OPTIMALNO UPRAVLJANJE

Žarko Zečević
Elektrotehnički fakultet
Univerzitet Crne Gore

Tema 4

Dinamičko programiranje

Ishodi učenja:

Nakon savladavanja gradiva sa ovog predavanja studenti će moći da:

- Primjenom dinamičkog programiranja odrede optimalnu upravljačku sekvencu koja će diskretni sistem voditi po optimalnoj trajektoriji
- Riješe Belmanovu i Hamilton-Jakobi-Belmanovu (HJB) jednačinu u zatvorenoj formi za kontinualne/diskretne LTI sisteme
- Simulacijama riješe Rikatijevu diferencijalnu/diferencnu jednačinu i odrede optimalno upravljanje na ograničenom horizontu
- Interpretiraju LQR rješenje sa stanovišta klasične teorije upravljanja

Uvod u optimalno upravljanje

Problem optimalnog upravljanja se odnosi na pronalaženje upravljačkog signala $\mathbf{u}^*(t) \in U$ koji sistem

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

vodi po trajektoriji $\mathbf{x}^*(t) \in X$, takvoj da bude minimizovana funkcija performanse:

$$J = h(\mathbf{x}(t_f), t_f) + \int_{t_0}^{t_f} g(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau) d\tau.$$

U prethodnoj definiciji U predstavlja skup dopustivih vrijednosti upravljačkog signala, dok je X skup dopustivih vrijednosti promjenljivih stanja. U opštem slučaju funkcije g i h su proizvoljne skalarane funkcije. Simbol * označava da je par $(\mathbf{u}^*(t),\ \mathbf{x}^*(t))$ optimalan, odnosno da minimizuje zadati kriterijum performanse.

Funkcija performanse se bira u skladu sa konkretnom primjenom, dok su skupovi U i X najčešće definisani u skladu sa fizičkim ograničenjima implementacije.

Uvod u optimalno upravljanje

Problem minimalnog vremena:

Na primjer, ako želimo da sistem prevedemo iz početnog stanja $\mathbf{x}(t_0)$ u stanje $\mathbf{x}(t_f)$ za najkraće moguće vrijeme, funkciju performanse možemo definisati na sljedeći način:

$$J = \int_{t_0}^{t_f} d\tau.$$

Problem krajnjeg stanja:

Sa druge strane, ako želimo da minimizujemo razliku između krajnjeg stanja $\mathbf{x}(t_f)$ i njegove željene vrijednosti $\mathbf{r}(t_f)$, tada funkciju performanse možemo definisati kao:

$$J = \left(\mathbf{x}(t_f) - \mathbf{r}(t_f)\right)^2 = \left(\mathbf{x}(t_f) - \mathbf{r}(t_f)\right)^T \left(\mathbf{x}(t_f) - \mathbf{r}(t_f)\right).$$

U nešto generalnijem slučaju, funkcija performanse ima oblik:

$$J = \left(\mathbf{x}(t_f) - \mathbf{r}(t_f)\right)^2 = \left(\mathbf{x}(t_f) - \mathbf{r}(t_f)\right)^T \mathbf{H} \left(\mathbf{x}(t_f) - \mathbf{r}(t_f)\right),$$

gdje je $\mathbf{H}$ realna, simetrična, semi-pozitivno definitna matrica.

Uvod u optimalno upravljanje

Problem minimalnog upravljanja:

Dalje, ako je cilj da minimizujemo energiju koju ćemo uložiti u upravljanje, tada funkciju performanse definišemo na sljedeći način:

$$J = \int_{t_0}^{t_f} \mathbf{u}^T(t)\mathbf{R}\mathbf{u}(t)dt,$$

gdje $\mathbf{R}$ predstavlja realnu, simetričnu, pozitivno defintnu matricu.

Problem praćenja referentnog signala:

Ako želimo da stanja $\mathbf{x}(t)$ što vjernije prate referentni signal $\mathbf{r}(t)$ na čitavom vremenskom intervalu $[t_0,\ t_f]$, tada funkciju performanse zapisujemo na sljedeći način:

$$J = \int_{t_0}^{t_f} \left(\mathbf{x}(t) - \mathbf{r}(t)\right)^T \mathbf{Q} \left(\mathbf{x}(t) - \mathbf{r}(t)\right) dt.$$

Ukoliko je $\mathbf{r}(t)=0$ (problem regulacije), tada se prethodno integral svodi na:

$$J = \int_{t_0}^{t_f} \mathbf{x}(t)^T \mathbf{Q}\mathbf{x}(t)dt.$$

Uvod u optimalno upravljanje

Prethodno definisani kriterijum performanse predstavlja dobar izbor ukoliko je dozvoljeno upravljanje ograničeno (na primjer $|\mathbf{u}(t)|<1$). Međutim, ukoliko ne postoje ograničenja ulaznog signala, minimizacijom prethodnog integrala dobićemo upravljački signal u obliku impulsa i njegovih izvoda. Da bi izbjegli postavljanje ograničenja na upravljanje, i time lakše riješili optimizacioni problem, funkciju performanse možemo definisati na sljedeći način:

$$J = \int_{t_0}^{t_f} \left[\left(\mathbf{x}(t) - \mathbf{r}(t) \right)^T \mathbf{Q} \left(\mathbf{x}(t) - \mathbf{r}(t) \right) + \mathbf{u}(t)^T \mathbf{R} \mathbf{u}(t) \right] dt.$$

Na ovaj način ujedno minimizujemo razliku između promjenljivih stanja i željenih vrijednosti, kao i snagu upravljačkog signala. Kako se prvi sabirak podintegralne veličine smanjuj povećavanjem upravljanja, kompromis između ova dva oprečna zahtjeva se može definisati izborom matrice $\mathbf{Q}$ i $\mathbf{R}$.

U nastavu će biti pokazano da se kod linearnih sistema, minimizacojom gornjeg kriterijuma dobija kontroler koji se može jednostavno implementirati.

Uvod u optimalno upravljanje

Napomene:

Matrice $\mathbf{Q}$ i $\mathbf{R}$ su najčešće dijagonale.

$$J = \int_{t_0}^{t_f} \left[\left(\mathbf{x}(t) - \mathbf{r}(t) \right)^T \mathbf{Q} \left(\mathbf{x}(t) - \mathbf{r}(t) \right) + \mathbf{u}(t)^T \mathbf{R} \mathbf{u}(t) \right] dt.$$

Odgovarajućim odabirom vrijednosti matrica $\mathbf{Q}$ i $\mathbf{R}$ zapravo definišemo prioritet minimizacije. Na primjer, ako nam je najbitnije da minimizujemo prvi upravljački signal, tada ćemo prvi element matrice $\mathbf{R}$ postaviti na najveću vrijednost.

Ako se matrice $\mathbf{Q}$ i $\mathbf{R}$ (semi)-pozitivno definitne, tada važi:

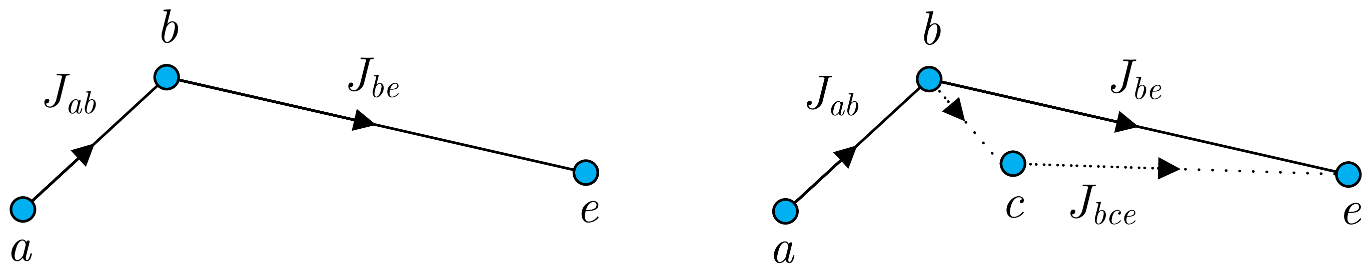
$$\mathbf{x}(t)^T \mathbf{Q} \mathbf{x}(t) \underset{(>)}{\geq} 0, \forall \mathbf{x}(t).$$

Drugim riječima, na ovaj način obezbjeđujemo da je integral koji minimizujemo uvijek ima bar jedan lokalni minimum.

Postoje dva pristupa u rješavanju optimalnih upravljačkih problema: dinamičko programiranje i varijaconi račun. U okviru ovog predavanja mi ćemo se baviti dinamičkim programiranjem.

Dinamičko programiranje

- Omogućava određivanje optimalnog zakona upravljanja
- Richard Ernest Bellman, 1953.
- Bazira se na **principu optimalnosti**



- Neka je $J^*_{ae} = J_{ab} + J_{be}$ optimalna putanja od čvora a do čvora e
- Tvrdnja: Ako je J^*_{ae} optimalna putanja, onda je i njen dio J_{be} optimalna putanja od čvora b do čvora e

Dokaz kontradikcijom

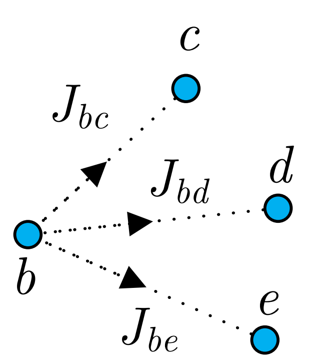
Pretpostavimo da je optimalna putanja od čvora b do čvora e putanja b-c-e. Tada je $J_{bce} < J_{be}$ tj. $J_{ab} + J_{bce} < J_{ab} + J_{be} = J^*_{ae}$. Ovo ne može biti tačno ukoliko je tačna pretpostavka da je J^*_{ae} optimalna putanja od čvora a do čvora e.

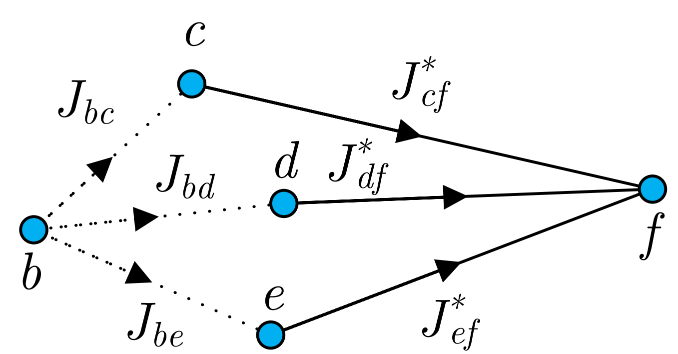
Dinamičko programiranje

Belmanov princip optimalnosti:

Optimalna strategija upravljanja sadrži svojstvo da bez obzira na početno stanje i odluke koje su dovele do trenutnog stanja, upravljanje na preostalom dijelu putanje treba biti optimalno u odnosu na trenutno stanje.

- Posmatrajmo neki proces čije je trenutno stanje b. Iz stanja b je dozvoljeno preći u neko od stanja c, d ili e. Optimalne putanje od stanja c, d i e do krajnjeg stanja f su poznate i prikazane na slici.
- Potrebno je donijeti odluku o prelasku u jedno od stanja c, d ili e, tako da putanja od stanja b do krajnjeg stanja f bude optimalna.





Dinamičko programiranje

Prema principu optimalnosti, ukoliko je b-c početni dio optimalne putanje od b do f, tada je i segment c-f terminalni dio ove putanje. Slično važi i za segmente b-d i b-e. Vrijednost funkcije performanse od stanja b do stanja f, ukoliko se ide preko stanja c je (*cost-to-go* funkcija):

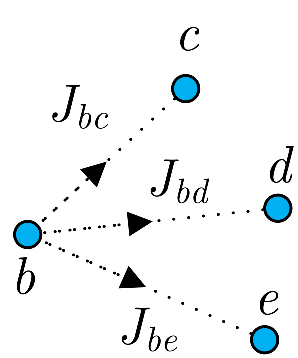
$$C_{bcf}=J_{bc}+J^*_{cf}.$$

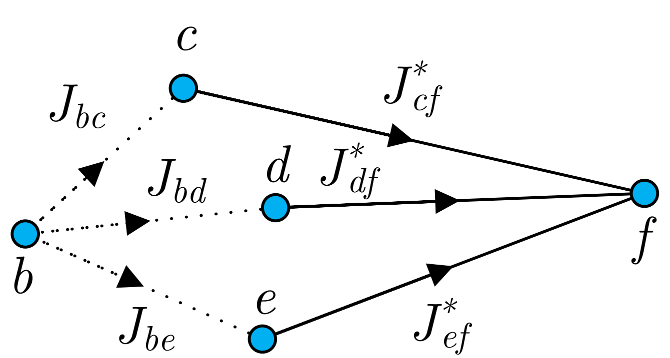
Slično, ukoliko se iz b do f ide preko d i e, *cost-to-go* funkcije će biti:

$$C_{bdf}=J_{bd}+J^*_{df},\ C_{bef}=J_{be}+J^*_{ef}.$$

Konačno, *optimalna cijena* od b do f se određuje kao minimum prethodnih *cost-to-go* funkcija:

