Embedded Information Systems

Nonconventional modelling

December 8, 2020

Nonconventional modelling

Reminder

Hybrid systems: Hybrid systems combine continuous and discrete dynamics.

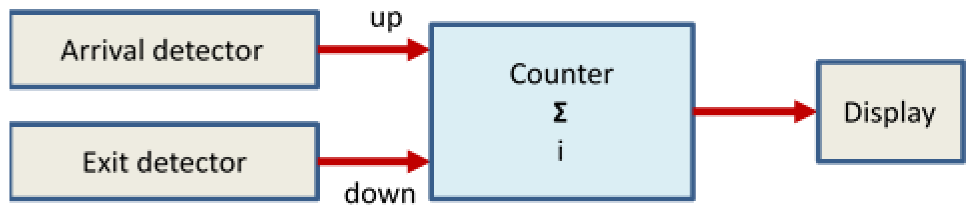
Sometimes they are called modal systems, because controlled by a Finite State Machine (FSM), they are switched into different modes of operation where they behave as continuous systems.

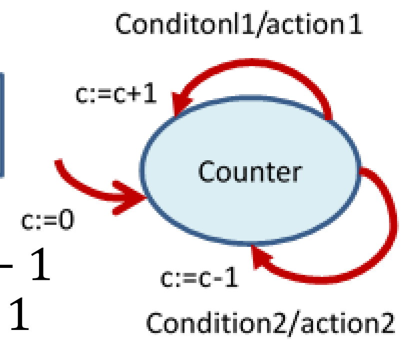
Concerning the mode changes, hybrid systems behave like discrete systems, but between these mode changes time dependency is present.

Discrete systems:

Example:

Number of cars in a parking house (max. *M*, which is the capacity of the house)

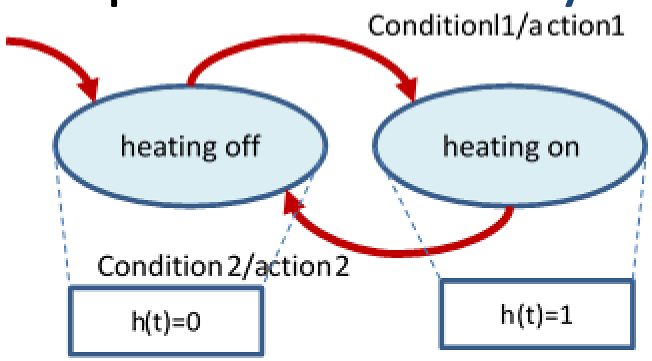




Condition1/action1: $up \wedge \neg down \wedge c < M / c + 1$

Condition2/action2: $down \wedge \neg up \wedge c > 0 / c - 1$

Example: Thermostat with hysteresis



Condition1/action1: Temperature ≤ 18 C°/heating on

Condition2/action2: Temperature ≥ 22 C°/heating off

System input: Temperature of the environment

System output: heating on/heating off commands:

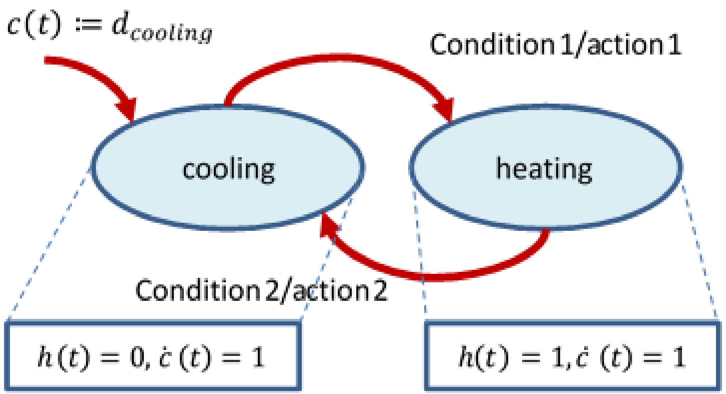
The corresponding **time functions:** $h(t) = 1, h(t) = 0.$

