U.PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Electrotechnical and Computer
Engineering Department

Flexible Production Line

Version 3 (2023/02/11)

1 Introduction

This document describes the simulator of a flexible production line, inspired by the kit that exists in the DEEC automation laboratory.

2 Flexible Production Line Simulator

The simulator consists of a Java program, which can be run on any platform that has a Java virtual machine (e.g. Windows, Linux, OSX). The simulator tries to follow as accurately as possible the kit mentioned above.

Unlike the production line that is set up with a fixed layout, the simulator allows production lines to be built with the most diverse layouts, and with any number of sub-components (i.e. machine tools, simple conveyors, rotary conveyors). In other words, the layout of the production line is configurable, both in terms of the amount of equipment of each type, as well as its distribution over the factory floor. Thus, the same simulation programme can simulate the most diverse distributions of equipment throughout the plant. The configuration is contained in a file, which can be manually edited. However, for the work that is proposed, you will be given a file with a distribution of equipment that follows the distribution of the production line previously described. For this reason, the format of the configuration file will not be described in this document. [1]

The layout of the manufacturing line on which you will carry out your work is probably quite different from the one in the following figure, which is a simulation of the physics kit in the automation laboratory. It will nevertheless serve to describe how each type of equipment works.

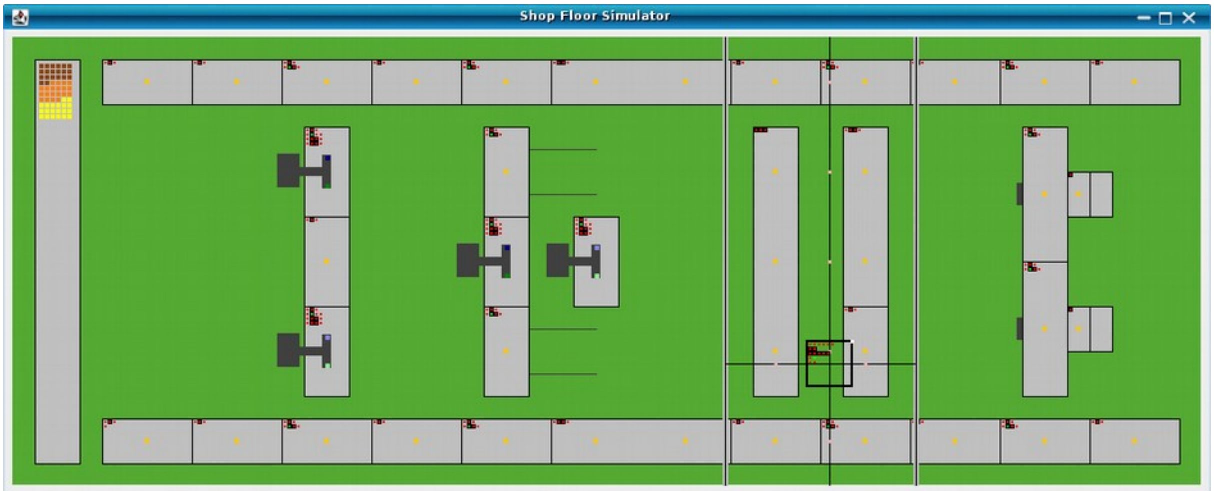


Figure 1: General aspect of the simulator

Each piece of equipment (conveyor, machine tool, etc.) simulated by the simulator will be controlled via inputs and outputs (mostly binary). Control of this equipment will be via the Modbus/TCP communications protocol. For all intents and purposes, the simulator behaves like a single Modbus/TCP slave that can be controlled by any PLC supporting this protocol.

1 The format of the configuration file is very intuitive and self-explanatory, so you are free to experiment as you see fit.

As usual with Modbus slaves, the inputs and outputs are referenced by their memory address (starting with 0), and occupy distinct memory locations. In other words, the binary input with address 6 is distinct from the binary output with the same address 6.

Each device uses a certain number of inputs and outputs. The exact number depends on the type of device. For example, a single conveyor occupies 2 outputs (to command the movement of the conveyor) and one input (to read the sensor in the middle of the conveyor). The exact address of these inputs and outputs depends on the position at which the subcomponent is set in the configuration file.