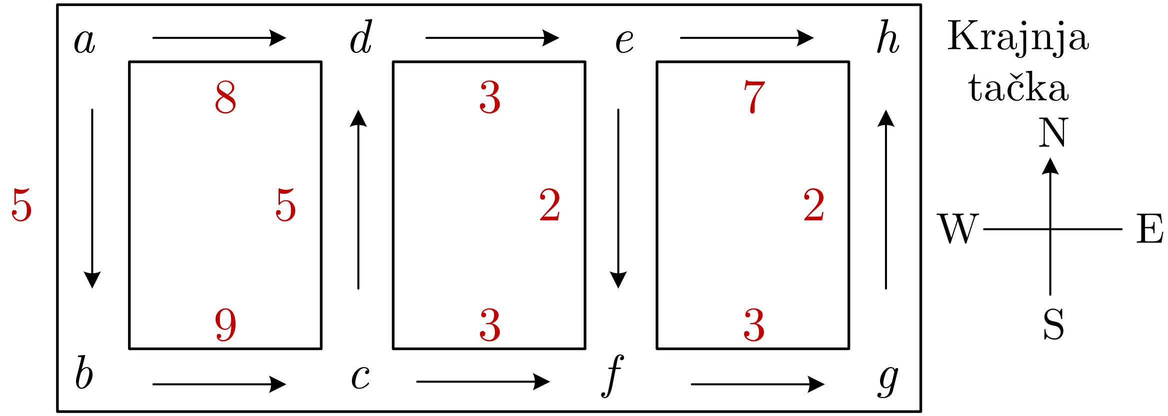
$$J^*_{bf}=\min\{C_{bcf},\ C_{bdf},\ C_{bef}\}.$$





Rješavanje problema rutiranja pomoću DP

- Dinamičko programiranje je računarska tehnika koja prethodno opisani postupak donošenja odluka proširuje na sekvencu stanja.
- Posmatrajamo problem rutiranja kod kojeg je potrebno odrediti optimalnu putanju od tačke a do tačke h. Dozvoljeno je kretanje samo po granama u smjeru strelica, pri čemu je cijena prelaska navedena pored grane.
- Jedan način rješavanja ovog problema bi bio u sistematičnom pretraživanju svih dozvoljenih putanja. Primjenom DP će se pretražiti daleko manji broj putanja.



Rješavanje problema rutiranja pomoću DP

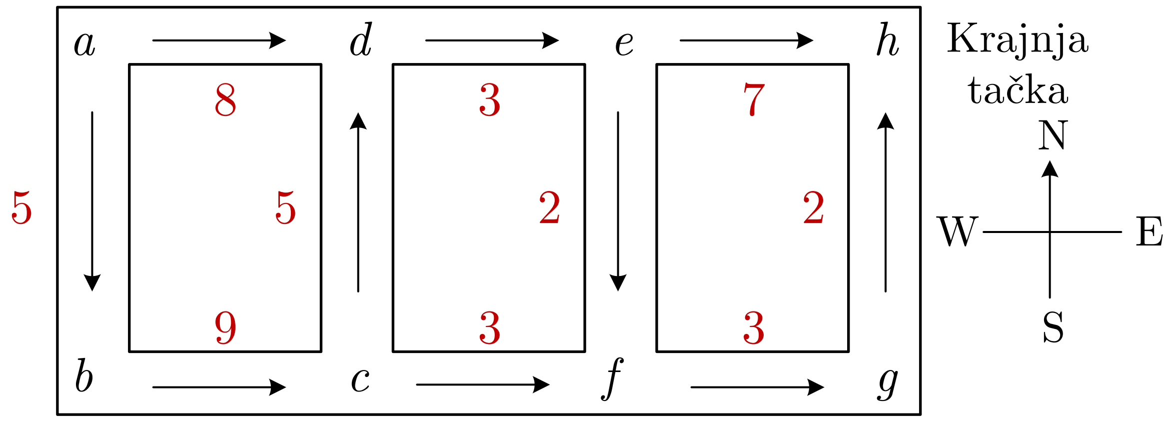
- Pretpostavimo da se nalazimo u tački c. Sa slike se vidi da je iz c moguće preći u stanja d i f. Vrijednosti *cost-to-go* funkcija će biti:

$$C_{cdh} = J_{cd} + J^*_{dh} = 5 + J^*_{dh},$$
$$C_{cfh} = J_{cf} + J^*_{fh} = 3 + J^*_{fh},$$

dok je optimalna cijena prelaska iz c u h određena sa:

$$J^*_{ch} = \min\{C_{cdh},\ C_{cfh}\}.$$

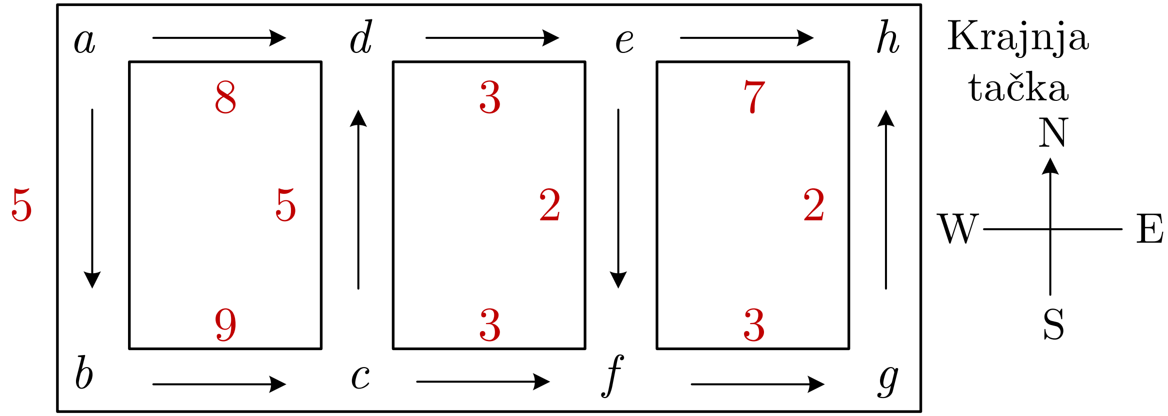
- Da bi se mogla donijeti odluka o prelasku u tačku d ili tačku f, odnosno odredila optimalna cijena prelaska J^*_{ch}, potrebno je znati vrijednosti J^*_{dh} i J^*_{fh}.



Rješavanje problema rutiranja pomoću DP

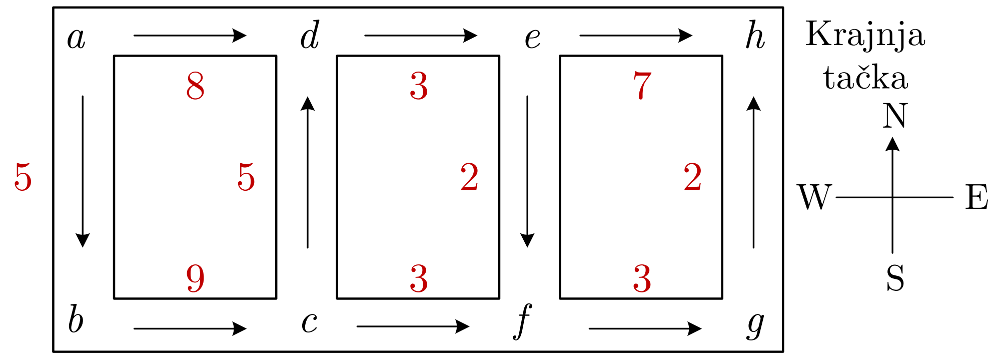
- Ako je $J^*_{dh} = 10$ i $J^*_{fh}=5$, tada su cost-to-go funkcije $C_{cdh}=5+10=15$ i $C_{cfh}=3+5=8$. Dalje, optimlana cijena prelasaka iz a u h je:
    - $J^*_{ch}=\min\{C_{cdh}, C_{cfh}\} = \min\{15,8\}=8$
- Međutim, postavlja se pitanje kako odrediti J^*_{dh} i J^*_{fh}. U suštini, problemu treba pristupiti unazad, počevši od krajnje tačke h. Iz tačke g se u tačku h može preći samo na jedan način, pa je $J^*_{gh} = 2$. Dalje, ova vrijednost se koristi za računanje J^*_{fh}:

$$J^*_{fh} = \min\{C_{fgh}\} = C_{fgh} = J_{fg} + J^*_{gh} = 3 + 2 = 5.$$



Rješavanje problema rutiranja pomoću DP

Rješenje problema rutiranja je prikazano ispod. Optimalna putanja iz tačake a u tačku h je a-d-e-f-g, pri čemu optimalna cijena prelaska iznosi 18.



a – trenutno stanje

u_i – dozvoljena odluka (upravljanje)

x_i – stanje u koje se prelazi iz stanja a, primjenom odluke u_i

h – krajnje stanje (cilj)

J_{ax_i} – cijena prelaska iz a u x_i

$J^*_{x_ih}$ – minimalna cijena dostizanja stanja h iz x_i

C_{ax_ih} – cijena prelaska iz a u h, ako se ide preko x_i

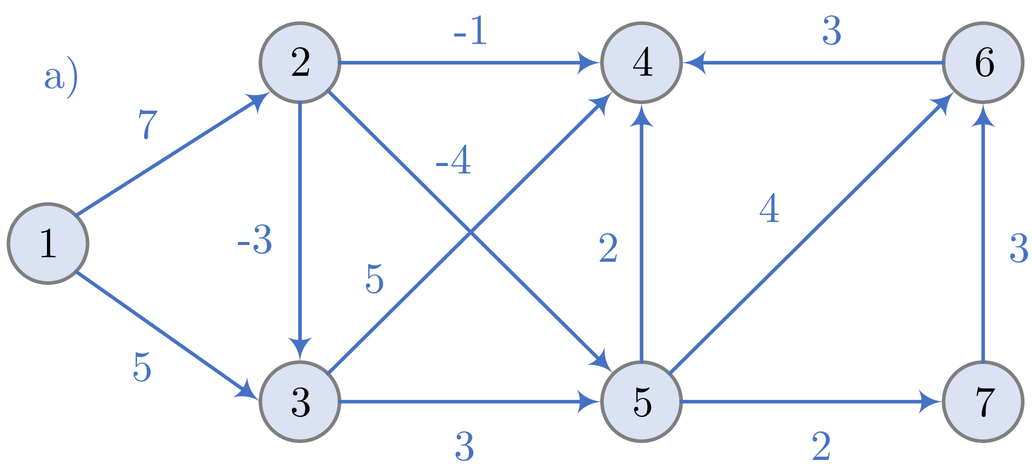
J^*_{ah} – minimalna cijena od a do h

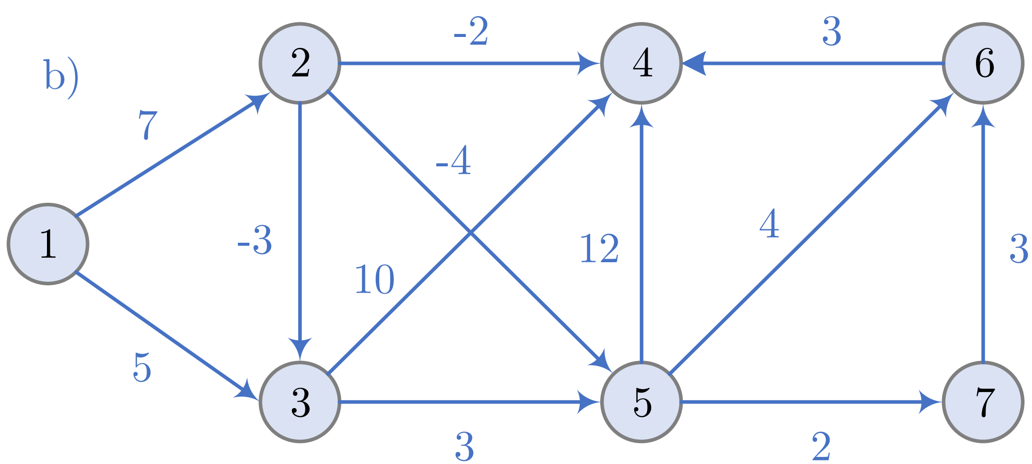
$u^*(a)$ – optimalna odluka iz stanja a

a	u_i	x_i	$J_{ax_i} + J^*_{x_ih} = C_{ax_ih}$	J^*_{ah}	$u^*(a)$
g	N	h	$2+0=2$	2	N
f	E	g	$3+2=5$	5	E
e	E S	h f	$8+0=8$ $2+5=7$	7	S
d	E	e	$3+7=10$	10	E
c	N E	d f	$5+10=15$ $3+5=8$	8	E
b	E	c	$9+8=17$	17	E
a	E S	d b	$8+10=18$ $5+17=22$	18	E

Primjer – pronalaženje optimalne rute

Odrediti optimalne putanje između bilo koja dva čvora grafa prikazanog na slici.





