INTERNATIONAL DESIGN CONFERENCE – DESIGN 2024
https://doi.org/10.1017/pds.2024.273

Bridging simulation granularity in system-of-systems: conjunct application of discrete element method and discrete event simulations in construction equipment design

Mubeen Ur Rehman ✉, Raj Jiten Machchhar and Alessandro Bertoni

Blekinge Institute of Technology, Sweden

✉ mrc@bth.se

Abstract

The paper addresses a critical challenge in System-of-Systems (SoS) simulations arising from the different granularity levels in SoS simulations, integrating non-coupled Discrete Element Method results into SoS-level Discrete Event Simulations using surrogate modeling. Illustrated with a wheel loader bucket use-case in mining, it enhances early design decision-making and lays the groundwork for improving SoS simulations in construction equipment design. This paves the way for broader research and application across diverse engineering design domains.

Keywords: *system of systems, engineering design, design support system, surrogate modelling, discrete element method*

1. Introduction

The challenges introduced by technological transitions, globalization, and sustainability in engineering design often need to be addressed with a more systemic approach that goes beyond the application of a design method in a specific discipline (e.g., (Bertoni et al., 2021; Tomiyama et al., 2019)). Concepts such as Product Service Systems design (Isaksson et al., 2009), System-of-Systems (SoS) design (Papageorgiou et al., 2020), design of Cyber-Physical systems (Wiesner & Thoben, 2017) are often used to contextualize the development of new methods and applications in specific design contexts encompassing the re-definition of the design boundaries beyond a physical product (often referred as "hardware") to include a broader systems transformation perspective. Computer-based simulations run in various disciplines support the engineering of new systems in approaches framed around set-based concurrent engineering (Sobek II et al., 1999) or model-based systems engineering (Henderson & Salado, 2021). Simulations-based or model-based approaches are applied in a wide array of manufacturing industries to exploit different types of models, experiment design parameters, or assess the expected properties of a system before a decision (Bertoni et al., 2021). However, when designing a new product impacting the effectiveness of an SoS configuration (such as a fleet of vehicles, collaborative network, etc.), there is an ingrained compatibility issue generated by the different levels of granularity of simulations that run at subsystem, system, or SoS level (Maier et al., 2016). The "portfolio" of available simulations can span from detailed finite element simulations of components to machine dynamics to agent-based simulations (ABS) or discrete event-based simulations (DES) of an overall operational context. In such a context, the paper aims to introduce an approach bridging simulation granularity between subsystem and SoS levels, utilizing advanced data science algorithms. This approach is demonstrated through a case study in the construction equipment field facing radical innovation in mining, driven by the technological transition toward electromobility and autonomy. Specifically, the following research objectives are defined:

- Present an approach leveraging data science algorithms to use the results of non-coupled Discrete Element Method (DEM) simulations at a subsystem level to populate DES of the operational performances of the SoS under a given scenario.
- Showcasing applicability of the proposed approach for a wheel loader case, a heavy equipment used for scooping, loading, and transporting materials at a mining site with an articulated bucket mounted at its front end.

The approach aims to reduce uncertainties in early design decision-making related to the impact that a design modification at the subsystem level incurs at SoS levels, especially when the system exhibits temporal dependencies with the operational context. In other words, analyzing a system's operation requires handling sequential data over time because its present state is influenced by past states. The paper is structured as follows: Section 2 presents the research approach adopted by the authors, and Section 3 overviews the relevant literature. Section 4 starts by describing the proposed approach, followed by its application in a demonstrative case involving a wheel loader. Section 5 discusses the findings and concludes the paper.

2. Research approach

The research approach employed in the study is characterized by a two-fold focus, research problem clarification, and definition through Action research (Avison et al., 1999), while the proposal of the approach applied to the specific industrial case can be framed as a case study (Yin, 2009). Three approaches were used for data collection. First, qualitative data were collected during weekly physical and online meetings with industrial stakeholders to explore issues and challenges in enhancing SoS simulations during early design. This supported an iterative cyclic problem-solving process of planning, acting, observing, and adapting, typical of action research. Second, a literature review focused on the combined application of DEM, DES, and surrogate modeling in the systems engineering and engineering design fields. Special attention was given to granularity, system behavior, and performance characteristics to extract valuable insights. Third, a stream of data was generated that stemmed from DEM simulations, enabling the systematic collection of performance matrices of the machine and particle interactions. The data was then fed to machine learning algorithms to analyze further and predict the dynamic behavior of the system of interest. The approach described in the paper reached a validation stage that can be defined as the "support evaluation" stage of the Design Research Methodology as described by (Blessing & Chakrabarti, 2009).

3. Value-driven System-of-Systems design

Systems engineering (SE) is an interdisciplinary approach to realizing an artifact capable of fulfilling the needs of the stakeholders, often referred to as the “system of interest” (INCOSE, 2015). During the development process, the development team must include the aspects that specifically do not belong to the system of interest but can affect its functioning. The collection of such aspects is often referred to as the operational environment or context, giving rise to the concept of system boundary (INCOSE, 2015). This means that the interactions of the system of interest are not limited to internal elements but also include interactions with external elements such as operational environments, users, enabling systems, etc. The aspects within this boundary are usually under the development team’s control. When the system of interest is a system-of-system (SoS), there is an added challenge of understanding the relationship between two or more of the systems, requiring a thorough investigation of the measure of effectiveness of SoS.

