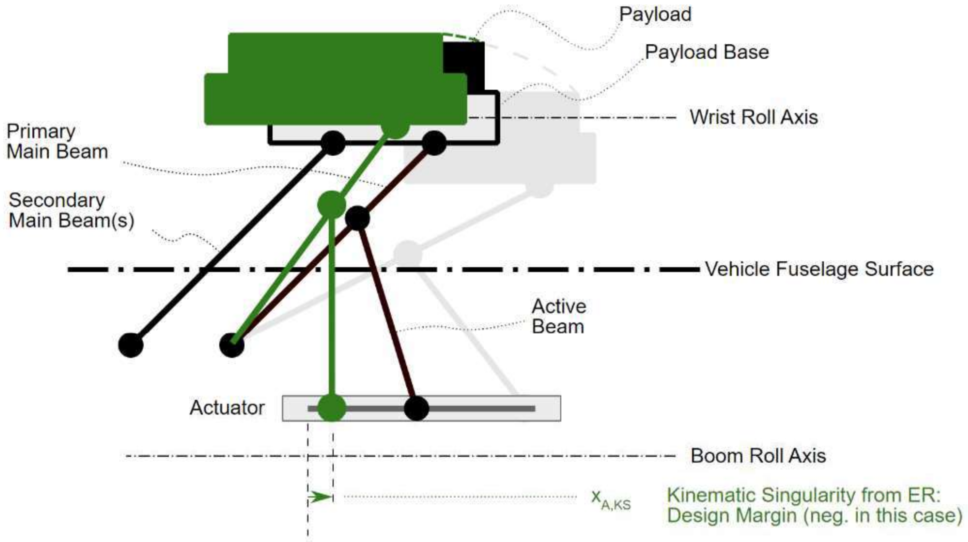


Figure 44: Kinematic singularity for the ER actuator in the independent configuration

The summation of these forces acting on the payload is transferred to the primary main beam, considering the number of main beams in the system configuration. The primary main beam experiences both axial (compression or tension) and transverse forces. The transverse force must be counterbalanced through the active beam at the joint. The active beam itself only handles axial forces. Once the axial force on the active beam is known, the actuator's axial force can be calculated. This force is assumed to be the only force the actuator can achieve. The remaining transverse force is entirely supported by the mechanism holding the actuator, which is also calculated.

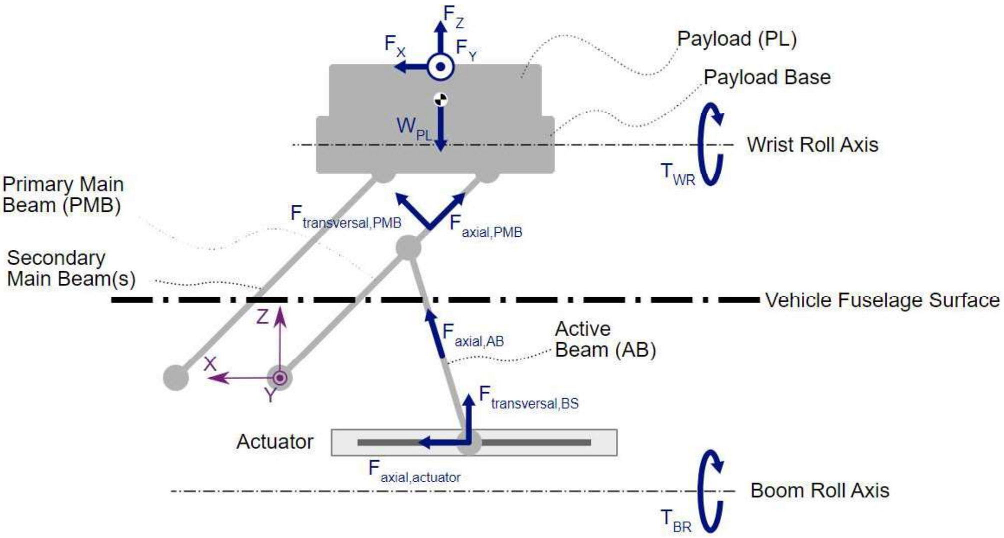


Figure 45: System forces acting in the independent configuration

Next, the forces on the BR are calculated. The external force in the X-direction (F_X) is parallel to the BR axis and thus does not influence the calculation. The critical factor in calculating the BR torque is the distances between the BR axis and the external forces F_Y and F_Z and the payload weight W_{PL} throughout the operational range. These distances depend on the payload extension and the WR angle and can be trigonometrically calculated. The BR torque (T_{BR}) is then calculated based on these variables.

Finally, the WR torque (T_{WR}) is calculated. The external force in the X-direction (F_X) is parallel to the WR axis and does not influence the calculation. Similarly, the external force in the Z-direction (F_Z) is in the same plane as the WR axis for every operational position and thus does not influence the calculation. Only the external force in the y-direction (F_Y) and the payload weight (W_{PL}) affect the calculation. The WR torque depends on the WR and BR angles, as the distance between the WR axis and the payload weight varies with these angles.

5.2.3. Calculations for the dependent in-line configuration

The calculation for the dependent in-line configuration is analogous to the previous calculation, with the difference that the systems of equations address three dimensions. The algorithm begins by determining the ER position and BR angle of the payload based on the displacement of the actuators. This calculation is performed geometrically by identifying the intersection points of three spheres and is illustrated in Figure 46. The trilateration methodology (Asmaa, Hatim, and Abdelaaziz 2014, 415-419), commonly used in GPS location due to its computational efficiency, is applied here.

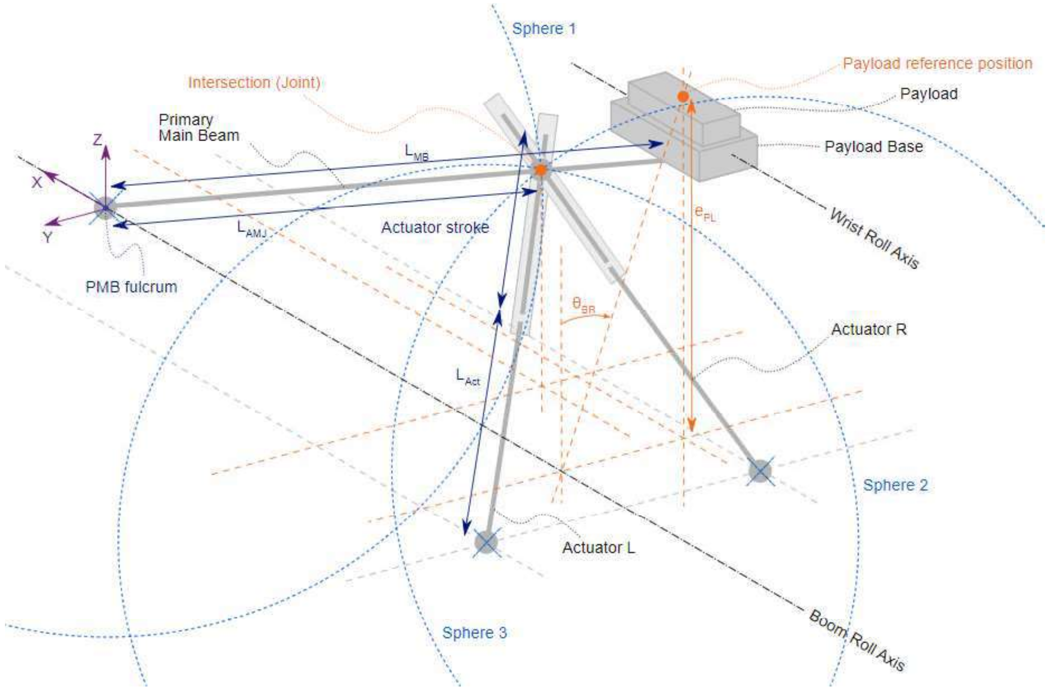


Figure 46: Calculation of payload position in the dependent in-line configuration

The first sphere is centred at the primary main beam (PMB) fulcrum, with a radius equal to the distance between the primary main beam fulcrum and the joint (L_{AMJ}). The second and third spheres are centred at the left and right actuator fulcrums, each with a radius equal to the length of the actuator (L_{Act}) plus the actuator stroke. Of the two solutions, the intersection with the highest z-coordinate is consistently selected as the correct one. This intersection point determines the joint's position, which is then used to calculate the payload position. Since the lengths of the actuator stroke can be actively adjusted, this calculation is iterated for all combinations of the entire range of both actuators. The joint's position, and therefore the payload position, are thus dependent on the left and right actuator displacements. Once all operational positions of the joint are determined, they can be translated to the corresponding ER positions (e_{PL}) and BR angles (θ_{BR}) for each coupled actuator displacement.

