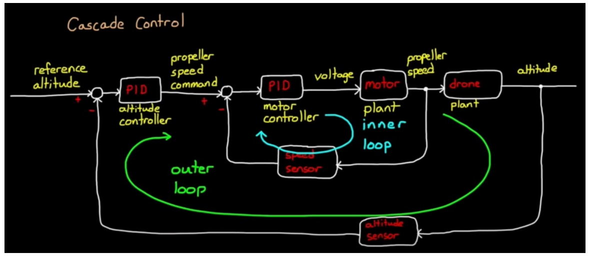
Model-based design of digital controllers

Hugues GARNIER

hugues.garnier@univ-lorraine.fr

The e-mail of the day



From: XXX.YYYY@etu.univ-lorraine.fr

Subject: Request for advice on controller design

Date: February 15, 2023 at 23:12:51 UTC+1

To: Hugues Garnier

Good morning, sir,

I hope you are well. I'm contacting you because I'd like your advice on designing a controller to control the vertical speed of a drone.

I'm currently working on my study project. The aim is to implement standard controllers such as P, PI and PID. But what bothers me in my conception of PI and PID correctors is that there is a delay in the transfer function.

My transfer function looks like:

G(z)= z^(-2) *(-0.023608/(z-0.6065))

So I'd like to determine the Kp, Ki and Kd gains with a concrete method so that I can explain my reasoning. I had thought of the pole placement method, but the calculations quickly become very complicated. I don't have any other ideas in mind, apart from a trial-and-error approach, but I'd like to avoid this method.

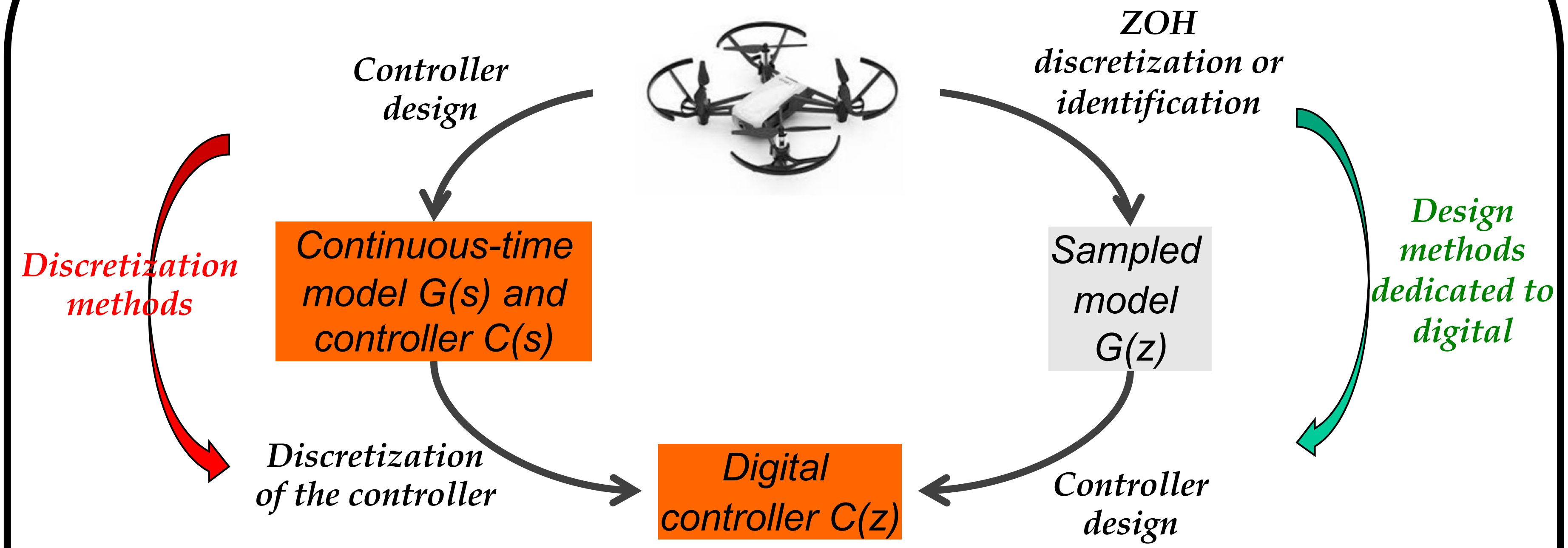
Could you tell me another way of designing these controllers?

