CEIG - Spanish Computer Graphics Conference (2022)
J. Posada and A. Serrano (Editors)

Simulation based initial feasibility analysis pipeline for small-sized part picking

I.Mendizabal-Arrieta[1], A. Tammaro[2], M. Ojer[1] and X. Lin[1]

[1]Vicomtech Foundation, Basque Research and Technology Alliance (BRTA), Mikeletegi 57, San Sebastian-Donostia, 20009, Guipuzcoa, Spain
[2] ant.tammaro.prof@gmail.com

Abstract

Random bin picking is still one of the main tasks for robotics in the current days. When the environment is very cluttered, the calculation of grasping positions can be highly demanding in terms of time and computing power. To ease the computation load, some parts arranging operations can be performed before the segmentation stage. For instance, for small and light parts, a feeder-vibrating table system can be used to separate the components, allowing them to be easily grasped, and increasing the overall performance of the solution. However, as the geometry and requirements for piece type are different, one or more feasibility tests need to be done for each case. These analyses are usually very time and cost intensive and require the use of expensive hardware such as robots, grippers, and prototype cells. The use of virtual reproductions of the environment like digital twins or physical-based simulations could help reduce the time and effort spent on designing the settings, nevertheless, their correct configuration is not trivial. This paper presents a simulation based analysis method for picking small-sized parts. It aims to supply the tools and define a streamlined procedure for efficient feasibility testing. Those concepts are applied in a specific bin picking scenario of multiple small electronic components. For each part type, a set of case-specific initial and boundary conditions are taken into account, then a series of performance metrics for both bin and vibrating table part picking are computed. The obtained information is decisive to make strategic decisions regarding the hardware requirements, the profitability, and the success probability of the project.

CCS Concepts

• ***Computer systems organization*** → *Robotics;* • ***Computer graphics*** → *Physical simulation;* • ***Hardware validation*** → *Simulation and emulation;*

1. Introduction

Industry 4.0 is based on several technologies that have improved and settled in recent years. Robotics, high-quality simulations, and system integration flexibility are some of them ([JASG21], [OG20], [GCSR21]).

One suitable field of application of these techniques is the industry of electronic component manufacturing. Here, a variety of small parts, such as Pin-Through-Holes (PTH) components, need to be taken from bins or containers and mounted with high accuracy on Printed Circuit Boards (PCBs). Currently, those operations are performed by specialized machines or human operators. The former excels in productivity and accuracy, but lacks flexibility for manufacturing small batches, which many state as a goal of Industry 4.0 ([ANL*20], [SKWK20]), [MW17]. The latter on the contrary, has low productivity but great flexibility and adaptation to new references. The use of robots in this task could be a compromise between both of them.

The mentioned task of picking parts from bins is known as Bin Picking ([Buc16]), and it is one of the most common tasks associated with robotics. The parts to be grasped can be arranged in a structured, or random formation (see Figure 1). The latest has been deeply studied in the recent decades, because the upstream cost of randomly piling parts is significantly lower than placing them in a structured arrangement. For the specific case of light and small pieces, another solution exists that can be applied to randomly arranged parts: the use of feeders and vibrating tables (see Figure 2). These systems allow having a controlled number of components simultaneously on a table that has the capacity of shaking the parts in up to three directions (X, Y, Z). This way, they can be rearranged and placed in a way that a robot can grasp them. Some electronic components might meet this weight and shape criteria, which allows the use of vibrating tables to pick them.

Bin picking and vibrating table solutions usually require an extensive analysis specific to the geometry and characteristics of the part, which adds cost and delays the deployment time. These analyses take into account vision systems and robots, and output metrics such as the part picking success rate or cycle time. However, this

DOI: 10.2312/ceig.20221139

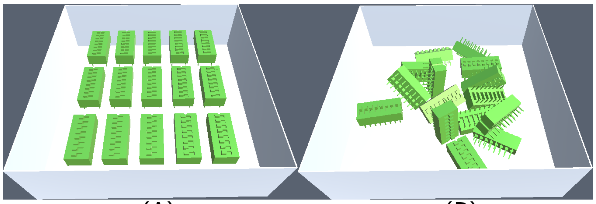

Figure 1: *Possible part arrangements in bins: (A) structured, (B) random.*

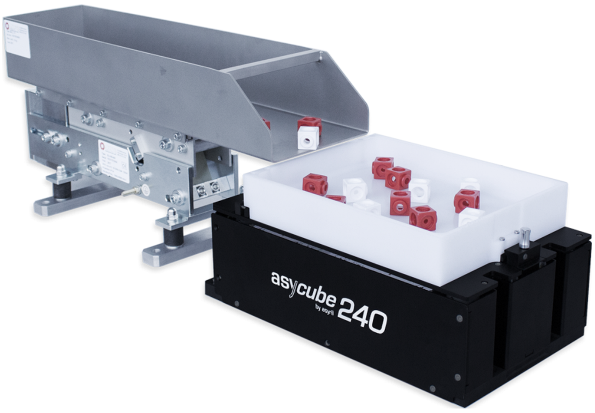

Figure 2: *Commercial vibrating table ©Asyril Asycube 240, Switzerland (source Asyril).*

paper proves that before experimenting with physical hardware like cameras or robots, virtual simulations can be used as a primary tool to validate if a prototype is feasible or not. By creating a virtual copy of the environment using detailed 3D models, reproducing the sensor's output and the robot behavior, and utilizing an accurate physics engine, it is possible to systematically test multiple solutions with relatively low effort.

This paper proposes a parametrical pipeline to analyse the feasibility of picking small components, with different settings, e.g. bin and vibrating table, based on virtual simulations. In Section 2 the literature related to cell optimization, part picking, and small part grasping is analysed. Section 3 explains the methodology used in the pipeline. In Section 4, how simulations were designed and implemented is explained. Also, a test-case example is described. Section 5 shows the results of that test-case. Finally, Section 6 states the obtained conclusions and the future work.

