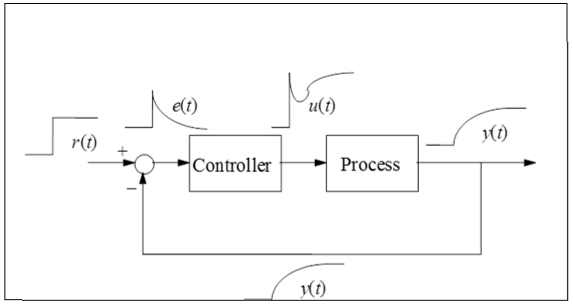
Discrete-time approximations of continuous controllers; discrete PID controller

Discrete approximation of a continuous controller



Aalto University
School of Electrical
Engineering

Approximation of the transfer function

The aim in the approximation of a continuous transfer function is to develop a discrete system, which corresponds to the continuous transfer function. This has earlier been done, but then assuming a zero order or first order hold.

But the hold assumtion is not always valid (the output signal *y* changes arbitrarily and the controller measures the signal as such). Usually a discrete approximation starts from the approximations of the derivative and the integral, assuming that the signal is smooth.

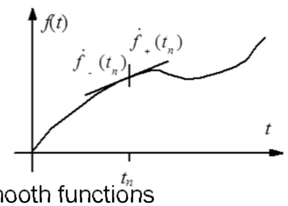
Aalto University
School of Electrical
Engineering

Approximations of the derivative

Starting from the definition of the derivative

$$\frac{df_-(t_n)}{dt} = \dot{f}_-(t_n) = \lim_{\Delta t \to 0} \frac{f(t_n) - f(t_n - \Delta t)}{\Delta t}$$

$$\frac{df_+(t_n)}{dt} = \dot{f}_+(t_n) = \lim_{\Delta t \to 0} \frac{f(t_n + \Delta t) - f(t_n)}{\Delta t}$$



The results are the same for smooth functions

$$\dot{f}(t) = \dot{f}_-(t) = \dot{f}_+(t)$$

Aalto University
School of Electrical
Engineering

Approximations of the derivative

From these definitions the backward and forward approximations are obtained. The ”Euler method” is the same as forward derivation.

$$p \cdot f(t) = \dot{f}(t) = \dot{f}(kh) \approx \frac{f(kh) - f(kh-h)}{h} = \frac{1-q^{-1}}{h} f(kh)$$

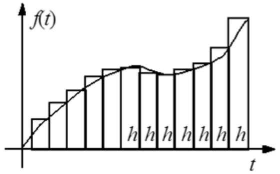
$$p \cdot f(t) = \dot{f}(t) = \dot{f}(kh) \approx \frac{f(kh+h) - f(kh)}{h} = \frac{q-1}{h} f(kh)$$

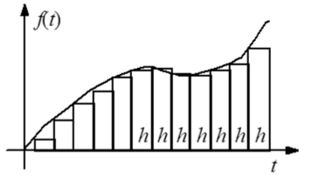
Approximations are

$$p \approx \frac{1-q^{-1}}{h} = \frac{q-1}{qh} \qquad p \approx \frac{1-q^{-1}}{q^{-1}h} = \frac{q-1}{h}$$

Approximations of the integral

Correspondingly, integrals can be approximated as sums





$$\frac{1}{p} f(t) = \int_{-\infty}^{t} f(\tau) d\tau \approx \sum_{i=-\infty}^{k} f(ih) \cdot h \qquad \frac{1}{p} \cdot f(t) = \int_{-\infty}^{t} f(\tau) d\tau \approx \sum_{i=-\infty}^{k-1} f(ih) \cdot h$$

Approximations of the integral

$$\sum_{i=-\infty}^{k} f(ih) \cdot h = \sum_{j=0}^{\infty} q^{-j} \cdot f(kh) h = f(kh) h \sum_{j=0}^{\infty} \left(q^{-1}\right)^{j} \cdot = \frac{h}{1-q^{-1}} f(kh)$$

$$\sum_{i=-\infty}^{k-1} f(ih) \cdot h = \sum_{j=0}^{\infty} q^{-(j+1)} \cdot f(kh) h = q^{-1} h \cdot f(kh) \sum_{j=0}^{\infty} q^{-j} = \frac{q^{-1} h}{1-q^{-1}} f(kh)$$

