

AdaptPack Studio: an automated intelligent framework for offline factory programming

André Luiz Castro

INESC TEC and Department of Computer Systems, Universidade Federal da Paraíba, Joao Pessoa, Brazil

João Pedro Carvalho de Souza

INESC TEC and Porto University, Porto, Portugal

Luís F. Rocha

INESC TEC, Porto, Portugal, and

Manuel F. Silva

INESC TEC and Departamento de Engenharia Electrotécnica, ISEP – Instituto Superior de Engenharia do Porto, Porto, Portugal

Abstract

Purpose – This paper aims to propose an automated framework for agile development and simulation of robotic palletizing cells. An automatic offline programming tool, for a variety of robot brands, is also introduced.

Design/methodology/approach – This framework, named AdaptPack Studio, offers a custom-built library to assemble virtual models of palletizing cells, quick connect these models by drag and drop, and perform offline programming of robots and factory equipment in short steps.

Findings – Simulation and real tests performed showed an improvement in the design, development and operation of robotic palletizing systems. The AdaptPack Studio software was tested and evaluated in a pure simulation case and in a real-world scenario. Results have shown to be concise and accurate, with minor model displacement inaccuracies because of differences between the virtual and real models.

Research limitations/implications – An intuitive drag and drop layout modeling accelerates the design and setup of robotic palletizing cells and automatic offline generation of robot programs. Furthermore, A* based algorithms generate collision-free trajectories, discretized both in the robot joints space and in the Cartesian space. As a consequence, industrial solutions are available for production in record time, increasing the competitiveness of companies using this tool.

Originality/value – The AdaptPack Studio framework includes, on a single package, the possibility to program, simulate and generate the robot code for four different brands of robots. Furthermore, the application is tailored for palletizing applications and specifically includes the components (Building Blocks) of a particular company, which allows a very fast development of new solutions. Furthermore, with the inclusion of the Trajectory Planner, it is possible to automatically develop robot trajectories without collisions.

Keywords Offline programming, Path planning, Robot simulation, Robotic cells

Paper type Research paper

1. Introduction

Factory automation tasks have changed in the past few years due to the increased demand for heterogeneous and mixed assembly lines. These changes imply that automation processes with short life cycles are more common nowadays. All these advancements are happening because of the Fourth Industrial Revolution, which has brought improvements in artificial intelligence, robotics, materials, among other fields. Distinct processes throughout the plants are becoming connected in a way that a virtual intelligence can operate. Virtualization is one of the techniques that has supported remote monitoring, tracing processes in real-time, and allowing fast response

and decision making. Another significant aspect is that products are being manufactured according to demand, using adaptive and modular manufacturing cells, which offers flexibility to modify machine tasks promptly (Do *et al.*, 2015).

These advancements have affected a special kind of automated process, namely palletizing cells, since product life cycle is shortening and dependent on the inconstant market. Therefore, competition is fierce, and profits increase in companies that conduct adaptable responses while producing a

The current issue and full text archive of this journal is available on Emerald Insight at: <https://www.emerald.com/insight/0143-991X.htm>



Industrial Robot: the international journal of robotics research and application
© Emerald Publishing Limited [ISSN 0143-991X]
[DOI 10.1108/IR-12-2019-0252]

This is an extended and improved version of a paper that was originally presented at the ICARSC 2019 conference, that took place last April in Porto, and which received, ex-aequo, the best paper award granted by the Industrial Robot journal to the papers presented at this event. The original version of the paper was entitled “AdaptPack Studio: Automatic Offline Robot Programming Framework for Factory Environments” and is available for download from the IEEE Xplore, at the following URL: <https://ieeexplore.ieee.org/document/8733626/>

Received 12 December 2019

Revised 7 March 2020

16 June 2020

Accepted 17 June 2020

vast collection of items, as indicated by the client's determinations. Production flexibility and volume are directly related to the time spent in setting up a robotic cell. In consequence, it is crucial to have tools that allow the rapid programming of those robots. To accomplish the previous requirements, using offline programming is the usual approach. This type of programming can simulate the mechanized environment, and the original idea is to use computer graphics to create a reproduction of the working cell. Inside this abstraction, it is possible to handle all tasks and variables, corresponding to the real situation.

Robot-based palletizing cells have captured attention in the last decade in the scientific sphere. Yu *et al.* (2011) developed an offline tool for stacking objects using a MOTOMAN-HP20 robot and the brand's simulator (Yaskawa, 2020). On its turn, Kito *et al.* (2017) produced an offline teaching tool for robots to pick bolts from a pallet. However, there is no automation in the previous solutions: the user still programs the robot manually, and high efficiency is still not achieved. Following a different approach, the work presented in Cheng and Chen (2013) presents an autonomous smart robot with advanced sensing and decision-making capabilities to teach "child" robots in the production line. This last work, however, does not deal with the palletizing problem. For the specific problem addressed in this paper, Moura and Silva (2018) achieved automatic palletizing solutions, but in the case of their work, the programs generated are limited to the programming of Asea Brown Boveri (ABB) robots, the path planning does not take into consideration possible collisions and the pallet arrangement is not as flexible as the solution presented here.

Although the previous solutions focus on a particular brand of a robot and its native simulation software, there are several manufacturers of generic offline programming software (Visual Components, 2020; RoboDK, 2020; OCTOPUZ, 2020), which can perform programming of global robotic platforms. There are also alternative simulation software packages in the market, such as the FlexSim (2020), Anylogic (2020) and SIMIO (2020), but they do not perform detailed simulation of robot motions and offline robot programming. The work proposed by Silva *et al.* (2017) presents a solution for the automatic offline generation of collision-free robot programs. In their work, a framework that aids the programming and path generation of palletizing routines has been developed. This platform has its base on the Visual Components (VC) simulation software (Visual Components, 2020), owing to its capability to perform detailed simulations and offline programming for several distinct robot brands, over other candidates.

