## Introduction

This exercise is meant to handle a real case of production machine using synchronous OBs (flowpack OB). This example is related to a machine preparing hamburger packs, where each box contains 2 pieces. Hamburgers have a diameter of 150 mm and come from a meat slicer machine not controlled by our system, which provides pieces in a random way, moved by a belt: distance between two consecutive products can vary so much, as shown in **figure 1**. It is also possible that a group of hamburgers comes out from the slicer, then some meters of empty space, then another group, and so on.

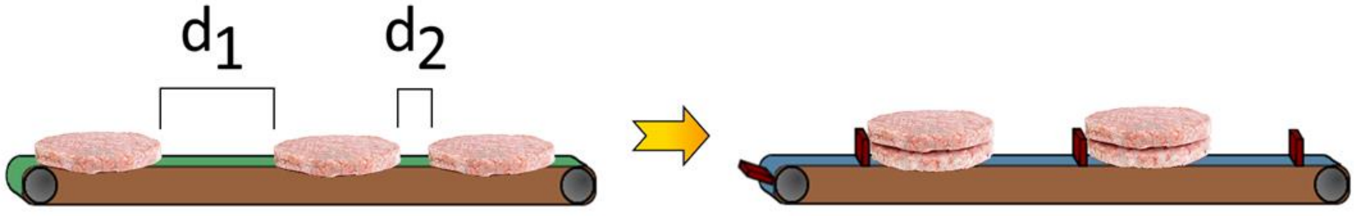


**Figure 1** - Machine input belt and output chain

Machine builder needs a solution to deal with incoming products at different distance, put them in piles of two and organize them on a rod chain to feed a flowpack machine.

### 1. Machine structure

In this exercise we don’t focus on the complete flowpack machine, but just on the feeding system that prepares the product. Machine axes are listed below:

- **Launcher**: this belt is controlled by an inverter (no servo motor) and brings products coming from slicer machine to load the step trains.
- **Step trains**: 3 coaxial step trains (each one has 6 slots) to load products and carry them to the chain feed position.
- **Loader**: an axis moving in a transversal way to push products from the wagons of the trains (slots) to be loaded on the chain.
- **Chain**: a rod chain to feed the flowpack machine. Each slot must contain a pile of two hamburgers, as shown in **figure 1**.

The launcher belt carries hamburgers coming out from the slicer machine and loads them on a step train, also called flowpocket. This object can be used to make the pile of two hamburgers, loading two of them in each wagon of the train, one over the other. After a wagon takes two products, the train makes a step, in order to bring a new empty wagon in the loading station position. When it’s fully loaded, a train moves to the other station and a loader pushes hamburger out from the train to feed the chain. A structure scheme is shown in **figure 2**.

