6. Vremenski optimalno upravljanje nelinearnim sustavima

U četvrtom poglavlju razmatrali smo problem optimalnog upravljanja unutar zadanog vremenskog intervala. Ovdje ćemo razmatrati problem vremenski optimalnog upravljanja (*Time-Optimal Control*, TOC), odnosno, upravljanja u slučaju kada vremenski interval treba minimizirati.

Postoje dva pristupa numeričkom rješavanju tog problema. Kod vremenske diskretizacije ukupni vremenski interval jednak je produktu broja podintervala N s periodom diskretizacije T, tako da varijabilnost vremenskog intervala možemo izraziti jednom od tih veličina uzimajući drugu za konstantu. Dosadašnja primjena BPTT algoritma na vremenski optimalno upravljanje [22], problem tretira uzimajući konstantnim period diskretizacije T i varirajući broj podintervala N. U tom pristupu je problematična cjelobrojnost varijable N kod primjene gradijentne metode. Uzimanjem perioda diskretizacije kao kontinuirane varijable, izbjegavamo gornji problem, međutim, u tom slučaju javlja se problem nestabilnosti koji se može rješiti jedino uzimanjem jako male vrijednosti koeficijenta konvergencije u gradijentnom algoritmu za varijablu T, čime drastično usporavamo konvergenciju algoritma.

U ovom poglavlju prikazat ćemo numeričku metodu rješavanja vremenski optimalnog upravljanja, koja relativno efikasno izbjegava gore navedene probleme. Metoda je bazirana na periodu diskretizacije T kao varijabli po kojoj se minimizira, i koristi svojstva kaznenih funkcija za rubne uvjete i ograničenja.

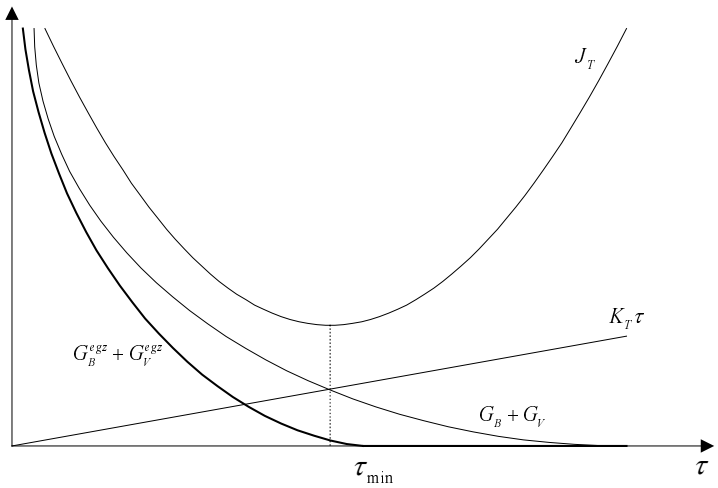
6.1 Direktna primjena BPTT algoritma na TOC

Ovdje ćemo prikazati proširenje BPTT algoritma za vremenski optimalno upravljanje (problem minimalnog vremena) za slučaj varijabilnog perioda diskretizacije, $\tau \equiv T$. Osnovna ideja je u tome da se zadrži algoritam za izračunavanje upravljačkih varijabli za zadani period diskretizacije, dok se slijedeća iteracija perioda diskretizacije izračunava na osnovi gradijenta nove funkcije cilja koja sadrži vremenski kriterij optimalnosti (minimum vremena $t_{\min} = N\tau_{\min}$, za prijelaz iz početnog u konačno stanje) i kaznene funkcije za ograničenja i rubne uvjete

$$J_T = K_T \tau + G_B(\mathbf{x}_N) + G_V(\mathbf{u}_0, \mathbf{u}_1, \ldots, \mathbf{u}_{N-1}). \tag{6.1}$$

Minimalno vrijeme za prijelaz iz početnog u konačno stanje ima smisla određivati jedino ako su zadana ograničenja na upravljačke varijable. To znači da bi za neko vrijeme manje od minimalnog, $t_f < t_{\min}$, BPTT algoritam za fiksnu vrijednost vremenskog

intervala neizbježno dao vrijednosti upravljačkih varijabli koje bi kršile ograničenja, što bi dovelo do povećanja vrijednosti kaznene funkcije G_V, a time ujedno dolazi do kršenja rubnih uvjeta i povećanja vrijednosti G_B. S druge strane, za vrijeme veće od minimalnog, $t_f > t_{\min}$, upravljačke varijable zadovoljavaju ograničenja i kaznene funkcije teže nuli (za egzaktna, analitička rješenja, vrijednosti kaznenih funkcija su jednake nuli). Kada se kaznenim funkcijama doda linearna ovisnost o vremenu, ukupna funkcija cilja ima minimum koji teži egzaktnoj vrijednosti $t_{\min}$, s povećavanjem koeficijenata kaznenih funkcija. Slika 6.1 daje grafički prikaz naprijed navedenog



Slika 6.1 *Prikaz ovisnosti kaznenih funkcija i J_T o periodu diskretizacije.*

Ako u diskretiziranoj formulaciji problema optimalnog upravljanja, sa specificiranim vremenom trajanja procesa, zamjenimo konstantnu vrijednost perioda diskretizacije T s varijablom $\tau \equiv T$, koja će se određivati na osnovu gradijentnog algoritma, imamo za *k*-ti korak iteracije gradijentnog algoritma

$$\mathbf{x}_{i+1}^{(k)} = \mathbf{x}_i^{(k)} + \tau^{(k)}\,\phi(\mathbf{x}_i^{(k)}, \mathbf{u}_i^{(k)}), \qquad \mathbf{x}_0^{(k)} = \mathbf{x}_0, \qquad \mathbf{x}_N^{(k)} = \mathbf{x}_f, \tag{6.2}$$

$$g_j(\mathbf{x}_i^{(k)},\ \mathbf{u}_i^{(k)}) \geq 0, \qquad j = 1,2,\ldots,p, \tag{6.3}$$

$$J(\mathbf{u}_0^{(k)}, \mathbf{u}_1^{(k)}, \ldots, \mathbf{u}_{N-1}^{(k)}) = \min_{\mathbf{u}} \tau^{(k)} \sum_{i=0}^{N-1} F'(\mathbf{x}_0^{(k)}, \mathbf{u}_0^{(k)}) + G_B(\mathbf{x}_N^{(k)}) \tag{6.4}$$

za $i = 0,1,\ldots,N-1$, gdje je $\mathbf{x}_i = \begin{bmatrix} x_i^1 & x_i^2 & \cdots & x_i^n \end{bmatrix}^T$, $\mathbf{u}_i = \begin{bmatrix} u_i^1 & u_i^2 & \cdots & u_i^m \end{bmatrix}^T$.