The addresses are taken up/used sequentially (i.e. with addresses increasing from 0) in the order in which the devices are defined in the configuration file. For a complete listing of the addresses of all the devices in the factory, run on the command line:

```
$ java -jar sfs.jar --csv enderecos.csv
```

or

```
$ java -jar sfs.jar --map enderecos.txt
```

With the 'csv' option, a file will be created with the information in a table, separated by commas, so that it can be easily used by a spreadsheet editing program. With the 'map' option, the file created will have more text in order to make it easier for humans to read.

For example, a plant with only two conveyor conveyors would have the two actuation commands of the first conveyor allocated to addresses 0 and 1, while the commands of the second conveyor would be allocated to the following addresses, i.e. 2 and 3. The sensors of these conveyors (one for each conveyor) occupy a second address space dedicated to binary sensors, so the sensor of conveyor 1 would be allocated to address 0, while the sensor of the second conveyor would occupy address 1.

When you are controlling the simulator, you can record the sequence of commands it has received (i.e., record the simulation) as long as you run it with the command

```
$ java -jar sfs.jar --record teste1
```

To view again the simulation you have just saved, execute the command

```
$ java -jar sfs.jar --playback teste1
```

3 Representation of equipment and parts

In the simulator each device (conveyor, machine) appears as a few coloured dots in the top left-hand corner of its representation. The points with a grey background indicate the state of the commands that the device is receiving, and the points with a black background the state of their respective sensors. As you would expect, red indicates deactivated (FALSE) and green indicates active (TRUE).

3.1 Linear conveyors

A linear conveyor can move a part in two opposite directions (axes **XX** or **YY**, depending on the position of the conveyor). The movement always takes place at the same speed so that the conveyor motor can be controlled using a binary signal for each direction. In principle it should not be permitted to control the movement in both directions simultaneously (with the risk of damaging the equipment), however the production line is protected (electrically) against this eventuality, resulting in both motors stopping, so the simulator behaves in the same way.

All conveyors have one or more sensors enabling the presence of an item on the conveyor to be detected (depending on the size of the conveyor). In shorter conveyors, the sensor is located at the centre of the conveyor. In longer conveyors there are two sensors per conveyor, placed approximately at each end of the conveyor. Each of these sensors has a corresponding binary output.

The linear conveyor has the following input and output signals (Table 1).

Type	Name	Description
Binary Actuator	mp	movement in positive direction (XX or YY)
Binary Actuator	mm	movement in the negative direction (XX or YY)
Binary Sensor 1	p1	presence of work-piece/part
Binary Sensor 2	p2	presence of work-piece/part
Binary Sensor 3		...

Table 1: Actuators and sensors of the linear conveyor

For conveyors with a horizontal position (**XX** axis) the nomenclature **me** / **md** (movement to the left / right) is used, while for conveyors with a vertical position (**YY** axis) **mc** / **mb** (movement up / down) is used.

Linear conveyors with a sensor in the middle have three squares (Figure 10) which indicate respectively:

- square (grey background) on the left: left/up scroll order;
- square (black background) in the middle: part presence sensor;
- square (grey background) on the right: right/down scroll order.

Figure 2: Aspect of two linear conveyors in the simulator

Some longer conveyors have two sensors, so their representation includes four squares in the top left corner, two for the controls, and another two (in the centre) for the sensors.

3.2 Rotating Conveyors

The rotating conveyor behaves in the same way as a linear conveyor, but its rotation about the **ZZ** axis can be controlled.

To control this extra rotation, the conveyor has two additional binary signals, one for each direction of rotation. In principle it should not be permitted to control the movement in both directions of rotation simultaneously (with the risk of damaging the equipment), but the production line is protected (electrically) against this eventuality, resulting in both motors stopping, so the simulator behaves in the same way.

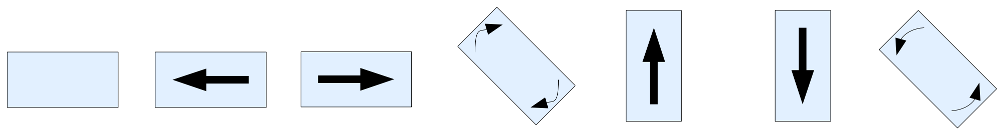
The conveyor can only be turned by a maximum of 90°. The two extremes of these 90° are signalled by two end-of-stroke sensors. If the conveyor continues to rotate beyond these limits, the device may be damaged.

The following table describes the respective input and output signals.

