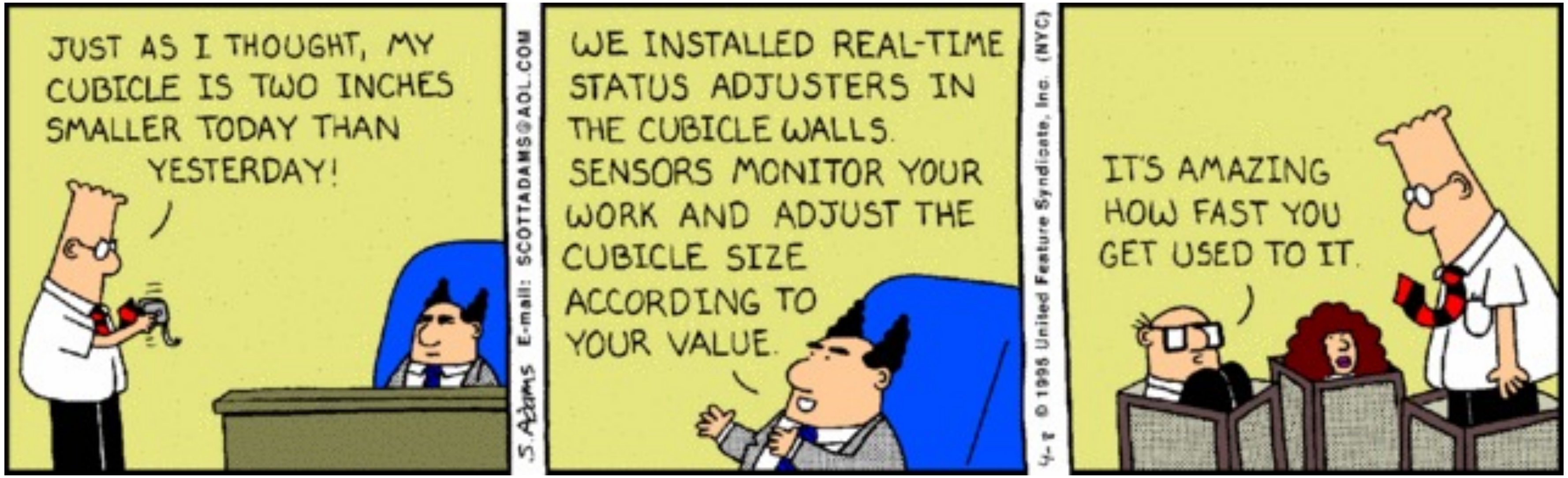
Sisteme Încorporate

Cursul 7
Sisteme de control

Facultatea de Automatică și Calculatoare
Universitatea Politehnica București



http://dilbert.com/strips/comic/1995-04-08/

- Un sistem încorporat este un sistem dinamic
  - Sistemul acționează în funcție de stimulii primiți
  - Atunci când una sau mai multe ieșiri ale sistemului trebuie să se conformeze anumitor reguli, un controller manipulează intrarea sistemului pentru a aduce ieșirile la valorile dorite.
  - Sistemul poate fi afectat de perturbații exterioare
  - Parametrii de intrare trebuie calculați în funcție de erori și de parametrii de ieșire.

- Exemple:
  - Termostatarea unei incinte
  - Controlul vitezei de rotație a unui hard-disk
  - Controlul altitudinii de zbor
  - Cruise control/Traction control
  - Surse de alimentare
  - Controlul mișcării pentru roboții industriali

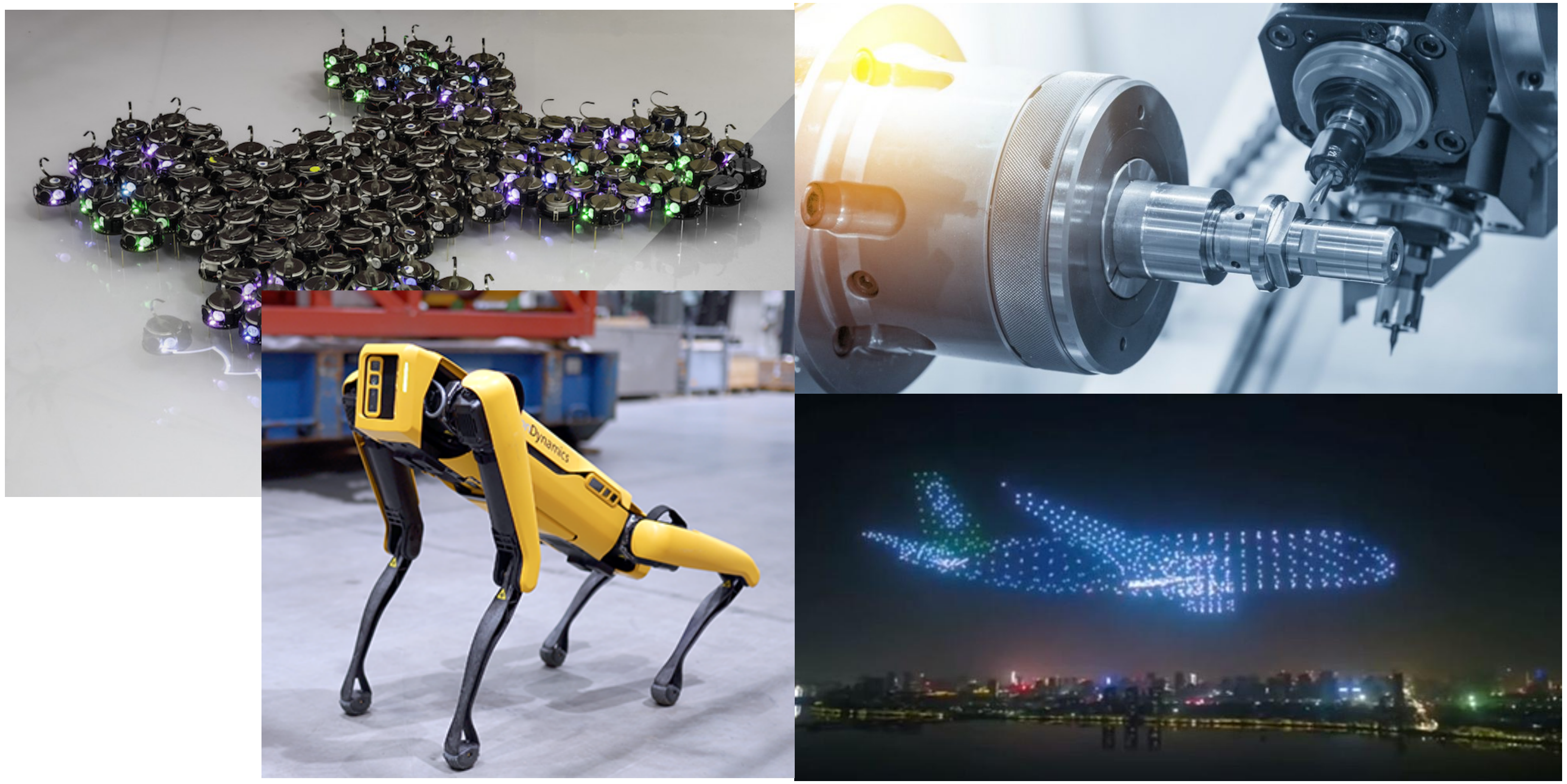
Controlul Sistemelor

- Aplicarea unor valori la intrarea sistemului pentru care ieșirea acestuia se conformează unei valori de referință.

  – Cruise-control: f_motor(t)=? → viteza=60 km/h
  – Server E-commerce: Alocarea resurselor? → T_răspuns = 5 sec
  – Networking: Rata de transfer? → Întârziere = 1 sec

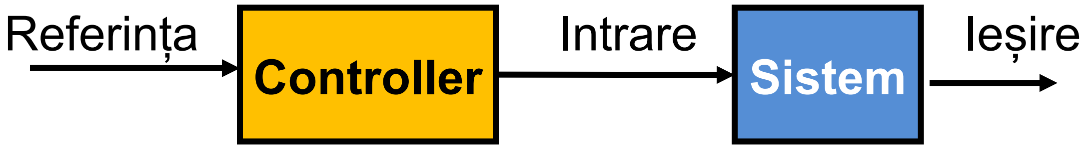
Ce este teoria controlului?

- Sistem = Un obiect sau ansamblu care se schimbă cu timpul
- Control = Modalitate de a influența schimbarea respectivă
- Exemple:
  - **Roboți**
  - Epidemii
  - Bursa de valori
  - Termostate
  - Circuite
  - Motoare
  - Rețele electrice
  - Sisteme autopilot

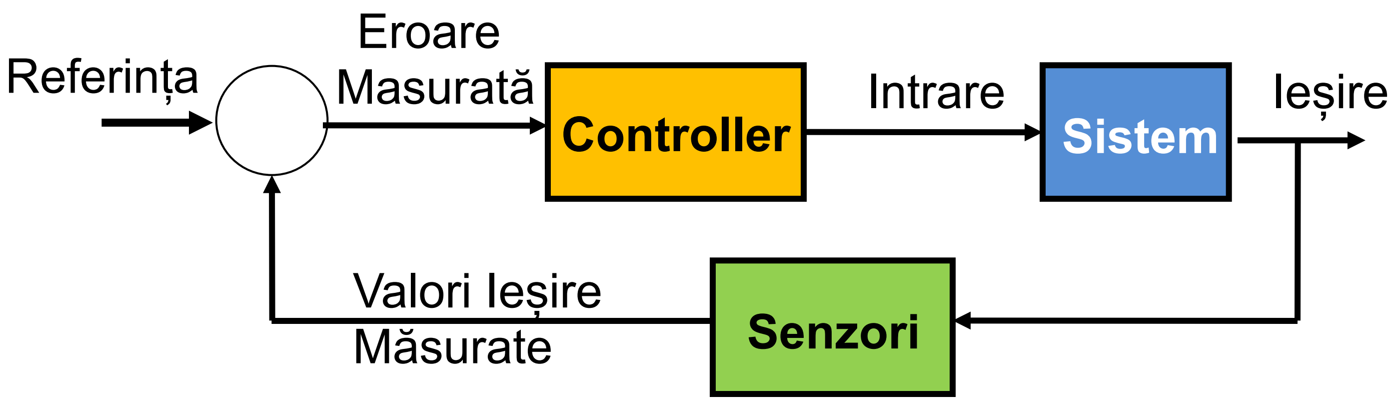


Tipuri de sisteme de control

- Sisteme în buclă deschisă



- Sisteme cu reacție



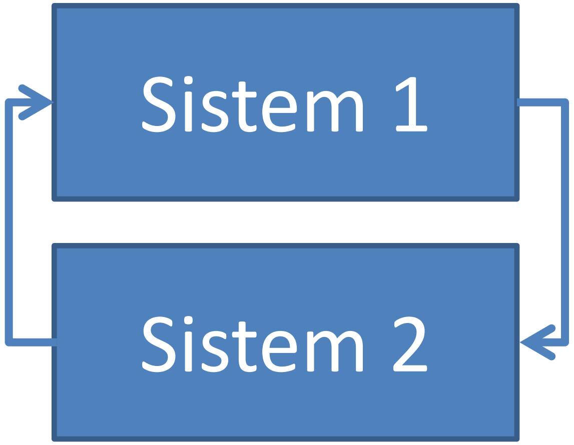
Sisteme în buclă deschisă

- Calculează valorile de intrare fără a măsura variabilele de sistem
  - Simplu de implementat
  - Trebuie cunoscute exact TOATE VARIABILELE din sistem ca totul să meargă cum trebuie
    - Cruise-control: frecare(t), unghi_plan(t)
    - Server E-commerce: Încărcarea(rata de sosire a cererilor? Consumul de resurse?); sistem (timp de service? Defecțiuni?)

- Sistemele în buclă deschisă dau greș atunci când
  - Nu știm totul
  - Facem erori de modelare
  - Lucrurile se schimbă

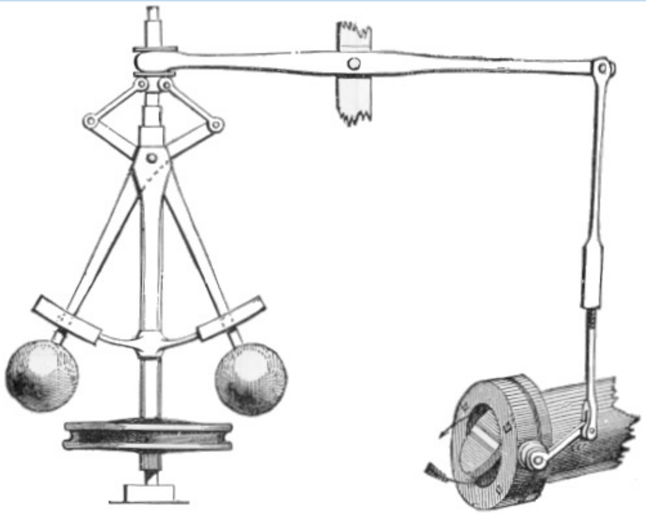
Sistemele cu reacție

- Ce este reacția?
  - Întoarcerea unei părți din ieșirea unui sistem la intrarea acestuia în scopul auto-corectării.
  - Presupune interconectarea mutuală a două sau mai multe sisteme
  - Relația cauză – efect e dificil de stabilit. Sisteme interdependente
  - Feedback-ul este prezent oriunde în sistemele naturale și artificiale



Exemplu #1 – Regulator de viteză

- “Flyball governor” (1788)
  - Reglează viteza unui motor cu aburi
  - Reduce efectele variației de sarcină (rejecția perturbațiilor)
  - Produce accelerarea revoluției industriale



Greutățile se indepartează odată cu creșterea vitezei de rotație

Supapa se închide, micșorând turația motorului

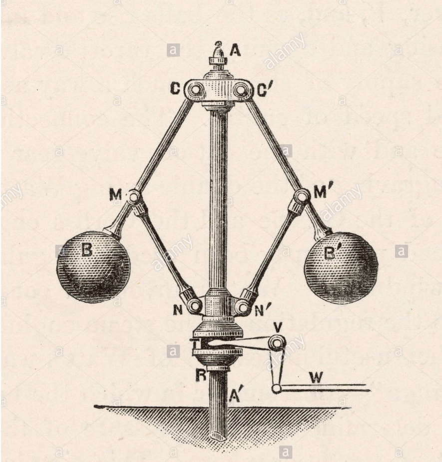
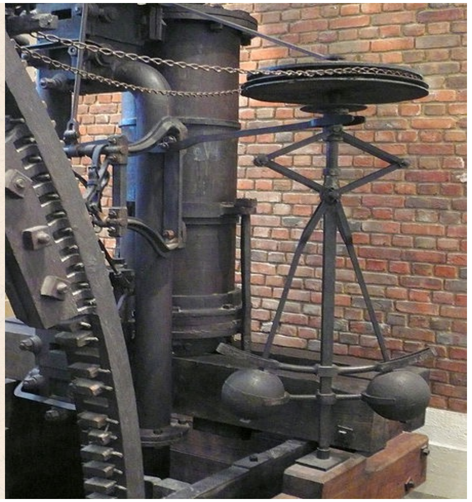
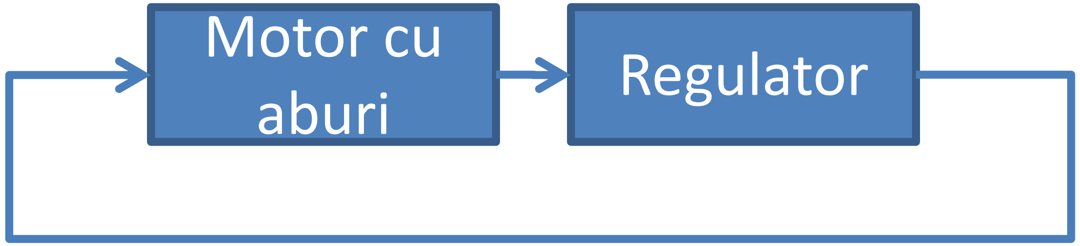


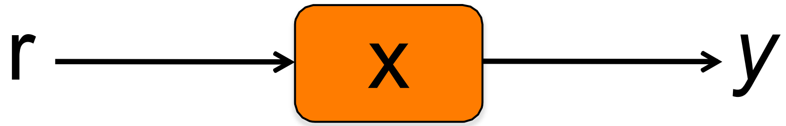
FIG. 29.—The Governor.





