CONTROL OF NONLINEAR SYSTEMS

J. Slačka, E. Miklovičová

[a]Slovak University of Technology, Faculty of Electrical Engineering and Information Technology Ilkovičova 3, 812 19 Bratislava, Slovak Republic

Abstract

In real life we often come into contact with nonlinear systems. In this paper we will concentrate on two methods dealing with the problem of nonlinear system control. As the first method we will use the self-tuning regulator and as the second multiple-model control. Both of the methods mentioned above will be verified by computer simulation. As a controlled system a simple cylindrical tank with one drain and one adjustable tributary is chosen. In conclusion we will compare these two methods of nonlinear control and try to discover their strengths and weaknesses.

1 Self-tuning controller

Consider that the process behavior can be described by the following ARX model

$$A(z^{-1})y(k) = B(z^{-1})u(k) + \xi(k) \tag{1.1}$$

where $u(k)$ is the control signal, $y(k)$ the measured output and $\xi(k)$ is white noise. In the control design we will consider that noise has no effect on process ($\xi(k) = 0$).

In self-tuning controllers recursive identification is used to update the parameters of the process model as new plant measurements become available at each sampling period (Fig. 1). Based on the updated model parameters, the controller parameters are recalculated and new value of controller output at time t is obtained.

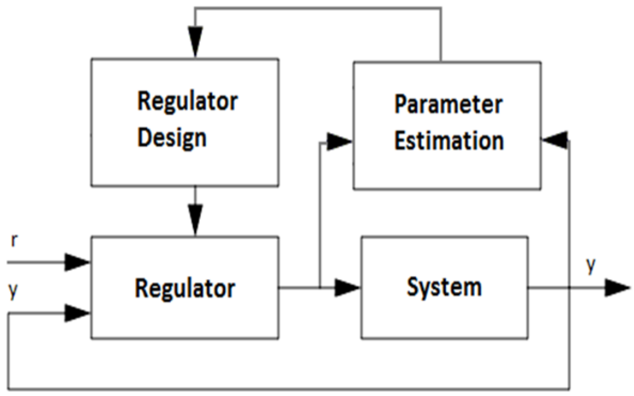

Figure 1: Self-tuning controller

The control law may be designed using the standard pole-placement control approach [1].The control objective is to assign the closed loop poles to pre specified positions. The control law has the form

$$R(z^{-1})u(k) + S(z^{-1})y(k) = T(z^{-1})r(k) \tag{1.2}$$

The polynomials

$$R(z^{-1}) = 1 + r_1 z^{-1} + \cdots + r_{nr} z^{-nr} \tag{1.3}$$

$$S(z^{-1}) = s_0 + s_1 z^{-1} + \cdots + s_{ns} z^{-ns} \tag{1.4}$$

$$T(z^{-1}) = t_0 + t_1 z^{-1} + \cdots + t_{nt} z^{-nt} \tag{1.5}$$

are solutions of the following Diophantine equation

$$P(z^{-1}) = A(z^{-1})R(z^{-1}) + B(z^{-1})S(z^{-1}) \tag{1.6}$$

where $P(z^{-1})$ is the desired closed loop characteristic polynomial

$$P(z^{-1}) = 1 + p_1 z^{-1} + \cdots + p_{np} z^{-np} \tag{1.7}$$

The polynomials $R(z^{-1})$, $S(z^{-1})$, $T(z^{-1})$ need to be calculated in every sampling period. In order to calculate these polynomials we need to continuously identify the process model polynomials $A(z^{-1})$, $B(z^{-1})$, which may change depending on actual working point, using, for example, the recursive least squares algorithm, with exponential weighting [2].

The self-tuning controller Simulink model scheme is shown in Figure 2. As it can be seen, there are two Matlab functions called Regulator and RMNS. The function RMNS calculates ARX model parameters using recursive least squares method and the function Regulator calculates $R(z^{-1})$, $S(z^{-1})$, $T(z^{-1})$ polynomials and the controller output signal.

