Parking assist for multi-articulated vehicles

Distributed Systems Minor Project Report

Figure 1: TRENS Solar Train (Solico, n.d.)

HAN_UNIVERSITY OF APPLIED SCIENCES

Project title: **Parking assist for multi-articulated vehicles.**

Name/number of the group: Group 3

Name of students:

Abhijeet Vijay Kotwal	1652725
Joeri Bosman	2115571
Mattia Imperatori	1662246
Tjoma Dyundikov	2106085
Anton van Veen	1679717

HAN Supervisor: Nikhil Muthakana, Dixon Devasia

Date: 13th June, 2023

SUMMARY

This project, titled "Parking Assist for Multi-Articulated Vehicles," presents a comprehensive approach to address the steering control challenges encountered when reversing a tractor-trailer combination. The project encompasses a tractor and trailer articulation model, showcased in Figure 1 on the report's cover. By employing a distributed system architecture, the project aims to demonstrate successful steering control, along with various concepts such as membership function, voting logic, fault tolerance, redundancy, and the establishment of effective communication between boards using a carefully selected communication protocol derived from theoretical considerations.

Moreover, this project surpasses the limitations of traditional single articulation systems by incorporating a tractor and two trailers. By showcasing the functionality and steering control in scenarios involving both single and double articulation, it highlights the versatility and adaptability of the system. Furthermore, this demonstration opens up possibilities for the application of these concepts in even more complex scenarios, such as triple articulation or beyond, broadening the project's potential impact.

In this configuration, the tractor and each of the two trailers are equipped with microcontrollers. The system demonstrates successful membership service by seamlessly recognizing the addition of a new trailer to the tractor. Consequently, it updates the respective kinematics of the extended system, enabling accurate calculation of the steering angle. Remarkably, this is achieved without the need to reflash or update the older boards, showcasing the system's efficiency and flexibility.

To ensure fault tolerance and redundancy, multiple microcontrollers are incorporated into each unit (tractor and trailer). A voting mechanism is implemented to identify and isolate any faulty board, notifying users of the issue. This implementation employs Triple modular redundancy (TMR) to enhance system reliability.

Furthermore, this concept is extended to address failures in communication between the units as well as within each individual unit. This is achieved by implementing two Controller Area Network (CAN) channels between the units, ensuring robust communication, and mitigating the impact of potential communication failures.

Lastly, the findings are showcased through a process of validation and verification. This involves conducting a series of tests across various scenarios, aligning with the carefully planned and organized project structure. These tests serve to confirm and demonstrate the effectiveness of each section of the project, providing a thorough evaluation of its performance.

HAN_UNIVERSITY OF APPLIED SCIENCES

CONTENTS

1 INTRODUCTION

1.1 Background

Multi-articulated vehicles, also known as articulated vehicles or articulated lorries, are a type of vehicle that consists of two or more rigid sections that are connected by a pivoting joint. This joint, called an articulation, allows the vehicle to bend in the middle, making it easier to maneuver through tight spaces and around corners. These vehicles are typically used for transporting goods over long distances and are commonly seen on highways and major roads. They are popular in the transportation industry because they are able to carry large amounts of cargo while still being able to navigate through urban areas with ease.

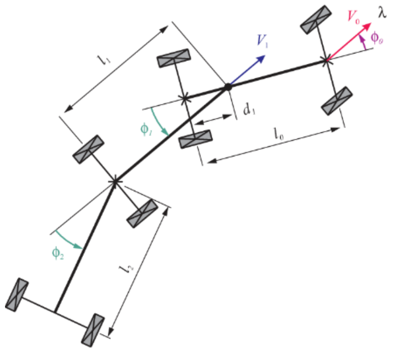

Figure 2. Inverse Kinematic Model (Sklyarenko, Schreiber, & Schumacher, 2013)

The TRENS Solar Road Train is such an articulated vehicle, aimed for use in urban areas to transport cargo and people. The fully electric "citytrain" named "Solar People mover" consists of one tractor followed by up to three wagons sections, allowing it to carry up to 90 people (TRENS B.V., n.d.). A key feature of the Road Train is the modular nature of the vehicle design. Multiple variations of trailers can be added, allowing the vehicle to be fully multi-purpose. A key challenge that comes with the modular nature of the vehicle is ensuring the robustness of the system. The vehicle's control system must at all times know what the current configuration of the vehicle is and in what order the trailer are connected. This requires a highly available, fault tolerant distributed system which ensures that the vehicle remain operational when hardware components might fail. For this to be achieved fault-tolerance mechanics such as redundancy in hardware components are introduced.

While multi-articulated vehicles offer many advantages in terms of cargo capacity and maneuverability, they also require skilled and experienced drivers to operate safely. Due to the chain-like configuration of sections, maneuvering the vehicle backwards is extremely complicated, where small errors in steering direction can lead to large deviations from the desired path of the rear section.

1.2 Problem definition

It is true to say that the major drawback of a multi-articulated vehicle is that the driving in the reverse direction is an open loop unstable control problem. When a lead vehicle "pulls" passive trailers, the car leads the system and trailers follow the vehicle. The last trailer is at the head of the system when an articulated vehicle is driving in the reverse, which turns the control of the trailers into a pushing motion. It becomes hard to predict the direction of the last trailer of the vehicle since it is a highly unstable system, which becomes increasingly more unstable with the number of trailers. The admissible direction of pushing is also limited by the maximum steering angle of front wheels.

1.3 Project Objectives

A reverse driving assistant system for articulated vehicles can reduce the difficulty of reversing for drivers. This system controls the steering operation of the tractor based on the number of trailers and the desired path direction of the last trailer, which the driver provides. The design and build of such systems should be economic, easily implementable and have redundancy measures built into it for real-life fault possibilities. The tractor would be considered as the main node and each consecutive trailer as a node. The steering mechanism of the tractor will be emulated using a DC-motor.
Hence, the primary goal of this project is to exemplify the logic and concept behind developing a fault-tolerant reverse assist system for multi-articulated vehicles incorporating a membership function, enabling seamless articulation and effortless operation. Through this demonstration, the project aims to showcase the practical implementation and effectiveness of the designed system.

1.3.1 Sub-Objectives

The sub objectives of this project can be defined as:

- Implementing a control algorithm using the provided kinematic model.
- Designing a system architecture that enables articulation angle sensing, membership service, and steering actuation.
- Making the final system fault-tolerant, highly available, and redundant.

1.4 Research questions

The goal of report is to answer the following research questions, which consists of the main research questions and a set of sub-questions. These are listed as follows.

Main research question:
How can a fault-tolerant and redundant real-time system architecture be designed for a reverse assist system in multi-articulated vehicles?

Sub-questions:
- What strategies can be implemented to ensure the availability of the system in the face of hardware failures and other potential disruptions?
- What are the key design considerations for a fault-tolerant and redundant real-time system architecture?
- How can membership service be provided in the system architecture to enable communication between the tractor and each trailer node?
- How can we develop nodes that are highly available to the system, ideally that can be freely swapped in the system?
- What is the best approach to implement position control for steering actuation, and how can it be integrated into the system architecture?
- How can we test the different functionalities in the different nodes?
- How can the system architecture be tested and validated to ensure that it is robust and fault-tolerant, and how can real-time performance be evaluated under different failure scenarios?
- How can the availability of the components be evaluated?

1.5 Requirement Analysis

Within this chapter the requirements of the system and the project will be given. The requirements of the system are given by MoSCoW Prioritization method. MoSCoW is a prioritization technique for helping to understand and manage priorities. The letters stand for: "Must Have", "Should Have", "Could Have", "Won't Have this time" (Agile Business Consortium, 2016). It is not relevant for this project to mention "Won't Have this time" criteria since it won't be introduced for this time due to simplification purposes.