Value-driven design (VDD) is one of the commonly used frameworks to drive design decisions throughout the development project in SE (Collopy & Hollingsworth, 2011). To quantify value, the “portfolio” of available simulations can span from detailed finite element simulation of components to machine dynamics, to ABS or DES of an overall operational context. Particularly, when the development team requires a lifecycle-oriented approach in decision-making, DES is one of the most popular choices to quantify value (Jahangirian et al., 2010). DES is a process simulation method where

https://doi.org/10.1017/pds.2024.273 Published online by Cambridge University Press

the operations are performed as a sequence in time. The system's state shall only change at discrete instances and is assumed to remain constant immediately. DES enables the tracing of conditions of sequential processes over time and is relevant for investigating the operational performance of scenarios comprising many interconnected and interdependent systems (Greasley, 2009; Moon, 2017). To include some dynamism and rule-based decision-making in the DES model, it is often combined with System Dynamics (SD) and ABS, known as hybrid simulation (Brailsford et al., 2019). In ABS, entities called "agents" can assess the situation and make the most suitable decision based on rules and logic incorporated in the model. As Brailsford et al. (2019) further argue, such hybrid simulations provide multi-fidelity control over the microscopic operational level view and macroscopic strategic level view.

A powerful simulation method is the DEM, which uses particle dynamics to study the forces acting on the particles due to their interaction with other particles (Bhalode & Ierapetritou, 2020). These simulations are based on Newton's laws of motion and contact laws applied to discrete particle physics. It analyzes the behavior of an object by solving the numerical equations at each time step, focusing on the impingement distance, which represents the displacement to reproduce the behavior of the fine granular material (Iwata et al., 2022). Typical equations used in DEM can be found in Bhalode & Ierapetritou (2020), where it can be seen that the properties of the particles and the machine geometry (the feeder unit in their case) play a vital role in how the material behaves.

Yet another method that is commonly used in SoS simulation is Surrogate or meta-modeling (Yondo et al., 2018). It is a practice primarily used in engineering design to alleviate some of the computational burden by approximating the input-output relationship of a simulation. Artificial Neural Networks (ANN) are generally better at approximating non-linear data or multimodal function landscapes but are limited to static mapping of input-output relationships. For feeding signals from previous timestamps, recurrent neural networks (RNNs) that have recurrent connections are used (Staudemeyer & Morris, 2019). RNNs suffer from exploding or vanishing gradients, so their look-back time is limited in the number of timestamps (Staudemeyer & Morris, 2019). One way to deal with the vanishing gradient problem is to use Long Short-Term Memory (LSTM) networks that can bridge many timestamps (Yu et al., 2019). LSTM networks are a special kind of RNNs, where gate functions are introduced as cell structures, allowing them to handle problems involving large timestamps (Yu et al., 2019). LSTM networks have been widely utilized for various problems, including trajectory prediction, speech recognition, acoustic modeling, etc. (Yu et al., 2019).

4. The approach for System-of-Systems simulations in construction equipment

4.1. The logic of the proposed approach

As mentioned above, DES, often combined with ABS, is one of the choices to quantify value from a lifecycle perspective. In such simulations, the state of the system only changes at discrete times. In Figure 1, the DES can be discretized into several events marked by E. Although the depiction is linear, a DES could have several interconnected and cyclic events. State change for a system is possible only at those events, and the simulation time jumps to the next event. The effect of these transitions is usually user-defined parameters or functions based on historical data, expert opinion, or experience-based judgments, especially in the early design stages when there is a lot of uncertainty about the system-context interactions. Defining these parameters can be challenging if the operation involves interacting with material particles, including material transportation, mixing, compressing, milling, and so on (Ketterhagen et al., 2009).

In the wake of driving these definitions via a simulation, the approach presented in Figure 1 is proposed. A model is needed to capture the effect of state transitions in the DES environment. This model is expressed as $y = \hat{f}(x)$ in Figure 1, where x represents a vector of features (including design and contextual variables), $\hat{f}$ is the model, and y is the desired output. To support the formulation of such a model, the proposed approach unfolds through three distinct steps, marked with numbers in yellow circles in Figure 1. The first step is developing an environment for a DEM simulation based on whether a particle simulation is desirable (marked with Nr. 1 in Figure 1). Defining machine and particle properties are crucial aspects of creating the DEM environment. From an engineering design standpoint, machine properties are the design configuration and control variables that effectively affect its functioning in the given context. Eventually, different control policies (Frank et al., 2018) can be formulated to provide the development team a reasonable grip on various ways to achieve a function. The material properties primarily include particle size, shape, and the number of particles in the environment. Also, a contact model needs to be selected based on the application. With these definitions, a Design of Experiments (DoE) (Yondo et al., 2018) can be run to understand the interactions between the machine and the particles. An additional sampling can be executed and adapted based on the focus of the experiments. For instance, in wheel loader bucket and gravel simulations, the trajectories that did not reach the targeted fill factor (Filla, 2015) can be eliminated. The results of such simulations are then stored in a database and serve as an input to developing the surrogate model.