Following this, the forces that the system must passively endure and actively exert are calculated for each operational position. For simplification, the WR angle is set to zero unless otherwise specified. Analogous to the calculation for the independent configuration, the external forces (F_X, F_Y and F_Z) move with the payload and therefore their vectors are independent of the WR angle, BR angle, and ER position. On the other hand, the payload weight (W_{PL}) depends on the BR angle (and the WR angle in case it would not be maintained as 0° for simplification).

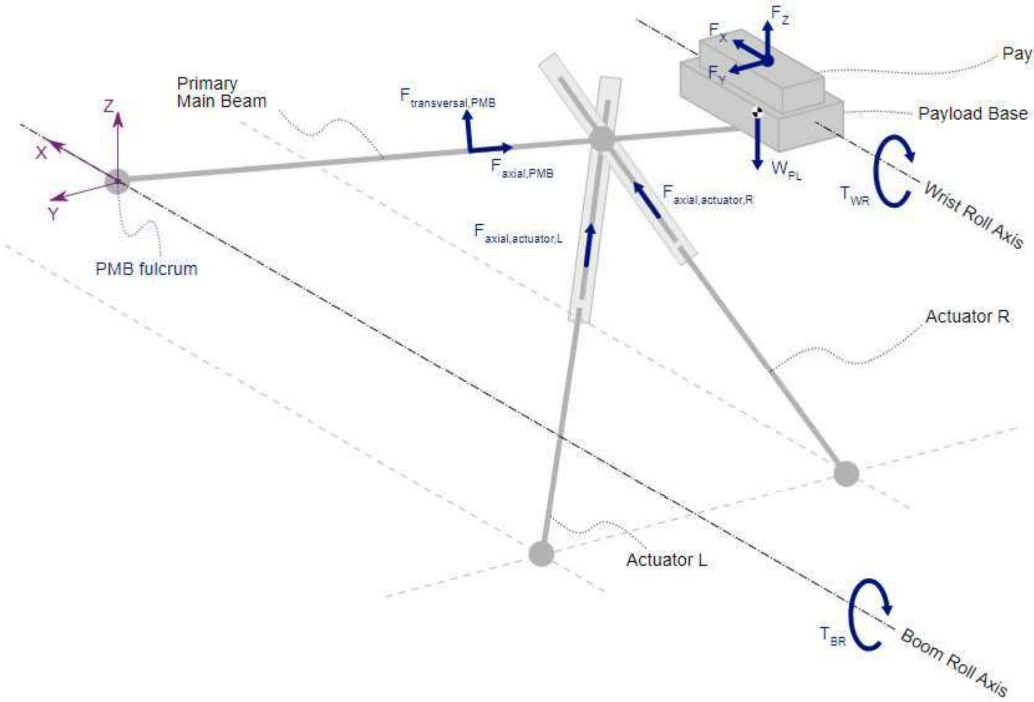


Figure 47: System forces acting in the dependent in-line configuration

The forces acting on the payload are transferred to the primary main beam. The primary main beam experiences both axial (compression or tension) and transverse forces. The transverse force of the primary main beam must be counterbalanced through the left and right actuators at the joint, which only handle axial forces.

Finally, the calculation of the WR torque is identical to that for the independent configuration.

5.2.4. Calculations for the dependent transverse configuration

The calculation for the dependent transverse configuration is like the previous calculation. The ER position and BR angle of the payload, based on the displacement of the actuators, are again determined using trilateration as illustrated in Figure 48.

The first sphere is centred at the primary main beam fulcrum (PMB) with a radius equal to the distance between the primary main beam fulcrum and the joint (L_{AMJ}). The second and third spheres are centred at the ends of the left and right actuators, respectively, each with a radius corresponding to the active beam length (L_{AB}). The calculation is iteratively performed the entire range of the left and right actuator strokes, which alters the positions of the actuator ends. The results for this configuration also depend on two variables: the left and right actuator displacements. The solution with the highest z-coordinate is always considered correct. Once the joint position is determined, the payload position can be computed for all coupled actuator positions. These positions are then translated to the corresponding ER positions (e_{PL}) and BR angles (θ_{BR}).

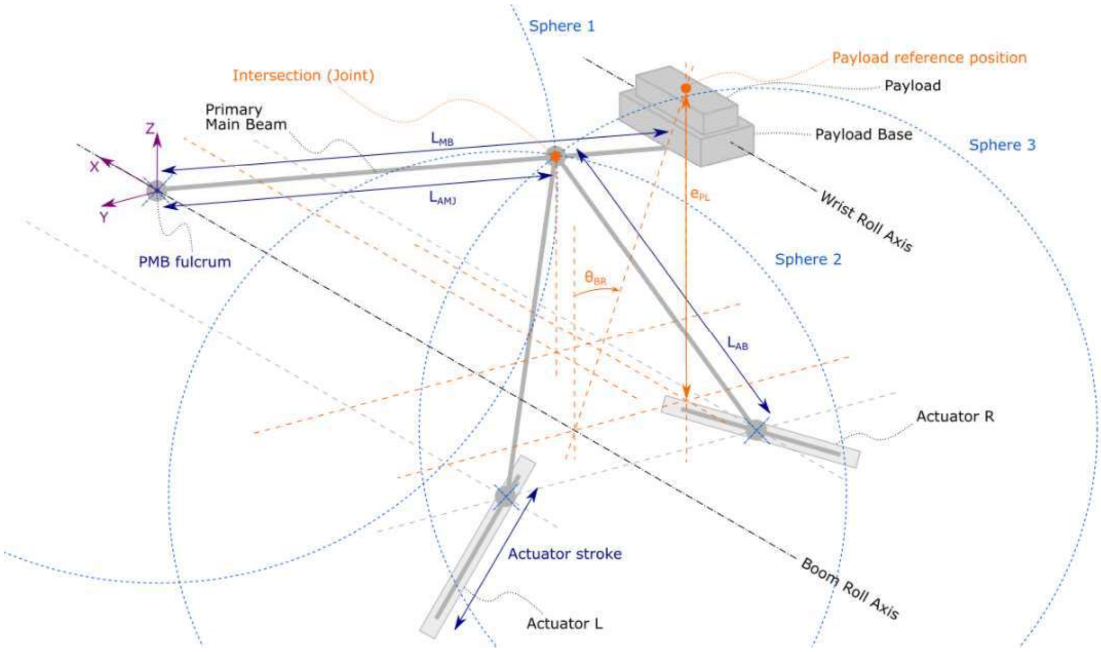


Figure 48: Calculation of payload position in the dependent transverse configuration

Subsequently, the forces of the system actuators and beams shown in Figure 49 are calculated for each operational position, following the same procedure as in the previous configuration. The only distinction is that in this case, the actuator forces are now axial forces of the active beams, which are supported by the actuators at the active beam fulcrum. The WR torque calculation remains unchanged from the previous configurations.

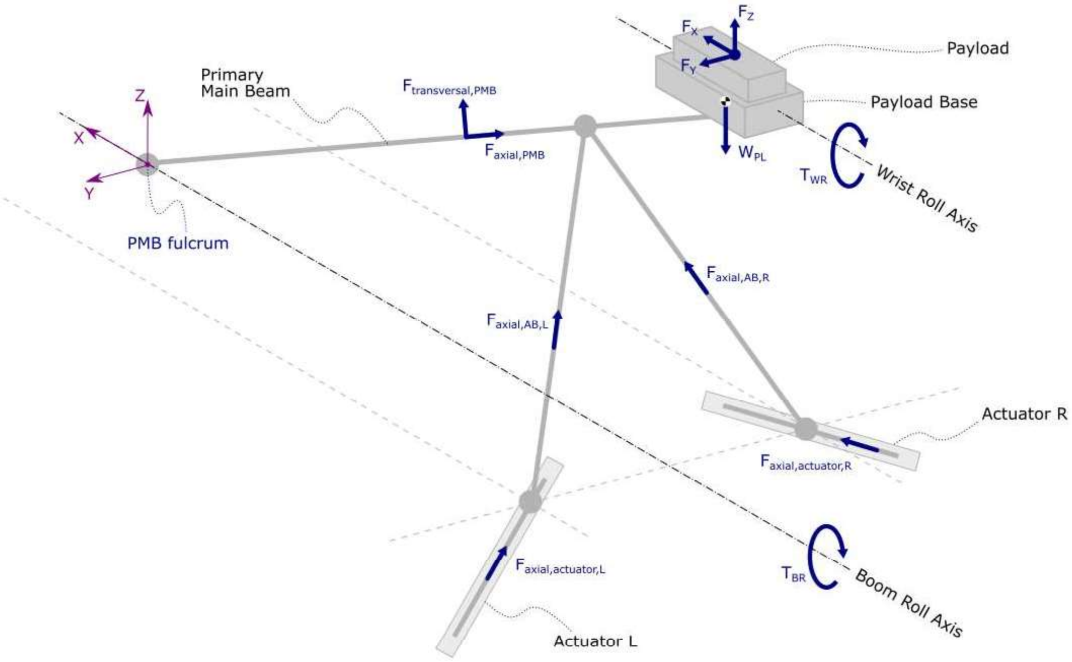


Figure 49: System forces acting in the dependent transverse configuration

5.3. Results and demonstration

This chapter focuses on the output section of the algorithm, presenting graphical representations of the system's behaviour. The calculations include displacement analysis, internal forces within the beams, and the forces required from the actuators. To narrow the scope of the thesis, only the results related to displacement and actuator forces are discussed. These results are used to determine the three main requirements for the actuators: stroke (COMP_REQ_1), velocity (COMP_REQ_2), and force (COMP_REQ_3) as outlined in Section 4.4.4. This information aids in specifying the actuators required for the BS system.

