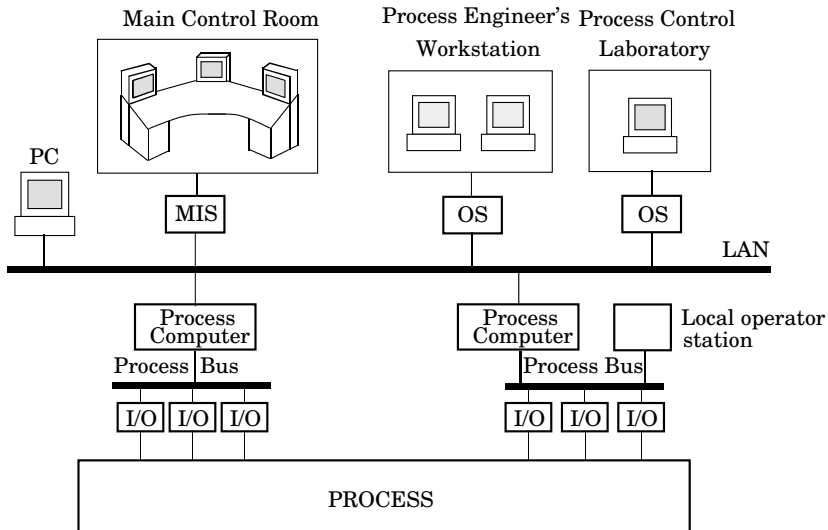


Processreglering – FX

- ▶ Datorstyrning
 - ▶ Industriella styrsystem
 - ▶ Samplade system
 - ▶ Diskretisering av regulatorer
- ▶ Logik och sekvensstyrning
 - ▶ Boolesk algebra
 - ▶ GRAFCET

Läsanvisning: *Process Control*: 10.1–10.8

Industriella styrsystem



Styrning på flera olika nivåer

Låg nivå:

- ▶ Logik, förreglingar
- ▶ Enkla reglerloopar, oftast PI(D)

Mellannivå:

- ▶ Övergripande sekvensstyrning
- ▶ Samordning m.h.a. olika regulatorstrukturer
- ▶ Avancerad MIMO-reglering, t.ex. Model Predictive Control

Hög nivå:

- ▶ Produktionsplanering
- ▶ Processoptimering

Hård- och mjukvara på olika nivåer

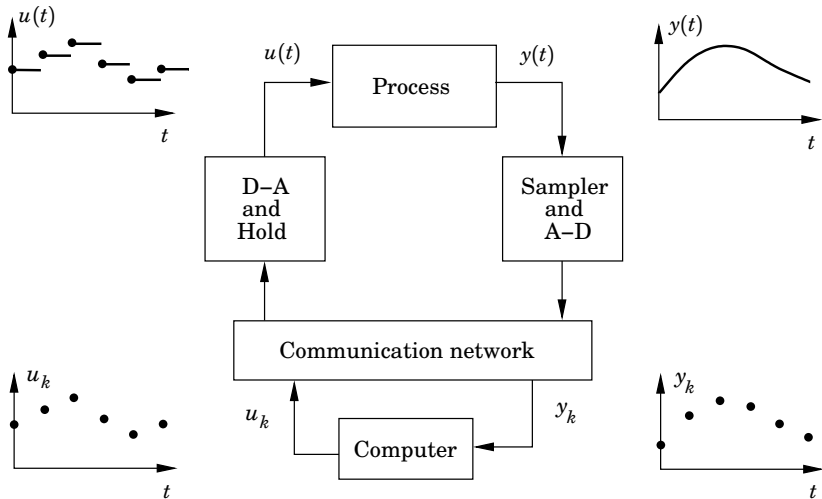
Låg nivå:

- ▶ PLC (styrdatorer)
- ▶ Fältbussar (lokala styrenätverk)
- ▶ Sensorer och ställdon (ofta med inbyggda styrprogram)
- ▶ Servodrifter och frekvensomvandlare (för motorstyrning)
- ▶ Enkla operatörsgränssnitt (knappar och pekskärmar)

Högre nivåer:

- ▶ PC-datorer
- ▶ Ethernet
- ▶ SCADA-system (Supervisory Control and Data Acquisition)
- ▶ MES (Manufacturing Execution System)
- ▶ ...