System requirements:

Table 1: System Requirements

Number	Requirement	MoSCoW
1	The system must successfully perform a steering action based on a given reversing direction set point of the last trailer, for articulated system up to 4 trailers.*	**Must**
2	The system architecture must be fault tolerant, alerting the operator in case of complete communication or node failure while bringing the system to a safe ground state. **	**Must**
3	The system must be able to identify each node in the system and recognize the order of nodes in the articulated chain.	**Must**
4	The system must be able to detect and isolate faulty nodes.	**Must**
5	The system should be able to correct its steering action based on live events on the system. (i.e., different static articulation angles between trailers)	**Should**
6	The system operator should be able to change the desired trailer direction at all times.	**Should**
7	The articulation angle between trailers can be emulated using potentiometers as angle sensors between articulation points.	**Could**
8	The system could have a simple HMI to monitor the systems state and allow for operator warnings.	**Could**

*: requirement changed during implementation of project to 2 trailers.

**: System is fault operational, operator is alerted

1.6 Approach

The project began by establishing a well-structured plan, which outlined the necessary steps and identified key milestones to be achieved. Extensive research was conducted to gather relevant literature, serving as a valuable resource throughout the project's duration. This research not only provided a foundation of knowledge but also inspired new ideas and explored alternative methods to accomplish the project's objectives.

By studying the experiences and insights of professionals in related fields, the project team gained a deeper understanding of the concepts and principles behind the membership function, voting systems, and fault tolerance. This knowledge played a crucial role in shaping the system design and implementation strategies.

Furthermore, the team leveraged Diego's TTCAN template as a starting point. This decision not only saved time but also provided valuable guidance for handling complex tasks efficiently. By building upon this template, the team was able to streamline the development process and focus on customizing it to meet the specific requirements of the project.

Overall, the combination of meticulous planning, extensive literature research, and leveraging existing templates enabled the team to make informed decisions, avoid unnecessary trial and error, and achieve the project's goals effectively. It laid a solid foundation for success by drawing upon the expertise of others, exploring different approaches, and optimizing the implementation process.

1.7 Outline of the minor project report

With the project objectives and approach now defined, we have the following order to present a brief overview of the outline of what is presented in the report

Section 2 : Literature review

Within this section, we offer a concise summary of the extracted and valuable information obtained from these papers. This summary serves to provide an overview of the content that proved insightful and aided in understanding the project's working principles. These papers were carefully selected based on their relevance and applicability to facilitate the project's development.

By including this overview, we aim to acknowledge the contribution of these papers in shaping the project's design and highlighting their significance in providing valuable insights.

Section 3: Methodology

In this section, we provide an overview of the project's development process, outlining the steps taken to achieve the project goals and ultimately conclude the work. The purpose of this section is to offer guidance and insight to individuals interested in recreating the system.

The section begins by discussing the system design and kinematic architecture, providing a comprehensive understanding of the overall structure and functionality. We then delve into the design of the individual units, starting with the tractor, where we explore its unique features and considerations Next, we address the implementation of fault tolerance through a voting mechanism. This aspect of the project focuses on ensuring system reliability and robustness by incorporating redundancy and the ability to detect and isolate faulty components.

Finally, we delve into the implementation of the membership service, which plays a crucial role in managing the communication and coordination between the various components of the system

By providing an overview of these key steps and aspects of the project, this section aims to offer valuable insights and guidance to those interested in replicating and understanding the system's construction and functionality.

Section 4: Results

This section will present images of achieved results displaying the functionality and working of concepts like membership service, fault tolerance (via voting), successful communication in a distributed architecture and will include section in verification and validation.

Verification shall be caried out via vigorous testing and demonstrated by going through an established test plan based on the requirements defined above.

Validation is demonstrated by coherence to expected theory and verification tests.

Section 5: Conclusion

This section will return to the primary project objective and give a short description of achieved milestones.

Section 6: Discussion

This will include some critical thoughts and analysis, along with future scope of project.

1.8 GitHub Software Repository

The software developed for this project is stored in a GitHub repository, which serves as a centralized location for version control and collaboration. GitHub is a popular platform for hosting and managing code repositories, providing a robust set of tools for software development teams.

By utilizing GitHub, we ensure that the software is securely stored and accessible to the project team members. The repository contains the complete source code, documentation, and any additional resources related to the software.

To access the software repository, please follow this link: https://github.com/JoeriBosman111/MES-Distributed-Systems-Minor-Project.git

Through this link, you will be able to view and download the software developed during this project.

2 LITERATURE REVIEW

This chapter contain a literature review

2.1 Fault Tolerance/ Redundancy

Fault tolerance and redundancy are crucial components for maintaining the dependability and uninterrupted operation of distributed systems. These techniques serve as safeguards against hardware failures, software errors, and various fault scenarios, ensuring that systems can sustain their functionality without substantial interruptions.
In the world of distributed systems, there are various types of faults that can happen in both the hardware and software parts. These faults have different characteristics and can be classified accordingly. To make sure the system keeps running smoothly and reliably, it's important to use suitable redundancy measures to tackle these faults.

For the purposes of this project various papers were studied and a few were shortlisted based on the valuable insight and applicability to the project.
The first (Sanjay Bansal, 2019) titled A Detailed Review of Fault-Tolerance Techniques in Distributed System explores two essential aspects of distributed architecture: the capability to handle multiple faults and the performance of distributed systems. This paper focuses on different techniques for fault handling and tolerance that can be implemented in a distributed architecture. These techniques are designed to be applicable across a wide range of fault types and can be deployed in large- and small-scale distributed systems. This paper inspired the implementation of N-modular redundancy approach which was selected for the project.

The Second paper (P. Balasubramanian, 2016) titled "A Fault Tolerance Improved Majority Voter for TMR System Architectures" delves deeper into the N-Modular Redundancy by Triple modular redundancy implementation. This paper lists the importance and wide use of TMR in safety critical applications such as Avionics and shipboard electronics, TMR is suited to handle a wide range of fault possibilities across both software and hardware infrastructure (for one module). This paper introduces the general scheme of TMR and voting implementation.

Third, the paper (György Györök, 2022) titled "Fault-tolerant Software Solutions in
Microcontroller Based Systems" presents a very unique simplistic way of designing a voting system for TMR implementation. Additionally, the paper addresses various types of system faults that may occur and demonstrates how the TMR approach effectively mitigates them. This paper significantly influenced the design and implementation of a TMR (Triple Modular Redundancy) voting system in the project, resulting in the successful realization of reliable voting and fault tolerance capabilities.

By comparing these papers, the project gains insights into fault-tolerant techniques in distributed systems, the application of N-modular redundancy, the specific implementation of Triple Modular Redundancy (TMR), and a simplified voting system design for TMR. These studies contribute to the project's objective of achieving fault tolerance and uninterrupted operation in the distributed system being developed.

2.2 Membership service

In the context of distributed systems, the membership service plays a crucial role in managing redundancy and ensuring the operational status of processing nodes. One influential paper in implementing a membership service is "a process membership service for active safety systems." the paper describes a process membership protocol specifically designed for distributed real-time systems that utilize both time-triggered and event-triggered message passing for communication between processing nodes (ecus).

By presenting the process membership protocol for distributed real-time systems, the paper offers valuable insights into the design and functioning of the membership service. It addresses the need for redundancy management in architectures where distributed applications share a common set of processing nodes. The protocol enables a group of co-operating processes to establish a consistent view of each other's operational status, ensuring the correct functioning of the system.

Through its detailed description of the protocol, the paper provides a deeper understanding of the concepts, challenges, and mechanisms involved in implementing a membership service. This knowledge has been instrumental in developing an effective approach to managing redundancy and ensuring the reliable operation of the distributed system.

In terms of their focus, the first paper provides a comprehensive review of fault-tolerance techniques, while the second and third papers delve deeper into the implementation of TMR and its voting schemes. The first paper lays the groundwork for understanding various fault-handling techniques, while the second and third papers provide more specific insights into TMR and its application in fault-tolerant systems.

Overall, the paper has played a vital role in advancing the understanding and implementation of membership services in distributed systems, enabling better decision-making.

Furthermore, Diego López's master thesis from 2022 proved to be valuable in comprehending the design principles of basic cycles and the potential integration of a membership service within the system. This, in turn, greatly aided and served as inspiration for the architecture design and implementation in the current project.

