

1 Popis úlohy

Chceme řídit systém s výstupem y_t popsaným jako

$$y_0 = h_0(\theta, v_0), \quad y_{t+1} = h_t(I_t, \theta, u_t, v_{t+1}), \quad t = 0, \dots, N-1, \quad (1)$$

kde $I_t = (y_{0:t}, u_{0:t-1})$.

Označme T_t dostatečnou statistiku pro parametr θ založenou na informacích dostupných v čase t . Pokud dostatečná statistika neexistuje, pak bude T_t označovat nějakou její vhodnou aproximaci. Označme dále $H_t = (I_t, T_t)$ tzv. hyperstav systému.

Předpokládejme dále, že o parametru θ máme nějakou apriorní informaci v podobě hustoty pravděpodobnosti $f(\theta|T_0)$. Aposteriorní hustotu $f(\theta|T_{t+1})$ získáme pomocí Bayesova vzorce

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

Rekurzivní použití vzorce (2) pro odhad parametru θ se nazývá postup Bayesovského učení.

Pro vývoj hyperstavu H_t v čase můžeme na základě (2) psát

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

Ztrátová funkce je

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

Úlohou je nalezení řídicí strategie $\pi = \mu_{0:N-1}$, která by minimalizovala očekávanou ztrátu

$$J_\pi = \mathbb{E}_{\theta_0, v_{0:N-1}} \left\{ \sum_{t=0}^{N-1} g_t(y_{t+1}, \mu_t(H_t)) \right\}, \quad (5)$$

za apriorní informace $f(\theta|T_0)$, známého rozdělení šumu v_t a podmínek (3) a (1).

Úloha je řešitelná pomocí dynamického programování, tedy postupnou minimalizací očekávané ztráty od konce řídicího horizontu

$$J_N(H_N) = 0, \quad (6)$$
$$J_t(H_t) = \min_{u_t \in \bar{U}_t} \mathbb{E} \{g_t(y_{t+1}, u_t) + J_{t+1}(H_{t+1})|H_t, u_t\}, \quad t = 0, \dots, N-1, \quad (7)$$

kde H_{t+1} se počítá dle (3). Střední hodnota vzhledem k y_{t+1} se počítá pomocí (1) a $f(\theta|T_t)$ jakožto aktuálního odhadu na parametr θ .

Pro konkrétní systém však vlastní výpočet obvykle není možné provést. Důvodem je obtížný výpočet parametrického integrálu (parametr je u_t) a jeho následná minimalizace vzhledem k u_t . V další sekci je proto představen aproximační algoritmus k nalezení suboptimálního řízení.

2 Algoritmus SIDP se separovanou budící složkou (SIDPS)

Algoritmus je založený na SIDP, tj. na aproximaci dynamického programování pomocí iterování suboptimálního řízení a metody Monte Carlo.

Oproti klasickému SIDP specializujeme tvar hledaného řízení. To hledáme v tvaru

$$u_t(H_t) = u_t^{(1)}(H_t) + u_t^{(2)}(\tilde{H}_t), \quad t = 0, \dots, N-1. \quad (8)$$

kde $u_t^{(1)}$ je zásah starající se o minimalizaci ztráty a $u_t^{(2)}$ budí systém za účelem jeho lepší identifikace. Tvar $u_t^{(1)}$ předpokládáme za známý, předmětem optimalizace je $u_t^{(2)}$, který závisí pouze na $\tilde{H}_t \subset H_t$. Část H_t , na které $u_t^{(2)}$ nezávisí označíme $\tilde{H}_t^{(c)}$. Tedy je $H_t = \{\tilde{H}_t, \tilde{H}_t^{(c)}\}$.

Další změnou oproti SIDP je minimalizace ztrátové funkce na ubíhající horizontu délky \tilde{N} (obvykle volíme $\tilde{N} \ll N$). Namísto iterování řízení na mřížkách $H_{0:N-1}$ za účelem nalezení optimálního u_t pro každý bod H_t (případ SIDP) tak iterujeme pouze mřížku obsahující \tilde{H}_0 a v každém bodě optimalizujeme jen $u_0^{(2)}$. Optimální zásah pro j -tý bod diskretizovaného $H_0 (= H_{0,j})$ označíme $u_{0,j}$. Připomeňme, že optimalizace probíhá na ubíhající horizontu délky \tilde{N} . Vzhledem k tomu, že v každém časovém okamžiku používáme řízení napočtené na jediné mřížce \tilde{H}_0 , ztrácí se výhodnost lokální diskretizace hyperstavu (viz článek o SIDP). To vyplývá z předpokladu, že v průběhu řízení se bude neurčitost v systému vlivem procesu učení měnit (i výrazně). Bude tedy třeba širší spektrum hodnot \tilde{H}_0 pro které se bude hledat optimální řízení. V případě jediné mřížky bychom tak měli diskretizovat \tilde{H}_0 v celém rozsahu předpokládaných hodnot, pro které chceme systém řídit.

