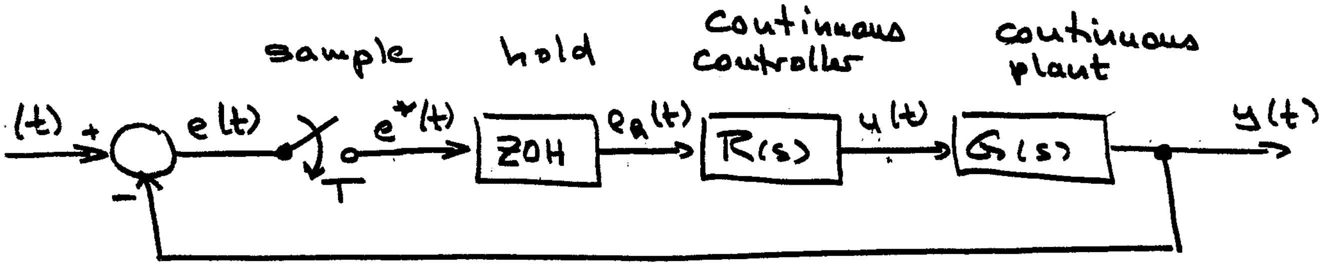
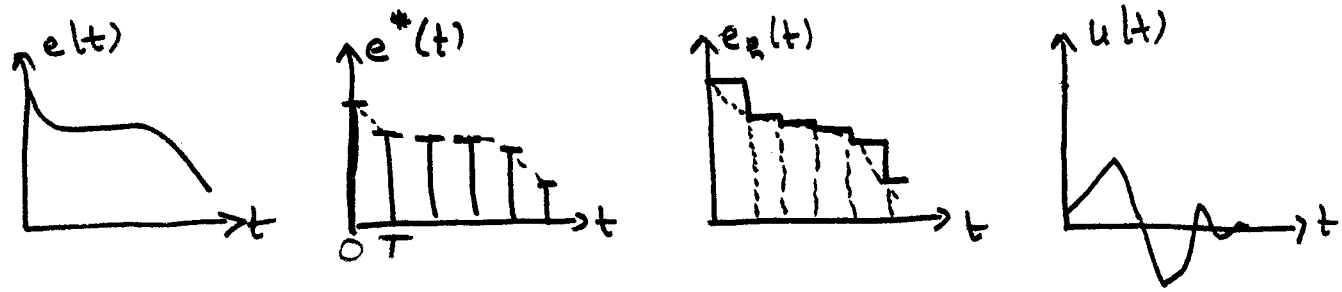
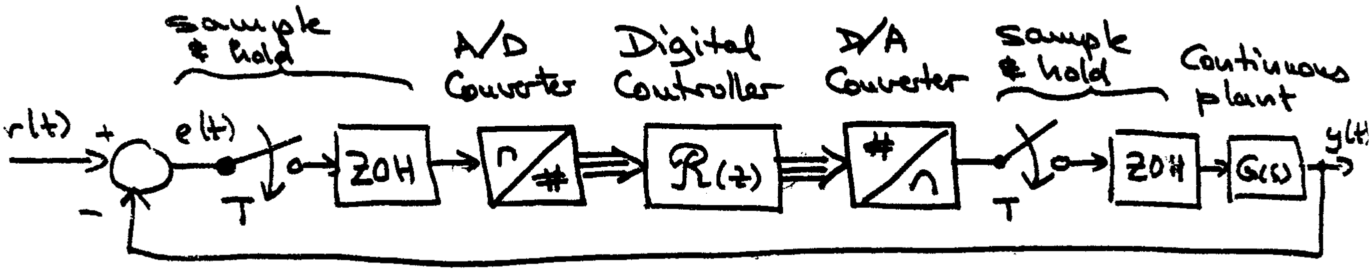
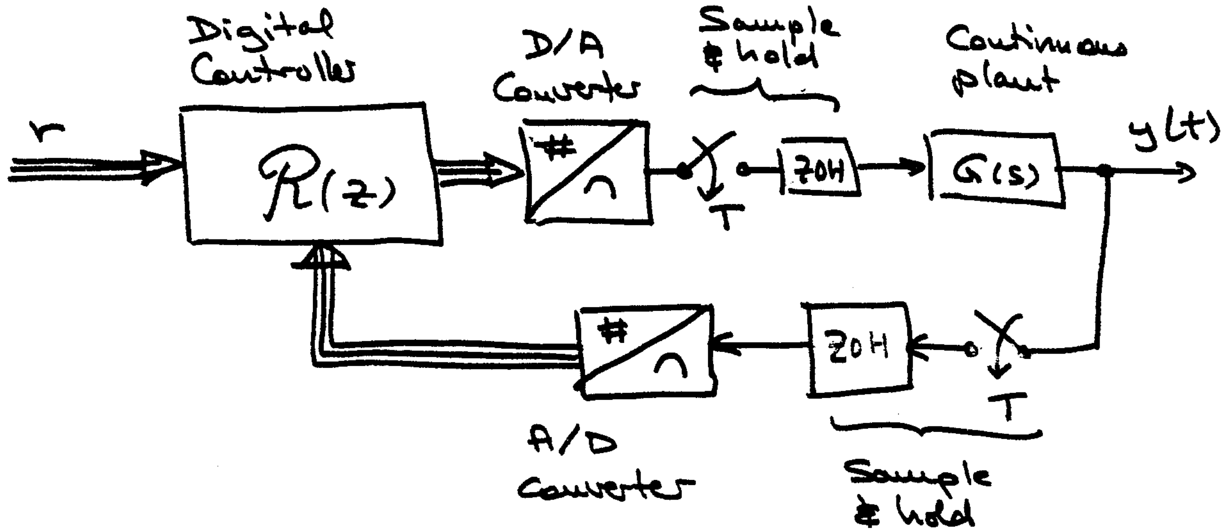
Types of Sampled-Data Systems









