Regulacije

Procesna avtomatika

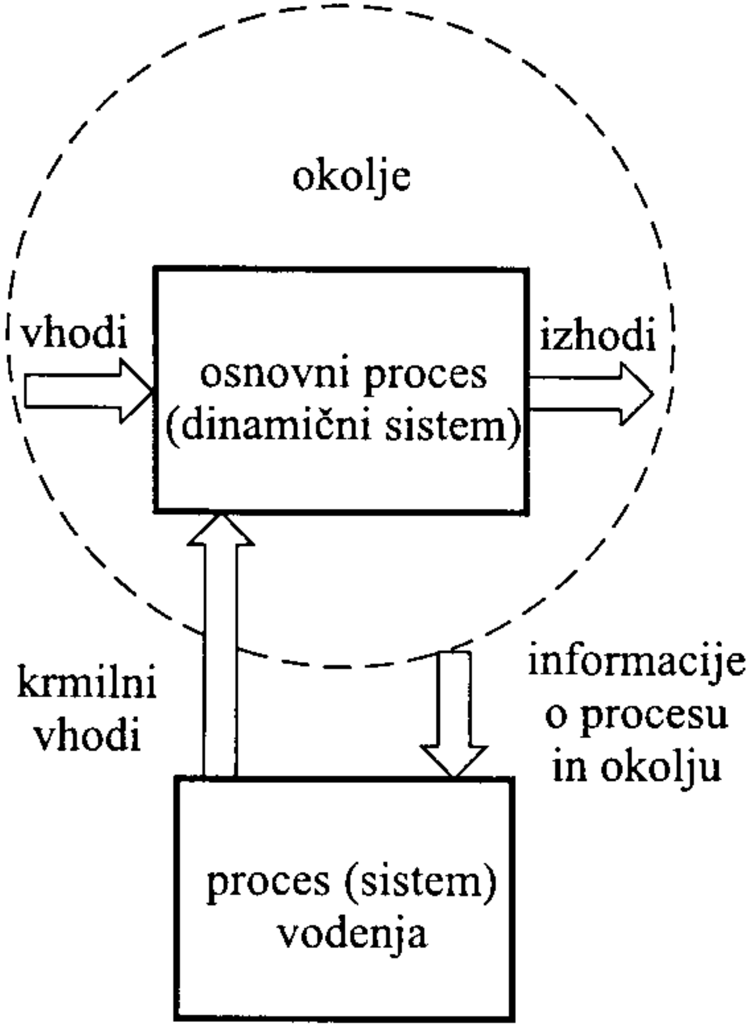
Uroš Lotrič, Nejc Ilc

Osnove vodenja

Vsem sistemom je skupno vodenje

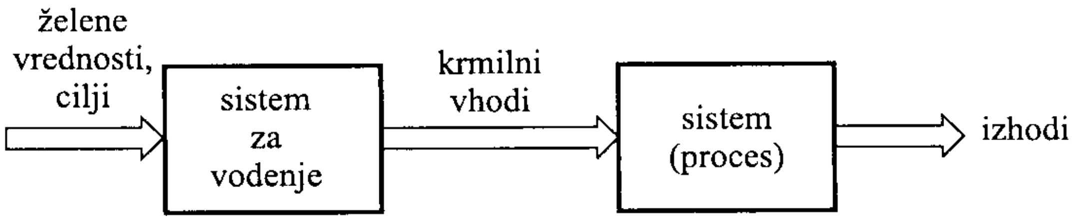
Omejili se bomo na dinamične ciljno usmerjene sisteme

Proces vodenja in primarni proces

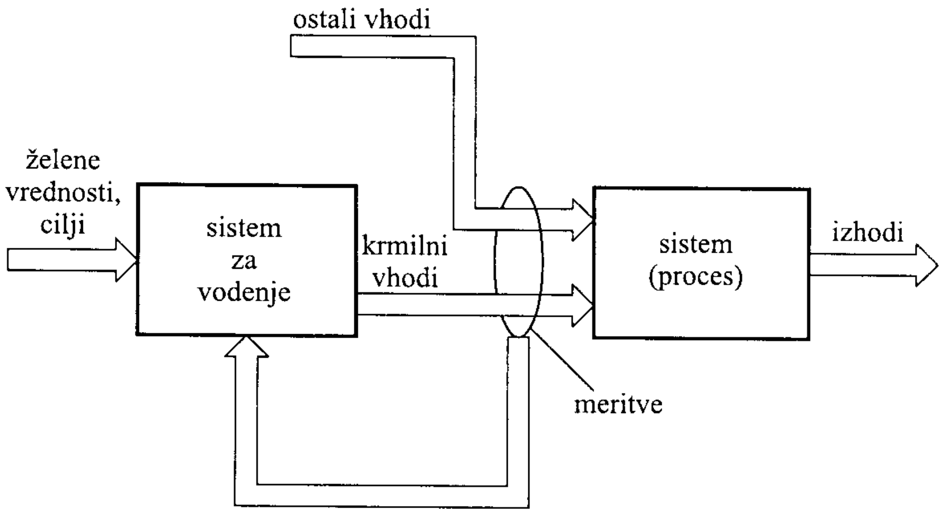


Tipi vodenja

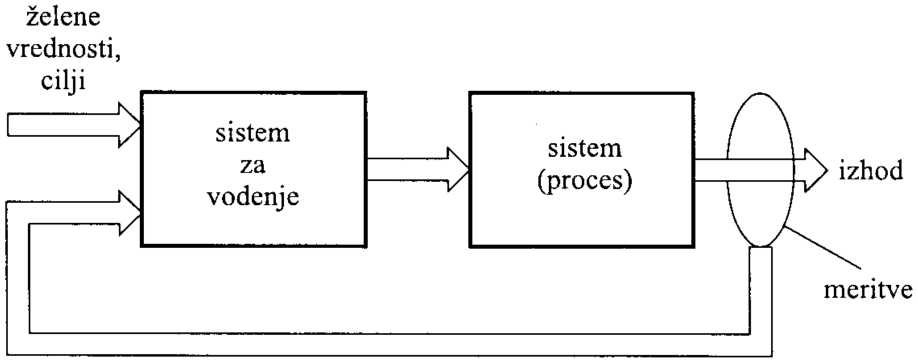
Odprto-zančno



Kompenzacijsko



Zaprto-zančno
(regulacija)



Regulacija

Zaprto-zančni regulator poskuša voditi proces tako, da poskuša merjeno vrednost kar najbolj približati določeni referenčni točki

Regulator za vodenje procesa potrebuje senzorje za merjenje stanja in aktuatorje za vplivanje na proces

Regulatorji se pri vodenju procesa običajno ozirajo na napako, to je razliko med referenčno točko in dejansko meritvijo

Do napake pride, če

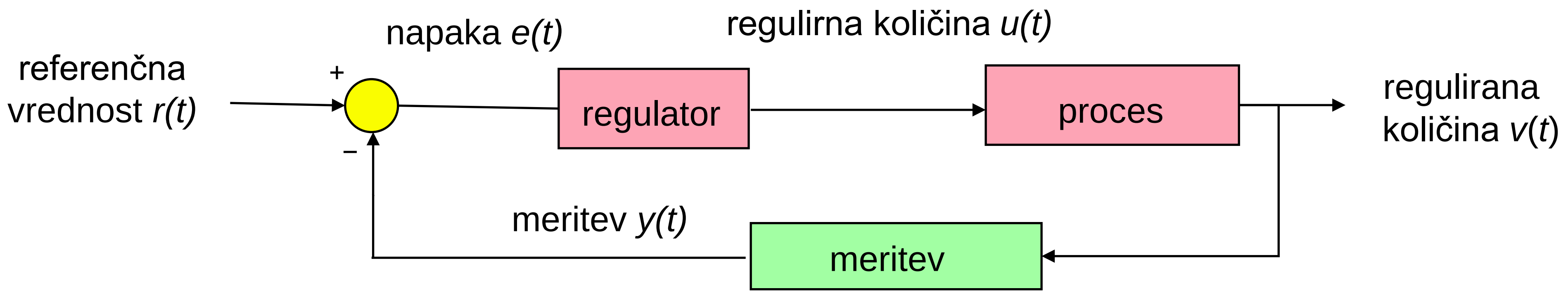
- operater namenoma spremeni referenčno točko ali
- pride do naključnih motenj ali obremenjenosti procesa

Naloga regulatorja je, da motnje avtomatsko odpravi

Regulacija

Vzdrževanje zvezne spremenljivke (regulirane količine) na želeni vrednosti (referenci)

- Odpravljanje vpliva motenj
- Sledenje spremembam želene vrednosti



Regulacija

Dva pristopa

- Standardni regulatorji
  - Zahtevajo matematični model procesa
  - Primeri: dvotočkovni, PID, Otto-Smith
- Moderni pristopi
  - Model procesa ustvarijo iz meritev
  - Primeri: mehka logika, nevronske mreže, kombinacija