Training a surrogate model is the second step in the proposed approach (marked with Nr. 2 in Figure 1). The machine and particle interaction are a non-Markovian process. Thus, a history of visited states is necessary to predict the future state in the machine particle interaction. A class of neural networks that allow previous states to be used as inputs is proposed to capture the dynamics of such a non-Markovian process. As seen in Figure 1, the architecture consists of LSTM layers, followed by fully connected layers. A dropout layer is added to avoid overfitting. Finally, since the desired output from this surrogate model is a sequence of data, a regression output layer is added. The hyperparameters of this model, such as the number of layers, number of neurons per layer, dropout ratio, etc., can be tuned until the desired prediction accuracy is reached.

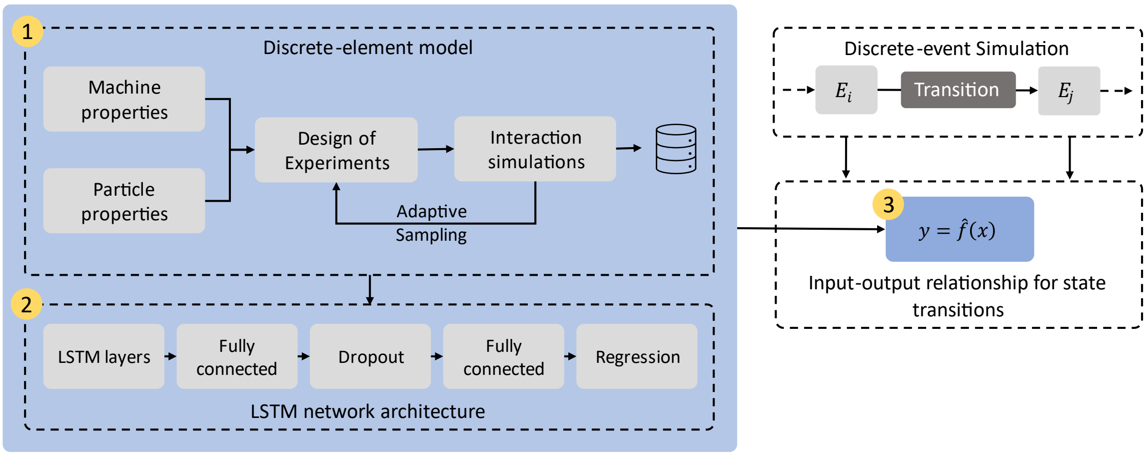

Figure 1. The logic of the approach applied in the case study

The trained LSTM network model can be used in the DES environment to provide a basis for state transition. Since the LSTM network predicts sequential data, integration may be used to calculate the cumulative impact of adopting a control policy. However, different control policies can generate different outputs from the same setup of machines and particles. Thus, the LSTM network is rather seen as a replica of the DEM environment, enabling a reduced computational complexity once the network is trained. Different unseen control policies can then be inputted into this network to generate more data, essentially reducing sampling bias. Such an approach is particularly relevant when a single LSTM network cannot capture the complete dynamics of machine-particle interactions, and a separate LSTM network needs to be developed for different features (like machine properties or particle properties) per se. Thus, the third step in Figure 1 comprises using the data generated from the LSTM network to approximate an input-output relationship for state transitions, i.e., to develop the desired model $\hat{f}$ (marked with Nr. 3 in Figure 1). Since regressions are a static mapping between input and output, integrating sequential data is suggested before performing regression. ANNs may replace the regression model if the input-output relationship is highly non-linear. The goal is to have a reasonable approximation for the machine-particle interaction that can be fitted into the DES model as a transition rule. Figure 1 provides an overview of the complete development process. However, the first and second steps may be excluded once the regression model is constructed.

https://doi.org/10.1017/pds.2024.273 Published online by Cambridge University Press

4.2. Demonstrative case of a wheel loader

The practical application of the proposed approach was realized in an industrial use case, focusing on a wheel loader, a versatile heavy construction equipment also known as a front-end loader. Wheel loaders are crucial in material handling across various industries, effectively scooping up materials like dirt, gravel, or ore and transporting them to their destinations. However, these machines are inherently complex, comprised of multiple subsystems, each contributing to the overall energy efficiency of the process. The demonstrative case takes a pragmatic route of isolating the bucket from the system, i.e., the wheel loader. Four crucial stages during the operational phase of the wheel loader include loading at point A, material transportation from point A to B, unloading at point B, and returning to point A with an unloaded bucket. Within this complete cycle, loading the bucket emerges as a complex task, significantly influencing the overall energy efficiency of the process. As per Filla et al. (2014), in a short loading cycle, the energy consumption rate is highest during the bucking-filling process compared to the other operations. In absolute terms, the process of filling the bucket can represent as much as 40% of the overall energy consumption per cycle. Thus, when developing new concepts for the loading subsystem, including bucket geometry or its control policy, the development team can gain significant advantages from understanding how these elements will interact with the material early in the development process.