The same approximations can also be derived recursively. For example

$$\mathrm{Int}_f(t) = \int_{-\infty}^{t} f(\tau) d\tau \approx \sum_{i=-\infty}^{k} f(ih) \cdot h$$

Approximations of the integral

$$\Rightarrow \quad \mathrm{Int}_f(kh) = \int_{-\infty}^{kh} f(\tau) d\tau \approx \sum_{i=-\infty}^{k} f(ih) \cdot h$$

$$\Rightarrow \quad \mathrm{Int}_f(kh+h) \approx \sum_{i=-\infty}^{k+1} f(ih) \cdot h = \sum_{i=-\infty}^{k} f(ih) \cdot h + f(kh+h) \cdot h$$

$$= \mathrm{Int}_f(kh) + f(kh+h) \cdot h$$

$$\Rightarrow \quad \mathrm{Int}_f(kh+h) - \mathrm{Int}_f(kh) = f(kh+h) \cdot h$$

$$\Rightarrow \quad (q-1) \cdot \mathrm{Int}_f(kh) = q f(kh) \cdot h$$

$$\Rightarrow \quad \mathrm{Int}_f(kh) = \frac{qh}{q-1} f(kh) = \frac{h}{1-q^{-1}} f(kh)$$

Approximations of the integral

The integral approximations are

$$\frac{1}{p} \approx \frac{h}{1-q^{-1}} = \frac{qh}{q-1} \quad \text{and} \quad \frac{1}{p} \approx \frac{q^{-1}h}{1-q^{-1}} = \frac{h}{q-1}$$

By comparing to the approximations of the derivative, we see that they are analogous.

$$p \approx \frac{1-q^{-1}}{h} = \frac{q-1}{qh} \qquad p \approx \frac{1-q^{-1}}{q^{-1}h} = \frac{q-1}{h}$$

Aalto University
School of Electrical
Engineering

Approximations of the integral

Let us derive one more integral approximation. The use of the trapetsoidal rule to the integral leads to the Tustin approximation or the bilinear approximation.

The result is:

$$\frac{1}{p} f(t) = \int_{-\infty}^{t} f(\tau) d\tau \approx \frac{1}{2} \sum_{i=-\infty}^{k} f(ih) \cdot h + \frac{1}{2} \sum_{i=-\infty}^{k-1} f(ih) \cdot h$$

$$= \frac{1}{2} \cdot \frac{qh}{q-1} f(kh) + \frac{1}{2} \cdot \frac{h}{q-1} f(kh) = \frac{h}{2} \cdot \frac{q+1}{q-1} f(kh)$$

$$\frac{1}{p} \approx \frac{h}{2} \cdot \frac{q+1}{q-1} = \frac{h}{2} \cdot \frac{1+q^{-1}}{1-q^{-1}} \quad \Rightarrow \quad p \approx \frac{2}{h} \cdot \frac{q-1}{q+1} = \frac{2}{h} \cdot \frac{1-q^{-1}}{1+q^{-1}}$$

Aalto University
School of Electrical
Engineering

Approximations of differential equations

Each derivative operator *p* is substituted by the shift operator *q* or by a function of it.

Approximation of backward differentiation $p \approx \frac{1-q^{-1}}{h} = \frac{q-1}{qh}$

Euler approximation $p \approx \frac{1-q^{-1}}{q^{-1}h} = \frac{q-1}{h}$

Tustin approximation $p \approx \frac{2}{h} \cdot \frac{1-q^{-1}}{1+q^{-1}} = \frac{2}{h} \cdot \frac{q-1}{q+1}$

Aalto University
School of Electrical
Engineering

Approximations of the transfer functions

When playing with transfer functions and pulse transfer functions, then instead of *p* and *q* the variables *s* and *z* are used.

