Aalto-yliopisto
Aalto-universitetet
Aalto University

Master’s Programme in Mechanical Engineering

Multibody Simulation and Energy Usage of a Forestry Forwarder

Henri Wahlman

Master’s thesis
2025

Aalto-yliopisto
Aalto-universitetet
Aalto University

Author Henri Wahlman

Title of thesis Multibody Simulation and Energy Usage of a Forestry Forwarder

Programme Master's Programme in Mechanical Engineering

Thesis supervisor Assistant Professor Jari Vepsäläinen

Thesis advisor(s) PhD Researcher Juho Lehto, D.Sc. (Tech.) Jyrki Kajaste

Date 26.05.2025 **Number of pages** 47 + 8 **Language** English

Abstract

The energy efficiency of forestry machinery is considered low by modern standards, with total system efficiency typically remaining below 40%. Energy losses occur primary in powertrain and in hydraulics, especially in valves due to pressure throttling. To be able to achieve higher efficiencies it is needed to research and develop these systems. This can be most effectively performed using simulation models. This thesis presents the development of a forestry forwarder multibody simulation model and highlights the key aspects that needed to be considered during model creation. Working cycles studied with the simulation involved the basic tasks of a forwarder - loading logs onto the trailer and driving. The objectives were to simulate these basic functionalities with manual control and to investigate the energy consumption of the simulated forwarder's actuators. The study examined whether the simulated energy consumption was comparable to a real forwarder consumption. The simulation worked as expected and provided reasonable results. The results reveal the simulation speed and the comprehensively collected energy consumption data.

Keywords Forestry forwarder, multibody simulation, multibody modelling, energy consumption, 3D modelling, contact modelling

Aalto-yliopisto
Aalto-universitetet
Aalto University

Tekijä Henri Wahlman

Työn nimi Kuormatraktorin monikappalesimulaatio ja energiankulutus

Koulutusohjelma Konetekniikan maisteriohjelma

Vastuuopettaja/valvoja Apulaisprofessori Jari Vepsäläinen

Työn ohjaaja(t) Väitöskirjatutkija Juho Lehto, TkT Jyrki Kajaste

Päivämäärä 26.05.2025 **Sivumäärä** 47 + 8 **Kieli** Englanti

Tiivistelmä

Metsätyökoneiden energiatehokkuutta pidetään huonona kokonaishyötysuhteen jäädessä tyypillisesti alle 40 %. Häviöitä tapahtuu pääasiassa voimansiirrossa ja hydrauliikassa, erityisesti venttiileissä painehäviöiden vuoksi. Jotta energian kulutusta voidaan parantaa, on kannattavampaa tutkia näitä systeemejä simulaatiomalleilla kuin todellisilla prototyypeillä. Tässä työssä esitetään kuormatraktorin monikappalesimulaatiomallin rakentaminen ja mitä asioita mallia luotaessa oli huomioitava. Simulaatiossa työkoneella pystyttiin suorittamaan tyypillisiä kuormatraktorin työtehtäviä, kuten tukkien lastaamista lavalle ja ajoneuvolla ajamista. Tavoitteena oli simuloida kuormatraktorin perustoimintoja manuaalisella ohjauksella ja tutkia simuloidun kuormatraktorin toimilaitteiden energiankulutusta. Työssä tarkasteltiin, onko simuloitu energiankulutus verrattavissa oikean kuormatraktorin kulutukseen. Simulaatio toimi odotetulla tavalla, ja siitä saatiin kerättyä realistisia tuloksia. Tuloksista paljastuu simulaation toimintanopeus ja kattavasti kerätyt energiankulutustiedot.

Avainsanat Kuormatraktori, monikappalesimulaatio, monikappalemallinnus, energiankulutus, 3D-mallinnus, kontaktimallinnus

Table of contents

Preface and acknowledgements

This thesis was carried out for the university as the final part of my studies. Here is what I built and it does exactly what it was designed to do. Assembling the simulation model was highly engaging. Now I look forward to working on similar projects. Keeping this work in mind, I feel confident about future challenges. Solving the small things made the biggest difference.

Helsinki, Finland, 26 May 2025
Henri Wahlman

1 Introduction

Finland is the most forested country in Europe, with over 75% of its land area covered by forest containing 20.3 million hectares of forest suitable for timber production [1]. This makes Finland an ideal place for forestry machinery and forestry machine manufacturers. Companies such as Ponsse, John Deere, and Logset operate in Finland, making the country both a user and producer of modern forest technologies.

One of the key machines used in forestry operations is the forwarder. Forwarder is a forestry vehicle used for picking up logs from harvesting sites and transporting them to the roadside. Forwarder often works with a harvester that fells and processes trees for a forwarder to pick up. Typically, a forwarder is a 6- or 8-wheeled vehicle with a cockpit and a trailer with a hydraulic log-loading crane. It has excellent maneuverability due to an articulated steering allowing tight turns. It is designed to operate on rough, sloped and soft forest terrain.

Climate targets push forwarder development towards more efficient and environmentally sustainable solutions. The demand for low emissions and minimal forest impact drives forestry machine manufacturers to focus on fuel efficiency, electric systems and reduced soil disturbance. These improvements are important because operating in challenging environments consumes a significant amount of energy with low efficiency. [2].

To support the development of more efficient and sustainable forwarders, simulation methods have become important. One such method is multibody simulation that allows us to describe real-life mechanical systems as a set of bodies connected by joints and actuators with defined constraints. These connections allow movement and interaction between parts of the system. Multibody simulation is a computational method to study the dynamic behaviour of these bodies enabling accurate prediction of their motion under various forces within a mechanical system [3].

It is expensive and time-consuming to build a real mechanical system for testing purposes, especially in the early development stages. Using a simulation model can reduce the need for physical prototypes. In this thesis, the multibody model was built in a simulation environment, allowing the study of the system’s motion without a physical machine. The model includes the basic movements of the forwarder: driving, crane operations and log handling. This approach allows both motion and performance studies of the whole forwarder.

The research question of the thesis is: “What is the energy usage of the simulated forwarder and how does it compare to real-life systems?”. In this study, energy usage is estimated based on ideal multibody simulation data. The model does not include detailed hydraulic or powertrain systems or their losses, so the energy usage represents an idealized estimation. To obtain the results, a multibody simulation model of forwarder was created and tested. The results were then compared with real-world data based on studies about forwarder fuel consumption.

To compare real-life systems to a simulation model, the variable being compared must be clearly defined. The simulation model and the real system should be as similar as possible to make the comparison meaningful. The measured quantity must also be something that can be compared for example, fuel consumption cannot be evaluated from a simulation if fuel usage is not modelled in any way. However, real fuel usage can be converted to energy consumption using the energy density of the fuel, as demonstrated by Haavikko et al., which makes it possible to compare energy usage between the real system and the simulation [4].

The thesis includes a background chapter that discusses the technologies used in forwarders as well as possibilities for modelling and simulating their features and operations. This provides an overview of the broader context in which the work has been conducted. Following the background, state-of-the-art is introduced where previous research closely related to forwarders and their simulations were reviewed. These works give understanding of what has already done related to forwarder simulations. After that the construction of the simulation model is explained. It tells how a basic 3D model can be turned into controllable multibody system. The following chapters go through the test setup, results and discussion. At the end the work is summarized.

2 Background

Forwarders are complex vehicles designed to operate in demanding forest environments, transporting logs from harvesting sites to roadsides for further transport. To study these machines in detail it is essential to first understand their overall structure, the technological advancements in forestry machinery and the different modelling approaches that can be applied to such machines. A solid understanding of forwarder technologies and awareness of various modelling methods provide context for the simulation study in this thesis.

2.1 Forwarder Systems and Technologies

Forwarder operation relies on electrical, hydraulic, and powertrain systems. The electric system supplies power to electrical components such as lights sensors, operator interface and safety features. Electrical control units (ECUs) control electrical signals for vehicle operation. With sensors all over the machine, ECUs manage the data and transmit the operator's commands to the actuators controlled.

The hydraulic system is primarily for log loading crane and forwarder steering. It converts mechanical engine power into fluid pressure and flow with hydraulic pumps. Actuators such as boom pistons are controlled with actuating valves. Valves control hydraulic flow directed to each cylinder, causing piston movement.

The powertrain handles mechanical power, providing energy from the engine to the wheel rotation. It includes a transmission that adjusts torque output to suit operating conditions. Driveshafts enable 8-wheel drive capability for good traction on rough terrain.

Overall interaction between these systems is shown in Figure 1, that helps illustrate their combined role in forwarder operation. In this simulation model, these subsystems are not included to save computational resources for faster multibody simulation. The hydraulic and powertrain systems are addressed in a separate thesis project at Aalto University that focuses specifically on the forwarder's hydraulic and powertrain systems which complements this work.

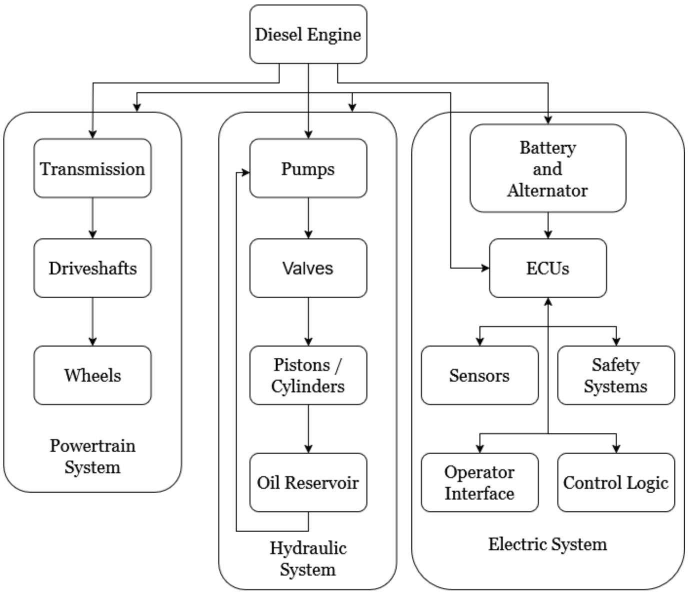

Figure 1. General forwarder subsystems.

Forwarder hydraulic systems commonly rely on load-sensing technology. In these systems, hydraulic pumps are regulated to deliver flow only when needed. A variable displacement pump is controlled to ensure a constant pressure differential across the valves of the hydraulic actuators. [5] Load-sensing systems are constantly researched to enhance their efficiency and performance because their overall efficiency is poor. This is because a single pump typically supplies multiple actuators simultaneously. The pump adjusts its pressure based on the demand of the most demanding actuator in the circuit. However, other actuators connected to the same circuit may not require as much pressure at the same time, resulting in wasted energy by flow throttling. [6]

Alongside load-sensing systems, electro-hydraulic systems have been introduced in the field. Their purpose is to boost hydraulic efficiency by removing the need of throttling valves and allowing energy recovery during operation. In these systems, each actuator is driven by its own adjustable electric motor and a double-acting hydraulic pump. This ensures that each actuator receives exactly the required flow, eliminating valve-related losses. The energy recovery is implemented with reversible acting pumps and electric motors. For example, when lowering a load, the energy is generated using motors and pumps in reverse converting hydraulic energy back into electricity to recharging batteries. [7]

The described electro-hydraulic systems typically run with a combustion engine during operation. However, Ponsse has developed an almost fully electric forwarder known as the EV1. It uses conventional electro-hydraulics but to run the system it uses energy stored in batteries. The system can operate for approximately 30-45 minutes using battery power alone. Once the battery reaches 40% capacity the combustion engine starts to recharge it. This way the machine operation is made part-time fully electric. Mergl V. et al. discussed this and other hybrid systems in their paper. [8]

Electrification has the potential to reduce the environmental impact of forestry machinery and improve its efficiency. Diesel-powered machines are major source of pollutant emissions globally. European regulations set strict emission limits for nonroad mobile machinery under Stage V standards [9]. The development of forestry machinery must meet the standards which is why alternative solutions to traditional diesel engines must be researched. However, the transition to electric machinery involves high costs as developing and integrating new systems is expensive. Also, current battery technologies do not provide sufficient energy density for prolonged operations. Despite these challenges, new hybrid solutions, like Ponsse EV1, show a promising direction of development in the field. [10]

2.2 Modelling approaches

When developing a forestry machine through modelling, it is important to first understand the different modelling options available for such systems. Exploring these options helps explain why a particular approach was chosen and reveals what other possibilities exist. In addition to multibody modelling, a forestry machine can be analysed using finite element analysis (FEA) to evaluate the structural strength and deformations. A detailed hydraulic system model can be created to simulate pressures and flow rates in the system. Longitudinal vehicle modelling can be used to study pure driving performance.

Multibody simulation is a method to study the motion of bodies that are connected to each other by different joints and constraints. The motion calculations are based on the basic laws of motion, Newton's and Euler's equations. These describe the translational and rotational dynamics of each body in the system. The equations are expanded to include all connected bodies, which results in a set of coupled differential equations. These equations require the inertia and geometric properties of each body as well as initial conditions, which are set before simulation. The differential equations are integrated numerically using time-stepping methods. At each time-step, the solver calculates new positions and orientations of the bodies, creating a complete motion history of the system. Multibody dynamics is discussed in more detail in the book Fundamentals of Multibody Dynamics. [11]

While multibody simulation focuses on system motion, structural and durability studies can be performed using finite element analysis. FEA is used to determine whether components can withstand operational stresses, enabling design optimization. The method works by dividing bodies into smaller elements, each assigned material properties and constraints. Calculations then show how the structure behaves under various loads and conditions. [12] FEA can be applied to single bodies, for example to a wheel, to study deformations in a rubber tire during operation. It can also be used to interconnected bodies, like a crane structure, to determine how much load the structure can withstand before failing.

Hydraulics consumes a significant share of the total energy consumption in mobile working machinery. The portion of hydraulics is approximately 40% but the amount of usable energy is much lower due to losses, such as valve losses. To identify, analyse and reduce these losses, it is beneficial to model hydraulic systems. Simulations help compare energy efficiency, performance and control of different hydraulic topologies. Detailed models can be built from individual components, allowing for the study of each component's impact on the overall system performance.

Forestry machine driving performance can be studied using longitudinal vehicle models. These models are constrained to two-dimensional analysis and focus on the characteristics of straight-line motion. Typical quantities of interest include the vehicle's acceleration and speed. It is also possible to study how quickly the vehicle stops when braking. Typically, forestry machines operate on challenging terrain, which is why longitudinal models do not capture the complex behaviour between the machine and uneven ground. However, if the primary objective is to evaluate raw driving performance, then these methods are highly suitable.

