This paper is available online at www.praiseworthyprize.org
International Review of Mechanical Engineering (I.RE.M.E.), Vol. 13, N. 12
ISSN 1970 - 8734 *December 2019*

A Design Process to Parameterize a Real-Time Simulation Model of a Commercial Vehicle

Manouchehr Mohammadi, Emil Kurvinen, Aki Mikkola

Abstract – *This paper introduces a method for building a real-time simulation model with adjustable user-selected parameters. The proposed design process model consists of eight steps with four decision points. Parameterization is a technique enabling real-time simulation with different combinations of parameters. Currently, there is no unique way to incorporate user input and switch between model combinations. The proposed method is presented in the form of a flowchart. Based on the data, a 3D design of the model was constructed. Two alternative approaches were introduced to construct a parameterized real-time simulation model with user inputs. The approach used was selected based on the number of parameterized specifications. The feasibility of each case was analyzed analytically and by simulation. Finally, a version of the model was selected based on the given initial requirements. To illustrate the developed approach, an excavator model was selected for parameterization. In the excavator model, two parts are considered to have adjustable parameters: the bucket and the hydraulics. Each part has three options that can be selected by users. The approach enables easy adaptability of user-generated parameter inputs, thus permitting evaluation of multiple scenarios, while simultaneously maintaining realistic representation.* ***Copyright © 2019 The Authors.***

Keywords*:* *Excavator Model, Parameterization, Multibody System, Parameterized Specifications, Real-Time Simulation*

Nomenclature

A	Area
CAD	Computer aided design
F_b	Bucket weight
F_d	Dipper arm weight
F_h	Hydraulic force
F_p	Force generated by the accumulated sand particles in the bucket
I_{xx}, I_{yy}, I_{zz}	Moments of inertia
L_1, L_2, L_3	Lengths of different parts of the excavator
M_b	Moment generated by F_B
m_b	Mass of the large bucket
M_d	Moment generated by F_d
m_d	Mass of the dipper arm
M_h	Moment generated by F_h
M_p	Moment generated by F_p
m_p	Mass of the accumulated sand particles in the bucket
M_T	Equivalent moment
P_h	Hydraulic pressure
XML	Extensible Markup Language

I. Introduction

Dynamic simulation has proven to be a valuable tool; for this reason, it is commonly implemented to a number of product processes. To evaluate the performance of a machine using computerized methods, the equations of motion must be formulated and solved. Several studies on how to write and solve equations of motion for multibody system dynamics can be found in the literature [1]-[4]. In the design and appraisal of industrial vehicles, a simulation model can significantly reduce required design time and the cost of prototypes [5], [6]. When considering the dynamic performance of machines, it is important to note, however, that the operators' experience often plays an important role. Nevertheless, most simulation studies have focused on modelling methods, whereas studies that account for the influence of the operator have received little attention. The user can be given more consideration by employing real-time simulation of dynamic systems. When using a real-time simulator, users must feel as if they are operating a real machine, which can be achieved only if the real-time simulation model is accurate and couples the different engineering areas: such as hydraulics, pneumatics, and electronics. Currently, however, such real-time models are usually case-specific and thus tailored to specific applications. Furthermore, the development of such real-time models is labor-intensive and incurs considerable costs. The problem of high cost can be mitigated using a real-time simulation approach based on multibody system dynamics. Real-time simulation in multibody

Available online by December 31st, 2019 https://doi.org/10.15866/ireme.v13i12.18018

systems has been studied in a number of research areas such as aviation and industrial vehicles [7]-[9], hybrid vehicles [10], the automotive industry [11], four-bar mechanisms, and flexible multibody systems [12], [13].

However, increased usage of real-time simulation has been limited, because hours of work are needed to build a model and it may not guarantee in useful results [14].

The problem of the high cost and high-specificity of current machine simulation approaches can be addressed by developing adjustable physics-based, real-time, simulator-driven processes for product development.

This idea can be accomplished, in practice, by developing a toolset that will allow users to access machine research and development. This can be accomplished, in practice, through virtual worksites providing fully configurable, real-time, virtual prototyping. From a system engineering point of view, a multi-step approach is required for the design and construction of a product. First, the problem and the requirements of any possible solution are identified.

Then, actions to achieve the target are introduced in steps, followed by the respective analysis of each step [15]. The analysis steps assess whether the constructed product meets the requirements at each step, and comments on the feasibility of the chosen approach or the desired requirements. A feasible product development procedure should be generic and modifiable to suit the final product [16], [17]. Despite numerous studies on product development, constructing generic models in different fields, and the utilization of multibody system simulation, limited attention has been paid to interaction with users during the design phase and provision of simulation models with adjustable parameters. [18]-[21]

The objective of this paper is to introduce a method for building a real-time simulation model with adjustable parameters. The design steps are presented in the format of a flowchart. An excavator model is selected as a case study to be parameterized. Using the presented approach to develop simulation models, parameters can be selected based on different operation scenarios. By extending the applicability of the real-time methodology, the approach allows the construction of system simulations that previously were prohibitively difficult to analyze or required extensive effort to update the real-time simulation model. As such, this research seeks to address techniques to generate feasible real-time simulation models with adjustable parameters.

The structure of the paper is as follows: Section two explains how to construct a flowchart to illustrate the design steps of a parameterized real-time simulation model. Section three introduces a numerical case example and the feasibility analysis. Section four details the differences between the two techniques and their advantages and disadvantages. Conclusions are presented in the final section.

II. Methods

To create a parameterized real-time simulation model of an industrial vehicle, a general design flowchart is introduced as depicted in Fig. 1. The design process consists of eight steps and four decision points.

