POLITECNICO DI MILANO
Dipartamento di Electronica e Informazione
Master of Automation and Control Engineering

Modelling and simulation of a scheduling algorithm for a pick-and-place packaging system.

Graduation thesis of:
Diana Paola Blanco Rendón
Mat. 769807

Supervisor:
Prof. Luca Ferrarini

Tutor:
Eng. Alessio Dede

Academic year 2012-2013

A Dios, mi fiel compañero y guía en esta inolvidable e intensa experiencia de vida. A mis padres, Edgar y Martha y, a mi hermana, Ivonne Lucia. Ustedes son mi motor de vida y sin su apoyo y amor incondicional no hubiera logrado salir adelante y cumplir este sueño

Acknowledgements

I would like to thank Politecnico di Milano for its contribution on my professional growth and Professor Luca Ferrarini for his permanent guidance and help during the complete process of elaboration of this thesis. I also want to express my gratitude to the team at DAISY Laboratory, for your help and constant support during the time taken to bring this work to a successful conclusion.

A Julian Peña y Diego Mora, invaluables amigos y excelentes colegas, por su apoyo y compaia.

A Dios por darme la oportunidad de vivir y cumplir una meta mas de mi vida, y a mis amigos y familia, por que nunca dudaron de mi y me apoyaron incondisionalmente durante todo el tiempo dedicado a vivir esta maravillosa experiencia.

Diana Paola Blanco Rendón

Abstract

In the last century packaging industry has evidence a trend which tends to be more mature as the demand of the sector is growing steadily. In order to improve the efficiency of the processes and save huge labor costs, the traditional packaging industry gradually started to implement the automation or semi-automation techniques to support the picking and placing operations. As an example of these techniques there exist the robotic systems implemented in pick and place packaging lines.

The use of robotic cells and conveyor systems has allowed the packaging industry to solve numerous problems presented regarding the efficiency and labour costs of the packaging processes. However, occasionally, robotic manipulators are incapable of solving pick and place operations without the implementation of accurate control strategies.

This work presents a mathematical formulation of a pick and place packaging system, in terms of dynamic equations, that allows the plant designers to describe the behavior of the whole system analytically and makes attainable an extensive simulation study of the performances of the plant under different operative settings or adopting different control strategies. In addition, a job assignment technique and some scheduling strategies are proposed that will lead to satisfy the requirements given by the plant layout and enhance the properly work of the plant.

The document starts presenting a motivation of the scheduling rules and strategies, for pick and place operations, that have been developed for industrial applications. Then, a detail description of the mathematical model derived and the product assignment strategy defined for the system. Finally, the scheduling strategies designed are presented and analysed by means of algorithms developed in MATLAB®.

Sommario

Nell'ultimo secolo lindustria di confezionamento mostra un andamento che tende ad essere più maturo poiché la domanda del settore è in costante crescita. Al fine di migliorare l'efficienza dei processi e ridurre i costi del lavoro, l'industria di confezionamento tradizionale gradualmente ha iniziato ad implementare le tecniche di automazione o semi-automazione per supportare le operazioni di prelievo e deposito. Come esempio di queste tecniche esistono sistemi robotici implementati nelle linee di confezionamento "pick and place".

L'uso di celle robotiche e sistemi di nastri trasportatori ha aiutato l'industria di confezionamento a risolvere i numerosi problemi presentati per quanto riguarda l'efficienza e i costi dei processi di confezionamento. Tuttavia, occasionalmente, i sistemi robotici non sono in grado di risolvere le operazioni di prelievo e deposito senza l'implementazione di strategie di controllo accurate.

Si presenta una formulazione matematica di un sistema di confezionamento "pick and place", in termini di equazioni dinamiche, che consente al progettista dell'impianto di descrivere il comportamento del sistema analiticamente e permette un ampio studio di simulazione delle prestazioni dell'impianto sotto impostazioni operative diverse o l'adozione di differenti strategie di controllo. Inoltre, una tecnica di assegnazione di lavoro e alcune strategie di scheduling sono proposte per soddisfare i requisiti indicati dal layout dell'impianto e migliorare il funzionamento dell'impianto..

Il documento inizia presentando la motivazione delle regole di pianificazione e le strategie, per le operazioni di "pick and place", che sono state sviluppate per applicazioni industriali. Poi, una descrizione dettagliata del modello matematico ottenuto e la strategia di assegnazione di prodotti definita per il sistema. Infine, le strategie di scheduling progettate sono presentate e analizzate mediante algoritmi sviluppati in MATLAB®.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the last century packaging industry has evidence a trend which tends to be more mature as the demand of the sector is growing steadily. Pick and place operations have been widely used within this kind of industry with the main feature of being labor-intensive and often requiring workers to pack and sort products into boxes, trays or cover boards for further packing. In order to improve the efficiency of the processes and save huge labor costs, and thanks to the development and popularity of the information and automation technologies, the traditional packaging industry gradually started to implement the automation or semi-automation techniques to support the picking and placing operations. As an example of these techniques there exist the robotic systems implemented in secondary packaging lines.

Behind the use of semi-automatic pick and place systems there are critical issues that need to be solved prior the implementation of the physical system. Once the layout of the plant has been defined, mostly described with a typical configuration of a set of work stations performing pick and place cycles from one moving conveyor to another parallel one, the first critical issue is to appropriately assign the items to be picked at each workstation in the conveyor system in order to balance workload of each workstation. The second issue corresponds to define the control laws and strategies that will ensure that all the incoming products are picked and placed from one conveyor to the other one before them leaving the plant. Finally, a simulation of the packaging line is needed to mimic the final behavior of the system and consequently to verify that its throughput is the desired one.

The study of each one of these stages when defining a suitable working strategy for a packaging line starts with the presentation of different layouts and strategies of picking and placing operations that are being implemented nowadays within the packaging industry, in chapter 2. Then, a detailed description of the mathematical model derived for the system, which is the starting point in order to be able to define, simulate and validate any working strategy, is presented in chapter 3. The product assignment strategy defined and some of the control problems encountered with the respective policies required that ensure the properly work of the plant are explained in chapter 4. Chapter 5 shows an analysis based on the results obtained by means of an algorithm developed in MATLAB® for each control strategy. The conclusions and future work are presented in chapter 6.

1.1 Problem Statement

Pick and place operations are commonly employed in industrial workspaces within the packaging industry. The use of robotic cells and conveyor systems has allowed the packaging industry to solve numerous problems presented regarding the efficiency and labour costs of the packaging processes. However, occasionally, robotic manipulators are incapable of solving pick and place operations without the implementation of accurate control strategies. These control strategies depends not only on the way the incoming product is fed but also on the layout chosen for the plant. Each of the necessaries steps, for achieving the best performance of the plant, should be programmed and tested in advance in order to avoid malfunctioning in the real plants, thus, huge losses economically speaking.

A qualitative analysis to evaluate plant performances is use nowadays and seems reasonable, however, plant designers are opting for implementing quantitative methods to priori evaluate the effects of the variables of the system. The derivation of a mathematical model for the plant layout, in terms of dynamic equations, generates a tool that allows the plant designers to describe the behavior of the whole system analytically and makes attainable an extensive simulation study of the performances of the plant under different operative settings or adopting different control strategies. Consequently, the bests control strategies for a specific plant layout can be validate either making use of virtual simulators or in the real system.

Chapter 2

State of the Art

As it is well known packaging plays an important part of society as it connects production and consumption of a wide variety of products. The demand and need for packaging has increased during the last years thanks to the demographic and economic growth, due to this reason, the packaging industry sometimes faces conflicting sustainability demands. The industry response to this issue is widespread, with most companies investing resources into the development of sustainable packaging redefining the areas involved in the sector; actively seek to encourage transformation, innovation, and optimization of the packaging processes.

Within the packaging industry the current packaging processes can be categorized at least in three different types depending on use. The first one, known as Primary or Sales Packaging, refers to the package that holds the product and it gets the attention and the rewards that come with being closest to the consumer. The second type of packaging process, called Secondary or Grouped Packaging, is the one in charge to assemble primary packaging and it is described as any outer wrappings that help to store, transport, inform, display and protect the primary packaging. Lastly, Tertiary or Transport Packaging, offers protection, transport and some navigation for the participants in the channel, combining the products for storage and hauling.

Nowadays, the Secondary Packaging is the blending point between the function-driven Tertiary Packaging and the more brand driven Primary Packaging, hence, most of industrial companies are investing their time and resources in optimizing the efficiencies obtained from this type of packaging through automated packaging lines that involve some robotics system on it. This chapter is organized as follows, section 2.1 presents a short review of the typical configuration of the Secondary Packaging plants pointing its main features. The discussion of standard scheduling rules to dynamic queues with their improved versions and some on-line real time control strategies for coordination of multi-robots systems are described in section 2.2

2.1 Plant Layout in Secondary Packaging

To fully understand the working principle of secondary packaging systems, it is necessary to have clear the inputs and outputs of the system. Inputs of secondary packaging process are either ordered or unordered products coming from the Primary Packaging Processes (PPP) as well as empty containers where those products are going to be packed, and, the output of the process is the containers containing ordered products. The process can be then decomposed into three fundamental sub processes as shown in figure 2.1.

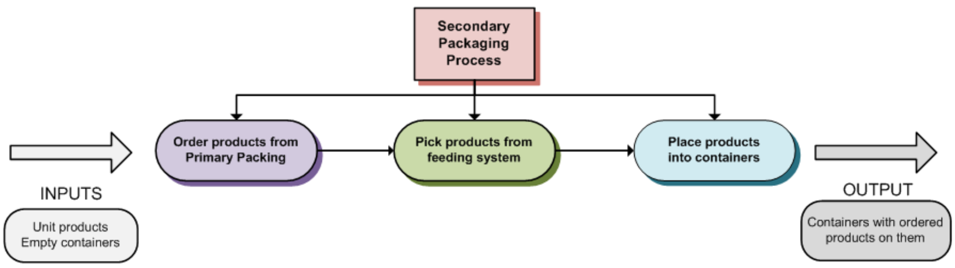


FIGURE 2.1: Sub processes of Secondary Packaging System

As it can be deduced from figure 2.1 the main aim of Secondary Packaging Plants (SPP) is to pick products given by a feeding system and to place them into containers. The layout of containers can be different from product to product, however, the number of units to be placed in each container and its layout are fixed and known for each given product.

In the same way that for the containers, the layout for automated packaging lines can vary depending of the product to be packed, however, typical configuration for these systems consists of a set of sequential robot stations performing pick-and-place cycles from the feeding system to the containers. In one hand , the use of robotics is supported by the fact that it provides many opportunities for improvements of productivity, it may also provide the means of overcoming shortages of skilled labour and the possibility of improvements in working conditions [3].

In the other hand, mostly, the feeding system and the system which transports the containers along the line are described by conveyor belt systems. The primary advantage of implementing a conveyor tracking application relies in that continuous operations may maximize robot utilization and minimize production cycle time, contributing to economic performance and system productivity. Finally, in order to make a robotic manipulator function intelligently, feedback from the environment is indispensable. During the last decade, vision sensors have become of utmost importance and their integration with the robotic cell makes the robot see the environment and adapt its work-flow to varying conditions.

Putting together all the components described previously, a detailed description of the nowadays most used layout of a secondary packaging plant is obtained as illustrated in figure 2.2. Accordingly, the plant consist of two aligned belts transporting the product ready to be packed and the empty containers to a series of robot stations arrange in the conveying direction. The product moved in one belt is to be gripped by the robots at each station and deposited in the containers carried by the other belt. In addition, there is at least one optical registration station to detect the position and/or direction of the individual product when they are supplied in an unordered fashion. In the case that products are supplied in an ordered fashion, the optical registration is rendered superfluous or is used merely for quality control.

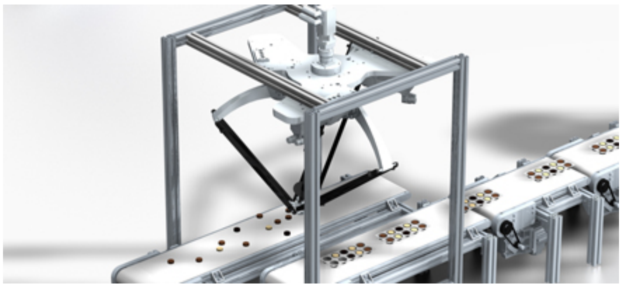


Figure 2.2: Example Secondary Packaging Plant layout. Taken from [12]

As previously introduced, the general purpose of a Secondary Packaging plant is to group primary packages together, picking them from a feeding system and placing them into available empty positions in containers. However, in a properly working plant, the cooperative activity of the robotic cells should ensure that all the incoming products from the feeding conveyor belt are cleared and all the containers entering the system are properly filled up when leaving the plant. In order to succeed in this goal there must exist a balance between the inflow of products on the feeding conveyor belt and the empty spaces in the containers so that each single unit of product can be placed into a suitable position within a container.

In addition, the work of the robotics cells should be distributed in such way that no matching between an incoming piece of product and an empty space in a container is missed. There may exist cases where the conditions previously stated are not completely fulfilled, as result, some products may not be picked from the feeding conveyor or not totally filled containers leave the last robotic cell. Under these cases, it is considered that the plant is malfunctioning, representing a loss. In general, the pick and place process is distributed over the robotic cells and it targets the whole working area of each robotic cell. To avoid dropouts, it is necessary that average flows of product units and of empty spaces within the containers are equal in the area in which the pick and place action is executed.

The direction with which each conveyor belt is moving plays as well an important role when avoiding the dropouts in the system. In order to clearly understand the contribution of this fact, it is important to underline that the different robotics cells operate in sequence along the same conveyor belts, hence, the relative movement of the product and container conveyor belts represents an important variable. There exist two possible solutions for this problem. The first one, referred as co-current flow setting, considers the two conveyor belts moving in the same direction, whereas, in the second solution, known as counter-current flow setting, they move in opposite directions. Figure 2.3 portrays the two cases.

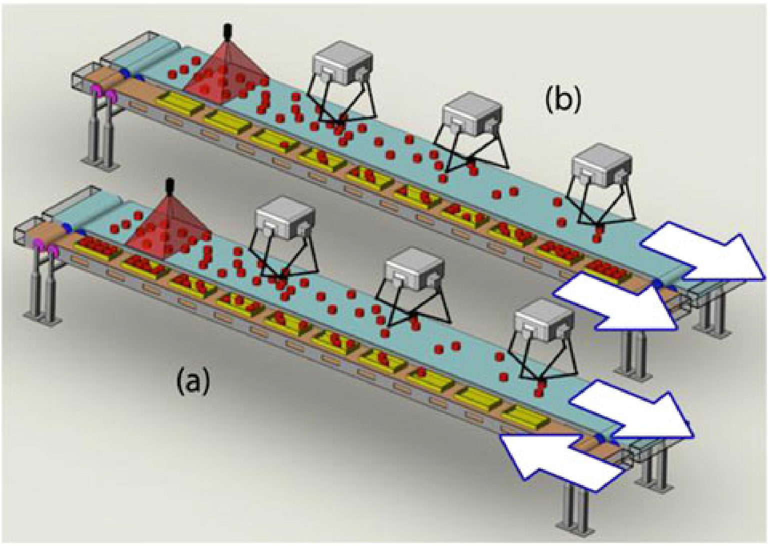


Figure 2.3: Secondary Packaging Plant with counter-current flow (a) and co-current flow (b) configurations.Taken from [2]

2.2 Scheduling Rules and Strategies for Pick and Place Operations

When using conveyor systems in semi-automatic and automatic picking systems, one of the critical issues is to appropriately assign the items to be picked to each work station from the conveyor system in order to balance workload of each workstation and to enhance the overall resource utilization of the process. The number of workstations and items to be picked at each workstation should be determined while generating a job assignment plan of a conveyor-aided-picking system. Generally, job assignment of the conveyor-aided picking system is carried out through a set of empirical or scheduling rules that achieve the balance needed to enhance the picking efficiency [5] .

Furthermore, when products are fed randomly, the task that each robot is going to perform when picking the products up from the conveyor belt should be as optimal as possible. In particular, the overall system must address the following concerns effectively [1] :

- Real-time operation with simple programming requirements.
- Working in all regions of its workspace while accommodating to changes in the working environment.
- Flexible integration and control.

Some examples of standard scheduling rules and strategies for multi-robot coordination, in pick and place operations, are presented below.

2.2.1 FIFO Rule

This rule establishes that the elements in the queue are served in the same order they enter it [4]. Considering the items on a moving conveyor as a queue of N elements waiting to be gripped by a robotic system for being placed in a fixed position. Therefore, each element which enters first to the working area of the robot is assigned to this latter. Its advantages are:

- Its complexity is constant, specially because it does not require computing and sorting the gripping times of each item within the queue, hence, it has a very low computational cost.
- In case of under-load, the rule proves to be optimal for many classes of scheduling problems.

On the other hand, its main disadvantage is:

- In case of over-load, the rule proves to be inefficient, it gives unacceptable values of gripping rate and the system tends to work under the boundaries of the working area.

2.2.2 SPT Rule

For static queues, the minimum average waiting time is achieved by always serving first the clients with the shortest processing time. In the dynamic case, in the same way that it was considered for the FIFO rule, this law still holds, though not with the same optimality. The aim to achieve is to grip as many items as possible and spending at each step the shortest time to grip an item allows to obtain it. As a matter of fact, this rule actually gives better results than the FIFO rule, however its disadvantages are:

- It schedules the items on the mere basis of the gripping rate, without taking into account the admissibility constrains in the position of the items.
- It is not flexible, in particular, it does not allow to spend ”a little bit longer” to grip an item that will otherwise exit the working area of the robot.

2.2.3 Improved FIFO Rule

In order to improve the performance of the standard FIFO rule, a modification of the latter was proposed at [5]. As it was stated at section 2.2.1, the main limitation of the FIFO rule is that in overload situations it causes the system to become unstable and, consequently, to work around the workspaces boundaries, where the gripping rate is maximum [5].

The proposed solution to overcome the problem it is shown in figure 2.4. The main idea is to recognize the robot workspace configuration and to restrict the working area of the robot to the optimal gripping area. In other words, the robot will not consider the objects that gone beyond the minimum gripping time point, resulting in a gripping rate not much worse than that achieved by the STP rule.

