

# ITERATIVE FEEDBACK TUNING APPLIED TO POWER PLANT CONTROL

*H. Procházka, J.M. Legros*

PROSYSTEMY Slovakia, EDF R&D France

## Abstract

**One of the most recent methods for controller tuning and optimization is Iterative Feedback Tuning (IFT). It is capable to optimize parameters of linear digital single-input, single-output controller of general structure (P, PI, PD, PID, transfer function) in closed loop. A software tool applying the IFT method has been developed in Matlab environment. This tool was validated on several examples of power-plant control.**

## 1 Introduction

Industrial processes are commonly controlled by PI/PID controllers and/or similar linear control structures. Generally, these are not optimally tuned. Even if the controllers were tuned in some way at the beginning of process operation, it is beneficial to re-optimize/re-tune these controllers. The process is evolving in time (wearing down, changing of conditions) and the optimization permits to approach once again an optimal performance, i.e. cost-effective, efficient, safe and robust operation.

Different approaches exist for controller optimization/tuning. There are ad-hoc methods, model-based optimization methods, etc. One of the most recent methods is Iterative Feedback Tuning (IFT) [1]. It is capable to optimize parameters of linear digital single-input, single-output controller of general structure (P, PI, PD, PID, transfer function). The main features of the method are:

- The process remains in closed-loop with initial controller during optimization.
- System signals are used for optimization, such as: set-point (reference), measured process variable, control input, measured disturbance.
- Performance criterion is optimized which defines: set-point tracking, disturbance rejection, control action limitation, robustness.
- An iterative step-by-step technique is used. New controller with superior performance is proposed at the end of each iteration step.

A software tool applying the IFT method has been developed in Matlab environment by R&D Department of Electricité de France, Université Catholique de Louvain Belgium, and Prosystemy Slovakia. The tool provides us with an automated interactive (easy-to-use) approach of using the IFT method. The tool was validated on several examples of power-plant control.

The paper is structured as follows. First part introduces IFT method basic theory, its algorithm and some of the method's advanced features. The second part presents the software tool, its essential data sets and graphical user interface. The third part describes one controller tuning example. The paper finishes with conclusion.

## 2 Iterative feedback tuning method

### 2.1 Principle

A one-degree-of-freedom linear discrete-time single-input, single-output controller optimization is considered. A corresponding closed-loop (CL) diagram is presented in Figure 1. The term  $G$  is either a plant model or a real plant with a sample and hold device and the block  $C$  represents the controller to be optimized. The external signals  $r(t)$ ,  $d(t)$ ,  $v(t)$  are an output reference (set-point), a possible external control action input, and a perturbation signal, respectively. The output signal  $y(t)$  (measured variable) and the control action  $u(t)$  are internal closed-loop signals. Considering  $G$  a linear transfer function, the internal closed-loop signals can be described via sensitivity functions as follows:

$$\begin{aligned}
u(t) &= CSr(t) - CSv(t) \\
y(t) &= GCSr(t) + Sv(t) \\
S &= (1 + GC)^{-1}
\end{aligned} \tag{1}$$

The IFT scheme for 1DOF controller optimization considers the closed-loop system as it is shown in Figure 1 but without the control excitation input:  $d(t) = 0$ . IFT minimizes the following criterion function:

$$J(\rho) = E\left[\left(L_y(y(t) - y_d(t))\right)^2\right] + \lambda E\left[\left(L_u u(t)\right)^2\right] \tag{2}$$

In (2),  $\rho = [\rho_1 \ \rho_2 \ \dots]^T$  is a vector of controller coefficients to be optimized, e.g.  $C = C(\rho)$ , and  $E[\ ]$  denotes expectation with respect to all random signals. The first term in the criterion reflects tracking performance specifications with  $y_d(t)$  a desired closed-loop time response and  $y(t)$  an actual output. The second term adds a possible control action energy restriction. The transfer functions  $L_y$  and  $L_u$  are some stable weighting/scaling filters adjusting the criterion and  $\lambda$  is a scaling factor added for the same purpose.

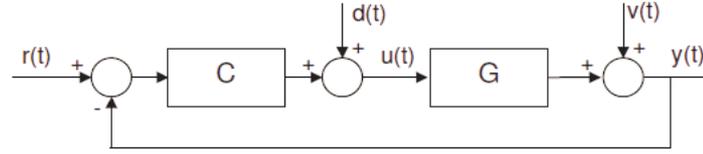


Figure 1: Closed-loop diagram.

