



Design Technique to Parameterize a Real-Time Simulation Model of an Industrial Vehicle with Customized Parameters.

By

Student Name: Harrisou Adamu

ID: 180011242

Supervise By:

Prof. Dr. Shamsuddin Ahmed

Department of Mechanical and Production Engineering

Islamic University of Technology

Dhaka, Bangladesh

May 2023

DECLARATION BY THE CANDIDATE

It is hereby declared that the work presented in this thesis is my own. The thesis or any part of it has not been submitted elsewhere for the award of any degree or professional qualification.

The content of this graduation exercise is the result of the work I have been carrying out since the official commencement date of the approved dissertation.

Name and signature of the Candidate

Name and Signature of the Supervisor

Department of Mechanical and Production Engineering
(MPE)Islamic University of Technology (IUT)
Dhaka, Bangladesh

CERTIFICATION OF RESEARCH

This Thesis is Entitled **Design Technique to Parameterize a Real-Time Simulation Model of an Industrial Vehicle with Customized Parameters**, submitted by Harrisou Adamu, student ID 180011242, and has been accepted as satisfactory in Partial Fulfilment of the Requirement for the Degree of Bachelor of Science in Mechanical Engineering in May 2023.

Prof. Dr. Shamsuddin Ahmed

Abstract

This study presents a strategy for building a real-time simulation model that runs in real-time with user-closable, customizable parameters. There are eight steps and four decision points in the proposed design process model. Parameterization is a method that lets you simulate different combinations of parameters in real time. The proposed technique is introduced as a flowchart. Given the information, a 3D plan of the model was developed. Distinct strategies comprising two types are presented toward building a defined real-time parametric simulation model that incorporates operator input and customizes it accordingly. The methodology utilized was chosen given the number of defined parameterized specifications. Analytical and simulation-based evaluations of each case's feasibility were used. In the end, a model version was chosen based on the initial requirements. Parameterization was demonstrated on an excavator model in Orthoclase the developed method. The two areas of the excavator model that are thought to have customizable features are the hydraulic cylinders and the bucket, there are the two components of focus in the real-time simulation results. The goal is to create an accurate and realistic simulation model that can be used for training, testing, and optimization. And because some parts are essential to the excavator's performance, the parameterization process will concentrate primarily on the hydraulic system and bucket. The excavator's safety and efficiency can be enhanced by accurately modeling these parts in the simulation model. This ultimately results in increased productivity and lower operating costs.

Keywords:

Customized parameters, Real-time simulation model, Parameterized Specifications, Excavator model

ACKNOWLEDGEMENT

I express our heartiest gratefulness to the Almighty Allaah SWT for His blessings, which made us live and possible to complete this thesis work successfully.

Developing a project is a long and intense effort. Auspiciously, there is assistance, recommendation, suggestion, advice and encouragement from many people. I could not list everyone who helped me here, but a few people must be thanked publicly. If I unintentionally omit anyone, I apologize in advance.

First, special thanks to Prof. Dr. Shamsuddin Ahmed for his valuable inspections, suggestions, and guidance throughout this final-year project. He has been giving out full support and compassionate encouragement during the implementation of this project. Without his full support and special guidance, this dissertation would not have been successfully completed.

Finally, I wish to express sincere appreciation to my parents, siblings and friends who willing to share the comments with me and support me during the whole period in launching the project. Without their support, inspiration and sacrifice, this project would never have been finished successfully.

Thanks the Author.

Table of Contents

DECLARATION BY THE CANDIDATE	ii
CERTIFICATION OF RESEARCH	iii
Abstract.....	iv
ACKNOWLEDGEMENT	v
SYMBOL AND ABBREVIATION LIST	ix
Chapter 1.....	1
Introduction	1
1.1 Background of the Study.....	1
1.2 Research Problem Statement	2
1.3 Research Questions	2
1.4 Aim and Objectives	3
1.5 Research Methods	4
Chapter 2.....	5
Literature Review	6
2.1 Introduction	6
2.2 Simulation in researches.....	6
2.3 Equations and Principles of a Multibody System.....	8
2.4 Coordinates both globally & locally	10
2.4.1 Equations of Kinematic Constraint about rotational coordinates	11
2.4.2 Kinematic Joints Constraints	12
2.4.3 Equations of Motion	14
2.4.4 Simulation in practice	16
2.4.6 Marketing.....	17
2.4.7 Customizable Mode	17
2.4.8 Employed Software.....	18
2.4.9 Vortex.....	18
2.5 CAD Software – SolidWorks	20
2.5.1 Blender	21
2.5.2 Excel and Python.....	22
2.6 Four-Bar Mechanism.....	23
Chapter 3.....	27
Methodology.....	27
3.1 Case study – An excavator	27
3.1.1 How an Excavator Works	27
3.2 Stimulated Industrial Vehicle	29

3.3	Editable Parameters.....	30
3.4	System for lifting and a bucket	30
3.4.1	Combination and comparison for the Bucket's examination	33
3.4.2	Combination and comparison for the Bucket's examination	41
	Chapter 4.....	42
4	Results and Analysis.....	42
4.1	Evaluation of various buckets for use	42
4.1.1	Consumption of fuel for three tests using three different buckets.....	44
4.1.2	Evaluation of various hydraulic circuits for use	46
4.2	Future work.....	49
4.2.1	Using Software and the connections between them	50
4.2.2	An intuitive user interface – Gamification	51
4.2.3	Further customizations - Analyze the section.....	52
4.3	User-friendlier interface - Gamification.....	52
4.3.1	Further customizations - Analyze section	52
4.4	Representation of models and conditions - Climate customization.....	52
4.4.1	Additional modifications: Examine the section.	52
4.4.2	Model and environment visualization - Environment customization.....	53
	Chapter 5.....	55
5	Conclusions	55
	References	58
	Appendix 1	61
	Appendix 2	62

LIST OF FIGURES

FIGURE 1. 1 FLOW CHAR.....5
.....

FIGURE 3. 1 THE EXCAVATOR, COMPLETE WITH ITS TWO CUSTOMIZATION OPTIONS.27
FIGURE 3. 2 TYPICAL APPLICATIONS FOR THE EXCAVATOR (CATERPILLAR, 2018).....29
FIGURE 3. 3 THE KINDS OF BUCKETS THAT ARE USED FOR CUSTOMIZATION.....31
FIGURE 3. 4 PORTIONS OF THE INTERFACE FOR VORTEX MODELER.....31
FIGURE 3. 5 THE GLOBAL Y POSITIONS OF THE SMALL BUCKET AND THE TOTAL TIME ARE SHOWN
.....34
FIGURE 3. 6 GASOLINE UTILIZATION FOR THE EXCAVATOR UTILIZING BUCKET A35
FIGURE 3. 7 SHOWS THE ANGULAR VELOCITY OF BUCKET A IN THE Y-AXIS FOR THE DIP ARMS
CYLINDERS.35
FIGURE 3. 836
FIGURE 3. 9 ANALYZE THE HYDRAULIC CIRCUIT'S PERFORMANCE37
FIGURE 3. 10 THE TOTAL DURATION AND LOCATIONS OF BUCKET B ARE SHOWN.....38
FIGURE 3. 11 FUEL CONSUMPTION GRAPH FOR THE MEDIUM BUCKET (BUCKET B).38
FIGURE 3. 12 : OVERALL TIME AND BUCKET C LOCATIONS IN GLOBAL Y-AXIS.....40
FIGURE 3. 13 EXCAVATOR FUEL USAGE WITH BUCKET C40
.....

FIGURE 4. 1 A SINGLE TEST'S GRAPHS OF THREE DISTINCT BUCKET TYPES.42
FIGURE 4. 2 PLOTS FOR THE PRECISE SPEED OF THE DIP ARM.....44
FIGURE 4. 3 FUEL CONSUMPTION IN THREE KINDS OF BUCKETS.....45
FIGURE 4. 4 COMPARISON OF THREE BUCKET TYPES' SPECIFICATIONS FOR AN EXCAVATOR46
FIGURE 4. 5 THE ANGULAR VELOCITY FOR EACH OF THE EXCAVATOR'S 3 CYLINDER-PISTON
INPUTS47
FIGURE 4. 6 THE CYLINDER-PISTON DIPPER ARM ACTUATOR VELOCITY FOR THREE VARIATIONS
.....48
FIGURE 4. 7 UTILIZATION OF FUEL FOR THE THREE SORTS OF DIPPER ARM CHAMBER CYLINDERS
.....49

List of Tables

TABLE 2. 1 DEPICTS THE VALUES FOR THE PARAMETERS.24
.....

TABLE 3. 1 PARAMETERS THAT NEED TO CHARACTERIZE EVERY THREE KINDS OF THE BUCKETS.
.....32
TABLE 3. 2 **BUCKET A** (SMALL BUCKET).....33
TABLE 3. 3 **BUCKET B** (THE MEDIUM BUCKET)37
TABLE 3. 4 **BUCKET C** (THE BIG BUCKET).....39

SYMBOL AND ABBREVIATION LIST

I - Unit vector

I - Inertia

m - Mass

M - Mass matrix

n - Number of coordinates

n1 - Order of a differential equation

O - The element of the final rotational transformational matrix x

q - A generalized coordinate vector

r - Vector of position

R - Position

ω - Angular velocity

φ - Angular variable

Φ - Constraint of the velocity

SD - System Dynamics

XML - Extensible Markup Language

3D - Three dimensions

Chapter 1

Introduction

1.1 Background of the Study

A useful tool has been dynamic simulation, which is frequently applied to several production processes. Formulation and solution of the motion equations are required before computerized methods can be used to evaluate a machine's performance. Numerous research capitalizing writing and calculations of motion equations in multi-body dynamic systems could be obtained from previous literature [1]-[4]. While bearing in mind the strong display of machinery, it is vital to note that regardless of the directors' familiarity regularly expect a huge part. Anyway, maximum studies on simulation concentrated on showing techniques, while perusing up that record for the effect of the manager sticks out. The use of real-time simulation of dynamic systems enables greater user consideration. When operating a real-time simulation model, operators exert as though they are working with physical machinery. The real-time physics base model must be precise and integrate the various engineering domains in order to accomplish this: like power through pneumatics, hardware, and pressure. However, at the moment, these real-time models are typically tailored to particular applications and case-specific. Additionally, developing such a physic base real-time model which is capable to running in real-time is labor-intensive and costly. The burden issue can be addressed by using a physic base real-time simulation technique centered on the dynamics of multi-body systems. Real-time simulation in multi-body systems has been studied in a variety of fields, including aeronautics, heavy vehicles [7, 9], crossbreed vehicles [10, 11], automotive [12, 13], four-bar mechanisms, and elastic multi-body systems [12, 13]. On the other hand, real-time simulation hasn't been used much because it takes a long time to build a model and might not produce valuable effects [14]. Movable physical science-based, continuous, test system-driven item improvement cycles can settle the issue of current machine reenactment approaches' significant expense and high particularity. This thought can be achieved by fostering a toolset that will permit clients to get to machine innovative work. In practice, virtual worksites that offer real-time, fully configurable virtual prototyping make this possible. Product design and construction necessitate a multi-step approach from a system engineering perspective. To start with, the issue and the prerequisites of any conceivable arrangement are distinguished.

Then, at that point, exercises to attain the goal are hosted ahead of time, trailed by a specific assessment of every step [15]. The examination stages measure if the fabricated thing meets the necessities at every stage and comment on the chance of the picked method or the best

essentials. Merchandise improvement methods ought to be universal and adaptable to the absolute artifact [16, 17]. Notwithstanding frequent research on product development, providing simulation models with variable factors, the creation of general models in various fields, the use of multi-body system simulation, and the interface with users in the planning phase have received little attention. [18]- [21]. The study aims to demonstrate a technique for developing a model with variable parameters for real-time simulation. A flowchart is used to show the steps of the design. A bucket model is chosen as a contextual investigation to be defined. Parameters can be chosen based on various operational scenarios when developing simulation models using the method that has been presented. The approach makes it possible to build simulations of the systems that were formerly too tough to analyze, and a lot of work to apprise the real-time simulation model by expanding the real-time methodology's applicability. As a result, this study is to investigate methods for creating attainable real-time simulation models with variable factors.

Development of this study is according to the accompanying: In the second section, you will learn how to design a real-time simulation model of parameters using a flowchart. The practicality examination and a mathematical case model are presented in the third segment. In Section 4, each method's benefits and drawbacks are explained in detail. The last part is the conclusion.

1.2 Research Problem Statement

The following problem are:

- There is a need for prospects to analyze compound systems' behavior via potent software that enables the design of unpretentious models that can precisely embody the systems and countenance the usage of simulation to accomplish mathematical combinations and dynamic examination in a quicker, meeker, and calmer way.
- There is a need for operators to check products, give vital facts to manufacturers, and operators, and make merchandise grow more customer-oriented.
- Engineers need to expand merchandise faster, with a decrease in the related cost.

1.3 Research Questions

When a research problem or issue is encountered, research questions frequently arise.

The primary issue with this project is: A simulation model's adjustable customization is always time-consuming and expensive. The Vortex programming utilizes a comprehensible text record that has information about a model that could run a model in its connection points. Modifying

the model's interface, the readable text file mentioned earlier, and other options are available for a customizable model. The research questions that come to mind are the ones listed below:

- i. How possible is it to gain admittance to all of the model's data in a way that allows for modification and simple saving?
- ii. What is the reasonable strategy to figure out positive information, choose amidst them, and create a prepared or ready to run model?
- iii. Will it be conceivable to devise a tweaked model that a client can choose an extensive variety of information, rather than just a couple of choices, among the sub-gatherings and congregations Vortex programming gathers them and makes the model with next to no disturbing about their change?
- iv. Is the user's chosen configuration practical?
- v. Is there a mandate for a widespread selection of preferences for the parts of the model?
- vi. Which Boundaries of the model is alterable?

1.4 Aim and Objectives

A customizable simulation model for Vortex software will be developed as part of this dissertation. Right now, according to a simulation perspective, a model has a bunch of consistent boundaries to mimic it and concentrates the positive outcomes, and examines them. The old methods take a long time to change a model's adjustable parameters, so it's very convenient to have a way to make a model that can change its data quickly and precisely.

The creation of a user-selectable, adjustable simulation model is the primary objective of this thesis. To put it another way, the Vortex software has a simulation model that takes user input and creates a model—in this case, an excavator—based on that information. The argument is, information is obtained from certain gatherings and sub-congregations and drives the model, this large number of congregations should be very much matched to one another so that the client could feel the impacts of her/his choices.

A simulated excavator model's changeable characteristics for this dissertation are as follows:

- a) The bucket's dimensions are in both visual and collision modes, as well as its mass.
- b) The dipper arm's cylinder and piston dimensions.
- c) The volume of the dipper arm cylinder's nominal flow rate.

Depending on the user's parameters, a simulation model may be able to complete a task faster than a normal real model, but it will use more fuel. On the other hand, it could be a model that uses gas but is slower than before. All designs for the climate and adaptable pieces of the model are changed with other static parts.

The goal for the future is to devise added customizable constraints and chunks of a model, similar to an excavator, which can be analyzed based on some real-world parameters. To put it another way, it will be both a model and a method for determining which model is more reliable in terms of wear, depreciation, ease of maintenance, and cost based on fuel consumption and working hours. After this dissertation, specific future objectives will be discussed.

1.5 Research Methods

Simulation utilized in this study were the Vortex programming as a constant simulation to make a reproduction model with another programming as its partner. It is important to note that SolidWorks, Blender, Python, and Excel are all used during the creation process. Subsequently, the simulation model can be run. In the plan section, Vortex generate an XML file containing the model's data. Various ways to edit that file to change the model's capabilities can be used. However, the most practical approaches to completing this dissertation work are lettering a Python code and running with an Excel code, creating an XML code containing assemblies, which will be discussed in the following chapters.

Methodology of this postulation, there is a reproduction model with static boundaries that are flexible. The model will be editable using the methods described in the design section. With the assistance of SolidWorks and Blender software, the visualization and collision graphics will adapt simultaneously, and the entire model can be run on the Vortex platform. A general design flowchart is used to construct a customizable excavator, as shown in Fig. 1. The plan interaction comprises eight stages and four choice focuses. In Given information, a 3D plan of the model was developed.

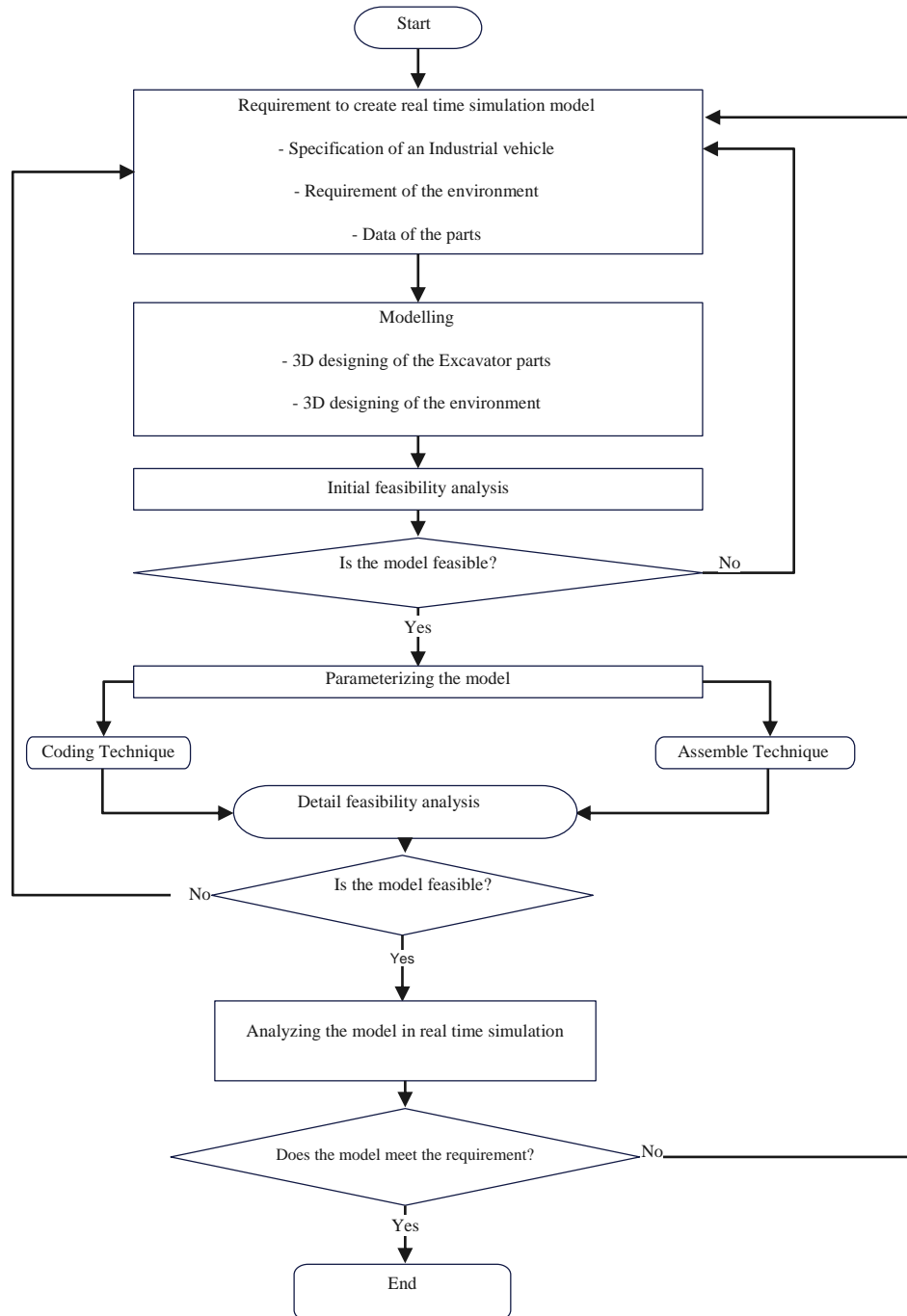


Figure 1. 1 An industrial vehicle's parametric real-time simulation model's design flowchart

Chapter 2

Literature Review

2.1 Introduction

At the point when a new project undertaking and thought comes as a primary concern, seeing and reviewing is dependably sensible to past explores in advance to see what investigates have proactively done and what is their way to deal with taking care of an issue. With a writing survey of past explores' outcomes, the examination can be found and it is feasible to have a correlation among them and sort out their coverage at work, their thought regarding the task, and the title which is getting looked at. In addition, it may be possible to identify obstacles and limitations that predate previous research by conducting a thorough literature review.