2.2.4 Improved SPT Rule

The basis of this rule are depicted in figure 2.5. To achieve the improvement of the SPT rule it is necessary to recognize that taking other object than the quickest one may increase the gripping rate. In order to fast evaluate the new scheduling rule, as portrayed in figure 2.5, let the *optimal gripping area* be defined as the conveyor region around the minimum-gripping-time-point, whose boundaries are characterized by a constant gripping time , and $t_g = T^*$, and whose width is $L = v_b \bullet T_M$, where T_M is the longest pick-and-place time. Thus, any point in the optimal gripping area is characterized by

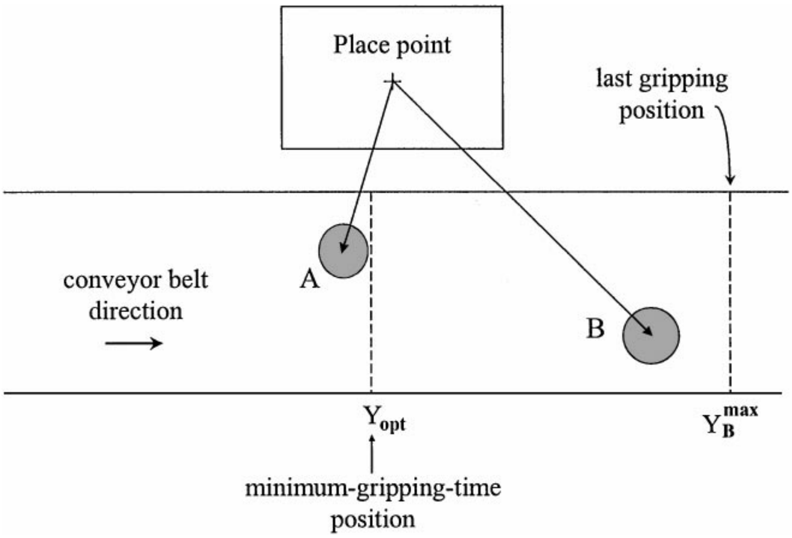


FIGURE 2.4: Situation where FIFO performance can be improved. Taken from [4]

a gripping time $t_g \leq T^*$. Then, the *upstream area* and the *downstream area* are, respectively, the admissible regions *before* and *after* the optimal gripping area [4] (see figure 2.5)

Finally, the improved SPT rule results as follows. Let O_1 be the SPT object in the list, and O_2 the SPT object among those in the downstream area, if O_1 is not in the upstream area, the take O_1, else take O_2 [4]

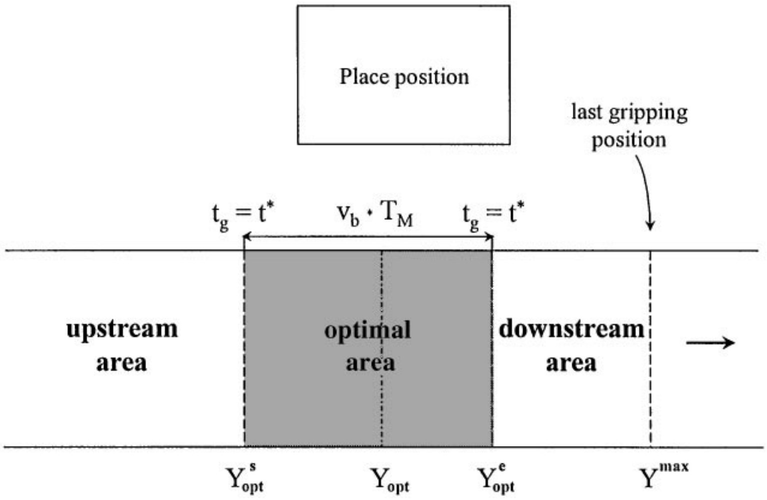


FIGURE 2.5: Descomposition of the working area of the robot into three areas for the Improved SPT rule. Taken from [4]

2.2.5 Distributed Constrain Optimization Problem (DCOP)

In multi-robot coordination strategies, the main concern is to achieve an instance of optimal assignment of a set of tasks to a set of robots based on maximizing the overall performance of the system, while taking their individual performances into consideration [1]. A DCOP is a COP in which the variables and constrains are distributed among automated agents, each of the many autonomous agents controls a single variable and together the agents have the joint goal of maximizing a global objective function. Within the classification of DCOPs alghoritms two groups can be recognized, namely search algorithms and consistency algorithms [1].

2.2.5.1 Search Algorithms

These algorithms aim to maximize a global objective function. The most trivial algorithm used is to select a leader agent among all the agents, and gather all information about the variables, their domains and their constrains, into the leader agent. The leader then solves the COP. However, due to the computational complexity involved with the message passing structure, these algorithms are hard to implement in real life, the cost of collecting all information about the problem can proved to be prohibited [7].

2.2.5.2 Consistency Algorithms

In these algorithms each agent reacts on the basis of local knowledge. However, since each agent's utility is defined based as the sum of the terms of the global objective function that it is involved, its decision-making has to consider all the associated robots pay offs as well.Furthermore,as the robots are changed,the individual utility functions have to be reconsidered and changed accordingly [1] .

2.2.6 Non-cooperative Dynamic Game

In contrast with the stated on section 2.2.5, in non-cooperative game theory a set of objective functions is simultaneously optimized. Here, each agent decides for its moves based on only individual considerations associated with its assigned task [1].

In the non-cooperative games, each robot is associated with a cost function that simultaneously takes itself and its neighbors into consideration. Each robot knows which action is the best for its self-interest, however, when choosing with action is the one to execute, it also considers its neighbors' actions and chooses an action based on the trade-off between doing the best for its own interest and picking up products that are least likely to be picked up by the neighbors within a non-cooperative dynamical game.

By non-cooperative, it is meant that the robots decide individually as opposed to forming coalitions to decide collectively. By dinamical, it is meant that each robot views the process as a repeated game with its neighboring robots. As the enviroment is time-varying with the continuous flow of products on the conveyor band, this is a dynamic game in which all the robots participate repeatedly as long as the conveyor is moving or contains products [1].

The framework of this strategy says that each robot needs to communicate with only its neighboring robots, as each robot operates in a manner that accommodates changes on the conveyor belt as well as neighbors' actions, robots end picking up products from the whole area contained within their workspaces.

Chapter 3

Mathematical Model of the System

A mathematical model is a representation in mathematical therms of the behavior of real devices and objects [8]. This is a general but complete definition of the concept of mathematical modeling, however, within this concept is not said that mathematical models can never make a completely precise model of the physical system, there will always be phenomena which are not able to be modeled. Thus, mathematical models will always carry a portion of error and uncertainty within them. But even if a mathematical model just achieve to describe a part of the reality it can be useful for analysis and design purposes, the only necessary condition is that the model must describe accurately the dominating dynamic properties of the system.

This chapter starts with a short introduction about the basic concepts of mathematical modeling and its importance within the engineering world. Subsequently, section 3.2 presents a short but concise description of the real system to be model mathematically. The procedure followed to derive the mathematical model of the system and the final result is described in section 3.3.

3.1 Mathematical Modeling: Basic Concepts

Roughly defined, mathematical modeling aims to describe the different aspects of the *real world*, their interaction and their dynamics through mathematics [9]. The term *real world* refers essentially to any system whose behaviors can be observed whether these behaviors are natural in origin or produce by artifacts. Mathematical models are characterized by their simplicity but also by the accuracy of the results of their analysis, the goal is usually obtain a ”sufficiently accurate” and flexible model at low cost. It is important to highlight that a mathematical model is not unique to a given system. A system may be represented in many different ways and, therefore, may have many mathematical models, depending on the builder’s perspective.

For engineers, who have as main roles to describe and analyze objects and devices and to design devices, processes and systems, there exist many reasons why modeling turns out to be useful and in most of the cases necessary, but all those reasons are related and summarized in the following two:

- **To gain understanding**. Generally speaking, an improved understanding of any real word system can be gained once that is built a model that accurately reflects some behavior of it. In addition, in the process of the construction of the model, it can be identified the most important factors of the system and how they are related among them.
- **To predict or simulate**. Mathematical models become an important tool that allows to know what a real system will do in the future without experimenting directly with the system. As consequence, process of design becomes cost effective and time saver.

3.1.1 Process of modeling

Mathematical modeling as any other process should follow a sequence or procedure in order to achieve the desired result, namely an accurate representation of the system of interest. This procedure or sequence can be described as below:

1. **Define system boundaries**. In most of the cases physical systems work in co-operation with other systems. Therefore it is necessary to define the boundaries of the system before beginning developing the respective mathematical model.

2. **Make simplifying assumptions**. Setting up assumptions when modeling helps to set limits to the problem and thus provide a framework within which to work. In addition, assumptions simplify the problem and make it more manageable making mathematical work possible.

3. **Use the Balance law for the physical balances in the system and define eventual additional conditions**. Breafly, Balance Law says that *the rate of*

change of any physical property in a system, Q(t), is equal to inflows minus outflows plus and less the rates of generated and consumed Q respectively, see figure 3.1. Therefore, the Balance Law can be expressed as follows:

$$\frac{dQ(t)}{dt} = q_{in}(t) + g(t) - q_{out}(t) - c(t) \tag{3.1}$$

Additional conditions can be defined to the model ($Q(0)$). Then, Q at the time t is:

$$\mathrm{Q}(t) = \mathrm{Q}(0) + \int_0^t q_{in}(t) + g(t) - q_{out}(t) - c(t)\,\mathrm{d}t \tag{3.2}$$

Where $Q(0)$ is the condition of the system at time $t = 0$

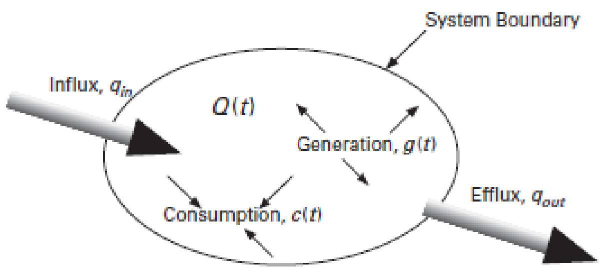


FIGURE 3.1: Illustration of the Balance Law . Taken from [8]

4. **Draw an overall block diagram showing inputs, outputs and parameters**. Using a block diagram makes the model appear clearly. An example of such representation is shown in Figure 3.2.

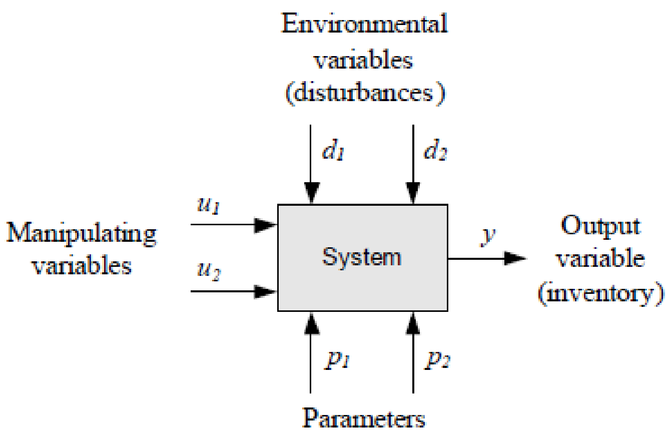


FIGURE 3.2: Overall block diagram. Taken from [11]

5. **Present the model in a proper form**. The most common model form are block diagrams, state models, transfer functions and differential equations.

Other way to analyze the process of mathematical modeling is looking at it as a principled activity. Figure 3.3 depicts a basic philosophical approach based on methodological modeling principles that express the intentions and purposes of mathematical modeling.

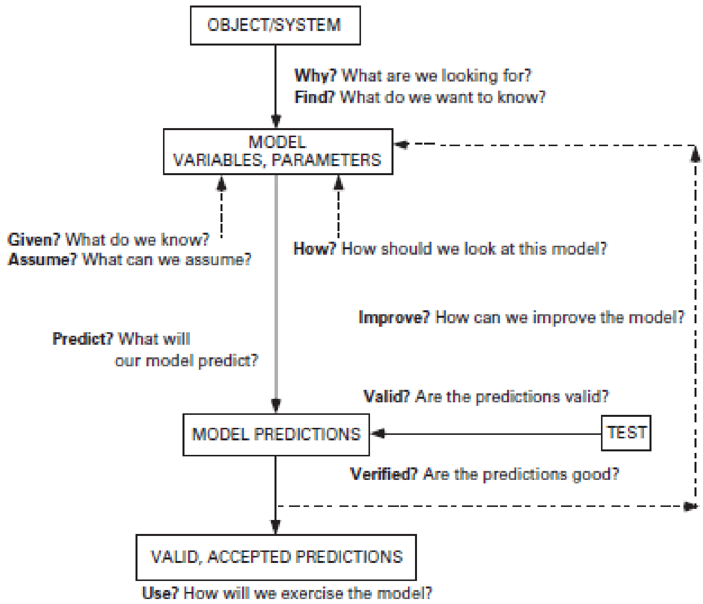


FIGURE 3.3: Philosophical approach to built a mathematical model. Taken from [8]

3.2 Secondary Packaging System Description

The secondary packaging system of interest is described in Figure 3.4. Specifically, the system has two conveyors belts, 1 and 2, one for feeding piece goods and other for feeding empty containers respectively. These two devices extend parallel to each other and work under a co-current principle, see section 2.1. The conveying directions of the individual conveyors are indicated by large black arrows. Individual grasping units or pickers *R1* to *R4* are situated in sequence along the conveying section; they can pick up piece goods from the feeding conveyor belt individually and place them into the containers. Depending on the type of piece good to be packaged, it is also possible to use pickers that are able to pick up the piece goods in groups and place them together into the containers.

Furthermore, the system has also a robot control unit *RS*, which controls each robot and an optical detection system *O* upstream of the robots in the conveying direction, which detects the position of the individual piece of goods and forward this data to the *RS*.

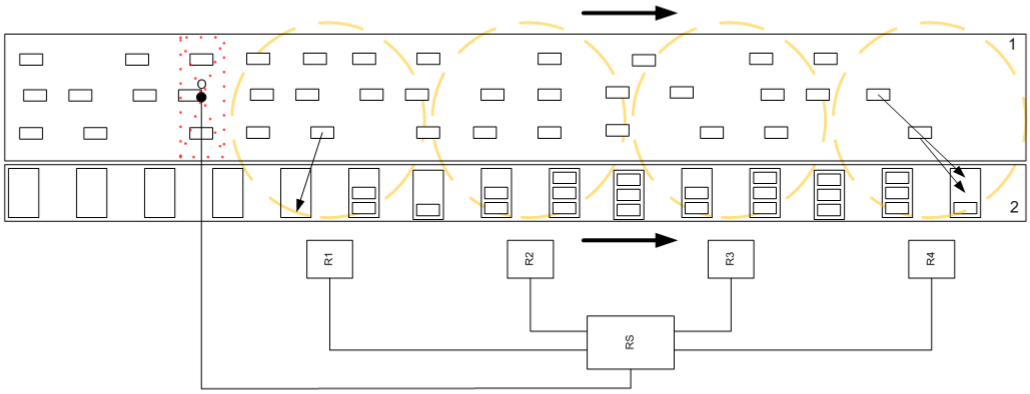


FIGURE 3.4: Layout of the system to be modeled mathematically

The objective of the system is to fill all the containers completely before leaving the area of use of *R4*, as well as, to clear all the piece goods from the feed conveyor belt 1. In order to fulfill the first condition, the definition of a control law for the action of the conveyor belt 2 is needed. The fulfillment of the second condition is guaranteed by the strategy according to which each robot decides which piece goods item should pick from those situated within its working range. The initial conditions for the system are the conveyor belts, 1 and 2, switched off and empty containers distributed uniformly along the working length of the conveyor belt 2.

3.3 Mathematical Model of the Secodary Packaging System

In Section 2.1 was mentioned that a Secondary Packaging System (SPS) can be decomposed into three fundamental sub processes, see Figure2.1. This work is focused mainly in the two last sub processes, the process of picking products from the feeding system (product stream) and the process of placing products into empty containers. Actually, these two sub processes, one dealing with the product stream and the other with the hole stream, can be recognized as discrete dynamical systems, and differential equations were used to derive the respective mathematical models.

For each sub process three different detail levels are illustrated. Each case analyzes the sub system within a different study scenario. Although, the working conditions keep being the same for all the cases, the level of complexity varies for each one of them; the bigger the level of detail, the bigger the complexity of the model and the smaller the flexibility of it. Following sections will describe the different levels evaluated, starting with the most general model that can be obtained until one that has certain level of detail and complexity that allows to describe more accurately the specific internal dynamic of the whole system.

3.3.1 Level 0. Modeling each Robot Cells

Level 0 makes reference to the most general way in which the SPS can be seen for modeling purposes. In this level, the system is analyzed as a plurality of successive cells as is illustrate in Figure 3.5.

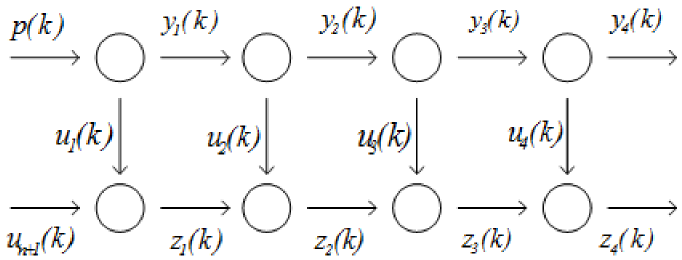


FIGURE 3.5: Level 0: Discrete block diagram of the system

3.3.1.1 Model of the Product Stream

The mathematical model for the product stream is based on the principle of maintaining the mass of the products in each robot cell. As Figure 3.5 portrays, the output of each robot is discretized, therefore, the model of the product stream can be represented by a set of differential equations as shown in equation 3.3.

$$\begin{aligned} y_1(k+1) &= p(k) - u_1(k) \\ y_2(k+1) &= y_1(k) - u_2(k) \\ y_i(k+1) &= y_{i-1}(k) - u_i(k) \\ y_n(k+1) &= y_{n-1}(k) - u_n(k) \end{aligned} \tag{3.3}$$

Where n is the total number of robots in the system, $p(k)$ represents the product stream entering the system, $y_i(k)$, with i=1,2,...,n, corresponds to the product units that leave the robot cell *i-esima* towards the cell *i+1* and $u_i(k)$ makes reference to the product units that leave the robot cell *i-esima* towards the containers.

