Stochastické iterativní aproximace dynamického programování (SIDP)

Miroslav Zima

FJFI ČVUT

1. září 2010

Obsah

Systém

- Systém, jeho odezva na vstup u_t při realizaci šumu v_{t+1}

$$y_{t+1} = h_t(I_t, u_t, v_{t+1}, \theta), \qquad t = 0, \dots, N-1, \tag{1}$$

kde $I_t = (y_{t:0}, u_{t-1:0})$ a θ je neznámý parametr. Známe I_0, rozdělení šumu v_t a apriorní informaci $f(\theta | T_0)$.

Systém

- Systém, jeho odezva na vstup u_t při realizaci šumu v_{t+1}

$$y_{t+1} = h_t(I_t, u_t, v_{t+1}, \theta), \qquad t = 0, \ldots, N-1, \tag{1}$$

kde $I_t = (y_{t:0}, u_{t-1:0})$ a θ je neznámý parametr. Známe I_0, rozdělení šumu v_t a apriorní informaci $f(\theta | T_0)$.

- Bayesovské učení k získání aposteriorní hustoty $f(\theta | T_{t+1})$

$$f(\theta | T_{t+1}) = \frac{f(y_{t+1}|\theta, I_t, u_t) f(\theta | T_t)}{\int f(y_{t+1}|\theta, I_t, u_t) f(\theta | T_t) \mathrm{d}\theta}. \tag{2}$$

Systém

- Systém, jeho odezva na vstup u_t při realizaci šumu v_{t+1}

$$y_{t+1} = h_t(I_t, u_t, v_{t+1}, \theta), \qquad t = 0, \dots, N-1, \tag{1}$$

kde $I_t = (y_{t:0}, u_{t-1:0})$ a θ je neznámý parametr. Známe I_0, rozdělení šumu v_t a apriorní informaci $f(\theta | T_0)$.

- Bayesovské učení k získání aposteriorní hustoty $f(\theta | T_{t+1})$

$$f(\theta | T_{t+1}) = \frac{f(y_{t+1}|\theta, I_t, u_t) f(\theta | T_t)}{\int f(y_{t+1}|\theta, I_t, u_t) f(\theta | T_t) \mathrm{d}\theta}. \tag{2}$$

- Hyperstav $H_t = (I_t, T_t)$ platí

$$H_{t+1} = f_t(H_t, u_t, y_{t+1}), \qquad t = 0, \dots, N-1. \tag{3}$$

Formulace úlohy

- Ztrátová funkce (určuje kvalitu řízení)

$$g(y_{1:N}, u_{0:N-1}) = \sum_{t=0}^{N-1} g_t(y_{t+1}, u_t). \tag{4}$$

Formulace úlohy

- Ztrátová funkce (určuje kvalitu řízení)

$$g(y_{1:N}, u_{0:N-1}) = \sum_{t=0}^{N-1} g_t(y_{t+1}, u_t). \tag{4}$$

- Řídící strategie

$$\mu_t(H_t) = u_t, \qquad t = 0, 1, \ldots, N-1. \tag{5}$$

Formulace úlohy

- Ztrátová funkce (určuje kvalitu řízení)

$$g(y_{1:N}, u_{0:N-1}) = \sum_{t=0}^{N-1} g_t(y_{t+1}, u_t). \tag{4}$$

- Řídící strategie

$$\mu_t(H_t) = u_t, \qquad t = 0, 1, \dots, N-1. \tag{5}$$

- Hledáme $\pi = \mu_{0:N-1}$ minimalizující očekávanou ztrátu

$$J_\pi = \mathop{\mathrm{E}}_{\theta_0, v_{0:N-1}} \left\{ \sum_{t=0}^{N-1} g_t(y_{t+1}, \mu_t(H_t)) \right\}, \tag{6}$$

Použití dynamického programování při řešení úlohy řízení s aditivní ztrátou

- Princip optimality $\Longrightarrow$ postupně od konce řídícího horizontu minimalizujeme dílčí očekávané ztráty

$$J_N(H_N) = 0, \tag{7}$$

$$J_t(H_t) = \min_{u_t \in U_t} \mathop{\mathrm{E}}_{y_{t+1}} \left\{ g_t(y_{t+1}, u_t) + J_{t+1}(H_{t+1}) | H_t, u_t \right\}, \tag{8}$$

podmínky: $y_{t+1} = h_t(I_t, u_t, v_{t+1}, \theta)$ a $H_{t+1} = f_t(H_t, u_t, y_{t+1})$