5.3.1. Outputs of the independent configuration

For the next results, input parameters were based on those used for the linear engines of a smaller-scale demonstrator already built by TransPod. This approach allows for the validation of the results by comparing them with the calculations performed by the team responsible for designing this BS system and with the actual system itself.

The first results presented are for displacement. Figure 50 illustrates the payload extension as a function of the actuator displacement, the payload extends when the actuator stroke increases. The payload reaches the reaction surface in the worst operational scenario, where the FluxJet is positioned on the opposite side of the tube with the maximum roll angle, as described in Section 5.2.1. Conversely, the payload can be sufficiently retracted to align flush with the vehicle fuselage. The actuator's range accommodates both the minimum and maximum required motions, providing some margin at both ends. The singularity issue, discussed in Section 5.2.2, is effectively avoided, as it cannot be reached.

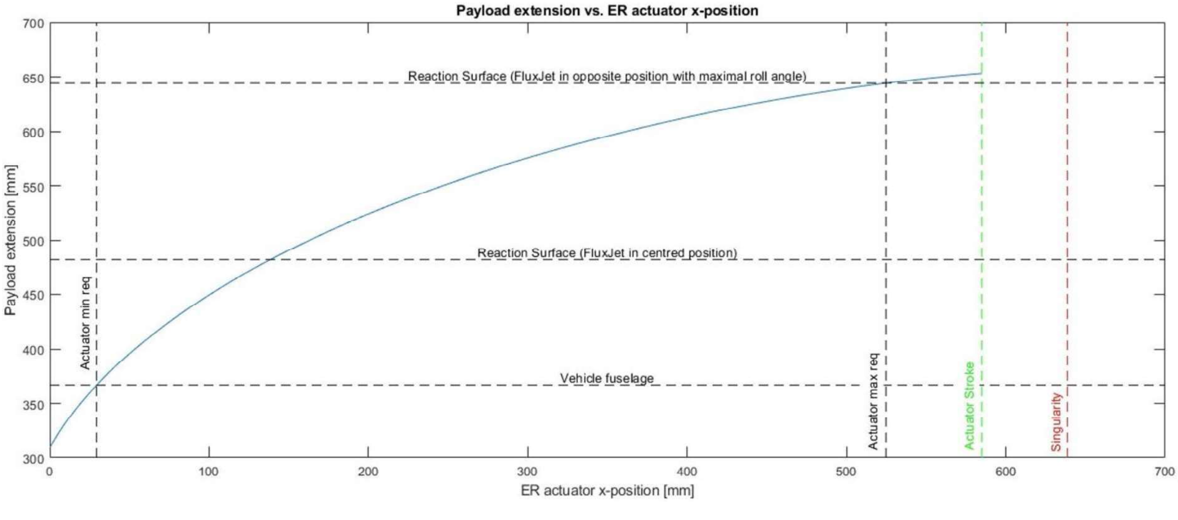


Figure 50: Payload extension vs. actuator position for the independent configuration

The extending factor depicted in Figure 51 indirectly indicates the required actuator velocity. Initially, as the payload begins to extend from the vehicle, the displacement of the payload exceeds the actuator's displacement (extending factor > 1). This means a small actuator displacement results in a significant payload movement. Conversely, for most of the actuator stroke, when the payload is already extended, the extending factor is less than 1, indicating that the actuator's velocity must exceed the payload extending velocity.

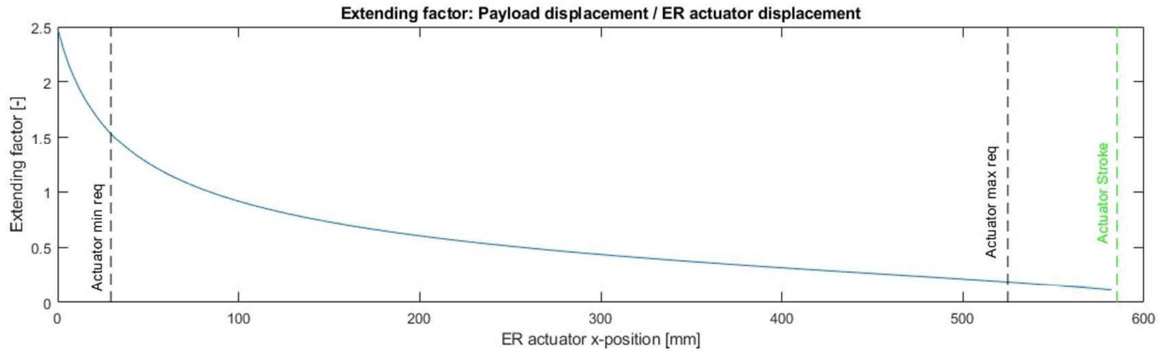


Figure 51: Extending factor for the independent configuration

In Figure 52, the actuator forces are shown. The top two graphics illustrate the actuator force for a BS system located at the top right of the vehicle (BS angular position = 45°, see Figure 41), while the bottom two graphics show the force for a BS system at the bottom right (BS angular position = 135°). The left graphics represent the resultant actuator force when no external forces act, meaning the system only supports the payload mass (45 kg). The right graphics show the actuator force with an additional longitudinal force of 10,000 N in the positive direction, simulating the vehicle being thrusted forward. A positive actuator force indicates a pushing action to maintain stroke position, while a negative force indicates pulling. The actuator force also varies with the BR angle, as the payload weight vector direction changes with the BR angle.