2.3 Articulated Vehicles/Inverse Kinematic model

The single and double articulation dynamic equations were provided, and based on these equations, a model was developed in Han Coder. This model was designed to perform necessary calculations using inputs such as rotation sensors. To test the model, multiple simulations were conducted using realistic sample values for articulation angles, wheelbase length, trailer length, hitch point, etc. These sample values were sourced from the master thesis report by (Luijten, 2010). The testing process instilled confidence in the accuracy of the kinematic model design.

Furthermore, insights from (Raksincharoensak, 2019) shed light on how the steering angle of a truck trailer combination is influenced by the articulation angle. Combining the information from these three sources aided in the design, testing, and implementation of the Inverse Kinematic model.

3 METHODOLOGY

3.1 Description of research design

3.2 Time-Triggered Architecture

The time-triggered architecture used for the project is based on the HANCoder TTA template provided by (López, 2022). This template provides a template for using TTCAN using HANCoder and Simulink and allows for board initialization and global time synchronization. This template is used as a solid base for the development of the TTA system.

3.2.1 Timing

The TTA system runs on Olimex STM32-E407 microcontrollers. These microcontrollers are run on a hardware clock frequency of 1 MHz. The TTA template uses interrupts in the hardware physical clock to increment the local clock every 100 ticks. This results in a local clock with a frequency of 10 KHz. This local clock is used for the matrix cycle of the TTA system, which contains either a computational task, or a communication task. To ensure all tasks have sufficient time to be completed successfully, tasks are assigned a certain amount of local ticks after being triggered. For computational tasks a window of 4 ticks is reserved, for communication 80 ticks are reserved. The reason why the communication window is so long, is because multiple boards perform a communicational task at the same instance, for which sufficient needs to be appointed to ensure all messages are successful.

3.2.2 Initialization and synchronization

The TTA template assigns all nodes in the system and board ID based on digital pins on the board which are pulled HIGH over 5V. These ID's are fixed and unique to each board. The initialization will only continue with board ID's which are predefined in the software to be part of the system. For this system, nine boards are part of the system, which create three sets of three Fault Containment Units (FCU). Each FCU can represent a trailer or tractor in the articulated chain. The following board ID's form the following FCU's:

Table 2:Board and trailer ID initialization

FCU (ID)	**Board 1 ID**	**Board 2 ID**	**Board 3 ID**
Tractor (3)	1	2	3
Trailer 1 (1)	4	5	6
Trailer 2 (2)	7	8	9

Each board will recognize which FCU they are part of with their trailer ID, which is based on their board ID. The master initialization of the TTA template is developed for use with 3 boards in the system. With the expansion of the system, this initialization remains unchanged due to time constraints. To facilitate this, only the tractor boards will participate in the race for time master, as the tractor is the only FCU to always be present in the system.

During initialization, all boards wait for a complete matrix cycle duration (1116 local ticks) to receive an message from the time master containing temporal data. At first initialization, no board will be master so no other board will receive any message. After the initialization timeout, board 1 of the tractor FCU is set master at default and will start the first basic cycle BC0. BC0 is adopted form the TTA template and provides time synchronization and master voting and assignment.

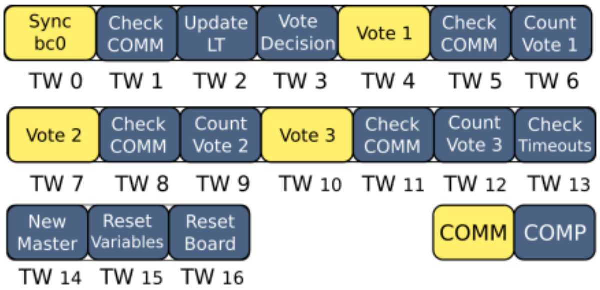

Figure 3: Basic Cycle 0 of TTA template (López, 2022)

Figure 3 shows BC0 as per the TTA template, where blue squares represent computational tasks and yellow squares represent communication tasks. When the cycle is first started by the first master, it will send a synchronization message which will initialize all other boards in the system to the same local ticks and basic cycle. Afterwards, the three boards of the tractor FCU vote who is or should be the master, based on amount of errors per board. While only the tractor boards vote, all boards perform their own assessment. The outcome of the voting decides which boards is to be master for the next cycle.

3.2.3 Matrix Cycle

The matrix cycle of the system consists of three basic cycles which are run consecutively by all boards. This includes BC0 as explained before. Basic Cycle 1 (BC1) and Basic Cycle 2 (BC2) are separate by intra- and inter-FCU functions.

During BC1 all intra-FCU functions are performed. This means that during this cycle, the CAN busses are reserved for communication between nodes inside the FCU's. This cycle starts with another synchronization tasks per the TTA template, after which the first implemented task is reading the potentiometers. All nodes of each FCU is connected to a potentiometer, which they all read during this task. Different FCU's use this reading for different purposes, but all FCU's are fit with Triple Modular Redundancy by implementing three nodes. All nodes in the FCU will communicate its reading of the potentiometer, after which an voting logic is performed to identify faulty values and nodes. The node which is guaranteed to have a correct reading, will be assigned FCU master. The FCU master is later in the matrix cycle allowed to share its potentiometer reading with the other FCU masters of other FCU's.

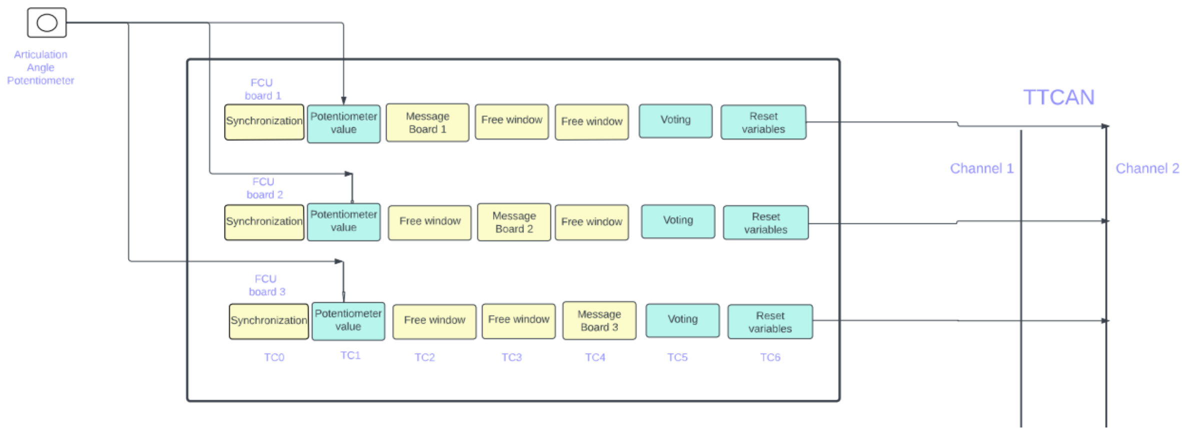

Figure 4: Basic Cycle 1

During BC2 all inter-FCU functions are performed. Again, this cycle starts time synchronization by the master. During this cycle, the FCU masters of trailer 1 and 2 communicate their potentiometer value, which is accepted by the tractor as the articulation angle between trailers and/or tractor. After receiving these angles, the tractor FCU master will use its potentiometer reading as the user input for desired reversing direction for the last trailer. An Inverse Kinematic Model is used to calculate the steering angle required to maneuver the last trailer towards its desired direction.