3 State-of-the-Art

Modelling approaches have been applied in the field of forestry machinery before. To increase the efficient operation of forwarders, the use of automation in routine motions should be researched. At Umeå University, a log loader crane alone was built and used as a testing platform for automation and control studies in the forestry industry. Morales D. and La Hera P. studied the dynamics of a forwarder crane. They developed both a kinematic and a dynamic model of crane motion. The purpose of this was to enable future research on automating routine crane tasks and to improve crane operation efficiency. As a result, they calculated the torques required to perform the desired simulated crane motion. This was validated using experimental data from the built crane, showing that the simulated torque closely matched the measured torque. The model is well-suited for studying the dynamics of log loader cranes. [13]

A forwarder can also be equipped with tracks instead of wheels. Ramachandran P. studied a tracked forwarder in multibody simulation in his master's thesis. Forest machines need to cause as little environmental damage as possible. Their operation also needs to become more sustainable without compromising productivity. To help achieve these goals, methods such as reducing soil damage and minimizing machine vibrations should be used. The study involved replacing the wheels with a track unit in a conventional wheeled forwarder. The model did not include the crane. The aim was to evaluate the vehicle's driving performance and behavior on a simple straight track with bumps. Results showed that the simulated vehicle experienced significant vertical displacement and roll motion. The model can be used in follow-up studies, but it would require verification with a real machine to ensure credible results. [14]

In 2014, Baharudin M. et al. developed a multibody application for a tree harvester, which can be applied to the real-time simulation of a hydraulically driven harvester. The model was tested in a real-time simulation setup including a visual display, a motion platform, and an I/O interface. A simplified version of the hydraulic system was also included in the model. During the tests, they collected pressure, flow rate, and collision force data from the simulation. The results were reported to be reasonable. [15]

Autonomous operation is a major research topic aimed at optimizing forest machines' work cycles. In February 2024, a self-driving forwarder was successfully tested in real forestry operations, demonstrating its capability to perform effectively. The task was for the machine to autonomously pick up and transport logs to the designated pick-up point, and the test was a success. [16]

A 4-wheeled small-scale autonomous forwarder has been built at Luleå University of Technology. It serves as a research platform for testing different technologies. Alongside the physical vehicle, a digital twin was also developed. The modelled digital twin includes features such as real-time multibody simulation, hydraulics, contacts and collisions, and co-simulation capabilities. The digital twin was able to operate in a scanned virtual environment. It was programmed to follow a prerecorded GPS track previously driven by the physical vehicle. The study validated the simulation of the digital twin using real machine data, and real-time simulation was also achieved. [17]

Autonomous unmanned forwarders are not yet available for the forestry industry. Existing ones have been built for research and development purposes. These autonomous forwarders utilize artificial intelligence mostly for navigation and accurate tree identification tasks. Ringdahl O. studied navigation and path planning of autonomous forwarders in one of his doctoral thesis papers. He describes methods for navigating challenging environments using different algorithms for path tracking and obstacle avoidance [18].

Isbak J. further discussed the role of AI in identifying tree species in forestry operations. Identification can be done using 3D LiDAR scans or 2D images. There were challenges in identifying trees, since the accuracy was about 80%, and in dense forests, identifying individual trees became difficult. [19] The navigation and tree identification tasks described above were part of the autonomous forwarder project mentioned earlier [16].

4 Simulation model

To obtain a functioning multibody model in MATLAB Simulink, one effective method is to first create an accurate 3D model of the forwarder in CAD software and then export this assembly into MATLAB. This allows the use of detailed 3D parts directly within Simulink. These imported parts can then be used to make a moving multibody model of the forwarder, enabling the addition of controllable interactions between individual components.

4.1 SolidWorks 3D Model

The 3D model was implemented using SolidWorks. It is a general representation of a forestry forwarder. The size of the forwarder corresponds to a real-life 8-wheeled machine, although the individual dimensions are generalized and not based on a specific design. Figure 2 illustrates the modelled forwarder.

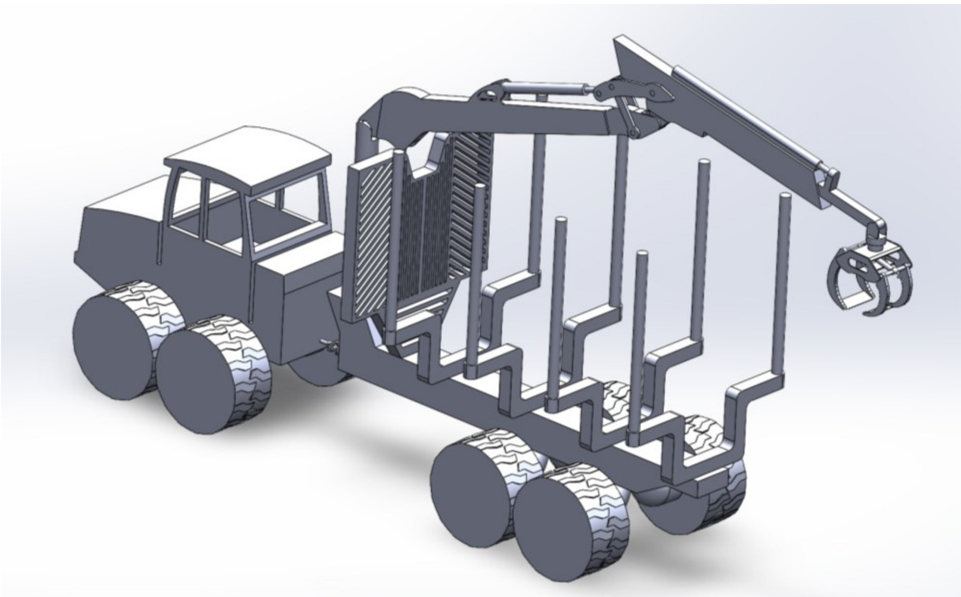

Figure 2. 3D model of a generic forestry forwarder.

The model includes many simplifications. There is no suspension, the frame structure is simple rigid body without edge fillets or beam constructions and wheels are modelled as a single part without separate components for the rim, tire, or hub. The articulated steering is made possible using a universal joint. The grapple rotator is not modelled to work in detail. The focus is on creating a working multibody simulation, so these details have been intentionally neglected.

While building the 3D assembly, extra attention was paid to define the relationships between parts through mating in SolidWorks. The Simscape Multibody Link add-in was used to export the assembly into an XML format compatible with MATLAB [20]. This export included information about relationships between bodies. Ensuring that all parts were properly constrained in SolidWorks directly affected how well the multibody model worked in MATLAB, especially when it comes to joint types and allowed movements.

A specific example was the modelling of cylinders and pistons. To represent realistic behaviour in the simulation the piston's rotational degree of freedom had to be restricted. If this constraint wasn't defined in SolidWorks, the exported model ended up with a cylindrical joint, which allows both translational and rotational movement. The wanted joint type was a prismatic joint which allows only linear motion along the piston's axis.

4.2 MATLAB Multibody Model

The multibody model was created from the SolidWorks assembly using MATLAB, which reads the exported XML file and converts assembly into a Simscape multibody model. This model includes Simscape joints between the parts and defines each part's position relative to others. To enable joint control and to define interactions such as collisions, the imported model required manual modifications.

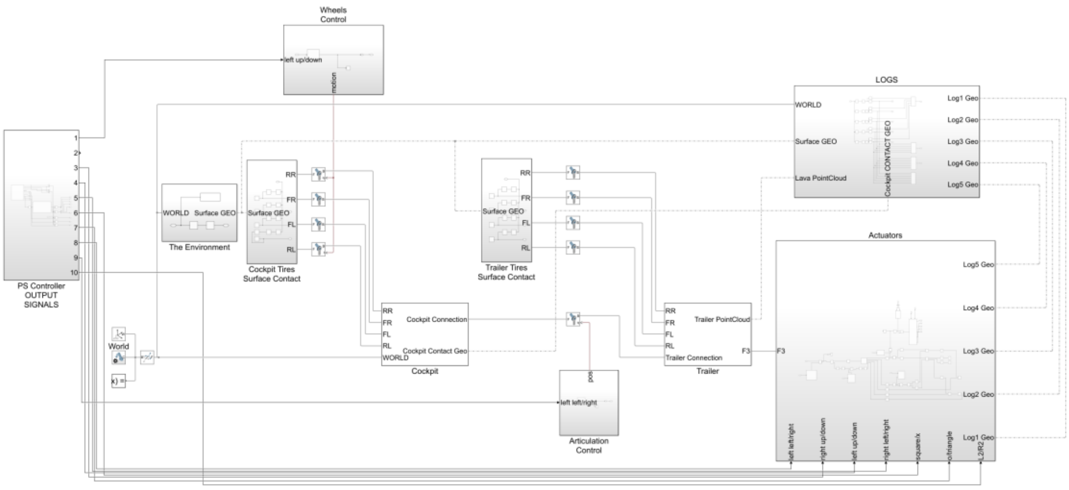

Figure 3. Controllable MATLAB multibody model of forestry forwarder.

Figure 3 presents Figure 3. Controllable MATLAB multibody model of forestry forwarder.the forwarder system used in simulation model. The model includes multibody parts, controllable joints, an environment for the forwarder to operate in and contacts between parts. The multibody model is created to simulate forwarder movement including log handling. During operation, actuating joints can be evaluated for example by analysing energy consumption. Appendix A presents figures of the key subsystems in the simulation model, and Appendix B includes the MATLAB initialization code.

4.2.1 Multibody parts

Every part of the model has been created individually. Parts have been modelled and assembled in SolidWorks and further exported into MATLAB with geometry-, inertia- and graphic properties. These properties can also be changed for each part in multibody model. MATLAB multibody model of the forwarder is shown in Figure 4. Name of the parts are also included, and these names are used later when speaking of the specific part.

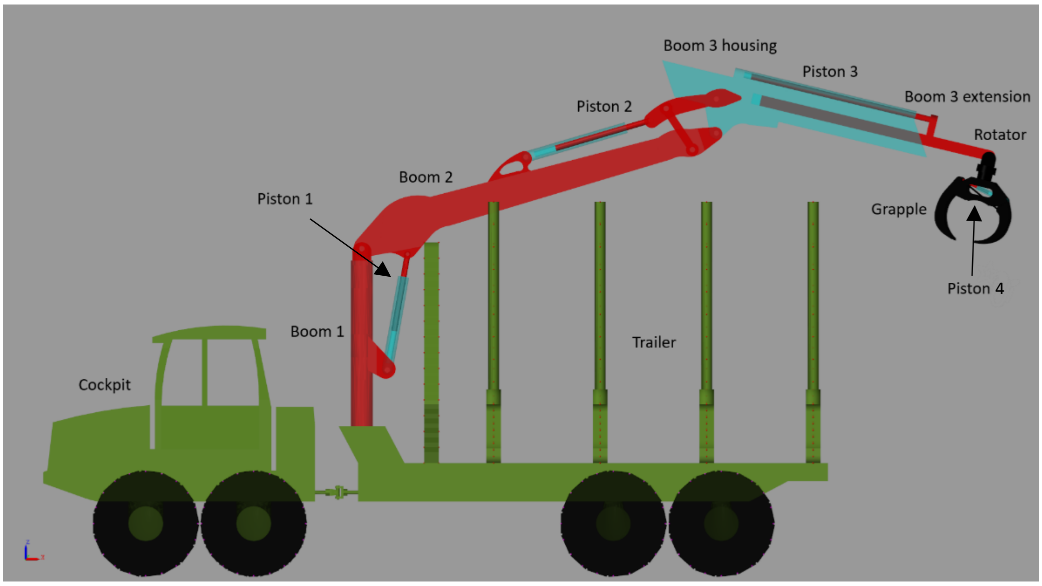

Figure 4. Multibody model of the forwarder.

The material used in the model is steel to keep the model as simple and generic as possible. Changing the material changes the inertia properties of the parts thus affecting the movements of the forwarder. Too low inertia on main bodies of the forwarder causes unwanted movement due swinging of the heavy crane. Mass properties are shown in the Table 1 below.

Table 1. Mass of forwarder bodies.

Body	Mass (kg)
Cockpit	6000
Trailer	6500
8 Wheels	2800
Boom 1	900
Boom 2	600
Boom 3 housing	250
Boom 4 extension	150
Grapple	300

Exported SolidWorks parts were converted to STEP format for use in MATLAB. The Simscape *File Solid* -block creates bodies in Simscape and utilizes the STEP format when importing parts from SolidWorks. This block imports the true visual shape of the body into the simulation but uses a convex hull representation for physical calculations in the multibody environment. The convex hull geometry is just an approximation of the true geometry seen in Figure 5 which causes problems later if used in contacts between bodies. This is due to the imperfections of the approximated geometry. [21]

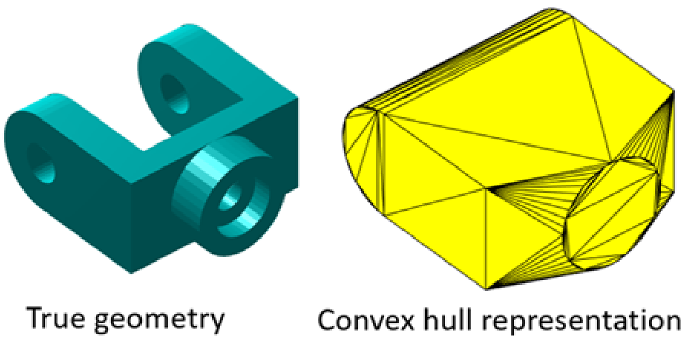

Figure 5. Differences between the true geometry and the convex hull representation of the body. [11]

4.2.2 Joint control

There are generally two ways to control a forwarder crane: either by controlling each piston individually or by controlling only the grapple tip position, where the system automatically calculates and adjusts the necessary piston movements to reach the desired tip location. Controlling each piston individually requires a skilled operator but allows for lower system complexity (fewer sensors, no advanced control algorithms) and offers full control and flexibility. Boom tip control, on the other hand, is easier to operate than the conventional system, and the movements are optimized, which leads to reduced energy consumption. Manner J. and Englund M. compared these control systems to each other and found out that boom tip control allows best productiveness [22].

Traditional individual piston control has been chosen for this study primarily for research purposes. This approach allows full manual control of each arm making it easier to analyse the behaviour and the performance of each joint individually. Implementing individual control is also simpler than boom tip control as it makes the system constraints easier to understand. Boom tip control would have required inverse kinematics and an additional control system which are additional features which can be considered for future research.

To implement this individual piston control in the simulation, a general PC controller was used to operate both the vehicle and the crane actuators. Since the controller doesn't have enough joysticks and buttons to handle all movements simultaneously, a logical state chart is used to manage the forwarder's control states.

There are two states: *drive* and *grapple* (Figure 6). The idea of these states is just to tell the model whether the controller controls the basic movement of the forwarder (driving) or the crane actuators. Pressing the start button switches between the two states.