II.1. Step 1: Primary Requirements

In the parameterization, there is a base model to which the other parameterized parts are added. To be able to build a parameterized model, several primary requirements must be met, and some fundamental data must be provided. The requirements can be divided into three major groups, the first of which are requirements related to the vehicle, for example, data about the mass, inertia, and geometry of the vehicle parts. The second group is environment requirements. The interaction between the vehicle and the environment is critical, and the limitations, borders and other specifications of the environment are defined in this step. Ground conditions, such as its material and slope, should be explicitly described. The third group is data regarding the design specifications to be parameterized. For instance, technical properties of parameterized parts, such as weight or material should be defined in a way that their effect on the feasibility of the model is readily comparable.

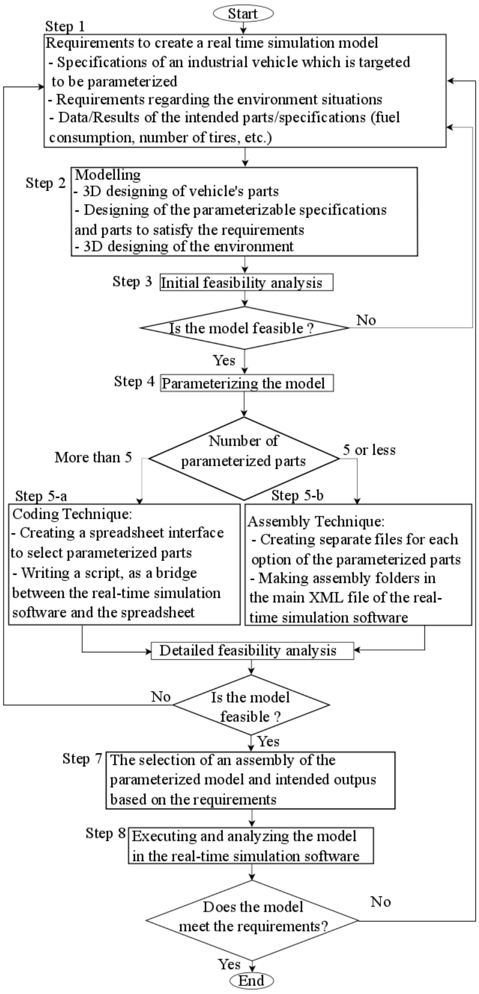

Start
Step 1
Requirements to create a real time simulation model
- Specifications of an industrial vehicle which is targeted to be parameterized
- Requirements regarding the environment situations
- Data/Results of the intended parts/specifications (fuel consumption, number of tires, etc.)
Step 2
Modelling
- 3D designing of vehicle's parts
- Designing of the parameterizable specifications and parts to satisfy the requirements
- 3D designing of the environment
Step 3
Initial feasibility analysis
Is the model feasible ?
No
Yes
Step 4
Parameterizing the model
Number of parameterized parts
More than 5
5 or less
Step 5-a
Coding Technique:
- Creating a spreadsheet interface to select parameterized parts
- Writing a script, as a bridge between the real-time simulation software and the spreadsheet
Step 5-b
Assembly Technique:
- Creating separate files for each option of the parameterized parts
- Making assembly folders in the main XML file of the real-time simulation software
Detailed feasibility analysis
No
Is the model feasible ?
Yes
Step 7
The selection of an assembly of the parameterized model and intended outpus based on the requirements
Step 8
Executing and analyzing the model in the real-time simulation software
Does the model meet the requirements?
No
Yes
End

Fig. 1. Design flowchart of a parameterized real-time simulation model of an industrial vehicle

II.2. Step 2: Modelling

To parameterize the vehicle, its 3D geometry and operating environment is needed. In this procedure, the vehicle parts are assumed as either rigid or flexible bodies. To decrease the simulation time, the graphic of the vehicle should be simplified, for example, by neglecting detailed parts that do not affect simulation results. The vehicle parts are modeled using CAD or 3D graphic software and the environment is constructed with graphics software. Standard parts of the vehicle can be used via prepared 3D models that exist in CAD libraries.

In addition, 3D vehicle parts may be available from the manufacturer or component manufacturers.

II.3. Step 3: Initial Feasibility Analysis

Whenever a parameterized model is planned, its feasibility must be taken into account. For example, the engine size and hydraulic circuitry of a vehicle have specific operating ranges and capacities. In addition, each parameterized specification/part can be considered as having maximum and minimum loading cases based on functionality as a function of weight, size, and space considerations. For instance, the tire of a parameterized vehicle has maximum and minimum values for size or mass, i.e., the tire must fit the mountings. In this step, the extreme cases are identified from the parameterized cases and the most demanding case feasibilities are studied. For example, the combination of the minimum hydraulic force and the maximum arm weight of a lifting vehicle would be studied to assess if there is sufficient force to lift the components. In this step, the feasibility is estimated with an analytical method, which is called initial feasibility analysis. If the parameterized parts and their extreme combinations seem reasonable, parameterization can proceed. Otherwise, an iteration round is needed to refine the requirements in Step 1.