Použití dynamického programování při řešení úlohy řízení s aditivní ztrátou

- Princip optimality $\implies$ postupně od konce řídícího horizontu minimalizujeme dílčí očekávané ztráty

$$J_N(H_N) = 0, \tag{7}$$

$$J_t(H_t) = \min_{u_t \in U_t} \mathsf{E}_{y_{t+1}} \left\{ g_t(y_{t+1}, u_t) + J_{t+1}(H_{t+1}) | H_t, u_t \right\}, \tag{8}$$

podmínky: $y_{t+1} = h_t(I_t, u_t, v_{t+1}, \theta)$ a $H_{t+1} = f_t(H_t, u_t, y_{t+1})$

- Problémy v řešení

Použití dynamického programování při řešení úlohy řízení s aditivní ztrátou

- Princip optimality $\Longrightarrow$ postupně od konce řídícího horizontu minimalizujeme dílčí očekávané ztráty

$$J_N(H_N) = 0, \tag{7}$$

$$J_t(H_t) = \min_{u_t \in U_t} \mathop{\mathsf{E}}_{y_{t+1}} \left\{ g_t(y_{t+1}, u_t) + J_{t+1}(H_{t+1}) | H_t, u_t \right\}, \tag{8}$$

podmínky: $y_{t+1} = h_t(I_t, u_t, v_{t+1}, \theta)$ a $H_{t+1} = f_t(H_t, u_t, y_{t+1})$

- Problémy v řešení
  - výpočet střední hodnoty

Použití dynamického programování při řešení úlohy řízení s aditivní ztrátou

- Princip optimality $\Longrightarrow$ postupně od konce řídícího horizontu minimalizujeme dílčí očekávané ztráty

$$J_N(H_N) = 0, \tag{7}$$

$$J_t(H_t) = \min_{u_t \in U_t} \mathop{\mathsf{E}}_{y_{t+1}} \left\{ g_t(y_{t+1}, u_t) + J_{t+1}(H_{t+1}) | H_t, u_t \right\}, \tag{8}$$

podmínky: $y_{t+1} = h_t(I_t, u_t, v_{t+1}, \theta)$ a $H_{t+1} = f_t(H_t, u_t, y_{t+1})$

- Problémy v řešení
  - výpočet střední hodnoty
  - minimalizace vzhledem k u_t

Použití dynamického programování při řešení úlohy řízení s aditivní ztrátou

- Princip optimality $\Longrightarrow$ postupně od konce řídícího horizontu minimalizujeme dílčí očekávané ztráty

$$J_N(H_N) = 0, \tag{7}$$

$$J_t(H_t) = \min_{u_t \in U_t} \mathop{\mathsf{E}}_{y_{t+1}} \left\{ g_t(y_{t+1}, u_t) + J_{t+1}(H_{t+1}) | H_t, u_t \right\}, \tag{8}$$

podmínky: $y_{t+1} = h_t(I_t, u_t, v_{t+1}, \theta)$ a $H_{t+1} = f_t(H_t, u_t, y_{t+1})$

- Problémy v řešení
  - výpočet střední hodnoty
  - minimalizace vzhledem k u_t
  - Analytické řešení obvykle nejde nalézt $\Longrightarrow$ aproximační metody.

SIDP

- Iterativní dynamické programování

SIDP

- Iterativní dynamické programování
  - Namísto přímého nalezení π konstruujeme $\pi_n \to \pi$

SIDP

- Iterativní dynamické programování
  - Namísto přímého nalezení π konstruujeme $\pi_n \to \pi$
- Stochastická aproximace = Metoda Monte Carlo

SIDP

- Iterativní dynamické programování
  - Namísto přímého nalezení π konstruujeme $\pi_n \to \pi$
- Stochastická aproximace = Metoda Monte Carlo
  - Očekávaná ztráta se aproximuje průměrem z n realizací ztráty

SIDP

- Iterativní dynamické programování
  - Namísto přímého nalezení π konstruujeme $\pi_n \rightarrow \pi$
- Stochastická aproximace = Metoda Monte Carlo
  - Očekávaná ztráta se aproximuje průměrem z n realizací ztráty
  - Generování realizace ztráty pro konkrétní bod H_t

SIDP

- Iterativní dynamické programování
  - Namísto přímého nalezení π konstruujeme $\pi_n \to \pi$
- Stochastická aproximace = Metoda Monte Carlo
  - Očekávaná ztráta se aproximuje průměrem z n realizací ztráty
  - Generování realizace ztráty pro konkrétní bod H_t
    1. generujeme v_t a θ (použijeme $f(\theta, T_t)$)