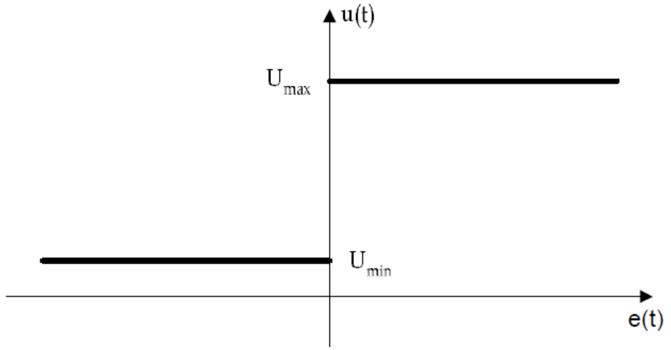
Praksa

- Mnogi realni procesi so nelinearni in jih je težko matematično opisati
- V praksi se je pokazalo, da se večino nelinearnih procesov obvladuje z regulatorji PID
- Izkušnje kažejo na smiselnost uporabe regulatorjev PID, saj so preprosti in zato razumljivi
- Namesto majhnega števila kompleksnih regulatorjev lahko uporabimo množico enostavnih regulatorjev PID

Regulacija – dvotočkovni regulator

Dvotočkovni regulator glede na vrednost napake *e(t)* = *r(t)* – *y(t)* preklaplja med dvema vrednostma regulirne količine *u(t)*

$$u(t) = \begin{cases} U_{max}; e(t) \geq 0 \\ U_{min}; e(t) < 0 \end{cases}$$



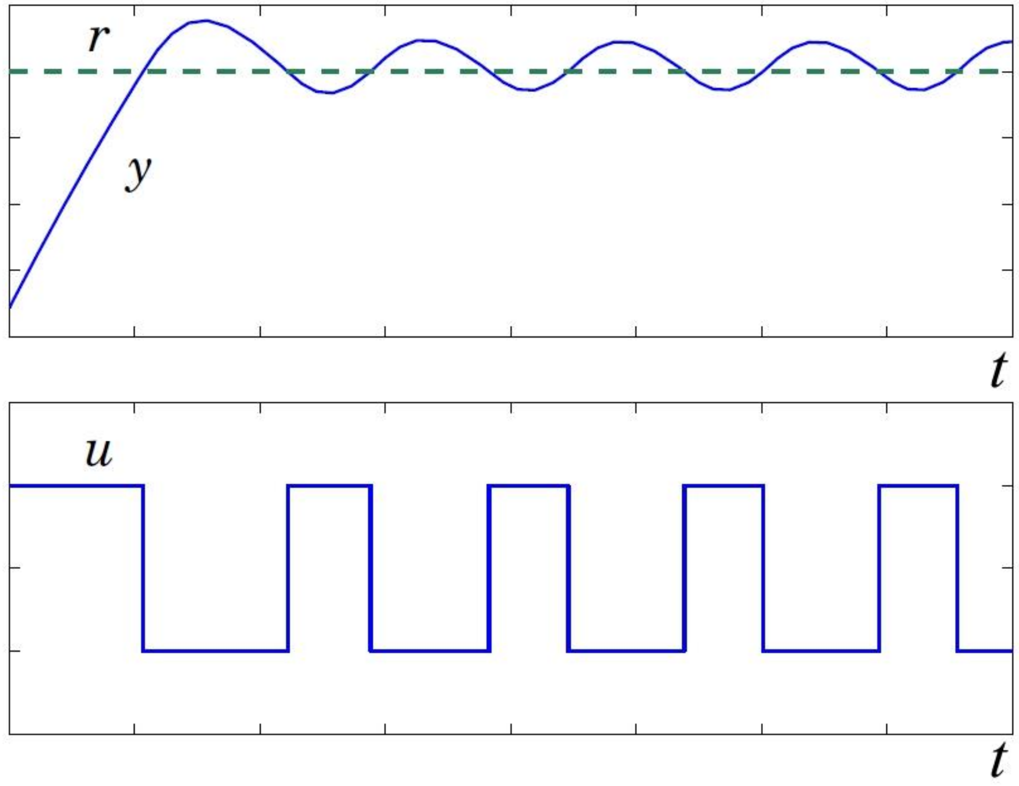
Regulacija – dvotočkovni regulator

Če ima regulirani proces pozitivni odziv, bo:

- visoka vrednost regulirne količine U_{max} povzročila povečanje regulirane količine,
- nizka vrednost regulirne količine U_{min} pa njeno zmanjšanje

Problem

- Regulirana količina neprestano oscilira okrog referenčne vrednosti
- Obraba preklopnih elementov pri hitrih procesih



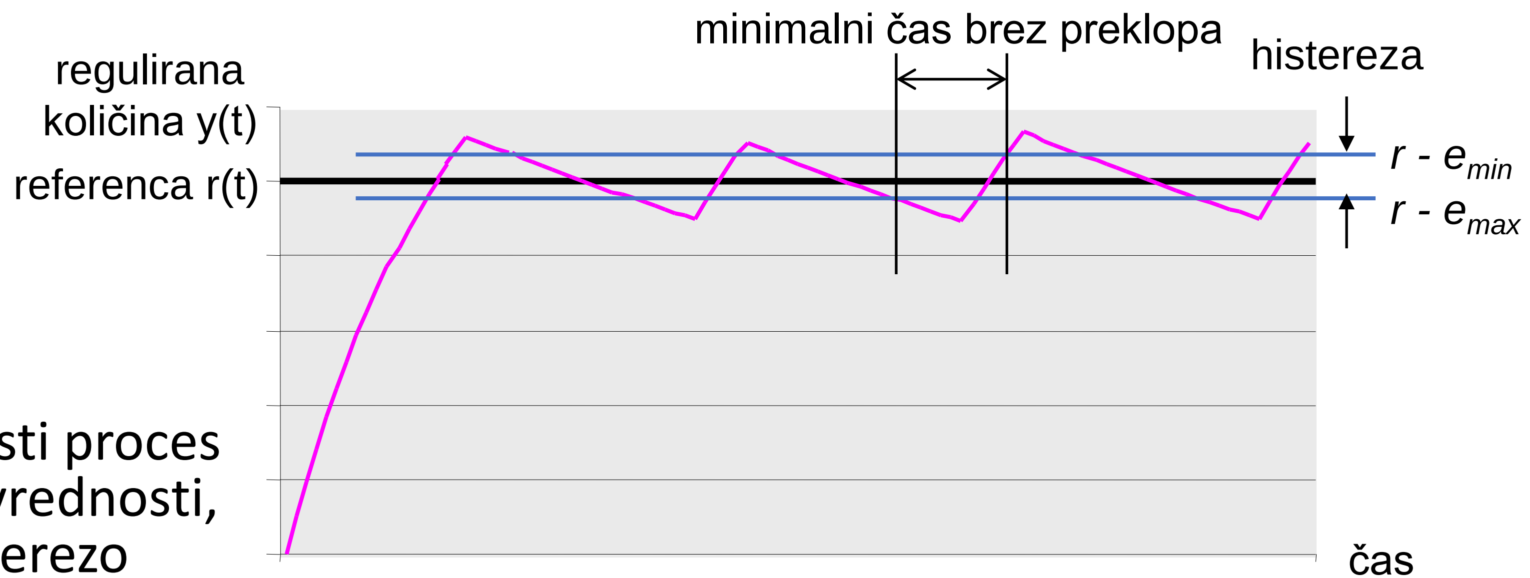
Rešitev?

Regulacija – dvotočkovni regulator

Histerezna zanka

- Ohlapne meje za preklop
- Minimalni čas brez preklopa

$$u(t) = \begin{cases} U_{max}; e(t) > e_{max} \\ U_{min}; e(t) < e_{min} \\ U_0; drugače \end{cases}$$



- Zaradi vztrajnosti proces lahko preseže vrednosti, določene s histerezo
- Boljša rešitev?

Regulator PID

Regulator PID je posplošitev dvotočkovnega regulatorja

Regulatorji PID uporabljajo tri režime dela:

- P – proporcionalni
- I – integrirni in
- D – diferencirni

Načina P in I se lahko uporabljata samostojno, način D skoraj ne

Kombinacije načinov PI, PD in PID so v praksi zelo pogoste

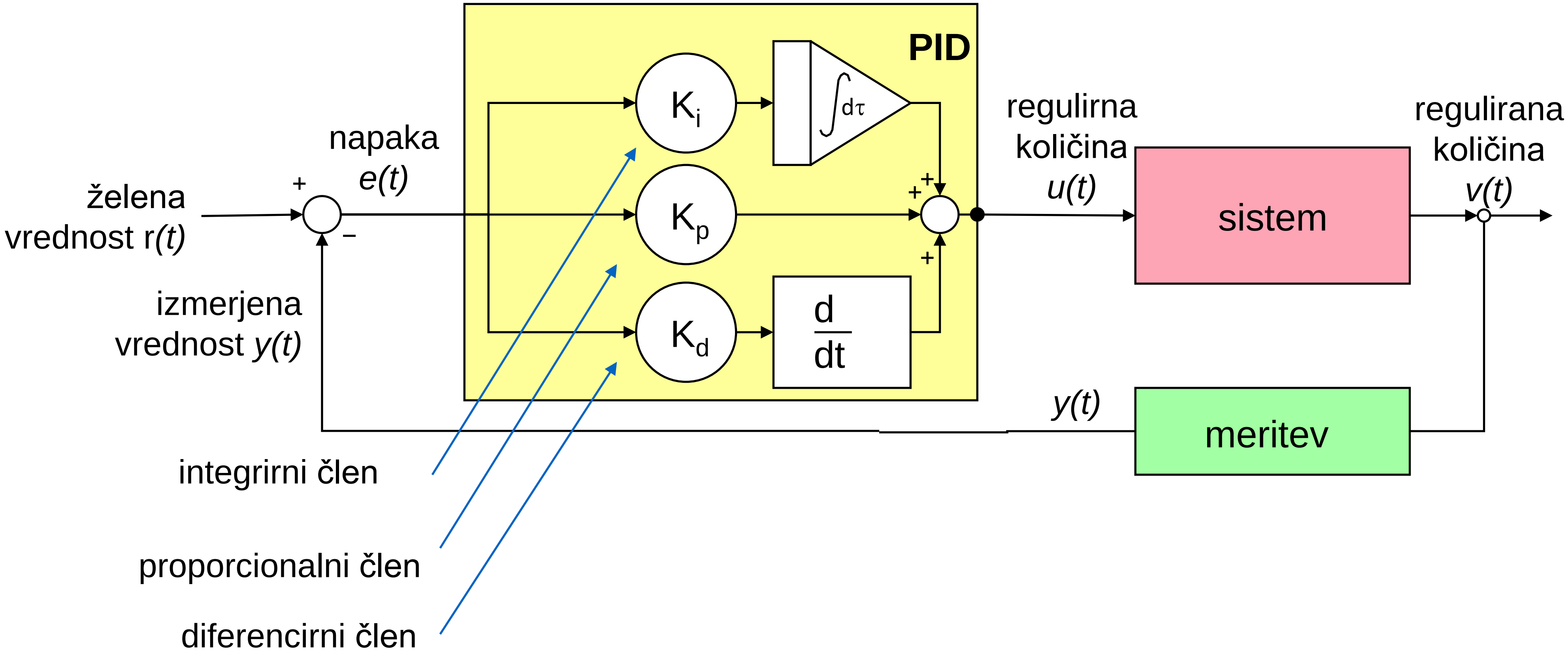
Zgodovina

- mehanska, pnevmatska, električna analogna regulacija PID
- danes večinoma digitalna regulacija PID z mikrokrmilniki

Regulator PID

Regulirna količina *u(t)* je podana kot vsota treh členov

- Vsak člen ustreza enemu režimu (proporcionalni, integrirni, diferencirni)
- Vplive posameznih členov določamo s konstantami K_p, K_i in K_d

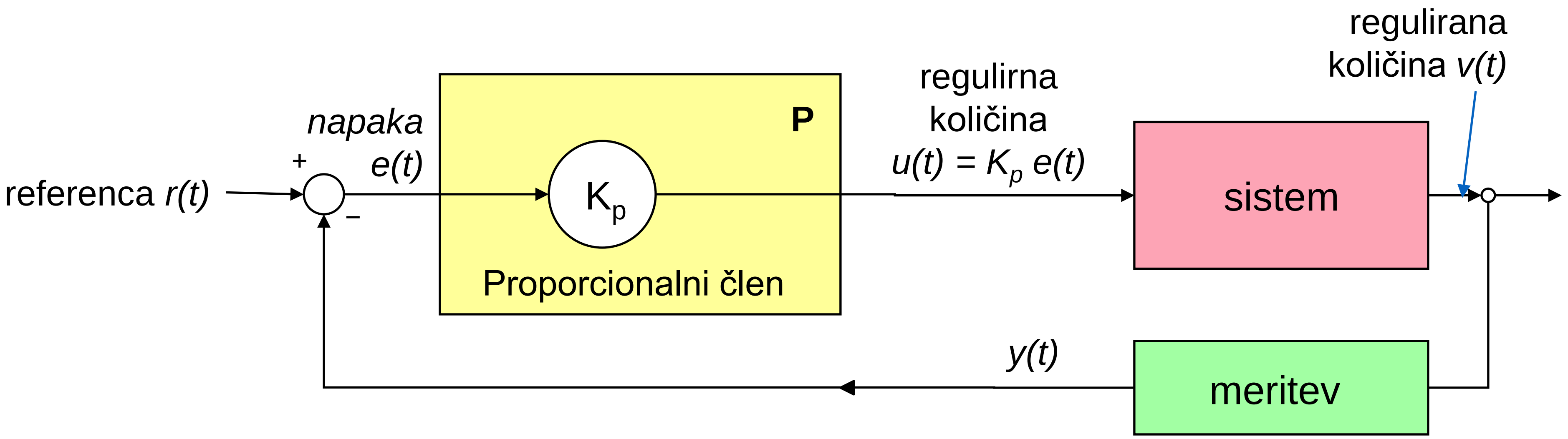


Regulator P

Najpreprostejši zvezni regulator

Posplošitev diskretnega dvotočkovnega regulatorja

Regulirna količina je proporcionalna napaki

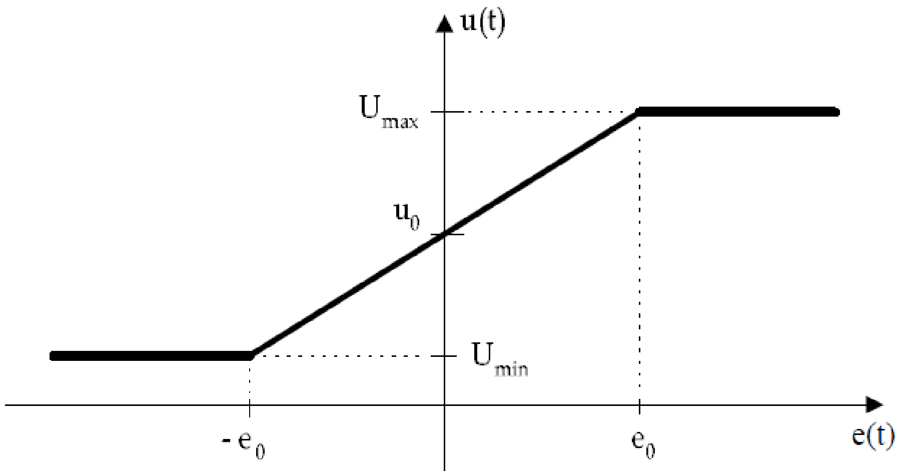


Regulator P

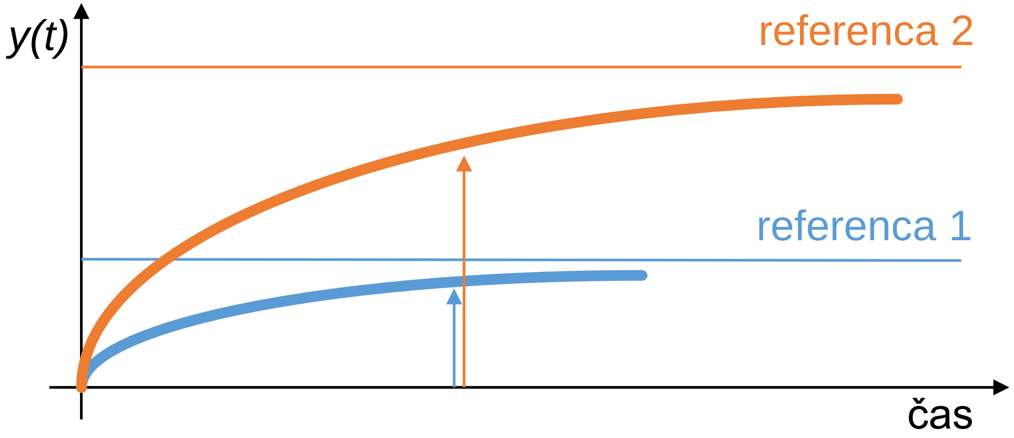
Zaradi fizikalnih omejitev je regulirna količina navzgor in navzdol omejena

Enačba

$$u(t) = \begin{cases} U_{max}; e(t) > e_0 \\ U_0 + K_p\, e(t); |e(t)| < e_0 \\ U_{min}; e(t) < -e_0 \end{cases}$$



Delovanje



majhna napaka → majhen popravek

velika napaka → velik popravek

Regulator P

Z regulatorjem P lahko odstranimo oscilacije, ki jih povzroči dvotočkovni regulator

Problem: napaka v mirovanju

- V delovnem območju regulatorja $[-e_0, +e_0]$ velja

$$u(t) = U_0 + K_p\, e(t) \quad \rightarrow \quad e(t) = \frac{u(t) - U_0}{K_p}$$

- Napaka je lahko nič, če je
  - $K_p = \infty$
    - dvotočkovni regulator, ki pripelje nazaj oscilacije
  - $u(t) = U_0$
    - izvedljivo, če lahko za vsako referenčno vrednost *r(t)* posebej nastavimo konstanto U_0

Regulator P

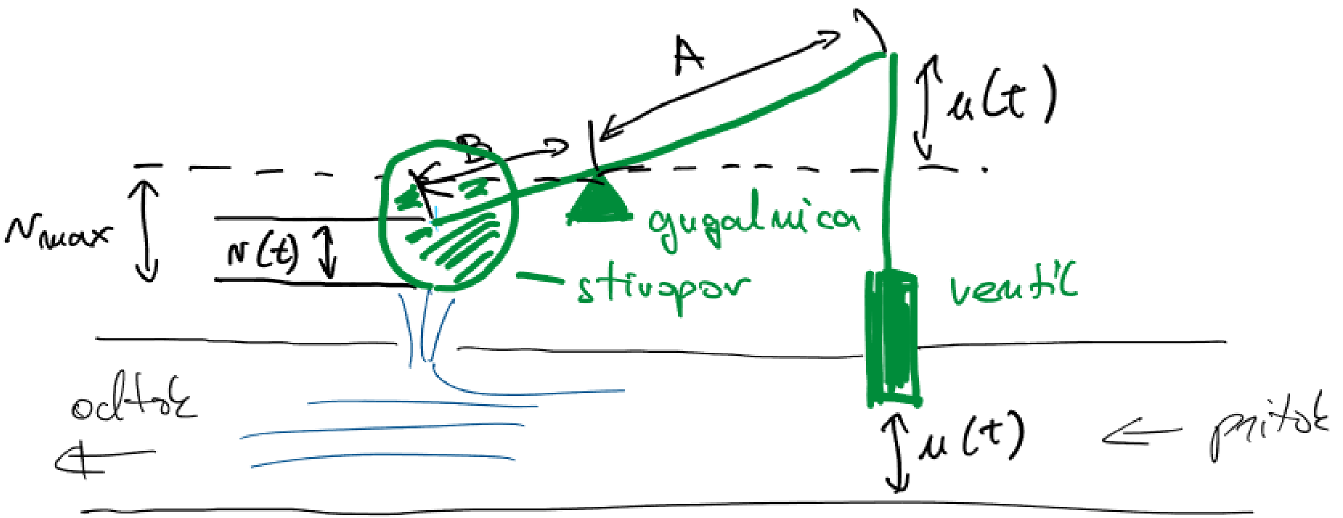
Z večanjem konstante K_p dosežemo:

- Manjšo napako v mirovanju in s tem boljše sledenje referenčni vrednosti
- Manjšo amplitudo napake
- V proces vnesemo hitrejšo dinamiko in s tem večjo občutljivost na merski šum

Regulator P

Primer

- Nadzor pretoka
- Tekočina teče po cevi, izliv
- Enačba procesa:
  - Ob velikem pretoku v(t) imamo visok curek u(t)
  - v(t) = C u(t)
    - u(t) – položaj ventila v cevi
    - v(t) – položaj plovca (stiropor)



- Regulator določa razmerje A: B
  - Iz razmerja stranic trikotnikov sledi: v_{max}-v(t) : B = u(t) : A
  - $u(t) = K_p(v_{max} – v(t))$, $K_p = A/B$
  - $u(t) = K_p(r – v(t)) + U_0$, $U_0 = K_p(v_{max} - r)$, r - referenčna vrednost
- Primer
  - C=1, r = 60 mm, v(t) = 55 mm, K_p = 1, U_0 = 50 mm → u(t) = 55 mm – reference ne dosežemo

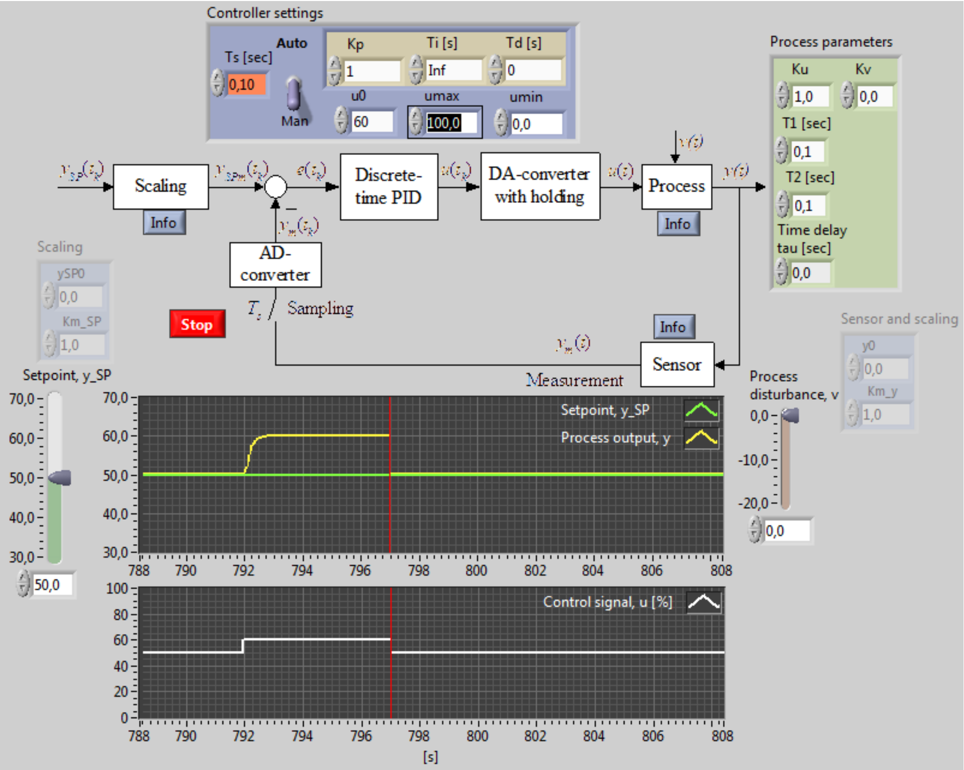
Regulator P

Primer

- Enačba procesa:
  - Za T1=T2 in Kv = 0 velja

$$y(t) = v(t) = K_u e^{-\frac{t-\tau}{T_1}} u(t-\tau)$$

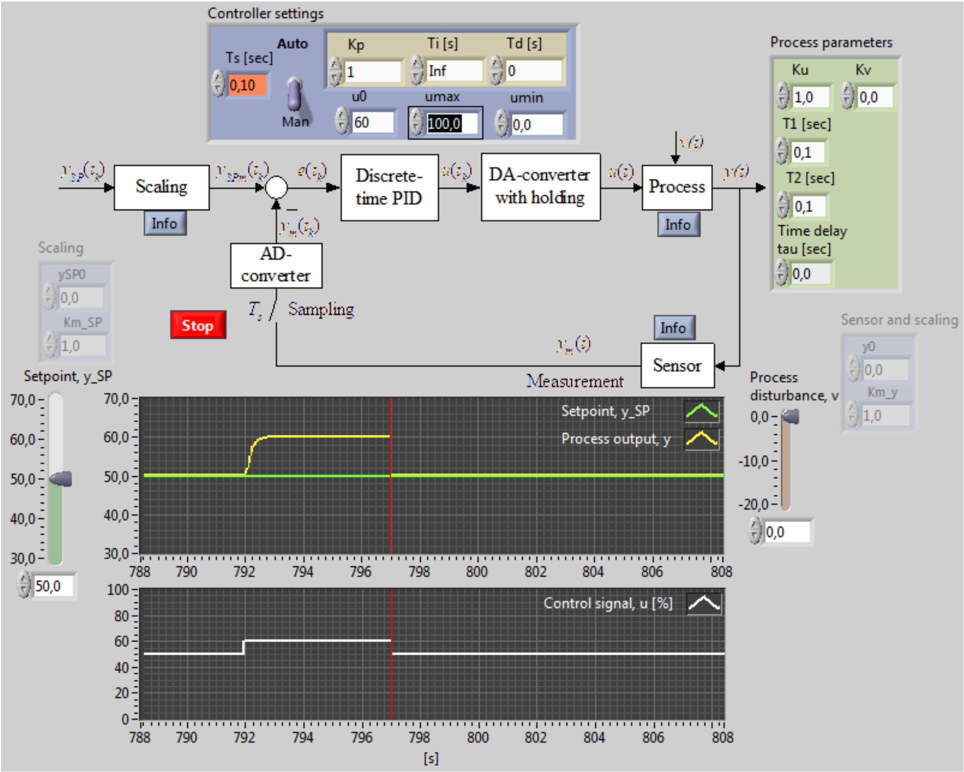
  Izberemo T1 = 0,1 s, τ=0 s, Ku = 1
- Poskus 1:
  - Stikalo postavimo na Man (ročno)
  - Spreminjamo U_0 in opazujemo odziv
- Poskus 2:
  - Nastavimo K_p = 1, T_i = Inf in T_d = 0
  - Stikalo postavimo na Auto (avtomatsko)
  - Po spremembi Setpoint iz 50 na 60 vidimo, da se zaradi prej omenjenih težav izhod sistema ustali na 55
  - Šele potem, ko postavimo U_0 = 60 se proces postavi na želeno vrednost



Regulator P

Primer

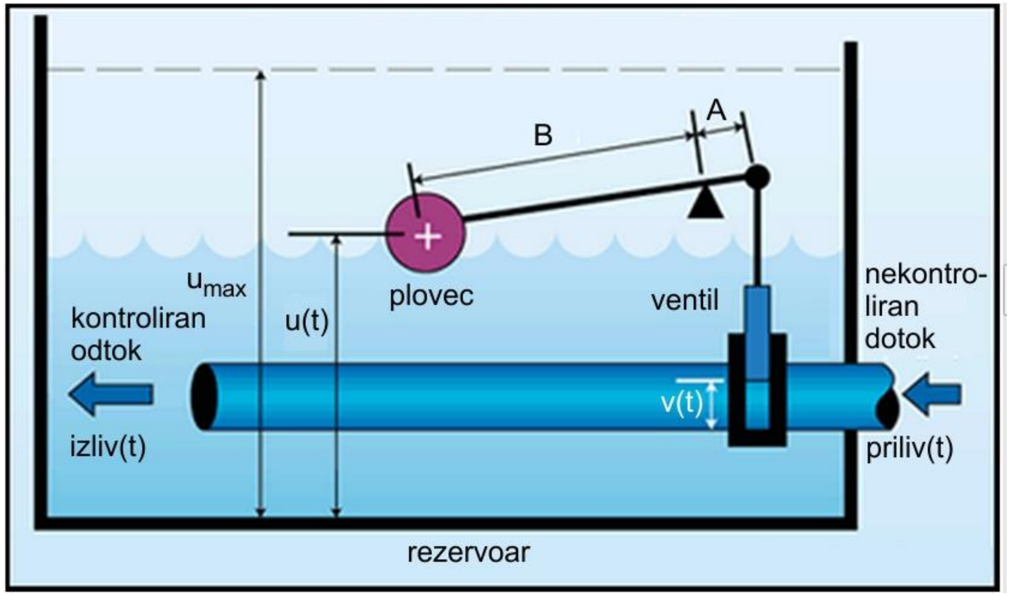
- Poskus 3:
  - Povečujemo K_p in spreminjamo Setpoint
  - Opazimo, da je proces pri večjih vrednosti K_p manj stabilen, napaka v mirovanju pa je manjša



