Control Laboratory: **a.a. 2015/2016**

Lezione 19 — 21 April

Instructor: Luca Schenato *Scribes: M. Baccega, A. Carraro, F. Marchetti*

19.1 Process's discretizing P(s)

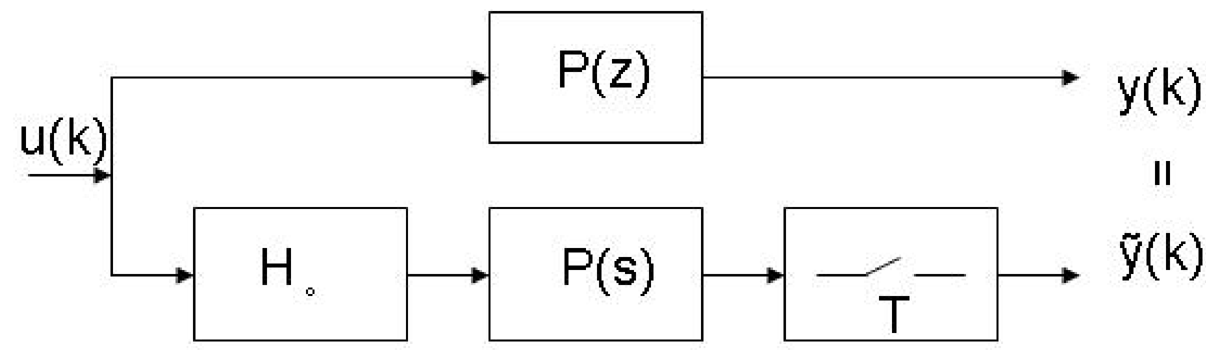


Figura 19.1. Process's discretizing

We consider P(s) writter in space state:

$$\begin{cases} \dot{x(t)} = Ax(t) + Bu(t) \\ y = Cx(t) + Du(t) \end{cases}$$

then we can write x(t) as:

$$x(t) = e^{A(t-\bar{t})}x(\bar{t}) + \int_{\bar{t}}^{t} e^{A(t-\tau}Bu(\tau)d\tau \tag{19.1}$$

Applying the sostitution $t = (k+1)T$ e $\bar{t} = kT$

$$x((k+1)T) = e^{AT}x(kT) + \int_{KT}^{(K+1)T} e^{A(t-\tau)}Bu(\tau)\mathrm{d}\tau \tag{19.2}$$

we know that u(t)=u(kT) for $kT \leq t \leq (k+1)T$ we have:

$$x((k+1)T) = e^{AT}x(kT) + (\int_{KT}^{(K+1)T} e^{A(t-\tau)}B\mathrm{d}\tau)u(kT) \tag{19.3}$$

result that:

$$x((k+1)T) = e^{AT}x(kT) + (\int_{0}^{T} e^{A\tau}B\mathrm{d}\tau)u(kT) \tag{19.4}$$

The following equations are provided:

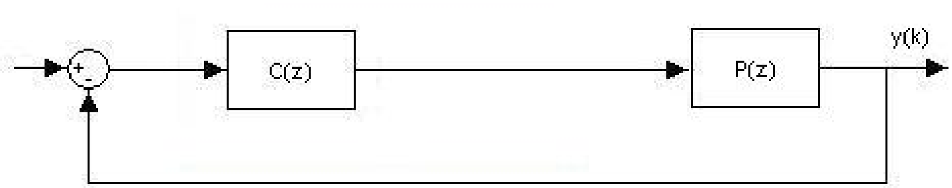
$$A_d = e^{AT} \tag{19.5}$$

$$B_d = \int_0^T e^{A\tau}B\mathrm{d}\tau \tag{19.6}$$

$$C_d = C \tag{19.7}$$

$$D_d = D \tag{19.8}$$

This transformation make the following systems equivalent:



in this way, we have $y(K) = \tilde{y}(kT)$.