The present work focuses on the development of a software framework responsible for aiding the design, development, simulation and effective installation of robotic palletizing cells, in the scope of a project named AdaptPack. The AdaptPack project's main objective is to develop a framework, based on concepts of modularity, to expedite the programming of robots for palletizing tasks. To address this problem, the following project objectives were specified:

- provide a solution to the design and simulation of robotic cells, based on a library of existing components, allowing the assembly of palletizing systems in a short time;

- strengthen a solution to generate automatic offline routines for palletizing products with different configurations, formats, and dimensions;
- develop a robot path planning solution with collision avoidance; and
- test and validate the framework in real-life processes.

The core result of this project was the AdaptPack Studio, a modular system designed as an add-on to the VC simulation software. This add-on attacks the presented challenges in three modules:

- 1 palletizing cell design, modeling and simulation;
- 2 automatic offline palletizing routine generation; and
- 3 translation and exporting of generated programs to real systems.

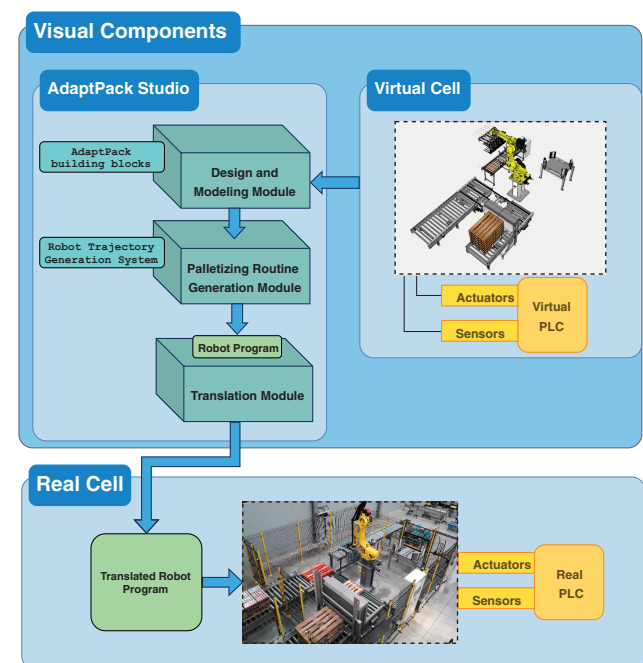
For this accomplishment, it was constructed three specific systems inside the central platform to handle each module individually. Those modules are connected and exchange data among themselves, as illustrated in Figure 1.

Carrying the past impressions in mind, the remainder organization of this paper is as follows. Section 2 describes the general concepts behind this project and introduces the central architecture of the proposed system. Section 3 presents the performed tests and results. Finally, in Section 4, the main conclusions of this work are exposed, followed by a few ideas concerning possible future improvements.

2. AdaptPack Studio architecture

The AdaptPack Studio further divides into three modules, which are the "Design and Modeling", the "Palletizing Routine Generation" and the "Translation". Each one receives inputs from the previous, as illustrated in Figure 1.

Figure 1 Overall system architecture



2.1 AdaptPack modules

2.1.1 Design and modeling

The first module tackles solutions for structural and architectural design. The VC software facilitates this process by offering a Modeling tab in which components get connected as blocks. Besides the broad set of default components, provided by the VC library, in this work were developed models of general components used in palletizing operations (specific for custom requirements). Named “AdaptPack Building Blocks”, these components, together with the Modeling Tab, form the “Design and Modeling” module.

2.1.2 Palletizing routine generation

By interacting with a custom-developed Graphical User Interface (GUI), targeted for palletizing operations, the operator can set several custom parameters (explained in Subsection 2.2) for the robot offline programming algorithm. This GUI (called “ADAPTPACK” tab) was developed using the C# Application Programming Interface (API) provided by VC, granting the possibility to program robot instructions without writing any code or using the teach pendant (Figure 3). Simple steps, like filling in text fields, clicking buttons, and selecting components on the screen, are responsible for storing the virtual cell configurations. This information is sent to the *Palletize.cs* Class, responsible for computing the main instructions, as observed in Figure 2. The code for the robot is generated automatically after setting all configurations and pressing the *Init Routine* button (depicted far-right in Figure 3).

Figure 2 Palletize Routine Generation module in details

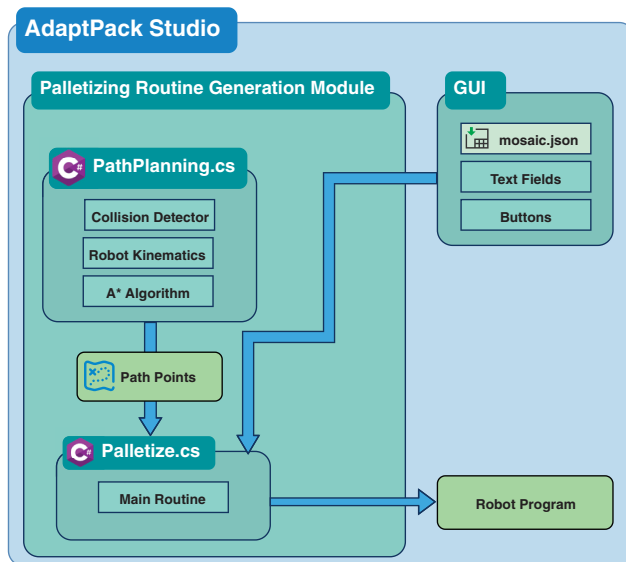
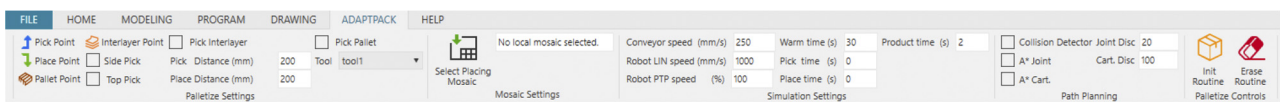


Figure 3 GUI subdivided among Palletize, Mosaic, Simulation, and Path planning



Note: It holds all options required for automatic robot programming

This process forms the “Palletizing Routine Generation” module, which, on its turn, will receive information from the “Robot Trajectory Generation System”, as illustrated in Figure 1. The trajectory generation system aims to program the robot movements between the pick and place points, considering operations within the modeled cell, in a way that results in collision-free trajectories. This last system is implemented inside the *PathPlanning.cs* class (Figure 2). Afterward, three dimensional (3D) simulation allows the system validation. By simulating the robot’s virtual models and the palletizing cells in which they operate, there may be a significant improvement in the design efficiency of a robotic cell. Also, considering the accuracy and level of detail offered by VC, it is possible to achieve models behaving very close to live equipment; this allows studying and testing a solution, with accuracy, before its implementation.

