University of Ljubljana

J. Stefan Institute, Ljubljana, Slovenia

Report **DP-7267**

Transient control by the integration method

Damir Vrančić
Đani Juričić

September 1995

Table of contents

1. Introduction 2
2. Theoretical evaluation 3
3. Experiments 18
4. Conclusions 33
References 34

1. Introduction

Each closed-loop control system must execute at least two tasks: to provide good control and tracking performance. There are many different types of controllers available to fulfil such tasks like PID types of controllers, state-space controllers, controllers with rational transfer functions, fuzzy and neural network controllers, etc. However, frequently happens that it is not easy to find such controller parameters that will satisfy given performances.

In this report we will give some guidelines for designing feed-forward controller for changing the process set-point. After transition is completed, the original closed-loop controller has to be switched on again. The advantages of such feed-forward controller (algorithm) is in its simplicity and quite interesting performance (overshoot for most usual processes equals to zero). The drawbacks are in feed-forward structure and unstable properties, so it must be switched again to closed-loop when process approaches new set-point.

For evaluating the feed-forward algorithm we used the idea of [Strejc, 1959]. He expressed the surface between process input and process output (method of multiple integration). The aim of the method was to identify process model on stepwise change at process input [R. Isermann, 1971] [R. Isermann, 1974] [R. Isermann, 1988] [H. Rake, 1987]. The method is specially useful for processes running in noisy environments. Drawback of the method lays in a fact that the process model is usually not described well for higher frequencies (more than three multiple integrations are not recommended) [R. Isermann, 1971].

In our evaluation, we used only first integration and showed that the same surface between process input and process output appears in a closed-loop systems. Based on this fact, a new feed-forward algorithm for process transient is proposed.

2. Theoretical evaluation

Consider a stable and linear process given by Fig. 1 and equation (1).

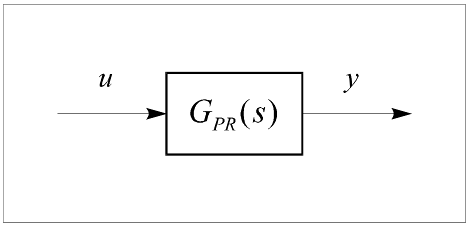

Fig. 1. A stable and linear process

$$G_{PR}(s) = \frac{1 + b_1 s + b_2 s^2 + \cdots + b_m s^m}{1 + a_1 s + a_2 s^2 + \cdots + a_n s^n} e^{-sT_d} \tag{1}$$

From (1), process output can be expressed as

$$Y(s) = \frac{1 + b_1 s + b_2 s^2 + \cdots + b_m s^m}{1 + a_1 s + a_2 s^2 + \cdots + a_n s^n} e^{-sT_d} U(s) \tag{2}$$

where *Y(s)* and *U(s)* represent Laplace transforms of process output and process input, respectively. Typical process output response on unity-gain step function at process input is shown in Fig. 2.

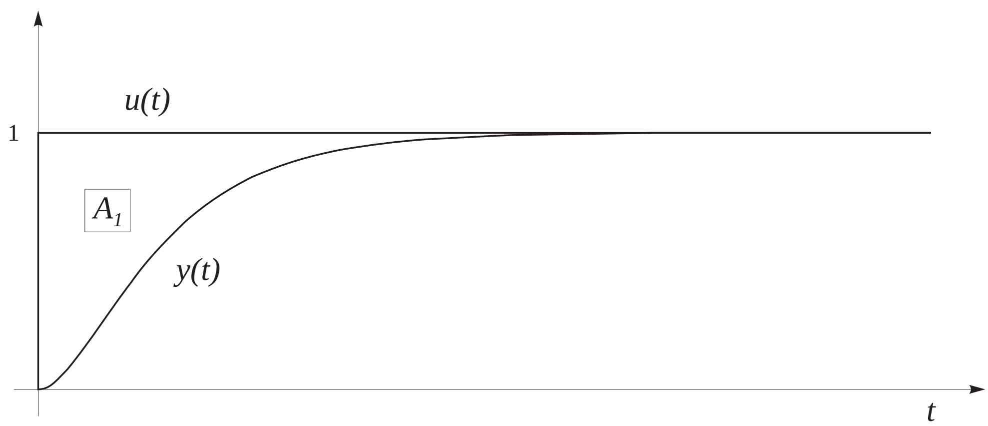

Fig. 2. Typical process response on unity-gain step function of u

where A_1 denotes the surface between $u(t)$ and $y(t)$:

$$A_1 = \int_0^{\infty} \left[u(t) - y(t) \right] dt \tag{3}$$

Equation (3) can be expressed by Laplace transforms $U(s)$ and $Y(s)$ in another way:

$$A_1 = \lim_{s \to 0} \left[U(s) - Y(s) \right] = \lim_{s \to 0} \left[\left(1 - G_{PR}(s)\right) U(s) \right] \tag{4}$$

When taking into account that $u(t)$ is the unity-gain step function

$$U(s) = \frac{1}{s} \tag{5}$$

and that delay can be expressed by Taylor series as

$$e^{-sT_d} = 1 - sT_d + \frac{s^2 T_d^{\,2}}{2!} - \cdots, \tag{6}$$

A_1 becomes

$$\begin{aligned} A_1 &= \lim_{s \to 0} \left[\left[1 - \frac{1 + b_1 s + b_2 s^2 + \cdots + b_m s^m}{1 + a_1 s + a_2 s^2 + \cdots + a_n s^n} \left(1 - sT_d + \frac{s^2 T_d^{\,2}}{2!} - \cdots \right) \right] \frac{1}{s} \right] = \\ &= a_1 - b_1 + T_d \end{aligned} \tag{7}$$

If $G_{PR}(s)$ is expressed in zero-pole representation,

$$G_{PR}(s) = \frac{\left(1 + sT_{z1}\right)\left(1 + sT_{z2}\right) \cdots \left(1 + sT_{zm}\right)}{\left(1 + sT_{p1}\right)\left(1 + sT_{p2}\right) \cdots \left(1 + sT_{pn}\right)} e^{-sT_d}, \tag{8}$$

and when asserting (1) and (8) equal, we can see that

$$\begin{aligned} b_1 &= T_{z1} + T_{z2} + \ldots + T_{zm} = \sum_{i=1}^{m} T_{zi} \\ a_1 &= T_{p1} + T_{p2} + \ldots + T_{pn} = \sum_{i=1}^{n} T_{pi} \end{aligned} \tag{9}$$

From (7) and (9), the surface A_1 (see Fig. 2) can be expressed as

$$A_1 = \sum_{i=1}^{n} T_{pi} - \sum_{i=1}^{m} T_{zi} + T_d \tag{10}$$

Now, consider a stable closed-loop system (see Fig. 3)

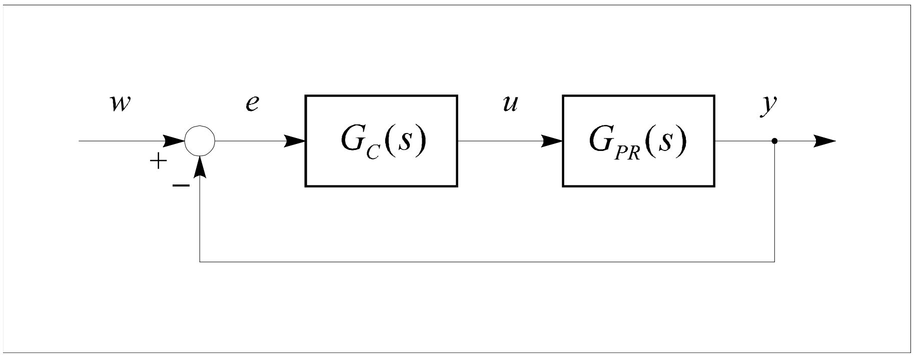

Fig. 3. Process and controller in closed-loop

where w, e and G_C are reference (set-point), control error and controller transfer function, respectively. G_{PR}(s) is described in (1) and (8). Let the controller transfer function equals to

$$G_c(s) = \frac{d_0 + d_1 s + \ldots + d_q s^q}{c_0 + c_1 s + \ldots + c_p s^p} \tag{11}$$

and consider the unity gain step function as a reference signal

$$W(s) = \frac{1}{s} \tag{12}$$

Typical system response would be as shown in Fig. 4.

From Fig. 3, u can be expressed as

$$U(s) = \frac{G_C(s)}{1 + G_C(s) G_{PR}(s)} W(s) \tag{13}$$

Using (4), (9), (12) and (13), we can calculate the surface A_2 in Fig. 4 as

$$A_2 = \lim_{s \to 0}\left[\left(1 - G_{PR}(s)\right)U(s)\right] =$$
$$= \frac{d_0}{c_0 + d_0}\left(a_1 - b_1 + T_d\right) = \frac{d_0}{c_0 + d_0}\left[\sum_{i=1}^{n} T_{pi} - \sum_{i=1}^{m} T_{zi} + T_d\right] \tag{14}$$

To have no steady-state error, controller must have integral character. This yields

$$c_0 = 0 \tag{15}$$

Expression (15), when inserting into (14), gives (7). Surfaces A_1 and A_2 (see Figs. 2 and 4) are the same, respectively. To conclude, if we want to move the process output from one steady state $y(0)=0$ to another steady state $y(\infty)=1$, the input signal $u(t)$ should be such that

$$A = \int_0^\infty \left[u(t) - y(t)\right]dt = a_1 - b_1 + T_d = \sum_{i=1}^{n} T_{pi} - \sum_{i=1}^{m} T_{zi} + T_d \tag{16}$$

