

A03 — Variable-speed conveyor belt

FMU model and guide for PLC programming

Version 1.0

Student exercise — Homa Virtual Plant (Model-in-the-Loop)

Homa inženjering

www.homa.hr · info@homa.hr

© 2026 Homa inženjering. All rights reserved.

Contents

1. Exercise objective.....	3
2. Process description (what the model represents).....	3
3. FMU interface — signals.....	3
3.1. FMU inputs.....	3
3.2. FMU outputs (measurements the PLC reads).....	3
4. What the PLC must achieve (control objectives).....	4
5. Suggested control structure (guideline).....	4
6. Recommended sample times.....	4
7. Test scenarios and acceptance criteria.....	4
8. Integration into Homa Virtual Plant.....	5
9. Notes on the model.....	5

1. Exercise objective

You are given an **FMU model** representing a variable-speed (VFD) conveyor belt. Your task is, through **Homa Virtual Plant** to connect a PLC to this model and write a **PLC program that controls the process** according to the given objectives.

This is a **Model-in-the-Loop (MIL)** setup: the model mimics a real plant, your PLC drives it, and you develop and test your program on a virtual system. The model is “open” — it contains no controller of its own; **all control is performed by your PLC** through the model inputs.

2. Process description (what the model represents)

The variable-speed belt controls the **material flow** via a frequency converter (VFD). Belt load (u_load_estimate) and inflow affect the flow and the motor power.

Available / in the process:

- **VFD drive:** speed/flow is commanded via u_speed_sp.
- **Load:** y_belt_load and y_motor_power rise with load; overload causes a **jam**.

3. FMU interface — signals

This is the most important part. Pay attention to the signal direction:

Signal direction: what is an **INPUT** for the FMU is an **OUTPUT** (command) for your PLC, and vice versa. The actuators your PLC **writes**; the measurements your PLC **reads**.

3.1. FMU inputs

a) Actuators — controlled by your PLC (PLC outputs):

Variable (FMU)	Type	Unit	Range	Description
u_enable	Real (0/1)	–	0 or 1	Enable
u_speed_sp	Real	–	0.0 – 1.0	Commanded speed / VFD (for flow control)
u_reset_jam	Real (0/1)	–	0 or 1	Jam reset

b) Disturbances / scenario — the environment, NOT controlled by your PLC:

Variable (FMU)	Type	Unit	Range	Description
u_material_in	Real	kg/s	0 – ~12	Material inflow
u_load_estimate	Real	–	0.0 – 1.0	Belt load estimate

You set the disturbances on the **PLC HMI** (as a parameter / simulated sensor), and the PLC forwards them to the FMU. They are not an output of your controller.

3.2. FMU outputs (measurements the PLC reads)

Variable (FMU)	Type	Unit	Range	Description
y_speed_norm	Real	–	0.0 – 1.0	Normalized speed
y_material_out	Real	kg/s	0 – ~12	Material discharge

Variable (FMU)	Type	Unit	Range	Description
y_motor_power	Real	kW	~2 – 40	Motor power
y_belt_load	Real	–	0 – 1.5	Belt load
y_jam	Real (0/1)	–	0 or 1	Jam

4. What the PLC must achieve (control objectives)

Your PLC program must:

1. Control the flow (y_material_out) to a target value by varying u_speed_sp (VFD).
2. Limit load/power (avoid overload and jam).
3. Jam detection and safe stop/reset.
4. Start and stop sequences.

The objectives are given; **you design the control method yourself** (a suggested structure is provided below as an aid).

5. Suggested control structure (guideline)

Flow loop (PI):

- Compare target flow with y_material_out → u_speed_sp (VFD).

Load monitoring:

- If y_belt_load / y_motor_power are too high → reduce speed or trip.

Jam and sequences:

- Load above the threshold for a long time → alarm, u_speed_sp = 0; reset as in A01.

6. Recommended sample times

Layer	Recommended cycle
Drive control (VFD)	20 – 100 ms
Material-flow logic	100 – 500 ms
FMU co-simulation communication step	~0.1 s

7. Test scenarios and acceptance criteria

Scenario	How to trigger	Expected PLC behaviour	Pass criterion
Load ramp	u_load_estimate rises	adjust the speed	flow stays stable
Overload	u_load_estimate / inflow high	reduce speed / trip	no persistent jam
No inflow	u_material_in = 0	belt empty, low power	no false alarm
Speed saturation	demand flow above maximum	speed at the limit	stable, no windup
Restart after a jam	u_reset_jam after the load drops	controlled restart	safe restart

General criteria: controlled variables within limits, no instability / excessive cycling, correct response to faults.

8. Integration into Homa Virtual Plant

5. Load `Plant.fmu` (the compiled model) into Homa VP.
6. Map the FMU variables: **actuators** (`u_*`) as PLC outputs, **measurements** (`y_*`) as inputs.
7. Disturbances (`u_material_in`, `u_load_estimate`) are set on the HMI as operating conditions and tests; the PLC forwards them to the FMU.
8. Select a communication channel (e.g. OPC UA / S7) and run the closed loop.

The model is configured for **StopTime 600 s** (Interval 1 s). Real-time pacing is driven by Homa VP via a speed factor.

9. Notes on the model

Important: the model is a simplified starter plant for developing and testing PLC logic — **it is not a validated industrial model**. The parameters are illustrative; calibrate them before real use.

- The dynamics are approximate; the purpose is a control exercise, not exact process analysis.
- `y_jam` is a simulated fault (jam) on overload.