2. Related work

Robotics is one of the backbones of Industry 4.0. However, during operation, robots usually acquire data from other sensors (e.g. 2D, 3D cameras, safety light curtains) and physically interact with other components (e.g. more robots, bins, or conveyors). All these elements together shape what is called a robotic cell. Since these independent units need to work together, their position and the way they interact in the workspace is a key component for achieving the required production needs. For these reasons, many researchers have been analyzing the problem. Zhang et al. [ZF17] for example, look for the main challenges to get a good design. It mentions the importance of choosing the correct type and number of robots, the cell layout, the tasks scheduling, a tool design optimized for the task, and the importance of human-robot collaboration. Another study ([BSG*15]) talks about how robotic cells can be designed in a modular way, to increase the flexibility of adding, removing or moving the robots that compose it. Laemmle et al. ([LG19]) mention the benefits of using simulation tools for such a task. But, they also show how time-intensive the task of generating a correct virtual scenario can be. Additionally, Gadaleta et al. ([GBP17]) analyses the reduction of the energy consumption of the cell depending on its layout design. However, no literature has been found that specifically addresses the bin or vibrating table design optimization problem.

Research about Bin Picking is developed in many directions. [SK14] shows a general virtual framework for path planning, design, and simulation of Bin Picking applications. Regarding motion planning, [MKS*21] uses Deep learning to separate entangled parts; [IDX*20] proposes an algorithm that takes into account the dynamics of the robot and some candidate grasps (generated by a grasp planner) to produce optimal planning; and [VVS17] uses motion primitives for generating the planning. Finally, [FDN*20] and [MTS*20] research new metrics and benchmarks to evaluate Bin Picking.

Innovation has also been applied in the design of the grippers. [BG18] for example, proposes a two-finger gripper to grasp flat parts on a surface. [ZWW*20] on the other hand, proposes a gripper with a rotational degree of freedom on the jaws to rotate the parts in the gripper without involving the robot joints in it. Finally, other studies (such as [DTA20]) propose custom grippers for picking parts in cluttered environments.

Regarding virtual simulations, in the last years, an increasing adoption of digital twins in industrial settings has been seen. Especially during the design phase of manufacturing lines and intelligent manufacturing [ZZL*20]. In [KKT*18] the authors categorize the relation between a digital model and its corresponding physical model, based on the level of integration and the data flow between the real and simulated world. Nowadays exist many simulation environments that have a built-in physics engine and can produce high fidelity results. Some are: ROS [QCG*09], the Robot Operating System which with his second iteration fixed many issues that prevented it to become the industry standard [MVPR*20]; Pybullet [CB16], a python-based physics engine very popular in deep reinforcement learning applications; MuJoCo [TET12], and advanced physics simulator with a focus on contact forces calculation; Unity, a 3D simulator supported by the Nvidia Physx engine, initially designed for creating video games, but that has been increasingly adopted to simulate industrial environments and has good integration with 3rd party tools, e.g. ROS [SHT*17].

3. Methodology

The goal of this paper is to formulate a procedure for small part picking analysis and test its effectiveness. Therefore, in this section we present a standard methodology that models this procedure. In practice, it takes a real use case as input and abstracts it by forming a simpler simulated environment with respect to the task requirements and boundary conditions. This formulates the real use case verification problem as a task of searching an optimal solution in the solution space provided by the constrained simulated environment. To evaluate the solution, validation strategies are exploited with respect to the specific task involved in the use case. Finally, the analysis is performed on top of repeated experiments. Certain metrics are used to compare the solutions with the test-case requirements. In the following paragraphs, the proposed standard methodology is explained along with some examples in specific tasks:

The requirements for a use case are normally task-related. For instance, in a bin picking task, those consist in the number of parts to be grasped per hour or to be able to provide a continuous flow of parts during certain amount of time.

The solution space is created by simulating the real use case with boundary conditions, where trade-offs are made between the fidelity and the complexity of the solution space. Boundary conditions constrain the solution space with respect to the specific use case.

The validation strategy describes how good a solution is. In a bin picking task, it could be a binary function which verifies that a part can be picked and placed without colliding with the scene. A more sophisticate validation strategy could also be defined, such as given a grasping pose, the probability that a path planner can solve the path planning problem within a time limit.

The metrics define how the repeated experiments will be analysed comparing with the requirements. Commonly used metrics are the probability to find a valid part or emptying a bin in a picking task.

4. Implementation

Here is presented an implementation of the general methodology explained in Section 3. This section explains its specific details and extends it with a test-case example.

4.1. Part validity analysis

The core concept behind the feasibility tests is the definition of a graspable piece. A test piece is considered valid for pick-and-place if it satisfies a series of subsequent constraints. The testing is performed following the specific order shown in Figure 3, and if one requirement is not met, the subject is considered not valid for pick and place. To satisfy a condition, all previous ones need to be satisfied too. In the following paragraphs, these constraints will be presented in detail.

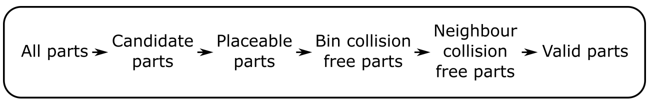

Figure 3: *Part validity process.*

A part is considered a valid candidate if it is placed in a position close to one or more grasping poses predefined by the user. The definition of these poses is given by the axis of the part that is aligned with the world axis Z (Z^w) and by the one aligned with the gripper closing direction. Figure 4 shows two grasping pose examples. If the orientation of the analysed part is closer than a rotation of 45º in any direction from a user-defined grasping pose, the subject is considered a valid user candidate ([IDX*20]). In that case, all the pose information is saved (e.g. grasping position, gripper opening during grasping). In Figure 5 it can be seen how by changing the part angle, the chosen user-defined pose is different.