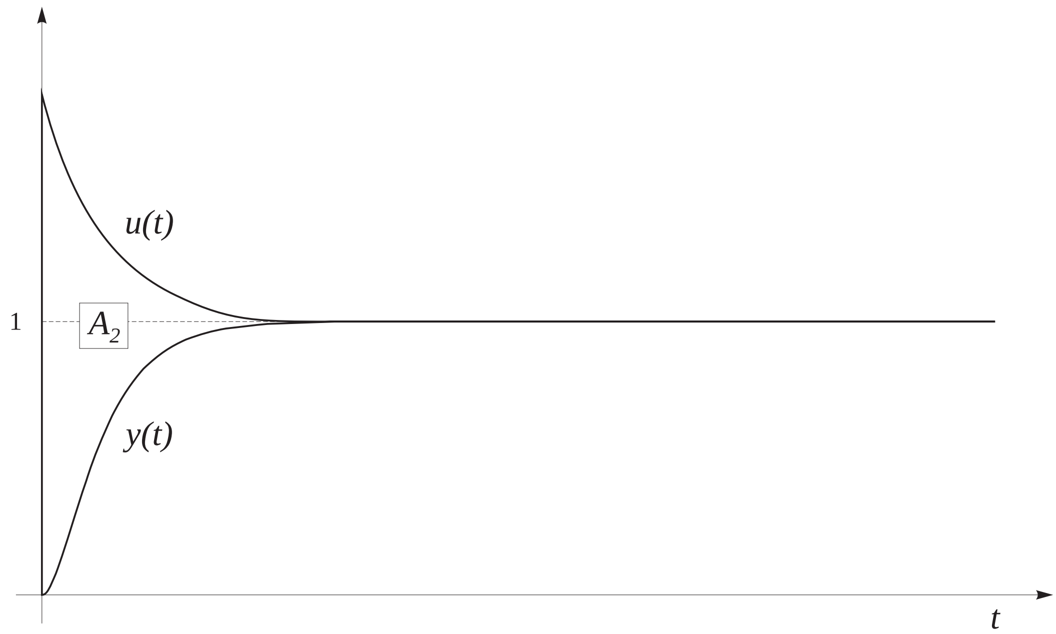

Fig. 4. Typical closed-loop system response

When process is linear (1), we can make further conclusions. If we want to move the process output from one steady state $y(0)=y_0$ to another steady state $y(\infty)=y_\infty$, we have to use such input signal *u(t),* that

$$A = \int_0^\infty [u(t) - y(t)]dt = (y_\infty - y_0)(a_1 - b_1 + T_d) = (y_\infty - y_0)\left[\sum_{i=1}^{n} T_{pi} - \sum_{i=1}^{m} T_{zi} + T_d\right] \qquad (17)$$

Expression (17) is valid if process steady state gain is 1 as expressed in (1). If process steady state gain is different than 1, we can add a gain in front of the process, to make it equal to 1 as shown in Fig. 5.

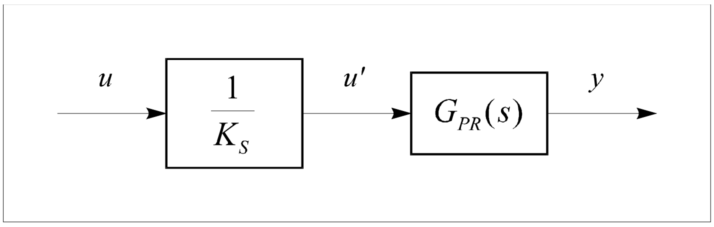

Fig. 5. Compensation of the steady-state process gain

where K_S represents a steady state gain of the process G_{PR}:

$$G_{PR}(s) = K_S \frac{1 + b_1 s + b_2 s^2 + \cdots + b_m s^m}{1 + a_1 s + a_2 s^2 + \cdots + a_n s^n} e^{-sT_d} \qquad (18)$$

Process with additional *zero in the origin* needs different approach. Let us have next process transfer function:

$$G_{PR}(s) = \frac{s + b_1 s^2 + b_2 s^3 + \cdots + b_m s^{m+1}}{1 + a_1 s + a_2 s^2 + \cdots + a_n s^n} e^{-sT_d} = \frac{s(1 + sT_{z1})(1 + sT_{z2})\cdots(1 + sT_{zm})}{(1 + sT_{p1})(1 + sT_{p2})\cdots(1 + sT_{pn})} e^{-sT_d} \qquad (19)$$

Now, instead of unity gain step function, we can use the unity ramp signal u:

$$U(s) = \frac{1}{s^2} \qquad (20)$$

Fig. 6 shows us typical system response

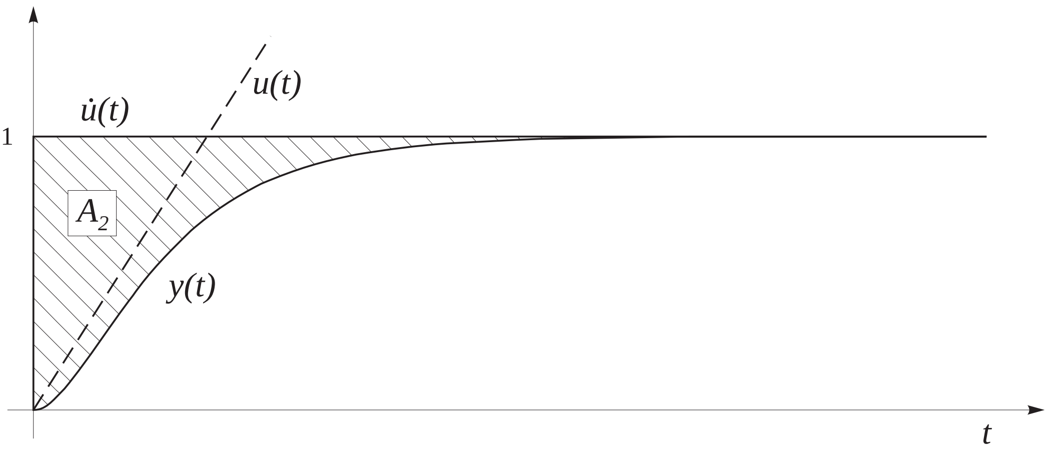

Fig. 6. Typical system response on ramp signal at u for processes with zero in origin

where $\dot{u}(t)$ is time derivation of *u(t)*. A_3 represents a surface between $\dot{u}(t)$ and *y(t)* as

$$A_3 = \int_0^{\infty}\left[\dot{u}(t) - y(t)\right]dt \tag{21}$$

With the same derivation as in (4) to (10), we can express A_3 as

$$A_3 = a_1 - b_1 + T_d = \sum_{i=1}^{n} T_{pi} - \sum_{i=1}^{m} T_{zi} + T_d \tag{22}$$

The same result (22) would appear in closed-loop system (Fig. 3) if in steady state the process output $y(t)=y_\infty$ would become equal to w=1. As in (17), because the system is linear, we can make the following assumption. If we want to move the process output from one steady state $y(0)=y_0$ to another steady state $y(\infty)=y_\infty$ and process transfer function is described by (19), we have to use such signal u(t), that

$$A_3 = \int_0^{\infty}\left[\dot{u}(t) - y(t)\right]dt = \left(y_\infty - y_0\right)\left(a_1 - b_1 + T_d\right) = \left(y_\infty - y_0\right)\left[\sum_{i=1}^{n} T_{pi} - \sum_{i=1}^{m} T_{zi} + T_d\right] \tag{23}$$

If actual process transfer function has a steady-state gain K_S,

$$G_{PR}(s) = K_S \frac{s + b_1 s^2 + b_2 s^3 + \cdots + b_m s^{m+1}}{1 + a_1 s + a_2 s^2 + \cdots + a_n s^n} e^{-sT_d} = K_S \frac{s\left(1 + sT_{z1}\right)\left(1 + sT_{z2}\right)\cdots\left(1 + sT_{zm}\right)}{\left(1 + sT_{p1}\right)\left(1 + sT_{p2}\right)\cdots\left(1 + sT_{pn}\right)} e^{-sT_d} \tag{24}$$

the solution of such problem can be in adding gain $1/K_S$ in front of the process as shown in Fig. 5.

The last derivation is made for processes with pole in origin:

$$G_{PR}(s) = \frac{1 + b_1 s + b_2 s^2 + \cdots + b_m s^m}{s + a_1 s^2 + a_2 s^3 + \cdots + a_n s^{n+1}} e^{-sT_d} = \frac{(1 + sT_{z1})(1 + sT_{z2})\cdots(1 + sT_{zm})}{s(1 + sT_{p1})(1 + sT_{p2})\cdots(1 + sT_{pn})} e^{-sT_d} \tag{25}$$

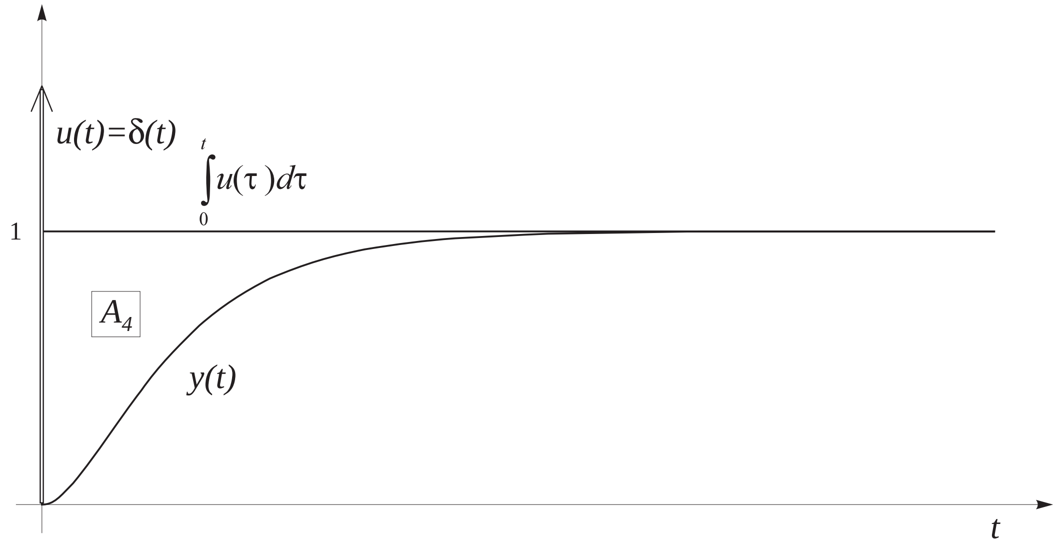

Fig. 7. Typical system response on delta signal at u for processes with pole in origin

Typical process response on input delta pulse ($u=\delta(t)$) is shown in Fig. 7, where we are searching for the surface A_4 between the integral of control signal and the process response $y(t)$.

$$A_4 = \int_0^\infty \left[\int_0^t u(\tau)d\tau - y(t) \right] dt \tag{26}$$

With the same derivation as in (4) to (10) and taking into account linear behaviour of the process, we can express A_4 as

$$A_4 = (y_\infty - y_0)(a_1 - b_1 + T_d) = (y_\infty - y_0)\left[\sum_{i=1}^{n} T_{pi} - \sum_{i=1}^{m} T_{zi} + T_d \right] \tag{27}$$

where y_0 and y_∞ are steady states at t=0 and t=∞, respectively. Also, if additional process gain is such that

$$G_{PR}(s) = K_S \frac{1+b_1 s + b_2 s^2 + \cdots + b_m s^m}{s + a_1 s^2 + a_2 s^3 + \cdots + a_n s^{n+1}} e^{-sT_d} = K_S \frac{(1+sT_{z1})(1+sT_{z2})\cdots(1+sT_{zm})}{s(1+sT_{p1})(1+sT_{p2})\cdots(1+sT_{pn})} e^{-sT_d} \quad (28)$$

we can add additional gain $1/K_S$ in front of the process as shown in Fig. 5. Process with additional gain is now the same as expressed in (25).