Backward differences $H_{bd}(z) \approx G\left(\frac{1-z^{-1}}{h}\right) = G\left(\frac{z-1}{zh}\right)$

Euler approximation $H_e(z) \approx G\left(\frac{1-z^{-1}}{hz^{-1}}\right) = G\left(\frac{z-1}{h}\right)$

Tustin approximation $H_t(z) \approx G\left(\frac{2}{h} \cdot \frac{1-z^{-1}}{1+z^{-1}}\right) = G\left(\frac{2}{h} \cdot \frac{z-1}{z+1}\right)$

Aalto University
School of Electrical
Engineering

Approximations of the transfer function, example

For a given system develop a discrete approximation and pulse transfer function.

Continuous differential system : $\dot{y}(t) + 2y(t) = u(t)$

The corresponding transfer function: $G(s) = \dfrac{1}{s+2}$

First, the ZOH equivalent is calculated

$$y(k+1) = e^{-2h}y(k) + \tfrac{1}{2}(1-e^{-2h})u(k) \qquad H_{zoh}(z) = \frac{\frac{1}{2}(1-e^{-2h})}{z-e^{-2h}}$$

Aalto University School of Electrical Engineering

Approximations of the transfer function, example

By using the backward difference approximation

$$\dot{y}(t) + 2y(t) = u(t) \quad \Rightarrow \quad \frac{q-1}{qh}y(kh) + 2y(kh) = u(kh)$$

$$\Rightarrow \quad (q-1)y(kh) = -2hqy(kh) + hqu(kh)$$

$$\Rightarrow \quad y(kh+h) - y(kh) = -2hy(kh+h) + hu(kh+h)$$

$$\Rightarrow \quad (1+2h)y(kh+h) = y(kh) + hu(kh+h)$$

$$\Rightarrow \quad y(kh+h) = \frac{1}{1+2h}y(kh) + \frac{h}{1+2h}u(kh+h)$$

Aalto University School of Electrical Engineering

Approximations of the transfer function, example

the Euler approximation

$$\dot{y}(t) + 2y(t) = u(t) \quad \Rightarrow \quad \frac{q-1}{h}y(kh) + 2y(kh) = u(kh)$$

$$\Rightarrow \quad (q-1)y(kh) = -2hy(kh) + hu(kh)$$

$$\Rightarrow \quad y(kh+h) - y(kh) = -2hy(kh) + hu(kh)$$

$$\Rightarrow \quad y(kh+h) = (1-2h)y(kh) + hu(kh)$$

the Tustin approximation

$$\dot{y}(t) + 2y(t) = u(t) \quad \Rightarrow \quad \frac{2}{h}\cdot\frac{q-1}{q+1}y(kh) + 2y(kh) = u(kh)$$

Aalto University School of Electrical Engineering

Approximations of the transfer function, example

$$\Rightarrow \quad 2(q-1)y(kh) = -2h(q+1)y(kh) + h(q+1)u(kh)$$

$$\Rightarrow \quad y(kh+h) - y(kh) = -hy(kh+h) - hy(kh) + \tfrac{1}{2}hu(kh+h) + \tfrac{1}{2}hu(kh)$$

$$\Rightarrow \quad (1+h)y(kh+h) = (1-h)y(kh) + \tfrac{1}{2}hu(kh+h) + \tfrac{1}{2}hu(kh)$$

$$\Rightarrow \quad y(kh+h) = \frac{1-h}{1+h}y(kh) + \frac{\frac{1}{2}h}{1+h}u(kh+h) + \frac{\frac{1}{2}h}{1+h}u(kh)$$

The pulse transfer functions can be derived

$$G(s) = \frac{1}{s+2} \quad \Rightarrow \quad H_{bd}(z) = G\left(\frac{z-1}{zh}\right) = \frac{1}{\left(\frac{z-1}{zh}\right)+2} = \frac{zh}{z-1+2zh}$$

