EPICS Database Exercises

Kay Kasemir

Many slides from Andrew Johnson,
APS/ANL

July 2026

Extend /ics/examples/01_*/first.db

```
# A ramp from 0 to 'limit', were limit
# can be configured via a separate record
record(ao, "$(S):limit")
{
   field(DRVH, "100")
   field(DOL,  "10")
   field(PINI, "YES")
}

record(calc, "$(S):ramp")
{
   field(SCAN, "1 second")
   field(INPA, "$(S):ramp")
   field(INPB, "$(S):limit")
   field(CALC, "A<B ? A+1 : 0")
}
```

1. Copy the example:
   cp first.db exercise.db
2. Add a “$(S):step” record and use it in the CALC to allow stepping the ramp in increments between 0.1 and 5.
3. Create a display for all 3 records.
4. Make the “..ramp” display units of “a.u.” and have it show 2 digits after the decimal point
5. Add a widget to the display that allows controlling the rate at which the logic processes
6. Configure the “..ramp” to generate an alarm when the value is above 8
7. Configure the “..ramp” to only send values to an archive when the value changes by 2 or more
8. Add an analog output record which, when processed, resets the “..:limit” to 10. Add a button to the display which triggers this reset.

Binary records

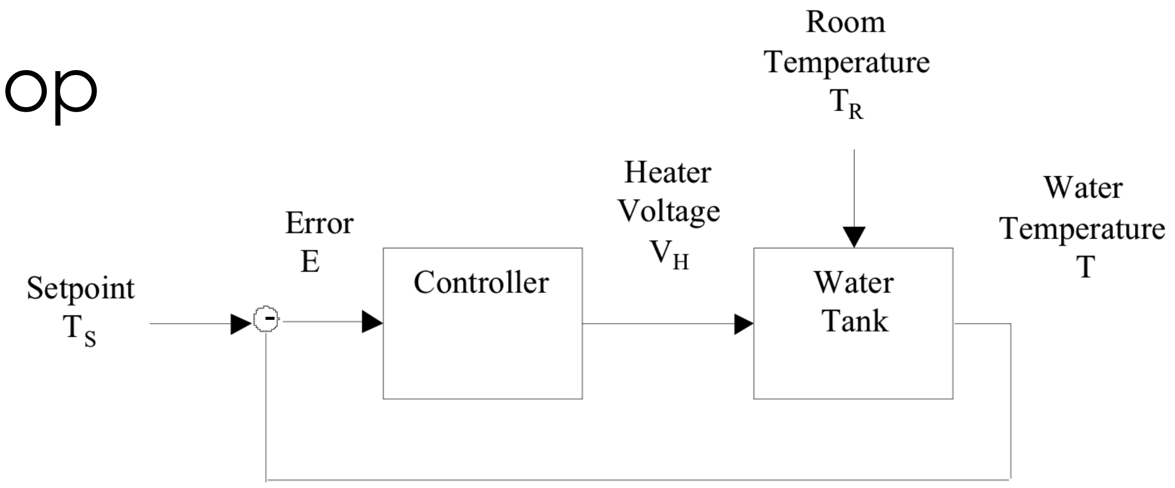
1. Create a BO with values “Normal” and “Doubled”, add to display
2. Use in the “..:ramp” to double the effective step size
3. Configure the BO such that when setting it to “Doubled”, it will revert to “Normal” after 5 seconds

In the following, we ask you to study a larger example

- Read the database files
- Understand what each record does
- Know the meaning of each field
- Get a feeling for the effect of "P" and "I" gains in a PID controller