For numerical evaluation the criterion (1) has the following form ( $k$  is signal sample index):

$$J_{num}(\rho) = \frac{1}{N} \sum_{k=1}^N \left[ \left( L_y(y(k) - y_d(k)) \right)^2 \right] + \lambda \frac{1}{N} \sum_{k=1}^N \left[ \left( L_u u(k) \right)^2 \right] \tag{3}$$

A minimum of the criterion (2) is searched using a gradient-based local minimization technique such as steepest descent, Gauss-Newton, etc. To use these techniques gradient of the criterion function (2) is needed:

$$\nabla J(\rho) = \begin{bmatrix} \frac{\partial J(\rho)}{\partial \rho_1} \\ \frac{\partial J(\rho)}{\partial \rho_2} \\ \vdots \end{bmatrix} = 2E \left\{ (y(t) - y_d(t)) \left[ \frac{\partial y}{\partial \rho} \right] \right\} + 2\lambda E \left\{ u(t) \left[ \frac{\partial u}{\partial \rho} \right] \right\} \tag{4}$$

where the signal derivatives has the following form:

$$\begin{aligned}
\frac{\partial u}{\partial \rho} &= S \frac{\partial C}{\partial \rho} (Sr(t) - Sv(t)) \\
\frac{\partial y}{\partial \rho} &= GS \frac{\partial C}{\partial \rho} (Sr(t) + Sv(t))
\end{aligned} \tag{5}$$

Notice that in (5) the first terms  $S$  and  $GS$  are sensitivity functions of the closed loop without the term  $C$  (see eq. (1)). The second term  $\frac{\partial C}{\partial \rho}$  is a derivative of the actual controller. Finally let's call the last term  $r_2(t)$ , then  $r_2(t) = (Sr(t) - Sv(t))$ . The signal  $r_2(t)$  can be obtained from a closed loop response as follows (knowing that  $GCS + S = 1$ ):

$$\begin{aligned}
r_2(t) &= r(t) - y(t) \\
r_2(t) &= (1 - GCS)r(t) + Sv(t) \\
r_2(t) &= Sr(t) + Sv(t)
\end{aligned} \tag{6}$$

An estimation of the signal derivatives (5) is generated in the IFT method by applying a special experimental reference  $r_2(t)$  to the closed loop. This will be shown in description of the IFT method procedure below.

## 2.2 Application

The IFT method is an iterative procedure. At the end of each iteration step a new controller is proposed providing superior closed-loop performance (see Figure 2).

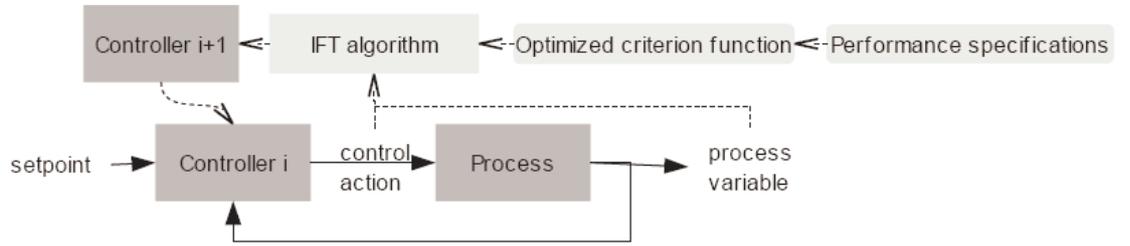


Figure 2: IFT method procedure.