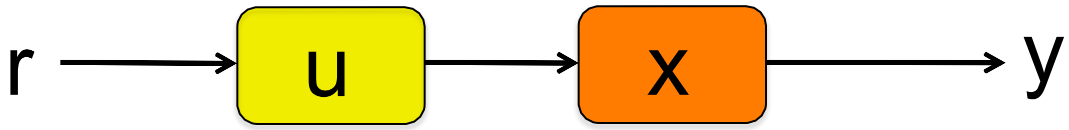
Elemente de bază

- ***Stare*** = Reprezentare a sistemului la un moment dat de timp
- ***Dinamică*** = Descriere a cum se schimbă starea sistemului
- ***Referință*** = Ce vrem ca sistemul să facă
- ***Ieșire*** = Măsură a (anumitor aspecte ale) sistemului



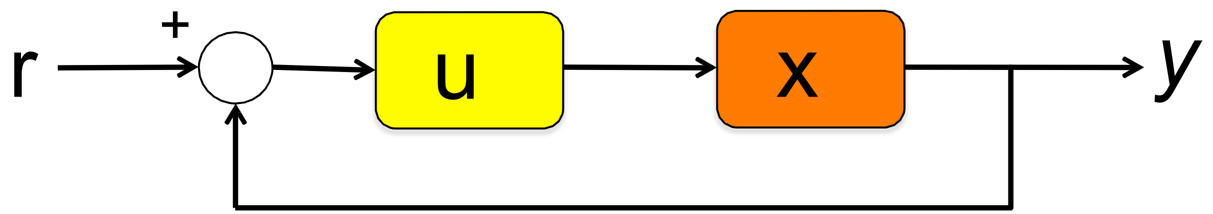
Elemente de bază

- ***Stare*** = Reprezentare a sistemului la un moment dat de timp
- ***Dinamică*** = Descriere a cum se schimbă starea sistemului
- ***Referință*** = Ce vrem ca sistemul să facă
- ***Ieșire*** = Măsură a (anumitor aspecte ale) sistemului
- ***Intrare*** = Semnal de control



Elemente de bază

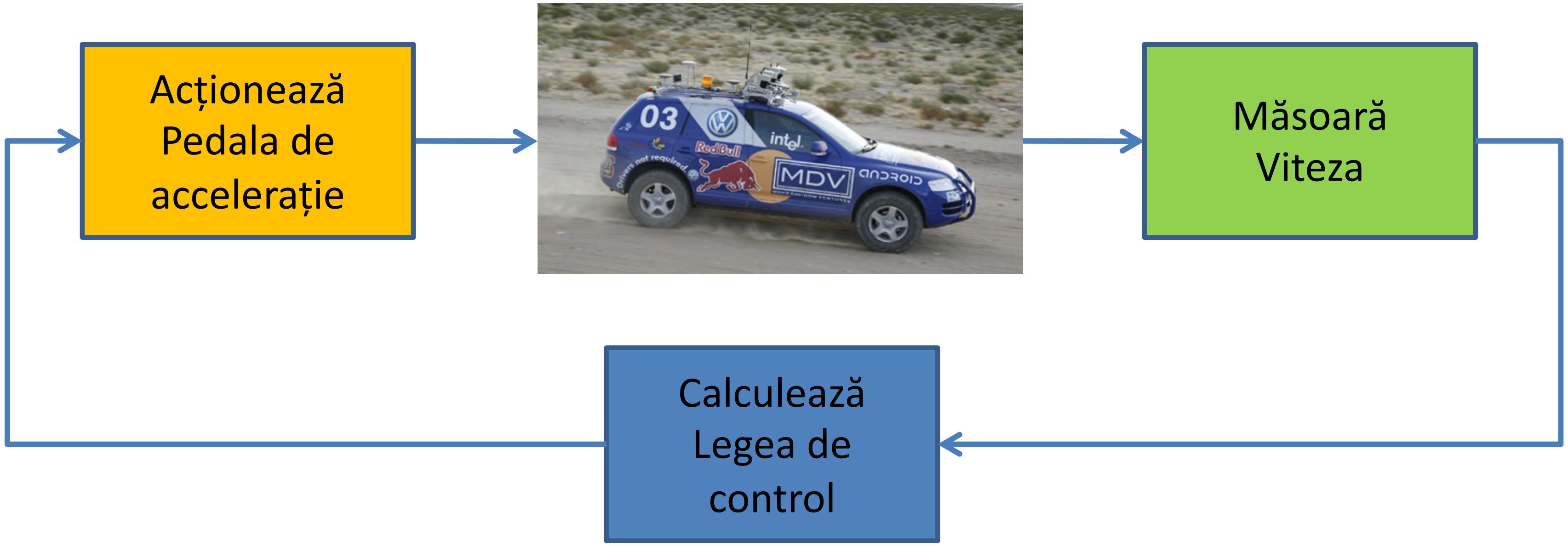
- ***Stare*** = Reprezentare a sistemului la un moment dat de timp
- ***Dinamică*** = Descriere a cum se schimbă starea sistemului
- ***Referință*** = Ce vrem ca sistemul să facă
- ***Ieșire*** = Măsură a (anumitor aspecte ale) sistemului
- ***Intrare*** = Semnal de control
- ***Feedback*** = Mapare a ieșirii la intrare



Înapoi la exemplele noastre

- **Roboți**
- Epidemii
- Bursa de valori
- Termostate
- Circuite
- Motoare
- Rețele electrice
- Sisteme autopilot

Control = Senzori + Calcule + Efectoare

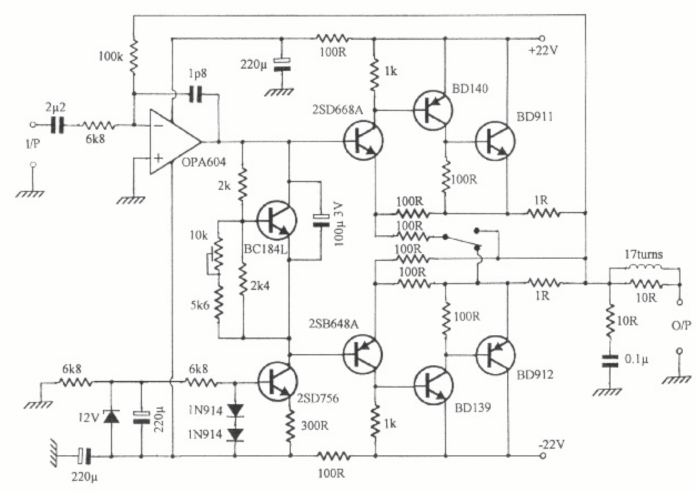


Țeluri:

1. Stabilitate: Sistemul își menține starea de-a lungul timpului (rulează cu viteză constantă)
2. Performanța: Sistemul reacționează rapid la schimbări (accelerează de la 0 la 100 km/h)
3. Robustețe: Sistemul tolerează perturbațiile (masă, frecare, unghiul pantei etc.)

Cele două principii de control

- Robustețe la nesiguranță prin feedback
  - Reacția asigură performanțe mărite chiar și la variațiile neprevizibile ale variabilelor de sistem
  - Amplificatoare care funcționează corect chiar dacă valorile componentelor variază
  - Ideea de bază: măsurarea cu acuratețe a diferenței dintre comportamentul obținut și cel dorit; corectare prin calcul și efectori.
- Designul comportamentului dinamic prin feedback
  - Reacția permite modificarea caracteristicilor dinamice ale unui sistem
  - Exemplu: Îmbunătățirea manevrabilității pentru avioanele instabile
  - Ideea de bază: Interdependența modifică comportamentul normal

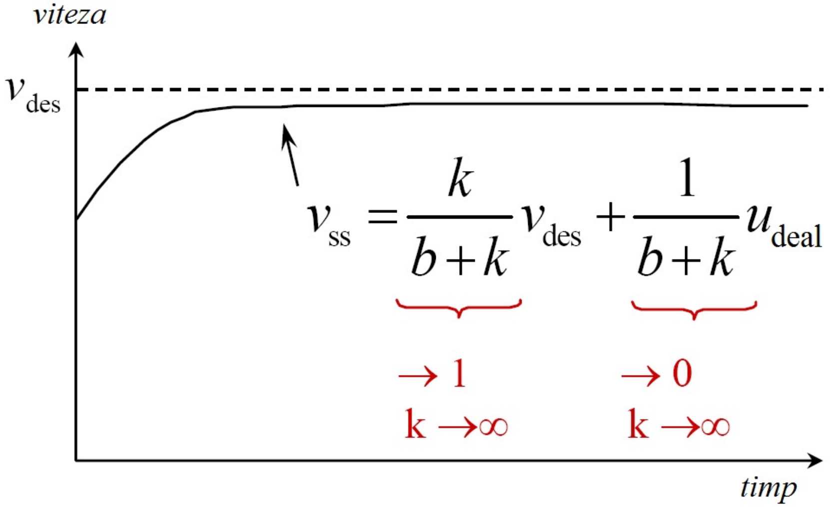


Dryden Flight Research Center EC85 33297-23 Photographed 1985
X-29

Exemplul #2 – Cruise Control

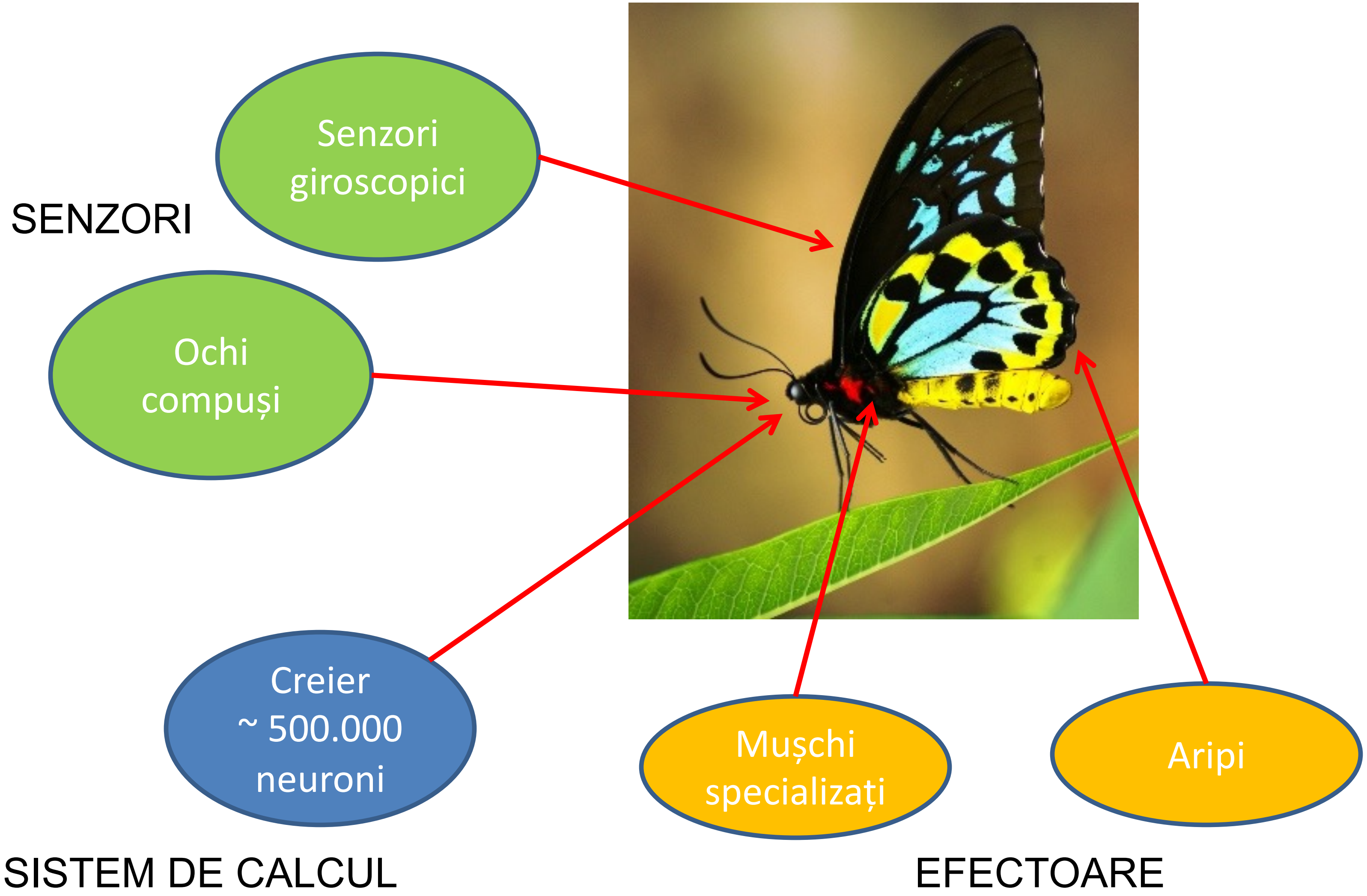
$$m\dot{v} = -bv + u_{motor} + u_{drum}$$

$$u_{motor} = k(v_{des} - v)$$



- Stabilitate/performanță
  - Viteza steady state (Vss) se apropie de viteza dorită pentru k-> ∞
  - Răspuns lin, fără depășire sau oscilații
- Rejectia perturbațiilor
  - Efectele perturbațiilor (dealurile) sunt eliminate când k-> ∞
- Robustețe
  - Rezultatele nu depind de valorile factorilor m, b, k pentru k suficient de mare.

Exemplul #3 – Zborul unei insecte



Aplicațiile moderne ale sistemelor cu reacție

Embedded Systems Laboratory

- Sisteme de zbor
  - Fly-by-wire
  - UAV
- Robotică
  - Determinarea exactă a poziției pentru operații precise
  - Medii greu accesibile: spațiu, mare, operații non-invazive
- Procese chimice
  - Reglarea temperaturii, vitezei de reacție, dozarea reactanților
- Comunicații și rețelistică
  - Amplificatoare și repetoare de semnal
  - Power management pentru comunicațiile wireless
- Automobile
  - Controlul motorului, tracțiunii, climatizării, stabilității etc.

Și multe altele….