Heater Control Simulation

- Typical control loop



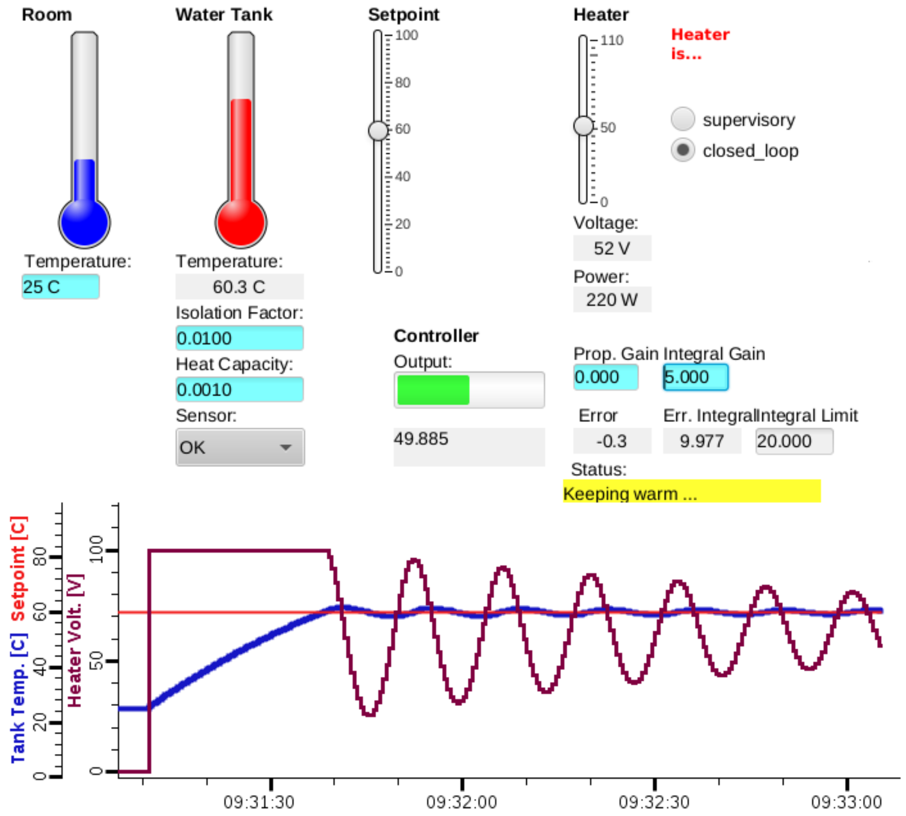
- PID Controller

$$O(n) = K_P\ E(n) + K_I \Sigma_i E(i)\ dT + K_D\ [E(n)-E(n-1)]/dT$$

