U.PORTO

FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Industrial Informatics

Integrated Masters on Eletrotecnical and Computer Engineering

2020/2021

Automation of a Production Line

WARNING

The customer may need to make changes to the specifications up to 15 days before the delivery deadline.

Mário de Sousa 2021-03-12 ver. 1.0

1 Introduction

This document describes the work you will be doing throughout the Industrial Informatics course. In a simplified way, it consists of:

Requirement 1: automating the production of a flexible production line simulator,

Requirement 2: development of a MES (Manufacturing Execution System) to monitor and manage the production on the plant floor

Typically, the lowest level control (requirement 1) will be achieved using PLCs or equivalent equipment. For the control of the immediately higher level (requirement 2, MES) it is advisable to use higher level programming languages and tools (C ++, Java, Python, C #, ...), and the use of a DBMS (database management system) is also mandatory (e.g. PostgreSQL, MariaDB, ...).

The description of the equipment in this line, and how to interact with them, is made in a separate document. In this document, only the requirements that the control and monitoring systems must comply with will be presented. In order to understand these requirements, it is essential to know the equipment under control, so it is suggested to read the document describing the equipment of the production line in advance.

2 Overview of the Flexible Production Line

The Flexible Production Line to be automated consists of 4 cells: 1 automatic warehouse, 2 machining cells (with 4 machines each), and 1 cell for loading raw material and unloading processed parts.

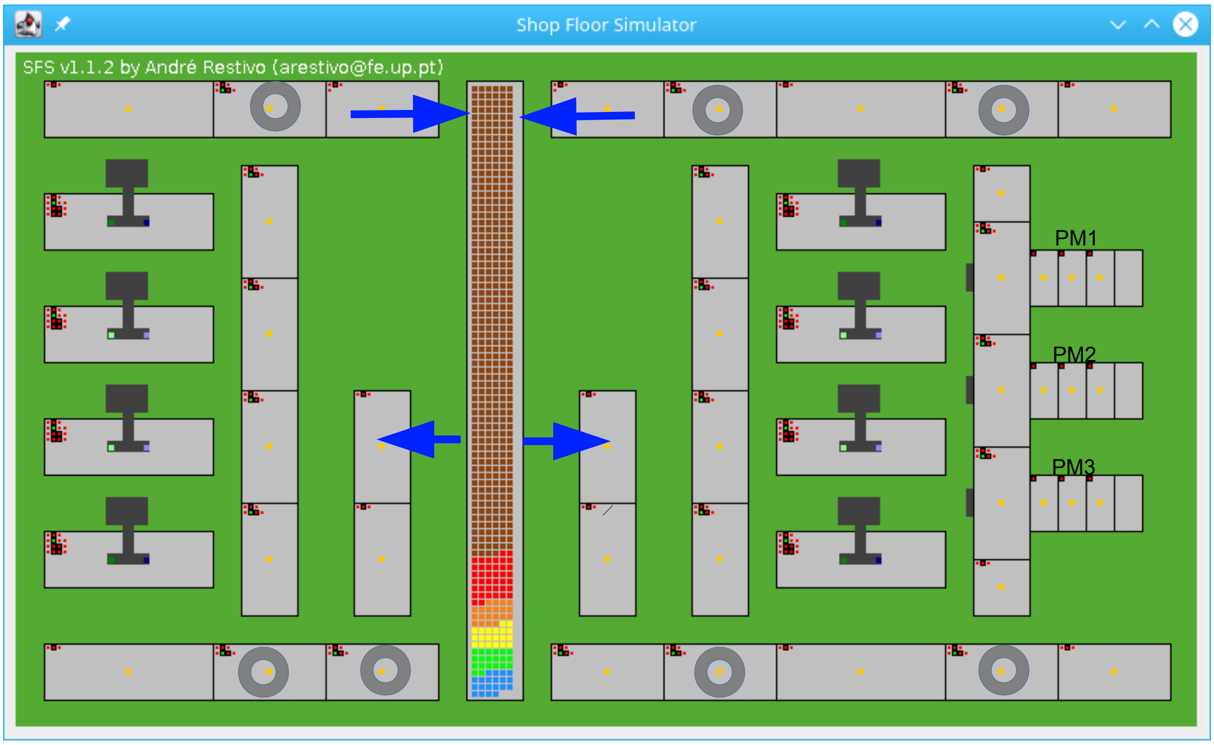

Figura 1: Linha de produção flexível – disposição dos equipamentos.