Rješavanje problema upravljanja pomoću DP

Neka je sistem opisan diferencijalnom jednačinom:

$$\dot{x}(t) = ax(t) + bu(t).$$

Odrediti upravljanje $u(t)$ koje minimizuje funkciju performanse:

$$J = x^2(t_f) + \lambda \int_0^{t_f} u^2(t)dt,$$

pri čemu su dozvoljene vrijednosti promjenljive stanja i upravljačkog signala:

$$0 \leq x(t) \leq 1.5, -1 \leq u(t) \leq 1.$$

Prije nego što se pristupi primjeni numeričke metode dinamičkog programiranja, potrebno je diskretizovati kontinualni sistem i funkciju performanse. Primjenom metode diferenciranja unazad dobija se sljedeća diferencna jednačina:

$$x(n+1) = (1 + aT)x(n) + bTu(n).$$

Na sličan način se diskretizuje funkcija performanse:

$$J = x^2(N) + \lambda T \sum_{n=0}^{N-1} u^2(n).$$

Rješavanje problema upravljanja pomoću DP

Radi jednostavnosti pretpostavimo da je $a = 0$, $b = 1$, $\lambda = 2$, $t_f = 2$, $T = 1$, ($N = 2$). Prethodne jednačine se svode na:

$$x(n+1) = x(n) + u(n), n = 0,\ 1,$$

gdje $u(1)$ i $u(2)$ treba odabrati tako da se minimizuje funkcija:

$$J = x^2(2) + 2u^2(0) + 2u^2(1),$$

uz ograničenja:

$$0 \leq x(n) \leq 1.5, -1 \leq u(n) \leq 1.$$

Sada treba primijeniti DP na sličan način kao kod problema rutiranja. Da bi ograničili broj potrebnih računarskih operacija, dozvoljene vrijednosti stanja i upravljanja treba kvantizovati. Usvojimo da je $x(n) \in \{0.0, 0.5, 1.0, 1.5\}$ i da je $u(n) \in \{$-1.0, -0.5, 0, 0.5, 1.0$\}$.

Procedura pronalaženja optimalnog upravljanja je sljedeća.

- Usvojiti da je n=1, a zatim za svaku moguću vrijednost stanja $x(1)$ isprobati svaku moguću vrijednost upravjanja $u(n)$ i izračunati minimalnu vrijednost cost funkcije.

Rješavanje problema upravljanja pomoću DP

Rezultujuća tabela za slučaj kad se kreće iz stanja $x(1)$

Trenutno stanje $x(1)$	Upravljanje $u(1)$	Sljedeće stanje $x(2)=x(1)+u(1)$	Cijena prelaska iz $x(1)$ u $x(2)$ primjenom $u(1)$ $x^2(2)+2u^2(1)=J_{12}(x(1),u(1))$	Minimalna cijena $J^*_{12}(x(1))$	Optimalno upravljanje $u^*(x(1),1)$
1.5	0.0 -0.5 -1.0	1.5 1.0 0.5	$(1.5)^2+2(0.0)^2=2.25$ $(1.0)^2+2(-0.5)^2=1.50$ $(0.5)^2+2(-1.0)^2=2.25$	$J^*_{12}(1.5)=1.50$	$u^*(1.5,1)=-0.5$
1.0	0.5 0.0 -0.5 -1.0	1.5 1.0 0.5 0.0	$(1.5)^2+2(0.5)^2=2.75$ $(1.0)^2+2(0.0)^2=1.00$ $(0.5)^2+2(-0.5)^2=0.75$ $(0.0)^2+2(-0.5)^2=0.50$	$J^*_{12}(1.0)=0.75$	$u^*(1.5,1)=-0.5$
0.5	1.0 0.5 0.0 -0.5	1.5 1.0 0.5 0.0	$(1.5)^2+2(1.0)^2=4.25$ $(1.0)^2+2(0.5)^2=1.50$ $(0.5)^2+2(0.0)^2=0.25$ $(0.0)^2+2(-0.5)^2=0.50$	$J^*_{12}(1.0)=0.25$	$u^*(1.5,1)=0.0$
0.0	1.0 0.5 0.0	1.0 0.5 0.0	$(1.0)^2+2(1.0)^2=3.00$ $(0.5)^2+2(0.5)^2=0.75$ $(0.0)^2+2(0.0)^2=0.00$	$J^*_{12}(1.0)=0.00$	$u^*(1.5,1)=0.0$

Rješavanje problema upravljanja pomoću DP

Rezultujuća tabela za slučaj kad se kreće iz stanja $x(0)$

Trenutno stanje $x(0)$	Upravljanje $u(0)$	Sljedeće stanje $x(1)=x(0)+u(0)$	Cijena pr. iz $x(0)$ u $x(2)$ $J_{01}(x(0),u(0))+J^*_{12}(x(1))=2u^2(0)+J^*_{12}(x(1))$	Minimalna cijena $J^*_{02}(x(0))$	Optimalno upravljanje $u^*(x(0),0)$
1.5	0.0 -0.5 -1.0	1.5 1.0 0.5	$2(0.0)^2+1.50=1.50$ $2(-0.5)^2+0.75=1.25$ $2(-1.0)^2+0.25=2.25$	$J^*_{02}(1.5)=1.25$	$u^*(1.5,0)=$-0.5
1.0	1.0 0.5 0.0 -0.5	1.5 1.0 0.5 0.0	$2(0.5)^2+1.50=2.00$ $2(0.0)^2+0.75=0.75$ $2(-0.5)^2+0.25=0.75$ $2(-1.0)^2+0.00=2.00$	$J^*_{02}(1.0)=\begin{Bmatrix}0.75\\0.75\end{Bmatrix}$	$u^*(1.0,0)=\begin{Bmatrix}0.0\\-0.0\end{Bmatrix}$
0.5	1.0 0.5 0.0 -0.5	1.5 1.0 0.5 0.0	$2(1.0)^2+1.50=3.50$ $2(0.5)^2+0.75=1.25$ $2(0.0)^2+0.25=0.25$ $2(-0.5)^2+0.00=0.50$	$J^*_{02}(0.5)=0.25$	$u^*(0.5,0)=0.0$
0.0	1.0 0.5 0.0	1.0 0.5 0.0	$2(1.0)^2+0.75=2.75$ $2(0.5)^2+0.25=0.75$ $2(0.0)^2+0.00=0.00$	$J^*_{02}(0.0)=0.00$	$u^*(0.0,0)=0.0$

Rješavanje problema upravljanja pomoću DP

- Dinamičko programiranje se sastoji od dvije faze:
  - Faza od kraja prema početku
  - Faza od početka prema kraju
- Tokom faze od kraja prema početku se određuju vrijednosti optimalnih cijena koje omogućavaju donošenje odluka o optimalnim prelazima. U opštem slučaju, optimalna cijena se određuje na osnovu izraza:

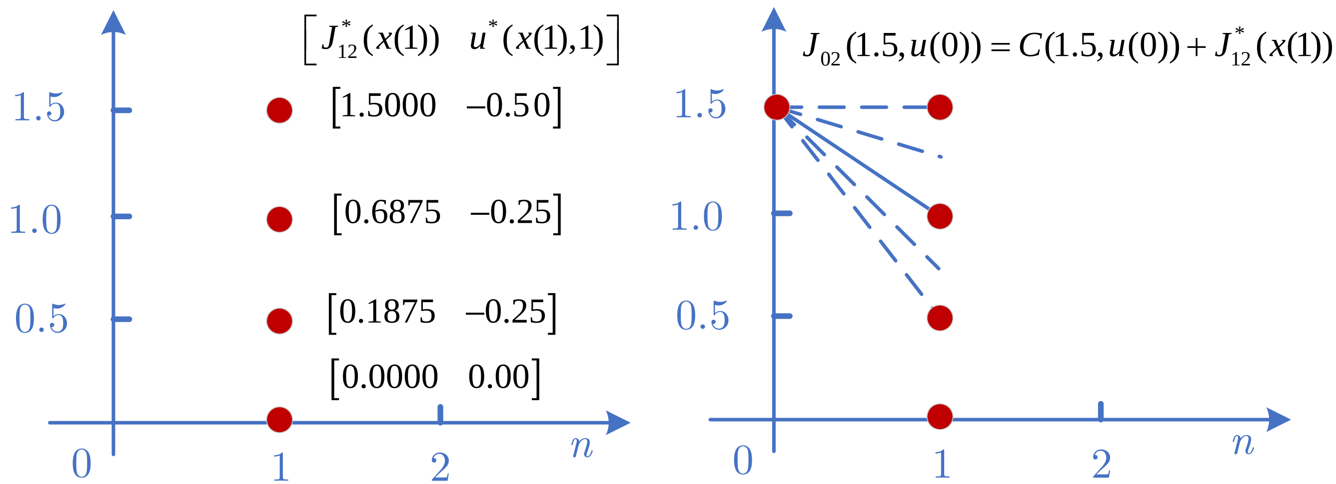
$$C_{n,N}(x(n),u(n)) = J_{n,n+1}(x(n),u(n)) + J^*_{n+1,N}(x(n+1))$$

$$J^*_{n,N} = \min_{u(n)}\left[C_{n,N}(x(n),u(n))\right]$$

- Tokom faze od početka prema kraju se donose odluke o optimalnim prelazima (optimalnim upravljanjima), te određuje optimalna putanja. Na primjer, u razmatranom primjeru, ako je početno stanje $x(0)=0.5$, tada je optimalno upravljanje jednako $u^*(n)=[0.0,\ 0.0]$, dok je optimalna trajektorija $u^*(n)=[0.5,\ 0.5]$. Konačno, optimalna cijena je jednaka $J^*_{02}(0.5)=0.25$.

Interpolacija

U analiziranom primjeru diskretizacija stanja i upravljanja je izvršena na takav način da svako upravljanje iz svakog stanja sistem prevodi u novo stanje čija vrijednost odgovara jednom od usvojenih diskretnih nivoa. Ukoliko diskretne vrijednosti nijesu pažljivo odabrane, potrebno je primijeniti interpolaciju.



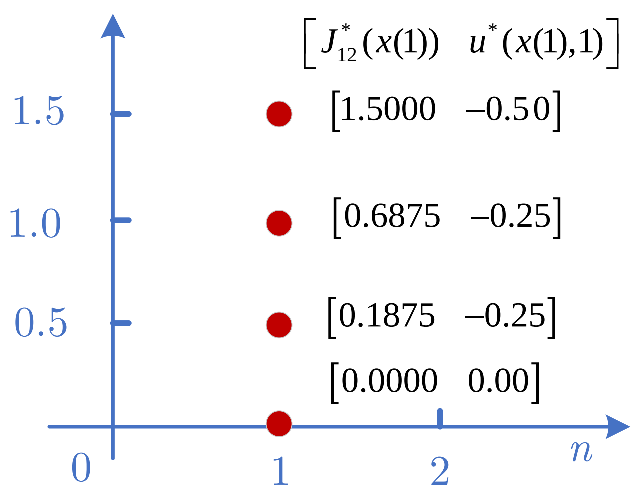
Na primjer, pretpostavimo da su upravljanja diskretizovana sa korakom 0.25: $u(n) \in \{$-1.0, -0.75 -0.5, -0.25, 0, 0.25, 0.5, 0.75, 1.0$\}$. Vrijednosti optimalnih cijena i upravljanja kada se kreće iz stanja $x(1)$ su prikazana na gornjoj slici, lijevo.

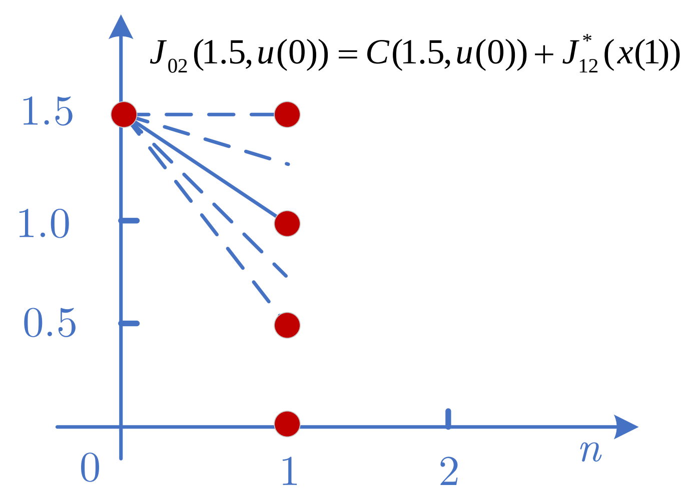
Interpolacija

Pretpostavimo da sada želimo da izračunamo optimalnu cijenu za slučaj kada se kreće iz stanja $x(0)=1.5$, pri čemu ispitujemo upravljanje $u(0)=-0.25$. Ovo upravljanje prevodi sistem u stanje $x(1)=1.25$. Da bi izračunali optimalnu cijenu potrebno poznavati $J^*_{12}(1.25)$. S obzirom da se $J^*_{12}(1.25)$ u tabeli za $x(1)$ nalazi između $J^*_{12}(1.5)$ i $J^*_{12}(1.0)$, primjenjuje se linearna interpolacja:

$$J^*_{12}(0.75) = J^*_{12}(1.0) - \frac{1}{2}\left(J^*_{12}(1.0) - J^*_{12}(0.5)\right) = 1.0938.$$

Interpolacija se na sličan način primjenjuje kada se iz memorije čita optimalno upravljanje za zadato početno stanje.