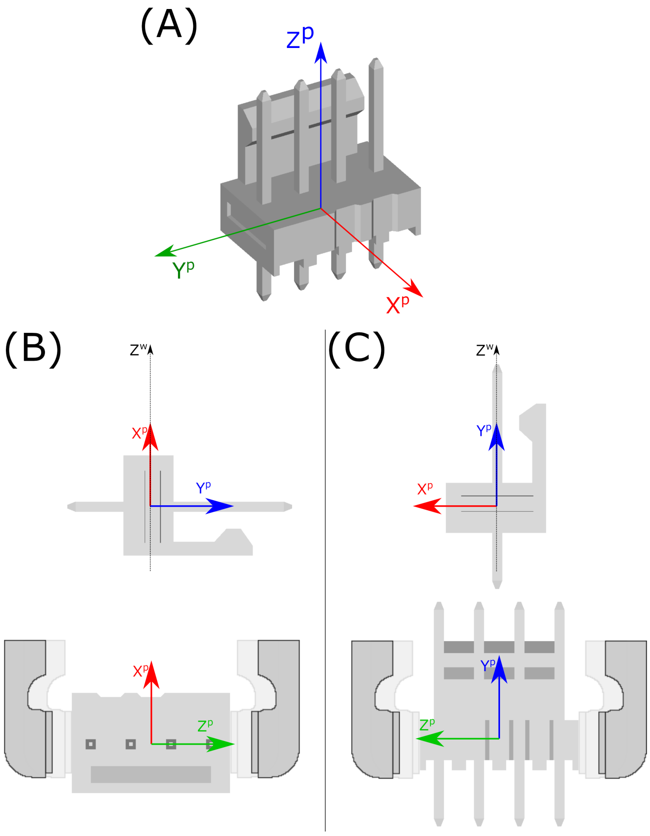

Figure 4: *Example of user defined grasping poses: (A) isometric view of the PTH, (B) is defined by part axis X^P (red) coinciding with world coordinate Z^w, and (C) by part axis Y^P (blue). In both cases the gripper closes parallel to part axis Z^P.*

Once the part was confirmed as a candidate, its placeability is tested, because there might be some cases in which the gripper collides the table when placing the parts on the target region. Since the case of a parallel jaw gripper is considered, this is done by iteratively looking for gripper-table collisions, in a range of rotations specified by: the grasping point, the jaw closing axis, and a range of angles (-45º to 45º in this case [IDX*20]). Figure 6 presents the case in which a grasping pose is considered valid for grasping but not for placing. In (A) the minimum (pose 1) and maximum (pose 2) values of the range are collision-free, while during the placing operation (B) the relative position of the grippers and the parts are kept the same, but they are rotated 90º to the placing orientation of

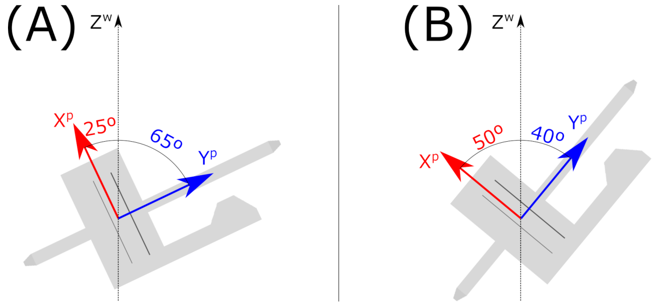

Figure 5: *Example of influence of part angle in chosen user-defined pose: (A) X^p (red) has an angle of 25º respect of Z^w. Therefore, the grasping pose defined by X^p is chosen; (B) Y^p (blue) has an angle of 40º respect of Z^w. Therefore, the grasping pose defined by Y^p is chosen.*

the part. It can be seen that position 1 collides against the placing surface. Therefore, that particular orientation is not valid. All valid orientations are saved for the next steps.

At this stage, the remaining poses are tested for collisions between the gripper and the container (see Figure 7). The simulator built-in physics engine is used to compute collisions and interactions between objects. In this and the next step, we used a simplified version of the real gripper for faster collision checking (see Figure 8). The gripper model is approximated by a cylinder for the body and a cuboid for the workspace of the jaw. Parameter D is the diameter of the gripper, which is kept constant throughout the whole project; U is the width of the jaws in the approach position. This value depends on the part to be grasped as well as the grasping orientation; parameters V and W are the height and depth of the jaws, which are also kept constant.

The last stage is similar to the previous one and involves the collision check between the gripper and the other pieces close to the target (see Figure 9). The tests are also made with the built-in physics engine and if there are no neighbour parts that could interfere with grasping, the part is finally considered valid and ready to be picked.

4.2. Simulator

The simulator was developed using ©Unity, USA, V2020.3.24f1. We chose it because it has a high-fidelity physics engine, namely PhysiX SDK 4, with support of joint articulations and very fast collision detection. In addition, this engine is very famous and therefore has a very rich library of plugins, components, and tutorials.

At the simulation initialization stage, a file containing all initial and boundary conditions is loaded and parsed. The main variables in this file are the following:

- Experiment ID
- Number of simulation instances (N_{it})
- Bin size (x, y, z) in mm (BIN_{dim})
- PTH type to analyse (PTH_{type})
- Number of pieces in bin or feeder (N_{parts})
- Presence of vibrating table.

