# ADAPTIVE CONTROL THROUGH ON-LINE SIMULATION PRACTICAL ASPECTS

**Radu Bălan**

*Technical University Cluj-Napoca, Faculty of Mechanics/ Mechatronics Dept.*
*B-dul Muncii 103-105 Cluj-Napoca, tel 064-155967 e-mail: radubalan@yahoo.com*

ABSTRACT

There are proposed a few applications examples of an algorithm that uses on-line simulation and rule-based control. A compare between this algorithm and PID control is presented. Some examples are used to indicate the control parameters choosing.

KEYWORDS: adaptive-predictive control, on-line simulation, rule-based control.

## 1. INTRODUCTION

In [7], it is presented some theoretical aspects of an adaptive-predictive algorithm that uses on-line simulation and rule based control. In a practically implementation, the algorithm A1 [7] can be write:

STEP 1: Output measuring y(t); up-date input u[.] and output y[.] vectors
STEP 2: Up-date counters that define conditions of algorithm function:
IF $V_{sta}=0$
THEN $C_{sta}=0$; $C_{sta1}=0$; { transitory regime }
IF $u_{min}<u(t)<u_{max}$ THEN $C_{sta}=1$; $C_{sta1}=1$; {Start stationary regime}
ELSE $C_{sta}$++; $C_{sta1}$++; { stationary regime }
IF $C_{sta1}>V_{sta2}$ THEN $C_{sta1}=V_{sta1}$
{$0<C_{sta}\leq V_{sta1}$ -initial part of stationary regime
$V_{sta1}<C_{sta}\leq V_{sta2}$ -stabilization part of stationary regime
$C_{sta}>V_{sta2}$ -final stationary regime}
STEP 3: Test if the algorithm works well in stationary regime:
IF $|y(t)-yr(t)|>\Delta p$ AND $C_{sta}> V_{sta2}$
THEN $C_{sta}=0$ {perturbation regime}
STEP 4: Identification:
IF $C_{sta}<V_{sta3}$ { $V_{sta3}> V_{sta2}$ }
THEN execute RLS algorithm
ELSE execute additive correction
STEP 5: Up-date $u_{maxst}(t)$, $u_{minst}(t)$:
IF $C_{sta}<V_{sta1}$ THEN $a_{med}(t)=0$; $u_{maxst}(t)=u_{max}$; $u_{minst}(t)=u_{min}$;
ELSE $a_{med}(t)=k_a a_{med}(t-1)+(1- k_a)|y_r(t)-y(t)|$
$u_{maxst}(t)= u_{maxst}(t-1)-k_{st}[u_{maxst}(t-1)- u_{st}(t)]+k_a a_{med}(t)$
$u_{minst}(t)= u_{minst}(t-1)+k_{st}[u_{st}(t)-u_{minst}(t-1)]-k_a a_{med}(t)$

Limit $u_{minst}$, $u_{maxst}$ to perform next conditions:

$u_{min} \le u_{minst}(t) < u_{maxst}(t) \le u_{max}$

$u_{maxst}(t) - u_{minst}(t) > d_{ust}$

STEP 6: Simulate the behaviour of system for control sequences:

$u_1(t) = \{ u_{minst}, u_{minst}, .., u_{minst} \}$ Find $y_{max0}$.

$u_2(t) = \{ u_{maxst}, u_{minst}, .., u_{minst} \}$ Find $y_{max1}$.

$u_3(t) = \{ u_{minst}, u_{maxst}, .., u_{maxst} \}$ Find $y_{min0}$.

$u_4(t) = \{ u_{maxst}, u_{maxst}, .., u_{maxst} \}$ Find $y_{min1}$.

STEP 7: Use simulated dates:

Compute variable reference: $y_{r1}(t) = y_r(t) + k_{ref}[y(t) - y_r(t)]$

IF $y_{max0} < y_{r1}(t)$ (corresponding to $u_1(t)$ sequence) AND
$y_{max1} > y_{r1}(t)$ (corresponding to $u_2(t)$ sequence)
THEN choose:

$$u(t) = \frac{u_{max\,st}(t) - u_{min\,st}(t)}{y_{max1} - y_{max0}} y_{r1}(t) + \frac{u_{min\,st}(t) y_{max1} - u_{max\,st}(t) y_{max0}}{y_{max1} - y_{max0}}$$