Aalto University School of Electrical Engineering

Approximations of the transfer function, example

$$\Rightarrow \quad H_{bd}(z)=\frac{zh}{(1+2h)z-1}=\frac{\frac{h}{1+2h}z}{z-\frac{1}{1+2h}}$$

For the other approximations

$$G(s)=\frac{1}{s+2} \quad \Rightarrow \quad H_e(z)=G\left(\frac{z-1}{h}\right)=\frac{1}{\left(\frac{z-1}{h}\right)+2}=\frac{h}{z-1+2h}$$

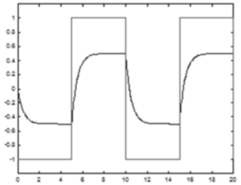
$$G(s)=\frac{1}{s+2} \quad \Rightarrow \quad H_t(z)=G\left(\frac{2}{h}\cdot\frac{z-1}{z+1}\right)=\frac{\frac{\frac{h}{2}}{h+1}z+\frac{\frac{h}{2}}{h+1}}{z+\frac{h-1}{h+1}}$$

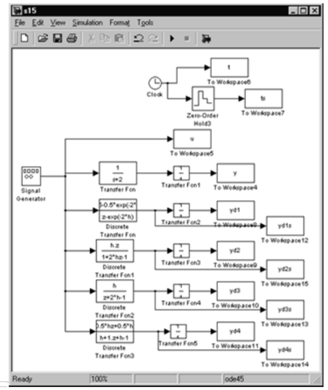
Aalto University School of Electrical Engineering

Approximations of the transfer function, example

Compare the different approximations by simulation.

Try the step function input (ZOH-assumption is valid)

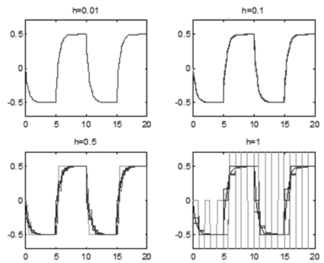




Aalto University School of Electrical Engineering

Approximations of the transfer function, example

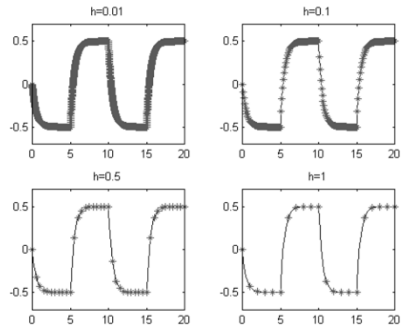
With high sampling frequencies all approximations work well; however, this is not the case for lower sampling rates.



Aalto University School of Electrical Engineering

Approximations of the transfer function, example

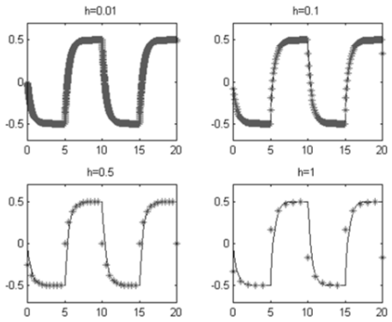
The ZOH-approximation works well for all sampling rates, because the input signal happens to be constant between the sampling instants.



Aalto University School of Electrical Engineering

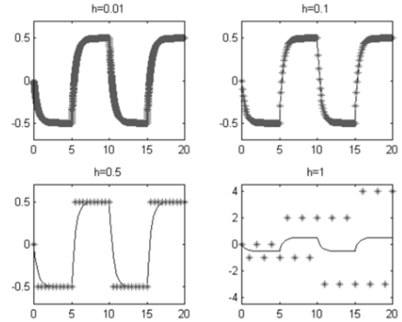
Approximations of the transfer function, example

The backward difference approximation seems to work well.



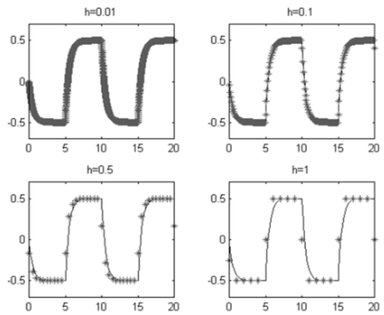
Approximations of the transfer function, example

The Euler approximation gives an unstable response for low sampling rates.