2.1.3 Translation

After simulation validation, the robot’s program “code”, written in the VC’s software proprietary language (Robot Simulation Language – RSL), is translated into the ones used by the different robots’ controllers (Asea Brown Boveri (ABB), Keller und Knappich Augsburg (KUKA), Fuji Automatic NUMerical Control (FANUC), and Yaskawa). For the translation process the “Translation” module conducted the main process. The approach used a different GUI, responsible for receiving the auto programmed code and translating to the proprietary languages, being this process described in detail in Souza et al. (2019) (a video illustrating this process can be found at <http://bit.ly/2PJfzgh>).

2.2 AdaptPack tab

To work along with the VC interface, which organizes itself in tabs, the “AdaptPack” tab was inserted between the existing ones (see Figure 3). Each tab is responsible for a specific task, such as layout construction, components modeling/editing, programming and others. This new tab, further divided into subgroups, allows configuring robotic instructions intuitively.

The “Palletize Settings” subgroup contains buttons and text fields that define the pick and place settings. The determination of the Pick, Pallet, and Interlayer locations occurs by clicking on the correspondent 3D component, where the products, pallets, and interlayers are stored, respectively. The place location sets up by clicking on the element defined as the drop point. Since a gripper may have multiple Tool Center Points (TCP), it is also available a drop-down menu (“Tool”) with all the robot tools available in the library. Furthermore, there is an option that determines the methodology for the definition of the pick and place approach points, in agreement with the kind of gripper used. The ones that have a side approximation point (involving toothed grippers), or the ones that have a top

approximation point (for example, suction grippers) are covered in this scope. As acknowledged as a good practice to use approach points just before the grasping/dropping location – the framework in study generates these points when entering the distances in two provided text boxes (“Pick Distance” and “Place Distance”).

The “Mosaic Settings” subgroup allows the operator to select a file, written in the JavaScript Object Notation (JSON) scripting language, containing a palletizing stack pattern (mosaic). This file consists of structures that determine the placing order and positions of the boxes to be stored in the pallet. It also contains information about the boxes and pallet dimensions, complemented with the number of product layers on this pallet. The overall mosaics’ structure divides into layers, each layer containing one or more groups of products, and each group including one or more boxes. Consequently, the robot can simultaneously handle one or multiple products, depending on the mosaic layout defined by each customer. The JSON file is generated automatically by a software developed in collaboration with the AdaptPack Project team. Given input settings such as pallet and product dimensions, the number of layers, the use of interlayers and others (as depicted on the left image of Figure 4), this software generates the palletizing coordinates, as presented in the right image of Figure 4.

Finally, the last two subgroups, “Simulation Setting” and “Path Planning”, are responsible for further improving the simulation. The first presents text fields to perform adjustments to the simulation environment, such as robot’s and conveyor’s speeds, stop times, product production rate time, among others. The last presents buttons which enable the activation of the “Robot Trajectory Generation System”, based on the A* search algorithm, developed by Tavares et al. (2016). Besides the joint space trajectory generated by Silva et al. (2017), the current work introduces a Cartesian space trajectory generation. This new approach allowed decreasing the computation time and the smoothing of the robot trajectory.

After completing all the configurations, the system is ready to compute and generate the palletizing routine automatically. When the *Init Routine* button is activated, the *Palletize.cs* class receives the configurations from the GUI, as pictured in Figure 2. The pseudocode presented in Code Script 1 shows, step by step, the computation steps that occur inside the class.

Figure 4 Representation of the palletizing mosaic in the JSON file

```

{
  "pallet": [
    {
      "Code": "G",
      "Id": 1,
      "length": 1200,
      "width": 1000
    }
  ],
  "items": [
    {
      "Id": 121,
      "length": 296,
      "width": 246,
      "height": 220,
      "MaxHeight": 2000,
      "Layers": 4
    }
  ]
}