II.4. Step 4: Parameterization

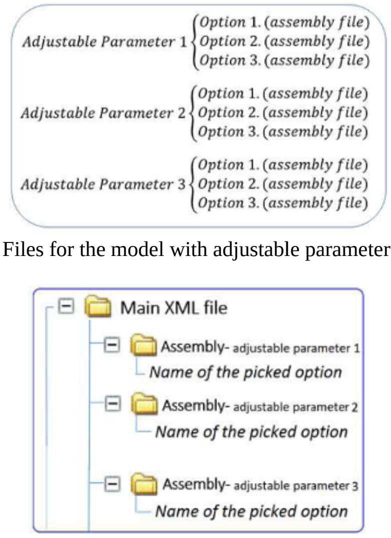
In the design process, parameterization can be done using two fundamentally different techniques: the coding technique or, alternatively, the assembly technique. The coding technique requires effort in the initial preparation, but as the number of parameterizations increases the relative amount of work required decreases. With the assembly technique, each combination preparation demands the same amount of work and, thus the assembly technique is considered feasible only for cases with a small number of combinations. In this design process, five different combinations are considered to be the limit where the coding technique should be preferred over the assembly technique. The number of different parameters and combinations are defined in this step.

II.5. Step 5-a: Coding Technique

In the coding technique, all data and specifications regarding parameterized specifications are coded in a spreadsheet. The spreadsheet is the user interface. Additionally, a script is written in a programming language as a bridge between the user interface and the real-time simulation software. In this work, the programming language Python was selected for coding the script and it has already been used for connector coding in previous studies. For example, Rubenstein et al. [22] used a Python framework as a bridge between their analysis and interface. The procedure for using this technique is to choose favored options from the spreadsheet in the first stage. Then, the script reads the data from the spreadsheet and implements them into the real-time simulation software. In the real-time simulation software, each designed model has four main files: an executable file and three Extensible Markup Language (XML) files. The XML files comprise all necessary data about the model and its environment. One of the created XML files, entitled ‘the main XML file’ consists of information about the mechanical and graphical properties of the model parts. As noted earlier, the script reads values that are selected by the user and substitutes them in the main XML file in place of existing values.

II.6. Step 5-b: Assembly Technique

In the assembly technique, an assembly file, a type of readable file that can be used for alternative options for each adjustable part, is generated for each option of the adjustable parameters. For example, if a model with three adjustable parameters is considered, each of which has three alternative options, nine assembly files are needed for the model as shown in Fig. 2(a). Additionally, assembly folders equal in number to the adjustable parameters must be placed in the main XML file to read the assembly files. The number of assembly folders in the main XML file depends on the number of adjustable parameters. In this example, three assembly folders are required for three adjustable parameters, Fig. 2(b).


Adjustable Parameter 1
Option 1. (assembly file)
Option 2. (assembly file)
Option 3. (assembly file)
Adjustable Parameter 2
Option 1. (assembly file)
Option 2. (assembly file)
Option 3. (assembly file)
Adjustable Parameter 3
Option 1. (assembly file)
Option 2. (assembly file)
Option 3. (assembly file)
Main XML file
Assembly- adjustable parameter 1
Name of the picked option
Assembly- adjustable parameter 2
Name of the picked option
Assembly- adjustable parameter 3
Name of the picked option

(a) Files for the model with adjustable parameters X

(b) Required assembly folders in the main XML file

Figs. 2. Procedure to create a real-time simulation model with adjustable parameters with assembly technique

As Fig. 2(b) demonstrates, when users want to build a model, they should select files from those given in Fig. 2(a). Then, the names of the picked files must be written in the assembly folders.

II.7. Step 6: Detailed Feasibility Analysis

In Step 6, the detailed feasibility analysis regarding the extreme combinations is carried out using real-time simulation software. In this step, the combination of options considered in the initial feasibility analysis is evaluated in the real-time simulation software. If the results are feasible the procedure can progress further. Otherwise, an iteration is needed to modify the requirements in Step 1.

II.8. Step 7: Preparing the Case

In step 7, among all possible combinations of user inputs to the parameterized model, any combination can be readily selected for simulation. The user-input defined model, parameterized with either the coding or assembly technique, is ready for examination in the real-time simulation software for the outputs of interest selected.

For instance, outputs may be the hydraulic force of a specific cylinder or the pressure impulse for a specific work cycle.

II.9. Step 8: Executing and Analyzing the Model

In this step, the user-selected combination of inputs for the parameterized model is analyzed. Thus, the model is run in the real-time simulation software to evaluate the specifications of the vehicle. The results of the simulation test can explicitly demonstrate whether the model has met the given requirements in Step 1. If the model fails to meet the given requirements, either the requirements should be re-examined, or modifications should be made to the design of the parameterized parts.

The results of the simulated model explicitly demonstrate how changing or modifying values in the model lead to changes in its performance. The eight-step design process described enables the construction of a real-time simulation model with parameterized specifications that can be analyzed explicitly and individually. Moreover, switching among possible combinations of user-generated input parameters is made possible, which provides an opportunity to compare the efficiency of the various combinations available in the model. In the next section, a real-time simulation excavator is parameterized based on the design process shown in Fig. 1. Two specifications of the excavator model are parameterized.

III. Case Study

In this study, an excavator model is selected for parameterized specifications. The layout of the model is shown in Fig. 3.

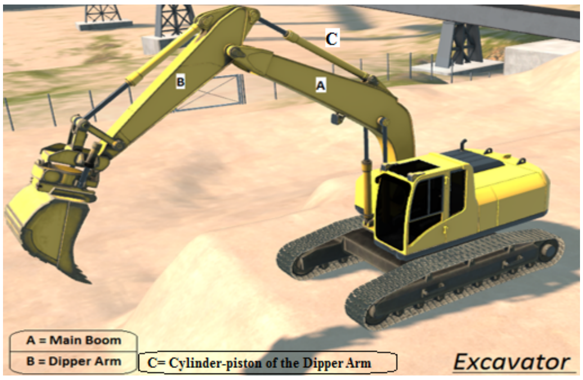


Fig. 3. Studied excavator with adjustable parameters

The simulated excavator model is a representation of a real excavator with 22 tons operating weight. The excavator model is built by following the design process described in the flowchart in Fig. 1. The following sections describe the steps. In this study, a semi-recursive method [23] is employed to construct the equations of motion for the excavator. This method was selected, because it has been found to be efficient in tree-like mechanisms, and it has previously been used to form and build structures in efficient simulation models [24]-[26].