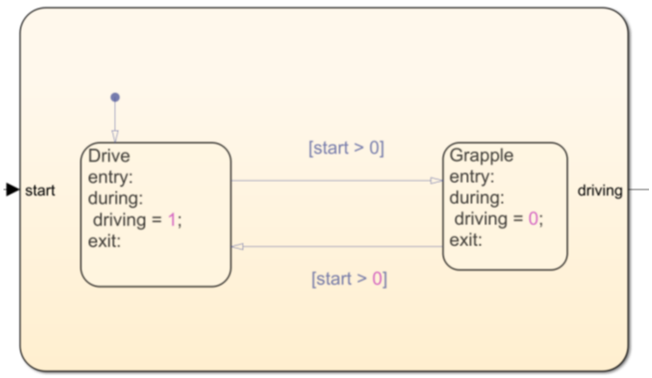

Figure 6. Control states of the forwarder.

PC controller signals from buttons always output 1 or 0 depending on whether the button is pressed or not. Joysticks output a continuous signal varying between 0 and 1 depending on how much the joystick is tilted. Joystick outputs 0 if it is untouched and 1 if the stick is tilted at its maximum.

Controlling of the pistons was made by using a *discrete-time integrator* -blocks (Figure 7). The block allows to add gain and limits to the controller's signal. Gain is used to control how quickly the piston position changes in response to the joystick movement. Limits are used to ensure that the piston movement stays within its physical limits like the piston stroke length. The piston cannot move physically more than it is allowed due to physical limitations.

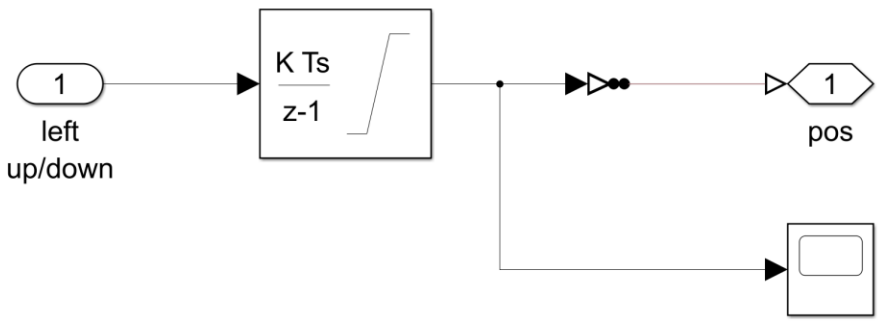

Figure 7. Piston 2 controller.

Piston 2 has a stroke length of 975mm. Figure 8 shows the control signal for the Boom 2 piston. The difference between maximum and minimum values corresponds to the 975mm stroke length.

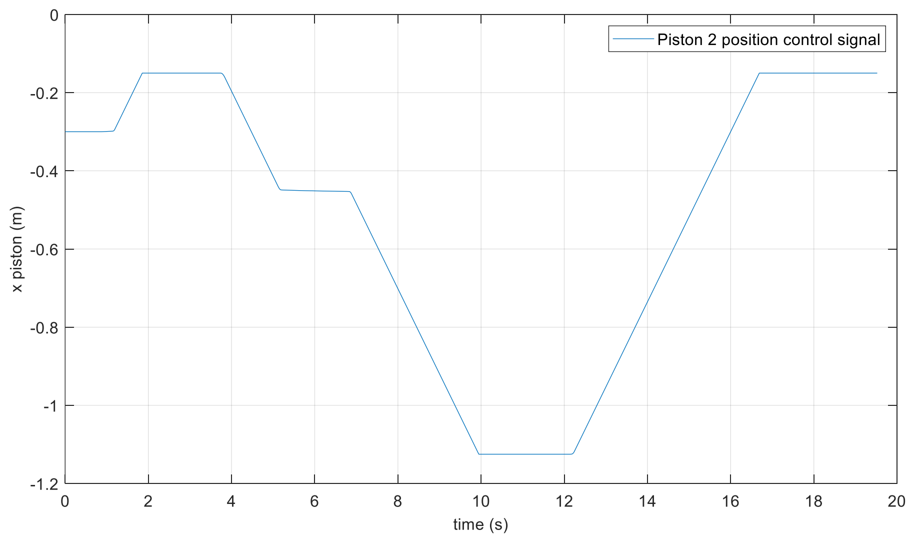

Figure 8. Piston 2 control signal.

The control of the actuators has been designed to be nearly ideal. The actuator movement is modelled as immediate without delays or spring stiffness. However, a minor value of damping was added to joints to improve simulation stability.

Realistic crane movement in simulation requires piston speeds adjusted to match real-life forwarder performance. In general, piston speeds can be calculated using the hydraulic flowrate and the geometry of the piston. Using the eq. 1, realistic piston velocities can be estimated.

$$v_{piston} = \frac{Q_{piston\_chamber}}{A_{piston}} \qquad \text{eq. 1}$$

v_{piston} is the piston speed (m/s), $Q_{piston\_chamber}$ is the flow rate into the piston chamber (m^3/s) and A_{piston} is the piston cross-section area (m^2). These parameters can be selected using real-life values from manufacturers' data resulting realistic piston speeds for the simulation. In Table 2 piston speeds have been calculated using real parameters for eq. 1. In addition, effective flow rate coefficient η_{flow} has been used. Hydraulic systems rarely operate at their maximum flow rates. The flow rate is often restricted by directional valves even if the pump can provide a higher flow rate. This coefficient is used to account for partial usage of the maximum flow rate. Figure 9 presents the validated piston speeds corresponding to the calculations in Table 2. It shows that piston velocities in the simulation were calibrated to match these calculated values, ensuring realistic actuator motion.

Table 2. Piston velocities for simulation [23].

Piston	Q_{max} $(\frac{l}{\min})$	η_{flow}	d (mm)	v_{max} $(\frac{m}{s})$
Piston 1	360	0.4	100	0.306
Piston 2	360	0.4	90	0.377
Piston 3	360	0.4	80	0.477
Piston 4	360	0.2	60	0.424

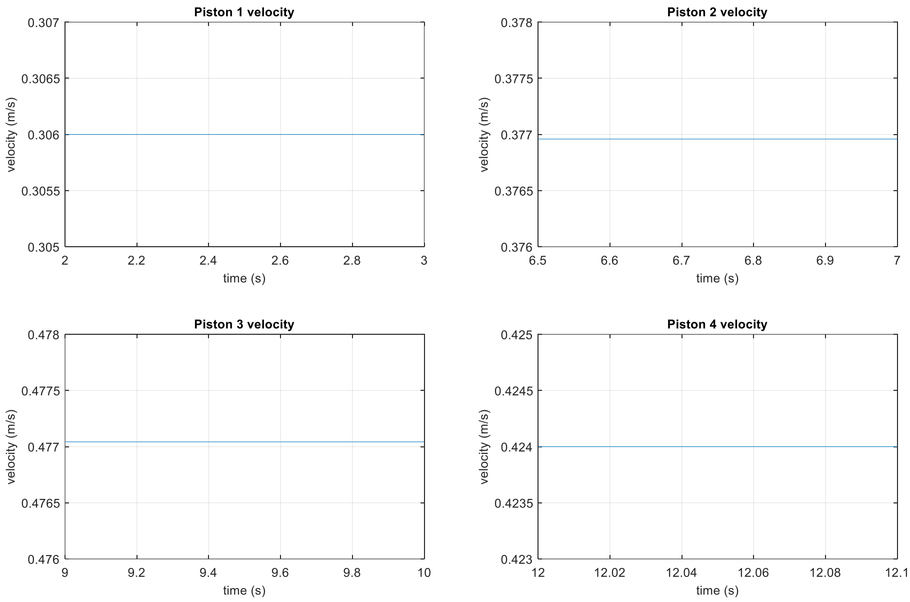

Figure 9. Adjusted simulation piston velocities.

Based on the study of forwarder crane loading speeds, it takes 6 to 16 seconds for lifting bunch of logs onto the trailer depending on the load size and the distance [24]. The piston velocities calculated for the simulation allows the crane movement that can meet the loading speed of real-life operation. Matching behaviour makes the model suitable for comparison.

Controlling the grapple needed also extra attention. The grapple is connected to the extension boom with a controllable revolute joint (Figure 10). This joint is controlled to allow tilting of the grapple around the joint axis. Normally the grapple hangs freely under gravity with only rotation controllability. In real-life operation this causes swaying of the grapple. Controlling the joint between the boom and the grapple, the swaying in the simulation model has been prevented. While this does not reflect real-life behaviour it simplifies log handling in the simulation by reducing unnecessary motion. Antisway systems have been studied but they are not commercially implemented [25].

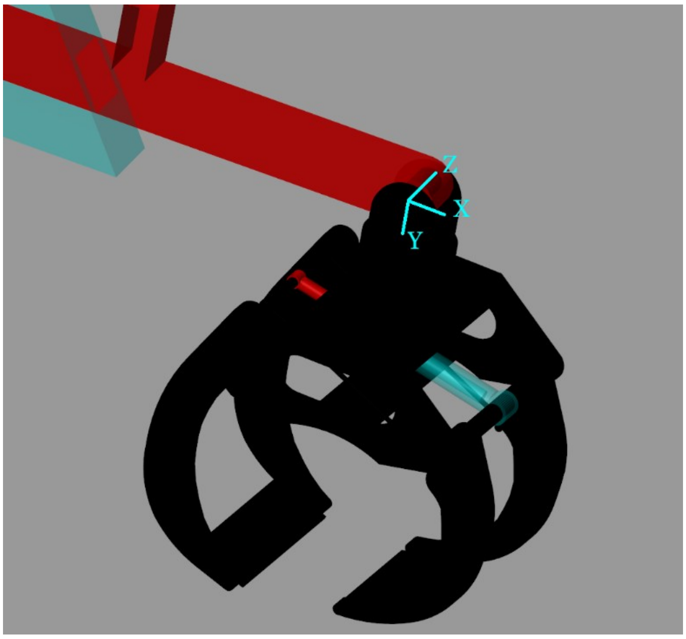

Figure 10. Controlled revolute joint between the extension boom and the grapple.

4.2.3 The environment

The environment was modelled using individual solid bodies, as shown in Figure 11. The position of each body was defined manually. In the simulation, there were no interaction with these objects. The forwarder was able to drive through them without collision. This was done intentionally just for the look to save computational resources for more essential parts of the simulation. Interaction with these bodies would not have added any value to the study.

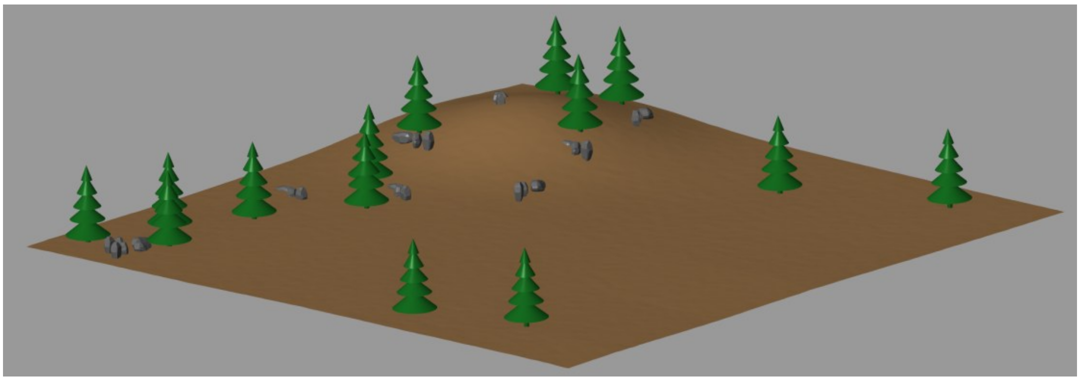

Figure 11. The simulation environment.

The terrain was modelled using a *grid surface* -block. It allowed detailed surface generation through a MATLAB script. The terrain has a little bumpiness to represent a more realistic surface. In the script the bumpiness was generated using a combination of Perlin noise [26], sine-based waves and random roughness for z-direction. Perlin noise was used to add realistic small-scale height variations, sine-based waves were for large-scale variations and random roughness was for tiny irregular bumps. The grid surface was used to simulate interactions with both the forwarder's wheels and the logs.

4.2.4 Contact modelling

Contacts between bodies have been made using *spatial contact force* -blocks. There were many restrictions when using this block because it works only between specific geometry pairs [27]. There were also limitations when working directly with solid blocks' geometries. Convex hull geometries are approximations of the true geometries, so using solid blocks' geometry outputs does not meet the needs of every contact. With very simple geometries like perfect spheres and cylinders, the convex hull geometry can be used.

Table 3. Used geometry pairs.

Contact between	**Base**	**Follower**
Ground – Wheel	Grid Surface	Point Cloud
Ground – Log	Grid Surface	Point Cloud
Forwarder - Log	Convex Hull of Solid	Convex Hull of Solid
Trailer – Log	Point Cloud	Convex Hull of Solid
Log – Log	Convex Hull of Solid	Convex Hull of Solid
Grapple – Log	Point	Convex Hull of Solid

Contacts between complex bodies were made using points and point clouds. These points were attached to the surfaces of real geometries to imitate the real shape of each body. Point clouds were defined in the MATLAB script and placed in the positions of the real geometries. This leaves the actual geometries visible only for visual purposes, as the contacts occur through point clouds instead of the actual geometry. The different contact geometry pairs used in the simulation are listed in Table 3.

In Figure 12, the point cloud of the trailer is shown. Points were placed along the trailer's true geometry for contacts with the logs piled onto the trailer. Figure 13 shows the points created for the grapple – log contacts. These points were created on the grabbing surface of the grapple, defining the area that directly interacted with the logs during operation.

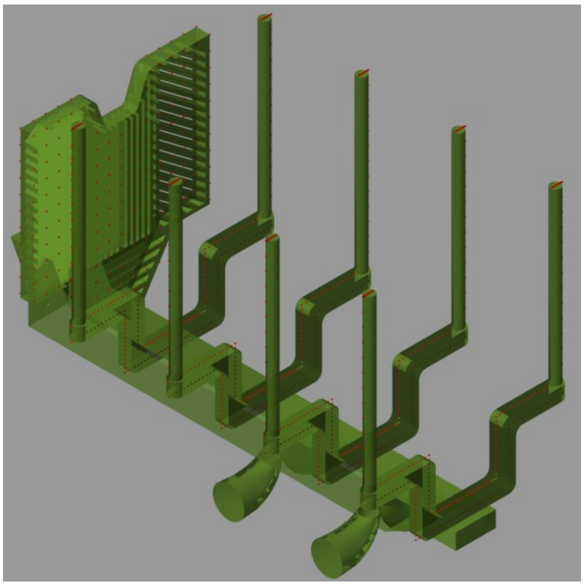

Figure 12. Point cloud created for the trailer geometry.

