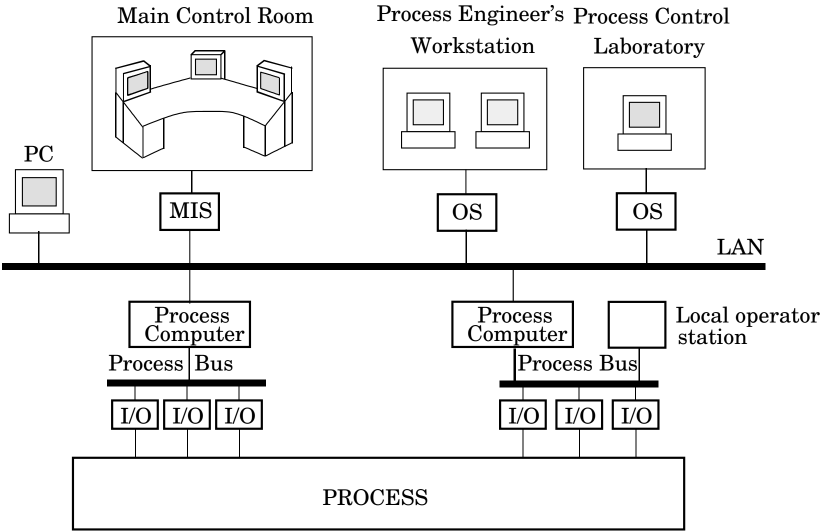
Process control – FX

- Computer control
  - Industrial control systems
  - Sampled systems
  - Controller discretization
- Logic and sequence control
  - Boolean algebra
  - GRAFCET

Reading: *Systems Engineering and Process Control:* X.1–X.8

Industrial control systems



Control in several levels

Low level:

- Logic
- Simple control loops, often PI(D)

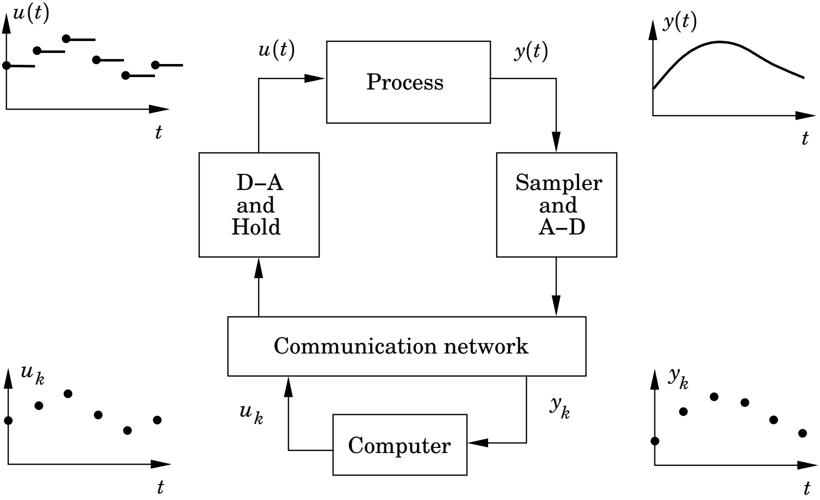
Mid level:

- Sequence control
- Coordination using different control structures
- Advanced MIMO control, e.g., Model Predictive Control

High level:

- Production planning
- Process optimization

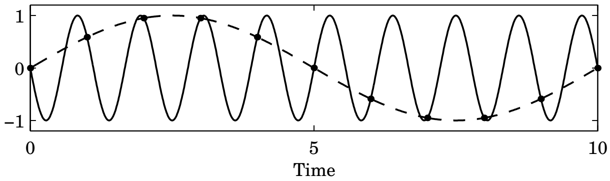
Sampled control systems



Sampled control systems

- Mix of continuous and discrete time – hard to analyze
  - Simplification: Only look at sampling time points
- Potential problems
  - Lost information through sampling
  - Quantization effects in D-A och A-D converters
  - Effects on communication delays

Lost information through sampling

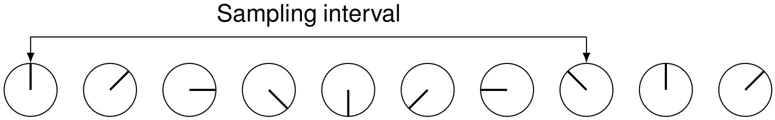


- Sampling: Discrete points with given time interval *sampling interval*, h are measured.