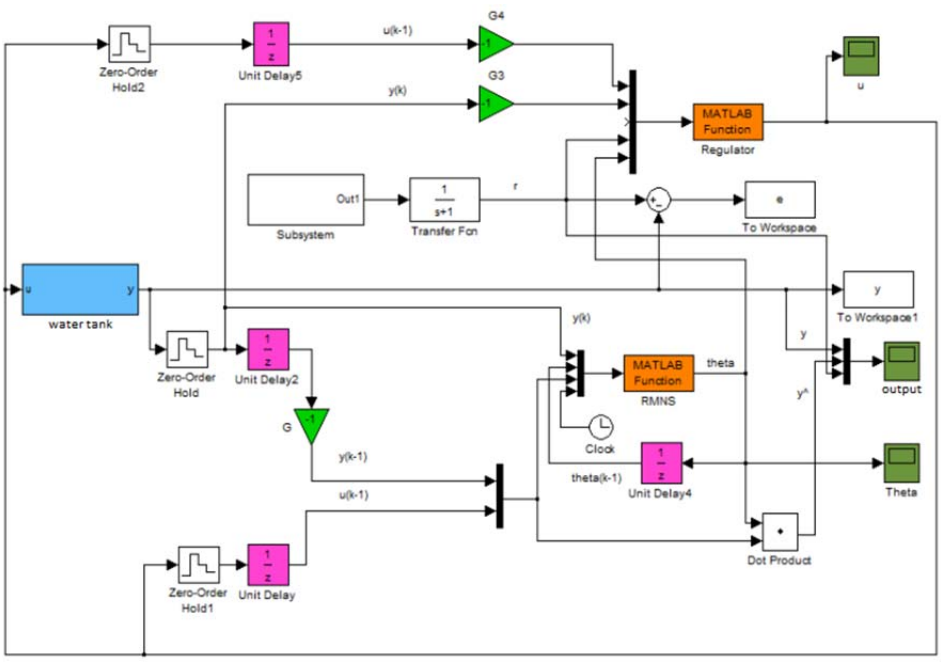

Figure 2: Simulink model of self-tuning controller

2 Multiple-model control

The multiple model control uses different approach. The process working range is divided into several working points. For each working point the process model is identified and the regulator parameters are calculated in advance using the pole placement method mentioned above. The global controller output can be obtained either by the hard switching between the locally valid controllers or by merging the local controller outputs. In our implementation the weighted combination of all linear regulator outputs has been used as shown in Fig. 3.

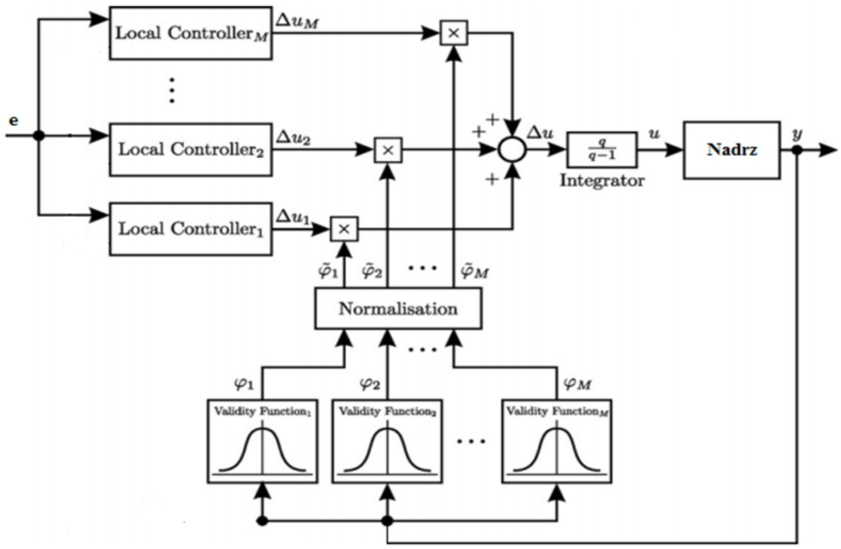

Figure 3: Multiple-model control

The weight for locally valid linear i-th controller is calculated using the following Gaussian function [3]

$$\Delta u^*(k) = \sum_{i=1}^{M} \varphi_i(\Phi(k) u_i(k)) \tag{2.1}$$

$$\varphi_i(\Phi(k)) = e^{\frac{-\frac{1}{2}(\Phi(k)-C_i)^T(\Phi(k)-C_i)}{\sigma_i^{2}}} \tag{2.2}$$

where:

- C_i- is the center of Gaussian function,
- σ–represents the width of the curve,
- $\Phi(k)$–actual working point in k-step.

The weights have to be normalized, because there is more than one regulator. The arithmetic average is used to compute new normalized weights [2]

$$\rho_i\big(\Phi(k)\big) = \frac{\varphi_i(\Phi(k))}{\sum_{j=1}^{M} \varphi_j(\Phi(k))} \tag{2.3}$$

The weights are calculated in every sampling period. The local controller parameters remain without change. Also there is no need to continuously identify the process model as in the self-tuning control. Simulink model representation of the multiple-model control can be found in Fig. 4.