SIDP

- Iterativní dynamické programování
  - Namísto přímého nalezení π konstruujeme $\pi_n \to \pi$
- Stochastická aproximace = Metoda Monte Carlo
  - Očekávaná ztráta se aproximuje průměrem z n realizací ztráty
  - Generování realizace ztráty pro konkrétní bod H_t
    1. generujeme v_t a θ (použijeme $f(\theta, T_t)$)
    2. aplikací řídícího zásahu získáme $y_{t+1} = h_t(I_t, u_t, v_{t+1}, \theta)$

SIDP

- Iterativní dynamické programování
  - Namísto přímého nalezení π konstruujeme $\pi_n \to \pi$
- Stochastická aproximace = Metoda Monte Carlo
  - Očekávaná ztráta se aproximuje průměrem z n realizací ztráty
  - Generování realizace ztráty pro konkrétní bod H_t
    1. generujeme v_t a θ (použijeme $f(\theta, T_t)$)
    2. aplikací řídícího zásahu získáme $y_{t+1} = h_t(I_t, u_t, v_{t+1}, \theta)$
    3. k dosavadní ztrátě přičteme $g_t(y_{t+1}, u_t)$

SIDP

- Iterativní dynamické programování
  - Namísto přímého nalezení π konstruujeme $\pi_n \to \pi$
- Stochastická aproximace = Metoda Monte Carlo
  - Očekávaná ztráta se aproximuje průměrem z n realizací ztráty
  - Generování realizace ztráty pro konkrétní bod H_t

1. generujeme v_t a θ (použijeme $f(\theta, T_t)$)
2. aplikací řídícího zásahu získáme $y_{t+1} = h_t(I_t, u_t, v_{t+1}, \theta)$
3. k dosavadní ztrátě přičteme $g_t(y_{t+1}, u_t)$
4. dopočítáme $H_{t+1} = f_t(H_t, u_t, y_{t+1})$

SIDP

- Iterativní dynamické programování
  - Namísto přímého nalezení π konstruujeme $\pi_n \to \pi$
- Stochastická aproximace = Metoda Monte Carlo
  - Očekávaná ztráta se aproximuje průměrem z n realizací ztráty
  - Generování realizace ztráty pro konkrétní bod H_t

1. generujeme v_t a θ (použijeme $f(\theta, T_t)$)
2. aplikací řídícího zásahu získáme $y_{t+1} = h_t(I_t, u_t, v_{t+1}, \theta)$
3. k dosavadní ztrátě přičteme $g_t(y_{t+1}, u_t)$
4. dopočítáme $H_{t+1} = f_t(H_t, u_t, y_{t+1})$
5. interpolací/extrapolací π_n určíme optimální zásah pro H_{t+1}

SIDP

- Iterativní dynamické programování
  - Namísto přímého nalezení π konstruujeme $\pi_n \to \pi$
- Stochastická aproximace = Metoda Monte Carlo
  - Očekávaná ztráta se aproximuje průměrem z n realizací ztráty
  - Generování realizace ztráty pro konkrétní bod H_t

1. generujeme v_t a θ (použijeme $f(\theta, T_t)$)
2. aplikací řídícího zásahu získáme $y_{t+1} = h_t(I_t, u_t, v_{t+1}, \theta)$
3. k dosavadní ztrátě přičteme $g_t(y_{t+1}, u_t)$
4. dopočítáme $H_{t+1} = f_t(H_t, u_t, y_{t+1})$
5. interpolací/extrapolací π_n určíme optimální zásah pro H_{t+1}
6. pokud $t < N - 1 \implies 1.$

Integrátor s neznámým ziskem

- Výstup systému

$$y_{t+1} = y_t + \theta u_t + v_{t+1}, \qquad t = 0, \dots, N-1, \qquad (9)$$

kde $\theta \neq 0$ neznáme, šum $v_{t+1} \sim N(0, 0.1)$ a $y_0 = 1$.

Integrátor s neznámým ziskem

- Výstup systému

$$y_{t+1} = y_t + \theta u_t + v_{t+1}, \qquad t = 0, \ldots, N-1, \tag{9}$$

kde $\theta \neq 0$ neznáme, šum $v_{t+1} \sim N(0, 0.1)$ a $y_0 = 1$.