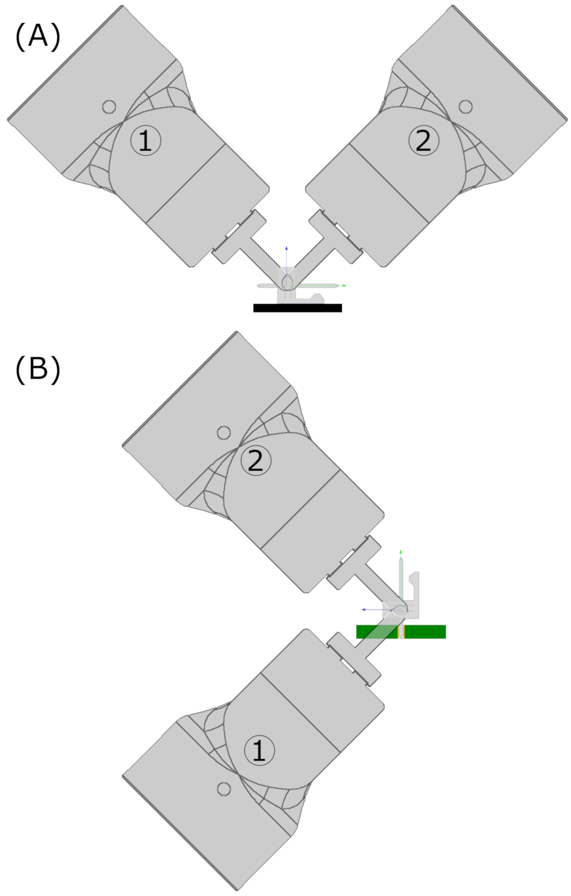

Figure 6: *Gripper position analysis during (A) part picking, (B) placing.*

- Type of experiment (Static analysis, bin emptying simulation, etc.)
- Optimal range of parts in bin (only applies in vibrating table scenario)
- Maximum number of vibrations (only applies in vibrating table scenario)

Right after parsing it, N_{it} demo instances are created with randomized initial part positions. This allows testing the same boundary conditions multiple times in just one simulation (see Figure 10). Each instance is composed of three main objects: a part (PTH) factory (responsible for creating the PTH_{type} instances), a gripper, and a bin (with its dimensions defined by BIN_{type}).

Mesh bodies were attached to every model, so the physics engine could register the collisions between them. Physical properties were only applied to the pieces, so they could be affected by gravity and move realistically. The bin is considered to be fixed in space, and therefore, no physical properties were needed for it. Finally, the simulation checks if there are any collisions between the gripper and the other elements in the scene.

Afterwards, the type of experiment associated with the current input is executed. For example, Figure 11 shows an experiment

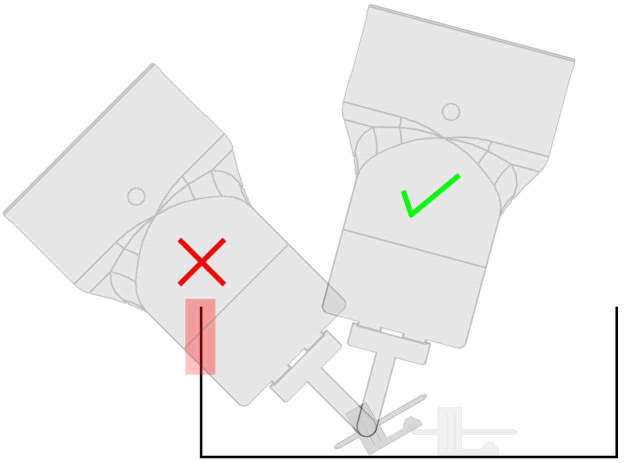

Figure 7: *Gripper colliding (left) and not colliding (right) the bin.*

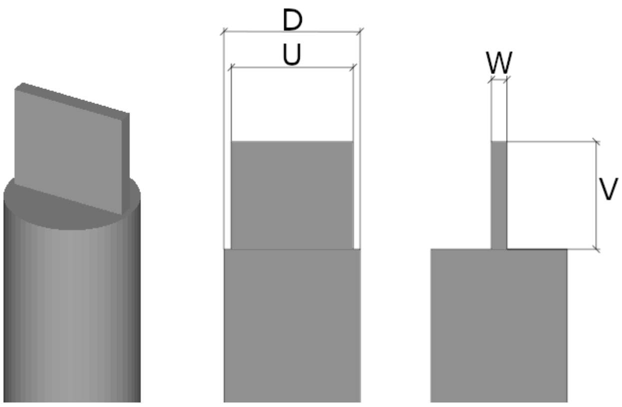

Figure 8: *Simplified gripper model: (A) Isometric view, (B) front view, (C) side view.*

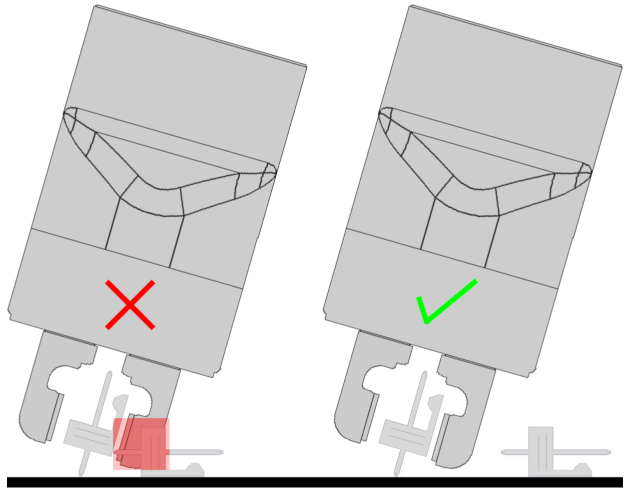

Figure 9: *Gripper colliding (left) and not colliding (right) with the neighbour parts.*

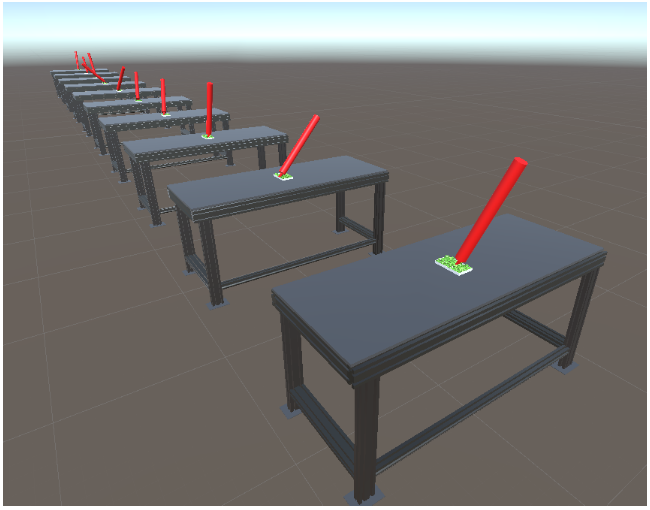

Figure 10: *Multiple demo instances in a single simulation.*

where the validity of each part in a bin is checked, and if it satisfies all previously explained conditions, the part is removed from the bin. These experiments are explained in detail in sections 4.3.1 and 4.3.2.