Real-time simulation, earlier efforts, and previous works in the field of simulation can all be discussed in this chapter. However, since the simulation is very practical in many different fields, it makes sense to review real-time simulation regarding manufacturing and normally used vehicles, particularly previous efforts regarding parameterization.

2.2 Simulation in researches

FINNA – Wilma was used as a practical database for the review in previous studies. It has reviewed the majority of scientific databases and journals and has covered a large number of books, conferences, and articles. Steffen Bang Sow tried, among other things, to find a solution for the simulation. He suggested using a rational chart to figure out how to solve the simulation problem by formulating problems and goals, collecting data, modeling and running the model, and analyzing the results. Bang sow, 2010, p. 2). Steffen Bang Sow's steps are depicted in Figure 2.1; however, his research did not include a method for determining the calibrations of the simulated model.

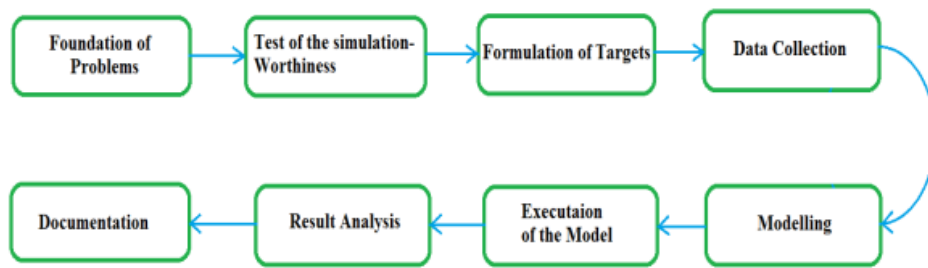


Figure 2. 1 Stages of the simulation delinquent that are built on research by Steffen Bangsow

The periods of a simulation issue in view of exploration by Steffen Bangsow. The process of creating a model of a real-world system and studying its behavior under various conditions is known as simulation. The periods of a reenactment issue incorporate issue definition, model conceptualization, model execution, and simulation explore plan, and result examination. Their framework interfaces all parts together and incorporates an information-driven reproduction bit. This invention aims to provide a training environment that is both realistic and dependable for technical skills related to software-based systems, machinery, and equipment.

A method for real-time simulation in preparation that all system components are interconnected has been patented by Edward Robert Comer and his coworkers. A data-driven simulation grain with subdivisions, as shown in Figure 2.2. This flagrant is imagined " for preparing specialized abilities on gear, hardware, and software-based frameworks". With the aid of simulation, they attempted to create a realistic and dependable training environment in this creation. (Comer. 2005)

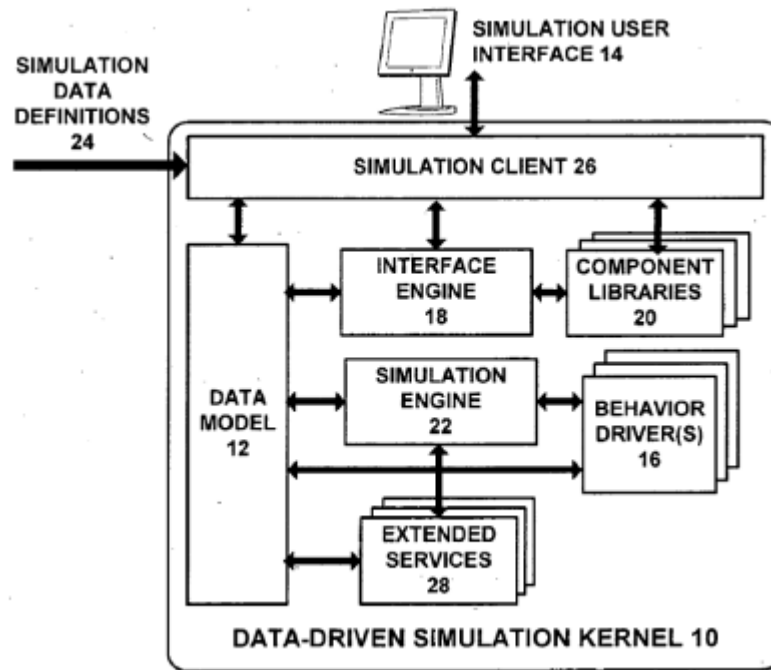


Figure 2. 2 Protected solicitation for preparing with the assistance of simulation (Comer. 2005).

Comer has attempted to connect a simulation client and the crucial of the simulation drill model, such as the XML edge and simulation data. This feature, which has access to useful some part of the model, is useful; even though, its system has limitations that make it difficult to work with. In different words, it can't rise out of the preparation part and reach out to different parts which can be useable for not-prepared clients. In light of past explores, there are two reproduction draws near, discrete-occasion simulation, and framework elements, which are called DES and SD all the while. The development performance of a simulated vehicle over time serves as the foundation for both discrete event simulation and system dynamics approaches. Moreover, they can recognize a few upgrades for models which should be possible coming down the line for a model. The differential equations serve as the foundation for the system dynamics approach. (Tako, 2010, p. 784)

2.3 Equations and Principles of a Multibody System

Multibody System Dynamics (MSD), universal and indigenous synchronizes, kinematic restraint equivalences and equations of motion are all discussed from a broader perspective in this chapter. The classic mechanical mechanism, satellites, and robots are the foundations upon which knowledge of multibody system dynamics is built, as evidenced by numerous signs from the history of mechanical engineering. Multibody framework elements are portrayed by calculations or formalism, individually, prepared for PC execution." Exact and useful collaboration

with computer-aided design programming, definition, constant simulation, links and the association between the parts, control mechatronic frameworks, and the investigation of the entire multi-bit are the principal viewpoints on Multibody Framework Elements (MSD). Also, the examination of MSD's conduct, utilizing decreased techniques in excessively outstands exact consequences in joining codes for Tribute, Conventional Distinction Conditions, and DAE, Difference-logarithmically conditions. (Schiehlen, 1997, p. 149).

The conditions that make sense of a movement for multibody frameworks are called Newton-Euler conditions. In the following sections of this report, we will review the MSD motion equation fundamentals. Lagrange provided an analysis of the mechanical constraints system in 1788. Lagrange's equations DAE and ODE explain the system's total kinetic and potential energy. Constraints and generalized coordinates ought to be taken into consideration in this system. (Schiehlen, 1997 1-2). Reasonably, a multibody framework (MBS) has associations with double gatherings with essential letterings and machine-driven constituents that can demonstrate displacements are the first, followed by relationships between bodies, constraints, and kinematic joints. In other words, an assembly of inflexible or flexible physiques linked by kinematic joints and probably some dynamism features are known as a multi-body system. In light of the requests of an MBS, the body for a multibody framework is portrayed as an inflexible or adaptable body. With utilizing six summed-up facilitates of six degrees of opportunity, DOF, the movement of the inflexible frame is shown whereas 3D planetary is characterized. (Flore. 2016, p. 1)

Description of tough junctions among parts in an MBS framework, of the three-dimensional plane, but break will decrease. (Figure 2.9 depicts the kinds of coordinates that are frequently used to explain the MBS (Flore. 2016, p. 7)

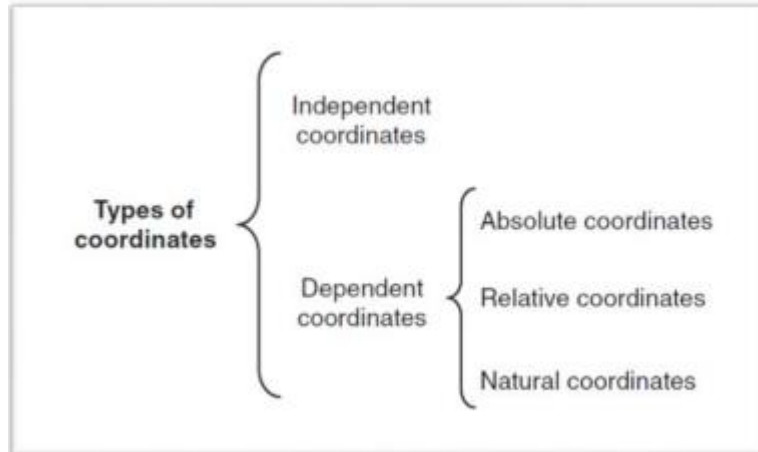


Figure 2. 3 Coordinates that are often used in MBS (Flore. 2016, p. 7).

Using Euler angles, Bryant angles, or Euler parameters, rotational coordinates can be used to describe the orientation of a rigid body in three-dimensional space. We can locate any rigid body using six coordinates, three for translation and three for rotation.

A restriction on the movement of one or more bodies is referred to as a constraint. Generalized coordinates, which are exemplified by a vector q with n coordinates, can be used to define the orientation and location of bodies. The constraint is represented in this report by the parameter and number.

$$r_{i1} = \{x_{i1} \ y_{i1} \ z_{i1}\}^T \dots \dots \dots (2.1)$$

2.4 Coordinates both globally & locally

The dislocation of a free-moving particle ($i1$) in three-dimensional can be described by three independent variables and the position vector (r) (Flore. 2016, p. 11):

$$r_{i1} = \{x_{i1} \ y_{i1} \ z_{i1}\}^T \dots \dots \dots (2.1)$$

The aforementioned definition can also be applied to an inelastic frame and its position. Additionally, its alignment is explained using a locus system. The idea of the global coordinate system can be found in Appendix 1, Flore. 2016, page 12.

2.4.1 Equations of Kinematic Constraint about rotational coordinates

There are a few ways to deal with making sense of rotational directions in 3D MBS that are Euler Points, Bryant Points, and Euler Boundaries. Any rigid body's location can be determined using six coordinates, three translational and three rotational. All steps for spatial MBS with rotational coordinates can be found in Appendix 2. The constraint of rigid bodies is due to their degrees of freedom. The position and angle of bodies can now be determined using the idea of generalized coordinates. The vector

$$q = \{q_1, q_2, q_3, \dots, q_n\}^T \dots \dots \dots (2.2)$$

The generalized coordinates that define the location and orientation of bodies are represented by the vector $q = q_1, q_2, q_3, \dots, q_n^T$ in the context of mechanical systems. These coordinates can be used to describe the translational and rotational displacement of a rigid body in three-dimensional space. For gaining an understanding of mechanical systems' constraints and degrees of freedom, the concept of generalized coordinates is crucial.

T indicates that n is the coordinate's value that demonstrates the restraint using a constraint and a value in this report. The boundary shows the kind of imperative and the number is the quantity of condition. Example, $\Phi(s, 2)$ displays the limitation of two conditions. (Flore. 2016, p. 31)
The kinematic condition of the trajectory figure plane is displayed (Flore. 2016, p. 33)

$$\Phi \equiv \Phi(q) = 0 \dots \dots \dots (2.3)$$

The condition $\Phi \equiv (q) = 0$ addresses a kinematic limitation condition in view of the vector of body-organizes, where \mathbf{q} is a vector that incorporates three translational directions and Euler boundaries for a body.

This condition is utilized to infer speed requirements, which are addressed by $\Phi = Dv = 0$, where \mathbf{D} is the Jacobian framework and \mathbf{v} is the speed vector that incorporates translational and rakish speeds.

The equation $\mathbf{qi} = \{\mathbf{ri} \ \mathbf{pi}\}$ represents a vector of body-coordinates that includes three translational coordinates (\mathbf{ri}) and Euler parameters (\mathbf{pi}). Based on the body-coordinate vector, this equation is used to get the kinematic equation, which looks like $(q) = 0$. The velocity constraints can be deduced as $= Dv = 0$, where v is the velocity vector that includes the angular and translational velocities, respectively, and D is the Jacobian matrix.

The velocity vector of a body, $v_i = \{ \dot{r}_i \}$, denoted by the symbol v_i , is depicted in Equation 2.4. It is made up of the time derivatives of the translational coordinates (r_i) and the angular velocities ($\dot{\theta}_i$). The rotational components of the x, y, and z axes are all represented by the angular velocities by the vector $\dot{\theta}_i$. This condition is utilized in the determination of speed imperatives in multibody frameworks.

$$v_i = \begin{Bmatrix} \dot{r}_i \\ \dot{\theta}_i \end{Bmatrix} \dots \dots \dots (2.4)$$

The angular velocities path is ω_i (Flore. 2016, p. 28)

$$\omega_i = \{ \omega_x \ \omega_y \ \omega_z \}^T \dots \dots \dots (2.5)$$

The kinematic equation is founded on the vector of body coordinates and relates the position and orientation of a rigid body to its velocity and acceleration, making use of this vector. The use of kinematic joint constraints, such as spherical, revolute, and spherical-spherical joints, in real-world simulations of 3D multi-body systems is the subject of the context.

Simulations of 3D multi-body systems is the subject of the context.

$$\Phi \equiv D\dot{v} + D v = 0 \quad (2.6)$$

The second derivative of the kinematic equation, which describes the velocity constraints of a multi-body system, is denoted by the expression

$$Dv + D v = 0 \dots \dots \dots (2.7)$$

The Jacobian matrix D and the velocity vector v , both of which are derived from the body-coordinates vector q , are necessary for comprehending the motion and behavior of complex mechanical systems like those studied in real-world simulations of spherical, revolute, and spherical-spherical joints.

2.4.2 Kinematic Joints Constraints

A great deal of 3D MBS can be studied from the perspective of real simulation with the help of the following joints; the spherical, revolute, and spherical-spherical. It permits three relative rotations for a spherical joint (see Figure 2.10). "In this manner, the focal point of the circular joints has consistent directions concerning any of the neighborhood facilitates frameworks of the associated frames such as a round joint which is characterized by the form of point P_i on the body (D)harmonize on the points P_j on body j . This condition is based on circular limitation, and it is written here in a scalar structure ": (Flore. 2016, p. 43)

$$\Phi(s,3) \equiv r_{jP} - r_{iP} = r_j + s_{jP} - r_i - s_{iP} = 0 \quad (2.8)$$

The condition $\Phi(s,3) = r_j + s_{jP} - r_i - s_{iP} = 0$, addresses the condition of the speed imperative for a circular joint between two associated bodies. Where r_i and r_j are the velocities of the bodies and s_{iP} and s_{jP} are the velocities of the contact points in relation to their respective bodies, this equation describes the relationship between the velocities of the two bodies at the point of contact. This condition is gotten from the circular limitation, which requires the two contact focuses to correspond. $(s,3) = \dot{r}_j - \tilde{s}_j P \omega_j - \tilde{s}_j P \omega_j$.

$$r_i + s_{iP} + s_{iP} = 0 \dots \dots (2.9)$$

The acceleration constraint for a spherical joint between two connected bodies is outlined in Equation 2.9. The equation establishes a connection between the positions, velocities, and accelerations of the two bodies and the second derivative of the constraint function.

$$\Phi(s,3) \equiv r_{jP} - r_{iP} = r_j + s_{jP} - r_i - s_{iP} = 0 \dots \dots \dots (2.10)$$

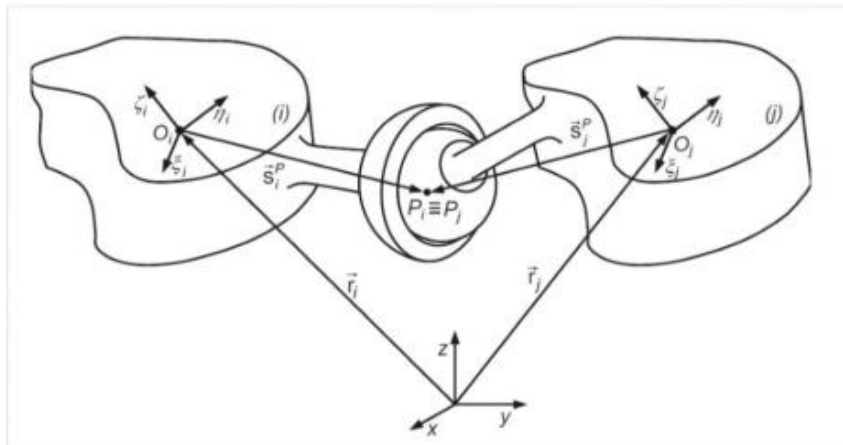


Figure 2. 4 The sphere-shaped joint connects the bodies I and J (Flore. 2016, p. 44).