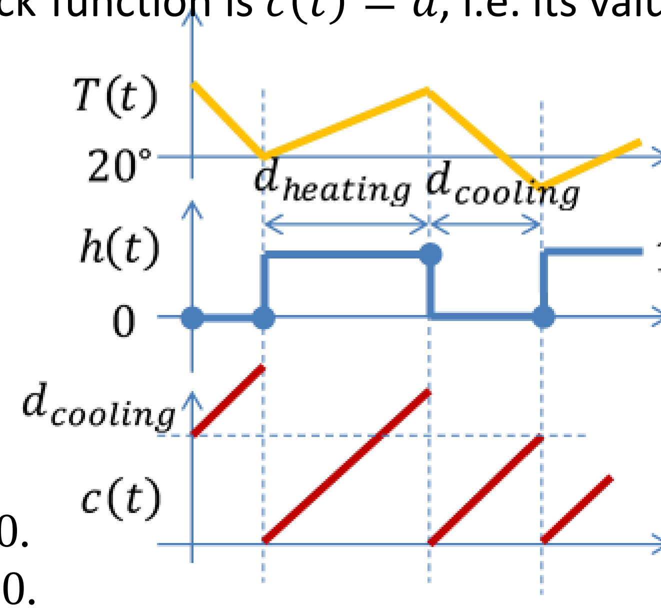
Timed automaton:

Example: ***Thermostat with timing instead of hysteresis:*** this is solved by the so-called **timed automaton**, which is the simplest nontrivial hybrid system.

Reminder

These automata, behind their states measure the evolvement of time for a given value of duration: $\forall t \in d_m$, and the derivative of the clock function is $\dot{c}(t) = a$, i.e. its value changes with the evolvement of time.





Condition1/action1: $T(t) \leq 20 \wedge c(t) \geq d_{cooling} / c(t) = 0.$

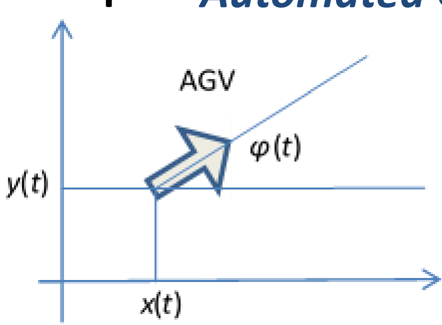
Condition2/action2: $T(t) \geq 20 \wedge c(t) \geq d_{heating} / c(t) = 0.$

Comments:

(1) $h(t)$ and $c(t)$ can be considered as tools of state refinement.They define some details of the operation (Modal systems).

(2) On the time diagram $T > 20\ C^{\circ}$. If it would be lower, then immediately the heating mode would start. This is served by the initial condition of the clock.

Example: ***Automated Guided Vehicle, AGV***



$$\dot{x}(t) = v(t)\cos(\varphi(t))$$
$$\dot{y}(t) = v(t)\sin(\varphi(t))$$
$$\dot{\varphi}(t) = \omega(t)$$

Two-level control: The AGV runs with a constant speed of 10 *km/h*. It has four operational mode: **left, right, straight, stop.**

To every mode of operation, a separate differential equation is assigned.

original:	**straight:**	**left:**	**right:**	
$\dot{x}(t) = v(t)\cos(\varphi(t))$ $\dot{y}(t) = v(t)\sin(\varphi(t))$ $\dot{\varphi}(t) = \omega(t)$	$\dot{x}(t) = 10\cos(\varphi(t))$ $\dot{y}(t) = 10\sin(\varphi(t))$ $\dot{\varphi}(t) = 0$	$\dot{x}(t) = 10\cos(\varphi(t))$ $\dot{y}(t) = 10\sin(\varphi(t))$ $\dot{\varphi}(t) = \pi$	$\dot{x}(t) = 10\cos(\varphi(t))$ $\dot{y}(t) = 10\sin(\varphi(t))$ $\dot{\varphi}(t) = -\pi$	$\dot{x}(t) = 0$ $\dot{y}(t) = 0$ $\dot{\varphi}(t) = 0$

stop:

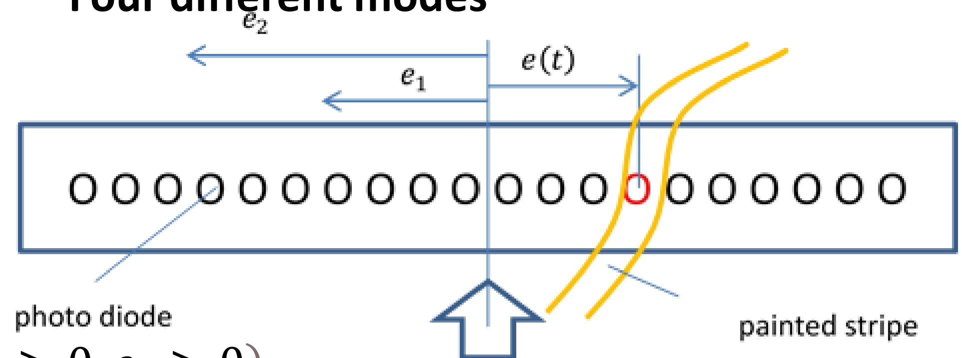
Four different modes

The sensor of the AGV:

a set of photodiodes perpendicular to the direction of the movement.