Regulator P

Malo drugačen proces

- Straniščni kotliček
  - Podobno kot prej, le da je vse skupaj zaprto v tank, plovec ni na ustniku ampak na gladini
  - Nivo ni odvisen samo od velikosti odprtine temveč tudi od časa odprtja
- Enačba procesa
  - $u(t) = \int_{t_1}^{t_2} (\text{priliv}(t) - \text{izliv}(t))dt$
- Regulator P enak kot prej
  - $v(t) = K_p(r - u(t)) + U_0$
- Ne glede na dogajanje, se bo ventil zaprl takrat, ko bo rezervoar poln → integracija nam odpravi napako v mirovanju

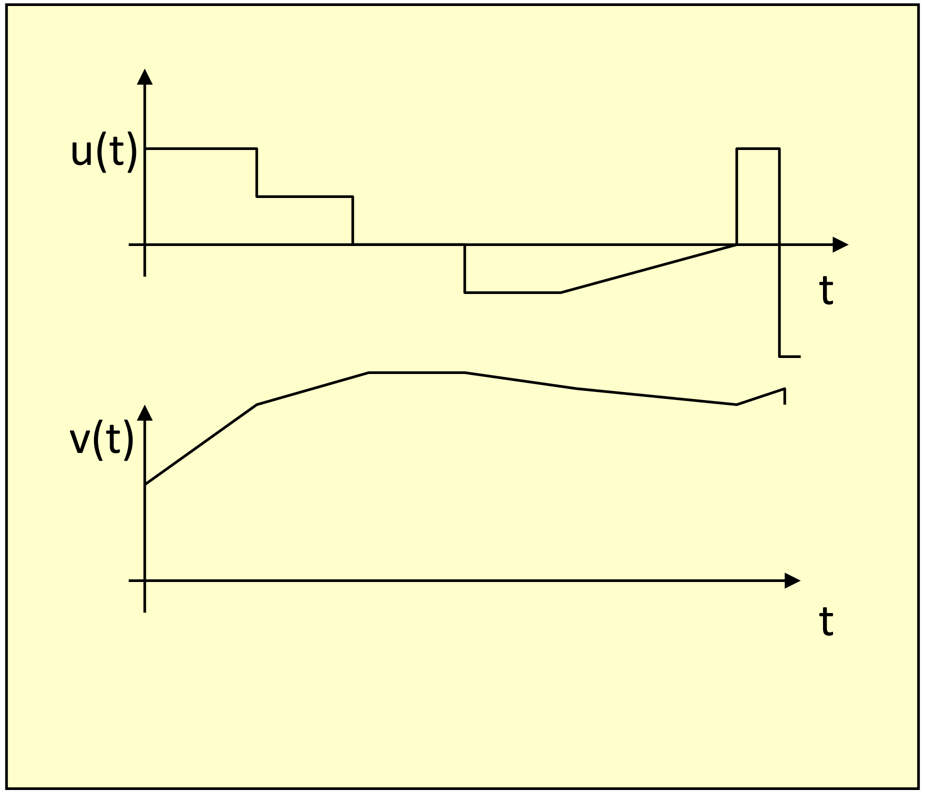


Očitno kombinacija procesa, ki vključuje integracijo, in regulator P deluje.

Integrator

Odziv integratorja

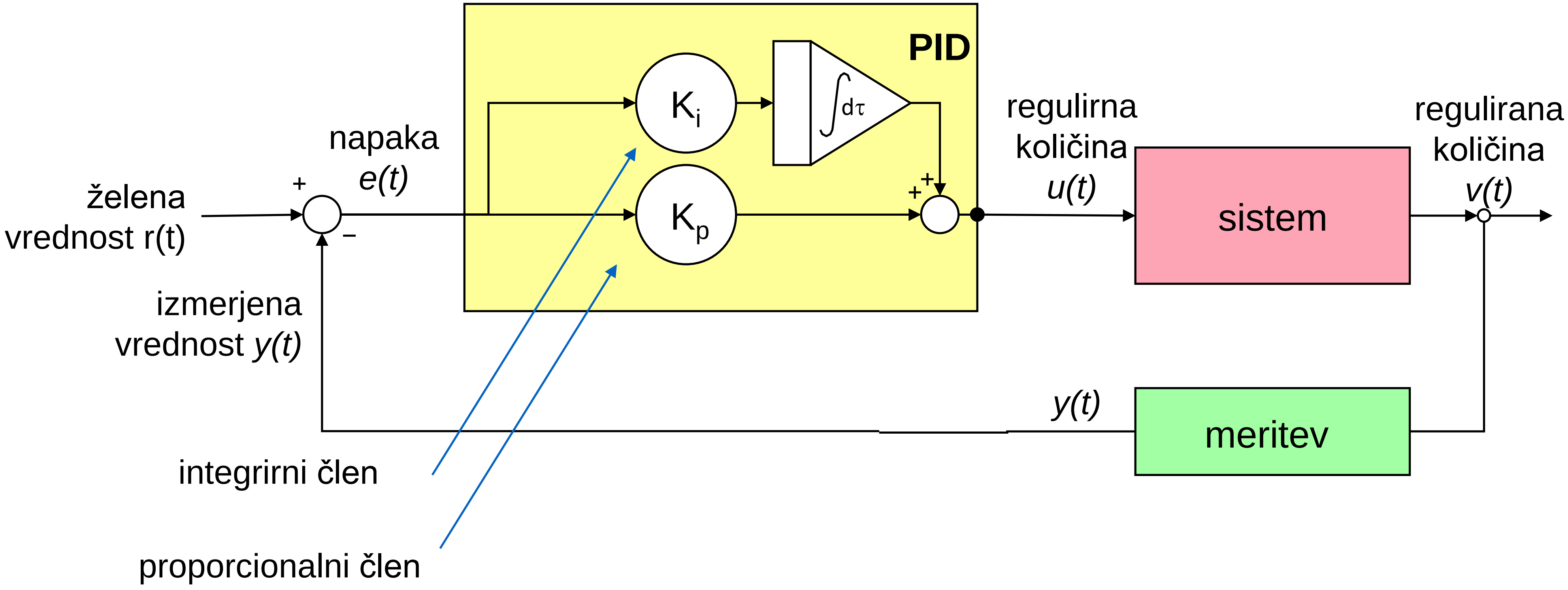
- Enačba:
  $\mathrm{v}(t) = K_i \int_0^t u(\tau)d\tau$
- primer



Regulator PI

Rešitev

- Regulatorju dodamo integrirni člen
- $u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau$



Regulator PI

Odziv integrirnega člena ni vezan na trenutno napako temveč na vsoto vseh prejšnjih napak

Primerjava

- Regulator P: $u(t) = K_p e(t) + U_0$
- Regulator PI: $u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau$
- Zamenjava U_0 z integralom nam omogoča odpravljanje napake v mirovanju
    - Glede na trenutno napako se vrednost U_0 stalno popravlja

Regulator PI bo odstranil oscilacije, značilne za dvotočkovni regulator in napako v mirovanju, značilno za regulator P

Slabost

- Uvedba integralnega člena privede do počasnejšega odziva in slabo vpliva na stabilnost celotnega sistema
- Regulatorji PI se pogosto uporabljajo v industriji kadar hitrost odziva ni preveč pomembna

Regulator PI

Regulator PI se večkrat zapiše nekoliko drugače:

- $$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau$$
  $$u(t) = K_p \left[e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau \right]$$
- Konstanta K_p / T_i se imenuje tudi ničenje (ang. reset mode)
  - Pri regulatorju P je potrebno s spreminjanjem U_0 ročno ničiti napako
  - Regulator PI opravlja to avtomatsko, odzivnost je odvisna od vrednosti konstante

Odziv regulatorja

- Imamo regulirani proces s pozitivnim odzivom
  - Pri e(t) > 0 sledi povečanje u(t), povečanje y(t) in zmanjšanje napake e(t)=r(t)-y(t)
  - Pri e(t) < 0 sledi zmanjšanje u(t), zmanjšanje y(t) in zmanjšanje napake