The criteria defined to derive the respective mathematical model for each robot cell are:

- Discrete instant k given by seconds
- The time interval between successive$k$ is chosen in such a way that product units entering a robot cell leave it at the end of the time interval.
- All product units entering at a time k (at the beginning of the time interval) plus product units taken at the same time k (during the time interval from k to $k+1$), leave the robot cell at time instant $k+1$
- All the product units entering a robot cell, leave it as well, hence, the model does not represent accumulation of product inside a robot cell, in other words, the robot cell can be considered as empty at each k

3.3.1.2 Model of the Hole Stream

The model of the hole stream follows the same principle than the product stream explained in section 3.3.1.1. Similarly, the dynamics of the hole stream at each robot cell can be represented by a differential equation as expressed in equation 3.4

$$\begin{aligned}
z_1(k+1) &= u_{n+1}(k) - u_1(k) \\
z_2(k+1) &= z_1(k) - u_2(k) + u_{n+1}(k-1) - z_{n+1}(k) \\
z_i(k+1) &= z_{i-1}(k) - u_i(k) + u_{n+1}(k-i-1) - z_{n+1}(k-i-2) \\
z_n(k+1) &= z_{n-1}(k) - u_n(k) + u_{n+1}(k-n-1) - z_{n+1}(k-n-2) \\
z_{n+1}(k) &= u_{n+1}(k)
\end{aligned} \tag{3.4}$$

In this case, n and $u_i(k)$ keep having the same meaning like in the model of the product stream. $z_i(k)$, with i=1,2,...,n, corresponds to the holes that leave the robot cell *i-esima* towards the cell *i+1* and $u_{n+1}(k)$ represents the hole stream which enters the system.

Analyzing eguation 3.4 it can be observed that the first differential equation, counting from top to bottom, states that the holes leaving the first robot cell are equal to the ones entering at the previous k instant less the ones occupied in the current k instant. However, with the successive robot cells, this interpretation does not apply. Equation 3.5 shows a further interpretation for these differential equations in order to understand the dependency of the hole stream of any robot cell with respect with the output of the previous robot cells .

$$
\begin{aligned}
z_2(k+1) &= u_{n+1}(k-1) - u_1(k-1) - u_2(k) \\
z_3(k+1) &= u_{n+1}(k-2) - u_1(k-2) - u_2(k-1) - u_3(k) \\
&\vdots \\
z_n(k+1) &= u_{n+1}(k-n-1) - u_1(k-n-1) - u_2(k-n-2) - \cdots - \\
&\quad u_{n-1}(k-n-(n-1)) - u_n(k)
\end{aligned}
\tag{3.5}
$$

The criteria defined to derive the respective mathematical model are:

- Time instant k the same as described in section 3.3.1.1
- There is no accumulation of holes in the robot cells during the time interval $[k, k+1]$
- Containers do not have zero velocity at any time instant

3.3.2 Level 1. Modeling Filling Positions of Robot Cells : Only one pick and place area

Level 1 sees the SPS in a more deep way than the level 0. The system keeps being seen as a discrete one comprised by a plurality of successive cells but, in addition, each robot cell is divided in sub cells that coincide with the number of boxes seen by each robot within its working area. Furthermore, the first robot sub cell is chosen as the picking area of the robot meanwhile the others are considered as transit. The same concept goes for the placing area, see Figure 3.6.

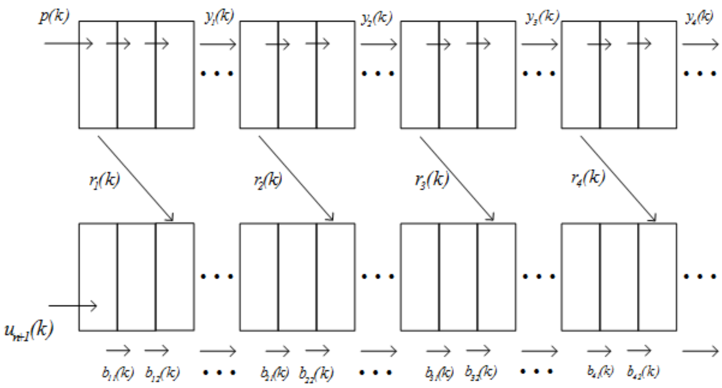


FIGURE 3.6: Level 1: Discrete block diagram of the system

On one side, the criteria considered for modeling both, product stream and hole stream, are as follow:

- Discrete instant k given by seconds
- The time interval between successive k is given by the distance between two consecutive containers and the maximum velocity with which the container conveyor works
- The velocity of the product conveyor is considered constant and is chosen in such way that the product units travel from one sub cell to the next one in a time interval $[k, k+1]$
- The incoming of product is considered constant at each discrete instant
- Containers can have zero velocity at any time instant

On the other side, once the level of detail to be modeled increases, the system should be restricted in such way that the output of each robot cell will describe the intended behavior for the plant. Thus, the system is restricted through the implementation of some specifications which are:

- At each time instant k the constant incoming of product units is equal to the holes inside each container
- When the placing positions within the container located in the placing area of each robot cell are not available, the robot must not pick any product unit from the respective picking area over the product conveyor. Specifically $r_i(k) = 0$

3.3.2.1 Model of the Product Stream

The mathematical model is based on the principle of maintaining the mass of the product units inside each robot cell as it was done in Level 0. Equation 3.6 describes the model of the product stream in each robot cell. Where n is the total number of robots in the system, m is the number of time instants between two contiguous cells, $p(k)$ is the product stream entering the system, $y_i(k)$, with i=1,2,...,n, corresponds to the product units that leave the robot cell *i-esima* towards the cell *i+1* and $r_i(k)$ refers to the product units that leave the picking area of the robot towards the containers.

$$
\begin{aligned}
y_1(k+m) &= p(k) - r_1(k) \\
y_2(k+m) &= y_1(k) - r_2(k) \\
y_i(k+m) &= y_{i-1}(k) - r_i(k) \\
y_n(k+m) &= y_{n-1}(k) - r_n(k)
\end{aligned} \tag{3.6}
$$

3.3.2.2 Model of the Hole Stream

For this case, the modeling of the hole stream can not be done for each robot cell like it was done in the previous cases. In particular, if the system considers the fact that the containers can have zero velocity at any time instant, it means the model should not describe the dynamic of holes within the robot cell but the dynamic of holes within each robot cell's sub cell. Equations 3.7, 3.8 and 3.9 characterize the hole stream at each robot cell's sub-cell.

$$\begin{aligned}
b_{11}(k+1) &= u_{n+1}(k)\cdot v(k) + b_{11}(k)\cdot(1-v(k)) \\
b_{12}(k+1) &= b_{11}(k)\cdot v(k) + b_{12}(k)\cdot(1-v(k)) \\
&\vdots \\
b_{1m}(k+1) &= b_{1(m-1)}(k)\cdot v(k) + b_{1m}(k)\cdot(1-v(k)) - r_1(k)
\end{aligned} \tag{3.7}$$

$$\begin{aligned}
b_{i1}(k+1) &= b_{(i-1)m}(k)\cdot v(k) + b_{i1}(k)\cdot(1-v(k)) \\
b_{i2}(k+1) &= b_{i1}(k)\cdot v(k) + b_{i2}(k)\cdot(1-v(k)) \\
&\vdots \\
b_{im}(k+1) &= b_{i(m-1)}(k)\cdot v(k) + b_{im}(k)\cdot(1-v(k)) - r_i(k)
\end{aligned} \tag{3.8}$$

$$\begin{aligned}
b_{n1}(k+1) &= b_{(n-1)m}(k)\cdot v(k) + b_{n1}(k)\cdot(1-v(k)) \\
b_{n2}(k+1) &= b_{n1}(k)\cdot v(k) + b_{n2}(k)\cdot(1-v(k)) \\
&\vdots \\
b_{nm}(k+1) &= b_{n(m-1)}(k)\cdot v(k) + b_{nm}(k)\cdot(1-v(k)) - r_n(k)
\end{aligned} \tag{3.9}$$

The notation used in Equations 3.7, 3.8 and 3.9 is described as follow: n is the total number of robots in the system, m is the number of sub cells within each robot cell, $u_{n+1}(k)$ represents the hole stream which enters the system, $b_{im}(k)$ refers to the holes remaining in the container placed in the sub cell *m-esima* of the robot cell *i-esima*, $r_i(k)$ corresponds to the product units that leave the picking area of the robot cell *i-esima* towards the containers and $v(k)$ characterizes the velocity of the container conveyor belt. Equation 3.10 shows how $v(k)$ is defined.

$$v(k) = \begin{cases} 1, & \text{if } b_{nm} == 0, \\ 0, & \text{Otherwise} \end{cases} \tag{3.10}$$

3.3.3 Level 2. Modeling Filling Positions of Robot Cells : No restrictions in the pick and place area

Level 2 is a more complete approach for the mathematical model of the SPS chosen. It follows the same concept of sub cells inside each robot cell like level 1, but in contrast, in level 2 there is no restriction about the picking and placing area of each robot cell. However, it must be pointed out that there exists a parity between a sub cell on the product conveyor and a sub cell on the container conveyor as is indicated in Figure 3.7.

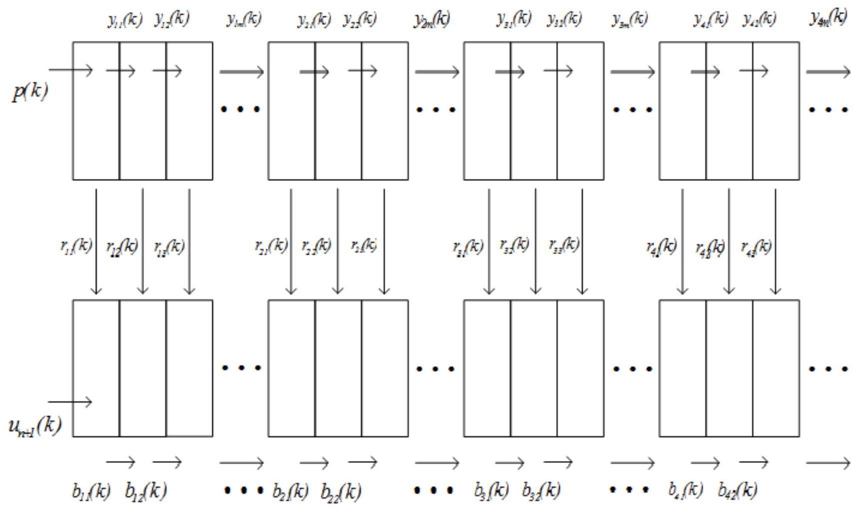


FIGURE 3.7: Level 2: Discrete block diagram of the system

The criteria considered when modeling the product stream and hole stream are the same as described in section 3.3.2. However, the same does not apply for the specifications of the system. Two cases are taking into account within this level, *case 1* and *case 2*. The respective specifications for both cases are summarized below.

1. Case 1

   - At each time instant k the constant incoming of product units is equal to the holes inside each container
   - The robots will always give priority to the filling of the container placed in their first sub cell. If the placing positions of the container in that position are not available, the robot will fill the container located in the next sub cell and so on.

- When any robot finds that none of the containers placed on its sub cells does not have available placing positions, the robot must not pick any product from the product conveyor

2. Case 2

- At each time instant k the constant incoming of product units is equal to the holes inside each container
- The robots will always give priority to the filling of the container placed in their last sub cell. If the placing positions of the container in that position are not available, the robot will fill the container located in the previous sub cell and so on.
- When any robot finds that none of the containers placed on its sub cells does not have available placing positions, the robot must not pick any product from the product conveyor

3.3.3.1 Model of the Product Stream

The modelling principle is exactly the same as the one used when deriving the hole stream model in level 1, see section 3.3.2.2 . Equations 3.11, 3.12 and 3.13 describe the model of the product stream in each robot cell's sub cell. Where n is the total number of robots in the system, m is the number of sub cells within each robot cell, $p(k)$ is the product stream entering the system, $y_{im}(k)$, with i=1,2,...,n, ... corresponds to the product units that leave the sub cell *m-esima* of the robot cel *i-esima*l towards the next sub cell and $r_{im}(k)$ refers to the product units that leave the picking area of the sub cell *m-esima* of the robot cell *i-esima* towards the respective placing area.

$$
\begin{aligned}
y_{11}(k+1) &= p(k) - r_{11}(k) \\
y_{12}(k+1) &= y_{i2}(k) - r_{12}(k) \\
&\vdots \\
y_{1m}(k+1) &= y_{1(m-1)}(k) - r_{1m}(k)
\end{aligned}
\tag{3.11}
$$

$$
\begin{aligned}
y_{i1}(k+1) &= y_{(i-1)m} - r_{i1}(k) \\
y_{i2}(k+1) &= y_{i1}(k) - r_{i2}(k) \\
&\vdots \\
y_{im}(k+1) &= y_{i(m-1)}(k) - r_{im}(k)
\end{aligned}
\tag{3.12}
$$

$$
\begin{aligned}
y_{n1}(k+1) &= y_{(n-1)m} - r_{n1}(k) \\
y_{n2}(k+1) &= y_{n1}(k) - r_{n2}(k) \\
&\vdots \\
y_{nm}(k+1) &= y_{n(m-1)}(k) - r_{nm}(k)
\end{aligned}
\tag{3.13}
$$

3.3.3.2 Model of the Hole Stream

Hole stream model applies the same modeling principle than the product stream . Equations 3.14, 3.15 and 3.16 characterize the hole stream at each robot cell's sub-cell. Where n is the total number of robots in the system, m is the number of sub cells within each robot cell, $u_{n+1}(k)$ represents the hole stream which enters the system, $b_{im}(k)$ refers to the holes remaining in the container placed in the sub cell *m-esima* of the robot cell *i-esima*, $r_{im}(k)$ refers to the product units that leave the picking area of the sub cell *m-esima* of the robot cell *i-esima* towards the respective placing area and $v(k)$ characterizes the velocity of the container conveyor belt. The notation for $v(k)$ is the same used in level 1, see equation 3.10

$$
\begin{aligned}
b_{11}(k+1) &= u_{n+1}(k)\cdot v(k) \ + \ b_{11}(k)\cdot(1-v(k)) - r_{11}(k) \\
b_{12}(k+1) &= b_{11}(k)\cdot v(k) \ + \ b_{12}(k)\cdot(1-v(k)) - r_{12} \\
&\vdots \\
b_{1m}(k+1) &= b_{1(m-1)}(k)\cdot v(k) \ + \ b_{1m}(k)\cdot(1-v(k)) - r_{1m}(k)
\end{aligned} \tag{3.14}
$$

$$
\begin{aligned}
b_{i1}(k+1) &= b_{(i-1)m}(k)\cdot v(k) \ + \ b_{i1}(k)\cdot(1-v(k)) - r_{i1}(k) \\
b_{i2}(k+1) &= b_{i1}(k)\cdot v(k) \ + \ b_{i2}(k)\cdot(1-v(k)) - r_{i2}(k) \\
&\vdots \\
b_{im}(k+1) &= b_{i(m-1)}(k)\cdot v(k) \ + \ b_{im}(k)\cdot(1-v(k)) - r_{im}(k)
\end{aligned} \tag{3.15}
$$

$$
\begin{aligned}
b_{n1}(k+1) &= b_{(n-1)m}(k)\cdot v(k) \ + \ b_{n1}(k)\cdot(1-v(k)) - r_{n1}(k) \\
b_{n2}(k+1) &= b_{n1}(k)\cdot v(k) \ + \ b_{n2}(k)\cdot(1-v(k)) - r_{n2}(k) \\
&\vdots \\
b_{nm}(k+1) &= b_{n(m-1)}(k)\cdot v(k) \ + \ b_{nm}(k)\cdot(1-v(k)) - r_{nm}(k)
\end{aligned} \tag{3.16}
$$

3.3.3.3 Automata representation of the system

Figure 3.8 shows the automata representation of the system.

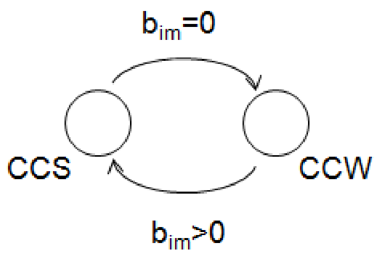


FIGURE 3.8: Level 2: Automata representation for the system modeled in Level 2

Container conveyor stopped (CCS)

At this state the system fulfills the equalities shown in Equation 3.17

Container conveyor working (CCW)

At this state the system fulfills the equalities shown in Equation 3.18

The notation used in Equations 3.17 and 3.18 is the same than the one used in sections 3.3.3.1 and 3.3.3.2

$$
\begin{aligned}
b_{11}(k) - b_{11}(k+1) &= p(k) - y_{11}(k+1) \\
b_{12}(k) - b_{12}(k+1) &= y_{11}(k) - y_{12}(k+1) \\
&\vdots \\
b_{1m}(k) - b_{1m}(k+1) &= y_{1(m-1)}(k) - y_{1m}(k+1) \\
b_{i1}(k) - b_{i1}(k+1) &= y_{(i-1)m} - y_{i1}(k+1) \\
b_{i2}(k) - b_{i2}(k+1) &= y_{i1}(k) - y_{i2}(k+1) \\
&\vdots \\
b_{im}(k) - b_{im}(k+1) &= y_{i(m-1)}(k) - y_{im}(k+1) \\
b_{n1}(k) - b_{n1}(k+1) &= y_{(n-1)m} - y_{n1}(k+1) \\
b_{n2}(k) - b_{n2}(k+1) &= y_{n1}(k) - y_{n2}(k+1) \\
&\vdots \\
b_{nm}(k) - b_{nm}(k+1) &= y_{n(m-1)}(k) - y_{nm}(k+1)
\end{aligned} \tag{3.17}
$$

$$
\begin{aligned}
u_{n+1}(k) - b_{11}(k+1) &= p(k) - y_{11}(k+1) \\
b_{11}(k) - b_{12}(k+1) &= y_{11}(k) - y_{12}(k+1) \\
&\vdots \\
b_{1(m-1)}(k) - b_{1m}(k+1) &= y_{1(m-1)}(k) - y_{1m}(k+1) \\
b_{(i-1)m}(k) - b_{i1}(k+1) &= y_{(i-1)m} - y_{i1}(k+1) \\
b_{i1}(k) - b_{i2}(k+1) &= y_{i1}(k) - y_{i2}(k+1) \\
&\vdots \\
b_{i(m-1)}(k) - b_{im}(k+1) &= y_{i(m-1)}(k) - y_{im}(k+1) \\
b_{(n-1)m}(k) - b_{n1}(k+1) &= y_{(n-1)m} - y_{n1}(k+1) \\
b_{n1}(k) - b_{n2}(k+1) &= y_{n1}(k) - y_{n2}(k+1) \\
&\vdots \\
b_{n(m-1)}(k) - b_{nm}(k+1) &= y_{n(m-1)}(k) - y_{nm}(k+1)
\end{aligned} \tag{3.18}
$$

Chapter 4

Scheduling Algorithms

In pick-and-place packaging systems the manipulation strategy or job assignment is the base or starting point for the definition of the best approach to handling an object. Once a technique for the job assignment has been selected, a handling method or scheduling strategy should be selected. The set of strategies and methods must satisfy the requirements given by the plant layout and enhance the properly work of the plant. In this chapter the description of the control problem, follow by the description of the job assignment chosen for the picking schedule of the conveyor system and, subsequently, the study of some control strategies developed.

