Beágyazott információs rendszerek

Szenzorhálózatok
Biztonságkritikus rendszerek

2020. november 19.

Szenzorhálózatok

Emlékeztető

A szenzorhálózatokról részletes fólia-sorozat található a tantárgy tanszéki honlapján.
Az alábbiak csak néhány kiemelt jellemzőt foglalnak össze.

A szenzorhálózati csomópontok jellegzetes megjelenési formája a **Berkeley Mica2 mote**, amely a fényképen látható. Mérete a nyomtatott áramköri lap alatt elhelyezett **2*AA elem** alapján becsülhető.
Felépítése és hardver jellemzői a **Szenzorhálózatok I.** című dokumentumban leírtak alapján ismerhető meg.
Ugyanitt olvasható az eszköz szóba jövő alkalmazásainak listája.
Az alkalmazások jellemzője a **térbeli kiterjedés**, és a szükséges **csomópontok nagy száma**.
A működés során lényeges az **energiatakarékosság**.

A TinyOS operációs rendszer

A TinyOS operációs rendszerről részletes fólia-sorozat található a tantárgy tanszéki honlapján.
Az alábbiak csak néhány kiemelt jellemzőt foglalnak össze.

Miért van rá szükség?

A tradicionális operációs rendszerekkel nehézségek vannak szenzorhálózatok esetében, mert a **többszálas architektúra nemigen használható** kellő hatékonysággal, nagy a **memóriaigény**, az **energiafelhasználás** minimalizálását nem támogatják.

A vezeték nélküli szenzorhálózatok esetében lényeges:
(1) a konkurens végrehajtás, (2) az energiafelhasználás hatékonysága,
(3) kis memóriaigény (small memory footprint), és (4) a sokrétű felhasználás támogatottsága.

Emlékeztető

Főbb jellegzetességei:

A TinyOS nyílt hozzáférésű operációs rendszer, amely kifejezetten vezeték nélküli szenzorhálózati alkalmazásokhoz készült. Komponens alapú, NesC (Networked embedded system C) nyelven íródott a University of California, Berkeley és az Intel Research együttműködésében.

A komponens alapú architektúra lehetővé teszi a gyakori változtatásokat, és eközben a kódméret minimális szinten tartható. A végrehajtás eseményvezérelt, és ebből adódóan nagymértékben konkurens. Energia hatékony, mert a processzor - amint lehetséges – *sleep* állapotba kerül.

Kicsi a “lábnyoma”, mert FIFO alapú, nem megszakítható ütemezést alkalmaz.

Statikus memória allokációt használ, a memória követelmények fordítási időben dőlnek el.

A lokális változók mentése a stack-re történik.

Energia hatékony, kétszintű ütemezést használ:

(1) Hosszan futó taszkok és események okozta interruptok,

(2) Sleep üzemmód hacsak nincs taszk a sorbaállási sorban, ébresztés eseményre.

A taszkok idő-flexibilis háttér jobok, egymáshoz viszonyítva atomikusak, azaz egymás nem szakítják meg, futásukat csak interruptként megjelenő események szakíthatják meg.

Az események időkritikus, rövidebb idejű programrészek, LIFO (Last-in First-Out) logika szerint kerülnek feldolgozásra, kezdeményezhetik taszkok késleltetett futását.

A programok komponensekből épülnek fel, minden komponens specifikál egy interfészt, és ezek segítségével kerül sor a “huzalozásra”, aminek eredménye a konfiguráció.

A komponenseknek kétirányú interfészeket használnak (use), ill. biztosítanak (provide).

A komponensek parancsokat (command) hívnak és implementálnak, és eseményeket (events) jeleznek és kezelnek.

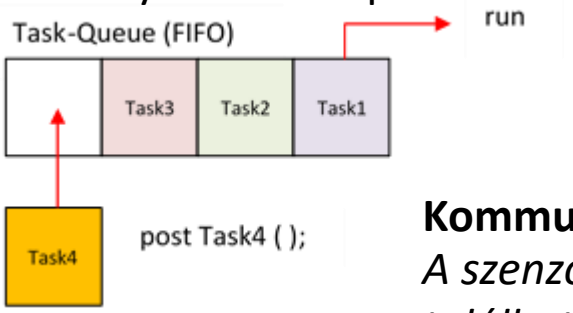
Emlékeztető

A komponensek a használt (used) interfészeken érkező eseményeket kezelik, és a parancsokat implementáló interfészeket biztosítanak (provide).

A komponensek hierarchiája:

A parancsok “lefelé” haladnak, nem blokkoló kérések, a vezérlés a hívóhoz kerül vissza.

Az események “felfelé” haladnak, taskot helyeznek el a várólistán (function queue scheduling), alacsonyabb szintű parancsot hívnak. A vezérlés a jelzést adóhoz kerül vissza.



Ciklusok elkerülésére azáltal kerül sor, hogy az események hívhatnak parancsokat, de a parancsok nem tudnak eseményt kezdeményezni.

Kommunikáció szenzorhálózatokban

A szenzorhálózatokon belüli kommunikációról részletes fólia-sorozat található a tantárgy tanszéki honlapján. Az alábbiak csak néhány kiemelt jellemzőt foglalnak össze.

Szabványos megoldások:

tipikusan az ISM (Industrial, Scientific, Medical) 2.4 GHz-es sávban, szórt spektrummal:

ZigBee/IEEE 802.15.4, IEEE 802.11b (Wi-Fi) WLAN (Wireless Local Area Network),

Bluetooth WPAN (Wireless Personal Area Network).

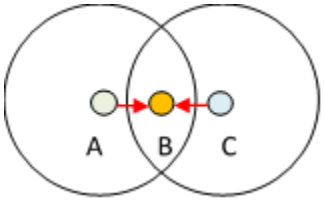
Szenzorhálózatokban tipikus a dinamikus (igény szerinti) csatorna-hozzáférési jog kiosztás, ezen belül is a CSMA: Carrier Sense Multiple Access.

Az ütközés elkerülés módja: adás előtt belehallgat a csatornába, ha nem érzékel adást, akkor adni kezd, ha adást érzékel, akkor vár.

Emlékeztető

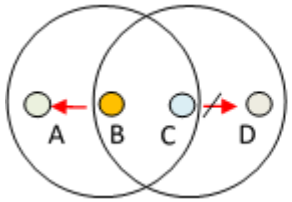
CSMA problémák:

Rejtett terminál problémája:



- A ad B-nek
- C nem hallja A-t!
- C is ad B-nek
- B egyik adást sem tudja venni

Látható terminál problémája:



- B ad A-nak
- C is szeretne adni D-nek!
- C hallja B-t
- C nem ad, bár nem okozna ütközést

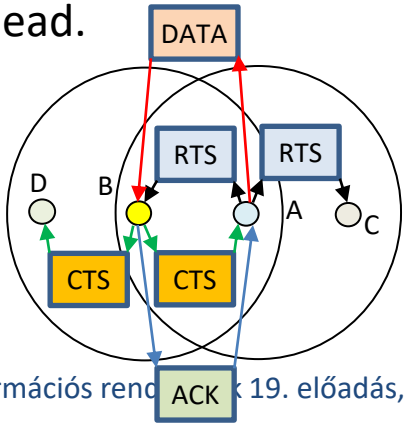
CSMA módosítások:

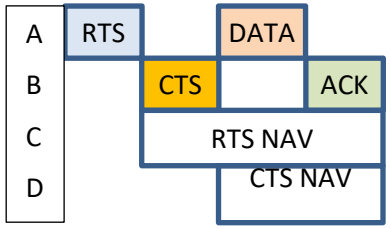
CSMA foglaltság jelzéssel: két csatornát használunk, az egyiket az adat továbbítására, a másikat a foglaltság jelzésére. A vevő a foglaltság csatornán folyamatosan jelez.
Adás előtt az adó ellenőrzi mind az adatcsatornát, mind a foglaltság-csatornát.
A csomópontnak egyszerre kell adni és venni, ami költséges.
Az egyidejű két csatorna nagyobb sávszélességet köt le.

Request To Send/Clear To Send (RTS/CTS): Két fázisban működik:

(1) Handshake, (2) adattovábbítás. Az alapgondolat: az ütközés a vevőnél történik.

Kizárja a rejtett terminál problémát. Hosszabb üzenetek esetén előnyös, egyébként nagy az overhead.





Az „A” adó RTS üzenetet küld (“B”-nek)
A „B” vevő CTS üzenettel válaszol
Az adó a CTS vétele után továbbítja az adatcsomagot
A többi csomópont RTS, CTS vétele után nem adhat!
(NAV = Network Allocation Vector)

Emlékeztető

Routing (Adásvonal vezetés, útvonalválasztás)

A szenzorhálózat ad-hoc. A csomópontok véletlen eloszlásúak, a kapcsolatok véletlenszerűen jönnek létre, nem megbízhatóak (fading), a csomópontok lehetnek mobilak, és lehetnek sokan.

Tipikus feladatok:

- Egy forrás → sok (akár minden csomópont) cél. Pl. egy központi csomópont utasításokat terjeszt a hálózatban.
- Sok forrás → egy cél. Pl. adatgyűjtés és továbbítás a központba.
- Egy forrás → egy cél. Pl. adatcsere csomópontok között.