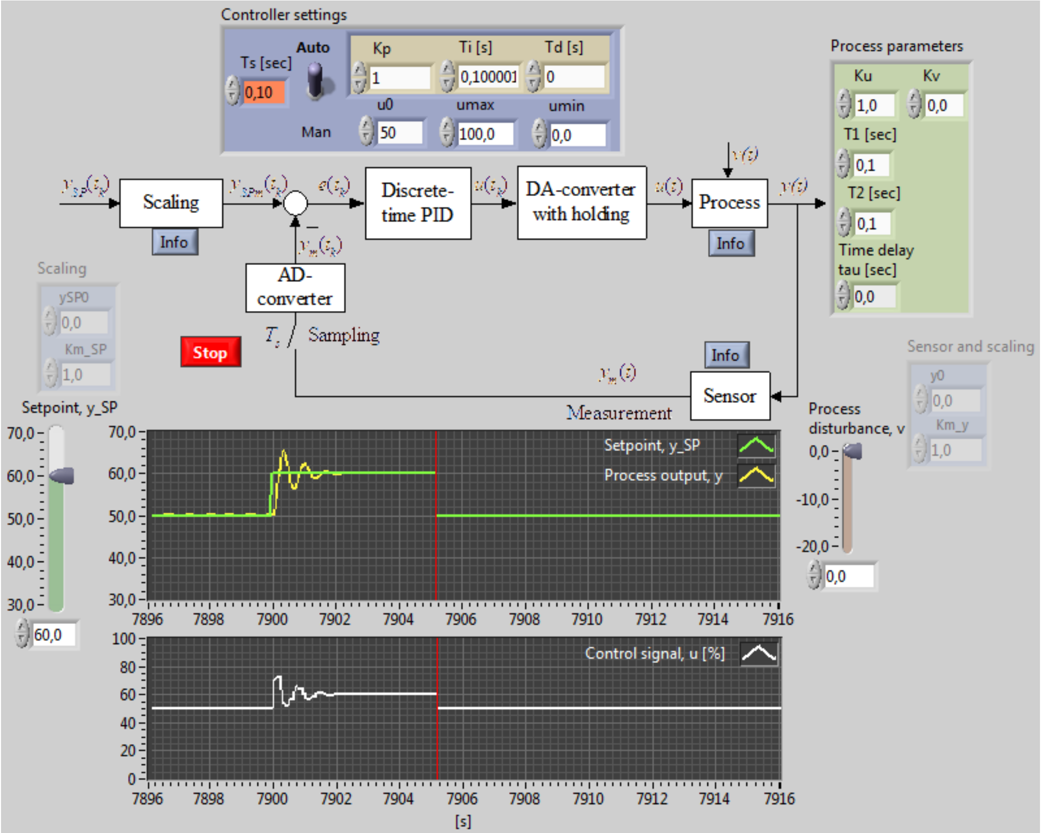
Regulator PI

Primer

- Enačba procesa:
  - Za T1=T2 in Kv = 0 velja

$$y(t) = v(t) = K_u e^{-\frac{t-\tau}{T_1}} u(t-\tau)$$

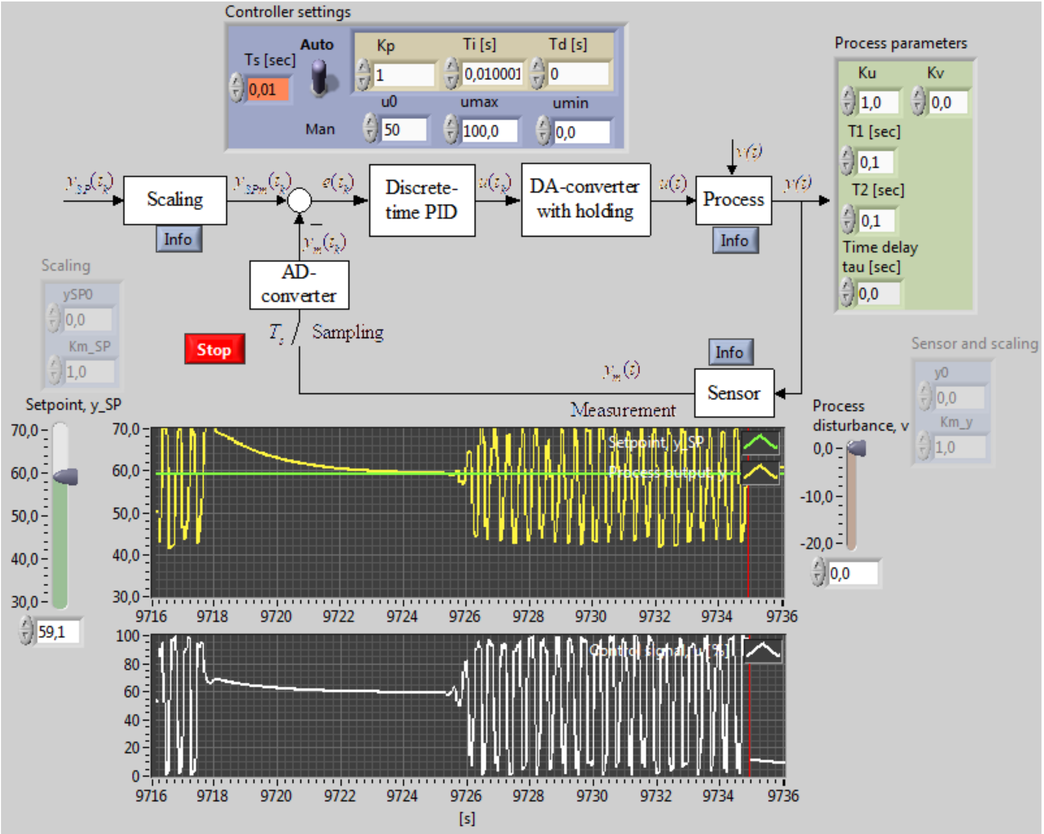
  - Izberemo T1 = 0,1 s, τ=0 s
- Poskus 4:
  - Nastavimo K_p = 1, T_d = 0
  - Stikalo postavimo na avtomatsko
  - Preverimo odziv za različne T_i:
    2 s, 0,5 s, 0,1 s, 0,04 s
  - Opazujemo dogajanje ob spremembi Setpoint
  - Opažanja:
    - Ni več težav z napak v mirovanju
    - Manjši T_i prinese hitrejši odziv in tudi manj stabilen sistem



Regulator PI

Primer

- Poskus 5:
  - Proporcionalni člen naj bo zanemarljiv v primerjavi z integrirnim:
    - K_p = 1, T_i = 0,01 s
  - V avtomatskem načinu opazujemo dogajanje ob spremembi Setpoint
  - Opažanja:
    - Samo integrirni člen je neuporaben, saj v mnogih primerih kompenzacija napake ni mogoča
    - Če bi pri poskusu P lahko popolnoma odpravili, bi se izkazalo, da se oscilacije ojačujejo
    - Pomembno vlogo igra čas vzorčenja
    - Zaradi pretiranega nihanja in nekontrolirane nestabilnosti nevarnost strojeloma



Regulator PI

Z integrirnim členom si regulator zapomni kaj se je dogajalo v preteklosti

- Na ta način lahko odpravi napako v mirovanju

Dolgotrajni spomin privede tudi do nestabilnosti

- Zaradi inercije, ki jo uvede integrirni člen, se regulator prepočasi odziva na spremembe

Regulator I skoraj nikoli ne nastopa samostojno, vedno samo kot dodatek k regulatorju P

- Za stabilizacijo procesa vedno rabimo prispevek proporcionalnega člena

Da ne pride do težav zaradi zasičenja, je potrebno integrirni člen omejiti

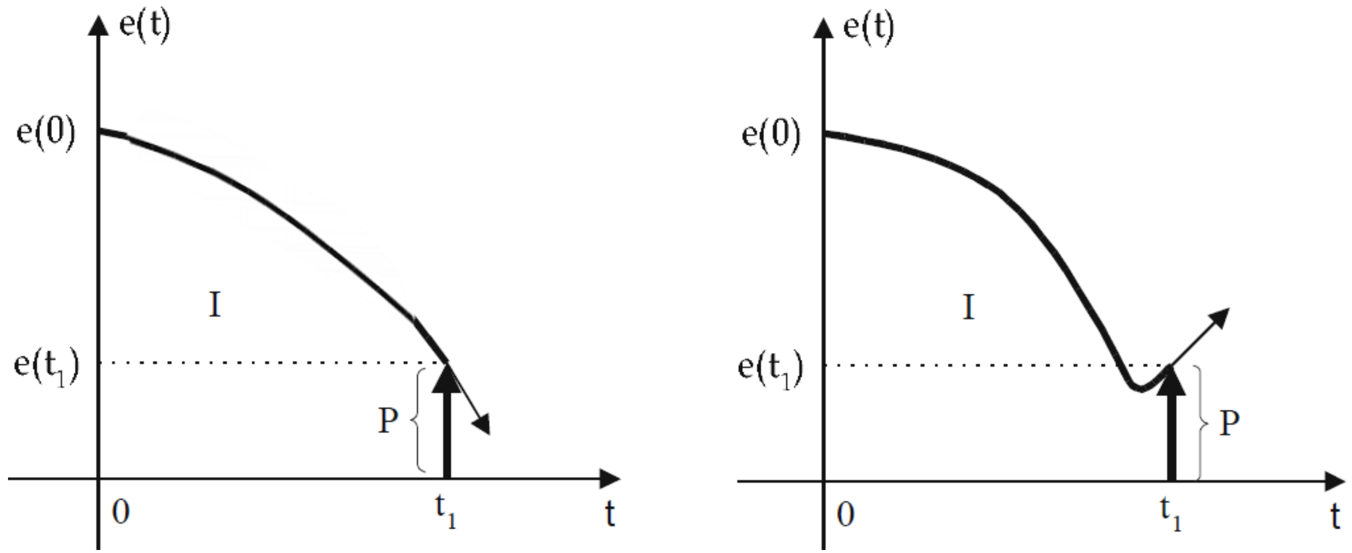
- Enostavna rešitev je, da z minimalno in maksimalno dovoljeno vrednostjo omejimo vrednost intergrala

V mnogih primerih je odziv prepočasen → dodamo še diferencirni člen

Regulator PI

Primerjava dveh potekov

- proporcionalni in integralni člen sta v obeh primerih enaka
- razlika je samo v smeri gibanja napake v času t_1