Type	Name	Description
Binary Actuator	mp	movement in positive direction (XX or YY)
Binary Actuator	mm	movement in the negative direction (XX or YY)
Binary Actuator	rd	rotation in the direct direction
Binary Actuator	ri	rotation in the reverse direction
Binary Sensor	p	presence of work-piece/part
Binary Sensor	d	end of rotation in direct direction
Binary Sensor	i	end of rotation in reverse direction

Tabela 2: Actuators and sensors of the rotating conveyor

A sequência de comandos, movimentos e posições possíveis está representada na seguinte figura.



mp=0	mp=1	mp=0	mp=0	mp=1	mp=0	mp=0
mm=0	mm=0	mm=1	mm=0	mm=0	mm=1	mm=0
rd=0	rd=0	rd=0	rd=0	rd=0	rd=0	rd=1
ri=0	ri=0	ri=0	ri=1	ri=0	ri=0	ri=0
d =1	d =1	d =1	d =0	d =0	d =0	d =0
i =0	i =0	i =0	i =0	i =1	i =1	i =0

Figure 3: Movement of the rotating conveyor

The rotary conveyors, in addition to an upper line with the same meaning as the linear conveyors, have a second line with the following meaning (Figure 11):

- square (grey background) on the left:

order of rotation in the direct direction;

- square (black background) on the left:

conveyor in the extreme position of rotation in the direct direction;

- square (black background) on the right:

conveyor in the extreme position of rotation in the reverse direction;

- square (grey background) on the right:

order of rotation in the reverse direction.

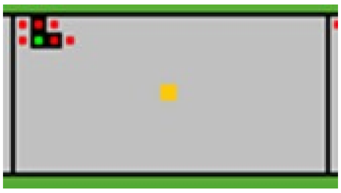


Figure 4: Aspect of a rotating conveyor in the simulator

3.3 Sliding Conveyors

The sliding conveyor behaves in much the same way as a rotary conveyor, the only difference being that the rotary movement is replaced by a linear translation movement (**XX** axis). Thus, the commands and sensors are the same as for the rotary conveyor, only with different names:

- square (grey background) on the left: translation order to the left;
- square (black background) on the left: conveyor in the extreme left-hand position;
- square (black background) on the right: conveyor in the extreme right-hand position ;
- square (grey background) on the right: translation order to the right.

The controls and sensors are the same as the rotary conveyor, just with different names.

Type	Name	Description
Binary Actuator	mp	movement in the positive direction (YY)
Binary Actuator	mm	movement in the negative direction (YY)
Binary Actuator	tp	translation in the positive direction (XX)
Binary Actuator	tm	translation in the negative direction (XX)
Binary Sensor	p	presence of work-piece/part
Binary Sensor	fp	end of translation in positive direction (XX)
Binary Sensor	fm	end of translation in negative direction (XX)

Tabela 3: Comandos e sensores do tapete deslizante

3.4 Pusher

A pusher allows a piece to be pushed off a conveyor. It is used mainly for sorting and arranging pieces. The conveyor is considered an integral part of the pusher and is controlled by the same controls as those of a linear conveyor.

The pusher uses two controls, one to move to the right (removal direction), and one to move to the left (return to home position). There are two end-of-stroke sensors that indicate the limit of each of these movements. To prevent physical damage to the equipment, you must never allow this movement to go beyond the limits.

Care should also be taken to only move the conveyor when the pusher is in its resting position.

Type	Name	Description
Binary Actuator	mp	movement in positive direction (XX or YY)
Binary Actuator	mm	movement in negative direction (XX or YY)
Binary Actuator	pr	contracts the pusher
Binary Actuator	pe	extends the pusher
Binary Sensor	p	presence of work-piece/part
Binary Sensor	fr	end of pusher contraction
Binary Sensor	fe	end of pusher extension

Tabela 4: Comandos e sensores do pusher

3.5 Work Table

A work table is only used to temporarily place workpieces. They are used within the range of the gantry robot so that it can place parts on these tables and stack parts on top of each other.

From the control point of view it behaves like a conveyor that never moves. It has only one presence sensor, and no control.

Type	Name	Description
Binary Sensor	p	presence of work-piece/part

Tabela 5: Comandos e sensores da mesa de trabalho

The work table has a single sensor (square with black background) representing the part presence sensor (Figure 12).

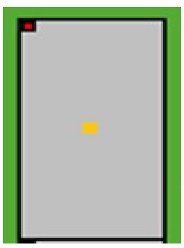


Figure 5: Aspect of a work table in the simulator

3.6 Warehouse

In order to simplify the work required from students, as well as the complexity of the simulator itself, the warehouse simulation assumes that the warehouse is already automated by an existing program. Thus, the interaction with the warehouse is limited to the request for storage of a part, or the withdrawal of a part of a certain type.