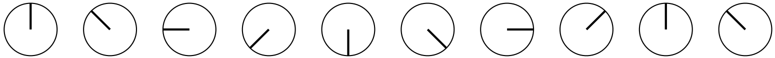
  Sampling frequency: $\omega_s = 2\pi/h$
- Aliasing: Higher frequencies are seen as lower frequencies
- *The sampling theorem*: At least two samples per period needed to avoid aliasing

Aliasing example 1

Rotating disc:

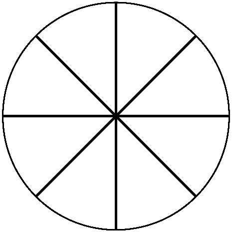


Sampled sequence:

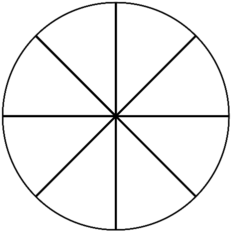


Aliasing example 1

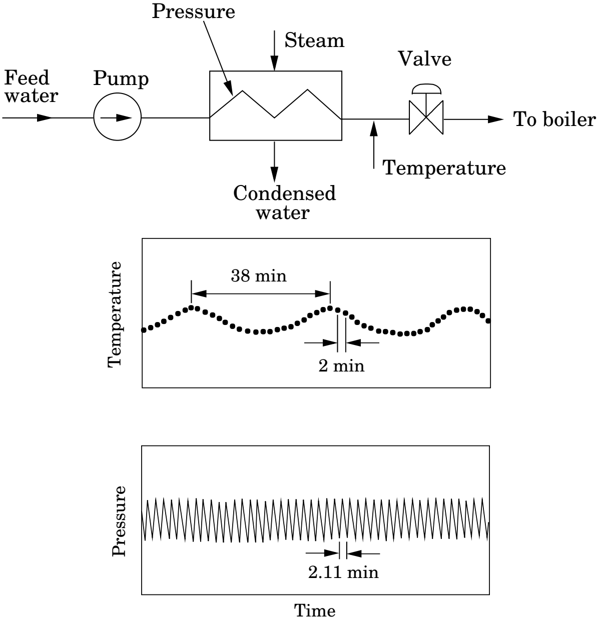
Rotating disc



Sampled disc

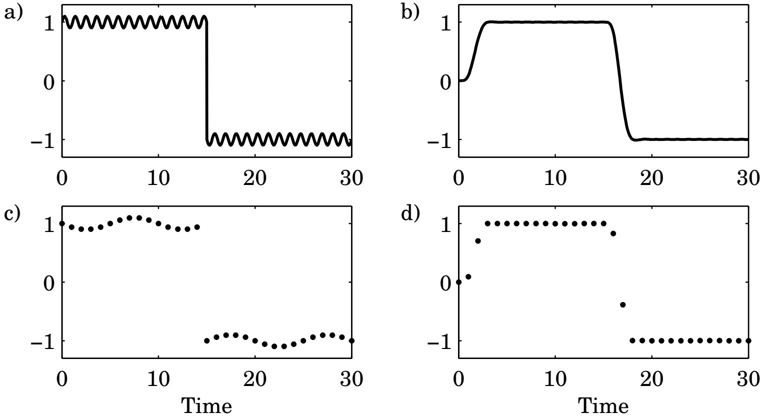


Aliasing example 2



Avoiding aliasing

All signal components above the *Nyquist frequency* $\omega_N = \omega_s/2 = \pi/h$ should be filtered away before sampling



a) not filtered signal, b) filtered signal, c) sampled not filtered signal, d) sampled filtered signal

Mathematical system descriptions

- Continuous time systems:
  - Differential equations, e.g.,: $T\frac{dy}{dt} + y = Ku$
  - Laplace transform, e.g.,: $Y(s) = \frac{K}{1+Ts}U(s) = G(s)U(s)$