```

Input

```

{
  "boxes": [
    {
      "X1": 0,
      "Y1": 0,
      "Z1": 0,
      "X2": 296,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 592,
      "Y1": 0,
      "Z1": 0,
      "X2": 888,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 888,
      "Y1": 0,
      "Z1": 0,
      "X2": 1184,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 1184,
      "Y1": 0,
      "Z1": 0,
      "X2": 1480,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 1480,
      "Y1": 0,
      "Z1": 0,
      "X2": 1776,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 1776,
      "Y1": 0,
      "Z1": 0,
      "X2": 2072,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 2072,
      "Y1": 0,
      "Z1": 0,
      "X2": 2368,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 2368,
      "Y1": 0,
      "Z1": 0,
      "X2": 2664,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 2664,
      "Y1": 0,
      "Z1": 0,
      "X2": 2960,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 2960,
      "Y1": 0,
      "Z1": 0,
      "X2": 3256,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 3256,
      "Y1": 0,
      "Z1": 0,
      "X2": 3552,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 3552,
      "Y1": 0,
      "Z1": 0,
      "X2": 3848,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 3848,
      "Y1": 0,
      "Z1": 0,
      "X2": 4144,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 4144,
      "Y1": 0,
      "Z1": 0,
      "X2": 4440,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 4440,
      "Y1": 0,
      "Z1": 0,
      "X2": 4736,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 4736,
      "Y1": 0,
      "Z1": 0,
      "X2": 5032,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 5032,
      "Y1": 0,
      "Z1": 0,
      "X2": 5328,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 5328,
      "Y1": 0,
      "Z1": 0,
      "X2": 5624,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 5624,
      "Y1": 0,
      "Z1": 0,
      "X2": 5920,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 5920,
      "Y1": 0,
      "Z1": 0,
      "X2": 6216,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 6216,
      "Y1": 0,
      "Z1": 0,
      "X2": 6512,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 6512,
      "Y1": 0,
      "Z1": 0,
      "X2": 6808,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 6808,
      "Y1": 0,
      "Z1": 0,
      "X2": 7104,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 7104,
      "Y1": 0,
      "Z1": 0,
      "X2": 7400,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 7400,
      "Y1": 0,
      "Z1": 0,
      "X2": 7696,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 7696,
      "Y1": 0,
      "Z1": 0,
      "X2": 7992,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 7992,
      "Y1": 0,
      "Z1": 0,
      "X2": 8288,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 8288,
      "Y1": 0,
      "Z1": 0,
      "X2": 8584,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 8584,
      "Y1": 0,
      "Z1": 0,
      "X2": 8880,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 8880,
      "Y1": 0,
      "Z1": 0,
      "X2": 9176,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 9176,
      "Y1": 0,
      "Z1": 0,
      "X2": 9472,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 9472,
      "Y1": 0,
      "Z1": 0,
      "X2": 9768,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 9768,
      "Y1": 0,
      "Z1": 0,
      "X2": 10064,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 10064,
      "Y1": 0,
      "Z1": 0,
      "X2": 10360,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 10360,
      "Y1": 0,
      "Z1": 0,
      "X2": 10656,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 10656,
      "Y1": 0,
      "Z1": 0,
      "X2": 10952,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 10952,
      "Y1": 0,
      "Z1": 0,
      "X2": 11248,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 11248,
      "Y1": 0,
      "Z1": 0,
      "X2": 11544,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 11544,
      "Y1": 0,
      "Z1": 0,
      "X2": 11840,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 11840,
      "Y1": 0,
      "Z1": 0,
      "X2": 12136,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 12136,
      "Y1": 0,
      "Z1": 0,
      "X2": 12432,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 12432,
      "Y1": 0,
      "Z1": 0,
      "X2": 12728,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 12728,
      "Y1": 0,
      "Z1": 0,
      "X2": 13024,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 13024,
      "Y1": 0,
      "Z1": 0,
      "X2": 13320,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 13320,
      "Y1": 0,
      "Z1": 0,
      "X2": 13616,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 13616,
      "Y1": 0,
      "Z1": 0,
      "X2": 13912,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 13912,
      "Y1": 0,
      "Z1": 0,
      "X2": 14208,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 14208,
      "Y1": 0,
      "Z1": 0,
      "X2": 14504,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 14504,
      "Y1": 0,
      "Z1": 0,
      "X2": 14800,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 14800,
      "Y1": 0,
      "Z1": 0,
      "X2": 15096,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 15096,
      "Y1": 0,
      "Z1": 0,
      "X2": 15392,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 15392,
      "Y1": 0,
      "Z1": 0,
      "X2": 15688,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 15688,
      "Y1": 0,
      "Z1": 0,
      "X2": 15984,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 15984,
      "Y1": 0,
      "Z1": 0,
      "X2": 16280,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 16280,
      "Y1": 0,
      "Z1": 0,
      "X2": 16576,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 16576,
      "Y1": 0,
      "Z1": 0,
      "X2": 16872,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 16872,
      "Y1": 0,
      "Z1": 0,
      "X2": 17168,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 17168,
      "Y1": 0,
      "Z1": 0,
      "X2": 17464,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 17464,
      "Y1": 0,
      "Z1": 0,
      "X2": 17760,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 17760,
      "Y1": 0,
      "Z1": 0,
      "X2": 18056,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 18056,
      "Y1": 0,
      "Z1": 0,
      "X2": 18352,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 18352,
      "Y1": 0,
      "Z1": 0,
      "X2": 18648,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 18648,
      "Y1": 0,
      "Z1": 0,
      "X2": 18944,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 18944,
      "Y1": 0,
      "Z1": 0,
      "X2": 19240,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 19240,
      "Y1": 0,
      "Z1": 0,
      "X2": 19536,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 19536,
      "Y1": 0,
      "Z1": 0,
      "X2": 19832,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 19832,
      "Y1": 0,
      "Z1": 0,
      "X2": 20128,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 20128,
      "Y1": 0,
      "Z1": 0,
      "X2": 20424,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 20424,
      "Y1": 0,
      "Z1": 0,
      "X2": 20720,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 20720,
      "Y1": 0,
      "Z1": 0,
      "X2": 21016,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 21016,
      "Y1": 0,
      "Z1": 0,
      "X2": 21312,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 21312,
      "Y1": 0,
      "Z1": 0,
      "X2": 21608,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 21608,
      "Y1": 0,
      "Z1": 0,
      "X2": 21904,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 21904,
      "Y1": 0,
      "Z1": 0,
      "X2": 22200,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 22200,
      "Y1": 0,
      "Z1": 0,
      "X2": 22496,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 22496,
      "Y1": 0,
      "Z1": 0,
      "X2": 22792,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 22792,
      "Y1": 0,
      "Z1": 0,
      "X2": 23088,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 23088,
