Syddansk Universitet

Configuration System for Simulation Based Design of Vibratory Bowl Feeders

Mathiesen, Simon; Natapon Hansen, Michael; Ellekilde, Lars-Peter; Madsen, Ole

Published in:
International Conference on Simulation, Modeling, and Programming for Autonomous Robots

Publication date:
2017

Citation for pulished version (APA):
Mathiesen, S., Natapon Hansen, M., Ellekilde, L-P., & Madsen, O. (2017). Configuration System for Simulation Based Design of Vibratory Bowl Feeders. In International Conference on Simulation, Modeling, and Programming for Autonomous Robots IEEE.

Configuration System for Simulation Based Design of Vibratory Bowl Feeders

Michael Natapon Hansson* , Simon Mathiesen†, Lars-Peter Ellekilde† and Ole Madsen*
*Department of Mechanical and Manufacturing Engineering
Aalborg University, 9220 Aalborg, Denmark
Email:{mhanss,om}@m-tech.aau.dk
†The Maersk McKinney Moller Institute
University of Southern Denmark, 5230 Odense M, Denmark
Email:{simat,lpe}@mmmi.sdu.dk

Abstract— **Vibratory bowl feeders are still among the most commonly used production equipment for automated part feeding, where parts are correctly oriented for further manipulation by being conveyed through a set of orienting devices. Designing vibratory bowl feeders involves selecting and sequencing a number of these devices that either reorients or rejects the part until a desirable orientation is achieved. To aid the designer in this task, this work presents a configuration system where knowledge of the behaviour for each device is acquired through dynamic simulation, and used to solve the configuration task. To test the approach, the configuration system is used to find three device sequences for feeding three parts in specific orientations. The sequences are validated through simulation and real world experiments, showing good consistency.**

I. INTRODUCTION

Automated production and assembly requires that parts can be presented in a stable and correct orientation for further manipulation. This task is referred to as part feeding, where an efficient means is to use of a vibratory bowl feeder (VBF). Figure 1 shows an example of a VBF and illustrates terms frequently used throughout this paper. Bulks of parts can be loaded into a VBF, which uses vibration to produce forward motion of the parts along a track on the edge of the bowl. This separates the parts from the bulk and conveys them to an exit-coupling into the succeeding automation system. Along the track, orientation devices (henceforth referred to as traps) can be employed to control the orientation of the parts according to either a reorientation, or a rejection-and-recirculation principle. Unfortunately, this type of feeding system is a dedicated piece of hardware, in the sense that the bowl can be used to feed just one specific part type. Therefore, reusing a vibratory bowl feeder on a different part requires the bowl to be redesigned.

A standard un-tooled bowl can be made fairly cheap, but when a custom solution is necessary, prices can easily increase fivefold or more. This is a result of the current practise of the design process, where skilled technicians or engineers employ rules of thumb, experience and trial-and-error approaches (as presented in [1],[2] and supported by own investigations). This approach is a result of the complexity of the design problem where the designer has to find the right traps, figure out the order of placement and tune the individual trap parameters to perform optimally. The complexity of the design process leads to considerable variation in design and commissioning time , and the man-hours used in the process accounts for most of the production cost [2].

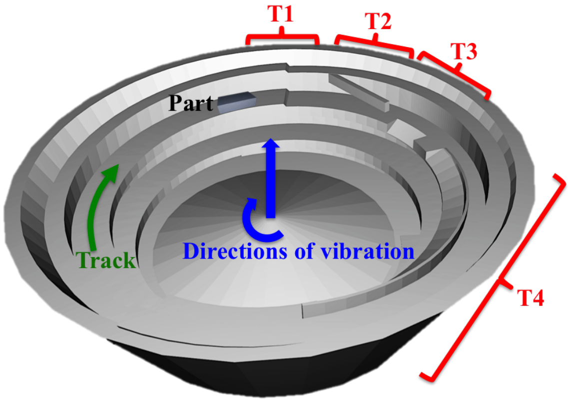


Fig. 1: 3D-model of a vibratory bowl feeder. An example of a part to be fed, directions of vibration, the track and four trap mechanisms (T1-4: *step*, *wiper blade*, *narrowed track* and *edge riser*) are shown.

In this paper, we present our work on making a configuration system capable of assisting the bowl designer in selecting and sequencing traps. Dynamic simulation is used to generate necessary data for facilitating this task as well as verifying results. In Section II, we review related work, and in Section III, the system and the underlying approach is elaborated. Relevant implementation details are listed in Section IV. Section V presents results from the use of the system to configure VBFs in three test-cases.