Nevoia de a avea un model

- Teoria controlului = Cum să găsim semnalul de intrare *u*?
- Obiective:
  - Stabilitate
  - Tracking
  - Robustețe
  - Respingerea perturbațiilor
  - Optimalitate

Modele dinamice

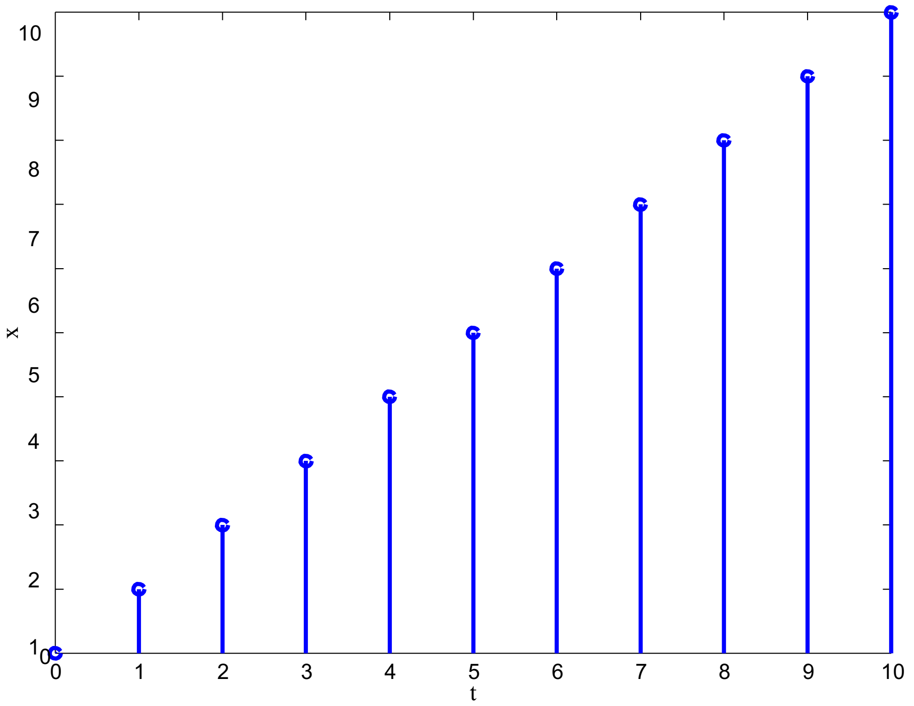
- Strategiile eficiente de control se bazează pe modele predictive
- Timp discret:

$$x_{k+1} = f(x_k, u_k) \longleftarrow \text{Ecuație a diferențelor}$$

Exemplu: Ceas

$$x_{k+1} = x_k + 1$$

Ceas de timp discret



Dinamică = schimbare cu timpul

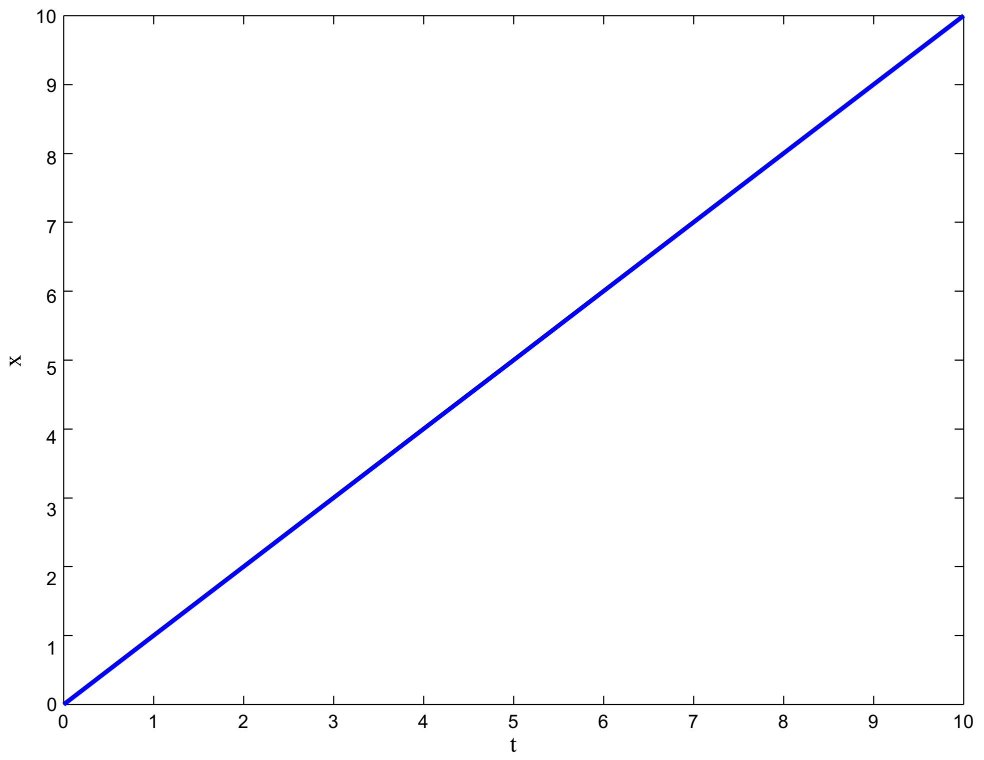
- Legile fizicii (și Natura) sunt scrise pentru timp continuu: În loc de starea următoare, avem nevoie de variația în raport cu timpul a mărimii
- Timp continuu:

$$\frac{dx}{dt} = f(x, u) \quad \sim \quad x^{\cdot} = f(x, u) \longleftarrow \text{Ecuație diferențială}$$

- Ceas:

$$\dot{x} = 1$$

Ceas de timp continuu



De la continuu la discret

- Dar, într-o implementare totul este discret/eșantionat!

$$\dot{x} = f\,(x, u)$$

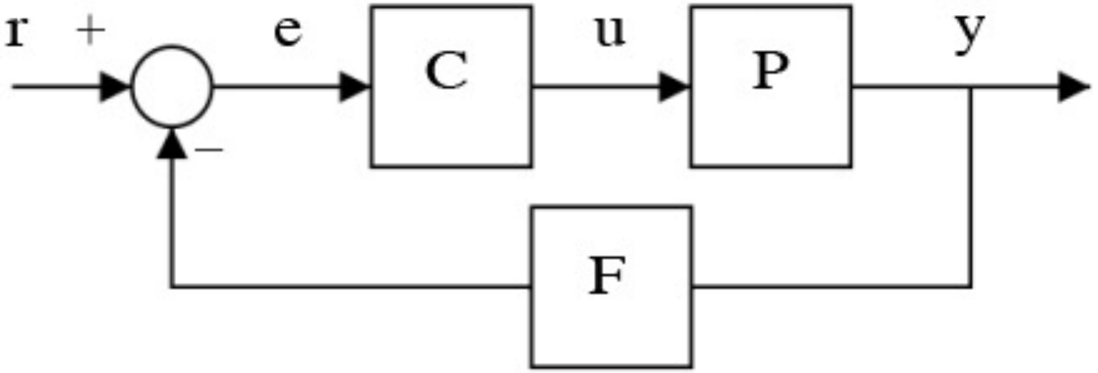
- Timpul de eșantionare: δt

$$x_k = x(k\delta t) \quad \Rightarrow \quad x_{k+1} = x((k+1)\delta t) = ???$$

$$x(k\delta t + \delta t) \approx x(k\delta t) + \delta t \dot{x}\,(k\delta t)$$

$$\boxed{x_{k+1} = x_k + \delta t f\,(x_k\,,\, u_k\,)}$$

Funcția de transfer



$$Y(s) = P(s)U(s) \qquad U(s) = C(s)E(s) \qquad E(s) = R(s) - F(s)Y(s).$$

$$Y(s) = \left(\frac{P(s)C(s)}{1 + F(s)P(s)C(s)}\right) R(s) = H(s)R(s).$$

$$H(s) = \frac{P(s)C(s)}{1 + F(s)P(s)C(s)}$$ Funcția de transfer a sistemului

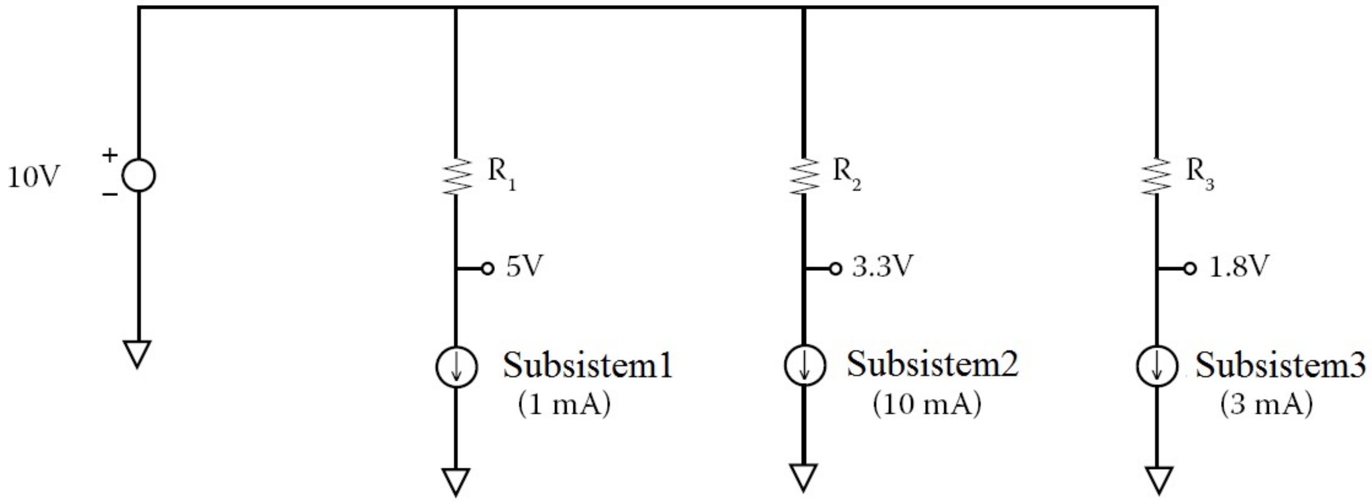
Exemplu: Sursă de tensiune

- Cerințe:
- Vreau să proiectez o sursă de tensiune care
  - să furnizeze o tensiune fixă indiferent de variația tensiunii de intrare sau a sarcinii de la ieșire
  - sa aibă o eficiență cât mai mare

Power Supply ver. 0.1

http://www.dilbert.com/strips/comic/1999-02-17/

Soluția 1: Divizor Rezistiv



- Puterea totală: P = 10V * 14mA = 140mW
- Puterea disipată în rezistențe:

  P_d = 5*1 + 6.9*10 + 8.2*3 = 96.6mW

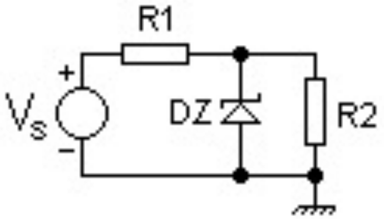
  70% din putere este pierdută în conversia de tensiune

  Este, probabil, cea mai proastă soluție

Soluția 2:

- Stabilizator cu diodă Zener
  - Poate furniza o tensiune stabilă numai pentru curenți mici
  - Tensiunea la ieșire poate varia odată cu variația sarcinii

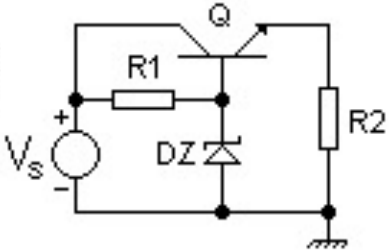
$$R1 = \frac{V_S - V_Z}{I_Z + I_{R2}}$$



- Ver. 2.1: Diodă Zener cu repetor
  - Adaugă un repetor pentru a micșora curentul prin dioda Zener
  - Potrivit pentru o gamă mai mare de variație a curentului prin sarcină

$$R1 = \frac{V_S - V_Z}{I_Z + K.I_B}$$

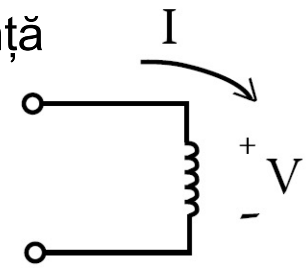
$$I_B = \frac{I_{R2}}{h_{FE(min)}}$$



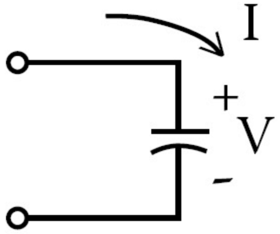
Alte soluții mai bune?

- Ce-ar fi să folosim componente fără pierderi?

Inductanță



Condensator



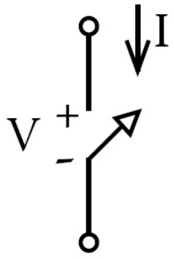
$$U = I_L Z = I_L j\omega L$$

$$U = I_C Z = I_L(-j\frac{1}{C\omega})$$

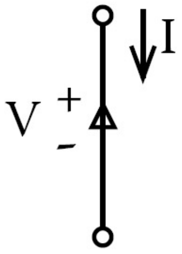
$$P_a = \mathrm{Re}\{S\} = \mathrm{Re}\{UI_L\} = \mathrm{Re}\{I_L^2 j\omega L\} = 0$$

$$P_a = \mathrm{Re}\{S\} = \mathrm{Re}\{UI_C\} = \mathrm{Re}\{I_C^2(-j\frac{1}{C\omega}\} = 0$$

Comutator



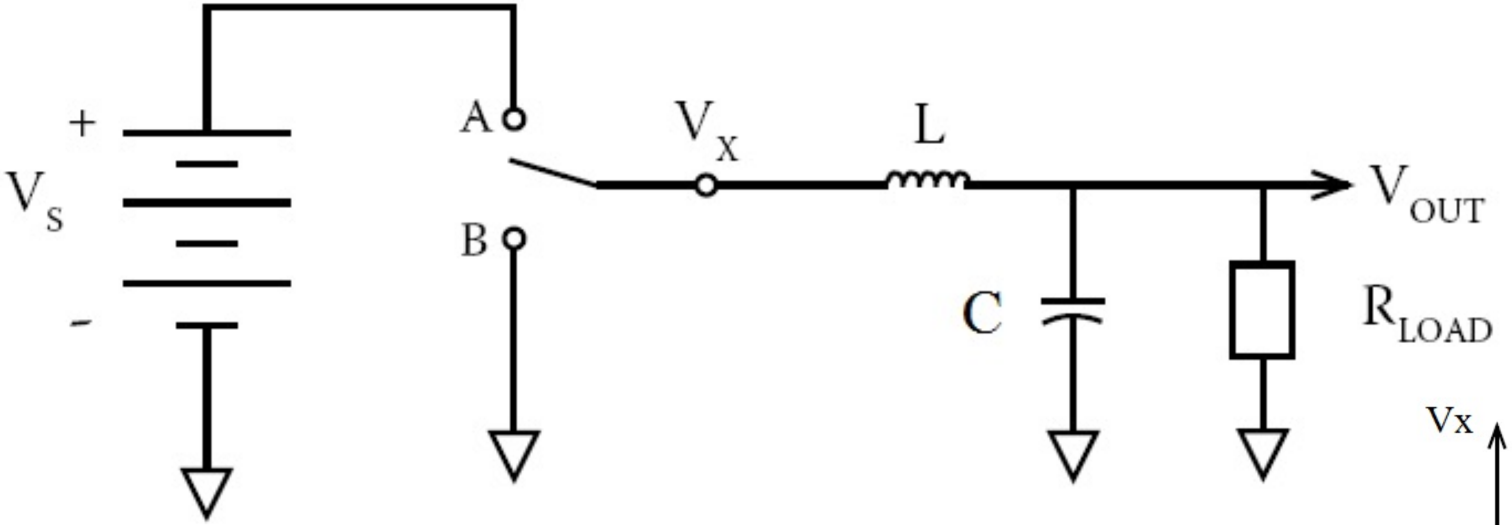
$$P = I \cdot U = 0 \cdot U = 0$$

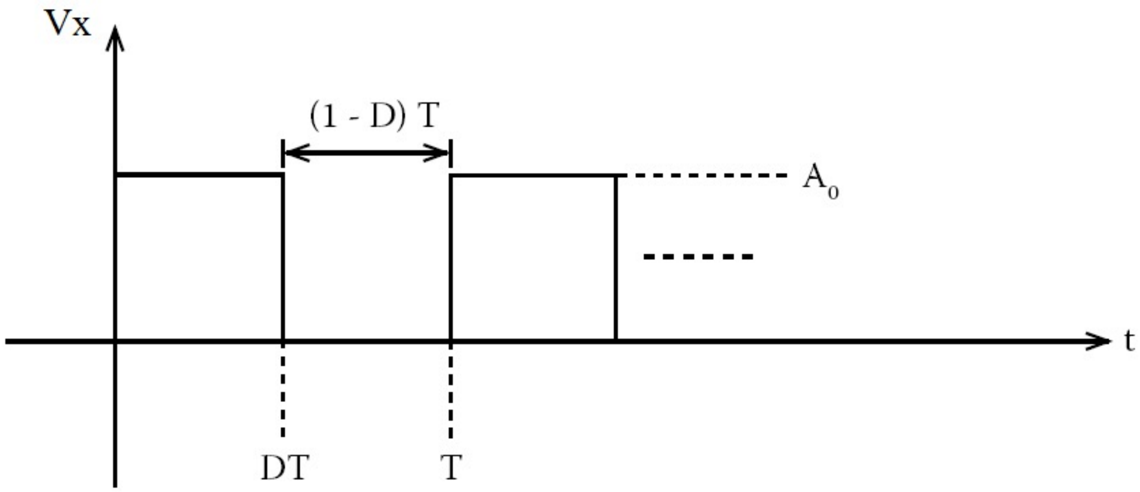


$$P = I \cdot U = I \cdot 0 = 0$$

Soluția 3: Sursă în comutație

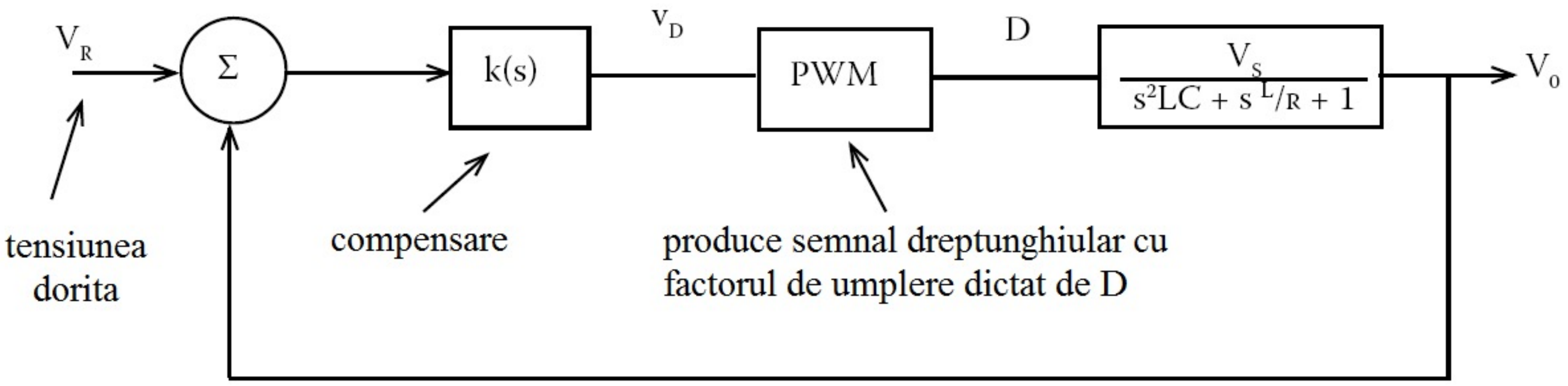
- Folosind condensatoare, bobine și comutatoare se poate face eficient conversia