The exact disctretization method can be applied directly also to transfer functions in the z-domain using the following transformation:

$$P(z) = (1 - z^{-1})Z\left[\frac{P(s)}{s}\right] \tag{19.9}$$

Continuous	Forward Euler	Backward Euler	Tustin	Exact
A	$A_d = I + TA$	$(I - TA)^{-1}$	$(I + \frac{AT}{2})(I - \frac{AT}{2})^{-1}$	e^{AT}
B	$B_d = TB$	$T(I - TA)^{-1}B$	$(I - \frac{AT}{2})^{-1}B\sqrt{T}$	$\int_0^T e^{A\tau}B\mathrm{d}\tau$
C	$C_d = C$	$C(I - TA)^{-1}$	$\sqrt{T}C(I - \frac{AT}{2})^{-1}$	C
D	$D_d = D$	$D + C(I - TA)^{-1}BT$	$D + C(I - \frac{AT}{2})^{-1}B\frac{T}{2}$	D
$P(s)$	$s = \frac{1}{T}(z - 1)$	$s = \frac{1}{T}\frac{(z-1)}{z}$	$s = \frac{2}{T}\frac{(z-1)}{(z+1)}$	$(1 - z^{-1})Z[\frac{P(s)}{s}]$

Tabella 19.1. Summarizing table for different discretization methods

or equivalently as:

$$P(z) = C_d(zI - A_d)^{-1}B_d + D_d \tag{19.10}$$

The previous table provide different discretization methods that can be applied both in the transfer function domain as well as in the state space domain to pass from a continuous process P(s) to its discrete process P(z) to have a good approximation on the outputs.

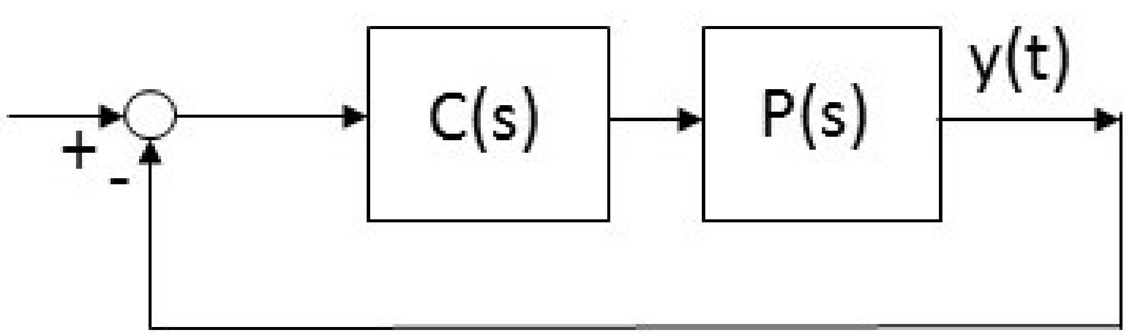


Figura 19.2.

The equivalent block diagram with a discrete controller is shown in Figure 19.3[1].
This isn't the only approach to control design, we can also implement the controller digitally.

[1]Note that in SIMULINK it is not necessary to insert the 'zoh'-block and the 'sampler' block, since SIMULINK automatically performs those steps with the '$C(z)$' block.

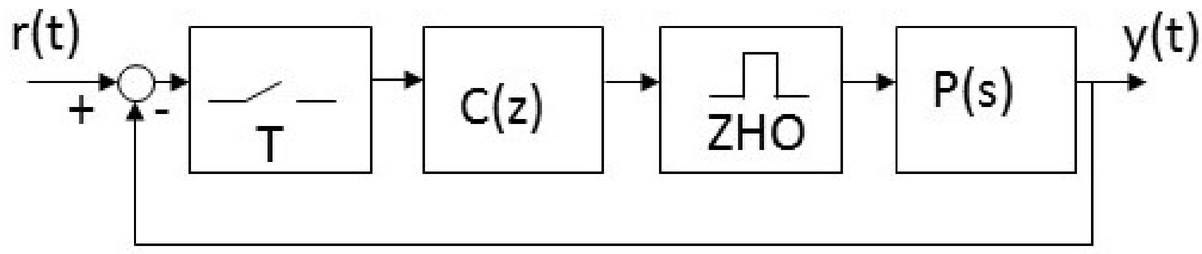


Figura 19.3.