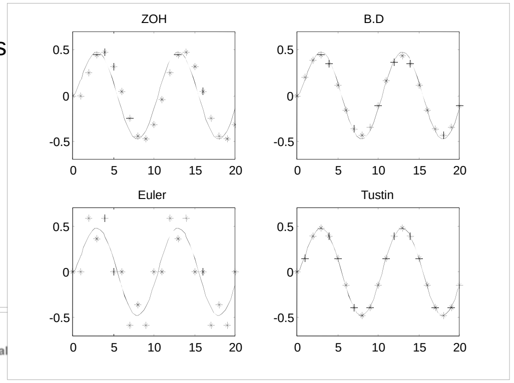
Approximations of the transfer function, example

The Tustin approximation works reasonably well



Approximations of the transfer function, example

For smooth signals all approximations give a good response. A sinusoidal signal with sampling interval h = 1.



Stability of the approximated transfer functions

As shown, the approximation of a stable transfer function may be unstable (in the above example the Euler approximation). Each pole s_P is mapped into the discrete pole z_P according to the approximation.

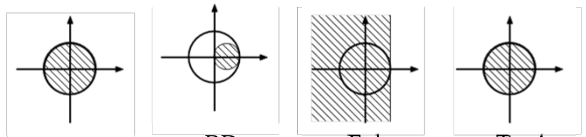
$$z_{P,\mathrm{ZOH}} = e^{s_P h} \qquad z_{P,\mathrm{BD}} = \frac{1}{1 - s_P h} \qquad z_{P,\mathrm{Euler}} = 1 + s_P h \qquad z_{P,\mathrm{Tustin}} = \frac{1 + \frac{1}{2} s_P h}{1 - \frac{1}{2} s_P h}$$

With high sampling rates all the approximated transfer functions behave identically (approach the continuous system); their poles approach the point 1.

$$\lim_{h \to 0} z_{P,i} = 1$$

Stability of the approximated transfer functions

Regardless of the method used, the discretized pulse transfer function is stable only if its poles are located inside the unit circle. It is therefore interesting to consider how the LHP (left half plane) is mapped into discrete domain. In the figure below the shadowed areas denote those points, into which the LHP is mapped.



ZOH BD Euler Tustin

Stability of the approximated transfer functions

Only for the ZOH and Tustin method stability is identical in the continuous and discrete time case. Backward derivation maps stable systems to stable systems, but it also gives to some unstable systems a stable discrete equivalent. The Euler method maps a group of stable continuous systems into unstable equivalents (as shown in the simulations).

Frequency folding

Many control design methods are based on the study of the frequency response (like the analysis of lead and lag compensators). Approximations on the other hand fold the frequency scale, which means that the response in a critical frequency may give inproper information.

Consider an example case. For the Tustin approximation it holds that

$$H_t(z) \approx G\left(\frac{2}{h} \cdot \frac{1 - z^{-1}}{1 + z^{-1}}\right) = G\left(\frac{2}{h} \cdot \frac{z - 1}{z + 1}\right)$$

Frequency folding

Compare the frequency response of the continuous process $G(s)$ to that of a Tustin equivalent $H(z)$.

Continuous response: $F(\omega) = G(i\omega)$

Discrete response: $F(\omega) = H(e^{i\omega h})$

By the Tustin approximation:

$$H(e^{i\omega h}) = G\left(\frac{2}{h}\cdot\frac{e^{i\omega h}-1}{e^{i\omega h}+1}\right) = G\left(\frac{2}{h}\cdot\frac{e^{\frac{i\omega h}{2}}-e^{-\frac{i\omega h}{2}}}{e^{\frac{i\omega h}{2}}+e^{-\frac{i\omega h}{2}}}\right) = G\left(\frac{2}{h}\cdot\frac{i\sin(\frac{\omega h}{2})}{\cos(\frac{\omega h}{2})}\right)$$

$$= G\left(i\cdot\tfrac{2}{h}\tan(\tfrac{\omega h}{2})\right) \qquad G(i\omega) \neq G\left(i\cdot\tfrac{2}{h}\tan(\tfrac{\omega h}{2})\right)$$