In praxis, it is not easy to control the surface A_4 as given in (26). In general, we should have infinite range of control signal u that process would be successfully controlled. This is not possible, due to process input limitations [Vrančić et al., 1995].

Because of integral behaviour of the process, we can move process output from one steady state $y(0)=y_0$ to another steady state $y(\infty)=y_\infty$ if using

$$\int_0^\infty u(t)dt = y_\infty - y_0 \quad (29)$$

Processes with different steady-state gains were introduced in (18), (24) and (28). For processes without pole or zero in the origin (18), K_S can be simply calculated as

$$K_S = \frac{y_\infty - y_0}{u_\infty - u_0} \quad (30)$$

where u_∞ and u_0 represent steady-state process inputs at t=∞ and t=0, respectively. Another way is to determine K_S with other shape of process input signal u:

$$U(s) = \frac{U_0}{s}\left(1 - e^{-\tau s}\right) \quad (31)$$

Typical process response on u is as shown in Fig. 8. If using u in (31), K_S can be calculated as

$$K_S = \frac{1}{U_0 \tau} \int_0^\infty y(t)dt \quad (32)$$

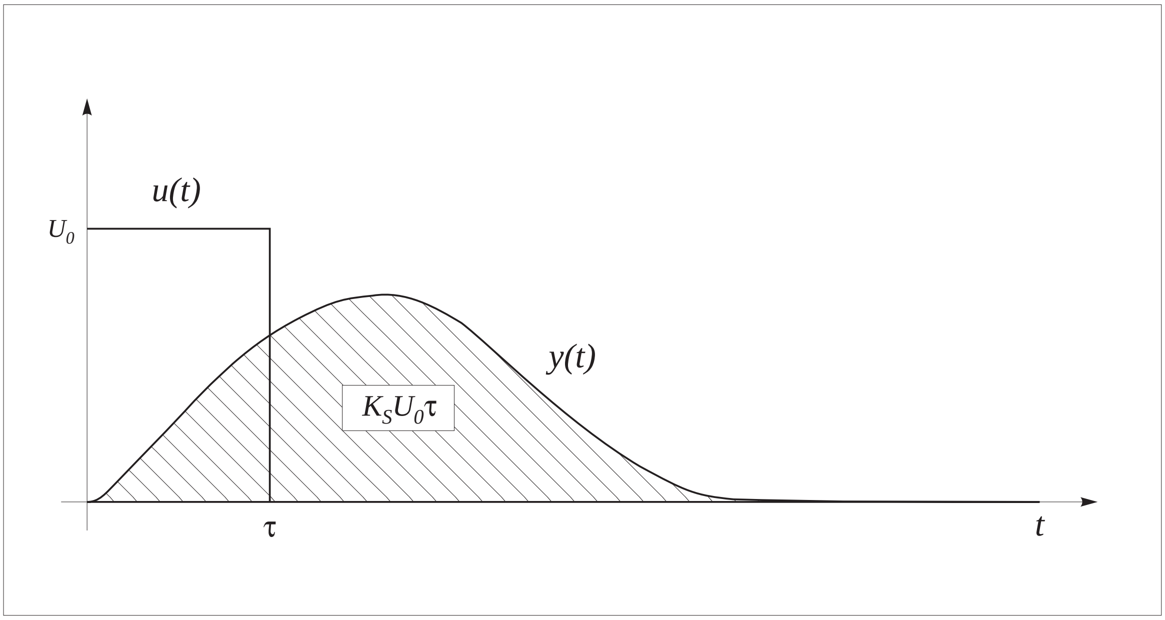

Fig. 8. Typical response for processes without pole or zero in origin

For processes with *zero in origin* (24), K_S can be calculated as

$$K_S = \frac{1}{U_0} \int_0^\infty y(t)dt \ , \tag{33}$$

where the process input is a step function with amplitude U_0:

$$U(s) = \frac{U_0}{s} \tag{34}$$

Typical process response is shown in Fig. 9.

For processes with *pole in origin*, we can determine K_S by using the following input signal *u:*

$$U(s) = \frac{1}{s}\left[U_{\max} - \left(U_{\max} - U_{\min}\right)e^{-\tau_1 s} - U_{\min}e^{-\tau_2 s}\right] \tag{35}$$

Note that

$$\int_0^\infty u(t)dt = 0 \tag{36}$$

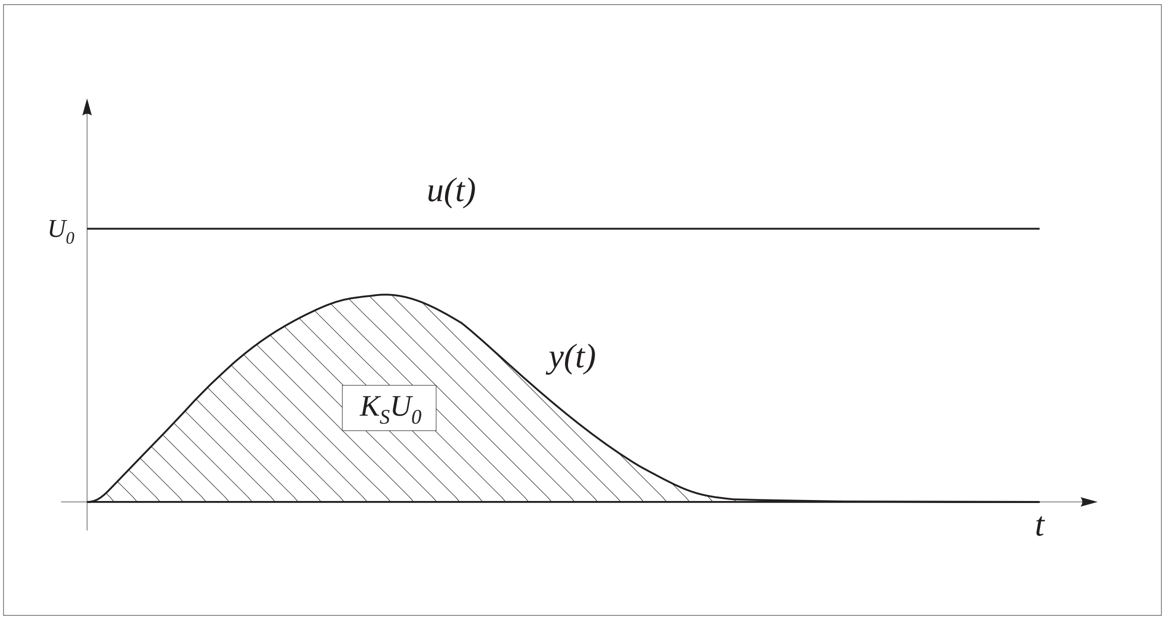

Fig. 9. Typical response of the processes with zero in the origin

From (35) and (36), τ_2 can be expressed as

$$\tau_2 = \tau_1 \frac{U_{\min} - U_{\max}}{U_{\min}} \tag{37}$$

and K_S is expressed as

$$K_S = \frac{2U_{\min}}{\tau_1^{\,2}\left(U_{\min} - U_{\max}\right)U_{\max}} \int_0^{\infty} y(t)\,dt \tag{38}$$

Typical process response on signal (35) is shown in Fig. 10.

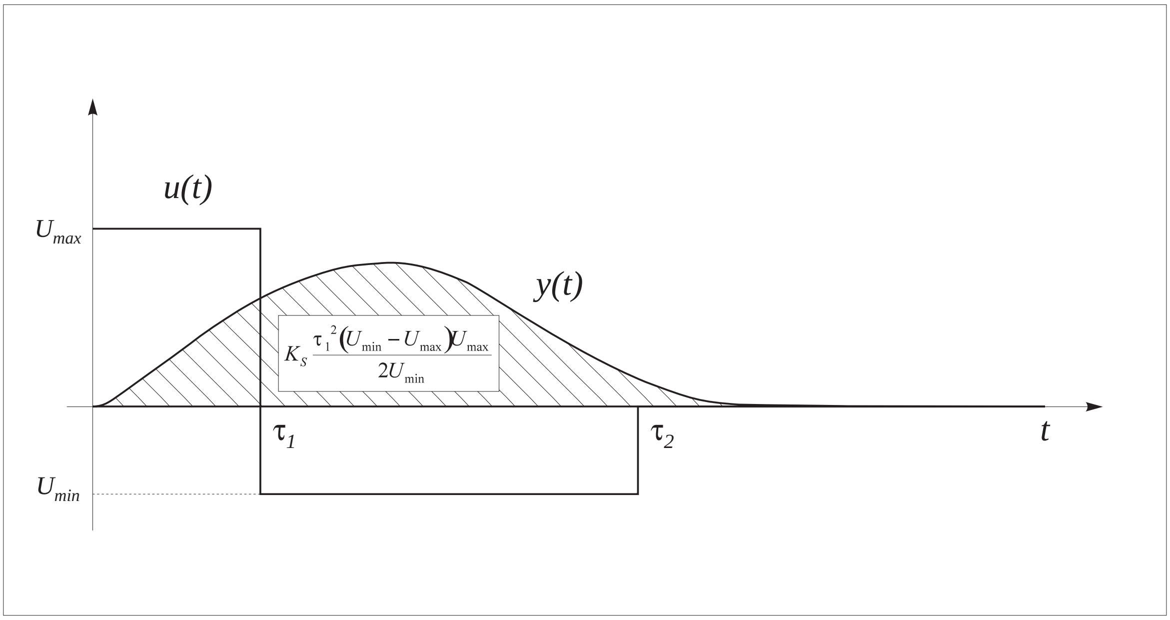

Fig. 10. Typical response of the processes with pole in the origin

Described feed-forward control has some drawbacks. The first is that system must remain stable when closing process input and output (it is not suitable for unstable processes). The second drawback is that process can stay in such closed-loop limited amount of time. Problem appears if process gain G_S is not exactly equal to 1. If this happens, process output will increase or decrease to uncontrolled values. In this case a special mechanism has to be added to switch from "feed-forward" to closed-loop in the right moment. The results of some experiments can be seen in next section.