Reasons for Sampling:

Signal Processing

- sampled signals are easier to transmit
- transmitted signals can be regenerated without transmission error
- sampled signals can more easily be coded (cryptology)
- sampled signals can better be modulated (multiplexing)

Control

- multiple use of expensive equipment (multiplexing)
  - digital computer
  - data transfer channel
- data are available at particular time instants only
  - chemical analysis
  - radar signal
- data can be modified at particular times only
  - silicon controlled rectifiers
- data are naturally discrete
  - stepping motor
  - optical position sensor

- sampler is introduced to improve the dynamic behavior of the control loop
  - ☐ Better sensitivity behavior
  - ☐ Better reliability
  - ☐ No drift
  - ☐ Noise reduction
  - ☐ Less weight (e.g. space)
  - ☐ Less hardware cost
  - ☐ Less software/maintenance cost

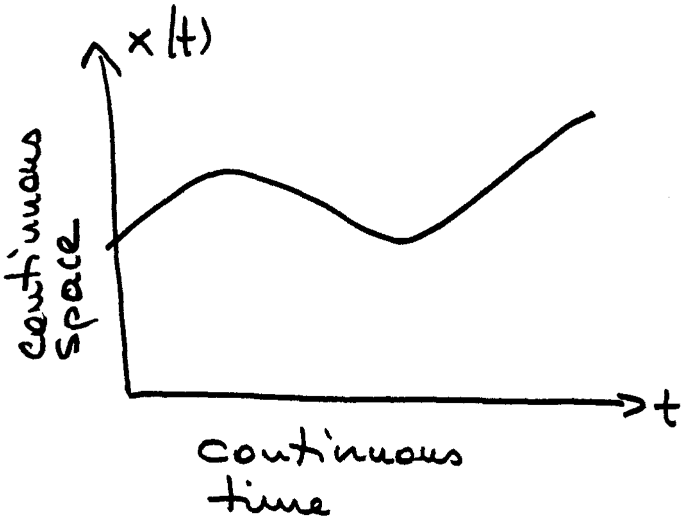
- Signal can be stored over longer time span

Types of Sampling

PAM: pulse amplitude modulation
PBM: pulse width modulation
PFM: pulse frequency modulation
...

n control, we use only **PAM**, because this is the only form of sampling that preserves linearity of a circuit.