Aalto University School of Electrical Engineering

Frequency folding

That means that if the continuous system performance has been optimized at a certain frequency *w'*, the Tustin approximation folds the frequency scale into the frequency *w*, for which it holds:

$$\omega' = \frac{2}{h}\tan\left(\frac{\omega h}{2}\right) \quad\Rightarrow\quad \omega = \frac{2}{h}\tan^{-1}\left(\frac{\omega' h}{2}\right) \approx \omega'\left(1-\frac{(\omega' h)^2}{12}\right)$$

It is easy to modify the Tustin approximation such that the frequency folding is removed at a given frequency w_1.

$$H_{t,pw}(z) = G\left(\frac{\omega_1}{\tan(\omega_1 h/2)}\cdot\frac{z-1}{z+1}\right)$$

Aalto University School of Electrical Engineering

Discrete PID-controller

The discrete PID-controller is the most frequently used controller in process industry today. It can easily be derived from the continuous PID-controller. The "*textbook version*" of the continuous PID-algorithm is

$$u(t) = K\left(e(t) + \frac{1}{T_I}\int_{-\infty}^{t} e(s)ds + T_D\frac{de(t)}{dt}\right) = P(t) + I(t) + D(t)$$

$$\Rightarrow\quad U(s) = G_{PID}(s)E(s) = K\left(1 + \frac{1}{T_I s} + T_D s\right)E(s) = P(s) + I(s) + D(s)$$

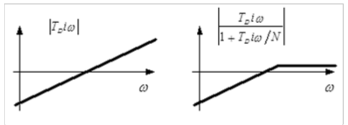
Ideal derivation cannot (and must not) be realized in a PID-controller. Practical systems always contain high frequency disturbances (e.g. white noise), which are attenuated by derivation.

Aalto University School of Electrical Engineering

Discrete PID-controller

Because of that a lag term is usually added to the derivation.

$$T_D s \approx \frac{T_D s}{1 + T_D s/N}$$



Other practical modifications are:

- Derivate only the output (not the reference, not the error signal)

- Only part of the set point (*b*) affects the gain.

Aalto University School of Electrical Engineering

Discrete PID-controller

$$U(s) = K\left(bY_{REF}(s) - Y(s) + \frac{1}{T_I s}(Y_{REF}(s) - Y(s)) - \frac{T_D s}{1 + T_D s/N}Y(s)\right)$$
$$= P_m(s) + I(s) + D_m(s)$$

For the "*textbook version*"

$$\begin{cases} P(t) = Ke(t) \\ I(t) = \dfrac{K}{T_I}\displaystyle\int_{-\infty}^{t} e(s)ds \\ D(t) = KT_D\dfrac{de(t)}{dt} \end{cases} \Rightarrow \begin{cases} P(kh) = Ke(kh) \\ I(kh) = \dfrac{K}{T_I}\displaystyle\sum_{i=-\infty}^{k-1} e(ih)h = K\frac{1}{\frac{T_I}{h}}\Sigma e(kh) \\ D(kh) = KT_D\dfrac{e(kh) - e(kh-h)}{h} = K\frac{T_D}{h}\Delta e(kh) \end{cases}$$

Aalto University
School of Electrical Engineering

Discrete PID-controller

The Euler approximation cannot be used, because the controller would not be causal (a future value would be needed to calculate the derivative). For the integral part the Euler approximation is usually used. A discrete PID-controller is obtained by substituting the integral of the error by a sum, the derivative by a difference, and by dividing the parameters T_I and T_D with the sampling interval h.

$$u(kh) = K\left(e(kh) + \frac{1}{\frac{T_I}{h}}\Sigma e(kh) + \frac{T_D}{h}\Delta e(kh)\right)$$