Schematickou podobu SIDPS zachycuje 1.

Algorithm 1 Schéma algoritmu SIDPS

diskretizuj \tilde{H}_0
v bodech diskretizace \tilde{H}_0 (je jich $|\tilde{H}_0|$) inicializuj π_1 (např. $\equiv 0$)
zvol libovolně $\tilde{H}_0^{(c)}$ (dle předpokladu na této volbě $u_0^{(2)}$ nezávisí)
for $i = 1$ to n_{iter} **do** {iteruj $\pi_i = \{u_{0,1:|\tilde{H}_0|}\}$
 for $j = 1$ to $|\tilde{H}_0|$ **do** {pro $\tilde{H}_{0,j}$ najdi (lokálně) nejlepší řízení}
 perturbací $u_{0,j}^{(2)}$ vygeneruj m kandidátů na řízení (ozn. $u_{1:m}$).
 for $k = 1$ to m **do** {spočti ztrátu J^* pro $\tilde{\pi}_i = \{\tilde{u}_{0,1:k-1}, u_k, u_{k+1:|\tilde{H}_0|}\}$
 for $l = 1$ to n_0 **do** { J^* aproximuj jako průměr z n_0 realizací $J^{(l)}$
 $J_0^{(l)} = 0$
 for $t = 0$ to $\tilde{N} - 1$ **do** {generuj realizaci ztráty J_l }
 generuj θ a v_t (použijeme $f(\theta, T_t)$ a známé rozdělení šumu)
 interpolací/extrapolací $\tilde{\pi}_i$ urči zásah $u_t^{(2)}$ pro \tilde{H}_t
 spočti $y_{t+1} = h_t(I_t, u_t, v_{t+1}, \theta)$
 přičti aditivní část ztráty, tj. $J_{t+1}^{(l)} = J_t^{(l)} + g_t(y_{t+1}, u_t)$
 dopočítej $H_{t+1} = f_t(H_t, u_t, y_{t+1})$
 end for
 $J^{(l)} = J_{\tilde{N}}^{(l)}$
 end for
 $\bar{J} = \frac{1}{n_0} \sum_{l=1}^{n_0} J_l$
 end for
 u_k s nejnižší hodnotou \bar{J} (ozn. $\tilde{u}_{0,j}$) uchovej jako řízení pro $\tilde{H}_{0,j}$.
 $\pi_i := \{\tilde{u}_{0,1:j-1}, \tilde{u}_{0,j}, u_{0,k+1:|\tilde{H}_0|}\}$
 end for
 $\pi_{i+1} := \{\tilde{u}_{0,1:|\tilde{H}_0|}\}$
 end for
 end for

3 Integrátor s neznámým ziskem

Výstup systému je popsán jako

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

$$v_{t+1} \sim N(0, \sigma^2), \quad (10)$$

kde $\theta \neq 0$ je neznámý parametr a rozptyl šumu σ^2 . Počáteční hodnota výstupu je nastavena na $y_0 = 1$ a optimální řízení je takové, pro které $y_t = 0$. Systém je možné v případě potřeby vhodně posunout či přeskálovat, není tedy potřeba uvažovat jiné hodnoty počátečního výstupu a referenčního signálu.