Its output signal: $e(t) = f(x(t), y(t))$.

If $e(t) > 0$, then it deviates to the left, if $e(t) < 0$, then to the right.



$(e_1 > 0, e_2 > 0)$

The control law of the AGV:

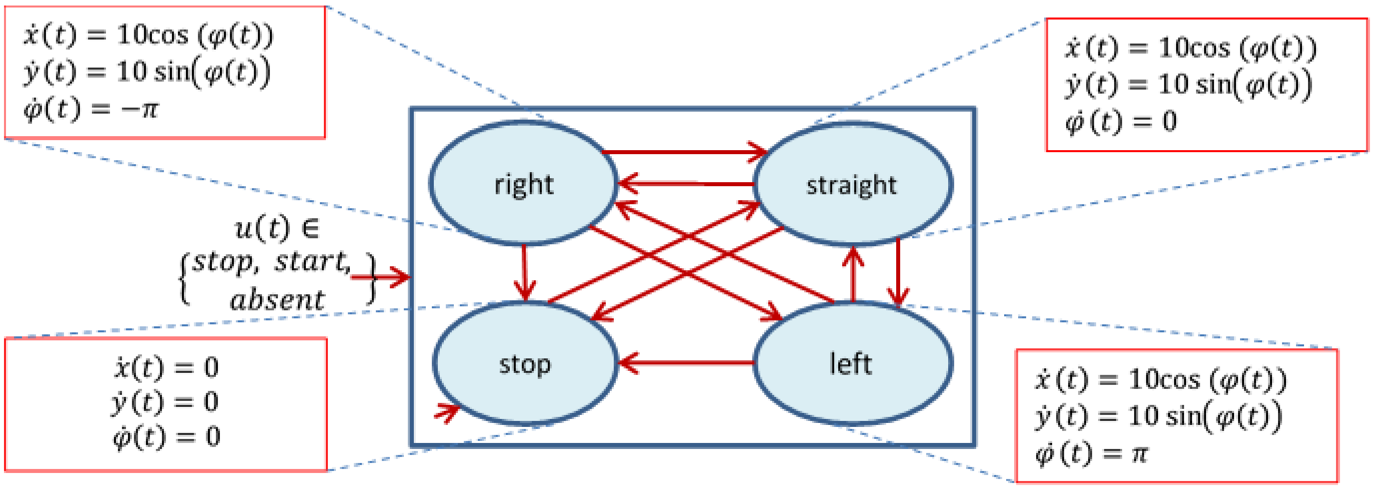
if $|e(t)| < e_1$, then go straight;
if $0 < e_2 < e(t)$, then go to right;
if $0 > -e_2 > e(t)$, then go to left.

The set of the input events: $u(t) \in \{stop, start, absent\}$. Since *stop* and *start* are instantaneous events, *absent* gives the interpretation for other time instants.

State-transition generating conditions:

start = $\{(v(t), x(t), y(t), \varphi(t)) | u(t) = start\}$
go straight = $\{(v(t), x(t), y(t), \varphi(t)) | u(t) \neq stop, |e(t)| < e_1\}$
go right = $\{(v(t), x(t), y(t), \varphi(t)) | u(t) \neq stop, e_2 < e(t)\}$
go left = $\{(v(t), x(t), y(t), \varphi(t)) | u(t) \neq stop, -e_2 > e(t)\}$
stop = $\{(v(t), x(t), y(t), \varphi(t)) | u(t) = stop\}$

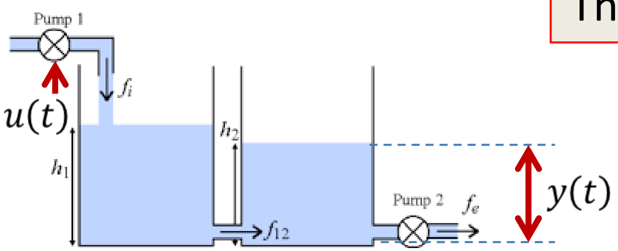
Reminder



Qualitative modelling and control I.