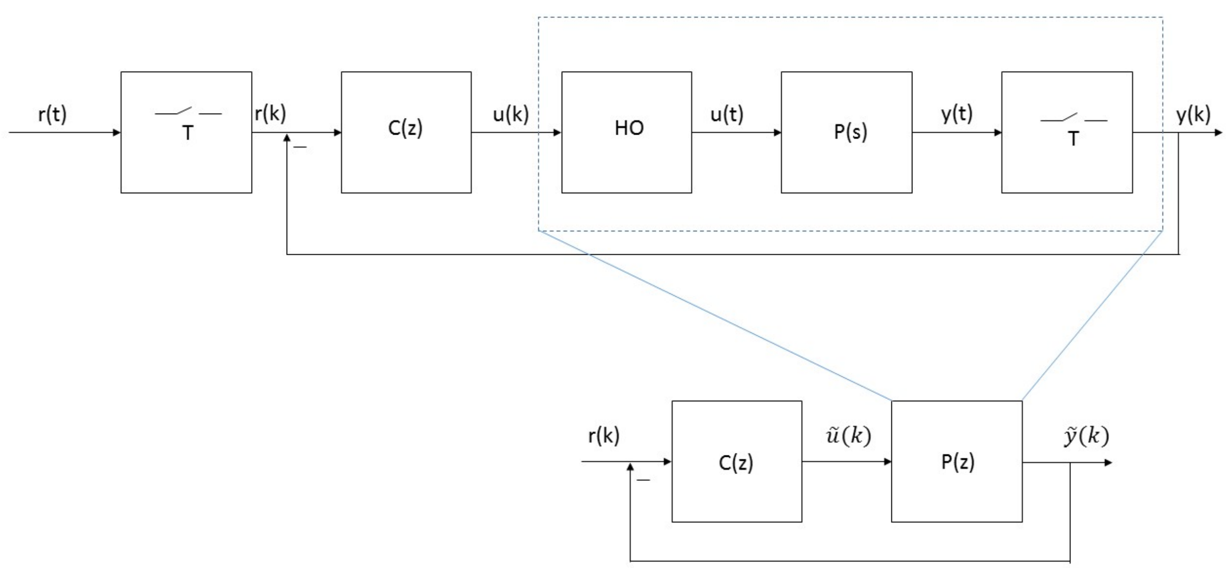


Figura 19.4.

19.1.1 Controller design in discrete time using state space approach

$$\begin{cases} \dot{x}(t) = Ax(t) + Bu(t) \\ y = Cx(t) + Du(t) \end{cases}$$

And the exact approximation is:

$$\begin{cases} x_{k+1} = A_d x_k + B_d u_k \\ y_k = C_d x_k + D_d u_k \end{cases}$$

With:

$$A_d = e^{AT} \qquad B_d = \int_0^T e^{A\tau} B \mathrm{d}\tau$$
$$C_d = C \qquad D_d = D$$

Note: The properties (A, B) reachable and (A, C) observable might not be preserved after the discretization process, in fact (A_d, B_d) and (A_d, C_d) may lose reachability and observability unless the sample period T is sufficiently small.

19.1.2 State feedback, nominal tracking, integral control

$$u(k) = -k_d x(k)$$
$$x_{k+1}{}^I = x_k{}^I + e_k$$

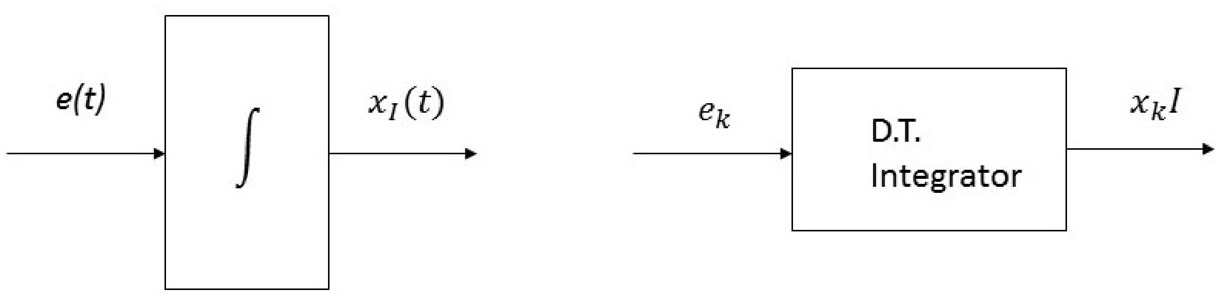


Figura 19.5.

