Reglering av händelsedrivna system (5 sp)

Control of Discrete Event Systems (5 cr)

The course can be a part of

1. Fördjupad kurs i reglerteknik (Chemical Engineers)

2. Fördjupad kurs i industriell systemteknik (Computer Engineers)

3. Fritt valbara studier (all)

Lectures?,

Assignments: Three
Exam: 24h or 4h + home exam
Homepage: www.abo.fi/student/tkfrtcdes
Systems and signals

\[ \dot{x} = g(x, u, t) \]

**Figure 1: Model of a system**

Where \( u = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \end{bmatrix} \) (inputs), \( x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \end{bmatrix} \) (states), \( y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \end{bmatrix} \) (outputs).

Linear system:

\[ \dot{x} = Ax + Bu \]
\[ y = Cx + Du \]
Feedback control

- Modeling, P is a model of the real system G
- Specification, Sp is derived based on P and specs for closed loop
- Synthesis, derive a controller S based on P and Sp
- Verification, check if $S \circ P \leq Sp$
- Implementation, derive a physical controller K which implements S
Discrete event systems

• Discrete event systems (DES) have states that take integer \( \{0, 1, 2, \ldots \} \) or symbolic \( \{idle, running, broken, in\text{-}service, \ldots \} \) values.

• State transitions are driven by events

• State transitions are instantaneous, take zero time

• Such system cannot be properly modeled by differential or difference equations

Observations:

• The state-space is a discrete set, which will be denoted \( Q \)

• Events can also be collected in a set, which will be denoted \( \Sigma \)

• Control of DES typically means supervision, thus the term supervisor

• Inputs to supervisor typically trigger events, and outputs are typically functions of the state
Example 1. *Random walk.* We consider a particle in a two-dimensional space, and we only model the number of steps taken in different directions, and assume equidistant steps. Thus we can model the position using integer-valued coordinates \((x, y)\) of the particle. The coordinates are the state of the system:

\[
Q = \{(i, j) : i, j = \ldots, -1, 0, 1, \ldots\}
\]

A natural set of events is

\[
\Sigma = \{N, S, W, E\}
\]

corresponding to the events "one step north", "one step south", "one step west", and "one step east". The figure below shows what happens if we start from an initial state \((0,0)\) and the events \(\{E, S, W, W, N, N, W\}\) take place.
Example 2.  *Stick picking game*

- You start with a certain number of sticks
- Two players pick sticks
- 1 or 2 (or 3) sticks must be picked at each turn
- The player picking the last stick loses (in newest version of Supremica it is the other way)

**Task:** Design a supervisor that plays to win
More examples of discrete event systems

- Queueing systems: arrival - queue - server - departure

- Communication systems: idle - transmitting - collision/waiting

- Manufacturing processes: idle - loaded - processing - finished

- Batch processes: recipes, sharing resources

- Traffic systems: intersections, traffic lights

- Cars: Central locking, vipers

- Database systems: read and write

- Hydraulic systems: Digital valves

- Monitoring of control systems: start up, shut down, change of operation, exception handling, failure diagnosis-isolation-recovery

DES are always man made
Control of DES

- Control of DES = Supervisory Control
- Restricting the system from doing bad things
- Allowing good things to happen
- Often implemented in Programmable Logic Controllers (PLCs)
  - Specialized computers
  - Specialized programming
  - IL, ST, LD, SFC, FBD, ...
  - Typically it involves a lot of fiddling with bits
Problems

- Combinatorial explosion
  - 7 robots can do $n$ things, $n$ states in each robot
  - $n = 2$, $2^7 = 128$ states in the combined system
  - $3^7 = 2187$ states in the combined system
  - $4^7 = 16384$ states in the combined system

- Exhaustive testing practically impossible

- Problems can occur far later
Valve battery example

• Valve battery
  – Collection of valves
  – Flexible routing
  – Operators
  – Automatic

• Manual verification of PLC program
  – 2 persons
  – in 3 shifts
  – for 8 weeks

• A full person-year!

• 700 logical errors were found!!
The course

1. Modeling
2. Specification
3. Synthesis
4. Verification
5. (Automatic code generation)
Tools

• Automata (= state machines) - Regular languages

• Supervisory Control Theory – Ramadge & Wonham, 1987–

• Supremica, a software tool from Chalmers, downloadable at www.supremica.org, unfortunately not available in the PC-class in Axelia.