      "Y1": 0,
      "Z1": 0,
      "X2": 23384,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 23384,
      "Y1": 0,
      "Z1": 0,
      "X2": 23680,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 23680,
      "Y1": 0,
      "Z1": 0,
      "X2": 23976,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 23976,
      "Y1": 0,
      "Z1": 0,
      "X2": 24272,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 24272,
      "Y1": 0,
      "Z1": 0,
      "X2": 24568,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 24568,
      "Y1": 0,
      "Z1": 0,
      "X2": 24864,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 24864,
      "Y1": 0,
      "Z1": 0,
      "X2": 25160,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 25160,
      "Y1": 0,
      "Z1": 0,
      "X2": 25456,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 25456,
      "Y1": 0,
      "Z1": 0,
      "X2": 25752,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 25752,
      "Y1": 0,
      "Z1": 0,
      "X2": 26048,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 26048,
      "Y1": 0,
      "Z1": 0,
      "X2": 26344,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 26344,
      "Y1": 0,
      "Z1": 0,
      "X2": 26640,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 26640,
      "Y1": 0,
      "Z1": 0,
      "X2": 26936,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 26936,
      "Y1": 0,
      "Z1": 0,
      "X2": 27232,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 27232,
      "Y1": 0,
      "Z1": 0,
      "X2": 27528,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 27528,
      "Y1": 0,
      "Z1": 0,
      "X2": 27824,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 27824,
      "Y1": 0,
      "Z1": 0,
      "X2": 28120,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 28120,
      "Y1": 0,
      "Z1": 0,
      "X2": 28416,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 28416,
      "Y1": 0,
      "Z1": 0,
      "X2": 28712,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 28712,
      "Y1": 0,
      "Z1": 0,
      "X2": 29008,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 29008,
      "Y1": 0,
      "Z1": 0,
      "X2": 29304,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 29304,
      "Y1": 0,
      "Z1": 0,
      "X2": 29600,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 29600,
      "Y1": 0,
      "Z1": 0,
      "X2": 29896,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 29896,
      "Y1": 0,
      "Z1": 0,
      "X2": 30192,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 30192,
      "Y1": 0,
      "Z1": 0,
      "X2": 30488,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 30488,
      "Y1": 0,
      "Z1": 0,
      "X2": 30784,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 30784,
      "Y1": 0,
      "Z1": 0,
      "X2": 31080,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 31080,
      "Y1": 0,
      "Z1": 0,
      "X2": 31376,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 31376,
      "Y1": 0,
      "Z1": 0,
      "X2": 31672,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 31672,
      "Y1": 0,
      "Z1": 0,
      "X2": 31968,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 31968,
      "Y1": 0,
      "Z1": 0,
      "X2": 32264,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 32264,
      "Y1": 0,
      "Z1": 0,
      "X2": 32560,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 32560,
      "Y1": 0,
      "Z1": 0,
      "X2": 32856,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 32856,
      "Y1": 0,
      "Z1": 0,
      "X2": 33152,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 33152,
      "Y1": 0,
      "Z1": 0,
      "X2": 33448,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 33448,
      "Y1": 0,
      "Z1": 0,
      "X2": 33744,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 33744,
      "Y1": 0,
      "Z1": 0,
      "X2": 34040,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 34040,
      "Y1": 0,
      "Z1": 0,
      "X2": 34336,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 34336,
      "Y1": 0,
      "Z1": 0,
      "X2": 34632,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 34632,
      "Y1": 0,
      "Z1": 0,
      "X2": 34928,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 34928,
      "Y1": 0,
      "Z1": 0,
      "X2": 35224,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 35224,
      "Y1": 0,
      "Z1": 0,
      "X2": 35520,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 35520,
      "Y1": 0,
      "Z1": 0,
      "X2": 35816,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 35816,
      "Y1": 0,
      "Z1": 0,
      "X2": 36112,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 36112,
      "Y1": 0,
      "Z1": 0,
      "X2": 36408,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 36408,
      "Y1": 0,
      "Z1": 0,
      "X2": 36704,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 36704,
      "Y1": 0,
      "Z1": 0,
      "X2": 37000,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 37000,
      "Y1": 0,
      "Z1": 0,
      "X2": 37296,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 37296,
      "Y1": 0,
      "Z1": 0,
      "X2": 37592,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 37592,
      "Y1": 0,
      "Z1": 0,
      "X2": 37888,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 37888,
      "Y1": 0,
      "Z1": 0,
      "X2": 38184,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 38184,
      "Y1": 0,
      "Z1": 0,
      "X2": 38480,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 38480,
      "Y1": 0,
      "Z1": 0,
      "X2": 38776,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 38776,
      "Y1": 0,
      "Z1": 0,
      "X2": 39072,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 39072,
      "Y1": 0,
      "Z1": 0,
      "X2": 39368,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 39368,
      "Y1": 0,
      "Z1": 0,
      "X2": 39664,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 39664,
      "Y1": 0,
      "Z1": 0,
      "X2": 39960,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 39960,
      "Y1": 0,
      "Z1": 0,
      "X2": 40256,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 40256,
      "Y1": 0,
      "Z1": 0,
      "X2": 40552,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 40552,
      "Y1": 0,
      "Z1": 0,
      "X2": 40848,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 40848,
      "Y1": 0,
      "Z1": 0,
      "X2": 41144,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 41144,
      "Y1": 0,
      "Z1": 0,
      "X2": 41440,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 41440,
      "Y1": 0,
      "Z1": 0,
      "X2": 41736,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 41736,
      "Y1": 0,
      "Z1": 0,
      "X2": 42032,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 42032,
      "Y1": 0,
      "Z1": 0,
      "X2": 42328,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 42328,
      "Y1": 0,
      "Z1": 0,
      "X2": 42624,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 42624,
      "Y1": 0,
      "Z1": 0,
      "X2": 42920,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 42920,
      "Y1": 0,
      "Z1": 0,
      "X2": 43216,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 43216,
      "Y1": 0,
      "Z1": 0,
      "X2": 43512,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 43512,
      "Y1": 0,
      "Z1": 0,
      "X2": 43808,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 43808,
      "Y1": 0,
      "Z1": 0,
      "X2": 44104,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 44104,
      "Y1": 0,
      "Z1": 0,
      "X2": 44400,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 44400,
      "Y1": 0,
      "Z1": 0,
      "X2": 44696,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 44696,
      "Y1": 0,
      "Z1": 0,
      "X2": 44992,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 44992,
      "Y1": 0,
      "Z1": 0,
      "X2": 45288,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 45288,
      "Y1": 0,
      "Z1": 0,
      "X2": 45584,
      "Y2": 246,
      "Z2": 220
    },
    {
      "X1": 45584,
      "Y1": 0,
      "Z1": 0,
      "
```