II. RELATED WORK

The goal of systematising the design of VBFs has been pursued by several researchers during the last decades. Aiding the designer to identify the required traps and sequencing them in a advantageous order have been attempted to be handled by introducing design guidelines and rule-based expert systems. Boothroyd [3] provides a thorough introduction to the mechanical properties to be considered in the design process of a mechanical part feeders. This work includes

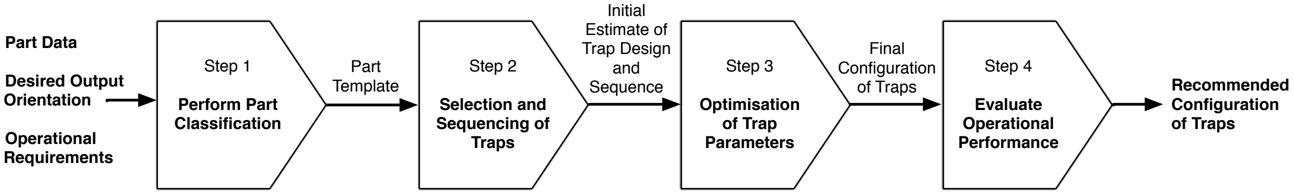


Fig. 2: Design methodology for VBF configuration system

a set of design guidelines to aid in the design of VBFs, including a part classification system. Tan et. al [1] developed an expert advisory system that can be used to recommend an appropriate sequence of traps. The recommendation is achieved by establishing a coding and classification system to describe the geometrical features that can be used to orient a part, and a set of rules to guide the selection and sequencing of traps based on these orienting features. Similarly, La Brooy et. al [4] also developed a rule-based expert system, but has further introduced the possibility of extracting the geometrical features of a part directly for CAD drawings.

Design guidelines and rule-based systems are established by gathering theoretical knowledge and prior experiences, to which they are primarily useful to guide the conceptual design of VBFs. However, the chosen configuration of traps still has to be tested experimentally to ensure that it provides the desired functionality. In this light, researchers have been looking towards using simulation to virtually validate the functionality of the traps. Berkowitz and Canny [5][6] modelled a vibratory bowl feeder using dynamic simulation to predict the behaviour of cuboids and cylinders interacting with a single trap. They present a comparison between simulated and physical experiments, which showed good results in general, but not without notable errors. Jiang et. al [7] described the development of a simulation software for evaluating the behaviour of cubical parts on a predefined sequence of traps. Chen et. al [8] used the multi-body simulation software ADAMS to evaluate the performance of a pre-designed vibrational hopper, feeding five cuboids. Mathiesen and Ellekilde [9] investigated the feasibility of using of the open source tools RobWork [10] and the Open Dynamics Engine [11] for the simulation of VBFs. Although finding inaccuracies in part motion on a microscopic level, the overall interaction between parts and traps has shown promising results.

Simulation has primarily been used to validate a specific design of traps, where the designer alters the trap parameters by observation [12]. To further reduce the production cost associated with manually altering trap parameters in an trial-and-error manner, researchers have been using algorithmic approaches in an attempt to automatise the activity of finding optimal trap parameters. Hofmann et. al [12] describe a simulation-based shape optimisation algorithm, to be used for automatically specifying optimal trap parameters and vibrational amplitude of a VBF. The optimisation algorithm have been tested with a single step trap, feeding a cuboid and a cylindrical-like object. Berretty et. al [13] used geometric algorithms to optimise trap parameters feeding polygonal parts, and have been tested on a narrowed track- and gap trap in sequence. Goemans et. al [2] also describe the use of geometric algorithms for trap parameter optimisation, and have been tested for designing a blade mechanism to reorient or rejecting polygonal parts.

The primary focus of introducing algorithmic approaches has been on optimising the design of individual traps or predetermined sequences of traps, to which a common characteristic is that the design activity of selecting and sequencing the traps is not included as part of the reported work.

To summarise, research has addressed many different aspects of the complex design process for VBFs. However, in order to intelligently facilitate design of bowl feeders, all aspects of the VBF design problem must be covered in its entirety.

III. DESIGN OF CONFIGURATION SYSTEM FOR VIBRATORY BOWL FEEDERS

Configuration can be described as the task of selecting and aggregating well-defined components into an artefact that satisfy a specific set of requirements [14]. Software systems that aid the configuration process are known as configuration systems or configurators [15]. Our goal for a VBF configuration system is that the design process can be automatised to a point where the system will recommend an efficient design of a bowl, given that part description, desired output orientation and operational requirements (such as feed rate) are provided to the system by a designer. Figure 2 illustrates the proposed methodology for such a configuration system and includes four steps; 1) performing classification of part geometry against part templates; 2) finding suitable arrangements of traps, by using behavioural data for available traps against chosen template; 3) optimising each trap parameter against the specific part geometry until the desired orientation is the only occurring output orientation; 4) evaluating performance against operational requirements.

In this paper, we will mainly address step 2 of the proposed methodology, focusing on how to design a configuration system that uses behavioural data, obtained from simulation, to guide the process of selecting and sequencing traps. Furthermore, we cover a sub-part of step 4 by the evaluation of output orientations for entire trap sequences.