One iteration step can be described by the following procedure:

- 1st real-time experiment. An external excitation signal  $r_1(t)$  (typically one or more setpoint step changes) is applied to the closed-loop system. In this experiment we obtain the following internal closed-loop signals  $y_1(t)$  and  $u_1(t)$  containing weighted sensitivity functions as shown by (1):

$$\begin{aligned}
u_1(t) &= CSr_1(t) - CSv_1(t) \\
y_1(t) &= GCSr_1(t) + Sv_1(t)
\end{aligned} \tag{7}$$

- Construction of the special excitation signal  $r_2(t)$  as it is shown in (6):

$$r_2(t) = r_1(t) - y_1(t) = Sr_1(t) + Sv_1(t)$$

- 2nd real-time experiment. The excitation signal  $r_2(t)$  is applied to the closed loop, in order to obtain an estimate of the signal derivatives (5). These derivatives are necessary to compute the gradient of the criterion (4). The following closed-loop responses are obtained:

$$\begin{aligned}
u_2(t) &= CSr_2(t) - CSv_2(t) = CS(Sr_1(t) + Sv_1(t)) - CSv_2(t) \\
y_2(t) &= GCSr_2(t) + Sv_2(t) = GCS(Sr_1(t) + Sv_1(t)) + Sv_2(t)
\end{aligned} \tag{8}$$

- An estimate of the signal derivatives (5) is obtained by filtering the obtained signals  $u_2(t)$  and  $y_2(t)$  with the filter  $C^{-1} \frac{\partial C}{\partial \rho}$ :

$$\begin{aligned}
est \frac{\partial u}{\partial \rho} &= C^{-1} \frac{\partial C}{\partial \rho} u_2(t) = S \frac{\partial C}{\partial \rho} (Sr_1(t) + Sv_1(t)) - CSv_2(t) \\
est \frac{\partial y}{\partial \rho} &= C^{-1} \frac{\partial C}{\partial \rho} y_2(t) = GS \frac{\partial C}{\partial \rho} (Sr_1(t) + Sv_1(t)) + Sv_2(t)
\end{aligned} \tag{9}$$

- The gradient (4) is estimated using the signal derivatives estimates (9) and closed-loop response (7). The terms  $-CSv_2(t)$  and  $Sv_2(t)$  in (9) produce gradient estimation errors. The gradient estimate has the following form:

$$est\nabla J(\rho) = 2E \left\{ (y(t) - y_d(t)) est \left[ \frac{\partial y}{\partial \rho} \right] \right\} + 2\lambda E \left\{ u(t) est \left[ \frac{\partial u}{\partial \rho} \right] \right\} \quad (10)$$

6. A new vector of controller coefficients is computed by a gradient-based local search method using the gradient estimate (10). A general formula for the local search methods is ( $\gamma_i$  is step size,  $R_i$  is a search direction matrix):

$$\rho_{i+1} = \rho_i - \gamma_i R_i est\nabla J(\rho) \quad (11)$$

### 2.3 Advances of IFT method

The basic IFT method presented in the previous sections is constantly improved, adapted, and modified by different researches [3]. Several adaptations have been developed and validated on example in connection with the software tool development as well. The following adaptations of the method are the most significant that have been produced during the software tool development:

- Robustness measures are introduced to the optimized criterion function [2].
- Procedure for simultaneous optimization of the controller in several operational set-points is developed and implemented.
- Different local optimum search techniques are evaluated and implemented such as Davidon-Fletcher-Powell (DFP) or Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm.
- Simultaneous optimization of the controller and corresponding feed-forward filter is implemented.

## 3 Software tool

An interactive software tool has been developed in Matlab environment. The tool was developed in collaboration between R&D Department of Electricité de France, Université Catholique de Louvain Belgium, and Prosystemy Slovakia. The main goal was to develop a software tool with the following main features:

- The IFT method is implemented.
- Advanced features of the IFT method are implemented as well.
- The tool has a user-friendly graphical user interface.
- A simulink model can be treated.
- A real process can be treated as well.
- The tool has control performance analysis tools.
- The tool has optimization history analysis tool.

A user work-flow diagram for the software tool is shown in Figure 3. Basic specifications are entered first. Next the specifications are semi-automatically converted to IFT method parameters. The conversion is made by the software tool, but user should verify and, if necessary, correct the method parameters. One or more iterative steps (see Section 2.2) are executed subsequently. Finally, obtained results are analyzed to decide on further action.

In the work-flow described above, three sets of data are considered: Specifications, IFT method parameters, and optimization results. The specifications are basic data to be entered by any automation engineer. The IFT method parameters can only be modified by an expert in IFT method. Results can again be treated by any automation engineer. A brief description of these three data sets is below.

Specifications:

- Structure of the optimized controller (PI, PID, ...)

