

A02 — Low-order MSW grate 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).....	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 low-order (identified) model of a municipal solid waste (MSW) incinerator for steam-load 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)

This is an **identified (black-box)** low-order model from a scientific paper (arXiv:2401.05221). It predicts the normalized steam load from the commanded load, primary-air temperature and flue-gas measurements. The signals are normalized (dimensionless).

Available / in the process:

- `u_steam_load_sp`: commanded steam load (main manipulated variable).
- `u_primary_air_temp`: primary-air temperature / air trim.
- Flue-gas measurements (`u_flue_h2o`, `u_dry_flue_co2`, `u_dry_flue_o2`) are the scenario.

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_steam_load_sp</code>	Real	–	0.60 – 1.05	Commanded norm. steam load
<code>u_primary_air_temp</code>	Real	–	~1.0	Norm. primary-air temperature / air trim

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

Variable (FMU)	Type	Unit	Range	Description
<code>u_flue_h2o</code>	Real	–	~0.14	Norm. H ₂ O in flue gas
<code>u_dry_flue_co2</code>	Real	–	~0.11	Norm. CO ₂ (dry flue gas)
<code>u_dry_flue_o2</code>	Real	–	~0.08	Norm. O ₂ (dry flue gas)

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_steam_load_norm	Real	-	~0.6 – 1.05	Norm. steam load (main PV)
y_steam_power_mw	Real	MW	~20 – 34	Steam power

4. What the PLC must achieve (control objectives)

Your PLC program must:

1. Control the steam load (y_steam_load_norm) to a setpoint by varying u_steam_load_sp.
2. Trim the primary air (u_primary_air_temp) according to the measured O₂.
3. Limit ramps and saturation; bumpless manual/automatic.

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 loops (0.5 – 2 s):

- Steam-load PI: y_steam_load_norm vs setpoint → u_steam_load_sp.
- O₂ trim PI: measured u_dry_flue_o2 → correction u_primary_air_temp.

Supervisory (5 – 30 s):

- Ramps, limits, bumpless manual/automatic transfer.

6. Recommended sample times

Layer	Recommended cycle
Control loops	0.5 – 2 s
Supervisory	5 – 30 s
FMU co-simulation communication step	~1 s

The model is slow (large time constants), so a 1 s step is sufficient.

7. Test scenarios and acceptance criteria

Scenario	How to trigger	Expected PLC behaviour	Pass criterion
Steam setpoint step	change the setpoint	smooth tracking	y_steam_load_norm within band
Sensor offset	bias on the measured O ₂	trim robustness	stable
Wet fuel	change u_flue_h2o	compensation	steam stays stable
Saturation	demand beyond limits	limits hold	no windup
Manual/automatic	switch mode	bumpless transfer	no bump

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_h2o`, `u_dry_flue_co2`, `u_dry_flue_o2`) 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.
- The model is identified from a paper (normalized signals); check the scaling per [arXiv:2401.05221](https://arxiv.org/abs/2401.05221) before real use.