4.1 Control Problem

The design of a scheduling algorithm changes depending on the conditions of the system to be controlled. Section 3.2 gives a complete scenario of the initial and working conditions of the system. Control problems related with pick and place strategies for packaging processes can be affected by different causes, such as the layout of the plant, the motion of the conveyor belts, or the fact that the robot picking time is, in general, not constant within its workspace among others.

Summarizing all those issues within a general idea, the main control problem to be addressed is to predetermine whether the speed of the supply of the containers, the speed of the supply of the individual products or the working speed of the pickers, in such way that containers leaving the working area of the last picker located in the conveying direction of the containers are just filled to completion and there are no individual products present any more on the conveying device for the individual products upon leaving the working area of the last picker. Furthermore, the assignment of component types and the working sequencing model for the pickers need to be defined either beforehand or by means of a on-line strategy, thus, the whole throughput of the system could be optimized .

In order to focus on the main peculiarity of the problem, for analysis purposes, it is assumed the job assignment for the pickers is solved before hand and the working speed of the picker is much bigger than the speed of the supply of the individual products, and consequently, much bigger than the speed or the supply of the containers. In addition, conveyor belt 2 is stopped each time the system encounters the last container is not filled to completion when leaving the working area of the last picker.

4.2 Job Assignment Technique

The design of a job assignment plan for a conveyor-aided picking system depends of the number of workstations and items to be picked at each workstation. An inappropriate job assignment plan might result in the unbalanced workload of operators. An optimal job assignment plan seeks to balance the workload of picking operators while enhancing their respective picking efficiency.

When referring to balancing workload the basic principle to implement, for a job assignment design, is to distribute equally incoming product stream among the operators of the system. With this idea on mind, the design of the picking flow should be such that each robot will pick and place the same portion of incoming product than the others. Figure 4.1 shows graphically this principle.

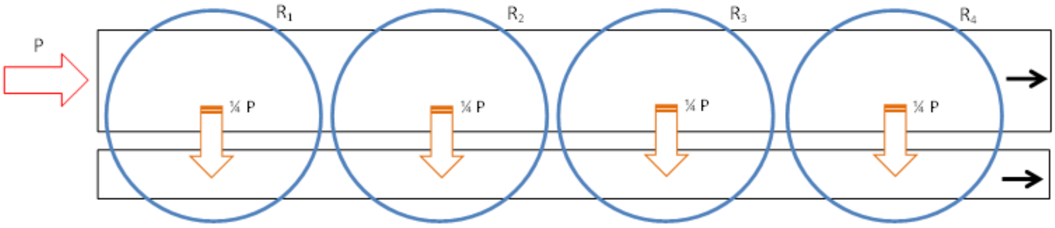


FIGURE 4.1: Distribution of the workload

Moreover, the balancing principle might apply to the place subprocess in the same way, explicitly, if each robot pick one piece each four, the placing should come each four available positions. In particular, the job assignment will define the sequence of filling at each container. Figure 4.2 portrays the specific case when each container has three empty places. Analyzing Figure 4.2 it can be appreciated that there exist a cyclic sequence that depends of the number of robots present in the system, for this specific case, each four containers the filling sequence repeats.

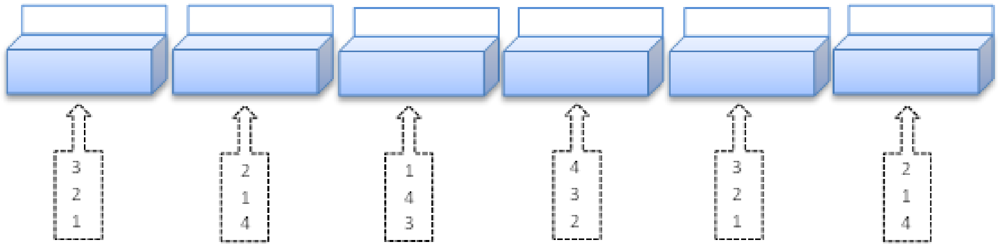


FIGURE 4.2: Sequence of filling of a container with three available spaces based in the balancing principle

The sequence in the product stream might coincide with the sequence of filling of each container; the first piece is assigned to robot 1, the second one to robot 2, the third one to robot 3 and it continuous in this way for all the incoming product stream, see Figure 4.3

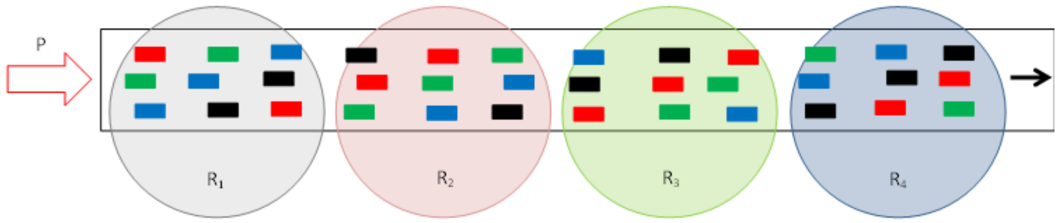


FIGURE 4.3: Sequence of picking based in the balancing principle

4.2.1 Mathematical Model

Job assignment technique will define the number of pieces each robot will pick at each k instant. The respective mathematical model and the criteria defined to derive it are based in the same principle than the entire system, see section 3.3.1.1. Equation 4.1 describes the model of the job assignment technique for each robot cell. Where n is the total number of robots in the system, m is the number of time instants between two contiguous cells, $p(k)$ is the product stream entering the system and $r_i(k)$, with i=1,2,...,n, refers to the product units that leave the picking area of the robot *i-esima* towards the containers.

$$
\begin{aligned}
r_1(k) &= 1 - floor\left(\frac{a(k)+1}{n}\right) - floor\left(\frac{a(k)+(p(k)-n)+1}{n}\right) \\
r_2(k+m) &= 1 - floor\left(\frac{a(k)+2}{n}\right) - floor\left(\frac{a(k)+(p(k)-n)+2}{n}\right) \\
r_i(k+(i-1)m) &= 1 - floor\left(\frac{a(k)+i}{n}\right) - floor\left(\frac{a(k)+(p(k)-n)+i}{n}\right) \\
r_n(k+(n-1)m) &= 1 - floor\left(\frac{a(k)+n}{n}\right) - floor\left(\frac{a(k)+(p(k)-n)+n}{n}\right)
\end{aligned} \tag{4.1}
$$

Equation 4.2 shows how $a(k)$ is defined with $a(0) = 1$

$$
a(k+1) = (a(k)+1) * \left(1 - floor\left(\frac{a(k)+1}{n}\right)\right) \tag{4.2}
$$

4.3 Control Strategies

Once it has been determined a control law that will not allow containers leave the system without full completion and, a job assignment technique has be decided, the only issue left to be solved is the working sequencing for the pickers. Several strategies are proposed and will be later compared among them with the aim of studying the strategy that will assure the desired throughput of the system.

4.3.1 Strategy 0: Basic Job Assigment

This strategy is the base of the study done for the layout of the plant proposed, see figure 3.4 . Robots will just pick and place one product unit during the time interval from k to $k+1$. The products to be picked for each robot correspond to those assigned to them selfs, following completely the job assignment proposed in section 4.2.

The robots will start picking unit of products as soon as they enter their respective working area. Due to the control action implemented, see section 4.1, in the case that any robot encounters there are not available spaces to place any unit of product within the containers inside its working area, the respective picker will not pick product units from the conveyor belt.

4.3.2 Strategy 1: Greedy robots

In this strategy the robots can pick and place more than one product unit during the time interval from$k$ to $k + 1$. As explained in section 3.2, empty containers distributed uniformly along the working length of the conveyor belt 2 are present at the time $k = 0$, thus, each robot will fill completely all the empty containers placed at the beginning of the process in their respective robot cells with all the products that enter their respective working areas. The robots will start picking products as soon as they enter their respective working area and, once the conveyor belt 2 stars moving, the robots will adopt the scheduling strategy proposed in section 4.3.1.

4.3.3 Strategy 2: Greedy robots with delayed action

The concept of having the robots picking and placing more than one product unit during the time interval from$k$ to $k + 1$ is kept. This strategy holds the same working principle as proposed in section 4.3.2 but with a small variant.

For this case, each robot will fill completely all the empty containers placed at the beginning of the process in their respective robot cells with any of the products entering their respective working area, but, all the robots will start filling the respective containers once last robot sees product units within its robot cell. Finally, once the firsts empty containers placed along the conveyor belt 2 has been filled, the robots will adopt the scheduling strategy proposed in section 4.3.1.

4.3.4 Strategy 3: Last robot greedy

Following the same concept than the exposed in section 4.3.3, the robots can pick and place more than one product unit during the time interval from k to $k+1$. On the other hand, this strategy claims that the last robot on the picking line will fill all the empty containers placed at the beginning of the process with all the product units entering its working area. The other robots will start to work once a new container enters their respective robot cell adopting the scheduling job proposed in section 4.3.1.

4.3.5 Strategy 4: Level of greediness constrained for all robots

Strategy 4 represents and extension of strategy 3, see section 4.3.4. The general concept is the same, each robot will fill completely all the empty containers placed at the beginning of the process in their respective robot cells with any type of product entering their respective working area, starting the filling action once the last robot sees product units within its robot cell.

However, the extension is given by the conditions that all the robots are constrained in the maximum number of product units that can picked at each time instant k and that only the last robot is in the capacity of filling totally the containers. The rest of the pickers just fill partially the containers located in their respective working areas. Once all the initial empty containers have been partially/fully filled with product units, the robots will adopt the scheduling strategy proposed in section 4.3.1.

4.3.6 Strategy 5: Level of greediness constrained for firsts robots and the last one free

Like strategy 4, this strategy represents an extension of the previous one. The strategy claims that each robot will fill the empty containers placed at the beginning of the process in their respective robot cells with any type of product entering their respective working area, starting the filling action once last robot sees product units within its robot cell.

The variation with respect to strategy 4 is that in this strategy all the robots but the last one are constrained in the maximum number of product units that can pick at each time. Meaning the last robot will cover all the spaces that are left empty for the other pickers. As before, once each robot have filled partially/totally the initial empty containers with product units, the robots will adopt the scheduling strategy proposed in section 4.3.1.

Chapter 5

Analysis and Results

This chapter contains an analysis of the six different scheduling strategies that were presented in chapter 4. The study of the behaviour of each strategy is evaluated on each level of the mathematical model derived in section 3.3. Furthermore, within this chapter, a comparison between the approaches used in this work. The analysis of each strategy is made by means of an algorithm developed in MATLAB®

5.1 Simulation Results

In order to operate, the algorithms developed in MATLAB® , one for each scheduling strategy and level of the mathematical model, require different data about the layout of the plant to simulate, such as, the number of robots used for the pick and place operation, number of product units entering the system and the number of containers that each robot can see inside its working area, both data for each discrete instant k, and, lastly, the number of instants that the user would like to run the simulation.

The input data is compiled and processed depending on the desired strategy and the respective algorithm will return as a result a set of plots that describe the dynamic behavior of the product stream and the hole stream for each robot cell respectively. For simulation purposes, two out of the three levels of the mathematical model presented in section 3.3.1 were studied. In particular, Level 1 and Level 2. These two approaches describe more accurately the specific internal dynamic of the system having into account both, the initial and the working conditions of the system.

5.1.1 Strategy 0: Basic Job Assignment

The description of input data of the different parameters that describe the layout to the plant to simulate will be presented using a base scenario with the values reported in table 5.1. Those values were chosen based on the mathematical model criteria and they will keep being the same for most of the simulations unless it is specified otherwise.

Parameter	**Value**
Number of robots	4
Number of product units entering the system at each instant	3
Number of boxes seen by each robot	3
Number of holes by container	3

Table 5.1: Layout of the plant. Base scenario parameters

5.1.1.1 Mathematical model level 1

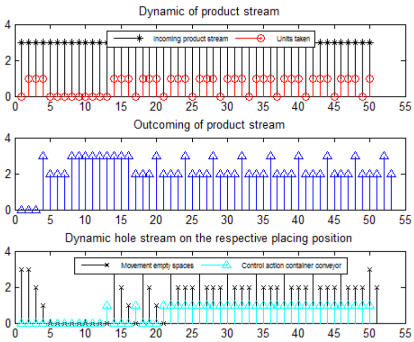
Figure 5.1 portrays the dynamical behavior of each robot cell described by the equations 3.6, 3.7, 3.8 and 3.9. For each robot cell three plots are generated. The firsts two, from top to bottom, describe the dynamic of the product stream at each robot cell and, the last plot, shows the dynamic of holes on the assigned placing position.

5.1.1.2 Mathematical model level 2

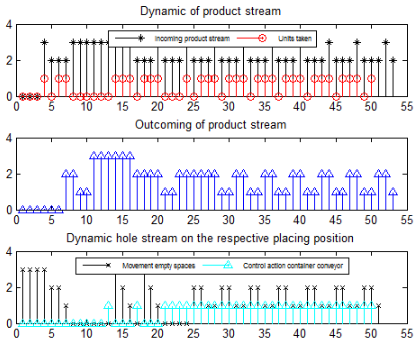
Figure 5.2 depicts both, the dynamical behavior of the product stream of each robot cell described by the equations 3.11, 3.12 and 3.13, and, the dynamical behavior of the hole stream described by the equations 3.14, 3.15 and 3.16. The same description applies for figure 5.3.

Both figures, figure 5.2 and figure 5.3, illustrate the behavior of the system when it is analyzed in the two different cases that are presented in section 3.3.3.

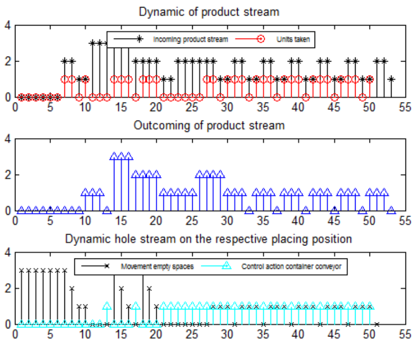
In each case, for each robot cell, six plots are generated. In particular, three correspond to the dynamic of the product stream at each robot cell sub-cell, and three for the dynamic of the hole stream at each robot cell sub-cell.



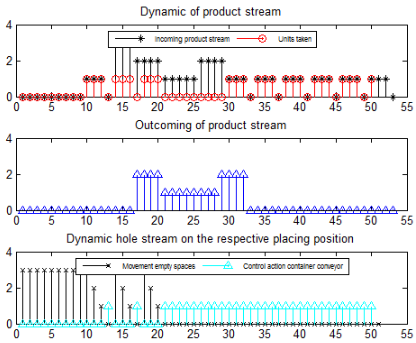
a) Robot cell 1



b) Robot cell 2



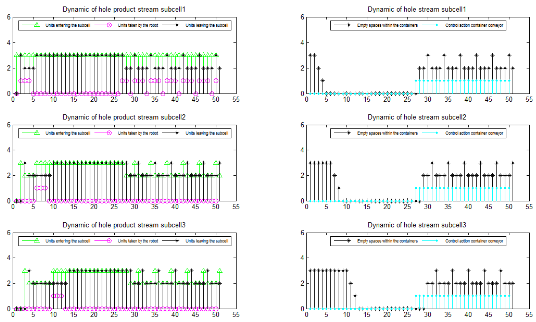
c) Robot cell 3



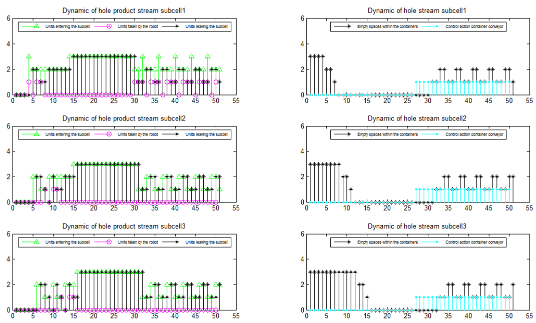
d) Robot cell 4

FIGURE 5.1: Dynamic behavior of scheduling strategy 0 simulated with mathematical model level 1

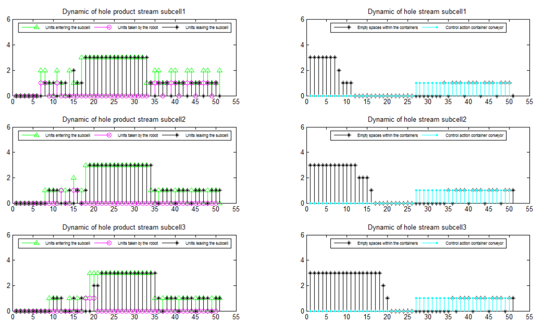
1. Case 1



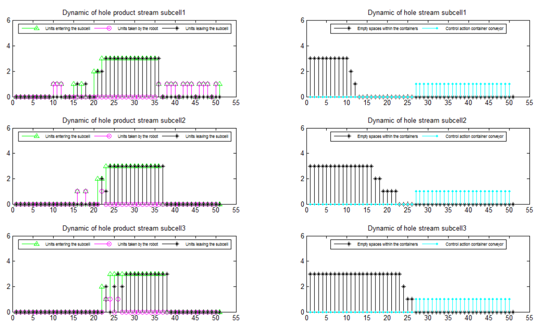
a) Robot cell 1



b) Robot cell 2



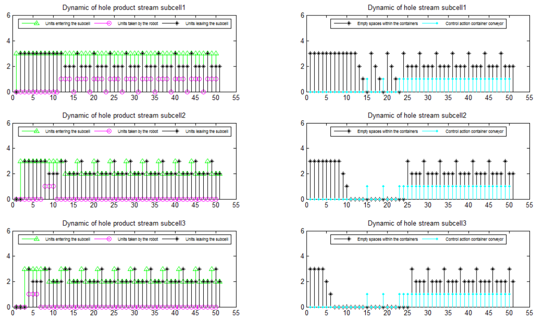
c) Robot cell 3



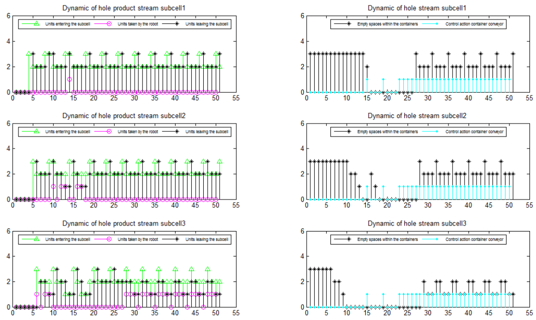
d) Robot cell 4

FIGURE 5.2: Dynamic behavior of scheduling strategy 0 simulated with mathematical model level 2 case 1