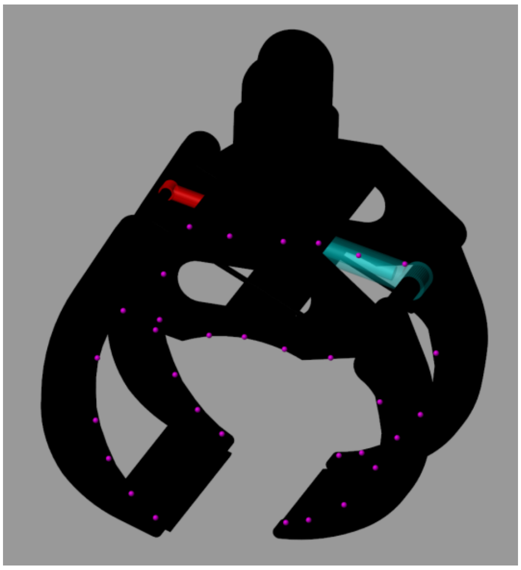

Figure 13. Points created for the grapple geometry.

Since the shape of the grapple was complex, the convex hull geometry could not be used. For this reason, points were created separately on the grapple's surface. Every point needed a spatial contact force block to enable contact with a log. Because there were a total of 5 logs and 12 points on one grapple body surface, the total number of spatial contact force blocks for that body was 60. Figure 14 shows a couple of these points with their corresponding contact setups. On the left, each log's geometry is connected to a spatial contact force block, which is then linked to a point, further connected to the solid body of the grapple.

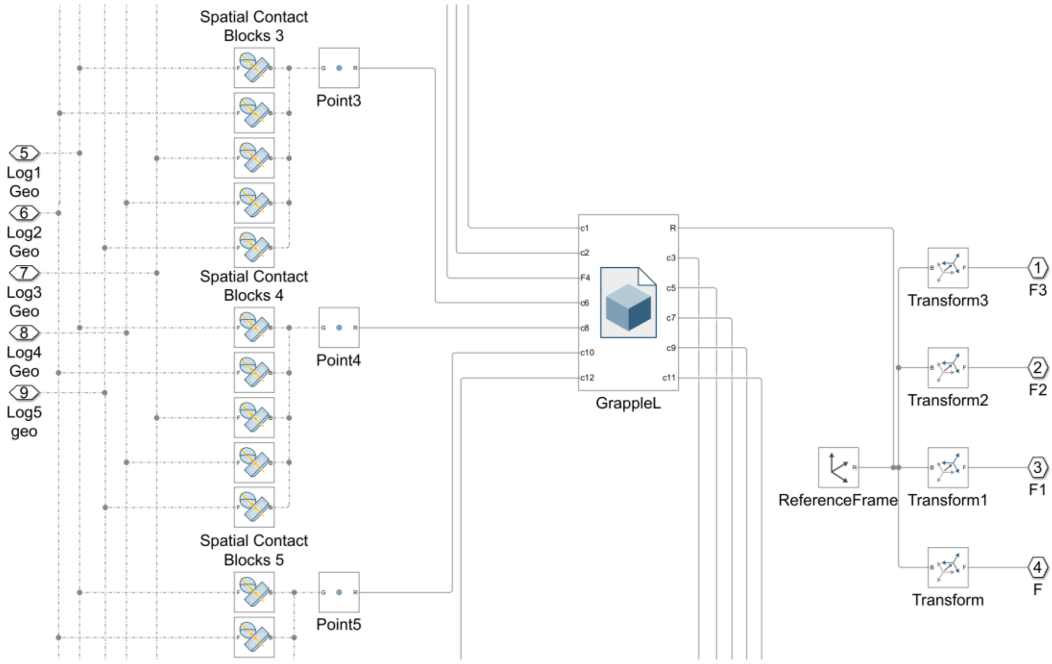

Figure 14. Grapple – log contact modelling.

Log contacts were made using two different geometries due to restrictions with supported geometry pairs. For example, the convex hull of the solid log geometry could not be used with the grid surface -block, which meant that that interaction between the terrain and the logs had to be implemented using an additional geometry. The spatial contact force block only works with point-type geometries when used with the grid surface. That is why point clouds were created on the surface of the logs to enable contact with the terrain. The number of points in the point cloud shown in Figure 15 is very low. Their purpose is simply to keep the logs resting on the surface. Adding more points would have slowed down the simulation due to increased number of simultaneous contact calculations.

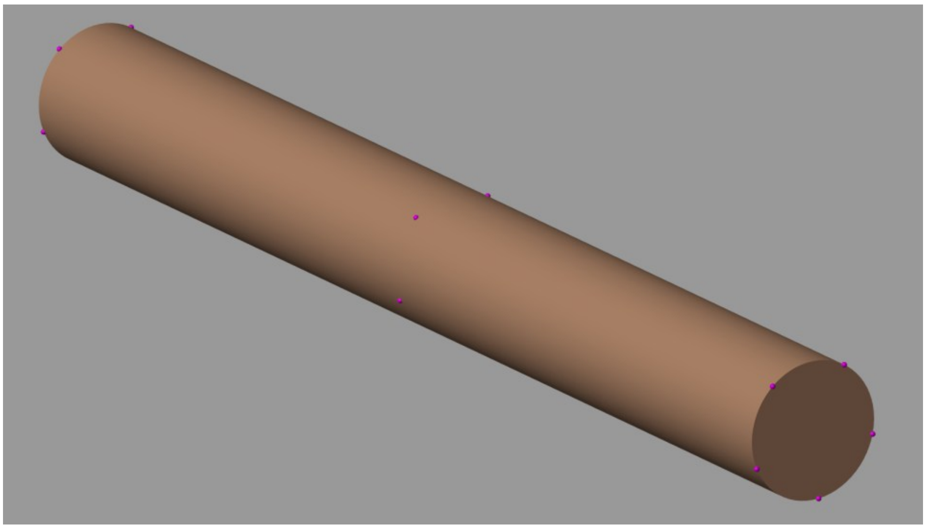

Figure 15. Point cloud on the surface of the log.

The contact modelling method created some uncertainty for the present study. For simplification, contacts were modelled without an additional custom contact model. The spatial contact force block allows elasticity through normal force stiffness and damping but these parameters were calibrated just to get the simulation working realistically. Increasing the stiffness caused the model to react very strongly and sensitively, resulting in unwanted motion and instability. Too low stiffness value let the bodies penetrate each other unrealistically. The damping affected how quickly the contact force settled. To make the contact behaviour even more accurate, a separate and more detailed contact model, such as a custom force implementation within the spatial contact force block, should have been implemented alongside the standard block.

4.3 Model constraints

A high number of simultaneous contacts limits the simulation speed. To achieve faster performance, the number of contact points had to be limited. Collisions between many bodies were not modelled at all, and those that were included used a low number of contact points. Limited and simplified contact modelling can be seen during the simulation: logs could not fully roll on the surface, and the forwarder was able to move through environmental objects. These are purely the result of the limited contact modelling.

Another constraint related to the simulation speed was the choice of MATLAB solver. The used solver was ode3, Bogacki-Shampine. It is a fixed-step solver for fast simulations that uses Bogacki-Shampine formula integration method to compute the state derivatives [28]. This solver was chosen because it allowed the simulation to run almost in real-time. However, its numerical accuracy is not too high, and it is less stable for fast dynamics than other solvers. This affected the precision of the simulation results but kept the simulation running fast.

Neglecting subsystems such as the hydraulic, powertrain, and electric systems had a direct impact on the simulation results. These systems were left out to reduce model complexity and to save computational resources in order to achieve faster simulation times. This limited the model's ability to represent the actual performance of a real forwarder. As a result, the quantities of interest could not be calculated directly from the subsystems and had to be estimated using data from the joints. Joints provided information about the acting forces and velocities during motion. With this data, the desired values were indirectly approximated. Neglecting these subsystems added uncertainty into the results.

More specific details also limited the realistic model behaviour. The modelled terrain was static. There was no deformation of the ground or the wheel during operation. All bodies were rigid, meaning no deformation during contacts either. The control of the forwarder and crane movements was immediate, with no delay or control error. Operation included no energy losses due to the lack of modelled systems that would normally allow them. The environment was kept very simple, without external conditions like wind or mud. Many details could have been considered but were neglected. In basic forwarder operation, the effects of these details were considered small which is why they were excluded from the model.

5 Testing cycles

Test courses were defined for the simulation so that energy usage data could be gathered from machine operations. Courses were designed to represent realistic forwarder tasks. Several test runs were conducted to ensure consistent data collection and to identify variations between manual operation cycles. Tests include two basic tasks: log handling with the crane and forwarder driving. Simulations were conducted on a computer equipped with an NVIDIA RTX 3060 Ti, an Intel Core i5-12600K and 32 GB of DDR4 RAM.

5.1 Loading and unloading operations

Picking up and loading logs involved gathering five logs from the forest onto the trailer. Typically, forwarders gather multiple logs in a single grab. In these tests, however, only one log is moved at a time. This simplification was made to keep the test as simple as possible. The difference in gathering has been taken into an account by adjusting the mass of the single log; one log in the simulation represents four real logs, keeping the model realistic and comparable. After the trailer was loaded, the forwarder moved to the end position next to the roadside for further transportation. Figure 16 presents the test course used for the simulation.

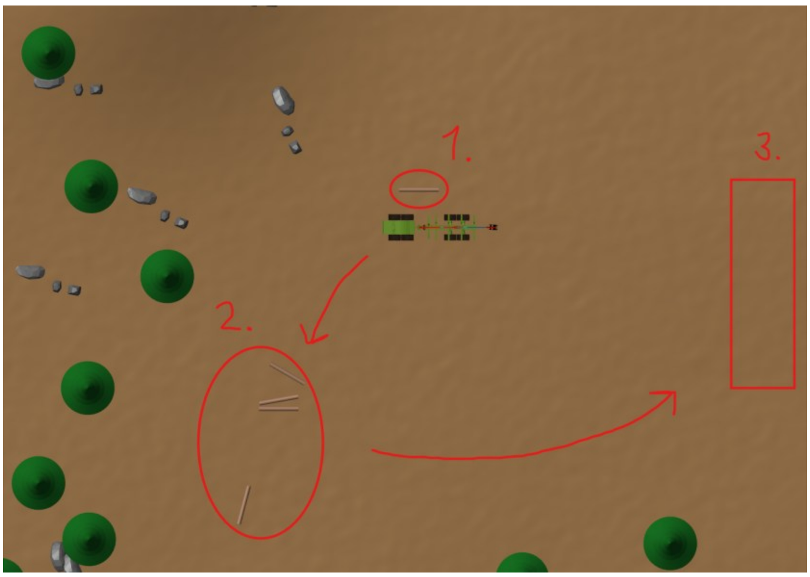

Figure 16. Test course for gathering and transporting logs. 1: The first log pick-up location. 2: Drive to the second position and load the remaining logs. 3: Transport the loaded logs to the final drop-off point for further transportation.

During this run, energy usage was calculated from joint data. For each controlled joint, except controlled wheels and articulated steering joint, the acting force or torque and displacement velocity were recorded. Using these variables the energy consumed by each joint was calculated using eq. 2 and eq. 3 where only positive power was considered (when actuators were doing work). Negative power occurs when the system does work back onto the actuators. For example, this can happen when a load pulls the piston down causing it to move without the piston itself generating force. In real systems, this energy could even be returned to the use through accumulators or energy regenerating pump/motor units. Summing the individual joint energies gives the total forwarder energy usage.

$$E = \int P(t)\, dt = \int (F(t) * v(t))\, dt \qquad \text{eq. 2}$$

$$E = \int P(t)\, dt = \int (T(t) * \omega(t))\, dt \qquad \text{eq. 3}$$

Log handling tasks also included unloading the trailer. This course simulated the unloading of logs from a forwarder trailer onto a log drop zone. The forwarder started stationary already loaded with 5 logs on its trailer. The task was to drive next to the drop zone and unload logs one by one. Logs were placed carefully on the ground side by side parallel to each other. Again, the test was repeated several times for consistency.

5.2 Time performance

During testing, also the simulation time was studied. MATLAB provides a simulation output object for every simulation run, which was used to gather the real time used by each simulation. This allowed comparison between simulation time and real time in simulations.

In addition to the two main test courses, a minor side study was conducted to investigate how the number of contact points affected simulation speed. In this test, the forwarder had to lift a one log onto the trailer. Tests were performed using 400 contact points on each log and using only 15. Real-time performance data was collected from both main test courses and this additional mini test.

6 Results

Testing cycles and time performance studies were conducted as described in chapter 5. The main focus was to evaluate the energy usage of the forwarder during loading and unloading operations. The data was collected including piston force signals, cycle times and detailed energy consumption measurements. Tables and graphs were created to visualize the collected data. Additionally, data about the effect of contact points on simulation time were collected.

6.1 Loading operation

The simulation worked as it should have. The forwarder was drivable and by controlling the crane it managed to load every log onto the trailer. Figure 17 illustrates the logs placed on the trailer. Some slight swaying of the whole forwarder occurred if the grapple was moved aggressively while picking up a log. Also, aggressive collisions between the log, ground and grapple caused additional swaying but this was considered a minor issue. Another issue in the simulation was the sliding of logs when a log was held in the grapple. Logs behaved as if no friction force was holding the log properly in the grip of the grapple.

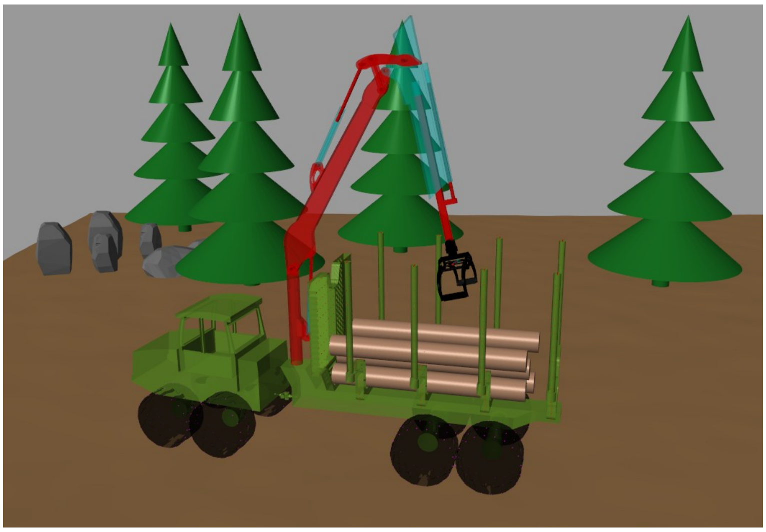

Figure 17. Final state of the forwarder after log loading.