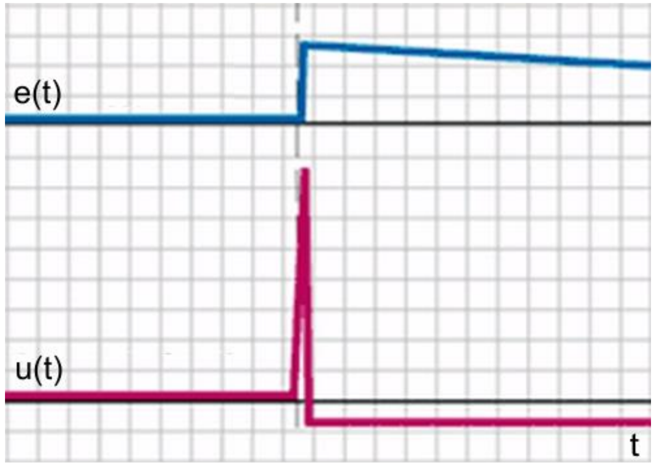


- Regulator PI ne zazna razlike med levim in desnim potekom in se bo v obeh primerih odzval enako
- Smer gibanja napake (napoved) nam pove tangenta na krivuljo oziroma odvod

Diferencirni člen

Napaka in njen odvod

- Večja kot je sprememba napake, večji je odziv
- Velikost odziva ustreza naklonu premice na signalu napake
- Z naklonom premice si lahko pomagamo pri napovedovanju naslednje vrednosti



Diferencirni člen

- $u(t) = K_d \frac{de(t)}{dt}$
- Če proporcionalni člen skrbi za odziv na trenutno stanje in integrirni člen za odziv na dogajanje v preteklosti, poskušamo z diferencirnim členom predvideti dogajanje v prihodnosti

Regulator D

Ideja: ob nenadni spremembi lahko z njim povzročimo močno korekcijo regulacijskega signala in tako v kar največji meri odstranimo napako

- Na ta način mnogim procesom močno povečamo odzivnost
- Ta popravek navadno
  - je v pravi smeri,
  - ni najbolj točen,
  - dosežemo ga mnogo hitreje kot z regulatorjem PI

Eden največjih problemov diferencirne kontrole je šum

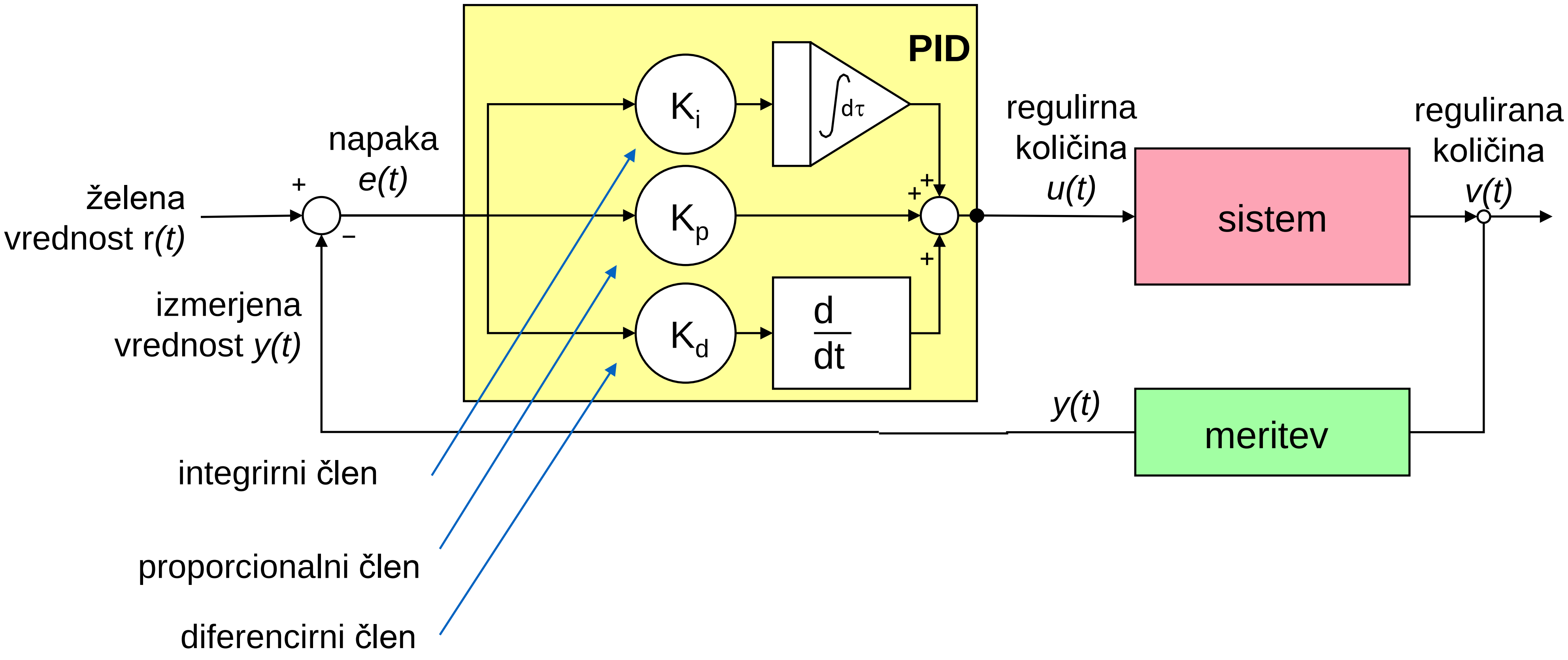
- Regulacije P šum ne moti, regulacija I ga s povprečenjem nevtralizira
- Regulacija D ojači nenadne spremembe in s tem tudi visoko frekvenčni šum

Regulator D nikoli ne nastopa samostojno, saj ne more odstraniti napake

Po prehodnem obdobju vpliv diferencirnega člena zamre, regulacijo lahko dobro zaključimo v kombinaciji z regulatorjem PI

Regulator PID

Enačba: $u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{de(t)}{dt}$



Regulator PID

Regulator PID se lahko zapiše nekoliko drugače:

- $$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{de(t)}{dt}$$
  $$u(t) = K_p \left[e(t) + \frac{1}{T_i} \int_0^t e(\tau)d\tau + T_d \frac{de(t)}{dt} \right]$$
- Vrednost diferencirnega člena v času t je proprocionalna napovedi gibanja napake v času $t + T_d$
- Večji kot je T_d , bolj daleč v prihodnost se gleda dogajanje

Regulator PID ima vso potrebno dinamiko

- primerno obnašanje znotraj kontrolnega območja $|e(t)| < e_0$ za odstranjevanje oscilacij (P),
- kompenzacijo napake za gibanje proti referenčni točki (I),
- hitro reakcijo na spremembe napake (D).

Prepočasen odziv regulatorja PI pohitrimo z dodatkom člena D

Regulator PID

Vpliv diferencirnega člena na regulacijo

- Primer
- Proces: $y(t) = v(t) = K_u e^{-\frac{t-\tau}{T_1}} u(t-\tau)$
  - T1 = T2 = 1 s, τ=0 s, Ku = 1, Kv = 0
- Nastavitve regulatorja PID
  - K_p = 4, T_i = 2 s
  - T_d: 0 s, 0,5 s, 1 s, 2 s, 5 s, 10 s
- Spreminjamo referenco in opazujemo signal
- Opažanja
  - Manjše, od nič različne vrednosti T_d zmanjšajo oscilacije in pohitrijo stabilizacijo procesa okrog nove referenčne vrednosti
  - Večje vrednosti T_d v sistem vnesejo dodatne oscilacije

Regulator PID

Težave

- Integrirni in diferencirni člen v sistem uvedeta nestabilnost
- Kompleksnost
  - Kljub temu, da razumemo, kaj je naloga vsakega člena posebej, je zelo težko nastaviti vse tri parametre tako, da bodo delovali usklajeno
  - Nastaviti jih moramo za vsako aplikacijo posebej – večkrat je težko izmeriti vse potrebne parametre za dobro nastavitev
- Sreča v nesreči
  - 90 % aplikacij v industriji deluje dobro kljub temu, da parametri niso optimalno nastavljeni

Regulator PID

Zvezno in diskretno

- Danes je večina regulatorjev PID izvedena programsko v mikro-krmilnikih
- Diskretizacija enačb
  - $\int e(t)dt \rightarrow \sum e(t)T_s$
  - $\frac{de(t)}{dt} \rightarrow \frac{\Delta e}{T_s}$
- Na integralni in diferencialni člen ima zelo močan vpliv čas vzorčenja
  - Čas vzorčenja naj bo med 1/10 in 1/100 želenega časa ustalitve signala na referenčni vrednosti, v primerih PI lahko tudi 1/1000
  - Čas je krajši, če
    - Nadziramo kompleksen proces,
    - Obstaja potreba po diferencirnem členu ali
    - Obstaja potreba po zelo natančni regulaciji
  - Prekratek čas vzorčenja v povezavi s šumom lahko povzroči velika nihanja diferencirnega člena