Thank you very much for your reply.

Best regards,

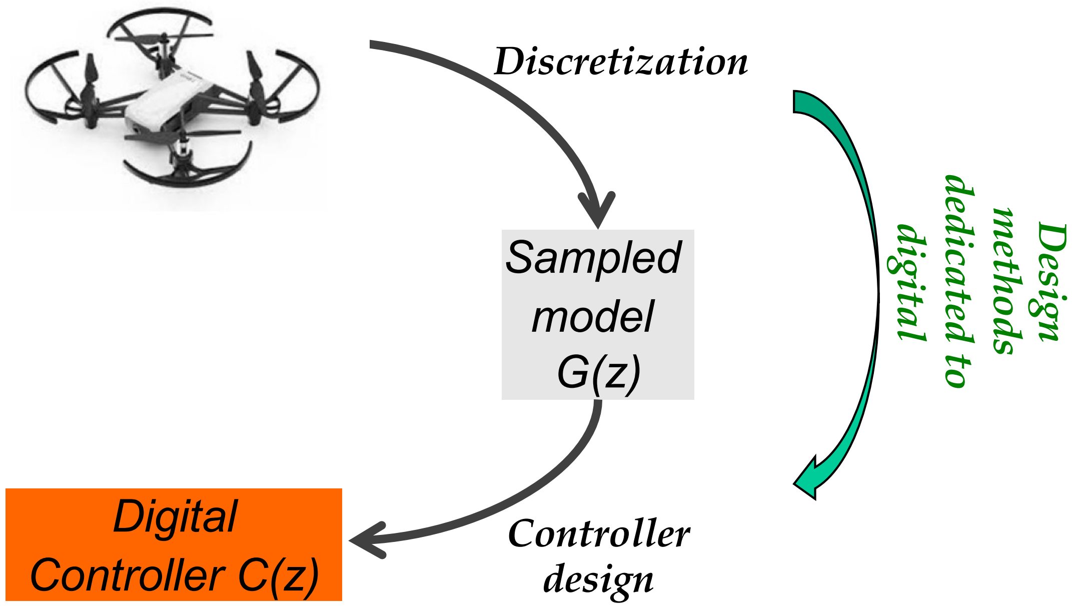
XXX YYYY

Approaches for model-based digital control design



- Two types of approach:
  - Digital methods
  - Methods for discretizing a continuous-time controller (including PID)

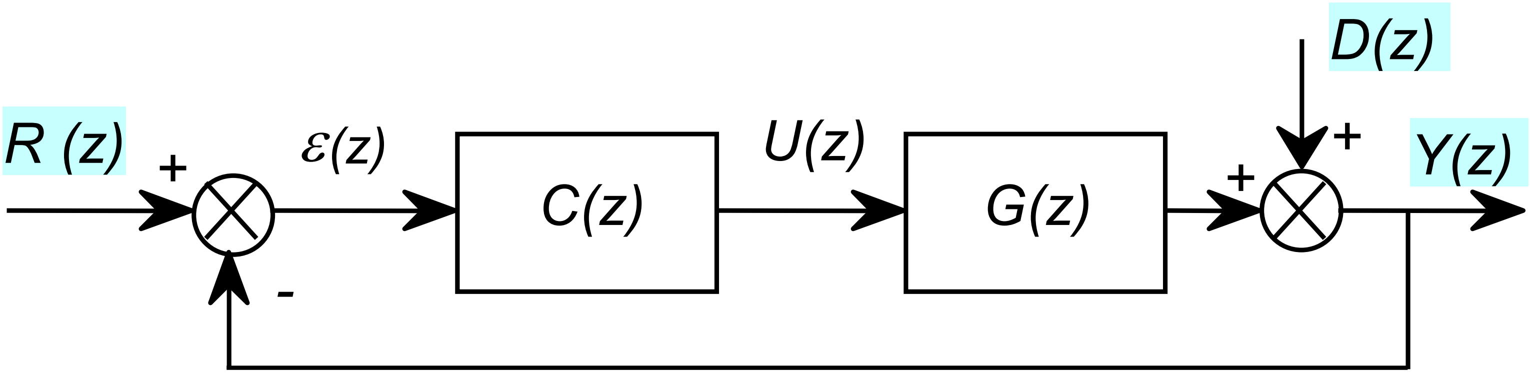
Approaches for model-based digital control design