4.2.1. Step 1: DEM simulations

In the first step in Figure 1, a commercial DEM tool is used to simulate machine-particle interactions. The required bucket CAD geometry was created and scaled down significantly to reduce the computation time. To have a better understanding of the impact of bucket size on energy consumption, three different kinds of buckets were modeled. For each successive bucket, the enclosed volume was increased by two-fold. The following input is the shape and size of the particles in the DEM environment. To this end, spherical particles were chosen to reduce the computation time. Particles were populated inside the conical shape of a certain angle to represent the gravel pile with some extent of realism. Different iterations were made for the particle generation scenarios. A deterministic approach was employed for populating the conical shape that simulates the gravel pile instead of a discrete probability distribution. This approach involves dense packing, aiming to fill the gravel pile with spherical granular material as efficiently as possible. In contrast, Poisson's distribution may introduce gaps or spaces within the pile. In the DEM environment, particles properties such as density, shear modulus, and Poisson’s ratio were first defined. Following, the particle-to-particle and particle-to-bucket interaction parameters such as coefficient of restitution, friction coefficients, and damping coefficients were defined. The normal and tangential forces were calculated using the Hertz-Mindlin contact model (Bhalode & Ierapetritou, 2020), as the granular particles were considered rigid elastic solids. The simulation time step was set at 20% of Rayleigh's time step to maintain the accuracy of the simulation. A snapshot of the setup used in the DEM environment, with the bucket and the gravel pile, can be seen in Figure 2.

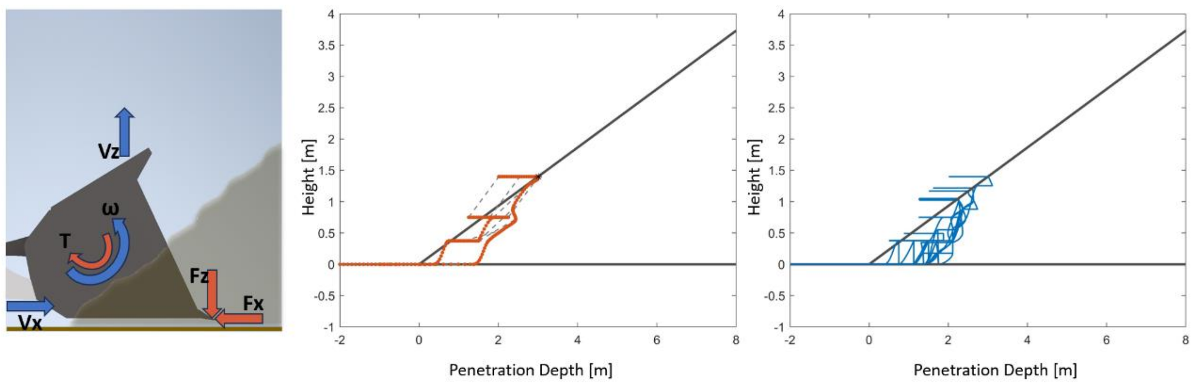

Figure 2. Snapshot of the bucket in a gravel pile and example of simulated trajectories in DEM (single trajectory and compilation of trajectories shown in middle and right images, respectively)

https://doi.org/10.1017/pds.2024.273 Published online by Cambridge University Press

Recursive programming was used to create the trajectories of the bucket moving in the gravel pile (Frank et al., 2018). To keep the problem's dimensionality lower, the bucket's motion was limited to the x-z plane, where x represents the penetration depth and z represents the height. With such a constraint, the bucket had three possible motion actuations during one discrete timestep, the first being a step in x direction, the second being a step in z direction, and the third being a rotational step in the x-z plane about a pivot. Several strategies for filling the bucket can be adopted for this application (Filla, 2015). The “stairway” type was chosen despite not being the most energy efficient for two reasons. Firstly, it can give a wide-spread dispersion of the states visited by the bucket in the gravel pile, and secondly, it can capture the effects of jerky motions in the gravel pile. The recursive programming script would generate numerous trajectories. A DoE was further used to sample these trajectories along with a filter that discards the trajectories that didn’t achieve the desired bucket filling. Figure 2 shows a single trajectory generated via recursive programming (to the left) and superimposition of many selected trajectories (to the right). The DEM simulation data were then post-processed and stored in a database.

