

PID Regulator

Introduction

StateWORKS is a system to develop and implement control systems [1]. Closed loop regulators belong also to this category. We do not have an RTDB object that realizes the PID regulation algorithm but we offer an Output Function for this purpose.

The technical note gives a short introduction to this technique and shows the implementation details by an example.

PID Regulation

The Output Function realizes the well known calculation algorithm calculating the regulator output:

$$\text{Out} = K_p e + K_i \int e dt + K_d de/dt$$

where:

- e – the error
- K_p - Proportional gain
- K_i – Integral gain
- K_d - Derivative gain
- In a discrete implementation the following formula will be used:
- $\text{Out} = K_p e + K_i \sum e + K_d (e_{\text{last}} - e)$

It is assumed that the calculations are done using a constant time interval, in other words a fixed sampling rate.

OfuRegulator

The actual calculation done by the OfuRegulator function corresponds to previous formulas. The function accesses several RTDB objects, among others:

- Inputs
 - ActValue – the regulated value
 - SetValue – the required value
- Output
 - OutValue – regulator output value
- Parameters
 - K_p – Proportional gain
 - K_i – Integral gain
 - K_d – Differential gain
 - SamplingTime – the sampling value used in the autonomous mode
- Intermediate values
 - Error - the error value
 - Integral – the integral value

The function has two modes: Autonomous and Sampling. In the Autonomous mode the function

samples the input values and calculates the output according to its internal clock defined by a parameter `SamplingTime`.

In the Sampling mode the calling state machine is responsible for the sampling period. Normally, it uses a timer generated clock or just changes of the input `ActValue` which equals the polling rate of the I/O Handler.

An application can specify any number of OFUN Objects which provide access to the `OfuRegulator` function. So, the application can have any number of PID regulators, each with its own set of required objects (input, outputs, intermediate values and parameters).

Example

The implementation of the PID algorithm has been tested using a state machine shown Figure 1.

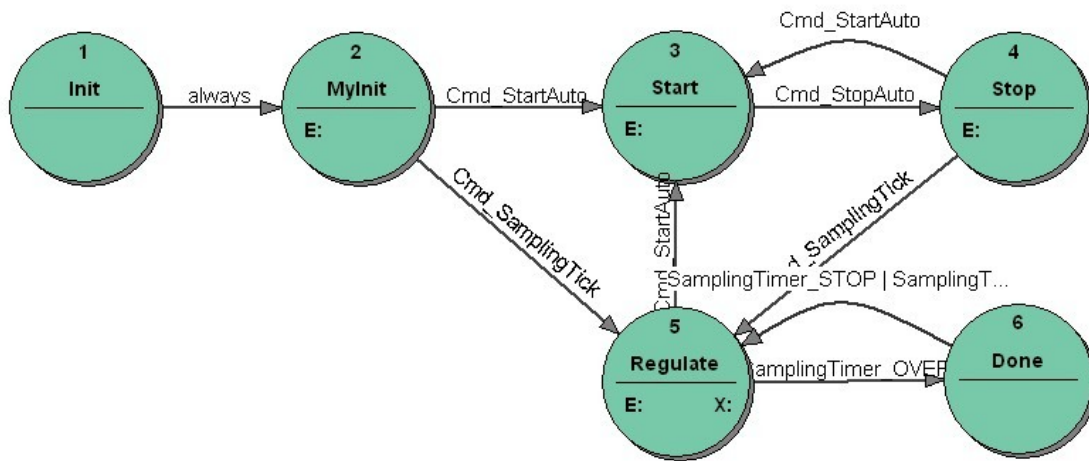


Figure 1: PID Test: State transition diagram

The object has been simulated by a delay of 2 sec. Figure 2 and Figure 3 at the end show two curves illustrating changes of the controlled `ActValue` by different PID parameters. There were no significant differences between Autonomous and Sampling Mode.

Conclusions

The `OfuRegulator` is an integral part of the `StateWORKS` versions Basic and Pro. As a standard solution we offer a `OfuRegulator` function with the Sampling mode. On request we can supply the `OfuRegulator` with the Autonomous mode.

The availability of this PID function means that `StateWORKS` covers a complete range of control tasks.

References

[1] Wagner, F. et al., *Modeling Software with Finite State Machines – A Practical Approach*, Auerbach Publications, New York, 2006.