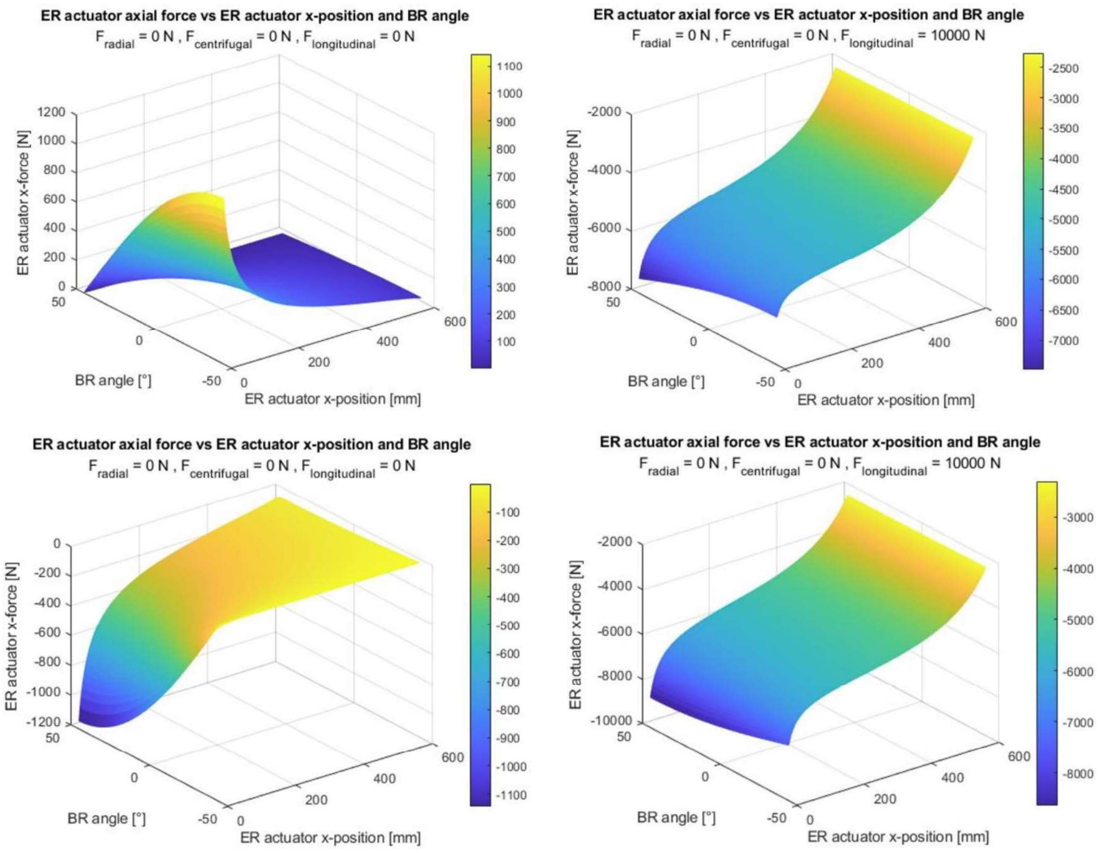


Figure 52: Actuator forces results for the independent configuration

Several conclusions can be drawn from these results. When no external forces act on the payload (left graphics), meaning the payload is inactive, the maximum forces occur in the initial range of actuator movement, where the angle between the active beam and actuator is small. As the payload extends, the active beam approaches a perpendicular position, reducing the actuator force to zero. The actuator force is also zero when the payload is positioned exactly lateral to the vehicle, where the payload weight vector points in the Y-direction (according to the coordinate system in Figure 45), and the modelled actuator support only forces in the XZ-plane. This lateral position corresponds to a BR angle of 45° for the top right BS system (top left subfigure) and -45° for the bottom right BS system (bottom left subfigure).

Finally, the BR and WR results are presented. For the independent configuration, BR and WR actuators are excluded from the analysis. Therefore, the results are calculated for the required range of roll motions, not for the actuator range. Four graphics in Figure 53 illustrate the BR torque at the BR axis, depending on the actuator position, which alters the payload extension, and the WR angle, which primarily shifts the payload weight vector. In the four cases shown, centrifugal and radial forces (see Figure 24) vary according to the subfigure subtitles. External forces remain constant relative to the payload, independent of the BR or WR angles.

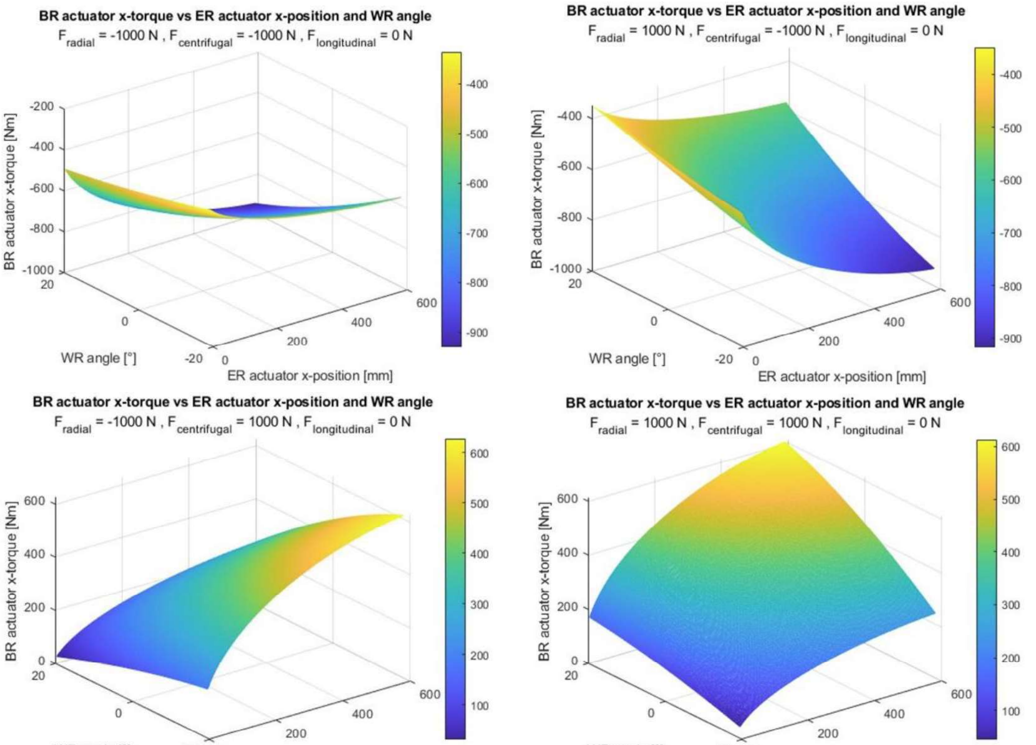


Figure 53: BR torque results for the independent configuration

The absolute BR torque increases when the payload is positioned laterally, increasing the distance from the BR axis to the payload weight vector. Conversely, the torque decreases when the payload is positioned closer to the top or bottom of the vehicle, reducing this distance. Similarly, the torque increases with greater payload extension due to the increased distance from the BR axis to the external force vectors and decreases when the payload is retracted.

Next, the WR torque results are shown in Figure 54. The WR torque depends on the BR and WR angles, as these determine the payload weigh vector's direction. External forces remain constant relative to the payload. The WR torque increases as the payload moves closer to the lateral position from a vehicle point of view, because in this position the perpendicular distance from the WR axis to the payload weight vector reaches the maximum possible. External forces, such as centrifugal forces, introduce an offset in the WR torque.

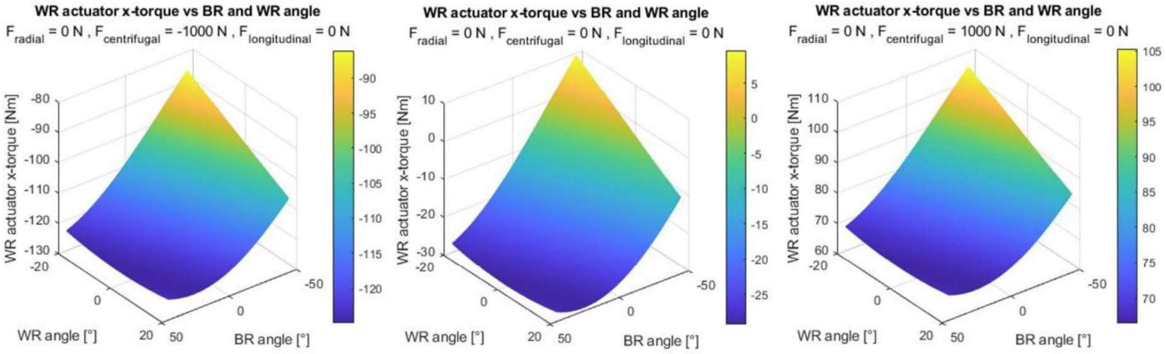


Figure 54: WR torque results for the independent configuration

Thus far, the results for the individual analysis mode have been presented for the independent configuration, which helps to understand the system's behaviour and the forces experienced at different payload positions. As explained in Section 5.2, the algorithm also offers a parameter analysis mode. This mode allows the designer to explore how system displacement or system forces vary across a variable design parameter. For example, the actuator force and payload extension distance are analysed for a variable distance between the BR axis and the ER actuator (Δe_A in Figure 37, referred to as “h_BR_axis_to_ER_actuator” in MATLAB) between 80 and 140 mm.

Figure 55 illustrates how the actuator force increases with increasing Δe_A. A smaller Δe_A value prevents the primary main beam from reaching positions where it aligns more parallel to the actuator, thereby minimizing the maximum required actuator force. Consequently, a minimum Δe_A value is recommended.