Since the measurement conditions were very simplified for joints, with no real damping or resistance, their force signals had a lot of vibration. Also, when the crane stopped moving or changed direction the force signal resulted large spikes. These spikes are seen in Figure 18. Vibrations and unwanted large spikes were reduced using a low-pass filter on the force signals. Time constant used in the filter was 0.075. Notice the difference in the magnitude of the force signals. The unfiltered signal reaches magnitude of $10^7 N$ while the filtered signal stays around $10^5 N$. Using signal filtering the force signal acts more naturally during piston movement.

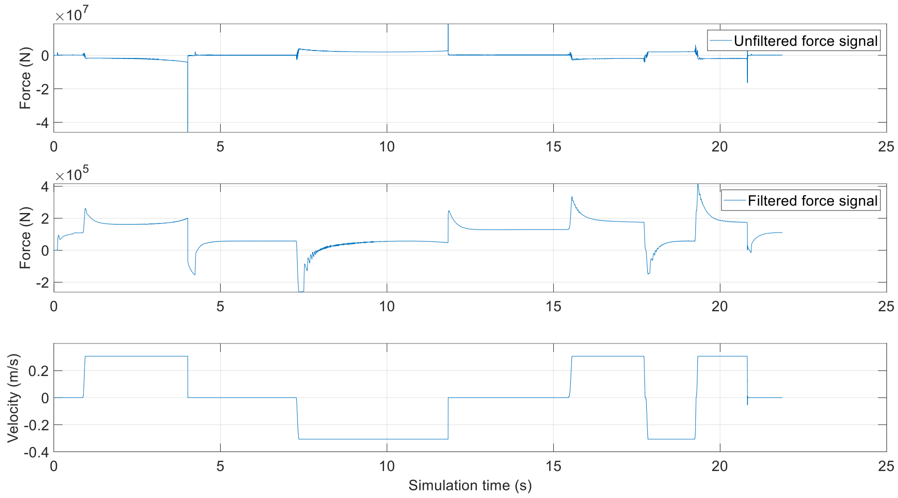

Figure 18. Unfiltered and filtered force signals during a piston movement.

Simulation did not run in real time. To get more accurate results and reduce unwanted random movement caused by aggressive collisions the simulation timestep had to be reduced. It is possible to increase simulation speed by increasing the fixed step size of the simulation, but this caused too much instability and random movement. Simulation was run with a fixed time step of 1.6ms. This resulted in an average simulation time of 88.02s while the actual real-world duration of the simulation was averaged of 126.15s (Table 4). This 38s difference was not really noticed during the simulation, but the difference is still significant especially considering the goal of real-time simulation.

Table 4. Forwarder energy usage during loading operation.

Test Cycle	Simulation time (s)	Real time (s)	Total Energy Used (MJ)
Run1	81.353	116.941	1.860
Run2	92.833	134.635	2.276
Run3	86.694	125.609	2.267
Run4	85.715	123.750	2.545
Run5	87.982	125.013	2.573
Run6	96.272	133.805	2.777
Run7	81.387	115.437	2.290
Run8	90.505	130.320	2.332
Run9	89.427	129.801	2.504
Run10	86.611	126.506	2.258
AVERAGE	**88.02**	**126.15**	**2.37**

Collected joint data resulted an average of 2.37MJ energy usage. In Table 4 all test runs are shown separately. There are clearly differences between runs while maximum energy usage was 2.777MJ and the minimum was 1.860MJ. Also, simulation time varied between 81.353s and 96.272s. Figure 19 presents test cycle run10 energy usage. Horizontal lines in the plot describes forwarder driving time when no crane actuators were used.

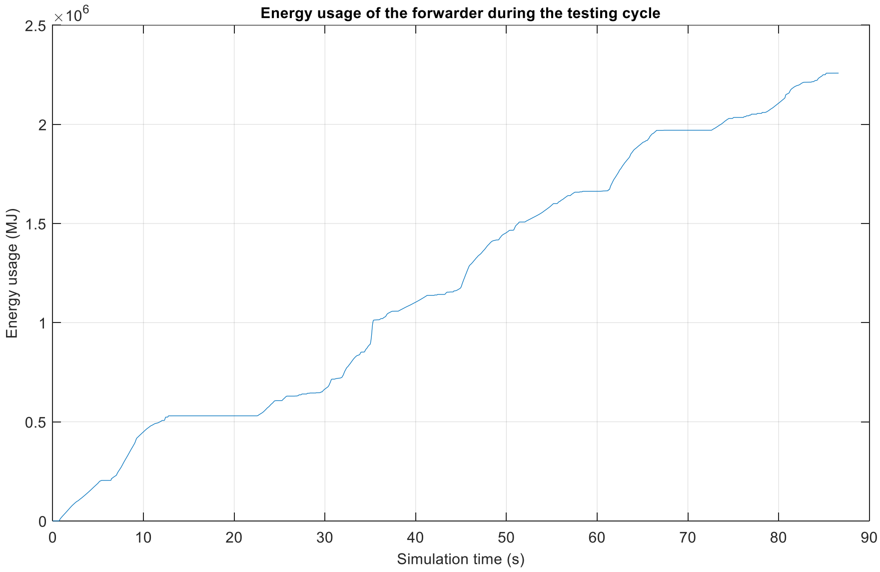

Figure 19. Energy usage of the forwarder during the test cycle.

Energy distribution data was also collected from the simulation. Energy usage of each joint was recorded for every run. Since joints acted ideally there was no energy losses presented. The total energy is entirely distributed directly to the ideal operation of individual joints without losses. Energy usage data is collected and presented in Table 5. A pie chart was created based on the average values in the Table 5 illustrating each joint's average energy consumption as a proportion of the total energy usage. The chart is shown in Figure 20.

Table 5. Energy distribution of the controlled joints during loading operation.

Test Cycle	Boom 1 rotator (MJ)	Piston 1 (MJ)	Piston 2 (MJ)	Piston 3 (MJ)	Piston 4 (MJ)	Rotator (MJ)
Run1	0.8123	0.4216	0.3971	0.005700	0.0862	0.1359
Run2	1.0561	0.4859	0.5460	0.009571	0.1004	0.07791
Run3	0.8881	0.5113	0.6716	0.01130	0.09458	0.08375
Run4	0.8740	0.4777	1.0524	0.005138	0.05605	0.08015
Run5	1.1233	0.4561	0.6783	0.004990	0.1967	0.1137
Run6	1.0139	0.4652	1.0746	0.002671	0.06349	0.1573
Run7	1.0234	0.4649	0.6590	0.005452	0.09279	0.04474
Run8	0.9268	0.5057	0.7252	0.009742	0.09949	0.06476
Run9	1.0457	0.4973	0.8054	0.006813	0.1045	0.05502
Run10	1.0816	0.4849	0.6129	0.003229	0.02888	0.04688
AVERAGE	**0.985**	**0.477**	**0.722**	**0.00646**	**0.0923**	**0.0860**

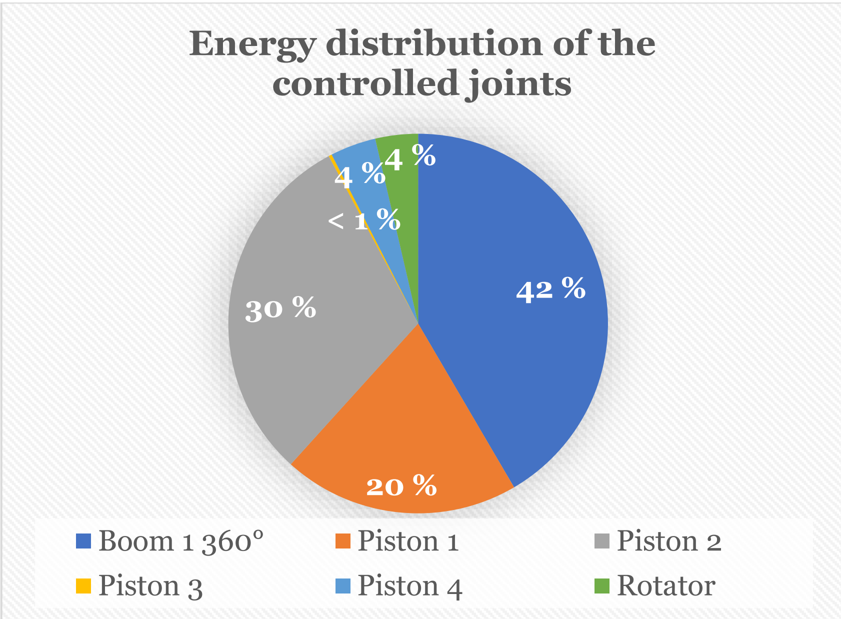

Figure 20. Joint energy usage during loading tasks.

6.2 Unloading operation

Unloading logs from the trailer worked as expected and the test course was finished successfully. Figure 21 presents the successful final state of unloaded logs. There were more challenges operating this test course than the loading course. Picking up the logs from the trailer one by one and placing them side by side parallel required precision and took more time.

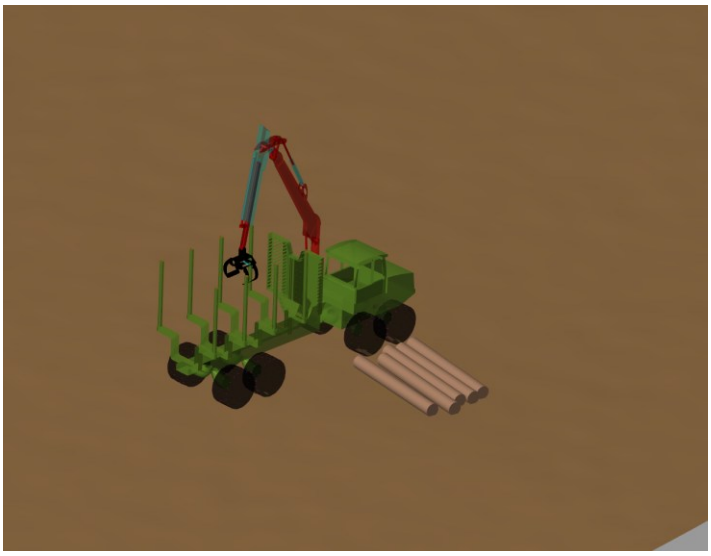

Figure 21. Final state of the forwarder after log unloading.

The same methods used to collect data for the loading operation were used to obtain data from the simulation during the unloading operation. This data is presented in Table 6 and Table 7. Also, a graph of energy distribution across the joints is presented in Figure 22. The average simulation time for the unloading tasks was 98.18s while the real time duration was 154.25s. The difference is 56s and this was already noticeable during the simulation. At the start when all the logs were on the trailer the simulation felt slow. The total energy usage was averaged 2.00MJ.

Table 6. Forwarder energy usage during unloading operation.

Test Cycle	Simulation time (s)	Real time (s)	Total Energy Used (MJ)
Run1	92.193	145.895	1.614
Run2	103.506	164.532	2.265
Run3	105.003	168.563	2.014
Run4	99.003	155.184	2.092
Run5	93.192	145.118	1.940
Run6	97.803	151.332	2.142
Run7	95.007	148.129	1.824
Run8	104.418	162.704	2.294
Run9	91.433	144.195	1.741
Run10	100.272	156.880	2.051
AVERAGE	**98.18**	**154.25**	**2.00**

Table 7. Energy distribution of the controlled joints during unloading operation.

Test Cycle	Boom 1 rotator (MJ)	Piston 1 (MJ)	Piston 2 (MJ)	Piston 3 (MJ)	Piston 4 (MJ)	Rotator (MJ)
Run1	0.5430	0.4476	0.2749	0.001774	0.2557	0.09162
Run2	0.5622	0.6977	0.6467	0.01158	0.2264	0.1199
Run3	0.5846	0.5756	0.4690	0.01224	0.2382	0.1359
Run4	0.8149	0.6425	0.3929	0.005976	0.2258	0.01120
Run5	0.6560	0.5468	0.4940	0.006681	0.1674	0.06903
Run6	0.6935	0.6132	0.5144	0.004194	0.2024	0.1142
Run7	0.4813	0.5049	0.3560	0.005678	0.2442	0.2319
Run8	0.7442	0.6682	0.4089	0.007812	0.2321	0.2324
Run9	0.5277	0.4399	0.2847	0.002351	0.2034	0.2842
Run10	0.6031	0.5778	0.4463	0.006128	0.2139	0.2033
AVERAGE	**0.621**	**0.571**	**0.429**	**0.00644**	**0.221**	**0.149**

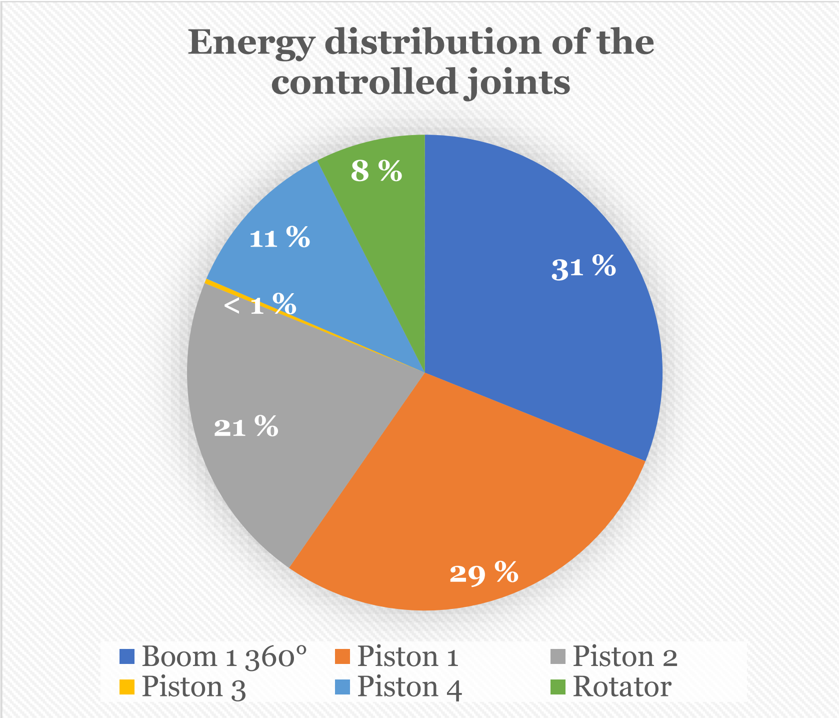

Figure 22. Joint energy usage during unloading tasks.

6.3 Effect of contact points on the simulation

During the test courses it became clear that the simulation speed slowed down when there were many simultaneous movements and collisions. Especially during the unloading test where the simulation started slower compared to the loading test. This is related to the contact modelling in the simulation. Effect of the number of contact points was tested on a shorter test course where one log was loaded onto a trailer. The test was repeated 5 times.

The number of contact points affects simulation time significantly. High number of points (400) on a log made it behave more realistic as the log rolled smoothly on the surface. With this many points on each log, the real time for loading one log in the simulation averaged 22.84s. With fewer points (15) the behaviour of a log on the surface was bumpy but the real time for the same simulation task averaged 18.61s. Figure 23 shows the difference between the number of contact points on logs. Results of these tests are presented in Table 8.