Example: **The design of such a controller which keeps the level of the liquid in the second tank at a prescribed level.**

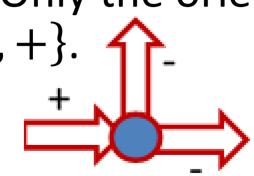
This is possible by setting *u(t)* at pump1 properly.



Problems of the quantitative model:
a) The physical limits are not modelled;
b) The equations are linearized;
c) Numerical values are inaccurate and change with time.

Qualitative Reasoning: Only the orientation of the quantities is considered.

Possible "values": $\{-, 0, +\}$.
The basic physical constraints are kept!



If at branching of a node the liquid flows out in two directions, then through the third tube the liquid should flow in.

The qualitative value of a quantity "*Q*" with respect to "*a*": $[Q]_a$

The qualitative value of the change of a quantity "*Q*" is the qualitative derivative:

$[\delta Q]_a, [\delta^2 Q]_a, \ldots$

Reminder

Operations: $(invert\ A)$: Changes the sign.

$(vote\ A_1, A_2, \dots, A_n)$: Gives back the value in majority.

Qualitative control of the level of tank2: L_2 denotes the level relative to the desired value:

$[L_2] = +$: higher than required.	$[\delta U] = +$: increase pumping rate.
$[L_2] = 0$: equals.	$[\delta U] = 0$: pumping rate is appropriate.
$[L_2] = -$: lower than required.	$[\delta U] = -$: decrease pumping rate.

$[\delta U] = +$: a fixed amount of increase of the pumping rate: ΔU.

The qualitative values exist only at the sampling instants.

Between sampling instants there is no level detection.

$$[L_2]_{(k)} = \left[actual\ level_{(k)} - required\ level_{(k)}\right]$$

A very simple control law:

$$Q1 \overset{def}{\triangleq} [\delta U]_{(k)} = (invert[L_2])_{(k)}$$

Comment:

If ΔU is larger, then larger overshoot and oscillation can be expected, but the reaction is faster.

If ΔU is smaller, then the overshoot and the oscillation will be smaller, but also the reaction is slower.

Improved controllers: Quantities considered:

Level error of tank2: $+, 0, -$

Speed of the level change of tank2: $+, 0, -$

Speed of the level change of tank1: $+, 0, -$

$$3 * 3 * 3 = 27$$

Reminder

$$Q2 \overset{def}{\triangleq} [\delta U]_{(k)} = \Big(invert\Big(vote\big(vote([L_2]_{(k)}, [\delta L_2]_{(k)}), [\delta L_1]_{(k)}\big)\Big)\Big)_{(k)}$$

$$Q3 \overset{def}{\triangleq} [\delta U]_{(k)} = \Big(invert\Big(vote([L_2]_{(k)}, [\delta L_2]_{(k)}, [\delta L_1]_{(k)})\Big)\Big)_{(k)}$$

Determination of $[\delta L_1]$: $\delta L_1 = \left(L_{2(k)} - L_{2(k-1)}\right) - \left(L_{2(k-1)} - L_{2(k-2)}\right) = \delta^2 L_2$

For the 27 combinations of the possible qualitative values the outputs of the three controllers can be summarized in the table below:

	$[L_2]$	$[\delta L_2]$	$[\delta L_1]$	$Q1$	$Q2$	$Q3$
1	+	+	+	-	-	-
2	+	+	0	-	-	-
3	+	+	-	-	0	-
4	+	0	+	-	-	-
5	+	0	0	-	-	-
…						
20	-	+	0	+	0	0
…						
27	-	-	-	+	+	+

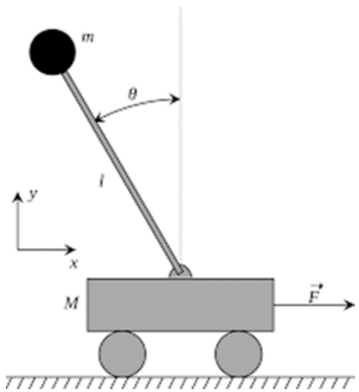
Comments:

(1) A rule-based system was also elaborated for this problem.
It could not handle the case: Tank2 shows a constant value above the required level, and the level of Tank1 lowers.

(2) Setting sampling rate and the value of ΔU is a critical issue, and a crucial decision of the designer.