Rekurzivna jednačina DP

Sada ćemo generalizovati proceduru DP na sisteme većeg reda. Posmatrajmo sistem opisan vektorskom diferencnom jednačinom:

$$\mathbf{x}(n+1) = f_d\left(\mathbf{x}(n), \mathbf{u}(n)\right),$$

i funkciju performanse:

$$J = h(x(N)) + \sum_{n=0}^{N-1} g_d(\mathbf{x}(k), \mathbf{u}(k)).$$

Potrebno je da odredimo optimalnu sekvencu upravljanja:

$$\mathbf{u}^*(\mathbf{x}(0), 0), \mathbf{u}^*(\mathbf{x}(1), 1), \ldots, \mathbf{u}^*(\mathbf{x}(N-1), N-1).$$

Optimalno upravljanje zavisi od trenutnog stanja i vremenskog indeksa.

Označimo sa $J_{NN}(x(N))$ cijenu dostizanja krajnjeg stanja:

$$J_{NN}(x(N)) \triangleq h(x(N)).$$

Dalje, definišimo cijenu prelaska iz stanja $\mathbf{x}(N-1)$ u stanje $\mathbf{x}(N)$:

$$\begin{aligned} J_{N-1,N}(\mathbf{x}(N-1), \mathbf{u}(n-1)) &\triangleq g_d(\mathbf{x}(N-1), \mathbf{u}(n-1)) + h(\mathbf{x}(N)) \\ &= g_d(\mathbf{x}(N-1), \mathbf{u}(n-1)) + J_{NN}(\mathbf{x}(N)) \\ &= g_d(\mathbf{x}(N-1), \mathbf{u}(n-1)) + J_{NN}(f_d(\mathbf{x}(N-1), \mathbf{u}(N-1))). \end{aligned}$$

Vrijednost J-a za $n = N-1$

Rekurzivna jednačina DP

Optimalna cijena je jednaka:

$$J^*_{N-1,N}(\mathbf{x}(N-1)) \triangleq \min_{\mathbf{u}(n-1)} \{g_d(\mathbf{x}(N-1),\mathbf{u}(n-1)) + J_{NN}(f_d(\mathbf{x}(N-1),\mathbf{u}(N-1)))\}$$

Na sličan način se zapisuje optimalna cijena za zadnja dva koraka:

$$J^*_{N-2,N}(\mathbf{x}(N-1)) \triangleq \min_{\mathbf{u}(N-2),\mathbf{u}(N-1)} \left\{g_d(\mathbf{x}(N-2),\mathbf{u}(n-2)) + J_{N-1,N}(f_d(\mathbf{x}(N-1),\mathbf{u}(N-1)))\right\}$$

Princip optimalnosti nalaže da bez obzira na početno stanje $\mathbf{x}(N–2)$ i inicijalnu odluku $\mathbf{u}(N–2)$, ostale odluke moraju biti opitmalne. Odnosno:

$$J^*_{N-2,N}(\mathbf{x}(N-1)) \triangleq \min_{\mathbf{u}(n-2)} \left\{g_d(\mathbf{x}(N-2),\mathbf{u}(n-2)) + J^*_{N-1,N}(\mathbf{x}(N-1))\right\}.$$

Dalje, u K-om koraku možemo zapisati sljedeće:

$$J^*_{N-K,N}(\mathbf{x}(N-K)) \triangleq \min_{\mathbf{u}(N-K)} \left\{g_d(\mathbf{x}(N-K),\mathbf{u}(N-K)) + J^*_{N-(K+1),N}(\mathbf{x}(N-K+1))\right\}.$$

Posljednja jednačina predstavlja **rekurzivnu jednačinu dinamičkog programiranja** ili **Belmanovu jednačinu**. Poznavajući $J^*_{(N-(K-1),N)}$, odnosno optimalnu cijenu za zadnjih K-1 koraka, možemo odrediti optimalnu cijenu $J^*_{(N-K,N)}$ za zadnjih K koraka. Rekurzivna procedura započinje inicijalizacijom $J^*_{NN}=J_{NN}$.

Primjer 1 – optimalna upravljačka sekvenca

Sistem prvog reda je opisan diferencnom jednačinom:

$$x(n+1) = -0.5x(n) + u(n).$$

Funkcija performanse koju treba minimizovati ima oblik:

$$J = \sum_{n=0}^{2} |x(n)|,$$

dok su stanja i upravljanja ograničena sa:

$$-0.2 \leq x(n) \leq 0.2, \;\; n = 0,1,2,$$

$$-0.1 \leq u(n) \leq 0.1, \;\; n = 0,1.$$

a) Primjenom dinamičkog programiranja odrediti optimalnu sekvencu upravljanja. Kvantizovati $x(n)$ i $u(n)$ sa korakom 0.1. Ukoliko je potrebno koristiti linearnu interpolaciju.

b) Na osnovu rezultata dobijenog u a) odrediti optimalnu upravljačku sekvencu $\{u^*(1),\ u^*(2)\}$ i optimalnu cijenu, ukoliko je početno stanje 0.2?

Primjer 2 – optimalna upravljačka sekvenca

Sistem prvog reda je opisan diferencnom jednačinom:

$$x(n+1) = 0.75x(n) + u(n).$$

Funkcija performanse koju treba minimizovati ima oblik:

$$J = u^2(0) + u^2(1),$$

dok su stanja i upravljanja ograničena sa:

$$0 \le x(n) \le 6,\ \ n = 0,1,2,$$

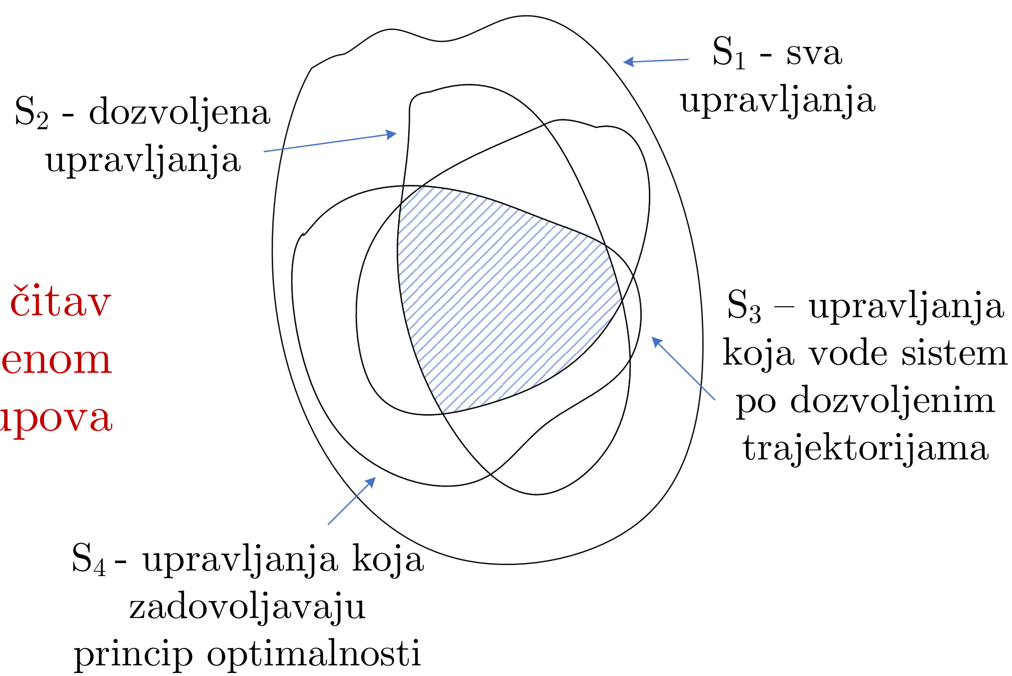
$$-1 \le u(n) \le 1,\ \ n = 0,1.$$

a) Primjenom dinamičkog programiranja odrediti optimalnu sekvencu upravljanja. Kvantizovati $x(n)$ i $u(n)$ sa korakom 0.5. Ukoliko je potrebno koristiti linearnu interpolaciju.

b) Na osnovu rezultata dobijenog u a) odrediti optimalnu upravljačku sekvencu $\{u^*(0),\ u^*(1)\}$ i optimalnu cijenu, ukoliko je početno stanje $x(0)=6$.

Karakteristike DP rješenja

- S obzirom da se koristi direktna pretraga za rješavanje rekurzivne jednačine, rješenje problema dobijeno pomoću DP predstavlja globalni minimum.
- Princip optimalnosti nameće dodatno ograničenje na dozvoljene vrijednosti upravljanja, čime se značajno smanjuje veličina skupa upravljanja koji je potrebno pretražiti. Procedura pretrage rješenja se dodatno pojednostavljuje ukoliko postoje ograničenja na upravljanje i varijable stanja.



Umjesto da pretažujemo čitav skup upravljanja S_1, primjenom DP se pretražuje presjek skupova S_2, S_3 i S_4

Karakteristike DP rješenja

- Optimalno upravljanje u svakoj iteraciji je zadato u formi povratne sprege po stanjima sistema: $\mathbf{u}(t)=\mathbf{f}(\mathbf{x}(t),t)$. Međutim, funkciju $\mathbf{f}$ je često nemoguće dobiti u zatvorenom obliku, već se optimalno upravljanje u tabelarnoj formi čuva u memoriji.
- Za sisteme većege reda broj vrijednosti koji je potrebno memorisati može biti jako veliki. Ovaj problem se naziva prokletstvo dimenzionalnosti. Na primjer, za sistem trećeg reda čija su stanja diskretizovana na 100 nivoa, za čuvanje vrijednosti dobijenih u samo jednom optimizacionom koraku potrebno je obezbijediti 10^6 memorijskih lokacija, što predstavlja 8MB memorijskog prostora. Ako smo sistem optimizovali na intervalu od 10s, pri čemu je korak odabiranja jednak 0.1s, to znači da je ukupno potrebno 8GB memorije.
- Razvijene su brojne tehnike za redukciju dimenzionalnosti optimizicaionog problema. Aproksimativno dinamičko programiranje je jedna od tih tehnika, međutim ona prevazilazi okvire ovog kursa.

LQR regulator

Diskretni LQR regulator

Posmatrajmo diskretni linearni sistem:

$$\mathbf{x}(n+1) = \mathbf{A}(n)\mathbf{x}(n) + \mathbf{B}(n)\mathbf{u}(n).$$

Smatraćemo da ne postoje nikakva ograničenja nad stanjima sistema i upravljačkim signalom. Problem se sastoji u pronalaženju optimalnog upravjanja koje minimizuje funkciju performanse zadate u obliku kvadratne forme:

$$J = \frac{1}{2}\mathbf{x}^T(N)\mathbf{H}\mathbf{x}(N) + \frac{1}{2}\sum_{n=0}^{N-1}\left[\mathbf{x}^T(n)\mathbf{Q}(n)\mathbf{x}(n) + \mathbf{u}^T(n)\mathbf{R}(n)\mathbf{u}(n)\right],$$

gdje su $\mathbf{H}$ i $\mathbf{Q}(n)$ realne simetrične semi-pozitivno definitne matrice dimenzija $n \times n$, dok je $\mathbf{R}(n)$ realna simetrična pozitivno definitna matrica dimenzija $m \times m$.

Dati problem možemo riješiti primjenom dinamičkog programiranja. Počnimo sa definisanjem cijene dostizanja krajnjeg stanja:

$$J_{NN}(\mathbf{x}(N)) = \frac{1}{2}\mathbf{x}^T(N)\mathbf{H}\mathbf{x}(N) \triangleq J^*_{NN}(\mathbf{x}(N)) \triangleq \frac{1}{2}\mathbf{x}^T(N)\mathbf{P}(N)\mathbf{x}(N).$$

Diskretni LQR regulator

Dalje, optimalna cijena prelaska iz stanja $\mathbf{x}(N\text{-}1)$ u stanje $\mathbf{x}(N)$ je:

$$J^{*}_{N-1,N}(\mathbf{x}(N-1)) \triangleq \min_{\mathbf{u}(N-1)} J_{N-1,N}(\mathbf{x}(N-1),\mathbf{u}(N-1))$$

Vrijednost J-a za $n=N$-1

$$= \min_{\mathbf{u}(N-1)} \left\{ \frac{1}{2}\mathbf{x}^T(N-1)\mathbf{Q}\mathbf{x}(N-1) + \frac{1}{2}\mathbf{u}^T(N-1)\mathbf{R}\mathbf{u}(N-1) \right.$$

$$\left. + \frac{1}{2}\underbrace{\left[\mathbf{A}\mathbf{x}(N-1)+\mathbf{B}\mathbf{u}(N-1)\right]^T}_{\mathbf{x}(n)^T} \mathbf{P}(N) \underbrace{\left[\mathbf{A}\mathbf{x}(N-1)+\mathbf{B}\mathbf{u}(N-1)\right]}_{\mathbf{x}(n)} \right\}.$$

S obzirom da su $\mathbf{Q}$ i $\mathbf{R}$ (semi)-pozitivno definitne matrice, prethodna funkcija je konveksna i ima jedan globalni minimum. Optimalno upravljanje $\mathbf{u}(N-1)$ možemo pronaći diferenciranjem prethodnog izraza:

$$\frac{\partial J_{N-1,N}}{\partial \mathbf{u}(N-1)} = \mathbf{R}\mathbf{u}(N-1) - \mathbf{B}^T\mathbf{P}(N)\left[\mathbf{A}\mathbf{x}(N-1)+\mathbf{B}\mathbf{u}(N-1)\right] = \mathbf{0}$$

$$\rightarrow \mathbf{u}^{*}(N-1) = -\left[\mathbf{R}+\mathbf{B}^T\mathbf{P}(N)\mathbf{B}\right]\mathbf{B}^T\mathbf{P}(N)\mathbf{A}\mathbf{x}(N-1) \triangleq -\mathbf{K}(N-1)\mathbf{x}(N-1).$$