Due to mentioned drawbacks, we also proposed another method of changing the process set-point. The integration method can be successfully used in optimal switching procedure. For the second order process, switching procedure looks as in Fig. 11.

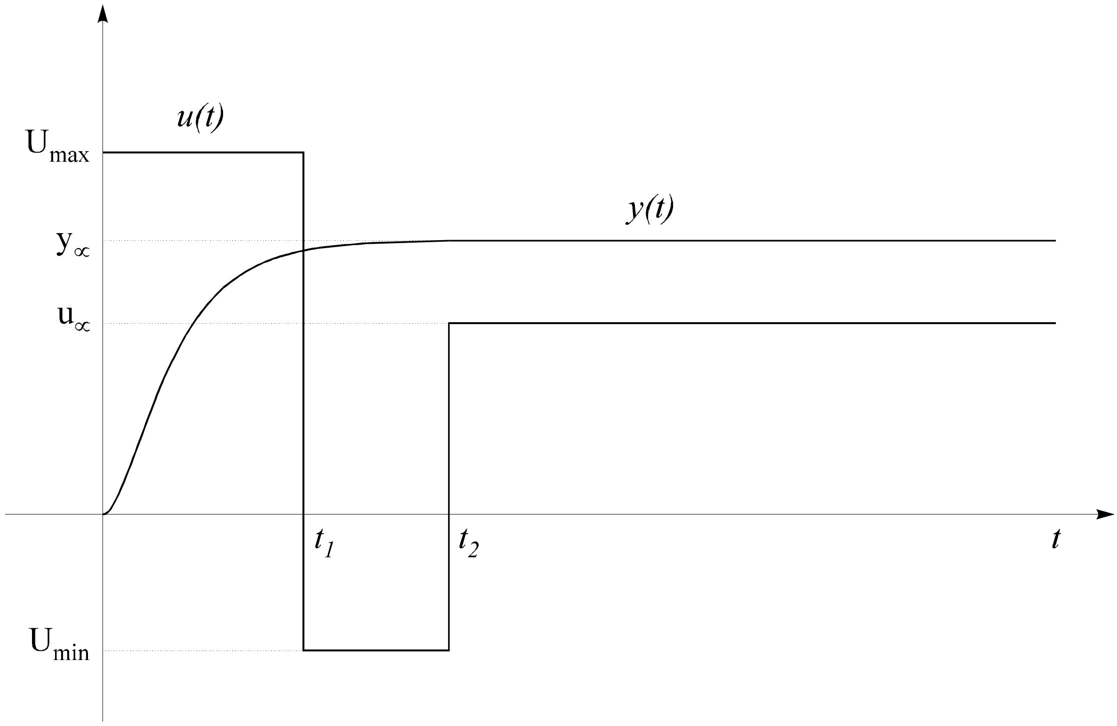

Fig. 11. The optimal switching

For higher order processes, switching procedure need more changes from U_{max} to U_{min} (or some other values of u) and back, so switching times would be at $t=t_1$, $t=t_2$,... $t=t_n$ where n denotes the process order. We will illustrate how to use the knowledge of the surface in finding switching times t_1 and t_2 for the second order process. Here, will use the process with steady-state gain 1, so $u_\infty=y_\infty$.

Usual procedure to find switching times t_1 and t_2 is to settle $u(t_2)=u_\infty$ and all time derivatives of u must be 0 at $t=t_2$ [1]. Because derivatives are not easy to be obtained directly from (noisy) processes, we rather used the integral (7) as additional data.

If we want that for $t\geq t_2$ process output $y(t)=y_\infty$, two conditions have to be fulfilled. First is that $y(t_2)=y_\infty$ and the second condition is that surface between input and output becomes equal to

$$\int_0^{t_2}\left[u(t)-y(t)\right]dt = y_\infty\left(T_{p_1}+T_{p_2}\right) \tag{39}$$

where T_{p1} and T_{p2} are poles of the second order system.

[1] Number of derivatives depends on the process order. For the second order process it is enough to use only the first derivative ($\dot{u}$ (t_2)=0).

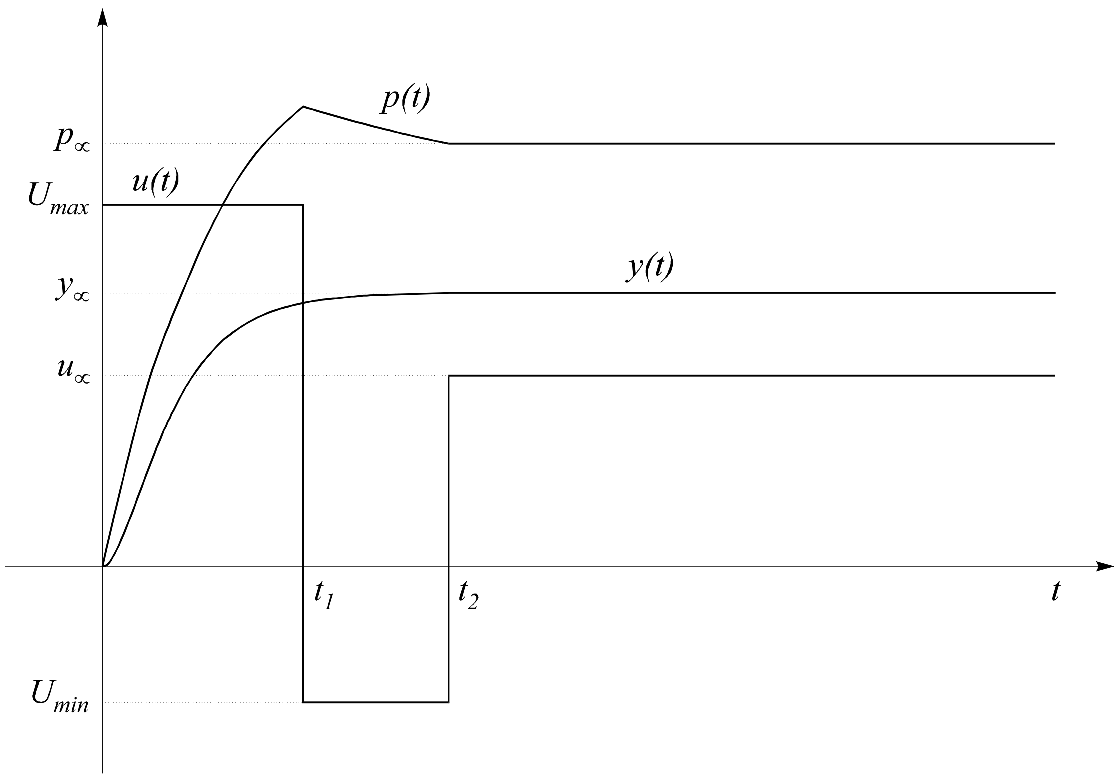

Fig. 12. The optimal switching for the second order system

Fig. 12 shows second order process response if process steady-state gain is equal to 1, where *p(t)* is

$$p(t) = \int_0^t [u(\tau) - y(\tau)] d\tau \tag{40}$$

If the process is linear, we can express *y(t)* when using *u(t),* shown in Fig. 11 as

$$y(t) = U_{\max} y_1(t) + (U_{\min} - U_{\max}) y_1(t - t_1) + (1 - U_{\min}) y_1(t - t_2) \tag{41}$$

where $y_1(t)$ is a process response on the unity-gain step function. Surface *p(t)* can be expressed as

$$p(t) = U_{\max} p_1(t) + (U_{\min} - U_{\max}) p_1(t - t_1) + (1 - U_{\min}) p_1(t - t_2) \tag{42}$$

where $p_1(t)$ is a surface between *u* and *y* where *u* is a unity-gain step function. At $t=t_2$, we want that process output and the surface between *u* and *y* reach the final value.

$$
\begin{aligned}
y(t_2) &= y_\infty \\
p(t_2) &= p_\infty = y_\infty\left(T_{p_1} + T_{p_2}\right)
\end{aligned}
\tag{43}
$$

where T_{p1} and T_{p2} represent process time constants (8). From (41) to (43), we can express

$$
U_{\max} y_1(t_2) + \left(U_{\min} - U_{\max}\right) y_1(t_2 - t_1) = y_\infty \tag{44}
$$

$$
U_{\max} p_1(t_2) + \left(U_{\min} - U_{\max}\right) p_1(t_2 - t_1) = p_\infty \tag{45}
$$

Multiplying (44) with $1/(y_\infty U_{max})$ and (45) with $1/(p_\infty U_{max})$, we get

$$
\frac{y_1(t_2)}{y_\infty} + \frac{U_{\min} - U_{\max}}{U_{\max}} \frac{y_1(t_2 - t_1)}{y_\infty} = \frac{1}{U_{\max}} \tag{46}
$$

$$
\frac{p_1(t_2)}{p_\infty} + \frac{U_{\min} - U_{\max}}{U_{\max}} \frac{p_1(t_2 - t_1)}{p_\infty} = \frac{1}{U_{\max}} \tag{47}
$$

where y_∞ is a new process steady-state. Defining

$$
e(t) = \frac{p_1(t)}{p_\infty} - \frac{y_1(t)}{y_\infty} \tag{48}
$$

and asserting (46) and (47) equal, leads to

$$
e(t_2 - t_1) = \frac{U_{\max}}{U_{\max} - U_{\min}} e(t_2) \tag{49}
$$

Typical time response of e is shown in Fig. 13.