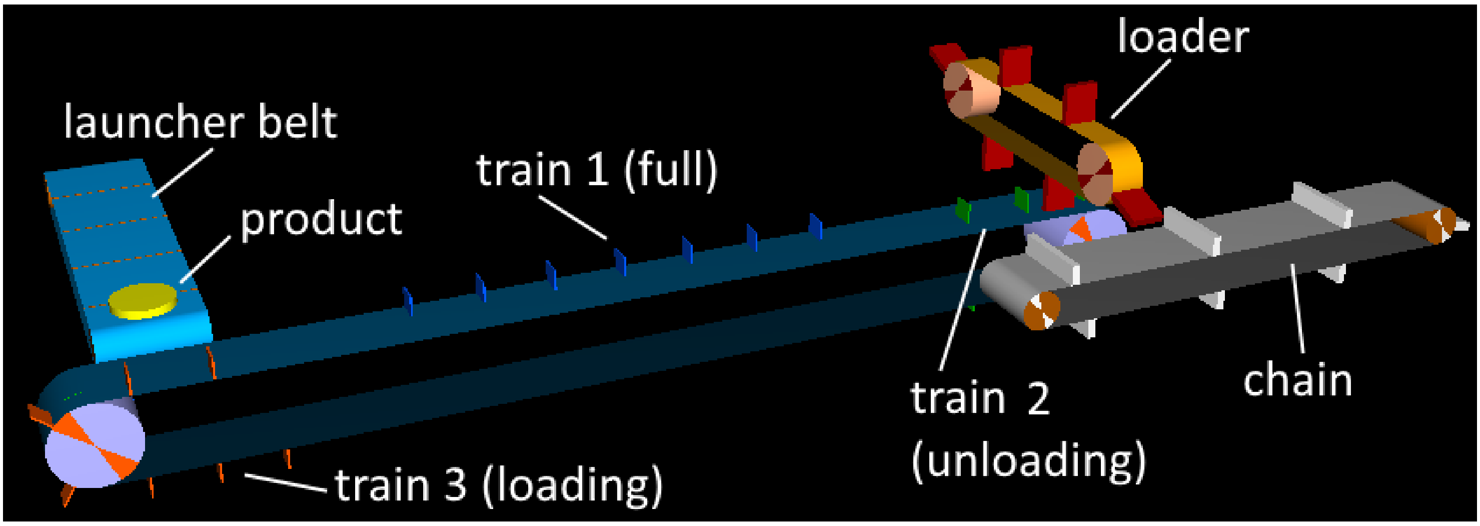


**Figure 2** - Machine scheme made with RDE 3D panel

To better understand the system, you can watch attached video (*chapter 5*).

## 1.1 Rod Chain

The rod chain is an axis bringing products to the flowpack machine and moves without any cam, just at linear speed. Each cycle is 400 mm long and fast stop time is 0.25 s. Oher parameters are listed as follows:

| | **Speed [cycle/s]** | **Acc [cycle/s$^2$]** | **Jerk [cycle/s$^3$]** |
|---|---|---|---|
| Jog | 0,5 | 5 | 25 |
| Reach master | 0,2 | 6 | 30 |
| Master lock | 3 | 6 | 30 |

Chain rod number is not set in this example. If you need it for the 3D panel of RDE, set the number you prefer (in **figure 2** there are 7 rods).

## 1.2 Loader

The loader axis is built as a rod chain with 6 rods with 300 mm of distance between each couple. The loader must perform a cam (**figure 3**) to push products in the chain at the correct speed and phase.

- Slowdown cam master angle: 120 deg
- Cam k_slave_speed: 0.5

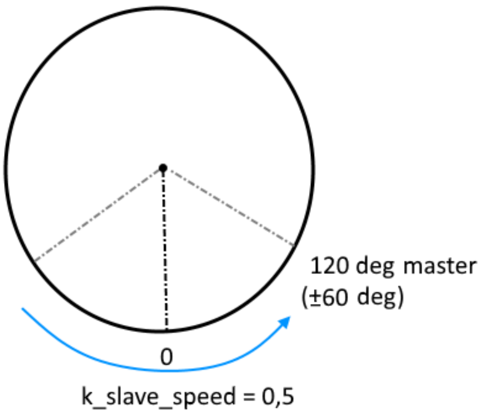


**Figure 3** – Loader cam scheme

Since the loader is mechanically moving at 90 degrees with the chain, a phase of 100 degrees is necessary to avoid hitting the chain rods and correctly push the products in the chain slots. All the other kinematics parameters are the same as the chain.

The loader moves across trains and chain, so it must also give the order to trains to make one step forward when a wagon is empty. The motion of the train must be phased and synchronized with the loader, to avoid a crash between trains and loader rods.
To do this, a **digicam** is necessary.

- Phase = 120 deg
- Width = 72 deg = 0,2 cycle

## 1.3 Step trains (flowpocket)

Step trains, also called flowpocket system, is a mechanism made of some belts (usually 3) with a series of slots, behaving like a train made of wagons. Each train goes to the loading station, where it takes the product, then it goes to the unloading station, where a loader system sends products into the chain.
Chain and loader are synchronous, but the products come from the launcher belt in asynchronous way, at random time and with variable distance; step trains must work in the

middle of these two opposite behaviors, loading in asynchronous way and downloading synchronously. This is the main purpose of step trains.
Another purpose of step trains is to load products in piles: in fact, in this example it is requested to feed the chain with piles of two hamburgers; so, each wagon waits in the loading position to receive two products from the launcher belt, one over the other. Only after two products have arrived, the train can do a step forward and start loading a new wagon.
To control the step trains, Robox system provides some OB: there are **rc_step_train** and **rc_step_train_manager**, which handles all the flow control of the various trains. You must read related manuals to know more.