Diskretni LQR regulator

Uvrštavanjem $\mathbf{u}^*(N-1)$ u izraz za $J^*_{N\text{-}1,N}$ dobija se optimalna cijena:

$$J^*_{N-1,N}(\mathbf{x}(N-1)) = \frac{1}{2}\mathbf{x}^T(N-1)\Big\{\big[\mathbf{A}+\mathbf{BK}(N-1)\big]^T \mathbf{P}(N)\big[\mathbf{A}+\mathbf{BK}(N-1)\big]\big)$$

$$+\mathbf{K}^T(N-1)\mathbf{RK}(N-1)+\mathbf{Q}\Big\}\mathbf{x}(N-1) \triangleq \frac{1}{2}\mathbf{x}^T(N-1)\mathbf{P}(N-1)\mathbf{x}(N-1).$$

Ako nastavimo da rješavamo optimalni problem unazad, dobićemo rješenje u istom obliku. Odnosno:

$$\mathbf{u}^*(N-2) = -\Big[\mathbf{R}+\mathbf{B}^T\mathbf{P}(N-1)\mathbf{B}\Big]\mathbf{B}^T\mathbf{P}(N-1)\mathbf{A}\mathbf{x}(N-2) \triangleq -\mathbf{K}(N-2)\mathbf{x}(N-2),$$

dok je optimalna cijena:

$$J^*_{N-2,N}(\mathbf{x}(N-1)) \triangleq \frac{1}{2}\mathbf{x}^T(N-1)\mathbf{P}(N-2)\mathbf{x}(N-1),$$

pri čemu je $\mathbf{P}(N-2)$:

$$\mathbf{P}(N-2) = \big[\mathbf{A}+\mathbf{BK}(N-1)\big]^T\mathbf{P}(N-1)\big[\mathbf{A}+\mathbf{BK}(N-1)\big]+\mathbf{K}^T(N-1)\mathbf{RK}(N-1)+\mathbf{Q}.$$

Diskretni LQR regulator

Optimalna vrijednost upravljanja u m-tom koraku je:

$$\mathbf{u}^{*}(N-m) = -\left[\mathbf{R} + \mathbf{B}^{T}\mathbf{P}(N-m)\mathbf{B}\right]\mathbf{B}^{T}\mathbf{P}(N-m)\mathbf{A}\mathbf{x}(N-m)$$
$$\triangleq -\mathbf{K}(N-m)\mathbf{x}(N-m),$$

pri čemu je:

$$\mathbf{P}(N-m) = \left[\mathbf{A} + \mathbf{B}\mathbf{K}(N-m)\right]^{T}\mathbf{P}(N-m+1)\left[\mathbf{A} + \mathbf{B}\mathbf{K}(N-m)\right]$$
$$+\mathbf{K}^{T}(N-m)\mathbf{R}\mathbf{K}(N-m) + \mathbf{Q}.$$

Važno je uočiti da je optimalno upravljanje dato u obliku linearne kombinacije stanja sistema. Drugim riječima, ako želimo da realizujemo ovo upravljanje na digitalnom kontroleru, dovoljno je sačuvati vrijednosti vremenski promjenljive matrice $\mathbf{K}(N-m)$:

$$\mathbf{K}(N-m) = \left[\mathbf{R} + \mathbf{B}^{T}\mathbf{P}(m-1)\mathbf{B}\right]\mathbf{B}^{T}\mathbf{P}(m-1)\mathbf{A}.$$

Prethodne jednačine se rješavaju numerički, počev od $m=1$ do $m=N$, pri čemu je $\mathbf{P}(N)=\mathbf{H}$. Primijetiti da je veza između m i indeksa vremena n data sa $n=N-m$.

Diskretni LQR regulator

Moguće je uvesti i smjenu promjenjljive $n = N - m$, pa se prethodne jednačine mogu zapisati i u sljedećem obliku:

$$\mathbf{P}(n) = \left[\mathbf{A} + \mathbf{BK}(n)\right]^T \mathbf{P}(n+1)\left[\mathbf{A} + \mathbf{BK}(n)\right] + \mathbf{K}^T(n)\mathbf{RK}(n) + \mathbf{Q}.$$

$$\mathbf{K}(n) = \left[\mathbf{R} + \mathbf{B}^T\mathbf{P}(n+1)\mathbf{B}\right]\mathbf{B}^T\mathbf{P}(n+1)\mathbf{A}.$$

Prethodne jednačine se rješavaju unazad u diskretnom vremenu, za n koje ide od N do 0, jer je granični uslov Rikatijeve jednačine zadat u krajnjoj tački: $\mathbf{P}(N) = \mathbf{H}$. Optimalni upravljački zakon je zadat u obliku:

$$\mathbf{u}^*(n) = -\mathbf{K}(n)\mathbf{x}(n).$$

Kao što se može primijetiti, primjena dinamičkog programiranja na linearne sisteme omogućava dobijeanje optimalnog upravljanje u zatvorenom obliku. Kontroler koji se implementira u ovom obliku se naziva linearni kvadratni regulator (eng. linear quadratic regulator, LQR) iz razloga što je dobijene minimizacijom kvadratne forme i što se matematički zapisuje u obliku linerane povratne sprege po stanjima.

Diskretni LQR regulator

Ako $N \to \infty$, tada će koeficijenti Rikatijeve jednačine da konvergiraju ka konstantama za $n \to 0$. Da bi odredili rješenje Rikatijeve jednačine u stacionarnom stanju, treba poći od uslova stacionarnosti: $\mathbf{P}_{k+1}=\mathbf{P}_k=\mathbf{P}_{ss}$.

$$\mathbf{P}_{ss} = \left[\mathbf{A}+\mathbf{B}\mathbf{K}_{ss}\right]^T \mathbf{P}_{ss}\left[\mathbf{A}+\mathbf{B}\mathbf{K}_{ss}\right] + \mathbf{K}_{ss}{}^T\mathbf{R}\mathbf{K}_{ss} + \mathbf{Q}.$$

$$\mathbf{K} = \left[\mathbf{R}+\mathbf{B}^T\mathbf{P}_{ss}\mathbf{B}\right]\mathbf{B}^T\mathbf{P}_{ss}\mathbf{A}.$$

Kombinujući prethodne dvije jednačine dobija se diskretna algebarska Rikatijeva jednačina (DARE):

$$\mathbf{P}_{ss} = \mathbf{A}^T\mathbf{P}_{ss}\mathbf{A} + \mathbf{A}^T\mathbf{P}_{ss}\mathbf{B}\left[\mathbf{R}+\mathbf{B}\mathbf{P}_{ss}\mathbf{B}^T\right]\mathbf{B}^T\mathbf{P}_{ss}\mathbf{A} + \mathbf{Q}.$$

Koeficijenati $\mathbf{K}_{ss}$ se na drugi način mogu odrediti iterativnim rješavanjem diferencne jednačine. Za veliko N i $\mathbf{H}=\mathbf{0}$, koeficijenti matrice $\mathbf{P}(n)$ i $\mathbf{K}(n)$ će konvergirati ka konstantama, odnosno treba usvojiti da je:

$$\mathbf{K}_{ss} = \mathbf{K}(0).$$

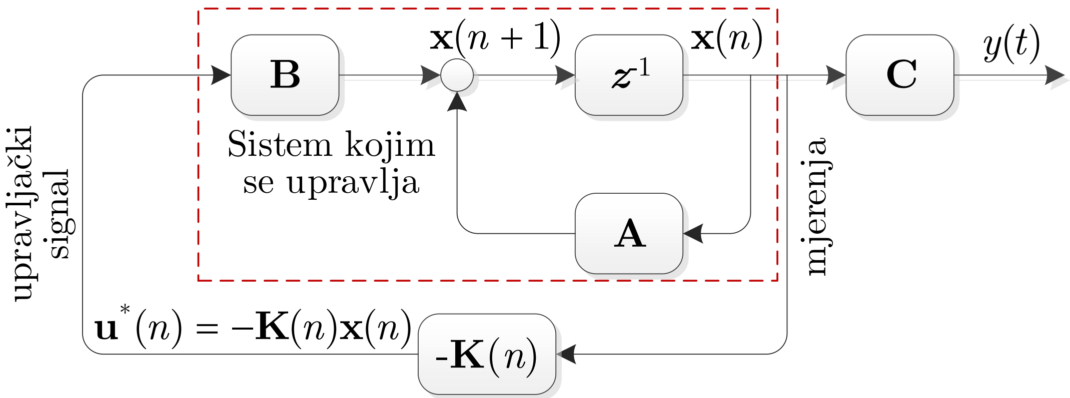
U Matlab-u algebarska Rikatijeva jednačina se rješava pomoću funkcije *dare*.

Diskretni LQR regulator

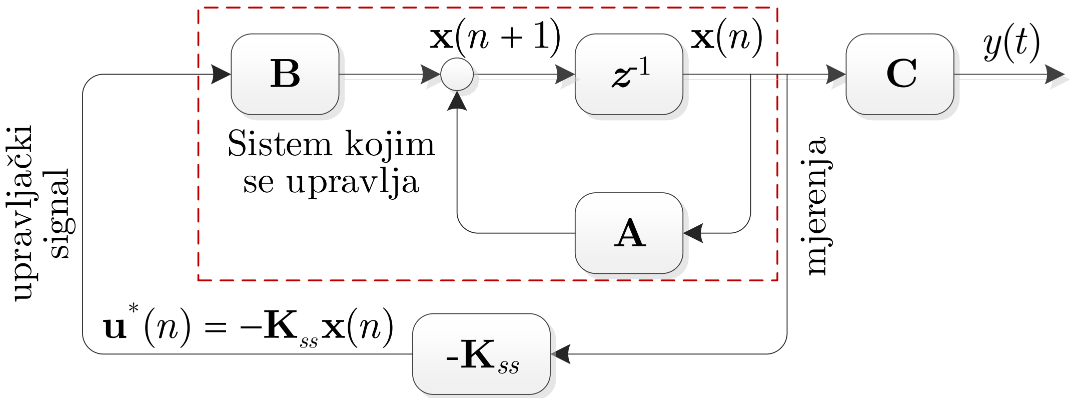
Na slikama su prikazana dva načina implementacije optimalnog upravljanja.

U prvom slučaju optimalno upravljanje se realizuje u obliku linearne vremenski promjenljive povr. sprege. U drugom slučaju upravljanje se realizuje u vidu linearne povratne sprege sa konstantnim pojačanjima koja su zapravo rješenje optimizacionog problema:

$$J = \frac{1}{2}\sum_{n=0}^{\infty} \mathbf{x}^T(n)\mathbf{Q}(n)\mathbf{x}(n) + \mathbf{u}^T(n)\mathbf{R}(n)\mathbf{u}(n).$$



Vremenski promjenljiva povratna sprega



Stacionarna povratna sprega

Primjer – diskretni LQR regulator

Diskretni LTI sistem je zadat u prostoru stanja:

$$\mathbf{x}(n+1)=\begin{bmatrix}0.9 & 0.05\\ -0.1 & 1.15\end{bmatrix}\mathbf{x}(n)+\begin{bmatrix}0.01\\ 0.05\end{bmatrix}u(n).$$

Odrediti optimalni upravljački zakon tako da bude minimizovana funkcija performanse:

$$J=\frac{1}{2}\sum_{n=0}^{N}0.25x_1^2(n)+0.05x_2^2(n)+0.05u^2(n).$$

a) Rikatijevu jednačinu riješiti na dva načina: u Simulinku i pomoću m-fajla.

b) Simulirati odziv spregnutog sistema, ako su početna stanja $[2 \;\; \text{-}1]^{\mathrm{T}}$.

c) Odrediti vrijednost optimalnih koeficijenata u stacionarnom stanju i optimalnu vrijednost funkcije performanse.