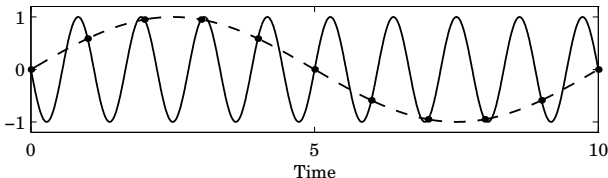
Samplade reglersystem



Samplade reglersystem

- ▶ Blandning av kontinuerlig och diskret tid – svårt att analysera
 - ▶ Förenkling: titta bara i samplingstidpunkterna
- ▶ Potentiella problem
 - ▶ Förlorad information genom sampling
 - ▶ Inverkan av kvantisering i D-A- och A-D-omvandlarna
 - ▶ Inverkan av kommunikationsfördröjningar

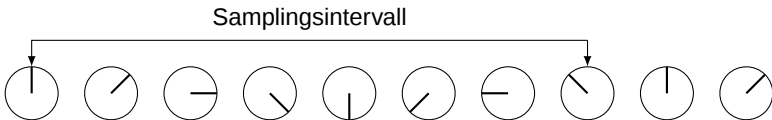
Förlorad information genom sampling



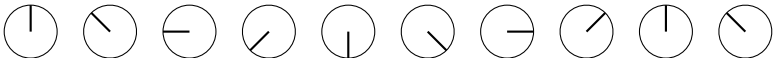
- ▶ Sampling innebär att man mäter i diskreta punkter med ett givet tidsintervall, *samlingsintervallet*, h .
Samplingsfrekvens: $\omega_s = 2\pi/h$
- ▶ Vikning (*aliasing*) kallas det då högre frekvenser tolkas som lägre frekvenser
- ▶ *Samplingsteoremet* säger att minst två sampel per period krävs för att undvika vikning

Exempel på vikning 1

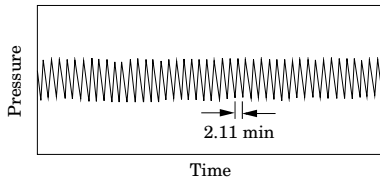
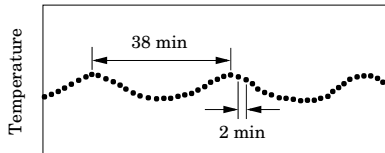
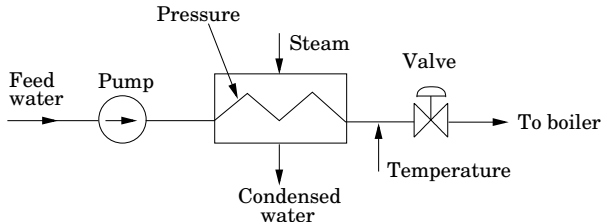
Snurrande skiva:



Samplad sekvens:



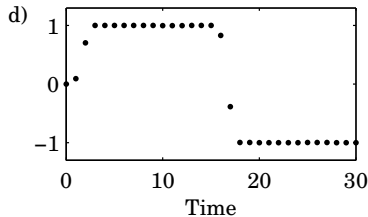
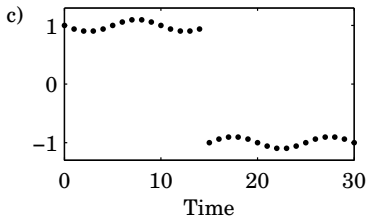
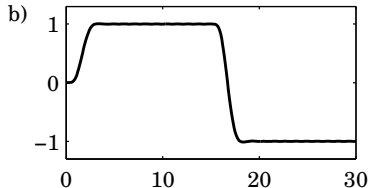
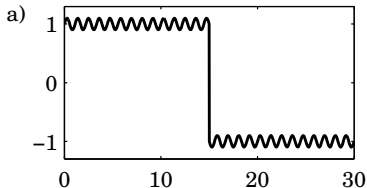
Exempel på vikning 2