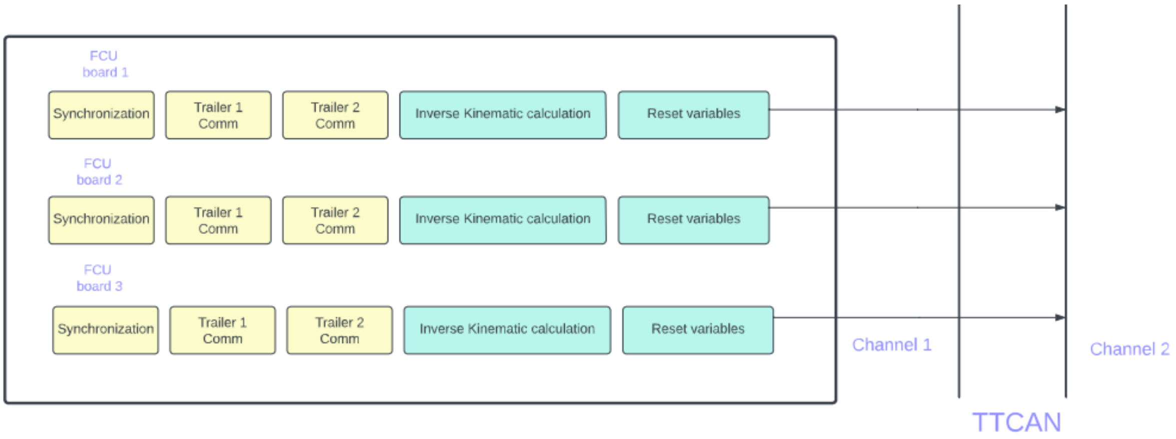

Figure 5: Basic Cycle 2

3.3 Inverse Kinematic Model

Inverse Kinematics is a fundamental concept that involves determining the joint angles required to achieve a desired end effector position or trajectory. In the context of tractors with single and double articulation, inverse kinematic modeling plays a crucial role in controlling and maneuvering these complex vehicles.

When modeling the inverse kinematics for single articulation tractors, the key task is to establish the relationship between the desired position or trajectory of the tractor's end effector and the joint angles of the articulation points. This involves considering the mechanical constraints and linkages of the tractor's components, such as the pivot points.

The general formula for the steering angle (δ_{I2K}) can be represented by the following equation:

$$\delta_{I2K} \ = \ tan \cdot \left(L_{0f} \cdot \frac{\dot{\theta}_0}{v_0} \right)$$

Depending on the scenario being used, the steering angle can be calculated using the following formulas:

For single articulation:

$$\dot{\theta}_1 \ = \ |v_1| \cdot \frac{\tan(\delta_c)}{L_{1f}}$$

$$\dot{\theta}_0 \ = \ sign(v_1) \cdot sign(L_{0b}) \cdot \left(-\frac{v_1}{L_{0b}} \cdot \sin(\gamma) + \frac{L_{1f}}{L_{0b}} \cdot \dot{\theta}_1 * \cos(\gamma) \right)$$

$$v_0 \ = \ v_1 \cdot \cos(\gamma) + \ L_{1f} \ \cdot (\dot{\theta}_1) \cdot \sin(\gamma)$$

For double articulation, the following formulas are added and modified to the existing formulas for single articulation:

$$\dot{\theta}_2 \ = \ |v_2| \cdot \frac{\tan(\delta_c)}{L_{1f}}$$

$$\dot{\theta}_1 \ = \ sign(v_2) * sign(L_{0b}) \cdot \left(-\frac{v_2}{L_{0b}} \cdot \sin(\gamma_2) + \frac{L_{1f}}{L_{0b}} \cdot \dot{\theta}_2 \cdot \cos(\gamma_2) \right)$$

$$v_1 \ = \ v_2 \cdot \cos(\gamma) + \ L_{1f} \ \cdot (\dot{\theta}_2) \cdot \sin(\gamma)$$

$$\dot{\theta}_0 \ = \ sign(v_1) \cdot sign(L_{0b}) \cdot \left(-\frac{v_1}{L_{0b}} \cdot \sin(\gamma_1) + \frac{L_{1f}}{L_{0b}} \cdot \dot{\theta}_1 * \cos(\gamma_1) \right)$$

It is important to note that in the single articulation scenario, v_1 is a constant, while in the double articulation scenario, v_1 is a variable.

These formulas provide a mathematical representation of the steering angle calculation based on various parameters such as velocity (v), angles (θ), distances (L), and orientations (γ). By applying these formulas, the inverse kinematic model can accurately determine the appropriate steering angle for precise control and maneuvering of the tractor in different articulation scenarios.

By accurately capturing these relationships and implementing the appropriate formulas, the inverse kinematic model enables the generation of the necessary joint angle commands for achieving the desired end effector position or trajectory for both single and double articulation tractors.

3.3.1 Model Overview

The inverse kinematic model developed for the reverse driving assistant system for articulated vehicles is designed to assist drivers in reversing maneuvers by controlling the steering operation of the tractor. The model takes into consideration the number of trailers and the desired path direction of the last trailer provided by the driver. The primary objective of this project is to develop a fault-tolerant reverse assist system for multi-articulated vehicles that incorporates a membership function and enables seamless articulation and effortless operation.

The switching logic plays a crucial role in the membership service and the overall system operation. The switching logic determines the appropriate mode or configuration for the system based on the inputs and conditions. It helps manage the switching between single and double articulation modes, ensuring that the system adapts accordingly to the desired path direction and number of trailers.

The switching logic is implemented using a combination of hardware and software components and based on communication-based approach: the switching logic can rely on communication between the tractor and the trailers to exchange information. Each trailer can transmit its status or configuration, such as whether it is attached or detached, to the tractor. The switching logic in the tractor can analyze this information to make the mode selection.

The membership service, which is an integral part system, relies on the switching logic to determine the operational status of the processing nodes (tractor and trailers) and establish a consistent view of each other's availability. It facilitates the coordination and communication between the nodes, ensuring seamless articulation and effortless operation.

3.3.2 Model Outcomes

The inverse kinematic model has two possible outcomes: single articulation (0) and double articulation (1). The model's logic determines the appropriate articulation mode based on the input parameters and the desired path direction of the last trailer. The single articulation outcome represents a configuration where only the tractor's joints are involved in the steering, while the double articulation outcome signifies the involvement of both the tractor and the trailer joints in the steering mechanism.

3.4 Fault Tolerance/ Redundancy Implementation via voting

Implemented Triple modular redundancy includes a voting logic design, that isolates any board with garbage/ suspicious or no values(disconnection).

The voting logic in question takes the potentiometer values read from each of the three boards as input. The ultimate objective is to identify and isolate the faulty board, while also designating one of the remaining two boards as the master board responsible for communication with the rest of the system

Here now we explore the description of the research design:

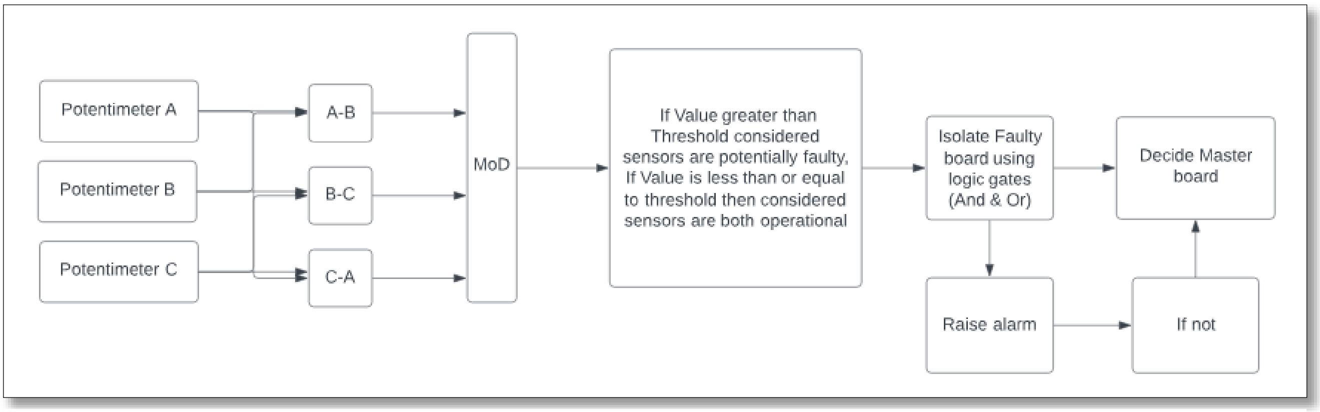

Figure 6: Voting System (logic block diagram)

Each board has a task of sharing its potentiometer value. Once the value is received, it is compared with the values from the other boards using arithmetic subtraction. The main idea is that when all the boards are working correctly and the readings are accurate, the subtraction should ideally result in zero. This happens because all the values come from the same potentiometer, just different boards. Next a modulus block is used since negative values are a possibility due to the reading of one potentiometer could be more than the other.