- Initial controller coefficients
- Sampling time of the controller
- Desired tracking performance (rising time, overshoot, ...)
- Process description (Simulink model, signal names, real process...)
- Reference (set-point) signal description
- Number of operating points.

IFT method parameters:

- Number maximal of iterations
- Actual step size of optimization
- Actual matrix  $R_i$
- Scaling filters  $L_y, L_u$ .
- Scaling factor  $\lambda$
- Robust stability terms of the criterion function

Optimization results:

- New computed controller  $C_{i+1}$
- Closed-loop responses obtained in the course of the iteration step
- Criterion function value
- Estimated responses for closed-loop with the new controller (closed-loop sensitivities, step response)

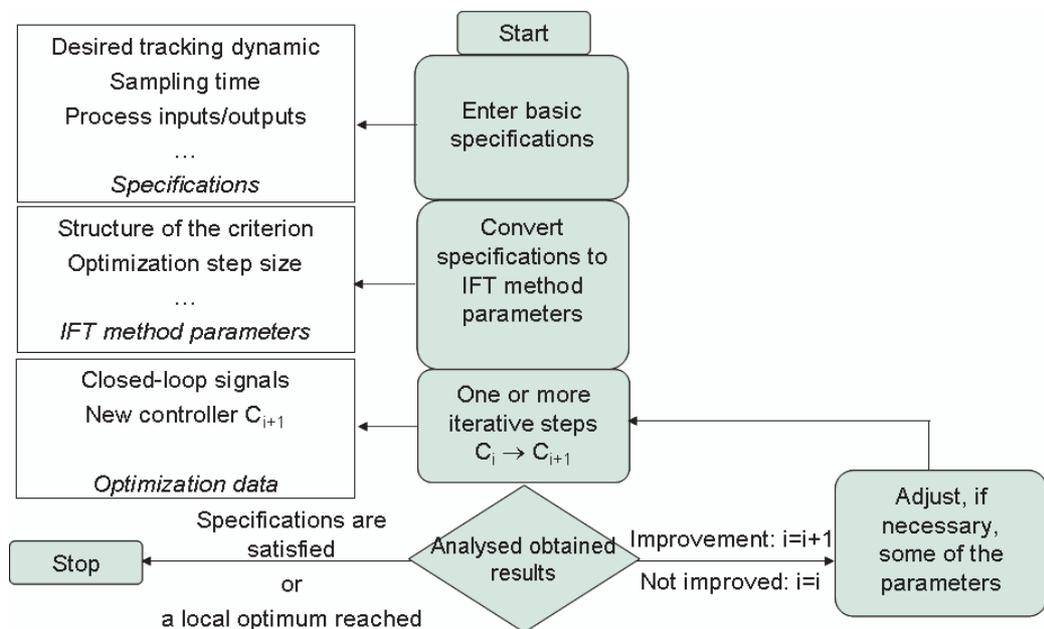


Figure 3: Workflow diagram.

Graphical user interface for the software tool main window is shown in Figure 4. The window left side (yellow background) displays user inputs (specifications, IFT method parameters). Right side (in light blue) presents optimization results and its analysis.