Gornji problem za fiksni τ riješen je u 4. poglavlju gradijentnom metodom

$$\mathbf{u}_j^{(k+1)} = \mathbf{u}_j^{(k)} - \eta \frac{\partial J(\mathbf{u}_0^{(k)},\ \mathbf{u}_1^{(k)},\ \ldots,\ \mathbf{u}_{N-1}^{(k)})}{\partial \mathbf{u}_j^{(k)}}. \tag{6.5}$$

Da bi izračunali slijedeću iteraciju $\tau^{(k+1)}$ primjenom gradijentnog algoritma, moramo funkciju cilja J_T implicitno izraziti u ovisnosti o $\tau^{(k)}$

$$J_T(\tau^{(k)}) = K_T \tau^{(k)} + G_V(\mathbf{u}_0^{(k+1)}, \mathbf{u}_1^{(k+1)}, \ldots, \mathbf{u}_{N-1}^{(k+1)}) + G_B(\mathbf{x}_N^{(k)}), \tag{6.6}$$

tako da imamo

$$\tau^{(k+1)} = \tau^{(k)} - \mu \frac{\partial J_T(\tau^{(k)})}{\partial \tau^{(k)}}. \tag{6.7}$$

Sada se problem svodi na računanje gradijenta funkcije cilja (6.6)

$$\frac{\partial J_T(\tau^{(k)})}{\partial \tau^{(k)}} = K_T + \frac{\partial G_V}{\partial \tau^{(k)}} + \frac{\partial G_B}{\partial \tau^{(k)}}. \tag{6.8}$$

Počet ćemo s gradijentom kaznene funkcije za rubne uvjete

$$\frac{\partial G_B}{\partial \tau^{(k)}} = \mathbf{G}_x^T(N) \cdot \mathbf{X}_\tau(N), \tag{6.9}$$

gdje je

$$\mathbf{X}_\tau(N) = \begin{bmatrix} \frac{\partial x_1^{(k)}(N)}{\partial \tau^{(k)}} & \frac{\partial x_2^{(k)}(N)}{\partial \tau^{(k)}} & \cdots & \frac{\partial x_n^{(k)}(N)}{\partial \tau^{(k)}} \end{bmatrix}^T,$$

s tim da je $x_j^{(k)}(N)$ j-ta komponenta vektora $\mathbf{x}_N^{(k)}$. Rekurzivnu relaciju za gornji vektor dobivamo derivirajući izraz (5.2)

$$\mathbf{X}_\tau(j) = \mathbf{X}_\tau(j-1) + \phi(j-1) + \tau^{(k)} \mathbf{X}^T(j-1) \cdot \mathbf{X}_\tau(j-1), \tag{6.10}$$

za $j = 1, \ldots, N$, s početnim uvjetom

$$\mathbf{X}_\tau(0) = \mathbf{0}.$$

Sad nam preostaje računanje gradijenta kaznene funkcije ograničenja G_V

$$\frac{\partial G_V}{\partial \tau^{(k)}} = \sum_{j=0}^{N-1} \mathbf{G}_{Vu}^T(j) \cdot \mathbf{U}_\tau(j), \tag{6.11}$$

gdje je

$$\mathbf{G}_{Vu}(j) = \begin{bmatrix} \frac{\partial G_V}{\partial u_1^{(k+1)}(j)} & \frac{\partial G_V}{\partial u_2^{(k+1)}(j)} & \cdots & \frac{\partial G_V}{\partial u_m^{(k+1)}(j)} \end{bmatrix}^T,$$

$$\mathbf{U}_\tau(j) = \begin{bmatrix} \frac{\partial u_1^{(k+1)}(j)}{\partial \tau^{(k)}} & \frac{\partial u_2^{(k+1)}(j)}{\partial \tau^{(k)}} & \cdots & \frac{\partial u_m^{(k+1)}(j)}{\partial \tau^{(k)}} \end{bmatrix}^T,$$

s tim da je $u_i^{(k+1)}(j)$ i-ta komponenta vektora $\mathbf{u}_j^{(k+1)}$. $\mathbf{x}_N^{(k)}$. Rekurzivnu relaciju za vektor $\mathbf{U}_\tau(j)$ dobivamo derivirajući izraz (5.5), što se svodi, kao što vidimo na derivaciju gradijenta funkcije cilja (5.4) po upravljačkim varijablama. To smo izračunali u prošlom poglavlju i dobili slijedeće rekurzivne relacije koje, prilagođene za ovaj problem poprimaju oblik

$$\mathbf{J}_u(j) = \tau^{(k)}\left(\mathbf{F}'_u(j) + \mathbf{S}(j)\right) + \mathbf{X}_u^T(j)\cdot\mathbf{G}_x(N),$$
$$\mathbf{X}_u(j) = \mathbf{U}(j)\cdot\mathbf{D}\ (j),$$
$$\mathbf{D}^T(j) = \mathbf{X}(j+1)\cdot\mathbf{D}^T(j+1),$$
$$\mathbf{S}(j) = \mathbf{U}(j)\cdot\mathbf{P}(j),$$
$$\mathbf{P}(j) = \mathbf{F}'_x(j+1) + \mathbf{X}(j+1)\cdot\mathbf{P}(j+1),$$

sa rubnim uvjetima

$$\mathbf{J}_u(N-1) = \tau^{(k)}\mathbf{F}'_u(N-1) + \mathbf{U}(N-1)\cdot\mathbf{G}_x(N)$$
$$\mathbf{P}(N-1) = \mathbf{0},$$
$$\mathbf{D}(N-1) = \mathbf{I},$$

za $j = N-2,\ N-3,\ \ldots, 0$.