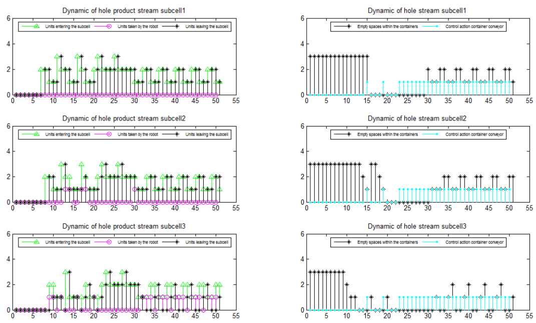
2. Case 2



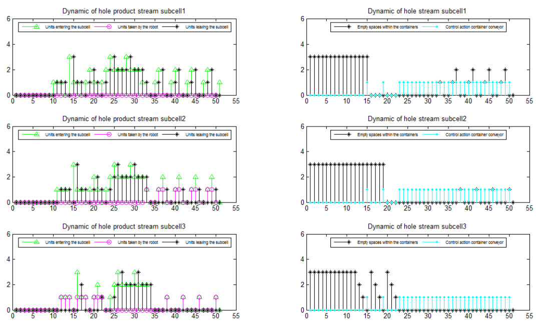
a) Robot cell 1



b) Robot cell 2



c) Robot cell 3



d) Robot cell 4

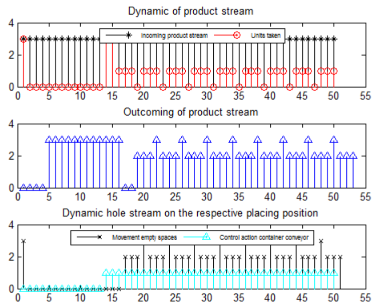
FIGURE 5.3: Dynamic behavior of scheduling strategy 0 simulated with mathematical model level 2 case 2

5.1.2 Strategy 1: Greedy robots

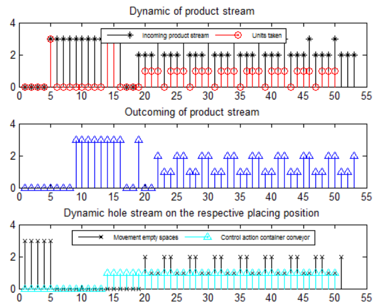
For the simulation of this strategy the different parameters that describe the layout to the plant to simulate are reported in table 5.1.

5.1.2.1 Mathematical model level 1

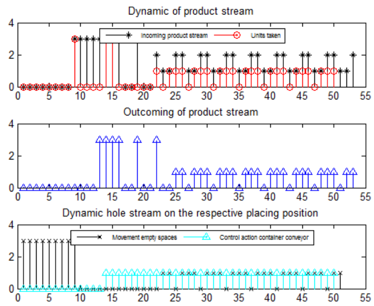
Figure 5.4 shows the dynamical behavior of each robot cell described by the equations 3.6, 3.7, 3.8 and 3.9 when implementing the scheduling strategy 1. The description of the plots is the same than the one given in section 5.1.1.1.



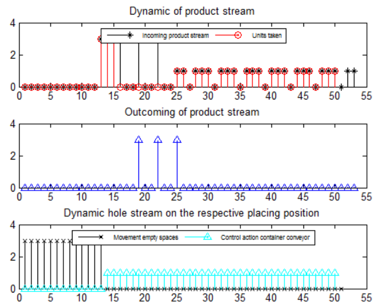
a) Robot cell 1



b) Robot cell 2



c) Robot cell 3



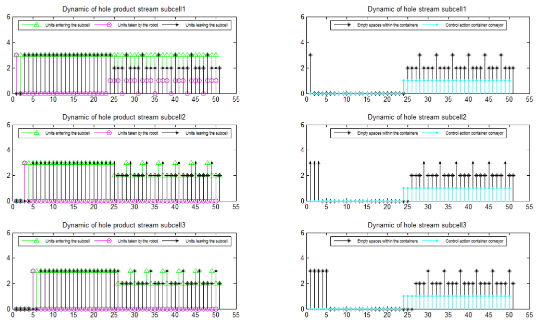
d) Robot cell 4

FIGURE 5.4: Dynamic behavior of scheduling strategy 1 simulated with mathematical model level 1

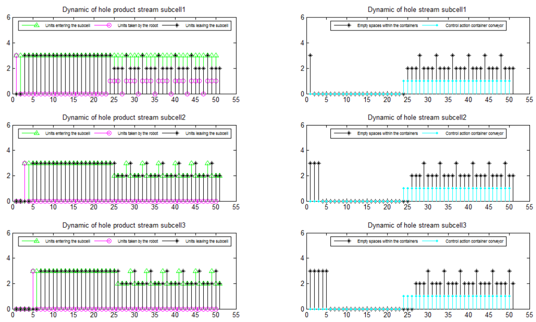
5.1.2.2 Mathematical model level 2

Figure 5.5 and figure 5.6 depict the dynamical behavior of both, the product stream and hole stream of each robot cell when the scheduling strategy 1 is applied to the system. The dynamical behavior of the system to simulate is represented by the equations 3.11, 3.12, 3.13, 3.14, 3.15 and 3.16. Figure 5.5 and figure 5.6, illustrate the output response of the system when it is analyzed within two different scenarios, as reported in section 3.3.3. For the description of the plots see section 5.1.1.2, last paragraph.

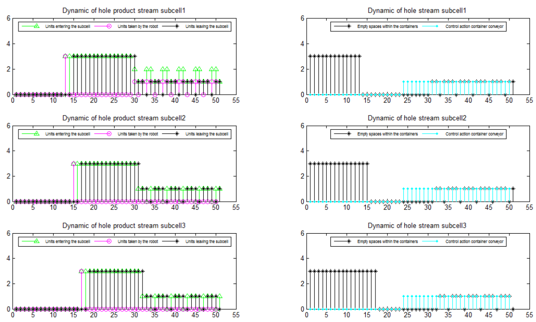
1. Case 1



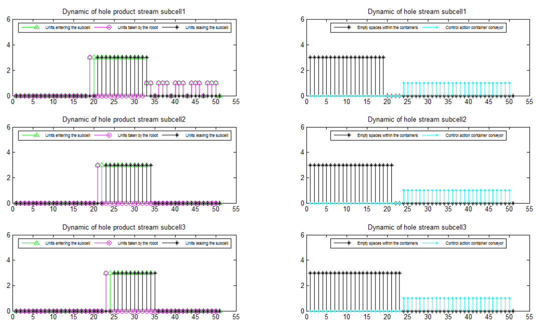
a) Robot cell 1



b) Robot cell 2



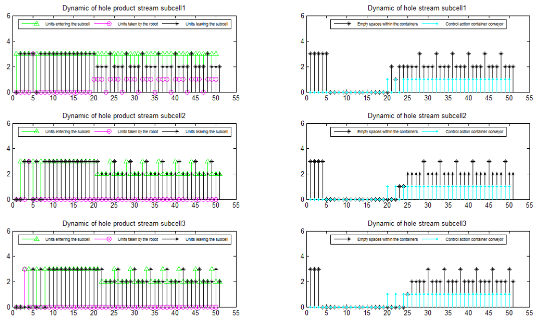
c) Robot cell 3



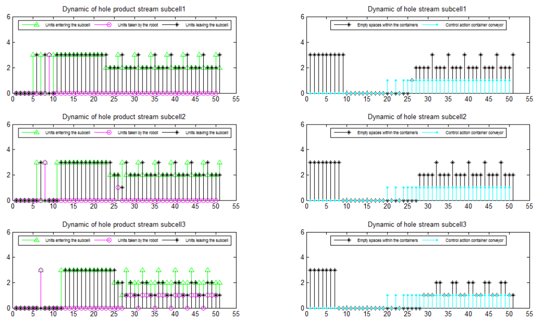
d) Robot cell 4

Figure 5.5: Dynamic behavior of scheduling strategy 1 simulated with mathematical model level 2 case 1

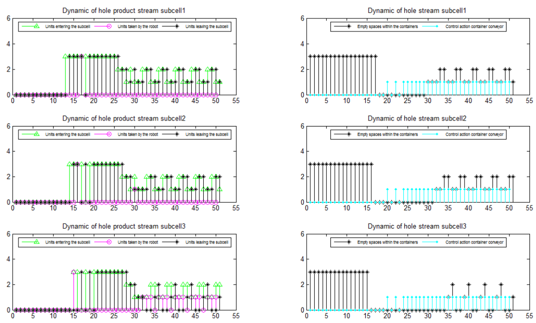
2. Case 2



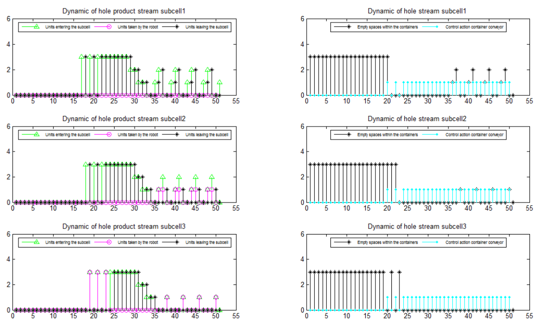
a) Robot cell 1



b) Robot cell 2



c) Robot cell 3



d) Robot cell 4

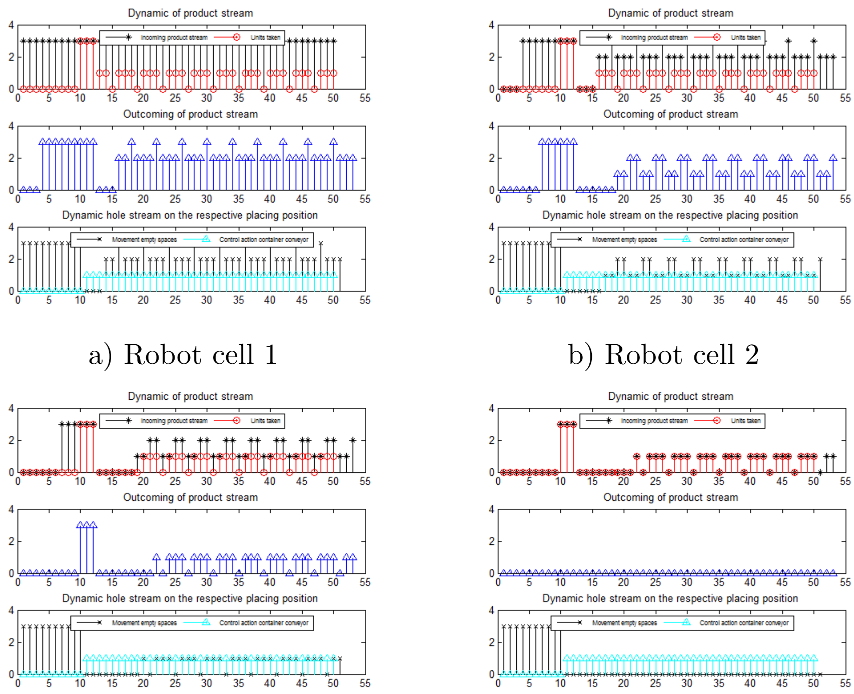
FIGURE 5.6: Dynamic behavior of scheduling strategy 1 simulated with mathematical model level 2 case 2

5.1.3 Strategy 2: Greedy robots with delayed action

The simulation of this strategy uses the values of the parameters reported in table 5.1 as done with the previous strategies.

5.1.3.1 Mathematical model level 1

The dynamical behavior of each robot cell described by the equations 3.6, 3.7, 3.8 and 3.9, when implementing the scheduling strategy 2, is shown in figure 5.7. The description of the plots is the same than in the previous strategies.



a) Robot cell 1 b) Robot cell 2

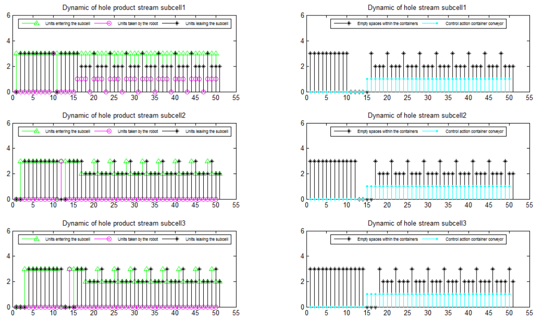
c) Robot cell 3 d) Robot cell 4

FIGURE 5.7: Dynamic behavior of scheduling strategy 2 simulated with mathematical model level 1

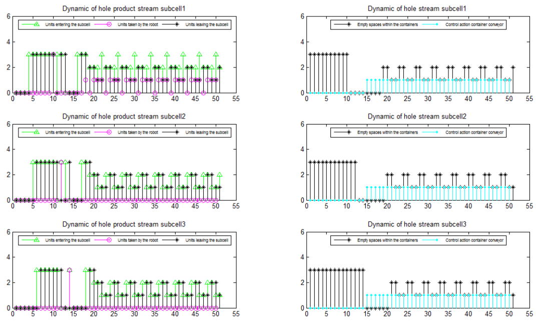
5.1.3.2 Mathematical model level 2

Figure 5.8 and figure 5.9 illustrate the dynamical behavior of both, the product stream and hole stream of each robot cell when the scheduling strategy 2 is applied to the system. Equations 3.11, 3.12, 3.13, 3.14, 3.15 and 3.16 represent the dynamical behavior of the system to simulate. As done in the previous strategies, there are two study cases and the description of the plots keep being the same as exposed in previous sections.

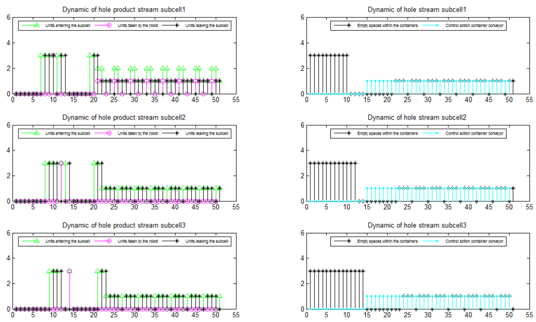
1. Case 1



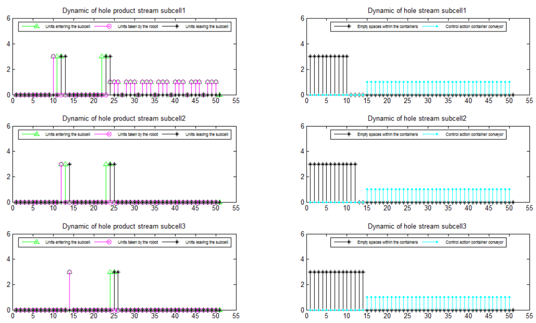
a) Robot cell 1



b) Robot cell 2



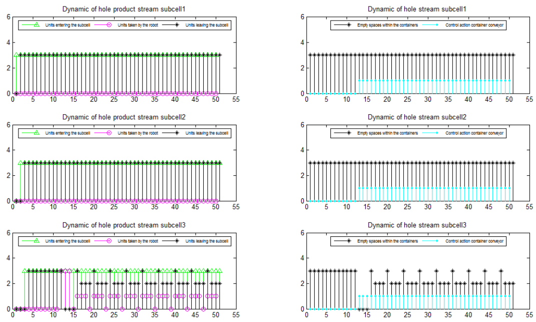
c) Robot cell 3



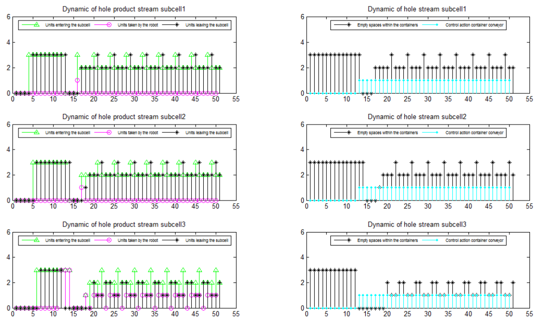
d) Robot cell 4

FIGURE 5.8: Dynamic behavior of scheduling strategy 2 simulated with mathematical model level 2 case 1

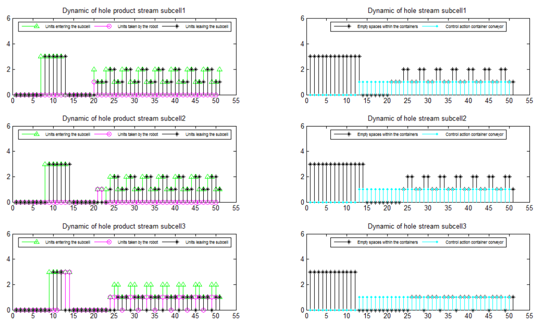
2. Case 2



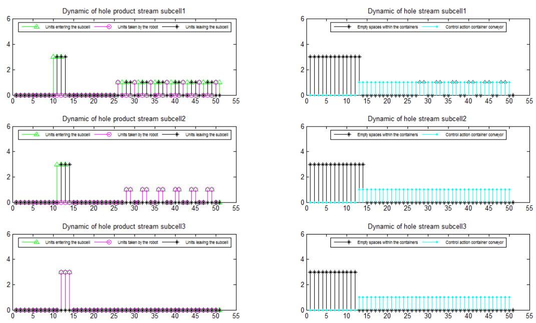
a) Robot cell 1



b) Robot cell 2



c) Robot cell 3



d) Robot cell 4

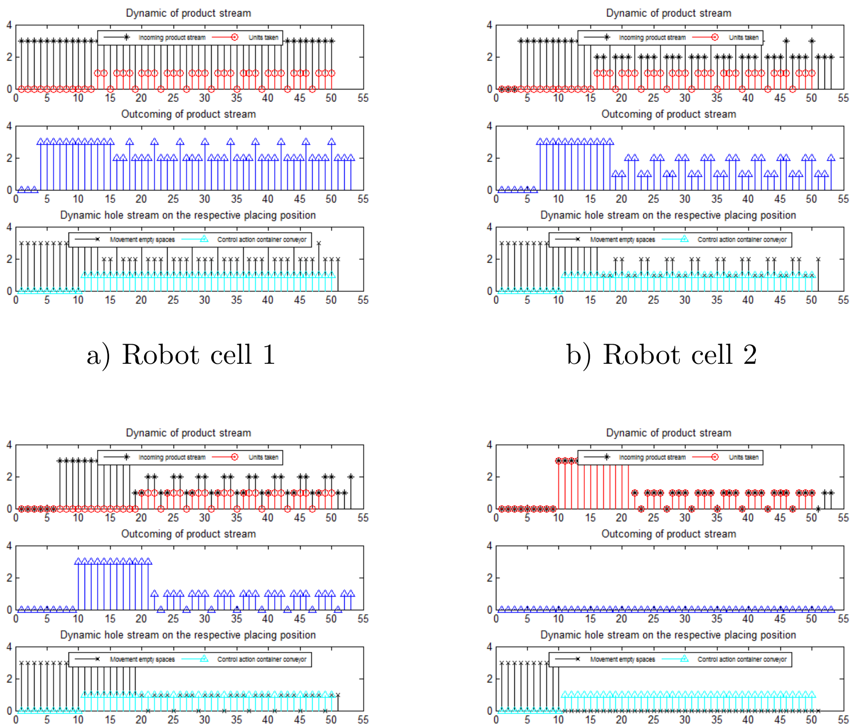
FIGURE 5.9: Dynamic behavior of scheduling strategy 2 simulated with mathematical model level 2 case 2

5.1.4 Strategy 3: Last robot Greedy

The input data that define the layout of the plant for the simulation of this strategy can be seen in table 5.1.