Example: Modelling an inverted pendulum with nondeterministic automaton.

This approach might be required for systems where the about the state vector $x(k)$ only quantized $[x(k)]$ values are available due to limited precision measurements of angles and angular velocity.



After linearisation of the equations around $\theta\ =\ 0$:

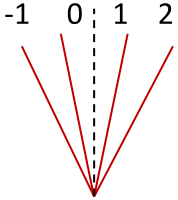
$$\dot{x}(t) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{mg}{M} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{(m+M)g}{Ml} & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ \frac{1}{m} \\ 0 \\ -\frac{1}{Ml} \end{bmatrix} u(t)$$

$$M = 1kg, m = 0.1kg, l = 0.5m, g = 9.81\frac{m}{s^2}$$

Measurement insensitivity: 0.0175 rad for ϑ, and 0.0175/20ms for $\dot{\theta}$.

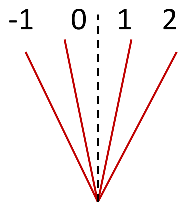
The pole can no longer be stabilised if $|x_3| > 0.21\ rad\ (12°)$, and $|x_4| > 0.87$.

For the angle (3rd element of the state vector) and for the angular speed (4th element of the state vector) the bounds corresponding to the figure:



$$g_{3,-1} = -0.210, g_{3,0} = -0.0175, g_{3,1} = 0.0175, g_{3,2} = 0.210$$
$$g_{4,-1} = -0.870, g_{4,0} = -0.0175, g_{4,1} = 0{,}0175, g_{4,2} = 0.870$$

If we denote staying in one of the two central regions by 0, by - 1 staying in the left-hand-side region, and by +1 staying in the right-hand-side one, we can define the following qualitative states:



$$z_1 = \begin{bmatrix}-1\\-1\end{bmatrix}, z_2 = \begin{bmatrix}-1\\0\end{bmatrix}, z_3 = \begin{bmatrix}-1\\1\end{bmatrix}, z_4 = \begin{bmatrix}0\\-1\end{bmatrix}, z_5 = \begin{bmatrix}0\\0\end{bmatrix},$$

$$z_6 = \begin{bmatrix}0\\1\end{bmatrix},\ z_7 = \begin{bmatrix}1\\-1\end{bmatrix},\ z_8 = \begin{bmatrix}1\\0\end{bmatrix},\quad z_9 = \begin{bmatrix}1\\1\end{bmatrix},\quad z_{10} = outside,$$

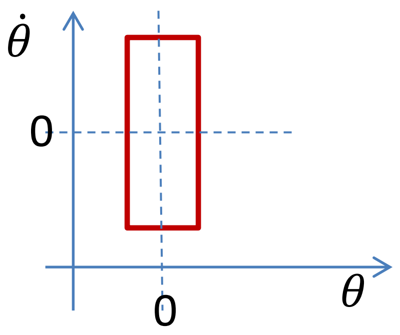
The qualitative values of the force on the vehicle (input signal):

$$u(k) = 10 \Leftrightarrow v(k) = 1,\quad u(k) = 0 \Leftrightarrow v(k) = 0,\ u(k) = -10 \Leftrightarrow v(k) = -1$$

By assigning proper input to the qualitative states, the pole can be stabilized:

$z(k)$	z_1	z_2	z_3	z_4	z_5	z_6	z_7	z_8	z_8
$v(k)$	-1	0	0	-1	0	1	0	0	1

The qualitative controller: $[u(k)] = f([z(k)])$



Comments:

Setting sampling rate and the value of *ΔU* is a critical issue, and a crucial decision of the designer.

The figure shows the idealized trajectories of the motion.

The real trajectories due to noise/disturbance do return to themselves.