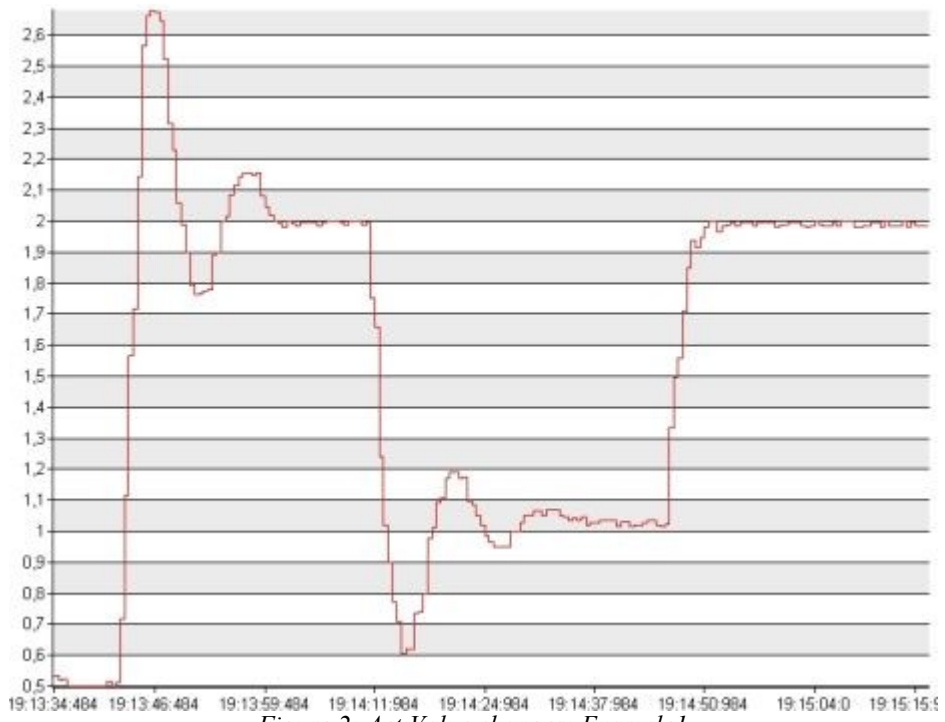


Figure 2: Act Value changes: Example1

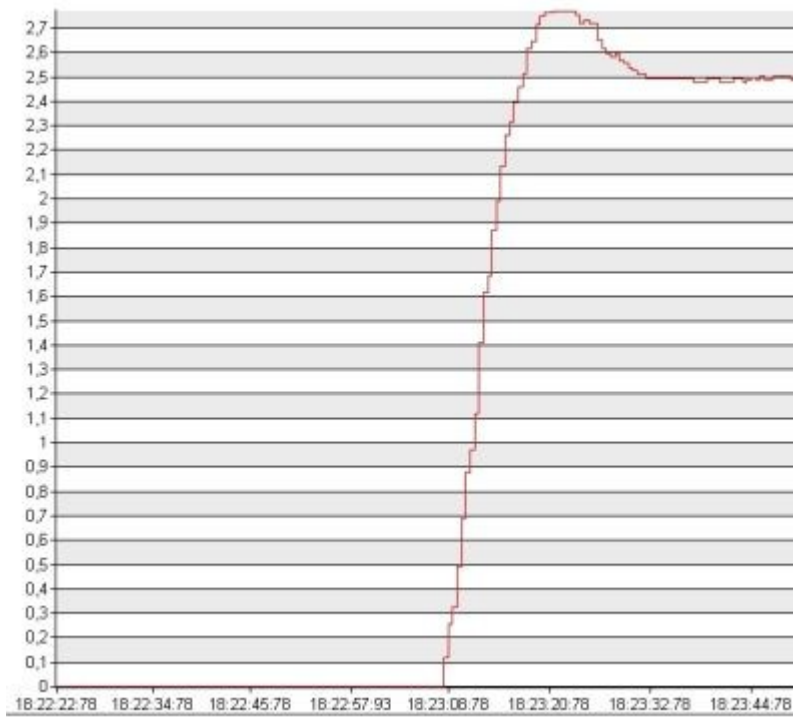


Figure 3: Act Value changes: Example2