$0 \le D \le 1$

Funcția de transfer

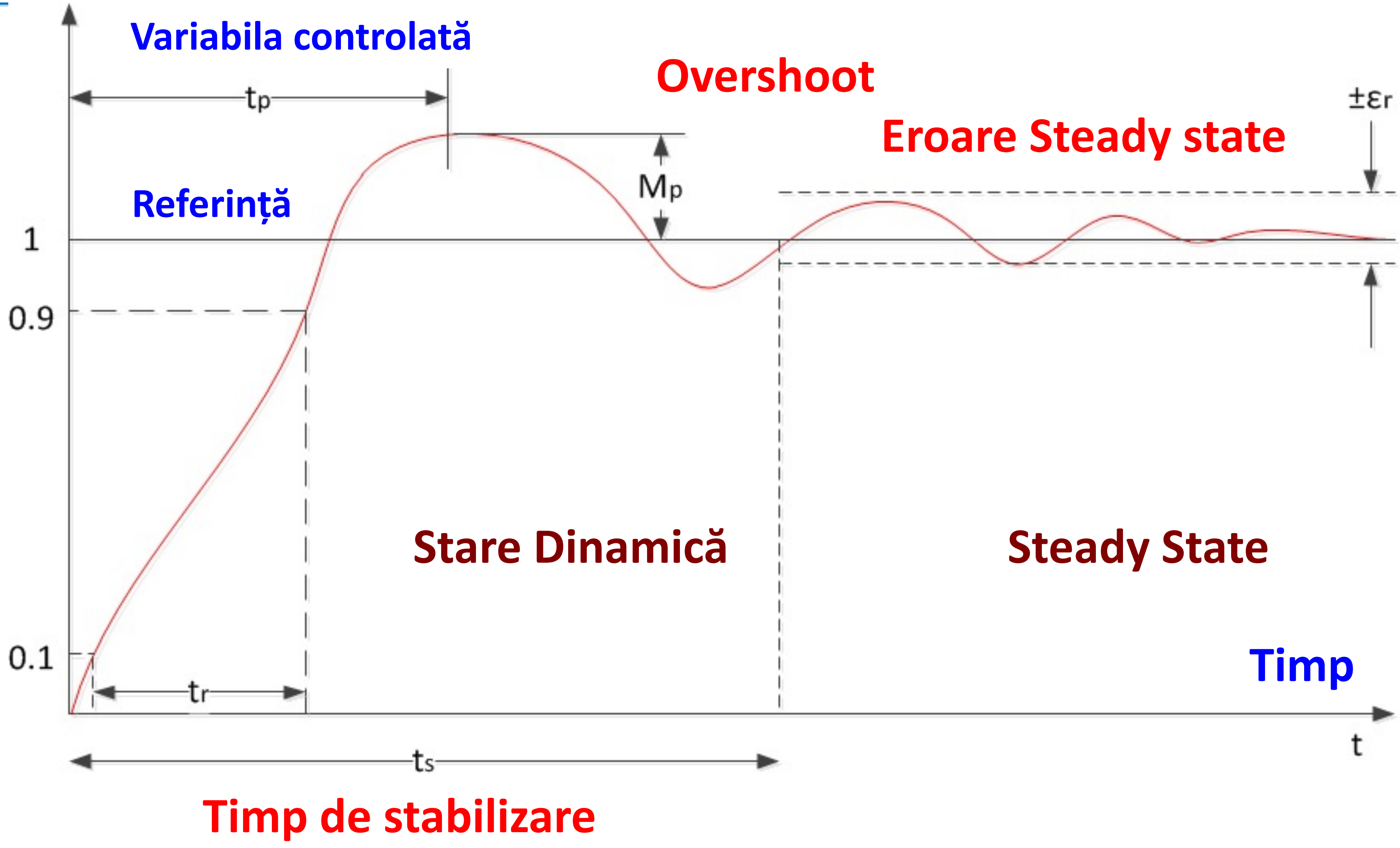


$$V_X = D \cdot V_S$$

$$V_O = \frac{1}{s^2LC + s\frac{L}{R} + 1} V_X$$

$$V_O = \frac{D}{s^2LC + s\frac{L}{R} + 1} V_S$$

Indicatori de Performanță



Proprietățile Regulatoarelor

- Stabilitate
- Ieșirea "urmărește" intrarea pentru orice tip de semnal de intrare
- Ex: în controlul proporțional, stabilitatea este estimată prin stabilirea dacă polii funcției de transfer au modulul mai mic ca 1

Proprietățile Regulatoarelor

- Acuratețe
- Acuratețea este dată de mărimea erorii de steady-state.
- Ex: în controlul proporțional, acuratețea este calculată în raport cu câștigul funcției de transfer pentru o valoare de referință a intrării. Eroarea este zero dacă și numai dacă acest câștig este 1.

Proprietățile Regulatoarelor

- Depășirea superioară (Overshoot)
- Este o proprietate a răspunsului la o excitație de tip treaptă a intrării.
- Depășirea superioară este caracteristică pentru un comportament oscilant al sistemului. O valoare mare pentru overshoot va fi urmată de obicei de o depășire inferioară (undershoot).

Tipuri de control

- Proporțional (P)
- Proporțional Integral (PI)
- Proporțional Diferențial (PD)
- PID

Cruise-Controller

- Face o mașină să meargă la o viteză dorită, de referință (*r*)
- A doua lege a lui Newton:
  $F = ma$
- Stare: viteza (*x*)
- Intrare: accelerează/frână (*u*)
  $F = cu$ (*c= coeficient transmisie electro-mecanică*)
- Dinamică:
  $$\dot{x} = a \Rightarrow m\dot{x} = cu \Rightarrow \dot{x} = \frac{c}{m} u$$

Proiectarea regulatoarelor

- Presupunem că măsurăm viteza (*y*=*x*)
- Sistemul de control trebuie să fie o funcție de *r*-*y* (=*e*)

- Ce proprietăți trebuie să aibă semnalul de control?
  - Un *e* mic va genera un *u* mic
  - *u* nu ar trebui să fie “zgâlțâit”
  - *u* nu trebuie să depindă de valorile *c* și *m* (nu putem oricum să le cunoaștem în toate cazurile)

Proiectarea regulatoarelor – elemente de bază

Embedded Systems Laboratory

- Modelul mașinii:

$$\dot{x} = \frac{c}{m} u$$

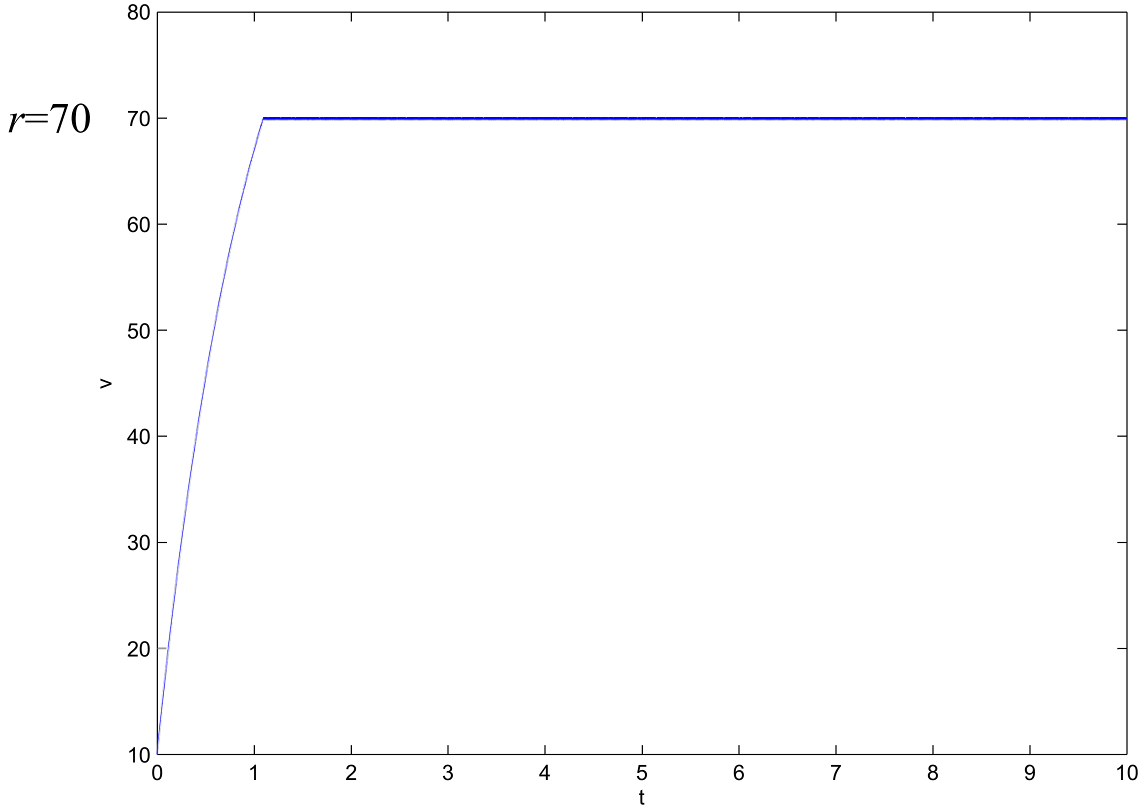
- Vrem să

$$x \rightarrow r \text{ când } t \rightarrow \infty \qquad (e = r - x \rightarrow 0)$$

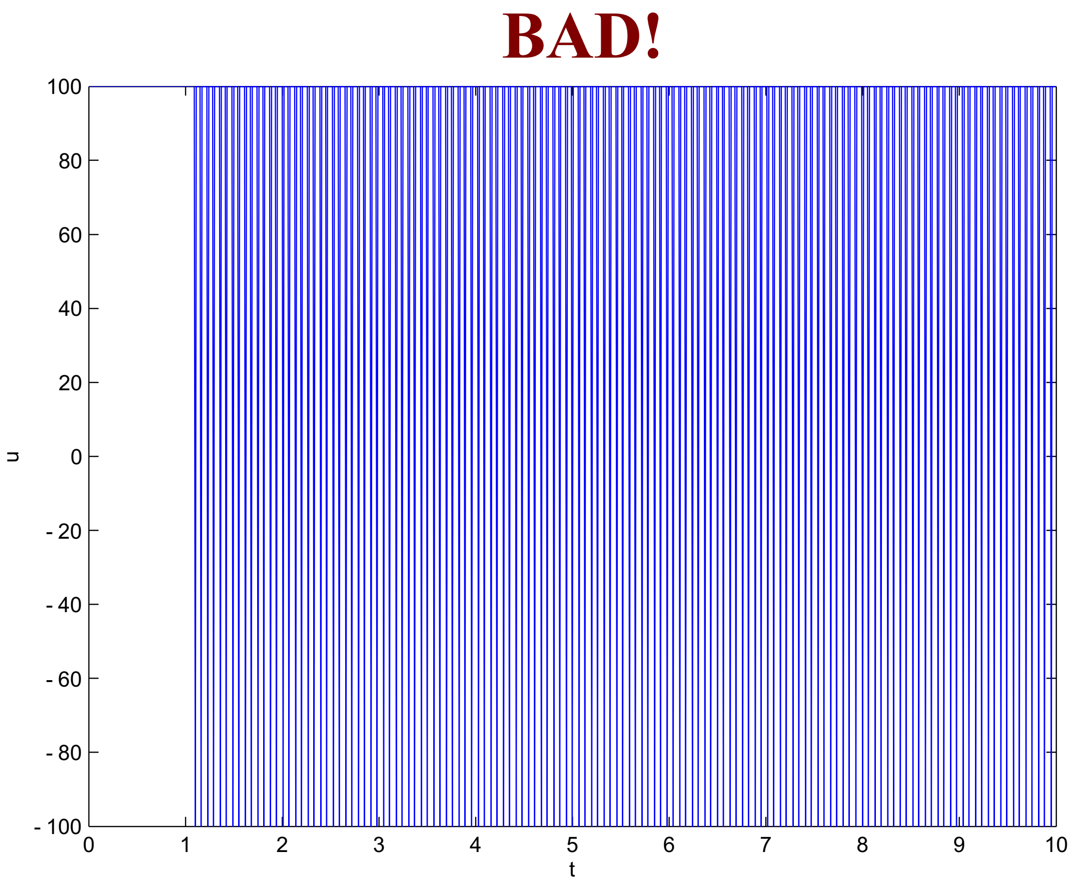
- *Încercarea 1*

$$u = \begin{cases} u_{max} & \text{if } e > 0 \\ -u_{max} & \text{if } e < 0 \\ 0 & \text{if } e = 0 \end{cases}$$

Control Bipozițional (Bang-Bang) - Teoretic



Bang-Bang Control - Realitatea



Bumpy ride

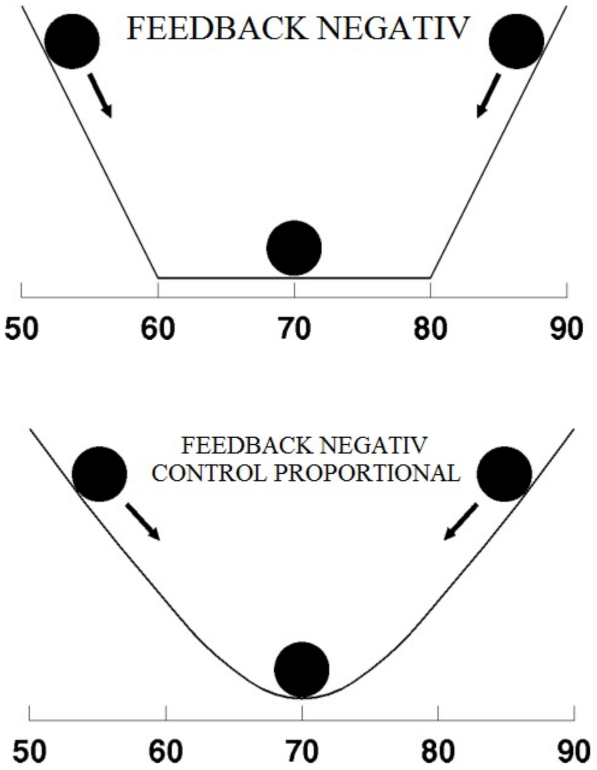
Burns out actuators

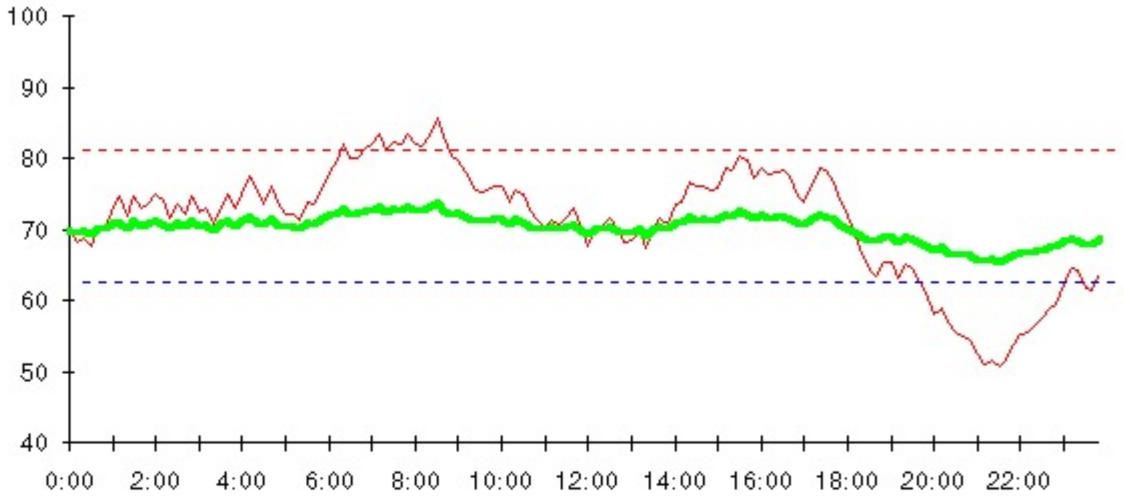
….