Att undvika vikning

Alla signalkomponenter ovanför *Nyquistfrekvensen*

$\omega_N = \omega_s/2 = \pi/h$ bör filtreras bort före sampling



a) ofiltrerad signal, b) filtrerad signal, c) samplad ofiltrerad signal,
d) samplad filtrerad signal

Matematiska beskrivningar av system

► Tidskontinuerliga system:

- Differentialekvationer, t.ex. $T \frac{dy}{dt} + y = Ku$
- Laplacetransform, t.ex. $Y(s) = \frac{K}{1+Ts} U(s) = G(s)U(s)$

► Samplade (tidsdiskreta) system:

► **Differensekvationer**

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

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

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

Sampling av processbeskrivningen

Linjär tidskontinuerlig process på tillståndsform:

$$\dot{x}(t) = Ax(t) + Bu(t)$$

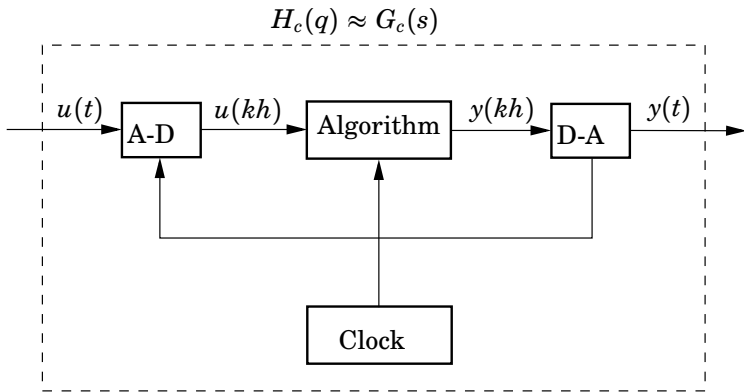
$$y(t) = Cx(t) + Du(t)$$

Hur utvecklas tillståndet x mellan två sampelpunkter t och $t + h$?
(antag att styrsignalen u hålls konstant):

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

Linjär differensekvation

Diskret approximering av kontinuerlig regulator



Ju kortare samplingsintervall h desto bättre approximation är möjlig

Hur översätta $G_c(s) \rightarrow H_c(q)$?

Diskretiseringsmetoder

Approximera derivatorna med differenser

- ▶ Framåtdifferens (Euler)

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

- ▶ Bakåtdifferens

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

OBS! Det finns många andra diskretiseringsmetoder (jämför med numerisk integration)

Exempel

Diskretisera det kontinuerliga systemet $\frac{dy(t)}{dt} = -3y(t) + 2u(t)$

- Framåtdifferens:

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

- Bakåtdifferens:

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

Analys av stabilitet

Stabilitet för en skalär differensekvation:

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

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

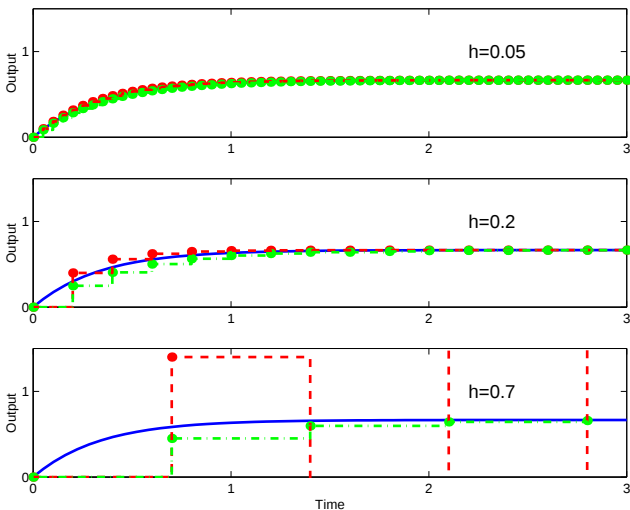
- ▶ $y(\infty) = 0$ om $|a| < 1$
- ▶ $|y(\infty)| = \infty$ om $|a| > 1$
- (Generellt: poler innanför enhetscirkeln medför asymptotisk stabilitet)