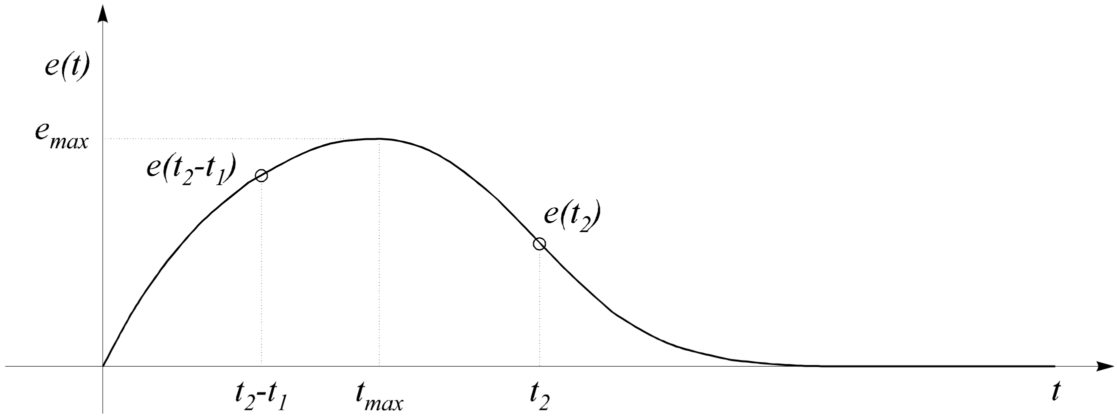

Fig. 13. The difference between the surface p_1/p_∞ and the process response y_1/y_∞

Now, we can find appropriate t_1 and t_2 in the following way. First, we must know the process response on the step function $y_1(t)$. Then we can calculate the surface between u (unity step function) and y as $p_1(t)$. Now, we have to define the change of steady-state process output as y_∞ and choose U_{max} and U_{min}. From (48), we can calculate $e(t)$. After that the searching procedure of t_2 and t_1 starts such that for different t_2 (see Fig. 13) we calculate correspondent $e(t_2-t_1)$ (49) and from already calculated e_1 we find t_2-t_1. The result of t_2-t_1 we put into (44) or (45) and see if it is fulfilled. The procedure of changing t_2 have to be repeated until (44) or (45) adequate.

Note, $t_2>t_1$ and $t_1>0$.

Quicker result can be obtained by calculating the relation between t_2 and t_1 if the second-order process time constants are already known. If this is the case, then

$$t_1 = t_2 - T_2 \ln\left[\frac{U_{\max} - U_{\min}}{U_{\max} e^{-\frac{t_2}{T_{p2}}} + y_\infty - U_{\min}}\right] \tag{50}$$

or

$$t_1 = t_2 - T_1 \ln\left[\frac{U_{\max} - U_{\min}}{U_{\max} e^{-\frac{t_2}{T_{p1}}} + y_\infty - U_{\min}}\right] \tag{51}$$

From (50) and (51) we can see that t_1 is expressed as a function of t_2. The optimisation procedure which finds appropriate t_1 and t_2 consists of optimising t_2 such that (44) or (45) adequate (t_1 is calculated by (50) or (51)).

3. Experiments

In previous section, an information was given, that to move a process (8) from one to another steady state, the surface between *u* and *y* have to be as expressed in (17). For the same process and same difference between steady states, the surface between *u* and *y* must be always constant, no matter what kind of *u* is used. To depict above sentences, we made some examples.

At first we used such signal *u*, that is equal to U_{max} from the time origin to the instant when integral between *u* and *y* become equal to expression (17). Since then, process input *u* is connected to process output *y*, so that expression (17) will be still valid:

$$u(t) = \begin{cases} U_{\max}; & \int_0^t [u(t) - y(t)]dt < a_1 - b_1 + T_d \\ y(t); & \int_0^t [u(t) - y(t)]dt = a_1 - b_1 + T_d \end{cases} \tag{52}$$

Figures 14 and 15 show results when using process (8) with time constants

$$T_{p_1} = 10s, \;\; T_{p_2} = 2s, \;\; T_{p_3} = 1s \tag{53}$$

and U_{max}=10.

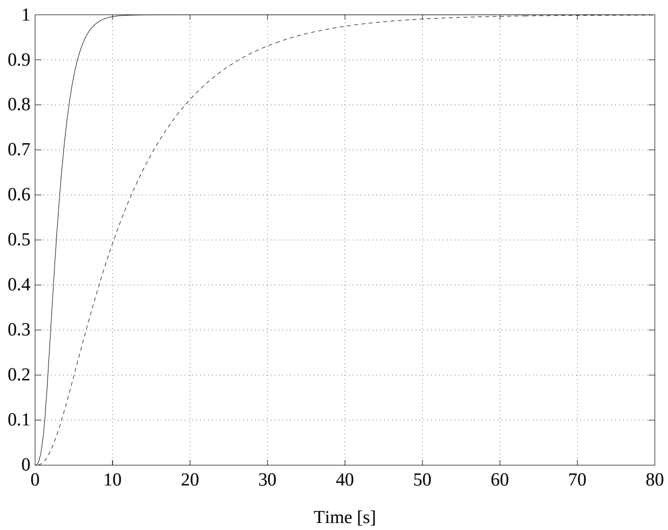

Fig. 14. Process output (y); __ "Feed-forward" response, -- "Step function" response

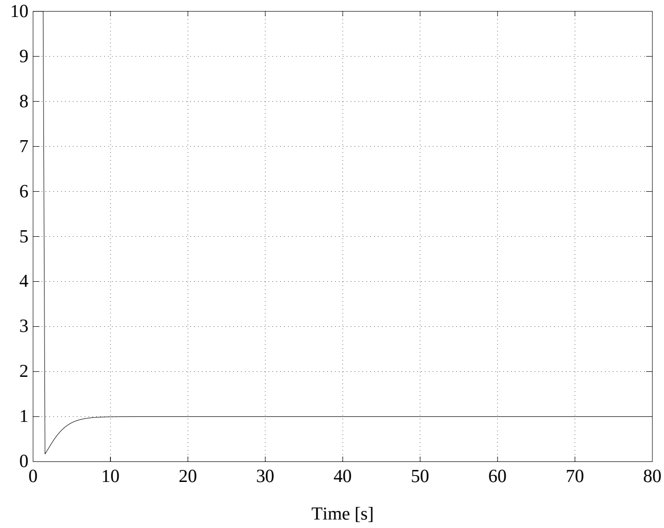

Fig. 15. Process input (u); __ "Feed-forward" response, -- "Step function" response

We can see that process came into higher steady state without overshoot. Next example (Figs. 16 and 17) show results obtained with the same process, but when using U_{max}=1.5. In such case process rise time becomes longer.

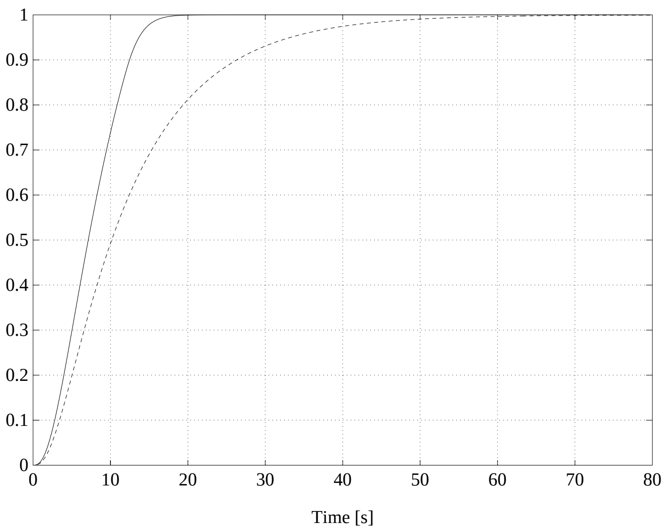

Fig. 16. Process output (y); __ "Feed-forward" response, -- "Step function" response

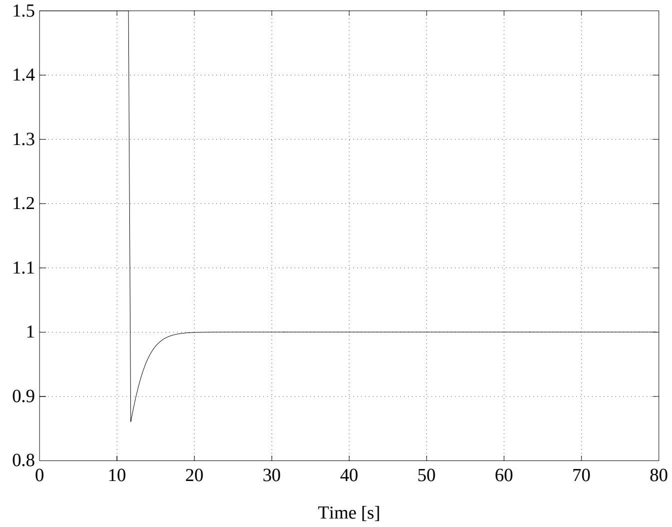

Fig. 17. Process input (u); __ "Feed-forward" response, -- "Step function" response

Next example (see Figs. 18 and 19) show simulation results for process with complex poles:

$$T_{p_1} = (2 + 3i)s, \quad T_{p_2} = (2 - 3i)s, \quad T_{p_3} = 1s \tag{54}$$

and when using U_{max}=10.