5.1.4.1 Mathematical model level 1

The dynamical behavior of each robot cell described by the equations 3.6, 3.7, 3.8 and 3.9, when implementing the scheduling strategy 3, is shown in figure 5.10. The description of the plots is the same than in the previous strategies.



a) Robot cell 1 b) Robot cell 2

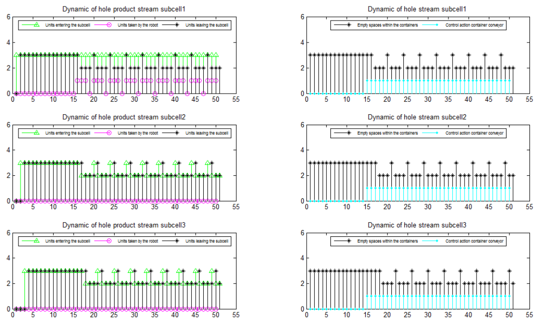
c) Robot cell 3 d) Robot cell 4

FIGURE 5.10: Dynamic behavior of scheduling strategy 3 simulated with mathematical model level 1

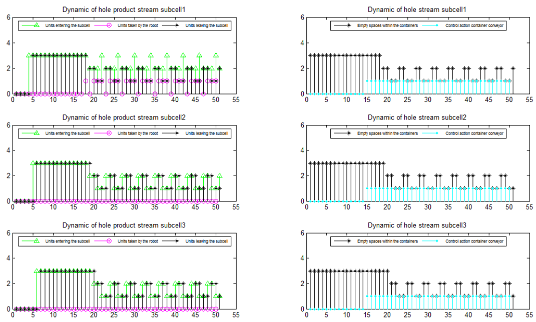
5.1.4.2 Mathematical model level 2

Figure 5.11 and figure 5.12 describe the dynamical behavior of both, the product stream and hole stream, of each robot cell when the scheduling strategy 3 is analyzed with the model of the plant defined by the equations 3.11, 3.12, 3.13, 3.14, 3.15 and 3.16. Each figure corresponds to a different study case, see section 3.3.3 for a detail description of each case. The description of the figures is the same as exposed in previous sections.

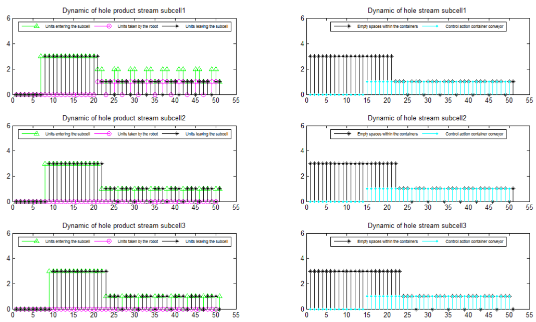
1. Case 1



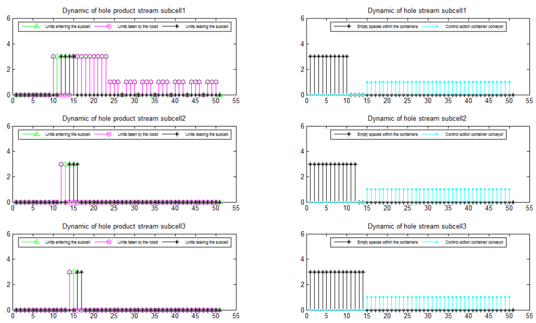
a) Robot cell 1



b) Robot cell 2



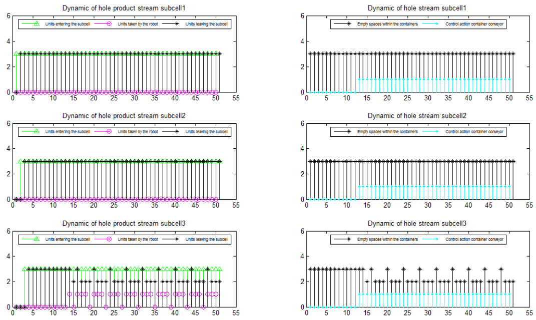
c) Robot cell 3



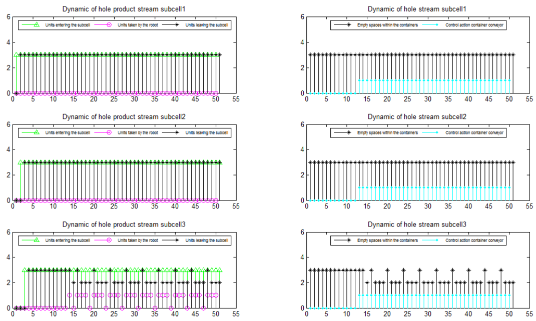
d) Robot cell 4

FIGURE 5.11: Dynamic behavior of scheduling strategy 3 simulated with mathematical model level 2 case 1

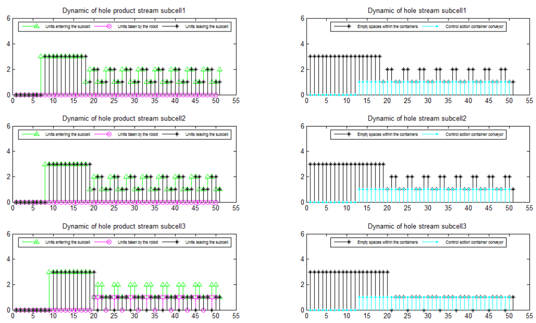
2. Case 2



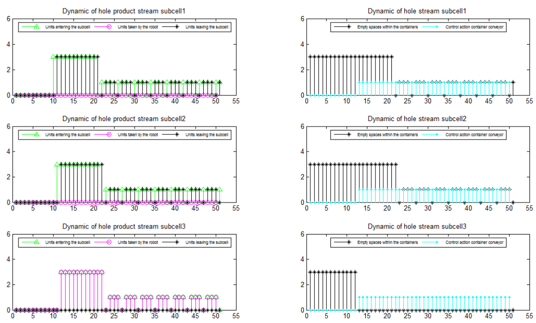
a) Robot cell 1



b) Robot cell 2



c) Robot cell 3



d) Robot cell 4

FIGURE 5.12: Dynamic behavior of scheduling strategy 3 simulated with mathematical model level 2 case 2

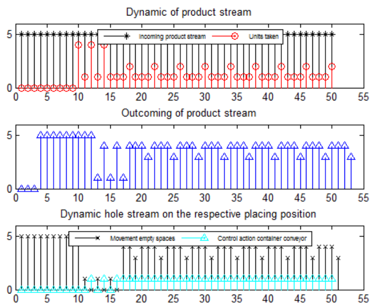
5.1.5 Strategy 4: Level of greediness constrained for all robots

The input data of the different parameters that describe the layout to the plant is the same as reported in table 5.1. In addition, a new parameter, $r_i(k)$, is included within this data and it describes the value of the constrained imposed to each robot according to the main concept of the strategy.

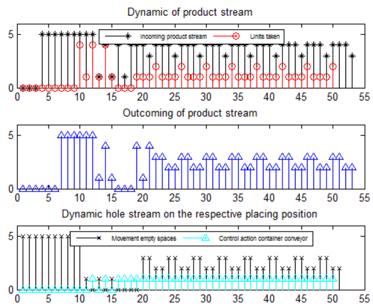
5.1.5.1 Mathematical model level 1

This scheduling strategy is run on the model of the plant given by the equations 3.6, 3.7, 3.8 and 3.9. Figures 5.13, 5.14 and 5.15 describe the internal dynamic behavior of the product and hole stream for each robot cell under different working conditions.

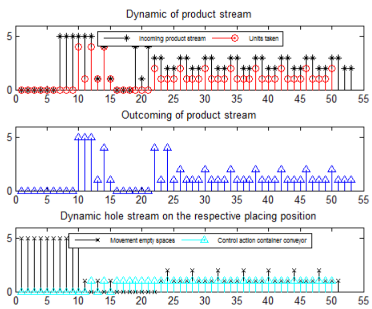
- $r_i(k)$ constrained to 4



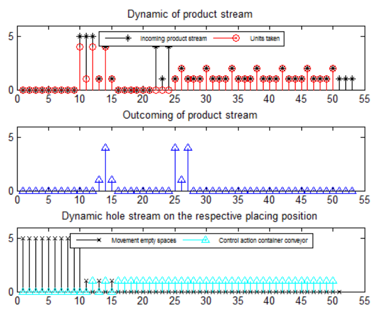
a) Robot cell 1



b) Robot cell 2



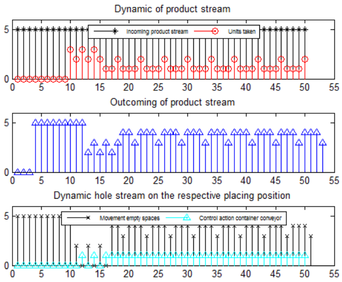
c) Robot cell 3



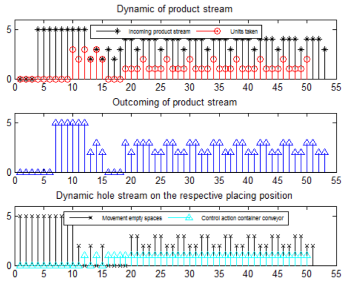
d) Robot cell 4

FIGURE 5.13: Dynamic behavior of scheduling strategy 4 simulated with mathematical model level 1 and $r_i(k)$ constrained to 4

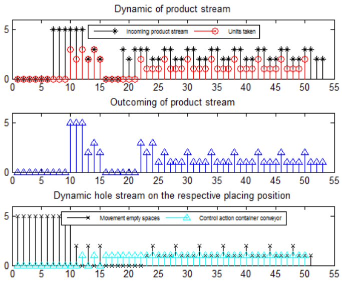
- $r_i(k)$ constrained to 3



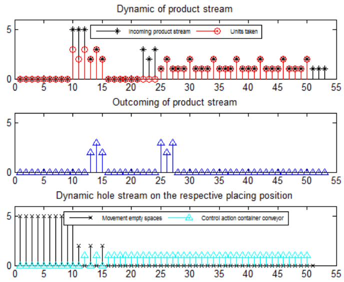
a) Robot cell 1



b) Robot cell 2



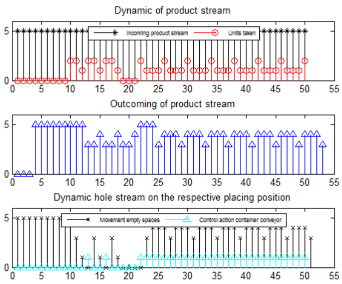
c) Robot cell 3



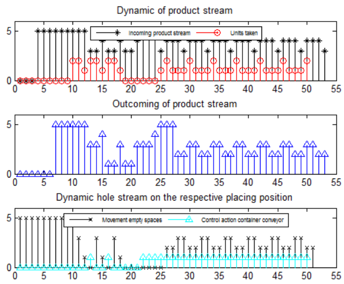
d) Robot cell 4

FIGURE 5.14: Dynamic behavior of scheduling strategy 4 simulated with mathematical model level 1 and $r_i(k)$ constrained to 3

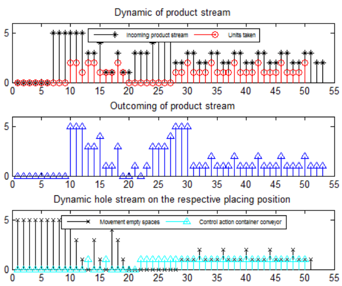
- $r_i(k)$ constrained to 2



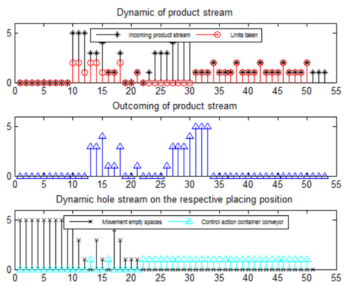
a) Robot cell 1



b) Robot cell 2



c) Robot cell 3



d) Robot cell 4

FIGURE 5.15: Dynamic behavior of scheduling strategy 4 simulated with mathematical model level 1 and $r_i(k)$ constrained to 2

5.1.5.2 Mathematical model level 2

The current scheduling strategy is simulated on the model of the plant defined by the equations 3.11, 3.12, 3.13, 3.14, 3.15 and 3.16. Figures 5.16, 5.17 and 5.18 describe the internal dynamic behavior of the product and hole stream for each robot cell under different working conditions.

1. Case 1

   - $r_i(k)$ constrained to 4

a) Robot cell 1

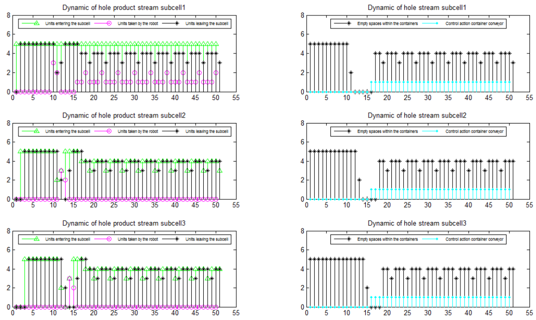
b) Robot cell 2

c) Robot cell 3

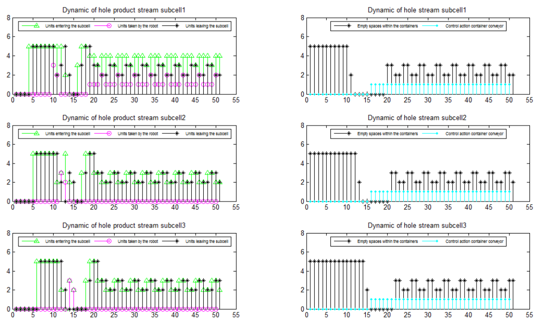
d) Robot cell 4

FIGURE 5.16: Dynamic behavior of scheduling strategy 4 simulated with mathematical model level 2 case 1 and $r_i(k)$ constrained to 4

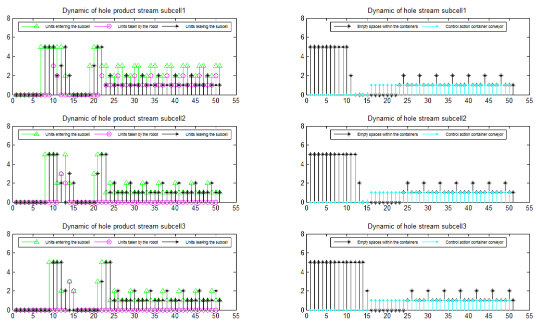
- $r_i(k)$ constrained to 3



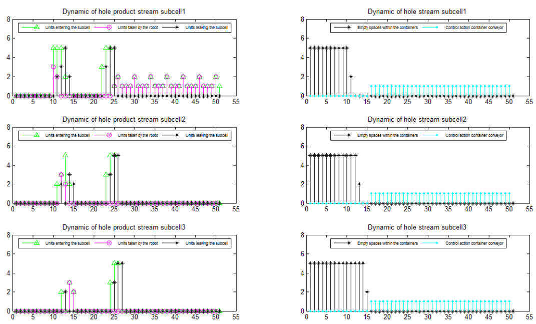
a) Robot cell 1



b) Robotcell 2



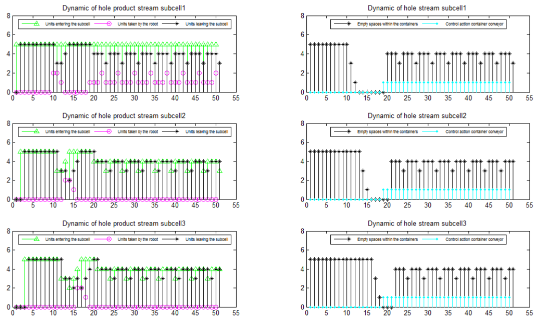
c) Robot cell 3



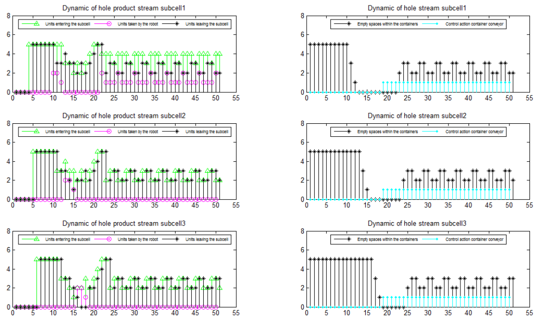
d) Robot cell 4

FIGURE 5.17: Dynamic behavior of scheduling strategy 4 simulated with mathematical model level 2 case 1 and $r_i(k)$ constrained to 3