Stabilitetsvillkor för exemplet ovan:

- ▶ Framåtdifferens: $|1 - 3h| < 1 \Rightarrow 0 < h < 2/3$
- ▶ Bakåtdifferens: $|\frac{1}{1+3h}| < 1 \Rightarrow h > 0$

Simulering av exemplet

$G(s) = \frac{2}{s+3}$. Exakt lösning (-) Framåtdifferens (Euler) (- -) Bakåtdifferens (-.)



Diskretisering av PI-regulatorn

PI-regulator med praktiska modifieringar (F9):

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

- P-delen statisk, ingen approximation behövs:

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

Diskretisering av PI-regulatorn

- I-delen diskretiserad med framåtdifferens:

$$I(t) = \int_0^t \left(\frac{K}{T_i} e(\tau) + \frac{1}{T_t} (u(\tau) - v(\tau)) \right) d\tau$$

$$\frac{dI(t)}{dt} = \frac{K}{T_i} e(t) + \frac{1}{T_t} (u(t) - v(t))$$

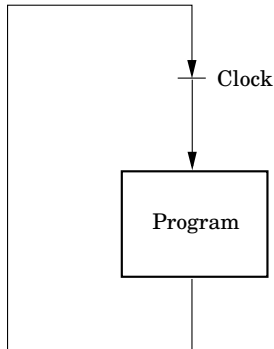
$$\frac{I(kh + h) - I(kh)}{h} = \frac{K}{T_i} e(t) + \frac{1}{T_t} (u(t) - v(t))$$

$$I(kh + h) = I(kh) + \frac{Kh}{T_i} e(kh) + \frac{h}{T_t} (u(kh) - v(kh))$$

Implementering av PI-regulatorn – pseudokod

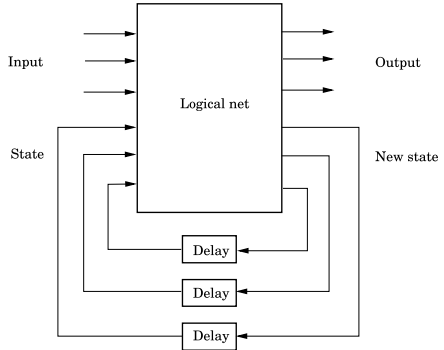
LOOP

```
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
```



Logik och diskret styrning

- ▶ Diskreta signaler
 - ▶ Mät signaler: *sant* eller *falskt*
 - ▶ Styr signaler: **på** eller **av**
- ▶ Logiknät
 - ▶ Statiskt nät
 - ▶ Ex: Alarm, förreglingar
- ▶ Sekvensnät
 - ▶ Dynamiskt nät
 - ▶ Ex: Uppstart/stopp, batchprocesser



Operationer och symboler

Tre operationer:

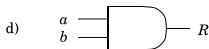
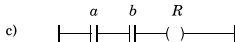
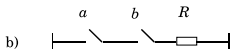
and: $a \cdot b$ a and b $a \wedge b$

or: $a + b$ a or b $a \vee b$

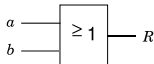
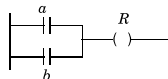
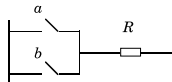
not: $\bar{a}$ not a $\neg a$

Symboler för and och or:

a) $a \cdot b$



$a + b$



Logiklagar

Boolesk 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 $a \cdot \bar{a} = 0$

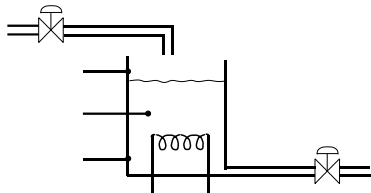
Logiska lagar:

- ▶ Kommutativ
 $a \cdot b = b \cdot a, a + b = b + a$
- ▶ Associativ
 $a \cdot (b \cdot c) = (a \cdot b) \cdot c, a + (b + c) = (a + b) + c$
- ▶ Distributiv
 $a \cdot (b + c) = a \cdot b + a \cdot c$
- ▶ de Morgans lag
 $\overline{a + b} = \bar{a} \cdot \bar{b}, \overline{a \cdot b} = \bar{a} + \bar{b}$

Exempel

Alarm för en batchreaktor:

Ge alarm om temperaturen i tanken är för hög, T , samt kylventilen är stängd, Q , eller då temperaturen är hög och inflödesventilen är öppen, V_1 .



Sanningstabell:

T	Q	V_1	$y = \text{alarm}$
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

Sekvensstyrning

Uppgifter ska göras efter varandra. Exempel:

- ▶ Hiss
- ▶ Tvättmaskin
- ▶ Kakbakning
- ▶ Uppstart och stopp av reaktorer

Kräver minne (tillstånd), ordningsföljd är väsentlig

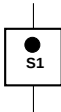
Sekvensnät

- ▶ Finite state machine (automatateori)
- ▶ Petrinät
- ▶ GRAFCET (är ett slags Petrinät)

GRAFCET – Steg och övergångar

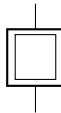
Steg:

- Aktiv respektive inaktiv



$S1.x = 1$ when active

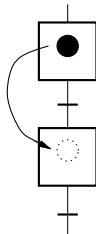
$S1.T$ = number of time units since the step last became active



Initial step

Övergång:

avfyras då föregående steg är aktivt samt övergångens villkor är eller blir uppfyllt.

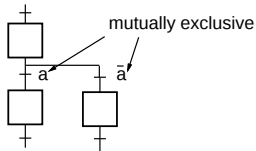


condition true and/or event occurred +
previous step active

GRAFCET – Kontrollstrukturer

► Alternativa vägar:

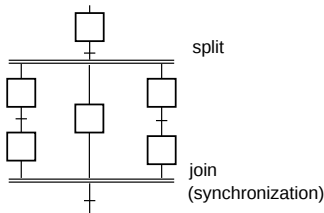
1. Förgrening (ömsesidigt uteslutande)



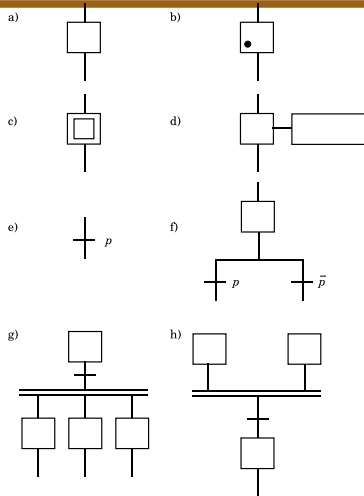
2. Repetition



► Parallella vägar med synkroniserad avslutning



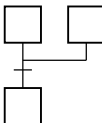
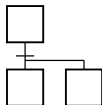
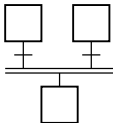
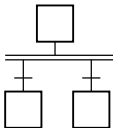
GRAFCET – Grundläggande symboler



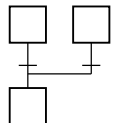
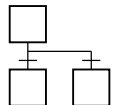
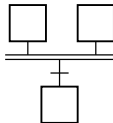
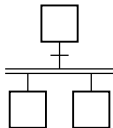
(a) Inaktivt steg, (b) Aktivt steg, (c) Initialsteg, (d) Steg med åtgärd, (e) Övergång,
(f) Förgrening med ömsesidigt uteslutande alternativ, (g) Förgrening i parallella vägar,
(h) Synkronisering