The hydraulic system is modelled using the lumped fluid theory [27]. Equations of motion based on the semi-recursive method and differential equations that describe the hydraulic system can be combined to form a set of differential equations. These equations can be solved using the efficient, Runge-Kutta time integration scheme [28]. The Runge-Kutta method has been used in previous studies to solve equations, i.e., mass and energy equations. [29]

III.1. Step 1: Primary Requirements of an Excavator

To build a model for the excavator shown in Fig. 3, a mathematical representation must be constructed. The specifications to be parameterized in this example are the bucket and a part of the hydraulic circuit system of the excavator. The bucket and the cylinder piston of the main-boom were selected as the adjustable components for parameterization in the model because they play crucial roles in the efficiency of real excavators. For instance, they considerably affect the time needed to accomplish a task and the fuel consumed. In the case under study, each parameterized component has three options, which are selected based on the work environment situation. The steering control and hydraulic systems of vehicles have been the subject of several studies in recent decades [30], [31]. Parameterization is a method that permits comprehensive control of vehicles with user-friendly techniques. The ground material is designed as deformable sand based on a particle description. A certain number of functional assemblies of the excavator model are the combinations of parameterized user inputs of the model, which are interchangeable without efforts and limitations. For example, the fuel consumption of the excavator can be accounted for by substituting different parameter values for the bucket. The requirements of the adjustable parts for the excavator are given in the tables below. The properties of the buckets are reported in Table I. The cylinder piston of the dipper arm (C) to be customized and investigated in the hydraulic circuit parameterization is shown in Fig. 3. The properties of the three types of cylinder piston controlling the dipper arm described in Table II. It should be noted that the same size of hydraulic pump is used for all the cylinder pistons.

III.2. Step 2: Modelling of the Excavator

The 3D model of the excavator and the environment was built using open-source Blender software and the parameterized components were generated with the computer-aided engineering and computer-aided design software SolidWorks. Fig. 4 shows the bucket design for the small, medium and large buckets. The center of mass (C.O.M) of each bucket and its corresponding coordinate system are shown in Fig. 4. All the buckets and cylinder pistons are designed such that users can choose any pair of bucket and cylinder piston in accordance with their needs. In this case, the buckets and cylinder pistons are assumed to be rigid bodies.

III.3. Step 3: Initial Feasibility Analysis of the Excavator

In this step, an initial calculation is done for a combination of parameters with the highest probability of inability to perform the given tasks, which in this case is the combination of the large bucket and the small cylinder piston. Fig. 5 presents a 2D schematic of the bucket, the main boom, the dipper arm showing their weight forces.

TABLE I
SPECIFICATIONS OF BUCKETS

Components	Small Bucket	Medium Bucket	Large Bucket
Mass (kg)	315	450	450
Moments of inertia (I_{xx}, I_{yy}, I_{zz})	111.2, 118.6, 107.6	187.0, 228.1, 168.1	370.0, 330.0, 206.9
Center of mass (m)	0.2, -0.7, 0	0.2, -0.7, 0	0.2, -0.7, 0
Capacity (m^3)	0.32	0.51	2.55

TABLE II
SPECIFICATIONS OF CYLINDER PISTON OF THE DIPPER ARM

Parameters	Small Cylinder Piston	Medium Cylinder Piston	Large Cylinder Piston
Mass (kg)	8	10	12
Diameter (mm)	120	140	160
Rod length (mm)	1630	1650	1670

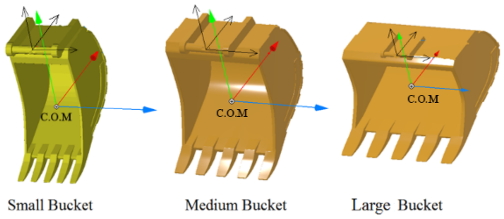


Fig. 4. Three types of bucket studied

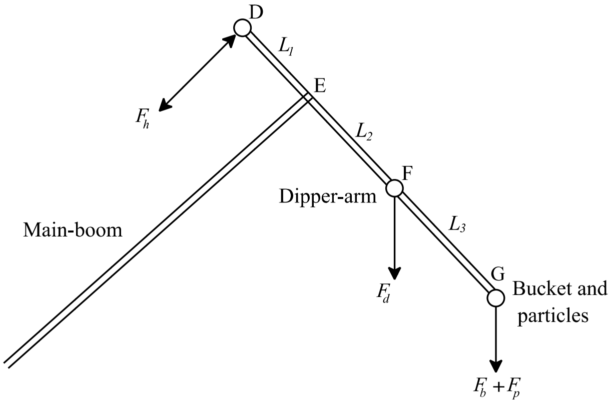


Fig. 5. Schematic of the forces acting on the bucket, the main boom, and the dipper arm

In Fig. 5, F_h is the hydraulic force of the small cylinder, F_d is the dipper arm weight, F_b is the bucket weight, and F_p is the weight of the accumulated sand particles.

The hydraulic force is implemented at point D, F_h. All moments are calculated based on point E. Points F and G are the centers of mass for the dipper-arm and the bucket with a load of sand particles, respectively.

Table III shows the values used to calculate the moments. If the amount of moment generated by F_h is higher than the amount of moment generated by F_d, F_b, and F_p, then the studied combination is feasible.