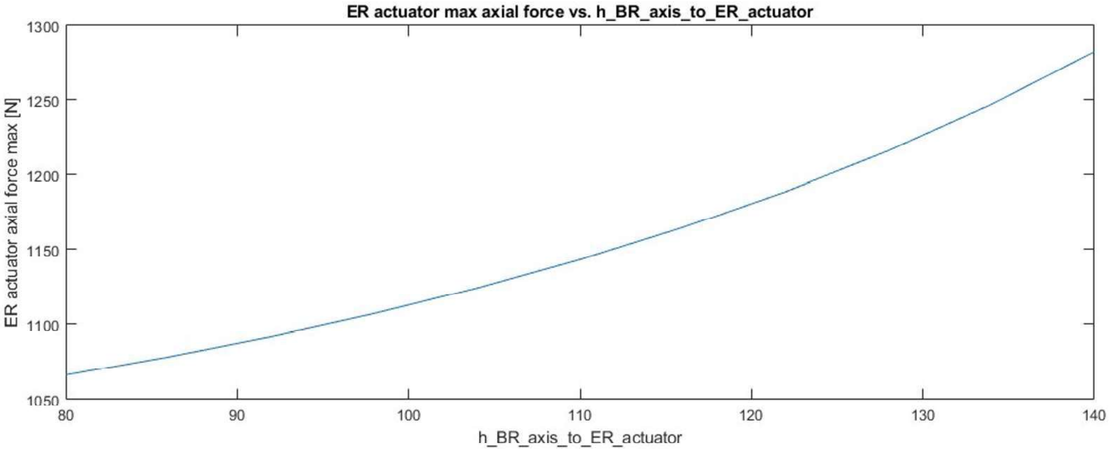


Figure 55: Maximal actuator force vs. distance between BR axis and actuator for the independent configuration

However, as shown in Figure 56, increasing Δe_A reduces the maximum achievable payload extension. Given a required extension of approximately 645mm (Figure 50), the distance between the BR axis and the actuator must be at least 100mm for the calculated configuration. It is important to note that this conclusion is specific to the current set of parameters under consideration, where Δe_A is the variable being analysed. Any changes to other design parameters may alter the results.

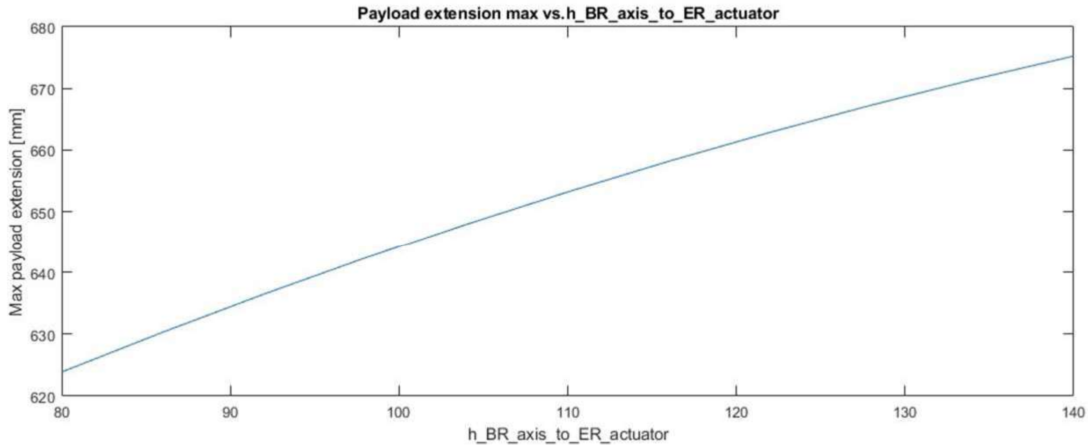


Figure 56: Maximal payload extension vs. distance between BR axis and actuator for the independent configuration

The design parameter analysis approach can be utilized for all design parameters, offering a method to derive conclusions and make optimal design decisions. Although further examples of the parameter analysis mode are not included in this thesis, the emphasis remains on the individual analysis mode. This mode enables a detailed visualization of the system's behaviour across various configurations.

5.3.2. Outputs of the dependent in-line configuration

For the calculations related to the dependent in-line configuration, a similar set of input parameters is used as for the independent configuration. The analysis follows the same sequence, beginning with displacement graphs and concluding with actuator force assessments. As before, internal forces within the beams are not displayed.

In the dependent in-line configuration, payload displacement is controlled by the left and right actuators. Figure 57 illustrates the payload extension and the BR angle depending on the displacement of both actuators. When both actuators extend, the payload extension increases, and when they retract, it decreases. The BR angle remains unchanged if both actuators extend or retract equally. However, the BR angle becomes positive when the left actuator is more extended than the right actuator and negative in the opposite scenario. The BR angle increases when the left actuator extends more than the right and decreases otherwise.

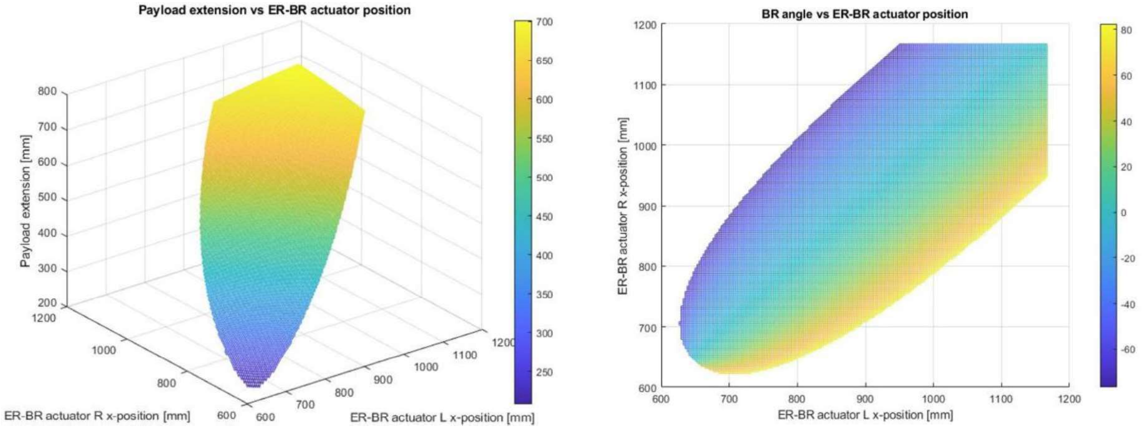


Figure 57: Payload ER position and BR angle vs. actuators position for the dependent in-line configuration

The results show a wide range of achieved BR angles (-80° to +80°) due to theoretical calculations (Section 5.2.3). In practice, the range required is significantly narrower, limiting the payload required positions to a smaller set of points.

The extending factor in Figure 58 represents the ratio between the payload extension displacement and the extension displacement of a single actuator. This factor is calculated solely based on ER displacement, not BR displacement, providing insight into the required actuator velocity. As in the independent configuration, this factor is less than one in most payload positions, indicating that the actuators must operate at a higher velocity than the payload velocity for the ER displacement.

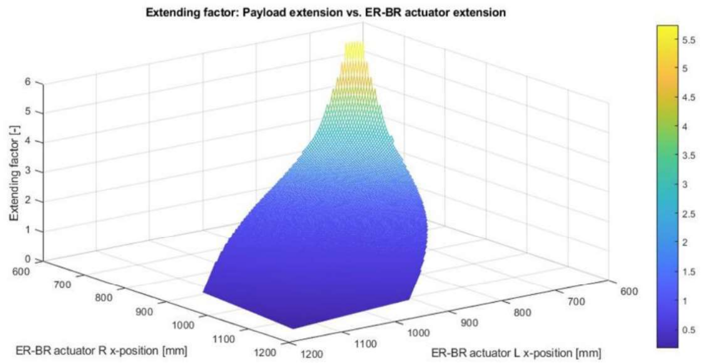


Figure 58: Extending factor for the dependent in-line configuration

Figure 59 presents the actuator forces for the left and right actuators, shown in the left and right subfigures, respectively, for a BS system located at the bottom right of the vehicle (BS angular position = 135°, see Figure 41). The top graphics depict the resultant actuator forces when no external forces act, supporting only the payload mass (45 kg). The bottom graphics illustrate the actuator forces with an additional longitudinal force of 10,000 N in the positive direction, simulating forward thrust. A positive actuator force indicates a pushing action to maintain the stroke position, while a negative force indicates pulling.

As observed, the actuators compensate for each other. Due to the system's symmetry and the actuators' specific positioning, when one actuator is pushing, the other is pulling, and vice versa in the scenarios presented. In other situations, such as when there are significant radial forces, both actuators work together, either both pulling or both pushing, to balance these forces.

The WR torque results are identical to those of the independent configuration, as the calculations are the same, and thus, they are not presented separately.