Bang-Bang Control – Proces lent

Putem avea un control mai precis?

- Controlul temperaturii unei incinte
  - Termostat cu 2 praguri: 14 si 20 grade C (bang-bang)
  - Regulator proporțional





Control P

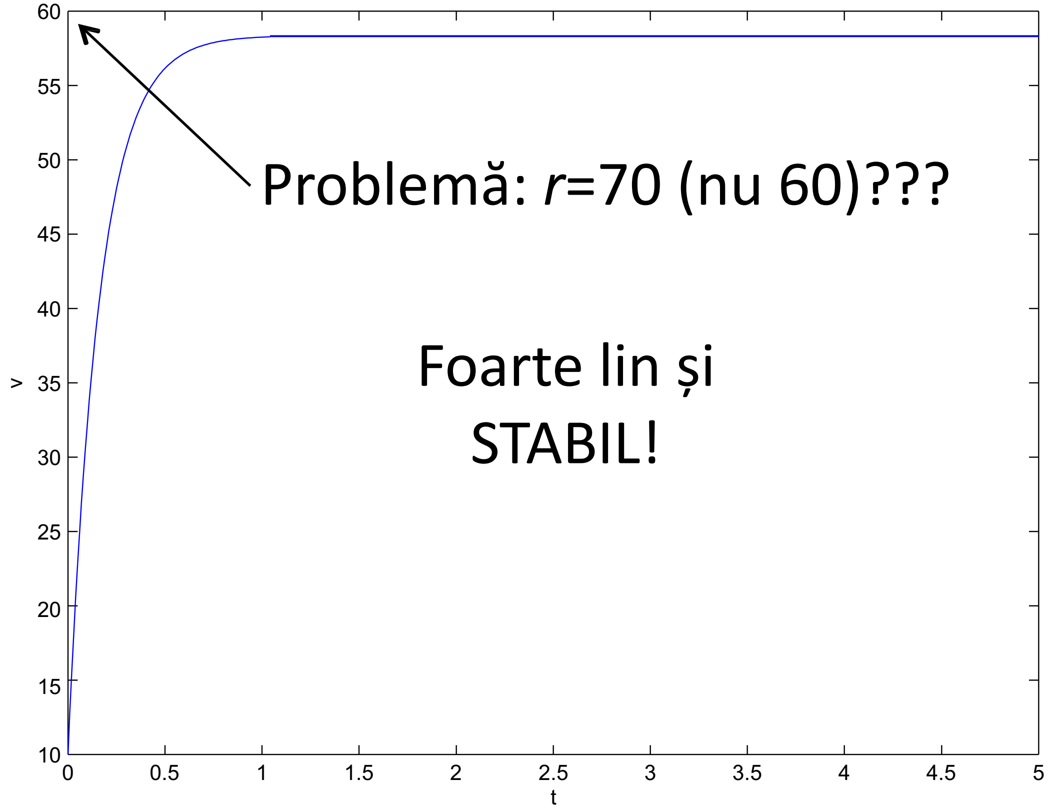
- Problemă: Regulatorul reacționează prea puternic la erori mici

- *Încercarea 2*

  $u = ke$

  - Erorile mici dau un semnal de control mic
  - Control foarte ”lin”
  - Numit și regulator proporțional (P-regulator)

Regulator P


Problemă: r=70 (nu 60)???
Foarte lin și
STABIL!
v
t
60
55
50
45
40
35
30
25
20
15
10
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5

Stabilitate dar nu și urmărire

- Deficiență: Modelul “real” este îmbunătățit pentru a include rezistența la înaintare:

$$\dot{x} = \frac{c}{m}u - \gamma x$$

- Acest model este folosit pentru a simula controller-ul
- La steady-state (x nu se mai schimbă)

$$\dot{x} = 0 = \frac{c}{m}u - \gamma x = \frac{c}{m}k(r - x) - \gamma x \Rightarrow (ck + m\gamma)x = ckr$$

$$\boxed{x = \frac{ck}{ck + m\gamma}r < r}$$

Obiective de performanță

Embedded Systems Laboratory

- Un regulator ar trebui să ofere:
  - Stabilitate (BIBO)
  - Tracking
  - Robustețe

$$\dot{x} = \frac{c}{m} u - \gamma x$$

Problemă: Dintr-odată trebuie să cunoaștem toți parametrii fizici pe care nu îi cunoaștem – *Nu este robust!*

- *Încercarea 3*

$$u = ke + \gamma \frac{m}{c} x$$

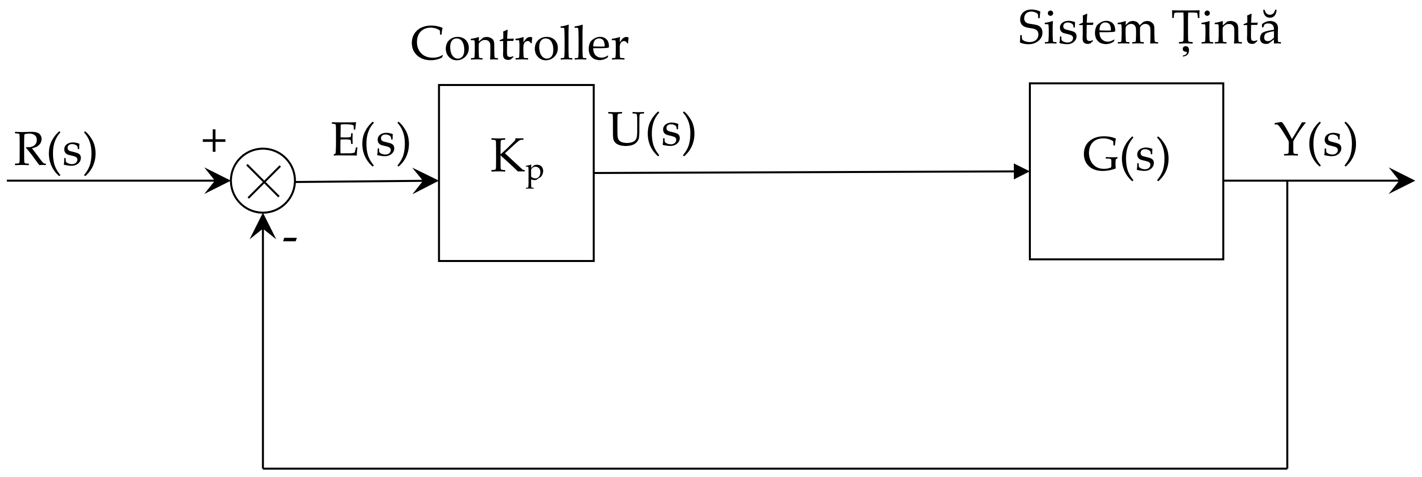
$$\dot{x} = 0 = \frac{c}{m} k(r - x) + \gamma x - \gamma x \quad \Rightarrow \quad x = r$$

Tracking!

Control Proporțional

- Una dintre cele mai des întâlnite bucle de control
- Ieșirea controllerului este direct proporțională cu eroarea
- Produce rezultate bune pentru majoritatea aplicațiilor unde parametrii de funcționare nu sunt critici (overshoot, stabilitate, eroare steady-state)

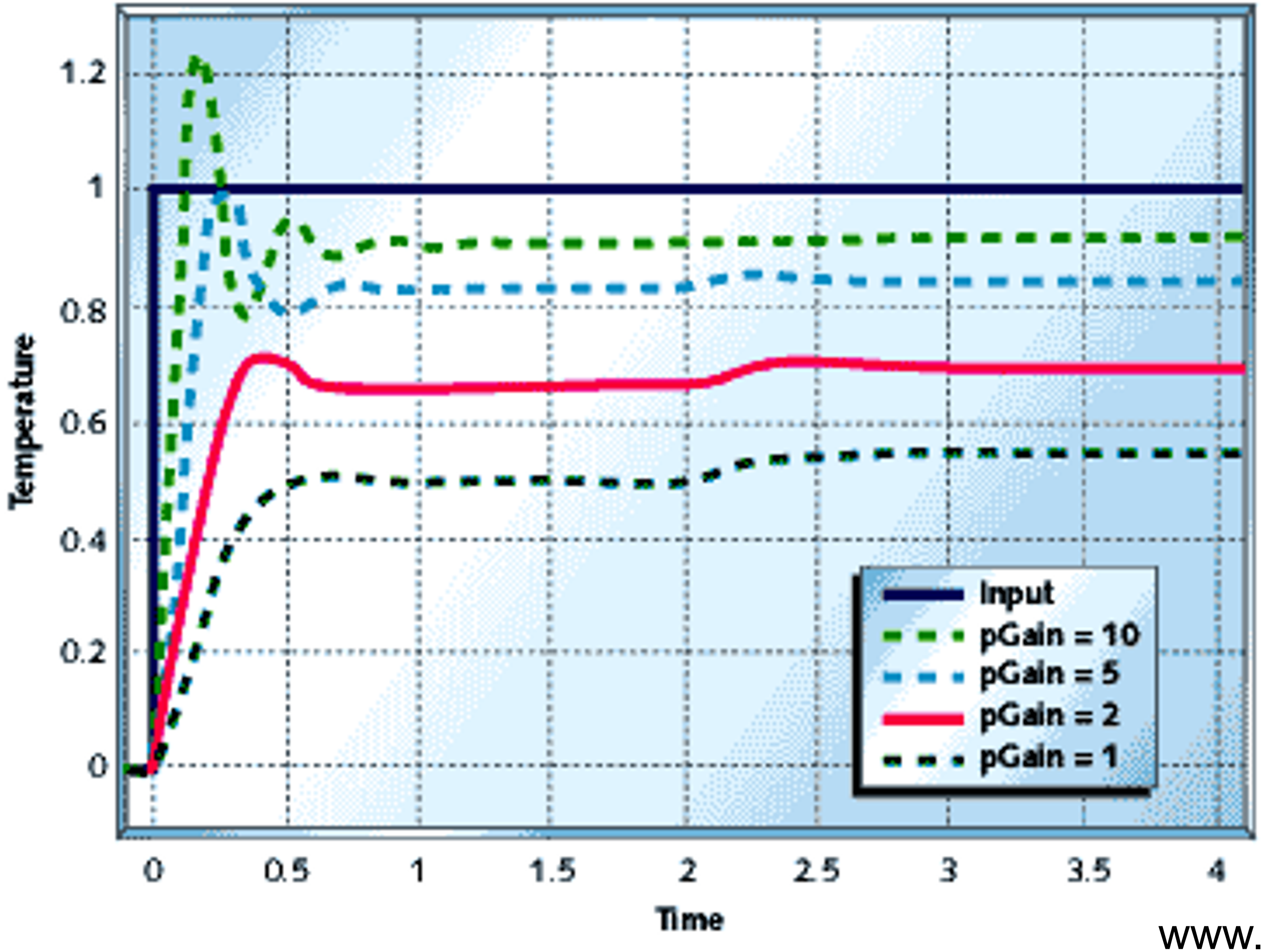
Control Proporțional



$$E(s) = R(s) - Y(s) \qquad U(s) = K_P E(s) \qquad Y(s) = G(s)U(s)$$

$$H(s) = \frac{K_P \cdot G(s)}{1 + K_P \cdot G(s)}$$

Răspunsul la semnal de tip treaptă



www.embedded.com

Caracterizarea Regulatorului P

- Ecuația caracteristică a sistemului:

$$1 + K_P \cdot G(s) = 0$$

- Când Kp -> 0, soluțiile următoarei ecuații sunt polii lui G(s):

$$\frac{1}{G(s)} + K_p = 0$$

- Când Kp->infinit, soluțiile următoarei ecuații sunt zerourile lui G(s):

$$\frac{1}{K_p} + G(s) = 0$$

Stabilitate

- Sistemul cu control proporțional este stabil dacă pentru o anumită valoare a lui Kp, toți polii sunt în interiorul cercului unitate

  (modul < 1).
- Orice sistem care are cel puțin un zero la infinit va deveni întotdeauna instabil pentru o valoare suficient de mare a lui Kp.

Acuratețe

$$e_{ss} = \lim_{k->\infty}(r(k) - y(k)) = r_{ss} - y_{ss}$$

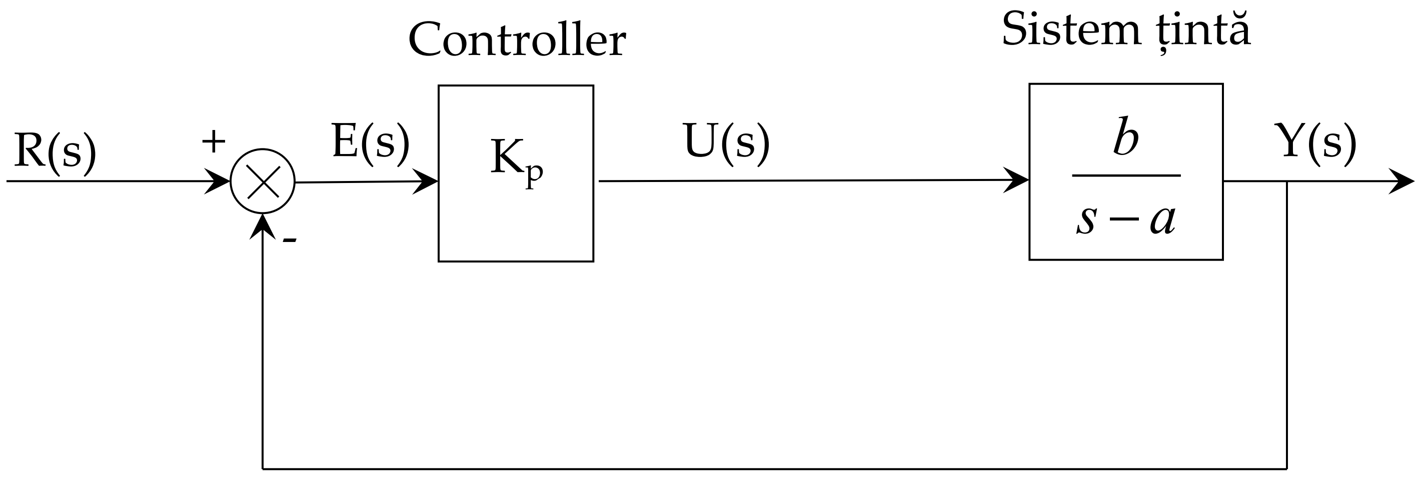
$$K_p = \frac{H(s)}{G(s)(1 - H(s))}$$

$$e_{ss} = \lim_{k->\infty} e(k) = \lim_{s->0} \frac{1}{1 + K_P G(s)}$$

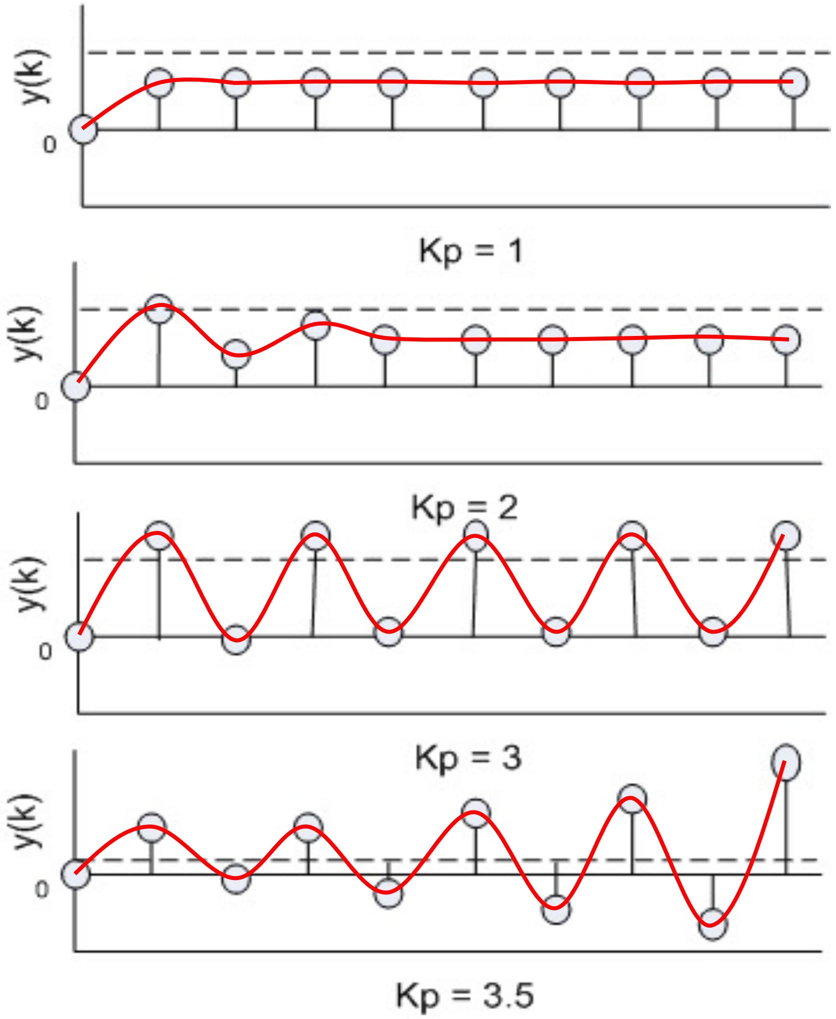
- Eroarea de stare stabilă este cu atât mai mică cu cât Kp este mai mare.