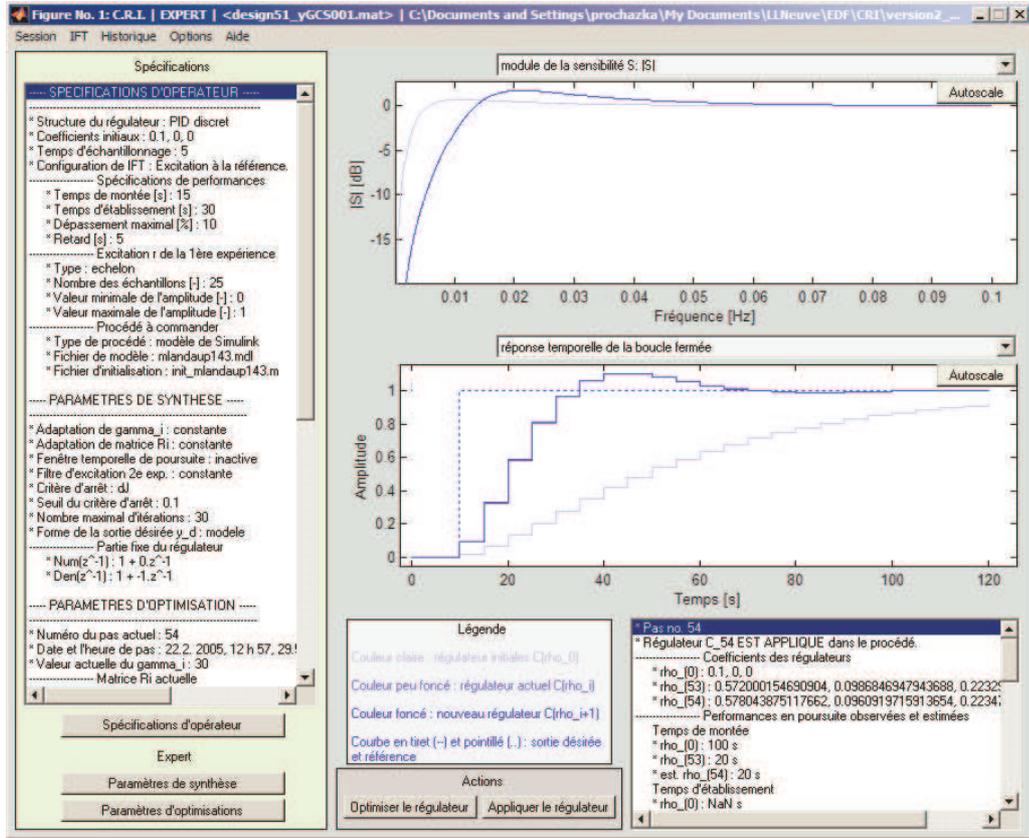


Figure 4: Software tool main window.

## 4 Application example: Multi-set-point tuning of barrage level control

### 4.1 Process

A case of dam level control for hydraulic power plant is considered. A non-linear dynamic model is used instead of real process. A level of water at the output  $z(t)$  [m] is controlled using a PI controller by acting on the output flow  $Q_s(t)$ . A disturbance is an input flow  $Q_e(t) \in (0, 2500) \text{m}^3/\text{s}$ . The desired level is  $z_d(t) \in (224.75, 228.5)$ .

An initial continuous-time PI controller is:

$$PI_c(s) = K \left( 1 + \frac{1}{sT_i} \right) \quad (12)$$

with  $K = -300 \text{ [m}^3/\text{s/m]}$ ,  $T_i = 5000 \text{ [s]}$ .

The IFT method optimizes a discrete-time controller only. We choose a sampling time  $T_s = 600 \text{ s}$ . The controlled system is only an integrator with variable gain and a delay. Hence the choice of the sampling time is a question of desired performance. For example, for a faster closed-loop dynamics a shorter sampling time is needed. A discrete-time equivalent of the initial PI controller is then:

$$PI_d(z^{-1}) = \frac{-327 + 273z^{-1}}{1 - z^{-1}} \quad (13)$$

### 4.2 One-set-point tuning

First the PI controller will be tuned only in one set-point for  $Q_e$  between 500 and 1000  $\text{m}^3/\text{s}$ . The input excitation signals are shown in Figure 5. After 19 iterations the obtained controller is:

$$PI_d(z^{-1}) = \frac{-1679.6 + 809.7z^{-1}}{1 - z^{-1}} \quad (14)$$

The criterion is  $J_0 = 0.056714$  for the initial controller and  $J_{19} = 0.001103$  for the optimized controller. The responses of the closed loop for initial and optimized controller are shown in Figure 6. However the optimized controller gives an unstable closed loop in other operating set-points as it is shown in Figure 7. For  $Q_e > 1700\text{m}^3/\text{s}$  instability appears.

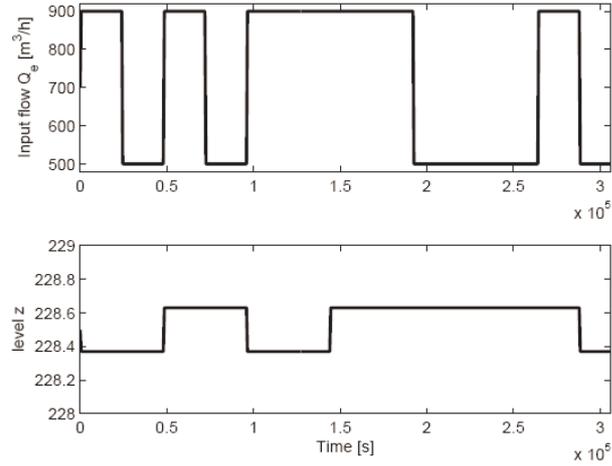


Figure 5: Experimental input excitations.