After all the iterations are over, an output file containing the simulation results is exported. The generated values are the following:

- Iteration number ($n \in [1, N_{it}]$)
- Initial positions of all the parts
- Candidacy of each part in the simulation
- Placeability of each part in the simulation
- Result of bin collision check of each part in the simulation
- Results of neighbour collision check of each part in the simulation
- Total number of valid parts
- Picking position of each part that has been removed from the bin (only applicable in bin picking experiments)
- Number of searches done before a valid part was found and removed from the bin (only applicable in bin picking experiments)
- Number of vibrations done before each part was removed from the bin (only applicable in vibrating table experiments)

Data is then analysed, and the performance metrics are computed.

When the simulation was being implemented, several problems and limitations appeared.

First, when the part factory created the instances, if the time between the generation of two objects was too small, the physics engine would detect a collision between them and would create a force sending them away from the bin. However, if the time between creating one part and the next was greater, this phase would have been taken too long. The solution consisted in defining several points in which the parts were created and then dropped in the container. Second, due to the difference in size between the PTHs

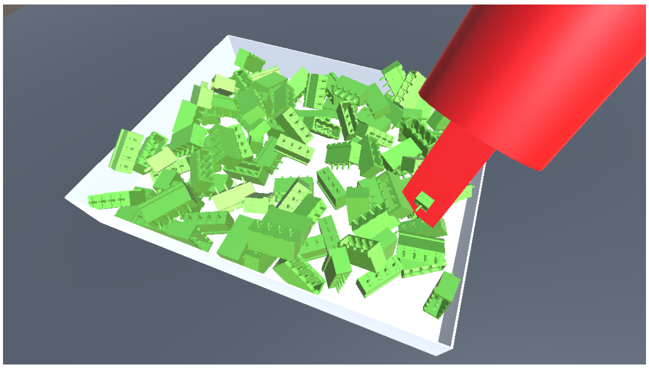

Figure 11: *Gripper in a part grasping position.*

and the other objects (gripper, container), many times the physics engine would not register the collisions and the parts would just pass through other surfaces. The solution consisted in using a continuous collision detection technique instead of the default discrete one and decreasing the physics solver time step.

Finally, the total number of parts in the scene depends on the number of simulation instances and the quantity of parts (PTHs) on each of them. If that value was too high, all the resources of the PC were used, slowing down the simulation and making them take too long time. Therefore, to get results faster, the number of iterations in each simulation had to be reduced.

4.3. Experiments

We define four types of experiments to evaluate the validity of picking parts from static bins and vibrating tables, two for each case. This section explains each of them.

4.3.1. Static bin picking

In the first group of experiments, we analyse the viability of bin picking with two different tests, a static analysis and a bin emptying simulation.

The goal of the static analysis is to find the optimal bin dimension for each part type. The solution is subjected to three limitations:

- The shape of the bin must be rectangular, meaning that circular or hexagonal bins for example are not considered.
- The maximum length on any dimension (x, y, z) is limited by the boundary conditions. This constraint comes from the assumption that the available space near the robot is limited.
- The volume of the bin needs to be sufficient to hold the number of parts stated in the boundary conditions.

Considering these limitations, a list of possible solutions (i.e. set of bins with different dimensions) is generated for each PTH type. Then, for each PTH - bin combination, we execute several simulations. In each one, the bin is filled up with a fixed number of randomly distributed parts and then their candidacy, placeability and bin collisions are checked. At this stage, the collision between the gripper and parts other than the candidate is not checked. This means that even if the target part is located in the middle of the container, surrounded by other parts, it is considered a valid candidate, being placeable and bin collision free. Once the simulations for each part type are finished, we consider the bin with the best success rate among all analysed options, and use it in the second experiment, bin emptying simulation.

In this second experiment, the goal is to check how probable is to empty an optimal bin full of parts. Therefore, once parts are randomly placed in the bin, besides testing the candidacy, placeability and bin collision of each part, also the collision of the gripper with the neighbour parts is evaluated. If a part satisfies all four conditions, it is considered valid, and removed from the bin. When this happens, as PTHs in the simulation are influenced by gravity, they will move within the bin, creating a new scenario. Therefore, once a part is removed, we have to test again the remaining parts. These simulations finish when there are no valid parts or the bin is emptied.

4.3.2. Vibrating table

In the second group of experiments, the viability of vibrating table picking is analysed. It is also split into static and bin emptying analysis.

There are two important differences between picking from a bin or a vibrating table: the latest has the ability to change the position of the parts, and it can also drop more parts on the table through a feeder. These two properties allow the system to always have the optimal number of parts on the table, to maximize the chances of having valid parts.

This experiment aims to find the optimal region, defined by a minimum and maximum number of parts on the table. This is done by testing different numbers of parts on the table and choosing the area that yield the best results. In this case, the testing range for each part type goes from 1 part up to 100. Figure 12 shows how 1 and 100 parts look like in a vibrating table simulation.

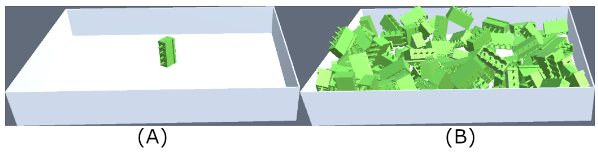

Figure 12: *Vibrating table static analysis: (A) 1 part, (B) 100 parts.*

Once the optimal region (min^*, max^*) is found, the next step is to simulate a feeder emptying procedure. In this case, the vibrating table is filled with max^* parts and an initial vibration is applied to scatter the parts, then the analysis is launched. We test each part on the table, and if it is found valid, it is removed from the simulation. This process will continue until one of the following situations occur:

- If there are no other valid parts in the table and the number of parts in the table is within the optimal region ($min^* \geq N_{parts} \geq max^*$) a vibration is triggered, which will modify the pose of the parts.