4.2.2. Step 2: Training LSTM network

Concerning step 2 in Figure 1, the results from the DEM simulation can be used to approximate the energy consumption for a bucket. However, energy consumption highly depends on bucket geometry, control policy, and gravel properties (Filla, 2015; Frank et al., 2018). Hence, numerous DEM simulations would be required to achieve a more accurate and generalized energy consumption, resulting in high computational complexity. To address this issue, a surrogate model was built from the DEM simulation data that can learn the relationship between bucket geometry, control policy, and gravel properties. 30 DEM simulations were conducted in this study, and the resulting data were divided into training and testing sets following a 90:10 ratio. Consequently, with three distinct bucket sizes, a single trajectory remained reserved for validation. A total of 27 trajectories were used to train the LSTM network, each comprising 130 data points. With a timestep of 0.1 sec, the total trajectory time was 13 secs. This consistent total time was maintained across all trajectories, primarily for the sake of simplicity. Hence, the trajectory data were not padded with synthetic data, even though this function was active during training.

The overall architecture of the LSTM network used for this case study is outlined in Figure 1. By default, it contained a sequence input layer. It was subsequently comprised of three LSTM layers with 300, 200, and 100 neurons, respectively. Figure 3 illustrates one LSTM layer consisting of a series of LSTM cells. With T total timesteps, the flow of data is shown from the input x_t to output y_t for each timestep t. Furthermore, h_t is the hidden state and c_t is the cell state at timestep t. The hidden h_T state is the output y_T at the last timestep if the layer outputs one value. Otherwise, the layer outputs a sequence, such as y_1 ... y_T, equivalent to h_1 ... h_T, in the case of full sequence outputs. Each LSTM operation is executed by an LSTM cell, comprising three gates (forget, input, and output) and a cell state, depicted to the right of Figure 3. Such an LSTM with a Forget Gate was used in this case study, and equations detailing this structure can be found in Yu et al. (2019). The forget gate supports the LSTM in deciding the information to be discarded. The input gate controls the update of the cell state by selecting the information to store. The output gate regulates the information from the current cell state to be passed to the output. In this cell, σ is the sigmoid function used to compute the gate activation.

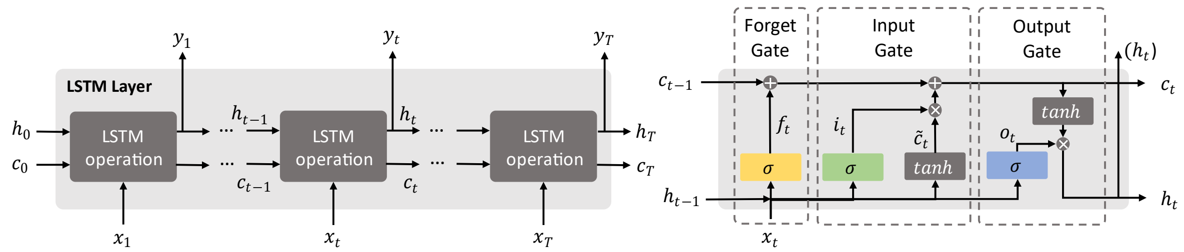

Figure 3. Illustrative LSTM layer (left) and LSTM cell architecture (right)

Following these, a fully connected layer with 50 neurons was added. A dropout layer with a 0.1 dropout rate was included to prevent overfitting. Another fully connected layer was incorporated with the

https://doi.org/10.1017/pds.2024.273 Published online by Cambridge University Press

number of neurons tailored to the desired response count. Finally, an additional regression layer was introduced to calculate the root mean square error (RMSE). All these hyperparameters of the LSTM networks were iteratively fine-tuned until the desired performance was reached. Figure 4 shows the prediction of force in the x and z directions and the torque along the x-z plane for a sample trajectory.

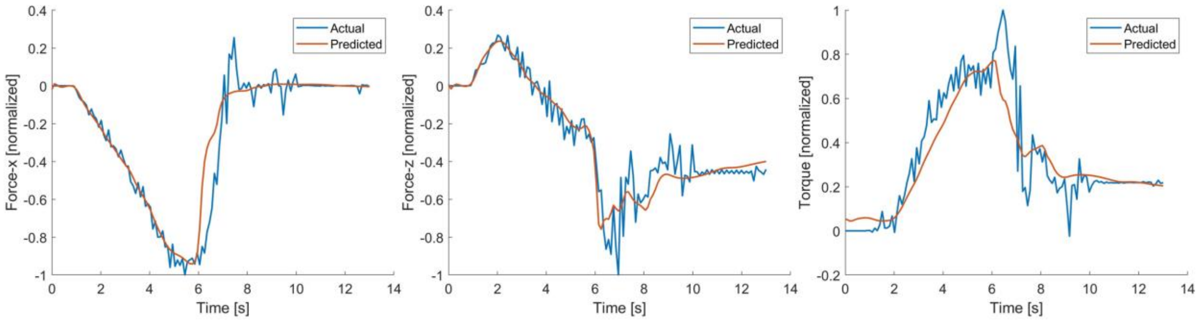

Figure 4. Actual vs predicted values from the LSTM network

From Figure 4, the LSTM network retains its nature but fails to predict the spikes in the forces and torque values over time. While a lower RMSE might appear promising, it becomes problematic when viewed from an integration standpoint within the DES environment. This study establishes a relationship between cumulative work done (or energy consumption) and bucket size, and this relationship is used as the transition model in DES simulation. This energy consumption is achieved by performing a numerical integration on the force and torque curves following a trajectory. The prediction exhibited a suboptimal performance with average error for all the training trajectories for bucket 1 = 10%, bucket 2 = 36%, and bucket 3 = 14%.