- Sampled (discrete time) systems:
  - **Difference equations**

$$y(kh + h) + ay(kh) = bu(kh)$$

  - **Shift operator**: $qy(kh) = y(kh + h)$

$$y(kh) = \frac{b}{q + a}u(kh) = H(q)u(kh)$$

Process description sampling

- Linear continuous time process on state-space form:

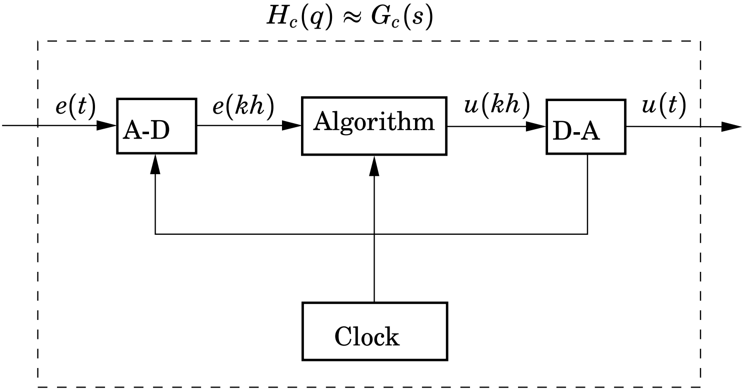
$$
\begin{aligned}
\dot{x}(t) &= Ax(t) + Bu(t) \\
y(t) &= Cx(t) + Du(t)
\end{aligned}
$$

- How does x change between sample points t and $t+h$? (supposing input u is constant):

$$
\begin{aligned}
x(t+h) &= e^{Ah}x(t) + \int_t^{t+h} e^{A(t-\tau)}Bu(\tau)\,d\tau \\
&= e^{Ah}x(t) + \int_0^h e^{A\tau}B\,d\tau\,u(t) \\
&= \Phi x(t) + \Gamma u(t)
\end{aligned}
$$

- Linear difference equation

Discrete approximation of continuous controller



- Shorter sampling interval h enables for better approximation
- How to translate $G_c(s) \rightarrow H_c(q)$?

Discretization methods

Approximate derivatives with differences:

- Forward difference

$$\frac{dy(t)}{dt} \approx \frac{y(t+h)-y(t)}{h}$$

- Backward difference

$$\frac{dy(t)}{dt} \approx \frac{y(t)-y(t-h)}{h}$$

Note! Many other (better) discretization methods exist

Example

Discretize the continuous system $\frac{dy(t)}{dt} = -3y(t) + 2u(t)$

- Forward difference:

$$\frac{dy(t)}{dt} \approx \frac{y(kh+h) - y(kh)}{h} = -3y(kh) + 2u(kh)$$
$$y(kh+h) = (1-3h)y(kh) + 2hu(kh)$$

- Backward difference:

$$\frac{dy(t)}{dt} \approx \frac{y(kh) - y(kh-h)}{h} = -3y(kh) + 2u(kh)$$
$$y(kh) = \frac{1}{1+3h}y(kh-h) + \frac{2h}{1+3h}u(kh)$$

Stability analysis

Stability for a scalar difference equation:

$$y(kh + h) = ay(kh) + bu(kh)$$

Suppose $u(kh) = 0$. $y(kh) = a^k y(0)$

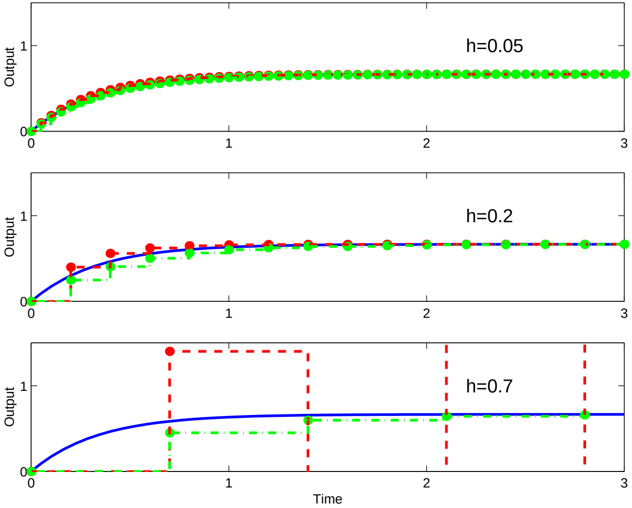
- $y(\infty) = 0$ if $|a| < 1$
- $|y(\infty)| = \infty$ if $|a| > 1$
- (Generally: poles inside unit circle $\Rightarrow$ asymptotic stability)