- If there are no other valid parts in the table and the number of parts in the bin is less than the minimum optimal ($N_{parts} < min^*$), the feeder will throw parts on the vibrating surface until the number of parts is the same as the maximum optimal ($N_{parts} = max^*$). Before starting checking the parts, a vibration is triggered.
- If the total number of picked parts is equal to the initial parts on the feeder (meaning that the feeder is empty), the simulation is stopped.

4.4. Test-case

As mentioned, the proposed pipeline is used in a test-case example. The boundary conditions and customer requirements of this industrial application example are the following:

- Each PCB is composed by 1xPTH0, 2xPTH1, and 2xPTH2 (see Figure 13). Their weight and geometry allow the use of both static bins and vibrating tables.
- 60 PCBs required per hour.
- A human operator fills a bin every 4 hours. Therefore, the bin capacity needs to be greater than 240 for PTH0 and 480 for PTH1 and PTH2.
- Due to limited available workspace, the largest length in any dimension (X, Y, Z) is 0.3 m (only applies to bin picking)
- Fixed vibrating table size is 195x150x27mm (only applies to vibrating table picking) (see Figure 2)

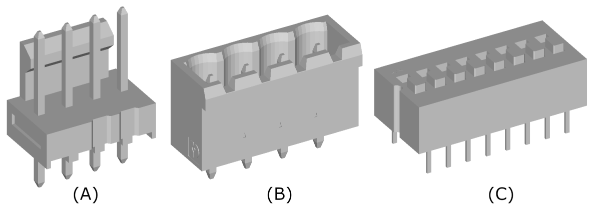

Figure 13: *Analysed parts: (A) PTH0, (B) PTH1, (C) PTH2.*

Table 1: *List of analysed bins and their dimensions*

Part type	Bin dimensions (X, Y, Z) [mm]
PTH0	100, 100, 27
PTH0	100, 125, 22
PTH0	100, 150, 18
PTH0	100, 175, 15
PTH0	75, 75, 48
PTH0	75, 100, 36
PTH0	75, 125, 29
PTH0	75, 150, 24
PTH0	75, 175, 21
PTH1	200, 75, 158
PTH1	300, 100, 79
PTH1	300, 125, 63
PTH1	300, 150, 53
PTH1	300, 175, 45
PTH1	300, 200, 40
PTH1	300, 225, 35
PTH1	300, 250, 32
PTH1	300, 275, 29
PTH1	300, 300, 26
PTH2	300, 75, 88
PTH2	300, 100, 66
PTH2	300, 125, 53
PTH2	300, 150, 44
PTH2	300, 175, 38
PTH2	300, 200, 33
PTH2	300, 225, 29
PTH2	300, 250, 26
PTH2	300, 275, 24
PTH2	300, 300, 22

5. Results

As previously mentioned, experiments are divided into two groups: static bin and vibrating table picking. This section analyses the data generated in those simulations with the test-case boundary conditions stated in Section 4.4.

5.1. Bin picking

As mentioned in section 4.3.1 the first experiment done in the bin picking scenario is used to search the optimal bin size for each PTH type. The dimensions of the candidate bins are shown in Table 5.1. Each bin has an unique XY surface for each part type. As we want to keep the volumen of the bin constant for each PTH type, the Z dimension will be automatically defined. Figure 14 shows the candidacy, placing and bin collision free probability for parts PTH0, PTH1 and PTH2 depending on the XY surface of the bin after 40 iterations. It shows that candidacy and placeability varies less than 7%, 5% and %3 for PTH0, PTH1 and PTH2 respectively with the analysed bin. However, bin collisions are very dependent on the XY surface of the bin, the greater this value is, the probability of not having a gripper-bin collision is higher.

In this case, the optimal bins for PTH0, PTH1 and PTH2 (bin_0^*, bin_1^* and bin_2^*) have the following dimensions:

- bin_0^* (x, y, z) = 100, 175, 15 mm
- bin_1^* (x, y, z) = 300, 300, 26 mm
- bin_2^* (x, y, z) = 300, 300, 22 mm

Once the optimal bin for each part type is found, the bin picking simulation is performed. It can be seen that the bin was not emptied in none of the iterations, and in the single best case, 11 parts were taken out of 480 (see Table 2).

The chosen optimal bins had an average bin collision free probability of 25.02%, 34.77% and 38.8%. However, those values decreased to 0.36%, 0.47% and 1.27% when the collision with the neighbours was checked. Therefore, these results show that with the current conditions, picking from static bins is not possible. The main reason being that the gripper is too bulky and hits other parts in the container.

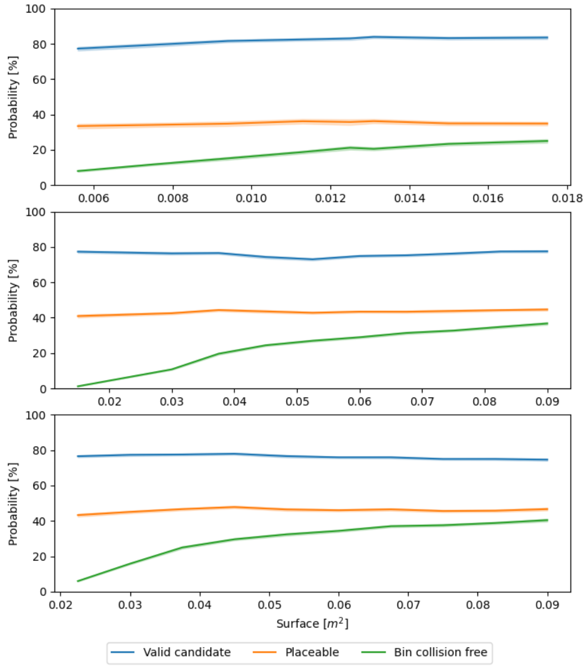

Figure 14: *Candidacy, placeability and bin-collision analysis for parts (A) PTH0, (B) PTH1 and (C) PTH2.*