- These digital methods are preferred when
  - *Fast sampling is not possible*
  - A sampled model *G(z)* is determined from *G(s)* or directly identified from input/output data
  - The control system is designed based on the sampled model
    - Examples: RST control, internal model control, predictive control

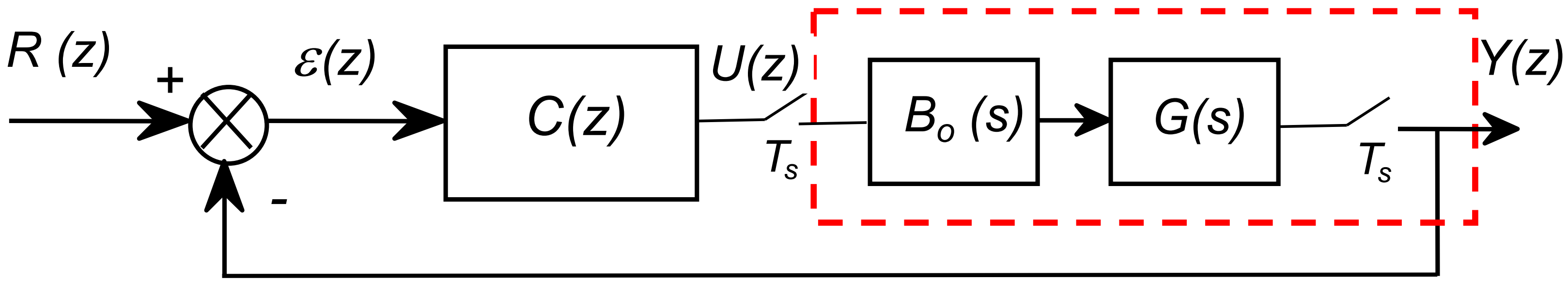
Closed-loop digital control block-diagram

- The search for a control law (and therefore for $C(z)$) using a full digital approach is based on:
  - a sampled model $G(z)$ of the cascaded ZOH + sampled model + sensor + sampler components
  - the type of external signals: reference $R(z)$, disturbance $D(z)$

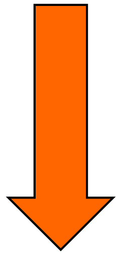


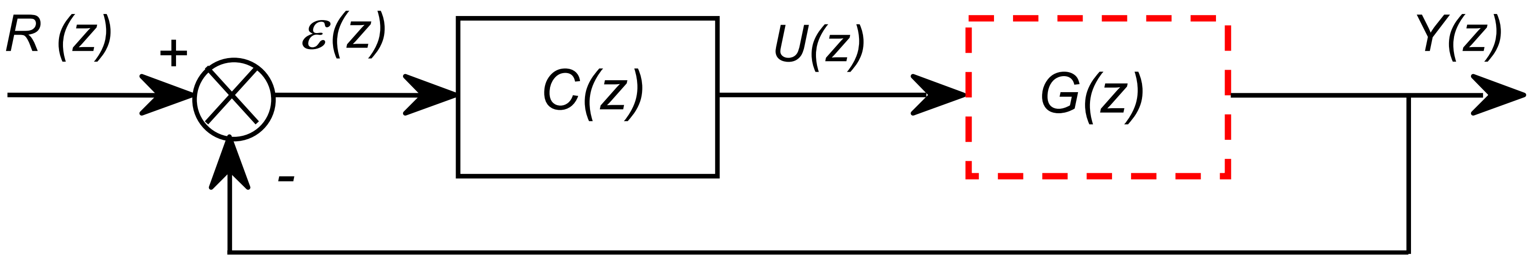
$$Y(z) = \frac{C(z)G(z)}{1+C(z)G(z)} Y_c(z) + \frac{1}{1+C(z)G(z)} D(z)$$