Acordarea K_p

- Ce valoare a lui K_p aleg pentru ca sistemul meu sa fie cât mai performant?
- Exemplu: controllerul de temperatură



Efecte pentru diferite valori ale Kp



$$G(s) = \frac{0.43}{s - 0.47}$$

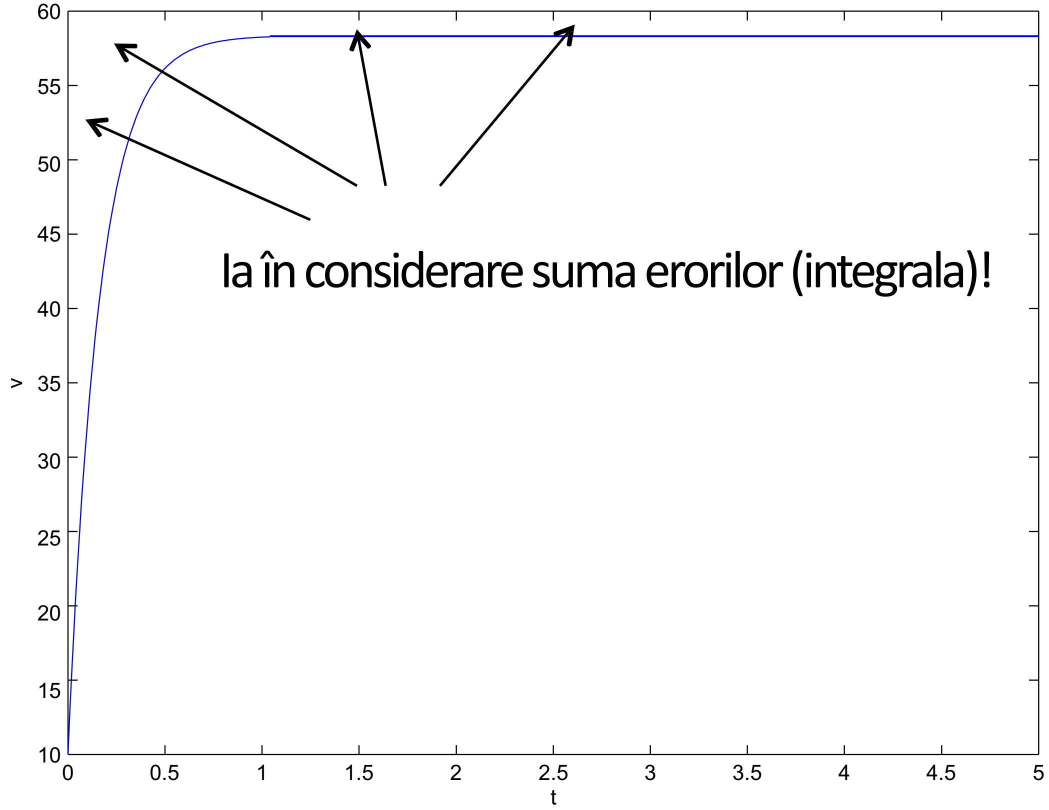
- Evoluția sistemului în timp în funcție de Kp
- K_P = 1, polul funcției de transfer este aproape 0, $|e_{ss}|$ este mare.
- K_P = 2, pol în -0.51, overshoot și oscilație.
- K_P = 3, pol la -1, oscilație.
- K_P = 3.5, pol la -1.25, sistemul este instabil.

Concluzii Control P

+ Este foarte ușor de implementat
+ Stabilitate bună dacă alegem K_p potrivit

- Timp destul de îndelungat pentru stabilizare
- Poate intra foarte ușor în oscilație
- Pentru valori din ce în ce mai mari ale K_p sistemul devine instabil
- În general, una sau mai multe cerințe (eroare de stare stabilă, depășire, oscilație) sunt imposibil de satisfăcut pentru orice valoare a K_p

Regulator P



60
55
50
45
40
35
30
25
20
15
10
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
t
v
Ia în considerare suma erorilor (integrala)!

Regulatoare PI

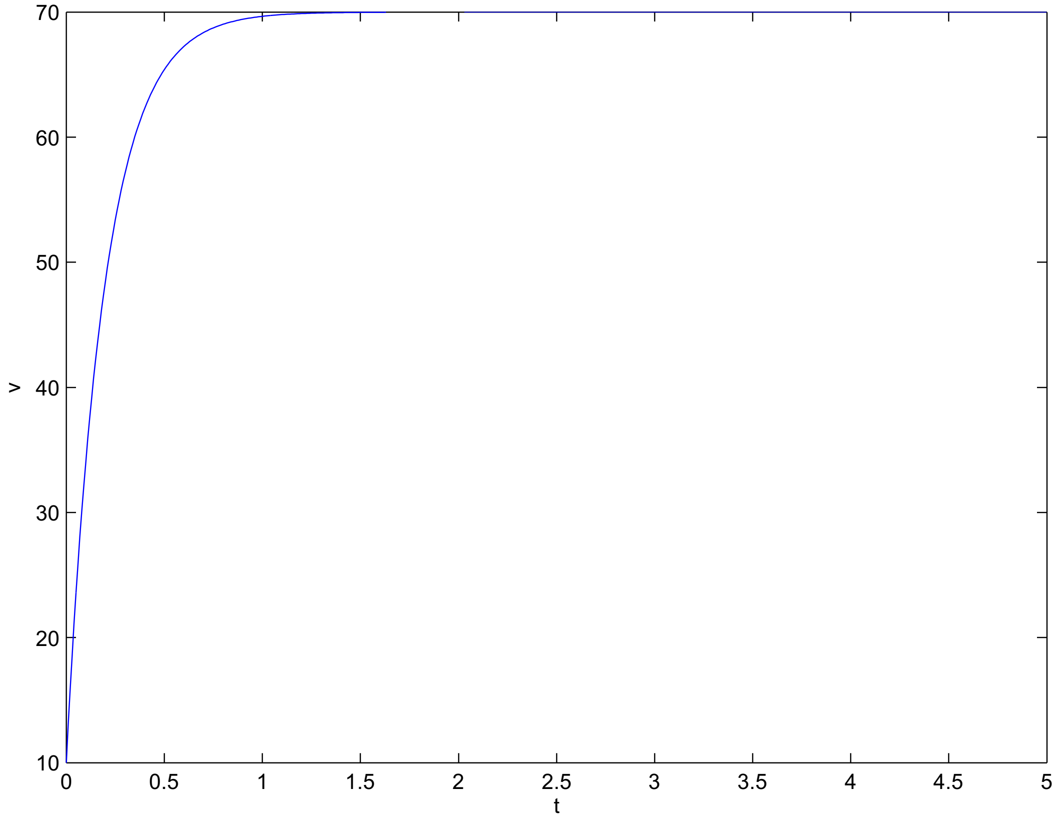
- *Încercarea 4*

$$u(t) = k_P e(t) + k_I \int_0^t e(\tau)d\tau$$

- Sau, de ce nu un Regulator PID?

$$u(t) = k_P e(t) + k_I \int_0^t e(\tau)d\tau + k_D \frac{de(t)}{dt}$$

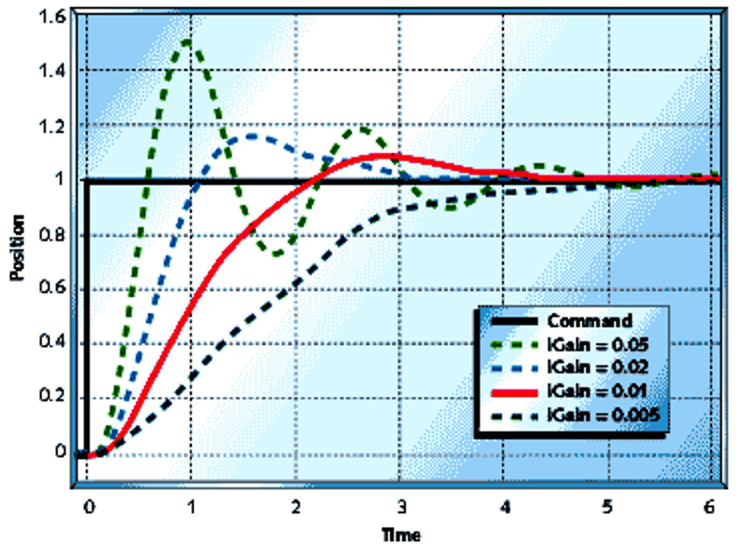
Regulator PI

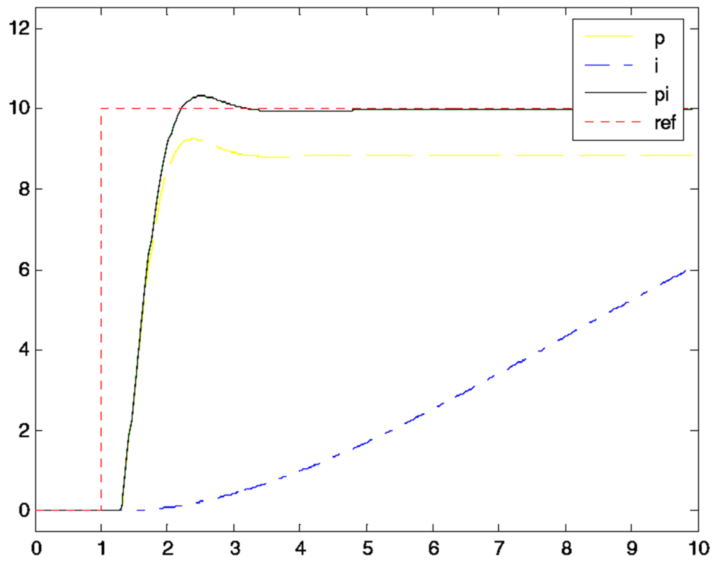


Control Integral

- Folosit pentru a adăuga precizie pe termen lung sistemului
- Aproape întotdeauna este folosit în conjuncție cu controlul proporțional (PI)
- Regulatorul însumează erorile trecute până când ieșirea ajunge la valoarea de referință

Control integral pur



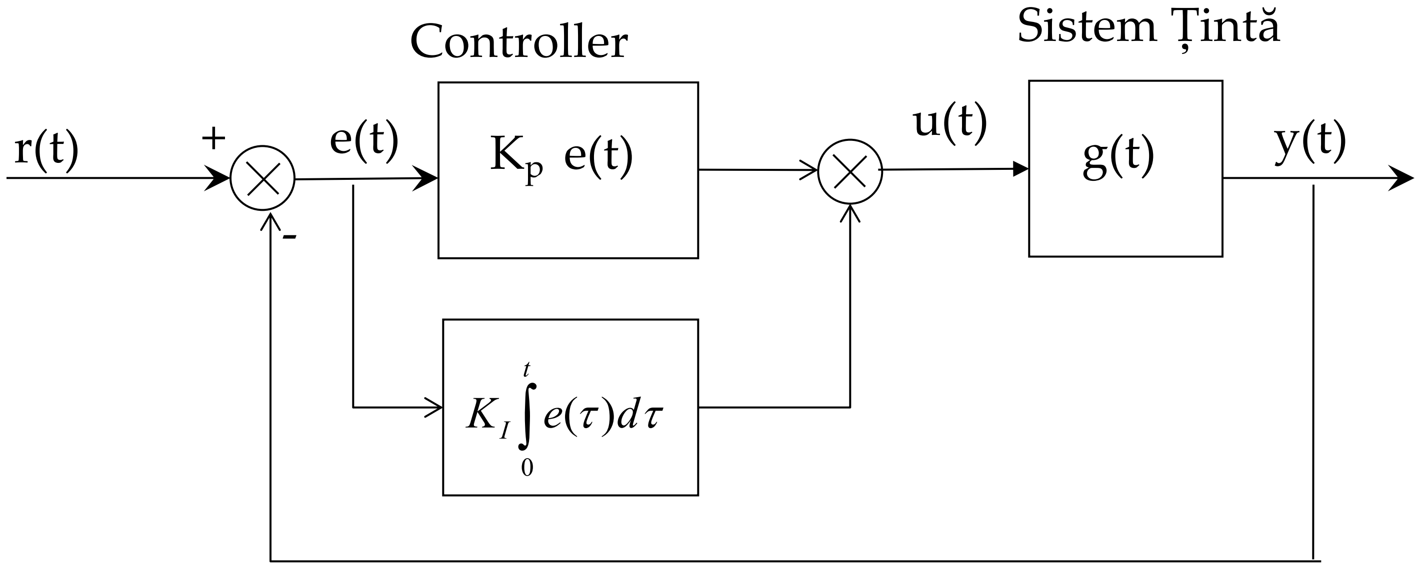


- Răspunsul controlului I este lent și uneori poate produce oscilația sistemului. De cele mai multe ori este preferat controlul PI.

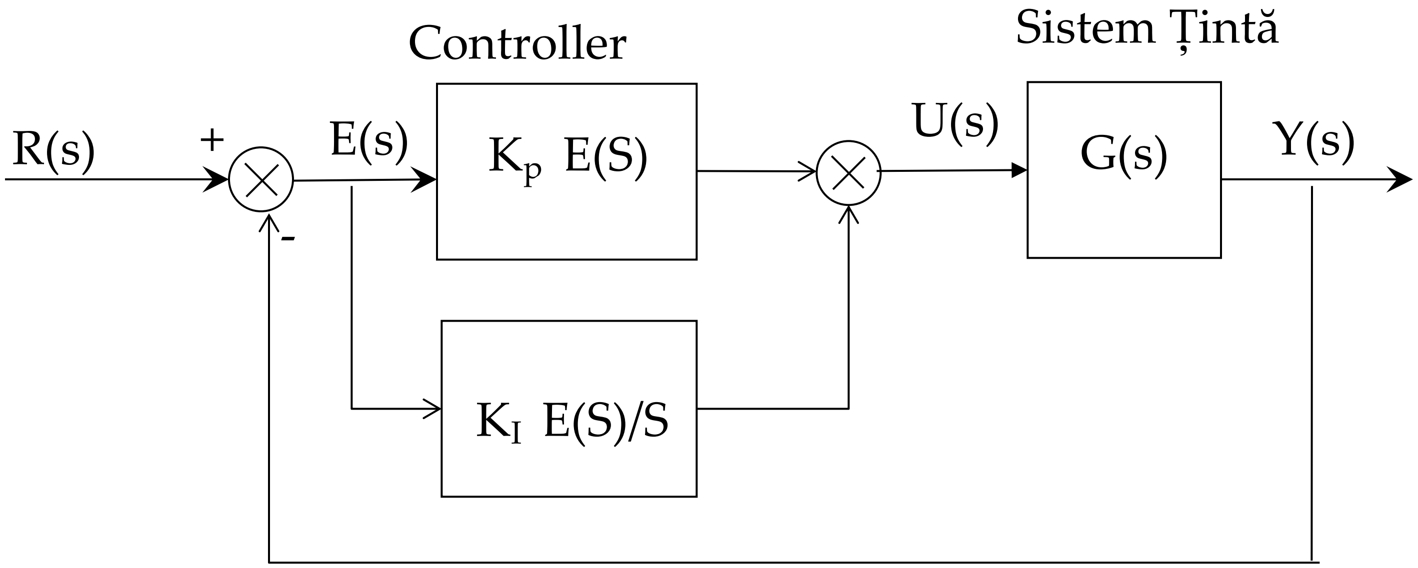
www.embedded.com

Control Proporțional-Integral

- Ieșirea regulatorului este direct proporțională cu integrala tuturor erorilor trecute.