Aalto University
School of Electrical Engineering

Discrete PID-controller

As a pulse transfer function the controller is

$$U(z) = K\left(1 + \frac{1}{\frac{T_I}{h}} \cdot \frac{1}{z-1} + \frac{T_D}{h}\frac{z-1}{z}\right)E(z) = H_{PID}(z)E(z)$$

The algorithm is called *absolute*, because it calculates the total value of the controller output. That means the calculation of a sum at each time instant, which is not effective algorithmically. The *velocity form* of the algorithm is much more effective, because only the change of the controller output is calculated at each step. Hence it is not necessary to calculate a large sum.

Aalto University
School of Electrical Engineering

Discrete PID-controller

The change becomes: $\Delta u(kh) = u(kh) - u(kh-h)$

$$u(kh) = K\left(e(kh) + \frac{1}{\frac{T_I}{h}}\sum_{i=-\infty}^{k-1} e(ih) + \frac{T_D}{h}\left(e(kh) - e(kh-h)\right)\right)$$

$$\Rightarrow \Delta u(kh) = K\left(e(kh) - e(kh-h) + \frac{1}{\frac{T_I}{h}}\sum_{i=-\infty}^{k-1} e(ih) - \frac{1}{\frac{T_I}{h}}\sum_{i=-\infty}^{k-2} e(ih)\right.$$
$$\left. + \frac{T_D}{h}\left(e(kh) - e(kh-h)\right) - \frac{T_D}{h}\left(e(kh-h) - e(kh-2h)\right)\right)$$

$$= K\left(e(kh) - e(kh-h) + \frac{1}{\frac{T_I}{h}}e(kh-h) + \frac{T_D}{h}\left(e(kh) - 2e(kh-h) + e(kh-2h)\right)\right)$$

$$= K\left(\left(1 + \frac{T_D}{h}\right)e(kh) + \left(\frac{1}{\frac{T_I}{h}} - 2\frac{T_D}{h} - 1\right)e(kh-h) + \left(\frac{T_D}{h}\right)e(kh-2h)\right)$$

Aalto University
School of Electrical Engineering

Discrete PID-controller

The velocity form can also be z-transformed

$$\Delta U(z) = K\left(\left(1+\tfrac{T_D}{h}\right)+\left(\frac{1}{\frac{T_I}{h}}-2\tfrac{T_D}{h}-1\right)\frac{1}{z}+\left(\tfrac{T_D}{h}\right)\frac{1}{z^2}\right)E(z)$$

It must be noticed that the above PID-algorithms are not the only interpretation of a discrete PID-algorithm. If backward integration is used in the integral part, the formula below follows.The structure of the used discrete PID algorithm must always be told together with the tuning values.

$$H_{PID}(z) = G_{PID}\left(\frac{z-1}{zh}\right) = K\left(1+\frac{1}{\frac{T_I}{h}}\frac{z}{z-1}+\tfrac{T_D}{h}\frac{z-1}{z}\right)$$

Discrete PID-controller

The discretization of a practical PID-controller is as straightforward.

$$\begin{cases} P_m(s) = K\left(bY_{REF}(s)-Y(s)\right) \\ I(s) = \dfrac{K}{T_I s}E(s) \\ D_m(s) = -\dfrac{KT_D s}{1+T_D s/N}Y(s) \end{cases} \Rightarrow \begin{cases} P_m(z) = K\left(bY_{REF}(z)-Y(z)\right) \\ I(z) = \dfrac{K}{T_I}\dfrac{h}{z-1} = K\dfrac{1}{\frac{T_I}{h}}\dfrac{1}{z-1} \\ D_m(z) = -KT_D\dfrac{\frac{z-1}{zh}}{1+\frac{T_D}{N}\frac{z-1}{zh}}Y(z) \end{cases}$$