Types of signals/systems:



$$\left|\begin{array}{l} \dot{\underline{x}} = A\underline{x} + B\underline{u} \\ \underline{y} = C\underline{x} + D\underline{u} \end{array}\right|$$

or:

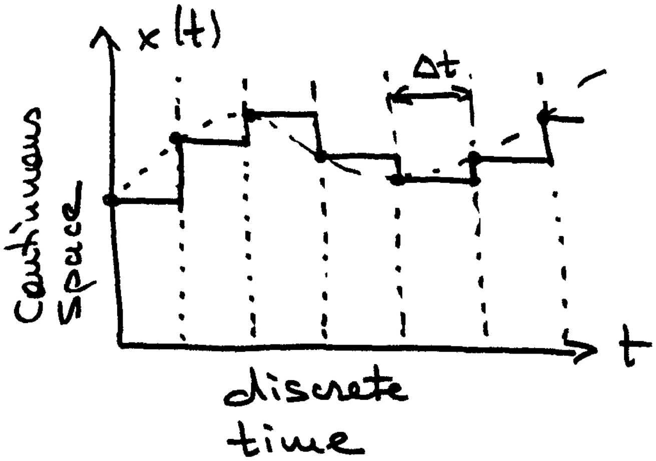
$$\underline{y}(s) = \underline{\underline{G}}(s)\cdot\underline{u}(s)$$

⇒ Continuous-Time Model

⇒ Differential equations

SISO: ECE 441

MIMO: ECE 501



$$\left|\begin{array}{l} \underline{x}(k+1) = F\underline{x}(k) + G\underline{u}(k) \\ \underline{y}(k) = H\underline{x}(k) + I\underline{u}(k) \end{array}\right|$$

or:

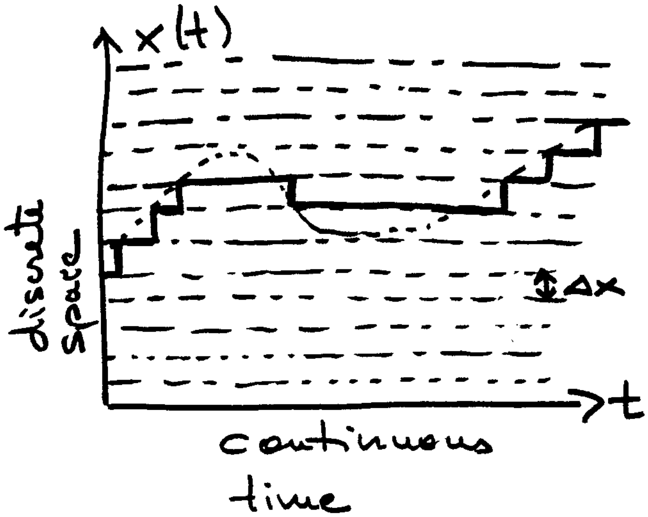
$$\underline{y}(z) = \underline{\underline{G}}(z)\cdot\underline{u}(z)$$