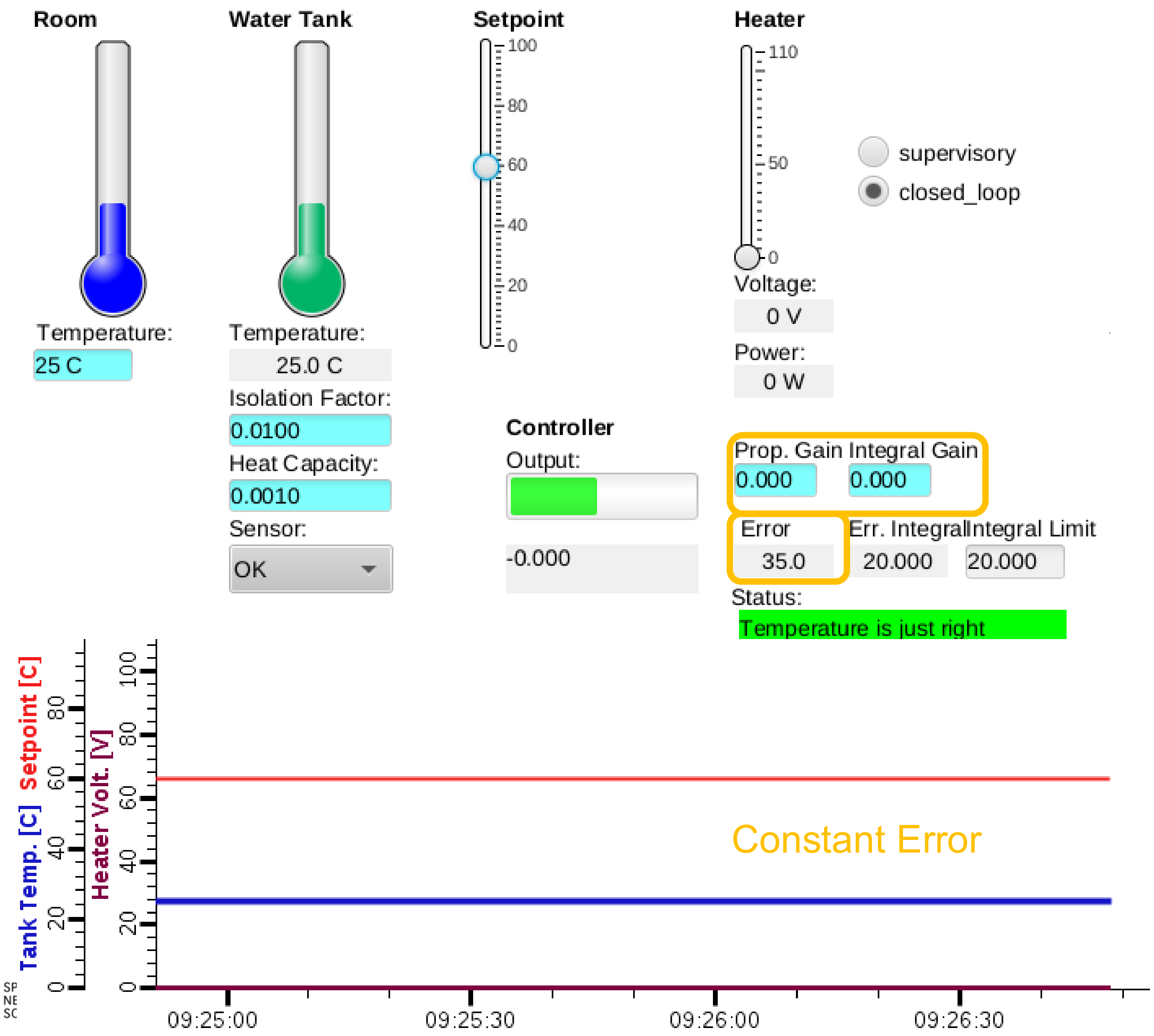
  - Update period dT
  - Error readings $E(n)$
  - Output $O(n)$
  - Proportional Gain K_P, Integral K_I, Derivative K_D