Discrete PID-controller

If a difference equation form is needed

$$\begin{cases} P_m(kh) = K\left(by_{REF}(kh)-y(kh)\right) \\ (1-q^{-1})I(kh) = K\,\tfrac{h}{T_I}\,q^{-1}\left(y_{REF}(kh)-y(kh)\right) \\ \left(1+\tfrac{T_D}{Nh}(1-q^{-1})\right)D_m(kh) = -K\,\tfrac{T_D}{h}(1-q^{-1})y(kh) \end{cases}$$

$$P_m(kh) = \left(by_{REF}(kh)-y(kh)\right)$$

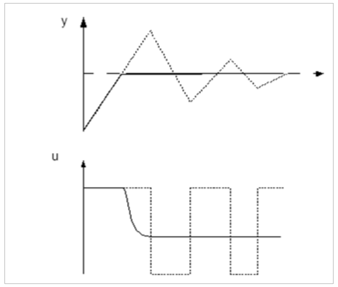
$$I(kh) = I(kh-h)+K\,\tfrac{h}{T_I}\left(y_{REF}(kh-h)-y(kh-h)\right)$$

$$D_m(kh) = \frac{\frac{T_D}{Nh}}{1+\frac{T_D}{Nh}}D_m(kh-h)-\frac{K\frac{T_D}{h}}{1+\frac{T_D}{Nh}}y(kh)+\frac{K\frac{T_D}{h}}{1+\frac{T_D}{Nh}}y(kh-h)$$

Discrete PID-controller

All modifications of the PID-controller can be realized with a discrete PID-controller. The most important ones for the integrator are the antiwindup-operation for saturating actuators, a soft mode change when switching from manual to automatic operation (and vice versa), and bumpless parameter value changes in self-tuning and adaptive PID algorithms. For instance, when an actuator saturates the integrator continues to grow into very high values. After the normal operation conditions have been restored it takes time for the integrator to return to normal values. During this time the controller does not operate well. That phenomenon is called ”integrator windup”. For cure, some ”antiwindup” mechanism is used. The simplest antiwindup is simply to stop integration, when the actuator saturates.

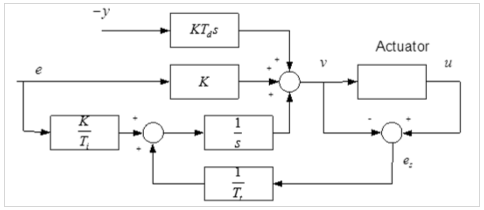
”Integrator windup” and ”antiwindup”



The dashed line shows the output of a controlled system under integrator windup. The solid line shows the response when antiwindup is used.

The antiwindup-operation must not be forgotten when realizing practical control algorithms!

”Integrator windup” and ”antiwindup”



Example of an antiwindup circuit

When the actuator performs normally, an orinary PID operation takes place (note: the derivative part acts only on the output signal). Under saturation the signal e_s deviates from zero, and corrects the value of the integrator towards the ”correct” direction (compare to the figure in the previous page).

Tuning of a discrete PID-controller

Discrete and continuous PID-controller behave identically with a high sampling frequency. Well-known tuning rules of the continuous PID-controllers can then well be used:

- Step response method (the step response of the open-loop process is measured and the dead time, time constant and static gain are measured; the PID tuning parameters are determined based on these values)

- Frequency limit method (the process is controlled (closed loop) by a P-controller; the gain of the controller is increased until the system oscillates in the stability boundary)

Choice of the sampling rate

In most commercial unit controllers the sampling rate is high and fixed (e.g. the sampling interval 200 ms).

If it must and can be chosen, for the PI-controller a rule of thumb says

$$\frac{h}{T_I} \approx 0.1 \cdots 0.3, \quad \frac{h}{L} \approx 0.3 \cdots 1, \quad \frac{h}{T_u} \approx 0.1 \cdots 0.3$$

where L is the dead time and T_u the period of the oscillation (frequency limit method)

For PID the sampling rate must be higher

$$\frac{hN}{T_D} \approx 0.2 \cdots 0.6, \quad N \approx 10, \quad \frac{h}{L} \approx 0.01 \cdots 0.06$$