The equation's 1st derivative. The velocity constraint equation is described in 2.5 (Flore, 2015, p. 44):

$$\dot{\Phi}(s,3) = \dot{r}_j + \dot{s}_{jP} - \dot{r}_i - \dot{s}_{iP} = 0 \dots \dots \dots (2.11)$$

The second subsidiary of the Equation. The equation for the acceleration constraint is shown in Eq 2.5 (Flore. 2016, p. 44):

$$\ddot{\Phi}(s,3) = \ddot{r}_j - \tilde{s}_j P \omega_j - \tilde{s}_j P \dot{\omega}_j - \ddot{r}_i + \tilde{s}_i P \omega_i + \tilde{s}_i P \dot{\omega}_i = 0 \dots \dots \dots (2.12)$$

2.4.3 Equations of Motion

A fundamental procedure for dynamic MBS analysis will be described in this section. A standard Lagrange multipliers technique serves as the foundation for this procedure. The initial form of the Newton-Euler-based equation of motion for a guarded MBS is given below, where the general force vector is g . (Flore. 2016. p. 61)

$$\dot{v} = M^{-1} (g + D^T \lambda) \dots \dots \dots (2.13)$$

Assumed that no valueless inertia (or mass) in the MBS matrix to reverse the matrix for M (Flore. 2016, p. 62).

$$\lambda = [DM - 1D^T]^{-1} (\gamma - DM^{-1}g) \dots \dots \dots (2.14)$$

The following equation can be obtained by using equation 2. In equation 2. (Flore. 2016, p. 62):

$$\dot{v} = M^{-1}g + M^{-1}D^T \{ [DM - 1D^T]^{-1} (\gamma - DM^{-1}g) \} \dots \dots \dots (2.15)$$

The equation $v = M^{-1} (g + D^T \lambda)$ is a component of the dynamic analysis procedure for multi-body systems (MBS) based on the Lagrange method. The constraint ensures that the points on the two bodies that are connected by this equation can be derived from a system of differential algebraic equations (DAE) that is obtained by combining the equation of motion for a constrained MBS with the differential equation at the acceleration level. This equation is used to calculate the system's acceleration vector, which is given the generalized force vector g , the Lagrange multipliers, the mass matrix M , and the damping matrix D .

$$\lambda = [DM - 1D^T]^{-1} (\gamma - DM^{-1}g) (2.16)$$

The best way to work out the Lagrange multipliers (λ) for a compelled multibody framework (MBS) utilizing the reverse of the result of three networks: D , M^{-1} , and D^T represent the system's external forces on the right side of the equation ($- D^T M^{-1} g$). For the purpose of dynamic analysis, the values of the linear set of equations of motion for a constrained MBS can be obtained using this equation.

$$\dot{v} = M^{-1}g + M^{-1} \{ [DM - 1D^T]^{-1} (\gamma - DM^{-1}g) \} \dots \dots \dots (2.17)$$

Using the inverse of the mass matrix (M^{-1}), the force vector (g), and the Jacobian matrix (D), the method for determining the acceleration of a mechanical system (v) is demonstrated in Equation 2.15. An additional term in the equation, denoted by the vector, takes into account system constraints. Assembling the mass matrix, evaluating the Jacobian matrix, and resolving

a set of linear equations are all parts of a larger algorithm for evaluating mechanical system motion.

Figure 2.5 represents a flowchart that gets the calculation of a standard arrangement of the condition of movement. The algorithm is explained in the steps that follow:

- t_0 , \mathbf{q}_0 and \mathbf{v}_0 are starting qualities.
 - The mass matrix, \mathbf{M} , needs to be put together. The Jacobian lattice ought to be assessed.
- It is necessary to hypothesize the restraint equations. \mathbf{V} , the right-hand side of the speed increases, ought not to be set in stone and the power vector \mathbf{g} , ought to be determined.
- To get values for $\dot{\mathbf{v}}$ and λ , the straight arrangement of conditions of movement for a compelled MBS ought to be tackled.

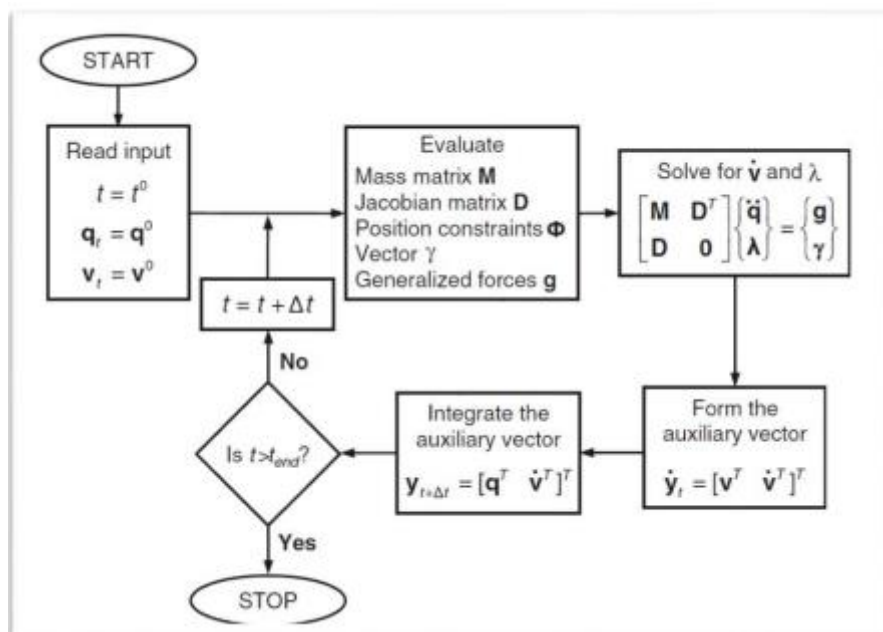


Figure 2. 5 The standard Lagrange multipliers method serves as the foundation for the flowchart for the dynamic analysis of MBS (Flore. 2016, p. 63).

The integration process begins to break the constraint's main equations in simulations. A technique known as the Baumgartner stabilization method may be of assistance in resolving this issue and maintaining control over the constraint break. The primary objective of this approach is to use the following equation in place of the previous differential equations. Figure 2.6 outlines the open and shut circles of controller frameworks (Flore. 2016, p. 64).

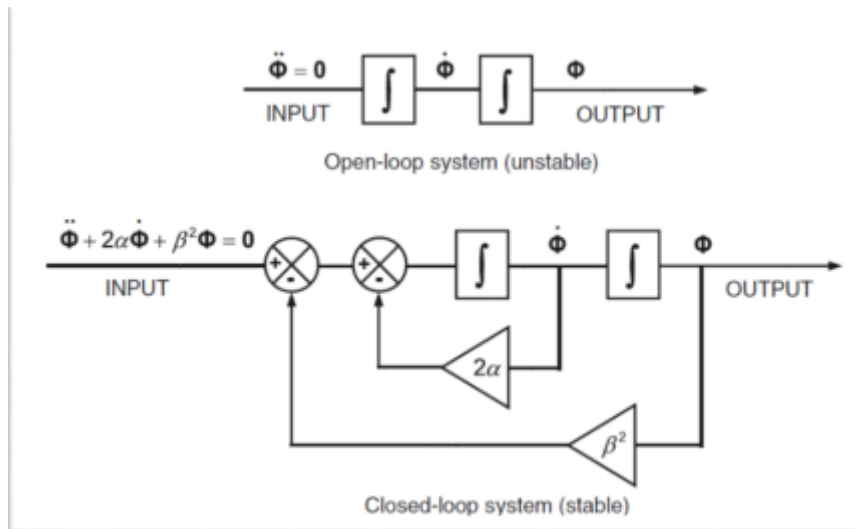


Figure 2. 6 Control systems' open & closed loops (Flore. 2016, p. 64)

2.4.4 Simulation in practice

Likewise, a logical object, simulation must be utilitarian of different regions like showcasing. To reach this goal, an extensive simulation model is required which can fulfill mentioned assumptions. In this chapter, simulators and simulations are discussed from a marketing perspective. Then, we'll talk about used software and customizable models. In the end, a four-bar mechanism's design and outcomes will be obtainable.

2.4.5 Real-time Simulation

A useful tool, real-time simulation provides a comprehensive perspective on the behaviors and responses of various vehicle components in a variety of situations. The real output can be achieved and utilized with a correct simulation design and real inputs and data for all components. The low cost of real-time simulation is yet another striking advantage. To plan and fabricate a genuine machine for getting results, a lot of spending plans must be cost, notwithstanding, with genuine simulation, this point can be reachable.

In addition, a simulated vehicle offers a significant opportunity for parameterization modification and modification. The fact that estimation data can differ from actual results for certain parameters and circumstances is a limitation of simulation. To reduce the number of errors in the final data, the minimum number of assumptions and simplifications should be used

2.4.6 Marketing

There are numerous applications for simulations and simulators in marketplaces. An impressive quantity of organizations that used to test systems and accomplish their advantages. To diminish the testing financial plan, an organization can utilize a test system that is suitable for its examination and concentrate results without burning through a lot of cash. An amazing point of stimulation in selling is parameterization. This indicates that the total spent can be increased if the business wishes to discover new results and modify some machine components. Any kind of change is easy to deal with in a parameterized simulation model. Not just recreating a machine in the test system is less expensive than assembling it, yet additionally substituting a few sections in simulation has no more charge. Because of this, a growing number of businesses are realizing the benefits of simulators and simulations.

2.4.7 Customizable Mode

The primary objectives of developing a generic simulation model are to prevent expenditures and time wastage. Centered on the idea of parameterization, a user can create a simulation model and get the outcomes from it in a variety of ways. Consider a simulation model with a variety of user-selectable assemblies, such as the engine, hydraulic, and power transmission, among others. There are sub-assemblies for each of these assemblies. Given the client's choices, there are an enormous number of mixes for congregations and sub-gatherings which will introduce various outcomes and examinations. An assembly and subassembly schematic of a simulation model is shown in Figure 3. There are a lot of options and combinations for building a single model, as shown in Figure 3, and every change to a sub-assembly will upset the data. As a result, when an operator user made their choices, a new model is created that is completely diverse with others.

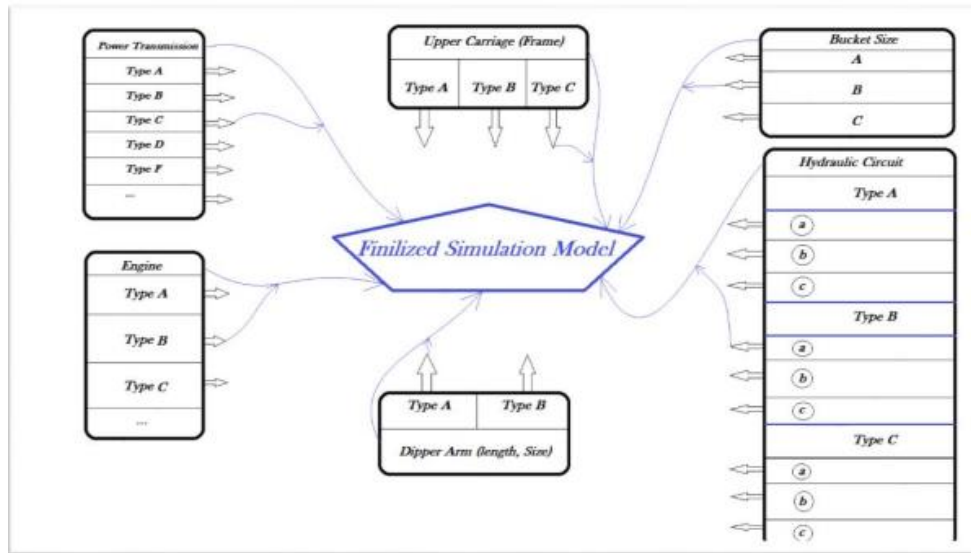


Figure 2. 7 Assembly and sub-assembly-based simulation model

Figure 2.7 shows a reproduction model comprised of various congregations and sub-gatherings, like motor get together, pressure driven gathering, and power transmission gathering. Users can select different combinations of these assemblies and sub-assemblies to create a unique simulation model with different results and analysis. Each sub-assembly has its own code, which is written using software, and when all the assemblies are combined, the simulation model is ready to run. Based on the software that was used to write the code, each sub-assembly has its enigma. Therefore, congregations accumulate and the genuine reenactment model prepared to run.

2.4.8 Employed Software

As previously stated, Vortex software is used to handle all real-time simulation issues in this dissertation. What's more, different viewpoints, for example, designs, and other software designs are utilized. SolidWorks, Blender, Python, and Excel. The obligation in every software will be made sense of the following parts.

2.4.9 Vortex

Using some rational simplifications, Vortex completes the task of real-time simulation. There are no designs while demonstrating in Vortex. Objects in Vortex have no shape so their impact illustrations and designs, associations, and conditions should be characterized in other software. These simplifications are crucial for expediting and simplifying analysis during simulation. Vortex has two points of interaction, one to make a simulation model and the other to run

the model to obtain outcomes. Figures 2.8 and 2.9 display the work point of interaction and vibrant simulation individually.

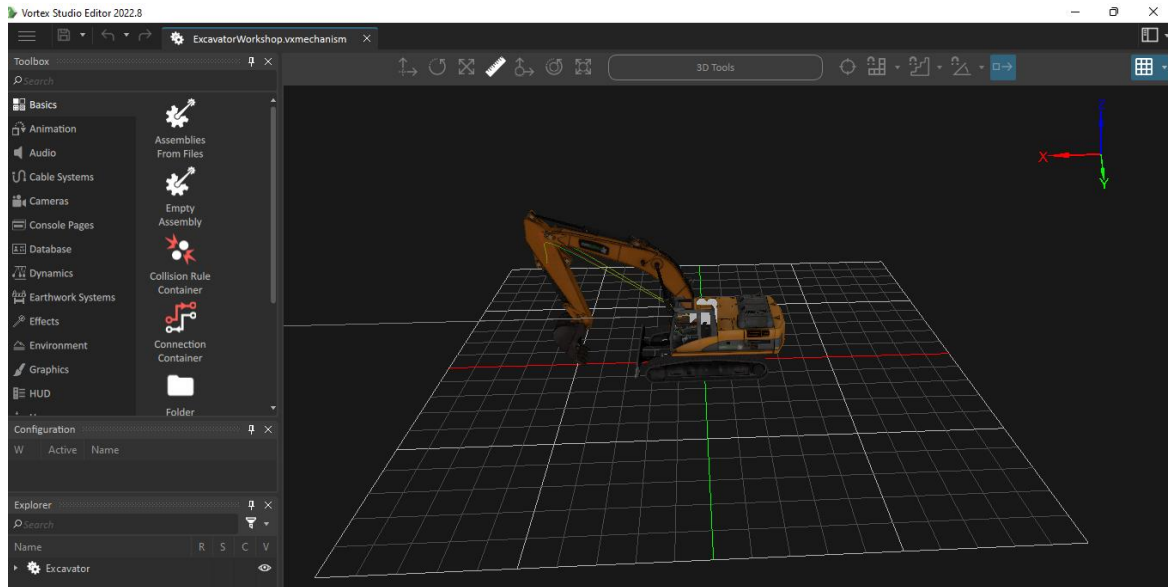


Figure 2. 8 Working interface in the Vortex software for creating an excavator-like model.

The above Figure shows the working interface of Vortex software, which is used to create models such as an excavator. Vortex is a real-time simulation software that simplifies the modeling process by not including graphics, so objects' shapes have to be designed in other software

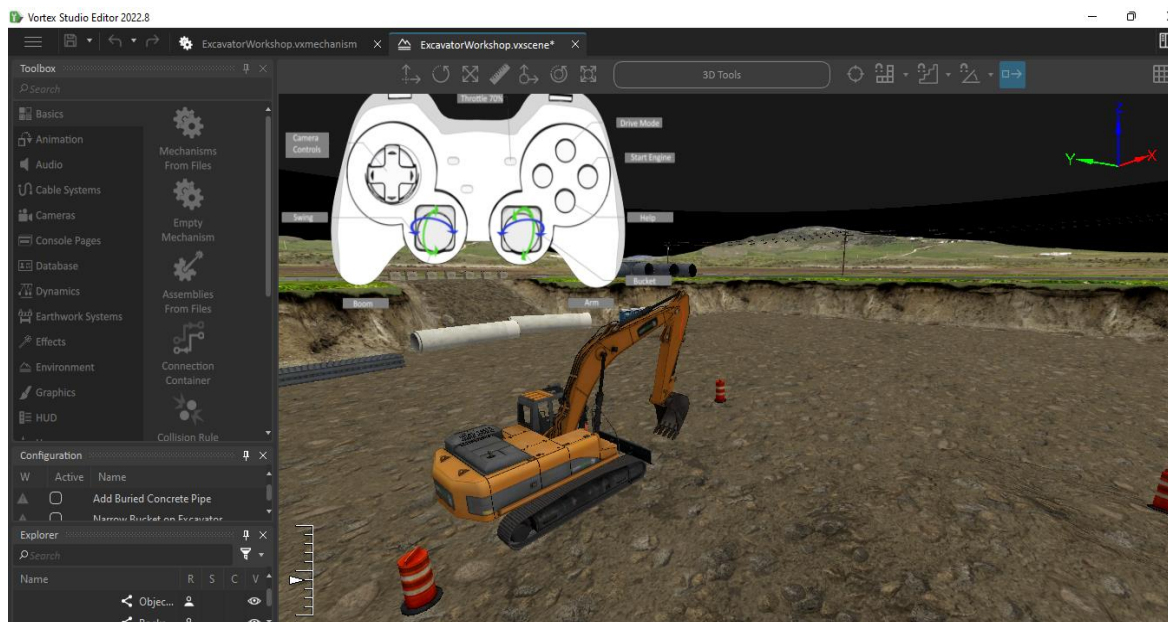


Figure 2. 9 Vortex software's interface for dynamic simulation.

It contained limitations of all graphics of bodies and environment, the hydraulic model outputs, outputs, sensors (virtual), and the data were implemented in the vortex as displayed in figure

2.9. A user can run the simulation model in the dynamic simulation interface to see if it works properly and obtain preferred outcomes and plots. Visuals in the operational interface lack specifics due to simplification, whereas graphics in the dynamic simulation interface are distinct and contain details. The model can be operated by the user using the keyboard or, if necessary, joysticks and a steering wheel. The user can plot diagrams while modeling, as shown in the figure below. These plots show how each component behaves when a simulation model is run. To put it another way, the most important part of the final results in Vortex are the plots. The user can select the specifications they want to see the simulation's behavior based on in the plot tab of the dynamic simulation interface.

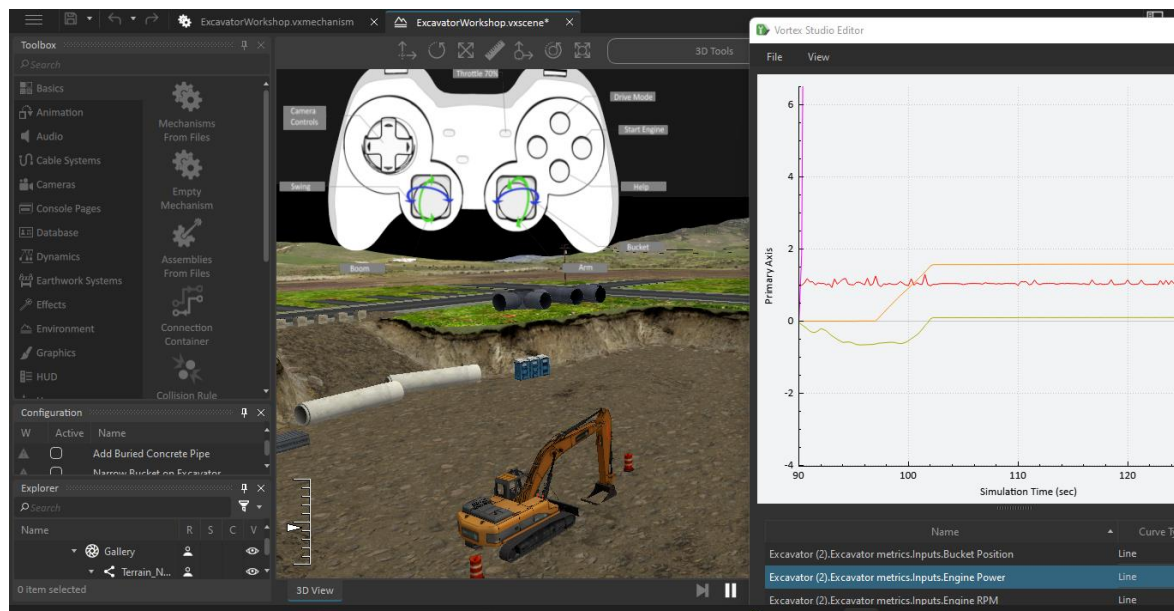


Figure 2. 10 Running scenario with a plot and keyboard control.