Stability conditions for above example:

- Forward difference: $|1 - 3h| < 1 \;\Rightarrow\; 0 < h < 2/3$
- Backward difference: $|\frac{1}{1+3h}| < 1 \;\Rightarrow\; h > 0$

Simulation of example with $G(s) = \frac{2}{s+3}$

Exact solution (–) Forward difference (- -) Backward difference (-.)



Discretization of PI controller

- PI controller with practical modifications (L9):

$$u(t) = \underbrace{K\Big(\beta r(t) - y(t)\Big)}_{P(t)} + \underbrace{\int_0^t \left(\frac{K}{T_i}e(\tau) + \frac{1}{T_t}\big(u(\tau) - v(\tau)\big)\right)d\tau}_{I(t)}$$

- P part static, no approximation needed:

$$P(kh) = K\left(\beta r(kh) - y(kh)\right)$$

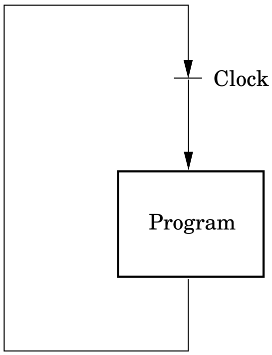
Discretization of PI controller

- I part discretized with forward difference:

$$
\begin{aligned}
I(t) &= \int_0^t \left(\frac{K}{T_i} e(\tau) + \frac{1}{T_t}\big(u(\tau) - v(\tau)\big) \right) d\tau \\
\frac{dI(t)}{dt} &= \frac{K}{T_i} e(t) + \frac{1}{T_t}\big(u(t) - v(t)\big) \\
\frac{I(kh+h) - I(kh)}{h} &= \frac{K}{T_i} e(t) + \frac{1}{T_t}\big(u(t) - v(t)\big) \\
I(kh+h) &= I(kh) + \frac{Kh}{T_i}\, e(kh) + \frac{h}{T_t}\big(u(kh) - v(kh)\big)
\end{aligned}
$$

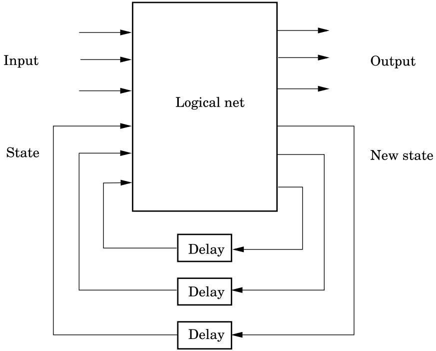
Implementation of PI controller – pseudo code

```
LOOP
   WaitForClockTick();
   r = ADIn(1);
   y = ADIn(2);
   P = K*(beta*r-y);
   v = P + I;
   IF v < umin
      u = umin;
   ELSEIF v > umax
      u = umax;
   ELSE
      u = v;
   END
   DAOut(1,u);
   I = I + K*h/Ti*(r-y) + h/Tt*(u-v);
END
```



Logic and discrete control

- Discrete signals
  - Measurements: *true* or *false*
  - Inputs: **on** or **off**
- Logical nets
  - Static nets
  - E.g.: Alarms
- Sequence nets
  - Dynamic nets
  - E.g.: Start-up, shutdown batch process

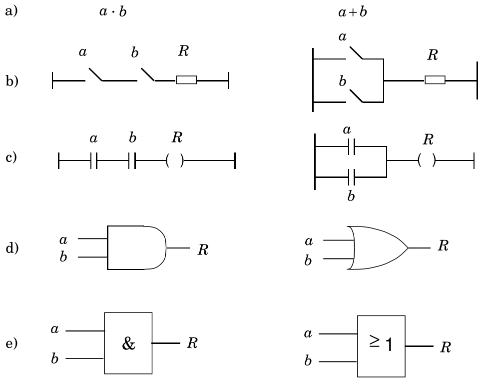


Operations and symbols

Three operations:

and:	$a \cdot b$	a `and` b	$a \wedge b$
or:	$a + b$	a `or` b	$a \vee b$
not:	$\bar{a}$	`not` a	$\neg a$