Imamo dakle

$$\mathbf{U}_\tau(j) = -\eta\frac{\partial\mathbf{J}_u(j)}{\partial\tau^{(k)}}, \tag{6.12}$$

odnosno

$$\frac{\partial\mathbf{J}_u(j)}{\partial\tau^{(k)}} = \mathbf{F}'_u(j) + \mathbf{S}(j) + \tau^{(k)}\left(\frac{\partial\mathbf{F}'_u(j)}{\partial\tau^{(k)}} + \frac{\partial\mathbf{S}(j)}{\partial\tau^{(k)}}\right) + \frac{\partial\mathbf{X}_u^T(j)}{\partial\tau^{(k)}}\mathbf{G}_x(N) + \mathbf{X}_u^T(j)\frac{\partial\mathbf{G}_x(N)}{\partial\tau^{(k)}}.$$

Prvo ćemo izračunati

$$\frac{\partial\mathbf{S}(j)}{\partial\tau^{(k)}} = \frac{\partial\mathbf{U}(j)}{\partial\tau^{(k)}}\mathbf{P}(j) + \mathbf{U}(j)\frac{\partial\mathbf{P}(j)}{\partial\tau^{(k)}}, \tag{6.13}$$

odnosno

$$\frac{\partial\mathbf{P}(j)}{\partial\tau^{(k)}} = \frac{\partial\mathbf{F}'_x(j+1)}{\partial\tau^{(k)}} + \frac{\partial\mathbf{X}(j+1)}{\partial\tau^{(k)}}\mathbf{P}(j+1) + \mathbf{X}(j+1)\frac{\partial\mathbf{P}(j+1)}{\partial\tau^{(k)}}. \tag{6.14}$$

Pošto su vrijednosti matrica $\mathbf{U}(j)$, $\mathbf{X}(j)$, koje sadrže parcijalne derivacije funkcije $\mathbf{f}(\mathbf{x}_j^{(k)}, \mathbf{u}_j^{(k)})$ po varijablama $\mathbf{u}_j^{(k)}, \mathbf{x}_j^{(k)}$, ovisne o $\tau^{(k)}$ preko izraza (5.2), na slijedeći način

$$\mathbf{U}(j) = \tau^{(k)}\Phi_u(j), \qquad \mathbf{X}(j) = \mathbf{I} + \tau^{(k)}\Phi_x(j),$$

gdje matrice $\Phi_u(j)$, $\Phi_x(j)$, analogno matricama $\mathbf{U}(j)$, $\mathbf{X}(j)$, sadrže parcijalne derivacije funkcije $\phi(\mathbf{x}_j^{(k)}, \mathbf{u}_j^{(k)})$ po varijablama $\mathbf{u}_j^{(k)}, \mathbf{x}_j^{(k)}$, što znači da nisu ovisne eksplicitno o $\tau^{(\ )}$, možemo napisati

$$\frac{\partial \mathbf{U}(j)}{\partial \tau^{(k)}} = \Phi_u(j) + \tau^{(k)} \frac{\partial \Phi_u(j)}{\partial \tau^{(k)}}. \tag{6.15}$$

Nadalje, imamo

$$\frac{\partial \Phi_u(j)}{\partial \tau^{(k)}} = \sum_{r=1}^{n} X_\tau^r(j)\, \Phi_{ux(r)}(j), \tag{6.16}$$

gdje je

$$\Phi_{ux(r)}(j) = \begin{bmatrix} \frac{\partial^2 \phi^1}{\partial u_j^1 \partial x_j^r} & \frac{\partial^2 \phi^2}{\partial u_j^1 \partial x_j^r} & \cdots & \frac{\partial^2 \phi^n}{\partial u_j^1 \partial x_j^r} \\ \frac{\partial^2 \phi^1}{\partial u_j^2 \partial x_j^r} & \frac{\partial^2 \phi^2}{\partial u_j^2 \partial x_j^r} & \cdots & \frac{\partial^2 \phi^n}{\partial u_j^2 \partial x_j^r} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 \phi^1}{\partial u_j^m \partial x_j^r} & \frac{\partial^2 \phi^2}{\partial u_j^m \partial x_j^r} & \cdots & \frac{\partial^2 \phi^n}{\partial u_j^m \partial x_j^r} \end{bmatrix}.$$

Isto tako, imamo

$$\frac{\partial \mathbf{X}(j+1)}{\partial \tau^{(k)}} = \Phi_x(j+1) + \tau^{(k)} \frac{\partial \Phi_x(j+1)}{\partial \tau^{(k)}}, \tag{6.17}$$

odnosno

$$\frac{\partial \Phi_x(j+1)}{\partial \tau^{(k)}} = \sum_{r=1}^{n} X_\tau^r(j+1)\, \Phi_{xx(r)}(j+1), \tag{6.18}$$

gdje je

$$\Phi_{xx(r)}(j) = \begin{bmatrix} \frac{\partial^2 \phi^1}{\partial x_j^1 \partial x_j^r} & \frac{\partial^2 \phi^2}{\partial x_j^1 \partial x_j^r} & \cdots & \frac{\partial^2 \phi^n}{\partial x_j^1 \partial x_j^r} \\ \frac{\partial^2 \phi^1}{\partial x_j^2 \partial x_j^r} & \frac{\partial^2 \phi^2}{\partial x_j^2 \partial x_j^r} & \cdots & \frac{\partial^2 \phi^n}{\partial x_j^2 \partial x_j^r} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 \phi^1}{\partial x_j^n \partial x_j^r} & \frac{\partial^2 \phi^2}{\partial x_j^n \partial x_j^r} & \cdots & \frac{\partial^2 \phi^n}{\partial x_j^n \partial x_j^r} \end{bmatrix}.$$