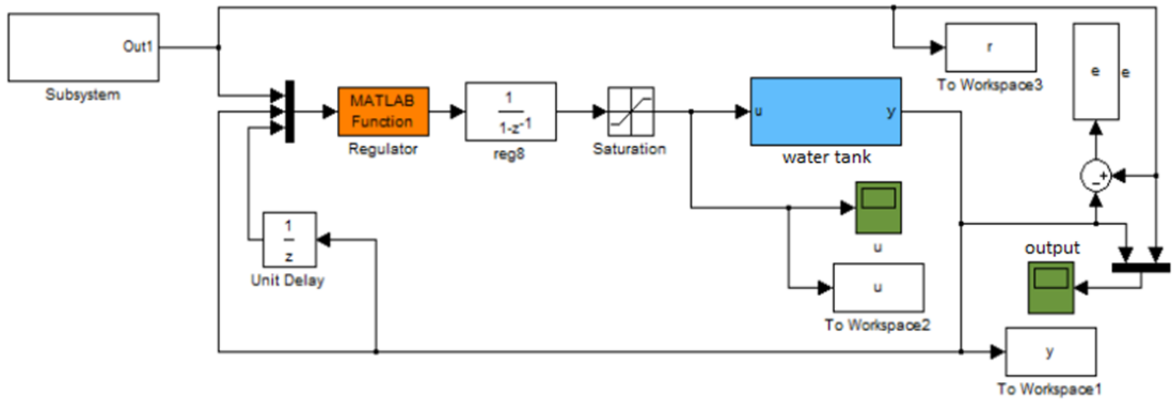

Figure 4: Simulink model of multiple model control

3 Case study

The effectiveness of the above described control strategies has been evaluated and compared in simulations in Matlab-Simulink environment using a simple cylindrical tank depicted in Fig.5.

The tank is a nonlinear system whose time constant and gain vary considerably throughout the operating range. The tank parameters are given in Table 1 and Simulink model is shown in Fig.6. The controlled variable is the liquid height h and the control variable is the inlet flow rate Q_1 with flow range between 0 to 0,5 m^3s^{-1}. The outlet flow rate varies due to liquid height level. Dynamic characteristics are described by following first order differential equation

$$S\frac{dh}{dt} = Q_1 - \mu S\sqrt{2gh} \tag{3.1}$$

Table 1: Water tank parameters

$S = 1\ m^2$	$h_{max} = 2\ m$
$S_0 = 0.1\ m^2$	$\mu = 0.62$
$g = 9.81\ ms^{-2}$	$\rho = 1000\ kgm^{-3}$

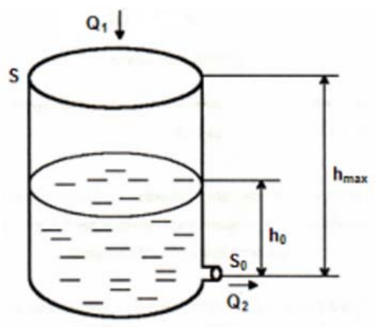

Figure 5: Cylindrical tank

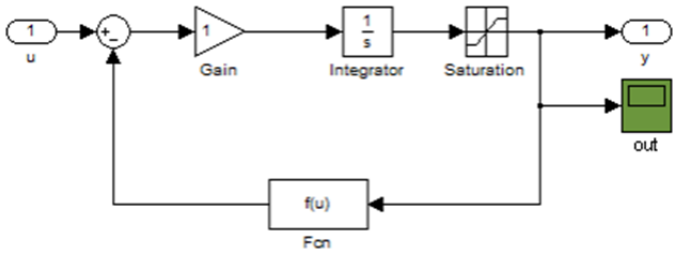

Figure 6: Simulink model

Simulation results of self-tuning controller are shown in Fig. 7. For both simulations as desired closed loop pole -0.4 was selected. In the beginning of simulation self-tuning control needs 0.5-2 seconds to adapt its parameters. After adaptation, the control quality was very good.

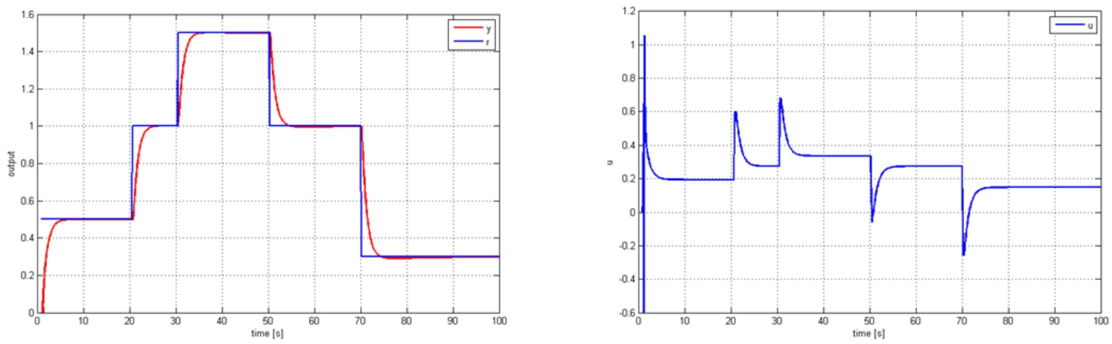

Figure 7: Self tuning control simulation results

Multiple-model control results can be found in Fig. 8. In order to ensure good control system performances across the whole working range the centers of Gaussian functions have been set in every third of the working range as shown in Fig. 9.