Funcția de transfer PI

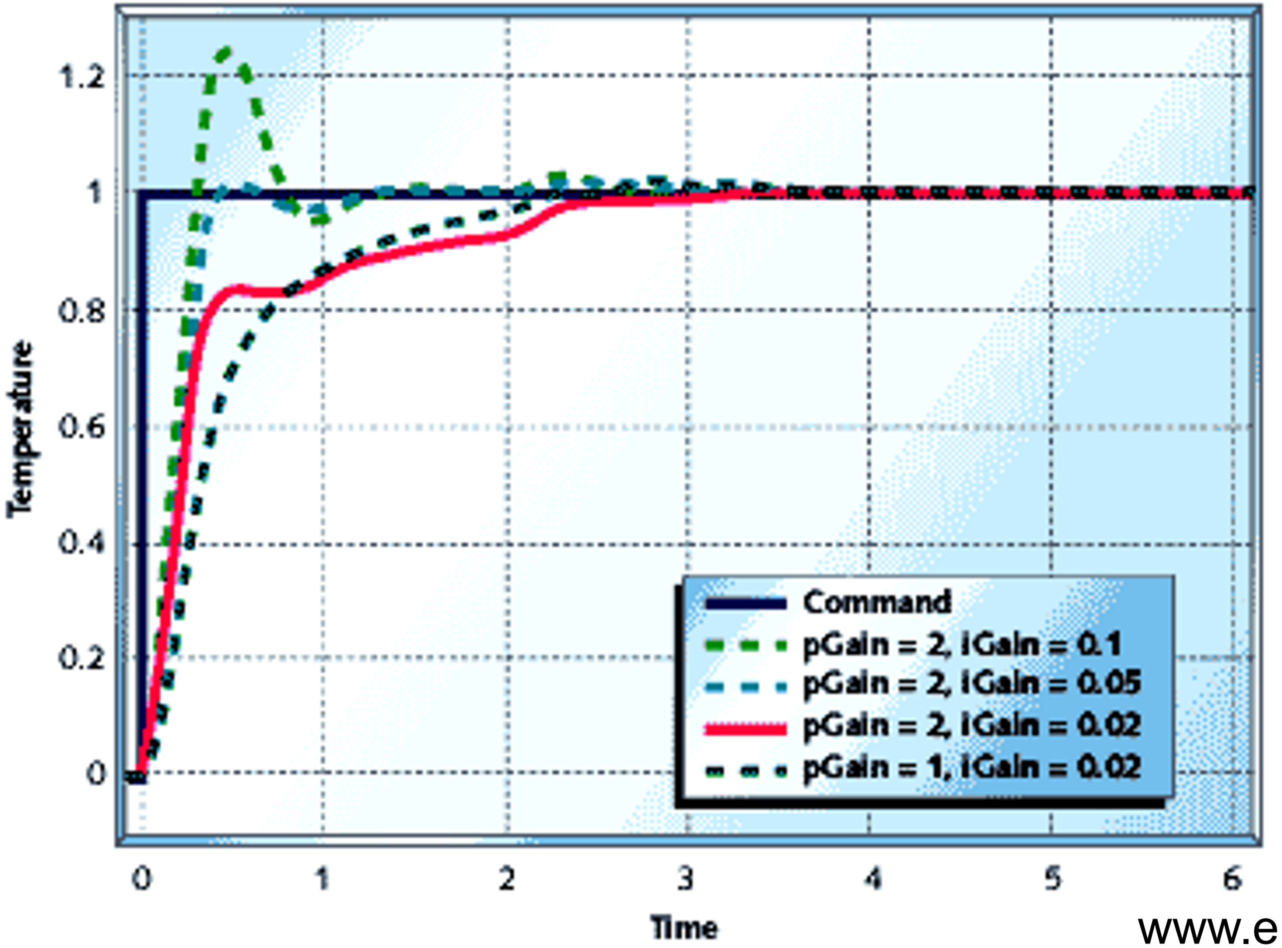


$$E(s) = R(s) - Y(s) \qquad U(s) = K_P E(s) + K_I \frac{E(s)}{s} \qquad Y(s) = G(s)U(s)$$

$$H(s) = \frac{(K_P + \frac{K_I}{s}) \cdot G(s)}{1 + (K_P + \frac{K_I}{s}) \cdot G(s)}$$

Exemplu

- Pentru regulatorul de temperatură:



www.embedded.com

Acuratețe

Ecuația caracteristică: $$1 + \left(K_P + \frac{K_I}{s}\right) \cdot G(s) = 0$$

$$e_{ss} = \lim_{k->\infty} e(k) = \lim_{s->0} \frac{1}{1 + \left(K_P + \frac{K_I}{s}\right) G(s)}$$

- Eroarea steady-state este zero cât timp sistemul este stabil.

Implementare Regulator PI

```
double iTerm;
// calculează  termenii PI
// între două limite
pid->iState  +=  error;
if (pid->iState  >  pid->iMax)
    pid->iState  =  pid->iMax;
else if (pid->iState  <  pid-> iMin)
    pid->iState  =  pid->iMin;
iTerm  =  pid->iGain * iState; // calculează factorul integral
```

Concluzii PI

- Controlul Integral pur are un timp de răspuns lent și poate produce oscilația sistemului

- Controlul PI are un timp de răspuns mai mic decât I pur.

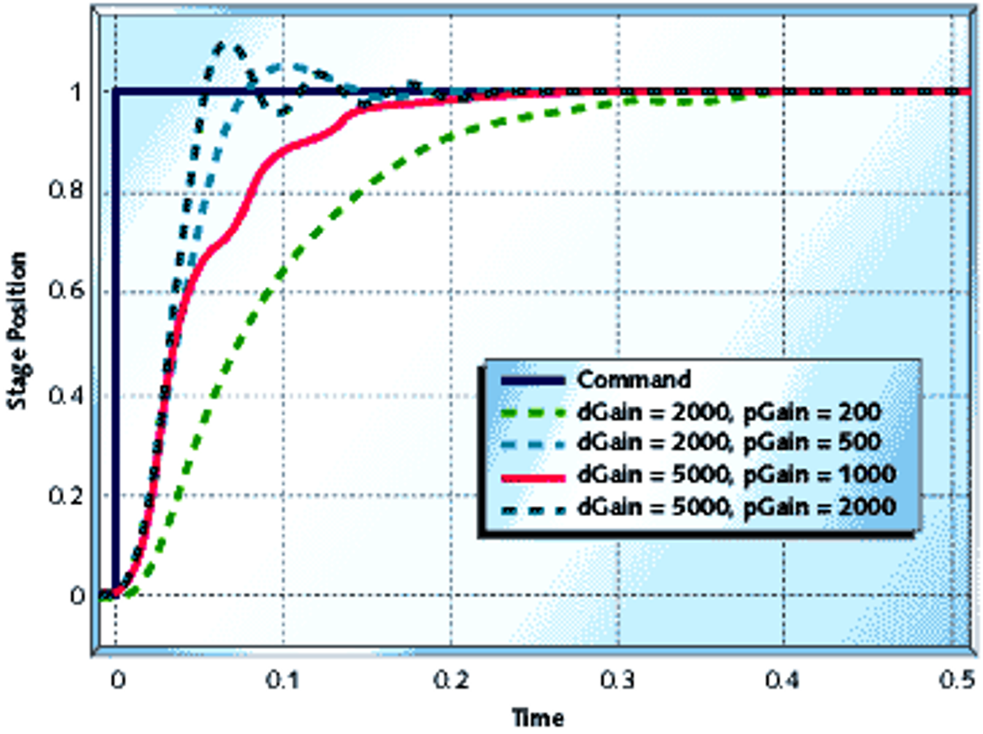
- Când e stabil, PI ajunge întotdeauna la valoarea de referință (e_{ss} = 0)

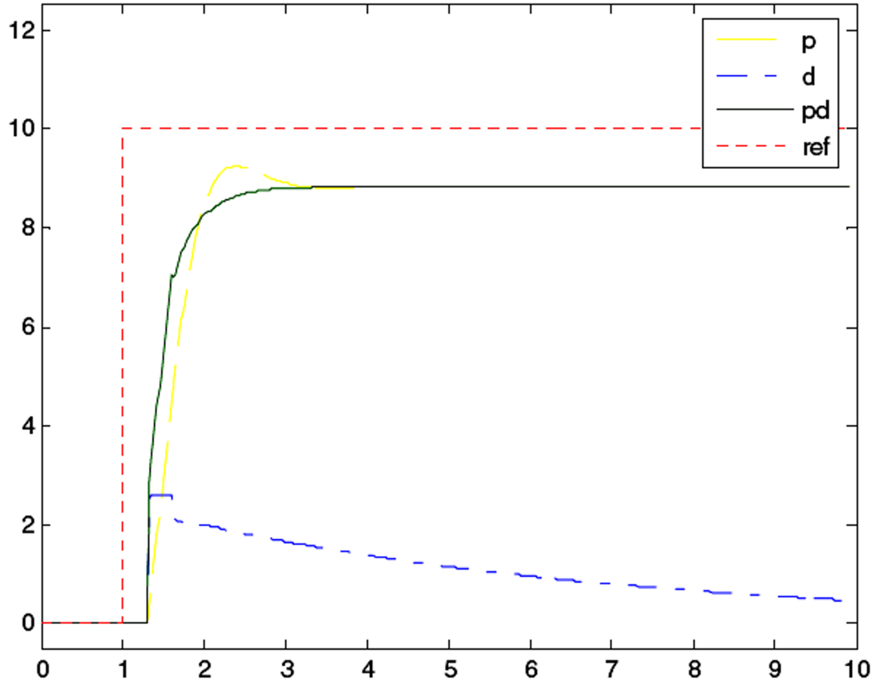
Control Diferențial

- Controlul P tratează comportamentul prezent al sistemului
- Controlul I tratează comportamentul trecut al sistemului
- Dacă nu stim cum o să se comporte sistemul, nu avem cum sa-l stabilizăm.

  Anticiparea comportamentului -> stabilitate marită
- Controlul D monitorizează viteza de schimbare a erorii sistemului

Control PD

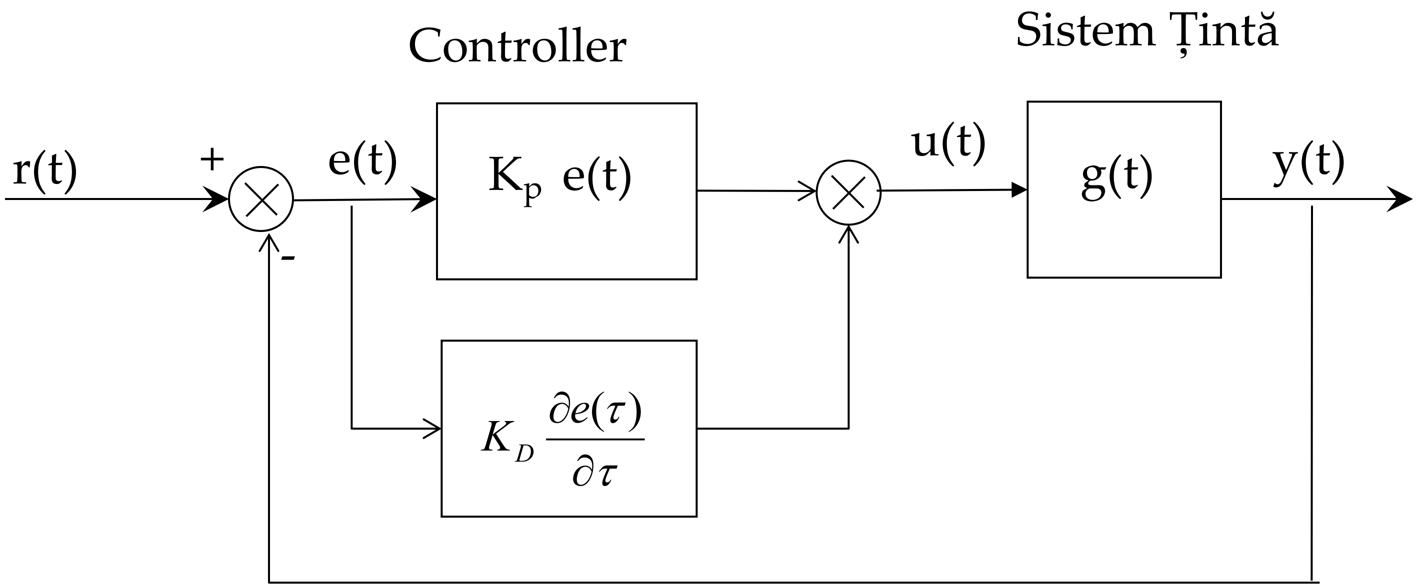




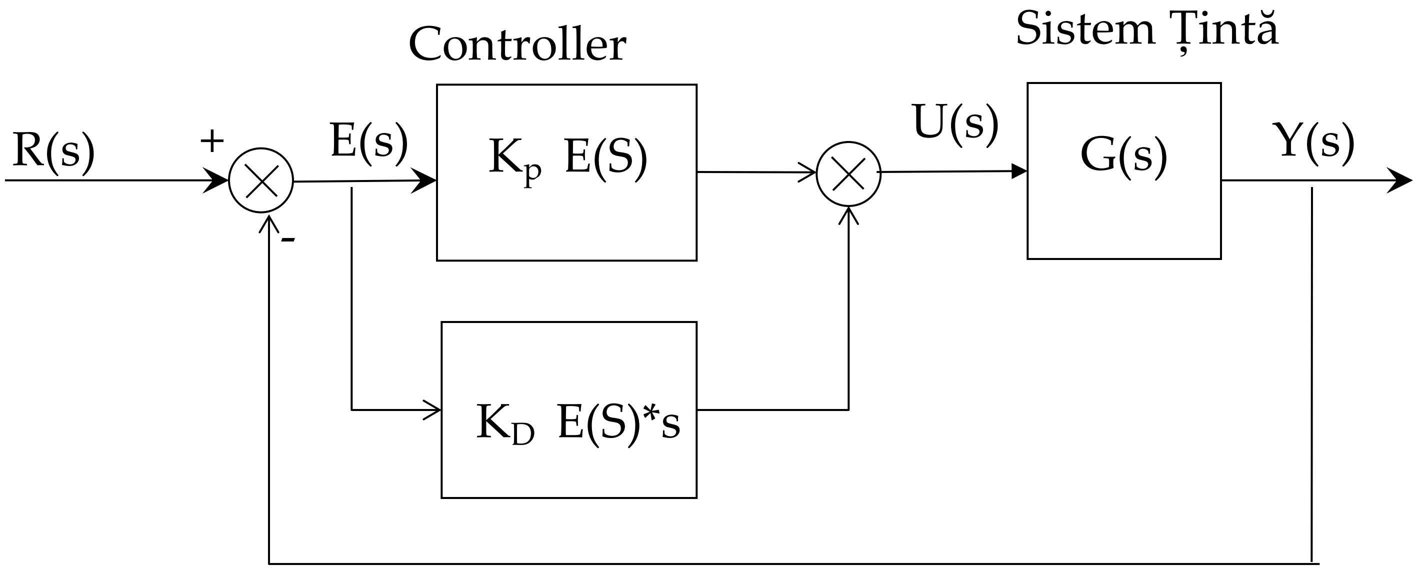
- Controlul diferențial este cel mai puternic dar și cel mai problematic. Sistemul devine instabil la zgomot și la neregularități de eșantionare.