Nadalje, imamo

$$\frac{\partial \mathbf{F}_u'(j)}{\partial \tau^{(k)}} = \mathbf{F}_{ux}'(j) \cdot \mathbf{X}_\tau(j), \tag{6.19}$$

$$\frac{\partial \mathbf{F}_x'(j+1)}{\partial \tau^{(k)}} = \mathbf{F}_{xx}'(j+1) \cdot \mathbf{X}_\tau(j+1), \tag{6.20}$$

gdje je

$$\mathbf{F}'_{ux}(j)=\begin{bmatrix} \frac{\partial^2 F'}{\partial u_j^1 \partial x_j^1} & \frac{\partial^2 F'}{\partial u_j^1 \partial x_j^2} & \cdots & \frac{\partial^2 F'}{\partial u_j^1 \partial x_j^n} \\ \frac{\partial^2 F'}{\partial u_j^2 \partial x_j^1} & \frac{\partial^2 F'}{\partial u_j^2 \partial x_j^2} & \cdots & \frac{\partial^2 F'}{\partial u_j^2 \partial x_j^n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 F'}{\partial u_j^m \partial x_j^1} & \frac{\partial^2 F'}{\partial u_j^m \partial x_j^2} & \cdots & \frac{\partial^2 F'}{\partial u_j^m \partial x_j^n} \end{bmatrix},$$

$$\mathbf{F}'_{xx}(j)=\begin{bmatrix} \frac{\partial^2 F'}{\partial x_j^1 \partial x_j^1} & \frac{\partial^2 F'}{\partial x_j^1 \partial x_j^2} & \cdots & \frac{\partial^2 F'}{\partial x_j^1 \partial x_j^n} \\ \frac{\partial^2 F'}{\partial x_j^2 \partial x_j^1} & \frac{\partial^2 F'}{\partial x_j^2 \partial x_j^2} & \cdots & \frac{\partial^2 F'}{\partial x_j^2 \partial x_j^n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 F'}{\partial x_j^n \partial x_j^1} & \frac{\partial^2 F'}{\partial x_j^n \partial x_j^2} & \cdots & \frac{\partial^2 F'}{\partial x_j^n \partial x_j^n} \end{bmatrix}.$$

Na kraju, ostaje

$$\frac{\partial \mathbf{G}_x(N)}{\partial \tau^{(k)}}=\mathbf{G}_{xx}(N)\cdot\mathbf{X}_\tau(N), \tag{6.21}$$

$$\frac{\partial \mathbf{X}_u^T(j)}{\partial \tau^{(k)}}=\frac{\partial \mathbf{U}(j)}{\partial \tau^{(k)}}\mathbf{D}^T(j)+\mathbf{U}(j)\frac{\partial \mathbf{D}^T(j)}{\partial \tau^{(k)}}, \tag{6.22}$$

$$\frac{\partial \mathbf{D}^T(j)}{\partial \tau^{(k)}}=\frac{\partial \mathbf{X}(j+1)}{\partial \tau^{(k)}}\mathbf{D}^T(j+1)+\mathbf{X}(j+1)\frac{\partial \mathbf{D}^T(j+1)}{\partial \tau^{(k)}}, \tag{6.23}$$

gdje je

$$\mathbf{G}_{xx}(N)=\begin{bmatrix} \frac{\partial^2 G_B}{\partial x_N^1 \partial x_N^1} & \frac{\partial^2 G_B}{\partial x_N^1 \partial x_N^2} & \cdots & \frac{\partial^2 G_B}{\partial x_N^1 \partial x_N^n} \\ \frac{\partial^2 G_B}{\partial x_N^2 \partial x_N^1} & \frac{\partial^2 G_B}{\partial x_N^2 \partial x_N^2} & \cdots & \frac{\partial^2 G_B}{\partial x_N^2 \partial x_N^n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 G_B}{\partial x_N^n \partial x_N^1} & \frac{\partial^2 G_B}{\partial x_N^n \partial x_N^2} & \cdots & \frac{\partial^2 G_B}{\partial x_N^n \partial x_N^n} \end{bmatrix},$$

što je zbog oblika kaznene funkcije (4.49) jednako

$$\mathbf{G}_{xx}(N)=2K_B\mathbf{I}.$$

Ovim smo izračunali sve što nam je potrebno za konačan oblik algoritma:

$$\tau^{(k+1)}=\tau^{(k)}-\mu\frac{\partial J_T(\tau^{(k)})}{\partial \tau^{(k)}},$$

$$\frac{\partial J_T(\tau^{(k)})}{\partial \tau^{(k)}}=K_T+\frac{\partial G_V}{\partial \tau^{(k)}}+\frac{\partial G_B}{\partial \tau^{(k)}},$$

$$\frac{\partial G_B}{\partial \tau^{(k)}} = \mathbf{G}_x^T(N) \cdot \mathbf{X}_\tau(N),$$

$$\frac{\partial G_V}{\partial \tau^{(k)}} = \sum_{j=0}^{N-1} \mathbf{G}_{Vu}^T(j) \cdot \mathbf{U}_\tau(j),$$

$$\mathbf{U}_\tau(j) = -\eta \frac{\partial \mathbf{J}_u(j)}{\partial \tau^{(k)}},$$

$$\frac{\partial \mathbf{J}_u(j)}{\partial \tau^{(k)}} = \mathbf{F}'_u(j) + \mathbf{S}(j) + \tau^{(k)} \left(\frac{\partial \mathbf{F}'_u(j)}{\partial \tau^{(k)}} + \frac{\partial \mathbf{S}(j)}{\partial \tau^{(k)}} \right) + \frac{\partial \mathbf{X}_u^T(j)}{\partial \tau^{(k)}} \mathbf{G}_x(N) + \mathbf{X}_u^T(j) \frac{\partial \mathbf{G}_x(N)}{\partial \tau^{(k)}},$$

$$\frac{\partial \mathbf{F}'_u(j)}{\partial \tau^{(k)}} = \mathbf{F}'_{ux}(j) \cdot \mathbf{X}_\tau(j),$$

$$\mathbf{S}(j) = \mathbf{U}(j) \cdot \mathbf{P}(j),$$

$$\frac{\partial \mathbf{S}(j)}{\partial \tau^{(k)}} = \frac{\partial \mathbf{U}(j)}{\partial \tau^{(k)}} \mathbf{P}(j) + \mathbf{U}(j) \frac{\partial \mathbf{P}(j)}{\partial \tau^{(k)}},$$