Adatküldési modellek:

- *Idővezérelt:* a szenzorok működése és az adatküldés idővezérelt. Tipikus alkalmazás: előre eltervezett adatgyűjtés. Energiatakarékos működéshez előnyös. Alvás → szinkronizált ébredés.
- *Eseményvezérelt:* a szenzorok működését környezeti események kezdeményezik. Időkritikus alkalmazásoknál célszerű. Energiatakarékos üzem nehezebben valósítható meg.
- *Lekérdezéses:* a szenzorok a központ lekérdező parancsára aktiválódnak.

Tipikus hálózati struktúrák:

- *Egyszintű (Flat):* Egyenrangú csomópontok, nehezen skálázható.
- *Hierarchikus:* Csoportok alakulnak: csoporton belüli és csoportok közötti kommunikáció működik. A csoportok között vezérlő csomópontok tartják a kapcsolatot. A vezérlők kitüntetett képességűek. A vezérlő szerep dinamikusan változhat.

Egyszintű: ***Elárasztásos adásvonal vezetés/útvonalválasztás (Flood routing):***

- ***Üzenetszórásos*** (broadcast) üzenettovábbítás.
- Minden üzenet első vételekor a vevő megjegyzi az üzenetet vagy ha nem neki szól akkor csak az azonosítóját, majd szétsugározza az üzenetet.
- ***Tipikus alkalmazás:*** egy forrás → sok cél (parancsot kap kvázi egyidejűleg).
- ***Előnye:*** egyszerű, hibatűrő a nagy redundancia miatt.
- ***Hátránya:*** rengeteg (feleslegesnek bizonyuló) üzenet és energiafogyasztás, továbbá ütközések (rejtett terminál.)
- ***Módosítások:***
- a vevő csak *p* valószínűséggel terjeszt tovább. A *p* topológia-függő.
- az ütközések elkerülése érdekében: a vétel után késleltetett továbbítás, véletlen várakozási idő.

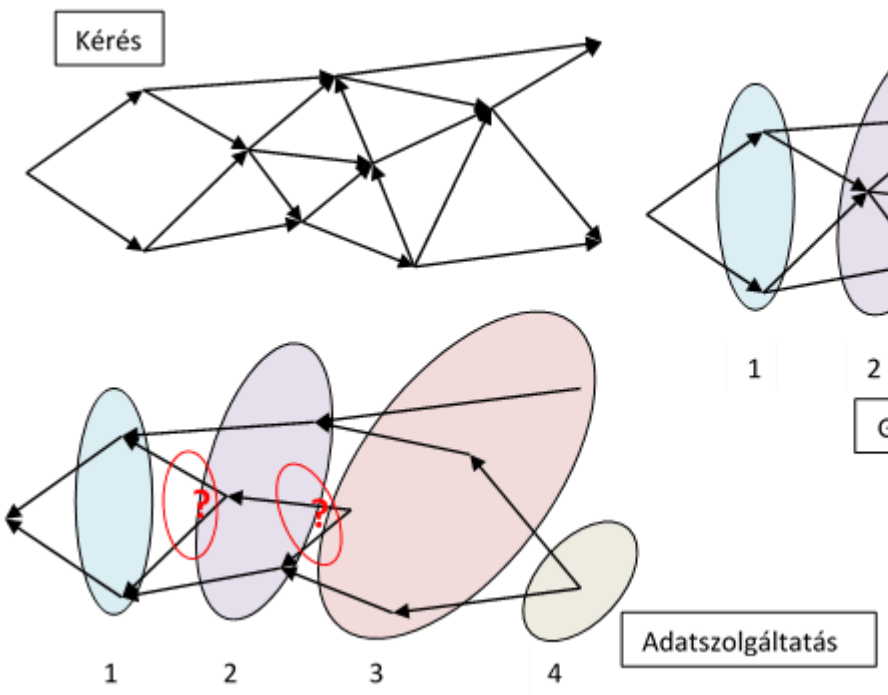
Egyszintű: ***Gradiens-alapú adásvonal vezetés/útvonalválasztás (Gradient Based Routing (GBR)):***

- *Három fázis:* (1) Kérés, (2) Gradiens számítás, (3) adatszolgáltatás.
- *Tipikus alkalmazás:* **sok forrás → egy cél (adatgyűjtés).**

(1) **Kérés:** a központ kérést küld a hálózatba: terjesztés elárasztással.

(2) **Gradiens számítás:** a kérés terjesztése közben „gradiens mérés” → A “gradiens” a legrövidebb “távolság” a központtól: Legkevesebb hány lépésben küldhető meg a kért adat a központnak.

(3) **Adatszolgáltatás:** adat továbbítás a legrövidebb távolságú úton, eközben aggregáció lehetséges.



GBR változatok: Több lehetséges útvonalból melyiket válasszuk?

sztochasztikus: véletlen választás

***energia-alapú* kiegészítés:** a kevés energiájú csomópont megemeli a saját „gradiens” értékét, így másfelé tereli a forgalmat.

Hierarchikus: csak említés szintjén, a név alapján visszakereshető az interneten:

Low Energy Adaptive Clustering Hierarchy (LEACH): Hierarchikus, dinamikusan létrejövő klaszterekre alapozva.

Geographic and Energy Aware Routing (GEAR): Elhelyezkedés alapú, üzenet csak a célzott régió felé halad.

Biztonságkritikus rendszerek (néhány alapvetés)

(Az alábbiak kivonatok **Dr. Majzik István** (BME MIT) a **"Valósidejű és biztonságkritikus rendszerek"** című tárgyhoz készített előadásvázlatából.)

Biztonsági követelmények rendszere

- Kockázatelemzés:

Tolerable Hazard Rate (THR): eltűrhető veszélygyakoriság, eltűrhető veszély ráta

- Folyamatos üzem esetén a veszélyt okozó hibajelenség gyakorisága óránként;
- Nem folytonos üzem esetén a veszélyt okozó hibajelenség valószínűsége a funkció meghívásakor

- Kategóriákba sorolás: **Safety Integrity Level (SIL)** – Biztonságintegritási szint

SIL	Biztonságkritikus funkció hibája/óra
1	$10^{-6}<THR<10^{-5}$
2	$10^{-7}<THR<10^{-6}$
3	$10^{-8}<THR<10^{-7}$
4	$10^{-9}<THR<10^{-8}$

1 év = 8760 óra. **SIL4** feltételezésével:
$10^8/8760 \cong 11415$ év hiba nélkül.
Ha **15** év az élettartam, akkor ~**750** berendezésből egyben lesz hiba, mert **15*750=11250.**

A szolgáltatásbiztonság alapjellemzői:

- Megbízhatóság: a rendszer folyamatos szolgáltatást nyújt;
- Rendelkezésre állás: a rendszer (javítva) használatra kész;
- Biztonság(osság): nincs káreset/baleset;
- Bizalmasság: nincs jogosulatlan információközlés;
- Karbantarthatóság: javítás és fejlesztés lehetősége;

A szolgáltatásbiztonság további jellemzői:

- Teljesítőképesség; teljesítmény és megbízhatóság.
- Tesztelhetőség: tesztelés lehetősége;

Megbízhatósági mértékek:

- Állapot particionálás: $s(t)$ rendszerállapot: **hibás** (D), **hibamentes** (U) állapotpartíció.

Várható értékek:

- Első hiba bekövetkezése: MTFF=E{u1} Mean Time to First Failure
- Hibamentes működési idő: MUT=E{ui} Mean Up Time
- Ugyanez: MTTF Mean Time To Failure
- Hibás állapot ideje: MDT=E{di} Mean Down Time
- Ugyanez: MTTR Mean Time To Repair
- Hibák közötti idő: MTBF=MUT+MDT Mean Time Between Failures

Valószínűségi időfüggvények:

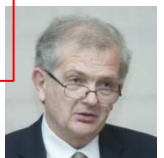
- Rendelkezésre állás: a(t)=P{s(t)∈ U} közben meghibásodhat (idővel csökken)
- Megbízhatóság: r(t)=P{s(t') ∈ U} $\forall t' < t$, nem hibásodhat meg
- Készenlét: K=$lim_{t\to\infty}$a(t) rendszeresen javított rendszer esetén (idővel nullára csökken)
- Készenlét: K=A=MTTF/(MTTF+MTTR)

Komponens jellemzők:

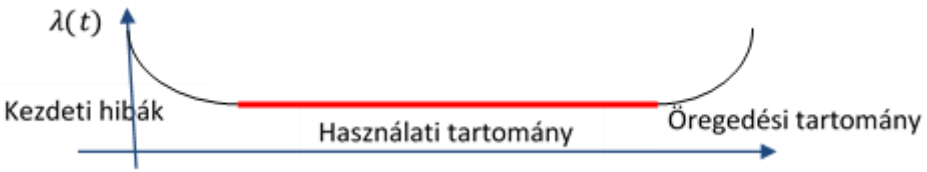
- Meghibásodási tényező: $\lambda(t)$ milyen valószínűséggel hibásodik meg t környezetében?

$$\lambda(t)\Delta t = P\{s(t+\Delta t) \in D | s(t) \in U\}, \Delta t \to 0.$$

részletezve (lásd az eloszlás és sűrűségfüggvények, valamint deriváltjaik kapcsolatát): $\lambda(t) = -\frac{1}{r(t)}\frac{dr(t)}{dt}$, amivel $r(t) = e^{-\int_0^t \lambda(t)dt}$.