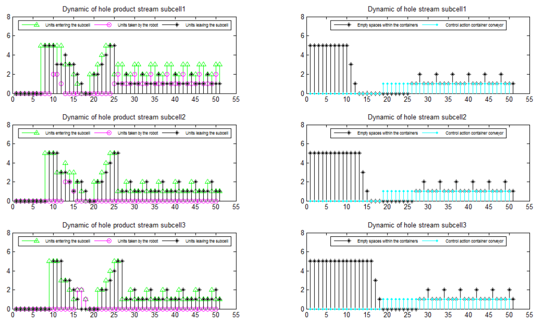
- $r_i(k)$ constrained to 2



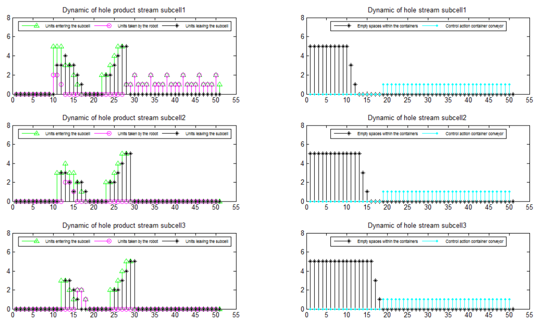
a) Robot cell 1



b) Robot cell 2



c) Robot cell 3

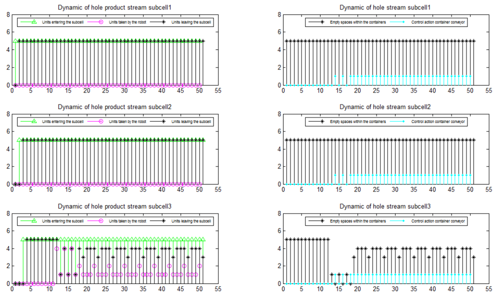


d) Robot cell 4

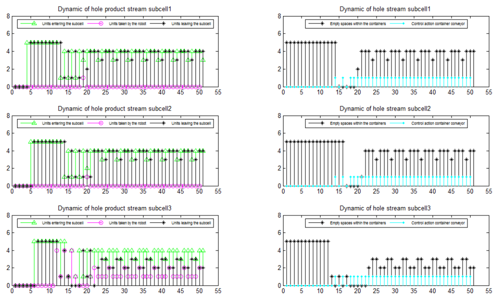
FIGURE 5.18: Dynamic behavior of scheduling strategy 4 simulated with mathematical model level 2 case 1 and $r_i(k)$ constrained to 2

2. Case 2

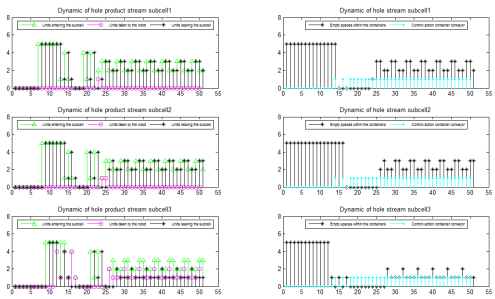
   - $r_i(k)$ constrained to 4



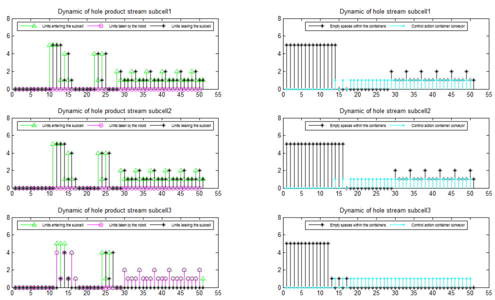
a) Robot cell 1



b) Robot cell 2



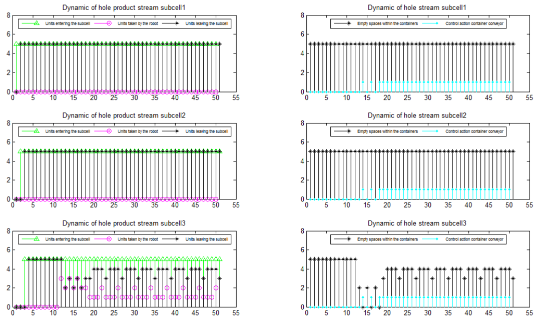
c) Robot cell 3



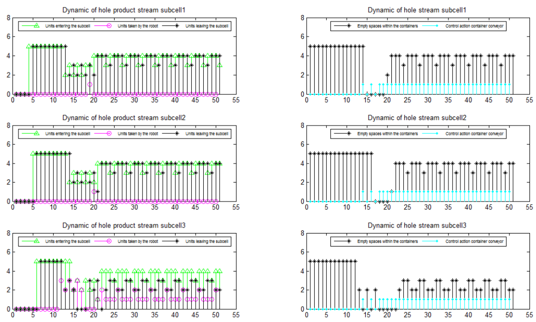
d) Robot cell 4

FIGURE 5.19: Dynamic behavior of scheduling strategy 4 simulated with mathematical model level 2 case 2 and $r_i(k)$ constrained to 4

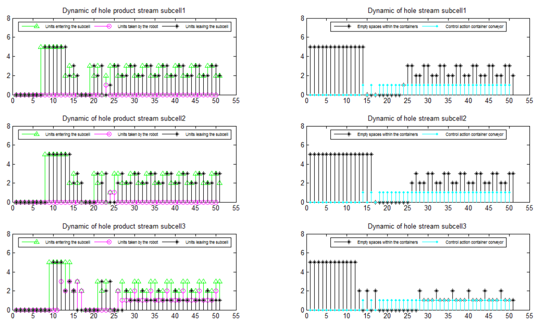
- $r_i(k)$ constrained to 3



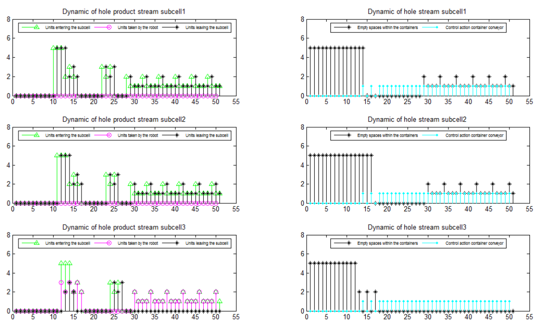
a) Robot cell 1



b) Robot cell 2



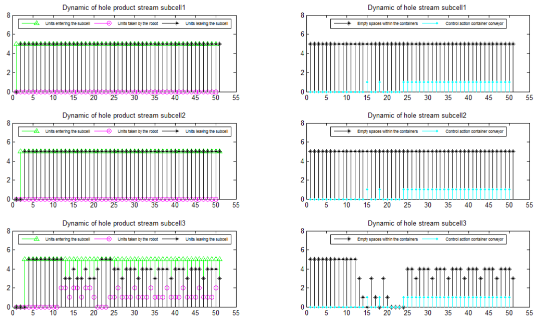
c) Robot cell 3



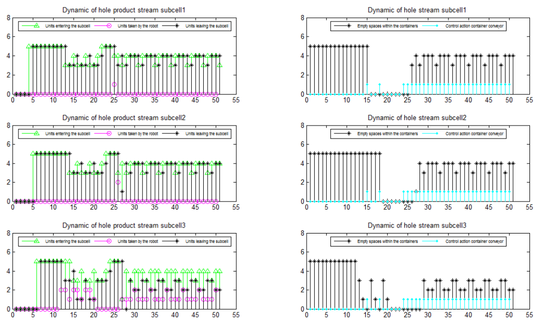
d) Robot cell 4

Figure 5.20: Dynamic behavior of scheduling strategy 4 simulated with mathematical model level 2 case 2 and $r_i(k)$ constrained to 3

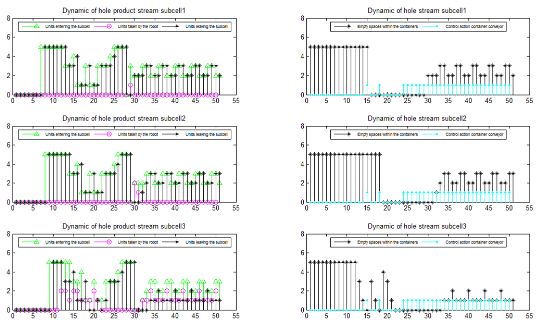
- $r_i(k)$ constrained to 2



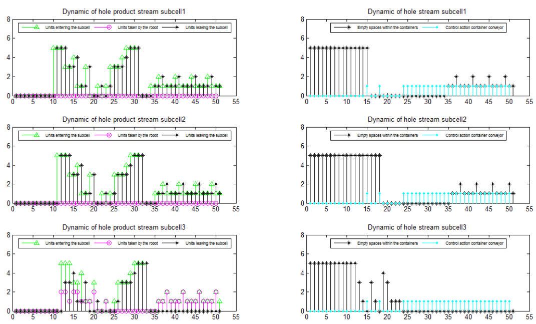
a) Robot cell 1



b) Robot cell 2



c) Robot cell 3



d) Robot cell 4

FIGURE 5.21: Dynamic behavior of scheduling strategy 4 simulated with mathematical model level 2 case 2 and $r_i(k)$ constrained to 2

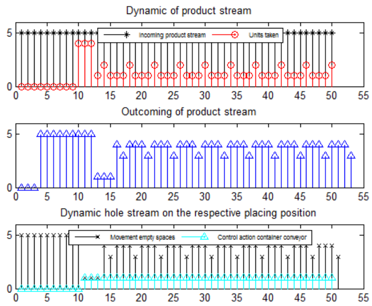
5.1.6 Strategy 5: Level of greediness constrained for firsts robots and the last one free

The simulation of this strategy holds the same input data than strategy 4, see section 5.1.5.

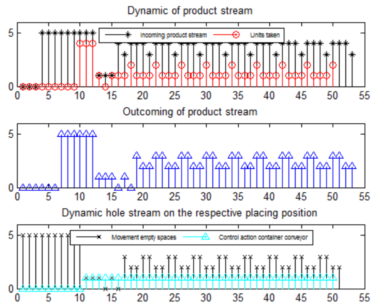
5.1.6.1 Mathematical model level 1

Equations 3.6, 3.7, 3.8 and 3.9 define the mathematical model used to simulate the current scheduling strategy. Figures 5.22 and 5.23 illustrate the internal dynamic behavior of the product and hole stream for each robot cell under different working conditions.

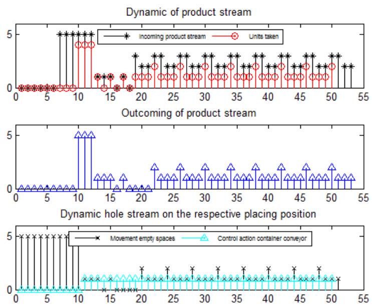
- $r_i(k)$ constrained to 4



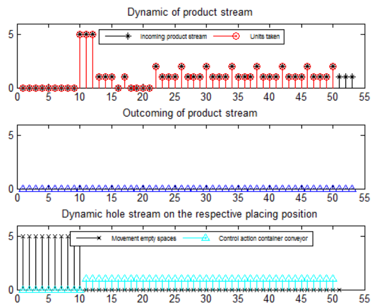
a) Robot cell 1



b) Robot cell 2



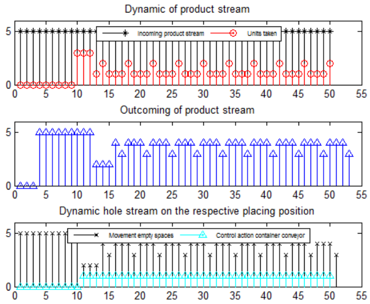
c) Robot cell 3



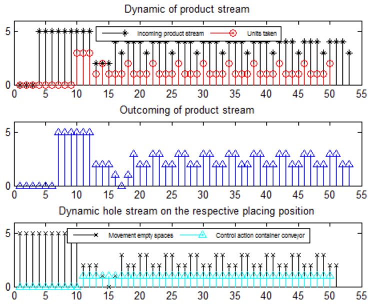
d) Robot cell 4

FIGURE 5.22: Dynamic behavior of scheduling strategy 5 simulated with mathematical model level 1 and $r_i(k)$ constrained to 4

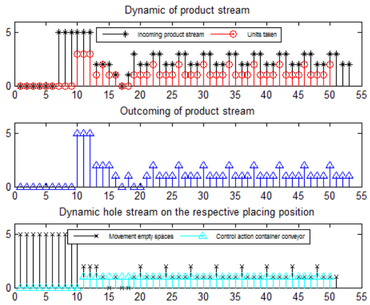
- $r_i(k)$ constrained to 3



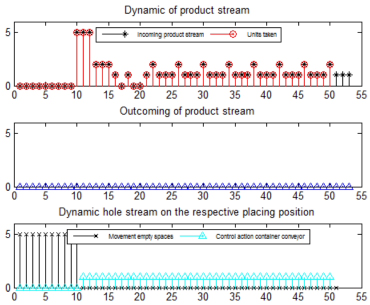
a) Robot cell 1



b) Robot cell 2



c) Robot cell 3



d) Robot cell 4

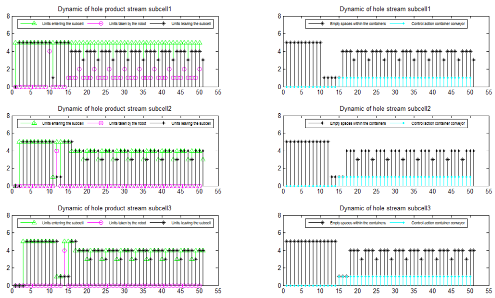
FIGURE 5.23: Dynamic behavior of scheduling strategy 5 simulated with mathematical model level 1 and $r_i(k)$ constrained to 3

5.1.6.2 Mathematical model level 2

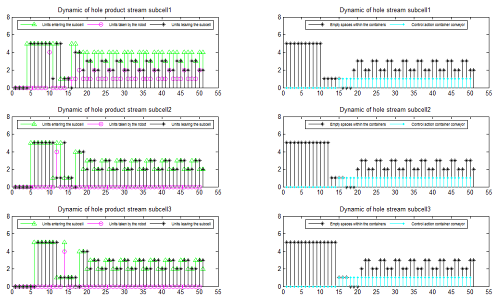
Equations 3.11, 3.12, 3.13, 3.14, 3.15 and 3.16 describe the mathematical model implemented for the simulation of the current scheduing strategy. Figures 5.24 and 5.25 show the internal dynamic behavior of the product and hole stream for each robot cell under different working conditions.

1. Case 1

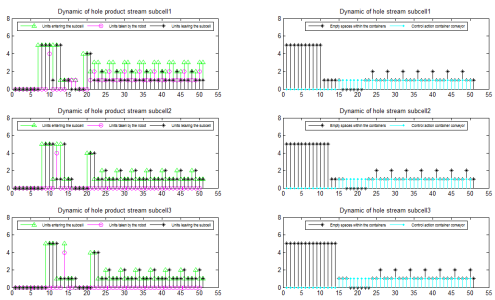
   - $r_i(k)$ constrained to 4



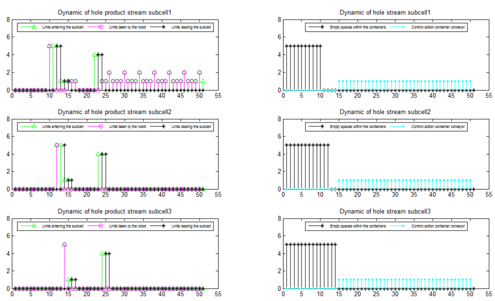
a) Robot cell 1



b) Robot cell 2



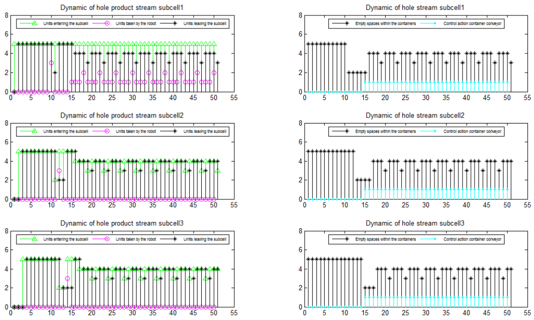
c) Robot cell 3



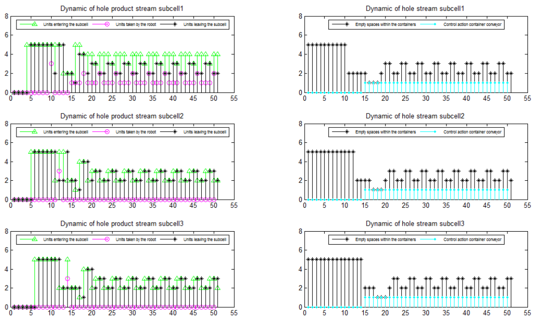
d) Robot cell 4

FIGURE 5.24: Dynamic behavior of scheduling strategy 5 simulated with mathematical model level 2 case 1 and $r_i(k)$ constrained to 4

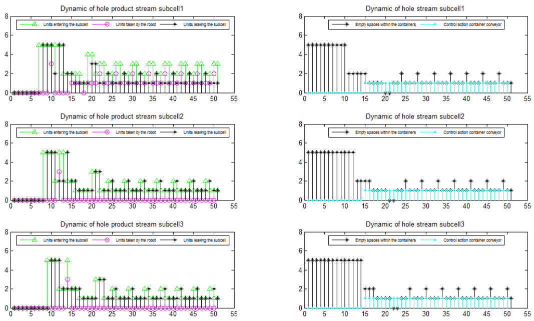
- $r_i(k)$ constrained to 3



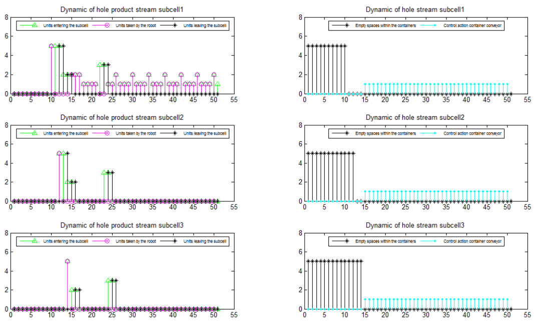
a) Robot cell 1



b) Robot cell 2



c) Robot cell 3

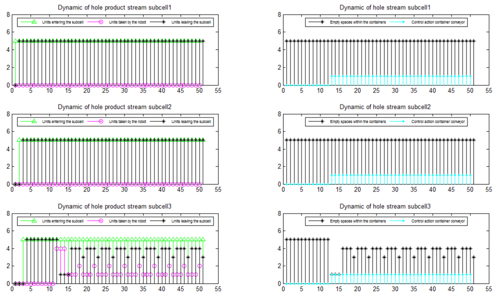


d) Robot cell 4

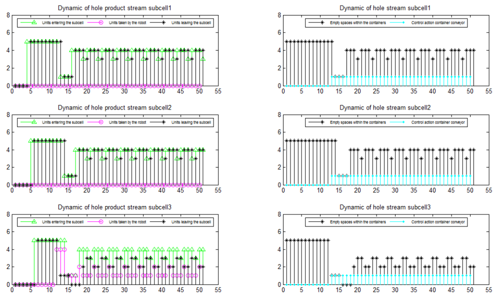
FIGURE 5.25: Dynamic behavior of scheduling strategy 5 simulated with mathematical model level 2 case 1 and $r_i(k)$ constrained to 3