O neznámém parametru θ máme v čase t informaci v podobě dostatečné statistiky $T_t = (\hat{\theta}_t, P_t)$ tvořené střední hodnotou a rozptylem. Předpokládáme nekorelovanost odhadu $\hat{\theta}_t$ s šumem, tedy že

$$\text{Cov}(v_{t+1}, \hat{\theta}_t) = 0. \quad (11)$$

Ztrátová funkce je kvadratická v y_{t+1} , čili

$$g(y_{1:N}, u_{0:N-1}) = \sum_{t=0}^{N-1} y_{t+1}^2. \quad (12)$$

Odhadovací procedurou pro parametr θ je Kalmanův filtr. Pro systém (9) má tvar

$$K_t = \frac{u_t P_t}{u_t^2 P_t + \sigma^2}, \quad (13)$$

$$\hat{\theta}_{t+1} = \hat{\theta}_t + K_t (y_{t+1} - u_t \hat{\theta}_t), \quad (14)$$

$$P_{t+1} = (1 - K_t u_t) P_t. \quad (15)$$

Hyperstav systému H_t tvoří vektor $(y_t, \hat{\theta}_t, P_t)$. Očekávaná ztráta je

$$J_t(H_t) = \min_{u_t \in U_t} \mathbb{E} \{ y_{t+1}^2 + J_{t+1}(H_{t+1}) | H_t, u_t \}, \quad t = 0, \dots, N-1. \quad (16)$$

Ta po dosažení z (9) a částečném provedení střední hodnoty přejde na tvar

$$J_t(H_t) = \min_{u_t \in U_t} \left\{ (y_t + \hat{\theta}_t u_t)^2 + u_t^2 P_t + \sigma^2 + \mathbb{E}_{y_{t+1}} (J_{t+1}(H_{t+1})) | y_t, \hat{\theta}_t, P_t, u_t \right\}. \quad (17)$$

3.1 Transformace rovnic systému

Od proměnných $(y_t, \hat{\theta}_t, P_t, u_t)$ přejdeme k $(\eta_t, \beta_t, \zeta_t, \nu_t)$ dle vztahů

$$\eta_t = \frac{y_t}{\sigma}, \quad (18)$$

$$\beta_t = \frac{\hat{\theta}_t}{\sqrt{P_t}}, \quad (19)$$

$$\zeta_t = \frac{1}{\sqrt{P_t}}, \quad (20)$$

$$\nu_t = \frac{u_t \sqrt{P_t}}{\sigma}. \quad (21)$$

Současně můžeme neurčitost ve výstupu (9) reprezentovat jedinou normalizovanou náhodnou veličinou

$$s_t = \frac{y_{t+1} - y_t + \hat{\theta}_t u_t}{\sqrt{u_t^2 P_t + \sigma^2}} \sim N(0, 1). \quad (22)$$

Rovnice pro výstup (9) a následující odhad neznámého parametru (14) tak přejde v

$$\eta_{t+1} = \eta_t + \beta_t \nu_t + \sqrt{1 + \nu_t^2} s_t, \quad (23)$$

$$\beta_{t+1} = \sqrt{1 + \nu_t^2} \beta_t + \nu_t s_t. \quad (24)$$

Přejdeme-li k vhodně upravené očekávané ztrátě, dostaneme

$$V_t(\eta_t, \beta_t, \zeta_t) = \frac{J_t(y_t, \hat{\theta}_t, P_t)}{\sigma^2} \quad (25)$$

$$= \min_{\nu_t} \left\{ (\eta_t + \beta_t \nu_t)^2 + \nu_t^2 + 1 + \mathbb{E}_{s_t}(V_{t+1}(\eta_{t+1}, \beta_{t+1}, \zeta_{t+1})) \right\}. \quad (26)$$

Nyní spočteme očekávanou ztrátu pro $N - 1$ jako

$$V_{N-1}(\eta_{N-1}, \beta_{N-1}, \zeta_{N-1}) = \min_{\nu_{N-1}} \left\{ (\eta_{N-1} + \beta_{N-1} \nu_{N-1})^2 + \nu_{N-1}^2 + 1 \right\}. \quad (27)$$

Pomocí diferenciálního počtu pak získáme optimální zásah ve tvaru

$$\nu_{N-1} = -\frac{\eta_{N-1} \beta_{N-1}}{1 + \beta_{N-1}^2} \quad (28)$$

a očekávanou ztrátu rovnu

$$V_{N-1}(\eta_{N-1}, \beta_{N-1}, \zeta_{N-1}) = \frac{\eta_{N-1}^2 + 1}{\beta_{N-1}^2 + 1}. \quad (29)$$

Protože optimální zásah ν_{N-1} ani očekávaná ztráta V_{N-1} nezávisí na ζ_{N-1} , díky tvaru V_t nebude rovněž optimální zásah ν_t a očekávaná ztráta V_t záviset na ζ_t . K nalezení optimálního řízení tedy stačí v každém čase t uvažovat pouze dvourozměrný hyperstav $H_t = (\eta_t, \beta_t)$. Navíc můžeme bez újmy na obecnosti počítat optimální zásah pouze pro kladné hodnoty η_t a β_t . Optimální řízení tedy napočteme pouze pro kladné hodnoty y_t a θ_t . Pro ostatní možnosti pak díky tvaru (9) dostaneme požadovaný řídicí zásah vhodnou volbou znaménka napočteného zásahu.