Regulator Proporțional-Diferențial

- Ieșirea regulatorului este direct proporțională cu rata de creștere a erorii.



Funcția de transfer PD



$$E(s) = R(s) - Y(s) \quad U(s) = K_P \cdot E(s) + K_D \cdot sE(s) \quad Y(s) = G(s)U(s)$$

$$H(s) = \frac{(K_P + sK_D)\cdot G(s)}{1 + \left(K_P + sK_D\right)\cdot G(s)}$$

Acuratețe

Ecuația caracteristică: $1+\left(K_P+sK_D\right)\cdot G(s)=0$

$$e_{ss}=\lim_{k->\infty} e(k)=\lim_{s->0}\frac{1}{1+\left(K_P+sK_D\right)G(s)}$$

- Pentru e_{ss}, PD se aseamănă cu controlul P. Sitemul poate avea probleme în atingerea valorii de referință.

Implementare Regulator PD

```
double dTerm;

dTerm = pid->dGain * (position - pid->dState);

pid->dState = position;
```

Concluzii PD

- Controlul PD are un timp de răspuns foarte mic
- Anticipează comportamentul viitor al sistemului folosindu-se de variația erorii măsurate
- Este foarte util într-un sistem de control fin al mișcării/poziției datorită vitezei crescute de răspuns
- Sensibilitate crescută la zgomote și perturbații datorată factorului de amplificare mare.
- Controlul PD pot fi folosit pentru micșorarea depășirii superioare (overshoot) în locul sistemelor cu control P.

Regulatorul PID

$$u(t) = k_P e(t) + k_I \int_0^t e(\tau)d\tau + k_D \frac{de(t)}{dt}$$

- P: Contribuie la stabilitatea sistemului, timp de răspuns mediu spre lent
- I: Tracking și rejecția perturbațiilor, timp de răspuns lent. Poate cauza oscilații
- D: Timp rapid de răspuns. Sensibil la zgomot.
- PID: Cel mai folosit tip de controller low-level.

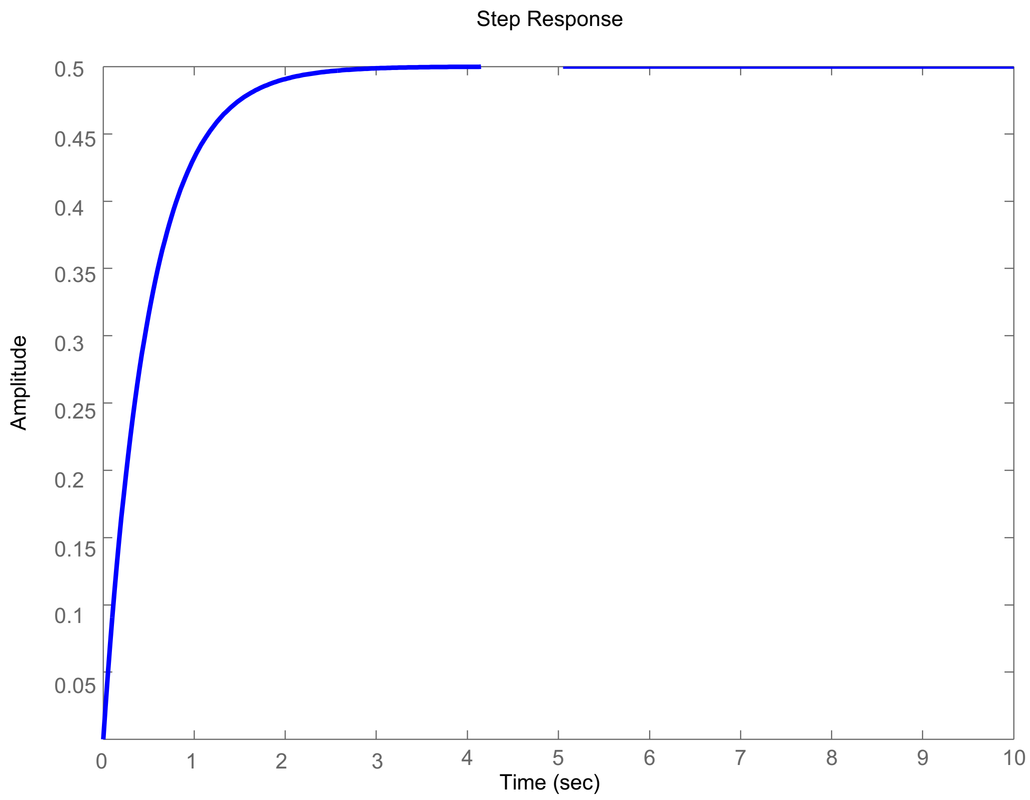
– Notă: stabilitatea nu este garantată.

- Mecanismul de feedback are o abilitate remarcabilă de a tolera incertitudinea indusă de modelul fizic adoptat!

Cruise-Controller (Again)

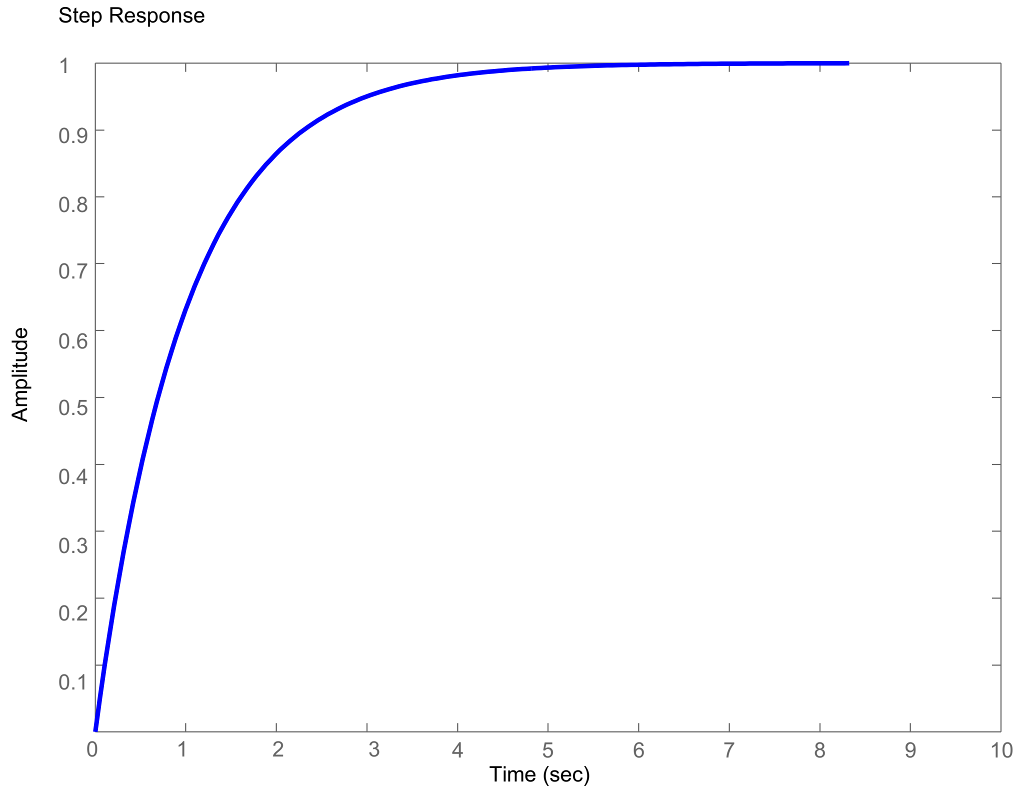
$$\dot{x} = \frac{c}{m}u - \gamma x \qquad c = 1, m = 1, \gamma = 0.1, r = 1$$

Cruise-Controller (Again)



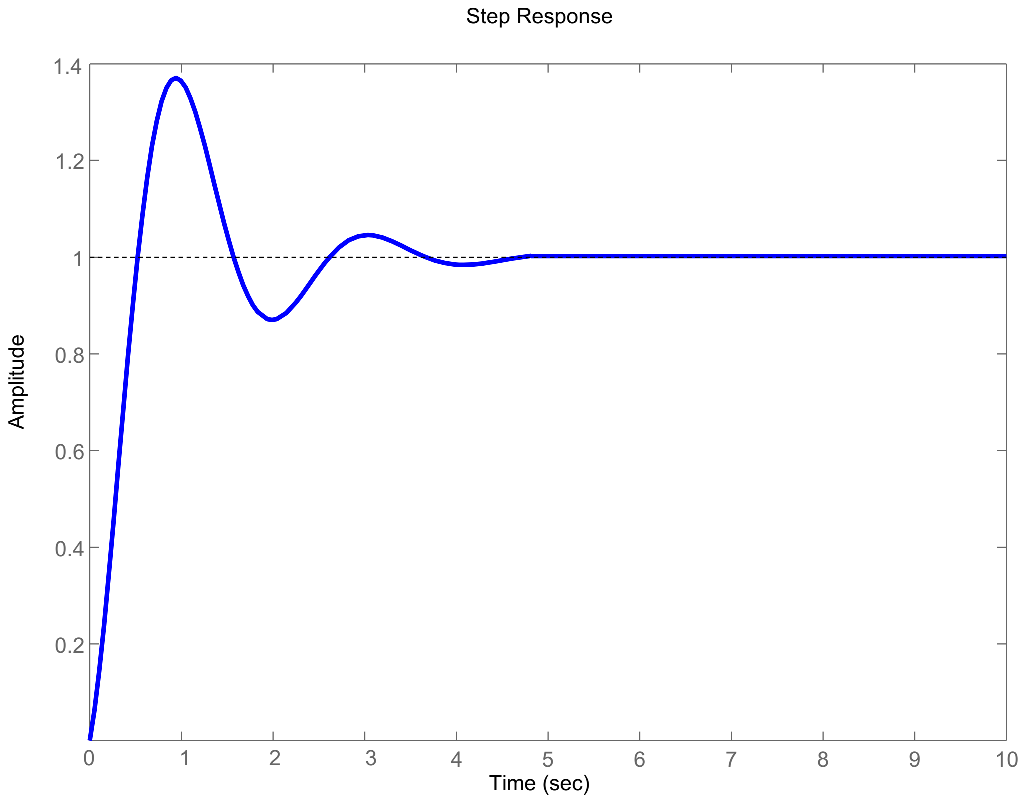
$k_P = 1,\qquad k_I = 0,\qquad k_D = 0$

Cruise-Controller (Again)



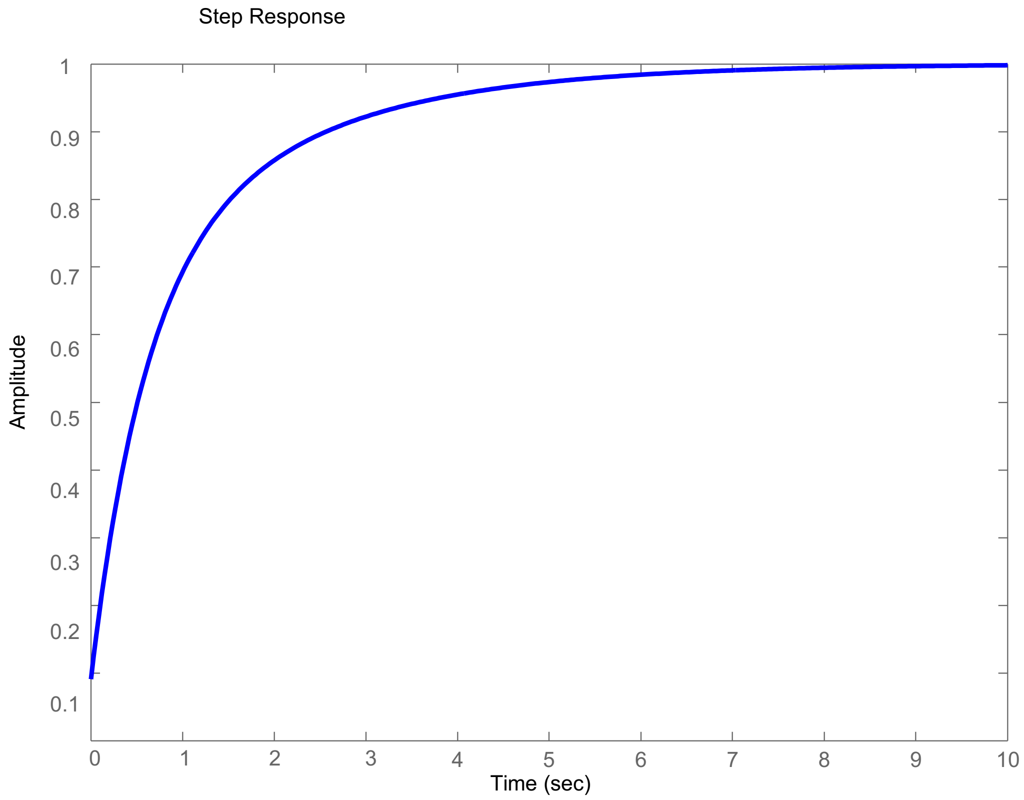
$k_P = 1, \quad k_I = 1, \quad k_D = 0$

Cruise-Controller (Again)



$k_P = 1, \quad k_I = 10, \quad k_D = 0$

Cruise-Controller (Again)



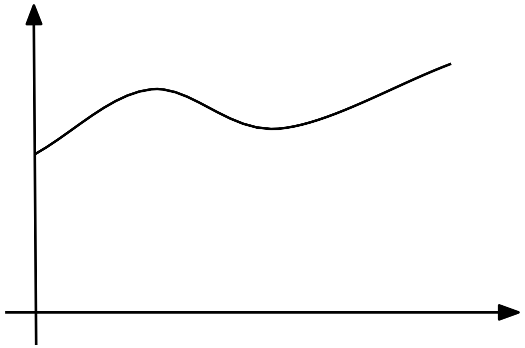
$k_P = 1, \quad k_I = 1, \quad k_D = 0.1$

Implementare

$$u(t) = k_P e(t) + k_I \int_0^t e(\tau)d\tau + k_D \frac{de(t)}{dt}$$

- Cum să transformăm o integrală în ceva implementabil în cod?

Δt (sample time) $\dot{e} \approx \dfrac{e_{new} - e_{old}}{\Delta t}$

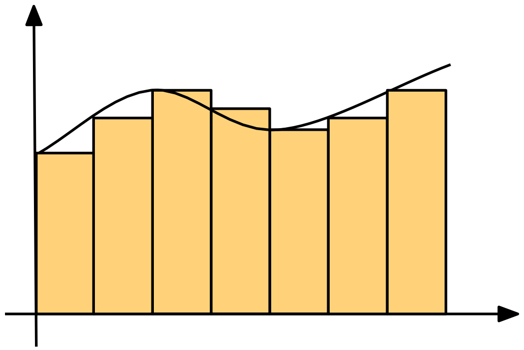


Implementare

$$u(t) = k_P e(t) + k_I \int_0^t e(\tau)d\tau + k_D \frac{de(t)}{dt}$$

- Cum să transformăm o integrală în ceva implementabil în cod?

Δt (sample time) $\qquad \dot{e} \approx \frac{e_{new} - e_{old}}{\Delta t}$

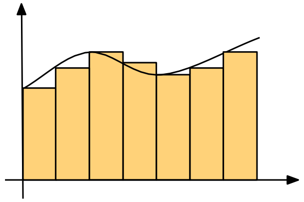


Implementare

$$u(t) = k_P e(t) + k_I \int_0^t e(\tau)d\tau + k_D \frac{de(t)}{dt}$$

- Cum să transformăm o integrală în ceva implementabil în cod?

Δt (sample time) $\quad \dot{e} \approx \dfrac{e_{new} - e_{old}}{\Delta t}$



$$\int_0^t e(\tau)d\tau \approx \quad {}^{N}_{k=0}$$

$N + 1$

$$\Delta t E_{new} = \Delta t$$
$$E_{new} = E_{old} + e$$

Implementare

Embedded Systems Laboratory

- De fiecare dată când este apelat regulatorul

```
read e; e_dot=e-old_e; E=E+e;
u=kP*e+kD*e_dot+kI*E; old_e=e;
```

Notă: Coeficienții includ acum și timpul de eșantionare și trebuie scalați în consecință

Exemplu: Regulator de altitudine pentru un quadcopter

$$\ddot{x} = cu - g \qquad u = k_P e + k_I \int e\, dt + k_D \dot{e}$$