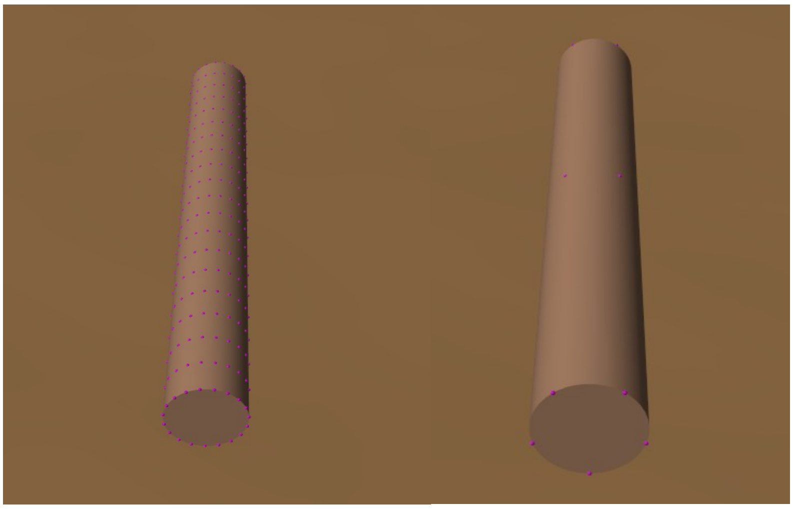

Figure 23. Logs with a different contact point number.

Table 8. Impact of contact points on real-time duration during log loading.

Contact Points	Simulation time (s)	Real time (s)	Contact points	Simulation time (s)	Real time (s)
15	13.830	19.518	**400**	12.649	21.325
15	13.852	18.757	**400**	13.737	23.098
15	13.624	18.499	**400**	12.811	21.816
15	13.539	18.268	**400**	14.174	24.105
15	13.049	18.000	**400**	13.832	23.878
AVERAGE	**13.58**	**18.61**		**13.44**	**22.84**

7 Discussion

The collected results showed consistency across multiple runs. Between the runs there were no significant differences in energy usage and the small variations can be simply explained by manual control. The test runs were also very close to each other in terms of time duration, which helped produce similar energy usage results. Repeated tests confirmed that the measured energy usage values were reliable across multiple runs.

The simulated energy usage is compared to real-life scenarios estimated in two different approaches. First, a general 8-wheeled forwarder has a power output of 190kW [29]. Assuming it operates at full power for 93s (the average simulation time calculated from both test cycles), the total energy consumption would be **17.67MJ**.

Second, Eliasson L. et al. studied fuel consumption in logging operations in Sweden. They found that the average fuel consumption of a large forwarder during basic operation was 13.7l/h. [30] The energy usage can be estimated using the energy density of diesel which is approximately 10700Wh/l [31]. In eq. 4 energy usage of a forwarder over 93 s has been estimated.

$$13.7\frac{l}{h} * 10700\frac{Wh}{l} * \frac{93}{3600}h * 3600\frac{J}{Wh} \approx \mathbf{13.63}\boldsymbol{MJ} \qquad \text{eq. 4}$$

In comparison, the loading operation simulation resulted in an average energy usage of 2.37MJ. This is ideal energy usage of hydraulic actuators so to make a fair comparison, following factors must be considered. The hydraulic efficiency of mobile working machines is typically around 30% [32], and the hydraulic system accounts for approximately 40% of the machine's total energy consumption. Taking these factors into account, the estimated total energy consumption of the simulated forwarder during loading tasks was **19.75MJ**.

During unloading operations, the simulated ideal energy usage was 2.00MJ. Applying the same efficiency and distribution assumptions, the estimated total energy consumption during unloading tasks was approximately **16.67MJ**.

These two real-life estimates fall within the same magnitude as the energy values estimated from the simulations. The maximum difference between the simulated and estimated real-world values is about 6MJ, while the minimum difference is just 1MJ. This close correspondence means that the simulated energy usage aligns well with real-world forwarder performance and that the simulation results are comparable with a real-world operation.

To make a more detailed energy usage comparison between the simulated forwarder and real-world operation, the operating points of a diesel engine could be considered. The efficiency of a diesel engine is highly dependent on its rotational speed and torque. Efficiency is typically highest at low rotational speeds with high torque. At higher speeds increased volumetric and mechanical losses tend to reduce efficiency. But since the model already includes other uncertainties from efficiency assumptions and model simplifications, there is no need to use this detailed method.

The joint that used the most energy was a revolute joint controlling Boom 1 rotation over vertical axis. An average it used 42% of total energy usage (Figure 20). That is a very large portion of the total energy usage, but it is very reasonable. In the loading operation test runs there was a lot of rotation movement over Boom 1. Five logs were separately loaded onto the trailer which required multiple back-and-forth repetitions for the Boom 1 rotational joint. Additionally, it is the first joint in the crane structure, so it carries the weight and inertia of all the parts that come after it. This explains its high energy usage.

Piston 1 and Piston 2 consumed the next highest amount of energy in the system. Piston 1 portion was 20% and Piston 2 30% (Figure 20). These were the pistons that handled up-and-down movements of the crane. That is why those portions had to be high since lifting heavy logs naturally required a lot of energy.

The actuator that used the least amount of energy was Piston 3. Its portion was less than one percent of total energy usage and significantly lower than the consumption of other actuators (Figure 20). This is simple explained by the fact that the actuator was not really used during the loading tasks. The forwarder was driven close enough to the logs so that the extension boom did not need to be used to reach further.

In the unloading operation simulations, the total energy usage averaged 2.00MJ. Even though the simulation time duration was about 10s longer than in the loading tasks, the energy usage was clearly lower. This can be explained by the fact that during unloading, the logs were initially positioned higher than in the loading operations. This reduces the vertical distance the logs needed to be lifted. During the loading operation when lifting the logs from the ground, the vertical lifting distance was greater which consumed more energy.

During unloading tasks, the simulation speed was slower than during loading operations. From the simulation time results (Table 4 and Table 6) it can be calculated how many real-time seconds are required for one simulation second. For loading operation one simulation second took 1.43 seconds. For unloading operation one simulation second took 1.57 seconds. Unloading tasks took 9.79% more real-time per simulation second compared to the loading tasks. In the unloading operation the logs start off next to each other and are in contact with both each other and the trailer. As the grapple picks up each log from the trailer there are many simultaneous collisions and small movements. These factors slow down the simulation.

A lot of simplifications were made in the model to pursue real-time simulation. During the test runs controlling the forwarder felt like it was running in real time. Still there was a noticeable difference between simulation time and real time. During the loading operations the difference was 38s while during the unloading operations the difference was even larger at 56s. Many factors affect the simulation speed. The biggest ones are the used simulation time step and contact modelling.

Reducing the number of contact points on the logs made the simulation run faster. Table 8 shows that the simulation speeds up significantly when the number of contact points on logs were reduced from 400 to 15. The simulation was 24% faster with reduced number of contact points.

System simplifications also affect the results. Leaving out the hydraulic system and powertrain from the model makes it challenging to calculate energy consumption accurately. Since energy cannot be measured directly from those subsystems it had to be estimated indirectly using joint data.

The forwarder was modelled successfully as a multibody system in MATLAB environment. The desired movements were made controllable with a PC controller allowing the machine to perform typical forwarder operations like driving and loading logs. This makes the model a good base for further research where more advanced systems and automation features can be built on top of the current setup.

8 Conclusions

The primary focus of this thesis was to determine the energy consumption of the simulated forwarder. To achieve that, a key objective was to build a functional and controllable multibody model of the forwarder. The goal was to find out whether the simulated energy consumption could be meaningfully compared to real-life values. The model created allows the user to control the forwarder within a simulation environment and perform simple logging tasks. The energy usage calculated from the simulation is comparable to real-life operation, though it reflects on idealized model. A more detailed simulation model would allow for more accurate results.

These objectives were achieved by first creating a working 3D model of the forwarder in CAD software. Then the assembly was imported into MATLAB, where the fully functional multibody system was created manually. Movable joints were made controllable using a general PC controller, and interactions between bodies were implemented through contact modelling. The forwarder was then tested under typical loading and unloading tasks, and simulation time and energy usage data were collected. With the forwarder model, the logging tasks were completed, and the energy usage was measured and compared to real-world data.

Energy usage from the simulation was comparable to that of a real-life forwarder operation. The results showed that the simulated energy usage was close in value and of the same magnitude as the real energy usage of the forwarder during a short operation period. Keeping in mind that simplifications in the simulation model made the results less accurate. However, a simplified model helps achieve faster simulation, enabling real-time-like drivability of the machine during operation.

Improvements to the model would include adding detailed hydraulic and powertrain systems. These subsystems would make the model more accurate in terms of results. Implementing a boom tip control with inverse kinematics would also increase the efficiency of system control, as no additional crane movements would affect the results. The issue with adding more systems and features is simulation speed. Every added detail would reduce simulation speed. Even now, the simulation is not running in real time. With additional features, the forwarder control should be preprogrammed instead of manually controlled during simulation, so that even with slow performance the desired results could be obtained.

An easier approach to develop this model, rather than using detailed hydraulic or powertrain systems, would be to create additional mathematical subsystems to describe real-life physical phenomena. For example, energy losses could be considered using mathematical models alongside the ideal model. Friction and hydraulic losses could be modelled this way to improve realism without adding heavy computational load.

The 3D simulation environment could be made more detailed by using third-party software like Unreal Engine. It provides high-quality graphics and tools for editing virtual environment. It also includes a built-in physics engine, which allows basic interactions such as collisions and contacts. However, for more advanced and accurate multibody dynamics, co-simulation can be used. In this case, MATLAB handles the physics and Unreal Engine provides the visualization. This way physics remains highly accurate but Unreal would make it look better.

This work shows that multibody modelling produces good results for studying energy consumption. The model can be continuously developed to improve its accuracy. It provides a solid foundation for further research, as different system solutions can be tested using this framework. The work is not limited to forwarders alone, the methods presented here can also be applied to wide range of mobile machine applications beyond forestry.

References

[1] Maa- ja Metsätalousministeriö, "Suomen metsävarat". Accessed: March 31, 2025. Available: https://mmm.fi/metsat/suomen-metsavarat.

[2] T. Nordfjell, D. Athanassiadis, and B. Talbot, "Fuel Consumption In Forwarders," International Journal of Forest Engineering, vol. 14, n. 2, 2003, doi: 10.1080/14942119.2003.10702474.

[3] Vaia, "Multi-body simulation". Accessed: April 4, 2025. Available: https://www.vaia.com/en-us/explanations/engineering/automotive-engineering/multi-body-simulation/?utm_source=chatgpt.com.

[4] H. Haavikko, K. Kärhä, A. Poikela, M. Korvenranta, and T. Palander, "Fuel Consumption, Greenhouse Gas Emissions, and Energy Efficiency of Wood-Harvesting Operations: A Case Study of Stora Enso in Finland," Croatian Journal of Forest Engineering, vol. 43, n. 1, 2022, doi: 10.5552/crojfe.2022.1101.

[5] T. Tyni, J. Lehto, V. Närvänen, J. Kajaste, O. Calonius, and P. Kuosmanen, "Energy efficient hydraulic system topologies for load-haul-dump machine," in: The 18th Scandinavian International Conference on Fluid Power, Tampere, Finland, May 30 - June 1, 2023.

[6] M. Heikkilä, "Energy Efficient Boom Actuation Using a Digital Hydraulic Power Management System," Dissertation, Dept. Faculty of Built Environment, Tampere University of Technology, Tampere, Finland, 2016. https://urn.fi/URN:ISBN:978-952-15-3763-9.

[7] D. Padovani, M. Rundo, P. Fresia, and G. Altare, "Energy analysis of a hybrid electro-hydraulic system for efficient mobile hydraulics," Journal of Physics: Conference Series, vol. 2648, 2023, doi: 10.1088/1742-6596/2648/1/012051.

[8] V. Mergl, L. Zeizinger, and M. Kománek, "Advancing sustainability in forestry machinery: Electro-Hybrid drives for greenhouse gas reduction and enhanced energy efficiency," Central European Forestry Journal, vol. 70, n. 1, 2024, doi: 10.2478/forj-2023-0024.

[9] DieselNet, "EU: Nonroad Engines". Accessed: May 24, 2025. Available: https://dieselnet.com/standards/eu/nonroad.php#s5.

[10] E. Scolaro, M. Beligoj, M. P. Estevez, L. Alberti, M. Renzi, and M. Mattetti, "Electrification of Agricultural Machinery: A Review," IEEE, vol. 9, 2021, doi: 10.1109/ACCESS.2021.3135037.

[11] F. M. L. Amirouche, *Fundamentals of Multibody Dynamics: theory and applications*, Boston, USA: Birkhäuser, 2006.

[12] S. Musa, A. V. Kulkarni, and V. K. Havanur, *Finite Element Analysis*, Virginia, USA: Mercury Learning and Information, 2014.

[13] P. La Hera and D. Morales, "Modeling dynamics of an electro-hydraulic servo," in: World Automation Congress, Puerto Vallarta, Mexico, June 24-28, 2012.

[14] P. Ramachandran, "Modelling and Dynamic Simulation of Tracked Forwarder in Adams ATV Module," Master's thesis, Dept. Industrial Engineering and Management, KTH, Stockholm, Sweden, 2015. Available: https://www.diva-portal.org/smash/get/diva2:905791/FULLTEXT01.pdf.

[15] M. Baharudin, A. Rouvinen, P. Korkealaakso, and A. Mikkola, "Real-time multibody application for tree harvester truck simulator," Sage Journals, vol. 228, n. 2, 2014, doi: 10.1177/1464419314521182.

[16] Forest Machine Magazine, "Autonomous Forwarder Project". Accessed: April 13, 2025. Available: https://forestmachinemagazine.com/autonomous-forwarder-project/.

[17] M. Lehto, A. Tikanmäki, T. Lindbäck, H. Lideskog, and M.Karlberg, "Modeling and verification of a full-scale forestry vehicle real-time multiphysics digital twin," in: 16th European-African Regional Conference of the ISTVS, Lublin, Poland, October 11-13, 2023.

[18] O. Ringdahl, "Automation in Forestry – Development of Unmanned Forwarders," Dissertation, Dept. Computing Science, Umeå University, Umeå, Sweden, 2011. Available: https://www.diva-portal.org/smash/get/diva2:412664/FULLTEXT02.

[19] J. Isbak, "AI in forestry: Can computers identify trees better than foresters?" Accessed: May 11, 2025. Available: https://www.forsler.com/newsletter/ai-in-forestry-can-computers-identify-trees-better-than-foresters.