with a custom modular gripper, capable of handling single or multiple objects.

To improve the simulation graphics rendering performance, the cell safety fences were invisible during the experiments; still, these fences had collision properties enabled. Additional equipment, such as conveyors, feeders, and pallets were indispensable; they allowed the simulation to act realistically, displaying the full capability to reproduce real operations. In the test scenario, the robot was assigned to pick packs of cans/cartoons and palletize them on top of the pallet. A live model of the virtual cell was assembled with a prior objective to verify the AdaptPack framework and support the system's usefulness. The live model presented in Figure 6 had the same components positions.

3.1 Simulation tests

Simulation tests accompanied the usage of the new custom component library modeled for this work, shown in Figure 7. This library formed the main components inside the cell, serving as a validation of the AdaptPack Studio concept. One of the objectives intended when constructing the virtual cell was to determine the components' optimal positions, first to reduce the steps from the pick to place points, and second to allow the robot to reach the highest stacking height possible. Figure 8 presents the robot's workspace envelope, one of the references which served to optimize the position of other components

Figure 5 AdaptPack Studio simulated test cell

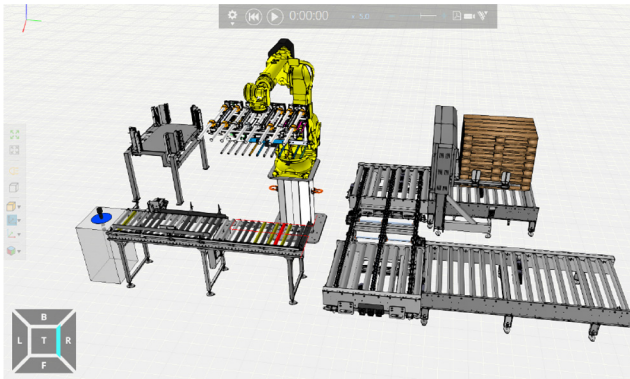


Figure 6 Real world test cell



around it. Besides shorter distance between components, to develop the optimal layout structure, also robot joint efforts, dynamic compensation, total reach and collision avoidance algorithms may improve the cell layout quality (Tavares *et al.*, 2019).

After defining the optimal layout structure, the auto programming step was ready to take action. As explained in Section 2, interaction with the AdaptPack tab took action to define pick and place points, gripper modules to use, the pallet mosaic structure (uploaded as a JSON file), and the rest of the general configurations. The auto generated robot code followed the pseudo algorithm described in Code Script 1.

Initially the tools and base frames are defined, followed by declarations of the pallet and interlayer routines. The pick and place functions were subdivided into subroutines, to keep track of the actions the robot is executing, so it always moves in steps. The signals were defined to work along the simulated PLC. This component works as a manager to the complete cell. For example, before the Pick function, the robot interacts with the PLC to get permission to continue the operation. The PLC allows the robot to pick products only if the picking conveyor has products ready to be picked. In Figure 9 is presented a snippet of the auto generated robot routine, after passing through the "Translation" module (Souza *et al.*, 2019). This figure depicts the Program Editor window, where the Main method and subroutines are declared. The tool and the base

Figure 7 Components modeled for the custom library

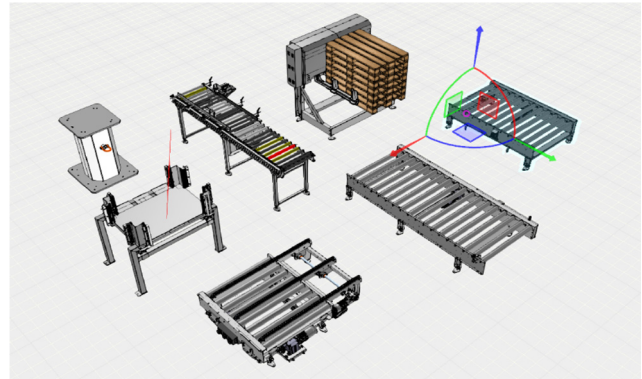
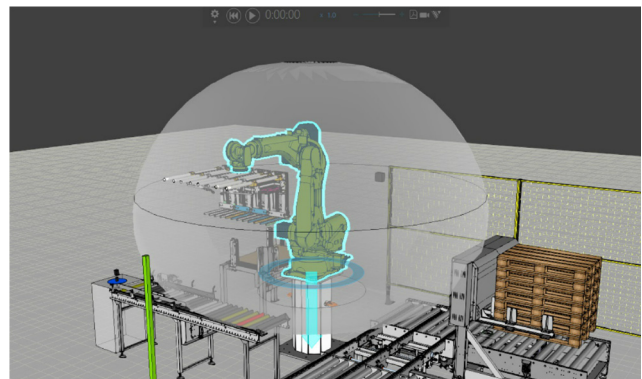


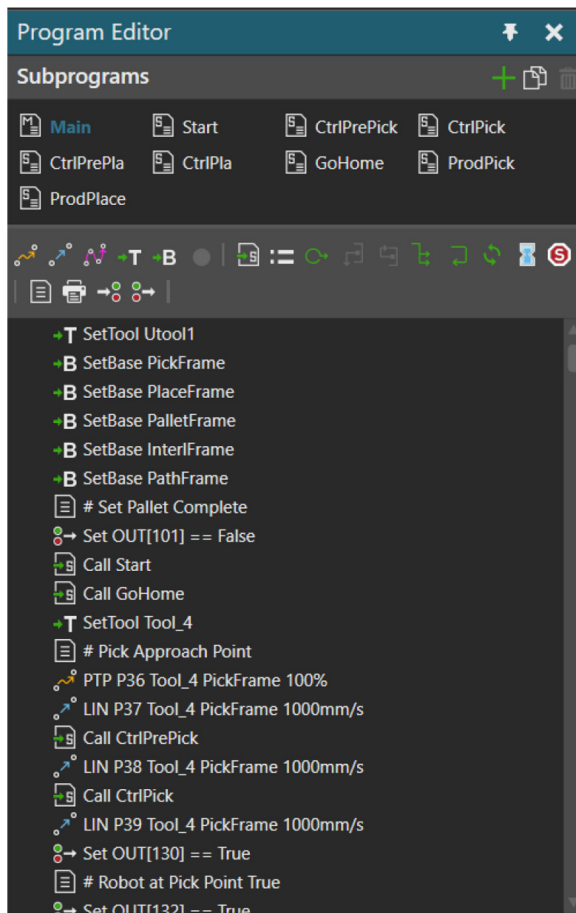
Figure 8 Robot workspace envelope



frames for the robot are defined inside the Main routine, followed by the interlayer handling subroutine, and the palletizing script. During the experiment, the robot was designated to pick and place packs of cans in specific positions, following the stack mosaic given as input to the system.