Otherwise, an iteration round is needed to change the requirements in Step 1.

In Table III, A is the difference between the areas of the cylinder and rod for the small cylinder piston. The values for the hydraulic force, F_h, and its corresponding moment, M_h, are given in Table IV. In addition, the maximum values of F_d, F_b, and F_p are also shown in the Table. In Table IV, g is gravity, which is taken as 9.806 m/s^2.

TABLE III
THE VALUES NEEDED TO CALCULATE THE MOMENTS

Parameters	Symbol	Values
Moment arm for the hydraulic force	L_1 (m)	0.98
Moment arm for F_d	L_2 (m)	0.80
Moment arm for F_d and F_p	L_3 (m)	2.87
Hydraulic Pressure	P_h (Pa)	34×10^6
Area of cylinder	A (m^2)	1.256×10^{-3}
Mass of the dipper arm	m_d (kg)	795
Mass of the sand particles	m_p (kg)	145.71
Mass of the large bucket	m_b (kg)	750

TABLE IV
VALUES OF THE HYDRAULIC FORCE, DIPPER ARM WEIGHT, BUCKET WEIGHT, PARTICLE WEIGHT, AND THEIR CORRESPONDING MOMENTS

Description	Parameters	Units
Hydraulic force	$F_h=P_h\cdot A=42.70$	kN
Moment, generated by F_h	$M_h=F_h\cdot L_1=42.15$	kNm
Dipper arm weight	$F_d=m_d\cdot g=7.80$	kN
Bucket weight	$F_b=m_p\cdot g=7.36$	kN
Particle weight	$F_p=m_d\cdot g=1.43$	kN
Moment, generated by F_d	$M_d=F_d\cdot L_2=6.24$	kNm
Moment, generated by F_b	$M_b=F_b\cdot(L_2+L_3)=27.01$	kNm
Moment, generated by F_p	$M_p=F_p\cdot(L_2+L_3)=5.24$	kNm

Moments, M_T, acting at point E, are written in Equation (1):

$$\begin{aligned} M_T &= M_h - \left(M_d + M_b + M_p\right) = \\ &= 42.15 - \left(6.24 + 27.01 + 5.24\right) = 3.677 \text{ kN m} \end{aligned} \quad (1)$$

It can be deduced that the extreme combination of the excavator model is feasible, and, therefore, the parameterization procedure can proceed. Because the value of M_T is positive, the excavator model is feasible, and the parameterization procedure can proceed without iteration and modification to the requirements. The excavator also requires a moment for digging the ground, which is taken into account and evaluated during the detailed feasibility analysis phase (Step 6).

III.4. Step 4: Parameterization of the Excavator

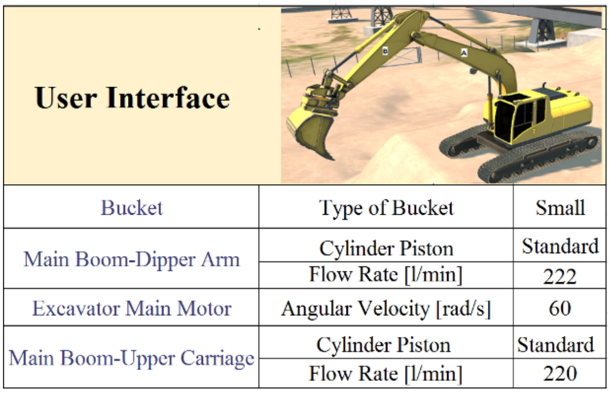
In Step 4, the excavator is modeled using one of two techniques, namely, the coding technique and the assembly technique. As described in Section II, the approach is selected based on the number of specifications to be parameterized. In the example under discussion, there are more than five specifications in total for the excavator model, i.e., three buckets and three cylinder pistons. Consequently, the coding technique was selected.

III.5. Step 5-a: Coding Technique

Fig. 6 gives an example of the spreadsheet used as the user interface for the excavator model. The intended bucket and cylinder piston of the main boom can explicitly be selected, and their details can be input into the spreadsheet.

Once the spreadsheet has been updated, the Python script is executed, and the software implements the selected bucket and cylinder piston in the excavator model.

The Python script reads all changes in the spreadsheet and revises the main XML file based on the changes made. The model is then ready to be run in the real-time simulation software.



User Interface

Bucket	Type of Bucket	Small
Main Boom-Dipper Arm	Cylinder Piston	Standard
	Flow Rate [l/min]	222
Excavator Main Motor	Angular Velocity [rad/s]	60
Main Boom-Upper Carriage	Cylinder Piston	Standard
	Flow Rate [l/min]	220

Fig. 6. Spreadsheet as the user interface

III.6. Step 6: Detailed Feasibility Analysis of the Excavator

Step 6 completes the feasibility analysis phases. The feasibility of the selected combination of parameters, which was estimated analytically in Step 3, is checked in that it is run in the real-time simulation software and checked to confirm that the excavator model can move freely, dig the ground, and transfer the material while in its extreme combination.

III.7. Step 7 and 8: Preparing, Executing and Analyzing the Excavator Case

The parameterized excavator model is ready for use in a way that the combination of parameters can be chosen easily. Using the available options for the types of buckets and cylinder pistons of the main-boom, the excavator model can effortlessly be assembled in different ways. All possible combinations of the excavator model are tested for feasibility. Fuel consumption and total operation time were selected in this study to demonstrate two outputs of a real-time simulation test. To provide results, the total consumed fuel and total spent time of the model are selected before running the model. Then, the model is run, and the results/data can be analyzed. The results for the three different buckets and the small cylinder piston were evaluated in a simulation test. The excavator had the same initial position and initial velocity for all three buckets. The bucket moves in three phases: bringing down the bucket, digging the ground, and lifting the bucket filled with material to its previous position. The total required time to complete the task was the first specification. As Fig. 7 shows, the large bucket required more time to complete the task than the small and medium buckets, 56.9% and 40% longer, respectively.