Control design by digital methods



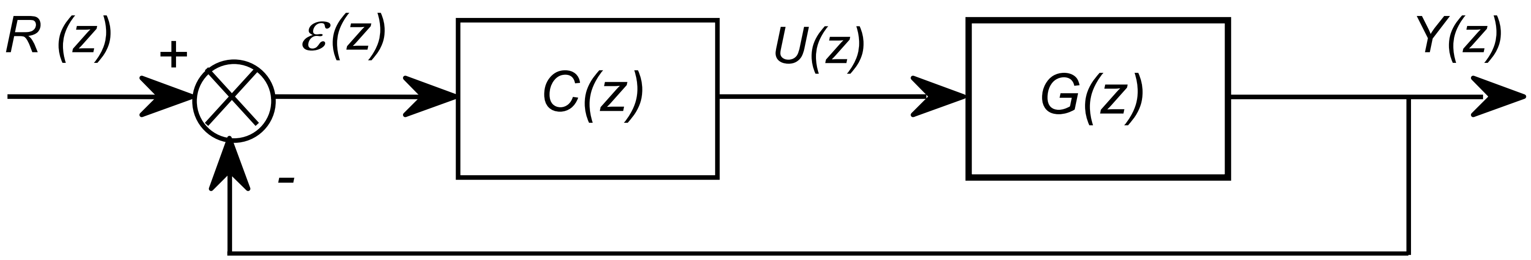
1. Determination of *G(z)* by discretization of *G(s)* *(ZOH method) or* identification from input/output data





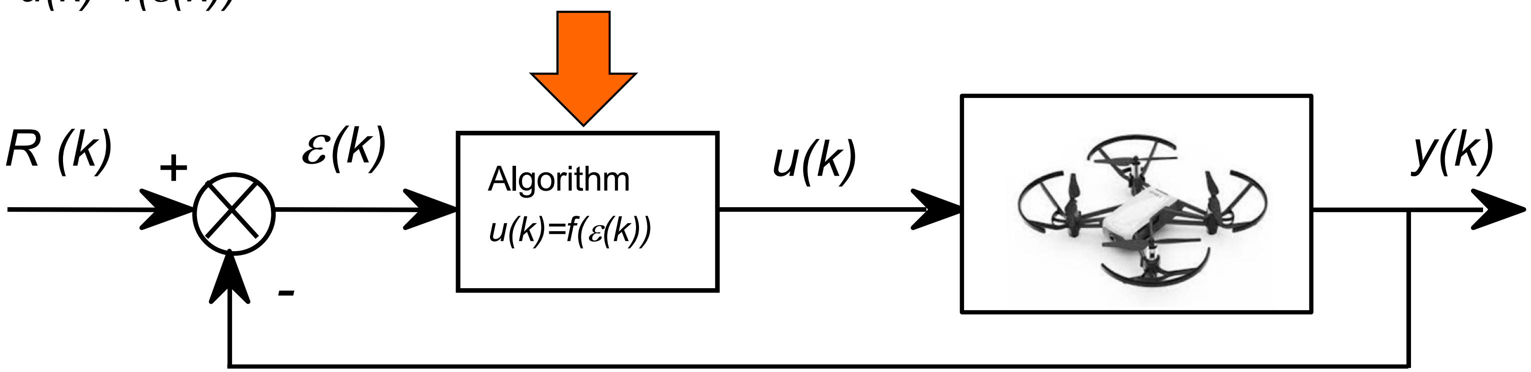
Control design by digital methods

2. Determination of *C(z)* from *G(z)*. Different approaches exist



3. Implementation of the control algorithm

$u(k)=f(\varepsilon(k))$



Digital control design using the reference model method

- Principle: impose that the closed-loop transfer function behaves as a desired response of a transfer function $F_{ref}(z)$ *(often 2nd order)*
- Workflow
  1. Determine a model *G(z)* by discretization of *G(s) or* identification
  2. Determine the closed-loop transfer function