Step trains data:

- Total race = 6000 mm
- Train number = 3
- Slots per train = 6
- Pocket length = 170 mm
- Load pos = 1200 mm
- Unload pos = 3500 mm
- Load/Unload acc = 2800 mm/s$^2$
- Fwd speed = 800 mm/s
- Rev speed = 1200 mm/s

Since this is just a simulation, you need to simulate products coming and have a register to change their speed runtime, like a percentage of time between two consecutive hamburgers. When a wagon is full with two pieces, you can tell the train to move a step using **train_manager.load_prod()** method.

### 1.4 Launcher belt

This belt is the begin of the system, receiving products from the slicer and sending them to load the step trains. This belt is controlled by a **rc_flow feed** OB and in this case, it doesn't have a related rc_axis OB, because this axis is controlled in speed mode by an inverter and not by a servo motor. The control output is an analog output word **OUT_W (50)**, whose value is 0 if the axis is stopped and 2048 if it reaches its **max_spe**.
When the train is making a step to change the loading wagon, it is important that no product arrives, otherwise it will crash. For this reason, the launcher makes a slowdown, reaching its **min_spe**, set by a NVRR; the slowdown lasts for a certain time, set by another NVRR. Use **speed_sel** to choose the speed to use.

This axis is not working in synchronous way, since it is receiving products from the meat slicer and will receive a slowdown order from the train when it must move a step.
Therefore, in automatic rule the launcher will be controlled with **move** command.
In **figure 4** you can see that when the train moves (**IV (3)** makes a jerked step), the **launcher.spe** is reduced for a while, reaching its minimum value.

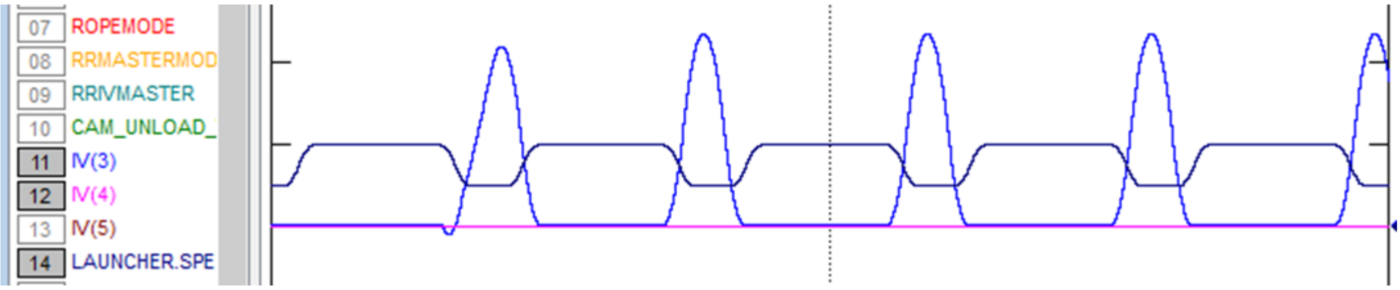


**Figure 4** – Launcher axis slows down when the train is making a step forward

Launcher data:

- Cycle_len = 300 mm ; double of product length, to keep space in between
- Move_spe = 1 cycle/s
- Min_spe = move_spe * nvrrKSlow ; default slowdown is 50%
- Min_spe time = 0.6 s
- Max_spe = 2 cycle/s; Max_acc = 20 cycle/$s^2$
- Move_acc, Stop_dec = 5 cycle/$s^2$; Move_jer/Stop_jer = 25 cycle/$s^3$
- Man_spe = 0,5 cycle/s; Man_acc = 3 cycle/$s^2$; Man_jer = 15 cycle/$s^3$

## 2. RDE project

Create a new RDE project.
Create 5 axes and a power set to handle them.
Write task and rule to control the system. RP1 or RP2 controllers should be used.
Program this synchronous machine using Robox library Flowpack OBs.

## 3. Write rule and task file in R3 language to control the machine

Prepare to handle these operative modes:

- Power off
- Fast stop
- Idle
- Zero cycle
- Manual

- Posit
- Auto

Remember that task is used to make FSM and switch among operative modes (SELECT command), together with OB parameters update. Rules are used to control the motion. Zero cycle is omitted in this sample, just set zero done to emulated axes.