Funkcija performanse se može zapisati u sljedećem obliku:

$$J=\frac{1}{2}\sum_{n=0}^{N}\left[x_1(n) \quad x_1(n)\right]^T\begin{bmatrix}0.25 & 0\\ 0 & 0.05\end{bmatrix}\left[x_1(n) \quad x_1(n)\right]+0.05u^2(n).$$

što znači da su matrice $\mathbf{Q}$ i $\mathbf{R}$ jednake:

$$\mathbf{Q}=\begin{bmatrix}0.25 & 0\\ 0 & 0.05\end{bmatrix},\ \mathbf{R}=0.05.$$

Primjer – diskretni LQR regulator

Koeficijenti $\mathbf{K}(0)$ se mogu odrediti simulacijom Rikatijeve jednačine ili rješavanjem algebarske Rikatijeve jednačine:

$$\mathbf{P} = \mathbf{A}^T\mathbf{P}\mathbf{A} + \mathbf{A}^T\mathbf{P}\mathbf{B}\left[\mathbf{R} + \mathbf{B}\mathbf{P}\mathbf{B}^T\right]\mathbf{B}^T\mathbf{P}\mathbf{A} + \mathbf{Q}.$$

U Matlab-u se diskretna algebarska Rikatijeva jednačina rješava pomoću funkcije *dare*:

```
[P polovi K]=dare(A,B,Q,R),
```

dok se vektor povratne sprege može dobiti i pomoću komande *dlqr*:

```
[K P polovi]=dlqr(A,B,Q,R).
```

Optimalna vrijednost funkcije performanse je jednaka:

$$J_{0,N}^* = \frac{1}{2}\mathbf{x}^T(0)\mathbf{P}(0)\mathbf{x}(0)$$

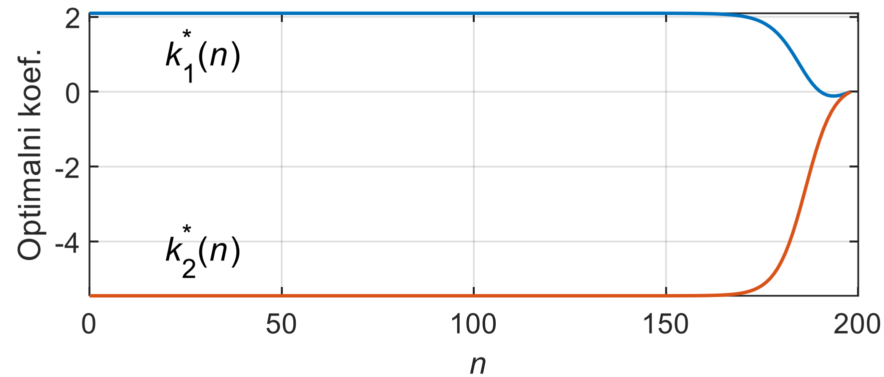
$$= \frac{1}{2}\begin{bmatrix} 2 & -1 \end{bmatrix}\begin{bmatrix} 2.9260 & -2.8912 \\ -2.8912 & 6.7742 \end{bmatrix}\begin{bmatrix} 2 \\ -1 \end{bmatrix}$$

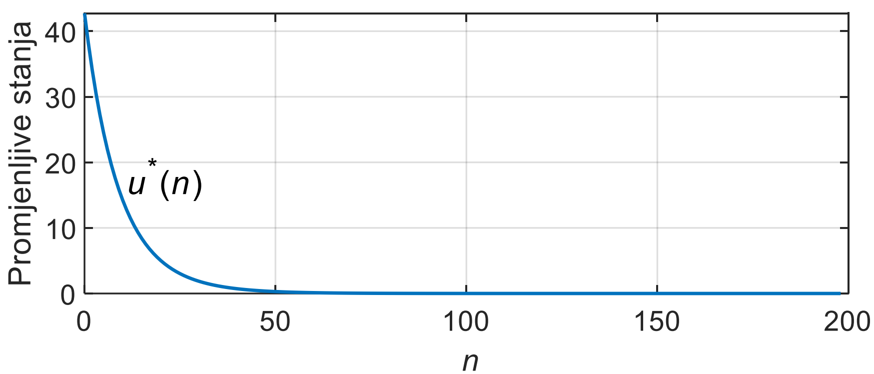
$$= 15.0213.$$

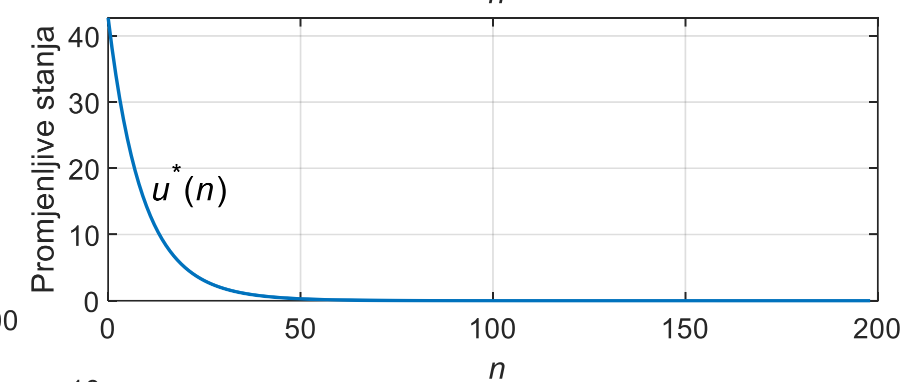
```
>> [P polovi K]=dare(A,B,Q,R)
P =
     2.9260   -2.8912
    -2.8912    6.7742

polovi =
     0.9205
     0.8781

K =
    -2.0944    5.4479

>> [K P polovi]=dare(A,B,Q,R)
```

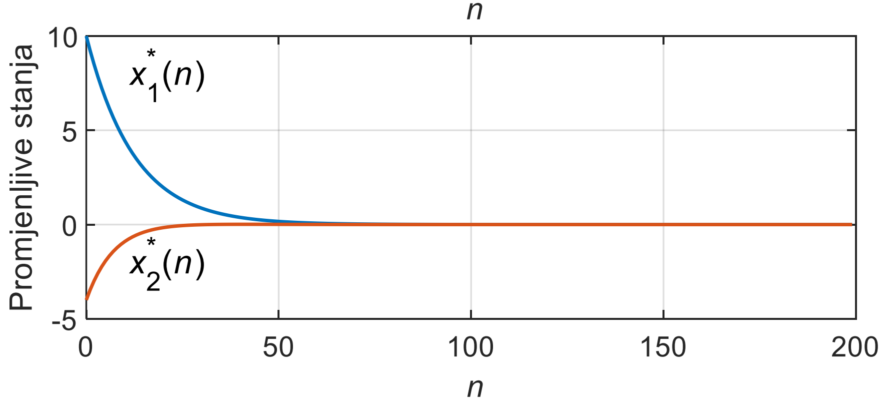
Primjer – diskretni LQR regulator

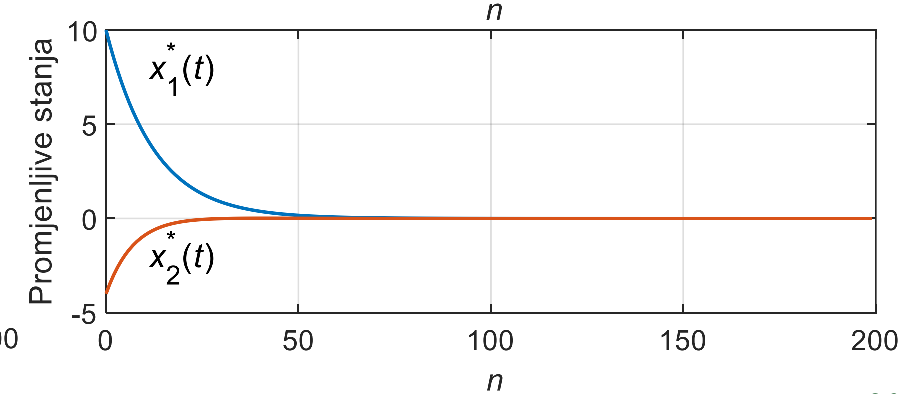
Na slikama su prikazane optimalne trajektorije i upravljanje za dva slučaja: sa konstantnim i vremenski promljenljvim koeficijentima linearne povratne sprege.











Primjer – diskretni LQR regulator

```
A=[0.9 0.05;-0.1 1.15]; B=[0.01;0.05]; Q=[0.25 0;0 0.05]; R=[0.05];
P{1}=zeros(2); N=200;
for k=1:N-1 % rjesavanje Rikatijeve jednacine
    K{N-k}=-[R+B'*P{k}*B]^-1*B'*P{k}*A;
    P{k+1}=[A+B*K{N-k}]'*P{k}*[A+B*K{N-k}]+K{N-k}'*R*K{N-k}+Q;
end
plot([0:N-2],element(K,1,1),'linewidth',1)
hold on
plot([0:N-2],element(K,1,2),'linewidth',1)
% simulacija upravljanja
x(:,1)=[10;-4]; % pocetni uslovi
for n=1:N-1
    u(n)=K{n}*x(:,n); % upravljacki signal
    x(:,n+1)=A*x(:,n)+B*u(n); % jednacine stanja
end
figure
plot([0:N-1],x','linewidth',1)
figure
plot([0:N-2],u,'linewidth',1)
% pomocna f-ja
function x=element(y,i,j)
    for n=1:length(y)
        x(n)=y{n}(i,j);
    end
end
```

Hamilton-Jakobi-Belmanova jednačina

Posmatrajmo kontinualni sistem zadat u prostoru stanja:

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

i funkciju performanse koju treba minimizovati:

$$J = h(\mathbf{x}(t_f), t_f) + \int_{t_0}^{t_f} g(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau) d\tau.$$

Funkciju performanse možemo razdvojiti na dva dijela:

$$J^*(\mathbf{x}(t), t) = \min_{\substack{\mathbf{u}(t) \\ t \le \tau \le t_f}} \left\{ \int_t^{t+\delta t} g(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau) d\tau + \int_{t+\delta t}^{t_f} g(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau) d\tau + h(\mathbf{x}(t_f), t_f) \right\}$$

pri čemu je δt ima malu vrijednost.

Prema Belmanovom principu optimalnosti minimalna vrijednost cost funkcije na intervalu $[t,\ t_f]$ je jednaka minimum zbira cost funkcije na intervalu $[t,\ t_0 + \delta t]$ i optimalne cijene na intervalu $[t_0 + \delta t,\ t_f\ ]$:

$$J^*(\mathbf{x}(t), t) = \min_{\substack{\mathbf{u}(t) \\ t \le \tau \le t_f}} \left\{ \int_t^{t+\delta t} g(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau) d\tau + J^*(\mathbf{x}(t+\delta t), t+\delta t) \right\},$$

Hamilton-Jakobi-Belmanova jednačina

Pod pretpostavkom da drugi parcijalni izvodi postoje i da su ograničeni, funkciju $J^*(\mathbf{x}(t+\delta t),t+\delta t)$ možemo razviti u Tejlorov red u okolini $(\mathbf{x}(t),t)$:

$$J^*(\mathbf{x}(t),t)=\min_{\substack{\mathbf{u}(t)\\ t\le\tau\le t_f}}\left\{\int_{t_0}^{t+\delta t} g(\mathbf{x}(\tau),\mathbf{u}(\tau),\tau)d\tau + J^*(\mathbf{x}(t),t)+\frac{\partial J^*(\mathbf{x}(t),t)}{\partial t}\delta t\right\}$$

$$+\frac{\partial J^*(\mathbf{x}(t),t)}{\partial \mathbf{x}}\big(x(t+\delta t)-\mathbf{x}(t)\big)+\text{članovi većeg reda.}$$

Kako je δt malo (teži nuli), prethodni izraz se može dodatno uprostiti:

$$J^*(\mathbf{x}(t),t)=\min_{\substack{\mathbf{u}(t)\\ t\le\tau\le t_f}}\left\{g(\mathbf{x}(t),\mathbf{u}(t),t)\delta t + J^*(\mathbf{x}(t),t)+\frac{\partial J^*(\mathbf{x}(t),t)}{\partial t}\delta t\right\}$$

$$+\frac{\partial J^*(\mathbf{x}(t),t)}{\partial \mathbf{x}}\mathbf{f}\big(\mathbf{x}(t),\mathbf{u}(t),t\big)\delta t+o(t).$$

Hamilton-Jakobi-Belmanova jednačina

Minimum prethodne funkcije možemo odrediti njenim diferenciranjem i izjednačavanjem izvoda sa nulom:

$$J^*(\mathbf{x}(t),t) = \frac{\partial J^*(\mathbf{x}(t),t)}{\partial t}\delta t + \min_{\mathbf{u}(t)}\{g(\mathbf{x}(t),\mathbf{u}(t),t)\delta t\}$$
$$+\frac{\partial J^*(\mathbf{x}(t),t)}{\partial \mathbf{x}}\mathbf{f}\left(\mathbf{x}(t),\mathbf{u}(t),t\right)\delta t + o(t).$$

Dijeljenjem prethodne jednačine sa δt dobija se:

$$\frac{\partial J^*(\mathbf{x}(t),t)}{\partial t} + \min_{\mathbf{u}(t)}\left\{g(\mathbf{x}(t),\mathbf{u}(t),t) + \frac{\partial J^*(\mathbf{x}(t),t)}{\partial \mathbf{x}}\mathbf{f}\left(\mathbf{x}(t),\mathbf{u}(t),t\right)\right\} = 0$$

Prethodna parcijalna diferencijalna jednačina treba da zadovolji graničnu vrijednost u trenutku $t=t_f$, odnosno:

$$J^*(\mathbf{x}(t_f),t_f) = h(\mathbf{x}(t_f),t_f).$$

Hamilton-Jakobi-Belmanova jednačina

Hamiltonijan $\mathcal{H}$ se definiše na sljedeći način:

$$\mathcal{H}\left(\mathbf{x}(t), \mathbf{u}(t), t, \frac{\partial J^*(\mathbf{x}(t), t)}{\partial \mathbf{x}}\right) = g(\mathbf{x}(t), \mathbf{u}(t), t) + \frac{\partial J^*(\mathbf{x}(t), t)}{\partial \mathbf{x}} \mathbf{f}\left(\mathbf{x}(t), \mathbf{u}(t), t\right),$$

dok je:

$$\mathcal{H}\left(\mathbf{x}(t), \mathbf{u}^*(t), t, \frac{\partial J^*(\mathbf{x}(t), t)}{\partial \mathbf{x}}\right) = \min_{u(t)} \mathcal{H}\left(\mathbf{x}(t), \mathbf{u}(t), t, \frac{\partial J^*(\mathbf{x}(t), t)}{\partial \mathbf{x}}\right).$$

Konačno dobijamo Hamilton-Jakobi-Belmanovu (HJB) jednačinu:

$$\frac{\partial J^*(\mathbf{x}(t), t)}{\partial t} + \mathcal{H}\left(\mathbf{x}(t), \mathbf{u}^*(t), t, \frac{\partial J^*(\mathbf{x}(t), t)}{\partial \mathbf{x}}\right) = 0,$$

koja zapravo predstavlja kontinualnu verziju Belmanove rekurzivne jednačine.