$$F_{BF}(z) = \frac{C(z)G(z)}{1 + C(z)G(z)} = F_{ref}(z)$$

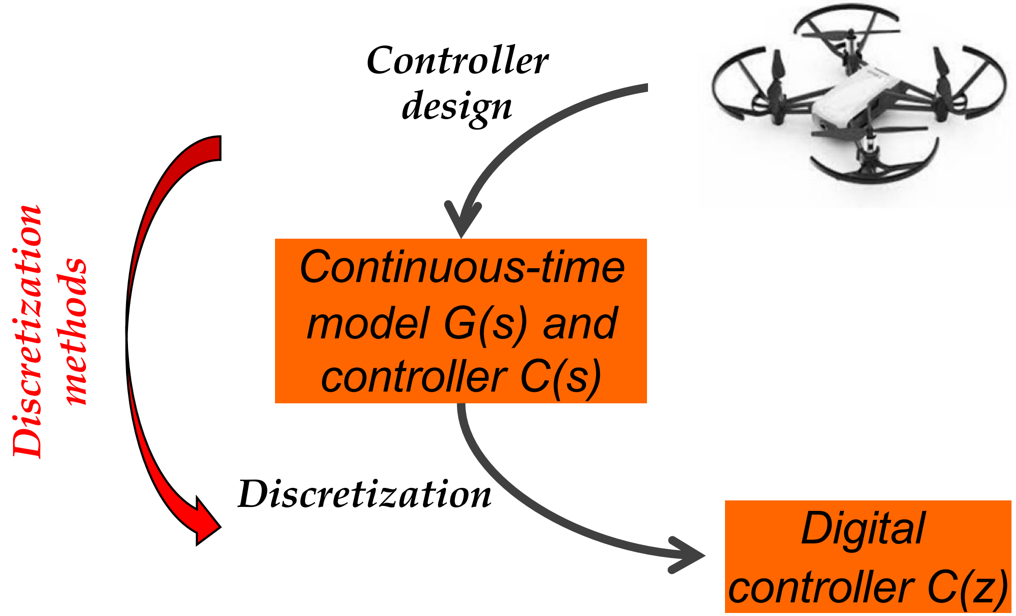
  3. Determine the controller parameters so that

$$C(z) = \frac{F_{ref}(z)}{G(z)\left(1 - F_{ref}(z)\right)}$$

Remarks

- A too restrictive choice of $F_{ref}(z)$ can lead to an unfeasible controller: non-causal or unstable
- Too fast dynamics of desired $F_{ref}(z)$ can lead to control values with too large amplitudes, damaging the equipment

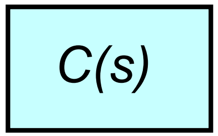
Transposition methods



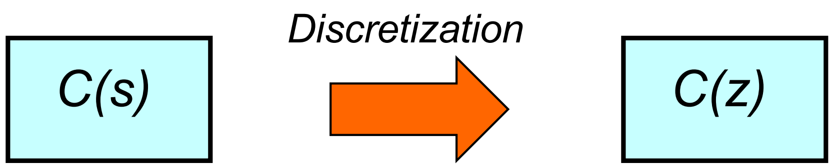
- The design of digital controllers by transposition of continuous-time controllers is preferred when:
  - *Fast sampling is possible*
- Continuous-time control design methods are generally well mastered in the industrial field: PID controllers, for example.
  - Specifications are easier to interpret with continuous-time models than with sampled models

Digital control design by transposing the continuous-time controller

- Methodology
  1. Design of a continuous-time controller *C(s)* by one of the traditional design methods (PID controller or others) determined from the system model *G(s)*, enabling the specifications to be met