As presented in [Castro et al. \(2019\)](#), the generation of the trajectory intermediary points, here referred as Path Planning, is optional from the programmer perspective. Without the activation of the Path planning, the framework only generates the pre-pick, pick, and pos-pick points, followed by the pre-place, place, and pos-place points. As a default, there are no delimited points between the pos-pick and pre-place point; the robot performs the lowest cost joint motion between these. For the test scenario considered, the “Robot Trajectory Generation System” befitted adequately. With the tight limits imposed by the fences, there are collisions when there is no trajectory computation. Considering this compact operational space, the production of a linear trajectory, consisting of multiple connected lines, was the outcome of the Path Planner. An interval of 300 mm between each point established the discretization parameter for the Cartesian algorithm. The experiments consisted of packing three different products. [Figures 10](#) and [11](#) display snapshots of the blue and green packing process. [Figure 12](#) displays red packs. As

Figure 9 Generated palletizing code



illustrated, the robot could successfully follow the trajectory points and stacked the products according to the mosaics provided.

3.2 Real world tests

One of the objectives of this work was to be able to perform the offline programming of the real robots used in palletizing cells. For this reason, the virtual cell presented in the last subsection served as a model for the real-world cell ([Figure 6](#)). Before

Figure 10 Stacking packs of blue boxes

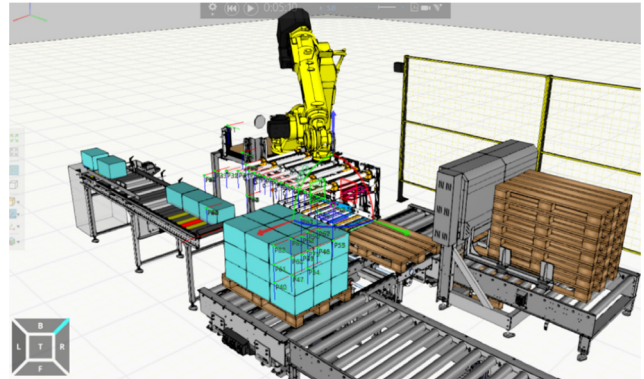


Figure 11 Stacking packs of green boxes

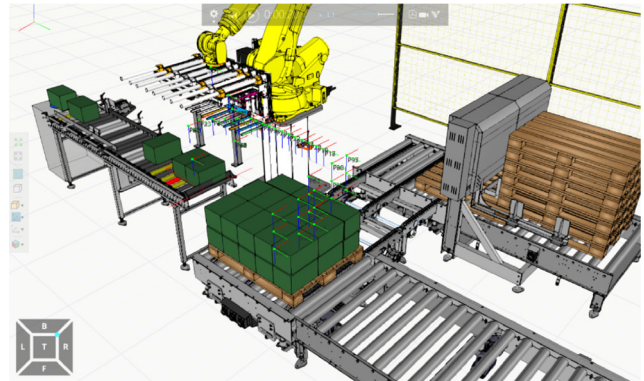
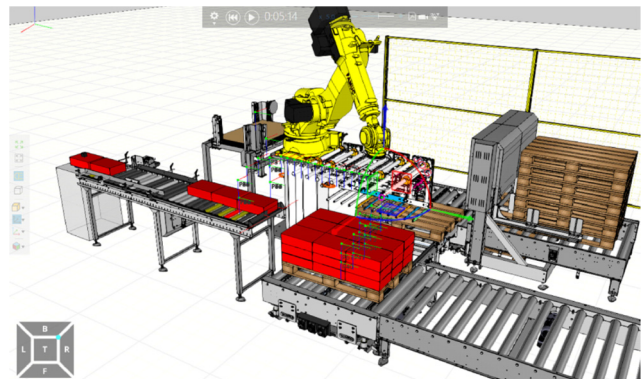


Figure 12 Stacking packs of cans in simulation



exporting the generated code from the previous simulation to the real system, it was necessary to correct minor displacements from the virtual to the real cell. This step is called “Cell Model Calibration”. It is hardly possible to assemble a robotic cell with precise and exact measurements from the simulated cell. In the design phase, external variables, not yet considered, have to be examined after the real cell assemblage. For example, the floor inclination scales rapidly when the distance between two points increases, causing range and angular error from simulation to the real world. However small those errors may remain, still is risky for collisions to happen, since there is an absolute need for millimeter accuracy in picking operations.

The calibration process used in the presented work is based on the measurement of the coordinates of some relevant points in the real robotized cell, using a specific robot tool (typically a pointy rod) as the measurement “instrument”, and adjusting the cell model, developed in the AdaptPack Studio, so that the coordinates of the same relevant points in the simulation model are the same as in reality. After performing the calibration process, the experiment displayed in Figure 12 was again generated automatically and translated to the real robot, with a full guarantee of being collision-free (Figure 13). The outcomes from the test scenario exhibited here confirm the previous results in simulation. The robot followed the same trajectory points and stacked the products with millimetric, however inevitable, displacements.