Table 2: *Bin picking simulation results for each part type.*

PTH	Iterations	Parts	Emptied bins	Average valid parts	Min valid parts	Max valid parts	Standard Deviation
0	40	240	0	0.8	0	4	0.92
1	40	480	0	2.7	0	6	1.67
2	40	480	0	7.5	2	11	2.28

5.2. Vibrating table

As mentioned in section 4.3.2, before performing the vibrating table picking, an analysis was done to find the range of parts that need to be on the table in order to maximize the success probabilities of part picking tests. In order to do it, some experiments were performed. In each of them, a exact number of parts (i.e. $n \in \{1,2,...,100\}$) were placed in the vibrating table and their pick-and-place validity was analysed. To introduce the randomness, each experiment was repeated 40 times by randomly dropping the parts onto the vibrating table. The average number of valid parts over those 40 experiments is computed to represent the performance in each experiment. Figure 15 shows the average number of valid parts depending on how many parts are located in the vibrating table. As it can be seen, when there are few parts on the table, the number of available valid parts is very low. This is caused by the part pose that is not a valid candidate, or the part is not placeable, or because the gripper collides with the bin when trying to grasp it. When the number of parts is increased, we reach the maximum number of valid candidates, until it decreases again due to collisions with the other parts, since the environment starts to be more cluttered.

In order to find the range of maximum performance, a non-linear regression is calculated to obtain the performance curve for each PTH type. In practice, a fourth order model is employed in the regression to fit with the discrete data and further eliminate the noise introduced by the randomness. Then the best performance of each PTH type is found at the maximum value of each curve.

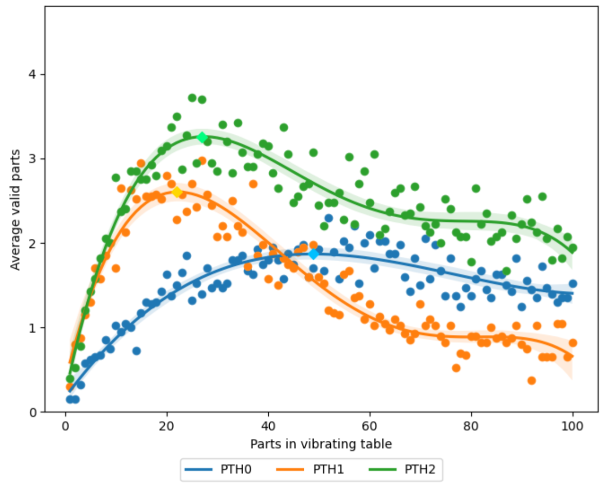

Figure 15: *Number of valid parts in vibrating table depending on total number of parts with their fourth order polynomial fittings.*

Table 3: *Vibrating table picking simulation results for each part type.*

PTH	Iterations	Parts	Optimal range	Emptied bins	Av. vibrations to empty feeder
0	50	240	44-54	50	123.14
1	50	480	17-27	50	166.96
2	50	480	22-32	50	139.07

For these part geometries and the analysed vibrating table dimensions, the maximum performance points are the following: 49 PTH0 parts on the table, 22 PTH1 and 27 PTH2.

Taking this information as the starting point, new experiments for the vibrating table picking are prepared. The input needed for them is not a single number of parts, but a range, since otherwise the feeder would be constantly placing new PTHs on the vibrating table. Therefore, a $\pm$ 5 part margin is considered, making the optimal ranges the following: $(min_0^*, max_0^*) = (44, 54)$, $(min_1^*, max_1^*) = (17, 27)$, $(min_2^*, max_2^*) = (22, 32)$.

The results of the vibrating picking process are shown in Table 3. As it can be seen, 100% of a total of 150 bins (50 for each PTH type) were emptied. In order to empty the feeder with the PTH1 components, the table had to vibrate on average 167 times. As the whole feeder emptying process needs to be done in 240 minutes (i.e. due to the process boundary conditions) a vibration was done every 1.43 minutes. Therefore, a vibrating table system is considered valid for picking these parts.

6. Conclusions and further work

In a bin picking scenario for small and light parts, picking can be done from static containers or feeder-vibrating table systems. The second one requires more expensive hardware, but has the ability to scatter and vibrate the parts on the surface, increasing the success probability. However, the feasibility test of either of them is very time-consuming and requires the use of specific hardware. In this paper, a simulation based initial analysis pipeline for picking small parts was presented. Obtaining a valid result in this test is a necessary but not sufficient condition for a bin type and small part geometry combination to be picking feasible.

As the simulations are fast and cheap to run, many robotic cell designs can be discarded quickly, focusing on those with higher success probabilities. Once a valid result is obtained, the subsequent tests involving other hardware (e.g. real robot or cameras) can be performed.

In the paper, the parameters of the simulator are explained with special focus on the configuration of the small parts. Then, a test-case defined by its boundary conditions is described and used as an example for the proposed strategy. The results reveal that the analysed parts are not valid for bin picking, but they are if a feeder-vibrating table system is used.

Further work could be extended in many directions. First, other kinds of grippers, such as vacuum or multi-finger, could be implemented. Focusing more on the simulation field, parallel programming on GPUs could be used to reduce the computation time. Finally, in the current paper, especially in the bin picking scenario, a large subset of parts is not valid because they did not satisfy the placeability condition. In future versions, the user could provide more information about how to grasp parts in that configuration, to place them in an intermediate surface and regrasp them.

Acknowledgement

This research is developed for the ACROBA project, funded by the European Union's HORIZON 2020 Research and Innovation program under grant agreement No 101017284.