The Figure 2.10 shows a picture of a running situation with keyboard control and plots. This image was likely created using CAD software, such as SolidWorks and Blender, which are used for designing and modeling 3D graphics. These software programs are important for creating professional-looking visualizations and collision graphics, which are necessary for accurately simulating real-world scenarios.

2.5 CAD Software – SolidWorks

SolidWorks played a crucial role in modeling an excavator for some of the graphics in my case study because of its precision in 3D assemblies. SolidWorks is the first choice for designing because of its user-friendly interface. In terms of graphics, Vortex can read STL files, but the

3ds file format best fit with Vortex. Therefore, additional graphics-related software is required in addition to SolidWorks.

2.5.1 Blender

Blender is a powerful piece of software that allows for the modeling of environmental graphs, rear-ender graphics, and vehicle parts. It also makes visualization look very professional.

In addition, Blender is simple-to-use open-source software with remarkable structures like faster modeling, rendering of realistic images, forming a real sense of resources (Blender, 2017). Blender can produce files in the 3DS format that can be read by Vortex software. The visualization graphic and the collision graphic are the two kinds of graphics that Vortex requires. The document design for collision illustrations in Vortex is as it were .3ds (Vortex, 2017). It is option realistic is utilized to see the model and the climate, while crash realistic is utilized to show portions of a model that should slam into other body parts and the climate. For instance, the excavator model's bucket requires collision graphics because it must dig the sand. For a simulation model to run smoothly, the crash and perception designs in Blender should always be visible and changing.

A pail representation realistic should be visible in Figure 3.1-a, and the impact and perception illustrations should be visible all the while in Figure 2.3-b. Scales, shades, and surfaces that could be altered with the blender.

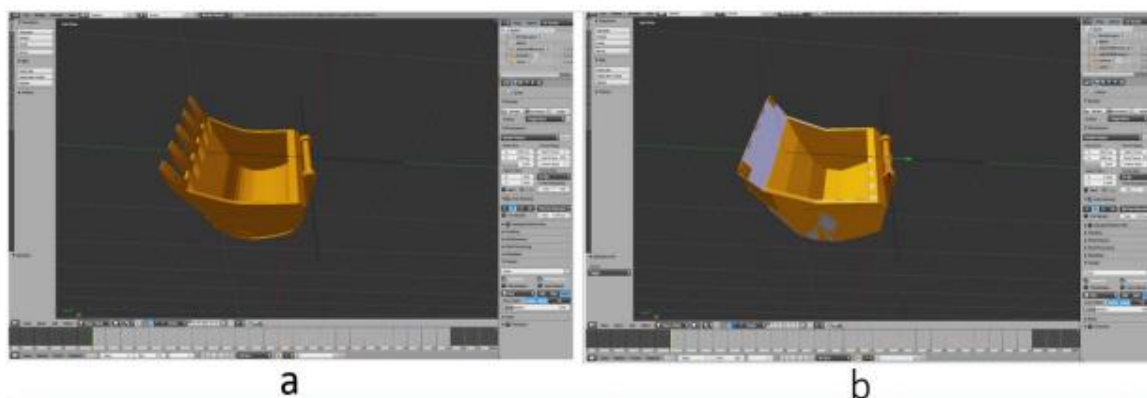
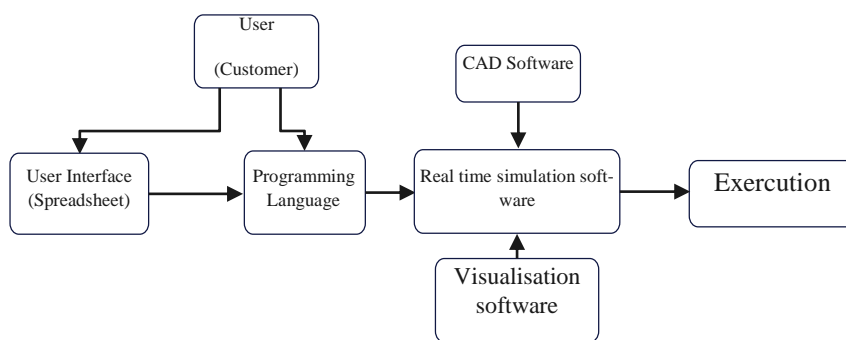


Figure 2. 11 Graphic for visualization; b Graphics for the collision & visualization in Blender

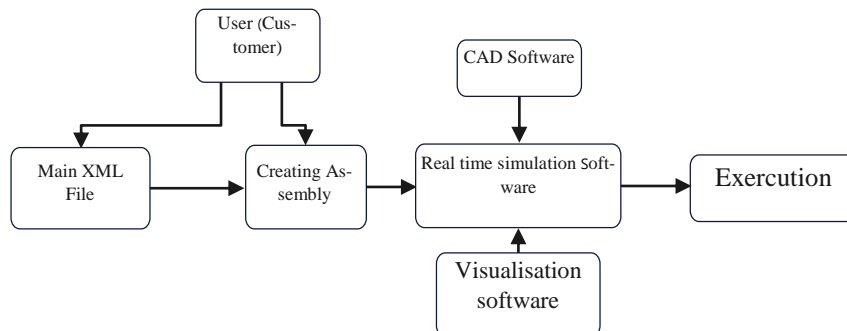
Figure 2.11 shows two images of a bucket in Blender, a software used for creating 3D models and simulations. The main picture (a) will be a representation of the container, while the subsequent picture (b) shows both the perception and impact illustrations of the pail. Impact designs are significant for making the reenactment more sensible by permitting the pail to communicate with the sand it is digging.

2.5.2 Excel and Python

The parameterization target can be reached using Python and Excel software. Python is a coding program that is utilized as an open-source program in a useful information base (Python, 2021). A user can create a vehicle based on their customizations using an Excel file as an interface file, for example, by selecting one of the model's available options. To act as a bridge, code-based software is required for this customization. All selected options can be viewed in Vortex by writing a Python script, and after the simulated vehicle, the desired alternatives can be seen in the Vortex environment. The case study chapter will explain how to create files in Excel and Python code.



(a) Steps in the coding technique



(b) Steps in the coding technique

2.6 Four-Bar Mechanism

Figure 2.21 considers a four-linked model for the purpose of evaluating and determining the real-simulation mechanism. The Lagrangian formulation, position analysis, velocity analysis, and equation of motion will all be explained to determine a mechanism's behavior.

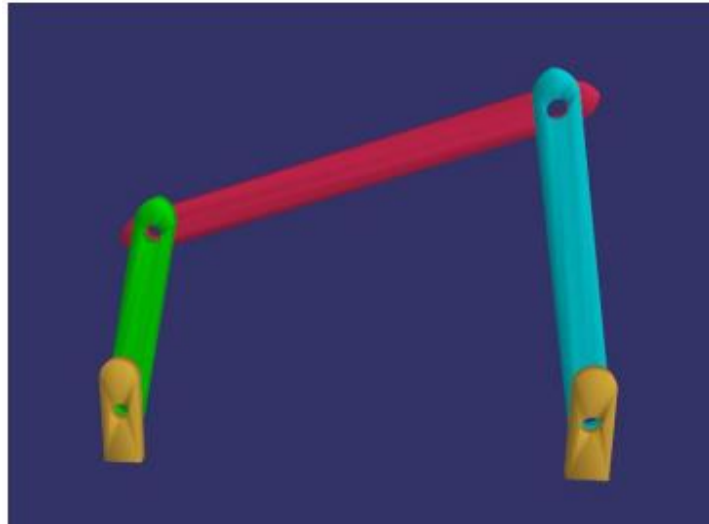


Figure 2. 12 Four-Bar Mechanism

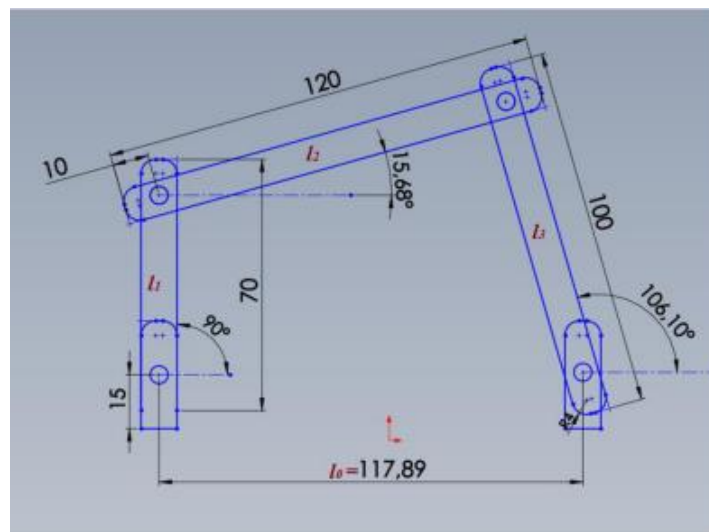


Figure 2. 13 bar analysis mechanism

Table 1.0 exhibits the aspects of 4th -bar component, displayed in Figure 3.2 It is important to note that the values in the table are viewed from center to center.

Table 2. 1 Depicts the values for the parameters.

Parameters	l_0	l_1	l_2	l_3	θ_0'	α_0'	φ_0'	m_1	m_2	m_3
Value (unit)	117 cm	5 cm	101 cm	82 cm	90°	15.8°	106.2°	4 kg	11 kg	9 kg

$$-l_1 \cos \theta - l_2 \cos \alpha + l_0 + l_3 \cos \varphi = 0 \dots \dots \dots (3.1)$$

$$-l_1 \sin \theta - l_2 \sin \alpha + l_3 \sin \varphi \dots \dots \dots (3.2)$$

" $l_1 \cos \theta, l_2 \cos \alpha, l_0 \cos \theta,$ and $l_3 \cos \varphi = 0$ " (3) $l_1 \sin \theta, l_2 \sin \alpha,$ and $l_3 \sin \varphi = 0$ (3.1) The positions of a 4-bar mechanism, a mechanical linkage made up of four rigid bodies connected by four joints, are outlined in the equations 3 and 3.1. The mechanism's stability is ensured by these equations, which relate the bars' lengths (l_0, l_1, l_2, l_3) to the angles between them. The mechanism's position for a given set of parameters is determined by applying the equations, which are derived from the mechanism's geometry.

The position of a 4-bar mechanism is described by the equation $(\theta', \alpha') = \tan^{-1} \frac{l_2(l_1 \sin \theta' + l_3 \sin \alpha', l_0 - l_1 \cos \theta' + l_3 \cos \alpha')}$, in which θ' is the angle between the coupler and the fixed link, α' is the angle between the input link and the fixed link, and φ' is the angle between the output link and the fixed link. The coupler's position for a given input angle can be determined by applying the values of the parameters in Table 2.1, which describe the mechanism's dimensions.

The final equation for the position:

$$\alpha'(\theta', \varphi') = \tan^{-1} \frac{l_2(-l_1 \sin \theta' + l_3 \sin \varphi', l_0 - l_1 \cos \theta' + l_3 \cos \varphi') \dots \dots \dots (3.3)$$

The velocity equation,

$$[\alpha' \varphi'] = \begin{pmatrix} S_1(\theta', \alpha', \varphi) \\ S_2(\theta', \alpha', \varphi) \end{pmatrix} \dot{\theta}' \dots \dots \dots (3.4)$$

Kinetic energy:

$$T = \frac{1}{2} (m_1 l_{c1}^2 (\dot{\theta}')^2 + I_1 (\dot{\theta}')^2) + \frac{1}{2} (m_2 l_1^2 (\dot{\theta}')^2 + l_{c2}^2 (\dot{\alpha}')^2 + 2 l_1 l_{c2} \cos(\theta' - \alpha') \dot{\theta}' \dot{\alpha}') + \frac{1}{2} (m_3 l_{c3}^2 (\dot{\phi}')^2 + I_3 (\dot{\phi}')^2) \dots \dots \dots (3.5)$$

Where T is the kinematic energy, m is the mass, I is inertia and l/2.

Potential energy:

$$V = m_1 g l_{c1} \sin \theta' + m_2 g l_1 \sin \theta' + l_{c2} \sin \alpha' + m_3 g l_{c3} \sin \phi' \dots \dots \dots (3.6)$$

V is the expected energy and g is the ground speed increase.

The conclusive motion equation:

$$M(\theta') (\dot{\theta}')^2 + V(\theta', \dot{\theta}') = \tau \theta \dots \dots \dots (3.7)$$

For the four-bar mechanism, contains the constraint equations, Jacobian matrix, Newton variance for point and velocity exploration, and a formula for $M(\theta') (\dot{\theta}')^2 + V(\theta', \dot{\theta}') = (3)$.

The potential energy of a 4-bar mechanism, a mechanical system with four rigid links connected by revolute joints, is described in the text. The mass of each link is multiplied by the gravitational acceleration and the height of its center of mass above a reference level to obtain the potential energy. The sum of the gravitational potential energies of each link is used to calculate the potential energy. The equation demonstrates that the potential energy is influenced by the lengths of the links, which determine the distances between the joints, and the angles of the links, which have an effect on the heights of the centers of mass.

2.6.1 The outcomes of 4-bar mechanics:

The simulation's outcome can be seen below, following modeling in Vortex (figure 2.23);

Figure 3.5 shows the precise speed in the y and z course for the center connection in x while reenacting.

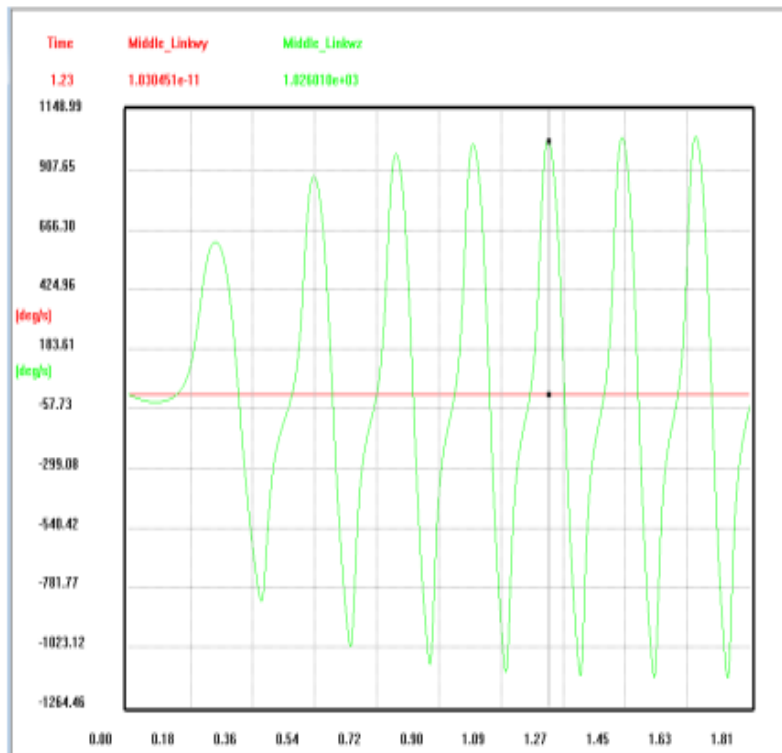


Figure 2. 14 The Middle-link's angular velocity in the y and z directions.

The middle-link's rate of change in angle as it moves in the y and z directions is referred to as its angular velocity in the y and z directions. Figure 2.25 shows how this rakish speed changes after some time during reproduction. This data is significant for figuring out the movement and conduct of the 4-bar-instrument. It is essential to comprehend how this 4-bar mechanism moves and acts to comprehend how angular velocity changes over time during the simulation.

Chapter 3

Methodology

3.1 Case study – An excavator

With the help of parameterization, executing a few possible choices in the excavator model, figure 3.1, should be possible. The hydraulic circuit and bucket of the excavator model can be modified in certain ways.

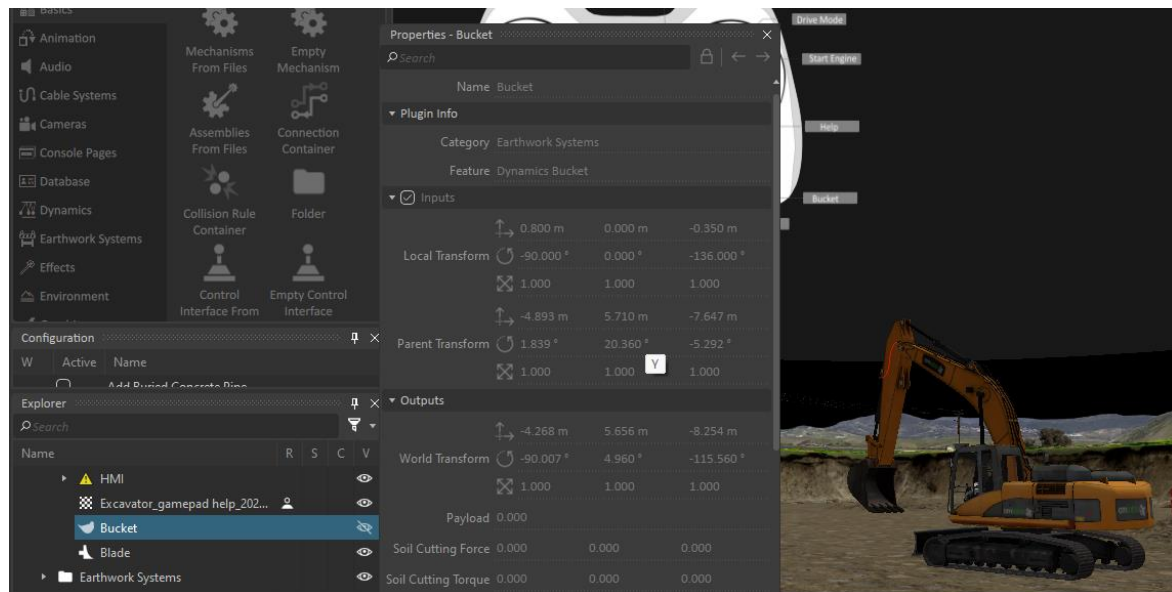


Figure 3. 1 The excavator, complete with its two customization options.

The excavator is used to dig up and lift any type of geologic material, such as soil, sand, rocks, construction debris, etc. The bucket, dipper arm, and main boom are moved by the driver using six main cylinder pistons (see Figure 3.1). There is no need to drive very quickly at high speed because of how the various types of excavator work. So there is no stuff record in this particular earthmover and with customization in water-powered circuits, variable velocities to lift the primary blast or the scoop arm are reachable. The bucket and the hydraulic circuit are the two main areas on which this thesis focuses on customizing. There are currently three different buckets and three different cylinder pistons that are stirring the dipper arm.