Study the "fishtank" example

1. Go to cd /ics/examples/fishtank
2. Start IOC: ./st.cmd
3. Open heater.bob

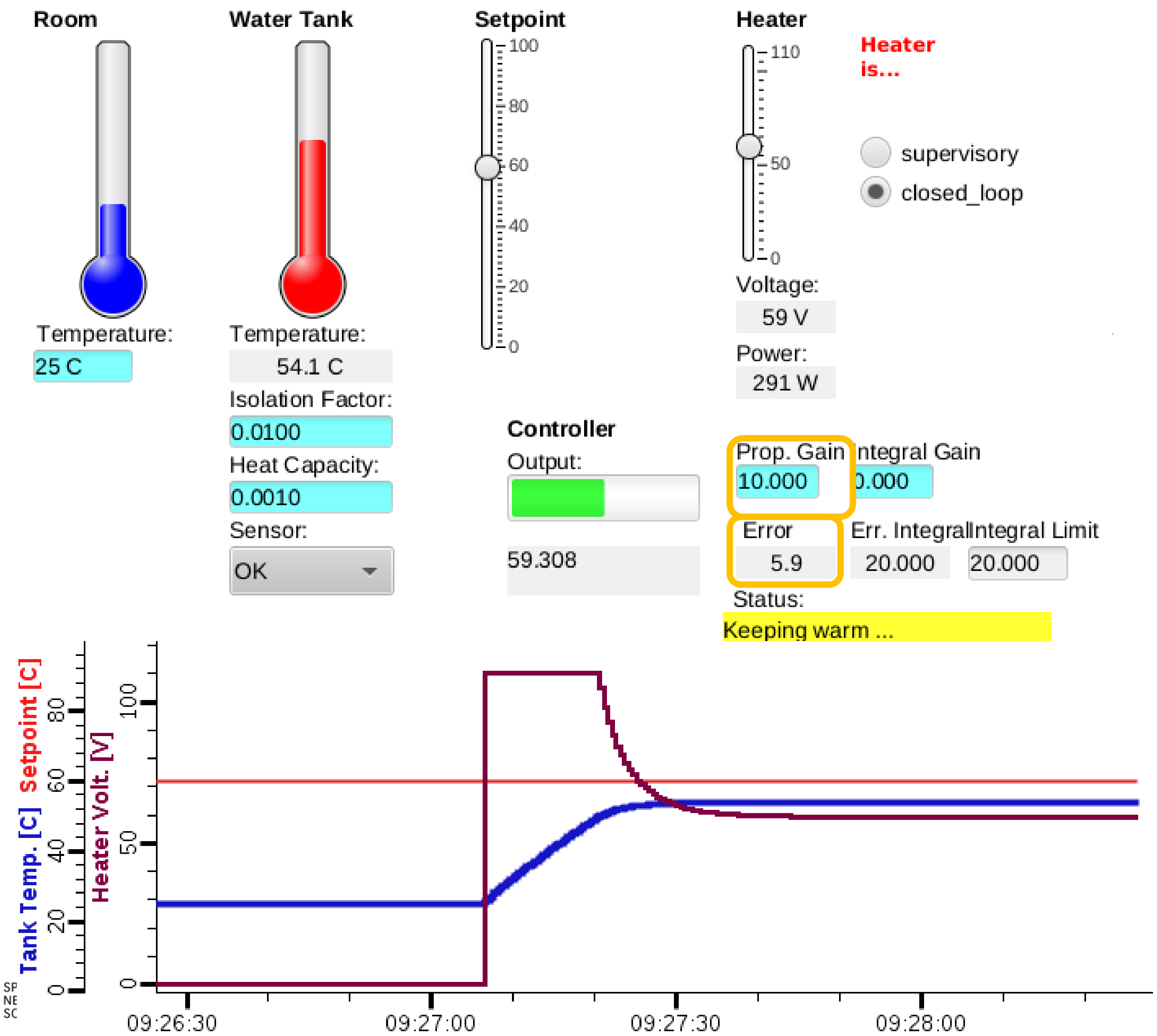


No Control



1. Set the Prop and Int. gain to zero
2. Verify that tank temperature ignores the Setpoint

Only Proportional Control

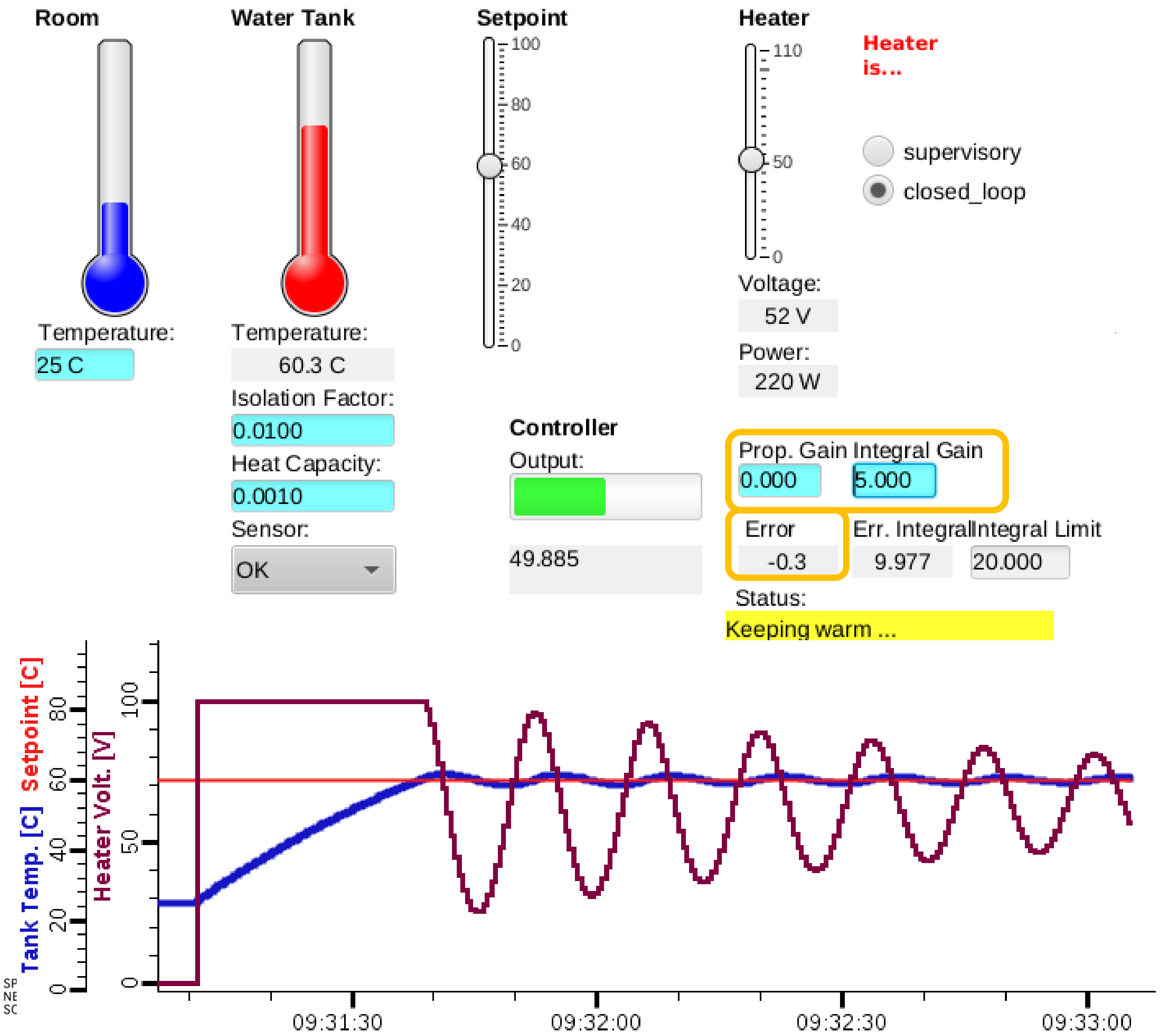


1. Set the Prop. gain
2. Verify that tank temperature reacts to setpoint, but doesn't quite reach it

Why is proportional gain insufficient for heater control?

Small residual Error

Only Integral Control



1. Set only the Integral gain
2. Verify that tank temperature reacts to setpoint. Can you find a good value for the int. gain?

Eventually, no Error!

.. but Ringing

Study database: User Inputs to Simulation

- Note use of Macros
- Note use of Analog *Output* for user *Input* because of DRVL/DRVL

```
record(ao, "$(user):room")
{
    field(DESC, "Room Temperature")
    field(EGU, "C")
    field(HOPR, "40")
    field(LOPR, "0")
    field(DRVL, "0")
    field(DRVH, "40")
    field(DOL, "25")
    field(PINI, "YES")
}
```

```
record(ao, "$(user):setpoint")
{
    field(DESC, "Temperature Setpoint")
    field(EGU, "C")
    field(HOPR, "0")
    field(LOPR, "100")
    field(DRVL, "0")
    field(DRVH, "100")
    field(PREC, "1")
    field(DOL,  "30")
    field(PINI, "YES")
}
```