References

[ANL*20] Adlin N., Nylund H., Lanz M., Lehtonen T., Juuti T.: Lean indicators for small batch size manufacturers in high cost countries. *Procedia Manufacturing 51* (2020), 1371–1378. 1

[BG18] Babin V., Gosselin C.: Picking, grasping, or scooping small objects lying on flat surfaces: A design approach. *The International Journal of Robotics Research 37*, 12 (2018), 1484–1499. 2

[BSG*15] Banas W., Sekala A., Gwiazda A., Foit K., Hryniewicz P., Kost G.: The modular design of robotic workcells in a flexible production line. In *IOP Conference Series: Materials Science and Engineering* (2015), vol. 95, IOP Publishing, p. 012099. 2

[Buc16] Buchholz D.: Bin-picking. *Studies in Systems, Decision and Control 44* (2016), 3–12. 1

[CB16] Coumans E., Bai Y.: Pybullet, a python module for physics simulation for games, robotics and machine learning. 2

[DTA20] D'Avella S., Tripicchio P., Avizzano C. A.: A study on picking objects in cluttered environments: Exploiting depth features for a custom low-cost universal jamming gripper. *Robotics and Computer-Integrated Manufacturing 63* (2020), 101888. 2

[FDN*20] Fujita M., Domae Y., Noda A., Garcia Ricardez G., Nagatani T., Zeng A., Song S., Rodriguez A., Causo A., Chen I.-M., et al.: What are the important technologies for bin picking? technology analysis of robots in competitions based on a set of performance metrics. *Advanced Robotics 34*, 7-8 (2020), 560–574. 2

[GBP17] Gadaleta M., Berselli G., Pellicciari M.: Energy-optimal layout design of robotic work cells: Potential assessment on an industrial case study. *Robotics and Computer-Integrated Manufacturing 47* (2017), 102–111. 2

[GCSR21] Gallo T., Cagnetti C., Silvestri C., Ruggieri A.: Industry 4.0 tools in lean production: A systematic literature review. *Procedia Computer Science 180* (2021), 394–403. 1

[IDX*20] Ichnowski J., Danielczuk M., Xu J., Satish V., Goldberg K.: Gomp: Grasp-optimized motion planning for bin picking. In *2020 IEEE International Conference on Robotics and Automation (ICRA)* (2020), IEEE, pp. 5270–5277. 2, 3

[JASG21] Jamwal A., Agrawal R., Sharma M., Giallanza A.: Industry 4.0 technologies for manufacturing sustainability: a systematic review and future research directions. *Applied Sciences 11*, 12 (2021), 5725. 1

[KKT*18] Kritzinger W., Karner M., Traar G., Henjes J., Sihn W.: Digital twin in manufacturing: A categorical literature review and classification. *IFAC-PapersOnLine 51*, 11 (2018), 1016–1022. 2

[LG19] Laemmle A., Gust S.: Automatic layout generation of robotic production cells in a 3d manufacturing simulation environment. *Procedia CIRP 84* (2019), 316–321. 2

[MKS*21] Moosmann M., Kulig M., Spenrath F., Mönnig M., Roggendorf S., Petrovic O., Bormann R., Huber M. F.: Separating entangled workpieces in random bin picking using deep reinforcement learning. *Procedia CIRP 104* (2021), 881–886. 2

[MTS*20] Mnyusiwalla H., Triantafyllou P., Sotiropoulos P., Roa M. A., Friedl W., Sundaram A. M., Russell D., Deacon G.: A bin-picking benchmark for systematic evaluation of robotic pick-and-place systems. *IEEE Robotics and Automation Letters 5*, 2 (2020), 1389–1396. 2

[MVPR*20] Mayoral-Vilches V., Pinzger M., Rass S., Dieber B., Gil-Uriarte E.: Can ros be used securely in industry? red teaming ros-industrial. *arXiv preprint arXiv:2009.08211* (2020). 2

[MW17] Mrugalska B., Wyrwicka M. K.: Towards lean production in industry 4.0. *Procedia engineering 182* (2017), 466–473. 1

[OG20] Oztemel E., Gursev S.: Literature review of industry 4.0 and related technologies. *Journal of Intelligent Manufacturing 31*, 1 (2020), 127–182. 1

[QCG*09] Quigley M., Conley K., Gerkey B., Faust J., Foote T., Leibs J., Wheeler R., Ng A. Y., et al.: Ros: an open-source robot operating system. In *ICRA workshop on open source software* (2009), vol. 3, Kobe, Japan, p. 5. 2

[SHT*17] Sita E., Horváth C. M., Thomessen T., Korondi P., Pipe A. G.: Ros-unity3d based system for monitoring of an industrial robotic process. In *2017 IEEE/SICE International Symposium on System Integration (SII)* (2017), IEEE, pp. 1047–1052. 2

[SK14] Schyja A., Kuhlenkötter B.: Virtual bin picking-a generic framework to overcome the bin picking complexity by the use of a virtual environment. In *2014 4th International Conference On Simulation And Modeling Methodologies, Technologies And Applications (SIMULTECH)* (2014), IEEE, pp. 133–140. 2

[SKWK20] Schuh G., Kelzenberg C., Wiese J., Kessler N.: Creation of digital production twins for the optimization of value creation in single and small batch production. *Procedia CIRP 93* (2020), 222–227. 1

[TET12] Todorov E., Erez T., Tassa Y.: Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems* (2012), IEEE, pp. 5026–5033. 2

[VVS17] Vonásek V., Vick A., Saska M.: Motion planning with motion primitives for industrial bin picking. In *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)* (2017), IEEE, pp. 1–4. 2

[ZF17] ZHANG J., FANG X.: Challenges and key technologies in robotic cell layout design and optimization. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science 231*, 15 (2017), 2912–2924. 2

[ZWW*20] ZHAO J., WANG X., WANG S., JIANG X., LIU Y.: Assembly of randomly placed parts realized by using only one robot arm with a general parallel-jaw gripper. In *2020 IEEE International Conference on Robotics and Automation (ICRA)* (2020), IEEE, pp. 5024–5030. 2

[ZZL*20] ZHOU G., ZHANG C., LI Z., DING K., WANG C.: Knowledge-driven digital twin manufacturing cell towards intelligent manufacturing. *International Journal of Production Research 58*, 4 (2020), 1034–1051. 2