ELSE

IF $y_{min0} < y_{r1}(t)$ (corresponding to $u_3(t)$ sequence) AND
$y_{min1} > y_{r1}(t)$ (corresponding to $u_4(t)$ sequence)
THEN choose:

$$u(t) = \frac{u_{max\,st}(t) - u_{min\,st}(t)}{y_{min1} - y_{min0}} y_{r1}(t) + \frac{u_{min\,st}(t) y_{min1} - u_{max\,st}(t) y_{min0}}{y_{min1} - y_{min0}}$$

ELSE

IF $y_{max0} > y_{r1}(t)$ (corresponding to $u_1(t)$ sequence)
THEN choose $u(t) = u_{minst}(t)$
ELSE

IF $y_{max1} < y_{r1}(t)$ (corresponding to $u_2(t)$ sequence)
THEN choose $u(t) = u_{maxst}(t)$
ELSE choose $u(t) = u(t-1)$

STEP 8: Compute the average of controller's output:

$u_{med}(t) = k_{umed} u_{med}(t-1) + (1 - k_{umed}) u(t)$

Compute the estimate value of controller's output in stationary regime:

$$u_{st}(t) = \frac{1 + a_1 + a_2 + .. + a_n}{b_1 + b_2 + .. + b_m} y_r(t)$$

STEP 9: u(t) filter:

IF $V_{sta2} < C_{sta}$
THEN $u(t) \leftarrow k_u u(t) + (1 - k_u) u_{med}$.

## 2. APPLICATIONS

Based on these rules and on-line simulation there were developed algorithms for linear/nonlinear processes, constant setpoint (A1 algorithm [2]) or variable setpoint (A2 algorithm [2]), adaptive/nonadaptive case. The algorithms were tested both in simulation (DELPHI applications) and in real time control (using 80C552 microcontroller and DELPHI)[2], [3], [4]. Next, it is presented a few examples, which indicate the control parameters choosing.

*Example 1*

Let's consider the process (P1) [1]:

$$y(t) = -a_1 y(t-1) - a_2 y(t-2) - a_3 y(t-3) + b_1 u(t-1-d) + b_2 u(t-2-d) + b_3 u(t-3-d)$$

where: *y[.]: process's output; u[.]: controller's output; $0 \le u[.] \le 250$;*
A[.]=[1 -2.43492 1.97629 -0.53468]; B[.]=[0.000948003 0.004438182 0.001296496];
Static gain is $k_0=1$; d=1 is dead time; t: discrete time ($0 \le t \le 75$).

*Remark*: In the next figures only the setpoint (SP- $y_r(t)$) and process's output (PV- y(t)) are represented at true scale. Controller's output (OP- u(t)) is represented u(t)/3. Notations from legend (figure1) are used in figure 2..13. Here, $y_r[0]=0$ and $y_r[t]=150$ for t>0, u[0]=0 for $t \le 0$ and u [t]=150 for t>0.
For PID tune, it was used Ziegler-Nichols criterion.

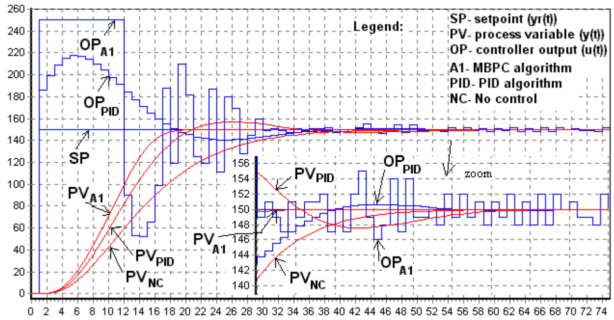


Fig. 1: Example 1

*This example shows the advantages of A1 algorithm, comparatively with PID: a shorter time response, no override. A possible drawback in some applications is a larger variance of u(t), but this variance can be reduced if it is necessary.*

*Example 2*
Conditions: the setpoint has a variable shape, the model is accurate (nonadaptive case), $u_{max}=250$, $u_{min}=0$, $k_{ref}=0.2$, $k_u=1$, $k_a=0$, $V_{sta1}=5$, $V_{sta2}=50$, $d_{ust}=5$, it is not use information about setpoint changes. This example shows the effect of $k_{st}$ choice in STEP 5.