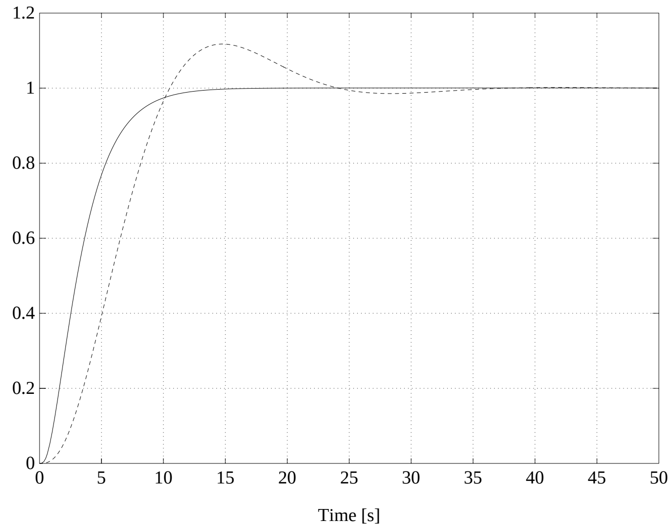

Fig. 18. Process output (y); __ "Feed-forward" response, -- "Step function" response

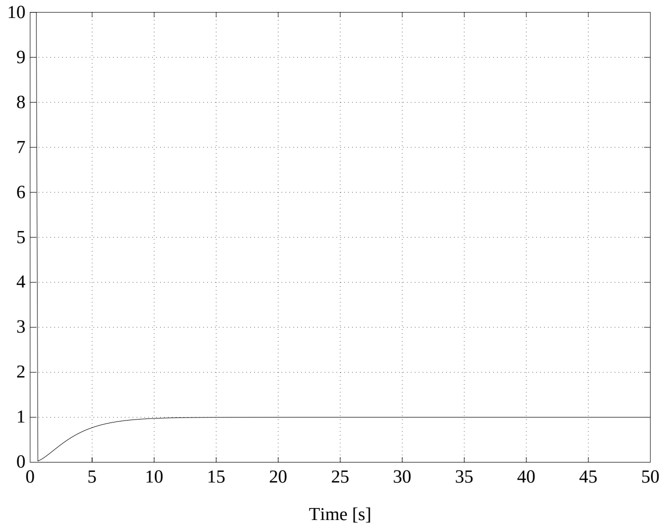

Fig. 19. Process input (u); __ "Feed-forward" response, -- "Step function" response

We can see that "feed-forward" response is again without overshoot. The next two Figures (20 and 21) show results obtained when using process with the following time constants:

$$T_{p_1} = 10s, \;\; T_{p_2} = 2s, \;\; T_{p_3} = 1s, \;\; T_{z_1} = 1.5s \tag{55}$$

and U_{max}=10.

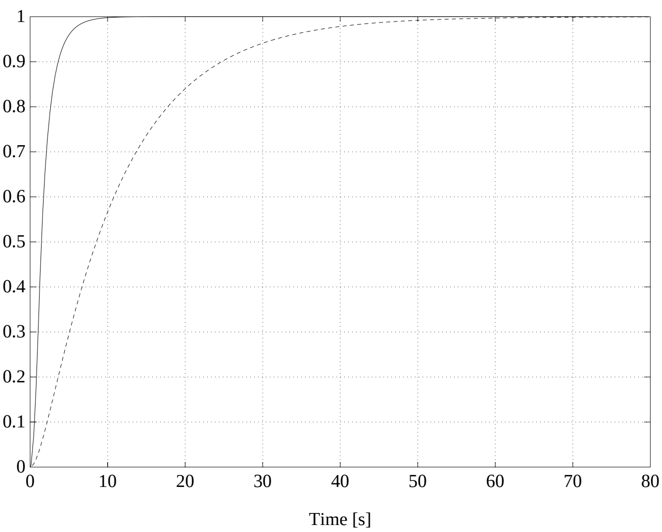

Fig. 20. Process output (y); __ "Feed-forward" response, -- "Step function" response

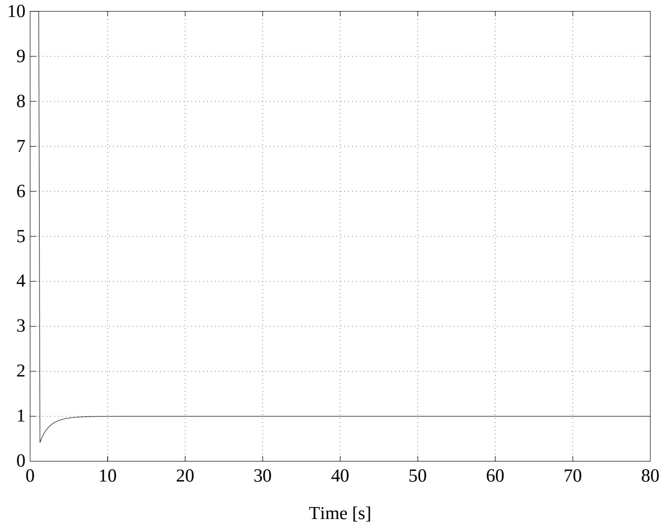

Fig. 21. Process input (u); __ "Feed-forward" response, -- "Step function" response

When increasing zero time constant to the value bigger than T_{p2}, the overshoot appears. Figs. 22 and 23 show results obtained when applying T_{z1}=3.

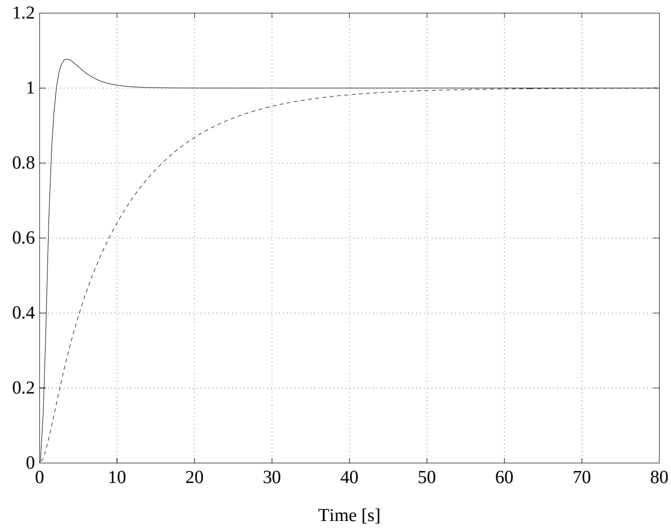

Fig. 22. Process output (y); __ "Feed-forward" response, -- "Step function" response

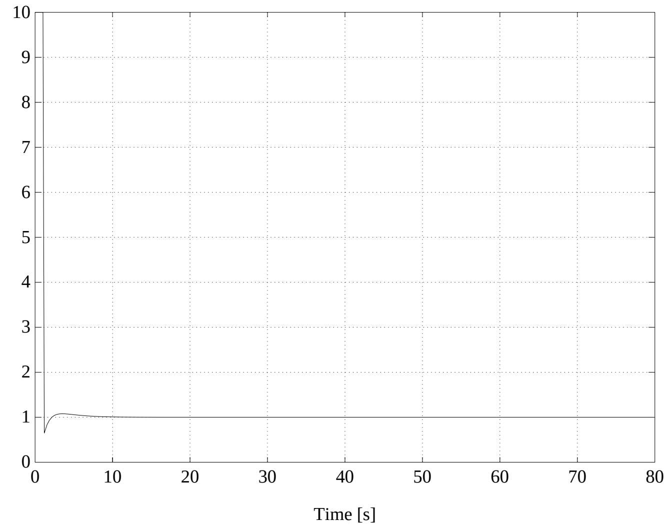

Fig. 23. Process input (u); __ "Feed-forward" response, -- "Step function" response

If process has a delay, it can also produce an overshoot using the feed-forward algorithm. For the system with two poles

$$T_{p_1} = 10s, \;\; T_{p_2} = 2s \tag{56}$$

and a delay, bigger than smaller time constant (2s), an overshoot can appear. Figures 24 and 25 show process response when applying a delay T_d=4s with U_{max}=10.

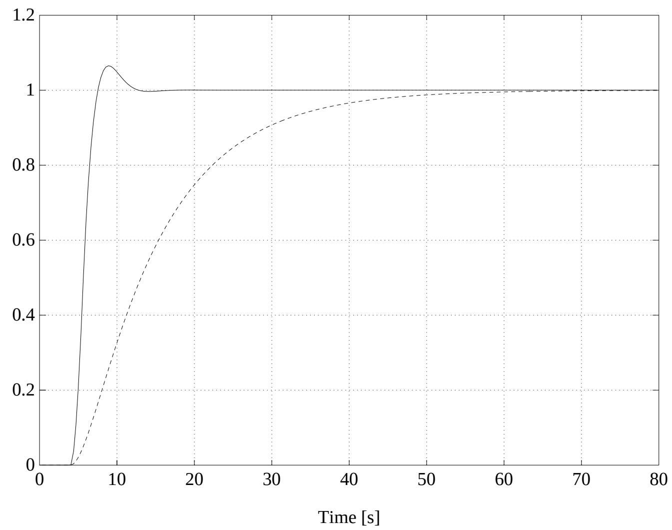

Fig. 24. Process output (y); __ "Feed-forward" response, -- "Step function" response

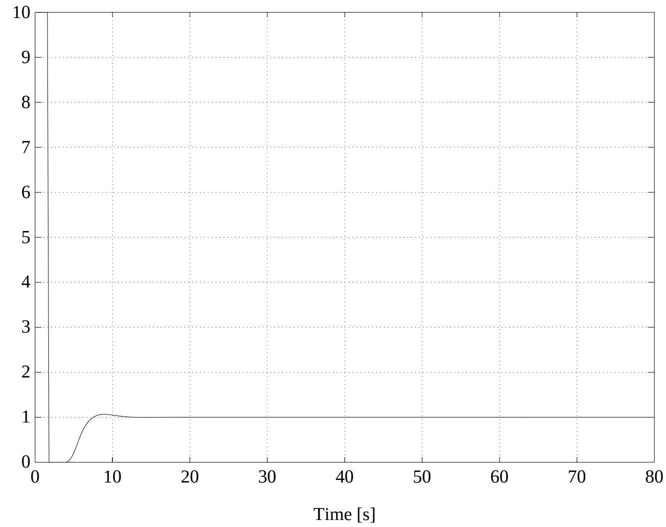

Fig. 25. Process input (u); __ "Feed-forward" response, -- "Step function" response

The last example is performed when we have a non-minimal phase process:

$$T_{p_1} = 10s,\;\; T_{p_2} = 2s,\;\; T_{p_3} = 1s,\;\; T_{z_1} = -2s \tag{57}$$

with U_{max}=10. The result of the simulation are shown in Figs. 23 and 24.