3.1.1 How an Excavator Works

In industries, excavators are accustomed to working in risky conditions and on the lopsided ground. They don't need to move quickly because they always move construction waste, sand, soil, and particles to another moving vehicle like a Camion. However, according to Volvo (2012), some excavators can travel at speeds of up to 8.9 kilometers per hour off-road and up

to 35 kilometers per hour on the road. Among the well-known manufacturers of construction equipment, including excavators, are Caterpillar, Komatsu, Volvo, Hitachi, and Libeler. Additionally, there are numerous applications for excavators: Volvo, 2012), figure 3.2, digging, ripper application, establishment penetrating, material dealing with, high destruction, long reach, etc. Additionally, there is a company known as Sandvik that manufactures a wide range of merchandise, in the assessment of drilling rigs and mining of resources (Sandvik, 2017).

The simulation model for this project is based on products manufactured by Sandvik. Manufacturers of excavators constantly strive to improve their products by taking into account several parameters, many of which are entangled (Caterpillar, 2017). This dissertation discusses the hydraulic system and the bucket's size, which is unswervingly linked to the number of particles dug each time. Moreover, information and data in regards to buckets were created by a spearheading organization, Volvo Organization in Reference section 6. To have a bucket with a high moving speed in its blasts, how much oil goes into the chambers, and does the stream rate ought to increase? The work cycle time will likely be significantly reduced simultaneously if the bucket is large. However, there are two significant considerations to make: To begin with, from a fuel-utilization perspective, this present circumstance can support fuel utilization too. Second, additional hydraulic circuit parameters should be taken into consideration when using a larger bucket or a very potent valve in a hydraulic circuit. Put in an unexpected way, because chosen boundaries for a vehicle generally speaking are entwined together, by changing a boundary, different boundaries must be altered or possibly patterned.

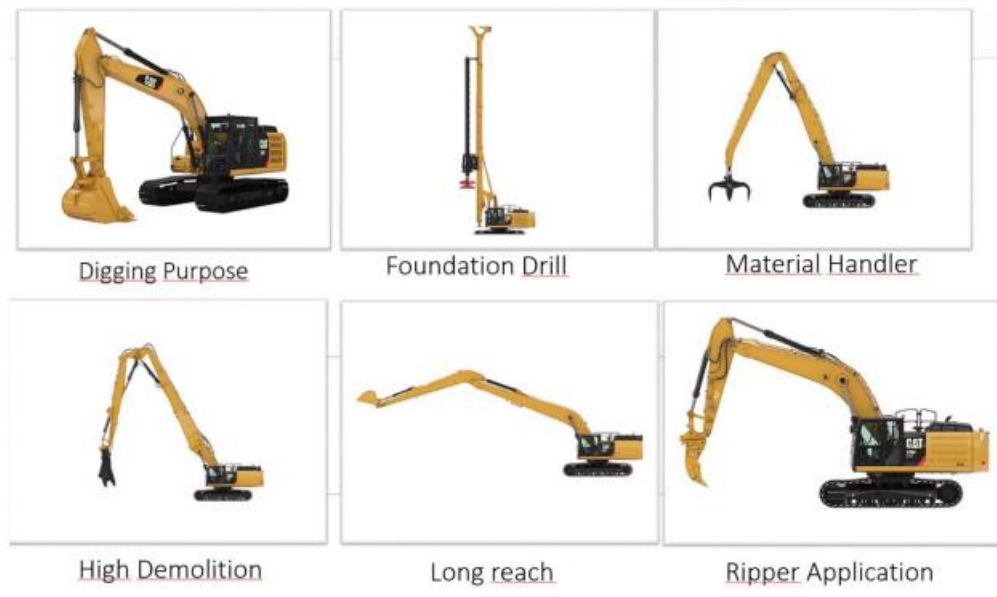


Figure 3. 2 Typical applications for the excavator (Caterpillar, 2018)

3.2 Stimulated Industrial Vehicle

A made-to-order model is primarily used to demonstrate a vehicle's versatility in various situations. It indicates that the capability of a vehicle can be determined using a customizable simulation model. Besides, it permits to make of a pragmatic correlation between a model of various choices, information, and chunks and readies a standpoint to improve the choices of the model for every particular circumstance. From a marketing perspective, this idea presents a huge opportunity because it permits a customer to choose from different options and cater to their target audience.

3.3 Editable Parameters

Vortex makes four XML (Extensible Markup Language) design records convertible which the executable document, MVS record, peruses them to run a model. One of the made XML is a record that has the primary information about the model in it. Other XML records are material library documents, critique records, and world records. For example, sceneries for cameras and lights are saved in the world file. The excavator's parts and parameters must first be identified before an editable simulation model can be created. The specifications of each component of the excavator must be carefully considered. To get an absolute model of a vehicle, there are various taxonomies to use. Portions of a vehicle can be categorized as an illustration of a classification. The undercarriage, upper carriage, main boom, dipper arm, bucket, and attachments are the excavator components in this instance.

A vehicle's systems and interactions can also be categorized. The hydraulic system, the engine and force system, and the parts system, for instance, are all components of the excavator. Every framework has its sub-frameworks and every one of them is associating together. In this dissertation, dual components are deemed user-selectable and user-operable. Digging is one of the primary functions of the excavator's head, as shown in Figure 3.2. The hydraulic system, which operates through the excavator and provides power for any excavator movement, is another significant system. The process of making a hydraulic system as a whole customizable is difficult and, in some cases, unreliable. On the other hand, certain parts of the system can be edited, allowing for a model with a variety of options for movement velocity and powerful arms to dig. These dual systems will be examined in subsequent chapters.

3.4 System for lifting and a bucket

The primary function of a bucket-type excavator is to excavate and transfer soil materials. At times of an earthmover operational time, the significant aspect is saving time, operation in a spot with many restricted spots, in this manner having a chance to use a couple of sorts of cans appears to be viable and helpful. For this situation there are three sorts of containers with assortment aspects and loads, figure 3.3.



Figure 3. 3 The kinds of buckets that are used for customization.

In Vortex Modeler, the bucket has some parameters that can be set in the bodies' sub-tree. There are Visualization Graphics and Collision Graphics for each part, both of which interact with the graphics of the bodies in the Graphics sub-tree. In addition, the position and mass of each component must be determined. The Vortex Modeler user boundary, which includes Model Tree, is shown in Figure 3.4.

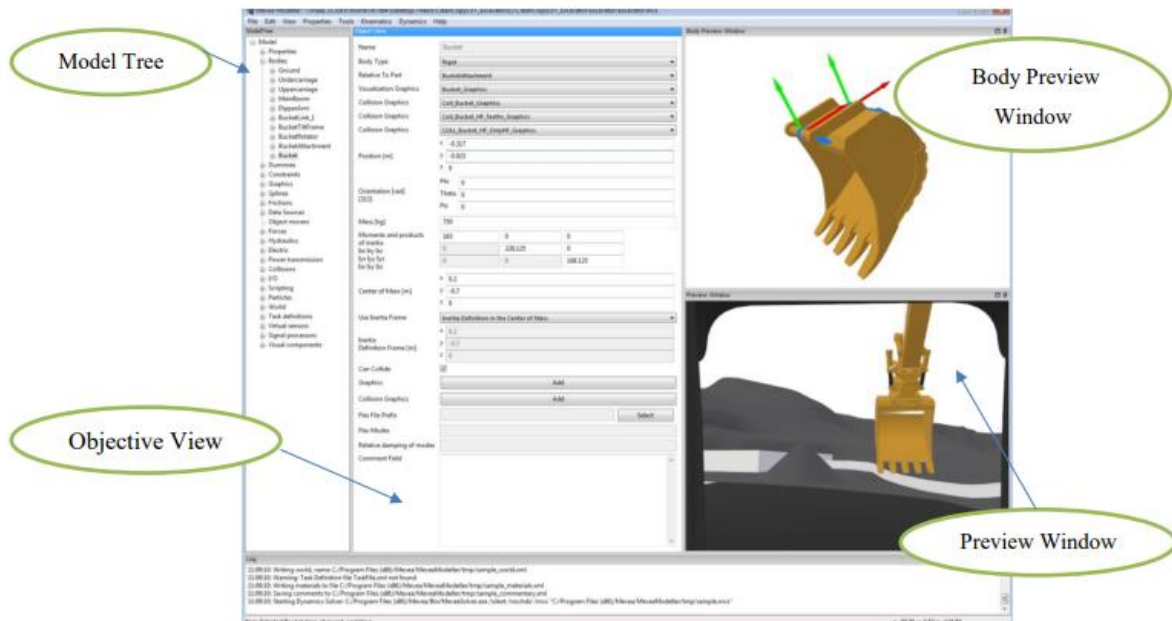


Figure 3. 4 portions of the interface for Vortex Modeler.

As referenced, each body and molecule have two sorts of designs. The body is shown to the user in a visualization graphic, and any collision between the parts and the environment is shown in a collision graphic. For instance, the function of digging takes place when a bucket and particles (like sand or gravel) collide, as defined by Vortex.

Table 3.1. Portrays boundaries that need to change to get an adjustable can. Figure 3.4 shows how the bucket's data is set in the Objective View window, with its graphics were chosen on the graphics in the Graphics sub-tree that have already been calculated and well-defined.

Table 3. 1 Parameters that need to characterize every three kinds of the buckets.

The BUCKET of the EXCAVATOR		
Parameters (unit)	Position	m
	Orientation	rad
	Mass (weight)	kg
	Moment of inertia	kg.m ²
	Centre of Mass	m

Analysis

A-Comparison between three kinds of buckets of the excavator on:

- Total cycle time (s)
- Fuel consumption (ml)
- The power transfer of the excavator's primary motor

B-Specifications of cylinder-pistons:

- Values of angular velocity for the dipper arm

Results

After the excavator model has been created, it may be run in the dynamic simulation interface and compared to other kinds of custom-made models. A user can execute by selecting her or his preferred parameters. The following is the procedure for running the model:

The model may be run in the dynamic simulation interface and the results are shown. Plots and outcomes of bucket selections, as well as hydraulic circuit selections and will be discussed in the next chapters. The excavator comprises an original model that has a bucket and hydraulic circuits that are developed together; however, this study, it is tried to produce various blends

to examine the impacts of these changes and find out if a simulated model might be more optimal.

3.4.1 Combination and comparison for the Bucket's examination

Three types of buckets are involved; each were three sets of parameters that were tested.

The capacity and weight of the bucket have an impact on its functions, working cycle time, and fuel consumption. The bucket specifications are listed in the table.

The most notable specifications during bucket customization will be illustrated and described in the figures below. When changing buckets is the primary goal, four Factors will be measured and addressed. The same test is run three times for three distinct buckets, with the findings and charts discussed in the following paragraphs.

Table 3. 2 **Bucket A** (Small bucket)

Parameters	Units	Values
Position	m	(0.31, 0.13 .0)
Orientation	rad	(0.0, 0.0, 0.0)
Centre of Mass	m	(0.4, 0.8, 0)
Mass	kg	316
Volume	m^3	0.317
Moments of inertia (I_{xx} , I_{yy} , I_{zz})	$kg.m^2$	(187, 229, 169)

The overall cycle duration and global y position for all three buckets are the first items to be considered. Figure.3.5. below shows the outcome for the smaller bucket.

The "Bucket Position vs. Time" graph is a visual representation of the position of the excavator bucket over time. The x-axis represents time and the y-axis represents the position of the bucket. The graph is generated by the real-time simulation of the excavator bucket and can be used to determine if the bucket is moving as desired.

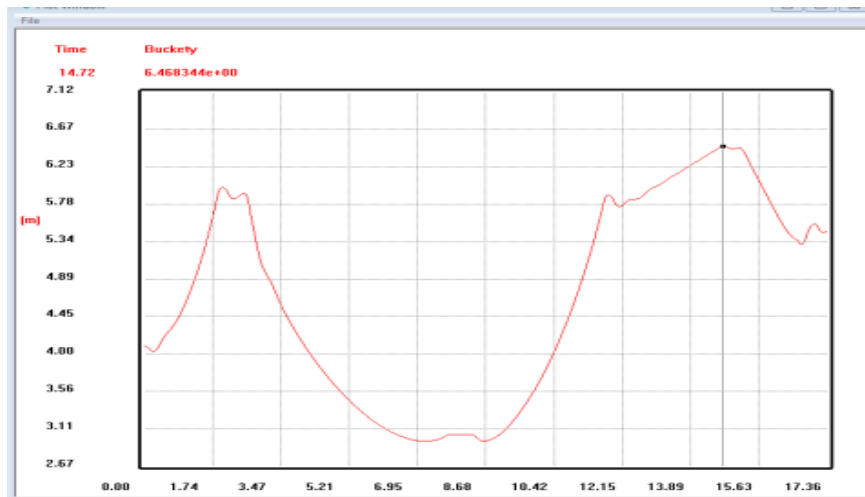


Figure 3. 5 The global y positions of the small bucket and the total time are shown

The graph can be used to analyze the following:

1. Bucket Movement: The shape of the graph can be used to determine the pattern of the bucket's movement, such as if it is moving in a smooth or jerky manner.
2. Bucket Accuracy: The graph can be used to determine the accuracy of the bucket's movement. If the bucket is not moving to the desired position, the graph will show deviations from the target position.
3. System Performance: The graph can be used to determine the performance of the control system, as it will show if the system can control the bucket's movement accurately and efficiently.

By analyzing the "Bucket Position vs. Time" graph, engineers and operators can improve the performance of the excavator bucket and ensure that it is efficiently and safely operating. During the test, the most important factor, fuel consumption, was tracked.

The graph depicts the fuel usage of the excavator while utilizing the little bucket. As can be seen, the final number, which is the most important prerequisite for fuel consumption, is $4.8e^{-2}$.

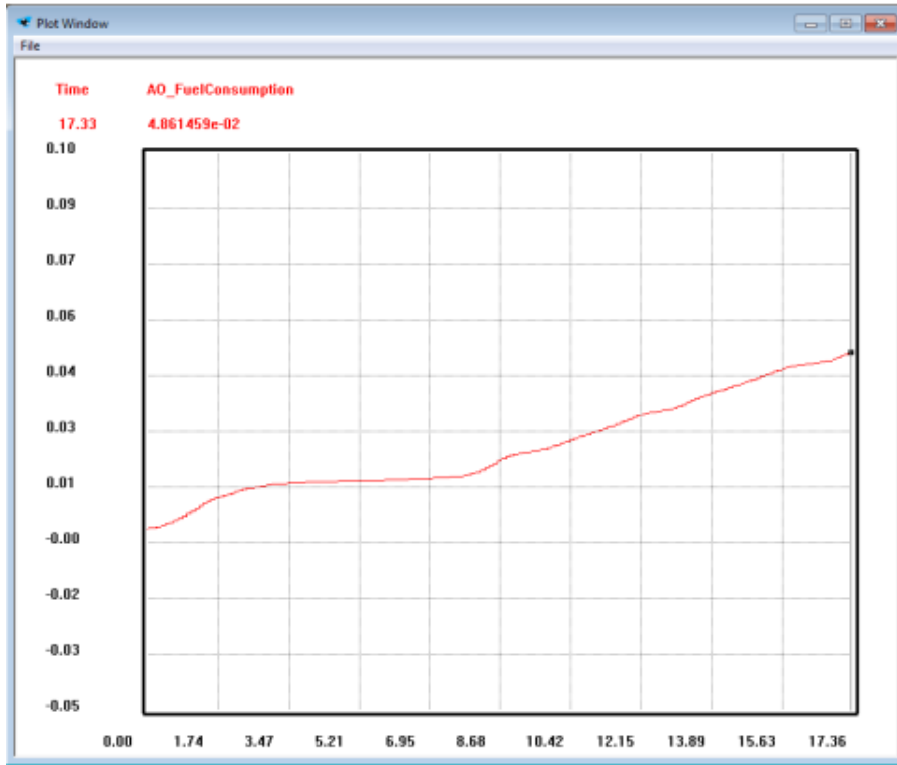


Figure 3. 6 Gasoline utilization for the excavator utilizing bucket A

Understanding the excavator's movement and performance is made easier when we know how the angular velocity changes over time. Next specification is the angular velocity in global y direction for the dipper arm cylinder during the test. As figure 3.7 shows, the amount of the angular velocity for the small bucket has a small-scale of oscillation around a certain value. It can help identify any potential issues or areas for improvement, as well as provide insights into how efficiently and effectively it is operating.

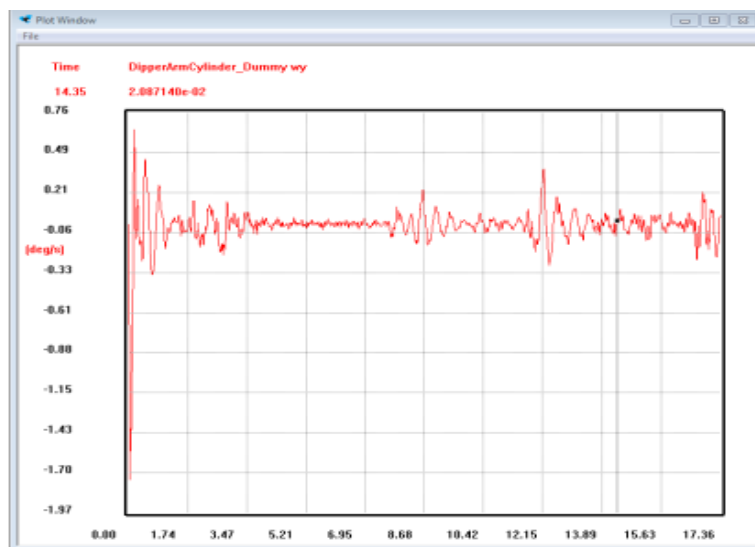


Figure 3. 7 Shows the angular velocity of bucket A in the y-axis for the dip arms cylinders.

Angular velocity values for the dipper arm cylinder in the global Y direction and also total working cycle time while employing the small hydraulic circuits are shown below.



Figure 3. 8 The values of the cylinder-piston velocity of the dipper arm in a small hydraulic circuit

When employing small hydraulic circuits, the figure displays the global Y-direction angular velocity readings for the dip arm's cylinders and the overall operating cycle duration. Understanding the dipper arm's performance and its capacity to lift and lower loads require this information.

The values of the cylinder-piston velocity of the dipper arm in a small hydraulic circuit are depicted in Figure 3.8. This number tells you how quickly the hydraulic system's cylinder piston is moving.

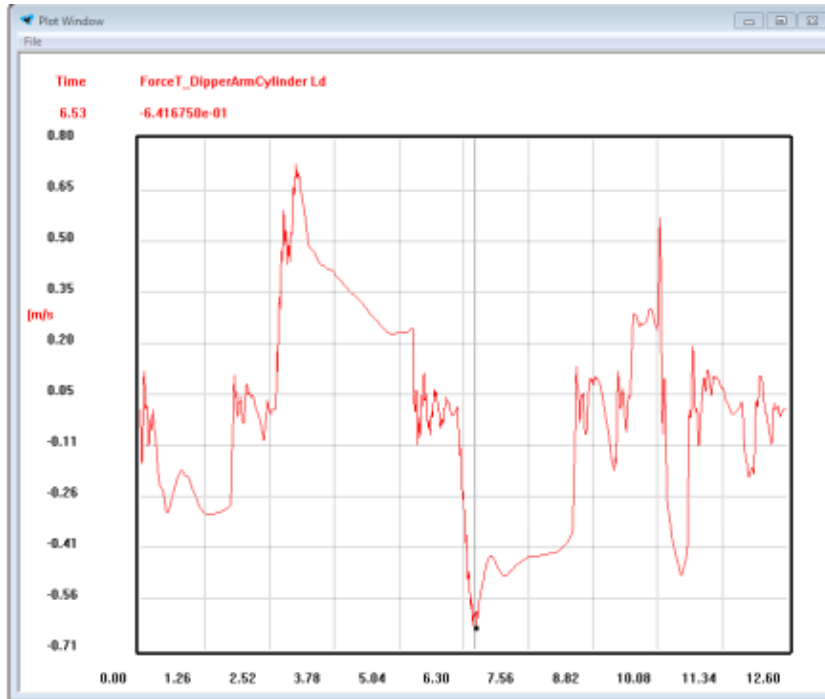


Figure 3. 9 Analyze the hydraulic circuit's performance

The data in Figure 3.18 can be used to analyze the hydraulic circuit's performance because it shows how quickly a small hydraulic circuit's cylinder and piston are moving. One can determine whether there are any problems with the performance of the system, such as inefficiencies or malfunctions, by comparing these velocity values to the expected or desired values.