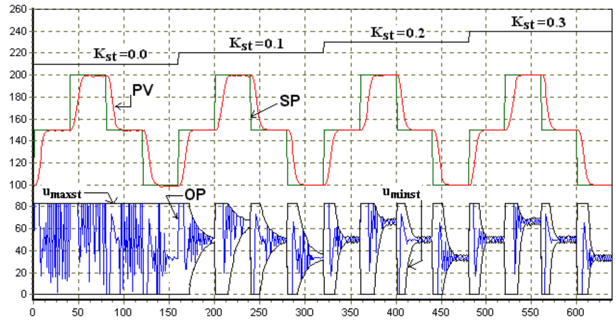


Fig. 2: Example 2

*Example 3*
Conditions: analogous with example 2; $k_{st}=0.0$. This example shows the effect of $k_{ref}$ choice in STEP 7 if $k_{st}=0.0$. In this case, the time response is minim, but the variance of u(t) is larger. Other parameters: $k_u=1$, $k_a=0$, $V_{sta1}=5$, $V_{sta2}=50$, $d_{ust}=5$.

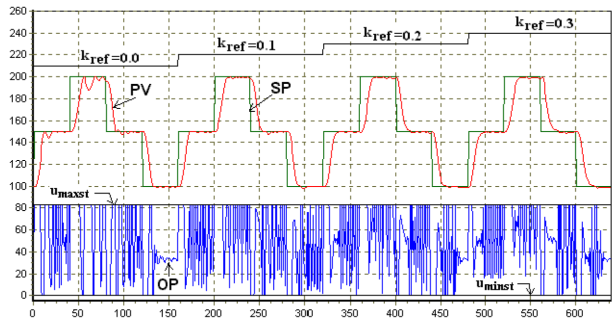


Fig. 3: Example 3

*Example 4*
Conditions: analogous with example 3; $k_{st}=0.1$. This example shows the effect of $k_{ref}$ choice if in STEP 7 if $k_{st} \ne 0.0$.
Other parameters: $k_u=1$, $k_a=0$, $V_{sta1}=5$, $V_{sta2}=50$, $d_{ust}=5$.
If $k_{st} \ne 0.0$, in stationary regime, the difference $u_{maxst}(t)$- $u_{mnist}(t)$ will decrease.

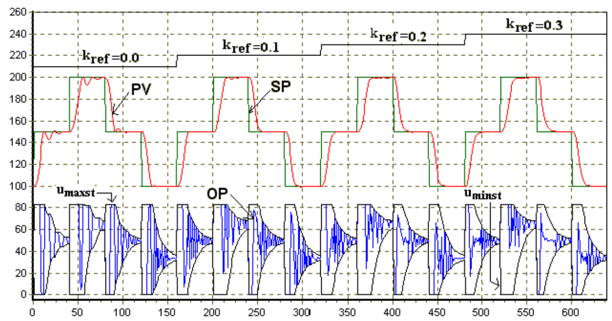


Fig. 4: Example 4

*Example 5*

Conditions: $k_{st}$=0.05, $k_{ref}$=0.1. This example shows the effect of $k_u$ choice and indicates a method to reduce the variance of control signal (STEP 9). But if $k_u$ is quite small, it is possibly small oscillation on output.

Other parameters: $k_a$=0, $V_{sta1}$=5, $V_{sta2}$=50, $d_{ust}$=5.

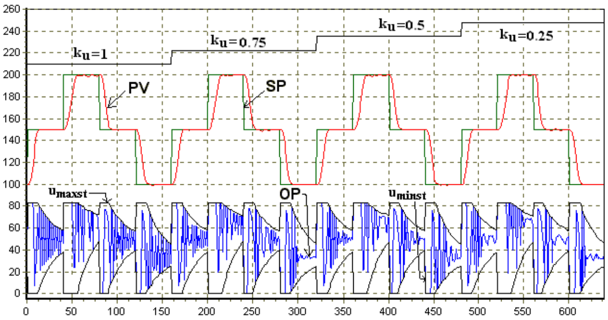


Fig. 5: Example 5

*Example 6*

Conditions: the static gain ($k_0$) is changed from 1 to 1.6 with 0.2 step; a model-based adaptive-predictive control ($A_1$) has been used;

The estimate of static gain is $k_{0est}$. The forgetting factor is $\lambda$=0.98, $k_{st}$=0.15, $k_{ref}$=0.2, noise: $\sigma$ =0.

Other parameters: $k_u$=0.5, $k_a$=2 $V_{sta1}$=5, $V_{sta2}$=10, $V_{sta3}$=10, $d_{ust}$=5.