A. The Configuration Task

Configuring a VBF involves two main tasks: choosing the required trap types and defining the parameters of these traps. Before that, it is necessary to choose a desired part orientation, O_d, in which the system is to deliver the part. Next, the task is to find an appropriate sequence of traps that forces the probability of occurrence of O_d to 100%. As traps work either by reorienting or rejecting parts in a specific orientation, a simple solution might involve finding a number of traps that reject all undesired orientations, O_u. However, it might not always be possible to find traps that reject all O_u without rejecting O_d as well. Alternatively, a trap sequence can consist of traps that reorient O_u to O_d, or, more likely, a combination of both rejecting and reorienting traps. Rejecting fewer parts will likely have a positive impact on the throughput. Therefore, it is also necessary to consider the ratio of rejection as a secondary parameter when determining a viable sequence of traps. Traps are of course chosen on the basis of their behaviour, but the same trap can be configured with different parameters values (such as the height of a step for toppling over parts), making their behaviour opaque to non-specialists. A configuration can consist of any number of traps in sequence, but in practise there is a limit to how many traps a single device can contain. Given a number of different trap mechanisms to be included in a configuration, N, and a bounding on the size of the sequence, M,

$$\sum_{j=1}^{M} \prod_{i=0}^{j-1} (N - i) \tag{1}$$

yields the minimum amount of evaluations needed to exhaust the entire configuration space under the assumption that a valid configuration only consists of one instance of each trap. If allowing for M of the same trap in a configuration,

$$\sum_{j=1}^{M} N^{j} \tag{2}$$

yields the maximum.

Furthermore, a large number of validations are required for different trap-settings to obtain reliable performance estimates. Solving this trap optimisation problem concurrently with the combinatorial problem of putting together a viable trap sequence, will result in a huge solution space of mixed discrete/continuous parameters. This solution space will shrink significantly if the combinatorial sub-problem is detached. In this work we therefore propose to equip a designer with the tools to establish a good estimate for a viable trap sequence that subsequently can be validated through simulation.

B. Part Representation

In the domain of vibratory feeders the system is highly probabilistic as the motion of the parts acting in the feeder is affected by somewhat chaotic conditions. As a result, sufficiently large numbers of samples are needed to provide reliable results. Collecting behavioural data from physical experiments requires building the system with the possibility for an appropriate amount of parameter variation. This is a time-consuming task, to which simulation provides an alternate approach that reduces this effort significantly.

However, whether collecting data from physical experiments or simulation, the generation of a knowledge repository containing the behaviour of all possible designs of different traps against all possible designs of parts is an impossible goal to achieve.

As mentioned in Section II, a predominant method for establishing guidelines and expert advisory systems for VBFs is through the use of part coding systems that captures the overall geometric shape and features of a part. Here, the basic idea is to establish an encapsulating geometry of the part and then identify significant features such as holes, protrusions, symmetry axes and skewed mass distributions. A major issue raised in the adoption of this process is that it requires extensive cataloguing of geometry-to-features-to-trap-relations and yet does not guarantee an optimal trap performance.

We propose to encapsulate the combination of shape and features in a combined representative shape, including also mass and material properties, that we refer to as a part template. These templates will serve as an abstractive representation of a specific part that will be the outset for selecting and sequencing traps. Matching a specific part to a template is conducted by choosing the template that most closely resembles the features of that specific part. Templates can be defined according to any level of detail, e.g. the shape of a template can range from basic shapes such as cuboids and cylinders, to more feature rich shapes, where the most extreme case will be to have a template that has the same shape as a specific part. Of course, the more feature rich the templates are defined, the bigger the risk is that a template may not be reused as a representative for more than one specific part. As a feasibility study, this work deals in eight instances of a cuboid template, all with uniformly distributed mass, but with different dimensions of aspect ratios based on a fixed height. Three of the templates are used for experiments in section V, these are listed in Table I.

C. Facilitating Selection and Sequencing of Traps

Traps can be seen as individual mechanisms with a behaviour and a set of parameters within a fixed range, and with proper discretisation, which results in a finite number of instances per trap. Having a finite number of templates and trap instances allows a procedural method to be employed for generating the necessary data of all relations. This provides a base of knowledge containing information on the effect of applying different traps to these templates, thus enabling simple browsing for a desired behaviour. Obtaining behavioural data for a specific sequence of traps can be

TABLE I: The three instances of templates used in section V. The material of all cuboids is steel with uniform distributed mass. Height is denoted as h.

Template	Aspect Ratio	Cuboid Dimensions [mm]
Cuboid1	h x (2.0h) x (4.0h)	7.00 x 14.00 x 28.00
Cuboid2	h x (2.0h) x (2.0h)	7.00 x 14.00 x 14.00
Cuboid3	h x (1.5h) x (2.25h)	7.00 x 10.50 x 15.75

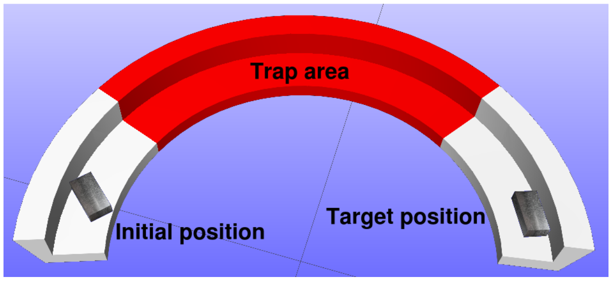


Fig. 3: Data is obtained by letting the parts convey from initial to target position. The area in between, marked in red, is designated *trap area*. This can be of varying length, depending on the number of traps or the length of the inserted track.

determined by aggregating the behavioural data for each trap in that arrangement. Data on the individual behaviour of the traps is obtained from simulation using the setup illustrated in Figure 3.