3.2 Opatrné řízení

Metoda opatrného řízení spočívá v optimalizaci očekávané ztráty (7) na horizontu délky $N = 1$. Minimalizujeme tedy

$$J_0(H_0) = \min_{u_0 \in U_0} E \{g_0(y_1, u_0) | H_0, u_0\}. \quad (30)$$

Stačí proto spočítat střední hodnotu $g_0(y_1, u_0)$ vzhledem k y_1 a výsledek minimalizovat vzhledem k u_t . Poznamenejme, že výsledné řízení nebude zcela jistě duální. To plyne z toho, že minimalizujeme očekávanou ztrátu pouze jeden krok dopředu a tedy se nemůže projevit výhoda identifikace parametrů pomocí vybuzení systému mimo požadovaný stav.

Optimální řídicí zásah je pro metodu opatrného řízení dán dle (28) jako

$$\nu_t = -\frac{\eta_t \beta_t}{1 + \beta_t^2}. \quad (31)$$

Pro malé hodnoty β_t (tedy pro velké hodnoty neurčitosti v identifikaci parametru θ , viz (19)) bude řízení velmi konzervativní (odtud název). Pro $\beta_t = 0$ (to odpovídá $\hat{\theta}_t = 0$) pak bude $\nu_t = 0$ a regulátor tedy nebude poskytovat žádné aktivní řízení. Opatrné řízení je vhodným kandidátem na $u_t^{(1)}$ v 8.

3.3 Klasický přístup k dynamickému programování

Při použití klasického numerického přístupu k dynamickému programování je prostor hyperstavů diskretizován do mřížky (zde 64x64). Pro každý bod hyperstavu se napočítá očekávaná ztráta, mimo body mřížky se použije interpolace (zde kubická). K numerické integraci se použije klasická Simpsonova metoda a k nalezení minima pak jednoduchá interpolační metoda (zde se každými třemi body na mřížce proložila parabola, našlo její minimum a to se pak testovalo, zda-li je globálním minimem očekávané ztráty).

Optimální řízení na výsledné mřížce bylo nakonec parametrizováno analytickou formulí tvaru

$$\nu_t = -\frac{0,56 + \beta_t}{2,2 + 0,08\beta_t + \beta_t^2} \eta_t - \frac{1,9}{1,7 + \beta_t^4}. \quad (32)$$

délka ubíhajícího horizontu	\tilde{N}	10
počet iterací algoritmu	n_{iter}	4
počet bodů v diskretizaci každé dimenze H_t	n_g	10
počet kandidátů na změnu řídicího zásahu	m	7
počáteční rozsah pro hledání optimálního řídicího zásahu	β_0	5
parametr pro redukcí β_0 při opakování algoritmu	γ	0,2
počet realizací pro odhad metodou Monte Carlo	n	20

Tabulka 1: Konkrétní volba parametrů algoritmu SIDPS

Analytická aproximace způsobí zvýšení ve ztrátové funkci o méně než 1 %.

První člen v (32) můžeme interpretovat jako modifikované opatrné řízení, druhý člen pak jako budící složku řízení.

3.4 Implementace SIDPS pro jednoduchý systém

Budící složka v (32) závisí pouze na β . I to nás vede k tomu, že řízení budeme hledat ve tvaru

$$\nu_t = -\frac{\eta_t \beta_t}{1 + \beta_t^2} - \nu_t^{(2)}(\beta_t), \quad (33)$$

Volíme tedy $\tilde{H}_t = \beta_t$ a optimalizujeme $\nu^{(2)}$.

Dle popisu (a následné transformace) systému (9) je pro výpočet optimálního $\nu_t^{(2)}$ třeba diskretizovat (kladnou) část 1-dimenzionálního prostoru proměnné \tilde{H}_0 .