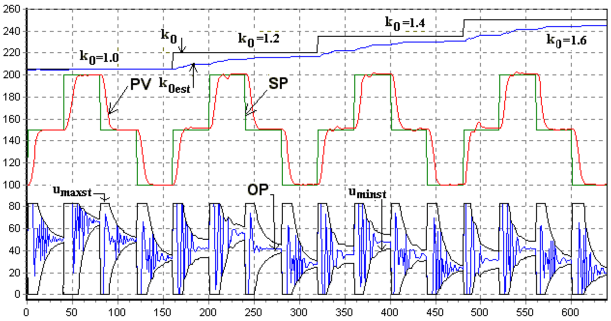


Fig. 6: Example 6

If the difference between process and model is quite larger, the control algorithm will compute a wrong control signal and it is possibly to appear significant errors (for example at step 498 the override is 16%). A method to reduce this effect is to choose cautions value for parameters, especially for $k_{st}$ and $k_{ref}$.

*Example 7*

Conditions: analogous with example 6 but $\lambda$=0.94.

This example shows the effect of forgetting factor ($\lambda$) choosing. If $\lambda$ is small, the identification algorithm works faster. Compare example 6 and example 7. But in real case, if $\lambda$ is quite small it is possibly to appear significant oscillations in parameters identification.

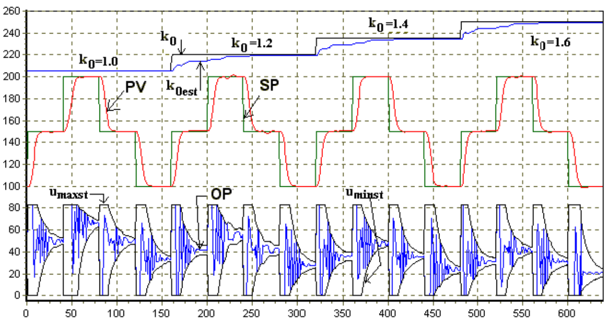


Fig. 7: Example 7

*Example 8*

Conditions: analogous with example 7 but $V_{sta3}$=30.

This example shows the effect of $V_{sta3}$ choosing. This parameter indicates when the RLS algorithm is stopped and it is changed with a additive correction (STEP 4). In this example, the RLS algorithm works a longer time than in example 7.

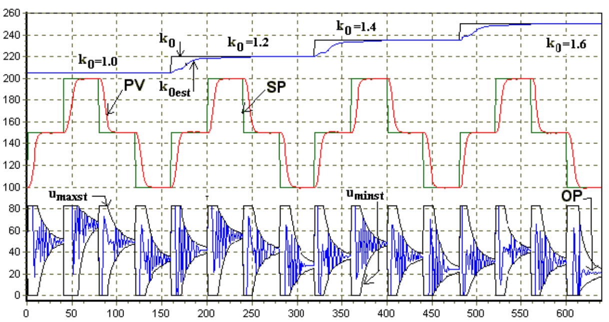


Fig. 9: Example 8

*Example 9*
Conditions: analogous with example 8 but noise: $\sigma = 10^{-3}$.
This example shows the effect of noise: small oscillations in parameters identification. The variance of control signal increases. Compare example 9 and example 8. But if the process permits this variance, it leads to faster parameters identification.

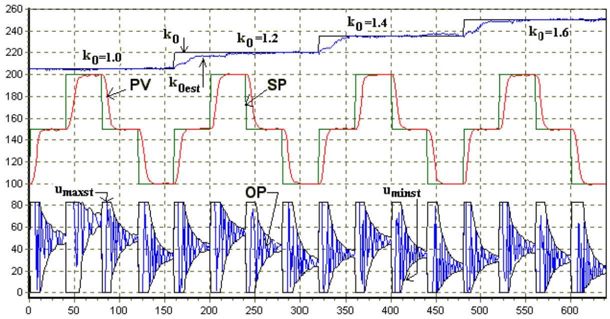

Fig. 9: Example 9

*Example 10*
Conditions: analogous with example 6, but it was used PID control (without static gain estimation).
If differences between process and model are larger, than appear big override. It is necessary to introduce a identification component.

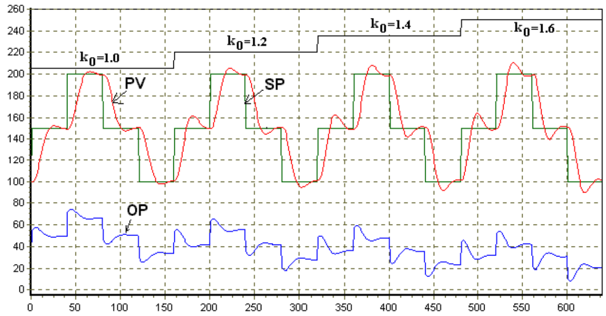

Fig. 10: Example 10

*Example 11*
Conditions: analogous to example 10 but it was used static gain estimation $k_{0est}$. The forgetting factor is $\lambda$=0.98. It is used a adaptive PID control. Comparatively with example 10, the quality of control increases, especially after parameters identification. But initial, when the difference between process and model are larger, appear big override (see steps 175, 340, 500), than, after parameters identification, these overrides decrease (see steps 225, 380, 540). Comparatively with example 8 in the same conditions, the variance of control signal is smaller, but this small variance leads to delay in parameters identification.