On the other hand, the considerable difference in capacity (Table I), the big bucket compensated the drawback by transferring much more amount of material than other two buckets.

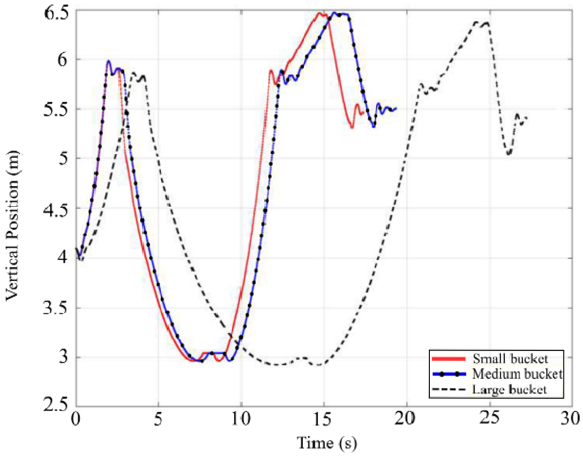


Fig. 7. Total spent time to complete the task for the three buckets studied; 17.36 s, 19.35 s, and 27.24 s for the small, medium, and large buckets, respectively

Fig. 8 depicts the fuel consumption in the test with the small cylinder piston and three different buckets. At the beginning of the test, the results for the three buckets are similar to each other. However, the excavator configuration with the small bucket has the lowest total fuel consumption. The discussed specifications are given in Table V. The large bucket has a longer total cycle time (27.2 s) due to a longer digging step (after the first local maximum spot) and higher fuel consumption (77.7 ml) for digging the ground and carrying the material.

However, its cycle time and fuel consumption per capacity are significantly lower than the other two buckets, which is an opportunity to save both time and fuel when handling a fixed amount of materials. On the other hand, controlling the large bucket and digging the ground with the large bucket poses greater challenges for position control than the smaller types, due to the higher weight and the amount of material transferred. As Fig. 7 shows, at the beginning of the movement, the large bucket shows more wobbling in the vertical direction. Moreover, the maximum values for the vertical positions when using the large bucket is less than with the other buckets. During bucket substitutions, the bulk modulus of the hoses and cylinder material, and the hydraulic motor size are the same.

Due to the flexibility of the hydraulic system, the large bucket with material exhibits more wobbling than the small and medium buckets. In addition, the value of the rotational inertia while using the large bucket is higher than for the small and medium buckets. Excavator operators play a vital role in controlling the bucket.

Proficient operators accomplish a given material handling task in less time [32]. Bucket selection for the excavator largely depends on the demands of the working environment.

For instance, if the amount of particles to be transferred is important, employing the large bucket can be practical. Place, type of materials and transferring distance are other notable factors for choice of the most appropriate bucket. Once the excavator model is complete, it is exported to a dynamic simulator to analyze its motions. A dynamic simulator gives users, especially proficient ones, the opportunity to get a real feel for the different excavator combinations modeled.

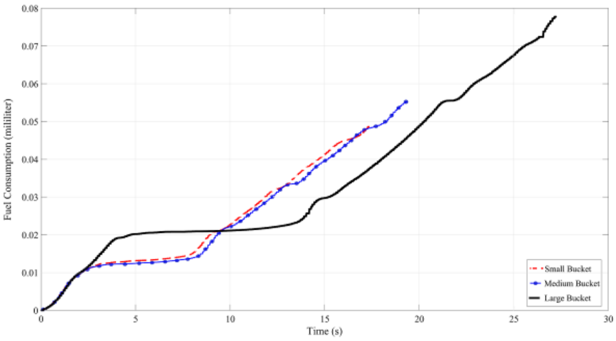


Fig. 8. Fuel consumption for tests with three types of the bucket; 48.6 ml, 55.2 ml, and 77.7 ml for the small, medium, and big buckets, respectively

TABLE V
COMPARISON OF SPECIFICATIONS BETWEEN THREE KINDS OF BUCKET FOR THE EXCAVATOR

Parameters	Small Bucket	Medium Bucket	Large Bucket
Total cycle time (s)	17.4	19.4	27.2
Fuel consumption (ml)	48.6	55.2	77.7
Time (s)/Capacity(m^3)	54.3	38.0	16.7
Fuel Consumption (ml)/Capacity (m^3)	151.8	108.2	30.5

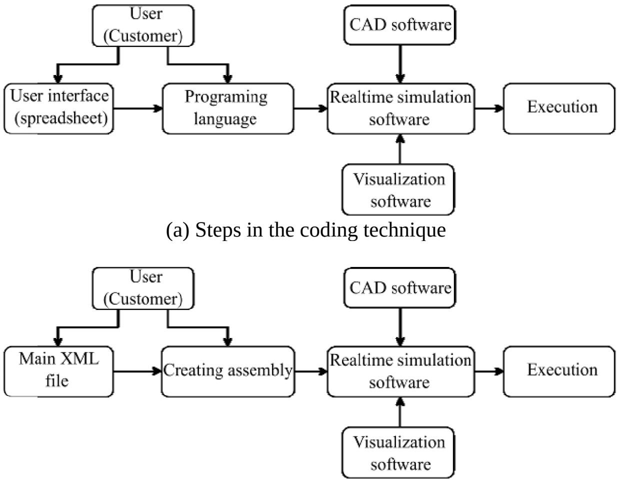
Other researchers also have employed a dynamic simulator to simulate and analyze the performance of moving machine models [33].