[20] MATLAB Help Center, "Install the Simscape Multibody Link Plugin". Accessed: March 13, 2025. Available: https://se.mathworks.com/help/smlink/ug/installing-and-linking-simmechanics-link-software.html.

[21] MATLAB Help Center, "File Solid". Accessed: March 13, 2025. Available: https://se.mathworks.com/help/sm/ref/filesolid.html.

[22] J. Manner, A. Mörk, and M. Englund, "Comparing forwarder boom-control systems based on an automatically recorded follow-up dataset," Silva Fennica, vol. 53, n. 2, 2019, doi: 10.14214/sf.10161.

[23] Komatsu Forest Machines, "Komatsu 875". Accessed: April 26, 2025. Available: https://www.komatsuforest.com/forest-machines/our-forwarders/875.

[24] M. Piskunov, "Investigation of Structure of Technology Cycle Time of Hydraulic Manipulators in the Process of Loading Forwarders with Logs," Croatian Journal of Forest Engineering, vol. 42, n. 3, 2021, doi: 10.5552/crojfe.2021.863.

[25] J. Kalmari, H. Hyyti, and A. Visala, "Sway Estimation using Inertial Measurement Units for Cranes with a Rotating Tool," in: 8th IFAC Symposium on Intelligent Autonomous Vehicles, Gold Coast, Australia, 2013, doi: 10.3182/20130626-3-AU-2035.00050.

[26] Khan Academy, "Perlin noise". Accessed: May 4, 2025. Available: https://www.khanacademy.org/computing/computer-programming/programming-natural-simulations/programming-noise/a/perlin-noise.

[27] MATLAB Help Center, "Spatial Contact Force". Accessed: April 29, 2025. Available: https://se.mathworks.com/help/sm/ref/spatialcontactforce.html.

[28] MATLAB Help Center, "Solver". Accessed: April 29, 2025. Available: https://se.mathworks.com/help/simulink/gui/solver.html.

[29] KOMATSU Metsäkoneet, "Komatsu 875". Accessed: May 3, 2025. Available: https://www.komatsuforest.fi/komatsu-mets%C3%A4koneet/kuormatraktorit/875.

[30] L. Eliasson, K. Kärhä, and J. Arlinger, "Fuel consumption in logging operations in Sweden," International Journal of Forest Engineering, vol. 34, n. 3, 2023, doi: 10.1080/14942119.2023.2229707.

[31] Beloit, "Energy Density". Accessed: May 18, 2025. Available: https://chemistry.beloit.edu/edetc/SlideShow/slides/energy/density.html.

[32] L. Ge, L. Quan, X. Zhang, Z. Dong, and J. Yang, "Power Matching and Energy Efficiency Improvement of Hydraulic Excavator Driven with Speed and Displacement Variable Power Source," Chinese Journal of Mechanical Engineering, vol. 32, 2019, doi: 10.1186/s10033-019-0415-x.

A. Simulation Model and Subsystems

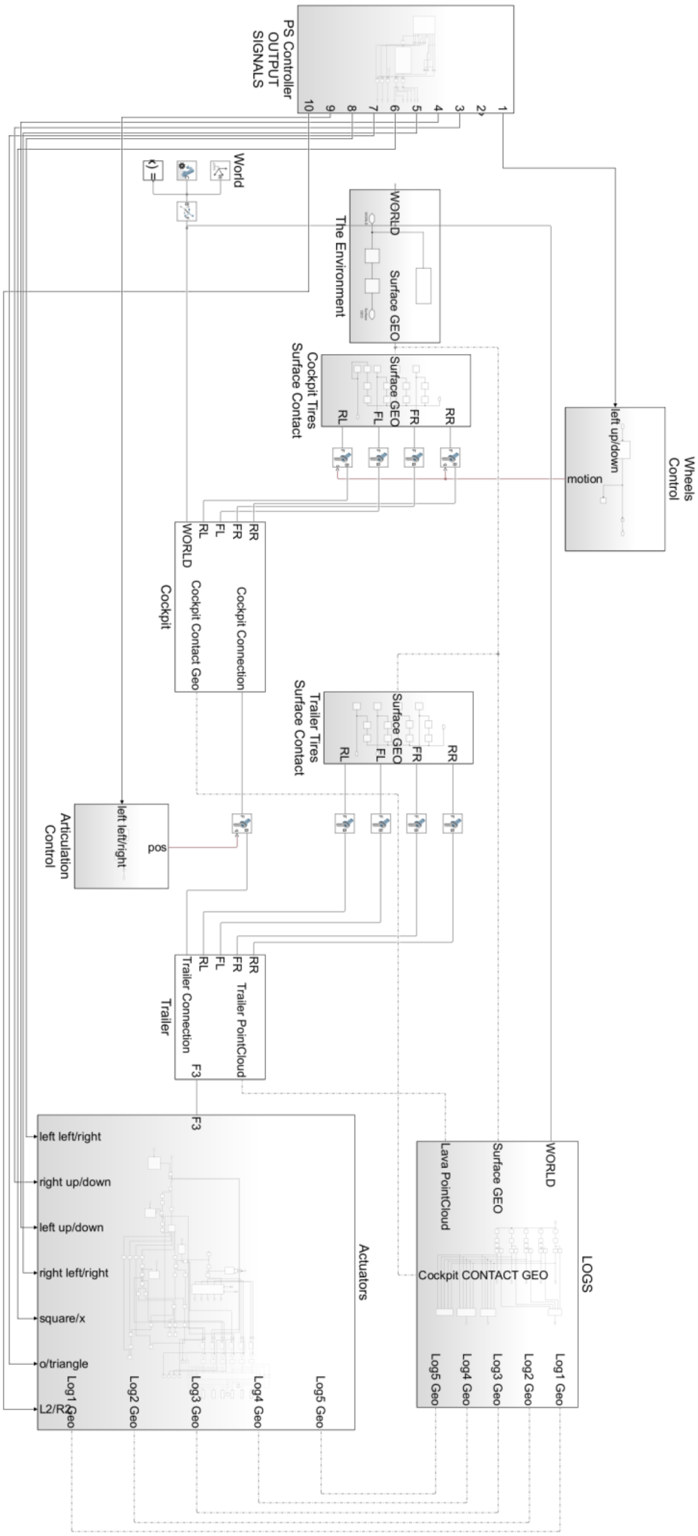

Figure A1. Top-level view of the forwarder simulation model in MATLAB Simulink.

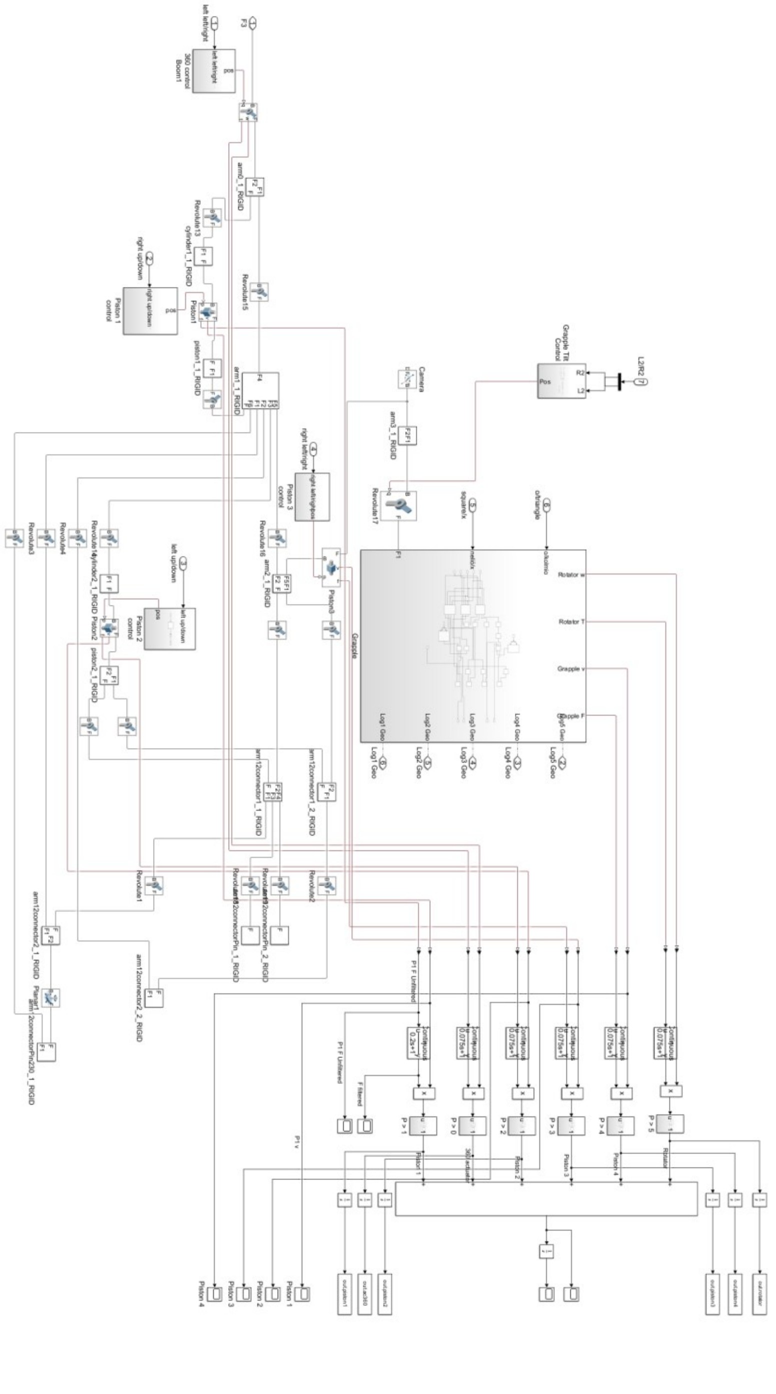

Figure A2. Actuators subsystem.

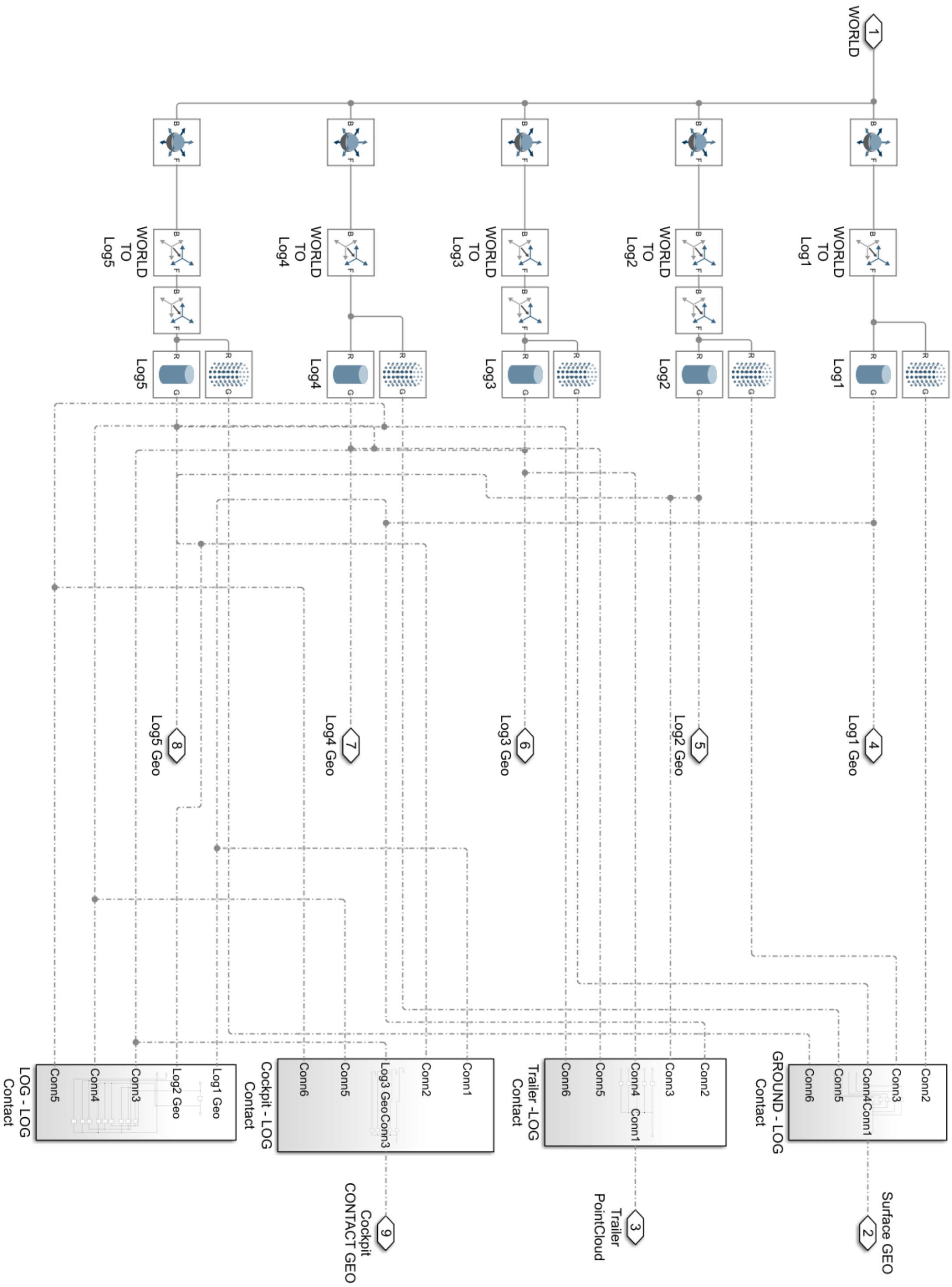

Figure A3. Logs subsystem.

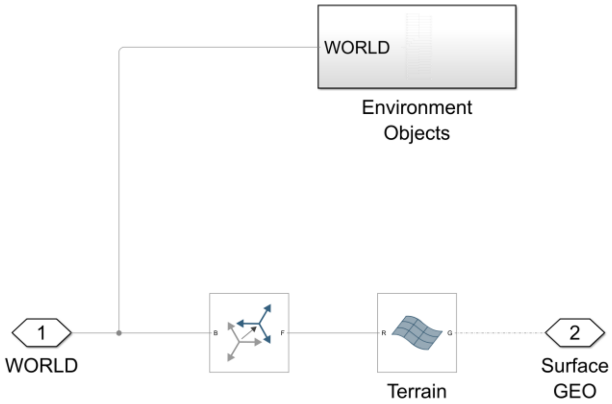

Figure A4. The environment subsystem.