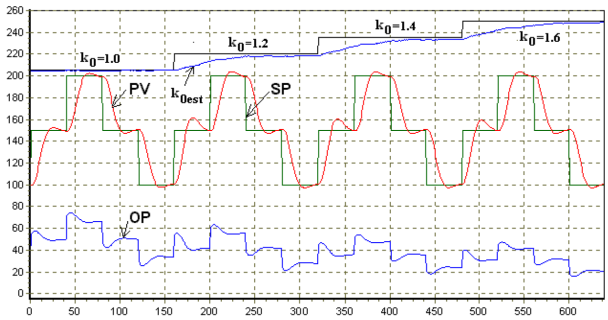

Fig. 11: Example 11

*Example 12*
Conditions: analogous to example 11 but the forgetting factor is $\lambda$=0.94. Comparatively with example 11, the parameters identification is faster, but the outputs have the same behavior. This is due to characteristics of PID

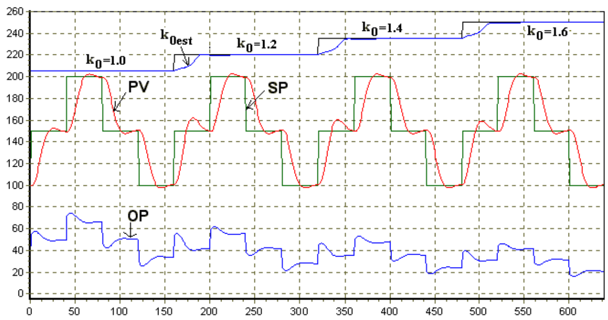

Fig. 12: Example 12

control. Compare with example 7 and 8 where was used A1 algorithm.

*Example 13*
Conditions: analogous to example 12, but noise: $\sigma = 10^{-3}$.
If variance of u(t) and $\lambda$ are quite small, it is possible an unwell identification (425..475 steps). Compare example 9,12,13. PID algorithm leads to a quite small variance of u(t).

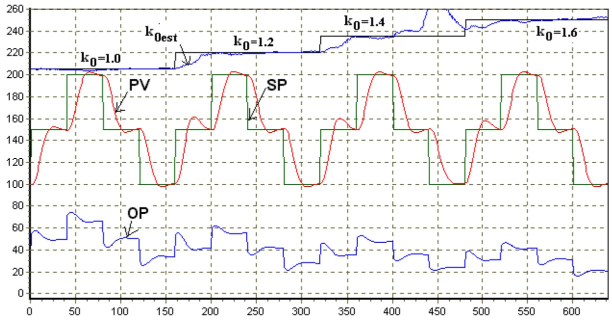

Fig. 13: Example 13

## 3. CONCLUSIONS

This paper presents some examples, which permit to choose the parameters of an adaptive-predictive algorithm that uses on-line simulation, and rule based control. It was used a DELPHI application; the user can easily choose and modify the parameters of process, model and control. Also, it is presented a compare between adaptive-predictive control and PID control. The parameters of algorithm can be choosing in large limits and can be optimized using a supervisor algorithm.

## 4. REFERENCES

[1] Camacho E., Bordons C. [1999], "Model Predictive Control" Spriger-Verlag
[2] Balan R. [2001] "Adaptive control systems applied to technological processes", Ph.D. Thesis 2001, Technical University of Cluj-Napoca, Romania.
[3] Control Engineering Europe February/March [2001] pp. 25..28
[4] Balan R. [2000], "Adaptive Control for Thermal Processes described by Strejc Equation", Q&A-R 2000 THETA 12
[5] Balan R. [2000], "A Method for Heat-Exchanger Simulation and Control" Q&A-R 2000 THETA 12
[6] Nikolau M., S.Alper Eker [1999], "Adaptive control through on-line optimization. The MPCI paradigm and variants. Int. J. Appl. Math and Comp. Sci, 1999 Vol 9, No. 1, pp 25-52
[7] Balan R. [2002] "Adaptive Control through on-line Simulation. Theoretical aspects" Q&A-R 2000 THETA 13, Cluj-Napoca, Romania