There are two stations/conveyors to remove parts from the warehouse, and another two to deposit parts in the warehouse. All conveyors are bi-directional, and there are conveyors that can rotate changing between an horizontal and vertical alignment (marked with a circle in figure 1).

This production line will be simulated by the SFS software (Shop Floor Simulator) which, being developed in Java, is compatible with Windows, Linux and OSX. Access to the logic signals (sensors and actuators) is done through the Modbus/TCP protocol. The SFS simulator implements a Modbus/TCP server, in which the sensors (Boolean signals) are mapped as Input Discretes, and the actuators (Boolean signals) as Coils. The addresses of each signal are available in a file io.csv (format comma separated values, which can be opened as a spreadsheet). This file is generated by the simulator itself with the command “java -jar sfs.jar –csv io.csv”.

The description of the signals available to control each equipment (machine, conveyor, rotating conveyor, pusher and warehouse) is described in another document.

2.1 The Production (Transformation) Process

The production line processes different types of parts (P1, P2, ...), represented with different colors. The color convention used (same as the color convention for electrical resistors) is defined in the following table.

Peça	Cor
P1	Castanho
P2	Vermelho
P3	Laranja
P4	Amarelo
P5	Verde
P6	Azul
P7	Violeta
P8	Cinza
P9	Branco

Tabela 1: Convenção de cores e tipos de peças.

When processing a part of a certain type on one of the machines, using a given tool, for a certain period of time, this part changes its type.

Each machine has a private warehouse of three tools (T1, T2, and T3), among which you can alternate in order to produce different parts according to the raw material supplied. Changing between two tools takes 30 seconds. The transformations of parts that may occur depend on the type of tool, and are described in the following table:

Starting Part	Produced Part	Tool	Processing Time
P1	P2	T1	15 s
P4	P5	T1	15 s
P6	P8	T1	15 s
P2	P3	T2	15 s
P5	P6	T2	30 s
P3	P4	T3	15 s
P5	P9	T3	30 s
P6	P7	T3	30 s

Tabela 2: Sequências de transformações possíveis

If you try to perform operations that are not found in the table above (for example: processing P8 with tool T3), the part will be unusable and will be represented in black. Similarly, if you perform an operation described in the table above, but for a longer period of time than indicated, the part is also unusable. If you start to perform a transformation operation but stop before the necessary time has elapsed, the transformation is incomplete, leaving the piece with the two colors (of the original piece, and of the final piece) in the correct proportion.

In order to produce a final piece, it is necessary to take into account the processing sequence of the raw materials. This sequence results from the successive combination of raw material and tool, which may generate by-products for further processing. In this way, if you want to produce a P3 piece from another P1 piece in the warehouse, it will be necessary to apply T1 to the P1 piece resulting in a by-product of the type P2, followed by T2 resulting in P3.

The production line starts with a fixed number of parts in the automated warehouse; 400 pieces P1, 40 pieces P2, and 20 pieces each of P3, P4, P5 and P6.

2.2 The Load/Unloading Process

The rightmost cell is used for loading and unloading parts. Unloading is carried out by one of the 3 unloading docks (PM1, PM2 and PM3). Loading is carried out by one of the two extreme conveyors of the upper and lower transport lines (i.e. conveyors with identifier C7T1b and C7T7b).

The loading stations function as normal conveyors on which pieces will be manually deposited, which must be transported to the warehouse.

Each unloading station consists of a pusher and a slider. The slider is an inclined ramp on which a maximum of 4 pieces can fit. The pusher is a mechanism that pushes the piece of the conveyor in line with the slider, to the respective slider. The pieces will be removed manually from the far right of each slider. Each position of the slider (except the last one, to the right) has a piece presence sensor.

3 Controling the Production Equipment

For your work you must develop the software to control the equipment that is part of the cells of the production line described above (requirement 1 mentioned in section §1). However, it will still have to develop the MES that supervises and monitors the production line (requirement 2).

