



Write a self-contained 61131-3 structured text program, not a function block, to implement cascade control for the pressure in an oil refinery. The primary loop controls the pressure in the vessel, while the secondary loop controls the flow of oil into the vessel. The output of the primary loop is used as the setpoint for the secondary loop.



Below is a self-contained IEC 61131-3 Structured Text program implementing a cascade control system for pressure in an oil refinery. The primary loop controls the pressure in the vessel, and the secondary loop controls the flow of oil into the vessel. The output of the primary loop is used as the setpoint for the secondary loop.

iecst Copy code

```
PROGRAM CascadeControl
VAR
    // Primary loop variables
    PV1: REAL; // Process variable: vessel pressure
    SP1: REAL; // Setpoint: target pressure
    OP1: REAL; // Output: secondary loop setpoint
    Kp1: REAL := 1.0; // Proportional gain
    Ki1: REAL := 0.1; // Integral gain
    Kd1: REAL := 0.05; // Derivative gain
    e1, e1_prev, e1_sum, e1_diff: REAL;

    // Secondary loop variables
    PV2: REAL; // Process variable: flow rate
    SP2: REAL; // Setpoint: target flow rate (OP1)
    OP2: REAL; // Output: control valve position
    Kp2: REAL := 2.0; // Proportional gain
    Ki2: REAL := 0.2; // Integral gain
    Kd2: REAL := 0.1; // Derivative gain
    e2, e2_prev, e2_sum, e2_diff: REAL;

    dt: TIME := t#100ms; // Sample time
    t_last: TIME;
END_VAR

METHOD RunCascadeControl
    // Read current pressure and flow rate values
    PV1 := ReadPressure();
    PV2 := ReadFlowRate();

    // Primary loop: pressure control
    e1 := SP1 - PV1;
    e1_sum := e1_sum + e1 * dt;
    e1_diff := (e1 - e1_prev) / dt;
    OP1 := Kp1 * e1 + Ki1 * e1_sum + Kd1 * e1_diff;
    e1_prev := e1;

    // Limit OP1 to a valid range
    IF OP1 > 100.0 THEN
        OP1 := 100.0;
    ELSIF OP1 < 0.0 THEN
        OP1 := 0.0;
    END_IF;

    // Secondary loop: flow control
    SP2 := OP1;
    e2 := SP2 - PV2;
    e2_sum := e2_sum + e2 * dt;
    e2_diff := (e2 - e2_prev) / dt;
    OP2 := Kp2 * e2 + Ki2 * e2_sum + Kd2 * e2_diff;
    e2_prev := e2;

    // Limit OP2 to a valid range
    IF OP2 > 100.0 THEN
        OP2 := 100.0;
    ELSIF OP2 < 0.0 THEN
        OP2 := 0.0;
    END_IF;

    // Set control valve position
    SetValvePosition(OP2);
END_METHOD

END_PROGRAM
```

Note that you will need to replace the ``ReadPressure``, ``ReadFlowRate``, and ``SetValvePosition`` methods with appropriate code to interface with your specific hardware for reading pressure and flow rate and controlling the valve position.