4.2.3. Step 3: Regression model from the LSTM network

Despite suboptimal prediction accuracy, the LSTM network was further used to build the surrogate model that can be used in the DES environment. Figure 5 (left) shows the energy consumed for unseen trajectories and different bucket sizes. A total of 19 trajectories were used for each bucket, and eventually, the mean of energy was calculated for the buckets. These averaged values are presented in Figure 5 (right) to create a regression model for future deployment in the DES environment. This model is simply the best fit for the available simulation data to have a reasonable approximation for the machine-particle interaction in the DES environment.

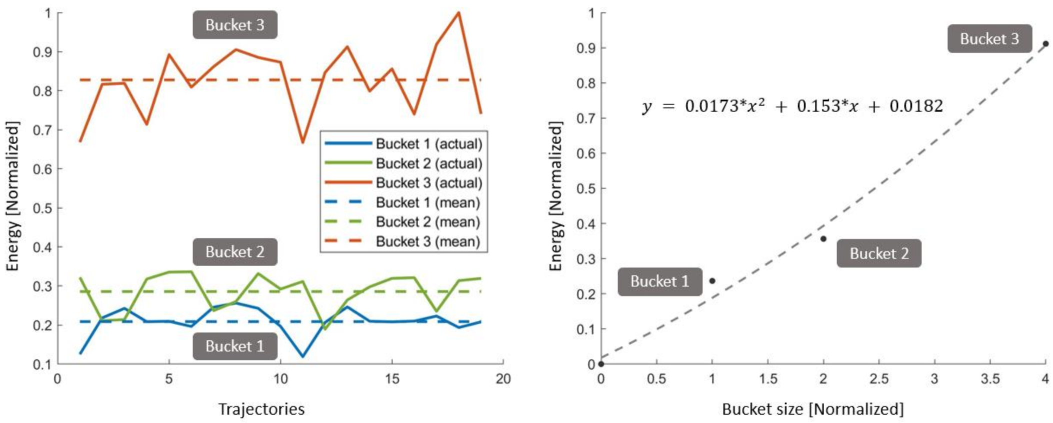

Figure 5. Energy output for different size buckets and unseen trajectories

4.2.4. State transition in the DES environment

A series of experiments were conducted within the DES environment to demonstrate the use of the developed regression model that maps the relationship between bucket size and energy. The study investigated the interplay between bucket size, energy consumption, and productivity. For

https://doi.org/10.1017/pds.2024.273 Published online by Cambridge University Press

demonstrative purposes, three distinct scenarios were developed in the DES environment. Throughout these scenarios, extraction points, routes, and dumping points remained consistent; the variations involved altering the number of haulers, their respective capacities, and infrastructural capabilities, such as processing time. In the first scenario, eight haulers were employed in sets of two, where each set was assigned to a specific extraction route. One set of haulers had a 20-ton capacity, while the other had a 10-ton capacity. Similarly, in the second scenario, a total of four haulers were employed in sets of two. One set of haulers had a 20-ton capacity, while the other had a 12-ton capacity. Finally, in the third scenario, a total of six haulers were employed in sets of two. One set of haulers had a 15-ton capacity, while the other had a 20-ton capacity. The rationale behind a varying number of haulers and their capacity is that it affects the number of scooping cycles and the overall energy consumption. Combinatorics within a set of haulers with different payload capacities was not investigated for this demonstrative case. This setup allowed for analyzing the impact of varying bucket sizes in terms of productivity levels amid the bottlenecks. For this study, each scenario ran for a 24-hour material handling cycle. The result of the first, second, and third scenarios is depicted in a normalized form in the left, center, and right in Figure 6. In this figure, the red line traces the maximum productivity from a scenario where productivity implies ore extracted per day. The 3D perspective can be deceiving; hence, the normalized productivity values are also pasted in round brackets.

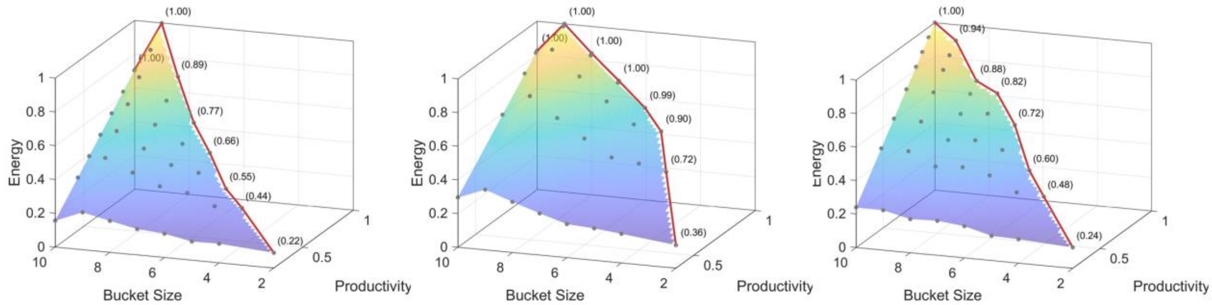

Figure 6. Trade-off maps between bucker sizes, productivity, and energy consumed for different operational scenarios