- Ztrátová funkce je kvadratická

$$g(y_{1:N}, u_{0:N-1}) = \sum_{t=0}^{N-1} y_{t+1}^2. \tag{10}$$

Integrátor s neznámým ziskem

- Výstup systému

$$y_{t+1} = y_t + \theta u_t + v_{t+1}, \qquad t = 0, \dots, N-1, \tag{9}$$

kde $\theta \neq 0$ neznáme, šum $v_{t+1} \sim N(0, 0.1)$ a $y_0 = 1$.

- Ztrátová funkce je kvadratická

$$g(y_{1:N}, u_{0:N-1}) = \sum_{t=0}^{N-1} y_{t+1}^2. \tag{10}$$

- Předpoklad $\mathrm{Cov}(v_{t+1}, \theta_t) = 0$ a $T_0 = (\hat{\theta}_0, P_0) \implies$ lze získat konkrétní tvar rce $H_{t+1} = f_t(H_t, u_t, y_{t+1})$.

Integrátor s neznámým ziskem

- Výstup systému

$$y_{t+1} = y_t + \theta u_t + v_{t+1}, \qquad t = 0, \ldots, N-1, \tag{9}$$

kde $\theta \neq 0$ neznáme, šum $v_{t+1} \sim N(0, 0.1)$ a $y_0 = 1$.

- Ztrátová funkce je kvadratická

$$g(y_{1:N}, u_{0:N-1}) = \sum_{t=0}^{N-1} y_{t+1}^2. \tag{10}$$

- Předpoklad $\mathrm{Cov}(v_{t+1}, \theta_t) = 0$ a $T_0 = (\hat{\theta}_0, P_0) \implies$ lze získat konkrétní tvar rce $H_{t+1} = f_t(H_t, u_t, y_{t+1})$.
- Hyperstav systému H_t tvoří vektor $(y_t, \hat{\theta}_t, P_t)$ (lze ho vhodnou transformaci redukovat na dim=2).

Konkurenční metody

- Certainty equivalence (CEC)

Konkurenční metody

- Certainty equivalence (CEC)
  - pro návrh řídícího zásahu je brán bodový odhad $\hat{\theta}$

Konkurenční metody

- Certainty equivalence (CEC)
  - pro návrh řídícího zásahu je brán bodový odhad $\hat{\theta}$
  - je možné získat analytické vyjádření pro optimální řídící zásah

Konkurenční metody

- Certainty equivalence (CEC)
  - pro návrh řídícího zásahu je brán bodový odhad $\hat{\theta}$
  - je možné získat analytické vyjádření pro optimální řídící zásah
- Cautious control (CC)

Konkurenční metody

- Certainty equivalence (CEC)
  - pro návrh řídícího zásahu je brán bodový odhad $\hat{\theta}$
  - je možné získat analytické vyjádření pro optimální řídící zásah
- Cautious control (CC)
  - hledá se optimální řízení na horizontu délky $N = 1$

Konkurenční metody

- Certainty equivalence (CEC)
    - pro návrh řídícího zásahu je brán bodový odhad $\hat{\theta}$
    - je možné získat analytické vyjádření pro optimální řídící zásah
- Cautious control (CC)
    - hledá se optimální řízení na horizontu délky $N = 1$
    - je možné získat analytické vyjádření pro optimální řídící zásah

Konkurenční metody

- Certainty equivalence (CEC)
  - pro návrh řídícího zásahu je brán bodový odhad $\hat{\theta}$
  - je možné získat analytické vyjádření pro optimální řídící zásah
- Cautious control (CC)
  - hledá se optimální řízení na horizontu délky $N = 1$
  - je možné získat analytické vyjádření pro optimální řídící zásah
- Klasický numerický přístup k dynamickému programování (DP)

Konkurenční metody

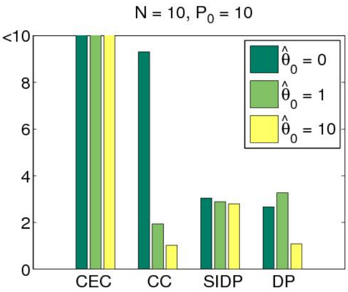
- Certainty equivalence (CEC)
  - pro návrh řídícího zásahu je brán bodový odhad $\hat{\theta}$
  - je možné získat analytické vyjádření pro optimální řídící zásah
- Cautious control (CC)
  - hledá se optimální řízení na horizontu délky $N = 1$
  - je možné získat analytické vyjádření pro optimální řídící zásah
- Klasický numerický přístup k dynamickému programování (DP)
  - Pro výpočet střední hodnoty je použita Simpsonova metoda

Konkurenční metody

- Certainty equivalence (CEC)
  - pro návrh řídícího zásahu je brán bodový odhad $\hat{\theta}$
  - je možné získat analytické vyjádření pro optimální řídící zásah
- Cautious control (CC)
  - hledá se optimální řízení na horizontu délky $N = 1$
  - je možné získat analytické vyjádření pro optimální řídící zásah
- Klasický numerický přístup k dynamickému programování (DP)
  - Pro výpočet střední hodnoty je použita Simpsonova metoda
  - K minimalizaci ztráty slouží jednoduchá interpolační metoda.