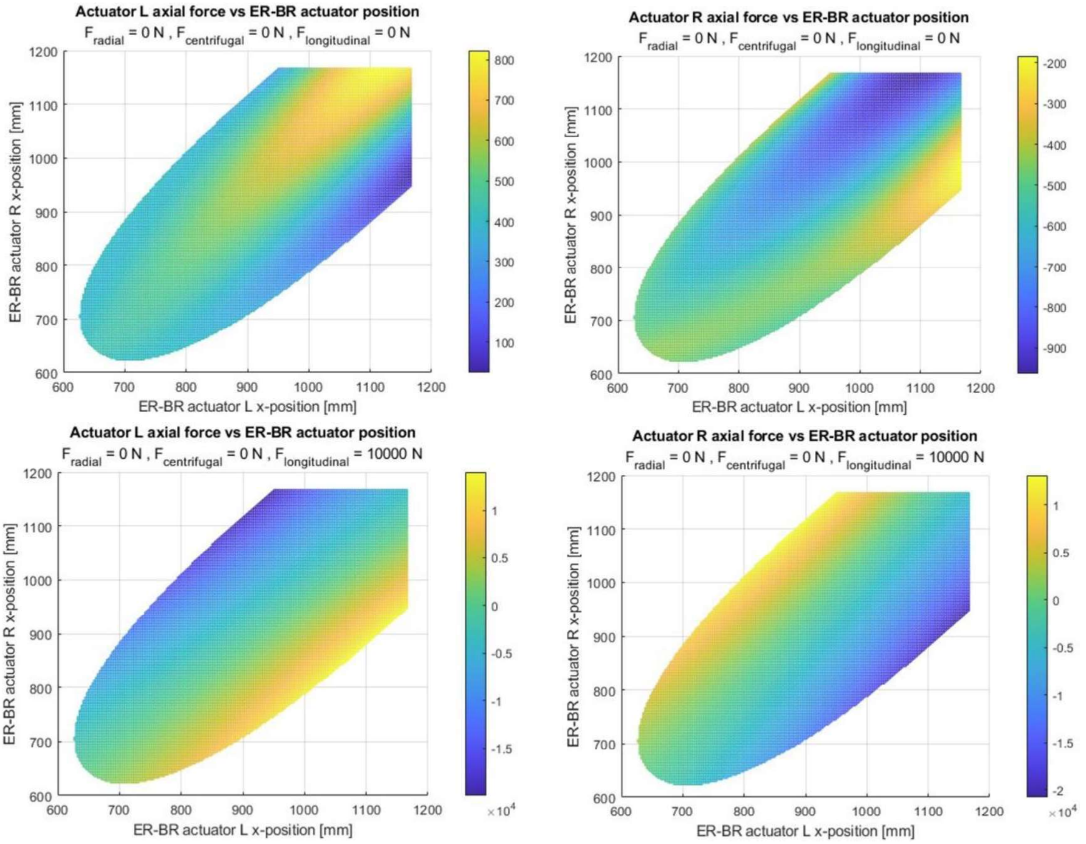


Figure 59: Actuator forces results for the dependent in-line configuration

5.3.3. Outputs of the dependent transverse configuration

The results presented for the dependent transverse configuration include parameters for a Boom Suspension (BS) system with an Intermediate Power Pickup (IPP) system as the payload, used in one of the small-scale FluxJet demonstrators. This BS is positioned laterally on the vehicle (BS angular position = 90°, see Figure 41) and its payload weighs approximately 0.25 kg. First, the displacement results are presented, followed by the actuator forces results, excluding the internal forces within the beams as previously done.

Figure 60 illustrates the payload extension and the BR angle as functions of the displacement of both actuators. When both actuators extend, the payload extension increases; conversely when they retract, the extension decreases. The BR angle remains unchanged if both actuators extend or retract equally. However, the BR angle becomes positive when the left actuator is more extended than the right actuator and negative when the right actuator is more extended than the left.

In the dependent transverse configuration, payload displacement is controlled by the left and right actuators similarly to the dependent in-line configuration, but with different actuator positions. In this configuration, the system has active beams, and the direction of the actuator displacement is in the transverse plane. This modifies the displacement range, as shown in Figure 60 , compared to Figure 57. In the dependent transverse configuration, all actuator positions are physically possible, whereas in the dependent in-line configuration, not every combination of actuator positions is feasible.

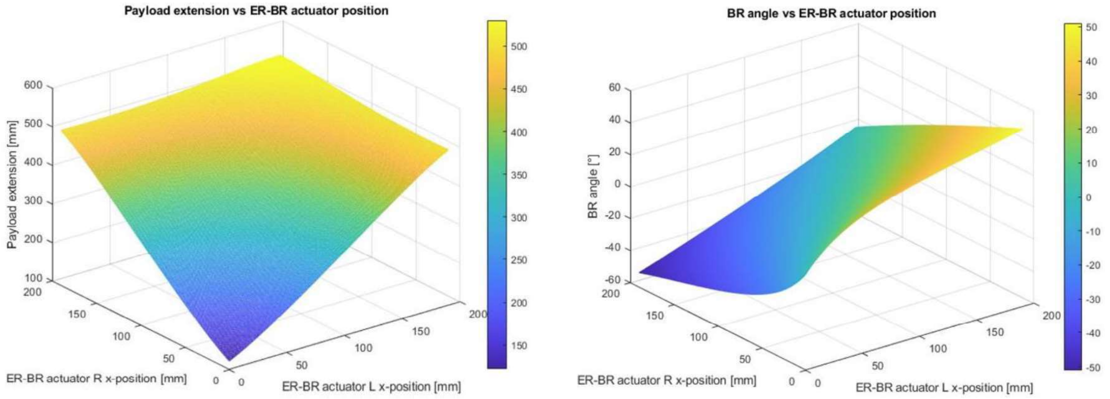


Figure 60: Payload ER position and BR angle vs. actuators position for the dependent transverse configuration

The extending factor in Figure 61 represents the ratio between the payload extension displacement and the extension displacement of a single actuator. This factor is calculated solely based on the ER displacement, not the BR displacement, providing insight into the required actuator velocity. Unlike the independent and dependent in-line configurations, this factor is greater than one in most payload positions, indicating that the actuators must operate at a lower velocity than the payload velocity for the payload ER displacement.

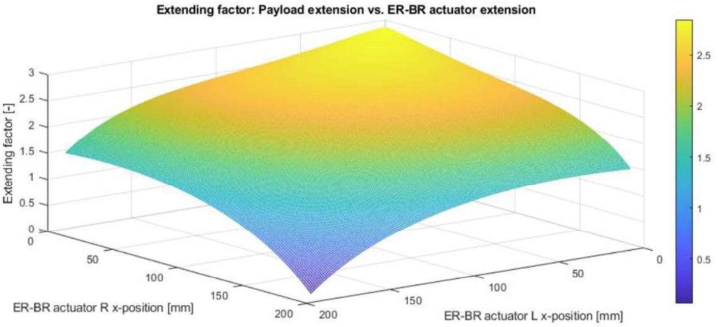


Figure 61: Extending factor for the dependent transverse configuration

Figure 61 presents the actuator forces for the left and right actuators, depicted in the left and right subfigures, respectively. The top graphics show the resultant actuator forces when no external forces act, supporting only the payload mass. The bottom graphics illustrate the actuator forces with an additional radial force of 100 N in the negative direction, simulating a force pushing the payload into the vehicle, akin to the force from contact between the IPP payload and the reaction surface. Note that the axes of the two bottom graphics are inverted compared to the top ones for illustrative purposes. A positive actuator force indicates a pushing action to maintain the stroke position, while a negative force indicates pulling.

In the scenario with only the payload weight (top subfigures), the left actuator, located on the upper side of the vehicle, exerts a pulling force (negative axial force), while the right actuator, located on the bottom, exerts a pushing force (positive axial force) to counterbalance the weight of the payload. The actuators work together to maintain stability.

In the scenario with the additional radial force pushing the payload into the vehicle (bottom subfigures), both actuators exert pushing forces to maintain the payload position against the radial force. This scenario is hypothetical; in reality, the payload would continuously adjust to new operational positions to match the position of the reaction surface.