IV. Discussion

There are a number of differences between the coding and assembly techniques. With the coding technique, parts and specifications selection can be straightforward because of the availability of the user interface. On the other hand, there is no concrete interface in the assembly file technique.

Therefore, users must have access to the main XML files of the model and manually modify the files to customize the model. Figs. 9 compare the main stages of the two techniques. With the coding technique, users first work with the spreadsheet and then the programming language, Fig. 9(a). With the assembly technique, users can access other parts of the main XML file and manipulate them as shown in Fig. 9(b). Enabling user access to the main XML file can be detrimental as users may unintentionally make model changes. In the coding technique, a designer must work with spreadsheet software and a programming language as well as the real-time simulation software. In the assembly technique, however, only the real-time simulation software, is required. The use of advanced analysis of real-time simulation [34] can also address some economic trends.

Owing to the anticipations of users, each of the aforementioned customized parts can practically be assembled with other parts of the model without any limitations.



(a) Steps in the coding technique

(b) Steps in the assembly technique

Figs. 9. Schematic of the designing procedure of different techniques

The digital tools and simulation are facilitators to respond to lifecycle related requirements, such as sustainability, traceability, repeatability, and the reusability of information in a manufacturing environment. The effective use of product and lifecycle information by simulation enables faster response to changes in customer needs and product-service related requirements.

V. Conclusion

This paper introduced a method to represent a real-time model with adjustable parameters into a real-time simulation software. An eight-step design flowchart was presented illustrating the parameterization procedure. Two techniques, the coding technique, and the assembly technique can be used for the construction of the real-time simulation model with adjustable parameters. With the coding technique, users use a spreadsheet to choose the parameters to be evaluated. A programming language script reads data from the spreadsheet and implements them in the real-time simulation software. With the assembly technique, the parameterization is carried out with specific files called assembly files, which are substituted into the main XML file. The XML file contains all the data to run the simulation. The proposed design approach was applied to a case study of an excavator. Following the eight steps of the design process, parameterized real-time simulation models were developed. The models were used to study the effect of bucket size on the overall performance of the excavator.

The proposed conceptual design process is applicable to other industrial applications in addition to excavators.

The construction of a real-time simulation model consists of several steps and requires information to be gathered from many departments, i.e., mechanical and electrical departments, to a single space. A design process flowchart is proposed to accomplish this in a straightforward manner. The development of a model makes it possible to include various components that can be easily changed.

References

[1] A. A. Shabana, *Dynamics of multibody systems* (5th edition, Cambridge University Press, Vol. 171, 2020).

[2] C. Pappalardo, D. Guida, On the computational methods for solving the differential-algebraic equations of motion of multibody systems, *Machines*, Vol. 6, n. 2, pp. 20, 2018.

[3] C. Pappalardo, D. Guida, On the use of two-dimensional Euler parameters for the dynamic simulation of planar rigid multibody systems, *Archive of Applied Mechanics,* Vol. 87, n. 10, pp. 1647-1665, 2017.

[4] J. J. Laflin, K. S. Anderson, M. Hans, Enhancing the performance of the DCA when forming and solving the equations of motion for multibody systems, *Multibody Dynamics,* (Springer, 2016, 19-31).

[5] M. A. Naya, D. Dopico, J. A. Pérez, J. Cuadrado, Real-time multi-body formulation for virtual-reality-based design and evaluation of automobile controllers, *Proceedings of the Institution of Mechanical Engineers, Part K: Journal of Multi-body Dynamics,* Vol. 221, n. 2, pp. 261-276, 2007.

[6] C. Schwarz, M. Bachinger, M. Stolz, D. Watzenig, Tool-driven design and automated parameterization for real-time generic drivetrain models, In: *MATEC Web of Conference*, EDP Science, Vol. 28, 2015, pp 03001.

[7] J. Ros, A. Plaza, X. Iriarte, JM. Pintor, Symbolic multibody methods for real-time simulation of railway vehicles, *Multibody System Dynamics,* Vol. 42, n. 4, pp. 469-493, 2018.

[8] A. Schmitt, H. Grossert, R. Seifried, Evaluation and experimental validation of efficient simulation models for optimization of an electrical formula car, ASME 2018 *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, American Society of Mechanical Engineers Digital Collection, 2018.

[9] R. W. Du Val, A real-time multi-body dynamics architecture for rotorcraft simulation, *The Challenge of Realistic Rotorcraft Simulation: Proceedings,* London, UK, Vol. 1, 2001.

[10] E. M. Dede, J. Lee, T. Nomura, Governing equations for electromechanical systems, *Multiphysics Simulation,* (Springer, London, 2014, 21-39)

[11] M. J. Tavernini, B. A. Niemoeller, P. T. Krein, Real-time low-level simulation of hybrid vehicle systems for hardware-in-the-loop applications, In: 2009 *IEEE Vehicle Power and Propulsion Conference,* IEEE, 2009, pp. 890-895.

[12] Tutunji, T., Salah, M., Jarrah, A., Ahmad, A., Alhamdan, R., Modeling and Identification of a Four-Bar Linkage Mechanism Driven by a Geared DC Motor, (2015) *International Review of Mechanical Engineering (IREME)*, 9 (3), pp. 296-306.
doi: https://doi.org/10.15866/ireme.v9i3.6035

[13] A. G. Schmitt, *Real-time simulation of flexible multibody systems in vehicle dynamics*, Ph.D. dissertation, Technische Universität Hamburg, 2019.