$$\mathbf{P}(j) = \mathbf{F}'_x(j+1) + \mathbf{X}(j+1) \cdot \mathbf{P}(j+1),$$

$$\frac{\partial \mathbf{P}(j)}{\partial \tau^{(k)}} = \mathbf{F}'_{xx}(j+1) \cdot \mathbf{X}_\tau(j+1) + \frac{\partial \mathbf{X}(j+1)}{\partial \tau^{(k)}} \mathbf{P}(j+1) + \mathbf{X}(j+1) \frac{\partial \mathbf{P}(j+1)}{\partial \tau^{(k)}}$$

$$\frac{\partial \mathbf{G}_x(N)}{\partial \tau^{(k)}} = \mathbf{G}_{xx}(N) \cdot \mathbf{X}_\tau(N),$$

$$\mathbf{X}_u^T(j) = \mathbf{U}(j) \cdot \mathbf{D}^T(j),$$

$$\frac{\partial \mathbf{X}_u^T(j)}{\partial \tau^{(k)}} = \frac{\partial \mathbf{U}(j)}{\partial \tau^{(k)}} \mathbf{D}^T(j) + \mathbf{U}(j) \frac{\partial \mathbf{D}^T(j)}{\partial \tau^{(k)}},$$

$$\mathbf{D}^T(j) = \mathbf{X}(j+1) \cdot \mathbf{D}^T(j+1),$$

$$\frac{\partial \mathbf{D}^T(j)}{\partial \tau^{(k)}} = \frac{\partial \mathbf{X}(j+1)}{\partial \tau^{(k)}} \mathbf{D}^T(j+1) + \mathbf{X}(j+1) \frac{\partial \mathbf{D}^T(j+1)}{\partial \tau^{(k)}},$$

$$\frac{\partial \mathbf{X}(j+1)}{\partial \tau^{(k)}} = \Phi_x(j+1) + \tau^{(k)} \sum_{r=1}^{n} X_\tau^r(j+1) \Phi_{xx(r)}(j+1),$$

$$\frac{\partial \mathbf{U}(j)}{\partial \tau^{(k)}} = \Phi_u(j) + \tau^{(k)} \sum_{r=1}^{n} X_\tau^r(j) \Phi_{ux(r)}(j),$$

za $j = N-2,\ N-3,\ \ldots, 0$, te

$$\mathbf{X}_\tau(j) = \mathbf{X}_\tau(j-1) + \phi(j-1) + \tau^{(k)} \mathbf{X}^T(j-1) \cdot \mathbf{X}_\tau(j-1),$$

za $j = 1, 2, \ldots, N$ i $k = 0, 1, \ldots, M$.

Iz svega ovoga jasno vidimo da je dobiveni algoritam puno složeniji i glomazniji u odnosu na BPTT algoritam sa konstantnim korakom diskretizacije koji smo izveli u prošlom poglavlju, te stoga prilično nepraktičan za primjenu.

S druge strane, aproksimacijom derivacije,

$$\frac{\partial J_T(\tau^{(k)})}{\partial \tau^{(k)}} \approx \frac{J_T(\tau^{(k)}) - J_T(\tau^{(k-1)})}{\tau^{(k)} - \tau^{(k-1)}}, \tag{6.24}$$

dobivamo nestabilna rješenja zbog greške koja se javlja kod oduzimanja bliskih brojeva, što je posljedica računanja s konačnim brojem decimala.

Zbog činjenice da je egzaktno računanje derivacije prekomplicirano za praktičnu primjenu, a numeričko nestabilno, slijedi da problemu vremenski optimalnog upravljanja moramo prići na drugi način, što će biti prikazano u slijedećem podpoglavlju.

6.2 Primjena kaznenih funkcija za TOC

Kao što smo već spomenuli, za vremenski interval koji je veći od $t_{\min}$ biti će zadovoljena sva ograničenja (kaznene funkcije za ograničenja i rubne uvjete jednake su nuli), dok za slučaj $t < t_{\min}$ ograničenja neće biti zadovoljena (kaznene funkcije biti će veće od nule), što je prikazano na slici 6.1. Kod numeričkog izračunavanja to znači da će za $t > t_{\min}$ vrijednost sume kaznenih funkcija konvergirati prema nuli, dok će za $t < t_{\min}$, ta vrijednost konvergirati nekom pozitivnom broju. Što je vrijednost sume kaznenih funkcija za zadani $t > t_{\min}$ manja, to je rješenje točnije (bliže optimalnom).

Međutim, slika 6.1 odgovara vrijednostima kaznenih funkcija nakon određenog broja iteracija za svaki τ, dok se za TOC zahtijeva podešavanje $\tau^{(k)}$ istovremeno s izvršavanjem BPTT algoritma. Drugim riječima, ako na početku izvršavanja algoritma krenemo s nekom vrijednošću $\tau > \tau_{\min}$, vrijednosti kaznenih funkcija će biti velike, iako će s povećanjem broja iteracija težiti nuli. To je glavni izvor problema s konvergencijom i stabilnošću iterativnih metoda tipa $\tau^{(k+1)} = f\left(\tau^{(k)}\right)$, gdje je $f(\tau)$ neprekidna funkcija. Ti problemi se mogu izbjeći nekom vrstom "ireverzibilne" iterativne metode koju ćemo sada izložiti.

Prvo, definiramo funkciju J_G, koja je jednaka sumi kaznenih funkcija za ograničenja i rubne uvjete

$$J_G\left(\tau^{(k)}\right) = G_V(\mathbf{u}_0, \mathbf{u}_1, \ldots, \mathbf{u}_{N-1}) + G_B(\mathbf{x}_N), \tag{6.25}$$

gdje je G_V jednak izrazu (4.55). Minimalna vrijednost funkcije J jednaka je nuli u slučaju zadovoljenja svih ograničenja. Da bi izbjegli eksplicitnu ovisnost funkcije J o τ, za koeficijent kaznene funkcije K_V^*, uzimamo konstantnu vrijednost, neovisnu o τ.