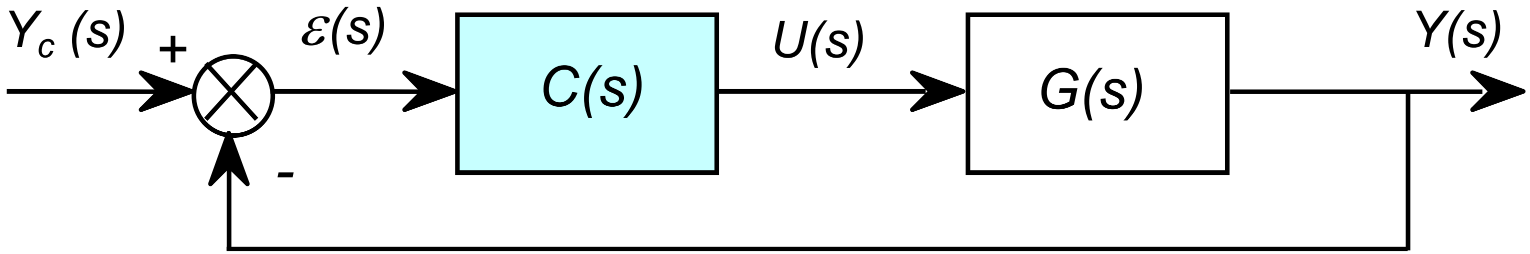


  2. Transpose the continuous-time transfer function *C(s)* into a digital controller *C(z)* to obtain a digital control algorithm that comes as close as possible to the behavior of continuous-time controller



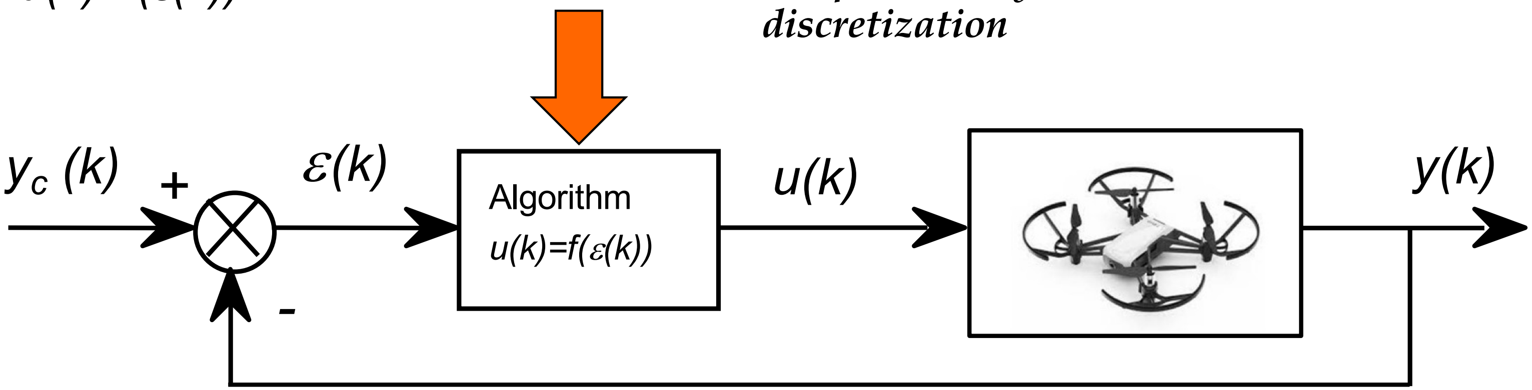
Digital control design by transposing the continuous-time controller

1. Determination of $C(s)$ from $G(s)$.
Various approaches (PID, ...) exist (course S5)



2. Transpose $C(s)$ to obtain $C(z)$
3. Implementation of the control algorithm
   $u(k)=f(\varepsilon(k))$

Transposition by discretization



Discretization methods

There are many to choose from, including:

- the impulse-invariant method
- the step-invariant method *(= ZOH method)*
- ***the matched pole-zero method***
- ***the forward Euler approximation method***
- ***the backward Euler approximation method***
- ***the Tustin (or bilinear) approximation method***

- ***Watch Brian Douglas' video***
  - *Discrete control #2: Discretize! Going from continuous to discrete domain*

- ***<u>Note</u>***
  - *The zero-order hold discretization method* (zoh) *to find C(z) from C(s) can be used, but it is not the most suitable here, as there is no ZOH in front of the controller!*