[14] C. Quesada, D. Gonzáles, I. Alfaro, E. Cueto, A. Huerta, F. Chinesta, Real-time simulation techniques for augmented learning in science and engineering, *The Visual Computer,* Vol. 32, n. 11, pp. 1465-1479, 2016.

[15] J. O. Grady, *System requirements analysis* (2nd edition, Elsevier, 2014).

[16] A. Bahill, Terry and Madni, M. Azad M, Discovering system requirements, In *Tradeoff Decisions in System Design,* (Springer, 2017, 373-457).

[17] D. M. Buede, W. D. Miller, *The engineering design of systems: models and methods* (3rd edition, John Wiley & Sons, 2016).

[18] O. Goury, C. Duriez, Fast, generic, and reliable control and simulation of soft robots using model order reduction, *IEEE Transactions on Robotics,* Vol. 34, n. 6, pp. 1565-1576, 2018.

[19] J. Chenevier, D. Gonzalez, J. V. Aguado, F. Chinesta, Reduced-order modeling of soft robots, *PloS One,* Vol. 13, n. 2, pp. 1-15, 2018.

[20] M. Kirchner, P. Eberhard, Simulation model of a gear synchronisation unit for application in a real-time HiL environment, *Vehicle System Dynamics,* Vol. 55, n. 5, pp. 668-680, 2017.

[21] A. Schmit, R. Seifried, Comparison of various models and integration method for real-time simulation of complex vehicle models with structural flexibility, *Proceedings of ISMA2016 International Conference on Noise and Vibration Engineering,* 2016, pp. 3599-3606.

[22] G. Rubenstein, D. M. Moy, A. Sridharan, I. Chopra, A python-based framework for real-time simulation using comprehensive analysis, In: 72nd *Annual Forum of the American Helicopter Society,* West Palm Beach, FL, 2016, pp. 1-15.

[23] P. M. A. Slaats, *Recursive formulations in multibody dynamics,* Ph.D. dissertation, Technische Universiteit Eindhoven, Instituut Vervolgopleidingen, 1991.

[24] A. Callejo, Y. Pan, J. L. Ricón, J. Kövecses, J. Garcia de Jalon, Comparison of semi-recursive and subsystem synthesis algorithms for the efficient simulation of multibody systems, *Journal of Computational and Nonlinear Dynamics,* Vol. 12, n. 1, 2017.

[25] J. C. Samin, O. Brüls, J. F. Collard, L. Saas, P. Fisette, Multiphysics modeling and optimization of mechatronic multibody systems, *Multibody System Dynamics,* Vol. 18, n. 3, pp. 345-373, 2007.

[26] J. Cuadrado, J. Cardenal, E. Bayo, Modeling and solution methods for efficient real-time simulation of multibody dynamics, *Multibody System Dynamics,* Vol. 1, n. 3, pp. 259-280, 1997.

[27] J. Watton, *Fluid power systems: modeling simulation, analog and microcomputer control* (Prentice-Hall, Inc, 1989).

[28] X. Yang, Y. Shen, Runge-kutta method for solving uncertain differential equations, *Journal of Uncertainty Analysis and Applications*, Vol. 3, n. 1, pp. 17, 2015.

[29] Benbih, H., Gueraoui, K., Driouich, M., Taibi, M., Saidi Hassani Alaoui, M., Modeling and Numerical Simulation of the Motion of a Solid Particle in a Fluid Flow, (2017) *International Review of Mechanical Engineering (IREME)*, 11 (9), pp. 677-682. doi: https://doi.org/10.15866/ireme.v11i9.11385

[30] S. E. Shladover, Review of the state of development of advanced vehicle control system (AVCS), *Vehicle System Dynamics*, Vol. 24, n. 6-7, pp. 551-595, 1995.

[31] T. Gordon, M. Howell, F. Brandao, Integrated control methodologies for road vehicles, *Vehicle System Dynamics*, Vol. 40, n. 1-3, pp. 157-190, 2003.

[32] 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.

[33] R. Ramya, K. Selvi, K. Murali, Real-time simulation and performance analysis of multimachine power systems using dSPACE simulator, *Simulation*, Vol. 92, n. 1, pp. 63-75, 2016.

[34] 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.

Authors' information

Faculty of Mechanical Engineering, Lappeenranta-Lahti University of Technology LUT, Lappeenranta 53850, Finland.

Manouchehr Mohammadi was born in Persia. He received the B. Sc. From QIAU, Persia in 2009, and M. Sc degrees in Department of Mechanical Engineering, from Lappeenranta-Lahti University of Technology, Finland in 2017. He is currently a junior researcher at the Lappeenranta-Lahti University of Technology. His current research interest includes designing of real-time simulation models, multibody system dynamics.
E-mail: Manouchehr.mohammadi@lut.fi

Emil Kurvinen was born in Ristiina, Finland. He received the B. Sc., M. Sc. and D. Sc. (Tech.) degrees in Department of Mechanical Engineering, from LUT University in 2011, 2012 and 2016, respectively. He is currently postdoctoral researcher at LUT. His current research interests include rotordynamics, multibody and machine design.
E-mail: Emil.kurvinen@lut.fi

Aki Mikkola received the Ph.D. degree in machine design, in 1997. Since 2002, he has been working as a Professor at the Department of Mechanical Engineering, Lappeenranta-Lahti University of Technology, (LUT University), Lappeenranta Finland. He is currently leading the research team of the Laboratory of Machine Design, LUT University. His research interests include machine dynamics and vibration, multibody system dynamics, and bio-mechanics. He has been awarded five patents and has contributed to more than 90 peer-reviewed journal articles.
E-mail: Aki.mikkola@lut.fi