Zatim, definiramo minimalnu vrijednost funkcije J koja je ujedno i mjera točnosti rješenja i označimo je sa $J_{G\min}$. Što je vrijednost $J_{G\min}$ manja, to je rješenje točnije, ali je također potreban veći broj iteracija da se dostigne ta vrijednost.

Sada možemo postaviti slijedeći općeniti algoritam

1. Inicijalizacija $\tau^{(0)}$, takvog da je $\tau^{(0)} > \tau_{\min}$,
2. *ako je* $J_G\left(\tau^{(k)}\right) < J_{G\min}$ *tada* $\tau^{(k+1)} < \tau^{(k)}$,
3. *ako je* $J_G\left(\tau^{(k)}\right) \geq J_{G\min}$ *tada* $\tau^{(k+1)} = \tau^{(k)}$,

za $k = 0,1,\ldots,M$. Pošto ne znamo vrijednost $\tau_{\min}$, početnu vrijednost τ odabiremo na osnovi procjene. U slučaju da stavimo $\tau^{(0)} < \tau_{\min}$, tada J tokom iteriranja neće dostići vrijednost $J_{G\min}$, što je znak da smo izabrali pogrešnu početnu vrijednost.

Drugim riječima, slijedeća iteracija perioda diskretizacije $\tau^{(k)}$, uslijedit će samo kada vrijednost funkcije J dostigne zadanu vrijednost $J_{G\min}$, a to znači da je $\tau^{(k)} > \tau_{\min}$. Sa smanjivanjem $\tau^{(k)}$, J će sve sporije konvergirati prema $J_{G\min}$, dok se ne dođe do vrijednosti $\tau^{(k)}$ za koju J neće moći dostići $J_{G\min}$ ili će mu jako sporo konvergirati, što znači da je $\tau^{(k)} < \tau_{\min}$. U tom slučaju, kao aproksimaciju za $\tau_{\min}$ uzimamo $\tau_{\min} \approx \tau^{(k-1)}$.

Ovakva struktura algoritma nam garantira stabilnost i konvergenciju prema $\tau_{\min}$, zbog toga što ne mjenja vrijednost $\tau^{(k)}$ sve dok vrijednost funkcije J ne padne ispod zadane, dovoljno male, vrijednosti $J_{G\min}$.

Drugi i treći korak navedenog algoritma možemo objediniti slijedećim izrazom

$$\tau^{(k+1)} = \tau^{(k)} - \varphi\left(\tau^{(k)}\right) H^{-}\left(J_G\left(\tau^{(k)}\right) - J_{G\min}\right). \tag{6.26}$$

Brzina konvergencije ovisit će o izboru funkcije $\varphi\left(\tau^{(k)}\right)$. Ako stavimo konstantu, $\varphi\left(\tau^{(k)}\right) = \Delta\tau$, dobivamo

$$\tau^{(k+1)} = \tau^{(k)} - \Delta\tau\, H^{-}\left(J_G\left(\tau^{(k)}\right) - J_{G\min}\right), \tag{6.27}$$

odnosno, svaki put kad je zadovoljeno ograničenje $J_G\left(\tau^{(k)}\right) < J_{G\min}$, τ se smanji za konstantnu vrijednost $\Delta\tau$.

Druga mogućnost je da uzmemo

$$\tau^{(k+1)} = \tau^{(k)} - \mu\left(J_{G\min} - J_G\left(\tau^{(k)}\right)\right) H^{-}\left(J_G\left(\tau^{(k)}\right) - J_{G\min}\right), \tag{6.28}$$

odnosno, promjena varijable τ proporcionalna je 'udaljenosti' funkcije J_G od zadane vrijednosti $J_{G\min}$. Gornjim izrazom omogućujemo kontinuiraniju promjenu varijable τ.

Ova dva pristupa usporediti ćemo međusobno na slijedećem primjeru.

Primjer 5.1 Razmotrimo problem optimalnog upravljanja za sustav drugog reda

$$\dot{x}_1 = x_2,$$

$$\dot{x}_2 = u,$$

sa ograničenjem

$$|u| \le 1.$$

Treba naći najkraće vrijeme, t_f, za prijelaz sustava iz početnog stanja

$$x_1(0) = 1, \qquad x_2(0) = 1,$$

u konačno

$$x_1(t_f) = 0, \qquad x_2(t_f) = 0.$$

Analitičko rješenje ovog problema je

$$x_1(t) = \begin{cases} -0.5t^2 + t + 1 & ; \ 0 \le t \le t_s \\ 0.5t^2 + A_1 t + A_2 & ; \ t_s \le t \le t_f \end{cases}, \qquad x_2(t) = \begin{cases} -t + 1 & ; \ 0 \le t \le t_s \\ t + A_1 & ; \ t_s \le t \le t_f \end{cases},$$

$$u(t) = \begin{cases} -1 & ; \ 0 \le t \le t_s \\ 1 & ; \ t_s \le t \le t_f \end{cases},$$

gdje je

$$t_s = 2.225, \qquad t_f = 3.45, \text{ odnosno } \tau_{\min} = 0.00345 \ (N = 1000),$$

