

A04 — Reactive absorber / flue-gas scrubber

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.....	4
9. Notes on the model.....	5

1. Exercise objective

You are given an **FMU model** representing a reactive absorber (scrubber) for removing SO₂ from flue gas. 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 scrubber removes SO₂ from the flue gas using a reagent. Efficiency depends on the reagent dose and circulation; the sump level and differential pressure must be kept within limits.

Available / in the process:

- u_reagent_flow_cmd: reagent dose (main influence on SO₂ removal).
- u_circulation_cmd: circulation (and protection at high Δp).
- u_drain_cmd: drain for sump-level control.

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_reagent_flow_cmd	Real	–	0.0 – 1.0	Reagent flow
u_circulation_cmd	Real	–	0.0 – 1.0	Circulation
u_drain_cmd	Real	–	0.0 – 1.0	Drain

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

Variable (FMU)	Type	Unit	Range	Description
u_flue_gas_flow	Real	kg/s	0 – ~10	Flue-gas flow
u_so2_in	Real	ppm	0 – ~2000	Inlet SO ₂

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_so2_out	Real	ppm	0 – ...	Outlet SO ₂

Variable (FMU)	Type	Unit	Range	Description
y_removal_efficiency	Real	-	0 – 1	Removal efficiency
y_sump_level	Real	-	0 – 1.2	Sump level
y_differential_pressure	Real	kPa	~0.5 – ...	Differential pressure

4. What the PLC must achieve (control objectives)

Your PLC program must:

1. Keep y_so2_out below the limit (reagent control).
2. Keep the sump level (y_sump_level) at the setpoint (drain).
3. Protect against high Δp (u_circulation_cmd) and pump trip.

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

5. Suggested control structure (guideline)

Control (0.5 – 2 s gas/reagent; 1 – 5 s level):

- SO₂ PI: y_so2_out vs limit → u_reagent_flow_cmd.
- Level PI: y_sump_level vs setpoint → u_drain_cmd.

Protection:

- High- Δp : if y_differential_pressure is too high → u_circulation_cmd / trip.

6. Recommended sample times

Layer	Recommended cycle
Gas / reagent	0.5 – 2 s
Level and supervisory	1 – 5 s
FMU co-simulation communication step	~1 s

7. Test scenarios and acceptance criteria

Scenario	How to trigger	Expected PLC behaviour	Pass criterion
SO ₂ step	u_so2_in rises	increase reagent	y_so2_out within the limit
Reagent loss	u_reagent_flow_cmd = 0	alarm, response	loss detected
Pump trip	circulation drops	protective mode	safe operation
High level	liquid inflow high	drain responds	y_sump_level within the limit
High Δp	u_flue_gas_flow high	protection at Δp	no overload

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

8. Integration into Homa Virtual Plant

4. Load `Plant.fmu` (the compiled model) into Homa VP.
5. Map the FMU variables: **actuators** (`u_*`) as PLC outputs, **measurements** (`y_*`) as inputs.
6. Disturbances (`u_flue_gas_flow`, `u_so2_in`) are set on the HMI as operating conditions and tests; the PLC forwards them to the FMU.
7. 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.