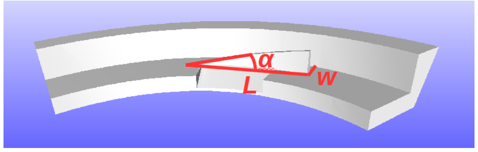
The parts are initialised in a random orientation at the *Initial position* on the track. Next, vibration is applied to the feeder resulting in the parts conveying clockwise along the track until reaching the *Target position*, where their orientations are saved. The *Trap area*, marked with red, can contain traps depending on the chosen configuration of traps.

The system so far covers four commonly used traps as well as a single track type. The track is consisting of a simple flat surface and the four traps are shown in Figure 4 with their associated parametrisation. The parameters of the four traps are set according to the guidelines provided by Boothroyd [3].

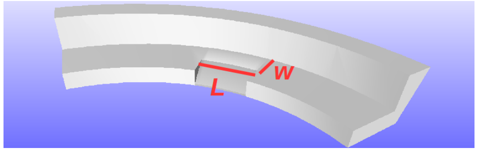
The parts can orient themselves in a number of stable poses on the track. For the cuboid templates, these are described using a finite number of discrete orientations. In the actual implementation, we distinguish between 24 different stable poses in which the cuboids lie on each of their six surfaces with all four neighbouring surfaces up against the track wall. Due to the symmetry of the cuboids, this can be reduced to six unique orientations. These orientations are shown in Figure 5, and for further reference, orientations 1 through 6 will be denoted *O1-6*.

The behaviour of a trap is described by determining the probability distribution of the different output orientations in which a template can occur in after interacting with a trap. This is represented as a histogram, where each bin represents a 3D-orientation of the template with bin height being the probability of occurrence. The actual behavioural data for *Cuboid3* interacting with the four traps is given in Figure 6. Aside from the trap behaviour, Figure 6 also includes the probability distribution of the observed stable poses and the probability of rejection. The stable pose distribution is obtained from 5000 successful simulations of the cuboid reaching the *Target position* from a random starting orientation. The actual orientations from this stable pose distribution are used as input for the trap simulations.

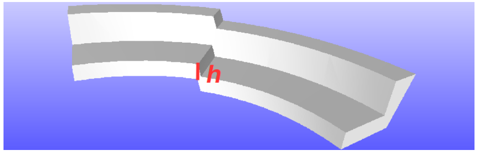
As an example, the figure shows that the *wiper blade* will



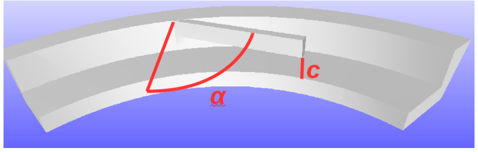
(a) The *edge riser* trap: w is the width from track wall to the rising edge, L is the length of the riser and α marks the angle. The intended purpose of the *edge riser* is to reorient a part to an orientation that can be conveyed between the track wall and the riser.



(b) The *narrowed track* trap: w is the width from the track wall to the slope and L is the length of the narrowed section. The intended purpose of the *narrowed track* is to reject all parts not conveying with their longest axis parallel to the track wall.

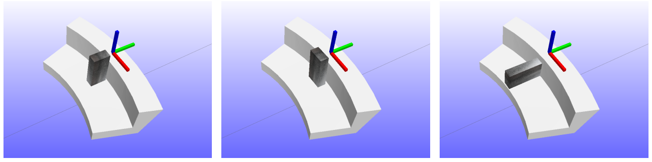


(c) The *step* trap: the height of the step is marked by h. The intended purpose of the *step* is to topple a part to one of its neighbouring stable poses.

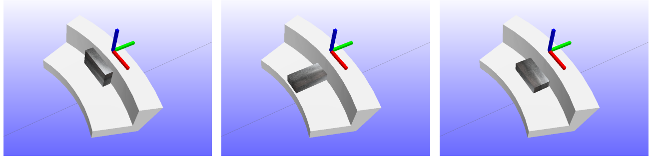


(d) The *wiper blade* trap: c denotes the clearance height from track to blade and α denotes the angle of the blade. The intended purpose of the *wiper blade* is to reject all parts unable to pass under the blade

Fig. 4: The four traps and their associated parameters. The build material for the traps is plastic.