Primjer – HJB

Kontinualni LTI sistem prvog reda je opisan diferencijalnom jendačinom:

$$\dot{x}(t) = x(t) + u(t).$$

Odrediti optimalni upravljački zakon koji minimizuje funkciju performanse:

$$J = \frac{1}{4}x^2(T) + \int_0^T \frac{1}{4}u^2(t)dt.$$

Optimalni upravljački signal treba da zadovolji HJB jednačinu:

$$\frac{\partial J^*(x(t),t)}{\partial t} + \mathcal{H}\left(x(t), u^*(t), t, \frac{\partial J^*(x(t),t)}{\partial x(t)}\right) = 0$$

Uvrštavajući jednačinu stanja u izraz za Hamiltonijan dobija se:

$$\mathcal{H}\left(x(t), u(t), t, \frac{\partial J^*(x(t),t)}{\partial x}\right) = \frac{1}{4}u^2(t) + \frac{\partial J^*(x(t),t)}{\partial x(t)}\big(x(t) + u(t)\big)$$

Diferenciranjem Hamiltonijana po upravljačkom signalu, dobija se:

$$\frac{\partial \mathcal{H}}{\partial u(t)} = \frac{1}{2}u(t) + \frac{\partial J^*(x(t),t)}{\partial x(t)} = 0.$$

Primjer – HJB

Odnosno, optimalni upravljački ima oblik:

$$u^*(t) = -2\frac{\partial J^*(x(t),t)}{\partial x(t)}.$$

Dalje, uvrštavanjem $u^*(t)$ u HJB dobijamo:

$$\frac{\partial J^*(x(t),t)}{\partial t} + \frac{\partial J^*(x(t),t)^2}{\partial x(t)} + \frac{\partial J^*(x(t),t)}{\partial x(t)}\left(x(t) - 2\frac{\partial J^*(x(t),t)}{\partial x(t)}\right) = 0$$

$$\frac{\partial J^*(x(t),t)}{\partial t} - \frac{\partial J^*(x(t),t)^2}{\partial x(t)} + \frac{\partial J^*(x(t),t)}{\partial x(t)}x(t) = 0$$

Jedan način da riješimo HJB je da pretpostavimo oblik rješenja. Praveći analogiju sa diskretnim LQR regulatorom, pretpostavimo sljedeće:

$$J^*(x(t),t) = \frac{1}{4}K(t)x^2(t).$$

$$\frac{\partial J^*(x(t),t)}{\partial x(t)} = \frac{1}{2}K(t)x(t). \qquad u^*(t) = -2\frac{\partial J^*(x(t),t)}{\partial x(t)} = -K(t)x(t).$$

Primjer – HJB

Najprije odredimo

$$\frac{\partial J^*(x(t),t)}{\partial t} = \frac{\partial}{\partial t}\left(\frac{1}{4}K(t)x^2(t)\right) = \frac{1}{4}\dot{K}(t)x^2(t) + \frac{1}{2}K(t)x(t)\dot{x}(t)$$

$$= \frac{1}{4}\dot{K}(t)x^2(t) + \frac{1}{2}K(t)x(t)\big(x(t) - K(t)x(t)\big) = \frac{1}{4}\dot{K}(t)x^2(t) + \frac{1}{2}K(t)x^2(t) - \frac{1}{2}K^2(t)x^2(t).$$

Uvrštavajući u HJB dobija se:

$$\frac{1}{4}\dot{K}(t)x^2(t) - K^2(t)x^2(t) + \frac{1}{2}K(t)x^2(t) = 0.$$

Dakle, optimalno upravljanje je jednako $u(t) = -K(t)$, pri čemu se dobija rješenje Rikatijeve diferencijalne jednačine $K(t)$:

$$\dot{K}(t) - 4K^2(t) + 2K(t) = 0.$$

U stacionarnom stanju gornja jednačina ima rješenje $K_{ss} = -2$, odnosno $u^*(t) = -2x(t)$.

Kontinualni LQR regulator

Sada ćemo uopštiti proceduru iz prethodnog primjera na kontinualne LTI sisteme n-tog reda:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}u(t).$$

Funkcija peroformanse je zadate u obliku kvadratne forme:

$$J = \frac{1}{2}\mathbf{x}^T(t_f)\mathbf{H}\mathbf{x}(t_f) + \int_{t_0}^{t_f} \frac{1}{2}\Big[\mathbf{x}^T(t)\mathbf{Q}(t)\mathbf{x}(t) + \mathbf{u}^T(t)\mathbf{R}(t)\mathbf{u}(t)\Big]d\tau.$$

Upravljački signal treba da zadovolji HJB jednačinu:

$$\frac{\partial J^*(\mathbf{x}(t),t)}{\partial t} + \mathcal{H}\left(\mathbf{x}(t), \mathbf{u}^*(t), t, \frac{\partial J^*(\mathbf{x}(t),t)}{\partial \mathbf{x}}\right) = 0$$

Hamiltonijan je jednak:

$$\mathcal{H}\left(\mathbf{x}(t), \mathbf{u}(t), t, \frac{\partial J^*(\mathbf{x}(t),t)}{\partial \mathbf{x}}\right) = \frac{1}{2}\mathbf{x}^T(t)\mathbf{Q}(t)\mathbf{x}(t) + \frac{1}{2}\mathbf{u}^T(t)\mathbf{R}(t)\mathbf{u}(t)$$

$$+ \frac{\partial J^*(\mathbf{x}(t),t)}{\partial \mathbf{x}}\big(\mathbf{A}\mathbf{x}(t) + \mathbf{B}u(t)\big).$$

Kontinualni LQR regulator

Izvod Hamiltonijana je jednak:

$$\frac{\partial \mathcal{H}}{\partial \mathbf{u}(t)} = \mathbf{R}(t)\mathbf{u}(t) + \mathbf{B}^T \frac{\partial J^*(\mathbf{x}(t), t)}{\partial \mathbf{x}} \mathbf{u}(t),$$

odakle se dobija:

$$\mathbf{u}^*(t) = \mathbf{R}^{-1}(t)\mathbf{B}^T(t) \frac{\partial J^*(\mathbf{x}(t), t)}{\partial \mathbf{x}}.$$

Dalje, uvrštavanjem $\mathbf{u}(t)$ u $\mathbf{H}$, dobija se:

$$\mathcal{H}\left(\mathbf{x}(t), \mathbf{u}^*(t), t, \frac{\partial J^*(\mathbf{x}(t), t)}{\partial \mathbf{x}}\right) = \frac{1}{2}\mathbf{x}^T(t)\mathbf{Q}(t)\mathbf{x}(t)$$
$$+\frac{1}{2}\frac{\partial J^*(\mathbf{x}(t), t)}{\partial \mathbf{x}}^T \mathbf{B}\mathbf{R}(t)\mathbf{B}^T \frac{\partial J^*(\mathbf{x}(t), t)}{\partial \mathbf{x}}$$
$$+\frac{\partial J^*(\mathbf{x}(t), t)}{\partial \mathbf{x}}^T \mathbf{A}\mathbf{x}(t) - \frac{\partial J^*(\mathbf{x}(t), t)}{\partial \mathbf{x}}^T \mathbf{B}\mathbf{R}(t)\mathbf{B}^T \frac{\partial J^*(\mathbf{x}(t), t)}{\partial \mathbf{x}}.$$

Kontinualni LQR regulator

Opet, praveći analogiju sa diskretnim sistemima, pretpostavimo da minimalna vrijednost funkcije performanse ima oblik:

$$J^*(\mathbf{x}(t), t) = \mathbf{x}^T(t)\mathbf{P}(t)\mathbf{x}(t).$$

Uvrštavanjem pretpostavljenog rješenja u HJB dobijamo:

$$\frac{1}{2}\mathbf{x}^T(t)\dot{\mathbf{P}}(t)\mathbf{x}(t) + \frac{1}{2}\mathbf{x}(t)^T\mathbf{Q}\mathbf{x}(t) - \frac{1}{2}\mathbf{x}^T(t)\mathbf{K}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{P}(t)\mathbf{x}(t) + \mathbf{x}^T(t)\mathbf{K}\mathbf{A}\mathbf{x}(t) = 0.$$

Prethodna jednačina se svodi na:

$$\begin{aligned}&\frac{1}{2}\mathbf{x}(t)^T\dot{\mathbf{P}}(t)\mathbf{x}(t) + \frac{1}{2}\mathbf{x}(t)^T\mathbf{Q}\mathbf{x}(t) - \frac{1}{2}\mathbf{x}(t)^T\mathbf{K}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{P}(t)\mathbf{x}(t)\\ &+\frac{1}{2}\mathbf{x}(t)^T\mathbf{K}\mathbf{A}\mathbf{x}(t) + \frac{1}{2}\mathbf{x}^T\mathbf{A}^T\mathbf{K}\mathbf{x}(t) = 0.\end{aligned}$$

Posljednja jednačina treba da važi za svako $\mathbf{x}(t)$, odnosno:

$$\dot{\mathbf{P}}(t) - \mathbf{P}(t)\mathbf{B}(t)\mathbf{R}(t)^{-1}\mathbf{B}^T(t)\mathbf{P}(t) + \mathbf{P}(t)\mathbf{A}(t) + \mathbf{A}^T(t)\mathbf{P}(t) + \mathbf{Q}(t) = 0.$$

pri čemu je granični uslov $\mathbf{P}(t_f) = \mathbf{H}$.

Kontinualni LQR regulator

Prethodna jednačina predstavlja vektorsku Rikatijevu diferencijalnu jednačinu. Rikatijeva jednačina je nelinearna i rješava se numeričkim putem, integracijom unazad, jer je granični uslov zadat u krajnjem vremenu. Optimalni upravljački signal ima oblik linearne povratne sprege po stanjima:

$$\mathbf{u}(t) = -\mathbf{R}^{-1}(t)\mathbf{B}^{T}(t)\mathbf{P}(t)\mathbf{x}(t).$$

U specijalnom slučaju kada je indeks performanse jednak:

$$J = \frac{1}{2}\int_{0}^{\infty}\frac{1}{2}\left[\mathbf{x}^{T}(t)\mathbf{Q}(t)\mathbf{x}(t) + \mathbf{u}^{T}(t)\mathbf{R}(t)\mathbf{u}(t)\right]d\tau,$$

optimalni koeficijenti su konstantni i mogu se dobiti rješavanjem algebarske Rikatijeve jednačine (ARE):

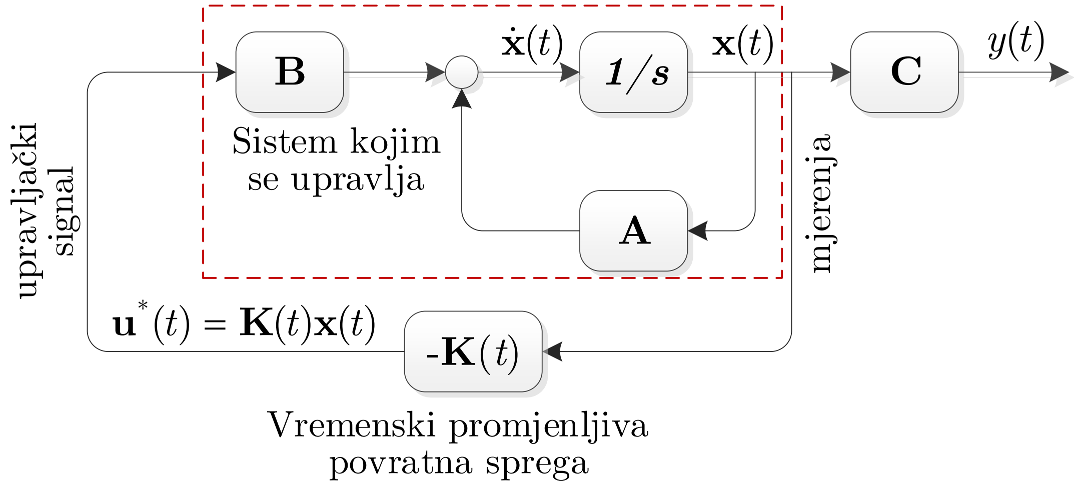
$$-\mathbf{P}_{ss}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^{T}\mathbf{P}_{ss} + \mathbf{P}_{ss}\mathbf{A} + \mathbf{A}^{T}\mathbf{P}_{ss} + \mathbf{Q} = 0.$$