Implementace všech částí algoritmu byla provedena v souladu s oddílem 2. Konkrétní nastavení parametrů algoritmu zachycuje tabulka 1. Mimo body diskretizace byla použita lineární interpolace. Pro porovnání kandidátů na řízení byl navíc implementován RSSS algoritmus, což vedlo k drobnému zlepšení výsledného řízení (snížení ztráty o cca 1% a stabilnější kovergence π_i).

Otázkou zůstává, jak volit $\tilde{H}_0^{(c)} = \eta_0$. Ačkoliv předpokládáme, že $\nu_t^{(2)}$ na $\tilde{H}_0^{(c)}$ nezávisí, skutečnost může být jiná. V tom případě je zřejmě vhodné volit $\tilde{H}_0^{(c)}$ tak, aby co nejvíce odpovídala stavu, ve kterém se bude systém skutečně nacházet. Jako vhodný se proto jeví návrh $\eta_0 = 0$. Opodstatnění volby spočívá v tom, že v případě úspěšného řízení bude systém v okolí optimální (tedy nulové) hodnoty.

Optimální zásah $\nu_0^{(2)}$ byl počítán pro hodnoty

$$\tilde{H}_0 = \beta_0 = \{0.0001 + 0.75 \times j | j = 0, \dots, 9\}. \quad (34)$$

Výsledné řízení se pak dalo s úspěchem použít pro širokou škálu hodnot (b_0, P_0, y_0, σ) bez nutnosti opakovat vlastní výpočet. Veškeré experimenty tak byly prováděny

s řízením získaným při tomto jediném výpočtu. Poznamenejme, že v případě klasického SIDP by to z důvodu lokální povahy algoritmu nebylo možné.

Výpočet za daných parametrů trval v řádech jednotek vteřin (cca 5). Charakter konvergence π_i je diskutován dále.

4 Srovnání jednotlivých přístupů

4.1 Kvantitativní srovnání

Řízení bylo testováno vzhledem k různým hodnotám počátečního odhadu θ , tedy k různým hodnotám $(\hat{\theta}_0, P_0)$. Řídící horizont byl vždy $N = 100$, rozptyl šumu $\sigma = 1$. Každá simulace byla opakována 1000x, uvedené hodnoty celkové ztráty jsou průměrem přes jednotlivé realizace.

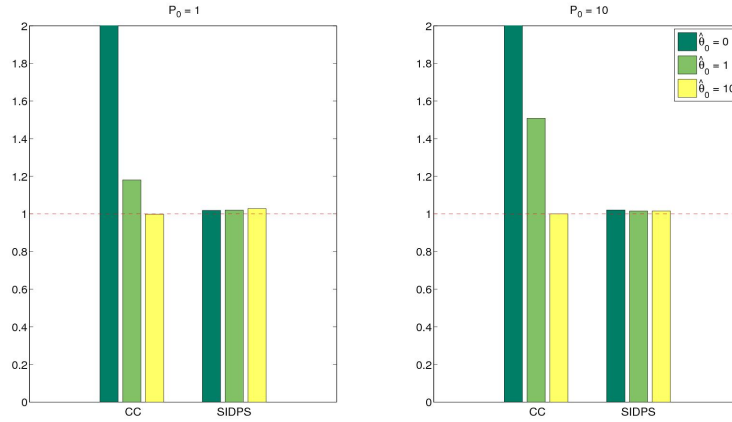
Pro každé jednotlivé opakování simulace byla skutečná hodnota parametru θ pro počáteční hodnoty $(\hat{\theta}_0, P_0)$ zvolena jako první nenulová realizace náhodné veličiny s rozdělením $N(\hat{\theta}_0, P_0)$. Tedy apriorní informace odpovídá skutečnosti. Pro takto vygenerovanou hodnotu θ byly postupně aplikovány jednotlivé řídicí algoritmy. Pro snížení vlivu náhodnosti při porovnání kvality řízení byly všechny realizace šumu (v rámci jednoho opakování simulace) v průběhu řízení jednotlivými algoritmy voleny stejně.

Na obrázku 1 jsou zachyceny výsledky pro různé hodnoty apriorní informace $(\hat{\theta}_0, P_0)$. Hodnoty ztráty jsou uvedeny relativně k průměrné ztrátě, které bylo ve stejné situaci dosaženo pomocí DP regulátoru. Dle obdržených výsledků tak v průměru nový algoritmus úspěšně konkuruje DP řízení, jehož výpočet je mnohonásobně složitější (dle informací v článku o SIDP je časová složitost cca $100000\times$ vyšší).