$$A_1 = -2t_s + 1, \qquad A_2 = -t_s^2 + (1 - A_1)t_s + 1.$$

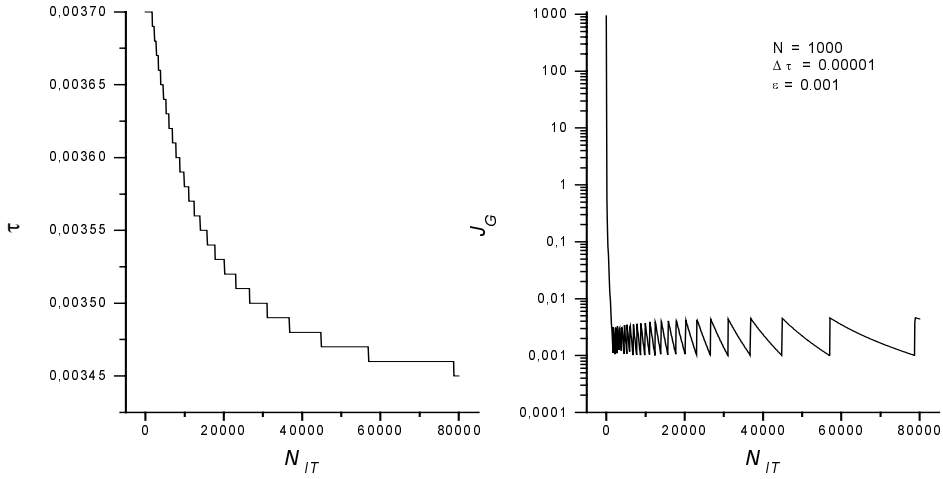
Na slici 6.2 prikazan je iterativni postupak za izračunavanje $\tau_{\min}$ prema izrazu (6.27) za $\varepsilon \equiv J_{G\min} = 0.001$ i $\Delta\tau = 0.00001$. Vidimo da se $\tau^{(k)}$ mijenja skokovito za $\Delta\tau$ svaki put kad se J_G spusti ispod $\varepsilon \equiv J_{G\min}$. Kada dođe do promjene τ, vrijednost funkcije J_G ponovo skoči iznad ε i proces se ponavlja. Što je manji $\Delta\tau$, to su manji skokovi za τ i J_G (slika 6.3). Također što je manja vrijednost ε, to su točniji rezultati, ali i sporija konvergencija.

Na slikama 6.4 i 6.5 prikazan je postupak izračunavanja $\tau_{\min}$ prema izrazu (6.28) za $\varepsilon \equiv J_{G\min} = 0.01$, te $\mu = 0.1$ i $\mu = 1.0$ respektivno. Na slici 6.4 za $N_{IT} = 80000$, dobili smo $t_{\min} = 3.45056$, a na slici 6.5 imamo $t_{\min} = 3.44981$. Vidimo da za manju vrijednost μ dobivamo kontinuiranu ovisnost τ o N_{IT}, ali i nešto sporiju konvergenciju.

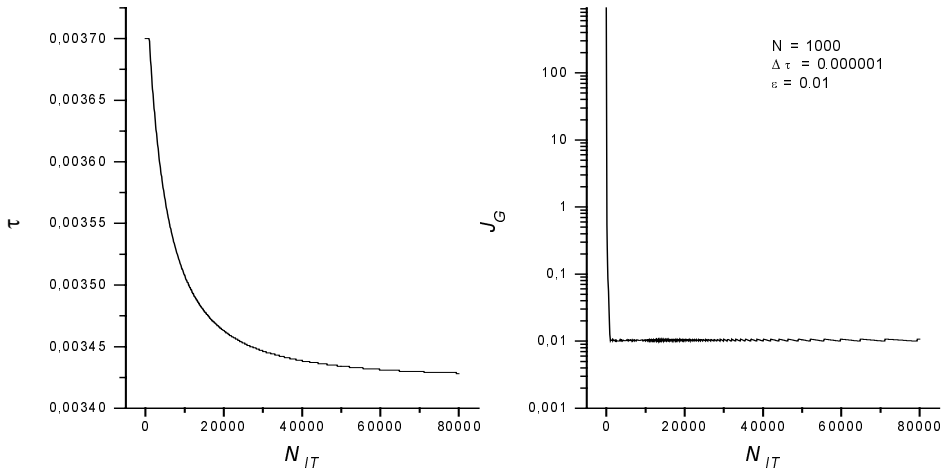
Slike 6.6 do 6.8 prikazuju vrijednosti varijabli stanja i upravljačkih varijabli za parametre prikazane na slici 6.5.

Prednost ovog algoritma, posebno (6.28), je očigledna u odnosu na algoritam opisan u prethodnom podpoglavlju, posebno zbog činjenice da algoritam ne zahtijeva računanje gradijenata i potpuno je neovisan o BPTT algoritmu kojim se računaju vrijednosti upravljačkih varijabli.

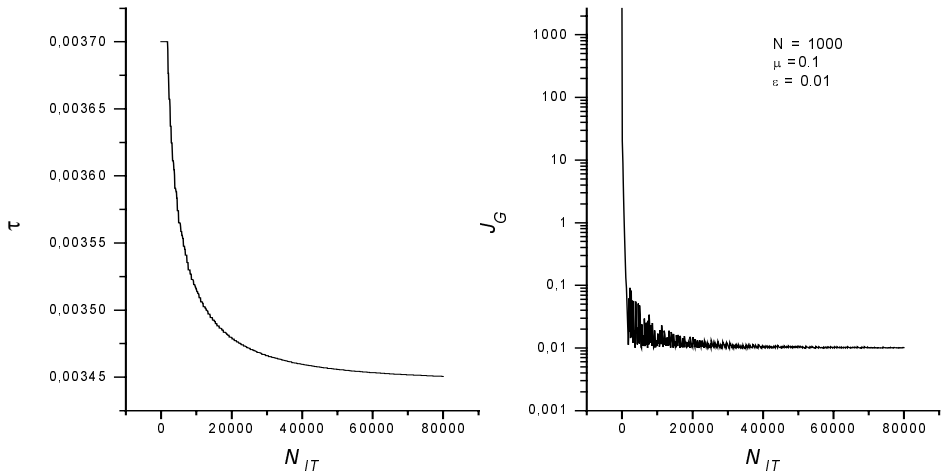
S druge strane, algoritam u obliku (6.27) ili (6.28) analogan je gradijentnom algoritmu i možemo ga lako priključiti strukturi neuronske mreže za BPTT algoritam. Time dobivamo jedinstvenu neuronsku mrežu za rješavanje problema optimalnog upravljanja, kao i vremenski optimalnog upravljanja.