Simulated Tank Temperature

```
# supervisory: user can adjust voltage
# closed_loop: PID (in separate control.db) sets voltage
# When PID is INVALID, go back to 0 voltage
record(ao, "$(user):heat_V")
{
    field(DESC, "Heater Voltage")
    field(EGU, "V")
    field(DRVL,"0")
    field(DRVH,"110")
    field(DOL, "$(user):PID MS")
    field(OMSL,"closed_loop")
    field(IVOA, "Set output to IVOV")
    field(IVOV, "0")
}

# ~1100 Watt heater when run with 110V:
# P = U I = U^2 / R,  R~12 Ohm
record(calc, "$(user):heat_Pwr")
{
    field(DESC, "Heater Power")
    field(EGU, "W")
    field(INPA, "$(user):heat_V PP NMS")
    field(CALC, "A*A/12.1")
}
```

```
# Every second, calculate new temperature
# based on current temperature,
# room temperature and heater
#
# A - current temperature
# B - room temperature
# C - heater power
# D - isolation factor (water <-> room)
# E - heat capacity (would really depend on water volume)
#
# Very roughly with
# T(n+1) = T(n) + [Troom-T(n)]*Isolation_factor
#          + heater_pwr * heat_capacity
record(calc, "$(user):tank_clc")
{
    field(DESC,"Water Tank Simulation")
    field(SCAN,"1 second")
    field(INPA,"$(user):tank_clc.VAL")
    field(INPB,"$(user):room")
    field(INPC,"$(user):heat_Pwr PP NMS")
    field(INPD,"0.01")
    field(INPE,"0.001")
    field(CALC,"A+(B-A)*D+C*E")
    field(FLNK,"$(user):tank")
}
```

- What causes all these records to process?

PID (without D) Computation

```
# Error computation's SCAN drives the rest
record(calc, "$(user):error")
{
    field(DESC, "Temperature Error")
    field(SCAN, "1 second")
    field(INPA, "$(user):setpoint")
    field(INPB, "$(user):tank MS")
    field(CALC, "A-B")
    field(PREC, "1")
    field(FLNK, "$(user):integral")
}
# Integrate error (A) but assert that
# it stays within limits (C)
record(calc, "$(user):integral")
{
    field(DESC, "Integrate Error for PID")
    field(PREC, "3")
    field(INPA, "$(user):error PP MS")
    field(INPB, "$(user):integral")
    field(INPC, "20.0")
    field(CALC, "(B+A>C)?C:(B+A<-C)?(-C):(B+A)")
    field(FLNK, "$(user):PID")
}
```

```
# PID (PI) computation of new output
# A - Kp
# B - error
# C - Ki
# D - error integral
record(calc, "$(user):PID")
{
    field(DESC, "Water Tank PID")
    field(PREC, "3")
    field(LOPR, "0")
    field(HOPR, "110")
    field(INPA, "10.0")
    field(INPB, "$(user):error MS")
    field(INPC, "5.0")
    field(INPD, "$(user):integral MS")
    field(CALC, "A*B+C*D")
}
```

Add Differential Control

- **”Patch” database**
- **How does it change the processing of records?**

```
# Update 'error':
# Make passive (now triggered by new 'error_diff'
record(calc, "$(user):error")
{
    field(SCAN, "Passive")
}

# New computation of change in error triggers
# the error computation
record(calc, "$(user):error_diff")
{
    field(DESC, "Temperature Difference")
    field(SCAN, ".5 second")
    field(INPA, "$(user):error")
    field(INPB, "$(user):error MS PP")
    field(CALC, "(B-A)/0.5")
}

# Every second, calculate new heater voltage via PID (PI)
# A - Kp
# B - error
# C - Ki
# D - error integral
# E - Kd
# F - error differential
record(calc, "$(user):PID")
{
    field(INPE, "0.0")
    field(INPF, "$(user):error_diff MS")
    field(CALC, "A*B+C*D+E*F")
}
```

Adding Differential Control