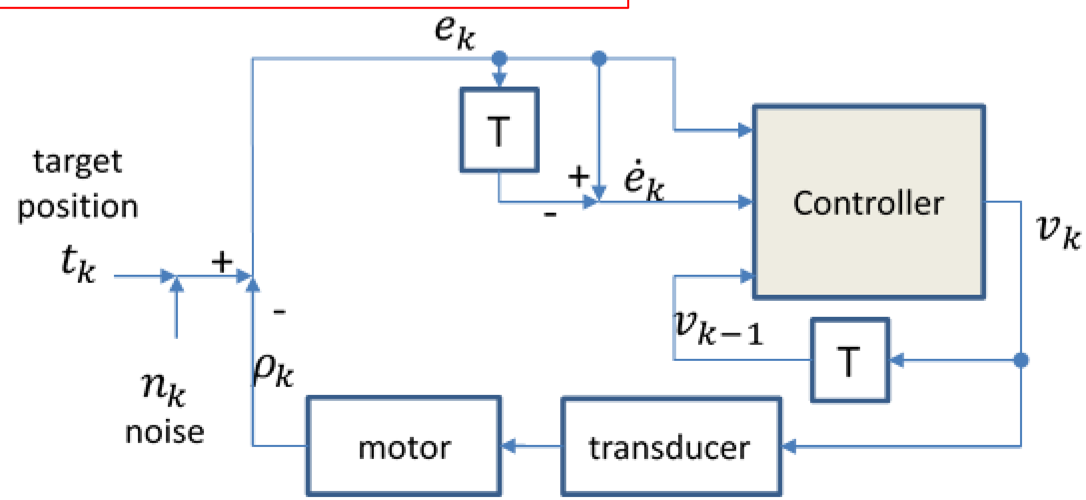
Example: Adaptive target tracking with fuzzy modelling and control

The target tracking system consists of two channels: it maps azimuth-elevation inputs to motor control outputs. The nominal target moves through azimuth-elevation space. Two motors adjust the platform to continuously point towards the target.

azimuth (0 … 180 degrees), elevation (0 … 90 degrees).

Sensor:

The platform can be any directional device that accurately points towards the target.
The device may be a laser, radar, video camera or high-gain antenna.



Notation:

t_k	target position;
n_k	observation noise;
ρ_k	tracking position;
e_k	tracking error;
$\dot{e}_k$	tracking error change;
v_k	estimated angular velocity;
T	sampling time.

$$\rho_k = \rho_{k-1} + T v_{k-1} + error$$

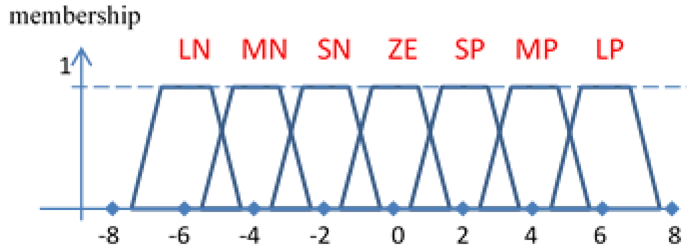
error = positioning uncertainty

Fuzzy controller: We restrict the output angular velocity of the fuzzy controller to the interval: [-6,6]. (This is a decision of the designer, a scaling factor)

Since $|v_k| \le \frac{9.0}{T} degrees/sec$ azimuth, and $|v_k| \le \frac{4.5}{T} degrees/sec$ elevation, thus the output gains of the channels are: $1.5/T$ and $0.75/T$.

The fuzzy controller uses heuristic control set-level "rules" or fuzzy-associative-memory (FAM) associations, based on quantised values of $e_{k,}$, $\dot{e}_k$ and v_{k-1}.

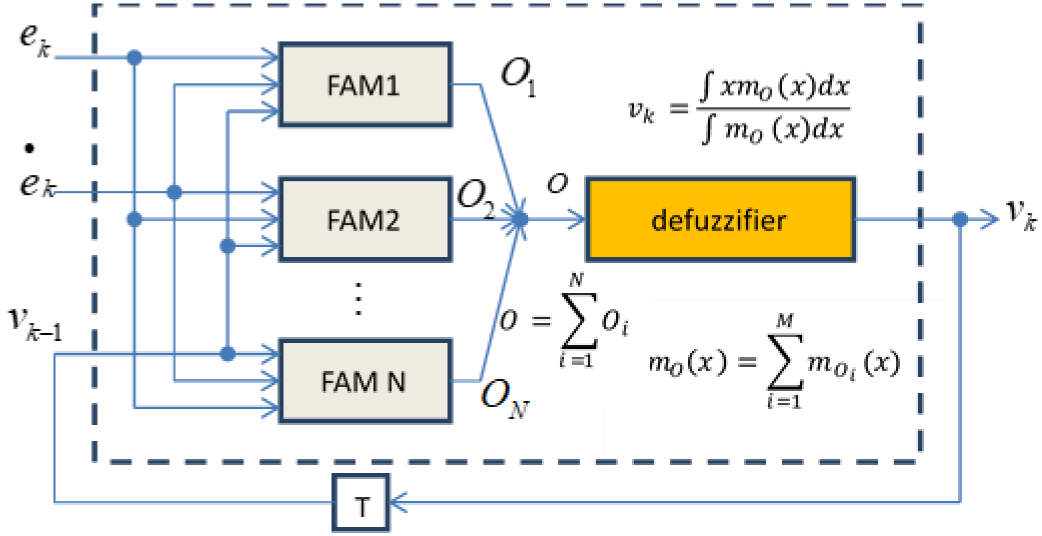
We define seven fuzzy levels by the following library of fuzzy-set values of the fuzzy variables e_k, $\dot{e}_k$ and $v_{k\text{-}1}$:



LN=Large Negative
MN=Medium Negative
SN=Small Negative
ZE=Zero
SP=Small Positive
MP=Medium Positive
LP=Large Positive

We assign each system input to a fit vector of length 7:

1 → (0 0 0 0.7 0.7 0 0)
-4 → (0 1 0 0 0 0 0)
3.8 → (0 0 0 0 0.1 1 0)



We formulate control FAM rules by associating output fuzzy sets with input fuzzy sets: For example the *i*-th rule:

$$IF\ e_k = MP \wedge \dot{e}_k = SN \wedge v_{k-1} = ZE\ THEN\ v_k = SP$$

We abbreviate this to: (*MP,SN,ZE;SP*).

The scalar value of the *i*-th FAM rule : $w_i = \min(membership\ values)$.

Example: $e_k = 2.6,\ \dot{e}_k = -2.0, v_{k-1} = 1.8.$ The fit vectors of length 7:

LN	MN	SN	ZE	SP	MP	LP
0	0	0	0	1	0.4	0
0	0	1	0	0	0	0
0	0	0	0.1	1	0	0

The membership values associated to the rule:

$$m_{MP}(e_k) = 0.4$$
$$m_{SN}(\dot{e}_k) = 1$$
$$m_{ZE}(v_{k-1}) = 0.1$$

The scalar value of the *i*-th rule:

$$w_i = \min(0.4, 1, 0.1) = 0.1$$

The form of the output fuzzy set depends on the encoding of the FAM rule:

Correlation-product encoding:

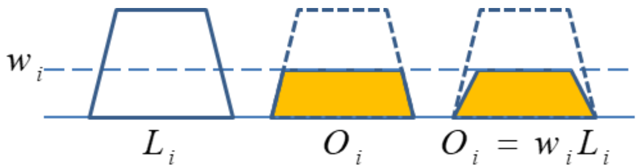
$$m_{O_i}(x) = w_i m_{L_i}(x)$$

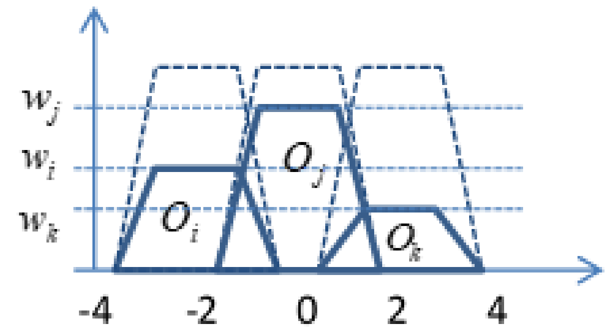
Correlation-minimum encoding:

$$m_{O_i}(x) = \min(w_i, m_{L_i}(x)).$$

Here $m_{L_i}(x)$ stands for the membership function of the output associated to the *i*-th FAM rule.

Two possible encoding strategies of the output fuzzy sets:

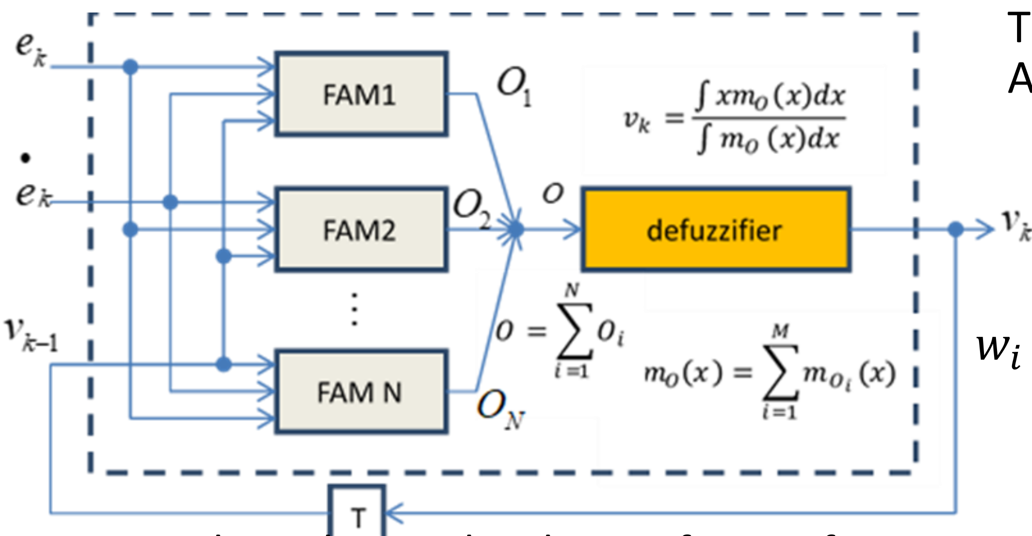




The ***defuzzifier*** assigns numerical value to the sum of the output fuzzy sets associated with the FAM rules.
This summed set is the sum of weighted trapezoids.

It is like the probability density function of probability theory, except the integral of the summed function differs from one.

The ***defuzzifier*** computes the v_k value as a centroid, therefore it is called: ***fuzzy centroid***.

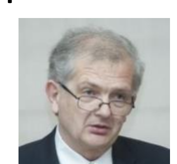


The implementation of the fuzzy controller:
A FAM$_i$ rule: (*MP,SN,ZE;SP*).

At the *k*-th time instant: $e_k = 2.6$, $\dot{e}_k = -2.0, v_{k-1} = 1.8$.

$$w_i = min\left(m_{MP}(e_k), m_{SN}(\dot{e}_k), m_{ZE}(v_{k-1})\right) = \\ = min(0.4,1,0.1) = 0.1$$

Since in this solution the shape of every fuzzy set is the same, we can write: e.g. $m_{SP}(x) = m_{ZE}(x-2)$. In general $m_{L_i}(x) = m_{ZE}(x - c_{L_i})$, where c_{L_i} is the centroid of the given membership function.

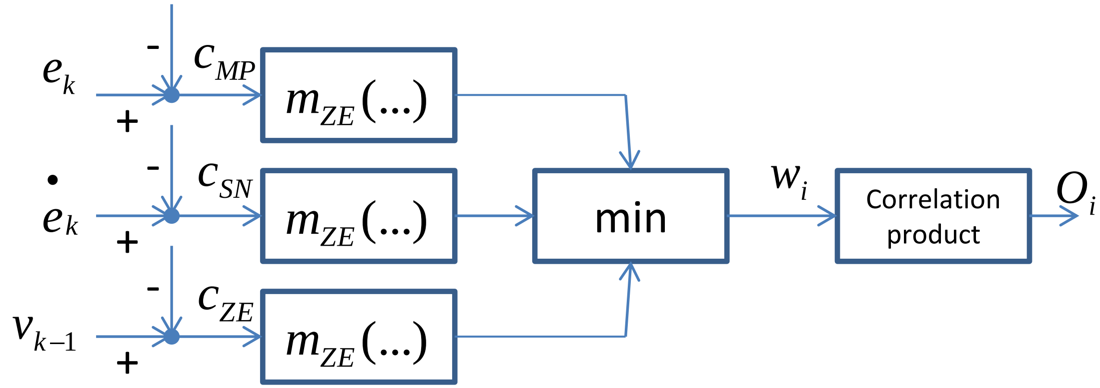


$$w_i = min\left(m_{MP}(e_k), m_{SN}(\dot{e}_k), m_{ZE}(v_{k-1})\right) = min(0.4,1,0.1) = 0.1$$

$$w_i = min\left(m_{ZE}(e_k - c_{MP}), m_{ZE}(\dot{e}_k - c_{SN}), m_{ZE}(v_{k-1} - c_{ZE})\right)$$

$$w_i = min\left(m_{ZE}(-1.4), m_{ZE}(0), m_{ZE}(1.8)\right) = min(0.4,1,0,1) = 0.1$$

In case of correlation-product encoding: $m_{O_i}(x) = w_i m_{ZE}(x - c_i)$, thus the implementation of the i-th FAM rule can have the following form:



Thank you for your attention!