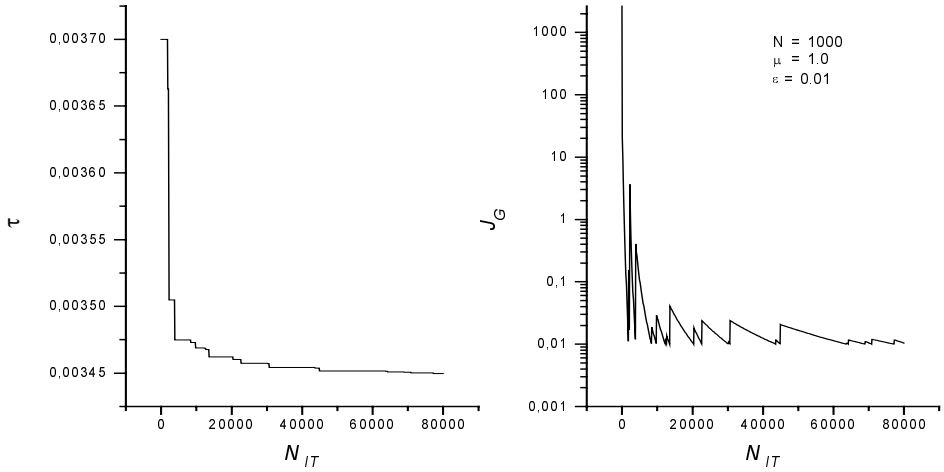
Slika 6.2 *Ovisnost* τ *i* J_G *o broju iteracija za (6.27).*



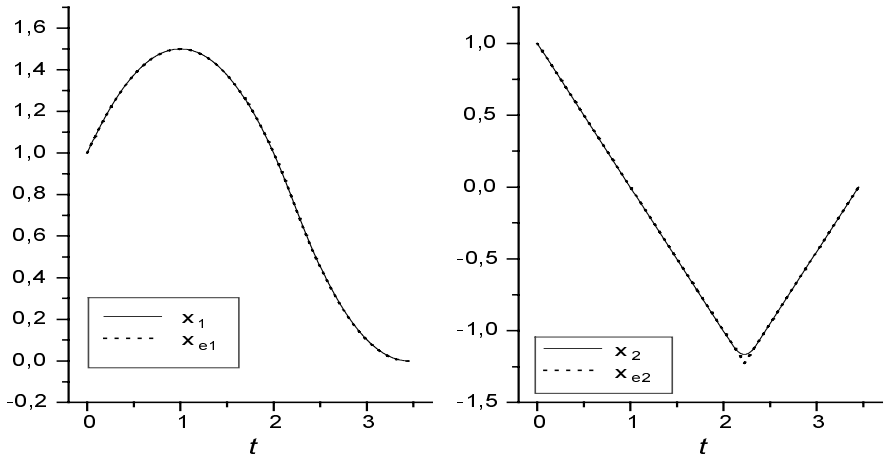
Slika 6.3 *Ovisnost* τ *i* J_G *o broju iteracija za (6.27).*



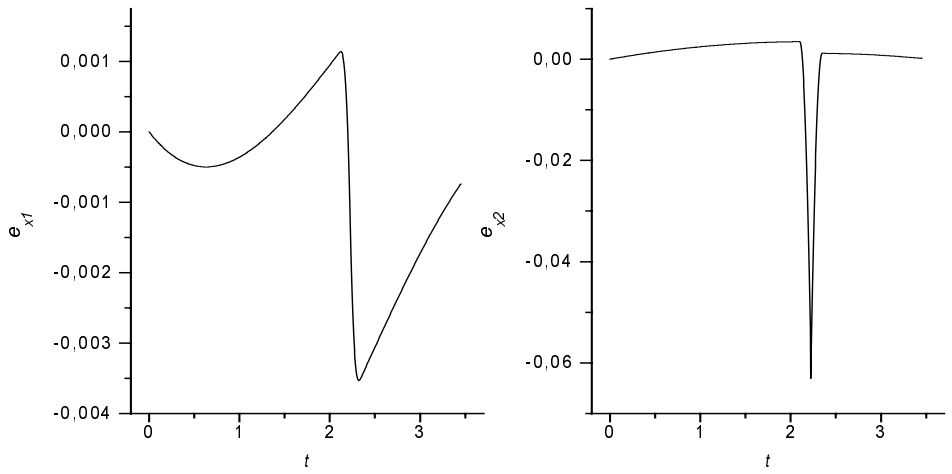
Slika 6.4 *Ovisnost* τ *i* J_G *o broju iteracija za (6.28).*



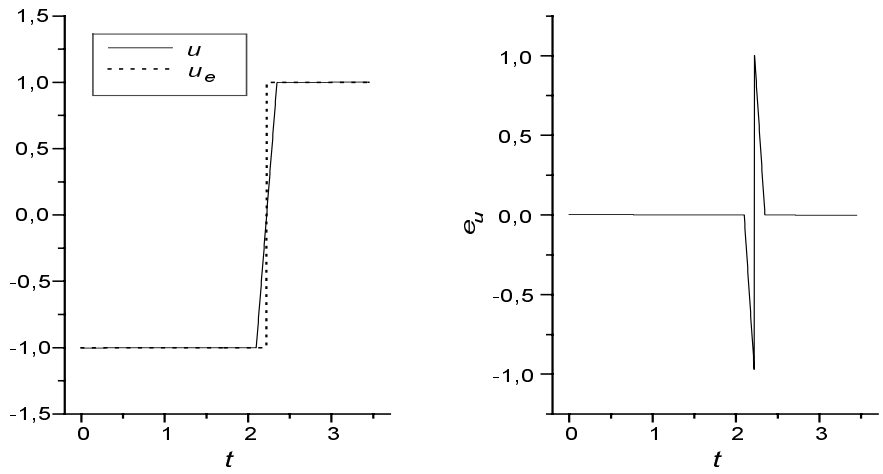
Slika 6.5 *Ovisnost* τ *i* J_G *o broju iteracija za (6.28).*



Slika 6.6 *Usporedba analitičkih rješenja (*x_{e1}, x_{e2}*) s numeričkim (*x_1, x_2*).*



Slika 6.7 *Odstupanja analitičkih rješenja od numeričkih* $\mathbf{e}_x = \mathbf{x}_e - \mathbf{x}$.



Slika 6.8 *Usporedba i međusobno odstupanje analitičkog rješenja* u_e, *od numeričkog* u.