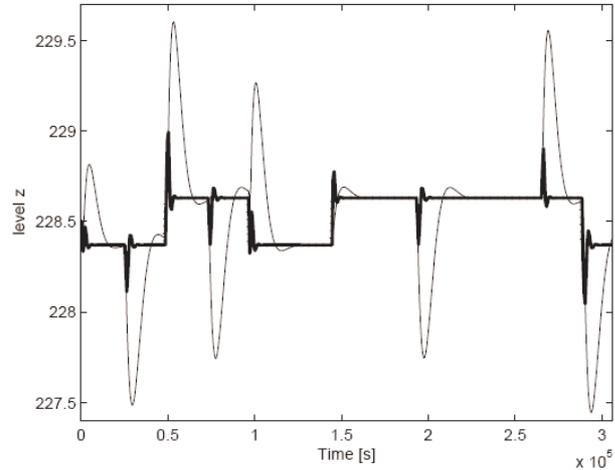


Figure 6: Closed-loop output responses of initial (thin line) and optimized (thick line) PI controller in the case of one-set-point tuning (dotted line is reference level).

### 4.3 All set-point tuning

The operating set-point depends on the input flow  $Q_e$  which ranges from  $0\text{m}^3/\text{s}$  to  $2500\text{m}^3/\text{s}$ . We split the operating range into three operating set-points:

- Set-point no. 1:  $Q_e \in (0, 1000)$
- Set-point no. 2:  $Q_e \in (1000, 2000)$
- Set-point no. 3:  $Q_e \in (2000, 2500)$

For these three operating set-points we define input signals as it is shown in Figure 8. We use the proposed procedure to tune the controller in all three set-points simultaneously. After 6 iterations the obtained controller is:

$$PI_d(z^{-1}) = \frac{-623.3 + 304.5z^{-1}}{1 - z^{-1}} \quad (15)$$

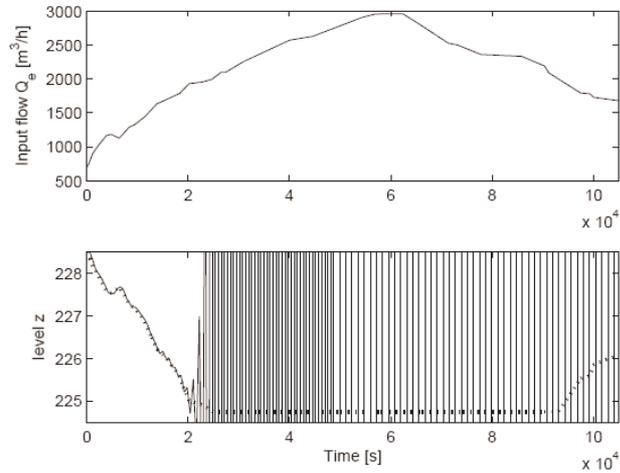


Figure 7: Closed-loop output response of the optimized PI controller in the case of one-set-point tuning, input signals pass all set-points (dotted line is reference level).

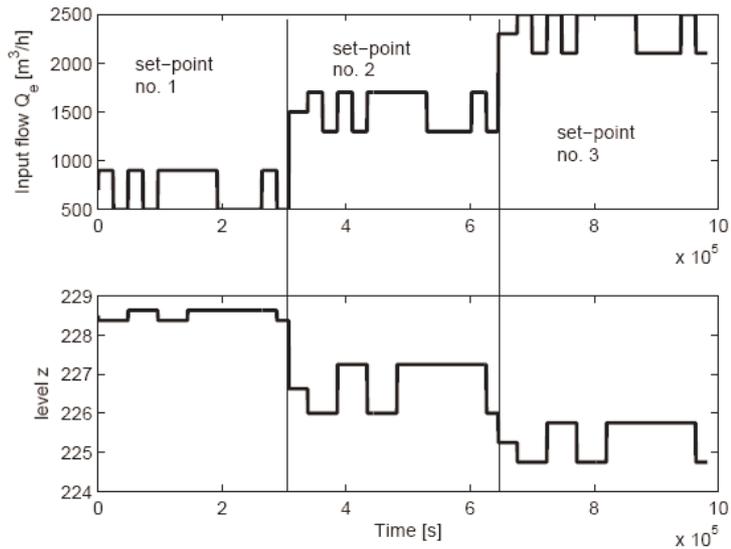


Figure 8: Input signals for tuning in all three set-points.