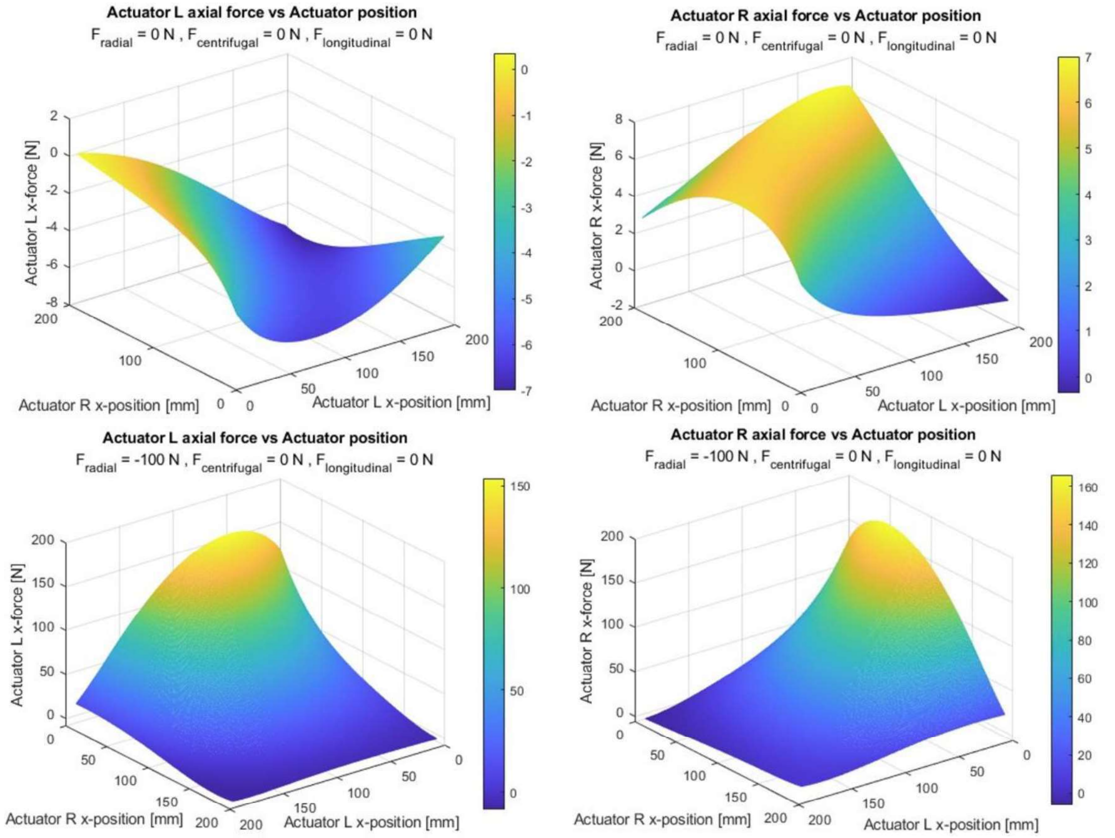


Figure 61: Actuator forces results for the dependent transverse configuration

The WR torque results are the same as those for the other configurations, given that the calculations are identical. Therefore, they are not presented separately.

6. Conclusion

This thesis explored the design and analysis of the Boom Suspension (BS) system within the FluxJet project, employing Model-Based Systems Engineering (MBSE) techniques to enhance system performance and adaptability. The research established a reusable, parameter-based design foundation, highlighting the significant advantages of MBSE in managing complex mechatronic systems. The use of MBSE, specifically through the V-Model and ARCADIA methodology, facilitated clear communication among stakeholders and streamlined system development thanks to the creation of a precise and unambiguous system model.

A primary goal of this study was to reduce the development time for the BS system. To achieve this, the research involved a comprehensive examination of logical architectures, with a particular focus on actuator technologies and their impact on system efficiency. This led to the development of a specialised algorithmic tool, designed to assist system designers in analysing and optimising BS configurations. The tool provided visualisations of payload displacement, actuator velocities, and required forces, offering a detailed understanding of system behaviour under various configurations.

The algorithm proved effective in identifying valid solutions quickly, ensuring that designs met all operational requirements while optimising efficiency and performance. Additionally, it facilitated a more efficient decision-making process regarding actuator selection, providing a robust framework for choosing suitable system configurations based on specific requirements.

A key achievement of this work is the streamlined development process for the BS system, which significantly reduces the time and effort required to design and evaluate different configurations. The use of MBSE methodologies not only enhanced system transparency and traceability but also facilitated a more structured and systematic approach to design, ensuring that all critical aspects of the system were thoroughly considered.

In summary, this thesis contributes a robust framework for the design and analysis of the BS system within the FluxJet project, highlighting the practical benefits of MBSE in the development of complex mechatronic systems. The methodologies and tools developed in this work are aimed at ensuring the success of the project and contributing to advancements in the field of mechatronic system design.

7. Outlook

The development of the Boom Suspension (BS) system within the FluxJet project presents several promising avenues for future research and practical implementation, particularly through the continued application and expansion of Model-Based Systems Engineering (MBSE). While this thesis has laid a solid foundation, further advancements can provide more comprehensive and robust solutions.

A milestone for future work is the completion of the physical analysis following the definition of the logical architecture. Transitioning from conceptual designs to tangible implementations is essential to ensure that all components and system requirements are accurately defined and met. This includes specifying the actual components required for the BS system, which is a necessary step to move from theory to practice.

Moreover, a more detailed analysis of the Wrist Roll (WR) function is needed. The current study does not fully address WR functionality, as the actuators for this motion were not investigated in depth. Incorporating this aspect would allow for a more thorough understanding of the system's dynamics, offering a complete analysis of all movements and their implications on system performance.

The algorithm developed in this thesis, while effective, currently addresses only three BS configurations. Expanding the algorithm to include additional configurations would enhance its utility and applicability, allowing for a broader analysis of potential design solutions. This would cater to a wider range of operational scenarios and payload requirements, providing a more flexible tool for future projects.

Improving the efficiency and speed of the algorithm will be particularly beneficial when handling a large volume of calculations across a wide range of parameters. Optimising the code to increase its processing speed will facilitate more extensive simulations and analyses, allowing for faster iterations and more detailed exploration of different design scenarios.

The inclusion of a yielding mechanism in the algorithm is another critical area for further development. By modelling and integrating this mechanism, the algorithm could perform calculations related to the behaviour of the yielding mechanism and its interaction with the overall BS system. This would provide deeper insights into how the system behaves under yielding conditions, enhancing the precision of the design process, and ensuring that the system can meet specific operational requirements related to this mechanism.

The tool currently allows for the analysis of each configuration in isolation. Enhancing it to introduce comparative metrics or graphics would allow for a more intuitive comparison of different logical architectures. Developing coefficients or other comparative measures could help identify the optimal configuration by providing clear, quantifiable criteria for evaluation.

The broader implications of this research extend well beyond the specific case of the FluxJet BS system. The use of MBSE in this context has demonstrated its value in modelling and developing complex systems. The techniques employed in this study can be applied to other engineering challenges, offering methods that are transferable across mechatronic systems or other domains. This highlights the versatility and effectiveness of MBSE as a tool for managing complexity and ensuring system reliability and performance.

In practical terms, the next critical step involves transitioning from algorithmic calculations to physical testing. Hardware-In-the-Loop (HIL) testing is essential for validating the theoretical models developed in this thesis, ensuring that they align with real-world performance. This phase will verify that the designed systems meet the rigorous demands of real-world applications.

Continued refinement of the BS system, alongside the expansion of its scope and the inclusion of practical testing, will further advance the FluxJet project. These efforts are likely to influence the development of other advanced transportation systems, providing a model for rapid and efficient system development. The structured approach outlined in this thesis lays a solid foundation for future endeavours, aiming for a more efficient, reliable, and adaptable design process. This methodology is instrumental in developing the various versions of BS systems necessary for different payloads and the various TransPod vehicle prototypes planned for future construction.

Bibliography

Millar, C., Lockett, M., Ladd, T. 2018. Disruption: Technology, innovation and society. Technological Forecasting and Social Change, 129, pp. 254–260.

Herrera-Quintero, Luis Felipe, Jaime Chavarriaga, Klaus Banse, Diego Bermudez, and George Proeller. 2019. "Disruptive Technologies in Intelligent Transportation Systems." 2019 2nd ITS LATAM. IEEE.

Barber, Ramon, David R. Rosa, Antonio Flores-Caballero, and Santiago Garrido. 2023. "Deployment of Model-Based-Design-Adaptive Controllers for Monitoring and Control Mechatronic Devices." Applied Sciences, vol. 13, no. 22, Article 12432. DOI: 10.3390/app132212432.

TransPod Inc. "TransPod - Ultra-High-Speed Tube Transportation." Accessed August 2, 2024. https://www.transpod.com.

TransPod Inc. 2021. "Public Summary Alberta Feasibility Study: Summary of Study on the Implementation of a TransPod Line from Calgary to Edmonton." June.

Walden, David D., Garry J. Roedler, Kevin J. Forsberg, R. Douglas Hamelin, and Thomas M. Shortell, eds. 2015. INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities. 4th ed. San Diego, CA: International Council on Systems Engineering. Published by John Wiley & Sons, Inc. ISBN: 9781118999400.

ISO. 2011. ISO/IEC/IEEE 42010:2011 Systems and Software Engineering - Architecture Description. International Organization for Standardization.

Friedenthal, Sanford, Alan Moore, and Rick Steiner. 2014. A Practical Guide to SysML: The Systems Modeling Language. 3rd ed, pp. 46-230. San Francisco: Morgan Kaufmann. ISBN: 9780128002025.

ISO. 2015. ISO/IEC/IEEE 15288:2015 Systems and Software Engineering - System Life Cycle Processes. International Organization for Standardization.

Sheard, Sarah A. 1996. "Twelve Systems Engineering Roles." INCOSE International Symposium, vol. 6.