(a) Orientation 1 (b) Orientation 2 (c) Orientation 3



(d) Orientation 4 (e) Orientation 5 (f) Orientation 6

Fig. 5: The stable poses for the cuboid template on a flat track. For illustrative purposes the cuboid orientations are shown using the cuboid template *Cuboid1*.

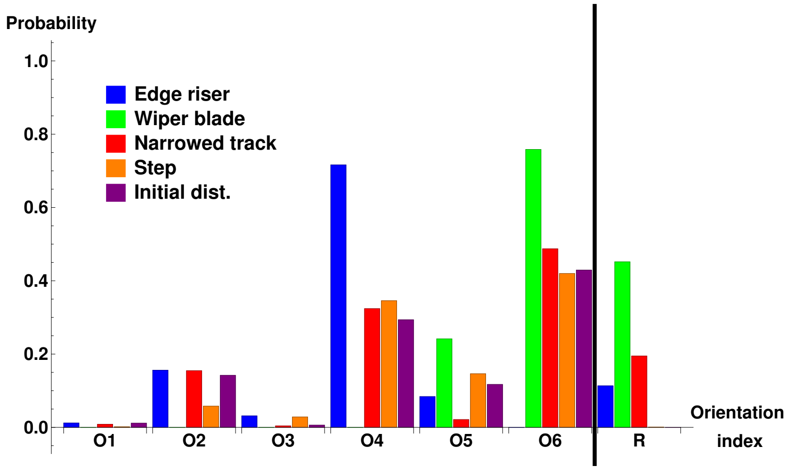


Fig. 6: This shows the probability of occurrence for the six stable poses after being subjected each of the implemented trap mechanisms. The *R*-bins represent the observed probability of rejection. The data shown is for the template *Cuboid3* and also contains information on the initial distribution of stable poses from a random sampling (denoted *Initial dist.*).

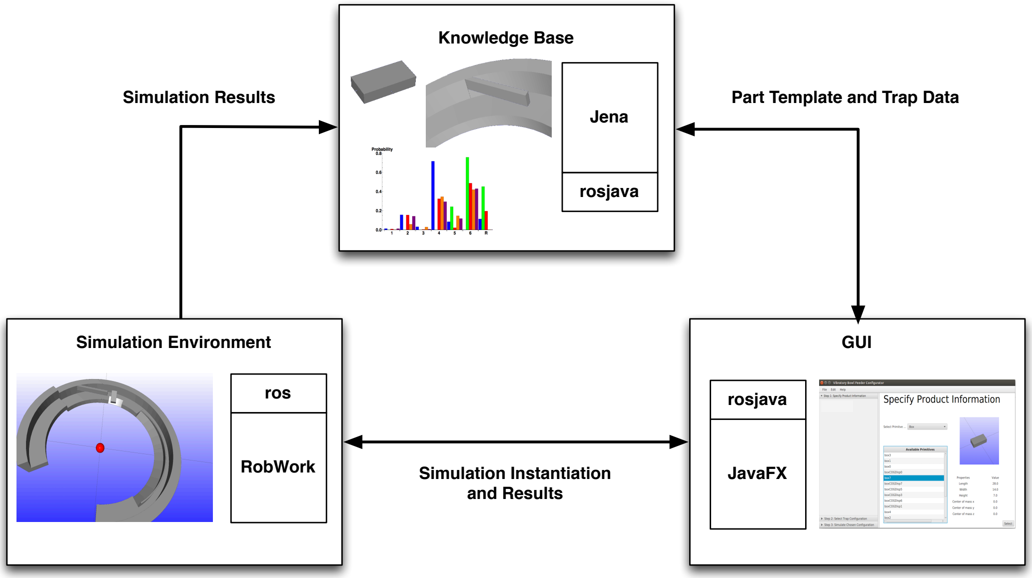


Fig. 7: The configuration systems main components

reject all orientations except *O5* and *O6* and has a high rate of rejection. Similarly the *edge riser* orients the vast majority of cuboids to orientation *O4*, the *step* topples most parts to *O4* or *O6*, and less than 3% of the cuboids are allowed to pass the *narrowed track* in *O3* and *O5*, even though their cumulative probability of occurrence before any traps are inserted is ≈ 12%. Such behavioural data can then be used to facilitate the configuration task described in section III-A, in terms of establishing an initial estimate of the trap design and sequence that may be applicable for a feeding task.

In this paper, the developed configuration system relies on a designer to conduct the reasoning towards choosing a trap sequence. This is facilitated by presenting the designer with behavioural data of traps, which includes the probability distribution of the output orientations as well as the input orientations. The behavioural data of traps of course relies on the sequence itself, to which it is necessary to simulate a specific sequence. As more trap sequences are simulated, individual trap behaviour will be available for the designer in context of a specific sequence.

IV. SYSTEM ARCHITECTURE AND IMPLEMENTATION

Figure 7 shows the main components of the configuration system and consist of: a knowledge base for storing and retrieving data; a simulation environment to facilitate both the acquisition of this data and the validation of a chosen configuration; and a user interface for guiding the configuration process. The communication between each component is achieved through ROS [16].