As the physical interface of the warehouse cell with the other cells is via two conveyors, the warehouse is represented by two types of conveyors. One type of conveyor allows parts to be removed from the warehouse, while the other type of conveyor allows parts to be stored in the warehouse.

Each conveyor is controlled in the same way as a previously described linear conveyor, so it has two digital outputs for the conveyor movement commands, as well as a digital input for indicating the presence of a piece on the conveyor.

To store a part, it must first be placed on the appropriate conveyor in the middle position. The storage order is indicated by a transition from 0 to 1 of a specific digital signal (in). The storage operation is considered to end as soon as the piece 'disappears' from the conveyor.

To remove a part from storage, the conveyor must initially be free. Next, the number of which identifies the type of part to be removed from storage must be indicated (via an 8-bit output). Note that for the removal request to be considered, this register must initially have the value 0. The part removal operation is considered to be terminated as soon as the conveyor appears to be occupied by the part.

The control signals for each of the warehouse interface conveyors are summarised below.

Type	Name	Description
Binary Actuator	mp	movement in positive direction (XX or YY)
Binary Actuator	mm	movement in negative direction (XX or YY)
Binary Actuator	in	insert part in the warehouse
Binary Sensor	p	presence of work-piece/part

Tabela 6: Actuators and sensors of the warehouse input conveyor

Type	Name	Description
Binary Actuator	mp	movement in positive direction (XX or YY)
Binary Actuator	mm	movement in negative direction (XX or YY)
Word Command	tp	type of part to be removed
Binary Sensor	p	presence of work-piece/part

Tabela 7: Actuators and sensors of the warehouse output conveyor

3.7 Machine Tool

A machine tool carries out operations on the workpiece on the conveyor attached to the machine. This conveyor is considered an integral part of the machine tool and is controlled by controls which are identical in every respect to the controls of a linear conveyor.

The machines have 3 tools mounted in a turret. The tool change is performed by rotating the turret (in one of two directions) until the desired tool is in the machining position. A sensor is activated whenever a tool (either one) is in the machining position. Since this sensor does not indicate which tool, but only the presence of any tool, it is the responsibility of the machine control program to store the number of revolutions performed and thus determine which tool is in the machining position. Each time the machine is started, it is assumed that the tool at the machining position is tool T1. The tools are mounted in the order T1, T2 T3.

The turret on which the tool is mounted can be moved in two independent axes - the **ZZ axis** (top-bottom), and the **YY axis** (left-right). For the position control in each of these axes there are 2 binary controls and two sensors which indicate the arrival at one of the extreme positions. Note however that the transformation operation can be carried out with the tool in any position.

Operation at fixed tool speed at the machining position is indicated by a simple binary control.

In order to prevent physical damage to the equipment, you must never allow movements in the **ZZ** and **YY** axes to exceed the limits.

Type	Name	Description
Binary Actuator	mp	movement in positive direction (XX or YY)
Binary Actuator	mm	movement in negative direction (XX or YY)
Binary Actuator	tc	change tool
Binary Actuator	--	change tool (other direction)
Binary Actuator	tr	rotate tool
Binary Actuator	yp	y-axis tower movement (positive direction)
Binary Actuator	ym	y-axis tower movement (negative direction)
Binary Actuator	zp	z-axis tower movement (positive direction)
Binary Actuator	zm	z-axis tower movement (negative direction)
Binary Sensor	p	presence of work-piece/part
Binary Sensor	pt	presence of tool (i.e. correctly aligned)
Binary Sensor	yp	End of movement y axis (positive direction)
Binary Sensor	ym	End movement y-axis (negative direction)
Binary Sensor	zp	End of movement z axis (positive direction)
Binary Sensor	zm	End movement z axis (negative direction)

Tabela 8: Actuators and sensors of a machine tool

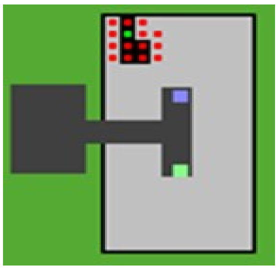


Figure 6: Aspect of a machine tool in the simulator

The machine tool conveyors, in addition to an upper line with the same meaning as the linear conveyors, have a second line with the status of the associated machine (Figure 13). On this second line:

- square (grey background) on the left: machine tool drive order;
- square (black background) in the middle: tool aligned and ready to be driven;
- square (grey background) in the middle: tool change order;
- square (grey background) on the right: tool change order (reverse direction);

The next two lines show the movement of the machine in the YY and ZZ axes respectively (in the simulator, the movement in the ZZ axis is represented by a movement in the XX axis). Each of the lines has:

- square (grey background) on the left: translation order left/up;
- square (black background) on the left: conveyor in the extreme left/up position;
- square (black background) on the right: conveyor in the extreme right/down translation position;
- square (grey background) on the right: translation order right/down.