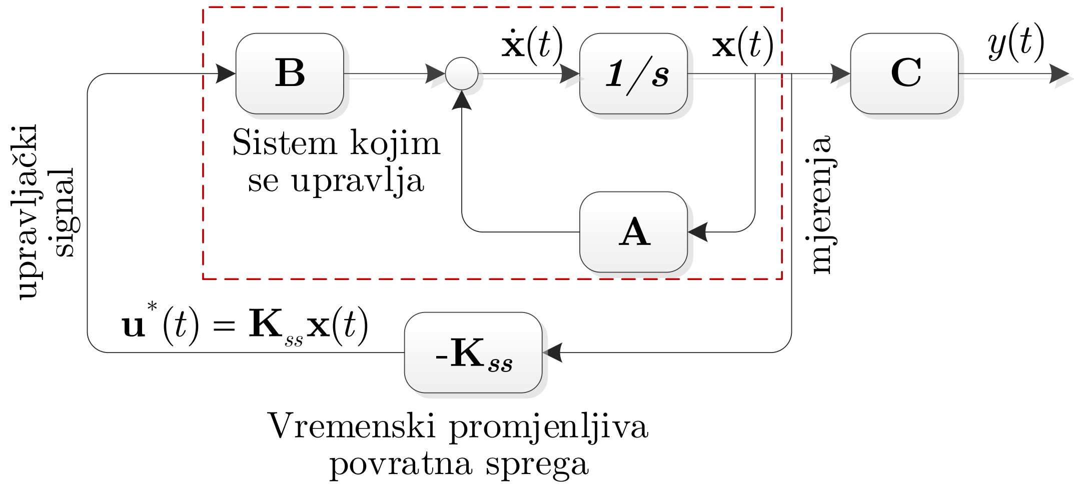
Optimalno upravljanje je tada jednako:

$$\mathbf{u}(t) = -\mathbf{R}^{-1}\mathbf{B}^{T}\mathbf{P}_{ss}\mathbf{x}(t).$$

Kontinualni LQR regulator

Dva načina implementacije optimalnog upravljanja.





Primjer – kontinualni LQR regulator

Kontinualni LTI sistem je zadat u prostoru stanja:

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} 1 & 0 \\ 2 & 0 \end{bmatrix}\mathbf{x}(t) + \begin{bmatrix} 1 \\ 0 \end{bmatrix}u(t), \;\; y(t) = \begin{bmatrix} 0 & 1 \end{bmatrix}\mathbf{x}(t).$$

Odrediti upravljački signal koji minimizuje sljedeću funkciju performanse:

$$J = \mathbf{x}^T(t)\begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}\mathbf{x}(t) + \int_0^5 \left[y^2(t) + u^2(t)\right]dt$$

a) Diferencijalnu Rikatijevu jednačinu riješiti u Simulinku.

b) Odrediti vrijednost vektora pojačanja **K** u stacionarom stanju.

c) Simulirati odziv spregnutog sistema, ako su početna stanja $[2 \;\; -1]^{T}$. Usvojiti konstantne vrijednosti pojačanja **K**.

Funkcija performanse se može zapisati u sljedećem obliku:

$$J = \mathbf{x}^T(t)\begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}\mathbf{x}(t) + \int_0^5 \left[\mathbf{x}^T(t)\mathbf{C}^T\mathbf{C}\mathbf{x}(t) + u^2(t)\right]dt,$$

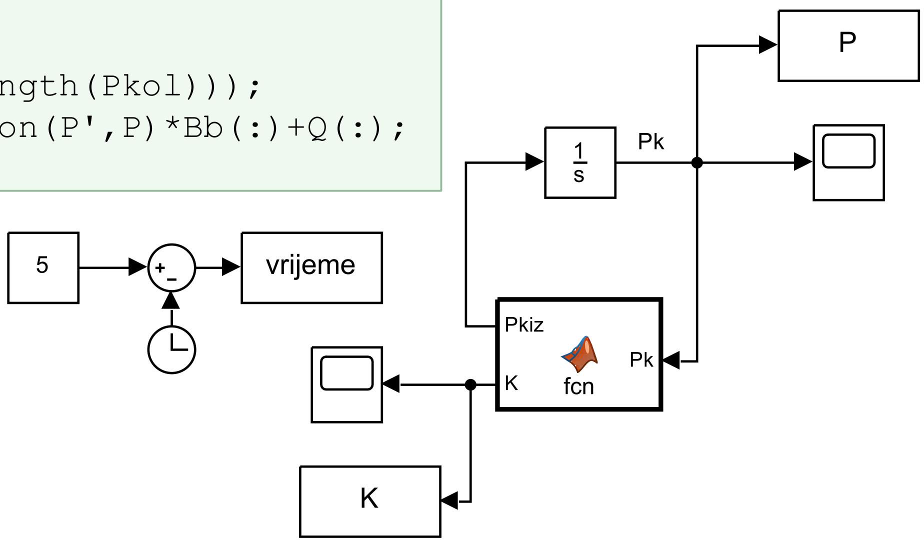
što znači da su matrice **Q** i **R** jednake:

$$\mathbf{Q} = \mathbf{C}^T\mathbf{C} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}, \;\; \mathbf{R} = 1.$$

Primjer – kontinualni LQR regulator

```
function [Pkolol_iz,K] = fcn(Pkol)
A=[1 0; 2 0];
B=[1;0]; C=[0 1]; Q=C'*C;
R=1; I=eye(2);
%%
Bb=B*inv(R)*B';
P=reshape(Pkol,sqrt(length(Pkol)),sqrt(length(Pkol)));
Pkolol_iz=[kron(A',I)+kron(I,A')]*Pkol-kron(P',P)*Bb(:)+Q(:);
K=inv(R)*B'*P;
```

Simulink ne može da riješi matričnu Rikatijevu jednačinu (matrica **P** je kvadratna). Da bi prevazišli taj problem, Rikatijeva jednačina se može zapisati u vektorskom obliku:



$$\dot{\mathbf{P}}_{col}(t) = \mathbf{P}(t)^T \otimes \mathbf{P}(t)\left[\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\right]_{col} - \left[\mathbf{A}^T \otimes \mathbf{I} - \mathbf{I} \otimes \mathbf{A}^T\right]\mathbf{P}_{col}(t) - \mathbf{Q}.$$

Provjerite!

$\otimes$ - Kronekerov proizvod

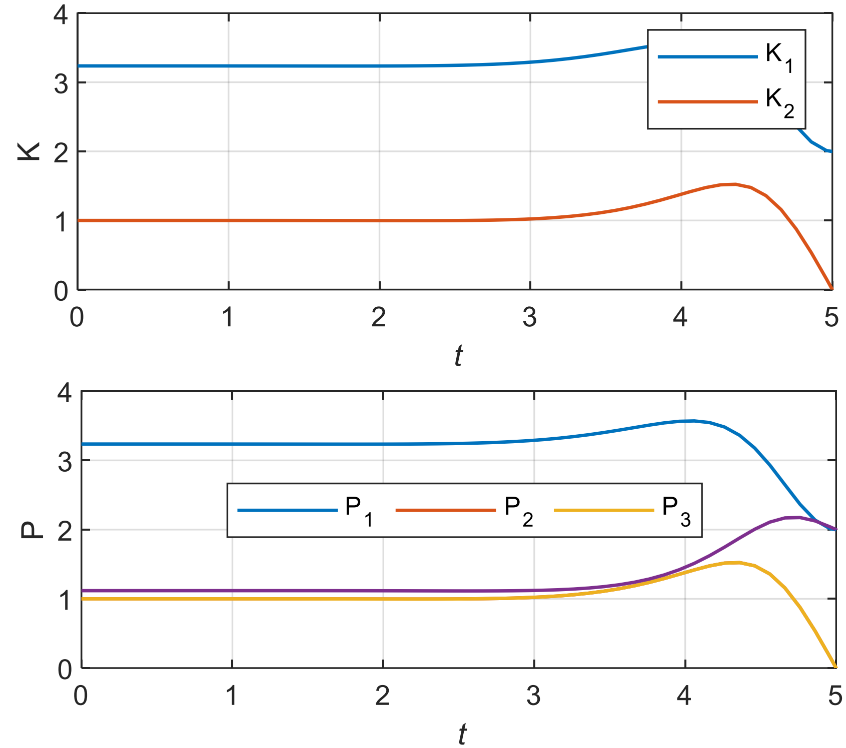
$\mathbf{A}_{col}$ – vektor koji se formira nadovezivanjem kolona matrice $\mathbf{A}$

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{A} & \cdots & a_{1n}\mathbf{A} \\ \vdots & \ddots & \vdots \\ a_{m1}\mathbf{A} & \cdots & a_{mn}\mathbf{A} \end{bmatrix}$$

Primjer – kontinualni LQR regulator

Na slikama je su prikazani optimalni koeficijenti povratne sprege i koeficijenti matrice $\mathbf{P}(t)$. Kao što se može vidjeti matrica $\mathbf{P}(t)$ u trentuku t=5s ima vrijednosti zadate krajnjim uslovom. Ovi koef. unazad u vremenu konvergiraju ka konstatnim vrijednostima. Isto važi i za koeficijente vektora $\mathbf{K}(t)$.

U praksi se najčešće tokom čitavog vremenskog intervala upravljanja koriste konstantne vrijednosti matrice $\mathbf{K}(t)$, odnosno $\mathbf{K}(0)$.




```
plot(5-vrijeme.time,shiftdim(K.signals.values(1,1,:)),'linewidth',1)
hold on
plot(5-vrijeme.time,shiftdim(K.signals.values(1,2,:)),'linewidth',1)
figure(2)
plot(5-vrijeme.time,shiftdim(P.signals.values(1,1,:)),'linewidth',1)
hold on
plot(5-vrijeme.time,P.signals.values,'linewidth',1)
```

Primjer – kontinualni LQR regulator

Koeficijenti $\mathbf{K}(0)$ se mogu odrediti simulacijom Rikatijeve jednačine ili rješavanjem algebarske Rikatijeve jednačine:

$$\mathbf{PBR}^{-1}\mathbf{B}^T\mathbf{P}-\mathbf{PA}-\mathbf{A}^T\mathbf{P}+\mathbf{Q}=0.$$

Prethodna jednačina se svodi na:

$$\begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix} \times \begin{bmatrix} 1 \\ 0 \end{bmatrix} \times 1^{-1} \begin{bmatrix} 1 \\ 0 \end{bmatrix}^T \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix} - \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} - \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}^T \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} = 0.$$

Prilikom rješavanja zadnje jednačine treba voditi računa da je matrica $\mathbf{P}$ simetrična i pozitivno definitna (odbaciti rješenja sa nepozitivnim sopstvenim vrijednostima).

U Matlab-u se algebarska Rikatijeva jednačina rješava pomoću funkcije *care*:

```
[P polovi K]=care(A,B,Q,R),
```

dok se vektor povratne sprege može dobiti i pomoću komande *lqr*:

```
[P polovi K]=care(A,B,Q,R).
```

Uporediti sa rezultatima simulacije

```
[P polovi K]=care(A,B,Q,R)
P =
    3.2361    1.0000
    1.0000    1.1180

polovi =
  -1.1180 + 0.8660i
  -1.1180 - 0.8660i

K =
    3.2361    1.0000
```

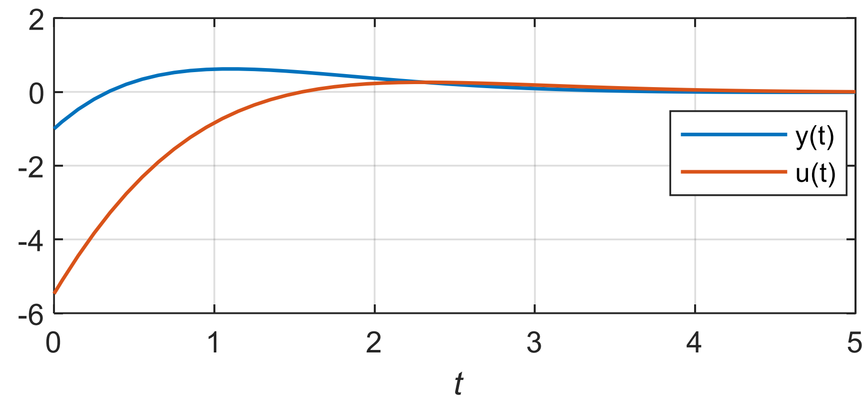
Primjer – kontinualni LQR regulator

Na slikama ispod su prikazani izlazni i upravljački signal.

Optimalna vrijednost funkcije performanse je jednaka:

$$J = \int_0^\infty \left[y^2(t) + u^2(t) \right] dt = \mathbf{x}(0)^T \mathbf{P}(0)\mathbf{x}(0)$$

$$= \begin{bmatrix} 2 \\ -1 \end{bmatrix}^T \begin{bmatrix} 3.23 & 1 \\ 1 & 1.11 \end{bmatrix} \begin{bmatrix} 2 & -1 \end{bmatrix} = 10.06.$$




```
[P polovi K]=care(A,B,Q,R)
P =
    3.2361    1.0000
    1.0000    1.1180

polovi =
  -1.1180 + 0.8660i
  -1.1180 - 0.8660i

K =
    3.2361    1.0000
```