Table 3. 3 Bucket B (The medium bucket)

Parameters	Units	Values
Position	m	(0.31, 0.13 .0)
Orientation	rad	(0.0, 0.0, 0.0)
Centre of mass	m	(0.4, 0.8, 0)
Mass	kg	450
Volume	m^3	0.512
Moments of inertia (I_{xx} , I_{yy} , I_{zz})	$kg.m^2$	(189, 229, 169)

Results for the Bucket B (Medium Bucket):

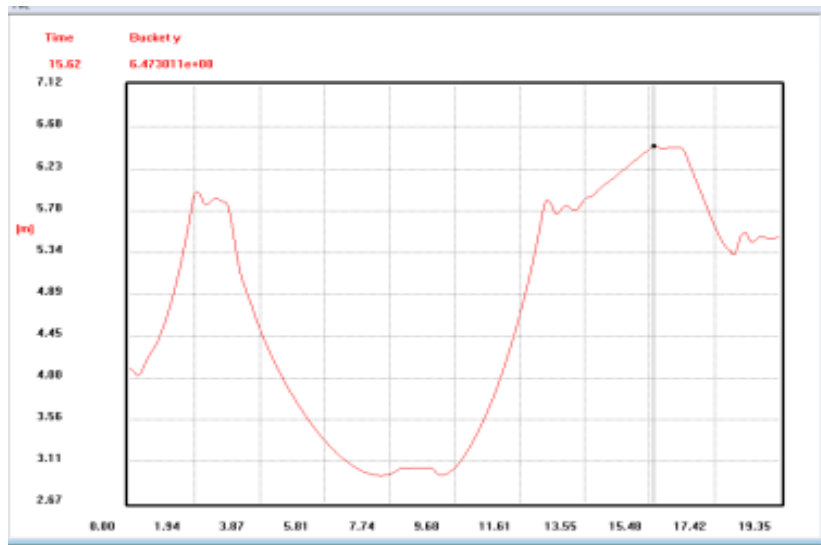


Figure 3. 10 The total duration and locations of Bucket B are shown

Output: Fuel Consumption:

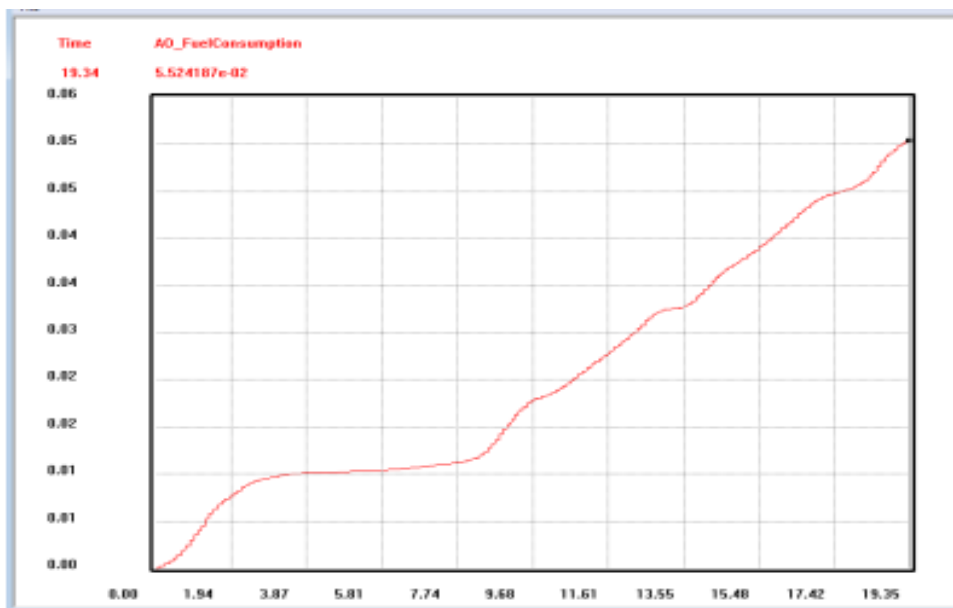


Figure 3. 11 Fuel consumption graph for the medium bucket (Bucket B).

Table 3. 4 Bucket C (The Big Bucket)

Parameters	Units	Values
Position	m	(0.31, 0.13 .0)
Orientation	rad	(0.0, 0.0, 0.0)
Centre of mass	m	(0.4, 0.8, 0)
Mass	kg	750
Volume	m^3	23
Moments of inertia (I_{xx} , I_{yy} , I_{zz})	$kg.m^2$	(183, 229, 169)

Results for the Bucket C (Big Bucket):

The "Bucket Position vs. Time" graph is a visual representation of the position of the excavator bucket over time. The x-axis represents time and the y-axis represents the position of the bucket. The graph is generated by the real-time simulation of the excavator bucket and can be used to determine if the bucket is moving as desired.

The graph can be used to analyze the following:

1. **Bucket Movement:** The shape of the graph can be used to determine the pattern of the bucket's movement, such as if it is moving in a smooth or jerky manner.
2. **Bucket Speed:** The slope of the graph can be used to determine the speed of the bucket. A steeper slope indicates a faster bucket movement, while a shallower slope indicates a slower bucket movement.
3. **Bucket Accuracy:** The graph can be used to determine the accuracy of the bucket's movement. If the bucket is not moving to the desired position, the graph will show deviations from the target position.
4. **System Performance:** The graph can be used to determine the performance of the control system, as it will show if the system is able to control the bucket's movement accurately and efficiently.

By analyzing the "Bucket Position vs. Time" graph, engineers and operators can optimize the performance of the excavator bucket and ensure that it is operating efficiently and safely.

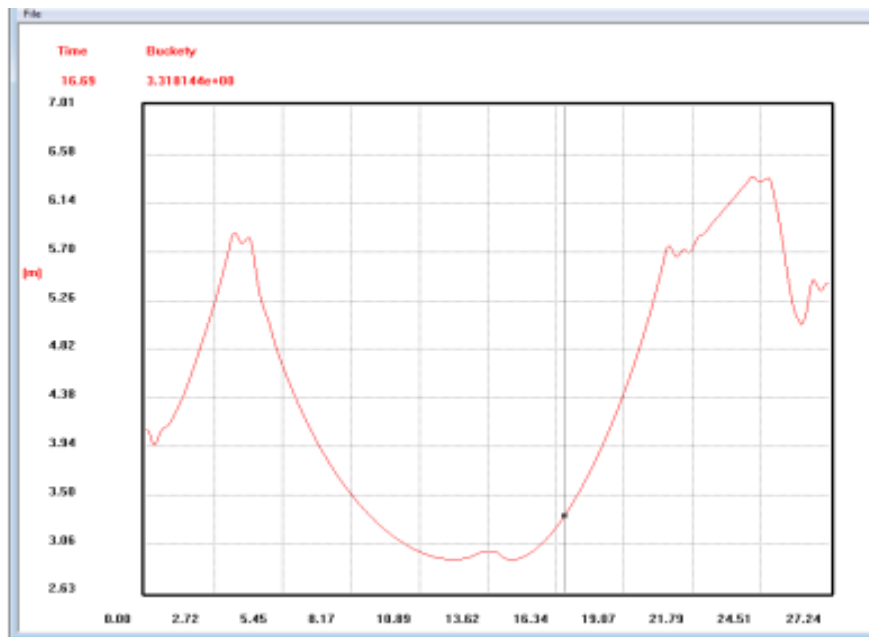


Figure 3. 12 : Overall time and bucket C locations in global y-axis.

Output: Fuel Consumption:

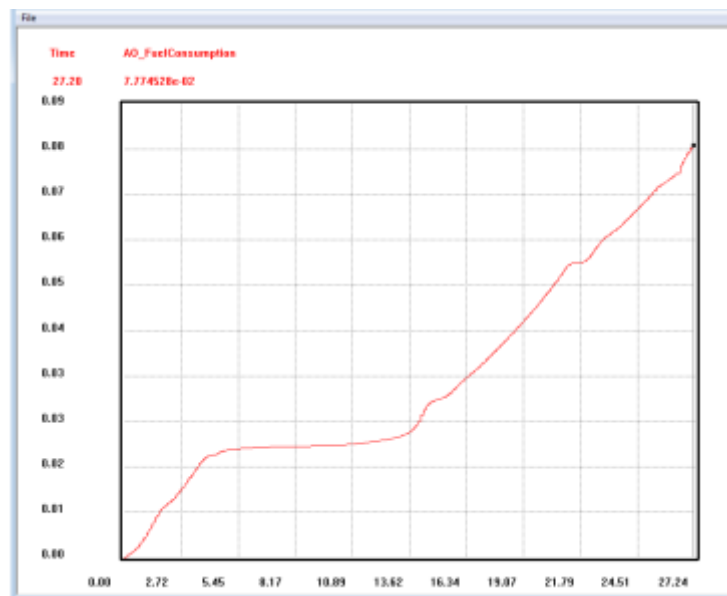


Figure 3. 13 Excavator fuel usage with bucket C

3.4.2 Combination and comparison for the Bucket's examination

There are three types of buckets and three sets of parameters that have been tested: working cycle time, fuel consumption, and the impact of the bucket's weight on its functions. In the table above, bucket specifications are shown.

In the accompanying figures, the most astounding particulars during the customization of containers will be shown and talked about. When the primary objective is the replacement of buckets, four specifications will be measured and discussed. A similar test is rehashed multiple times for three unique containers and the outcomes and plots will examine in the following passages.

The total cycle time and global Y Position for each of the three buckets are the first considerations.

The graphs of the results for the small bucket are shown in Figure 1. The graphs of the results for the medium bucket and the big bucket are shown in Figures 2 and 3.

Figure 1 shows that the test took 17.36 seconds to complete for the small bucket.

Finally, the test tracked fuel consumption, one of the most important factors. The graph of fuel consumption when using the excavator's small bucket is shown in Figure 4. As it is clear, the last worth as the most critical detail for the fuel utilization is $4.86e-2$.

Chapter 4

4 Results and Analysis

4.1 Evaluation of various buckets for use

The mined results tested with three different buckets for the excavator will be examined in this section. In certain circumstances, working process duration is one of the most basic viewpoints that manufacturing organizations and their clients focus on it. Figure 6 shows a comparison between three bucket types when they were used in the same tests that were discussed in chapter three (3).

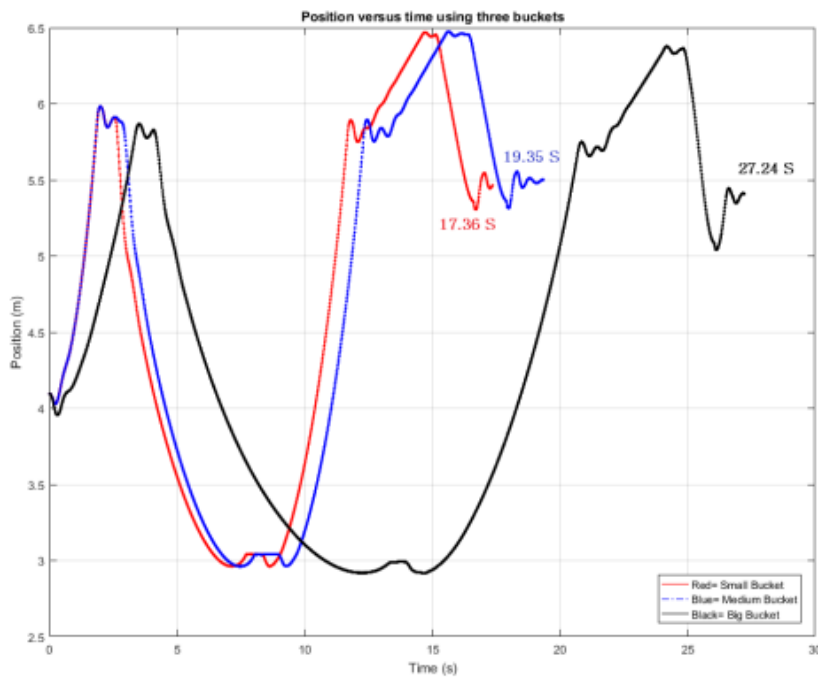


Figure 4. 1 A single test's graphs of three distinct bucket types.

About the position vs. time graph of an excavator bucket:

The excavator's bucket is controlled by the hydraulic system, which uses cylinders to move the arm and bucket. The operator controls the movement of the bucket using multiple levers (joystick) that direct hydraulic fluid to the appropriate cylinder.

I have found out that when the control signal on the joystick remains the same, the valve characteristic curve can affect the cylinder's movement. Here are 3 basic types of the valve characteristic curve:

- Linear valve characteristic curve: The flow rate through the valve increases linearly as the valve is opened. This type of curve is often used in simple hydraulic systems with few or no changes in load or pressure.
- Equal percentage valve characteristic curve: The flow rate through the valve increases exponentially as the valve is opened. Applications that require accurate flow control over a broad range of valve positions benefit from the use of valves with equal percentage characteristic curves, which provide more precise control at both high and low flow rates.
- Quick opening valve characteristic curve: The valve opens rapidly when the control signal is first applied, and the flow rate remains constant as the valve is opened further. This type of curve is often used in emergency shut-off valves or in systems where quick actuation is required to prevent damage or injury.

The results of tests on three different buckets for an excavator. The working cycle time, which is a crucial aspect for manufacturing businesses and their customers, is the focus of the analysis. The medium bucket takes 11.5% longer to complete the test than the small bucket, as shown in Figure 4.1, when compared to the other three bucket sizes.

The text makes sense of that while utilizing a bigger can on an excavator, it requires 40% greater investment to finish the job contrasted with utilizing a medium-sized can. This suggests that the excavator's efficiency may be affected by the bucket's size

The ultimate time required to complete the test for large, medium, and small buckets is depicted in Figure 6. During the use of the medium can, the buckets need an additional time 11.5 percent more than utilizing the little container. Additionally, the excavator takes 40% longer to use the big bucket than the medium bucket.

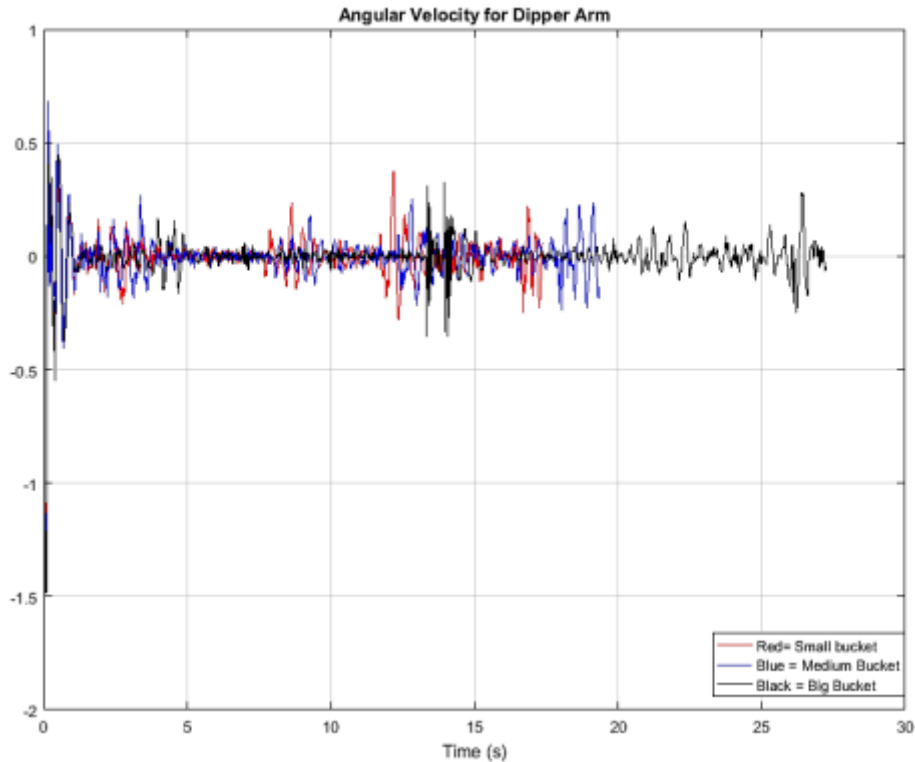


Figure 4. 2 Plots for the precise speed of the dip arm

Figure 4.2 shows plots for the precise speed of the scoop arm while utilizing three different pail sizes. The fact that the angular velocity did not significantly shift when buckets were switched indicates that the dipper arm's forces were unaffected by the change in bucket size

4.1.1 Consumption of fuel for three tests using three different buckets.

The factor that has been examined is, last but not least, fuel consumption. As Figure 6.1 show-cases, the last worth of fuel utilization for the medium container is $5.52e2$ which is 13% more than this incentive for the little pail. However, the big bucket consumes 7.77 gallons of fuel per hour, 40.6 % more than bucket B.

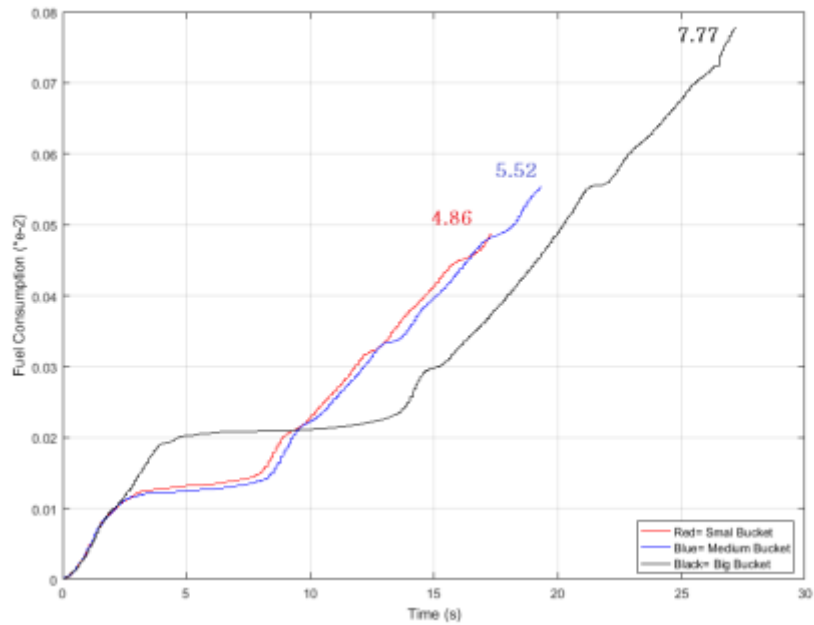


Figure 4. 3 Fuel consumption in three kinds of buckets.