Elm, Joseph P., and Dennis R. Goldenson. 2012. "The Business Case for Systems Engineering Study: Results of the Systems Engineering Effectiveness Survey." Software Engineering Institute, Carnegie Mellon University.

Graessler, Iris, and Julian Hentze. 2020. "The New V-Model of VDI 2206 and Its Validation." at – Automatisierungstechnik, vol. 68, no. 5, pp. 312–324. DOI: 10.1515/auto-2020-0015.

Samon, Jean Bosco, and Damasse Harold Fotsa Tchouazong. 2022. "A Review Analysis of Design Models for Mechatronic Product Design Perspectives." International Journal of Innovative Technology and Exploring Engineering (IJITEE), vol. 11, no. 4, March, pp. 71-78.

INCOSE. 2007. Systems Engineering Vision 2020. Version 2.03, October 3.

Borky, John M., and Thomas H. Bradley. 2019. Effective Model-Based Systems Engineering. Springer International Publishing. ISBN: 978-3-319-95668-8. DOI: 10.1007/978-3-319-95669-5.

Ramos, A., Ferreira, J. V., & Barcelo, J. 2012. “Model-Based Systems Engineering: An Emerging Approach for Modern Systems.” IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 42(1), 101-111.

Holt, J., & Perry, S. 2010. “SysML for Systems Engineering: A Model-Based Approach”. Institution of Engineering and Technology, 95-153.

Estefan, J. A. 2007. "Survey of Model-Based Systems Engineering (MBSE) Methodologies." INCOSE MBSE Initiative. 14-26.

PivotPoint Technology Corp. "MBSE with SysML Training." PivotPoint Technology. Accessed August 2, 2024.

Bahill, Terry, and Wayne Wymore. 1999. Model-Based Systems Engineering. CRC Press. ISBN: 9780203746936. DOI: 10.1201/9780203746936.

Guo, Yuqiang, Hui Zhao, Keke Qi, and Junxian Guo. 2022. "Analysis of the Mainstream MBSE Methodologies from the Modeling Practice View." In Proceedings of CECNet 2022, edited by A.J. Tallón-Ballesteros, 146-154. China Helicopter Research and Development Institute. DOI: 10.3233/FAIA220529.

Thales Group. "Arcadia: A Tooled Method to Define, Analyse, Design and Validate System, Software, Hardware Architectures." Datasheet.

Voirin, Jean-Luc. 2023. Arcadia Reference: Workflow and Activities. Thales Group.

Milecki, A. (2015). 45 Years of Mechatronics – History and Future. In: Szewczyk, R., Zieliński, C., Kaliczyńska, M. (eds) Progress in Automation, Robotics and Measuring Techniques. ICA 2015. Advances in Intelligent Systems and Computing, vol 350. Springer, Cham. https://doi.org/10.1007/978-3-319-15796-2_13.

Nof, Shimon Y., ed. 2023. Springer Handbook of Automation. 2nd ed. Springer International Publishing. ISBN: 978-3-030-96729-1. DOI: 10.1007/978-3-030-96729-1.

Sima, Liviu Mihai. 2022. "Design and Testing Ways for Mechatronic Systems." International Journal on Cybernetics & Informatics (IJCI), vol. 11, no. 1/2, April, pp. 61-69.

Van Amerongen, Job. 2007. "Mechatronic Design: A Port-Based Approach." Proceedings of the 4th International Symposium on Mechatronics and its Applications (ISMA07), Sharjah, U.A.E., March 26-29.

Toshiyoshi, Hiroshi, and Stefan M. Westervelt, eds. 2006. Emerging Actuator Technologies: A Micromechatronic Approach. Wiley-VCH. ISBN: 9780470091982. DOI: 10.1002/9780470091982.

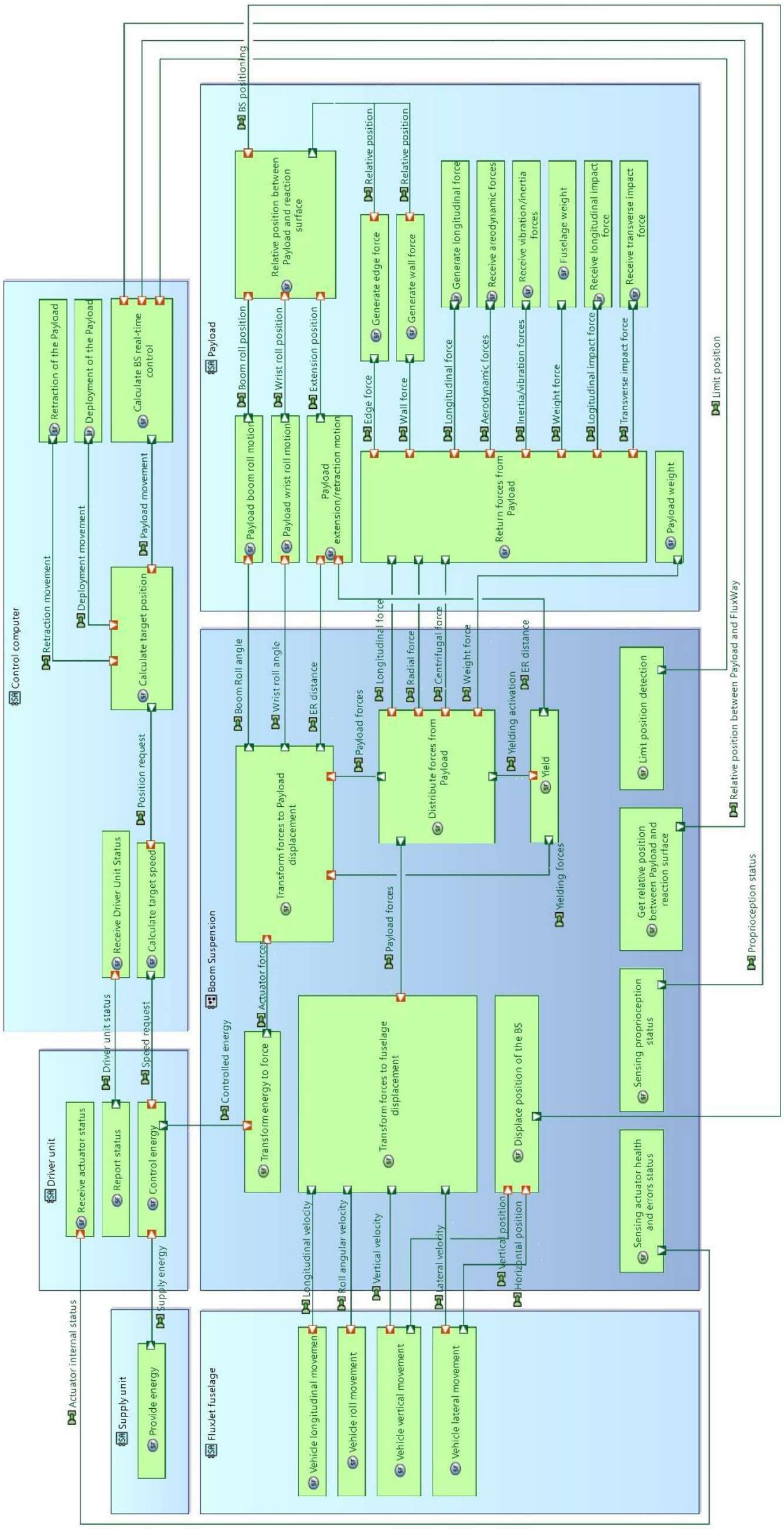
Poole, A.D., and J.D. Booker. 2011. "Design Methodology and Case Studies in Actuator Selection." Mechanism and Machine Theory, vol. 46, no. 5, pp. 647-661. ISSN: 0094-114X. DOI: 10.1016/j.mechmachtheory.2010.12.009.

ISO. 2018. ISO/IEC/IEEE 29148:2018 Systems and Software Engineering - Life Cycle Processes - Requirements Engineering. International Organization for Standardization.

Janzen, Ryan. 2017. "TransPod Ultra-High-Speed Tube Transportation: Dynamics of Vehicles and Infrastructure." Procedia Engineering, vol. 199, pp. 8-17. ISSN: 1877-7058. DOI: 10.1016/j.proeng.2017.09.142.

L. Asmaa, K. A. Hatim and M. Abdelaaziz, "Localization algorithms research in wireless sensor network based on Multilateration and Trilateration techniques" 2014 Third IEEE International Colloquium in Information Science and Technology (CIST), Tetouan, Morocco, 2014, pp. 415-419, DOI: 10.1109/CIST.2014.7016656.

Appendix 1: System architecture diagram



Appendix 2: System architecture diagram