In the scheme below we can observe that the estimated state has the same evolution of the nominal state except for the term $L(y_k - \hat{y}_k)$. In the next lines we will see that the matrix L will be essential to control the dynamics error.

$$\begin{cases} \hat{x}_{k+1} = A_d \hat{x}_k + B_d u_k + L(y_k - \hat{y}_k) \\ \hat{y}_k = C_d \hat{x}_k + D_d u_k \end{cases}$$

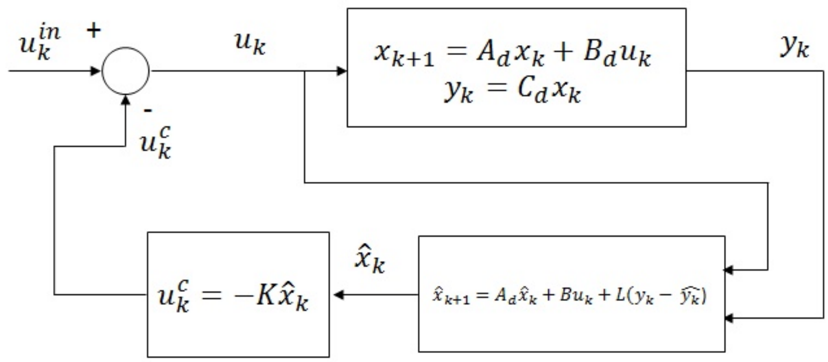


Figura 19.6.

$$e_k = x_k - \hat{x}_k \quad \hat{x}_k = x_k - e_k$$
$$z = \begin{bmatrix} x_k \\ e_k \end{bmatrix}$$

The regulator:

$$\begin{bmatrix} x_{k+1} \\ e_{k+1} \end{bmatrix} = \left[\begin{array}{c|c} A_d - B_d K & B_d K \\ \hline 0 & A_d - LC_d \end{array}\right] \left[\begin{array}{c} x_k \\ \hline e_k \end{array}\right] + \left[\begin{array}{c} B_d \\ \hline 0 \end{array}\right] u_k^{in}$$

$$\begin{aligned} e_{k+1} = x_{k+1} - \hat{x}_{k+1} &= A_d x_k + B_d u_k - (A_d \hat{x}_k + B_d u_k + L(y_k - \hat{y}_k)) \\ &= A_d e_k - L(C_d x_k - C_d \hat{x}_k) \\ &= (A_d - LC_d)e_k \end{aligned}$$

We note that the dynamics of the error does not depend on the matrix $B_d u_k$
$e_k = x + k - \hat{x}_k \quad \Rightarrow \quad \hat{x}_k = x_k - e_k$

$$\begin{cases} x_{k+1} = A_d x_k + B_d u_k \\ u_k = u_k^{in} - u_k^c \\ u_k^c = k\hat{x}_k = kx_k - ke_k \end{cases}$$

$$\begin{aligned} x_{k+1} &= A_d x_k + B_d u_k^{in} - B_d K x_k + B_d K e_k && (19.11) \\ &= (A_d - B_d K)x_k + B_d K e_k + B_d u_k^{in} && (19.12) \end{aligned}$$

$$y = \left[\begin{array}{c|c} C_d & 0 \end{array}\right] \left[\begin{array}{c} x_k \\ \hline e_k \end{array}\right]$$

As we can see, the estimated state is pushed into the feedback block and used to stabilized the system. So we must find K and L to stabilize A_k^c.

$$\begin{aligned} u_k^{in} = constant = u_{DC} \Rightarrow x_k &\to x_{DC} \\ y_k &\to y_{DC} \\ z_k &\to z_{DC} \Rightarrow e_k \to e_{DC} \end{aligned}$$

$$e_{DC} = (A_d - LC_d)e_{DC} \Rightarrow e_{DC} = 0$$

this is true because $(A_d - LC_d)$ is invertible.