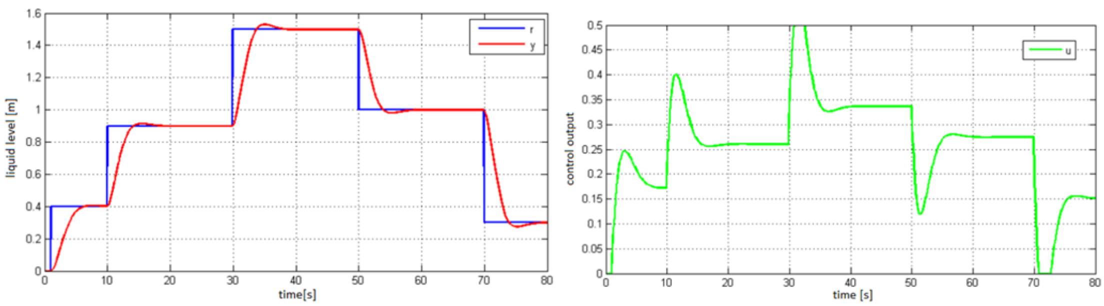

Figure 8: Multiple model control simulation results

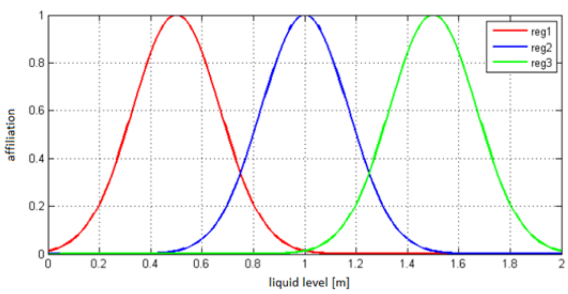

Figure 9: Gaussian functions

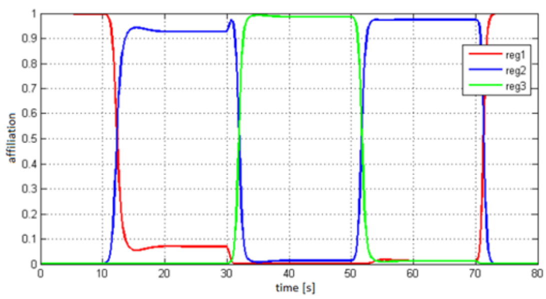

Figure 10: Contribution of regulators

4 Conclusion

Both of the methods mentioned above provided stable control with good quality. The main disadvantage of the self-tuning regulator is the need for very accurate initial values of the process model parameters. Without accurate initial parameters the closed loop system begins to diverge and becomes unstable. Another disadvantage is the computational complexity. In small embedded applications it may cause problems, because of the need to compute the model and regulator parameters in every single sampling period. The main advantage of the multiple model control is the simpler calculation of the controller output.The computational demand depends on the number of operating points.

When we try to compare the control system performances over the whole operational range, the self-tuning regulator is a clear winner. Multiple model controller has good control quality only in range between the designed regulators. When the process working point is out of the pre-calculated regulators, the control quality drops. Also when the process range is wide and the number of local regulators is low, quality of control is declining. Self-tuning regulator has good quality over the whole process range, because of its ability to adapt and to change the regulator parameters in every sampling period.

Both of tested methods worked as expected. In final conclusion we would recommend to use the multi model control for systems with low computing capacity and for processes with known working range. Self-tuning regulator is better suited for processes with unknown or very wide working range.

References

[1] K.J. Åström, B. Wittenmark. *Adaptive control* (2nd ed).Addison-Wesley Publishing Company, (2nd ed). 1995.

[2] L. Ljung. *System identification: Theory for the user*(2nd ed). Prentice Hall PTR, 1999.

[3] G. Gregorčič, G. Lightbody. *Nonlinear model-based control of highly nonlinear processes.* Computers and Chemical Engineering,Vol. 34, pp. 1268-1281, 2010

Acknowledgement

This work has been supported by the Slovak Scientific Grant Agency, Grant No. 1/2256/12.

Juraj Slačka
juraj.slacka@stuba.sk

Eva Miklovičová
eva.miklovicova@stuba.sk