Elektronikai alkatrészek kádgörbéje:



A használati tartományban $\lambda(t) = \lambda$.

Exponenciális eloszlást feltételezve:

$$r(t) = e^{-\lambda t}$$

$$MTFF = E\{u1\} = \int_0^\infty r(t)dt = \frac{1}{\lambda}$$

Megjegyzés: A kezdeti hibákat gyártás utáni teszttel szűrik ki.

Szolgáltatásbiztonság befolyásoló tényezői:

- **Hibajelenség (failure):** a specifikációnak nem megfelelő szolgáltatás (értékbeni/időbeni, katasztrofális/"jóindulatú");
- **Hiba (error):** hibajelenséghez vezető rendszerállapot (lappangó → detektált);
- **Meghibásodás (fault):** a hiba feltételezett oka;
- **Hatás:** alvó → aktív;
- **Fajta**: véletlen vagy szándékos, időleges vagy állandósult;
- **Eredet:** fizikai/emberi, belső/külső, tervezési/működési;

A fejlesztési folyamat: V-modell, verifikáció, validáció, tesztelés, …

A szervezeti rend: Szereplők:

- tervező (elemző, tervező, kódoló, unit tesztelő funkciók);
- verifikátor (igazoló);
- validátor (érvényesítő);
- értékelő (független felülvizsgáló);
- projekt menedzser;
- minőségbiztosítási felelős.

SIL0 esetén: a tervező, verifikátor, validátor lehet ugyanaz a személy, az értékelő más kell legyen;

SIL1 és SIL2 esetén: a tervező, a verifikátor-validátor és az értékelő más kell legyen;

SIL3 és SIL4 esetén: a menedzser, a tervező, a verifikátor-validátor és az értékelő más kell legyen; akár a verifikátor és a validátor is.

Architektúra tervezés a veszély elkerülése érdekében

Fail-safe működés: (1) fail-stop (cél a rendszer leállítása),
(2) fail-operational (a leállás nem biztonságos, valamilyen szinten a szolgáltatást biztosítani kell).

Jellegzetes megoldások fail-stop működéshez

- Egycsatornás feldolgozás öntészttel;
- Két- vagy többcsatornás feldolgozás: (a) ugyanazzal a programmal, (b) nem ugyanazzal a programmal (független ellenőrzés);

Hibatűrő működés

Redundancia: (1) Hardver, (2) Szoftver, (3) Információ, (4) Idő.
Redundancia típusai: hideg tartalék, langyos tartalék, meleg tartalék:

Redundancia/tulajdonság	Hideg tartalék	Langyos tartalék	Meleg tartalék
Alapelv	Csak hiba esetén aktiválva	Csökkentett terheléssel működik	Ugyanúgy működik, mint az elsődleges
Előnye	Nem hibásodik meg a passzív komponens	Kisebb meghibásodási tényező	Gyorsan átveheti az elsődleges helyét
Hátránya	Lassan veszi át az elsődleges helyét	Közepes sebességű feladat átvétel	Azonos meghibásodási tényező
Példa	Kikapcsolt tartalék számítógép	Naplózó számítógép belép elsődlegesként	Árnyék számítógép

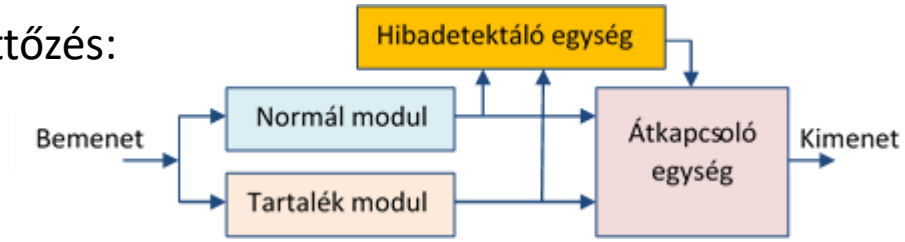
Milyen redundancia használandó?

- Hardver tervezési hibák (<1%): hardver redundancia, eltérő tervezésű.
- Hardver állandósult működési hibák (~10%): hardver redundancia, pl. tartalék processzor.
- Szoftver tervezési hibák (~10-20%): szoftver redundancia, eltérő tervezésű.

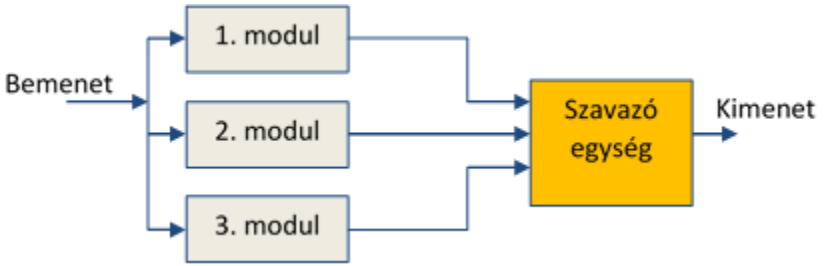
- Hardver időleges működési hibák (~70-80%): idő-redundancia (pl. utasítás újra-végrehajtás), információ redundancia (pl. hibajavítás), szoftver redundancia (pl. állapotmentés és helyreállítás).

Állandósult hardver hibák kezelése:

Kettőzés:



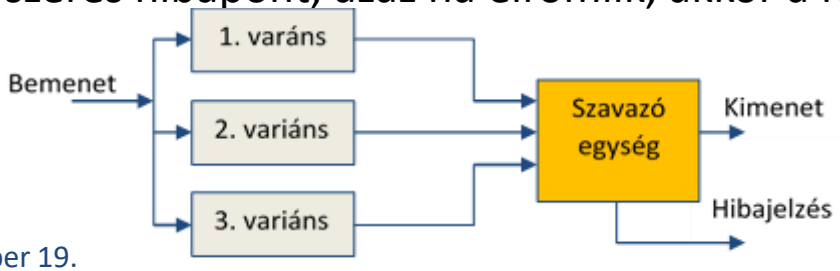
Alapesetben csak hibadetektálás, a hibatűréshez diagnosztikai támogatás és átkapcsolás kell.



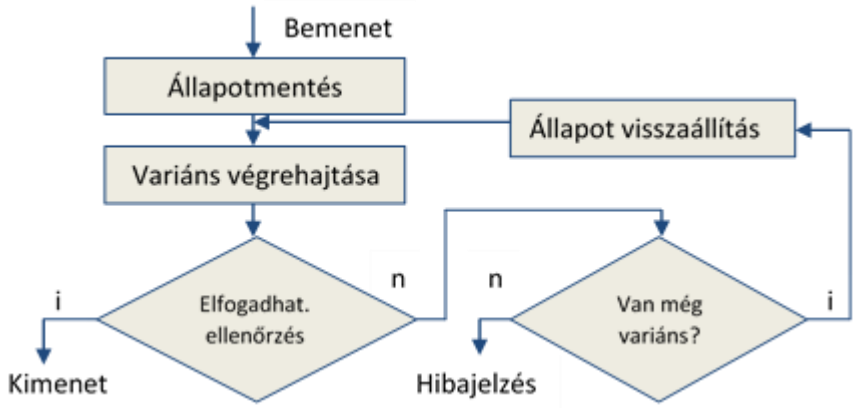
TMR: Triple-modular redundancy:
Hiba maszkolása többségi szavazással.
NMR: N-modular redundancy:
Hiba maszkolás többségi szavazással.
A missziós idő túlélése nagyobb esélyű, utána javítás lehetséges.
Repülőgép fedélzeti eszközök: 4MR, 5MR, esetenként 7MR.

Szoftver hibák kezelése: Variánsok alkalmazása: azonos specifikáció; de eltérő algoritmus, adatstruktúra; más fejlesztési környezet, programnyelv; elszigetelt fejlesztés.

N-verziós programozás: aktív redundancia, a variánsok párhuzamos végrehajtása, többségi szavazás. Ha a variánsok kimeneteire elfogadhatósági tartományt is adunk, akkor a szavazó azt is ellenőrzi. A szavazó maga ún. egyszeres hibapont, azaz ha elromlik, akkor a funkció kiesik, de a szavazó egyszerű, ezért kisebb a kockázat.



Javító blokkok technikája: passzív redundancia, csak hibaesetén aktiválódik.



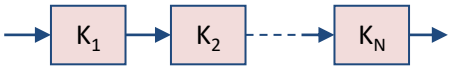
A variánsok kimenetének elfogadhatóságát ellenőrizzük, ha erre nincs lehetőség, akkor a módszer nem alkalmazható.
Ha hiba lép fel, akkor tartalék variáns soros végrehajtására kerül sor.
Az alkalmazás feltétele, hogy legyen lehetőség elfogadhatósági ellenőrzésre.