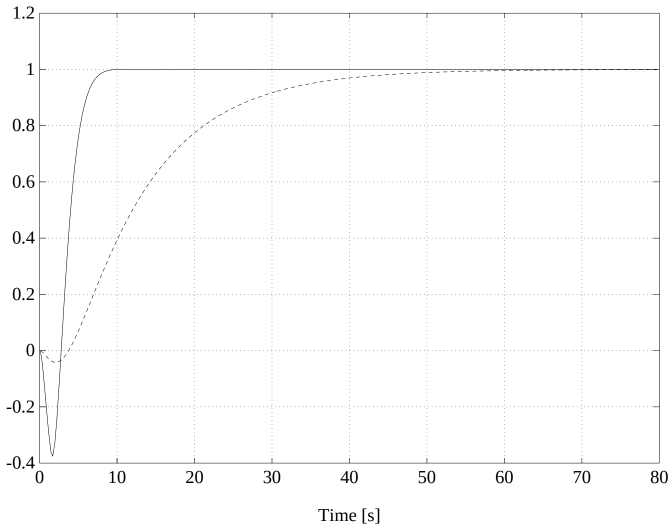

Fig. 26. Process output (y); __ "Feed-forward" response, -- "Step function" response

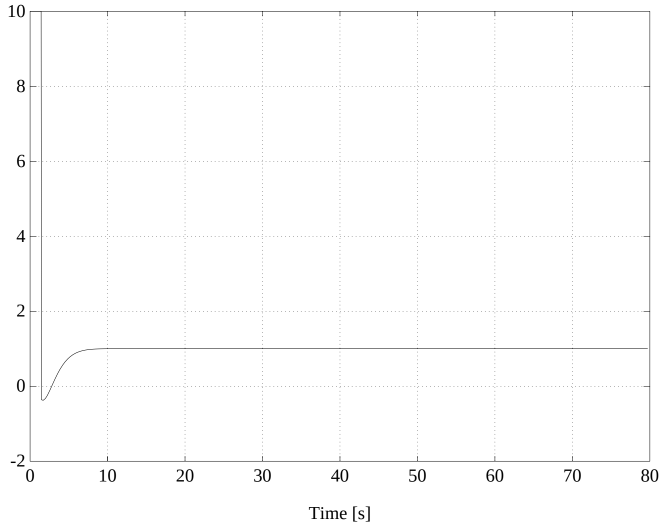

Fig. 27. Process input (u); __ "Feed-forward" response, -- "Step function" response

Next example shows a response of the process with zero in the origin (19). Now, process input is changed:

$$\dot{u}(t) = \begin{cases} U_{\max}; & \int_0^t [\dot{u}(t) - y(t)]dt < a_1 - b_1 + T_d \\ y(t); & \int_0^t [\dot{u}(t) - y(t)]dt = a_1 - b_1 + T_d \end{cases} \tag{58}$$

We used the process constants

$$T_{p_1} = 10s, \quad T_{p_2} = 10s \tag{59}$$

with one zero in origin. The results, when using U_{max}=10, are shown in Figs. 28 and 29.

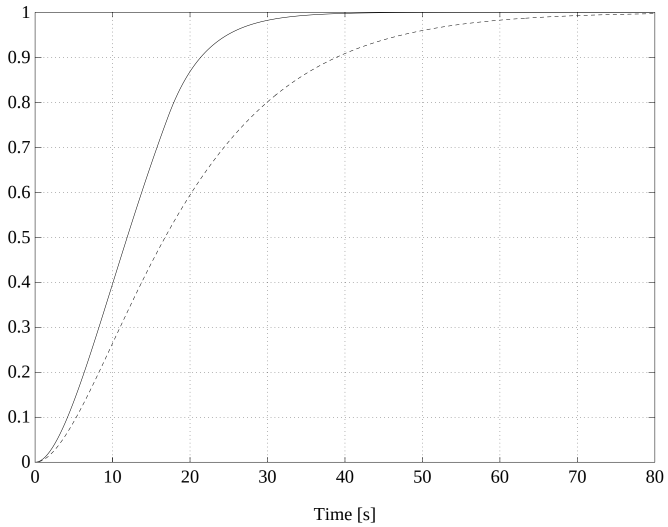

Fig. 28. Process output (y); __ "Feed-forward" response, -- "Step function" response

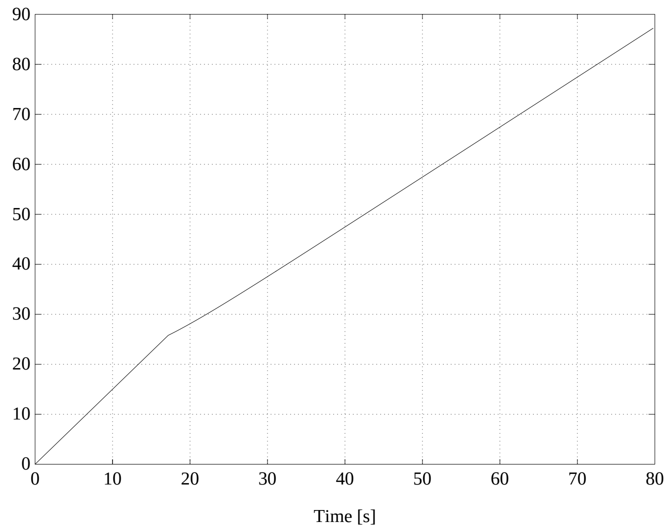

Fig. 29. Process input (u); __ "Feed-forward" response, -- "Step function" response

Drawbacks of proposed steady-state changing method are:

a) When closing the loop from process output (y) to process input (u), the system must be stable.

b) In the case of constant disturbance, process goes toward infinity ($\pm\infty$). The same happens if process steady state gain K_S is not the same as estimated (in our case $K_S \neq 1$). The first example

(Figs. 30 and 31) show results when using process with real steady-state gain K_S=1.1 (10% error in estimation of steady-state gain). Process time constants were as in (53) and U_{max}=10.

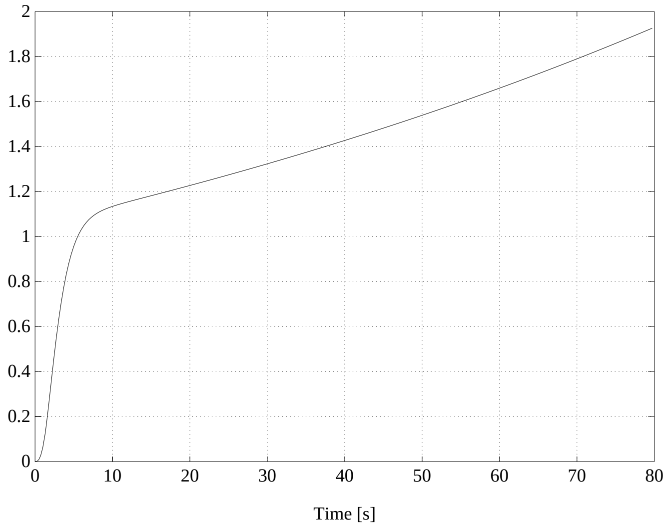

Fig. 30. Process output (y); __ "Feed-forward" response, -- "Step function" response

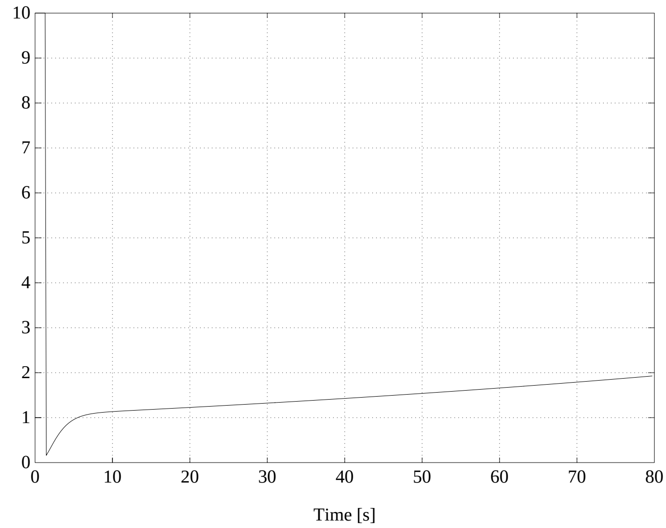

Fig. 31. Process input (u); __ "Feed-forward" response, -- "Step function" response

Next example (Figs. 32 and 33) show process response if real steady-state gain equals to K_S=0.9.

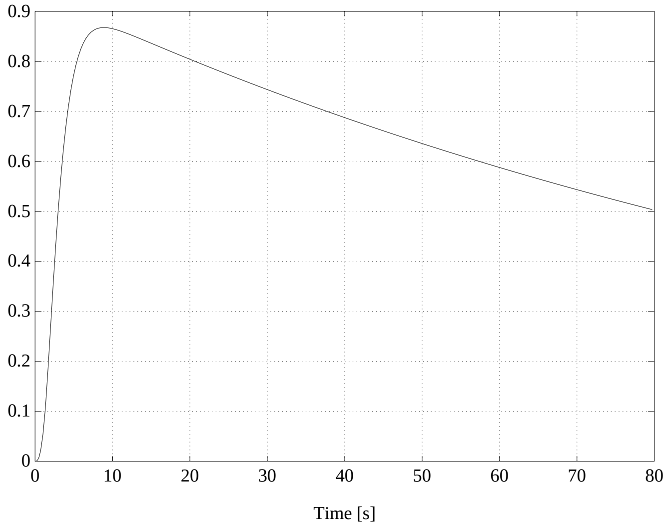

Fig. 32. Process output (y); __ "Feed-forward" response, -- "Step function" response

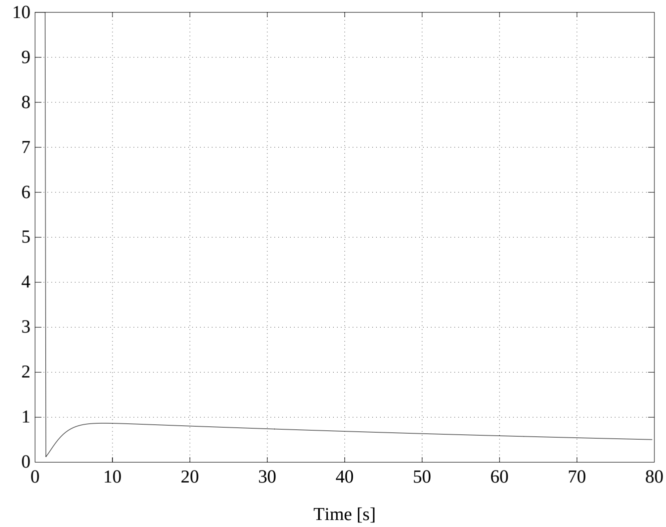

Fig. 33. Process input (u); __ "Feed-forward" response, -- "Step function" response

In both cases we can see process response goes toward infinity or toward zero. It lead us to conclusion that such kind of "control" can be used only with controller which would "take control" when process output will be close to desired. The optimal switching algorithm, adapted for the second order processes, seems to have less drawbacks.