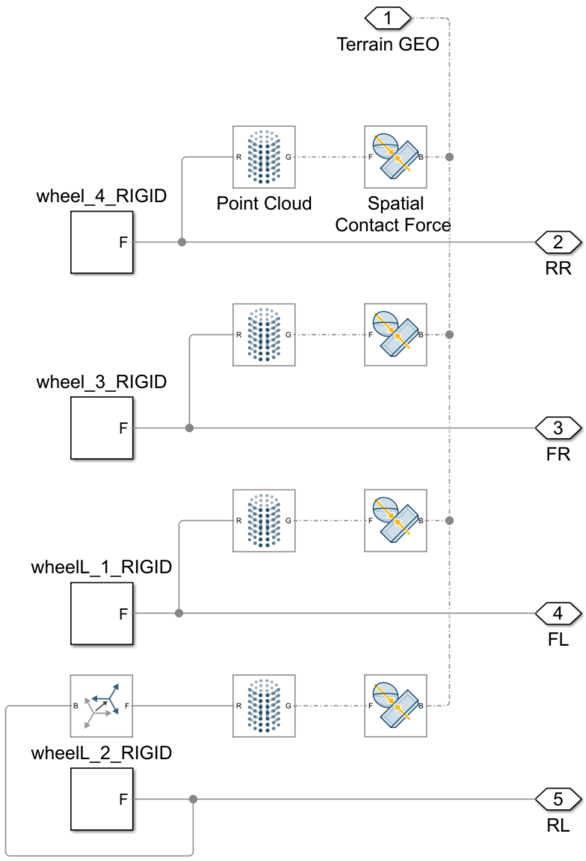

Figure A5. Cockpit – Wheels contacts.

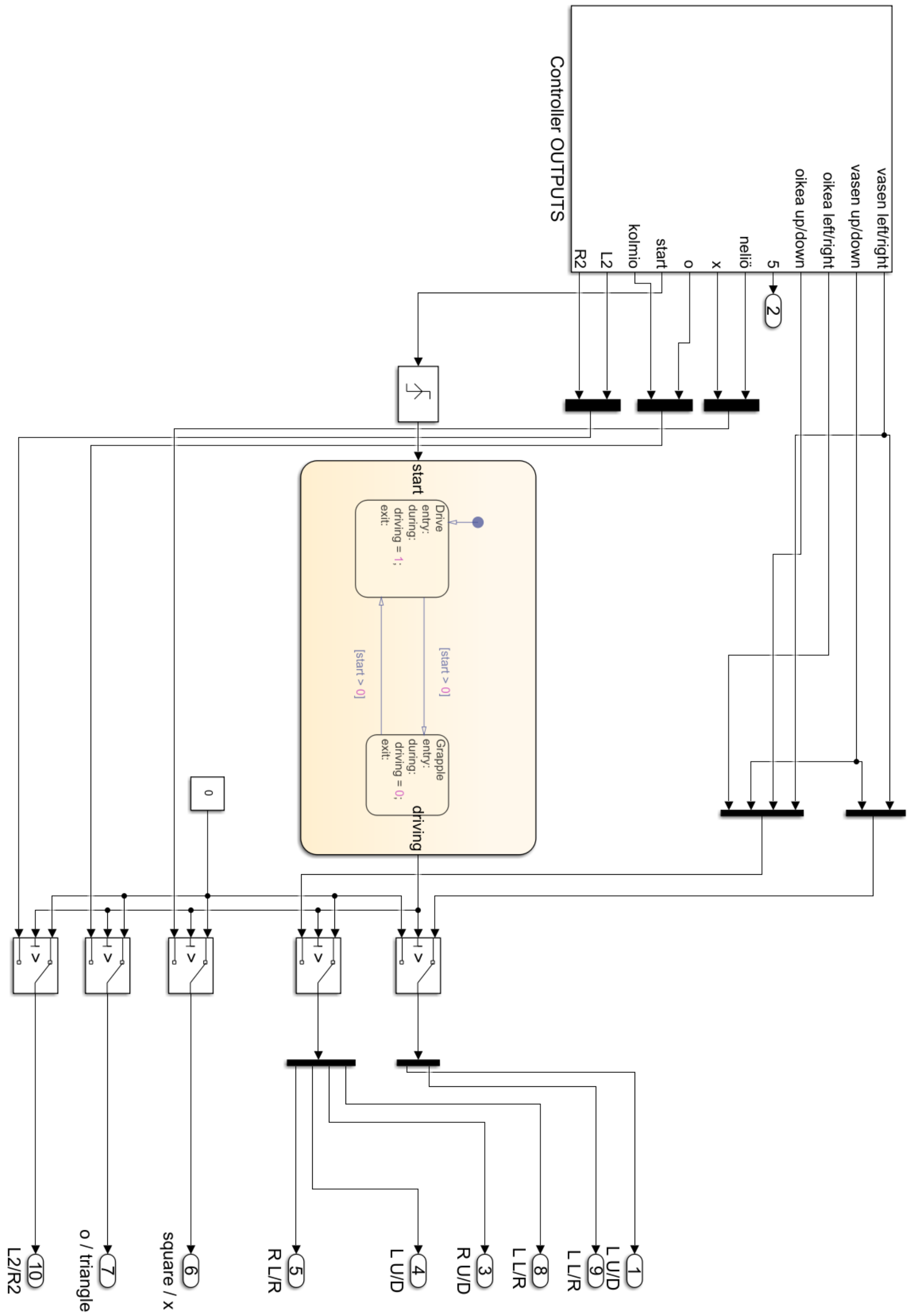

Figure A6. PC controller subsystem.

B. MATLAB initialization code

Editor - E:\GRADU\Matlab\Forwarder\parameters.m

parameters.m

```matlab
clear
clc

% GRAPPLE POINT OPACITY
% ________________________________
gpo = 0;
% ________________________________



% POINT CLOUD FOR TIRES
% ________________________________
% Parameters for the cylinder
radiusTire = 0.71;              % Diameter = 2 * 710 mm, so radius = 0.71 m
heightTire = 0.71;              % Height = 0.75 m

% Number of points
numPointsTire = 3;

% Generate points along the cylinder
thetaTire = linspace(0, 2*pi, numPointsTire * 7);       % Angle around the cylinder
zTire = linspace(0, heightTire, numPointsTire);         % Height points along the cylinder

% Create a grid of points
[ThetaTire, ZTire] = meshgrid(thetaTire, zTire);        % Create a grid for cylindrical coordinates
XTire = radiusTire * cos(ThetaTire);                    % X-coordinate for the points on the cylinder surface
YTire = radiusTire * sin(ThetaTire);                    % Y-coordinate for the points on the cylinder surface

% Convert the mesh to a point cloud
pointCloudTire = [XTire(:), YTire(:), ZTire(:)];
% ________________________________



% GROUND TIRE CONTACT
% _____________________________________
g_stiffness = 6e6;      % 6e5 toimii gridin kanssa ja 100000 terrainin
g_damping = 4e5;        % 4e4 toimii gridin kanssa ja 1000 terrainin
g_static_friction = 1.0;
g_dynamic_friction = 0.7;
% _____________________________________



% LOG GRAPPLE CONTACT
% _____________________________________
lg_stiffness = 1e6;                                     % alkuperäinen arvo 1e6
lg_damping = 10000;                                     % alkuperäinen arvo 1000
lg_coefficient_of_static_fiction = 1.2;                 % alkuperäinen arvo 0.7
lg_coefficient_of_dynamic_friction = 0.8;               % alkuperäinen arvo 0.5
% _____________________________________



% GROUND LOG CONTACT
% _____________________________________
gl_stiffness = 1e6;                             % Original 1.5e6
gl_damping = 1e4;                               % Original 1.9e4
gl_coefficient_of_static_fiction = 1.2;           % Original 0.9
gl_coefficient_of_dynamic_friction = 0.9;         % Original 0.8
% _____________________________________



% GRID GROUND
% __________________________________
Floor.l = 100;
Floor.w = 100;
Floor.h = 0.2;

% Grid parameters
Grid.clr = [1 1 1]*1;
Grid.numSqrs = 50;
Grid.lineWidth = 0.02;
Grid.box_h = (Floor.l-(Grid.lineWidth*(Grid.numSqrs+1)))/Grid.numSqrs;
Grid.box_l = (Floor.l-(Grid.lineWidth*(1+1)))/1;
Grid.extr_data = Extr_Data_Mesh(Floor.w,Floor.w,Grid.numSqrs,1,Grid.box_h,Grid.box_l);
% ____________________________________

```

Figure B1(a). MATLAB initialization code (part 1/3).

Editor - E:\GRADU\Matlab\Forwarder\parameters.m

parameters.m

```matlab



% ==============================================
% COMBINED POINT CLOUD FOR LOG BASE & TRAILER WALL
% ==============================================

% LOG BASE POINT CLOUD
cross_section_LOGBASE = [
    0, 0; 700, 0; 700, 800; 1490, 800; 1490, 3550;
    1340, 3550; 1340, 1000; 500, 1000; 500, 200; 0, 200;
    -500, 200; -500, 1000; -1340, 1000; -1340, 3550; -1490, 3550;
    -1490, 800; -700, 800; -700, 0; 0, 0
];

% Specific Z positions for LOG BASE
z_positions_LOGBASE = [100, 1550, 3000, 4450];
resolution_LOGBASE = 10;  % Number of points along cross-section edges

point_cloud_LOGBASE = [];
for i = 1:length(z_positions_LOGBASE)
    z_LOGBASE = z_positions_LOGBASE(i);
    for j = 1:size(cross_section_LOGBASE, 1)-1
        x1 = cross_section_LOGBASE(j,1); y1 = cross_section_LOGBASE(j,2);
        x2 = cross_section_LOGBASE(j+1,1); y2 = cross_section_LOGBASE(j+1,2);
        x_interp = linspace(x1, x2, resolution_LOGBASE);
        y_interp = linspace(y1, y2, resolution_LOGBASE);
        new_points = [x_interp', y_interp', repmat(z_LOGBASE, resolution_LOGBASE, 1)];
        point_cloud_LOGBASE = [point_cloud_LOGBASE; new_points];
    end
end

% Remove duplicates
point_cloud_LOGBASE = unique(point_cloud_LOGBASE, 'rows');

% ---- TRAILER WALL POINT CLOUD (Unchanged) ----
cs_wall = [
    0, 0; 300, 0; 807.5, 800; 1315, 800; 1315, 3000; 455, 3000;
    192, 2543.5; 0, 2543.5; -192, 2543.5; -455, 3000; -1315, 3000;
    -1315, 800; -807.5, 800; -300, 0; 0,0
];

z_min_wall = 5196.33;  % Ensuring trailer wall starts after log base
z_max_wall = z_min_wall + 200;
num_z_samples_wall = 2;
resolution_wall = 15;

x_range = linspace(min(cs_wall(:,1)), max(cs_wall(:,1)), resolution_wall);
y_range = linspace(min(cs_wall(:,2)), max(cs_wall(:,2)), resolution_wall);
[X, Y] = meshgrid(x_range, y_range);
[in, ~] = inpolygon(X, Y, cs_wall(:,1), cs_wall(:,2));
x_in = X(in);
y_in = Y(in);

z_samples_wall = linspace(z_min_wall, z_max_wall, num_z_samples_wall);
point_cloud_wall = [];

for i = 1:length(z_samples_wall)
    z_wall = z_samples_wall(i) * ones(size(x_in));
    new_points = [x_in, y_in, z_wall];
    point_cloud_wall = [point_cloud_wall; new_points];
end

% Remove duplicates
point_cloud_wall = unique(point_cloud_wall, 'rows');

% MERGE BOTH POINT CLOUDS
point_cloud_combined = [point_cloud_LOGBASE; point_cloud_wall];
% ______________________________________________

```

Figure B1(b). MATLAB initialization code (part 2/3).

Editor - E:\GRADU\Matlab\Forwarder\parameters.m

parameters.m

```matlab


% GRID SURFACE (Adjusted with a hill in the corner)
% ________________________________________________
% Define grid size
xMin_terrain = 0; xMax_terrain = 100; % X range in meters
yMin_terrain = 0; yMax_terrain = 100; % Y range in meters
gridResolution_terrain = 0.25; % Grid resolution in meters

% Generate grid vectors
xVec_terrain = xMin_terrain:gridResolution_terrain:xMax_terrain; % X Grid Vector
yVec_terrain = yMin_terrain:gridResolution_terrain:yMax_terrain; % Y Grid Vector
[X_terrain, Y_terrain] = ndgrid(xVec_terrain, yVec_terrain);

% Generate terrain height using Perlin noise and sine waves (reduced amplitude)
Z_terrain = 0.3 * perlinNoise2D(X_terrain, Y_terrain, 0.05) + ... % Reduced Perlin noise amplitude
    0.5 * sin(0.02 * X_terrain) .* cos(0.02 * Y_terrain) + ... % Smaller and smoother undulations
    0.05 * randn(size(X_terrain)); % Small random roughness

% Add a hill feature at the corner of the terrain
hill_center = [20, 80]; % Position of the hill (meters)
hill_height = 6; % Height of the hill (meters)
hill_radius = 12; % Radius of the hill (meters)
hill = hill_height * exp(-((X_terrain - hill_center(1)).^2 + (Y_terrain - hill_center(2)).^2) / (2 * hill_radius^2));
Z_terrain = Z_terrain + hill;

% Apply Gaussian smoothing filter to reduce spikiness
Z_terrain = imgaussfilt(Z_terrain, 2);

%% Function to generate Perlin noise
function Z_terrain = perlinNoise2D(X_terrain, Y_terrain, scale_terrain)
    rng(1); % Ensure reproducibility
    noiseSize_terrain = max(size(X_terrain));
    noise_terrain = rand(noiseSize_terrain, noiseSize_terrain);
    Z_terrain = interp2(noise_terrain, X_terrain * scale_terrain + noiseSize_terrain/2, Y_terrain * scale_terrain + noiseSize_terrain/2, 'linear', 0);
end

% ________________________________________________



% POINT CLOUD FOR LOGS
% ________________________________
% Log dimensions
radius_log = 0.25; % meters
length_log = 4.5; % meters
numPointsCircumference_log = 6;     % 6 toimii Number of points around the circumference
numPointsLength_log = 3;            % 3 toimii Number of points along the length of the log

% Generate evenly spaced angles around the cylinder's circumference
angles_log = linspace(0, 2*pi, numPointsCircumference_log); % Evenly spaced angles
heights_log = linspace(-length_log/2, length_log/2, numPointsLength_log); % Evenly spaced heights along the length

% Create a meshgrid to generate points
[angleGrid_log, heightGrid_log] = meshgrid(angles_log, heights_log);

% Convert to Cartesian coordinates (cylinder surface)
X_log = radius_log * cos(angleGrid_log); % X-coordinates on the surface
Y_log = radius_log * sin(angleGrid_log); % Y-coordinates on the surface
Z_log = heightGrid_log; % Z-coordinates, evenly spaced along the length of the log

% Store point cloud for Simulink
logPointCloud = [X_log(:), Y_log(:), Z_log(:)];

% ________________________________
```

Figure B1(c). MATLAB initialization code (part 3/3).