As a result, we get an error term which is the difference of value between any two boards, now ideally this value should be zero since all boards read same potentiometer, but in practicality there are small differences between each transmitted value. This is due to noise in the system, a filter is used to abate the noise although we cannot design it to eliminate all noise as it would lead to a delay in the final system and loss information about any minor changes applied to potentiometer.

So, we get a small error term upon subtraction of potentiometer values between any two operational boards. Thus, let's assume if Sensor C is faulty then the operation of A-C and C-A would yield large error terms while A-B would still yield a small error term. This would signify that both boards A and B are operational while Board C is non-operational.

Thus, we need to set a threshold for the error, as per variations of noise in the system a threshold of 250 was chosen after observation. So, for example if any one board were to fail or give anomalous values inconsistent with the other boards, its error term would be larger than the threshold, and then a flag is raised that notifies the operator of the faulty board.

Now, In the case when there are no faulty boards and when the faulty one is isolated from the system, the operational boards need to make a decision on who takes up the master role in the system. Two ways in which this could be achieved are; one is to predefine conditions where if A and B are operational then A is master, another is to simply select the board with the lower board ID among the operational boards. The former was followed in this approach. This was done for simplicity.

Thus, this system performs successful voting among all three boards, and notifies the operator of any fault, inconsistency, or anomaly in any one of the nodes.

Justification to the selected approach of voting: After a through literature survey, two types of approaches were implementable in this project.

The first was a weight-based voting mechanism, where each node allocates a weight to similar values received in the system. Thus, any faulty node would give the least amount of allocated weight and can be singled out.

The second was the basic arithmetic approach inspired by the literature (György Györök, 2022) which is the method followed in this project.

The arithmetic approach offered an equal level of voting capability compared to the weight-based approach. Moreover, the weight-based approach necessitates memory blocks for allocating weights based on previous values, resulting in increased computational requirements. Therefore, the arithmetic approach was favored for designing the voting mechanism, considering its simplicity, and reduced computational load.

3.5 Membership service

The goal of the membership service is to recognize and identify the amount of active members in the system, in this case trailers or FCU's. For this system, the amount trailers and their order in the articulated chain, affect the outcome of the inverse kinematic model. The TTA of the system facilitates for using the 'heartbeats' of the FCU's to identify which are present and active in the system.

During BC2, after receiving the articulation angle message form a trailer FCU master, a counter is incremented. This counter checks how many members are active in the system during a cycle. After each BC2, this counter is reset to zero to be able to check for changes in the number of members in the next cycle. Based on the value of the member counter, when a message from trailer 1 or 2 is received, it's trailer ID is saved as either member 1 (the first trailer in line) or member2 (the second trailer in line). This requires trailer to be added sequentially to the system to properly identify the order of trailers.

However, due to the fixed message order, this system would result in trailer 1 to always be recognized as the first trailer of the system, if it were to be added to system later then trailer 2. To prevent this, the system saves the previous configuration of members, and uses a comparative logic to only allow trailer 1 to be the first member, if trailer 2 wasn't the first member in the cycle before. The same logic applies for a reversed situation. In the case trailer 1 is indeed the first member, no ID would be assigned as member 1 and saved as the previous member, which would allow trailer 1 to be recognized by as member 1. This means that the system requires 1 matrix cycle to recognize changes in the members of the system.

```
if message received by trailer 1:
    if member_counter == 1 and previous_member_1 != 2:
        member_1_ID = 1 #ID of trailer 1 is saved as member 1

    else if member_counter == 2 and previous_member_2 != 2:
        member_2_ID = 1 #ID of trailer 1 is saved as member 2


else if message received by trailer 2:
    if member_counter == 1 and previous_member_1 != 1:
        member_1_ID = 2 #ID of trailer 2 is saved as member 1

    else if member_counter == 2 and previous_member_2 != 1:
        member_2_ID = 2 #ID of trailer 2 is saved as member 2

else #No message received or no statements correct
    #All member ID's reset to 0
    member_1_ID = 0
    member_2_ID = 0
    previous_member_1 = 0
    previous_member_2 = 0
```

Figure 7. Pseudo code for membership service

4 SYSTEM DESIGN AND IMPLEMENTATION

4.1 Introduction

In our architectural design, we have incorporated certain principles for simplicity. Abstraction is utilized in the top-level modules (Tractor, Trailer 1, and Trailer 2) and within the TTCAN architecture, employing multiple layers of abstraction to enhance clarity and comprehension of the system.

Each node (Tractor and Trailers) comprises three FCUs (Triple Modular Redundancy) that operate using identical software, ensuring independence among the different boards.

Furthermore, the system maintains consistency by employing global time in the TTCAN implementation, facilitating communication between the various boards. All boards have the same code for the Tractor and Trailer, further contributing to consistency in the system.

4.2 System architecture

In the designed system, each board threaten as node, and each connection of a three boards is treated as a FCU. The steering mechanism of the tractor is emulated using a DC motor, which controls the steering angle of the tractor based on the outputs of the inverse kinematic model. By incorporating a membership function and fault-tolerant mechanisms, the system ensures smooth articulation of the articulated vehicle and facilitates effortless operation during reversing maneuvers.

The design decisions and considerations for fault tolerance in the system include:

- 2 CAN Buses per Board: Redundancy at the communication level with two CAN buses per board.
- 3 Boards per FCU: Increased fault tolerance and redundancy by incorporating three boards per FCU.
- Voting Mechanism: Implementation of a voting mechanism for decision-making based on majority agreement among the boards.

In figure 8, an overview of the design of the system with the communications connections and the components used.

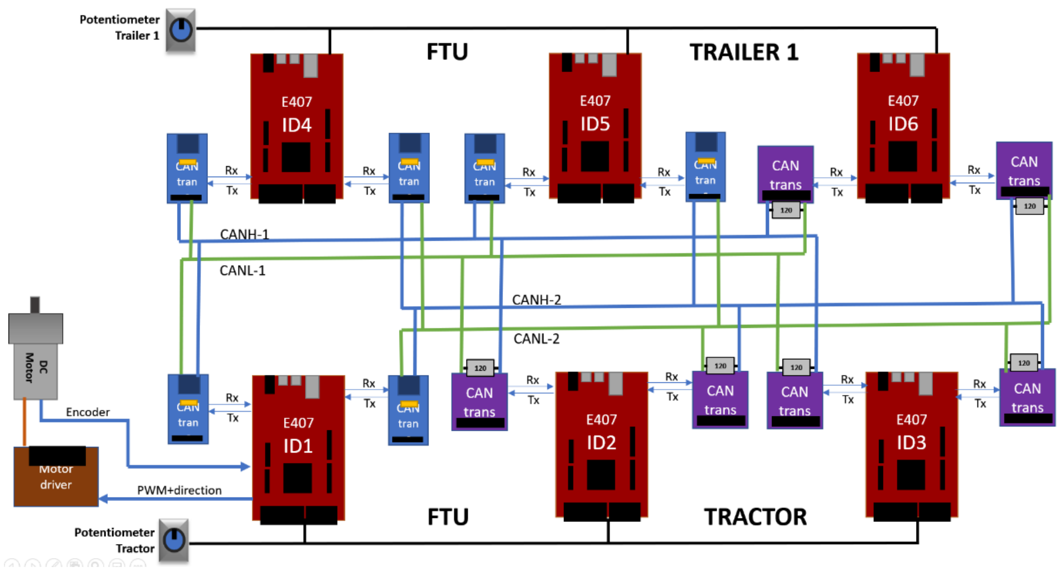

Figure 8. Design of the System (components and connections)

This IO-list (appendix c) serves as a structured overview of the system's inputs and outputs, providing a clear understanding of the connections between components. In appendix C the IO list is included.

