

A07 — Sludge buffer and thickener

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 sludge buffer/thickener in wastewater treatment (level and solids 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)

The tank receives sludge; a pump draws it off, and polymer improves thickening. The PLC controls the level (pump), doses polymer according to solids and signals a high level. Inflow and inflow solids are disturbances.

Available / in the process:

- `u_sludge_pump_cmd`: sludge pump (level and outlet-flow control).
- `u_polymer_cmd`: polymer dosing (thickening).
- Duty/standby pump operation is intended (in the PLC).

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_sludge_pump_cmd</code>	Real	–	0.0 – 1.0	Sludge pump
<code>u_polymer_cmd</code>	Real	–	0.0 – 1.0	Polymer dosing

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

Variable (FMU)	Type	Unit	Range	Description
<code>u_influent_flow</code>	Real	m ³ /h	0 – ...	Inflow
<code>u_influent_solids</code>	Real	kg/m ³	0 – ...	Inflow solids

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
<code>y_tank_level</code>	Real	–	0 – 1.2	Tank level
<code>y_solids_inventor</code>	Real	kg	≥ 0	Solids inventory

Variable (FMU)	Type	Unit	Range	Description
y				
y_sludge_out_flow	Real	m ³ /h	0 – 40	Sludge outlet flow
y_outlet_solids	Real	kg/m ³	0 – ...	Outlet solids
y_high_level	Real (0/1)	–	0 or 1	High level

4. What the PLC must achieve (control objectives)

Your PLC program must:

1. Keep the level (y_tank_level) at the setpoint via the pump, without overflow.
2. Dose polymer according to solids (u_influent_solids).
3. Duty/standby pump changeover; high-level alarm.

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 pump/level; 5 – 30 s solids):

- Level PI: y_tank_level vs setpoint → u_sludge_pump_cmd.
- Polymer proportional to solids → u_polymer_cmd.

Sequences:

- Duty/standby pump changeover; alarm at y_high_level = 1.

6. Recommended sample times

Layer	Recommended cycle
Pumps / level	0.5 – 2 s
Solids control	5 – 30 s
FMU co-simulation communication step	~1 s

7. Test scenarios and acceptance criteria

Scenario	How to trigger	Expected PLC behaviour	Pass criterion
High inflow	u_influent_flow rises	increase pumping	no overflow
Pump trip	duty pump drops	switch to standby	stable outlet
High solids	u_influent_solids rises	more polymer	outlet solids within the limit
Low level	inflow drops	reduce pumping	no dry running
Duty/standby	pump swap	smooth changeover	no interruption

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_influent_flow`, `u_influent_solids`) 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.
- Duty/standby pumps and alarms are implemented in the PLC (the model provides level, solids and flow).