Some of the methods are available in Matlab

```
Command Window
>> help c2d
 c2d  Converts continuous-time dynamic system to discrete time.

    SYSD = c2d(SYSC,TS,METHOD) computes a discrete-time model SYSD with
    sample time TS that approximates the continuous-time model SYSC.
    The string METHOD selects the discretization method among the following:
  ➡  'zoh'           Zero-order hold on the inputs
       'foh'           Linear interpolation of inputs
  ➡  'impulse'       Impulse-invariant discretization
  ➡  'tustin'        Bilinear (Tustin) approximation.
  ➡  'matched'       Matched pole-zero method (for SISO systems only).
       'least-squares' Least-squares minimization of the error between
                       frequency responses of the continuous and discrete
                       systems (for SISO systems only).
       'damped'        Damped Tustin approximation based on TRBDF2 formula
                       (sparse models only).
    The default is 'zoh' when METHOD is omitted. The sample time TS should
    be specified in the time units of SYSC (see "TimeUnit" property).
```

The matched pole-zero method

We know *C(s)*. How can we deduce *C(z)* ????

We know the relationship between *z* and *s* : $z = e^{sT_s}$

1. Computation of the continuous-time zeros z_c and poles p_c of *C(s)*
2. Determination of the discrete-time zeros z_d and p_d poles of *C(z)*

$$z_d = e^{z_c T_s} \qquad p_d = e^{p_c T_s}$$

3. Add a zero at *z* = *-1* if relative order (deg. den. - deg num.) of C(s) >1
4. Adjust the steady-state gain if necessary, such as

$$C(z)\Big|_{z=1} = C(s)\Big|_{s=0}$$

```
In Matlab: Cd=c2d(C,Ts,'matched')
```

Differentiation approximations

A transfer function represents a differential equation. It is natural to obtain a difference equation by approximating the derivatives with a forward difference (Euler's method)

$$px(t) = \frac{dx(t)}{dt} \approx \frac{x(t+h) - x(t)}{h} = \frac{q-1}{h}\,x(t)$$

or a backward difference

$$px(t) = \frac{dx(t)}{dt} \approx \frac{x(t) - x(t-h)}{h} = \frac{q-1}{qh}\,x(t)$$

In the transform variables, this corresponds to replacing s by $(z-1)/h$ or $(z-1)/zh$. Section 2.8 shows that the variables z and s are related in some respects as $z = \exp(sh)$. The difference approximations correspond to the series expansions

$$z = e^{sh} \approx 1 + sh \qquad \text{(Euler's method)} \tag{8.1}$$

$$z = e^{sh} \approx \frac{1}{1 - sh} \qquad \text{(Backward difference)} \tag{8.2}$$

Another approximation, which corresponds to the trapezoidal method for numerical integration, is

$$z = e^{sh} \approx \frac{1 + sh/2}{1 - sh/2} \qquad \text{(Trapezoidal method)} \tag{8.3}$$

Euler forward and backward approximations

We know $C(s)$. How can we deduce $C(z)$????

We know the relationship between z and s :

$$z = e^{sT_s}$$

$$s = \frac{1}{T_e} ln(z)$$

Non-linear relationship!

$$z = e^{sT_e} \approx 1 + T_e s + \ldots \quad \Rightarrow \quad s = \frac{z-1}{T_e} = \frac{1-z^{-1}}{T_e z^{-1}}$$

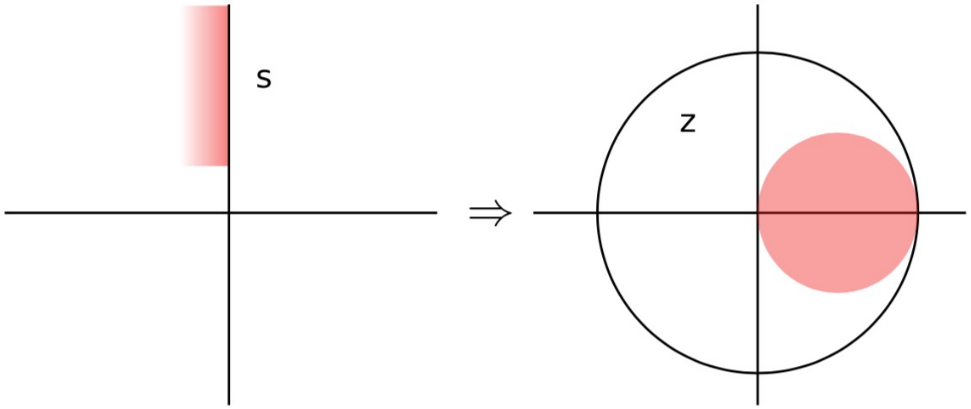
Forward approximation

$$z = \frac{1}{e^{-sT_e}} = \frac{1}{1 - T_e s + \ldots} \quad \Rightarrow \quad s = \frac{z-1}{T_e z} = \frac{1-z^{-1}}{T_e}$$