4.3 Implementation

For our project we used the following step-by step implementation. The step-by-step approach offers advantages such as enabling manageable and modular implementation. It also facilitates iterative improvement, allowing for continuous enhancements based on testing feedback. Although step-by-step, there was also a certain degree of overlap (if possible) between the different task, for efficiency reasons.

1. The TTCAN template in Simulink was adapted for 1 Matrix Cycle. This involved analyzing the documentation of TTCAN in HANcoder and adapting the schematics for use within our project.
2. CAN bus functioning: Then the system was expanded to include two boards connected via the CAN bus. Establish proper communication and verify the functioning of the CAN bus for reliable data exchange between the boards.
3. As next step, voting mechanism was implemented to our system. This included three boards with a voting mechanism.
4. We added the potentiometers into the system to capture articulation angle data. Ensure that the boards accurately receive and process the potentiometer signals.
5. The kinematic model that relates the vehicle's steering input and articulation angle, was added to the system.
6. Integrate a membership service into the system to enable identification and communication between the nodes (tractor and trailers). This service facilitates seamless coordination and information exchange between the nodes
7. The actuator (DC-motor + motor control) was added. The proven setup as in the Home Taken Exam was used.
8. The 2nd trailer was not added due to time restriction of the project, although the software/Simulink schematics is already prepared for this 2nd trailer.

Documentation throughout the implementation process is done in GitHub (appendix A).

5 TESTING, VERIFICATION AND VALIDATION

5.1 Test approach

We adopted a step-by-step testing approach for the system, including module testing:

- Two CAN bus communication
- Intra-communication within a node consisting of three boards
- Inter-communication between two nodes

  We gradually added functionality, including a voting mechanism, kinematic model, and membership service, followed by comprehensive functional system testing.

5.2 System testing

As last step we conducted testing of the implemented system to see if all the system requirements were met and if the functioning of the system could be validated. This included integration testing to verify interaction between different modules, and system-level testing.

Table 4 contains the system tests which are derived from the system requirements as stated in the approved project plan.

Table 4: Systemic Testplan

Nr.	Requirement	MoSCoW	Tests	Outcome tests
1	The system must successfully perform a steering action based on a given reversing direction set point of the last trailer, for articulated system up to 4 trailers.	Must	A) change potentiometer value of all trailers, read values in HMI B) potentiometer value of last trailer actuates steering wheel tractor	A) potentiometer values from all trailers is received by tractor HMI B) potentiometer value from last trailer is used as input for motor steering value at tractor
2	The system architecture must be fault tolerant, alerting the operator in case of complete communication or node failure while bringing the system to a safe ground state.	Must	A) Power off 1 trailer node board B) Power off 1 tractor node boards C) Disrupt a CAN bus D) Disconnect potentiometer wire of a board	A) On HMI is visible which node/board is faulty, systems keep functioning B) On HMI is visible which node/board is faulty, systems keep functioning C) System keeps functioning D) On HMI is visible which board is faulty
3	The system must be able to identify each node in the system and recognize the order of nodes in the articulated chain.	Must	A) Tractor and each trailer node are recognized by unique ID B) It is recognizable which trailer node is not available	A) ID's are visible on HMI (by ID-number) B) missing trailer is recognizable on HMI of Tractor

Nr.	Requirement	MoSCoW	Tests	Outcome tests
4.	System must be able to identify connected Trailer and number of connected trailers (Membership Service)	Must	A) Check if connected trailers are automatically recognized by the tractor	A) Membership Service concept demonstrated, and steering angle then corrected accordingly
5	The system should be able to correct its steering action based on live events on the system. (i.e., different static articulation angles between trailers)	Should	A) Repeat tests of the first requirement under live action	A) Expected outcomes are same as test 1, should be visible in real time (live action)
6	The system operator should be able to always change the desired trailer direction.	Should	A) Test tractor function of manual steering	A) change potentiometer of tractor and tractor steering-motor should move in right direction
7	The articulation angle between trailers can be emulated using potentiometers as angle sensors between articulation points.	Could	A) is covered in test 1	
8	The system could have a simple HMI to monitor the systems state and allow for operator warnings.	Could	see previous tests for monitoring and operator warnings on Tractor and Trailer HMI (HANtune)	A) Tractor HMI: see figure 9 B) Trailer HMI: see figure 9

5.3 Verification and Validation

To verify and validate the system performance, The testing plan (Table 4) is executed to ensure that the system performance meets the requirements specified (Table 2).

To begin a general standardized output of the HMI is shown in figure 9 below, each test is carried out and the respective displays are explained with regard to each test. Each board in a tractor or trailer unit is referred to as a node in the system.

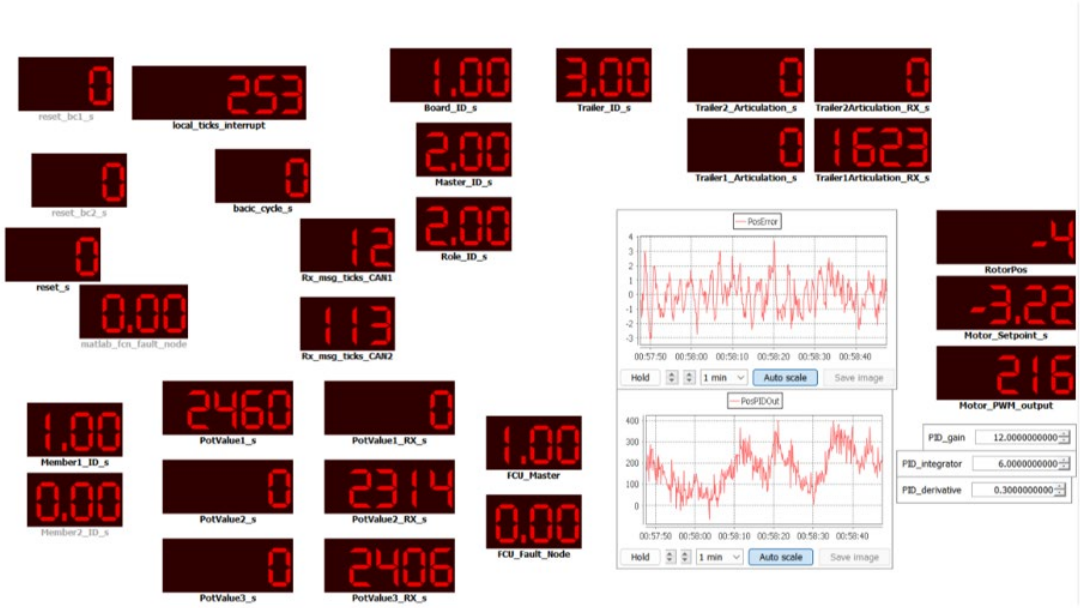

Figure 9. Standard complete HMI interface

In this section we shall now compare the tests listed in section 5.2 with the achieved outcome:

1. Requirement: The system must successfully perform a steering action based on a given reversing direction set point of the last trailer, for articulated system up to 4 trailers.

A) Test A: Change potentiometer value of all trailers, read values in HMI.

Figure 10. Potentiometer Values

Figure 10 shows change in potentiometer value which is sent across system and read in HMI as shown. "_S" denotes value being sent and "RX" denotes value being received across other two trailers.