Összehasonlítás:

Tulajdonság/típus	N-verziós programozás	Javító blokkok
Ellenőrzés	Szavazás, relatív	Elfogadhatóság, abszolút
Végrehajtás	Párhuzamos	Soros
Időigény	Leghosszabb variáns v. time-out	Hibák számától függ
Redundancia aktiválása	Mindig	Csak hiba esetén
Tolerált hibák	[(N-1)/2]	N-1
Hibakezelés	Maszkolás	Helyreállítás

Megbízhatósági blokkdiagram (Reliability Block Diagram)

1. Soros rendszer: a komponensek sorba kapcsolódnak:



A rendszer akkor hibátlan, ha valamennyi komponens az.
A rendszer megbízhatósága a komponensek megbízhatóságának szorzata: $r_R(t) = \prod_{i=1}^{N} r_i(t)$.
Ha a komponensek meghibásodási tényezője λ_i, akkor a rendszer

$$MTFF = \frac{1}{\sum_{i=1}^{N} \lambda_i}$$

2. Párhuzamos rendszer: A komponensek párhuzamosan kapcsolódnak:
A rendszer akkor hibás, ha valamennyi komponens hibás.
A hiba valószínűsége: (1- megbízhatóság).

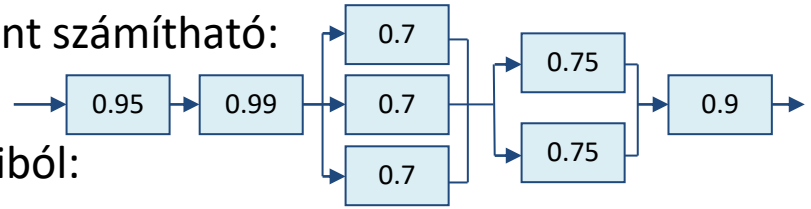
$$1 - r_R(t) = \prod_{i=1}^{N}(1 - r_i(t)).$$

Ha a komponensek megbízhatósága azonos: $r_K(t)$, akkor $r_R(t) = 1 - \left(1 - r_K(t)\right)^N$

Ha a komponensek meghibásodási tényezője λ, akkor a rendszer

$$MTFF = \frac{1}{\lambda}\sum_{i=1}^{N}\frac{1}{i}.$$

3. Összetett rendszer: részenként számítható:



A rendszer készenlét a komponensek készenléti adataiból:

$$\boldsymbol{K_R} = 0.95 \cdot 0.99 \cdot [1 - (1 - 0.7)^3] \cdot [1 - (1 - 0.75)^2] \cdot 0.9$$
$$= 0.95 \cdot 0.99 \cdot 0.973 \cdot 0.9375 \cdot 0.9 = \mathbf{0.77}$$

4. N-ből M hibás komponens esete:

N egyforma komponens, M vagy több komponens hiba esetén a rendszer is hibás.
A rendszer megbízhatósága (a komponensek megbízhatósága egyforma: *r*):

$$r_R = \sum_{i=0}^{M-1} P\{\textit{éppen i hiba van}\} = \sum_{i=0}^{M-1}\binom{N}{i}(1-r)^i r^{N-i}$$

Ideális többségi szavazás (TMR): N=3, M=2 esetén:

$$r_R = \sum_{i=0}^{1}\binom{3}{i}(1-r)^i r^{3-i} = \binom{3}{0}(1-r)^0 r^3 + \binom{3}{1}(1-r)^1 r^2 = 3r^2 - 2r^3.$$

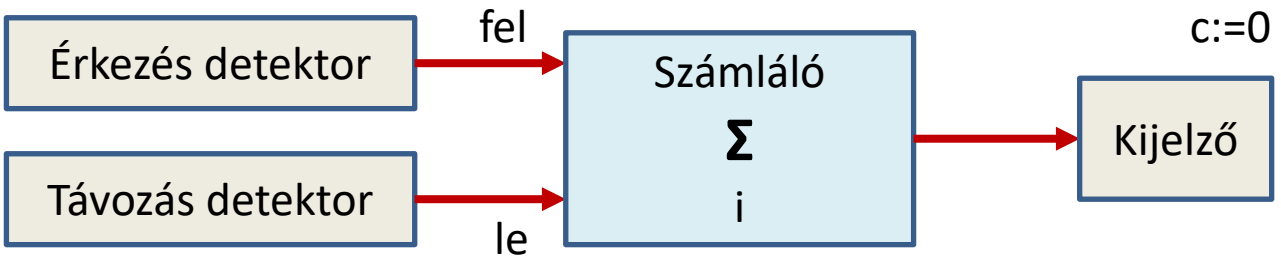
Exponenciális eloszlást feltételezve: $r(t) = e^{-\lambda t}$ alkalmazásával

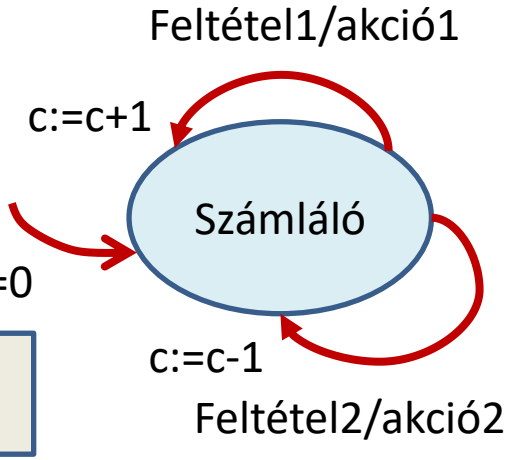
$$MTFF = \int_0^\infty r_R(t)dt = \int_0^\infty (3r^2 - 2r^3)dt = \frac{3}{2\lambda} - \frac{2}{3\lambda} = \frac{5}{6\lambda},$$

ami kisebb, mintha csak egy komponens lenne.

Diszkrét rendszerek

Példa: Parkoló gépkocsik száma egy parkolóházban (max. *M*)



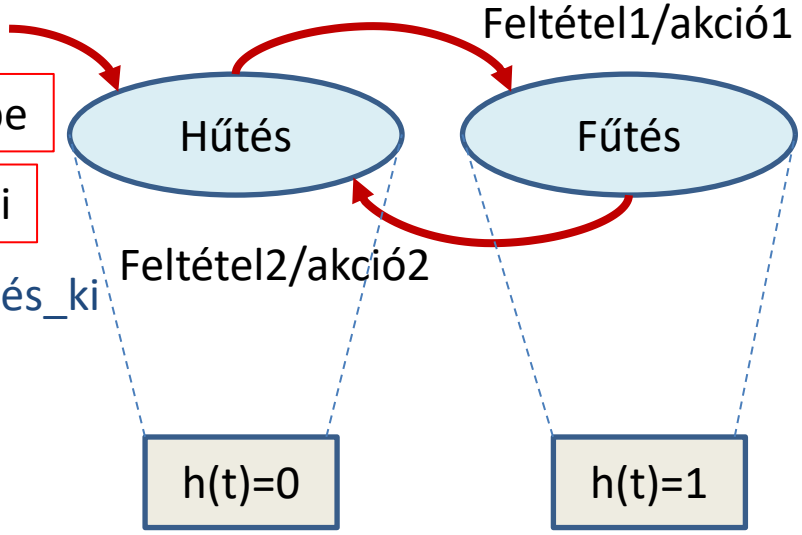


Feltétel1/akció1: $fel \wedge \neg le \wedge c < M \;/\; c+1$ Feltétel2/akció2: $le \wedge \neg fel \wedge c > 0 \;/\; c-1$

Példa: Termosztát hiszterézissel

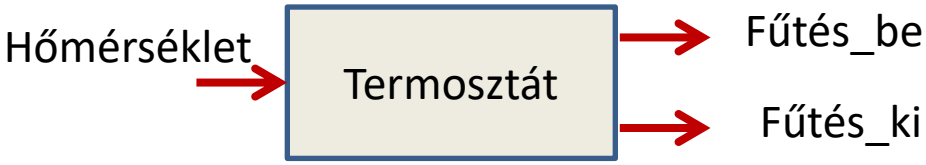
Feltétel1/akció1: Hőmérséklet ≤ 18 fok/fűtés_be

Feltétel2/akció2: Hőmérséklet ≥ 22 fok/fűtés_ki



Bemenet: hőmérséklet. **Kimenet:** fűtés_be, fűtés_ki

Ehhez rendelhető ún. **aktor modell**:

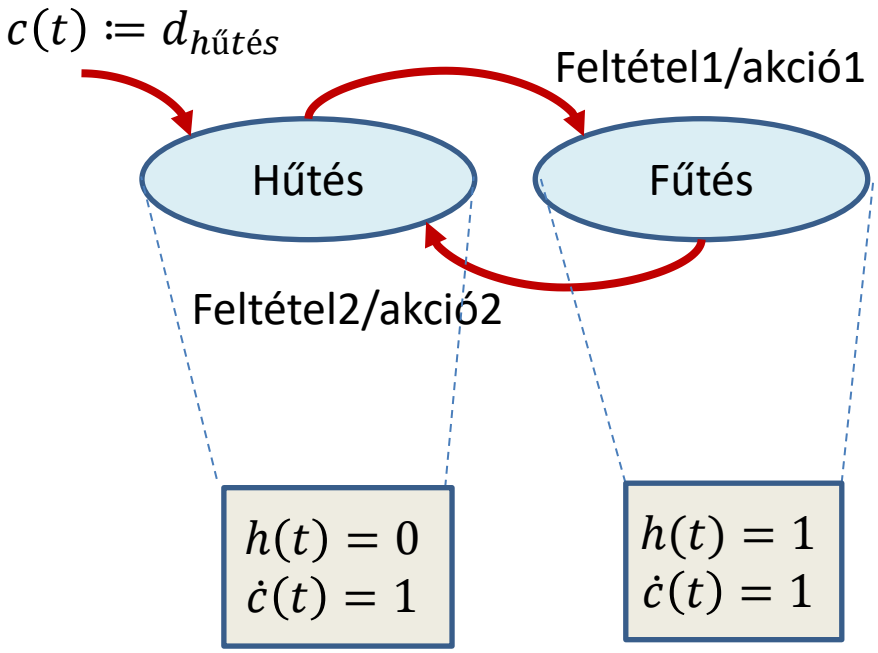


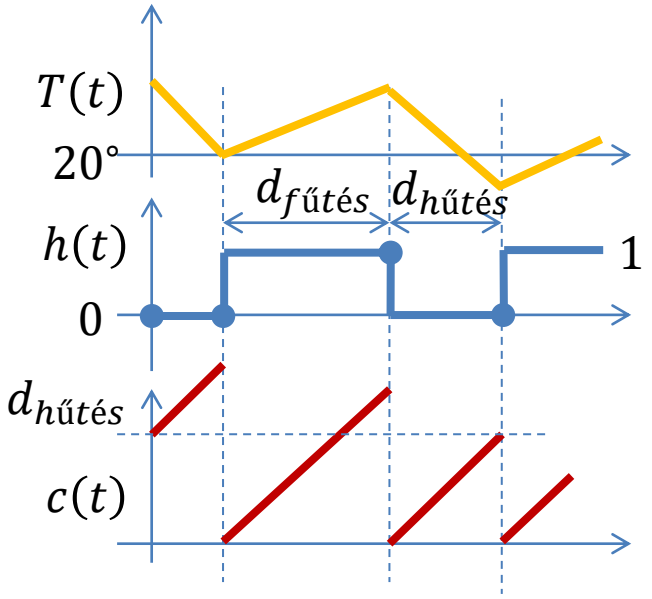
Hibrid rendszerek

A CPS rendszerekben szükség van az idő mérésére, és időzített akciók végrehajtására.

Az **időzített automata (timed automata)** a legegyszerűbb nem-triviális hibrid rendszer.
Az állapotaik mögött (adott időtartamig) mérik az idő múlását: $\forall t \in d_m$ $\dot{c}(t) = a$

Példa: Termosztát hiszterézis helyett időzítéssel





Megjegyzés: $h(t)$ és $c(t)$ az állapotfinomítás eszközei. Szokás (üzem)módról beszélni. (Modal systems)

Feltétel1/akció1: $T(t) \leq 20 \wedge c(t) \geq d_{hűtés} / c(t) \coloneqq 0$

Feltétel2/akció2: $T(t) \geq 20 \wedge c(t) \geq d_{fűtés} / c(t) \coloneqq 0$

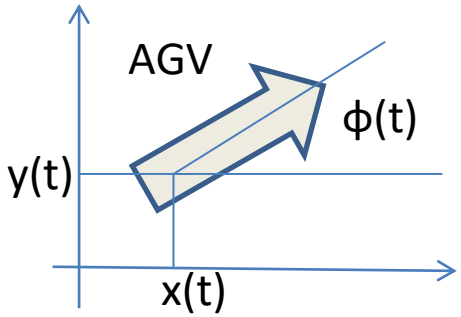
Hibrid rendszerek

Példa: Önjáró targonca (Automated Guided Vehicle, AGV)

Két szabadságfokú jármű, felfestett csík követésére képes. Minden *t* időpontban a hossztengelye mentén $v(t)$ sebességgel mozog azzal, hogy $0 \le v(t) \le 10\ km/h$

A súlypontja körül fordulni is tud $\omega(t)$ szögsebességgel, azzal hogy:

$-\pi \le \omega(t) \le \pi\ rad/sec$



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

Kétszintű szabályozás: a targonca mindig $10\ km/h$ sebességgel halad. Négy működési módja van: **balra, jobbra, egyenesen, megállás**.

Minden működési módhoz külön differenciálegyenlet tartozik:

egyenesen:
$$\dot{x}(t) = 10cos\big(\phi(t)\big)$$
$$\dot{y}(t) = 10sin\big(\phi(t)\big)$$
$$\dot{\phi}(t) = 0$$

balra:
$$\dot{x}(t) = 10cos\big(\phi(t)\big)$$
$$\dot{y}(t) = 10sin\big(\phi(t)\big)$$
$$\dot{\phi}(t) = \pi$$

jobbra:
$$\dot{x}(t) = 10cos\big(\phi(t)\big)$$
$$\dot{y}(t) = 10sin\big(\phi(t)\big)$$
$$\dot{\phi}(t) = -\pi$$

megállás:
$$\dot{x}(t) = 0$$
$$\dot{y}(t) = 0$$
$$\dot{\phi}(t) = 0$$

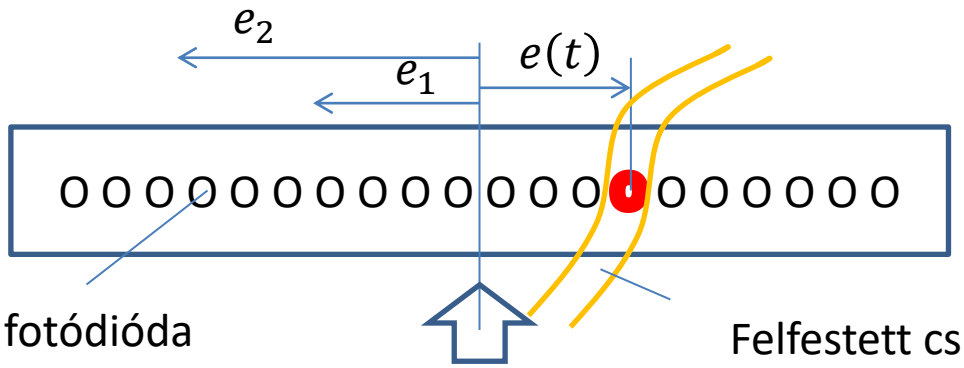
Hibrid rendszerek

A targonca érzékelője:

Kimenőjele: $e(t) = f\big(x(t), y(t)\big)$

$e(t) > 0$ balra tér el.

$e(t) < 0$ jobbra tér el.



A targonca vezérlése:

$|e(t)| < e_1$ egyenesen haladjon tovább!

$0 < e_2 < e(t)$ túlságosan eltér balra, forduljon jobbra!

$0 > -e_2 > e(t)$ túlságosan eltér jobbra, forduljon balra!

Bemeneti események halmaza: $u(t) \in \{stop, start, nincsesemény\}$

Állapotátmenetet generáló feltételek:

$$induljel = \{\big(v(t), x(t), y(t), \phi(t)\big) | u(t) = start\}$$

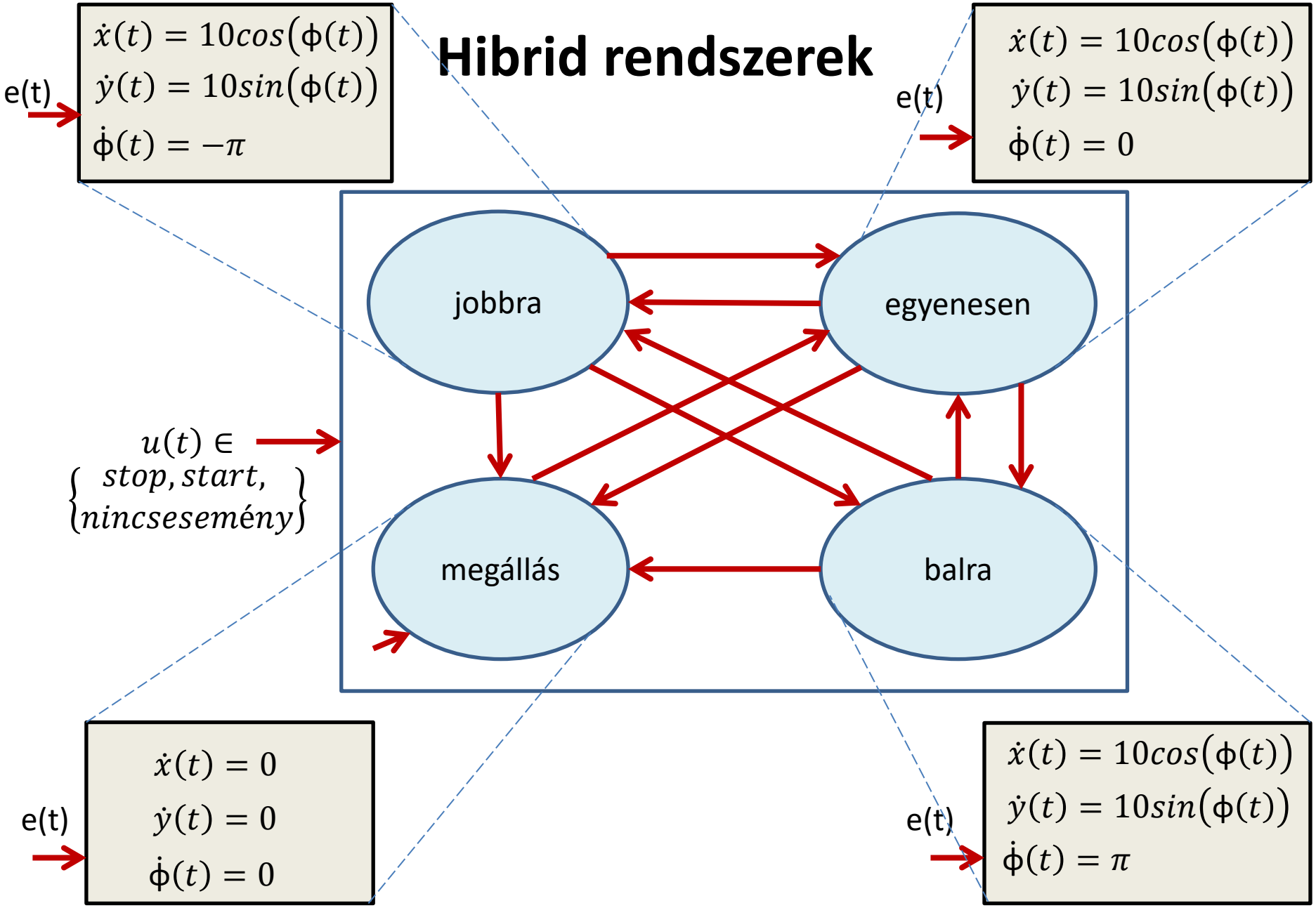
$$menjegyenesen = \{\big(v(t), x(t), y(t), \phi(t)\big) \big| u(t) \neq stop, |e(t)| < e_1\}$$

$$menjjobbra = \{\big(v(t), x(t), y(t), \phi(t)\big) | u(t) \neq stop, e_2 < e(t)\}$$

$$menjbalra = \{\big(v(t), x(t), y(t), \phi(t)\big) | u(t) \neq stop, -e_2 > e(t)\}$$

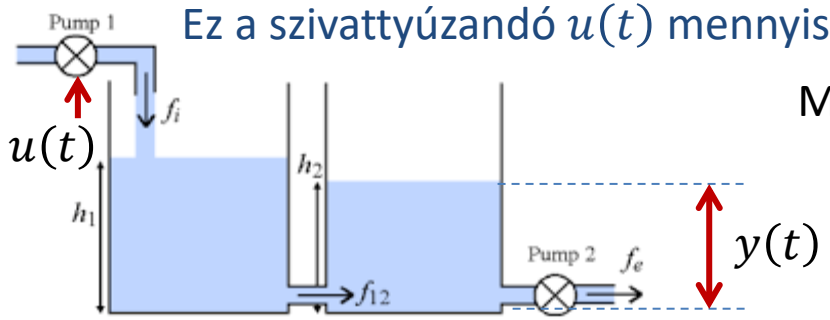
$$álljmeg = \{\big(v(t), x(t), y(t), \phi(t)\big) | u(t) = stop\}$$

Hibrid rendszerek



Kvalitatív modellezés és szabályozás

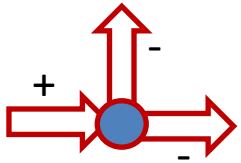
Feladat: Olyan szabályzó tervezése, amely a második tartály szintjét előírt értéken tartja. Ez a szivattyúzandó $u(t)$ mennyiség megfelelő beállításával lehetséges.



Megjegyzés: A kvantitatív modell problémái:

a) A fizikai korlátok nincsenek beépítve;
b) Az összefüggéseket linearizáltuk;
c) A numerikus értékek nem pontosak és időben változnak.

Kvalitatív okoskodás (Qualitative Reasoning): Csak a mennyiségek irányultságát vesszük figyelembe, az értékkészlet: $\{-, 0, +\}$. Alapvető fizikai kényszereket betartunk!



Ha egy csomóponti elágazásnál két ágon kifolyik az "anyag", akkor a harmadikon befolyik.

Egy "Q" mennyiség kvalitatív értéke egy "a"" értékre vonatkoztatva: $[Q]_a$

Egy "Q" mennyiség megváltozásának kvalitatív értéke a kvalitatív derivált: $[\delta Q]_a, [\delta^2 Q]_a, \ldots$

Műveletek: $(invert\ A)$: Megfordítja az előjelet.

$(vote\ A_1, A_2, \ldots, A_n)$: Értéke a többségi előjel.

Kvalitatív modellezés és szabályozás

A 2. tartály szintjének kvalitatív szabályozása: L_2 jelöli a második tartály szinthibáját.

$[L_2]=+$: magasabb, mint kellene.
$[L_2]=0$: megegyezik.
$[L_2]=-$: alacsonyabb, mint kellene.

$[\delta U]=+$: a szivattyúzás mértéke növelendő.
$[\delta U]=0$: a szivattyúzás mértéke megfelelő.
$[\delta U]=-$: a szivattyúzás mértéke csökkentendő.

A kvalitatív értékek csak a mintavételi időpontokban léteznek.
A mintavételi időpontok között nincsen detektálás.

$[\delta U]=+$ Rögzített értékű növekmény: ΔU

$$[L_2]_{(k)} = [aktuális\ szint_{(k)} - megkívánt\ szint_{(k)}]$$

Egy igen egyszerű szabályzó:

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

Ha ΔU nagyobb érték, akkor nő a túllövés és az oszcilláció, de gyors.

Ha ΔU kisebb érték, akkor csökken a túllövés és az oszcilláció, lassúbb a működés.

Javított szabályzók:

Figyelembe vett mennyiségek:

A 2. tartály szintjének hibája: +,0,-
A 2. tartály szintváltozási sebessége: +,0,-
Az első tartály szintváltozási sebessége: +,0,-

3*3*3=27 eset

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

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

Kvalitatív modellezés és szabályozás

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

alapján méréssel.

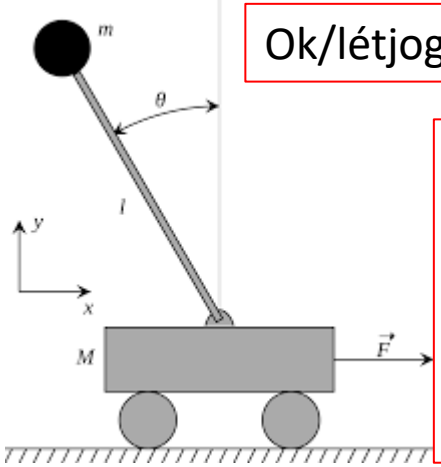
	$[L_2]$	$[\delta L_2]$	$[\delta L_1]$	$Q1$	$Q2$	$Q3$
1	+	+	+	-	-	-
2	+	+	0	-	-	-
3	+	+	-	-	0	-
4	+	0	+	-	-	-
5	+	0	0	-	-	-
…						
20	-	+	0	+	0	0
…						
27	-	-	-	+	+	+

Megjegyzés: (1) A feladatra empirikusan kidolgozott szabályrendszer nem tudta kezelni a "A 2. tartály a kívánt szint felett állandó értéket mutat, az 1. tartály szintje esik."
(2) A mintavételezési idő és a *ΔU* érték megválasztása kritikus tervezői döntés.

Kvalitatív modellezés és szabályozás

Példa: Kvalitatív modellezés nem determinisztikus automatával.

Olyan rendszerek esetében, amikor az $x(k)$ állapotvektorról csak egy $[x(k)]$ kvantált érték ismert.



Ok/létjogosultság: szög és szögsebesség mérés pontatlansága

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

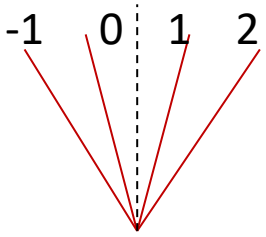
$$x(t) = \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix}$$

$$u(t) = F$$

Linearizált modell $\theta = 0$ környezetében. $M = 1kg, m = 0.1kg, l = 0.5m, g = 9.81\frac{m}{s^2}$

A mérési érzékenység: $0.0175\ rad\ a\ \theta - \ ra, és\ 0.0175/20ms\ \ a\ \dot{\theta} - ra.$

Nem stabilizálható a rúd, ha $|x_3| > 0.21rad\ (12°)$ $|x_4| > 0.87$



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

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

Kvalitatív modellezés és szabályozás

A kvalitatív állapotok:

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

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

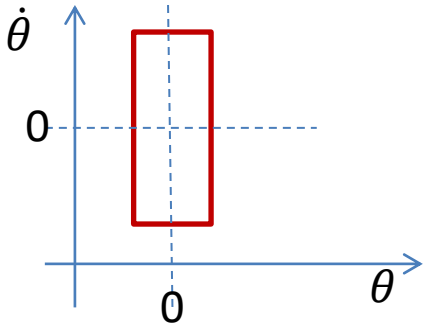
A bemenőjel (“rántás”) kvalitatív értékei:

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

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

Kvalitatív szabályzó: $[u(k)] = f([x(k)])$

Megjegyzés: A mintavételezési idő és az *F* érték megválasztása kritikus tervezői döntés.