GRAFCET – Några exempel

Illegal Grafcet



Legal Grafcet



GRAFCET – Exekveringsprinciper

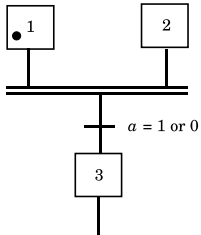
GRAFCETs exekveringsregler:

- ▶ Initialsteget är aktivt när sekvensnätet är initierat.
- ▶ En övergång är avfyringsbar om:
 - alla steg före övergången är aktiva
 - övergångsvillkoret är sant

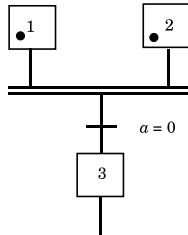
En avfyringsbar övergång måste avfyras

- ▶ alla stegen före övergången inaktiveras, och alla följande steg aktiveras, när en övergång avfyras.
- ▶ alla avfyringsbara övergångar avfyras samtidigt
- ▶ när ett steg både måste inaktiveras och aktiveras så kvarstår steget som aktivt utan avbrott

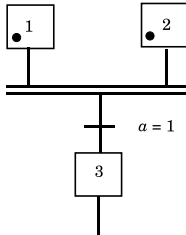
GRAFCET – Exempel på avfyring



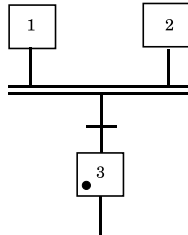
a) Not enabled



b) Enabled but not firable

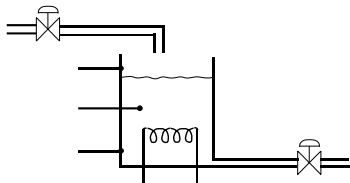


c) Firable



d) After the change from c)

Exempel: specifikationer

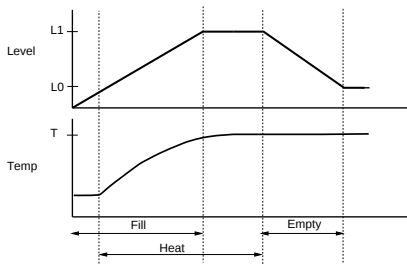


Verbal beskrivning:

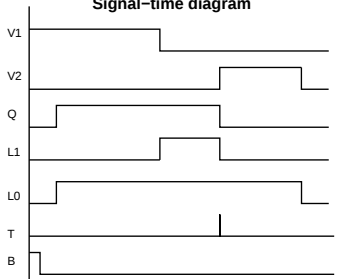
1. Starta sekvensen med att trycka på knappen B
2. Öppna ventil V_1 för påfyllning av vatten till övre nivå L_1
3. Värm vattnet tills temperaturen är högre än T . Värmning kan starta så fort det finns vatten över nivå L_0
4. Töm tanken genom att öppna ventil V_2 tills nivån når L_0 .
5. Stäng ventilerna och gå till 1) för att vänta på en ny startorder.

Exempel: tidsfunktioner

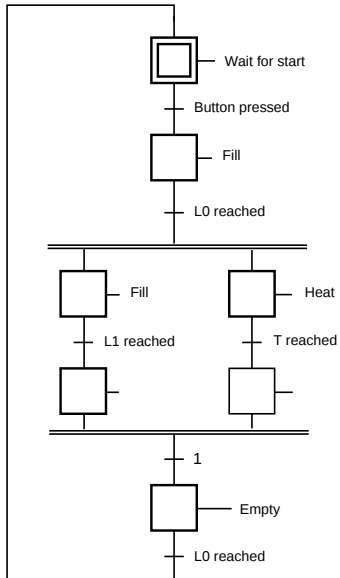
Way – time diagram (våg-tid-diagram)



Signal-time diagram



Exempel: Sekvensnät



Miniprojekt

- ▶ Skapa sekventiellt styrprogram för en CSTR-process i JGrafchart (Java-baserad implementation av GRAFCET)
- ▶ Digital implementering av PI-regulatorer för vattennivå och temperatur