Pro ilustraci závislosti výsledků na rozptylu šumu σ a počátečním stavu y_0 byl zopakován výše uvedený pokus pro $\sigma = 10$ a poté ještě pro $y_0 = 100$. Bylo použito stejné řízení, jako prve. Výsledky byly velmi podobné těm na obrázku 1. Celkově vykazovalo řízení pomocí popsaného algoritmu v průměru srovnatelně kvalitní řízení jako DP regulátor a to pro všechny testované $(\hat{\theta}_0, P_0, y_0, \sigma)$ se zanedbatelnými výpočetními nároky.

4.2 Kvalitativní srovnání

Při kvalitativním srovnání byly zkoumány četnosti konkrétních realizací ztráty. Obrázky 2, 3, 4 udávají četnosti realizací relativních ztrát, kterých bylo nabyto při použití konkrétního regulátoru pro různé hodnoty $\hat{\theta}_0$ ($P_0 = 10$ a 1000 opakování). Referenční hodnotou byla průměrná ztráta DP regulátoru. Ostatní nastavení bylo stejné jako v případě kvantitativního srovnání. Uvedené histr-



Obrázek 1: Kvantitativní porovnání regulátorů pro různé hodnoty $(\hat{\theta}_0, P_0)$ - Na ose y jsou relativní hodnoty průměrné ztráty, které dosáhly uvedené algoritmy (osa x). Referenční hodnotou byly průměrné ztráty, kterých dosáhl DP regulátor pro tytéž parametry.

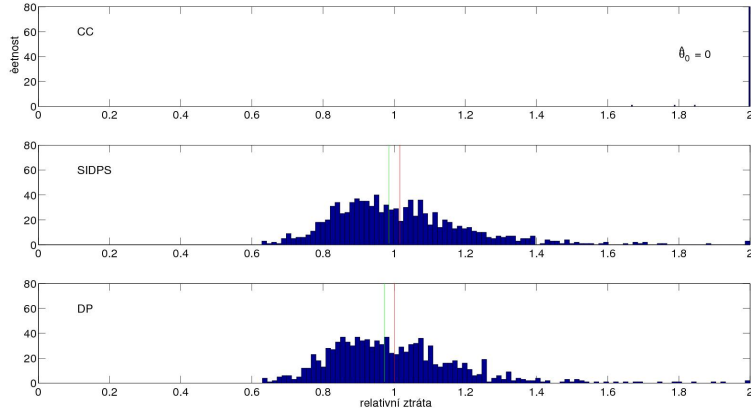
gramy dokumentují, že kvalitativně je řízení získané novým algoritmem velmi odobné jako řízení DP regulátorem.

5 Charakter konvergence SIDPS

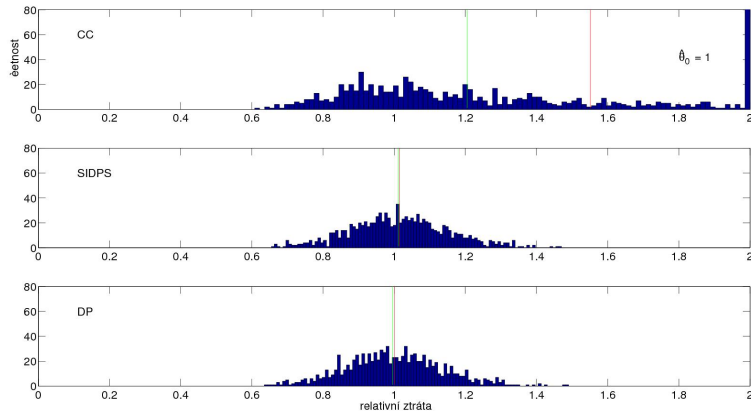
Obrázek 5 dává představu o charakteru konvergence π_i . První část obrázku zachycuje jednotlivé iterace π_i (hodnoty $u_0^{(2)}$ v závislosti na β). Druhá pak relativní průměrnou ztrátu při řízení za použití π_i (průměr přes 1000 opakování). Při simulacích bylo $(\hat{\theta}_0, P_0) = (1, 10)$, ostatní parametry zůstaly stejné jako v předešlé sekci. Referenční hodnotou byla opět průměrná ztráta DP regulátoru.