Symbols for `and` and `or`:



Computing with logic

Boolean algebra:

- Ex: $1 + a = 1$ och $0 + a = a$
- Ex: $1 \cdot a = a$ och $0 \cdot a = 0$
- Ex: $a + \bar{a} = 1$ och $a \cdot \bar{a} = 0$

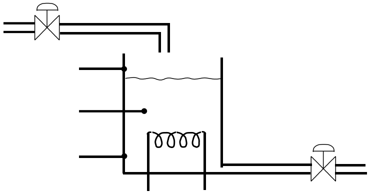
Logic laws:

- Commutative
  $a \cdot b = b \cdot a$, $a + b = b + a$
- Associative
  $a \cdot (b \cdot c) = (a \cdot b) \cdot c$, $a + (b + c) = (a + b) + c$
- Distributive
  $a \cdot (b + c) = a \cdot b + a \cdot c$
- de Morgan's law
  $\overline{a + b} = \bar{a} \cdot \bar{b}$, $\overline{a \cdot b} = \bar{a} + \bar{b}$

Example

Alarm for a batch reactor, sound alarm if:

- temperature T in tank too high and cooling valve Q off
- temperature T is high and inflow valves is open V_1



Truth table:

T	Q	V_1	y =alarm
0	0	0	0
1	0	0	1
0	1	0	0
1	1	0	0
0	0	1	0
1	0	1	1
0	1	1	0
1	1	1	1

Sequence control

Tasks should be done in sequence. Example:

- Elevator
- Washing machine
- Cake baking
- Start-up and shutdown of reactor

Requires memory (state), order is important

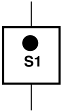
Sequence net

- Finite state machine (automata theory)
- Petri net
- GRAFCET (a kind of Petri net)

GRAFCET – Steps and transitions

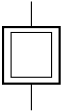
Steps:

- Active and inactive



S1.x = 1 when active

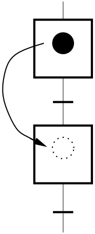
S1.T = number of time units since the step last became active



Initial step

Transitions:

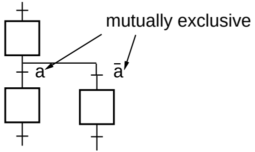
fired when preceding step is active and transition condition satisfied



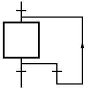
condition true and/or event occurred + previous step active

GRAFCET – Control structures

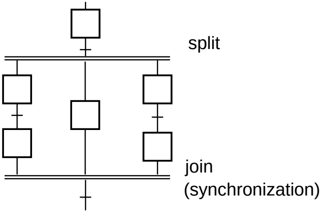
- Alternative ways:
  1. Branches (mutually exclusive)



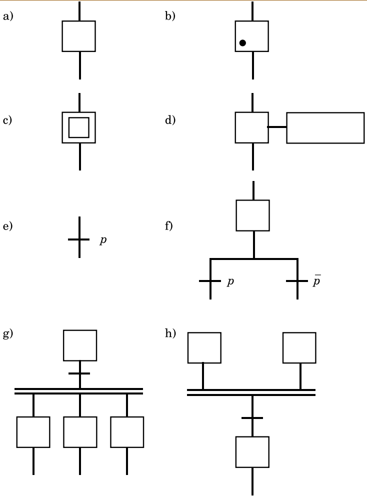
  2. Repetition



- Parallel ways with synchronized exit

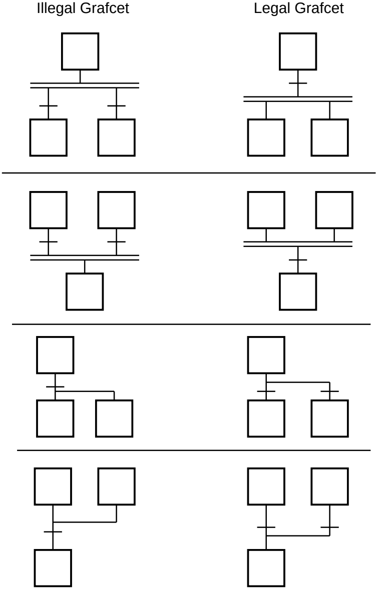


GRAFCET – Fundamental symbols



GRAFCET symbols. (a) Step (inactive); (b) Step (active); (c) Initial step; (d) Step with action; (e) Transition; (f) Branching with mutually exclusive alternatives; (g) Branching into parallel paths; (h) Synchronization.