It is strongly recommended that the lower level control is not done holistically, but rather in a modular way. It is suggested that you start by organizing / dividing the production line into control cells - cells that should be able to operate as independently as possible from the rest of the cells. Note that the cells that you define for your control do not have to coincide with the division into cells that was made when describing the production line (figure 1).

You should then define how the two components (MES and PLC) will interact. In technological terms, it should adopt the OPC-UA communication technology.

4 Services Provided by the Production Line

The company has an ERP software (ERP - Enterprise Resource Planning, MRP - Material Requirements Planning, ...) that makes the decision on the production orders that must be executed, and that will be communicated to the MES that will develop. Thus, your MES must accept and execute the commands it receives from the ERP.

Typically, new transformation orders will be sent by the ERP at the beginning of each day. If there are enough orders from customers, the ERP will send orders that allow MES to plan production for the next 5 days. In other words, pending orders stored and managed by the MES should, in principle, take up to 5 days to be completed. However, urgent orders may appear at any time, which are generally of a smaller size.

NOTE: For the purpose of faster simulation, we will consider a 'day' as corresponding to 50 s of real time. That is, the ERP will send orders every 50 s.

It will be up to your control system to manage the stocks of parts in the warehouse. Sometimes the ERP will ask your control software for the number of parts in the warehouse at that moment.

4.1 Orders

The MES will receive ERP orders encoded in XML files, which will be sent to port 54321 of the UDP / IP protocol. Each file may include one or more manufacturing orders, coded according to the following examples.

- **Workpiece Transformation/Processing:**

  The XML will contain:
  <Order Number=”nnn”>
  <Transform From=”Px” To=”Py” Quantity=”XX” Time=”TT” MaxDelay=”DD” Penalty=”PP”/>
  </Order>

  In which:
  nnn – order number
  Px – starting workpiece type
  Py – target workpiece type
  XX – quantity to produce
  TT – time instant in which the order is sent to the MES (in seconds)
  DD – relative deadline (in seconds) to finish executing this order (counting from the time in which the order is sent)
  PP – penalty (number between 0 and 1000) incurred for each day of delay in completing the order

  Upon receiving a transformation order, your control system should transform XX pieces of type Px (original pieces, x being a number from 1 to 6) into pieces of type Py (final pieces, with y from 2 to 9). The starting (final) parts will always have the warehouse as source (destination).

  It is possible that you will receive transformation orders for which there is not enough Px parts in the warehouse to complete the entire order. In these cases, you must start by executing the transformation order until the Px parts are exhausted in the warehouse, leaving the remaining parts of the transformation order in suspension until they can be executed (for example, after completing another transformation order that is generating Px parts). Therefore, it will be up to the MES (which will implement) to manage the existence of parts in the warehouse. On the other hand, you can assume that there will always be places available in the warehouse to store the produced Py parts.

  The order also indicates the maximum order completion time. This time is counted from the moment the order is delivered (i.e. a relative time).

The penalty is a value to be applied for each 'day' of delay in completing an order. For example, orders that end with a delay between 1 s and 50 s will have a penalty value of ZZ. Orders delayed from 51 s to 100 s will incur a penalty of 2 x ZZ.

– **Unloading of Workpieces**

The XML file will contain:
<Order Number=”nnn”>
<Unload Type=”Px” Destination=”Dy” Quantity=”XX”/>
</Order>

In which:
nnn – order number (ID)
Px – workpiece type to unload
Py – Unloading dock on which to unload
XX – number of work+ieces to unload

Indicates a request to deposit at the specified discharge location Dy (y = 1 → PM1, y = 2 → PM2, y = 3 → PM3), XX specimens of Px type parts. The deposited parts will later be removed manually. If the unloading site is complete, and the operator is late in removing the parts, the remaining parts of the order must remain on the conveyors (if they have already been removed from the warehouse) or in the warehouse, until there is again availability at the destination location;

– **Current Stores**

The ERP will ask your software for the number of pieces of each type existing at that time in the warehouse. To do this you will send an XML file that includes:

<Request_Stores/>

Your software should respond, to the same IP / port that sent the request, with an XML file that should contain:
<Current_Stores>
<WorkPiece type=”Px” quantity=”XX”/>
<WorkPiece type=”Px” quantity=”XX”/>
...
</Current_Stores>

In which:
Px – workpiece type
XX – number of workpieces of this type currently in the warehouse

- **List of Orders**

The ERP will ask the MES for the total list of transformation orders, including orders already completed, in process, and yet to be initiated. To do this you will send an XML file that includes:

<Request_Orders/>

Your software should respond, for the same IP / port that sent the request, with an XML file that should contain the total odors list with the following information:

<Order_Schedule>
<Order Number=”nnn”>
<Transform From=”Px” To=”Py” Quantity=”XX” Quantity1=”X1” Quantity2=”X2” Quantity3=”X3” Time=”TT” Time1=”T1” MaxDelay=”DD” Penalty=”PP” Start=”ST” End=”ET” PenaltyIncurred=”PI”/>
</Order>
...
</Order_Schedule>

In which:

nnn – order number (ID)
Px – starting workpiece type
Py – target workpiece type
XX – total number of workpieces to process/transform
X1 – number of workpieces already processed
X2 – number of worpieces in production / being processed
X3 – number of outstanding workpieces, yet to process
TT – time instant in which the order was sent by the MES (in seconds)
T1 – itime instant in which the order was effectively received form the MES (in seconds)
DD – relative deadline (in seconds) to finish executing this order (counting from the time in which the order is sent)
PP – penalty (number between 0 and 1000) incurred for each day of delay in completing the order

ST – time instant (in seconds) at which the order started to be executed (in case it has already begun execution) or time instant at which it is predicted to start (in case it has not yet begun execution)
ET – time instant (in seconds) at which the order finished execution (in case it has already finished) or time instant at which it is predicted to finish (in case it has not yet finished executing)
PI – penalty incurred (in case it has finished execution) or predicted penalty (in case it has not yet finshed execution)

- **Workpiece Loading**

Your control system should store in the warehouse all parts that are deposited in either of the two loading locations (i.e. the two rightmost conveyors). This order must be triggered by the presence of a piece in any of these conveyors.

The parts deposited on the conveyor immediately above PM1 will always be of type P1. The parts deposited on the conveyor immediately below PM3 will always be of type P2.

All of the orders described above must be processed simultaneously, as long as the factory layout and available resources allow it. This is not to say that the completion of orders is always the same by which they entered (for example, an order that requires only machines A may end before another order that uses only machine B, although the latter arrived first).

You should try to give priority to orders that have a shorter delivery time. The deadline of orders that are not for transformation (loading, unloading, ...) should be considered to be immediate (i.e., relative deadline = 0 s). Deadlines should be considered as objectives. Sometimes, due to excessive orders, it may not be possible to meet all deadlines.

4.2 Statistics:

To allow users to monitor the production process, the MES must provide the following information:

- **Machine Statistics**

  A list with the statistics of the operation of each machine. For each machine, provide:
  - total operating time,
  - nnumber of operated workpieces (total and for each type).

- **Unloaded Workpieces**

  For each unloading dock, provide:
  - the number of unloaded workpieces (total and by type).

The interface through which the monitoring is carried out is not specified, so you are free to implement it as you see fit. If you wish, you can extend the communication protocol so that they are made available in XML format on the UDP port.

4.3 Persistance

The information managed by the MES must be persistent, i.e. it should not be lost if the MES has to be turned off. This requirement must be achieved using a database managed by a database management system (e.x .: MariaDB, MySQL, PostgreSQL, ...).

The information to be placed in the database should be such as to allow the MES to be disconnected and resumed later without loss of information regarding pending orders, or statistics produced up to that time. During the period in which the MES is disconnected, it is acceptable to stop accepting new orders from the ERP. With MES turned off, production operations may also be affected, and the level at which they are affected will depend on the way in which the architecture of the production system is organized.

5 Expected Outcomes

This project should be carried out by groups of 4 students.