6 Závěr

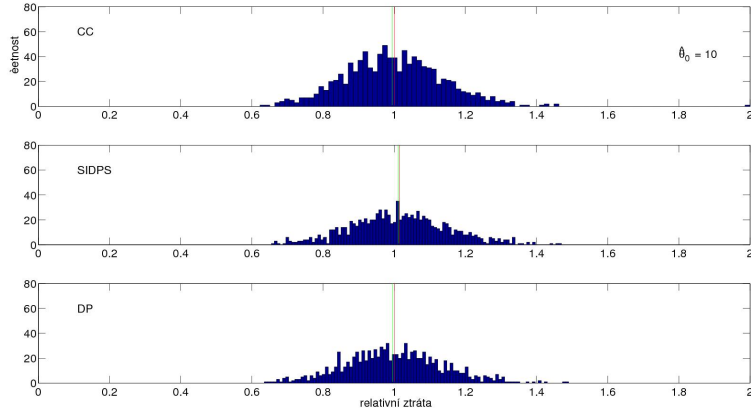
Na základě provedených pokusů se řízení získané pomocí SIDPS jeví jako velmi dobré. Dává srovnatelné výsledky s DP regulátorem, navíc v zanedbatelném čase. Nevýhodou algoritmu SIDPS je nutnost vybrat veličiny, které budou tvořit \tilde{H}_0 , předepsat $u_t^{(1)}$ a zvolit $\tilde{H}_0^{(c)}$. Dimenze \tilde{H}_0 má vliv zejména na náročnost výpočtu, $u_t^{(1)}$ a $\tilde{H}_0^{(c)}$ pak ovlivňují kvalitu výsledného řízení. Volba výše uvedených komponent algoritmu by měla vycházet z rozboru konkrétního systému, pro který má být algoritmus SIDPS implementován.



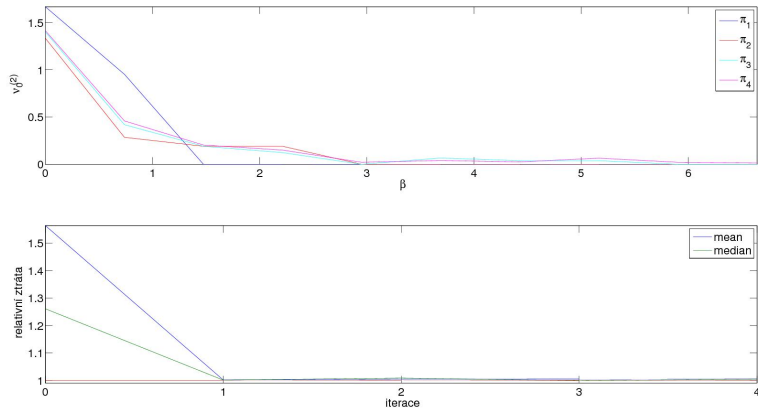
Obrázek 2: Kvalitativní porovnání regulátorů pro $\hat{\theta}_0 = 0$ - Na ose x jsou hodnoty relativní ztráty (referenční hodnotou je průměrná ztráta DP regulátoru), na osa y pak četnosti relativní ztráty z 1000 opakování. Zvýrazněna je hodnota průměrné ztráty (červená) a mediánu (zelená).



Obrázek 3: Kvalitativní porovnání regulátorů pro $\hat{\theta}_0 = 1$ - Na ose x jsou hodnoty relativní ztráty (referenční hodnotou je průměrná ztráta DP regulátoru), na osa y pak četnosti relativní ztráty z 1000 opakování. Zvýrazněna je hodnota průměrné ztráty (červená) a mediánu (zelená).



Obrázek 4: Kvalitativní porovnání regulátorů pro $\hat{\theta}_0 = 0$ - Na ose x jsou hodnoty relativní ztráty (referenční hodnotou je průměrná ztráta DP regulátoru), na osa y pak četnosti relativní ztráty z 1000 opakování. Zvýrazněny jsou pak hodnoty průměrné ztráty (červená) a mediánu (zelená).



Obrázek 5: Charakter konvergence SIDPS - Průběh π_i (první část) a relativní průměrná ztráta, které dosahuje (druhá část). Pro výpočet průměrné ztráty při řízení bylo $(\hat{\theta}_0, P_0) = (1, 10)$ a jako referenční hodnota byla zvolena průměrná ztráta DP regulátoru. Ostatní parametry simulace viz. sekce 4