Fuzzy modellezés és szabályozás

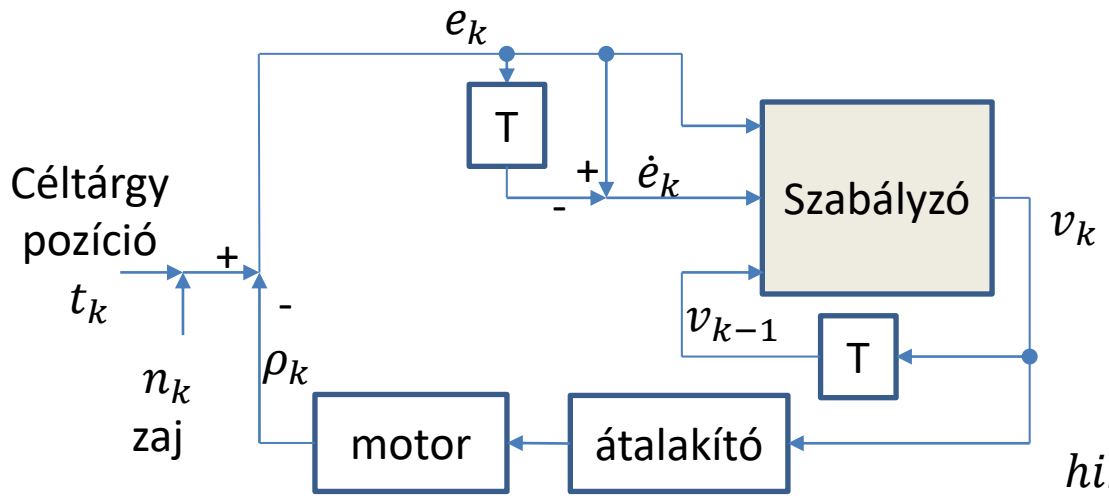
Példa: Adaptív célkövető rendszer:

egy azimut (0 … 180 fok) és egy emelkedés (0 … 90 fok) irányú forgató mechanizmussal.

Azimut: az a szög, amit a kijelölt irány vízszintes vetülete a déli vagy északi iránnyal bezár.

Szenzor: minden olyan eszköz alkalmas, amely kellő pontossággal képes a célra mutatni.

Laser, videokamera, nagy nyereségű antenna.



Jelölések:

t_k	céltárgy pozíció
n_k	megfigyelési zaj
ρ_k	célkövető pozíció
e_k	követési hiba
$\dot{e}_k$	követési hiba változás
v_k	becsült szögsebesség
T	mintavételi idő

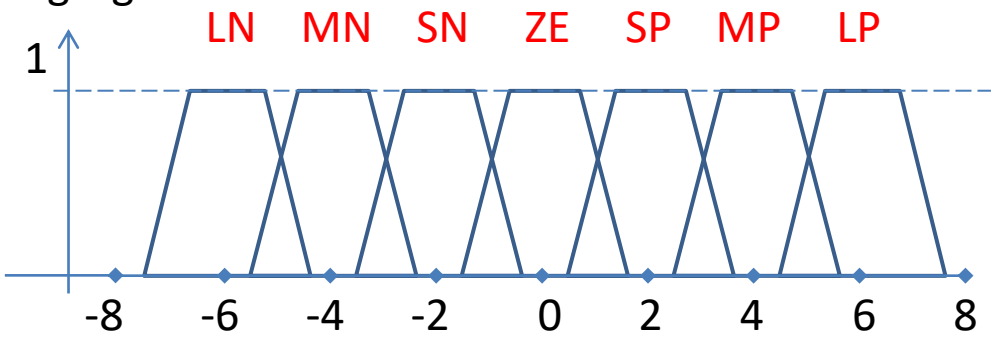
$$\rho_k = \rho_{k-1} + T v_{k-1} + hiba$$

$hiba =$ pozicionálási bizonytalanság

Fuzzy szabályozó: A becsült szögsebesség tartománya: [-6,6]. Mivel $|v_k| \le \frac{9.0}{T} fok/sec$ azimut irányban, $|v_k| \le \frac{4.5}{T} fok/sec$ emelkedés irányban, ezért az egyes csatornák erősítései: $\frac{1.5}{T}$ és $\frac{0.75}{T}$.

Fuzzy modellezés és szabályozás

Fuzzy szabályozó: Heurisztikus szintállító szabályokat tartalmaz az $e_k, \dot{e}_k$ és v_{k-1} értékei alapján. 7 fuzzy szint értéket definiálunk:



Minden bemenethez egy hételemű vektort rendelünk!

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

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)

Fuzzy-asszociatív-memoria (FAM) szabályok:

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

Az i-edik FAM szabály skalár értéke: $w_i = \min$(tagsági értékek).

Példa:

$e_k = 2.6$

$\dot{e}_k = -2.0$

$v_{k-1} = 1.8$

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

$m(\ldots)$: tagsági értékek

$m_{MP}(e_k) = 0.4$

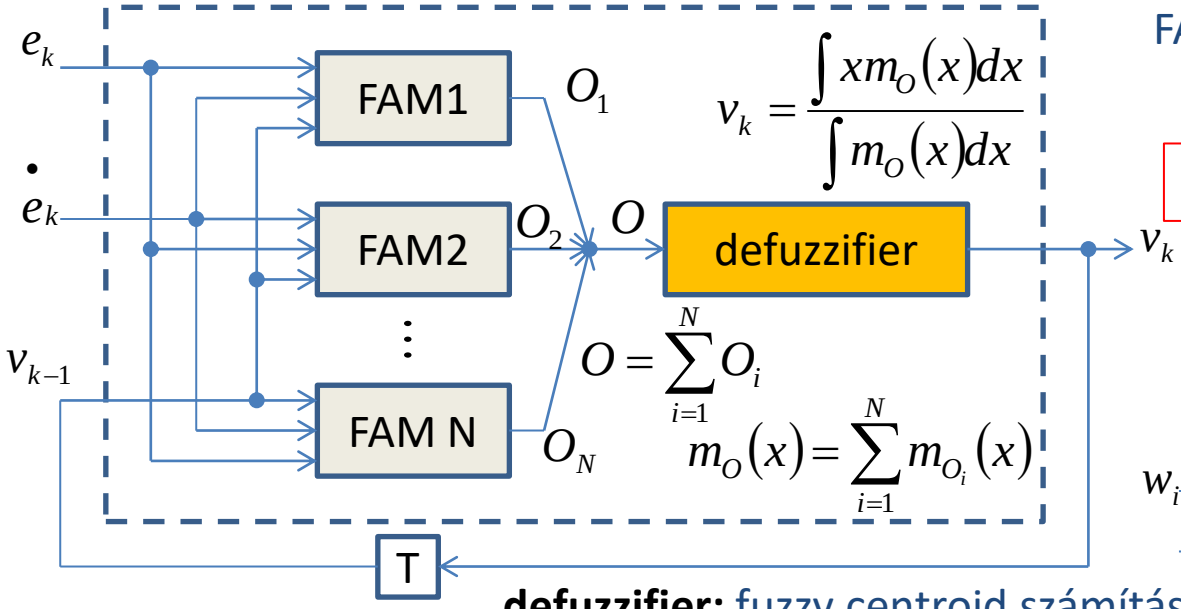
$m_{SN}(\dot{e}_k) = 1$

$m_{ZE}(v_{k-1}) = 0.1$

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

Fuzzy modellezés és szabályozás

A szabályozó kialakítása:



defuzzifier: fuzzy centroid számítás

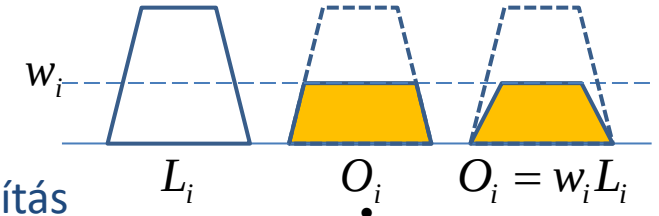
A kimeneti fuzzy halmaz alakja a FAM szabály kódolásától függ:

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

Korreláció-minimum kódolás

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

Korreláció-szorzat kódolás



A fuzzy szabályzó implementációja: FAMi szabály: (MP,SN,ZE;SP)

$e_k = 2.6 \quad \dot{e}_k = -2.0 \quad v_{k-1} = 1.8$

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

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

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

Minden fuzzy halmaz alakja azonos:

Pl.: $m_{SP}(x) = m_{ZE}(x-2).$

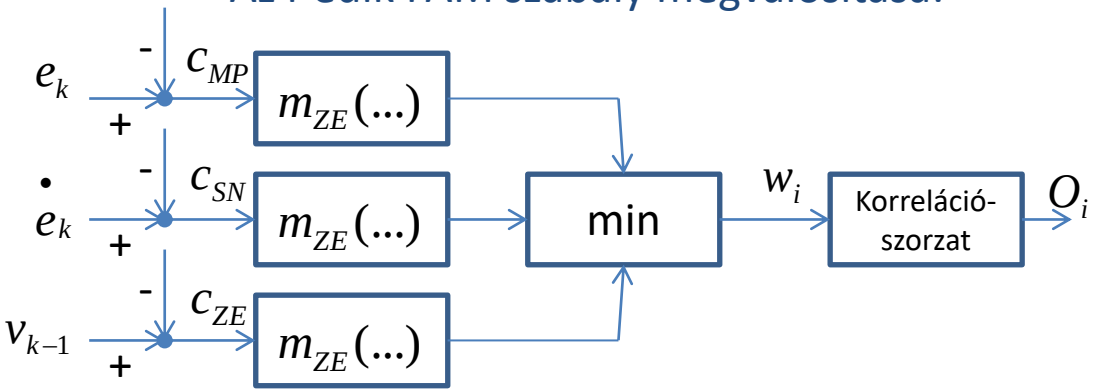
Általában: $m_{L_i}(x) = m_{ZE}(x - c_{L_i})$

c_{L_i} a tagsági függvény centroidja.