B) Test B: potentiometer value of last trailer actuates steering wheel tractor

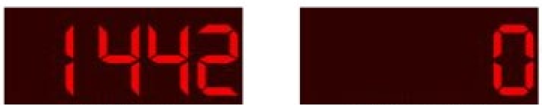

Figure 11. Change of Potentiometer value

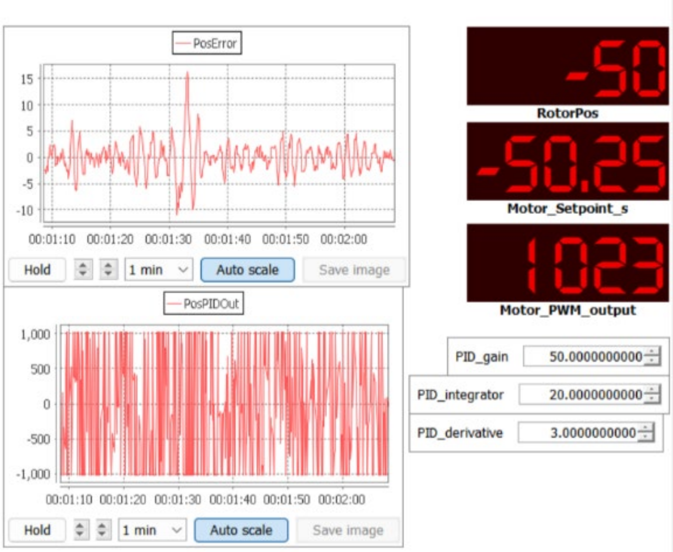

Figure 12. Actuation of steering(rotor) with respect to setpoint

"PotValue1_s" (in figure 11) denotes the potentiometer value of last trailer; "Motor_setpoint_s" (in figure 12) denotes the steering angle (in degrees) setpoint as calculated by the Inverse kinematic model. "RotorPos" denotes the actual position of the rotor (in degrees).

Additionally, "PosError" represents position error (difference between rotor position and setpoint) and "PosPIDOut" represents the output of the PID controller making adjustments to achieve setpoint. Thus, potentiometer value of last trailer was communicated through the designed system and resulted in effective steering in tractor.

2. Requirements:

A) Test A: Loss of Power to One of the Trailer boards(node)

Figure 13. Fault via disconnection of node in trailer

In figure 13 above "PotValue" displays represent potentiometer values being received or sent on each node of trailer respectively where a display ending with "_s" represents value being sent while displays ending with "_Rx" represents values being received.

Boards(nodes) 4, 5, 6 are a part of the trailer. "FCU_master" denotes the board in the trailer that is currently the master.

For this test board 5 is disconnected to simulate loss of power in one of the trailer boards

Upon disconnection "FCU_Fault_Node" display immediately alerts the operator of a fault in the fifth node, thus successfully completing the test and function requirement.

B) Test B: Loss of Power to One of the Tractor boards (node)

Figure 14. Fault via disconnection of tractor node

In this test, one board in the tractor was simulated to have loss of power for fault tolerance and redundancy. (Boards 1,2,3 represent the tractor). In figure 14 above, the board that experienced power failure was board 3, and consequently the system identifies this and raises a warning for the operator as shown by the display "FCU_Fault_Node". Additionally, (via "FCU_Master") we see that board one is the master of the fault containment unit. Finally, the system was still operational despite this failure and could continue with steering angle actuation.

C) Test C: Disrupt a CAN bus (simulation of Fault in one of the CAN buses)

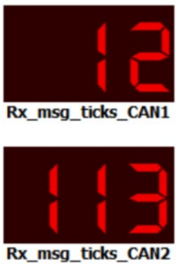

Figure 15. Fault via Disruption of CAN bus

Each CAN bus on receiver end receives message ticks, disruption is simulated via disconnection of CAN bus 2 (simulate fault), when this was done the digital display "Rx_msg_ticks_CAN2" as shown in figure 15 above froze, while the display "Rx_msg_ticks_CAN1" was still running. This represented the fault in CAN bus 2 to the operator via the HMI interface.
The system however even with this disruption was still operational and communication took place via CAN bus 1, effectively demonstrating redundancy in the system.
A demonstration video showing this performance is available via link on GitHub repository and in appendix.

D) Test D: Disconnect potentiometer wire of a board (Failure of Communication of articulation Angle)

Figure 16. Fault in potentiometer (angle sensor)

In this test scenario, fault was simulated via disconnection of potentiometer wire in Node number 3. In figure 16 above each of the displays ending with "_s" denote any values being sent; similarly, each of the displays ending with "Rx_s" denote the receiving of these values. "PotValue1" represents the first node (board) being able to send the value, similarly second node is able to receive it. However due to fault, "PotValue3_RX_s" is unable to receive the updated position of the potentiometer due to disconnection, thus this display freezes.
Additionally, the display "FCU_Fault_Node" alerts the operator of the node on which this fault has occurred (node 3).
By analyzing the display above the operator is able to recognize the nature of the fault and the location of the fault.
The system despite this fault is still operational demonstrating successful redundancy and fault tolerance.

3. Requirements: The system must be able to identify each node in the system and recognize the order of nodes in the articulated chain.

A) Test A: Tractor and each trailer node is recognized by unique ID

Figure 17: Unique ID allocation

As shown in figure 17, the displays "Board_ID_s" specifically show that each trailer and tractor node(board) has a unique ID. These ID's are displayed in the HMI interface when the system is run.

B) Test B: Recognizable which trailer node is not available

The boards displayed in figure 16, show the connected trailer nodes, when a node is unavailable it displays that as "0", additionally if any node is not available, the voting mechanism implemented for fault tolerance takes over and also raises an alert for the operator as shown in Test 2A and 2B.

4. Requirements: System must be able to identify connected Trailer and number of connected trailers (Membership Service).

    A) Test A: Check if connected trailers are automatically recognized by the tractor

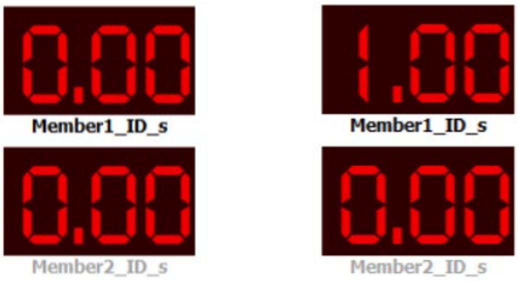

Figure 18: Membership Service

In figure 18, the display "Member1_ID_s" and "Member2_ID_s" are set to represent the number of connected trailers to the tractor thus demonstrating membership service. When only the tractor is simulated both of these displays read zero, when trailer 1 is added, it shows that the tractor has one additional member unit recognizing the newly added unit. Similarly, "Member2_ID_s" is allocated to represent connection of a second trailer.

5. Requirements: The system should be able to correct its steering action based on live events on the system. (i.e., different static articulation angles between trailers)

A) Test A: Change the articulation angles live and observe successful change in steering.

    This live action test is carried out during demonstration and evidenced in performance video link uploaded in GitHub (and in appendix).

6. Requirements: The system operator should be able to always change the desired trailer direction.

    A) Test A: change potentiometer of tractor and tractor steering-motor should move in right direction

This is demonstrated via test 1b (figure 12), the system operator is always in real time able to change the desired trailer direction. Demonstration of live (real time) shown on video link uploaded in GITHUB (and in appendix).

7. Requirements: The articulation angle between trailers could be emulated using potentiometers as angle sensors between articulation points.

A) Test A: Change in potentiometers (represent angle sensors) register as corresponding angle changes in articulation points.

Evident during live testing and represented in figure 12 and video demonstration link on GitHub (and appendix).

8. Requirement: The system could have a simple HMI to monitor the systems state and allow for operator warnings.

A) Test A: Functioning Tractor HMI
Achieved via all the above tests, interface shown in standard figure 9. (Live demonstration video link on GitHub and in appendix)

B) Test B: Functioning Trailer HMI
Achieved via all the above tests, interface shown in standard figure 9. (Live demonstration video link on GitHub and in appendix)

6 CONCLUSIONS AND FUTURE WORK

Conclusion

The main goal of this project, which was to develop a fault-tolerant reverse assist system for multi-articulated vehicles and showcase key concepts, has been successfully accomplished. The tested system demonstrates effective steering actuation for both non-articulated and single-articulation configurations. The verification and validation process were carried out using a carefully designed test plan, and the achieved results were presented.

It is important to note that the system was tested in a conceptual setting and not on actual trailers and tractors, and with short wire lengths not representable of in-vehicle wiring. However, it has been designed in a way that it can be readily implemented in a real-world scenario involving trailers and tractors. The system's capabilities have been sufficiently developed to support such an implementation.