To illustrate the optimal switching algorithm, we prepared two examples. We used the process:

$$G_{PR} = \frac{1}{(1+5s)(1+2s)} \tag{60}$$

and control constants

$$U_{\max} = 5, \;\; U_{\min} = 0, \;\; y_{\infty} = 1 \tag{61}$$

Optimisation procedure gave us the following values of t_1 and t_2:

$$t_1 = 1.87s, \;\; t_2 = 4.09s \tag{62}$$

System response is shown in Figures 34 to 36.

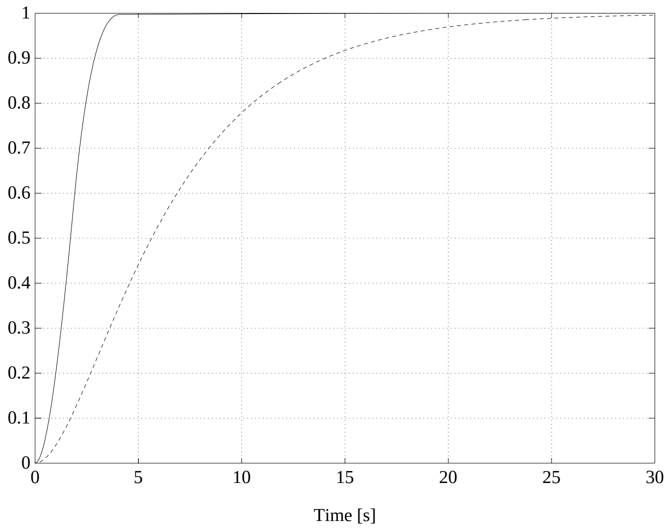

Fig. 34. Process output (y); __ Switching procedure, -- "Step function" response

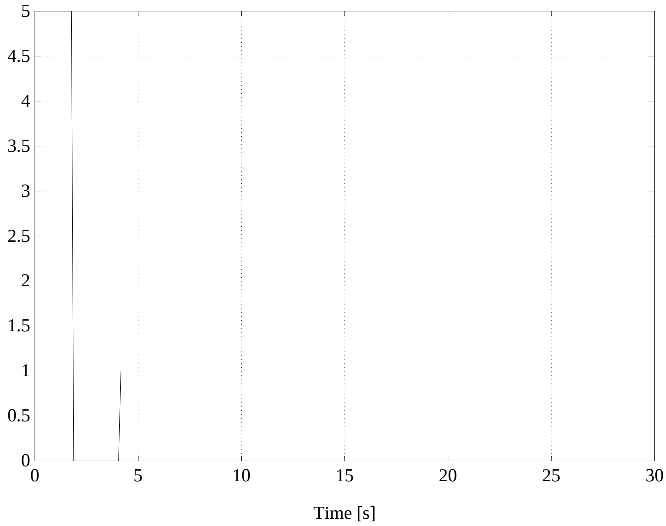

Fig. 35. Process input (u)

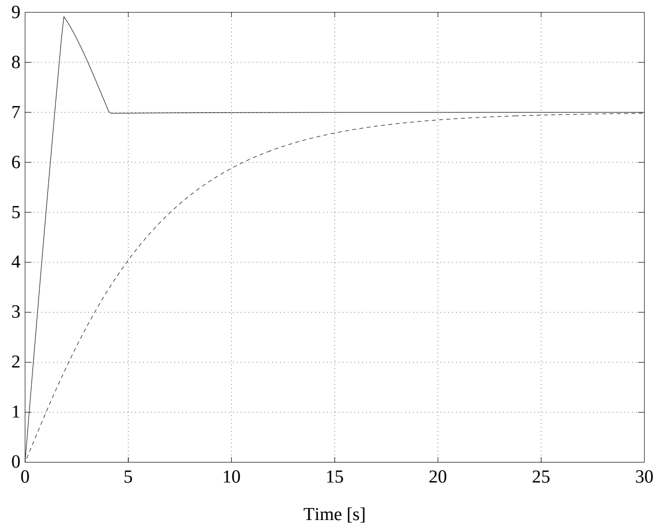

Fig. 36. p(t); __ Switching procedure, -- "Step function" response

Next example was made with the same process as before (60) and with the following control constants:

$$U_{\max} = 5,\ \ U_{\min} = -2,\ \ y_{\infty} = 0.6 \tag{63}$$

Optimisation procedure gave us the following values of t_1 and t_2:

$$t_1 = 1.45s,\quad t_2 = 2.59s \tag{64}$$

System response is shown in Figures 37 to 39.

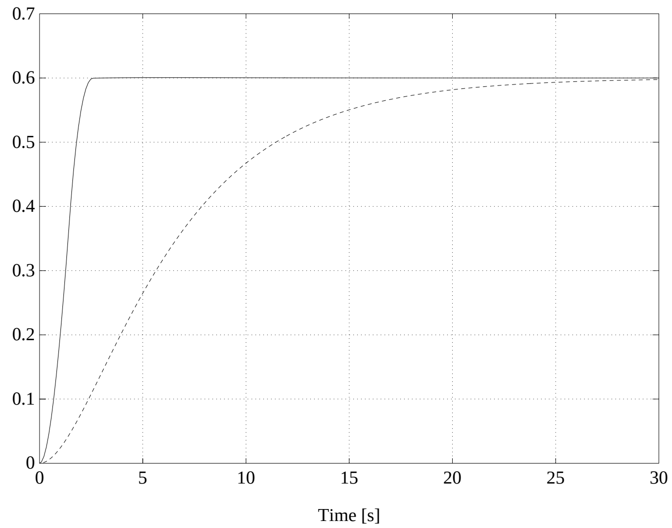

Fig. 37. Process output (y); __ Switching procedure, -- "Step function" response

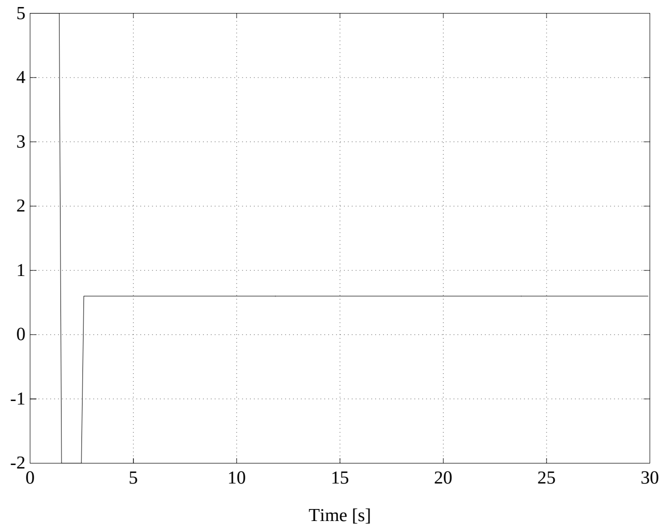

Fig. 38. Process input (u)

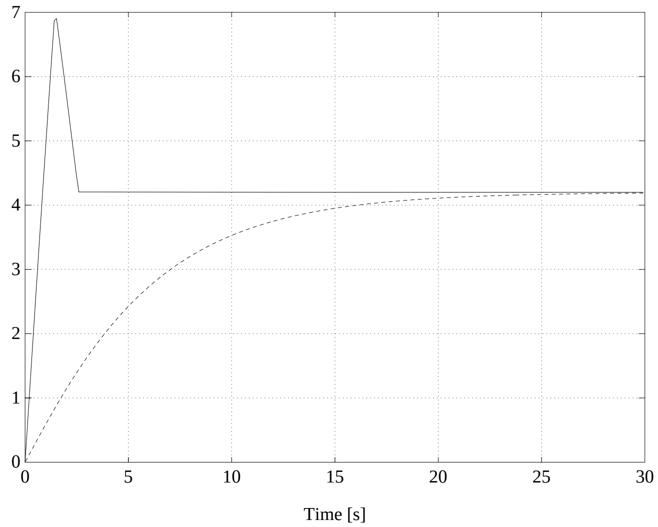

Fig. 39. p(t); __ Switching procedure, -- “Step function” response

4. Conclusions

Results of simulations show advantages and drawbacks of proposed set-point change method. Advantages are that process model can be unknown. All the information we need is to integrate the surface between process input and process output during transition and to determine process gain. Moreover, measurement can be also made in a closed-loop configuration (if the process is already controlled). The second advantage of the integration method is that the optimal switching procedure can also be made for noisy processes where process output derivation does not have to be used. For most usual processes the overshoot equals to zero.

Drawbacks of proposed method are also several. When closing process input with process output, system must stay stable. Another problem arises if process gain is not properly estimated. In such case we must pay more attention to determine right moment to switch from open-loop (feed-forward) to closed-loop configuration. In this case the optimal switching procedure can be used. Some difficulties can also arise if some disturbances appear during transition or if process is non-linear. If the last assumption is true, firstly we have to linearise the process characteristics and then to design appropriate feed-forward control.

Optimal switching procedure by which process output derivative does not have to be used in calculations is also proposed. The process output for the second order system can be made without overshoot.

References

R. Isermann, “**Experimentelle analyse der dynamik von regelsystemen (Identifikation 1)**”, 1971.

R. Isermann, “**Results on the simplification of dynamic process models**”, International Journal of Control, Vol. 19, No. 1, pp. 149-159, 1974.

R. Isermann, “**Identifikation dynamischer Systeme**”, Band II, Springer-Verlag, 1988.

H. Rake, “**Identification: Transient- and Frequency-Response Methods**”, Systems & Control Encyclopedia, Madan G. Singh eds., Vol. 4, pp. 2320-2325, 1987.

V. Strejc, “**Näherungsverfrahren für aperiodische Übergangscharakteristiken**”, Regelungstechnik 7, pp. 124-128, 1959.

D. Vrančić, Y. Peng and Đ. Juričić, “**Some aspects and Design of Anti-Windup and Conditioned Transfer**”, J. Stefan Institute, Ljubljana, Report DP-7169, 1995.