4. Conclusion

The use of automatic palletizing solutions has been expanding in the Fast Moving Consumer Goods (FMCG) business. Owing to the diversity of production, palletizing processes are changing from low assortment with high volume to large variety with little amount. Often, these changes require modifying machine tasks for different incoming supplies. It implies updating all the programmed routines of robots, as well as changing the palletizing mosaic. Accordingly, this work proposed an answer related to the development of a qualified framework to enhance automatic palletizing processes and solve preceding issues. This paper described a software framework for the automatic offline programming of palletizing applications. This improvement speeds up the task of designing palletizing cells, since it allows to model cells from scratch, using the functionalities presented in the AdaptPack Studio

Figure 13 Stacking packs of cans in the real world



software. Besides, it also enables the integrated and modular development of new component models, reducing the time to design, develop, build, and install new equipment.

The automatic generation of robot programs, based on inputs inside a GUI, has been detailed. These inputs consist of clicking at picking and placing positions inside a 3D modeled cell, selecting the mosaic which defines the product stacking aspect, and writing general configurations in text fields. Also included inside this new framework are two collision-free trajectory generators, based on the A* informed search algorithm. The first searches a path in the robot's joints space, different from the second, which explores a path in the Cartesian space of a palletizing cell. Finally, the “Robot Trajectory Generation System” was tested with a simulated and a real test cell simulation. Based on these tests, the system worked as intended, being able to generate collision-free palletizing routines automatically. This proposal improves the post-process procedure during offline programming and supports engineers in the project development. Furthermore, this methodology can handle the present demands of companies for fast design of flexible solutions, as verified during the tests at the JPM Industry facilities.

Concerning possible future enhancements to the AdaptPack Studio framework, the focus will be on the A* Trajectory Generator based on the Cartesian Discretization. The robot path generated using this method resulted in straight line trajectory segments between points, being appropriated to smooth this path to reach one that resembles more the robot's joints natural movements.

References

- Anylogic (2020), “Simulation modeling software tools & solutions for business”, available at: www.anylogic.com/ (accessed 12 June 2020).
- Castro, A., Souza, J.P., Rocha, L. and Silva, M.F. (2019), “AdaptPack studio: automatic offline robot programming framework for factory environments”, *19th IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC 2019)*, pp. 1-6, doi: [10.1109/ICARSC.2019.8733626](https://doi.org/10.1109/ICARSC.2019.8733626).
- Cheng, H. and Chen, H. (2013), “Autonomous robot teaching using a smart robot in production line”, *2013 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 1772-1777, doi: [10.1109/ROBIO.2013.6739724](https://doi.org/10.1109/ROBIO.2013.6739724).
- Do, H.M., Choi, T., Park, D. and Kyung, J. (2015), “Automatic cell production for cellular phone packing using two dual-arm robots”, *2015 15th International Conference on Control, Automation and Systems (ICCAS)*, pp. 2083-2086, doi: [10.1109/ICCAS.2015.7364713](https://doi.org/10.1109/ICCAS.2015.7364713).
- FlexSim (2020), “Simulation software for manufacturing, material handling, healthcare, etc.”, available at: www.flexsim.com/ (accessed 12 June 2020).
- Kito, K., Tatsuno, K., Otao, S., Hosotani, T. and Yohino, K. (2017), “A robot controller for a working cell”, *2017 International Symposium on Micro-NanoMechatronics and Human Science*, pp. 1-7, doi: [10.1109/MHS.2017.8305164](https://doi.org/10.1109/MHS.2017.8305164).
- Moura, F.M. and Silva, M.F. (2018), “Application for automatic programming of palletizing robots”, *2018 IEEE International Conference on Autonomous Robot Systems and Competitions*, pp. 48-53, doi: [10.1109/ICARSC.2018.8374159](https://doi.org/10.1109/ICARSC.2018.8374159).

- OCTOPUZ (2020), "Robot programming and simulation software", available at: <https://octopuz.com/> (accessed 12 June 2020).
- RoboDK (2020), "Simulator for industrial robots and offline programming", available at: <https://robodk.com/> (accessed 12 June 2020).
- Silva, R., Rocha, L.F., Relvas, P., Costa, P. and Silva, M. F. (2017), "Offline programming of collision free trajectories for palletizing robots", *ROBOT 2017: Third Iberian Robotics Conference. Springer International Publishing*, pp. 680-691.
- SIMIO (2020), "Simulation, production planning and scheduling software", available at: www.simio.com/ (accessed 12 June 2020).
- Souza, J.P., Castro, A., Rocha, L., Relvas, P. and Silva, M.F. (2019), "Converting robot offline programs to native code using the AdaptPack studio translators", *2019 IEEE International Conference on Autonomous Robot Systems and Competitions*, pp. 1-7, doi: [10.1109/ICARSC.2019.8733631](https://doi.org/10.1109/ICARSC.2019.8733631).
- Tavares, P., Lima, J. and Costa, P. (2016), "Double a* path planning for industrial manipulators", *Robot 2015: Second Iberian Robotics Conference*, Springer International Publishing, pp. 119-130.
- Tavares, P., Marques, D., Malaca, P., Veiga, G., Costa, P. and Moreira, A.P. (2019), "Optimal automatic path planner and design for high redundancy robotic systems", *Industrial Robot: The International Journal of Robotics Research and Application*, Vol. 47 No. 1, pp. 131-139., doi: [10.1108/IR-09-2018-0194](https://doi.org/10.1108/IR-09-2018-0194).
- Visual Components (2020), "3D manufacturing simulation and visualization software", available at: <https://www.visualcomponents.com/> (accessed 12 June 2020).
- Yaskawa (2020), "Motoman robot software", available at: www.motoman.com/products/software/ (accessed 12 June 2020).
- Yu, H., Shan, J. and Zhu, X. (2011), "Off-line programming and remote control for a palletizing robot", *2011 IEEE International Conference on Computer Science and Automation Engineering*, Vol. 2, pp. 586-589, doi: [10.1109/CSAE.2011.5952539](https://doi.org/10.1109/CSAE.2011.5952539).

Corresponding author

Manuel F. Silva can be contacted at: mss@isep.ipp.pt