Typical, without a regression model mapping the energy consumption for different bucket sizes, the rationale for selecting an appropriate bucket size for an operational scenario hinges on a productivity match. Additional knowledge about energy consumption presents a third dimension to design decision-making. For instance, as the bucket size increases, there is a corresponding rise in energy consumption. The grey points in Figure 6 are more interesting since they allow a head-to-head comparison for a specified productivity. For instance, if a productivity requirement needs to be fulfilled from a scenario, two or more bucket sizes can be compared to highlight the one that suits the context better, especially when an exact pass-matching is not possible. Pass-matching implies that the hauler capacity is a multiple of the bucket size. A larger bucket will consume more energy, but a comparison can be made to comprehend the extent of this correlation. In many cases, the wheel loader capacity is not the productivity bottleneck. For example, haulers emerge as a potential bottleneck in scenario 2; thus, the productivity threshold is reached rather early, and any bucket size beyond 6 tons will be underutilized. In such a case, the discussion can revolve around scaling down the bucket size if the energy consumption difference is drastic or assigning the wheel loader to prioritize different tasks during the intermediate time. Also, based on these graphs, the development team can gauge the effective loss in productivity and energy when the most optimal bucket configuration is not used. Such inferences can be a good starting point for discussion if developing a new bucket configuration is necessary.

5. Discussion and conclusion

Applying the proposed logic in the wheel loader case supports engineering design decisions by providing valuable insights into the interplay between bucket size, energy consumption, and productivity. Similar tools are used compared to previous studies (Filla et al., 2014; Frank et al., 2018); however, the underlying objective fundamentally differed. The intent was to add the missing dimensions at an SoS level that can further supplement the overall understanding of the problem and the solution space rather than finding optimal control policies. Given the pivotal role of SoS simulations in early decision-making, the authors argue that enhancing the accuracy of these definitions early in the design process can lead to reduced development time and cost, in line with the argument of Verhagen et al. (2012). However, several challenges and limitations were encountered during the study period, described as follows:

- DEM complexity: Particle shape significantly influences computational efficiency. Opting for spherical particles has notably expedited simulations, potentially at the expense of accuracy. Also, the stairway trajectory was selected in the trajectory generation through recursive programming. It's worth exploring alternative options like the cheese slice or zigzag trajectories (Filla et al., 2014) to compare and determine the most suitable trajectories.
- LSTM accuracy: Using LSTM seems like a viable choice for reducing the computational complexity induced by DEM. However, it seems to regularize the jerks to predict a smoother temporal dynamic response. Despite several iterations, the authors could not accurately represent spikes in the force and torque values. This inaccuracy cascades and gets amplified during the integration of time history, resulting in a suboptimal performance. Investigating if this phenomenon arises from not using an appropriate trajectory or lack of data is beyond the scope of this paper.
- Machine energy: Excluding other wheel loader components introduces a notable constraint. Notably, the study did not account for machine efficiency, a parameter that remained outside the study's scope. Evaluating energy consumption patterns with newer and more efficient machines using a similar approach would provide a compelling basis for comparison with the current findings.

In conclusion, this study can be seen as a step toward advancing the accuracy and applicability of Systems-of-Systems simulations in construction equipment design. The interdisciplinary nature of the research, coupled with the integration of DEM, DES, and surrogate modeling, showcases the applied approach. Future steps involve addressing one of the challenges presented above.

Acknowledgements

The work was performed in the frame of the FELD project funded by the Swedish Innovation Agency (VINNOVA) through the FFI Fossil-free mobile work machine initiative.

References

Avison, D. E., Lau, F., Myers, M. D., & Nielsen, P. A. (1999). Action research. Communications of the ACM, 42(1), 94–97. https://doi.org/10.1145/291469.291479

Bertoni, A., Larsson, T., Wall, J., & Askling, C. J. (2021). Model-Driven Product Service Systems Design: The Model-Driven Development and Decision Support (MD3S) Approach. Proceedings of the Design Society, 1, 2137–2146. https://doi.org/10.1017/pds.2021.475

Bhalode, P., & Ierapetritou, M. (2020). Discrete element modeling for continuous powder feeding operation: Calibration and system analysis. International Journal of Pharmaceutics, 585, 119427. https://doi.org/10.1016/j.ijpharm.2020.119427

Blessing, L. T., & Chakrabarti, A. (2009). DRM: A design research methodology. Springer.