Regulator PID

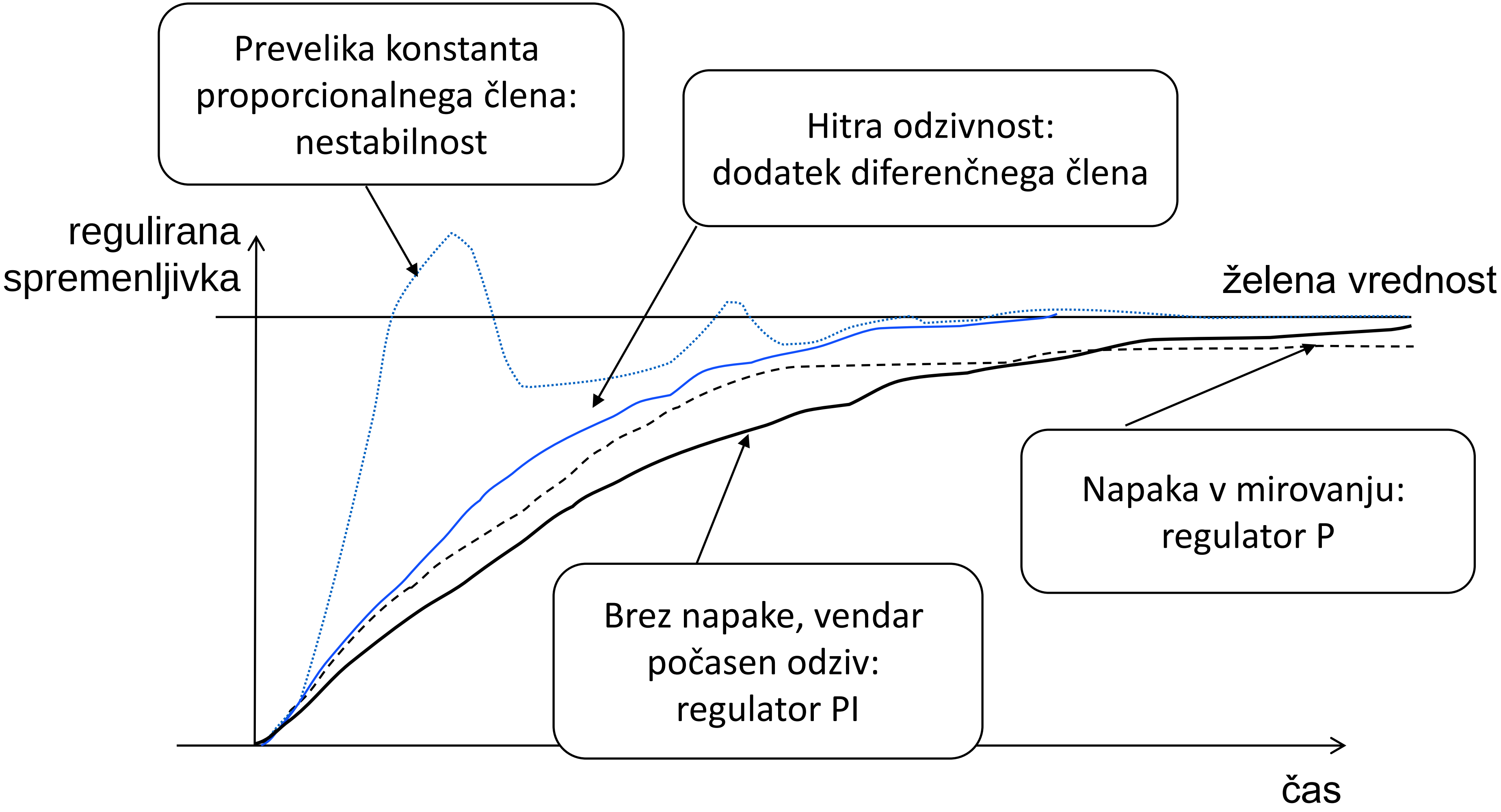
Zvezno in diskretno

- Enostavna izvedba diskretnega regulatorja PID

```
integral = 0    # začetna vrednost integrala
eOld = r - y    # razlika med referenco in meritvijo

while True:
     # razlika med referenco in meritvijo
     e = r - y

     # proporcionalni člen
     P = Kp * e;

     # integrirni člen
     integral = integral + e * Ts
     if integral > INTEGRAL_MAX:
        integral = INTEGRAL_MAX
     if integral < INTEGRAL_MIN:
        integral = INTEGRAL_MIN
     I = Ki * integral              # Ki = Kp / Ti

     # diferencirni člen
     D = Kd * (e - eOld) / Ts       # Kd = Kp * Td
     eOld = e

     # nova vrednost regulirne količine
     u = P + I + D

     # počakajmo na novo vzorčenje
     time.sleep(Ts)
```

Regulator PID

Kako nastaviti regulator PID?



Regulator PID

Kako nastaviti regulator PID?

- Vpliv povečanja konstante na sistem

Konstanta	Odziv	Prenihaj	Čas umirjanja	Napaka v ravnovesju	Stabilnost
K_p	Hitrejši	Povečanje	Majhen vpliv	Manjša	Slabša
K_i	Hitrejši	Povečanje	Daljši	Odpravljena	Slabša
K_d	Malo hitrejši	Zmanjšanje	Krajši	Nima vpliva	Boljša pri majhnih K_d

Regulator PID

Teoretični pristop

- Analiza odziva na testne funkcije
  - Regulator P za določitev K_p
  - Regulator PI za določitev T_i
  - Regulator PD za določitev T_d
- Običajno nimamo možnosti, da se igramo s sistemom

Regulator PID

Pristop s poskušanjem

- Vse člene postavimo na nič, nato pa jih dodajamo postopno
  - Proporcionalni člen
    - Konstanto K_p postavimo na tako vrednost, da sistem oscilira
    - Nato konstanto zmanjšujemo za faktor 8 do 10 dokler oscilacije ne izzvenijo
    - Z manjšimi spremembami (faktor 2) poiščemo všečen odziv
  - Integrirni člen
    - Začnemo z majhnimi vrednostmi T_i, na primer 1000, nato pa s faktorji 10 vrednost zmanjšujemo dokler odziv ni všečen
  - Diferencirni člen
    - Poiščemo točko kjer sistem začne oscilirati, nato T_d postavimo za faktor 10 nižje
    - Nadaljnje popravke delamo za faktor 2 navzgor ali navzdol
    - Če vemo, da bomo potrebovali diferencirni člen, ga nastavimo najprej

Regulator PID

Pristop s poskušanjem

- Primer
- Proces: $y(t) = v(t) = K_u e^{-\frac{t-\tau}{T_1}} u(t-\tau)$
  - T1 = T2 = 1 s, τ=0 s, Ku = 1, Kv = 0
- Postopek:
  - K_p:
    - Oscilacije opazimo pri K_p = 30
    - Pri K_p = 3 izzvenijo, odziv je OK
  - T_i:
    - 1000 s, 100 s, 10 s: prepočasi
    - 1 s: OK, 2 s : še bolje
  - T_d:
    - 40 s: zaznamo oscilacije,
    - 4 s: še ni OK
    - 0,4: še bolje

Regulator PID

Hevristične metode

- Dobro je poznati model procesa
- Z metodami določimo začetno nastavitev parametrov, nato pa jih s poskušanjem še fino nastavimo
- Nekaj metod:
  - Odprtozančni postopek z opazovanjem prenosnih funkcij procesa
    - Ziegler-Nichols, Cohen-Coon, Chien-Hrones-Reswick
  - Zaprtozančni postopek na podlagi opazovanja oscilacij procesa
    - Ziegler-Nichols

Regulator PID

Hevristične metode: zaprto zančni postopek po Ziegler-Nicholsu

- Ne predpostavlja modela procesa
- Najprej uporabimo samo proporcionalni člen
  - konstanto K_p povečujemo, dokler sistem ne začne oscilirati s konstantno periodo in amplitudo
  - Dobimo skrajno vrednost (ang. ultimate) proporcionalne konstante, K_u in periodo T_u
- Na podlagi zahtev določimo prve približke konstant iz spodnje tabele

Pravilo	K_p	T_i	T_d
Klasični Ziegler-Nichols	0,6 K_u	0,5 T_u	0,125 T_u
Pessenovo integracijsko pravilo	0,7 K_u	0,4 T_u	0,15 T_u
Manjši prenihaj	0,33 K_u	0,5 T_u	0,33 T_u
Brez prenihaja	0,2 K_u	0,5 T_u	0,33 T_u

  - Vir: A.S. McCormack, K.R. Godfrey: "Rule-Based Autotuning Based on Frequency Domain Identification", *IEEE Transactions on Control Systems Technology*, 6 (1), 1998

Regulator PID

Hevristične metode: zaprto zančni postopek po Ziegler-Nicholsu

- [Primer]
- Proces: $y(t) = v(t) = K_u e^{-\frac{t-\tau}{T_1}} u(t-\tau)$
  - T1 = T2 = 1 s, τ=0 s, Ku = 1, Kv = 0
- Postopek:
  - T_i = Inf, T_d = 0
  - Oscilacije opazimo pri K_p = 30 → K_u = 30
  - Približno 16 nihajev v 20 s → T_u = 20/16 = 1,25 s
  - Želimo sistem brez prenihaja
    - K_p = 0,2 x 30 = 6
    - T_i = 0,5 x 1,25 s ≈ 0,6 s
    - T_d = 0,33 x 1,25 s ≈ 0,4 s
  - Prenihaj je še prisoten → povečamo integracijski čas na 1,2 s in potem še na 2,4 s

Regulator PID

Primerjava parametrov

Metoda	K_p	T_i	T_d
Poskušanje	3	2	0,4 s
Ziegler-Nichols	6	2,4 s	0,4 s

- Precej različne vrednosti K_p, rezultat v obeh primerih zadovoljiv