In the examination, involving the big bucket for the excavator allows you a superior break to buzz constituent part of every cycle than utilizing average or little pails. The huge container has a volume that is multiple times bigger than the little pail and 4.4 times bigger than the medium can. Figure 4.1 depicts the total cycle period, the extreme number of elements that could be lifted in a single trial, and energy ingestion for each vessel type.

The figure 4.4 explains that using a big bucket for an excavator allows for more particles to be lifted in each cycle compared to using medium or small buckets. Bucket C is 4.3 times greater than bucket B can and 8 times greater than bucket A. However, using the big bucket also increases total cycle time and fuel consumption, so the choice of bucket size depends on the specific demands of the task.

Based on three factors, Figure 4.4 compares three excavator bucket types: complete process duration, greatest measure of particles lifted in each attempt, and fuel utilization. The figure shows that while utilizing a major pail can lift more particles, it likewise increments all out process duration and fuel utilization. Then again, the medium pail has a limit that is 59% more than the little can, with just a 13% increment in fuel utilization and a 11% increment in all out chance to achieve the mission. The requirements of the customer, such as the quantity of particles to be transferred, determine the type of bucket that is selected.

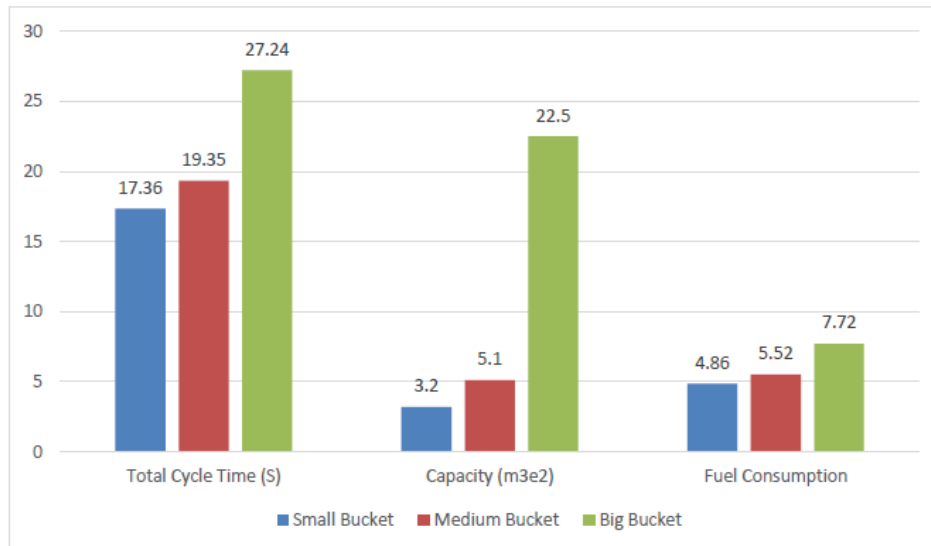


Figure 4. 4 Comparison of three bucket types' specifications for an excavator

Even though big bucket preparers offer the opportunity to transfer more particles, they will also significantly increase total cycle time and fuel consumption, as shown in Figure 4.7. However, while the medium bucket has a capacity that is approximately 59% greater than that of the small bucket, its fuel consumption is only 13% greater when compared to that of the small bucket. Additionally, the complete chance to achieve the mission for the medium can is 11% more prominent than the little one. It should be noted that choosing an excavator bucket is highly dependent on customer preferences. For example, when the number of constituent parts to be moved, is particularly important, using a large bucket may be feasible.

4.1.2 Evaluation of various hydraulic circuits for use

Analysis for the use of various hydraulic circuits UNDERGRAD The text analyzes the excavator's behavior when it uses various hydraulic circuits. The examination centers around the precise speeds of the scoop arm cylinder for three kinds of chamber cylinder and the time taken for each test. The outcomes show that the little chamber cylinder has the most noteworthy rakish speed, while the medium chamber cylinder sets aside some margin to finish the test. The angular velocities of the dipper arm piston for three distinct cylinder-piston types utilized in an excavator are depicted in Figure 4.6. The cylinder-piston with the lowest angular velocity is the small one, followed by the medium and large ones. The medium cylinder-piston has a test time that is 27% shorter than the big one and 27% longer than the small one

The excavator's four major specifications will be examined in this chapter to determine how the vehicle behaves when using each type of hydraulic circuit. The rakish speeds of the scoop

arm cylinder for three unique kinds of chamber cylinders are portrayed in Figure 4.6. The duration of each test can also be viewed.

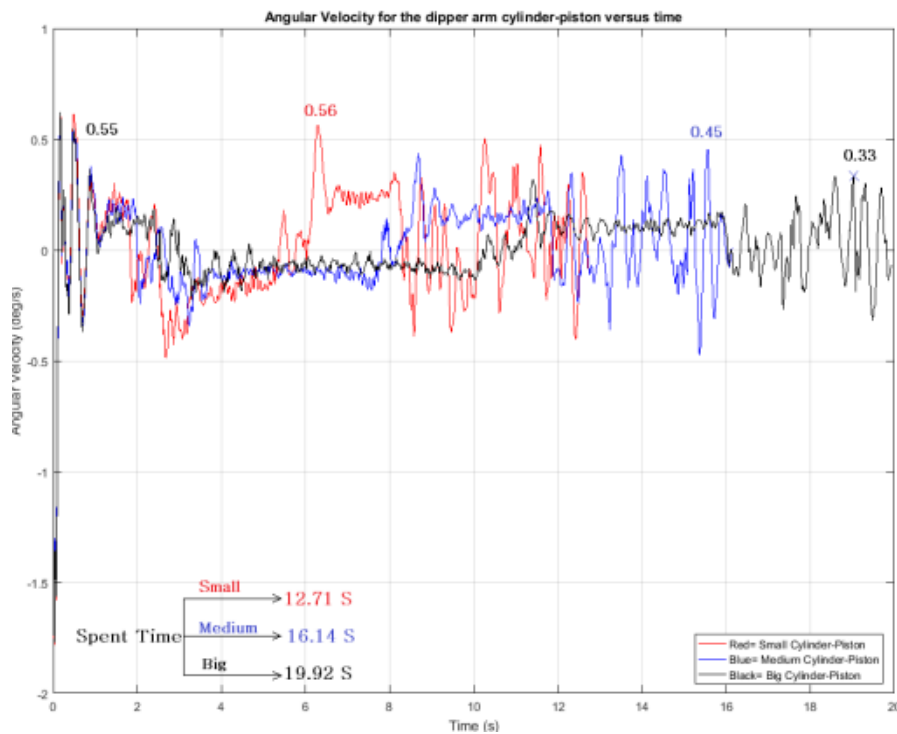


Figure 4. 5 The angular velocity for each of the excavator's 3 cylinder-piston inputs.

The findings of an examination of various excavator hydraulic circuits. According to the analysis, using a big cylinder piston takes 23% longer than using a medium one, while using a medium piston takes 27% longer than using a small one. Additionally, the small cylinder-piston has an average angular velocity of 0.2127 deg/s, which is 28 percent higher than the medium and 77 percent higher than the big one. This puts it ahead of the medium by 28 percent and ahead of the big one by 77 percent.

The cylinder-piston can move faster with a higher angular velocity, allowing the excavator to operate more quickly. The excavator will run slower as a result of the cylinder-piston's slower movement due to the lower angular velocity. When selecting a hydraulic circuit for an excavator, angular velocity alone may not be the only consideration.

The excavator's movement will be faster and more responsive at a higher angular velocity, whereas it will be slower and less responsive at a lower angular velocity. In this instance, the small cylinder-piston has a higher average angular velocity than the medium or large ones, making it possible that it is better suited for certain tasks that necessitate quick movements.

As expected, when the medium cylinder piston was used, the test took almost 27% longer than when the small one was used, as shown in Figure 4.6. Additionally, the excavator requires 23% more time than the medium cylinder piston to complete the test. It should come as no surprise that the small cylinder piston has a higher maximum angular velocity than the medium cylinder piston (24 percent) and the large cylinder piston (69 percent). The typical incentive for the little chamber cylinder is 0.213 deg/s which is 28% greater than the precise speed of the medium chamber cylinder and 77% more than the huge one. The typical speed for bucket A and large chamber cylinders are 0.167 deg/s and 0.120 deg/s individually. The dipper arm's cylinder pistons will change as the actuator velocity changes, resulting in a change in the generated force and working time for each cycle. Figure 4.8 represents actuator speeds for the scoop arm in three sorts of chamber cylinders of the scoop arm. The fact that the actual velocity for small cylinder pistons is more than for other pistons is confirmed by supreme values and local values on both cycle sides.

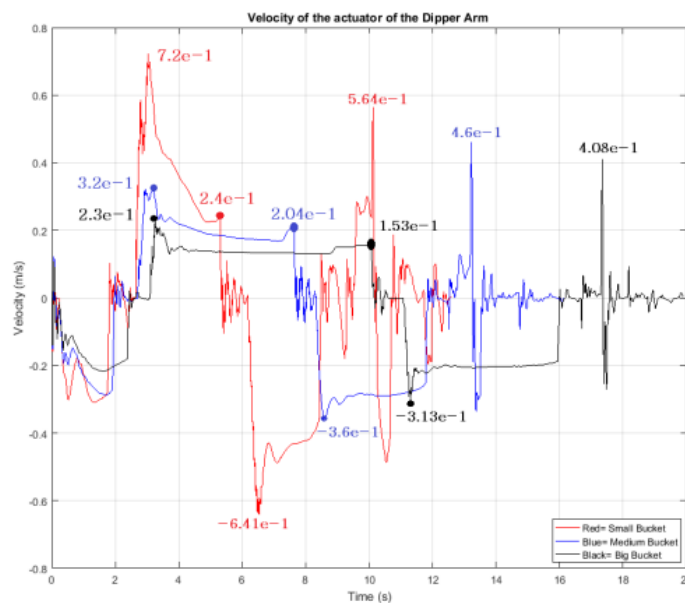


Figure 4. 6 The cylinder-piston dipper arm actuator velocity for three variations

The local supreme value of the actual velocity in the small case is 18% higher than in the average case, and the cylinder diameter of the dipper arm in the small case is 16% smaller than in the medium case. In addition, the small cylinder-piston case has a local maximum spot actuator velocity of 0.24 m/s, which is approximately 56% larger than the big cylinder-piston case. The big case has a diameter that is 33% larger than the small case, causing this difference. The final specification is fuel consumption, just as it was in section 4.9. Three different types of cylinder-piston are compared in Figure 4.10.

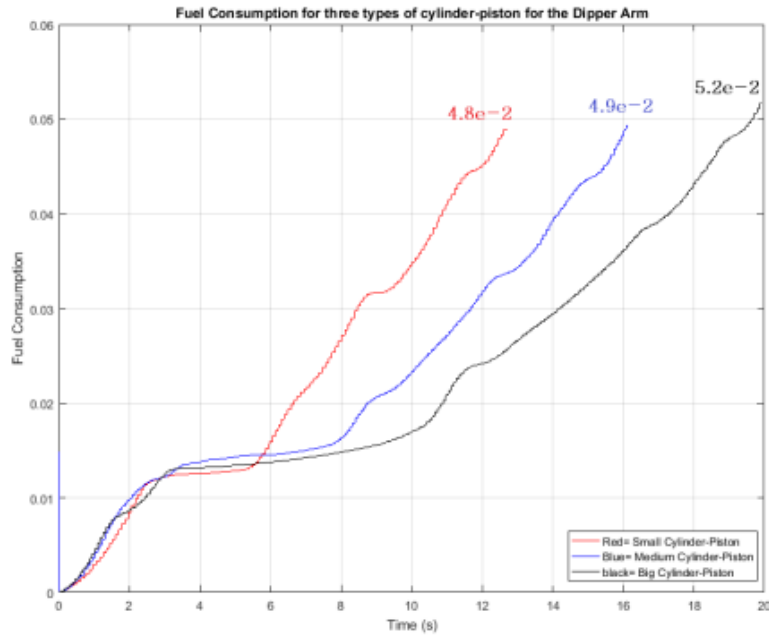


Figure 4. 7 Utilization of fuel for the three sorts of dipper arm chamber cylinders

Figure 4.7 compares the fuel consumption of three distinct dipper arm cylinder pistons. According to the graph, the big cylinder-piston case uses 8.4% more fuel than the small case and 6.1% more fuel than the medium cylinder-piston case. This information is important for understanding the efficiency of different cylinder-piston types and can help inform decisions about which type to use in the dipper arm. However, changing the cylinder-piston size does not have a significant impact on fuel consumption compared to changing the buckets. Unlike changing the buckets, which caused a noticeable difference.

According to Figure 4.7, the large bucket cylinder case uses 8.4% more fuel than the medium bucket cylinder and 6.1% more than the small case. Changes in bucket cylinder size do not significantly alter fuel consumption, unlike changes in containers, which resulted in significant differences in fuel consumption. There are a lot of different variables for each part of the excavator that can be looked at and broken down. In the subsequent section, one of the options that will be looked at is a plot for specific vehicles. Moreover, plots and results for each vehicle and its capability are distinct from those for any other vehicle or capability.

4.2 Future work

In many explorers and projects, there is dependably a perspective about the future and what will improve, change, and change in the following stages. From a variety of attainable perspectives, the improvement of this research can be considered. Software modification and interactions with other software, including the creation of a real model based on the simulation for

marketing purposes, etc. In this part, the absolute most possible future work will be talked about.

4.2.1 Using Software and the connections between them

The challenges of using multiple software programs to design a simulation model. Specifically, it is about the programs SolidWorks, Vortex, Blender, and Python, and how they are used in the design process.

SolidWorks and Blender are two programs that are commonly used for designing graphics. SolidWorks is good for creating accurate designs of bodies and assemblies, while Blender is better for creating more realistic designs of environments. However, there can be issues when trying to transfer designs between these programs and Vortex, another program used for simulation.

One of the challenges is that there isn't a standard file format that can be used to save files from SolidWorks and Blender and then used in Vortex. For example, SolidWorks can't create 3ds and Osgt document designs, which Vortex can understand, and Blender doesn't support the Osgt file format. Additionally, when designing in SolidWorks, all parts must be "dry" or without color because Blender can't read colors that come from SolidWorks.

To work around these issues. The first is to use a third-party program that can handle all aspects of design in a single program, avoiding the need to transfer files between multiple programs. The second solution is to write a code, such as a Python or Mat Lab code, to act as a bridge between the different programs and create a file format that Vortex can read.

In practice, designers will typically create the bodies in SolidWorks, export them to Blender for finishing touches and environment design, and then use them in Vortex. However, this process can be time-consuming and sometimes problematic, hence the suggested alternative solutions.

In this instance, SolidWorks, Vortex, Blender, and Python have all been used to design a simulation model. Two programs, Blender and SolidWorks, have been used to create visualization graphics; however, there are occasionally issues with communication between them and Vortex. SolidWorks is unquestionably useful for accurately designing bodies and their assemblies. Then again, Blender is an area of strength for establishing genuine designs, particularly for the climate. There isn't a standard file format that can be used to save created files and use them in Vortex if you use both programs. Vortex can understand 3ds and Osgt document designs and

SolidWorks can't make them. Additionally, Blender does not support the Osgt file format. Also, all parts made in SolidWorks ought to be dry since Blender can't perceive colors coming from SolidWorks.

4.2.2 An intuitive user interface – Gamification

The user-friendly interface and gamification elements in Vortex, which is a simulation software used to create real-time virtual models of industrial vehicles and equipment. Vortex offers three customizable interfaces for users to choose from before running their simulation models. Users can either use the assembly file approach or the coding file approach, as explained in Chapter 3.4. Alternatively, they can use the Vortex launcher, which is a third method for having an eye-catching point of interaction.

The Vortex launcher provides an intuitive and engaging user interface that is designed with stunning graphics. Users can play a simulation model and track its progress, such as how much pressure is applied to the motor or other specified parts, how much time is spent playing, how much fuel is used, and so on. Gamification elements can also be incorporated into the launcher, which allows users to compete with each other and be ranked according to their performance. For example, a leaderboard can be used to track the progress of users and provide a sense of competition.

In addition to the gamification elements, the Vortex launcher includes fuel and velocity meters, which can be used for straightforward missions. These meters provide users with an easy-to-understand way to monitor the simulation model's performance. The user interface is designed to be easy to use and intuitive, making it accessible to users with different levels of experience and technical knowledge.

Furthermore, Vortex offers a variety of settings for users to test and use their simulation models in additional games. This allows users to explore different scenarios and test the performance of their models under different conditions. Overall, Vortex provides a user-friendly and engaging interface that makes it easier for users to create and simulate real-time virtual models of industrial vehicles and equipment.

4.2.3 Further customizations - Analyze the section

Future work

4.3.1 Using Software and the connections between them

To design a simulation model, different software programs such as SolidWorks, Vortex, Blender, and Python are used. SolidWorks is good for designing detailed bodies and assemblies, while Blender is better for creating realistic graphics for the environment. However, there are some problems with communication between the software programs and a lack of a united file format to save created files.

4.3 User-friendlier interface - Gamification

The user interface of Vortex, a simulation software, can be made more user-friendly by adding gamification aspects. This means that the interface can be designed like a game, with meters such as fuel and velocity, to make it more engaging and easy to use. This can be done using Vortex launcher, which allows users to customize their simulation models in a fun and interactive way.

4.3.1 Further customizations - Analyze section

The "Further customizations - Analyze section" refers to the capability of customizing a simulation model to examine vital vehicle specifications like fuel consumption, mission completion time, and vehicle forces and torques. A company is able to offer a wider range of products to the market and classify its products according to market and customer demands thanks to this. Diagrams, plots, and comparison tables can also be provided by the simulation model to assess the vehicle's performance in various environments.

4.4 Representation of models and conditions - Climate customization

The idea of climate customization in the reenactment model considers various designs to be utilized for various pieces of a vehicle, like an excavator. In addition, this customization can be used to test and extract results from a simulation model in a variety of settings to evaluate the vehicle's performance and dependability.

4.4.1 Additional modifications: Examine the section.

This explains a design technique for customizing a simulation model of an industrial vehicle to evaluate its performance under different operating conditions. It suggests that additional parameterizations can be carried out using coding and assembly methods to find the best combinations of vehicle specifications, such as the duration of a mission, force acting on the car, abrasion of the vehicle's body, and fuel consumption.

The technique also allows a company to categorize its products based on market and customer demands, such as prioritizing time, fuel consumption, or vehicle lifetime. This enables the company to offer a wider range of products to the market.

The simulation model can be divided into 3 sections: customization, running the simulation, and analyzing the results. The client can modify the model based on their specific requirements and run it in the simulation environment. The results can be displayed through graphs, comparison tables, and diagrams to evaluate the vehicle's performance. The client can also customize the display options to choose from different designs and tables before running the simulation.