Future Work

The system (hardware and software interface) designed is capable to accommodate 2 trailers and can be further expanded to accommodate further trailers (with the right inverse kinematic equations). However, for the purposes of this project it has been tested with one trailer (single articulation) and only tractor (no articulation) due to deadline constraints. Despite of this, key concepts of distributed systems such as membership service, fault tolerance (voting mechanism), distributed communication architecture, inverse kinematic steering actuation have been implemented and tested in this setup as initially planned. Future scope and expansion could also include failure testing for more than 2 faulty boards across the system.

7 REFERENCES

György Györök, B. B. (2022). Fault-tolerant Software Solutions in Microcontroller Based Systems. *AIS 2017 - 12th International Symposium on Applied Informatics and Related Areas organized in the frame of Hungarian Science Festival*, 1-9.

López, D. M. (2022). *Deployment of a Time-Triggered Architecture schedule for a fully redundant Real-Time Embedded System with MATLAB-Simulink.* Arnhem: HAN UAS.

Luijten, M. (2010). *Lateral dynamic behaviour of articulated commercial vehicles.* Eindhoven: TUE.

P. Balasubramanian, K. P. (2016). A Fault Tolerance Improved Majority Voter for TMR System Architectures . *WSEAS TRANSACTIONS on CIRCUITS and SYSTEMS, Volume 15*, 108-122.

Raksincharoensak, Y. H. (2019). *Automated Steering Control System for Reverse Parking Maneuver of Semi-Trailer Vehicles Considering Motion Planning by Simulation of Feedback Control System.* Tokyo: Fuji Technology Press Ltd.

Sanjay Bansal, S. S. (2019). A Detailed Review of Fault-Tolerance Techniques in Distributed System. *International Journal on Internet and Distributed Computing Systems. Vol: 1 No: 1,*, 33-39.

Sklyarenko, Y., Schreiber, F., & Schumacher, W. (2013). IEEE International Conference on Mechatronics (ICM). *Maneuvering assistant for truck and trailer combinations with arbitrary trailer hitching*, (pp. 774-779). Vicenza, Italy.

Solico. (n.d.). *Composite Engineering for a promising Start-up: the 100% electronic Trens Solar Train.* Retrieved from solico.nl: https://www.solico.nl/composite-news/composite-engineering-for-a-promising-start-up-the-100-electronic-trens-solar-train

TRENS B.V. (n.d.). *TRENS SOLAR CITY TRAINS.* Retrieved from trens.eu: https://www.trens.eu/files/trens-specs-nl.pdf

APPENDIX A

GITHUB repository link:

https://github.com/JoeriBosman111/MES-Distributed-Systems-Minor-Project.git

Demonstration video link:

https://youtu.be/mVxf3LoFHLc

APPENDIX B

The following **Ошибка! Источник ссылки не найден.** contain a description of parameters that in use for a kinematic model.

Table 3: Nomenclature of Inverse Kinematic equations

Symbol	Description
Θ_0	Yaw angle of the tractor
Θ_1	Yaw angle of the 2nd unit
Θ_2	Yaw angle of the 3rd unit
L_{1f}	Wheelbase of the trailing units
L_{0b}	Distance between rear axle & hitch
L_{0f}	Wheelbase of tractor
γ_n	Articulation angle between n-1 & n. We use γ_1 and γ_2
δ_{I2K}	Steering angle at the wheels
δ_c	Desired steering at last unit
v_n	Velocity of the axle of the n^{th} unit, in use are v_1 and v_0

It is important to note that in the single articulation scenario, v_1 is a constant, while in the double articulation scenario, v_1 is a variable.

HAN_UNIVERSITY OF APPLIED SCIENCES

APPENDIX C

IO list of the connections

Node	Board_ID	Description signal	IO connections STM32-E407 boards Board connector	IO name	To board / remarks
Tractor	ID01	CAN2 Tx signal	con3	D1	to Tx of CAN2-transceiver
Tractor	ID01	CAN2 Rx signal	con4	D11	to Rx of CAN2-transceiver
Tractor	ID01	CAN1 Tx signal	PD	4	to Tx of CAN1-transceiver
Tractor	ID01	CAN1 Rx signal	PD	3	to Rx of CAN1-transceiver
Tractor	ID01	Board_ID	con3	D2	to +5V, Configures ID value of Board
Tractor	ID01	Potentiometer	con1	A5	to pin3 Tractor_pot (pin1 to +3.3V; pin2 to GND)
Tractor	ID01	Quad Encoder UEXT3	UEXT conn	pin3	to DC motor encoder1
Tractor	ID01	Quad Encoder UEXT4	UEXT conn	pin4	to DC motor encoder2
Tractor	ID01	direction digital port1	con3	D6	to IN1 of Motor driver board
Tractor	ID01	direction digital port2	con3	D7	to IN2 of Motor driver board
Tractor	ID01	PWM set duty cycle	con4	D13	to ENA of Motor driver board
Tractor	ID02	CAN2 Tx signal	con3	D1	to Tx of CAN2-transceiver
Tractor	ID02	CAN2 Rx signal	con4	D11	to Rx of CAN2-transceiver
Tractor	ID02	CAN1 Tx signal	PD	4	to Tx of CAN1-transceiver
Tractor	ID02	CAN1 Rx signal	PD	3	to Rx of CAN1-transceiver
Tractor	ID02	Board_ID	con3	D3	to +5V, Configures ID value of Board
Tractor	ID02	Potentiometer	con1	A5	to pin3 Tractor_pot (pin1 to +3.3V; pin2 to GND)
Tractor	ID03	CAN2 Tx signal	con3	D1	to Tx of CAN2-transceiver
Tractor	ID03	CAN2 Rx signal	con4	D11	to Rx of CAN2-transceiver
Tractor	ID03	CAN1 Tx signal	PD	4	to Tx of CAN1-transceiver
Tractor	ID03	CAN1 Rx signal	PD	3	to Rx of CAN1-transceiver
Tractor	ID03	Board_ID	con3	D2, D3	to +5V, Configures ID value of Board
Tractor	ID03	Potentiometer	con1	A5	to pin3 Tractor_pot (pin1 to +3.3V; pin2 to GND)
Trailer	ID04	CAN2 Tx signal	con3	D1	to Tx of CAN2-transceiver
Trailer	ID04	CAN2 Rx signal	con4	D11	to Rx of CAN2-transceiver
Trailer	ID04	CAN1 Tx signal	PD	4	to Tx of CAN1-transceiver
Trailer	ID04	CAN1 Rx signal	PD	3	to Rx of CAN1-transceiver
Trailer	ID04	Board_ID	con3	D4	to +5V, Configures ID value of Board
Trailer	ID04	Potentiometer	con1	A5	to pin3 Trailer_pot (pin1 to +3.3V; pin2 to GND)
Trailer	ID05	CAN2 Tx signal	con3	D1	to Tx of CAN2-transceiver
Trailer	ID05	CAN2 Rx signal	con4	D11	to Rx of CAN2-transceiver
Trailer	ID05	CAN1 Tx signal	PD	4	to Tx of CAN1-transceiver
Trailer	ID05	CAN1 Rx signal	PD	3	to Rx of CAN1-transceiver
Trailer	ID05	Board_ID	con3	D4, D2	to +5V, Configures ID value of Board
Trailer	ID05	Potentiometer	con1	A5	to pin3 Trailer_pot (pin1 to +3.3V; pin2 to GND)
Trailer	ID06	CAN2 Tx signal	con3	D1	to Tx of CAN2-transceiver
Trailer	ID06	CAN2 Rx signal	con4	D11	to Rx of CAN2-transceiver
Trailer	ID06	CAN1 Tx signal	PD	4	to Tx of CAN1-transceiver
Trailer	ID06	CAN1 Rx signal	PD	3	to Rx of CAN1-transceiver
Trailer	ID06	Board_ID	con3	D4,D3	to +5V, Configures ID value of Board
Trailer	ID06	Potentiometer	con1	A5	to pin3 Trailer_pot (pin1 to +3.3V; pin2 to GND)