

B05 — PV inverter (averaged model)

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).....	4
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 grid-connected PV inverter (averaged model) with supervisory P/Q control. 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)

An averaged model of a single-phase grid-connected PV inverter. The (supervisory) PLC commands active and reactive power within P/Q limits (apparent-power circle) and handles connection; the fast PWM/current loops stay in the inverter. Irradiance and temperature are disturbances.

Available / in the process:

- `u_active_power_cmd` / `u_reactive_power_cmd`: P and Q demands (supervisory control).
- `u_enable`: enable; a grid permissive (GridOK) is also intended in the PLC.
- The model protects the DC voltage and temperature (`y_fault`); the fast loops are in the inverter.

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
<code>u_active_power_cmd</code>	Real	kW	0 – 100	Active-power demand
<code>u_reactive_power_cmd</code>	Real	kvar	-50 – 50	Reactive-power demand
<code>u_enable</code>	Real (0/1)	–	0 or 1	Enable

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

Variable (FMU)	Type	Unit	Range	Description
<code>u_irradiance</code>	Real	W/m ²	0 – ~1000	Irradiance
<code>u_module_temperature</code>	Real	K	~270 – 330	Module temperature

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_available_power	Real	kW	0 – ~100	Available power
y_pac	Real	kW	0 – 100	AC active power
y_qac	Real	kvar	-50 – 50	AC reactive power
y_vdc	Real	V	~500 – 820	DC voltage
y_iac	Real	A	0 – ...	AC current (rms @ 230 V)
y_temperature	Real	°C	~25 – 80	Inverter temperature
y_fault	Real (0/1)	–	0 or 1	Fault (DC overvoltage / overheating)

4. What the PLC must achieve (control objectives)

Your PLC program must:

1. Command P (u_active_power_cmd) within the available power (y_available_power).
2. Command Q (u_reactive_power_cmd) within the P/Q circle (apparent power).
3. Connect only with a grid permissive; react to y_fault.
4. Thermal derating at high temperature.

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

5. Suggested control structure (guideline)

Supervisory control (100 ms – 1 s):

- Connection sequence: u_enable & GridOK & no fault → run.
- P/Q limiter: respect y_available_power and the apparent power (circle).

Protection:

- Monitor y_fault (y_vdc, y_temperature); thermal derating; safe disconnect.
- The fast current/PWM loops stay in the inverter (not part of the exercise).

6. Recommended sample times

Layer	Recommended cycle
Supervisory (P/Q, sequence)	100 ms – 1 s
Protection / derating	~100 ms
FMU co-simulation communication step	~0.1 – 1 s

7. Test scenarios and acceptance criteria

Scenario	How to trigger	Expected PLC behaviour	Pass criterion
Grid unavailable	GridOK = 0	prohibit operation	P/Q to 0
P/Q limit	demand beyond the circle	limit Q	within the apparent power
Irradiance step	u_irradiance step	track the available	y_pac ≤ available
Thermal derating	high temperature	reduce power	no overheating

Scenario	How to trigger	Expected PLC behaviour	Pass criterion
DC overvoltage	y_vdc rises	fault and disconnect	safe interruption

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_irradiance, u_module_temperature) 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.
- The fast current and PWM loops stay in the inverter; here only the supervisory P/Q control is done.