The knowledge base is implemented using JENA [17], which is an open source Java framework for creating and querying ontological knowledge models. Currently, the onto-

TABLE II: This table lists the setup for the three scenarios with cuboid dimensions (height, width and length), desired output orientation and included traps. Dashes represent no trap. The numbers listed with the traps are the associated parameter values in the following form *edge riser(width, length, angle)*, *narrowed track(width, length)*, *step(height)*, *wiper blade(clearance, angle)*. All measurements are in millimetres except angles which are given in degrees.

	Part	Orientation	Trap 1	Trap 2	Trap 3	Trap 4
1	*Cuboid1*	$O4$	*Step(8)*	*Wiper blade(10, 55)*	*Narrowed track(17, 42)*	*Edge riser(10.5, 250.5, 4)*
2	*Cuboid2*	$O2 \wedge O4$	*Step(8)*	*Edge riser(10.5, 250.5, 4)*	-	-
3	*Cuboid3*	$O6$	*Wiper blade(8)*	*Narrowed track(10, 23.63)*	-	-

logical model is used as a knowledge repository for capturing the orientation probability distribution of trap interaction for the templates. In the future, however, we intend to use ontological models for capturing heuristic knowledge that can be used in the configuration process.

The simulation environment is implemented using the the open source project RobWork [10], which is a framework for robotics simulation. This implementation uses the Open Dynamics Engine (ODE v. 0.13) [11] as the underlying physics engine. Due to inaccuracies in the underlying physics engine, the vibration amplitude in simulation is set to match the conveying speed of the template vibrating on the real feeder, and not the actual vibration amplitude of the real feeder. A detailed description and discussion of this can also be found in [9]. The system is capable of automatic generation of the 3D-models used for simulation of the feeder configuration. The generation of the 3D-models is done using the scripting language and command line tools of OpenSCAD (v. 2015.03) [18]. These models, as well as the models for the cuboids, consists of triangle meshes and is stored in the STL-format.

The graphical user interface is implemented using JavaFX [19]. The purpose of the user interface is to guide the user towards a VBF configuration that will feed the part in the desired orientation. First, the designer can select between the available templates. After a template is chosen, the designer will be able to add traps to form a sequence. This is guided by presenting data on the orientations in which the template can occur after interacting with a specific trap type. Finally, once a sequence of traps has been defined, the designer will be able to initiate a simulation and view the results of the templates orientation distribution against the entire sequence of traps.

V. APPLICATION AND VALIDATION OF THE CONFIGURATION SYSTEM

Three testing scenarios were set up guided by the use of behavioural data. The scenarios consisted of sequences of traps and a cuboid as the part to be fed. They are listed in Table II. Cuboids identical to the ones used for generating the behavioural data is used for validating whether the approach is viable for a scenario where the specific part of interest is identical to the matched template.

The trap configurations were validated using both simulation and a real-world test setup. The validation consists of initialising the parts in some orientation and applying vibration, thus forcing them to interact with the traps after which the orientation was saved to produce a post-trap-orientation distribution. If the traps forced the cuboid off the track, the sample was registered as a rejection, denoted *R*.

Each validation consisted of 300 experiments, where the initial orientations of the cuboids were uniformly distributed into each of the six observed stable poses (50 of each). This was done to ensure direct comparability from a well-defined input, but the uniform representation of all observed orientations also served to stress test the system. For the real-world test the traps were 3D-printed with the same parameter values as their simulation counterparts. The setup for scenario 3 is shown in Figure 8 as an example.

The results from both simulation and the real-world validation are shown in Figure 9.

Results for scenario 1: The trap configuration of scenario 1 gave the desired output orientation for the cuboid with no exception, but it also gave a high number of rejections caused by the *wiper blade*.

Results for scenario 2: In scenario 2, the majority of the output was in the desired orientations *O2* and *O4*, but there were also some occurrences of *O5* and *O6*. This deviation from expected results derives from two problems: *1)* some cuboids starting in *O5* and *O6* topples over at the *step*, thus being reoriented to the undesired orientations *O1* or *O3*; *2)* some cuboids in *O1-O4* were able to climb the *edge riser* and lean against the track wall until falling down in orientation *O5* or *O6*. Although more frequent in simulation, this phenomenon also occurred on the real feeder.

Results for scenario 3: The validation of scenario 3 in simulation showed that $\approx 86\%$ of the cuboids were oriented in *O6* after the traps, whereas $\approx 14\%$ passed the traps in *O5*, which the *narrowed track* should reject. In comparison with the real-world data the percentage of wrongly oriented cuboids were $\approx 4\%$.