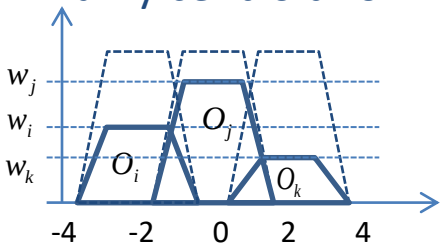
$$m_{O_i}(x) = w_i m_{ZE}(x - c_i)$$

Fuzzy modellezés és szabályozás

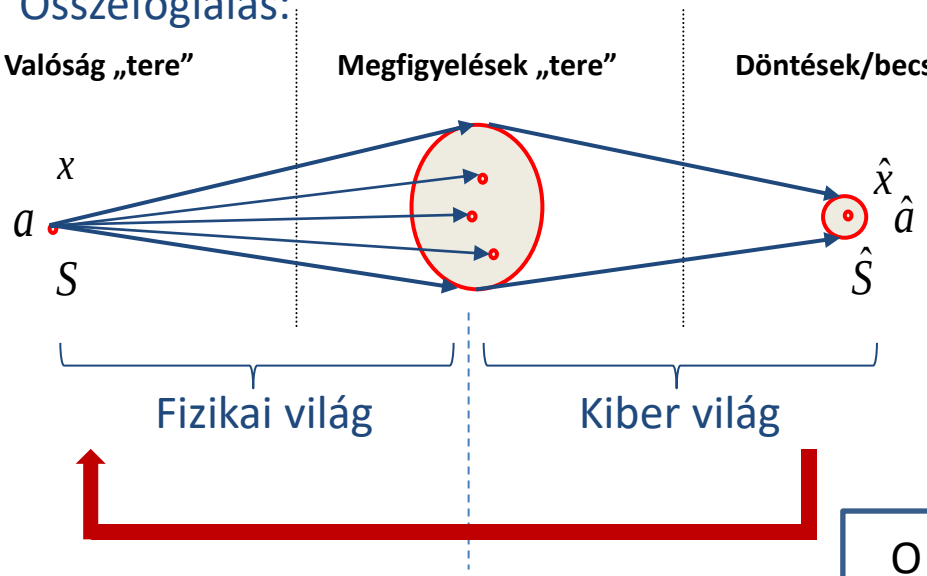
Az i-edik FAM szabály megvalósítása:

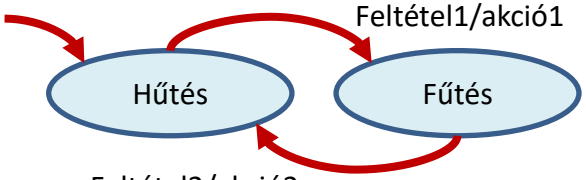


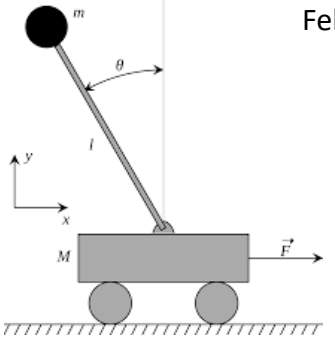
A fuzzy centroidhoz:

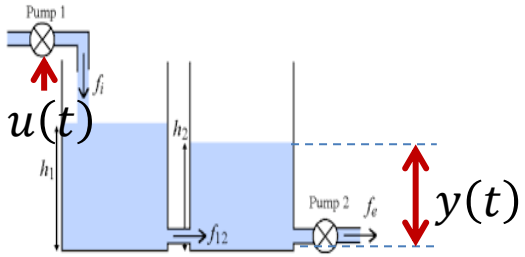


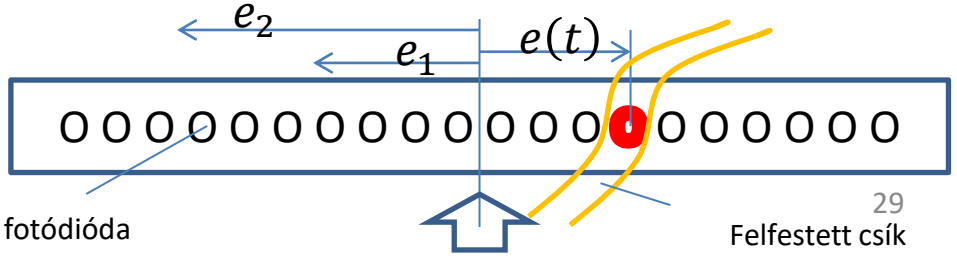
Összefoglalás:



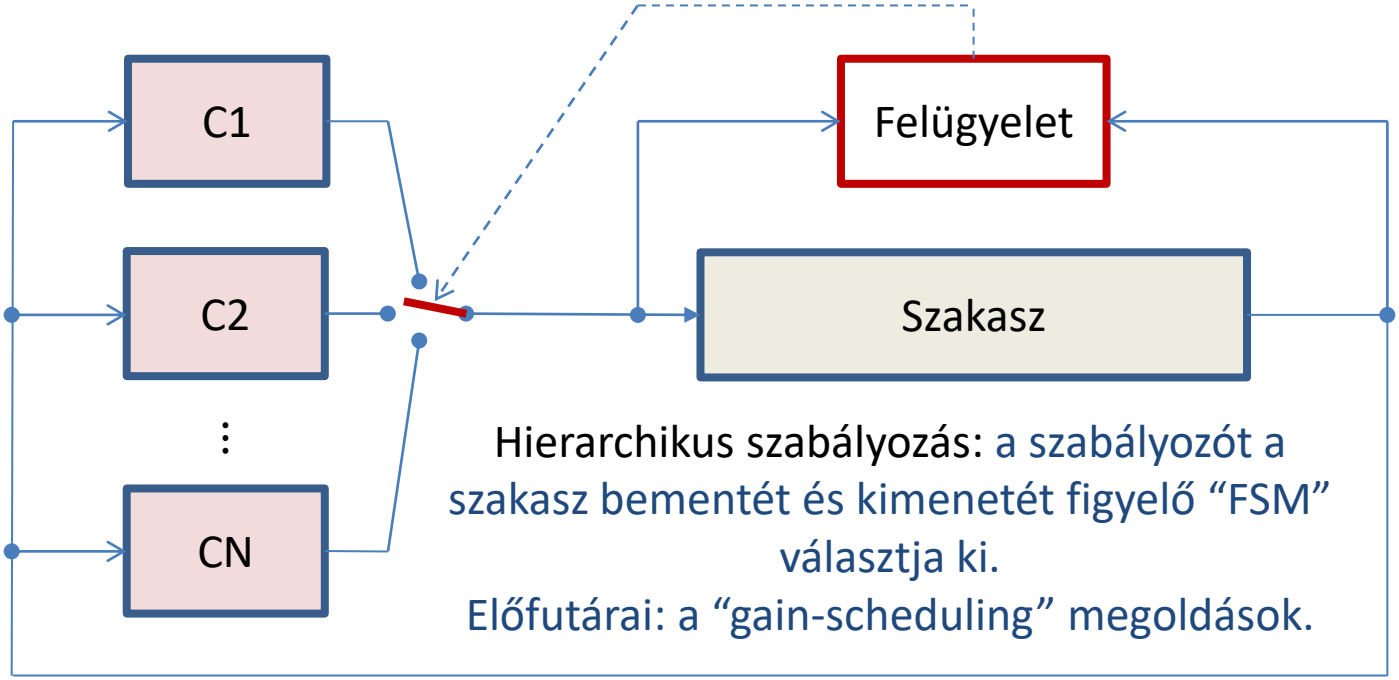








Hibrid rendszerek: Switching control



Kihívások: (1) Kapcsolgatott rendszer stabilitása (Hibrid rendszerek stabilitása!)
(2) Az átkapcsolások tranziensei.

Törekvések: (1) Stabilitás biztosítása a “passzivitás” fenntartásával.
(2) Tranziens menedzsment megvalósítása.

$$Z_{be}(s) = R + \frac{1}{sC}$$