2. Case 2

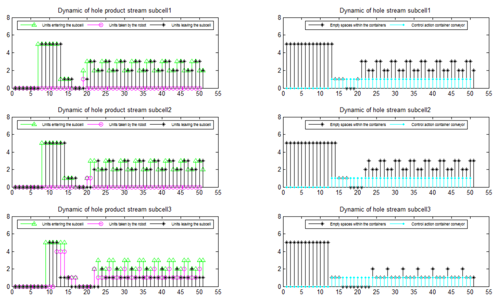
- $r_i(k)$ constrained to 4



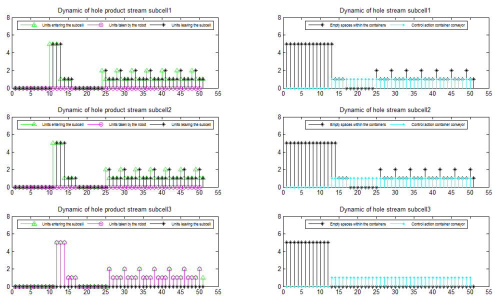
a) Robot cell 1



b) Robot cell 2



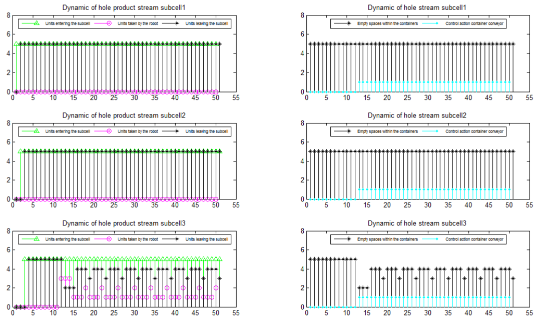
c) Robot cell 3



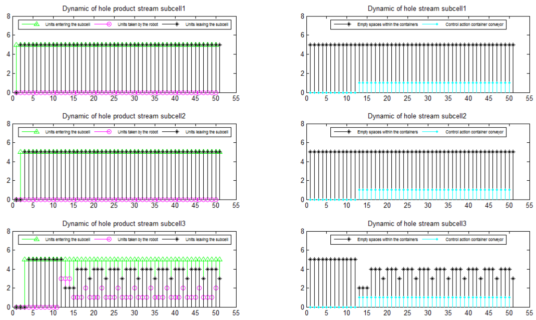
d) Robot cell 4

FIGURE 5.26: Dynamic behavior of scheduling strategy 5 simulated with mathematical model level 2 case 2 and $r_i(k)$ constrained to 4

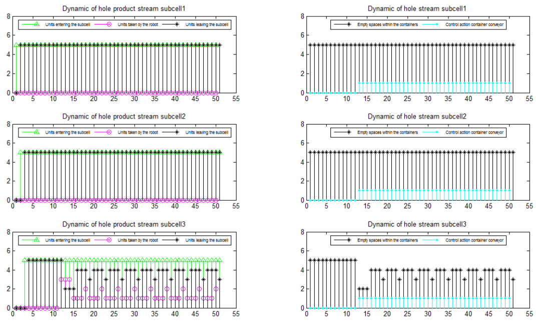
- $r_i(k)$ constrained to 3



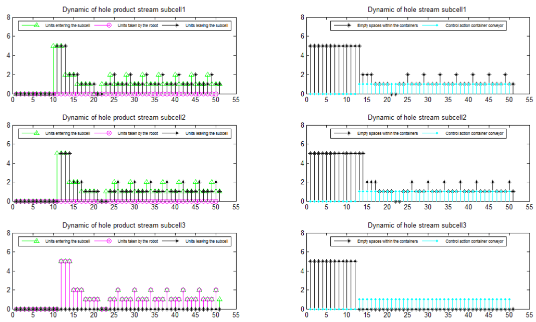
a) Robot cell 1



b) Robot cell 2



c) Robot cell 3



d) Robot cell 4

FIGURE 5.27: Dynamic behavior of scheduling strategy 5 simulated with mathematical model level 2 case 2 and $r_i(k)$ constrained to 3

5.2 Analysis of the Simulation Results

This section reports the respective analysis of the results exposed in the previous section for each scheduling strategy.

5.2.1 Strategy 0: Basic Job Assignment

From figures 5.1, 5.2 and 5.3 it can be seen that the implementation of a scheduling rule that is based only in a balancing workload among the pickers does not allow to obtain a satisfactory picking efficiency on the overall system regardless the mathematical model of the plant where the scheduling strategy is tested. This fact is mostly due to the initial conditions of the system, which are the same for both mathematical models.

The layout defined for the plant, specifically, the empty containers placed along the conveyor belt 2 at the beginning of the process, produces that the two conveyor belts, product and container conveyor belts, do not work at the same peace with respect to each other during the firsts time instants. This period will be called, from now on for analysis purposes , as *inital transient state (ITS)*.

During *ITS* the conveyor belt 1 is feeding the system continuously with new product units, however, conveyor belt 2 does not start feeding the system with new empty containers until the last container placed in the working area of the last picker is totally full. As consequence, there will be instants in which even though there are product units to be picked by the pickers in their respective picking areas, there are not available empty holes to place them, hence, the pickers will evidence dead working times, specifically, they will be forced to wait until new containers arrive to the respective placing areas. At the moment that new containers are introduced into the system, there will be a balance between product units and holes entering the system at each time instant k. Therefore, all the product units entering the system will be picked by the robots and all the containers will leave the plant completely full.

Table 5.2 compiles the output data obtained after a simulation of the current scheduling strategy was carried out on the mathematical models level 1 and 2. Analyzing this data, it can be seen that changing the system specifications, specifically, the picking and placing area within the working area of each robot, affects greatly the overall response of the system. The difference on the system output between the both models is mainly due to the fact that the level 2 model considers more than one placing area inside each robot cell. As a result, there will be more containers arriving the last sub cell of the system already full, thus, more product units on the conveyor belt that are not picked.

In addition, making a comparison between the two different scenarios when simulating model level 2, it is found that the system evidences a better performance when the pick and place action starts from the last sub-cell of each robot cell.

Level Model	**k when ITS finishes**	**Incoming product units**	**Lost product units**
Model level 1	30		18
Model level 2: case 1	33	150	27
Model level 2: case 2	34		24

Table 5.2: Results simulation scheduling strategy 0

5.2.2 Strategy 1: Greedy robots

From previous section it was deducted that the main problem of the system is encountered during the ITS of the process. In order to avoid the lost of product during this state, strategy one suggests that each robot will fill, as fast as they can, the empty containers placed in their working areas. This behavior is shown in figures 5.4, 5.5 and 5.6. Table 5.3 collects the output data obtained after the simulation of the current scheduling strategy on the mathematical models level 1 and 2.

Level Model	**k when ITS finishes**	**Incoming product units**	**Lost product units**
Model level 1	26		9
Model level 2: case 1	36	150	33
Model level 2: case 2	36		27

Table 5.3: Results simulation scheduling strategy 1

The study of the data contained in 5.2 and 5.3 leads to evidence that the performance of the system does not improve when implementing the current strategy, in particular, when the model level 2 defines the system. The total number of pieces that are left over the product conveyor belt, at each case, corresponds to the sum of the product units present in the last sub cell of the working area of the last picker when containers that are filled by the firsts robots reach the same sub cell. In other words, in order to avoid the lost of product units, when a full container reaches the last sub cell of the last picker, at the same time instant, there should not be any product units present in the same position over the conveyor belt 1.

5.2.3 Strategy 2: Greedy robots with delayed action

Having identified the main problem that causes lost of product on the system during the ITS, this strategy proposes a scheduling rule that will enhance the robots to reach the higher picking efficiency possible, therefore, the system will not evidence lost of product at any state of the process. Figures 5.7, 5.8 and 5.9 illustrate how the current scheduling strategy improves the output response of the systems when simulated in both mathematical models of the system, level 1 and level 2. The respective output data for each model is shown in table 5.4.

Level Model	k when ITS finishes	Incoming product units	Lost product units
Model level 1	0		0
Model level 2: case 1	26	150	6
Model level 2: case 2	0		0

Table 5.4: Results simulation scheduling strategy 2

Analyzing the data reported in table 5.4, as well as, the behavior evidenced in figures 5.8 and 5.9 it is possible to understand the difference between the two different study cases of the level 2 model. Even though the formulas that describe the system are the same for both study cases, the fact that the pick and place operation starts on different areas inside the same robot cells affects considerably the output response of the system.

As stated in previously sections, the main problem, that causes lost of product, is the containers that are fill to completion before they reach the last sub-cell of the system causing dead times in the action of the pickers. For this specific case, when the pick and place operation starts from the first sub cell inside each robot cell, namely case 1, each robot will evidence two dead times, as a result, there will be lost of product units at the end of the ITS that correspond to the pieces that were not picked from the conveyor at those time instants.

5.2.4 Strategy 3: Last robot greedy

Strategy 2 already offers a scheduling rule that satisfies the two main working conditions of a secondary packaging system, regardless the specifications given to the layout of the plant of this work. The aim of this strategy is to offer a new scheduling rule that will ensure the plant to work properly, in other words, it will give the same results that strategy 2 gives. However, once the new scheduling rule is implemented on the system, from figures 5.10, 5.11 and 5.12 it can be deduced that it does not enhance a balanced work load in all the robots. For this specific case, the last picker will carry a work load higher than the other robots.

As it was previously stated, the output data obtained, after the simulation of the current strategy was run, coincides with the ones obtained from the strategy 2, see table 5.4 for more detail. In the same way the analysis of the results is the same as exposed in section 5.2.3 last paragraph.

5.2.5 Strategy 4: Level of greediness constrained for all robots

Having identified a scheduling rule that allows all the pickers of the plant to work properly, with the best picking efficiency, the next step is to start playing with the parameters in order to identify the limitations of it. With this purpose on mind, strategy 4 offers an extension of the original scheduling strategy described in section 4.3.3 that will allow to analyze the conditions under which the strategy is considered optimal.

For the simulation of this strategy a new parameter was introduced. This parameter, which can vary as the user wants, denotes a restriction in the greediness of each robot. Analyzing just the first case, where the robots are constrained to pick maximum 4 product units at each time instant, see figures 5.13, 5.16 and 5.19, it is seen that there is again lost of product units during the ITS, however the lost of product is not as bigger as the ones encountered in strategies 0 and 1. Consult table 5.5 to see all the output data obtained for different values of the constrained imposed on the picking action of the robots. From data reported in 5.5 it is evident that the lower the value of the constrain, the bigger the number of lost product units. This fact is directly related to the number of dead times each robot evidences.

Level Model	**Constrain**	**k when ITS finishes**	**Incoming product units**	**Lost product units**
Model level 1	4	28	250	15
	3	28		15
	2	34		45
Model level 2: case 1	4	28	250	15
	3	28		15
	2	31		30
Model level 2: case 2	4	28	250	15
	3	28		15
	2	31		45

Table 5.5: Results simulation scheduling strategy 4

5.2.6 Strategy 5: Level of greediness constrained for firsts robots and the last one free

Strategy 5 can be seen as a combination between strategy 4 and strategy 3. In section 5.2.4 was exposed that strategy 3 makes the plant to work perfectly as it is desired, however, it has the disadvantage of overload the last picker of the system. From the simulation of the strategy 4, see figures 5.13, 5.16 and 5.19, it can be observed that the lost of product is mainly due to those instants in which pickers but last one fill to completion the empty containers placed at the beginning of the process. Taking advantage of the fact that the last picker can be the one in charge in finishing the filling of the containers placed on the working areas of the other pickers, then, those instants where product units are not taken for the first pickers are avoid. At the end, there is not lost of product units during the ITS. Figures 5.22 and 5.26 show this behavior. table 5.5 compiles the output data obtained for different values of the constrained imposed on the picking action of the robots.

Level Model	Constrain	k when ITS finishes	Incoming product units	Lost product units
Model level 1	4 3	0 0	250	0 0
Model level 2: case 1	4 3	31 31	250	15 15
Model level 2: case 2	4 3	0 0	250	0 0

Table 5.6: Results simulation scheduling strategy 5

5.2.7 Summary

Studying the data collected in tables 5.3, 5.4, 5.5, 5.6, it can be concluded the follow:

- Strategy 0 proved to be optimal just when the system is already in a stable state. That state is achieved once there is a balance between the incoming product and the incoming holes.
- The main reason that causes lost of product units during the ITS is the dead times produced in the pick and place action of all robots. The implementation of strategies 1 and 4 produces that all the robots evidence such dead times during the initial state of their working time due to the unbalanced number of product units with respect to the number of holes present in each robot cell.

- Regardless the scheduling strategy implemented, the option of starting the filling of the containers of each robot cell from the first sub cell causes dead times in the pick and place operation of each picker.

- Strategies 2, 3 and 5 ensure the system will work properly when the filling of the containers of each robot cell starts from their last sub cell, fulfilling the main two working conditions of a pick and place packaging system, see section 3.2. Under those cases, the robots will achieve the best picking efficiency with respect to the total amount of product stream that entered the system.

Chapter 6

Conclusions and Future Work

The description of a mathematical model for a co-current pick and place packaging system, in terms of dynamic equations, becomes a powerful tool that helps not only to gain understanding about the dynamic of the physical system and the variables involved on it, but also, it allows to simulate and, consequently, to analyze the behavior of the system under specific working conditions.

The use of differential equations when developing a mathematical model for a pick and place packaging system, permits to mimic the complex interaction between the sub-processes that comprise the system, in particular the pick and place activities performed by the pickers at each time instant.

Three different approaches are proposed to describe the dynamic of a pick and place packaging system and each one of them describes it considering a different level of complexity. The fact of having more than one mathematical model for the same system, makes possible to evidence that the bigger the level of detail described in the mathematical model, the closer the similarity in behavior between the simulated system and the real one. Consequently, the flexibility of the model will decrease up to the point it will only describe a specific physical system, in this case, a specific working layout of a pick and place packaging system.

A proper design of a scheduling strategy can be obtained with the utilization of a suitable mathematical model and the implementation of an algorithm. Together, they will allow to simulate the interaction of the pick and place activities performed by the pickers at each robotic cell. In addition, it is possible to analyze if the criteria used for the design of the scheduling strategy will define correctly the desired performance of the packaging line, or, on the other hand, it will not lead the entities of the process to achieve the desired behavior.

The methodology implemented for the development of this work allows to achieve an accurate, but not perfect, description of the behavior of the whole system analytically. Afterwards, it is highly recommended to simulate the plant and the scheduling strategies designed on 3D simulators in order to confirm the results obtained previously, and, to evaluate further limitations or problems that may arise due to some features of the system that are omitted or not considered in the mathematical modeling process.

Since the mathematical formulation of the packaging system is based on specific criteria that might not coincide all the times with the real specifications of currents industrial packaging plants, the mathematical model and the scheduling strategies developed may not prove to be accurate to describe the plant and achieve the desired output response of the system. If the layout specifications of the packaging system change with respect to the ones described on this work, a different scheduling strategy should be studied and tested.

As a primary future work, it is suggested to study the behavior of the system when the method in which the incoming product is supplied changes. For instant, considering that the incoming of product is not supplied uniformly.

Afterwards, if the scheduling strategies designed prove not to be successful in achieving an optimal performance for the whole system, it is necessary either to design a new job assignment technique, for instance, to explore the possibility of having an on-line job assignment technique, or, to design a new scheduling strategy that will allow the plant to fulfill completely its working objectives.

Finally, it can be included further extensions on the mathematical formulation of the system, such as, new state variables. In particular, the control of the speed in both conveyor belts or the action of an optical detection system which detects the position and alignment of the individual product units.

Bibliography

[1] H. Isil Bozma, M. E. Kalalioglu. "Multirobot coordination in pick-and-place tasks on a moving conveyor", *Robotics ans Computer-Integrated Manufacturing*, 28, pp. 530–538, 2012.

[2] Lorenzo Comba, Gustavo Belforte, Paolo Gay. "Plant layout and pick-and-place strategies for improving performances in Secondary Packaging Plants of food products". *Packaging Technology and Science*, 2012

[3] F. Erzincanli, J. M. Sharp. "Meeting the need for robotic handling of food products ",*Food Control*, Vol 8, 4, pp. 185–190, 1997

[4] R. Mattone, M Divona, A. Wolf. "Sorting of items on a moving conveyor belt. Part 2: performance evalutaion and optimization of pick and place operatons",*Robotics and Computed Integrated Manufacturing*, 16, pp. 81–90, 2000

[5] Jiang-Liang Hou, Nathan Wu, Yu-Jen Wu. "A job assignment model for conveyor-aided picked system", *Computers & Industrial Engineering*, 56, pp. 1254–1264, 2009

[6] Slim Daoud, Hicham Chehade, Farouk Yalaoui, Lionel Amodeo. "Efficient mataheuristics for pick and place robotic systems optimization", *Journal of Intelligent Manufacturing*, June 2012

[7] Makoto Yokoo, Edmund H. Durfee, Toru Ishida, Kazuhiro Kuwabara. "The Distributed Constraint Optimization Problem: Formalization and Algorithms". *IEEE Transactions on Knowledge and Data Engineering*, Vol 10, 5, pp. 673–685, 1998

[8] Clive L. Dym. *Principles of Mathematical Modeling*, 2^{nd} edition, ELSEVIER Sciencie, California, 2004

[9] Alfio Quarteroni. "Mathematical Models is Sciencie and Engineering". *Notices of the AMS*, vol 56, 1, pp. 10–19, 2009

[10] Robert D. King, Charles D. Turnitsa. "The Landscape of Assummptions". *Old Dominion University*, pp. 81–88. 2008

[11] In Jae Myung, Mark A. Pitt, Yu-Jen Wu. "Mathematical Modeling", *Blood*, Vol 3, 2, pp. x–901, 2002

[12] Pick and place with delta robot: for creating order. [Online]. Available: http://www.lenze.com/hu-hu/industry-sector-expertise/consumer-goods-industry/packaging-line/pick-and-place/. [Accessed: March, 2013]

[13] Florlan Herzog. "Method and apparatus for filling containers with piece goods". *US Patent 6826444*, 2004

[14] Emil Hüppi, Peter Dübendorfer, Frank-Peter Kirgis, Markus Sidler, Jacob Van Kogelenberg, Stephan Schüle. "Method and apparatus for putting piece goods into containers". *US Patent 6901726*, 2005