Brailsford, S. C., Eldabi, T., Kunc, M., Mustafee, N., & Osorio, A. F. (2019). Hybrid simulation modelling in operational research: A state-of-the-art review. European Journal of Operational Research, 278(3), 721–737. https://doi.org/10.1016/j.ejor.2018.10.025

Collopy, P. D., & Hollingsworth, P. M. (2011). Value-Driven Design. Journal of Aircraft, 48(3), 749–759. https://doi.org/10.2514/1.C000311

Filla, R. (2015). Evaluating the efficiency of wheel loader bucket designs and bucket filling strategies with non-coupled DEM simulations and simple performance indicators. Schriftenreihe Der Forschungsvereinigung Bau-Und Baustoffmaschinen: Baumaschinentechnik 2015–Maschinen, Prozesse, Vernetzung, 49, 273–292. https://doi.org/10.13140/RG.2.1.1507.1201

Filla, R., Obermayr, M., & Frank, B. (2014). A study to compare trajectory generation algorithms for automatic bucket filling in wheel loaders. 588–605.

Frank, B., Kleinert, J., & Filla, R. (2018). Optimal control of wheel loader actuators in gravel applications. Automation in Construction, 91, 1–14. https://doi.org/10.1016/j.autcon.2018.03.005

Greasley, A. (2009). A comparison of system dynamics and discrete event simulation. Proceedings of the 2009 Summer Computer Simulation Conference, 83–87.

Henderson, K., & Salado, A. (2021). Value and benefits of model-based systems engineering (MBSE): Evidence from the literature. Systems Engineering, 24(1), 51–66. https://doi.org/10.1002/sys.21566

INCOSE. (2015). INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities. John Wiley & Sons.

Isaksson, O., Larsson, T. C., & Rönnbäck, A. Ö. (2009). Development of product-service systems: Challenges and opportunities for the manufacturing firm. Journal of Engineering Design, 20(4), 329–348. https://doi.org/10.1080/09544820903152663

Jahangirian, M., Eldabi, T., Naseer, A., Stergioulas, L. K., & Young, T. (2010). Simulation in manufacturing and business: A review. European Journal of Operational Research, 203(1), 1–13. https://doi.org/10.1016/j.ejor.2009.06.004

Ketterhagen, W. R., am Ende, M. T., & Hancock, B. C. (2009). Process Modeling in the Pharmaceutical Industry using the Discrete Element Method. Journal of Pharmaceutical Sciences, 98(2), 442–470. https://doi.org/10.1002/jps.21466

Maier, J. F., Eckert, C. M., & Clarkson, P. J. (2016). Model granularity and related concepts (D. Marjanović, M. Štorga, N. Pavković, N. Bojčetić, & S. Škec, Eds.; pp. 1327–1336). https://www.designsociety.org/publication/38943/model_granularity_and_related_concepts

Moon, Y. B. (2017). Simulation modelling for sustainability: A review of the literature. International Journal of Sustainable Engineering, 10(1), 2–19. https://doi.org/10.1080/19397038.2016.1220990

Papageorgiou, A., Ölvander, J., Amadori, K., & Jouannet, C. (2020). Multidisciplinary and multifidelity framework for evaluating system-of-systems capabilities of unmanned aircraft. Journal of Aircraft, 57(2), 317–332. Scopus. https://doi.org/10.2514/1.C035640

Sobek II, D. K., Ward, A. C., & Liker, J. K. (1999). Toyota's Principles of Set-Based Concurrent Engineering. MIT Sloan Management Review. https://sloanreview.mit.edu/article/toyotas-principles-of-setbased-concurrent-engineering/

Staudemeyer, R. C., & Morris, E. R. (2019). Understanding LSTM -- a tutorial into Long Short-Term Memory Recurrent Neural Networks (arXiv:1909.09586). arXiv. https://doi.org/10.48550/arXiv.1909.09586

Tomiyama, T., Lutters, E., Stark, R., & Abramovici, M. (2019). Development capabilities for smart products. CIRP Annals, 68(2), 727–750. https://doi.org/10.1016/j.cirp.2019.05.010

Verhagen, W. J. C., Bermell-Garcia, P., van Dijk, R. E. C., & Curran, R. (2012). A critical review of Knowledge-Based Engineering: An identification of research challenges. Advanced Engineering Informatics, 26(1), 5–15. https://doi.org/10.1016/j.aei.2011.06.004

Wiesner, S., & Thoben, K.-D. (2017). Cyber-Physical Product-Service Systems. In S. Biffl, A. Lüder, & D. Gerhard (Eds.), Multi-Disciplinary Engineering for Cyber-Physical Production Systems: Data Models and Software Solutions for Handling Complex Engineering Projects (pp. 63–88). Springer International Publishing. https://doi.org/10.1007/978-3-319-56345-9_3

Yin, R. K. (2009). Case Study Research: Design and Methods. SAGE.

Yondo, R., Andrés, E., & Valero, E. (2018). A review on design of experiments and surrogate models in aircraft real-time and many-query aerodynamic analyses. Progress in Aerospace Sciences, 96, 23–61. https://doi.org/10.1016/j.paerosci.2017.11.003

Yu, Y., Si, X., Hu, C., & Zhang, J. (2019). A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures. Neural Computation, 31(7), 1235–1270. https://doi.org/10.1162/neco_a_01199