3.8 Gantry Robot

The gantry robot makes movements in the 3 axes (**XX**, **YY**, **ZZ**), and for each axis there are two movement commands (one for each direction).

For the **ZZ** axis there are two sensors which indicate the two extreme positions (top and bottom). For the **XX** axis there are also two sensors which indicate the two extreme positions, which coincide with the correct alignment with the conveyors/tables below the robot. For the **YY axis** there are 5 sensors, which indicate that the robot is aligned with one of the five conveyors/tables of this axis.

A single binary control is sufficient to control the gripper (0 → opens the gripper; 1 → closes the gripper). For mechanical reasons, it is necessary to wait approx. 1 second after the clamp closing command to allow time for the workpiece to be clamped properly. A sensor is mounted on the clamp and indicates the presence (or not) of a piece.

To prevent collisions from occurring, all movements of the gripper along the **XX** and **YY axes** may only be carried out when the gripper is in the upper position of the **ZZ axis**. In addition, the movement along the **XX**, **YY**, and **ZZ axes** should never be allowed to exceed the limits of the operation zone.

When the system starts, the robot's position is unknown. It is therefore necessary to have a procedure for initialising it that places it in a known position.

Type	Name	Description
Binary Actuator	xp	movement in the positive direction (XX)
Binary Actuator	xm	movement in negative direction (XX)
Binary Actuator	yp	movement in positive direction (YY)
Binary Actuator	ym	movement in negative direction (YY)
Binary Actuator	zp	movement in positive direction (ZZ)
Binary Actuator	zm	movement in negative direction (ZZ)
Binary Actuator	g	Gripper control
Binary Sensor	xp	End movement in positive direction (XX)
Binary Sensor	xm	End movement in negative direction (XX)
Binary Sensor	zp	End movement in positive direction (ZZ)
Binary Sensor	zm	End movement in negative direction (ZZ)
Binary Sensor	y1	Position 1 of the YY axis
Binary Sensor	y2	Position 2 of the YY axis
Binary Sensor	y3	Position 3 of the YY axis
Binary Sensor	y4	Position 4 of the YY axis
Binary Sensor	y5	Position 5 of the YY axis
Binary Sensor	p	presence of work-piece/part

Tabela 9: Actuators and sensors of the gantry robot

The gantry robot is represented by a square which moves on the XX and YY axes according to the commands received. Its position on the ZZ axis is represented by a white dot moving along the black line at the right end of the box (Figure 14).

In the first row of sensors/commands, six squares with a grey background appear, representing the commands to move in the XX (+, and - directions), YY (+, and - directions) and ZZ (+, and - directions) axes respectively.

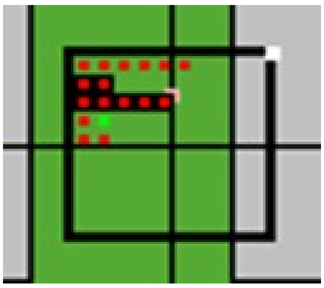


Figure 7: Aspect of the gantry robot in the simulator

The second and third lines show the status of the sensors that indicate the correct alignment of the robot with the conveyors underneath it. The second line represents the alignment when the robot moves on the XX axis, while the third line represents the alignment when the robot moves on the YY axis.

The fourth line shows the sensors that indicate the arrival of the robot at the extreme positions of the ZZ axis.

The fifth line shows, on the left, the gripper closing command and, on the right, the presence sensor of the part located in the robot's gripper.

4 Equipment controls

If you right-click on one of the simulator's resources (for example, on a linear pad), you can change the state of that resource's actuators, as well as insert or remove a part. Note, however, that the change of state of the resource's actuator is done momentarily, so it will only remain in the new state if there is no other command making new changes. If the PLC is commanding the simulator, it is continuously updating the state of its outputs, so the actuator state will quickly return to the state indicated by the PLC. However, if the PLC is not connected, the actuator will remain with the new value until it is changed again.