If stop button is pressed during automatic, the machine must perform a phase stop at 220 deg of master cycle. Standard machine speed is 60 ppm, but it can be modified runtime at any value from 10 ppm up to 60 ppm. Oher parameters are listed below:

| | **Speed** | **Acc [cycle/s$^2$]** | **Jerk [cycle/s$^3$]** |
|---|---|---|---|
| Jog | 0,4 [cycle/s] | 3 | 15 |
| Move (auto) | 10 – 60 [ppm] | 0,5 | 2,5 |

## 4. Debug

Switch on the system and make it work. On RDE make a control panel as shown in **figure 5**, with related control buttons to be able to handle all machine behaviors.
Remember to show also the output word to control the launcher axis inverter.

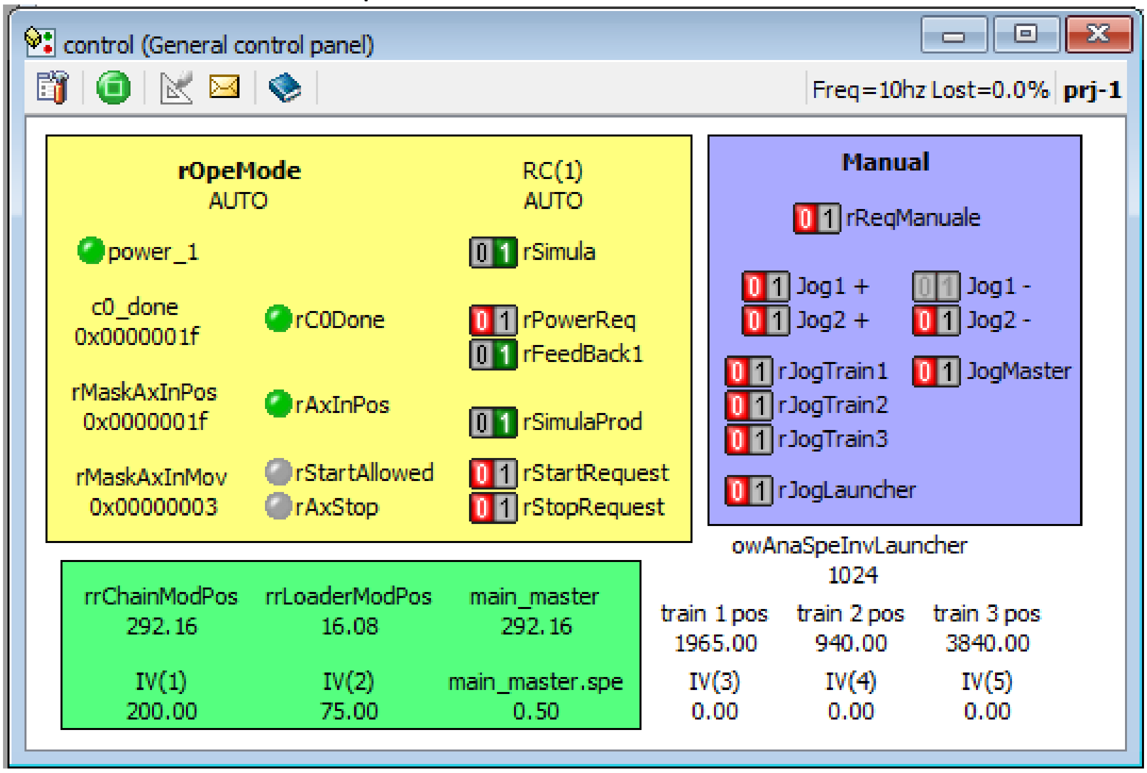


**Figure 5** – Example of control panel

An oscilloscope to show all axes' modulated positions and IV is required.
A 3D panel to see moving axes (**figure 2**) should be made as exercise. You can import a predefined 3D panel for trains from RDE sample panels store, in OneDrive.

## 5. Video

You can see attached video to well understand the system and how it must work: during the first minute the user is moving all axes in jog mode. Then, start button is pressed, so machine will perform posit and then go to automatic. In this video the machine works at a speed of 40 ppm. Sometimes production rate is increased (product wait time set at 60%) or decreased (wait time at 120% or 130%). Few seconds before 2 minutes you can see that slicer production is stopped, then enabled again after some time.
Then, production becomes very slow, with waiting time at 150%. You can see with your eyes the different frequency of product detected and loaded on the trains, shown in the 3D panel with a yellow disc on the launcher loading position.
During all this time and production rate changes, but step trains always feed the loader and chain, performing a very good synchronous production without missing products.