All the operating logic behind the control of the production line must be analyzed, designed and implemented by the students, who must identify, in addition to the components described above, all the components that will be necessary for the operation of the line. It is recommended that you start first by not considering the possible optimization of the use of available resources. However, the optimization of resources will not fail to be valued when evaluating the project.

The expected results of the project are:

1. **Application architecture model**, composed, at least, by UML class, objects, and sequence diagrams. To be delivered by **March 14, 2021**;

2. **Final Report** (50% of the grade), to be delivered by **28 May 2021.**

3. **Demonstration** and discussion of the application (50% of the grade), in the week of **24 to 28 of May 2021**.

Distribuiçao de Esforço Entre Elementos do Grupo:

The grades of the works will be attributed to the group. The report should indicate how the group wants this note to be distributed to each member of the group. This distribution must be made as a percentage in such a way that the sum of the percentages is 100.

Calculo da nota final individual	
Final Group Grade: G (0..20)	A = Max[0; SIN(pi/2 * (Px/P - 1) / (1/P - 1))]
Weight for each group member: P1, P2, P3, P4	B = Min[0; (Px/P - 1)]
P = Average(P1, P2, P3, P4)	C = (20-G)*G/20 * A
Individual Grade Gx = G + C + D	D = G * B

Final Report:

The report should be written in arial font with size 11pt and spacing between lines of 1.5 lines, and a single column, with a 2cm margin around A4 pages.

The report should also follow the following structure:

- (2%) **Introduction** (up to 0.5 pages of text)

Summary of the approach followed with identification of the problem, technologies used, some special characteristics of your solution, some results that you want to highlight and structure of the report. For this to fit in half a page, they will have to be succinct. Here it is only interesting to give a quick idea of what they did and what they will show next.

- (7%) **Structure of Code** (up to 1.5 pages of text, excluding images)

Description of the (static) structure of the code using class and / or object diagrams. The text should highlight the main components of the code and refer to the respective diagrams.

- (7%) **Code Operation** (up to 1.5 pages of text)

Description of the (dynamic) interaction between the elements of the code using interaction, activity and / or state diagrams. Once again, you should use the text to explain some more elaborate interactions or some details that you want to highlight, always referring to the respective diagrams.

- (7%) **Implementation** (up to 1.5 pages of text)

Justify the choice of technologies used, describe the way used to map classes in each technology, describe the high-level architecture of the program (infinite cycle vs threads, etc ...), explain how to invoke methods between MES and PLC (variables Modbus, etc ...)

- (5%) **Final vs Initial Architecture** (up to 1.5 pages of text)

Discuss the main changes introduced between the solution initially proposed and the one that was implemented.

- (7%) **Comparison with Common Design Patterns** (up to 1.5 pages of text)

Critical comparison of the architecture used in the project with the standards (design patterns) proposed by the ISA 95 standard, RAMI 4.0, and the article "Service Granularity in Industrial Automation and Control Systems".

- (5%) **Team Organization** (up to 1.0 pages of text)

Describe the assignment of tasks within the team, the development model, critical comments on the performance of team members and any changes made to the assignment of tasks. Indicate how the group wants the note to be distributed to each member of the group

- (5%) **Testing and Results** (up to 1.0 pages of text)

Describe the tests performed, in particular what complex situations were tested. Performance can also be quantified (e.g., order XXXX took xx seconds to be processed; sequence of orders X, Y and Z took yy seconds ...). Critical appreciation of these results.

- (5%) **Conclusions** (up to 1.0 pages of text)

Summary of what was said in the report, highlighting some results considered more relevant.

6 Competition

A competition will be launched at the end of the semester that will be won by the group that manages to complete a certain sequence of manufacturing orders with the lowest penalty. The final time to complete all orders will be considered as a tiebreaker criterion. The group that wins the competition will have a bonus of 2 values in the practical note. The group in second place will have a bonus of 1.5 values.

The sequence of manufacturing orders that will be used in the competition will be delivered after the project's delivery deadline. The sequence will be made available in the form of a script so that orders are launched at predetermined times (and not all at once at the start of the program).

Groups interested in competing must submit the parameters (total penalty incurred and processing time). The winning groups will also have to demonstrate to the teacher (in a public session) the execution of the orders live (i.e. with the control software running).