Kvantitativní srovnání

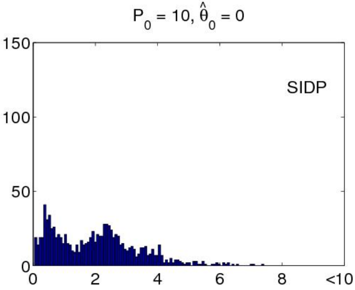
- Dosažená ztráta - průměr přes 1000 simulací
- θ = první nenulová realizace veličiny s rozdělením $N(\hat{\theta}_0, P_0)$



- Metoda CEC neposkytuje použitelné řízení
- Metoda CC funguje pouze při $\hat{\theta}_0 \neq 0$. Tehdy dosahuje nejnižší ztráty.
- SIDP a DP dobře řídí vždy. Při lepší počáteční identifikaci má SIDP horší výsledky (diskretizace).

Kvalitativní srovnání SIDP a DP

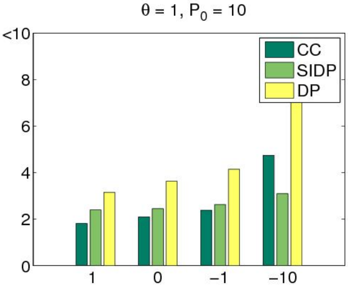
- Četnost konkrétních realizací ztráty - 1000 simulací.



- SIDP se vyhýbá výrazně špatnému řízení (> 10)
- DP častěji nabývá nižší celkové ztráty (< 1)
- SIDP navrhuje opatrnější zásahy $\Longrightarrow$ pomalejší identifikace θ

Test robustnosti

- Dosažená ztráta vhledem k apriorní informaci - 1000 simulací.
- $\theta =$ první nenulová realizace náhodné veličiny s rozdělením $N(1, 10)$, různá apriorní informace na $\hat{\theta}_0$, rozptyl $P_0 = 10$.



- Vyšší odolnost metody SIDP vůči špatné apriorní informaci
- SIDP poskytuje dobré řízení i pro $\hat{\theta}_0 = -10$, přitom skutečné $\theta = 1$

Závěr

- Předměty BP

Závěr

- Předměty BP
  - Úloha řízení za neurčitosti - teoretické řešení pomocí dynamického programování

Závěr

- Předměty BP
  - Úloha řízení za neurčitosti - teoretické řešení pomocí dynamického programování
  - Algoritmus SIDP jako možná aproximační metoda

Závěr

- Předměty BP
  - Úloha řízení za neurčitosti - teoretické řešení pomocí dynamického programování
  - Algoritmus SIDP jako možná aproximační metoda
  - Implementace SIDP pro řízení jednoduchého systému, porovnání s dalšími metodami

Závěr

- Předměty BP
    - Úloha řízení za neurčitosti - teoretické řešení pomocí dynamického programování
    - Algoritmus SIDP jako možná aproximační metoda
    - Implementace SIDP pro řízení jednoduchého systému, porovnání s dalšími metodami
- Možná další práce

Závěr

- Předměty BP
  - Úloha řízení za neurčitosti - teoretické řešení pomocí dynamického programování
  - Algoritmus SIDP jako možná aproximační metoda
  - Implementace SIDP pro řízení jednoduchého systému, porovnání s dalšími metodami
- Možná další práce
  - Navrhnout algoritmus, který by byl méně výpočetně náročný a stále poskytoval dobré řízení

Závěr

- Předměty BP
  - Úloha řízení za neurčitosti - teoretické řešení pomocí dynamického programování
  - Algoritmus SIDP jako možná aproximační metoda
  - Implementace SIDP pro řízení jednoduchého systému, porovnání s dalšími metodami
- Možná další práce
  - Navrhnout algoritmus, který by byl méně výpočetně náročný a stále poskytoval dobré řízení
  - např. $u_t = u_t^{(1)} + u_t^{(2)}$, kde $u_t^{(1)}$ minimalizuje ztrátu (například CC) a $u_t^{(2)}$ budí systém za účelem jeho lepší identifikace (použití SIDP).

Děkuji za pozornost