Overall, this technique allows for a customized simulation model to be developed for an industrial vehicle, providing valuable insights into its performance and aiding in design, operation, and maintenance optimization, as well as operator training and safety evaluations.

To find the best combinations, numerous additional parameterizations that are both feasible and reasonable can be carried out with the assistance of coding and assembly methods. This concept gives customers the right set of circumstances to look at important vehicle specifications. This indicates that this circumstance will be able to provide answers to important questions like the duration of a mission, the forces and torques applied to the vehicle, the abrasion of the vehicle's bodies, and fuel consumption.

A company can also categorize its products according to market and customer demands. For instance, while time may be the most important factor in some situations, fuel consumption or the vehicle's lifetime may be the most significant in others. This allows the company to provide and offer a wider range of its products to the market.

4.4.2 Model and environment visualization - Environment customization

To improve visualization in real-time simulation models of industrial vehicles by customizing the graphics and environments. One way to do this is by creating different graphics for various parts of the vehicle, such as different designs for each type of excavator bucket.

Customizing the environment is also important for testing and extracting results from the simulation model in different situations. By creating different environments with varying conditions, such as different terrains or weather patterns, it is possible to determine how the vehicle will perform and its reliability under different circumstances. This can be especially useful for

gamification purposes, where users can simulate different scenarios and evaluate the performance of the vehicle in various environments.

Overall, customization of graphics and environments in real-time simulation models can greatly enhance the visualization and accuracy of the model, leading to better design, operation, and maintenance of industrial vehicles.

Having different graphics for various parts while customizing is one way to improve visualization. For example, for a can of the Excavator, some representation designs can be set for each sort of pail.

Chapter 5

5 Conclusions

The creation of a real-time simulation model that can be customized with Vortex, Python, SolidWorks, and Blender software is the thesis's conclusion. Using a parameterized model that can be customized with user-selected data, the objective is to simplify and reduce the cost of modifying and analyzing simulation models. This approach can set aside time and cash by lessening the requirement for actual models and taking into account more exact investigation.

The creation of a simulation model that can be customized in real time is the primary objective of this project. This can be accomplished by utilizing Vortex programming, Python, SolidWorks, and Blender. With a defined model, it becomes simpler and savvier to change and break down reproduction models, lessening how much time and cash spent on planning and assembling genuine models.

A defined reenacted model is a PC program that can be modified to recreate various situations. A master's thesis utilized this kind of model, which was successful in achieving all of its objectives. The primary objective of this master's thesis is to develop a real-time simulation model that can be customized with Vortex software, Python, SolidWorks, and Blender software. In comparison to previous methods, using a customized simulation model can save time and produce results that are more accurate. When a parameterized model is used, it will take significantly less time and cost less to change and analyze a simulation model. This will make it easier and faster to create simulation models and put them on the market. When compared to previous ideas, this strategy has the potential to help conquer the real-simulation market and provide accurate analysis.

The comparison of fuel consumption and cycle time for excavators with different bucket sizes. The excavator with the smaller bucket was found to have the lowest total fuel consumption. The larger bucket, on the other hand, required more fuel to dig and transport materials, which resulted in a longer digging step and a longer total cycle time. Nonetheless, its process duration and fuel utilization per limit were essentially lower than the other two cans, which presents a valuable chance to save both time and fuel while dealing with a decent measure of materials.

Due to the large bucket's weight and the volume of material transferred, controlling it and digging with it are also highlighted in the conclusion.

Overall, the study provides insights into the performance of different excavator bucket sizes and their impact on fuel consumption and cycle time, which can inform decisions on equipment selection and operation.

Comparing the fuel consumption and cycle time of three different excavator configurations with varying bucket sizes. The specifications and results are presented in Table V, which is not included in the writing. However, based on the discussion, we can understand the following:

- The excavator with the small bucket had the lowest total fuel consumption.
- The large bucket had a longer total cycle time of 27.2 seconds compared to the other two buckets.
- The longer cycle time of the large bucket was due to a longer digging step after the first local maximum spot and higher fuel consumption of 77.67 ml for excavating the ground and carrying material particles.
- Despite the longer cycle time and higher fuel consumption per capacity, the large bucket was still more efficient than the other two buckets in terms of cycle time and fuel consumption. This suggests that using the large bucket can save both time and fuel when handling a fixed amount of materials.

Overall, the comparison highlights the trade-offs between bucket size, fuel consumption, cycle time, and position control in excavator operations. The findings can be useful for equipment selection and operation planning to optimize performance and efficiency.

The potential users of the information include:

1. Excavator operators: Excavator operators can use this information to better understand the movement and behavior of the excavator's bucket, which can improve their efficiency and productivity.
2. Excavator manufacturers: Excavator manufacturers can use this information to optimize the design of excavators and improve their performance, safety, and reliability.
3. Researchers and engineers: Researchers and engineers in the fields of mechanical engineering, hydraulic systems, and robotics can use this information to advance their understanding of excavator systems and develop new technologies and innovations.

4. Industry regulators: Industry regulators can use this information to develop and enforce safety standards and regulations for excavator operations.

About the position vs. time graph of an excavator bucket:

The excavator's bucket is controlled by the hydraulic system, which uses cylinders to move the arm and bucket. The operator controls the movement of the bucket using multiple levers (joystick) that direct hydraulic fluid to the appropriate cylinder

I have found out that when the control signal on the joystick remains the same, the valve characteristic curve can affect the cylinder's movement. Here are 3 basic types of the valve characteristic curve:

Preparing, Executing, and Analyzing the Excavator Case

The writing explains the process of parameterizing a real-time simulation model of an excavator, which involves defining the various parameters that are used to describe its characteristics and behavior. These parameters can be adjusted easily, allowing for different combinations of features to be tested quickly and efficiently.

To demonstrate the feasibility of the model, the simulation is run for several different bucket types and cylinder pistons. The results of the simulation are analyzed to evaluate the fuel consumption and operation time for each combination of parameters. The simulation is designed to simulate the excavator completing a task in three phases: bringing down the bucket, digging the ground, and lifting the bucket filled with the material back to its previous position.

The simulation results show that the larger bucket requires more time to complete the task than the smaller and medium buckets. However, despite the increased time, the larger bucket is able to transfer much more material due to its larger capacity, compensating for the additional time required. This type of analysis can be useful for optimizing the design and operation of an excavator, helping engineers to identify the best combination of parameters for a specific task or application.

Overall, this work demonstrates the importance of parameterizing real-time simulation models in the design and analysis of industrial vehicles like excavators. By defining and adjusting the various parameters, engineers can quickly test and evaluate different designs, allowing them to make informed decisions about the best configuration for a given task.

References

- [1] T. Gordon, M. Howell, F. Brandao, Integrated control methodologies for road vehicles, *Vehicle System Dynamics*, Vol. 40, n. 1-3, pp. 157-190, 2003. <https://doi.org/10.1076/vesd.40.1.157.15877>
- [2] M. González, A. Luaces, D. Dopico, J. Cuadrado, A 3D physics-based hydraulic excavator simulator, In: *ASME-AFM 2009 World Conference on Innovative Virtual Reality*, American Society of Mechanical Engineers Digital Collection, 2009, pp. 75-80. <https://doi.org/10.1115/winvr2009-734>

[3] R. Ramya, K. Selvi, K. Murali, Real-time simulation and performance analysis of multi-machine power systems using dSPACE simulator, *Simulation*, Vol. 92, n. 1, pp. 63-75, 2016. <https://doi.org/10.1177/0037549715620014>

[4] J. H. Jung, Power hardware-in-loop simulation (PHILS) of photovoltaic power generation using real-time simulation techniques and power interfaces, *Journal of Power Source*, Vol. 285, pp. 137-145, 2015. <https://doi.org/10.1016/j.jpowsour.2015.03.052>

Appendix 1

Some specifics on three different types of Volvo excavators. The data for the EW160D, EW180D, and EW210D may be found in the figures below (Volvo, 2017).

EW160D		
Engine	Volvo	D6H
Max power at	r/s / r/min	31,7 / 1,900
Gross, ISO 14396/SAE J1995	kW / hp	115 / 154
Net, ISO 9249/DIN 6271	kW / hp	112 / 150
Max torque at 1,350 rpm	Nm / lbf. ft	716 / 528
No. of cylinders		6
Displacement	l / cu. in	5.7 / 348
Bore	mm / in	98 / 3.86
Stroke	mm / in	126 / 4.96
EW180D		
Engine	Volvo	D6H
Max power at	r/s / r/min	33,3 / 2,000
Gross, ISO 14396/SAE J1995	kW / hp	129 / 173
Net, ISO 9249/DIN 6271	kW / hp	126 / 169
Max torque at 1,500 rpm	Nm / lbf. ft	758 / 559
No. of cylinders		6
Displacement	l / cu. in	5.7 / 348
Bore	mm / in	98 / 3.86
Stroke	mm / in	126 / 4.96
EW210D		
Engine	Volvo	D6H
Max power at	r/s / r/min	33,3 / 2,000
Gross, ISO 14396/SAE J1995	kW / hp	129 / 173
Net, ISO 9249/DIN 6271	kW / hp	126 / 169
Max torque at 1,500 rpm	Nm / lbf. ft	758 / 559
No. of cylinders		6
Displacement	l / cu. in	5.7 / 348
Bore	mm / in	98 / 3.86
Stroke	mm / in	126 / 4.96

		EW160D	EW180D	EW210D
Voltage	V		24	
Batteries	V		2 x 12	
Battery capacity	Ah		100	
Alternator	V / Ah		28 / 110	
Alternator rating	W		3,360	

		EW160D	EW180D	EW210D
Max. slew speed	r/min	10.0	9.1	9.7
Max. slew torque	kNm	50,4	65,6	77,5
	lbf. ft	37,200	48,400	57,200

		EW160D	EW180D	EW210D
Main pump, low noise axial piston pump				
Maximum flow	l/min	256	420	420
	gpm	67.6	111	111
Brake + steering pump, low noise gear pump				
Maximum flow	l/min	36.0	36.1	36.1
	gpm	9.5	10.0	10.0
Servo pump, low noise gear pump				
Maximum flow	l/min	15.0	15.2	15.2
	gpm	4.0	4.2	4.2
Maximum pressure:				
Implements	MPa	34,0 / 37,5	34,0 / 37,5	34,0 / 37,5
	Psi	4,930 / 5,440	4,930 / 5,440	4,930 / 5,440
Travel system	MPa	37,5	37,5	37,5
	Psi	5,440	5,440	5,440
Pilot system	MPa	4,0	3,5	3,5
	Psi	490	490	490

Appendix 2

Python code as a bridge between the Excel file, user interface, and Vortex software.

```

7## The Script file to handle all changes and options coming from the excel file
8## and implement it in the MeVEA Modellet.
9
10 import os
11 os.getcwd()
12 import openpyxl
13 import xml.etree.cElementTree as ET
14
15 tree = ET.ElementTree(file = 'Excavator.xml')
16 root = tree.getroot()
17
18 wb = openpyxl.load_workbook('Read_it.xlsx')
19 sheet = wb.active
20
21
22
23 # collecting data about the bucket
24 # Small / Standard / Big
25 # collecting data about the visualization and collision graphics
26 for bodies in root.iter('Bodies'):
27     for bucket in bodies.iter('Bucket'):
28         if (str(sheet['F2'].value) == "Small"):
29             bucket.set('VisualizationGraphics', 'Bucket_Small_Graphics')
30             bucket.set('CollisionGraphics', 'Coll_Bucket_Small_Graphics;COLL_Bucket_Teeths_HF_Small_Graphics;COLL_Bucket_HF_OnlyHF_Small_Graphics')
31             print (bucket.get('VisualizationGraphics'))
32         elif (str(sheet['F2'].value) == "Standard"):
33             bucket.set('VisualizationGraphics', 'Bucket_Graphics')
34             bucket.set('CollisionGraphics', 'Coll_Bucket_Graphics;Coll_Bucket_HF_Teeths_Graphics;COLL_Bucket_HF_OnlyHF_Graphics')
35             print (bucket.get('VisualizationGraphics'))
36         elif (str(sheet['F2'].value) == "Large"):
37             bucket.set('VisualizationGraphics', 'Bucket_Graphics_New_Big_Graphics')
38             bucket.set('CollisionGraphics', 'Coll_Bucket_big_Graphics;COLL_Bucket_Teeth_Big_Graphics;COLL_Bucket_HF_only_Big_Graphics')
39             print (bucket.get('VisualizationGraphics'))
40

```

```

40
41 # collecting data about the particles
42 for particles in root.iter('Particles'):
43     for pile in particles.iter('Pile'):
44         if (str(sheet['F2'].value) == "Small"):
45             contact_bucket.set('GraphName', 'Coll_Bucket_Small_Graphics')
46             print (contact_bucket.get('GraphName'))
47             contact_bucket.set('GraphName', 'Coll_Bucket_Graphics')
48             print (contact_bucket.get('GraphName'))
49         elif (str(sheet['F2'].value) == "Large"):
50             contact_bucket.set('GraphName', 'Coll_Bucket_big_Graphics')
51             print (contact_bucket.get('GraphName'))
52
53 # collecting data about the particles
54 for particles in root.iter('Particles'):
55     for pile in particles.iter('Pile'):
56         for mbscontactgraph in pile.iter('MBSContactGraph'):
57             for contact_bucket_onlyhf in mbscontactgraph.iter('Contact_Bucket_OnlyHF'):
58                 if (str(sheet['F2'].value) == "Small"):
59                     contact_bucket_onlyhf.set('GraphName', 'COLL_Bucket_HF_OnlyHF_Small_Graphics')
60                     print (contact_bucket_onlyhf.get('GraphName'))
61                 elif (str(sheet['F2'].value) == "Standard"):
62                     contact_bucket_onlyhf.set('GraphName', 'COLL_Bucket_HF_OnlyHF_Graphics')
63                     print (contact_bucket_onlyhf.get('GraphName'))
64                 elif (str(sheet['F2'].value) == "Large"):
65                     contact_bucket_onlyhf.set('GraphName', 'COLL_Bucket_HF_only_Big_Graphics')
66                     print (contact_bucket_onlyhf.get('GraphName'))
67
68 for particles in root.iter('Particles'):
69     for pile in particles.iter('Pile'):
70         for mbscontactgraph in pile.iter('MBSContactGraph'):
71             for contact_bucket_teeths in mbscontactgraph.iter('Contact_Bucket_Teeths'):
72                 if (str(sheet['F2'].value) == "Small"):
73                     contact_bucket_teeths.set('GraphName', 'COLL_Bucket_Teeths_HF_Small_Graphics')
74                     print (contact_bucket_teeths.get('GraphName'))
75                 elif (str(sheet['F2'].value) == "Standard"):
76                     contact_bucket_teeths.set('GraphName', 'Coll_Bucket_HF_Teeths_Graphics')
77                     print (contact_bucket_teeths.get('GraphName'))
78                 elif (str(sheet['F2'].value) == "Large"):
79                     contact_bucket_teeths.set('GraphName', 'COLL_Bucket_Teeth_Big_Graphics')
80                     print (contact_bucket_teeths.get('GraphName'))
81
82
83 # collecting data for graphics for collision graphics
84
85 for graphics in root.iter('Graphics'):
86     for COLL_Bucket_HF_TeethSimple_Graphics in graphics.iter('COLL_Bucket_HF_TeethSimple_Graphics'):
87         if (str(sheet['F2'].value) == "Small"):
88             COLL_Bucket_HF_TeethSimple_Graphics.set('FileName', '../Graphics/Excavator/Bucket/scale0.6z/COLL_Bucket_Teeths_HF_Simple_0.6.3ds')
89             print (COLL_Bucket_HF_TeethSimple_Graphics.get('FileName'))
90         elif (str(sheet['F2'].value) == "Standard"):
91             COLL_Bucket_HF_TeethSimple_Graphics.set('FileName', '../Graphics/Excavator/Bucket/COLL_Bucket_Teeths_HF_Simple.3ds')
92             print (COLL_Bucket_HF_TeethSimple_Graphics.get('FileName'))
93         elif (str(sheet['F2'].value) == "Large"):
94             COLL_Bucket_HF_TeethSimple_Graphics.set('FileName', '../Graphics/Excavator/Bucket/COLL_Bucket_Teeths_HF_Simple.3ds')
95             print (COLL_Bucket_HF_TeethSimple_Graphics.get('FileName'))
96
97
98 # collecting data to change the piston
99
100 for dummies in root.iter('Dummies'):
101     for dipperanrpiston_dummy in dummies.iter('DipperArmPiston_Dummy'):
102         if (str(sheet['F5'].value) == "Small"):
103             dipperanrpiston_dummy.set('VisualizationGraphics', 'BoomPiston_scale0.73_Graphics')
104         elif (str(sheet['F5'].value) == "Standard"):
105             dipperanrpiston_dummy.set('VisualizationGraphics', 'DipperArmPiston_Graphics')
106         elif (str(sheet['F5'].value) == "Big"):
107             dipperanrpiston_dummy.set('VisualizationGraphics', 'BoomPiston_scale1.31_Graphics')
108
109 #collecting data to change the cylinder
110

```



```

111 for dummies in root.iter('Dummies'):
112     for dipperarmcylinder_dummy in dummies.iter('DipperArmCylinder_Dummy'):
113         if (str(sheet['F5'].value) == "Small"):
114             dipperarmcylinder_dummy.set('VisualizationGraphics', 'BoomCylinder_scale0.73_Graphics')
115         elif (str(sheet['F5'].value) == "Standard"):
116             dipperarmcylinder_dummy.set('VisualizationGraphics', 'DipperArmCylinder_Graphics')
117         elif (str(sheet['F5'].value) == "Big"):
118             dipperarmcylinder_dummy.set('VisualizationGraphics', 'BoomCylinder_scale1.31_Graphics')
119
120 # collecting data to change the flow rate inside the cylinder-piston
121
122 for forces in root.iter('Forces'):
123     for hydraulics in forces.iter('Hydraulics'):
124         for dv63 in hydraulics.iter('DV63'):
125             for dv63_dipperarm2 in dv63.iter('DV63_DipperArm2'):
126                 if (str(sheet['F6'].value) == "125"):
127                     dv63_dipperarm2.set('QnomPA', '124.9')
128                 elif (str(sheet['F6'].value) == "170"):
129                     dv63_dipperarm2.set('QnomPA', '170')
130                 elif (str(sheet['F6'].value) == "222"):
131                     dv63_dipperarm2.set('QnomPA', '222.04')
132
133 for forces in root.iter('Forces'):
134     for hydraulics in forces.iter('Hydraulics'):
135         for cylinders in hydraulics.iter('Cylinders'):
136             for cylinder_dipperarm in cylinders.iter('Cylinder_DipperArm'):
137                 if (str(sheet['F5'].value) == "120"):
138                     cylinder_dipperarm.set('dinn', '120')
139                 elif (str(sheet['F5'].value) == "140"):
140                     cylinder_dipperarm.set('dinn', '140')
141                 elif (str(sheet['F5'].value) == "160"):
142                     cylinder_dipperarm.set('dinn', '160')
143
144
145 tree.write('Excavator.xml')

```