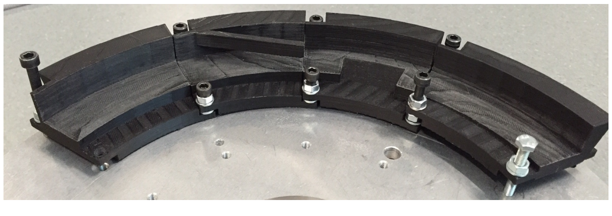
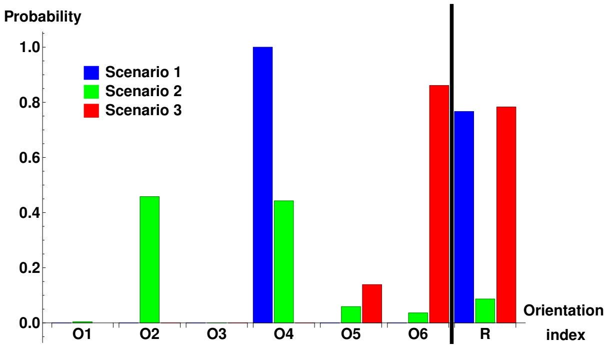
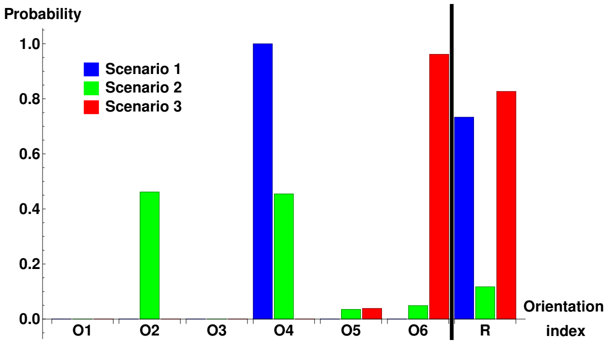


Fig. 8: Example of the real experimental setup. This shows the feeder configuration of scenario 3 with the 3D-printed traps



(a) Results from the three scenarios run in simulation.



(b) Results from the three scenarios tested on the physical platform.

Fig. 9: This figure show the probability of occurrence for the six stable poses after being subjected to the three trap sequences. The *R*-bins represent the observed probability of rejection.

VI. DISCUSSION

Although the trap configurations tested in the previous section does produce the overall desired output orientations of the parts, there are some rather obvious improvements that can be made. In scenario 3, it is the *narrowed track* trap that does not perform adequately. The possibility of this failure is already evident from Figure 6, where there is a small probability that orientation *O5* will occur. This probability grows from 2 percentage points to the 14 percentage points of Figure 9a due to the artificial distribution of start orientations imposed on this test, but even so, the trap parameters need adjusting to produce the desired behaviour.

For scenario 2, it could be argued that the trap configuration cannot adequately ensure the desired orientation. As in scenario 1, a *wiper blade* could be employed to deal with the standing cuboids that causes the problem, but a more desirable approach that does not reject as many cuboids could possibly be found by investigating whether another parameter value for the step height could guarantee that all cuboid were reoriented to orientation *O5* or *O6*. Similarly, a possibility for improvement of scenario 1 could be made by having an adjustment of the *wiper blade* so that it only rejects cuboids in orientation *O1* and *O2*, thereby producing a lower rejection ratio. However, this is not without risk, as it is likely to increase the probability of the cuboids being knocked over instead of being rejected, thus leading to problematic jamming situations when the cuboids are fed in succession. Additionally, it is worth stating that the high rejection ratio in scenario 1 is an unrealistic measure because most of the observed rejections comes from the cuboid being initialised in orientation *O1-5*, whereas the cumulative probability of occurrence for these orientations is ≈ 0.44. Using the actual probability distribution of initial orientations would result in a considerably lower probability for rejection than the shown ≈ 0.8.

All the problems discussed here could potentially be solved using two different approaches. One option is to accept the inaccuracies of the pre-generated data and leave the issue to be handled by the parameter optimisation represented by step 3 of Figure 2. This could involve tuning the *narrowed track* trap of scenario 2 to filter out all parts with orientation *O5*. A second option would be to acknowledge that the trap parameters for the specific instances were wrongly set and that more complete data needed to be produced. Having more instances of a specific trap, were parameters are set in steps of a fixed discretisation will provide the designer with better possibilities to find a sequence of traps with a desired behaviour (but with risk of increasing complexity in terms of data generation and choice navigation). This could potentially also shine a light on traps having multiple functionalities, such as the previous discussion on the *wiper blade*. That said, with the introduction of real parts instead of templates there is no way around the need for a parameter optimisation for the traps because the required amount of pre-generated data would otherwise be infinite (you cannot produce data for every conceivable part). Therefore, we believe that a combination of the two is necessary for a complete solution. Here the discretisation of the trap parameter values remains to be explored.