The criterion is  $J_0 = 0.05825$  for the initial controller and  $J_6 = 0.012972$  for the optimized controller. The optimized controller gives stable closed loops in all operating set-points and the reference tracking is excellent compared to initial PI controller, see Figure 9.

## 5 Conclusion

An interactive software tool implementing IFT method is developed and tested. The tool implements not only classical IFT method theory. Several advanced features of the method are introduced as well. Different control tuning and optimization case studies have been made using the software to verify and improve software tool efficiency. One of the cases is presented in this paper. A list of case studies made for the software tool testing is in Table 1.

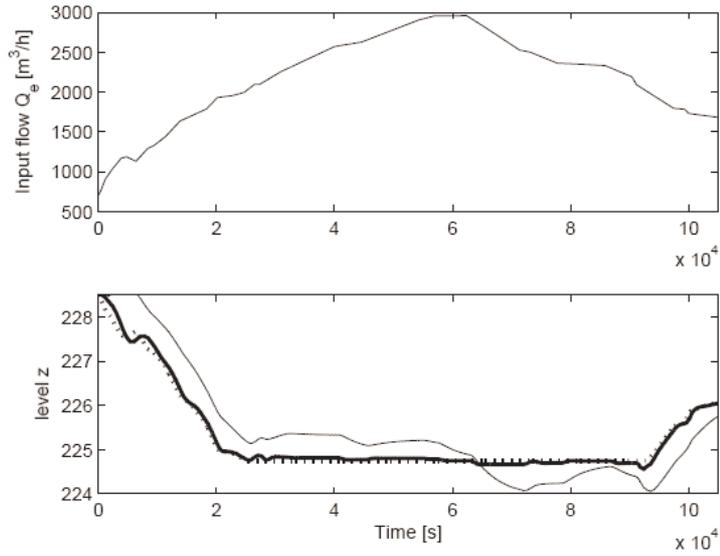


Figure 9: Closed-loop output response for the initial (fine line) and optimized (bold line) PI controller in the case of all-set-point tuning. Dotted line is level reference (set-point).

Table 1: CASE STUDIES FOR TESTS AND VALIDATION OF THE SOFTWARE TOOL.

<i>Case</i>	<i>Controller structure</i>	<i>Optimized performance</i>	<i>Process/model</i>	<i>Power plant</i>
Dam level control	PI	Multi-set-point tracking (robust performance)	Non-linear Simulink model	Hydraulic
Level control for steam generator	PID + feed-forward	Set-point tracking	Non-linear Simulink model	Nuclear
1 <sup>st</sup> order linear model control (see [2])	PI	Set-point tracking and robust stability	Linear Simulink model	-
Re-heater control	PI	Set-point tracking and output disturbance rejection	Linear Simulink model	Thermal
Pressure and power control	Two PI controllers	Multi-set-point tracking (robust performance)	Non-linear Simulink model	Thermal
Steam temperature control	Cascade of two PI	Set-point tracking and output disturbance rejection	Real process	Thermal

## References

- [1] H. Hjalmarsson, S. Gunnarsson, and M. Gevers. Model-free tuning of a robust regulator for a flexible transmission system. *European Journal of Control*, 1(2):148–156, 1995.
- [2] H. Prochazka, M. Gevers, B.D.O. Anderson, and C. Ferrera. Iterative feedback tuning for robust controller design and optimization. *Proceedings IEEE, CDC-ECC*, Seville, Spain, 2005.
- [3] H. Hjalmarsson and M. Gevers Eds. *Control Engineering Practice* (11), issue 9: Special Section on Algorithms and Applications of Iterative Feedback Tuning. Elsevier Science Ltd., September 2003.

H. Procházka  
hynek.prochazka@prosystemy.sk

J.M. Legros  
jean-michel.legros@edf.fr