⇒ Discrete-Time Model

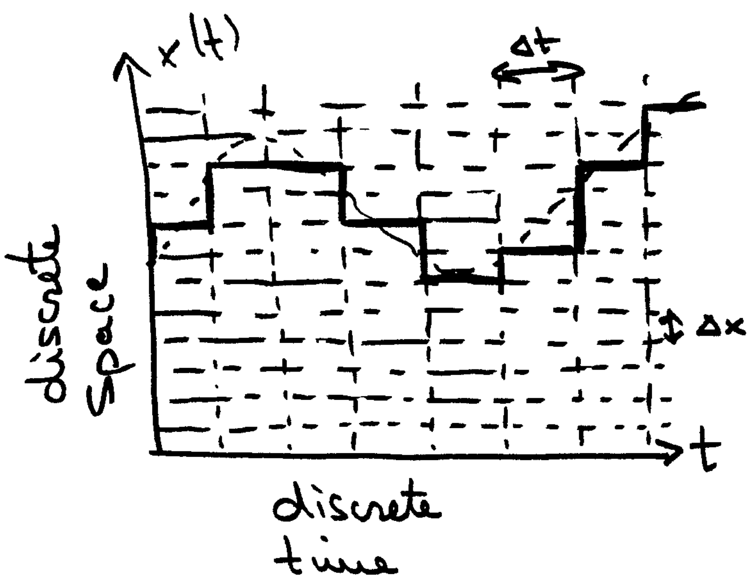
⇒ Difference equation

SISO: ECE 442

We currently have no class for MIMO discrete-time systems.



Application: A Shoemaker can produce a new pair of shoes at any time, but he always produces an entire pair at one time.



Application: Digital control system. The computer produces new control signals every Δt time units, but due to finite word length, only a finite # of levels is possible

e.g. 8 bit converter

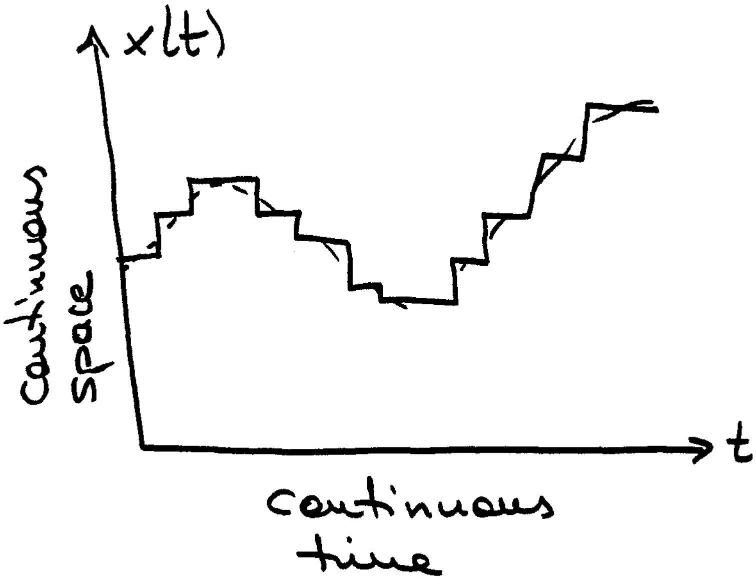
$\Rightarrow$ $2^n = 256$ different levels are possible

Problem: While discretization in time is linear, discretization in space is not !!! $\Rightarrow$ We have a problem with the mathematical description.

Solution: We interpret the discretization in space as noise:

$$\left| \begin{array}{l} \underline{x}(k+1) = F\,\underline{x}(k) + G\,\underline{u}(k) + \underline{n}(k) \\ \underline{y}(k) = H\,\underline{x}(k) + I\,\underline{u}(k) + \underline{w}(k) \end{array} \right|$$

$\underline{n}, \underline{w}$ are noise vectors.



State changes can occur at <u>any</u> time, and take <u>any</u> value. <u>But</u>: in a finite time interval, the states can change only a finite number of times.

A state-change is called a <u>discrete event</u>.

This type of situation is important for <u>sequencing control</u>, e.g. robots, assembly line manufacturing, <u>etc</u>. Unfortunately, good <u>control</u> strategies are not known. This is <u>open</u>

research. The only technique available is simulation and optimization of parameters through nonlinear programming.

⇒ e.g. { ECE 575
MIS 521 a,b

Better ideas for control purposes would be great and easy to sell — but the problem is very difficult. The operations are highly nonlinear.

We know much better how to control linear systems than nonlinear. Therefore, we try to avoid nonlinear operations as much as possible, and often even compensate inherent system nonlinearities to make the system (from outside) look like a linear system.