VII. CONCLUSION AND FUTURE WORK

In this paper, we have discussed the difficulties of vibratory part feeder design and presented an approach for the design of vibratory part feeders. The approach is based on simulation for the priori generation of the data needed to find a feasible feeder configuration. As it is impossible to generate data for all conceivable parts, data is generated only for representative shapes, denoted templates, that we believe can provide sufficient coverage of the solution space to aid a designer in his work. A configuration system has been implemented for solving the trap selection and sequencing problem of vibratory feeder design. Our tests show that given the assumption that a template has been chosen, a feeder configuration can be found from the data provided for a user of the system. Three feeder configurations have been tested for three different cuboid templates. Using the implemented configuration system, the three configurations were generated as 3D-models and then validated through simulation to show that the decoupled part/trap-interaction data, provides meaningful estimates for joined part-trap-interaction. The simulation results have then been validated on an equivalent real feeder configuration, and results for the simulation and the real-world results showed good consistency. These results suggests that the proposed approach is promising for aiding the design of vibratory bowl feeders.

As mentioned in the discussion, there is still work to be done in determining a feasible discretisation of trap instances to provide a complete solution space for a designer to explore. Although the system allows for the validation of feeder configurations with continuous trap parameter values, the solution space is too large to explore manually and therefore the application of an automatic optimisation technique is required. This has been explored by Hofmann et al. [12] to some extend, but not for multiple trap mechanisms in sequence. We are also currently looking into an approach for solving the selection and sequencing problem automatically as well investigating whether classifying an object to a representative shape/template can be left to the designer or whether a guided or fully automatic approach is needed.

ACKNOWLEDGEMENT

This work was supported by The Danish Innovation Foundation through the strategic platform MADE-Platform for Future Production.

REFERENCES

[1] P. Tan, B. Ngoi, S. Lee, and L. Lim, "A knowledge-based advisor for the automatic selection and sequencing of orienting devices for vibratory feeding," *Engineering Applications of Artificial Intelligence*, vol. 8, no. 1, pp. 1 – 13, 1995.

[2] O. C. Goemans, K. Goldberg, and A. F. van der Stappen, "Blades: a new class of geometric primitives for feeding 3d parts on vibratory tracks," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.* IEEE, 2006, pp. 1730–1736.

[3] G. Boothroyd, *Assembly Automation and Product design*, 2nd ed. CRC Press, 2005.

[4] R. La Brooy, C. Jiang, P. Chan, *et al.*, "Design of an expert system for automatic high-speed parts feeding," in *Sixth International Conference on Manufacturing Engineering: Manufacturing; a Global Perspective; Proceedings, The.* Institution of Engineers, Australia, 1995, p. 359.

[5] D. R. Berkowitz and J. Canny, "Designing parts feeders using dynamic simulation," in *Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, vol. 2, Apr 1996, pp. 1127–1132 vol.2.

[6] ——, "A comparison of real and simulated designs for vibratory parts feeding," in *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, vol. 3. IEEE, 1997, pp. 2377–2382.

[7] M. H. Jiang, P. S. K. Chua, and F. L. Tan, "Simulation software for parts feeding in a vibratory bowl feeder," *International Journal of Production Research*, vol. 41, no. 9, pp. 2037–2055, 2003.

[8] R. Chen, L. Chen, X. Wang, and X. Chen, "Dynamic design and simulation of a vibratory hopper," in *Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC), 2011 2nd International Conference on*, Aug 2011, pp. 3935–3938.

[9] S. Mathiesen and L.-P. Ellekilde, "Configuration and validation of dynamic simulation for design of vibratory bowl feeders," in *2016 12th IEEE International Conference on Control and Automation (ICCA).* IEEE, 2016, pp. 485–492.

[10] L.-P. Ellekilde and J. A. Jorgensen, "Robwork: A flexible toolbox for robotics research and education," in *Robotics (ISR), 2010 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK).* VDE, 2010, pp. 1–7.

[11] "Open dynamics engine," accessed: 24-08-2016. [Online]. Available: http://www.ode.org

[12] D. Hofmann, H. Huang, and G. Reinhart, "Automated shape optimization of orienting devices for vibratory bowl feeders," *Journal of Manufacturing Science and Engineering*, vol. 135, no. 5, 2013.

[13] R.-P. Berretty, K. e. Goldberg, M. H. Overmars, and A. F. van der Stappen, "Trap design for vibratory bowl feeders," *The International Journal of Robotics Research*, vol. 20, no. 11, pp. 891–908, 2001.

[14] D. Sabin and R. Weigel, "Product configuration frameworks-a survey," *IEEE Intelligent Systems*, vol. 13, no. 4, pp. 42–49, July 1998.

[15] A. Felfernig, L. Hotz, C. Bagley, and J. Tiihonen, *Knowledge-Based Configuration*. Morgan Kaufmann, 2014.

[16] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.

[17] "Jena," accessed: 24-08-2016. [Online]. Available: https://jena.apache.org/

[18] "Openscad scripting language and command line tools," accessed: 19-08-2016. [Online]. Available: http://www.openscad.org/

[19] "Javafx," accessed: 30-08-2016. [Online]. Available: http://www.oracle.com/technetwork/java/javase/overview/javafx-overview-2158620.html