GRAFCET – Some Examples



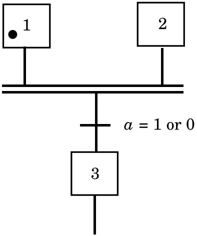
GRAFCET – Execution principles

Grafcet evolution rules:

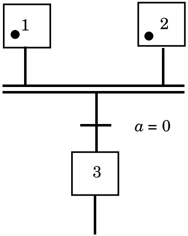
- The initial step(s) is active when the function chart is initiated.
- A transition is firable if:
  - all steps preceding the the transition are active (enabled).
  - the receptivity (transition condition and/or event) of the transition is true

  A firable transition must be fired.
- all the steps preceding the transition are deactivated and all the steps following the transition are activated when a transition is fired
- all firable transitions are fired simultaneously
- when a step must be both deactivated and activated it remains activated without interrupt

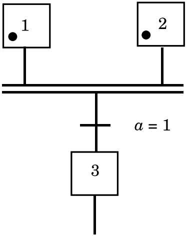
GRAFCET – Examples of firing



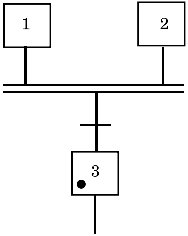
a) Not enabled



b) Enabled but not firable

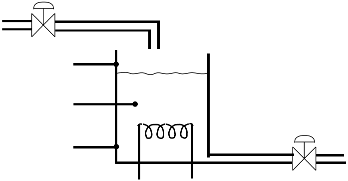


c) Firable



d) After the change from c)

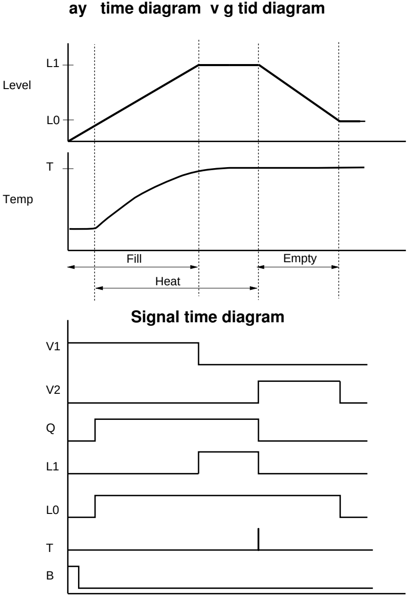
Example: Specifications



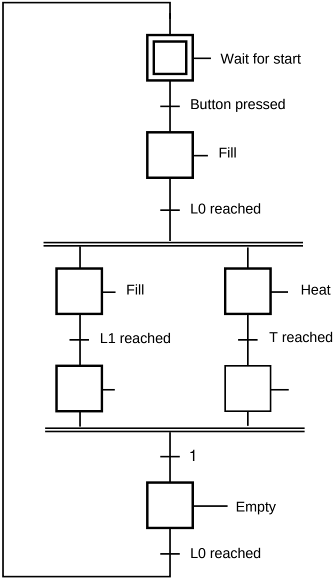
Verbal description:

1. Start the sequence by pressing the button B. (not shown)
2. Fill water by opening the valve V_1 until the upper level L_1 is reached.
3. Heat the water until the temperature is greater than T. The heating can start as soon as the water is above the level L_0.
4. Empty the water by opening the valve V_2 until the lower level L_0 is reached.
5. Close the valves and go to Step 1 and wait for a new sequence to start.

Example: Time functions



Example: Sequence net



Project

- Create sequential control program for a CSTR-process in JGrafchart (Java-based implementation of GRAFCET)
- Digital implementation of PI controller for water level and temperature