$$\begin{aligned} x_{DC} &= (A_d - B_dK)x_{DC} + B_dKe_{DC} + B_Du_{DC} \Rightarrow x_{DC} = (I - (A_d - B_dK))^{-1}B_d \\ y_{DC} &= C_d(I - (A_d - B_dK))^{-1}B_d \\ \bar{N} &= \frac{1}{C_d(I - (A_d - B_dK)^{-1}B_d} \end{aligned}$$

The corresponding transfer function of the close loop is:

$$P_{cc}(s) = \frac{P(s)C(s)}{1 + P(s)C(s)} = \frac{n_c(s)}{\prod_{i=0}(s - p_i^c)} \tag{19.13}$$

For simplicity of discussion and notation, let us assume that the poles p_i are distinct and inside the unit circle(i.e. $\mathbb{R}[p_i^c] < 0$); if we apply the step function the forced output looks like:

$$r(t) = 1(t)$$

$$y_f(t) = \alpha_0 + \sum_{i=1}^{n} \alpha_i e^{p_i^c t}$$

Where α_i are be the coefficients associated to the modes of the system $e^{p_i^c t}$.
After the discretization we obtain

$$P_{cc}(z) = \frac{P(z)C(z)}{1 + P(z)C(z)} = \frac{n_d(z)}{\prod_{i=0}(z - p_i)} \tag{19.14}$$

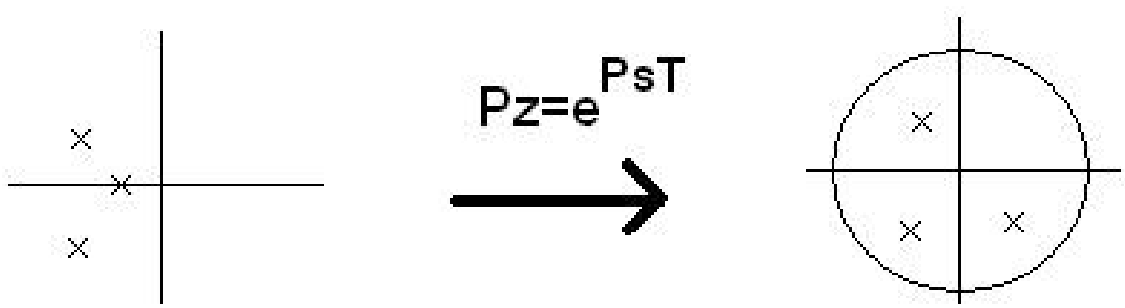
with the $|p_i^d| < 1$, and with $\alpha_i \simeq \beta_i$

$$\begin{aligned} y_k &= \beta_0 + \sum_{i=1}^{n} \beta_i (p_i^d)^k \\ &\simeq \alpha_0 + \sum_{i=1}^{n} \alpha_i (p_i^d)^k \end{aligned}$$

so the forced output will be approximately:

$$
\begin{aligned}
y_f(kT) &\approx \alpha_0 + \sum_{i=1}^{n} \alpha_i e^{p_i^c kT} \\
&= \alpha_0 + \sum_{i=1}^{n} \alpha_i (e^{p_i^c T})^k
\end{aligned}
$$

so if we set $p_i^d = e^{p_i^c T}$ we should expect a similar output between the continuous systems and the corresponding discretized system, at least on the sampling instants $t = kT$.



So we find the relationship between poles in continuos time and in descrete time. If we have C(s), and we know where we would like to place the poles of the closed loop system on s-plane, we can design directly the controller in discete time throught the map $p_i^d = e^{p_i^c T}$. We have to pay attention to the fact that a too large T provide poor performances while a to small one may forces some poles out of the unit circle. The following relation provides a guidance for the choice of the sampling period (whenever possible) and guarantees a good compromise between numerical robustness and approximation fidelity:

$$
\frac{1}{1000} t_r \leq T \leq \frac{1}{20 \sim 30} t_r
$$

where t_r is the desired raising time.