Backward approximation

Stability and distortion of the backward approximation

Conserve la stabilité : image de $\{s \in \mathbb{C} \ / \ Re(s) < 0\}$



Réponse fréquentielle : $z = e^{j\omega T_e}$

$$\frac{1 - e^{-j\omega T_e}}{T_e} = j\omega e^{-j\omega T_e/2} \frac{\sin(\omega T_e/2)}{\omega T_e/2}$$

Retard + distorsion

Tustin or bilinear approximation

We know $C(s)$. How can we deduce $C(z)$????

We know the relationship between z and s :

$$z = e^{sT_e}$$

$$s = \frac{1}{T_e} \ln(z)$$

Non-linear relationship!

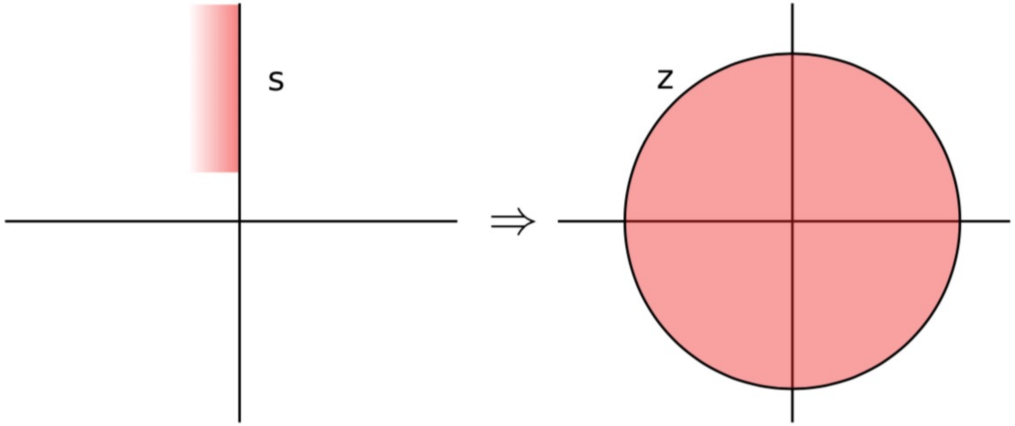
$$z = \frac{e^{\frac{sT_e}{2}}}{e^{\frac{-sT_e}{2}}} \approx \frac{1 + \frac{T_e}{2}s + \ldots}{1 - \frac{T_e}{2}s + \ldots}$$

$$s = \frac{2}{T_e}\frac{z-1}{z+1} = \frac{2}{T_e}\frac{1-z^{-1}}{1+z^{-1}}$$

Tustin approximation or bilinear

Stability and distortion of the Tustin approximation

Conserve la stabilité : image de $\{s \in \mathbb{C} \ / \ Re(s) < 0\}$



Réponse fréquentielle : $z = e^{j\omega T_e}$

$$\frac{2}{T_e}\frac{1-e^{-j\omega T_e/2}}{1+e^{j\omega T_e/2}} = j\omega \frac{\tan(\omega T_e/2)}{\omega T_e/2}$$

Pas de retard mais distorsion

Design of a digital conytroller by transposition of a continuous-time controller - Example

Let T_s =0.3s and the continuous-time controller be: $C(s)=\dfrac{1+0,53s}{1+0,21s}$

Forward approximation $$C(z)=\frac{0,53z-0,23}{0,21z+0,09}$$

Delayed approximation $$C(z)=\frac{0,83z-0,53}{0,51z+0,21}$$

Tustin approximation $$C(z)=\frac{1,89z-1,06}{z+0,17}$$

In Matlab: Cd=c2d(C,Ts,'tustin')